

The $\text{\textit{S}\text{\textit{T}\text{\textit{E}\text{\textit{X}}}}}_3$ Package Collection*

Michael Kohlhase, Dennis Müller
FAU Erlangen-Nürnberg
<http://kwarc.info/>

2023-11-24

Contents

1	Introduction & Setup	7
1.1	What is $\text{\textit{S}\text{\textit{T}\text{\textit{E}\text{\textit{X}}}}}_3$?	7
1.2	The $\text{\textit{S}\text{\textit{T}\text{\textit{E}\text{\textit{X}}}}}_3$ package	8
1.3	What is MMT?	8
1.4	Math archives and the MathHub Directory	8
1.5	Setting Up the $\text{\textit{S}\text{\textit{T}\text{\textit{E}\text{\textit{X}}}}}_3$ IDE	9
I	Tutorial	11
2	The Basics	12
2.1	Text symbols	15
2.1.1	Using Modules & Search in the IDE	15
2.2	Symbol References	18
2.3	Modules and Simple Symbol Declarations	21
2.4	Documenting Symbols	24
2.5	Sectioning and Reusing Document Fragments	26
2.6	Building and Exporting HTML	28
3	Mathematical Concepts	31
3.1	Simple Symbol Declarations	31
3.1.1	Semantic Macros and Notations	31
3.1.2	Types and Variables	34
3.1.3	Flexary Macros and Argument Modes	39
3.1.4	Precedences	41
3.1.5	Implicit Arguments	41
3.1.6	Finishing Equality	43
3.1.7	Variable Sequences	43
3.2	Statements	44
3.2.1	Definitions	44
	Semantic Macros in Text Mode	46

*Version 3.4.0 (last revised 2023-11-24)

Definientia	47
Using Symbols Without Semantic Macros and Exporting Code in Modules	48
3.2.2 Assertions	49
3.2.3 Proofs	52
3.3 Mathematical Structures	52
3.3.1 Declaring and Using Structures	52
Instantiating Structures	54
3.3.2 Extending Structures and Axioms	55
Conservative Extensions	56
3.3.3 Nesting Structures and \this	57
3.4 Complex Inheritance and Theory Morphisms	59
3.4.1 Glueing Structures Together	61
3.4.2 Realizations	63
4 Extensions for Education	65
4.1 Slides and Course Notes	65
4.2 Problems and Exercises	65
4.2.1 Background	65
4.2.2 The Package, Options, and Configuration	66
4.2.3 (Open) Problems	67
4.2.4 Structured Problems	68
4.2.5 Single/Multiple Choice Blocks	69
4.2.6 Filling-In Concrete Solutions	71
4.2.7 Answer Classes	72
4.2.8 Including Problems	72
4.2.9 Testing and Spacing	72
4.3 Homework Assignments and Exams	73
4.3.1 Introduction	73
4.3.2 Package Options	73
4.3.3 Assignments	73
4.3.4 Including Assignments	74
4.3.5 Typesetting Exams	74
II User Manual	76
5 Basics	77
5.1 Package and Class Options	77
5.2 Math Archives and the MathHub Directory	78
5.2.1 The Structure of Math Archives	79
5.2.2 MANIFEST.MF-Files	79
5.3 The lib-Directory	80
5.4 Basic Macros	81
6 Document Features	82
6.1 Document Fragments	82
6.2 Using and Referencing Document Fragments	83
6.3 Cross-Document References	83

7 Modules and Symbols	86
7.1 Modules	86
7.1.1 Signature Modules, Languages, and Multilinguality	87
7.2 Symbol Declarations	87
7.2.1 Returns	88
7.3 Referencing Symbols	88
7.4 Notations and Semantic Macros	90
7.4.1 Precedences and Bracketing	91
7.4.2 Notations for Argument Sequences	92
7.4.3 Semantic Macros	92
7.5 Simple Inheritance	93
7.6 Variables and Sequences	95
7.7 Structures	96
7.7.1 Semantic Macros for Structures	96
8 Statements	99
8.1 More on Definitions	100
8.2 More on Assertions	101
9 Customizing Typesetting	102
9.1 Highlighting Symbol References	102
9.2 Styling Environments and Macros	103
9.3 Custom CSS for Environments	104
10 Additional Packages	105
10.1 NotesSlides Manual	105
10.1.1 Introduction	105
10.1.2 Package Options	105
10.1.3 Notes and Slides	106
10.1.4 Customizing Header and Footer Lines	107
10.1.5 Frame Images	107
10.1.6 Ending Documents Prematurely	108
10.1.7 Global Document Variables	108
10.1.8 Excursions	109
10.2 Problem Manual	110
10.2.1 Introduction	110
10.2.2 Problems and Solutions	110
10.2.3 Markup for Added-Value Services	112
Multiple Choice Blocks	112
Filling-In Concrete Solutions	113
10.2.4 Including Problems	114
10.2.5 Testing and Spacing	115
10.3 HWExam Manual	115
10.3.1 Introduction	115
10.3.2 Package Options	115
10.3.3 Assignments	116
10.3.4 Including Assignments	116
10.3.5 Typesetting Exams	116
10.4 Tikzinput Manual	117

III Documentation	119
11 S^TE_X Developer Manual	120
11.1 Documents	121
11.2 Modules	121
11.3 Symbols	123
11.4 Notations	125
11.5 Structural Features	128
11.6 Imports and Morphisms	130
11.7 Expressions and Semantic Macros	131
11.8 Optional (Key-Value) Argument Handling	132
11.9 Stylistable Commands and Environments	133
11.10 Math Archives	134
11.11 SMS-Mode	135
11.11.1 Second Pass	135
11.11.2 First Pass	136
11.12 Strings, File Paths, URIs	136
11.12.1 File Paths	137
File Path Constants and Variables	138
11.12.2 URIs	138
URI Constants and Variables	139
11.13 Language Handling	139
11.14 Inserting Annotations	140
11.14.1 Backend macros	140
11.15 Persisting Content from Math Archives in sms-Files	141
11.16 Utility Methods	141
11.16.1 Group-like Behaviours	142
12 Additional Packages	144
12.1 NotesSlides Documentation	144
12.2 Problem Documentation	144
12.3 HWExam Documentation	144
12.4 Tikzinput Documentation	144
IV Implementation	145
13 The S^TE_X Implementation	146
13.1 Setting up	146
13.2 Utilities	147
13.2.1 Calling kpsewhich and Environment Variables	147
13.2.2 Logging	148
13.2.3 Files and File Paths	148
13.2.4 Languages	151
13.2.5 Group-like Behaviours	153
13.2.6 HTML Annotations	155
13.2.7 Auxiliary Methods	156
13.2.8 Persistence	162
13.2.9 Files, Paths and URIs	164
13.2.10 File Hooks	168

13.3	Math Archives	170
13.4	Documents	174
13.4.1	Title	174
13.4.2	Sectioning	175
13.4.3	References	179
13.4.4	Inputs	186
13.5	SMS Mode	192
13.6	Modules	196
13.6.1	The <code>smodule</code> -environment	196
13.6.2	Structural Features	207
13.7	Inheritance	212
13.7.1	<code>\importmodule</code> / <code>\usemodule</code>	212
13.7.2	Theory Morphisms	218
13.8	Symbols	224
13.8.1	Declarations	224
13.8.2	Notations	233
13.8.3	a/B-mode argument handling	245
13.8.4	Variables	246
13.8.4	Sequences	250
13.8.5	Expressions	254
	Invoking Semantic Macros	255
	Argument Handling and Annotating	262
	Term HTML Annotations	266
	Automated Bracketing	269
	Symname and Variants	270
	Highlighting	272
13.9	Mathematical Structures	273
13.10	Statements	287
13.11	Proofs	293
13.12	Metatheory	301
13.13	MMT Interfaces	302
14	Additional Packages	307
14.1	Implementation: The <code>notesslides</code> Package	307
14.1.1	Class and Package Options	307
14.1.2	Notes and Slides	311
14.1.3	Environment and Macro Patches	314
14.1.4	Styling Across Notes/Slides	316
14.1.5	Beamer Compatibility	316
14.1.6	TODO Excursions	317
14.2	Implementation: The <code>problem</code> Package	318
14.2.1	Package Options	318
14.2.2	Problems and Solutions	319
14.3	Implementation: The <code>hwexam</code> Package	335
14.3.1	Package Options	335
14.3.2	Assignments	335
14.3.3	Leftovers	338
14.4	Tikzinput Implementation	339



`sTeX` is – by now – relatively stable and ready to use for the general public. However, it is also actively being developed further and subject to ongoing research. Some of the features described in here might not fully work as expected, some are still experimental, there might occasionally be cryptic error messages, and in general bugs are expected.

We welcome all kinds of issues you might encounter at <https://github.com/slatex/sTeX>.

If you have questions or problems with `sTeX`, you can talk to us directly at <https://matrix.to/#/#stex:fau.de>.

Chapter 1

Introduction & Setup

1.1 What is $\text{\textcolor{blue}{S}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}}$?

$\text{\textcolor{blue}{S}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}}$ is a system for generating human-oriented documents in either [PDF](#) or [HTML](#) format, augmented with computer-actionable semantic information (conceptually) based on the [OMDoc](#) format and ontology.

At its core is the [\textcolor{teal}{S}\textcolor{blue}{T}\textcolor{teal}{E}\textcolor{blue}{X} package](#) for [L\textcolor{blue}{A}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}](#), that allows for semantically marking up document fragments; in particular concepts, [formulae](#) and [mathematical](#) statements (such as definitions, theorems and proofs). Running `pdflatex` over $\text{\textcolor{blue}{S}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}}$ -annotated documents formats them into normal-looking [PDF](#).

But $\text{\textcolor{blue}{S}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}}$ also comes with a conversion pipeline into semantically annotated [HTML](#), which can host semantic added-value services that make the documents *active* (i.e. interactive and user-adaptive) – essentially turning [L\textcolor{blue}{A}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}](#) into a document format for (mathematical) knowledge management (MKM).

The $\text{\textcolor{blue}{S}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}}$ system consists of the following components:

- The [\textcolor{teal}{S}\textcolor{blue}{T}\textcolor{teal}{E}\textcolor{blue}{X} package](#),
- the [Rus\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}](#) system for converting [L\textcolor{blue}{A}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}](#) documents to (semantically enriched) [HTML](#),
- the [MMT](#) system for advanced knowledge management service and semantically informed backend for hosting the [HTML](#) with integrated added-value services,
- a collection of [math archives](#) of $\text{\textcolor{blue}{S}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}}$ document fragments and [symbols](#) for reuse, available of [mathhub.info](#), and
- the [\textcolor{teal}{S}\textcolor{blue}{T}\textcolor{teal}{E}\textcolor{blue}{X} IDE](#): A [VS Code](#) extension that, besides the usual [IDE](#) functionalities like syntax highlighting, integrates [Rus\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}](#) and [MMT](#) and allows for accessing the public [math archives](#) on [mathhub.info](#) to make the entire toolchain easily accessible to authors.

If [PDF](#) is all you are interested in, the [\textcolor{teal}{S}\textcolor{blue}{T}\textcolor{teal}{E}\textcolor{blue}{X} package](#) is all you *need*. Either way, however, we recommend using the [\textcolor{teal}{S}\textcolor{blue}{T}\textcolor{teal}{E}\textcolor{blue}{X} IDE](#) – it very much helps with semantically annotating your [L\textcolor{blue}{A}\textcolor{teal}{T}\textcolor{blue}{E}\textcolor{red}{X}](#) documents.

1.2 The **STEX** package

The **STEX** package extends **LATEX** with:

- A mechanism to declare **symbols** – concepts, functions, relations, variables, etc., which can be used and referenced in text or via **semantic macros** in mathematical formulae,
- a **module system** based on logical identifiers – **modules** bundle declarations, definitions, theorems, document snippets, and **symbols** for reuse, and
- an analogous organizational structure for developing documents modularly from individual fragments and sections.

The **STEX** package has been designed to have minimal impact on other **packages** and **document classes**, or the actual document layout – formatting of semantic **environments**, **symbol** references and **semantic macros** can be fully customized.

1.3 What is **MMT**?

MMT is a **software system** and **Scala** API for generic knowledge management services. It is based on a version of the **OMDOC** ontology and **document format** (**MMT/OMDoc**).

Among the services **MMT** provides are compiling, building, converting and managing libraries, a built-in web-server for browsing content, various algorithms for generic computation, checking and translating expressions, and querying.

1.4 Math archives and the MathHub Directory

To make the most of **STEX**, it is strongly encouraged to follow a workflow of small document fragments and **modules** to maximize reuse.

One considerable weakness of **LATEX** is the way source files are referenced: they need to be either in a **texmf** directory, or else be referenced via file paths relative to the main **.tex**-file being compiled. This is highly inconvenient if we want to collaboratively develop many highly interrelated document fragments.

STEX therefore adds an organizational layer on top of **LATEX**'s: **math archives** stored in a fixed **MathHub** directory anywhere on your hard drive. Referencing source files and **modules** is then done relative to the containing **math archive**, and is thus *independent* of user's individual setups or the current **.tex**-file.

The drawback of this approach is that **STEX** needs to know the location of your **MathHub** directory. There are multiple ways to achieve that, but the simplest and recommended approach is to set an environment variable: Simply create a new directory **<path>/MathHub** somewhere on your hard drive and set the environment variable **MATHHUB** as the path to this new directory.

Alternatively, you can let the **STEX IDE** do the work for you (see [section 1.5](#)).

For more on **math archives**, see [section 5.2](#).



Figure 1: Installing the [sTeX IDE](#)

1.5 Setting Up the [sTeX IDE](#)

[sTeX](#) is based on [L^AT_EX](#), and adds additional layers of presentational and functional markup to it. As a consequence the source files of [sTeX](#) documents look quite different from the resulting [XHTML](#) and [PDF](#) documents. Thus the best way of interacting with the [sTeX](#) document collections is via an [integrated development environment \(IDE\)](#). In this tutorial we will use the [sTeX](#) plugin for the [VS Code](#), which you should set up as a first step (this also sets up the necessary auxiliary software).

Setting up [sTeX](#) with the dedicated [IDE](#) is easy:

1. Download and install [VS Code](#) here: <https://code.visualstudio.com/download>
2. Start [VS Code](#) and navigate to the *Extensions*-tab on the left. Here you can search for Extensions in the [VS Code](#) marketplace. Look for the [sTeX](#) extension by [KWARC](#), as in [Figure 1](#) on the left.
3. Having done so, upon opening any folder in [VS Code](#) containing a `.tex`-file the setup window will pop up, as in [Figure 1](#) on the right.

The [IDE](#) will attempt to determine your [Java](#) installation and your [MathHub](#) directory (if set via an environment variable). Alternatively, you can set the latter now.

4. Download the [MMT](#) `.jar`-file at the link provided in the setup and select it. The [IDE](#) should then be able to determine your [MMT](#) version.

And that's it. Click on *Finish* and your setup is finished. The extension will start and download **RuSTEX** and some fundamental **math archives** for you automatically (an internet connection is required when finishing the setup).

Part I

Tutorial

The dynamic [HTML](#) version of this part can be found at

[https://stexmmt.mathhub.info/:
sTeX/fullhtml?archive=sTeX/Documentation&filepath=tutorial.en.xhtml](https://stexmmt.mathhub.info/sTeX/fullhtml?archive=sTeX/Documentation&filepath=tutorial.en.xhtml)

[sTeX](#) is a system for generating human-oriented documents in either [PDF](#) or [HTML](#) format, augmented with computer-actionable semantic information (conceptually) based on the [OMDoc](#) format and ontology.

In this part, we will give a broad but shallow introduction to [sTeX](#), and what you can get out of it. Additionally, this serves as an introduction to the [sTeX IDE](#), and we consequently assume that you have that one set up, as described in [section 1.5](#).

Note that in [PDFs](#), the specific highlighting of semantically annotated text is fully customizable (see [chapter 9](#)). In this document, we use [this highlighting](#) for [notation](#) components, [this highlighting](#) for [symbol](#) references, [this highlighting](#) for (local) [variables](#) and [this highlighting](#) for definienda; i.e. new concepts being introduced.

Chapter 2

The Basics

This document itself uses [sTeX](#) and serves as a direct example for the following. You can download its source files, the generated [PDF](#) files, and the generated [HTML](#) documents directly from within the [IDE](#), by navigating to the [sTeX](#) tab in the menu on the left and finding [sTeX/Documentation](#) in the list of [math archives](#) and clicking the small “Install”-button next to it, see the screenshot on the left of [Figure 2](#).

Once downloading is finished (this may take a while since dependencies are also downloaded), you can then browse the `.tex`-files in [sTeX/Documentation](#) directly from the [math archives](#) panel in the [sTeX](#) tab, as you can see in the right screenshot in [Figure 2](#).

For example, you can now navigate to the file `tutorial/intro.en` to see the sources of this very chapter.

As a first example, consider the following document fragment from [section 1.1](#):

[sTeX](#) is a system for generating human-oriented documents in either [PDF](#) or [HTML](#) format, augmented with computer-actionable semantic information (conceptually) based on the [OMDoc](#) format and ontology.

If you were to look at the generated [HTML](#) from this fragment, you could hover over the highlighted words ([sTeX](#), [PDF](#), [HTML](#), [OMDoc](#)) and get a little popup with their definitions ([Figure 3](#)). Neat, huh?

Here, in the [PDF](#), hovering will only show you a unique identifier ([MMT-URI](#)) for the word, and link to a definition on the web. Still useful, but not quite as neat, of course.

A plain [LaTeX](#)-version of the above document fragment, without any [sTeX](#) markup, could look like this:

Example 1

Input:

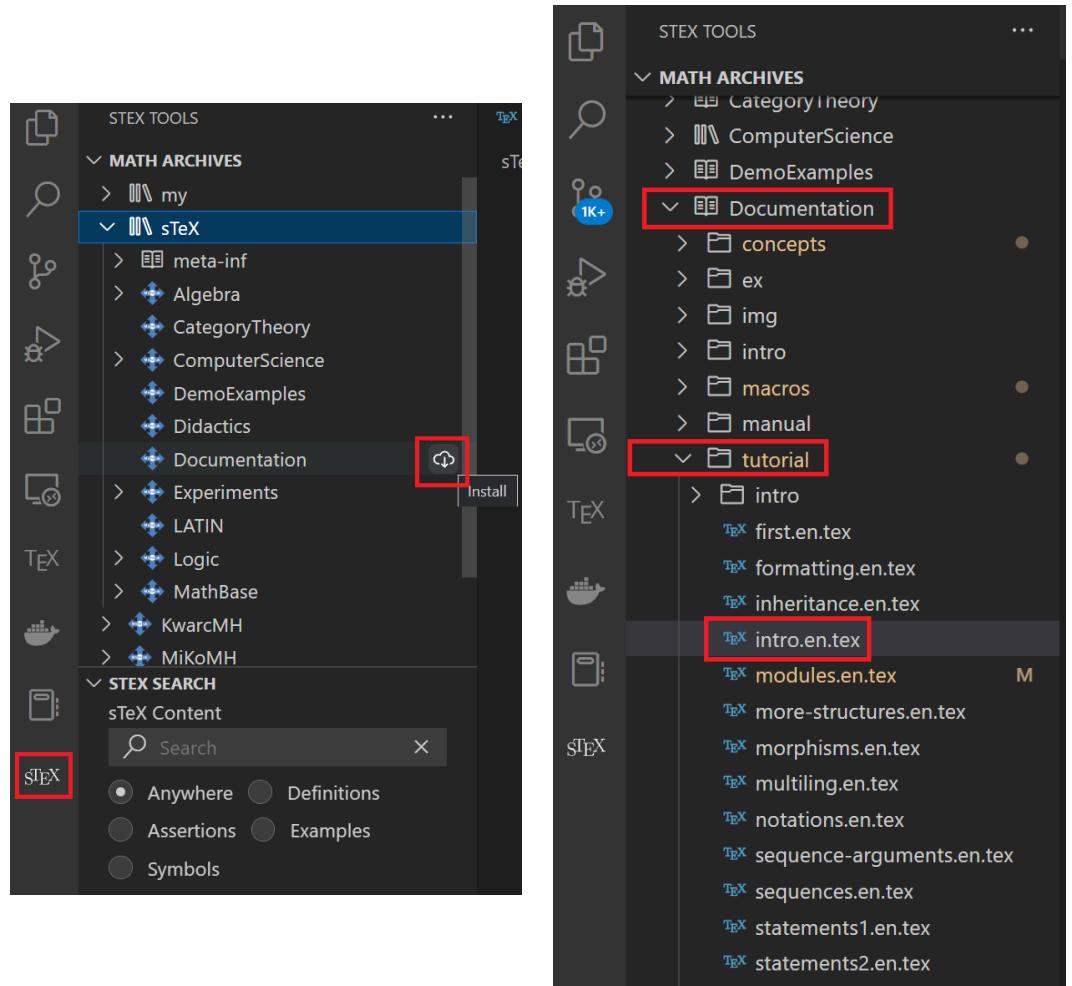


Figure 2: Installing Math Archives

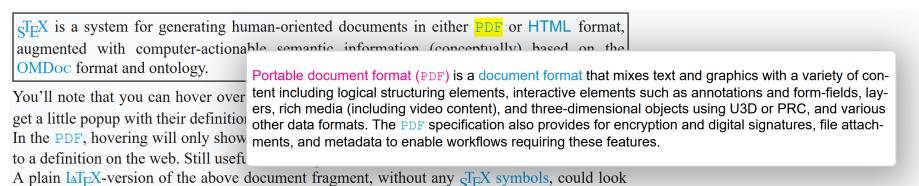


Figure 3: Definition on Hover

```

File [sTeX/Documentation]tutorial/intro/intro1plain.en.tex

1 \documentclass{article}
2 \usepackage{sTeX-logo}
3 \begin{document}
4
5   \sTeX{} is a system for generating human-oriented documents
6   in either \textsf{PDF} or \textsf{HTML} format, augmented
7   with computer-actionable semantic information (conceptually)
8   based on the \textsc{OMDoc} format and ontology.
9
10 \end{document}

```

Output:

sTeX is a system for generating human-oriented documents in either PDF or HTML format, augmented with computer-actionable semantic information (conceptually) based on the OMDoc format and ontology.

(Examples like the one above always show the file the source code is in, so if you have downloaded the sTeX/Documentation [math archive](#) you can toy around with it yourself)

If you save a file in the [IDE](#) (regardless of whether it has unsaved changes), a preview window will pop up, showing you the [HTML](#) generated from the .tex-file; see (Figure 4).



Figure 4: Previewing the document in the IDE

The \sTeX macro comes with the [stex package](#) as well, but if you only want to use the logo – e.g. to write eulogies about [sTeX](#) in plain [LaTeX](#) papers, you can load the much smaller [stex-logo package](#) instead.

2.1 Text symbols

The most central concept behind **s_TeX** is that of a *symbol*:

A **symbol** is a *named* concept that can be defined, documented and referenced. Examples for **symbols** are mathematical constants, functions, theorems, statements, principles – anything that has a (somewhat) precise meaning and can be referenced by name can be a **symbol**.

Before we explain how we can declare new **symbols** and associate them with definitions, **notations** and all that, let's assume an ideal world in which others have done that job already for us – after all, **s_TeX** is all about *reuse*, and naturally, there are **s_TeX symbols** for all of the above already. Let's start with the one for **s_TeX** itself:

2.1.1 Using Modules & Search in the IDE

In the **VS Code IDE**, navigate to the **s_TeX**-tab on the left. In the search panel, select the “**Symbols**” radio button and search for “**s_TeX**”. The second search result should be what we're looking for ([Figure 5](#)).

Search results are grouped into *local* and *remote* results. Local ones are the ones you already have in your local **MathHub** directory; remote ones you can download directly from within the **IDE**.

You can click the preview button to see the generated **HTML** for the document – the resulting window that pops up also has an **OMDoc** tab you can select, which (among other things) shows you the **semantic macros** provided by the respective **module**: In this case, it tells us that there is a **text symbol** named “**s_TeX**” with **semantic macro** `\stex` in the **module** `mod/systems/tex?sTeX` that is in the `\sTeX/ComputerScience/Software` archive. It produces the presentation “**s_TeX**” as we want ([Figure 6](#)).

A **text symbol** is a **symbol** `foo` with an associated **semantic macro** `\foo`. The **macro** `\foo` is allowed in text or math mode and produces a predefined piece of text output annotated with `foo`.

The variant `\fooname` produces the same output without annotation.

If we want to use the **s_TeX symbol** in a document – which we have open in the **IDE** – we simply click on the **use** button, and the **IDE** will automatically insert the line `\usemodule[sTeX/ComputerScience/Software]{mod/systems/tex?sTeX}`, making all **symbols** in that **module** available to use – in particular, we can now use the `\stex` **semantic macro** instead of the plain, non-semantic `\sTeX macro` – that is, of course, after we include the `stex` **package** first.

s_TeX The `\usemodule` macro takes as *optional* argument the name of a **math archive**, and as a regular argument the path to an **s_TeX module** (see [section 7.5](#)).

Analogously, we can also search for the **PDF**, **HTML** and **OMDoc** symbols, all of which are also **text symbols** and have the associated **semantic macros** `\PDF`, `\HTML` and `\omdoc`; the document should thus look like this:

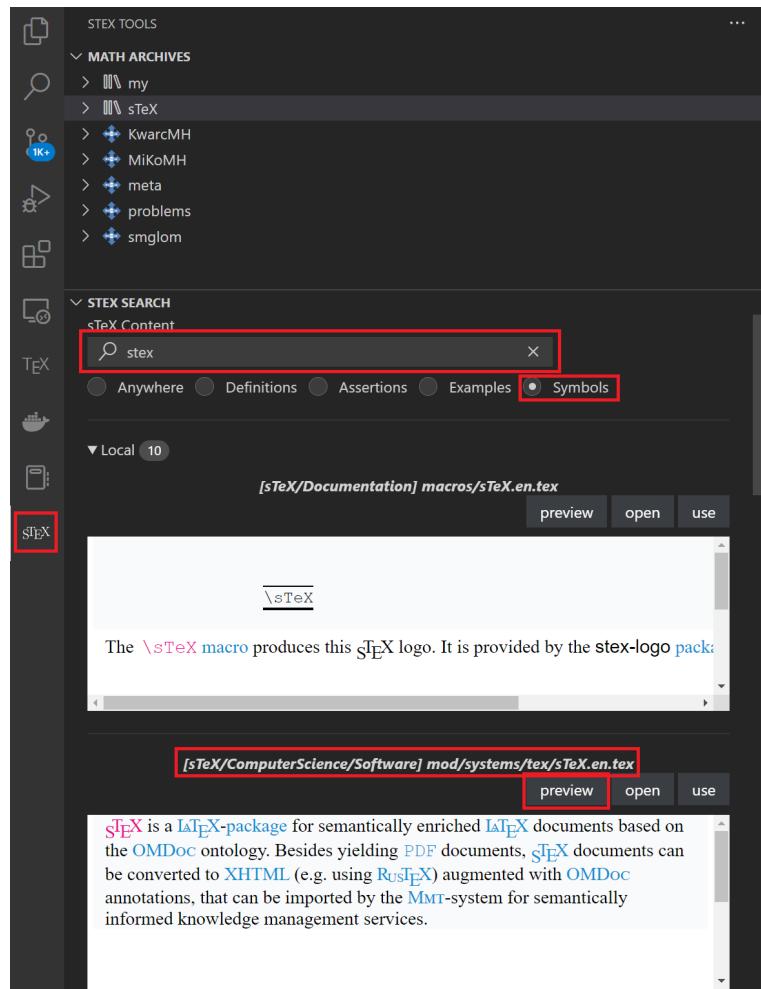


Figure 5: Search in the STeX IDE



Figure 6: OMDoc Preview

Example 2

Input:

```
File [sTeX/Documentation]tutorial/intro/intro1stex.en.tex
1 \documentclass{article}
2 \usepackage{sstex}
3 \begin{document}
4   \usemodule[sTeX/ComputerScience/Software]{mod/systems/tex?sTeX}
5   \usemodule[sTeX/ComputerScience/Software]{mod/formats?PDF}
6   \usemodule[sTeX/ComputerScience/Software]{mod/formats?HTML}
7   \usemodule[sTeX/ComputerScience/Software]{mod/formats?OMDoc}
8
9   \stex is a system for generating human-oriented documents
10  in either \PDF or \HTML format, augmented
11  with computer-actionable semantic information (conceptually)
12  based on the \omdoc format and ontology.
13 \end{document}
```

Output:

STeX is a system for generating human-oriented documents in either **PDF** or **HTML** format, augmented with computer-actionable semantic information (conceptually) based on the **OMDoc** format and ontology.

Now, our generated **HTML** looks a lot more interesting, with highlighting, pop-ups on hover and all that. Notably however, if we compile the file with `pdflatex`, it looks pretty much exactly as before.

That's because we haven't told **STeX** what to do with semantic annotations yet – and by default, it does not do anything fancy, except for wrapping them in an `\emph`. We can customize how we want **STeX** to highlight various semantic text fragments (see [chapter 9](#)). A default highlighting schema is provided in the `sstex-highlighting` package – including that will



Figure 7: Redundant Imports



Figure 8: Includes in the OMDoc Preview

- highlight semantically annotated text in [this color](#),
- show the MMT-URI of the corresponding [symbol](#) in a tooltip on hovering over the text,
- make the text link to the place the [symbol](#) is being defined in the current document (if it is), or, alternatively,
- make it link to an external resource, if one is known. In our case, they link to [stexmmmt.mathhub.info/:sTeX](#), where the [HTML](#) for all the [symbols](#) we use in this document are hosted.

Note that in the [IDE](#), the `\usemodule`-statement for [OMDoc](#) is underlined in blue ([Figure 7](#)) – [VS Code](#) is letting us know, that this `\usemodule` statement is *redundant*. That is because the [sTeX module](#) we imported earlier already imports the [OMDoc module](#); as such we have all [macros](#) therein available already. If we look at the [sTeX module](#) in the [VS Code](#) preview window again, we can see that ([Figure 8](#)).

We can consequently safely delete the `\usemodule` again.

2.2 Symbol References

Let's continue with the next paragraph of [section 1.1](#); for now unannotated:

Example 3

Input:

```

File [sTeX/Documentation]tutorial/intro/intro2plain.en.tex

1 \documentclass{article}
2 \usepackage{sTeX}
3 \begin{document}
4
5 At its core is the \sTeX{} package for \LaTeX{}, that allows for
6 semantically marking up document fragments; in particular
7 concepts, formulae and mathematical statements (such as
8 definitions, theorems and proofs). Running \texttt{pdflatex}
9 over \sTeX-annotated documents formats them into normal-looking
10 \textsf{PDF}.
11
12 \end{document}

```

Output:

At its core is the **sTeX** package for **L^AT_EX**, that allows for semantically marking up document fragments; in particular concepts, formulae and mathematical statements (such as definitions, theorems and proofs). Running **pdflatex** over **sTeX**-annotated documents formats them into normal-looking **PDF**.

We already know how to annotate “**sTeX**” and “**PDF**”; and if we use the search field in the **IDE** again, we can also find a **text symbol** for “**L^AT_EX**”. But if we look at the documentation, we will note that *more* is highlighted:

At its core is the **sTeX** package for **L^AT_EX**, that allows for semantically marking up document fragments; in particular concepts, **formulae** and **mathematical** statements (such as definitions, theorems and proofs). Running **pdflatex** over **sTeX**-annotated documents formats them into normal-looking **PDF**.

The “**package**”-symbol can be found in the **L^AT_EX** module too, and searching for the keywords “**formula**” and “**mathematics**” will yield the symbols “**well-formed formula**” and “**mathematics**”, but they are not **text symbols** and “**mathematics**” and “**package**” do not even have a **semantic macro** – and the one for “**well-formed formula**” would not work outside of math mode.

Text symbols are special in that way – they are intended for **symbols** that have a specific formatting associated (such as **L^AT_EX**, **OMDOC**, or **HTML**, which we prefer to typeset as sans serif). For those settings, it makes sense to associate that formatting with a **semantic macro** that does the typesetting for us.

Symbols without a **text macro** can be referenced with the **\symname** macro: **\symname{package}** prints the *name* of the “**package**”-symbol and annotates it accordingly, without any special formatting – in particular it is compatible with being in **\emph**, **\textbf** and similar **macros**. That takes care of *one* of the missing annotations.

More generally, the **\symref** macro can be used to annotate arbitrary text with a symbol: **\symref{mathematics}{mathematical}** associates the text **mathematical** with the symbol “**mathematics**”; thus, we get “**mathematical**” and similarly “**formulae**”.

sTeX

In general, any **macro** that expects a **symbol** name can be given either

1. the *name* of the **symbol**,

2. the name of its [semantic macro](#),
3. or any suffix of its MMT-URI containing at least the [module](#) name.

The second option is often short – and therefore convenient to write; for example, to achieve “[formulae](#)”, we can also write `\symref{wff}{formulae}`, since `\wff` is the [semantic macro](#) for “[well-formed formula](#)”.

sTeX

The third option allows for distinguishing between multiple [symbols](#) with the same name – the [IDE](#) can help in the latter case, by underlining ambiguous [symbol](#) references in yellow, and offering the [Quick Fix](#) functionality to let you select and autocomplete the specific [symbol](#) you want to reference.

Since `\symname` and `\symref` are a lot to type for something that should ideally be used as often as possible, the [macros](#) `\sn` and `\sr` exist as well and behave exactly the same way. We also provide some convenience abbreviations for `\sn`; namely `\Sn` (capitalizes the first letter of the [symbol](#) name), `\sns` (adds an “`s`” at the end, for the most common pluralization of a name), and `\Sns` (both).

Using all of the above, our annotated fragment now looks like this:

Example 4

Input:

```
File [sTeX/Documentation]tutorial/intro/intro2stex.en.tex
5 \usemodule[sTeX/ComputerScience/Software]{mod/systems/tex?sTeX}
6 \usemodule[sTeX/Logic/General]{mod/syntax?Formula}
7 \usemodule[sTeX/MathBase/General]{mod?Mathematics}
8 \usemodule[sTeX/ComputerScience/Software]{mod/formats?PDF}
9
10 At its core is the \stex \sn{package} for \LaTeX, that allows for
11 semantically marking up document fragments; in particular
12 concepts, \sr{wff}{formulae} and \sr{mathematics}{mathematical}
13 statements (such as definitions, theorems and proofs). Running
14 \texttt{pdflatex} over \stex-annotated documents formats them
15 into normal-looking \PDF.
```

Output:

At its core is the [sTeX package](#) for [L^AT_EX](#), that allows for semantically marking up document fragments; in particular concepts, [formulae](#) and [mathematical](#) statements (such as definitions, theorems and proofs). Running `pdflatex` over [sTeX](#)-annotated documents formats them into normal-looking [PDF](#).

There’s only one problem: *the document does not compile*, with an error [Undefined control sequence](#). The reason being that *some macro* in the [module](#) [Formula](#) uses the `\text` [macro](#). We can fix that by using the [amsfonts package](#) of course, but this points to a more general problem; namely that [modules](#) can make use of various [L^AT_EX](#) [packages](#) for typesetting [symbols](#).

Good practice suggests putting those packages into a *prelude* per [math archive](#), which we can then import from anywhere, using the `\libinput` [macro](#). For more on that, see [section 5.3](#).

For now, suffice it to say that we can import all [packages](#) required for the [module](#) [Formula](#) from the [math archive](#) [sTeX/Logic/General](#) by adding the line

```
\libinput[sTeX/Logic/General]{preamble}
```

before the `\begin{document}`.

2.3 Modules and Simple Symbol Declarations

Consider again the first two paragraphs of [section 1.1](#):

[sTeX](#) is a system for generating human-oriented documents in either [PDF](#) or [HTML](#) format, augmented with computer-actionable semantic information (conceptually) based on the [OMDoc](#) format and ontology.

At its core is the [sTeX package](#) for [LaTeX](#), that allows for semantically marking up document fragments; in particular concepts, [formulae](#) and [mathematical](#) statements (such as definitions, theorems and proofs). Running `pdflatex` over [sTeX](#)-annotated documents formats them into normal-looking [PDF](#).

Firstly, note that the first paragraph would be perfectly suitable to serve as a pop-up definition on hover for the [sTeX symbol](#). Secondly, what if all the [symbols](#) used in the above *didn't* already exist?

In this section, we will describe how to make your own [symbols](#) and collect them as reusable fragments in [modules](#) and [math archives](#) from scratch.

We start by creating a new [math archive](#). In the [IDE](#), switch to the [sTeX](#)-tab on the left and click the “New sTeX Archive” button ([Figure 9](#)). You will then be asked for the



Figure 9: New Math Archive in the IDE

name of the [archive](#), a [namespace](#) for its content, and a [url-base](#), where the content is supposedly going to end up online. You can safely keep the defaults for the latter two. In the following, we assume that your archive is named `my/archive`.

The [IDE](#) will then create the following files and directories in your MathHub directory:

```

- my
  - archive
    - lib
      - preamble.tex
    - META-INF
      - MANIFEST.MF
    - source
      - helloworld.tex

```

...and open the file `helloworld.tex` with the content

```

1 \documentclass{sTeX}
2 \libinput{preamble}
3 \begin{document}
4 % A first sTeX document
5 \end{document}

```

You can now reference any newly created content in your new `archive` using for example `\usemodule[my/archive]{...}`.

Let's start with the “`\TEX`” symbol. Rename the file `helloworld.tex` to something more meaningful, for example `latex.en.tex` – the `.en` will be picked up on by `sTeX` to signify that the fragment will be in *english* (see [subsection 7.1.1](#)).

What we want to achieve in this file is the following:

`\TEX` is a document typesetting software developed by Donald Knuth, with a focus on mathematical formulae. It is based on a powerful and extensible `macro` expansion engine.

`\LaTeX` is a (nowadays) default collection of `\TEX` macros developed by Leslie Lamport. Among other things, `\LaTeX` introduces `environments`, a distinction between preamble and document content, `packages` to bundle and distribute `macro` definitions, and `document classes`: special `packages` that govern the global layout of a document.

In particular, in the `HTML` the two paragraphs above should be shown when hovering over the `symbols` they define (as indicated by the magenta definiendum highlighting). So we need `symbols` and `semantic macros`, for: `\TEX`, `macro`, `\LaTeX`, `environment`, `package` and `document class`.

`Symbol` declarations are only allowed within `modules`:

`\sTeX` A `module` is a *named* block that bundles `symbol` declarations for subsequent reuse.
A `module` is introduced with the `smodule`-environment.

Let's name our `module` `LaTeX`. We then wrap the contents of our document in a `smodule` environment:

```

\begin{document}
\begin{smodule}{LaTeX}
...
\end{smodule}
\end{document}

```

Note, that the `IDE` immediately picks up on this and displays the full `MMT-URI` of our new `module` over the `\begin{smodule}{LaTeX}` ([Figure 10](#)) –

```
http://mathhub.info/my/archive?LaTeX
\begin{smodule}{LaTeX}
```

Figure 10: VS Code Code Lense

From this, we can glimpse that the `namespace` of the `module` is `http://mathhub.info/my/archive/latex`. This implies, that to use the `module` somewhere else, we will have to type `\usemodule[my/archive]{latex?LaTeX}` – the `latex`-part pointing to the `file` and `LaTeX` referring to the actual `module`.

If we rename the file to `LaTeX.en.tex`, we notice that the `namespace` changes to `http://mathhub.info/my/archive`, allowing us to now use it with `\usemodule[my/archive]{LaTeX}` directly. That's because the `module` name `LaTeX` and the file name `LaTeX` match now (see [section 7.5](#), [Figure 11](#)).

```
http://mathhub.info/my/archive?LaTeX
\begin{smodule}{LaTeX}
```

Figure 11: VS Code Code Lense



Note that “`LaTeX`” and “`latex`” only differ in capitalization – if your file system is case-insensitive (as e.g. MacOS’s was until quite recently), this distinction gets murky, but remains very important especially if you want to share your `math archive` with others!

It is therefore *highly recommended* to treat file names as case-sensitive either way.

Within the `module`, we can now declare new `symbols` using the `\symdecl`-macro. We start with those that are not `text symbols`:

```
\symdecl*{macro}
\symdecl*{environment}
\symdecl*{package}
\symdecl*{document class}
```

The `*` after the `\symdecl` indicates, that we do not want a `semantic macro` for the `symbol` – otherwise, it would generate one with the same name as the `symbol` itself and “pollute the `macro` space”, so to speak.

The `symbols` `TeX` and `LTeX`, however, have a definite way of being typeset associated with them, which can be produced using the standard `\TeX` and `\LaTeX macros`. So let's make them `text symbols`, using the `\textsymdecl` macro:

```
\textsymdecl{tex}{\TeX}
\textsymdecl{latex}{\LaTeX}
```

The first argument being the name of the generated `macro` (i.e. `\tex` and `\latex`) and the second one specifying the output to produce.

2.4 Documenting Symbols

We can now use the two new macros, `\symname/\sn`, `\symref/\sr` etc. to mark up the above two paragraphs. But the IDE also makes us aware of the symbols not yet being documented, via squiggly blue lines(Figure 12).



Figure 12: Undocumented Symbols

Among other things, this means that the system does not yet know what to show a reader when hovering over the symbol in the [HTML](#). The IDE also recommends two ways to fix that: The [sdefinition](#) or [sparagraph](#) environments.

Ignoring the former for now, which is more useful for mathematical concepts, we can use the following to mark up the first paragraph:

```
\begin{sparagraph}[style=symdoc,for={tex,macro}]
  \tex is a document typesetting
  software developed by Donald Knuth, with a focus on
  mathematical formulae. It is based on a powerful
  and extensible \sn{macro} expansion engine.
\end{sparagraph}
```

In general, the [sparagraph environment](#) can be used to mark up arbitrary paragraphs semantically, but the `style=symdoc` option tells [STEX](#) to use this paragraph as a documentation for the symbols provided in the `for=` option. And indeed, doing so makes the squiggly blue lines in the IDE under `\textsymdecl{tex}{TeX}` and `\symdecl*{macro}` disappear.

We just used the [semantic macro](#) `\stex` and the `\sn` macro to mark up the fragment – but we can do better. Both concepts are being *introduced* in the above paragraph, and we can let [STEX](#) know that that is the case:

Within an [sparagraph environment](#) with `style=symdoc` (or an [sdefinition environment](#)), we can mark up *definienda*, meaning the terms *being defined*, explicitly. Analogously to `\symname` and `\symref`, we have the macros `\definame` and `\definiendum` for that purpose.

Note that the `\tex` macro induced by the `text` symbol above already marks up the “TeX” it produces, so wrapping it in another `\definiendum` would be redundant. However, every `text` symbol also generates a *second* macro with the suffix `name` that generates a non-marked-up version of the same presentation. In other words, we get the macro `\texname` for free, that produces “TeX” (of course, we could just as well use the `\TeX` macro, but that one you probably know already).

Furthermore, every `\definiendum` or `\definame` automatically adds the symbol being referenced to the internal `for=`-list of the [sparagraph environment](#), obviating the need to list it explicitly.

As such, we can produce a better markup like this:

```
\begin{sparagraph}[style=symdoc]
  \definiendum{tex}{\texname} is a document typesetting
```

```

software developed by Donald Knuth, with a focus on
mathematical formulae. It is based on a powerful
and extensible \definename{macro} expansion engine.
\end{sparagraph}

```

Exercise

In your archive `my/archive`, create additional files that produce the following outputs:

Mathematics.en.tex

To do **mathematics** is to be, at once, touched by fire and bound by reason. This is no contradiction. Logic forms a narrow channel through which intuition flows with vastly augmented force.

– Jordan Ellenberg

PDF.en.tex

Portable Document Format (PDF) is a document format that mixes text and graphics with a variety of content.

HTML.en.tex

The **HyperText Markup Language (HTML)** is a representation format for web-pages.

OMDoc.en.tex

OMDoc is a document format for representing **mathematical** documents with their flexiformal semantics.

such that the following file compiles and shows the above snippets on hover:

sTeX.en.tex

```

1 \documentclass{sTeX}
2 \libinput{preamble}
3 \begin{document}
4 \begin{smodule}{sTeX}
5   \usemodule{OMDoc}
6   \usemodule{PDF}
7   \usemodule{HTML}
8   \textsymdecl{sTeX}{\sTeX}
9   \begin{sparagraph}[style=symdoc]
10     \definiendum{sTeX}{\stexname} is a system for generating
11     documents in either \PDF or \HTML format, augmented with
12     computer-actionable semantic information (conceptually)
13     based on the \OMDoc format and ontology.
14   \end{sparagraph}
15 \end{smodule}
16 \end{document}

```

sTeX is a system for generating documents in either **PDF** or **HTML** format, augmented with computer-actionable semantic information (conceptually) based on the **OMDOC** format and ontology.

The preamble of every file should only be

```
\documentclass{stex}
\libinput{preamble}
```

and the macros `\OMDoc`, `\PDF`, `\HTML` should produce `\textsc{\OMDoc}`, `\textsf{\PDF}` and `\textsf{\HTML}`, respectively (but with semantic annotations of course).

Lösung: Can be found in [sTeX/Documentation]source/tutorial/solution

2.5 Sectioning and Reusing Document Fragments

We know now how to import and reuse the `symbols` of some `module` (using `\usemodule`). What about the actual document `content`?

Assume we want to write a new article that includes all of the fragments in `my/archive` we made so far, in a file `all.en.tex` in the same `math archive`:

```
1 \documentclass{article}
2 \usepackage{stex}
3 \libinput{preamble}
4 \begin{document}
5   \author{Me}
6   \title{The \texttt{my/archive} Archive}
7   \maketitle
8   \tableofcontents
9 ...
10 \end{document}
```

In there, we want sections as follows:

```
- 1 Preliminaries
  (Mathematics)
  - 1.1 Document Formats
    (PDF)
    (HTML)
    (OMDoc)
- 2 \TeX and Friends
  (LaTeX)
  (sTeX)
```

We could of course do the following:

```
\section{Preliminaries}
\input{Mathematics.en}
\subsection{Document Formats}
\input{PDF.en}
\input{HTML.en}
\input{OMDoc.en}
\section{\TeX and Friends}
\input{LaTeX.en}
\input{sTeX.en}
```

...but this approach has two drawbacks:

Firstly, we need to manually keep track of the section levels, by explicitly writing `\section`, `\subsection` etc. This is fine as long as we are just interested in this particular

article. But what if we want to *reuse* the article's content in another document at some point? The section levels might be entirely different then – e.g. we might want the “Preliminaries” section to be a subsection instead.

Secondly, the `\input` macro considers the file name/path provided to be either *absolute* or relative to the *current tex file being compiled* – which means that the `\input{Mathematics.en}` only works for files in the same directory as `Mathematics.en.tex`.

In short: using `\section`, `\chapter` etc. explicitly, and `\input` to reuse fragments, breaks reusability.

Instead of using `\section` and `\subsection`, **StEx** therefore provides the `sfragment` environment.

`\begin{sfragment}{Foo}... \end{sfragment}` inserts a sectioning header depending on the current section level and availability. These are: `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` and `\ subparagraph`. This allows us to do the following instead:

```
\begin{sfragment}{Preliminaries}
  \input{Mathematics.en}
  \begin{sfragment}{Document Formats}
    \input{PDF.en}
    \input{HTML.en}
    \input{OMDoc.en}
  \end{sfragment}
\end{sfragment}
\begin{sfragment}{\TeX and Friends}
  \input{LaTeX.en}
  \input{sTeX.en}
\end{sfragment}
```

The only problem remaining now is that if we do this, **StEx** will insert a `\part` for the first `sfragment`. If we want the “top-level” sectioning level to be `\section` instead, we can insert a `\setsectionlevel{section}` in the preamble.

As a more reuse-friendly replacement of `\input`, **StEx** provides the `\inputref` macro. Using that has two advantages: Firstly, its argument is relative to some (optionally provided, or the current) `math archive` and is thus independent of the specific location of the file relative to the currently being compiled `.tex`-file. Secondly, when converting to `HTML`, it will *not* “copy” the referenced file's content in its entirety (as `\input` would), but instead dynamically insert the already existent (if so) `HTML` of the referenced file, avoiding content duplication and having to process the file all over again.

In general `\inputref[some/archive]{file/path}` inputs the file `file/path.tex` in the `archive` `some/archive`. As the `\input`-ed files in the example above are in the same `archive` anyway, we can simply substitute the `\inputs` by `\inputrefs` and call it a day.

Finally, we can make two more minor changes:

1. The *title* of our document is only supposed to be there, if we compile the document directly – if we were to `\inputref` our file into a “driver file” `all.en.tex`, the title and the table of contents should be omitted.

We can achieve this using the `\ifinptref` conditional: by wrapping the header in an `\ifinptref \else... \fi`, it will only be processed if the file is *not* being loaded using `\inputref`. `\ifinptref` is a “classic” `TEX` conditional and is treated as such in both `PDF` and `HTML` compilation. A smarter `macro` to use is `\IfInputref`, which takes two arguments for the *true* and *false* cases, respectively. Additionally, when compiling to `HTML`, *both* arguments to `\IfInputref` will be processed, and the backend will decide which of the two to present when serving a document.

2. The table of contents should also be omitted in `HTML` mode. To achieve that, we can use the `\ifstexhtml` conditional, which is *true* if the document is being compiled to `HTML`, and *false* if compiled to `PDF`.



Note, that since *both* arguments of `\IfInputref` are processed, they should *not* open `TEX` groups or `environments`!

In summary, we can modify our document to do the following:

```
\IfInputref{}{
  \author{Me}
  \title{The \texttt{my/archive} Archive}
  \maketitle
  \ifstexhtml \else \tableofcontents \fi
}
```

The final `all.en.tex` can be found in `[sTeX/Documentation]tutorial/solution/all.en.tex`.

2.6 Building and Exporting `HTML`

So far we know how to write `STEX` documents, (we assume) how to build `PDF` files from them (via `pdflatex` of course), and on saving documents the `IDE` will preview the generated `HTML`. But if we do that with our new `all.en.tex`, we get presented with [Figure 13](#). Where did all of our fragments go?



Figure 13: Missing Fragments in the `HTML` Preview

Well, they don’t exist yet as `HTML`. The `HTML` Preview window in the `IDE` is really just that: A *preview*. But when using `\inputref`, it has to find the `HTML` of the `\inputref`ed fragment *somewhere*. Meaning: we have to compile all of the fragments



Figure 14: The Build PDF/XHTML/OMDoc Button

we used to [HTML](#) first. Individually, we can compile the currently open file in [VS Code](#) using the button in [Figure 14](#).

This will do the following:

1. Run `pdflatex` over the file three times.
2. Store the resulting `.pdf` in `[archive]/export/pdf/<filepath>.pdf`.
3. Convert the file to [HTML](#) and store it in `[archive]/xhtml/<filepath>.xhtml`.
4. Extract all the semantics and store them as [OMDoc](#) in `[archive]/content/..., [archive]/narration/... and [archive]/relational/....`
5. Construct a search index in `[archive]/export/lucene/....`

Doing all of this for every individual file *in hindsight* would of course be a huge hassle. We can therefore just compile the full [archive](#), folders in an [archive](#), or whole *groups of archives* via right-clicking an element in the [Math Archives](#) viewer in the [STeX](#) tab ([Figure 15](#)).



Figure 15: Building Archives in the [IDE](#)

Once that's done, saving `all.en.tex` again yields the correct [HTML](#) in the preview window.

At this point, it should be noted that you can't actually just open the [HTML](#) files exported to [archive]/xhtml in your browser and get all of the expected functionality – that shouldn't be too surprising. Features like the fancy pop-up windows require a semantically informed backend infrastructure, in the form of the [MMT](#) system. However, [MMT](#) *can* dump a standalone version for you. Let's do that now:

With our `all.en.tex` file open and everything built as above, click the `Export Standalone HTML`-button in the [IDE](#) (see [Figure 16](#)).



Figure 16: Exporting [HTML](#) in the [IDE](#)

In the dialog box that opens now, select an **empty** directory and [MMT](#) will dump a standalone version of our `all.en.tex` document there. You will still not be able to open it in the browser directly, because most browser forbid javascript modules on the `file://` protocol, but opening the file via `http` will yield the desired result, and you can now upload the directory's content to wherever you might want to use it.

If you want to test this, a quick and easy way to do so is to use [VS Code](#): You can install the `Live Server` extension, open the directory and click the `Go Live` button on the lower right of the window, which will start a small web-server in the selected directory and open its `index.html` in the browser for you.

Chapter 3

Mathematical Concepts

So far, we have seen how to declare and reference `symbols` generate `semantic macros` for `text symbols`, collect them in `modules` and document them properly.

But where **sTeX** really shines is when it comes to mathematics and related subject areas: `semantic macros` are significantly more useful when used for generating symbolic `notations` in math mode, and by associating `symbols` with (flexi-)formal semantics, **sTeX** can even *check* that your content is (to some degree) formally correct, or at least well-formed.

Also **sTeX** provides specialized functionality for mathematical `statements`: the text fragments marked as Definition, Theorem, Proof that are iconic to mathematical documents.

The example snippets in this chapter can be found in the [math archive sTeX/MathTutorial](#). If you downloaded the [sTeX/Documentation archive](#) in the **sTeX IDE**, you already have that [archive](#). If not, you can download it from within the **IDE**, as described in [chapter 2](#).

3.1 Simple Symbol Declarations

We will start with `symbols` and `semantic macros` for mathematical concepts and objects and their contribution to mathematical formulae.

3.1.1 Semantic Macros and Notations

Let us start with a very fundamental concept; namely `equality`. As you should by now know, declaring a new `symbol` requires a `module`, so let's open a new one and use `\symdecl`:

```
\begin{smodule}{Equality}
\symdecl{equal}
\end{smodule}
```

As mentioned in [section 2.3](#), the starred variant `\symdecl*` does not create a `semantic macro`, so presumably, the variant without a `*` *does*. And indeed, we now have a macro `\equal`, which however will produce errors if we try to use it. That's because we haven't told **sTeX** what to do with it yet.

A **semantic macro** is a **LATEX**-macro that allows for referencing a **symbol** itself, or – in the case of e.g. a function – the *application* of a **symbol** to (one or multiple) *arguments*; primarily by invoking a **symbol**'s **notation** in *math mode*.

The command `\symdecl{macroname}` declares a new **symbol** with name **macroname** and a **semantic macro** `\macroname`. In the case where we want the name and the **semantic macro** to be distinct, the command `\symdecl{macroname}[name=some name]` declares the name of the **symbol** to be **some name** instead.

The starred variant `\symdecl*{name}` declares the concept with the given name, but does not generate a **semantic macro**.

So let's provide equality with a **notation**. As a first step, we should let **STEX** know that “**equal**” takes two arguments. We might also want to shorten the **semantic macro** to e.g. `\eq`, without changing the name. Hence:

```
\symdecl{eq}[name=equal,args=2]
```

Next, we add an infix notation with the **notation** macro:

```
\notation{eq}{#1 = #2}
```

That seems like a lot to write, so for the very common case where we want to declare a **symbol** with a **semantic macro** and a **notation** all at once, the `\symdef` macro does all three by combining the optional and mandatory argument of `\symdecl` and `\notation`:

```
\symdef{eq}[name=equal,args=2]{#1 = #2}
```

and indeed, we can now use the `\eq` **macro** in math mode to invoke our new **notation**: $\backslash eq{a}{b}$ now yields $a = b$ – notably without any highlighting (and hover interaction in the **HTML**) though. Since our **semantic macro** takes *arguments*, which should be differently highlighted, we need to let our **notation** know which parts of the **notation** are highlightable components.

We can do so with the `\comp` and `\maincomp` macros:

The `\comp`-macro marks components to be highlighted in a **notation** for a **symbol** taking (one or more) arguments.

This is necessary because it is (nearly) impossible for **LATEX** to figure out, which parts of a **notation** to highlight and which not on its own – in particular, the highlighting should stop for the *arguments* of a **semantic macro**.

Additionally, the `\maincomp` macro can be used to mark (at most) one **notation** component to represent the *primary* component of the **notation**.

Notations that do not take arguments, as well as **operator notations**, are automatically wrapped in `\maincomp`.

In our case, this applies only to the “ $=$ ”, symbol, so:

```
\symdef{eq}[name=equal,args=2]{#1 \mathrel{\maincomp{=}} #2}
```



You may be wondering about the role of the `\mathrel` macro in the example above: **TeX** determines spacing/kerning in math mode by assigning a *class* to every character. Both individual characters and whole subexpressions can be assigned one of these classes using dedicated macros. These are:



class	TeX macro	examples
ordinary (default class)	<code>\mathord</code>	$\alpha i \diamond$
large operator	<code>\mathop</code>	$\sum \prod \int$
opening	<code>\mathopen</code>	([{
closing	<code>\mathclose</code>)] }
binary relation	<code>\mathrel</code>	$\leq > =$
binary operator	<code>\mathbin</code>	$+ \cdot \circ$
punctuation	<code>\mathpunct</code>	, ;

TeX “forgets” the class of an expression if it is wrapped in a `\comp` macro. It is therefore a good idea to wrap any occurrence of a `\comp` in the corresponding TeX macro for the desired class (e.g. `\mathrel{\comp{\leq}}`).

Having done so, we can now type `$\leq{a}{b}$` to get $a = b$. Thanks to using `\maincomp`, we now also have an **operator notation**, which we can invoke using `$\eq!`, yielding $=$.

What if we want to add more **notations**? Say we want to be able to invoke `equality` to get the variant notation $a \equiv b$ (without changing the intended meaning). If we want to be able to choose one of several **notations**, we should give the **notation** an *identifier*.

Let’s again modify our earlier **notation** by adding the identifier `eq` to the optional arguments of `\symdef`, like so:

```
\symdef{eq}[name=equal,args=2,eq]{#1 \mathrel{\maincomp{=}} #2}
```

We can now invoke the specific **notation** provided here by writing `$\eq[eq]{a}{b}$` to the same effect. But we can also add more **notations** using the `\notation` macro:

```
\notation{eq}[equiv]{#1 \mathrel{\maincomp{\equiv}} #2}
```

which we can now invoke with `$\eq[equiv]{a}{b}$`, yielding $a \equiv b$.

By default, the *first* **notation** provided for a given **symbol** is considered the *default notation*, which is invoked if the **semantic macro** is used without an optional argument – hence, `$\eq{a}{b}$` still yields $a = b$.

If we use the starred variant of the `\notation` macro, the **notation** is set as the new default. Hence, had we done

```
\notation*[eq][equiv]{#1 \mathrel{\maincomp{\equiv}} #2}
```

then `$\eq{a}{b}$` would now yield $a \equiv b$.

Any already existing notation can be set as default using the `\setnotation` macro; e.g. instead of using `\notation*`, we could also do

```
\notation{eq}[equiv]{#1 \mathrel{\maincomp{\equiv}} #2}
\setnotation{eq}{equiv}
```

Exercise

Implement the **symbol** “equal” as above in a new **module** “Equality” and add a documentation such that hovering over the **symbol** in the **HTML** yields the following snippet:

Two objects a, b are considered **equal** (written $a = b$ or $a \equiv b$), if there is no property that distinguishes them.

Lösung: Can be found in [sTeX/MathTutorial]/mod/Equality1.en.tex

3.1.2 Types and Variables

You might have noticed – after you save the file – that the expressions `\eq{a}{b}` and `\eq[equiv]{a}{b}` are underlined in yellow in the **IDE** and have a warning attached to them (Figure 17). If we click on the **Invalid Unit** link in the error message, we get

```
$\eq{a}{b}$ or $\eq[equiv]{a}{b}$,
\eqab
invalid unit:
http://mathhub.info/sTeX/MathTutorial/mod/Equality1/Equality?en?term 1?definition: Judgment |-- (implicit bind
[a:/I/1, b:(/I/2 a)] (apply (apply equal a) b)) :: /omitted_type (Invalid Unit)
View Problem (Alt+F8) No quick fixes available
```

Figure 17: Type Checking Warning

a somewhat cryptic stacktrace-like window (Figure 18). The reason being, that **MMT**

<http://mathhub.info/sTeX/MathTutorial/mod/Equality1/Equality?en?term 1?definition>

- - Judgment $\{ \} \dashv \{a: /I/1, b: /I/2 (a) \}_{I,a=b} :: /omitted_type$
 - trying typing rules
 - trying to simplify /omitted_type
 - no rule applicable
 - trying inference/typing rules
 - inferring type
 - inferring type of $\{a: /I/1, b: /I/2 (a) \}_{I,a=b}$
 - applying inference rule rule LambdaLikeRule\$LambdaTypingRule for implicit bind
 - Judgment $\{ \} \dashv /I/1 INHABITABLE$

Figure 18: Type Checking Proof Tree

actually tries to formally verify *everything we write using semantic macros!* It does so, by attempting to infer the *type* of an expression – success implies that the expression is in fact well-typed.

If the former paragraph is difficult to comprehend for you, don't worry – you'll likely pick up on things as we go along. For now, suffice it to say that we can assign “*types*” to *symbols*, and the **MMT** system is smart enough to use those to check that what we're writing actually “makes sense”; for example, $a + b$ makes perfect sense if $+$ is addition and a and b are numbers, or elements of a vector space, but not if a and b are, say, triangles.



Every **symbol** or **variable** can be assigned a **type**, signifying what “kind of object” the **symbol** represents, or what (primary) set it is contained in.



In order to *formally verify* a mathematical statement, we have to rely on a set of *rules* that determine what is or isn’t a valid statement. There are many systems of such rules with very different flavours, called (**logical**) **foundations**.

The most commonly used **foundation** in (informal) mathematics is *set theory*, in particular *ZFC*; a set of axioms in (usually) *first-order logic*. However, in *computer proof assistants* and similar systems, *type theories* like *higher-order logic* or the *calculus of (inductive) constructions* are more popular, because they lend themselves better to computer implementations.

In as far as possible, we prefer to remain “foundationally agnostic”, or **foundation independent**: Every **foundation** has advantages and disadvantages, and which one is appropriate often depends on the particular setting one is working in. Nevertheless, certain “meta-principles” have proven themselves to be extremely effective in representing and checking mathematical content in software, and while we do not fix a particular **foundation** or specific checking rules, we will make use of those principles in general. These include e.g. the *Curry-Howard Correspondance*, or *Judgments-as-Types paradigm*, and *Higher-Order Abstract Syntax*.



Full formal verification of document content is an extremely lofty goal, and hardly realistic if you’re not willing to write your content in pretty specific ways, and informed by a decent amount of background knowledge in formal logic. Moreover, formally verifying content in **STEX** is an ongoing research project, so we will not go into the specifics in detail here.

While full formal verification is out of reach for now, annotating adequate **types** can strike a useful balance between the effort required and the benefit of automated meaning checking afforded by them. In this sense **STEX** is pragmatically similar to programming languages where adding types can raise the quality and correctness assurance in programs.



Keep in mind that getting **Invalid Unit** warnings does not impact at all what your document is going to look like – feel free to ignore them entirely.

Types are particularly useful for **variables**:



A **variable** represents a *generic* or *unspecified* object.

Variables can be declared using the `\vardef`-macro, whose syntax is analogous to `\symdef`.

Note that **variables** are local to the current T_EX-group (e.g. environment).

Let’s leave our equality-**module** aside for now and turn our attention to something simpler: **natural numbers**. Consider the following module:

Example 5

Input:

```
\begin{smodule}{Nat}
  \symdef{Nat}[name=natural numbers]{\mathbb N}
  \begin{sparagraph}[style=symdoc]
    The \defname{Nat} $\defnotation{\Nat}$ are the numbers
    $0,1,2,\dots$ 
  \end{sparagraph}
  \symdef{plus}[name=addition,args=2]{#1 \mathbin{\maincomp{+}} #2}
  \begin{sparagraph}[style=symdoc]
    \Defname{addition} $\defnotation{\plus{a}{b}}$-
    refers to the process of adding two \sn{Nat}.
  \end{sparagraph}
\end{smodule}
```

Output:

```
The natural numbers N are the numbers 0,1,2,....
Addition a+b refers to the process of adding two natural numbers.
```

(like `\defname` and `\definiendum`, the `\defnotation` macro is only allowed in documenting environments like `sparagraph[style=symdoc]` or `sdefinition`, and highlights the `notation` components marked with `\comp` or `\maincomp` the same way as `\defname` and `\definiendum` do.)

Note, that as the `\Nat` semantic macro does not take any arguments, we do not need to wrap the `notation` in a `\comp` or `\maincomp`.

Note also, that the `\plus{a}{b}` is again underlined in the IDE with an `Invalid Unit` warning.

The above fragment uses two `variables` *a* and *b*. In fact, `MMT` will consider them `variables` even though they are not marked up as such – but since they are not marked up, we are missing out on useful functionality.

Let's change that by adding two `variable` definitions¹:

Example 6

Input:

```
\begin{sparagraph}[style=symdoc]
  \vardef{va}[name=a]{a}\vardef{vb}[name=b]{b}
  \Defname{addition} $\defnotation{\plus{\va}{\vb}}$-
  refers to the process of adding two \sn{Nat}.
\end{sparagraph}
```

Output:

```
Addition a+b refers to the process of adding two natural numbers.
```

Okay, so now *a* and *b* are gray, but besides that, we haven't achieved much yet.

¹Technically, this is called a *variable reservation*, for those in the know.

Let's change that by giving the variables the type \mathbb{N} :

Example 7

Input:

```
\begin{sparagraph}[style=symdoc]
\vardef{va}{name=a,type=\Nat}{a}\vardef{vb}{name=b,type=\Nat}{b}
\Defname{addition} $\defnotation{\plus{\va}{\vb}}$ refers to the process of adding two \sn{\Nat}.
\end{sparagraph}
```

Output:

Addition $a+b$ refers to the process of adding two natural numbers.

Now if we hover over the a and b (in the [HTML](#)), it will show us that their type is \mathbb{N} !

We can of course also assign [types](#) to [symbols](#). In the [IDE](#), find the [symbol “function space”](#) with [semantic macro](#) `\funspace` (in `[sTeX/MathBase/Functions]{mod?Function}`). The [OMDoc](#) preview window shows you how to use this [symbol](#) ([Figure 19](#)). This tells

▼ Symbol <code>function space (\funspace{a_1, ..., a_n}{b})</code>		
Type	$(A : \text{SET}, B : \text{SET}) \rightarrow \text{SET}$	
Notations	id	notation
	arrowtimes	$a_1 \times \dots \times a_n \rightarrow b$
	arrowcurry	$a_1 \rightarrow \dots \rightarrow a_n \rightarrow b$
	Arrowtimes	$a_1 \times \dots \times a_n \Rightarrow b$
	Arrowcurry	$a_1 \Rightarrow \dots \Rightarrow a_n \Rightarrow b$

Figure 19: Syntax Preview

us that if we write `\funspace{a_1, ..., a_n}{b}` (depending on which notation we use), we will get $a_1 \times \dots \times a_n \rightarrow b$.

We want `addition` to have type $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, hence we do:

```
\syndef{plus}[name=addition,args=2,
type=\funspace{\Nat,\Nat}{\Nat}
]{#1 \mathbin{\maincomp{+}} #2}
```



So far (and when using the `use` button in the [IDE](#)), we have been using the `\usemodule` macro to import content. `\usemodule` is allowed anywhere and imports the referenced `module` content local to the current [TeX](#) group.

Now that we use imported `symbols` in `types` (and since we are *in* a `module`), we

need to make sure that the imported `modules` are also (transitively) *exported*, since our new `symbols` now *depend* on the imported `module`.

For that we use the `\importmodule` macro within the `module`; i.e. the file should now look something like this:



```
\begin{smodule}{Nat}
\importmodule[sTeX/MathBase/Functions]{mod?Function}
...
```

Note that the `HTML` is aware of this now (after you save): *Clicking* on any occurrence of `addition` now yields [Figure 20](#).

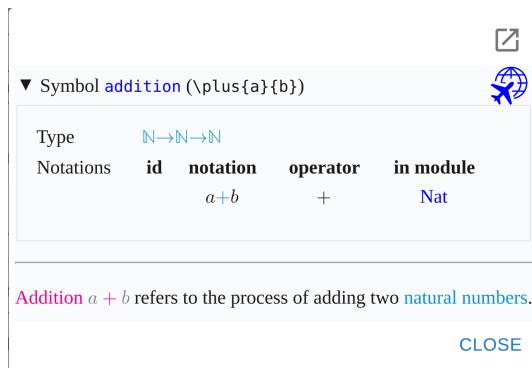


Figure 20: On-Click Popup in the `HTML`

However, the squiggly yellow `Invalid Unit` warnings are still there – that’s because everything we did with `types` so far still depends on our `natural numbers symbol`, which does not have a `type` yet.

By virtue of using `[sTeX/MathBase/Functions]{mod?Function}`, we also imported `[sTeX/MathBase/Sets]{mod?Set}`, which gives us the “`collection`” `symbol`. Let’s use this as a `type` for the `natural numbers`:

```
\symdef{Nat}[name=natural numbers,type=\collection]{\mathbb{N}}
```

Now if we save the file, all the squiggly lines are gone. Moreover, if you look at the `OMDoc` tab in the preview window, you can find [Figure 21](#). The **Document Elements** block collects all semantically annotated expressions in a `module` or document; including `variables` and the `$\plus{\va}{\vb}$`. Here, it tells us that it has checked the expression $a + b$ (in the context of $a : \mathbb{N}$ and $b : \mathbb{N}$), and inferred that it has `type N`.

Here’s what just happened:

1. The `MMT` system realized, that `$\plus{\va}{\vb}$` is the symbol “`addition`” applied to the two arguments a and b .
2. It knows, that “`addition`” has `type N × N → N`².

²Do not worry that the IDE actually reports the type `{a : N, b : N} ⊢ N`, this is an artefact of the underlying type system with dependent types used by `sTeX`; it just means $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ in this special case, but would also allow a and b to appear in the range type in more complex situations; see ?? for details.

▼ Document Elements

- Variable `a` (`\va`) of type `N`
- Variable `b` (`\vb`) of type `N`
- ▼ $\{a: N, b: N\}_I^{a+b}$

Inferred Type: $\{a: N, b: N\}_I^N$

Figure 21: Inferred Type

3. It knows, that this means that if the two arguments `a` and `b` both have type `N`, then the full expression has type `N`.

Here's something you can now try: If we *remove* the types from the variables `a` and `b` again, the warnings are *still* gone. We lose the type information on hover, but MMT still doesn't complain, because it now realizes that since `a` and `b` have no explicit types given, it should infer them. And by the same chain of reasoning as above, it can infer that since they are being used as arguments for addition, they need to have type `N`.

3.1.3 Flexary Macros and Argument Modes

Here is one thing you might wonder: Writing `$\plus{a}{b}$` is one thing, but what if we want to produce $a + b + c + d + e$? Do we really need to write `$\plus{a}{\plus{b}{\plus{c}{\plus{d}{e}}}}$`?

Of course not. We can declare the symbol such that the semantic macro `\plus` expects a (comma-separated) sequence of arguments instead of two “normal” arguments.

The optional `args`-argument of `\symdecl` expects a string of characters indicating the semantic macro's argument modes. There are four such modes:

- STEX**
- i a simple argument,
 - a a – (left or right) associative – sequence argument, represented as a single TeX-argument `{a,b,...}`,
 - b A binding argument that expects a variable that is bound by the symbol in its application, and
 - B A binding sequence argument of arbitrarily many bound variables by the symbol `({x,y,z,...})`.

If `args` is given as a number `n` instead, the semantic macro takes `n` arguments of mode i.

Example 8

- For `\plus{a,b,c}` yielding $a + b + c$, we do `\symdecl{plus}[args=a]`,
- for `\inset{a,b,c}{A}` yielding $a, b, c \in A$, we do `\symdecl{inset}[args=ai]`,
- in `\add{i}{1}{n}{f(i)}` yielding $\sum_{i=1}^n f(i)$, the variable `i` is bound in the ex-

pression, we hence do `\symdecl{add}[args=biii]`,

- in `\foral{x,y,z}{P(x,y,z)}` yielding $\forall x, y, z. P(x, y, z)$, the variables x, y, z are all **bound** by the \forall , we hence do `\symdecl{foral}[args=Bii]`.

So when we wrote `\symdecl{plus}[args=2]`, this was actually shorthand for `\symdecl{plus}[args=ii]`.

Let's revise our previous declaration and the syntax of the `\plus` macro:

```
\symdef{plus}[name=addition,args=a,
  type=\funspace{\Nat,\Nat}{\Nat}
]#1 \mathbin{\maincomp{+}} #2
\begin{paragraph}[style=symdoc]
  \vardef{va}[name=a]{a}\vardef{vb}[name=b]{b}
  \Defname{addition} $ \defnotation{\plus{\va,\vb}}$%
  refers to the process of adding two \sn{\Nat}.
\end{paragraph}
```

Now we get new errors, that are easy to explain: Our **notation**

`#1 \mathbin{\maincomp{+}} #2` refers to *two* arguments, but our **semantic macro** only takes *one* (albeit a **sequence argument**). We now need to let **STeX** know what to do with the **sequence argument** in our **notation**. Using the `\argsep` macro, we can tell **STeX** to insert the *separator* “ $+$ ” between the individual elements of the **argument sequence** #1:

```
\symdef{plus}[name=addition,args=a,
  type=\funspace{\Nat,\Nat}{\Nat}
]#\argsep{\#1}{\mathbin{\maincomp{+}}}}
```

Now we can finally write `$\plus{a,b,c,d,e}$` and get $a + b + c + d + e$ – hooray!

...expect that our squiggly yellow **Invalid Unit** warnings are back. That's because the **type** of **addition** still corresponds to a binary operation, rather than a unary function on sequences.

We *could* change the **type** of course, but we shouldn't *want* to or *have* to: platonically, **addition** is *still* a *binary function*; we just introduced the **a-mode** argument for *our convenience* as authors.

Instead, we can tell **MMT** how to “resolve” the **sequence argument** into a nested application of **addition**. In the very common case we have here, where the **symbol** represents an *associative binary operator*, we can just add the argument **assoc=bin** to the `\symdecl` (or `\symdef`) **macro**:

```
\symdef{plus}[name=addition,args=a,assoc=bin,
  type=\funspace{\Nat,\Nat}{\Nat}
]#\argsep{\#1}{\mathbin{\maincomp{+}}}}
```

and the warnings are gone again. Formally/internally, **MMT** will now turn the term `addition(sequence(a,b,c))` into `addition(a,addition(b,c))`.

Exercise

Analogously to the above, implement a **symbol** “multiplication” with **semantic macro** `\mult`, that takes a single **sequence argument** and has a default **notation** such that `\mult{a,b,c}` produces $a \cdot b \cdot c$.

Lösung: Can be found in [sTeX/MathTutorial]mod/Nat.en.tex

3.1.4 Precedences

If you have done the previous exercise, you now have *semantic macros* `\plus` and `\mult` at your disposal. We can of course nest them to produce e.g. $a + b \cdot c$ (with `\plus{a, \mult{b, c}}`). If we do `\mult{a, \plus{b, c}}` however, we get $a \cdot b + c$. Annoying – we now have to insert parentheses: `\mult{a, (\plus{b, c})}`... or do we?

We do *not*. Instead, we can assign *precedences* to *notations* to have *STEX* insert parentheses automatically.

notation (and hence *symdef*) take an optional argument `prec=<opprec>;<argprec1>x...x<argprec n>` consisting of an **operator precedence** `<opprec>` and for each argument `k` an **argument precedence** `<argprec k>`.

All *precedences* are integers, e.g. 10 or -500. It is good practice to use *precedences* that leave enough room to smuggle values inbetween, so that we can fine-tune them later as more symbols may intervene.

The precise numbers used for *precedences* are arbitrary – what matters is which *precedence* is higher than which other *precedence* when used together.

By default, all *precedences* are 0, unless the *symbol* takes no arguments, in which case the *operator precedence* is `\neginfprec` (negative infinity).

If we only provide a single number in `prec=`, this is taken as both the *operator precedence* and all *argument precedences*.

The *lower* a *precedence*, the *stronger* a *notation* binds its arguments. In our particular case, we want *multiplication* to bind stronger than *addition*, so we can (arbitrarily) assign them *precedences* e.g. 10 and 20:

```
\symdef{plus}[name=addition,args=a,assoc=bin,prec=20,
  type=\funspace{\Nat,\Nat}{\Nat}
] {\argsep{\#1}{\mathbin{\maincomp{+}}}}
\symdef{mult}[name=multiplication,args=a,assoc=bin,prec=10,
  type=\funspace{\Nat,\Nat}{\Nat}
] {\argsep{\#1}{\mathbin{\maincomp{\cdot}}}}
```

And now if we type `\mult{a, \plus{b, c}}`, *STEX* will automatically insert parentheses and yield $a \cdot (b + c)$ – and conversely, if we do `\plus{a, \mult{b, c}}`, *STEX* will *not* insert parentheses and yield $a + b \cdot c$.

3.1.5 Implicit Arguments

Let us turn our attention back to *equality*. Here's an almost philosophical question: *What is the type of “equality”?* Asking (the right kind of) mathematicians this question can cause fist fights to break out. As such, we will not give a definitive answer, *but* here is an informative approach that has proven to be quite effective in computational settings:

Equality is a *polymorphic binary relation* on an *implicit collection A*. And a *relation* is a function into a *type of propositions*.

We will see the advantage of this approach over time. For now, consider that given objects a and b , the expression “ $a = b$ ” is either true or false³, and “*equal*” takes two argu-

³Assuming classical logic – if you prefer to remain intuitionistic/constructive, note that *STEX*, being *foundation independent*, does not enforce the law of excluded middle!

ments, so if we have a `type` of “truth values”, it makes sense to model “`equal`” as a function taking two arguments and returning that `type`. So we do `type=\funspace{...}`?

Here’s the idea with respect to *implicit arguments*. Let’s first declare a new `variable` of `type “collection”`:

```
\vardef{vA}[name=a,type=\collection]{A}
```

We now assign the `type` $A \times A \rightarrow \text{Prop}$ to `equal`:

```
\symdef{eq}[name=equal,args=2,eq,
  type=\funspace{\vA,\vA}{\prop}
]#1 \mathrel{\maincomp{=}} #2
```

(The symbol “`proposition`” with `semantic macro` `\prop` comes with `STEX` directly; we say that it is part of the `STEX`.)

Now our `type` has a free variable A . For `MMT`, this now means that `equal` actually takes *one more argument*, but one whose value is uniquely determined from the other arguments. Indeed, if you consider `equal` to take three arguments (the first one being some A of `type collection`), then the *next* two arguments *enforce* that the first argument has to be the `type` of the other two.

In other words: A is now an implicit argument that `MMT` is tasked with inferring whenever we use `equal`, and that we never explicitly provide in `STEX`.

Indeed, if we use our `module Nat` from before, and apply `\eq` to a variable of type \mathbb{N} , `MMT` does not complain:

```
\usemodule{mod?Nat}
\vardef{vn}[name=n,type=\Nat]{n}
\$ \eq{\vn}{m} \$
```

And if we inspect the `OMDOC` tab in the `HTML` preview, we can see exactly what `MMT` did (Figure 22). We can see

The screenshot shows the OMDoc tab in an HTML preview. It displays the definition of the `equal` symbol and its usage in a document element.

Module Equality

Includes Function

Symbol `equal` (`\eq{a}{b}`)

Type $\{A: \text{SET}\}_I A \rightarrow A \rightarrow \text{Prop}$

Notations id notation operator in module

eq	$a = b$	=	Equality
equiv	$a \equiv b$	\equiv	Equality

Document Elements

Includes `Nat`

► Variable `n` (`\vn`) of type \mathbb{N}

▼ $\{n: \mathbb{N}, m: \mathbb{N}\}_{\frac{n=m}{\mathbb{N}}}$

Inferred Type: $\{n: \mathbb{N}, m: \mathbb{N}\}_{\mathbb{N}} \text{Prop}$

Figure 22: Implicit Arguments

1. (by the $\{\cdot\}_I \dots$) that `MMT` considers A an implicit argument in the `type` of `equal`,

2. that the *inferred* type of $n = m$ is `Prop`,
3. that `MMT` inferred the implicit argument of `equal` in $n = m$ to be `N` (by the $\underbrace{\dots}_{N}$), and
4. that it was enough to give `\vn` the explicit `type N` – `MMT` also inferred that hence m also has to have `type N`!

3.1.6 Finishing Equality

You might wonder if – as with `addition` – we can make “`equal`” take a `sequence` argument as well. Naturally, we can:

```

1  \symdef{eq}{name=equal,args=a,eq,
2   type=\funspace{\vA,\vA}{\prop}
3   ]{\argsep{\#1}{\mathrel{\maincomp}}}}
4   \notation{eq}[equiv]{\argsep{\#1}{\mathrel{\maincomp\equiv}}}}
```

and as before, we now get `Invalid Unit` warnings. Unlike before, however, we can not just fix this with adding `assoc=bin`. As mentioned, `bin` instructs `MMT` to “fold” the `symbol` over the arguments, so when doing `\eq{a,b,c}`, `MMT` would turn this into `equal(a,equal(b,c))`, i.e. the claim that “ a is equal to $b = c$ ” – but that’s not what $a = b = c$ means. What we mean by $a = b = c$ is really “ $a = b$ and $b = c$ ”.

For that, we can use `assoc=conj` – however, that requires that some `symbol` that can be used for *conjunction* (i.e. “and”) is in the current scope.

If we search for `conjunction` in the `IDE`, we should find the `module [sTeX/Logic/General]{mod/syntax?Conjunction}`.

Using that, we can now write the following:

```

\usemodule{mod?Nat}
\usemodule[sTeX/Logic/General]{mod/syntax?Conjunction}
\vardef{\vn}{name=n,type=\Nat}{n}
\$ \eq{\vn,m,p} $
```

Upon saving, `MMT` does not complain; and if we inspect the `OMDoc` tab in the `HTML` window again, we now notice that `MMT` correctly resolved this as in [Figure 23](#).

3.1.7 Variable Sequences

There is a special kind of `variable` in `STEX` for when we want to use *sequences* of `variables`.

We can use the `\varseq` macro to declare a new sequence `variable`; in the simplest case that looks something like the following:

```
\varseq{\seqn}{name=n,type=\Nat}{1,\ellipses,k}{\maincomp{n}_{\#1}}
```

We have just declared a new variable sequence of `type N`, that ranges over indices $1, \dots, k$, with `notation` n_i for some specific index i .

If we now do `\seqn{i}`, we get n_i , and if we do `\seqn!`, we get n_1, \dots, n_k .

We can also do multi-dimensional sequences, e.g.

```
\varseq{\seqm}{name=m,type=\Nat,args=2}
{\{1\}\{1\},\ellipses,\{\ell\}\{k\}}
{\maincomp{m}_{\#1}^{\#2}}
```

The screenshot shows a software interface for defining mathematical structures. At the top, there's a section titled "Module Equality". Below it, under "Includes Function", is a symbol "equal" (\eq{a_1, ..., a_n}) with a small globe icon next to it. Under "Document Elements", there's a section for "Nat, Conjunction". Inside this, a conjunction of equalities is shown: $\{n: \mathbb{N}, m: \mathbb{N}, p: \mathbb{N}\} \frac{n=m \wedge m=p}{\mathbb{N} \quad \mathbb{N}}$. Below this, the "Inferred Type" is given as $\{n: \mathbb{N}, m: \mathbb{N}, p: \mathbb{N}\} \text{Prop}$.

Figure 23: Conjunction of Equalities

Now $\text{\seqm}{i}{j}$ produces m_i^j , and $\text{\seqm}!$ produces m_1^1, \dots, m_ℓ^k .

Of course, we can manually change the way $\text{\seqn}!$ is typeset by providing an explicit [operator notation](#) using `op=`; e.g. if we do

```
\varseq{\seqn}[name=n,type=\Nat,op={(n_i)_{i=1}^k}]
  {1,\dots,k}\{\maincomp{n}_{\#1}\}
```

then $\text{\seqn}!$ produces $(n_i)_{i=1}^k$.

So far so nice, but sequence variables get especially useful in combination with [sequence arguments](#): Consider for example the `\plus` semantic macro for [addition](#). This expects one [sequence argument](#), or alternatively, a *sequence variable*: `\plus{\seqn}` now produces $n_1 + \dots + n_k$, and `\eq{\seqm}` now produces $m_1^1 = \dots = m_\ell^k$.

TODO⁴

3.2 Statements

Now that we have [equality](#), [natural numbers](#), [addition](#) and [multiplication](#) at our disposal, let's implement some *statements*. Both [addition](#) and [multiplication](#) are, for example, [associative](#) and [commutative](#).

We could state these properties directly for the two operations, but we can also first define [associativity](#) and [commutativity](#) in general, and then assert them specifically for [addition](#) and [multiplication](#).

3.2.1 Definitions

Let's define what it means to be [associative](#). This means, of course, declaring a new [symbol](#). Note that we don't need a [semantic macro](#) for [associativity](#), since there is no [notation](#) to attach to it. We will also for now ignore its [type](#). Note however, that [associativity](#) is still a property of (binary) operations, so it still makes sense to have the [symbol](#) take an [argument](#); namely the operation it applies to.

⁴TODO: seqmap

We will also finally provide an actual (more or less) formal *definition* for the `symbol`, so where we used the `sparagraph` environment with `style=symdoc` before, we will now use the `sdefinition` environment, which also gives us `\defname`, `\definiendum`, `\defnotation` and all that.

A first variant of a corresponding `module` could look like this:

Example 9

Input:

```
File [sTeX/MathTutorial]props/Associative1.en.tex
4 \begin{smodule}{Associative}
5   \importmodule{mod?Equality}
6
7   \symdecl*[associative][args=1]
8   \begin{sdefinition}[for=associative]
9     \vardef{vA}[name=A,type=\collection]{A}
10    \vardef{vop}[name=op,type=\funspace{\vA,\vA}\vA,args=a,assoc=bin]
11    {\argsep{\#1}{\mathbin{\maincomp{\circ}}}}
12    %
13    A binary operation \$\fun{\vop!}{\vA,\vA}\vA\$ is called
14    \defname{associative}, if
15    \$\eq{
16      \vop{(\vop{a,b}),c},
17      \vop{a,(\vop{b,c})}
18    }$ for all \$\inset{a,b,c}\vA$.
19  \end{sdefinition}
20 \end{smodule}
```

Output:

Definition 3.2.1. A binary operation $\circ : A \times A \rightarrow A$ is called **associative**, if $(a \circ b) \circ c = a \circ (b \circ c)$ for all $a, b, c \in A$.

Note, that the **semantic macros** `\fun` and `\inset` come from `[sTeX/MathBase/Functions]mod?Function` and `[sTeX/MathBase/Sets]mod?Set`, respectively. Also note, that the **variable** declaration for `\vop` makes use of all the fun features we already discussed for `addition`.



Note that the above is more than good enough, if you merely want to produce nice-looking, “wikified” **HTML** and **PDF** documents. The rest of this subsection will cover how to add more flexiformal semantics to the above.

If this seems laborious and/or difficult, keep in mind that this is to some degree experimental still, and you are not forced to go overboard with semantic annotations!

But if you aim to create a “library of symbols” for mathematical concepts, then all of the possibilities that we discuss here will add value for the community. Generally, the higher the ratio of readers to authors the more any investment in semantization will pay off.

Semantic Macros in Text Mode

The first thing we can do to further improve this document is marking up the “for all” in the definition – after all, there naturally is a `symbol` for the `universal quantifier`, which can be found in `[sTeX/Logic/General]mod/syntax?UniversalQuantifier` and has the `semantic macro` `\foral` (as to not conflict with the `TeX` primitive `macro` `\forall`).

The naive approach would be to replace the “for all” by e.g. `\sr{foral}{for all}`. That would (correctly) associate and highlight the text fragment with the `symbol` “universal quantifier”, *but* we are not just referencing the `symbol` here – we are actually using it, by *applying* it to the `variables` a, b, c and the expression $(a \circ b) \circ c = a \circ (b \circ c)$.

In *math mode*, we can just use the `semantic macro` `\foral` – that will take two arguments (of `modes` `B` and `i`) and produce the corresponding `notation`, so that

```
$\foral{\inset{a,b,c}{\vA}}{  
    \eq{\vop{(\vop{a,b}),c}, \vop{a,(\vop{b,c})}} }  
}$
```

will produce $\forall a, b, c \in A. (a \circ b) \circ c = a \circ (b \circ c)$.

In *text mode*, however, we don’t have a specific `notation` – instead, the specific “`notation`” is whatever sentence we want to mark up semantically. In text mode, `semantic macros` therefore behave differently:

1. They take *precisely* one argument, regardless of how many arguments the `macro` would take in math mode or (equivalently) the `args` property of the `symbol`.
2. *Within* that argument, we can use `\comp` to highlight arbitrary text fragments, and
3. we can use the `\arg` macro to mark up the *actual* arguments that the `symbol` is supposed to be applied to.

`\arg` takes as optional argument the index of the argument that is being marked up; if not they are used consecutively. The starred variant `\arg*` produces no output.

So we could now do

```
\foral{\comp{For all}}{$\arg{\inset{a,b,c}{\vA}}$, we have  
$ \arg{  
    \eq{\vop{(\vop{a,b}),c}, \vop{a,(\vop{b,c})}} }  
}$}
```

which produces “For all $a, b, c \in A$, we have $(a \circ b) \circ c = a \circ (b \circ c)$ ”.

In our case though, we want to “switch the arguments around” – first comes the equation, then the `variables` to be bound. Hence:

```
\foral{  
    $ \arg[2]{  
        \eq{\vop{(\vop{a,b}),c}, \vop{a,(\vop{b,c})}} }  
    }$  
    \comp{for all}  
    $ \arg[1]{\inset{a,b,c}{\vA}} $  
}
```

which produces “ $(a \circ b) \circ c = a \circ (b \circ c)$ for all $a, b, c \in A$ ”.

Definientia

Now we have a fully semantically annotated expression in the definition for “`associative`”. Can we let `MMT` know, that this expression really is *the* definition of the `symbol`?

Yes, we can. All we need to do is wrap the sentence in a `\definiens` macro (plural: `definientia`; like the word “`definiendum`” refers to “the term being defined”, “`definiens`” refers to “the thing the term is being defined *as*”).

The `\definiens` macro is only allowed within the `sdefinition` environment, and requires that the `environment` lists the `symbol` that gets the definientia attached explicitly in its `for=` argument. It is possible to attach definientia to multiple `symbols` within an `sdefinition environment`, in which case the symbol needs to be provided as an optional argument, e.g. we could do `\definiens[associative]{...}`. Since “`associative`” is the only `symbol` being defined in our definition, we can omit that argument.

Alternatively, as with `types` we can attach definientia to a `\symdecl` directly using the optional argument `def=....`

At this point, you might justifiably wonder, why we even still need to declare `associative` with `\symdecl*` before we define it. And indeed, we don’t – the `sdefinition environment` takes the same optional arguments as the `\symdecl` macro, and if we explicitly provide a `name=` (or a `macro=`), it will generate a `symbol` for us. We can hence get rid of the `\symdecl*` and instead do:

```
1 \begin{sdefinition}[name=associative,args=1]
2 ...
3 \end{sdefinition}
```

One more problem remains: We stated that `associative` is to take one argument – but we haven’t told `STEX` what it is yet. In our case, the argument is represented by the `variable` `\vop`. In fact, chances are that arguments to symbols in `types` or definientia are almost always represented by some `variable`.

We can use one of two ways to a `variable` as being an argument:

1. If the `variable` (e.g. `\vop` with name `op`) was already declared prior to the `sdefinition environment`, we can use the `\varbind` macro in the `environment`; e.g. by adding `\varbind{op}`.
2. We can move (or copy) the `\vardef` for the `variable` into the `environment` and add `bind` to its optional arguments.

In total, our fully annotated definition now looks like this:

Example 10

Input:

```

File [sTeX/MathTutorial]props/Associative.en.tex

8 \begin{sdefinition}[name=associative,args=1]
9   \vardef{vA}[name=A,type=\collection]{A}
10  \vardef{vop}[name=op,type=\funspace{\vA,\vA}\vA,
11    args=a,assoc=bin,bind % <- argument for the symbol
12  ]{\argsep[#1]{\mathbin{\maincomp{\circ}}}}
13  \vardef{va}[name=a,type=\vA]{a}
14  \vardef{vb}[name=b,type=\vA]{b}
15  \vardef{vc}[name=c,type=\vA]{c}
16 %
17 A binary operation $\mathit{\fun{vop}}(\vA,\vA)\vA$ is called
18 \definename{associative}, if
19 \definiens{$\forall a,b,c:A. \mathit{\eq{(\mathit{vop}{(\mathit{vA},\mathit{vb})},\mathit{vc})} = \mathit{vop}{(\mathit{vA},(\mathit{vop}{(\mathit{vb},\mathit{vc})})}}}$}
20   \mathit{vop}{(\mathit{vA},(\mathit{vop}{(\mathit{vb},\mathit{vc})}))}
21 }$ \comp{for all} $\mathit{arg}[1]{\mathit{inset}{\mathit{va},\mathit{vb},\mathit{vc}}}\vA$}.
22 \end{sdefinition}
23 %
24 %

```

Output:

Definition 3.2.2. A binary operation $\circ : A \times A \rightarrow A$ is called **associative**, if $(a \circ b) \circ c = a \circ (b \circ c)$ for all $a, b, c \in A$.

And indeed, if we look at the **OMDoc** tab of the **HTML** preview, we can see that not only does **MMT** attach the definiens to the **symbol**, it has also inferred the **type** of “**associative**” from the definiens (Figure 24).

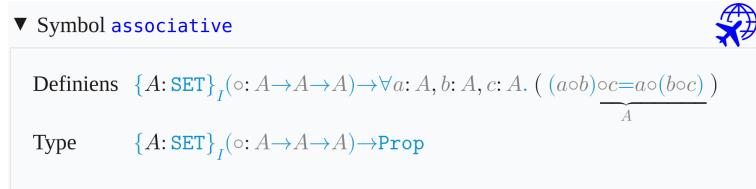


Figure 24: Type Inferred from Definiens

Using Symbols Without Semantic Macros and Exporting Code in Modules

So now we don't have a **semantic macro** for “**associative**”, but it *does* take an argument. How can we ever actually *use* the **symbol** now?

The answer is: with the **\symuse** macro. Like **\symref** and friends, **\symuse** takes a **symbol** name or the name of its **semantic macro** as argument, but behaves otherwise like using a **semantic macro** directly. So for, say, **addition**, **\symuse{addition}** and **\symuse{plus}** behave exactly like **\plus**.

In our case, this means we can do **\symuse{associative}**. “**associative**” does not have a **notation**, but in practice, we say something like “**+ is associative**” rather than using some specific mathematical **notation** for the same thing.

Combining this with what we just learned, we can now say that `addition` is associative by doing:

```
\symuse{associative}{$\arg{\plus!}$} \comp{is associative}
```

In fact, we would do the exact same thing every time we want to say that *some* operator is associative, so it makes sense to introduce a `macro` for this. In fact, such a `macro` is easy to define using standard `LATEX` methods. This is where `\STEXexport` becomes very handy:

In a `module`, we can put arbitrary `LATEX` code in an `\STEXexport`, and this code will be executed every time the `module` is imported via `\usemodule` or `\importmodule`. This is especially useful for `macro` definitions, and this way `modules` can almost act like `LATEX packages`!

So we can define a new `macro` `\isassociative` that applies “`associative`” to an arbitrary operation and produces the semantically marked-up text “#1 is `associative`”, and wrap that `macro` definition in an `\STEXexport`, and whenever we use the `Associative module`, we also get the `\isassociative` `macro`:

```
\STEXexport{
  \def\isassociative#1{
    \symuse{associative}{$\arg{\#1}$} ~is ~\comp{associative}
  }
}
```

And now, we can do e.g. `\isassociative{\plus}` to produce “`+ is associative`”.

For technical reasons, `\STEXexport` processes its content in the `expl3` category code scheme – what this means is that all spaces are ignored entirely, and the characters `_` and `:` are valid characters in `macro` names.



In practice, this means you will have to use the `-` character for spaces, and if you want to use a subscript `_`, you should use the `macro` `\c_math_subscript_token` instead.

Exercise

Analogously to all the above, implement a `module` for `commutativity`; i.e the property of a binary operation that $a \circ b = b \circ a$ for all a, b . Make the `module` export a macro `\iscommutative` analogously to `\isassociative`.

Lösung: Can be found in [sTeX/MathTutorial]props/Commutative.en.tex

TODO⁵

3.2.2 Assertions

Having defined `associativity` and `commutativity`, we can now assert that both properties hold for `addition` and `multiplication`.

For `assertions` (i.e. theorems, lemmata, axioms, claims,...), `sTeX` provides the `sassertion environment`.

In the simplest case, that can look like the following:

⁵TODO: intent?

```
\begin{sassertion}
  \isassociative{\Sn{plus}}
\end{sassertion}
```

which yields

Addition is associative

Do we want this to be typeset as a **Theorem**? For that we just add a `[style=theorem]` to the **sassertion environment**, provided we have a customization for that – (see [chapter 9](#)). We can also load the **stexthm package**, which uses the **amsthm package** to provide common typesettings for the types: **theorem**, **observation**, **corollary**, **lemma**, **axiom** and **remark**.

So far, this is not too useful – after all, we could have just as well used e.g. the **amsthm package** and gone straight for the non-**STEX** variant

```
\begin{theorem}
  \isassociative{\Sn{plus}}
\end{theorem}
```

But as with **sdefinition**, we can immediately add a corresponding **symbol** in the **sassertion environment**, and have it be documented directly by the **environment**:

```
\begin{sassertion}[style=theorem, name=addition is associative]
  \isassociative{\Sn{plus}}
\end{sassertion}
```

And now, if we do `\sn{addition is associative}`, we get **addition is associative** with a corresponding hover pop-up (in the [HTML](#)).

Of course, the usefulness of this grows with more elaborate assertions. For very short assertions such as the above, we might not even want to typeset them in such a space hungry manner.

For that purpose, we provide the **\inlineass macro** (and analogously: **\inlinedef** for **sdefinition**), which takes the same optional arguments as the **environment**. So we could also do:

```
\inlineass [name=addition is associative]{\isassociative{\Sn{plus}}}
```

So far, **MMT** is blissfully unaware of the semantic contents of our assertions. We can easily remedy that by wrapping the expression representing the assertion in a **\conclusion macro**, analogously to the **definiens macro** in **sdefinitions**:

```
\inlineass [name=addition is associative]{
  \conclusion{\isassociative{\Sn{plus}}}
}
```

We can now see the statement in the **OMDoc** tab of the [HTML](#) preview ([Figure 25](#)).

$$\triangleright \text{Assertion addition is associative} \vdash \text{apply } \left(\text{apply } \left(\underbrace{\text{associativeN}}_{\mathbb{N}} \right) + \right)$$

Figure 25: Assertion Statement in **OMDoc**

Not exactly pretty – the **OMDoc** tab uses **notations** to render content, and we did not provide any for **associative**.

Notice the \vdash symbol after the name of the assertion? As an aside for those who are curious:

The **judgments as types** paradigm represents the validity of **proposition** via a designated *type of proofs*: For any **proposition** P , we introduce a collection $\vdash P$ of *proofs* of P .
sTeX
To say that the **proposition** *holds* is then equivalent to positing that *some* element $p : \vdash P$ exists – in which case *proofs* become typed objects in their own right.

Let's consider a more interesting statement now. How about the **induction axiom**?

```
\begin{assertion}[style=axiom,name=induction axiom]
  Let  $\varphi(n)$  a property on  $\mathbb{N}$ . If
  \begin{enumerate}
    \item  $\varphi(0)$  and
    \item if  $\varphi(m)$  holds for some  $m$ , then
       $\varphi(\text{plus}\{m, 1\})$  also holds,
  \end{enumerate}
  then  $\varphi(n)$  holds for all  $n \in \mathbb{N}$ .
\end{assertion}
```

Axiom 3.2.3. Let $\varphi(n)$ a property on **natural numbers**. If

1. $\varphi(0)$ and
 2. if $\varphi(m)$ holds for some m , then $\varphi(m + 1)$ also holds,
- then $\varphi(n)$ holds for all $n \in \mathbb{N}$.

Exercise

Annotate the above by:

1. **Variables** with appropriate **notations** for φ , m and n , and
2. marking up the second premise (“if $\varphi(m)$ holds for some...”) in text mode as the formula $\forall m. \varphi(m) \Rightarrow \varphi(m + 1)$ using the **semantic macros** `\forall` (which we saw earlier already) and `\Rightarrow` (**implication**) from [sTeX/Logic/General]mod/syntax?Implication. The text fragments that should be highlighted are “if” and “then”.
3. marking up the conclusion (“ $\varphi(n)$ holds for all $n \in \mathbb{N}$ ”) in text mode as the formula $\forall n. \varphi(n)$. The text fragment that should be highlighted is “for all”.

Lösung: Can be found in [sTeX/MathTutorial]mod/NatTheorems.en.tex

So how can we teach **MMT** the semantics of this statement? Here's what we can do:

1. As with the simpler assertions (and hence the name), the *conclusion* of the assertion can be marked up with `\conclusion`.

2. As with `sdefinition`, we can mark `variables` as *bound* (using either `bind` in the `\vardef` or `\varbind`). If a `symbol` that can act as a `universal quantifier` is in scope, `variables` marked as bound are abstracted away using that `symbol`.
3. Similarly to `\conclusion`, `premises` can be marked up as such using the `\premise` macro. If a `symbol` is in scope that can act as an `implication`, that will be used to connect the premise(s) to the conclusion.

Hence, if we mark the variable φ as bound and use `\premise` and `\conclusion` (see [sTeX/MathTutorial]mod/NatTheorems.en.tex), we can inspect the OMDoc tab in the HTML preview again and see that MMT has now constructed the proposition (Figure 26).

```
> Assertion induction axiom ⊢ ∀φ:N→Prop.φ(0)⇒( ∀m:N.φ(m)⇒φ(m+1) )⇒( ∀n:N.φ(n) )
```

Figure 26: The Induction Axiom in OMDoc

3.2.3 Proofs



sTeX provides the `sproof` environment for marking up *proofs*. The markup mechanism for `sproof` is still highly experimental and likely subject to change in the near future. As such, we omit a closer explanation of its usage until the syntax and functionality have sufficiently stabilized.

3.3 Mathematical Structures

A common concept in mathematics is that of a `mathematical structure` – a *tuple* of interdependent components. For example: A `monoid` is a `structure` $\langle M, \circ, e \rangle$ such that certain axioms hold; where M is a set, \circ is a binary operation, and $e \in M$.

From a representational perspective, this is particularly interesting: M , \circ and e in the above are not `symbols` in the same way that the previous `symbols` we considered were – they don't represent definite objects. Instead, they are *components* of some other object, namely a monoid; where a *particular* monoid could either be a fixed object (such as $\langle \mathbb{Z}, +, 0 \rangle$) or an *indefinite* monoid; i.e. a `variable`. We call the components of a `mathematical structure` `fields`.

In this section, we will discuss how to declare and use `mathematical structures` in sTeX, build them up modularly, and connect them among each other to avoid duplication.

We will do so by considering *lattices* both algebraically and order-theoretically, and identify the two perspectives.

3.3.1 Declaring and Using Structures

The simplest kinds of `structures` are *magmas* and *(directed) graphs*, so we might as well start there:

Definition 3.3.1. A **magma** is a **structure** $\langle U, \circ \rangle$, where U is a **collection** and \circ a binary operation $U \times U \rightarrow U$.

The obvious start is to create a new **module** `Magma`. Within this **module**, we import the **Functions module** so we can later assign a **type** to the operation. We can then use the **mathstructure environment**, that creates a new symbol “**magma**”:

```
\begin{smodule}{Magma}
  \importmodule[sTeX/MathBase/Functions]{mod?Function}
  \begin{mathstructure}{magma}
    ...
  \end{mathstructure}
\end{smodule}
```

mathstructure behaves very similarly as **smodule** – within the **environment**, we can declare new **symbols**, **notations** and all that.

So within the **mathstructure**, we can add **symbols** for the two fields U and \circ :

```
\symdef{univ}[name=universe,type=\collection]{U}
\symdef{op}[name=operation,args=a,assoc=bin,
  type=\funspace{\univ,\univ}\univ
]{\argsep{\#1}{\mathbin{\maincomp{\circ}}}}
```

Once we close the **environment** (with `\end{mathstructure}`), the **symbols** are “gone”. However, we now have a new **symbol** “**magma**” with **semantic macro** `\magma`. Its usage is somewhat more complicated than “normal” **semantic macros**, but one thing we *can* do with it now is `$\magma!`, which will produce $\langle U, \circ \rangle$.

Notably however, the `\magma` **macro** is already available *within* the **mathstructure environment** as well.

This allows us to provide an **sdefinition** using the **semantic macros** declared in the **structure**:

Example 11

Input:

```
File [sTeX/MathTutorial]algebra/Magma.en.tex
7  \begin{mathstructure}{magma}
8    \symdef{univ}[name=universe,type=\collection]{U}
9    \symdef{op}[name=operation,args=a,assoc=bin,
10      type=\funspace{\univ,\univ}\univ
11      {\argsep{\#1}{\mathbin{\maincomp{\circ}}}}]
12
13  \begin{sdefinition}[for={magma,univ,op}]
14    A \definename{magma} is a \sr{mathstruct}{structure} $\magma$,
15    where $\univ$ is a \sn{collection} and $\op$ is
16    a binary operation $\funspace{\univ,\univ}\univ$.
17  \end{sdefinition}
18 \end{mathstructure}
```

Output:

Definition 3.3.2. A **magma** is a **structure** $\langle U, \circ \rangle$, where U is a **collection** and \circ a binary operation $U \times U \rightarrow U$.

Instantiating Structures

More importantly however, we can now declare a `variable magma`, using the optional `return=` argument. For example, we can now do

```
\vardef{vM}[name=M,return=\magma]{M}
```

and we get the semantic macro `\vM` with which we can do the following:

Syntax	Result
<code>\$\vM\$</code>	M
<code>\$\vM{}\$</code>	$\langle U_M, o_M \rangle$
<code>\$\vM{univ}\$</code>	U_M
<code>\$\vM{op}!\$</code>	o_M
<code>\$\vM{op}{a,b,c}\$</code>	$a o_M b o_M c$

In other words: Given a `symbol` or `variable` with `semantic macro` `\foo` and `return=\struct`, then `\foo{<fn>}` behaves like the `semantic macro` `\fn` *within* the `mathstructure environment` for `struct` – but instantiated for the specific instance `foo`.

By default, `STEX` attaches the `symbol`'s (or `variable`'s) `operator notation` as a subscript suffix to the notation component marked with `\maincomp` – e.g., since the “`\circ`” in the `notation` for `op` is marked with `\maincomp`, doing `$\vM{op}{a,b}$` ultimately outputs a `\circ_{\vM{op}}{a,b}`. Hence, we get $a o_M b$.

We can change the way the `\maincomp` notation component is modified, by using the optional argument `copm=` in the `semantic macro` for the `mathematical structure`. For example, to not change it at all, we can do:

```
\vardef{vM}[name=M,return={\magma[comp=##1]}]{M}
```

...or to suffix it with a `,` we can do

```
\vardef{vMp}[name=Mp,return={\magma[comp=##1']}]{M'}
```

This allows us to do things like:

```
Let $\vM! := \vM{}$ and $\vMp! := \vMp{}$. \sns{magma}. Then...
```

yielding

Let $M := \langle U, o \rangle$ and $M' := \langle U', o' \rangle$ magmas. Then...

We can also *assign* fields to (arbitrary) expressions, by doing `name=<tex>` in square brackets. For example we can do the following:

```
\vardef{vA}[type=\collection]{A}
\vardef{vM}[name=M,return={\magma[comp=##1][univ=\vA]}]{M}
\vardef{vMp}[name=Mp,return={\magma[comp={##1'}][univ=\vA]}]{M'}
```



```
Let $\vM! := \vM{}$ and $\vMp! := \vMp{}$. \sns{magma} on $\vA$....
```

Let $M := \langle A, o \rangle$ and $M' := \langle A, o' \rangle$ magmas.

Of course, we can also use `return=` with `variable` sequences – for example:

```
\varseq{vMs}[name=M,return={\magma[comp={##1}_{##1}],op=(M_i)_1^n}]
{1,\ellipses,n}{\maincomp{M}_{##1}}
Let $\vMs! := \vMs{i}_1^n$ a sequence of \sns{magma}...
```

Let $(M_i)_1^n := \langle U_i, o_i \rangle_1^n$ a sequence of **magma**s...

Note that in the above, it seems that using #1 in the **return** argument is allowed. Indeed, it is - the **return** statement takes the same arguments as the **semantic macro** itself does and is appropriately instantiated. Since the first (and only) argument to the sequence **\vMs** is the index, when doing **\vMs{i}...** the #1 in the **return**-statement will be replaced by **i**.

Also, note that if we want to produce M_i – i.e. the **magma** at index **i** in the sequence, we can do **\vMs{i}!**.



Think of the ! as a “stop sign” - if the expression up to the ! has an associated presentation, the ! tells **STEX** to “stop eating arguments” and present whatever it has until now.

3.3.2 Extending Structures and Axioms

It is extremely common to “build up” **structures** in a hierarchical manner by adding new fields or axioms: A *semigroup* is an associative magma. A *band* is an idempotent semigroup. A *monoid* is a semigroup with a unit. A *partial order* is an antisymmetric preorder.

We alluded to the fact earlier, that the **mathstructure** environment behaves like an **smodule** – that is literally true: Every **mathstructure** **foo** in a **module** **FooMod** is in fact also a **module** **?FooMod/foo-module**. We can therefore easily extend **structures** using **\importmodule{...?FooMod/foo-module}** – but extending **structures** is so common, and using **\importmodule** tiring, that there is a shortcut: the **extstructure** environment. It takes as second argument a comma-separated list of **structure** names. That allows us to easily define **semigroups**:

Example 12

Input:

```
File [sTeX/MathTutorial]algebra/Semigroup.en.tex
8 \begin{extstructure}{semigroup}{magma}
9   \begin{sdefinition}
10     A \definename{semigroup} is a \sn{magma} \$\semigroup!$,
11     where \inlineass[name=associative axiom]{
12       \conclusion{\isassociative{\op!}}}.
13   }
14 \end{sdefinition}
15 \end{extstructure}
```

Output:

Definition 3.3.3. A **semigroup** is a **magma** $\langle U, o \rangle$, where o is **associative**.

Note our usage of **\inlineass** to generate a new **symbol** for the **associative axiom**. If we look at the **OMDoc** tab in the **HTML** preview window, we can see the output in [Figure 27](#).

So **MMT** has decided that our statement is an *axiom*.



Figure 27: Axioms in OMDoc

Conservative Extensions

For **structures**, there is a *critical* distinction between *defined* and *undefined symbols*; and analogously between *theorems* and *axioms*.

Remember that **structures** are more like *templates* that are *instantiated* by particular objects. An *undefined* field in a **structure**, in that sense, is like an *obligation*: If something is supposed to be a **semigroup**, it *has to* have a **universe**, an **operation** and the **operation** needs to satisfy the **associative axiom**.

Defined fields on the other hand have a *definiens* on the basis of the remaining fields – they don't need to be explicitly provided for something to instantiate the **structure**; if all the *undefined* fields are provided, the *defined* ones we get “*for free*”.

The same holds for *theorems*: If a statement is *provable* from the axioms, then we don't need to explicitly prove it to hold for some particular instance – we have a proof already, provided the axioms hold.

The relation between axioms and theorems is not just analogous to that between undefined and defined **symbols**: It is the very same. Remember the **judgments as types** paradigm?

StEx For a **proposition** P , an assertion in **StEx** induces a **symbol** of type $\vdash P$. Without a proof, this **symbol** is *undefined* – and hence an *axiom*. A *proof* for P is a specific term of type $\vdash P$ – i.e. a potential *definiens*. To prove an assertion turns it into a *theorem*, which is to say that the **symbol** can be *defined*.

One consequence of this is: Extending a **structure** only by *defined* fields does not actually (conceptually) introduce a *new structure* – every instance of the old one *should* also be an instance of the new one. The new fields are basically just “syntactic sugar”.

There is a name for extending a **structure** only by defined fields (or theorems): A *conservative extension*.

StEx provides the **extstructure*** environment for that purpose. Unlike **extstructure**, it does *not* take a name (technically, **StEx** generates one internally). Instead, conceptually **extstructure*** modifies the extended **structure** directly, rather than generating a new **structure**. The caveat however is, that every **symbol** introduced in an **extstructure*** **must** be defined.

Consider the following conservative extension:

Example 13

Input:

```
File [sTeX/MathTutorial]algebra/MagmaSquare.en.tex
7 \begin{extstructure}{magma}
8   \begin{sdefinition}[macro=sq,args=1]
9     \notation{sq}[op=\cdot^2]{\#1^\text{comp} 2}
10    \vardef{va}[name=a,type=\univ,bind]{a}
11    Let $ \inset{va}{\univ} $. We define
12    $\defnotation{\sq{va}} := \definiens{\op{va,va}}$.
13  \end{sdefinition}
14 \end{extstructure}
```

Output:

```
Definition 3.3.4. Let  $a \in U$ . We define  $a^2 := a \circ a$ .
```

Via `\definiens`, the new symbol `sq` is now *defined* (note the `macro=` argument, that generates a `semantic macro` as well). Whenever we import the containing `module`, we now have an additional field `sq` in (any extension of) `magma` – e.g., as `semigroup` extends `magma`, the following is now valid:

```
\usemodule[sTeX/MathTutorial]{algebra?MagmaSquare}
\vardef{vsg}[name=S,return=\semigroup]{S}
$\vsg{sq}{a}$
```

...producing a^2 .

3.3.3 Nesting Structures and `\this`

A perhaps not too surprising, but a notable aspect of `structures` is that fields themselves can be instances. This is important for example for implementing *vector spaces*, but can also be used to bundle things that are not normally thought of as `structures`, such as objects with certain defining properties.

Take as an example, the notion of a (`magma`) homomorphism:

```
Definition 3.3.5. Let  $M_1 = \langle U_1, \circ_1 \rangle$  and  $M_2 = \langle U_2, \circ_2 \rangle$  magmas. A magma homomorphism is a function  $F : U_1 \rightarrow U_2$  such that  $F(a \circ_1 b) = F(a) \circ_2 F(b)$  for all  $a, b \in U_1$ .
```

So a `homomorphism` is a `function` with certain properties. And `structures` can be used to “bundle” the `function` itself with both the `magmas` on whose universes the `function` operates, as well as the *axiom* that *makes* it a `homomorphism`. After all, considered as a mere `function`, $F : U_1 \rightarrow U_2$ contains no information about the operation with respect to which it is homomorphic.

The first thing to note is that we can provide `mathstructure` with an optional argument for a `name` distinct from the name of its `semantic macro`. We then add two fields that `return` `magmas`. So far, so unexciting:

```
\begin{mathstructure}{magmahom}[magma homomorphism]
\symdef{dom}[name=domain,return={\magma[comp={##1}_1]}]{M_1}
\symdef{cod}[name=codomain,return={\magma[comp={##1}_2]}]{M_2}
```

For the `function` itself, we know how to give it a meaningful `type`, already:

```
\symdef{f}[type=\funspace{\dom{univ}}{\cod{univ}},args=1]{??}
```

...but what should its `notation` be? Ideally we would want it to just be the `notation` of whatever particular instance it is – in informal mathematics, we rarely distinguish notationally between a `homomorphism` and its underlying `function` (to the point where it's not clear, whether *informally* the distinction is even meaningful). Similarly, we rarely distinguish e.g. between a `magma` (or semigroup, monoid, group, ring, vector space,...) and its underlying universe.

This is where `\this` comes into play (pun intended). Within an `mathstructure` or `exstructure`, or in the context of a particular instance of one, `\this` represents “the” instance.

We can set it in the context of `mathstructure` as a further optional argument; e.g.

```
\begin{mathstructure}{magmahom}[magma homomorphism,this=F]
```

and then use `\this` in the `notation` for the `function`. We can further provide the `homomorphism condition` as an axiom using `\inlineass`:

Example 14

Input:

```
File [sTeX/MathTutorial]algebra/Homomorphism.en.tex
9 \begin{mathstructure}{magmahom}[magma homomorphism,this=F]
10   \symdef{dom}[name=domain,return={\magma[comp={##1}_1]}]{M_1}
11   \symdef{cod}[name=codomain,return={\magma[comp={##1}_2]}]{M_2}
12   \symdef{f}[op=\this,args=1,
13     type=\funspace{\dom{univ}}{\cod{univ}}
14   ]{\this \dobrackets{#1}}
15 
16 \begin{sdefinition}[for={magmahom,dom,cod,f}]
17   \vardef{va}[name=a,type=\dom{univ}]{a}
18   \vardef{vb}[name=b,type=\dom{univ}]{b}
19   Let $\dom!=\dom{}$ and $\cod!=\cod{}$ \sns{magma}.
20   A \definename{magmahom} is a function
21   $f!\{\dom{}\}\{\cod{}\}$ such that
22   \inlineass[name=homomorphism condition]{\conclusion{\forall{{
23     $arg[2]\{\leq{{
24       f\{\dom{}\}{va,vb}}, \cod{op}\{f\{va\},f\{vb\}\}
25     }\}\} \comp{for all} $arg[1]\{\inset{va,vb}{\dom{univ}}\}}}.
26   }}}
27 \end{sdefinition}
28 \end{mathstructure}
```

Output:

Definition 3.3.6. Let $M_1 = \langle U_1, \circ_1 \rangle$ and $M_2 = \langle U_2, \circ_2 \rangle$ magmas. A **magma homomorphism** is a function $F : U_1 \rightarrow U_2$ such that $F(a \circ_1 b) = F(a) \circ_2 F(b)$ for all $a, b \in U_1$.

Now if we instantiate our `magma homomorphism`:

```
\vardef{vh}[name=H,return={\magmahom[this=H]}]{H}
```

Here is a list of what we can do now:

Syntax	Result
$\$\\vh!$$	H
$\$\\vh{}$$	$\langle M_1, M_2, H \rangle$
$\$\\vh{f}!$$	H
$\$\\vh{f}{a}$$	$H(a)$
$\$\\vh{dom}!$$	M_1
$\$\\vh{cod}{}$$	$\langle U_2, o_2 \rangle$
$\$\\vh{cod}{univ}$$	U_2
$\$\\vh{dom}{op}!$$	o_1
$\$\\vh{cod}{op}{a,b,c}$$	$a o_2 b o_2 c$

Note how – as one would expect – we can treat $\backslash vh\{dom\}$ and $\backslash vh\{cod\}$ like any other instance of `magma`.

Note that some of the outputs in the above table are probably not quite what we want. Determining the precise typesetting of an expression involving *nested paths* of fields is difficult, to say the least (e.g., what exactly should `\this` refer to in a deeply nested sequence of fields?).

Using instances within `structures` is still very useful; at the very least when defining `structures`. When subsequently *using structures*, however, accessing fields of fields (of fields (of ...)) of an instance should be avoided.

Luckily, there is rarely a need for doing so – in practice, those fields we might want to access in such a way, we usually also want to provide specific `notations` and talk about independently of the “containing” instance, such that introducing a new `variable` (or `symbol`), and assigning the corresponding field to that `variable`, makes considerably more sense. And subsequently using the `variable` is easier than concatenating `{...}`, too.

3.4 Complex Inheritance and Theory Morphisms

We are starting to approach seriously experimental territory.

While the theory behind all the following is relatively well understood, and their implementation in `MMT` is mature, the same can not be said out the implementation in `sTeX`.

There are still kinks to be ironed out, but feel free to experiment.

We now have all the tools available to progress towards something more interesting. Here is a list of documents with respective `modules` and `symbols` we will build on in the following:

`[sTeX/MathTutorial]props/Idempotent.en.tex`

Definition 3.4.1. Let $e \in A$ and $\circ : A \times A \rightarrow A$. e is called **idempotent** with respect to \circ , if $e \circ e = e$.

Definition 3.4.2. The operation $\circ : A \times A \rightarrow A$ is called **idempotent**, if every element $a \in A$ is **idempotent** with respect to \circ .

[sTeX/MathTutorial]props/Distributive.en.tex

Definition 3.4.3. Let $\odot : B \times A \rightarrow A$ and $\oplus : A \times A \rightarrow A$. We say \odot **distributes over** \oplus , if $b \odot (a_1 \oplus a_2) = (b \odot a_1) \oplus (b \odot a_2)$ for all $a_1, a_2 \in A$ and $b \in B$.

[sTeX/MathTutorial]props/Absorption.en.tex

Definition 3.4.4. Let $\odot : A \times B \rightarrow A$ and $\oplus : A \times B \rightarrow B$. We say \odot **absorbs** \oplus , if $a_1 \odot (a_1 \oplus b) = a_1$ for all $a_1 \in A$ and $b \in B$.

[sTeX/MathTutorial]algebra/Band.en.tex

Definition 3.4.5. A **band** is an **idempotent semigroup**.

[sTeX/MathTutorial]algebra/Semilattice.en.tex

Definition 3.4.6. A **semilattice** is a **commutative band**.

[sTeX/MathTutorial]props/Reflexive.en.tex

Definition 3.4.7. A binary relation R on A is called **reflexive**, if $R(a, a)$ for all $a \in A$.

[sTeX/MathTutorial]props/Symmetric.en.tex

Definition 3.4.8. A binary relation R on A is called **symmetric**, if $R(a, b)$ implies $R(b, a)$ for all $a, b \in A$.

[sTeX/MathTutorial]props/Transitive.en.tex

Definition 3.4.9. A binary relation R on A is called **transitive**, if $R(a, b)$ and $R(b, c)$ implies $R(a, c)$ for all $a, b, c \in A$.

[sTeX/MathTutorial]props/Antisymmetric.en.tex

Definition 3.4.10. A binary relation R on A is called **antisymmetric**, if $R(a, b)$ and $R(b, a)$ implies $a = b$ for all $a, b \in A$.

[sTeX/MathTutorial]orders/Graph.en.tex

Definition 3.4.11. A **directed graph** is a **structure** $\langle U, R \rangle$, where U is a **collection** and R a binary relation on U .

Definition 3.4.12. An **(undirected) graph** is a directed graph $\langle U, R \rangle$, where R is **symmetric**.

[sTeX/MathTutorial]orders/Preorder.en.tex

Definition 3.4.13. A structure $\langle U, \leq \rangle$ is called a **preorder** (or **quasiorder**, or **preordered set**; in short **proset**), if \leq is **reflexive** and **transitive**.

[sTeX/MathTutorial]orders/Poset.en.tex

Definition 3.4.14. A preorder $\langle U, \leq \rangle$ is called a **partial order** (or **poset**), if \leq is **antisymmetric**.

[sTeX/MathTutorial]orders/InfSup.en.tex

Definition 3.4.15. Let $\langle U, \leq \rangle$ a poset. An element $a \in U$ is called an **infimum** or **greatest lower bound** of x_1 and x_2 , if $a \leq x_1$, $a \leq x_2$, and for any x with $x \leq x_1$ and $x \leq x_2$, we have $x \leq a$.

Definition 3.4.16. Let $\langle U, \leq \rangle$ a poset. An element $a \in U$ is called a **supremum** or **least upper bound** of x_1 and x_2 , if $x_1 \leq a$, $x_2 \leq a$, and for any x with $x_1 \leq x$ and $x_2 \leq x$, we have $a \leq x$.



Note that **infima** and **suprema** are more generally defined on *sets* of elements. Doing so in **STEX** is significantly more complicated *for now*, and will require some amount of research to make convenient – especially if we want to subsequently define *operators* on pairs of elements, as below. We therefore opt for the simpler version where it is defined as binary from the get go.

[sTeX/MathTutorial]orders/MeetJoinSemilattice.en.tex

Definition 3.4.17. A poset $\langle U, \leq \rangle$ is called a **meet semilattice** if for every two elements a, b the infimum $a \wedge b$ exists.

Definition 3.4.18. A poset $\langle U, \leq \rangle$ is called a **join semilattice** if for every two elements a, b the supremum $a \vee b$ exists.

Definition 3.4.19. An **(order) semilattice** is a meet and join semilattice.

Exercise

Try to implement all of the above yourself!

3.4.1 Glueing Structures Together

We now want to progress towards **lattices**, i.e. the following:

Definition 3.4.20. A lattice is a structure $\langle U, \wedge, \vee \rangle$ such that $\langle U, \wedge \rangle$ and $\langle U, \vee \rangle$ are semilattices, and \vee absorbs \wedge and vice versa; i.e. $a \vee (a \wedge b) = a$ and $a \wedge (a \vee b) = a$.

The operations \wedge and \vee are called **meet** and **join**, respectively.

So we make a new `module`, open an `extstructure environment` and... realize two problems:

1. We can't just extend `semilattice`: We need *two* copies of `semilattice` that share a universe, and importing `semilattice` twice is of course redundant.
2. We also want to *rename* the operations of the two `semilattices` to be subsequently called `join` and `meet`.

What we need is a way to *inherit* from `semilattice` while a) *modifying* the `symbols` therein, and b) not be `idempotent` – i.e. two imports from the same `structure` or `module` should not be identified. We can do that with the `\copymod` macro, which takes three arguments:

1. A *name* for the copy,
2. the `structure` or `module` to copy, and
3. a comma-separated list of renamings and redefinitions of the `symbol`. $\langle symbol \rangle = \langle def \rangle$ redefines $\langle symbol \rangle$, $\langle symbol \rangle @ \langle newname \rangle$ renames it, $\langle symbol \rangle = \langle def \rangle @ \langle newname \rangle$ (or $\langle symbol \rangle @ \langle newname \rangle = \langle def \rangle$) does both.

In our case, we want two copies of `semilattice`, which we will call `meetsl` and `joinsl`. In the first copy, we only want to rename `op` to `meet`. In the second, we want to rename `op` to `join`, and *also* redefine the universe to be the one from `meetsl`:

```
\copymod{meetsl}{semilattice}{
    op @ meet
}
\copymod{joinsl}{semilattice}{
    univ = \univ,
    op @ join
}
```

You might have already noticed some problem with that – which of the two universes does `\univ` refer to now? (They are *defined* as equal, but `LATEX` does not know that!) Or which of the two `commutative axioms` does “`commutative axiom`” refer to now? Everything is ambiguous now!

Not really - if you have wondered why the `\copymod` takes a *name* as argument: The name is prefixed to every `symbol` name. Hence, the `universe` in `joinsl` is now called `joinsl/universe`, and the one in `meetsl` is called `meetsl/universe`. Furthermore, `\copymod` by default generates no `semantic macros` for any of the imported `symbols` – except for those renamed with `@`. In fact, what the `@` syntax actually does, is to generate a `semantic macro` by that name. If we want to change the *name* (that is shown when using `\symname` et al), we add that new name in square brackets. Hence, what we really want to do is:

```
\copymod{meetsl}{semilattice}{
    univ @ univ,
    op @ [meet]meet
}
\copymod{joinsl}{semilattice}{
    univ = \univ,
    op @ [join]join
}
```

This now gives us two copies of `semilattice`, generates semantic macros `\univ` for `meetsl/universe`, `\meet` for `meetsl/op` and `\join` for `joinsl/op`, and renames `meetsl/op` to `meet` and `joinsl/op` to `join`.

That allows us to then add the `absorption` axioms, an `sdefinition` for `lattice` and subsequently `$\lattice!` produces $\langle U, \wedge, \vee \rangle$, with all axioms inherited (see [sTeX/MathTutorial]algebra/Lattice.en.tex).

3.4.2 Realizations

A very common situation we find in connection with `mathematical structures` is that “every *this* is a *that*” (or the concrete case “*this* is a *that*”).

With what we did so far, we are in this situation regarding the algebraic definition of `semilattices` and the order-theoretic one (exemplary `meet semilattice`).

In MMT parlance, this corresponds to a `total (implicit) theory morphism` from “*that*” to “*this*”.

In `STEX` words, we want to inherit from “*that*” by assigning all the `symbols` in “*that*” to concrete terms. In our case:

```
[sTeX/MathTutorial]algebra/SemiLatticeOrder.en.tex
```

Definition 3.4.21. Let $\langle U, \circ \rangle$ a `semilattice`. We let $a \leq b$ iff $a \circ b = a$.

Theorem 3.4.22. $\langle U, \leq \rangle$ is a `meet semilattice`.

Proof: We need to prove the following

`reflexivity` $a \leq a$: We need to show $a \circ a = a$. Follows from the `idempotent axiom`.

`antisymmetry` $a \leq b$ and $b \leq a$ implies $a = b$: Assume $a \circ b = a$ and $b \circ a = b = a \circ b$ (by the `commutative axiom`). Hence, $a = b$

`transitivity` If $a \leq b$ and $b \leq c$, then $a \leq c$. : Assume $a \circ b = a$ and $b \circ c = b$. Then $a \circ c = (a \circ b) \circ c = a \circ (b \circ c) = a \circ b = a$. Hence, $a \leq c$.

$a \circ b$ is the `infimum` of $\{a, b\}$: By definition (and the `commutative axiom`), $a \circ b \leq a$ and $a \circ b \leq b$. We need to show, that if $x \leq a$ and $x \leq b$, then $x \leq a \circ b$. Assume $x \circ a = x$ and $x \circ b = x$. Then $x \circ (a \circ b) = (x \circ a) \circ b = x \circ b = x$. Hence $x \leq a \circ b$

So to be precise, we want to provide *definientia* for all undefined `symbols` in `meet semilattice` (i.e. the `relation` and `meet`) and *proofs* for all `axioms` (`reflexive axiom`, `antisymmetric axiom`, `transitive axiom`, and `infimum axiom`), and by so obtain the fact that every `semilattice` is a `meet semilattice`.

For that purpose, we have the `\realize` macro. It behaves like `\copymod`, but does not take a name, and additionally requires that all undefined fields get assigned. So we could do the following:

Example 15

Input:

```

File [sTeX/MathTutorial]algebra/SemiLatticeOrder1.en.tex

8 \begin{extstructure*}{semilattice}
9   \realize{meets1}[
10     univ = \univ,
11     meet = \op!,
12     rel @ [order]order = \map{a,b}{\eq{\op{a,b},a}},
13     reflexive axiom = trivial,
14     transitive axiom = trivial,
15     antisymmetric axiom = trivial,
16     infimum axiom = trivial
17   }
18 \end{extstructure*}
19
20 \vardef{mysl}[return=\semilattice]{S}
21 $mysl{order}{a,b} \qquad \mysl{}[\univ,\op,order]$
```

Output:

$$a \leq_S b \quad \langle U_S, \circ_S, \leq_S \rangle$$

As we can see, we can now access the field `order`, which is renamed from `relation` in `meet semilattice` and also has the desired definiens in `MMT`. But of course this approach is very “declarative”: We do all the assigning in one `macro`, rather than narratively as what they *should* be: definitions and proofs.

If we want to achieve the more narrative version at the beginning of the subsection, we can use the `realization environment` instead. It behaves like the `\realize` macro, but allows us to do the assignments and renamings individual somewhere in the body of the `environment`, interleaved with arbitrary text. Additionally, within the `environment`, all `STEX` features that introduce *definientia* (like the `\definiens` macro) induce assignments instead.

To declaratively rename or assign fields, we can then use the `\assign` and `\renamedecl` macros instead. That allows us to do the following instead:

```

\begin{realization}{meets1}
\assign{univ}{\univ}
\assign{meet}{\op!}
\renamedecl{rel}[order]{order}
...
```

...and then use text to do the remaining assignments. For example, we can use the `sdefinition environment` to assign `rel` to the desired definiens:

```

\usestructure{meets1}
\begin{sdefinition}[for=order]
\varbind{va,vb}
Let $ \semilattice! [\univ,\op] $ a \sn{semilattice}.
We let $ \rel{\va,\vb} $.
iff $ \definiens{\eq{\op{\va,\vb},\va}} $.
\end{sdefinition}
```

And now `STEX` will use the `\definiens` to assign $a, b \mapsto a \circ b = a$ to the relation of `meet semilattice`.

Analogously, we can use the `sproof` and `subproof` environments to produce “definientia” (i.e. proofs) for the axioms (see [sTeX/MathTutorial]algebra/SemiLatticeOrder.en.tex)

Chapter 4

Extensions for Education

The last two chapters have shown generic markup and semantization facilities in $\text{\texttt{STEX}}$. As said before, investments in semantic markup pay off, iff the impact of a document is high, e.g. if there are many more readers than authors or if the semantic services afforded by the semantic markup can help reduce the help readers need to understand the material.

Educational documents constitute one category of high-impact documents which are supported by the $\text{\texttt{STEX}}$ ecosystem, we will cover them here. In fact, educational documents have been one of the initial document categories $\text{\texttt{STEX}}$ has been developed for. The idea is that if we can mark up the meaning and didactic role of learning objects, we can base learning support services on that and embed them into the documents.

Another reason educational documents are particularly interesting is that in a sense all academic communication is educational, as all documents try to “teach” the reader new concepts and results.

Concretely, we cover a document class for combining slides and course notes ([section 4.1](#)) and functionality for marking up problems and exercises ([section 4.2](#)) and for marking up homework assignments and exams ([section 4.3](#)).

4.1 Slides and Course Notes

TODO⁶

Contents

4.2 Problems and Exercises

4.2.1 Background

Problems/exercises are text fragments that contain a task assigned to the learner – e.g. computing the value of a specified quantity, simplifying an expression, modeling a described situation in a mathematical structure, or judging the veracity of a given statement. Problems/exercises are used in very different contexts, in

⁶TODO: `notesslides.sty`

- *homework assignments and formal exams*, where they are graded and points are awarded for course credit,
- *self-study materials* that allow learners explore applications of some concepts and/or practice,
- *automated tutoring systems* to determine learner competencies to trigger computer supported learning services.

In all of these contexts, diagnosis the correctness or suitability of an answer plays a great role, e.g. for grading, the generation of feedback, or the suggestion of remedial actions that may “heal” any diagnosed competency gaps or misconceptions. To support that we might want to specify “answer classes” that classify answers received for automating/triggering feedback and grading.

There are various types of problems/exercises in practical use. They are mainly distinguished by how they support diagnosis of student answers: there are

- open problems, where expected answers are free-form, and classification into answer classes is usually a manual process; see [subsection 4.2.3](#)
- single/multiple choice questions where the answer classes and thus feedback/grading correspond to the selection pattern of items; see [subsection 4.2.5](#).
- fill-in-the-blanks questions where we may want to specify how the answer string is interpreted; see [subsection 4.2.6](#).

Additionally, Problems and exercises often come with text fragments that serve auxiliary functions: hints, notes, and and grading specifications. Furthermore, we can specify how long solving a given problem is estimated to take and how many points will be awarded for a perfect solution – e.g. in an exam. See [subsection 4.2.9](#) for details.

4.2.2 The Package, Options, and Configuration

The `problem` package provides functionality for marking up problems and exercises as semantic sources from which the presentations for the various contexts can be generated. E.g. documents without solutions for paper or online exams, and the corresponding exams with master solutions for exam reviews. Similarly with/without hints, or points. Their visibility is specified in the options of the `problem` package, which can be used in any L^AT_EX class. The following is a typical preamble for a problem file:

```
\documentclass[lang=de]{article}
\usepackage[solutions,hints,pts,min]{problem}
```

Here we have specified the options `solutions` (solutions should be shown), `hints` (hints should be given), `pts` (display the points awarded for solving the problem?), `min` (display the estimated minutes for problem solving). Leaving out the options would make the corresponding functionality invisible.

The `problem` package generates content and labels according to the language specified in the `lang` key of the main document¹, these can be localized by in the files `1df/problem-<lang>.1df`⁷.

¹EdNOTE: MK: document the attribute somewhere

⁷The current crop of language definition files is quite limited. Please feel free to contribute the to the localization effort

The `problem` package also supplies the `test` option, which specifies that the content is intended for use in an assessment situation, where certain additional content (e.g. exam legalse) should be shown while other content (e.g. solutions) should not be.

4.2.3 (Open) Problems

The main environment provided by the `problem` package is (surprise surprise) the `sproblem` environment. It is used to mark up problems and exercises. The following example shows the main functionality:

Example 16

Input:

```
File [sTeX/Documentation]tutorial/ext/simple-problem.en.tex
1 \begin{document}
2 \begin{sproblem}[id=prob.elefants,pts=10,min=2,title=Fitting Elefants]
3   How many Elefants can you fit into a Volkswagen beetle?
4   \begin{hint}
5     Think positively, this is simple!
6   \end{hint}
7   \begin{exnote}
8     Justify your answer
9   \end{exnote}
10  \begin{gnote}
11    if they do not give the justification deduct 5 pts
12  \end{gnote}
13  \begin{solution}
14    Four, two in the front seats, and two in the back.
15  \end{solution}
```

Output:

Exercise (Fitting Elefants)

How many Elefants can you fit into a Volkswagen beetle?

Lösung: Four, two in the front seats, and two in the back.

The `sproblem` environment takes an optional key/value argument with the keys `id` as an identifier that can be reference later, `pts` for the points to be gained from this exercise in homework or quiz situations, `min` for the estimated minutes needed to solve the problem, and finally `title` for an informative title of the problem.

The `sproblem` can contain several `solution` environments, which take the well-known `id`, `style`, and `title` attributes, and also the `testspace` and `answerclass`. The former

The additional functionality is specified in the `hint` `exnote` (notes in exercises), `solution`, and `gnote` (grading notes) environments. Here, the first three are shown whereas the grading notes are hidden, since the corresponding option was not given in the `\usepackage[...]{problem}`. All of these environments can occur any number

of times in the `sproblem` environment. The `solution` environment takes an optional argument that is interpreted as the identifier.

4.2.4 Structured Problems

Problems can be structured into subproblems via the `subproblem` environment. It takes the same arguments and allows the same content model as as `sproblem`.

Example 17

Input:

```
File [sTeX/Documentation]tutorial/ext/structured-problem.en.tex
1 \begin{document}
2 \begin{sproblem}[id=prob.elefants-mod,title=Fitting Elefants Modularly]
3 Consider the problem of fitting elephants into a Volkswagen beetle.
4 \begin{subproblem}[pts=1,min=2]
5 Estimate the number of elephants on the front seats.
6 \begin{solution}
7 Two on each side one.
8 \end{solution}
9 \end{subproblem}
10 \begin{subproblem}[pts=1,min=2]
11 Estimate the number of elephants on the back seats.
12 \begin{solution}
13 Three little elephants.
14 \end{solution}
15 \end{subproblem}
```

Output:

Exercise (Fitting Elefants Modularly)

Consider the problem of fitting elephants into a Volkswagen beetle.

1. Estimate the number of elephants on the front seats.

Lösung: Two on each side one.

2. Estimate the number of elephants on the back seats.

Lösung: Three little elephants.

3. What is the total number?

Lösung: A family of five; we do not want elephants in the trunk.

By default, subproblems are numbered subordinate to the “mother problem”. The `problem` package does not make any assumptions about whether subproblems can be used independently from their sister problems, or on order requirements.

4.2.5 Single/Multiple Choice Blocks

Multiple choice blocks can be formatted using the `mcb` environment, in which single choices are marked up with `\mcc` macro.

`\mcc` `\mcc[<keyvals>]{<text>}` takes an optional key/value argument `<keyvals>` for choice metadata and a required argument `<text>` for the proposed answer text. The following keys are supported

- `T` for correct answers, `F` (the default value) for false ones,
- `Ttext` the verdict for correct answers, `Ftext` for false ones, and
- `feedback` for a short feedback text given to the student.

What we see when this is formatted to PDF depends on the context. In solutions mode (we start the solutions in the code fragment below) we get

Example 18

Input:

```
1 \startsolution
2 \begin{sproblem}[title=Functions,name=functions1]
3   What is the keyword to introduce a function definition in python?
4   \begin{mcb}
5     \mcc[T]{def} \mcc[F,feedback=that is for C and C++]{function}
6     \mcc[F,feedback=that is for Standard ML]{fun}
7     \mcc[F,Ftext=Nooooooooo,feedback=that is for Java]{public static void}
8   \end{mcb}
9 \end{sproblem}
```

Output:

Exercise (Functions)

What is the keyword to introduce a function definition in python?

- def^a
- function^b
- fun^c
- public static void^d

^aKorrekt

^bFalsch

that is for C and C++

^cFalsch

that is for Standard ML

^dNooooooooo

that is for Java

In “exam mode” where disable solutions (here via `\stopsolutions`) we get the questions without solutions (that is what the students see during the exam/quiz).

Example 19

Input:

```
1 \stopsolutions
2 \begin{sproblem}[title=Functions,name=functions1,autogradable]
3 What is the keyword to introduce a function definition in python?
4 \begin{mcb}
5   \mcc[T]{def} \mcc[F,feedback=that is for C and C++]{function}
6   \mcc[F,feedback=that is for Standard ML]{fun}
7   \mcc[F,Ftext=Nooooooooo,feedback=that is for Java]{public static void}
8 \end{mcb}
9 \end{sproblem}
```

Output:

Exercise (Functions)

What is the keyword to introduce a function definition in python?

- def
- function
- fun
- public static void

What we have seen is the default rendering of the multiple choice block, which is as an itemize like environment with check boxes. There are alternatives to that which we can choose with the `style` key in the optional attribute of the `mcb` environment. Currently the only alternative is `inline` style:

Example 20

Input:

```
1 \startproblems
2 \begin{sproblem}[title=Functions,name=functions1]
3 What is the keyword to introduce a function definition in python?
4
5 \begin{mcb}[style=inline]
6   \mcc[T]{def} \mcc[F,feedback=that is for C and C++]{function}
7   \mcc[F,feedback=that is for Standard ML]{fun}
8   \mcc[F,Ftext=Nooooooooo,feedback=that is for Java]{public static void}
9 \end{mcb}
10 \end{sproblem}
```

Output:

Exercise (Functions)

What is the keyword to introduce a function definition in python?

- def^a
- function^b
- fun^c
- public static void^d

^aKorrekt

^bFalsch

that is for C and C++

^cFalsch

that is for Standard ML

^dNoooooooooooo

that is for Java

This is the solutions mode, i.e. with solutions, otherwise, the footnotes will fully disappear.

Analogously, single choice blocks are marked up by the `scb` environment and the individual choices by the `\scc` macro. In PDF, there is little difference to the multiple choice case. In interactive formats like HTML, single choice blocks are typically realized via radio buttons to enforce single choice.²

4.2.6 Filling-In Concrete Solutions

The next simplest situation, where we can implement auto-grading is the case where we have fill-in-the-blanks

The `\fillinsol` macro takes a single argument, which contains a concrete solution (i.e. a number, a string, ...), which generates a fill-in-box in test mode:

Example 21

Input:

```
1 \stopsolutions
2 \begin{sproblem}[id=elefants.fillin,title=Fitting Elefants]
3 How many Elefants can you fit into a Volkswagen beetle? \fillinsol{4}
4 \end{sproblem}
```

Output:

Exercise (Fitting Elefants)

How many Elefants can you fit into a Volkswagen beetle?

and the actual solution in solutions mode:

Example 22

Input:

²EDNOTE: MK: are there any differences between mcc and scc we need to discuss here?

```

1 \startproblems
2 \begin{sproblem}[id=elefants.fillin,title=Fitting Elefants]
3 How many Elefants can you fit into a Volkswagen beetle? \fillinsol{4}
4 \end{sproblem}

```

Output:

Exercise (Fitting Elefants)

How many Elefants can you fit into a Volkswagen beetle?

4

If we do not want to leak information about the solution by the size of the blank we can also give `\fillinsol` an optional argument with a size: `\fillinsol[3cm]{12}` makes a box three cm wide.

Obviously, the required argument of `\fillinsol` can be used for auto-grading. For concrete data like numbers, this is immediate, for more complex data like strings “soft comparisons” might be in order.⁸

4.2.7 Answer Classes

³.

4.2.8 Including Problems

`\includeproblem` The `\includeproblem` macro can be used to include a problem from another file. It takes an optional KeyVal argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one problem in the include file). The keys `title`, `min`, and `pts` specify the problem title, the estimated minutes for solving the problem and the points to be gained, and their values (if given) overwrite the ones specified in the `problem` environment in the included file.

The sum of the points and estimated minutes (that we specified in the `pts` and `min` keys to the `problem` environment or the `\includeproblem` macro) to the log file and the screen after each run. This is useful in preparing exams, where we want to make sure that the students can indeed solve the problems in an allotted time period.

The `\min` and `\pts` macros allow to specify (i.e. to print to the margin) the distribution of time and reward to parts of a problem, if the `pts` and `pts` options are set. This allows to give students hints about the estimated time and the points to be awarded.

4.2.9 Testing and Spacing

The `problem` package is often used by the `hwexam` package, which is used to create homework assignments and exams. Both of these have a “test mode” (invoked by the

⁸For the moment we only assume a single concrete value as correct. In the future we will almost certainly want to extend the functionality to multiple answer classes that allow different feedback like in MCQ. This still needs a bit of design. Also we want to make the formatting of the answer in solutions/test mode configurable.

³EDNOTE: MK: describe them

package option `test`), where certain information –master solutions or feedback – is not shown in the presentation.

`\testspace` `\testspace` takes an argument that expands to a dimension, and leaves vertical space accordingly. Specific instances exist: `\testsmallspace`, `\testmedspace`, `\testlargepage` give small (1cm), medium (2cm), and big (3cm) vertical space.
`\testsmallspace` `\testnewpage` makes a new page in `test` mode, and `\testemptypage` generates an empty page with the cautionary message that this page was intentionally left empty.
`\testemptypage` The `solution` environment takes an `TODO`⁹
`solution`

4.3 Homework Assignments and Exams

4.3.1 Introduction

The `hwexam` package and class supplies an infrastructure that allows to format nice-looking assignment sheets by simply including problems from problem files marked up with the `problem` package. It is designed to be compatible with `problems.sty`, and inherits some of the functionality.

4.3.2 Package Options

`solutions` The `hwexam` package and class take the options `solutions`, `notes`, `hints`, `gnotes`, `pts`, `notes` `min`, and `boxed` that are just passed on to the `problems` package (cf. its documentation hints for a description of the intended behavior).
`gnotes`
`pts`
`min`

`multiple` Furthermore, the `hwexam` package takes the option `multiple` that allows to combine multiple assignment sheets into a compound document (the assignment sheets are treated as section, there is a table of contents, etc.).

`test` Finally, there is the option `test` that modifies the behavior to facilitate formatting tests. Only in `test` mode, the macros `\testspace`, `\testnewpage`, and `\testemptypage` have an effect: they generate space for the students to solve the given problems. Thus they can be left in the L^AT_EX source.

4.3.3 Assignments

`assignment` (*env.*) This package supplies the `assignment` environment that groups problems into assignment `number` sheets. It takes an optional KeyVal argument with the keys `number` (for the assignment number; if none is given, 1 is assumed as the default or — in multi-assignment documents `title` — the ordinal of the `assignment` environment), `title` (for the assignment title; this is `type` referenced in the title of the assignment sheet), `type` (for the assignment type; e.g. “quiz”, `given` or “homework”), `given` (for the date the assignment was given), and `due` (for the date `due` the assignment is due).

⁹TODO: check what is still undescribed `problem.sty` and make examples for it.

4.3.4 Including Assignments

\inputassignment

The `\inputassignment` macro can be used to input an assignment from another file. It takes an optional KeyVal argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one `assignment` environment in the included file). The keys `number`, `title`, `type`, `given`, and `due` are just as for the `assignment` environment and (if given) overwrite the ones specified in the `assignment` environment in the included file.

4.3.5 Typesetting Exams

`testheading (env.)` The `\testheading` takes an optional keyword argument where the keys `duration` specifies a string that specifies the duration of the test, `min` specifies the equivalent in number of minutes, and `reqpts` the points that are required for a perfect grade.

```
reqpts1 \title{320101 General Computer Science (Fall 2010)}
2 \begin{testheading}[duration=one hour,min=60,reqpts=27]
3   Good luck to all students!
4 \end{testheading}
```

Will result in

Name:

Matriculation Number:

320101 General Computer Science (Fall 2010)

2023-11-24

You have one hour (sharp) for the test;

Write the solutions to the sheet.

The estimated time for solving this exam is 23 minutes, leaving you 37 minutes for revising your exam.

You can reach 23 points if you solve all problems. You will only need 27 points for a perfect score, i.e. -4 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

	To be used for grading, do not write here								
prob.	4.1	1.1	1.2	2.1	2.2	2.3	2.4	2.5	2.6
total	0	0	0	10	0	0	0	0	0
reached									

good luck

10

¹⁰MK: The first three “problems” come from the stex examples above, how do we get rid of this?

Part II

User Manual

*The dynamic [HTML](#) version of this part can be found at
<https://stexmmt.mathhub.info/>:
<https://stexmmt.mathhub.info/stex/fullhtml?archive=stex/Documentation&filepath=manual.xhtml>*

Chapter 5

Basics

5.1 Package and Class Options

- `debug=(prefixes)`: (see Developer Manual)
- `lang=(languages)`: If set, **S_TE_X** will load the `babel` package with the provided languages. Supported languages (currently) are:

<code>ar</code>	arabic
<code>bg</code>	bulgarian
<code>de</code>	german (with option <code>ngerman</code>)
<code>en</code>	english
<code>fi</code>	finnish
<code>fr</code>	french
<code>ro</code>	romanian
<code>ru</code>	russian
<code>tr</code>	turkish (with option <code>shorthands=:</code> !)

- `mathhub=(path)`: Uses the provided file path as `MathHub` directory (see [section 5.2](#)).
- `usesms/writesms`: If `writesms` is set, content loaded from external `math archives` (i.e. `modules`) is persisted in the file `\jobname.sms`.

If `usesms` is set, the content of the `.sms`-file is loaded, obviating the need to reprocess the original files.

The options are not mutually exclusive, but care should be taken if dependencies have changed between builds.

This offers two advantages:

1. If a document has many (transitive) dependencies, `usesms` should significantly speed up the build process, and
2. setting `usesms` allows for distributing the `.sms`-file to make the document *standalone*, allowing for compilation without needing imported/used modules to be present.

The options `debug`, `mathhub`, `usesms` and `writesms` can also be set by the environment variables `STEX_DEBUG`, `MATHHUB`, `STEX_USESMS` and `STEX_WRITESMS`. In fact, the `MATHHUB` environment variable is the recommended way to set the `MathHub` directory. This is the only option where the *package option* overrides the environment variable.

The environment variables for `USE/WRITESMS` are particularly useful, in that they allow for convenient compilation workflows. For example, the `Build PDF/XHTML/OMDoc`-button in the `IDE` does the following:

```
STEX_WRITESMS=true pdflatex <job>.tex
[bibtex|biber] <job>
STEX_USESMS=true pdflatex <job>.tex
STEX_USESMS=true pdflatex <job>.tex
```

Guaranteeing (in the first run) that all dependencies are loaded from their respective sources and persisted, and in the two subsequent runs read from the generated `.sms` file, (likely) speeding up the subsequent runs significantly.

5.2 Math Archives and the MathHub Directory

`STEX` uses `math archives` to organize document content modularly, without a user having to specify absolute paths, which would differ between users and machines.

All `STEX` archives need to exist in the local `MathHub`-directory. `STEX` knows where this folder is via one of five means:

1. If the `STEX package` is loaded with the option `mathhub=/path/to/mathhub`, then `STEX` will consider `/path/to/mathhub` as the local `MathHub` directory.
2. If the `mathhub` package option is *not* set, but the macro `\mathhub` exists when the `STEX`-package is loaded, then this macro is assumed to point to the local `MathHub` directory; i.e. `\def\mathhub{/path/to/mathhub}\usepackage{stex}` will set the `MathHub` directory as `path/to/mathhub`.
3. Otherwise, `STEX` will attempt to retrieve the system variable `MATHHUB`, assuming it will point to the local `MathHub` directory. Since this variant needs setting up only *once* and is machine-specific (rather than defined in tex code), it is compatible with collaborating and sharing tex content, and hence recommended.
4. If that too fails, `STEX` will look for a file `~/.stex/mathhub.path`. If this file exists, `STEX` will assume that it contains the path to the local `MathHub`-directory. This method is recommended on systems where it is difficult to set environment variables, and is used by the `IDE` setup.
5. Finally, if all else fails, `STEX` considers `~/MathHub` to be the `MathHub` directory.

The `STEX IDE` allows you to directly download `math archives` from [gl.mathhub.info](#) – currently available `archives` there are:

- `sTeX/*` – a group of semi-experimental documents showcasing `STEX`3 features,
- `smgloM/*` – a vast collection of multilingual `modules` of concepts in mathematics and computer science. The SMGloM predates `STEX`3 and is thus largely “underannotated” with respect to (formal) semantics,
- `MiKoMH/*` – a vast collection of lecture slides and notes in computer science for courses held by Michael Kohlhase. They largely make use of SMGloM `modules`.

5.2.1 The Structure of Math Archives

An `archive` group/name is stored in the directory `<MathHub>/group/name`; e.g. assuming your local MathHub-directory is set as `/user/foo/MathHub`, then in order for the sTeX/Documentation `archive` to be found by the `sTeX` system, it needs to be in `/user/foo/MathHub/sTeX/Documentation`.

Each such `archive` needs two subdirectories:

- `/source` – this is where all your tex files go.
- `/META-INF` – a directory containing a single file `MANIFEST.MF`, the content of which we will consider shortly.

An additional `lib`-directory is optional, and is discussed in [section 5.3](#).

5.2.2 MANIFEST.MF-Files

The `MANIFEST.MF` in the `META-INF` directory consists of key-value-pairs, informing `sTeX` (and associated software, e.g. `MMT`) of various properties of an `archive`. For example, the `MANIFEST.MF` of the sTeX/Documentation archive looks like this:

```
id: sTeX/Documentation
ns: http://mathhub.info/sTeX/Documentation
narration-base: http://mathhub.info/sTeX/Documentation
format: stex
title: The sTeX Documentation
teaser: The full Documentation for the sTeX system
url-base: https://stexmmt.mathhub.info/:sTeX
dependencies:sTeX/ComputerScience/Software,sTeX/MathTutorial
ignore: */code/*|*/tikz/*|*/tutorial/solution/*
```

Many of these are in fact ignored by `sTeX`, but some are important:

`id`: The name of the `archive`, including its group (e.g. `sTeX/Document`). This is used by the `MMT` system in favor of the directory, but `TeX`'s limited access to the file system enforces the directory structure.

`source-base` or

`ns`: The namespace from which all `symbol` and `module MMT-URIs` in this `archive` are formed.

`narration-base`: The namespace from which all document `MMT-URI` in this repository are formed. It can safely match the `ns`-field.

`url-base`: A URL that is formed as a basis for *external references*. and hyperlinks. An `MMT` (or comparable system) instance should run there and host (`sTeX`-generated) `HTML`.

`dependencies`: All `archives` that this `archive` depends on. `sTeX` ignores this field, but `MMT` can pick up on them to resolve dependencies, e.g. when downloading `archives` in the `IDE`, which will also download all dependencies.

`ignore`: A regular expression of `.tex` files in the `source` directory that should be ignored; e.g. they will not be compiled when building a whole directory or archive in the `IDE`.

5.3 The lib-Directory

A `math archive`/`archive` may have a `lib` directory primarily intended for preamble code, `packages`, `.bib` files, etc., the files in which can be referenced in various ways.

Additionally, a *group* of archives `group` may have an additional `archive` `group/meta-inf`. If this `meta-inf` archive has a `/lib`-subdirectory, they too will be considered by the following.

`\libinput \libinput {some/file}` searches for a file `some/file[.tex]` in

- the `lib`-directory of the current archive, and
- the `lib`-directory of a `meta-inf`-archive in (any of) the archive groups containing the current archive

and `\inputs` all found files in reverse order; e.g. `\libinput{preamble}` in a `.tex`-file in `sTeX/Documentation` will *first* input `.../sTeX/meta-inf/lib/preamble.tex` and then `.../sTeX/Documentation/lib/preamble.tex`.

`\libinput` will throw an error if *no* candidate for `some/file` is found.

`\libinput[some/archive]{some/file}` will do the same, but starting in the `lib` directory of the `math archive` `some/archive`.

`\libusepackage \libusepackage [package-options]{some/file}` searches for a file `some/file.sty` in the same way that `\libinput` does, but will call `\usepackage[package-options]{path/to/some/file}` instead of `\input`.

`\libusepackage` throws an error if not *exactly one* candidate for `some/file.sty` is found.

`\addmhbibresource \addmhbibresource [some/archive]{some/file}` searches for a file like `some/file.bib` in `some/archive`'s `lib` directory and calls `\addbibresource` to the result.

`\libusetikzlibrary \libusetikzlibrary` behaves like `\libusepackage` but looks specifically for tikz libraries and calls `\usetikzlibrary` on the results.

throws an error if not *exactly one* candidate for the library is found.

A good practice is to have individual `sTeX` fragments follow basically this document frame:

```
\documentclass{sTeX}
\libinput{preamble}
\setsectionlevel{<your preference>}
\begin{document}
\IfInputref{}{
...
\maketitle
\ifstexhtml \else \tableofcontents \fi
}
...
\IfInputref{}{\libinput{postamble}}
\end{document}
```

Then the `preamble.tex` files can take care of loading the generally required `packages`, setting presentation customizations etc. (per archive or archive group or both), and a `postamble.tex` can e.g. print the bibliography, index etc.

`\libusepackage` is particularly useful in such a `preamble.tex` when we want to use custom packages that are not part of a `TeX` distribution, or on CTAN. In this case we commit the respective packages in one of the `lib` folders and use `\libusepackage` to load them.

5.4 Basic Macros

`\sTeX` The `\sTeX` macro produces this S^ITeX logo. It is provided by the `sTEX-logo` package, `\stex` included with the `stex` package.

`\ifstexhtml` The `\ifstexhtml` conditional is *true* if the current compilation generates `HTML`, and *false* otherwise (i.e. generates `PDF`).

`\STEXinvisible` `\STEXinvisible{\langle code \rangle}`

Processes `\langle code \rangle`, but does not generate any output. In the `HTML`, `\langle code \rangle` is exported with `display:none`.

Can be used to declare formal content and preserve its semantics in `HTML` without generating output.

Chapter 6

Document Features

6.1 Document Fragments

sfragment (env.) To make reusability of document fragments more feasible, **STEX** provides the **sfragment** environment. `\begin{sfragment}[id=<id>,short=<short title>]{section title}` calls `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` or `\ subparagraph` with argument `{section title}` depending on the current *section level* and availability, and increases the level accordingly.

The `<id>` can be used for cross-document references (see section 6.3).

blindfragment (env.) In the case where we want to increase the section level *without* producing a corresponding section header, the **blindfragment** environment can be used. This allows e.g. typesetting `\sections` before the first `\chapter`.

\skipfragment The `\skipfragment` macro “skips an **sfragment**”, i.e. it just steps the respective sectioning counter. This macro is useful, when we want to keep two documents in sync structurally, so that section numbers match up: Any section that is left out in one becomes a `\skipfragment`.

\setsectionlevel The `\setsectionlevel` macro sets the current section level to that provided as argument. This is particularly useful in the preamble of a document, as to be ignored in e.g. `\inputref` and make sure that sectioning proceeds as desired; e.g. `\setsectionlevel{section}` make sure that the first **sfragment** will be typeset as a `\section` (rather than e.g. a `\part`).

\currentsectionlevel The `\currentsectionlevel` macro produces the literal string corresponding to the current section level – e.g. within a chapter (but outside of a section), `\currentsectionlevel` produces “chapter”.
The `\Currentsectionlevel` macro does the same, but capitalizes the first letter; e.g. in the above situation, `\Currentsectionlevel` produces “Chapter”.

6.2 Using and Referencing Document Fragments

`\inputref` `\inputref [<archive>]{<file>}`

Inputs the file `<file>` in `<archive>`'s `source` directory. If `[<archive>]` is empty, the current `archive`'s `source` directory is used. If there is no current `archive`, `<file>` is resolved relative to the current file.

The file's content is processed within a `TEX` group when using `pdflatex`. When converting to `HTML` however, the file is not processed *at all*, and instead, a reference to the file is inserted, that can be replaced by the `HTML` generated by the referenced file by e.g. the `MMT` system.

This is the recommended method to assemble documents from individual `.tex` files.

`\mhinput`

Like `\inputref`, but actually calls `\input` in both `PDF` and `HTML` mode. Useful for small fragments or those without `modules`, but generally `\inputref` should be preferred.

`\ifinputref` `\IfInputref`

`\ifinputref` is a `TEX` conditional for whether the current file is currently processed via `\IfInputref`.

`\IfInputref {<true code>}{<false code>}` behaves like

`\ifinputref{true code}\else{false code}\fi` when using `pdflatex`; in `HTML` mode however, *both* arguments are processed and marked-up accordingly, so a hosting server (like `MMT`) can dynamically decide which parts to show or omit.

`\mhgraphics`
`\cmhgraphics`

If the `graphicx` package is loaded, `\mhgraphics` takes the same arguments as `\includegraphics`, with the additional optional key `archive`. It then resolves the file path in

`\mhgraphics[archive=some/archive]{some/image}` relative to the `source`-folder of the `some/archive` `archive`. If no `archive` is provided, the file path `some/image` is resolved relative to the current `archive` (if existent).

`\cmhgraphics` additional wraps the image in a `center`-environment.

`\lstinputmhlisting`
`\clstinputmhlisting`

Like `\mhgraphics`, but for `\lstinputlisting` instead of `\includegraphics`. Only defined if the `listings` package is loaded.

6.3 Cross-Document References

If we take features like `\inputref` and `\mhinput` (and the `sfragment` environment) seriously and build large documents modularly from individually compiling documents for sections, chapters and so on, cross-referencing becomes an interesting problem.

Say, we have a document `main.tex`, which `\inputrefs` a section `section1.tex`, which references a definition with label `some_definition` in `section2.tex` (subsequently also `\inputrefed` in `main.tex`). Then the numbering of the definition will depend on the *document context* in which the document fragment `section2.tex` occurs - in `section2.tex` itself (as a standalone document), it might be *Definition 1*, in `main.tex`

it might be *Definition 3.1*, and in `section1.tex`, the definition *does not even occur*, so it needs to be referenced by some other text.

What we would want in that instance is an equivalent of `\autoref`, that takes the document context into account to yield something like *Definition 1*, *Definition 3.1* or “*Definition 1 in the section on Foo*” respectively.

For that to work, we need to supply (up to) three pieces of information:

- The *label* of the reference target (e.g. `some_definition`),
- (optionally) the *file/document* containing the reference target (e.g. `section2`). This is not strictly necessary if the reference target occurs in the *same* document, but if not, we need to know where to find the label,
- (optionally) the document context, in which we want to refer to the reference target (e.g. `main`).

Additionally, the document in which we want to reference a label needs a title for external references.

```
\sref
\sref [archive=<archive1>,file=<file1>]
{<label>}[archive=<archive2>,file=<file2>,title=<title>]
```

This `macro` references *label* (declared in *file1* in `math archive <archive1>`). If the object (section, figure, etc.) with that label occurs (eventually) in the same document, `\sref` will ignore the second set of optional arguments and simply defer to `\autoref` if that command exists, or `\ref` if the `hyperref` package is not included.

If the referenced object does *not* occur in the current document however, `\sref` will refer to it by the object’s name as it occurs in the file *file2* in `archive <archive2>`, followed by the title.

In `HTML` mode, the reference additionally links to the `HTML` of the *file1*.¹¹

This works by storing labels during compilation in a file *jobname*.`sref`, analogous to e.g. the `.toc`. Note that this consequently requires both `file1.tex` and `file2.tex` to have been compiled previously, to generate the `.sref` file.

For example, doing

```
\sref[file=tutorial/full.en]{sec:basics}[file=tutorial.en,title=the \stex Tutorial]
in this very document fragment ([sTeX/Documentation]macros/sref.en.tex) will yield
Part I (The Basics) in the \stex Tutorial if compiled itself, or if compiled as part of the
\stex manual, and will yield the \autoref link chapter 2 in the documentation (which
includes the tutorial).
```

```
\srefsetin
\srefsetin [<archive2>]{<file2>}{<title>}
```

Sets a default value for the optional arguments *archive2*, *file2* and *title* of `\sref`. If the second set of optional arguments in `\sref` are omitted, these default values are used. Particularly useful to set in a preamble.

```
\sreflabel
\sreflabel {<label>} sets a label analogous to \label{<label>}, but for use in \sref.
Note that for every \stex macro or environment that takes an optional id=<id>
argument, the <id> (if non-empty) generates an \sreflabel automatically.
For example, \begin{sfragment}[id=foo]{Foo} is equivalent to
\begin{sfragment}{Foo}\sreflabel{foo}.
```

\extref `\extref [archive=<archive1>,file=<file1>]
{<label>} {archive=<archive2>,file=<file2>,title=<title>}`

Like **\sref**, but with the third argument mandatory, **\extref** will *always* produce the output as if *<label>* would *not* occur in the current document.

Chapter 7

Modules and Symbols

7.1 Modules

A `module` is required to declare any new formal content such as `symbols` or `notations` (but not `variables`, which may be introduced anywhere).

↳ An `STEX module` corresponds to an `MMT/OMDOC theory`. As such
↳ it gets assigned an `MMT-URI` (*universal resource identifier*) of the form
~~~T~~<namespace>?<module-name>.

`smodule (env.)` A new module is declared using the basic syntax

`\begin{smodule}[options]{ModuleName}... \end{smodule}`.

A module is required to declare any new formal content such as symbols or notations (but not variables, which may be introduced anywhere).

The `smodule`-environment takes several keyword arguments, all of which are optional:

`title` (*token list*) to display in customizations.

`style` (*string*)\* for use in customizations, see [chapter 9](#)

`id` (*string*) for cross-referencing, see `\sreflabel`.

`ns` (*URI*) the namespace to use. *Should not be used, unless you know precisely what you're doing.* If not explicitly set, is computed from the containing file and `archive`'s namespace.

`lang` (*language*) if not set, computed from the current file name (e.g. `foo.en.tex`).

`sig` (*language*) see below.

---

---

`\STEXexport` `\STEXexport{<code>}` executes *<code>* immediately and every time the current `module` is being used.



For technical reasons, `\STEXexport` processes its content in the `expl3` category code scheme – what this means is that all spaces are ignored entirely, and the characters `_` and `:` are valid characters in `macro` names.

In practice, this means you will have to use the `~` character for spaces, and if you want to use a subscript `_`, you should use the `macro` `\c_math_subscript_token` instead.

Also, note that no *global macro* definitions should happen in `\STEXexport`; this can lead to unexpected behaviour if the containing `module` has been used previously in the current document.

### 7.1.1 Signature Modules, Languages, and Multilinguality

if the current file is a translation of a file with the same base name but a different language suffix, setting `sig=<lang>` will preload the `module` from that language file. This helps ensuring that the (formal) content of both `modules` is (almost) identical across languages and avoids duplication.

For example, we can have a file `Foo.en.tex`, that declares and documents a `module` `Foo` (using `\begin{smodule}{Foo}`). If we put a file `Foo.de.tex` next to it, we can do `\begin{smodule}[sig=en]{Foo}` to have all the content in the `module` `Foo` (as declared in `Foo.en.tex`) available and translate its document content to german.

The `MMT` backend, when serving `STEX` content as `HTML`, will always attempt to find documentation in the language corresponding to the context; e.g. a user's preference.

## 7.2 Symbol Declarations

---

`\symdecl` `\symdecl {<mname>}[<options>]`

The `\symdecl` `macro` is the simplest way to introduce a new `symbol`. If `<options>` contains `name=<name>`, then `<name>` is the *name* of the `symbol`; otherwise, `<mname>` is used for the *name*. Additionally, a `semantic macro` `\mname` is generated.

The starred variant `\symdecl*` does not generate a `semantic macro`, in which case the `name`-option is superfluous.

→ `\symdecl` introduces a new `MMT/OMDoc constant` in the current `module` (i.e. → `MMT/OMDoc theory`). Correspondingly, they get assigned the `MMT-URI` ↵ `<module-URI>?<constant-name>`.

`\symdecl` takes the following optional arguments:

`name` see above,

`args` the arity of the `symbol` and its `semantic macro`; may be a number `0...9` or a string consisting of the characters `i`, `a`, `b` and `B` of length  $\leq 9$ ,

`type` the `symbol`'s `type`,

`def` the `symbol`'s `definiens`,

`return` the `symbol`'s *return code* (see below), most commonly the `semantic macro` of a `mathematical structure`,

`assoc` how to resolve arguments of `mode` `a` or `B`; may be `pre`, `bin`, `binl`, `binr` or `conj`,

`reorder` how to reorder the arguments in `OMDoc` (*advanced*),

`role` `symbols` with certain roles are treated in particular ways in `MMT/OMDoc` (*advanced*),

`argtypes` `TODO12`.

---

`\textsymdecl` `\textsymdecl{<symbol>}[<options>]{<code>}`

Like `\symdecl`, but requires that the `symbol` has arity 0 (hence `\textsymdecl` does not take the `args`-option), and generates a `semantic macro` that takes no arguments in either text or math mode, and produces marked-up `<code>` as output.

Additionally, a `macro` `\<symbol>name` is generated that produces `<code>` without any semantic markup.

---

`\symdef` `\symdef{<symbol>}[<options>]{<notation>}`

Combines the functionalities and optional arguments of `\symdecl` and `\notation` in one.

### 7.2.1 Returns

Assume we have a `symbol` `foo` with `semantic macro` `\foo`, (exemplary) taking two arguments, and `return=<code>`. If we do `\foo{a}{b}!`, the return code is simply ignored. If we do `\foo{a}{b}` *without* the `!`, here is what happens:

1. `STEX` will replace `#1` and `#2` in `<code>` by `a` and `b`, yielding `<retcode>`.
2. `STEX` will insert `<retcode>\foo{a}{b}!` in the input token stream.

This means that `<code>` should contain at most `<arity of foo>` argument markers, and eat precisely one argument appended to `<code>`.

When in doubt, we recommend only using `semantic macros` for `mathematical structures` (with only optional arguments) and `\apply` (with only optional arguments) in `return`.

## 7.3 Referencing Symbols

---

`\symref` `\symref{<symbol>}{<text>}`

The `\symref` macro (and its short version `\sr`) is the most general variant to mark-up arbitrary `LATEX` code `<text>` with the `symbol`.

---

<sup>12</sup>TODO: experimental

This is as good a place as any other to explain how **STEX** resolves a string `symbol` to an actual `symbol`.

If `\symbol` is a `semantic macro`, then **STEX** has no trouble resolving `symbolname` to the full `MMT-URI` of the `symbol` that is being invoked.

However, especially in `\symname` (or if a `symbol` was introduced using `\symdecl*` without generating a `semantic macro`), we might prefer to use the `name` of a symbol directly for readability – e.g. we would want to write A `\symname{natural number}` is... rather than A `\symname{Nat}` is.... **STEX** attempts to handle this case thusly:



If `symbol` does *not* correspond to a `semantic macro` `\symbol` and does *not* contain a ?, then **STEX** checks all `symbols` currently in scope until it finds one, whose name is `symbol`. If `symbol` is of the form `pre?name`, **STEX** first looks through all `modules` currently in scope, whose full `MMT-URI` ends with `pre`, and then looks for a symbol with name `name` in those. This allows for disambiguating more precisely, e.g. by saying `\symname{Integers?addition}` or `\symname{RealNumbers?addition}` in the case where several additions are in scope.

M → `\symref{\symbol}{\text}`    in    MMT/OMDoc    generates    the    term  
 M → `<OMS name="\symbol URI"/>`.  
 ~T →

---

`\symname`    `\symname[pre=\pre,post=\post]{\symbol}`

`\sn`    `\symname[pre=\pre,post=\post]{\symbol}`

`\Symname` If the `symbol` referenced by `\symbol` has name `name`, this is a shortcut for

`\Sn`    `\symref{\symbol}{\pre\name\post}`.

`\sns`    For example, given a symbol `agroup` with name `abelian group`, we can do `\symname[pre=Non-,post=s]{agroup}` to produce `Non-abelian groups`.

`\sn` is a shorter variant for `\symname`; `\Symname` and `\Sn` additionally capitalize the first letter. `\sns` and `\Sns` are short for `\sn [post=s]` and `\Sn [post=s]`, respectively.

---

`\srefsym`    `\srefsym{\symbol}{\text}`  
`\srefsymuri`    `\srefsymuri{\symbol}{\text}`

turns `\text` into a link to

- The documentation of `\symbol`, if it occurs in the same document, or
- the `symbol`'s documentation online, based on the containing `math archive`'s `url-base`.

`\srefsymuri` does the same, but expects a `symbol`'s full `MMT-URI` as first argument. This is particularly useful for e.g. customizing highlighting (see [chapter 9](#)).

---

`\symuse`    `\symuse{\symbol}` behaves exactly like a `semantic macro` for `\symbol`.

## 7.4 Notations and Semantic Macros

---

`\notation` `\notation{\symbol}{[options]}{code}`

introduces a new `notation` for the referenced `symbol`.

The starred variant `\notation*` sets this `notation` as the (new) default `notation`.

The optional arguments are:

- `prec=(opprec);(argprec 1)x...x(argprec n)`: An `operator precedence` and one `argument precedence` for each argument of the `semantic macro`. If no `argument precedences` are given, all `argument precedences` are equal to the `operator precedence`. By default, all `precedences` are 0, unless the `symbol` takes no argument, in which case the `operator precedence` is `\neginfpref` (negative infinity).
- `prec=nobrackets` is an abbreviation for `prec=\neginfpref;\infprec x...x\infprec`.
- `op=(code)`: An `operator notation`. If none is given, the notation component marked with `\maincomp` is used. If no `\maincomp` occurs in the `notation`, the default `operator notation` is `\symname{\symbol}`.
- `variant=(id)`: An id for this `notation`. The key `variant=` can be omitted; i.e. `\notation[foo]` is equivalent to `\notation[variant=foo]`.

---

`\comp` `\maincomp`

`\comp` is used to mark notation components in a `\notation` to be highlighted. Additionally, each `notation` can use `\maincomp` at most once to mark the *primary* notation component.



Ideally, `\comp` would not be necessary: Everything in a `notation` that is *not* an argument should be a notation component. Unfortunately, it is computationally expensive to determine where an argument begins and ends, and the argument markers `#n` may themselves be nested in other `macro` applications or `TeX` groups, making it ultimately almost impossible to determine them automatically while also remaining compatible with arbitrary highlighting customizations (such as tooltips, hyperlinks, colors) that users might employ, and that are ultimately invoked by `\comp`.



Note that it is required that

1. the argument markers `#n` never occur inside a `\comp`, and
2. no `semantic macros` may ever occur inside a `notation`.

Both criteria are not just required for technical reasons, but conceptionally meaningful:

The underlying principle is that the arguments to a `semantic macro` represent *arguments to the mathematical operation* represented by a `symbol`. For example, a `semantic macro` application `\plus{a}{b}` would represent *the actual addition of (mathematical objects) a and b*. It should therefore be impossible for `a` or `b` to be part of a notation component of `\plus`.

Similarly, a `semantic macro` can not conceptually be part of the `notation` of `\plus`,

since a `symbol` represents a *distinct (mathematical) concept* with *its own semantics*, and `notations` are syntactic representations of the very `symbol` to which the `notation` belongs.



If you want an argument to a `semantic macro` to be a purely syntactic parameter, then you are likely somewhat confused with respect to the distinction between the precise *syntax* and *semantics* of the `symbol` you are trying to declare (which happens quite often even to experienced `STEX` users, like us), and might want to give those another thought - quite likely, the concept you aim to implement does not actually represent a semantically meaningful (mathematical) concept, and you will want to use `\def` and similar native `LATEX` `macro` definitions rather than `semantic macros`.

---

`\setnotation` The first `notation` provided will stay the default `notation` unless explicitly changed:  
`\setnotation{\langle symbol \rangle}{\langle id \rangle}` sets the default `notation` of `\langle symbol \rangle` to that with id `\langle id \rangle`.

#### 7.4.1 Precedences and Bracketing

---

`\infprec` and `\neginfprec` represent *infinitely large* and *infinitely small precedences*, respectively.



`STEX` decides whether to insert parentheses by comparing `operator precedences` to a *downward precedence*  $p_d$  with initial value `\infprec`. When encountering a `semantic macro`, `STEX` takes the `operator precedence`  $p_{op}$  of the `notation` used and checks whether  $p_{op} > p_d$ . If so, `STEX` inserts parentheses.

When `STEX` steps into an argument of a `semantic macro`, it sets  $p_d$  to the respective `argument precedence` of the `notation` used.

##### Example 23

Consider `semantic macros` `\plus` and `\mult` taking two arguments, with `notations`  $a + b$  and  $a \cdot b$  respectively, and `precedences` 100 for `\plus` and 50 for `\mult`.

Consider `$(\plus{a, \mult{b, (\plus{c, d})}})` (i.e.  $a + b \cdot (c + d)$ ). Then:

1. `STEX` starts out with  $p_d = \infprec$ .
2. `STEX` encounters `\plus` with  $p_{op} = 100$ . Since  $100 \not> \infprec$ , it inserts no parentheses.
3. Next, `STEX` encounters the two arguments for `\plus`. Both have no specifically provided `argument precedence`, so `STEX` uses  $p_d = p_{op} = 100$  for both and recurses.
4. Next, `STEX` encounters `\mult{b, ...}`, whose `notation` has  $p_{op} = 50$ .
5. We compare to the current downward `precedence`  $p_d$  set by `\plus`, arriving at  $p_{op} = 50 \not> 100 = p_d$ , so `STEX` again inserts no parentheses.

6. Since the **notation** of `\mult` has no explicitly set **argument precedences**, **STEX** again uses the **operator precedence** for the arguments of `\mult`, hence sets  $p_d = p_{op} = 50$  and recurses.
7. Next, **STEX** encounters the inner `\plus{c, ...}` whose **notation** has  $p_{op} = 100$ . We compare to the current downward **precedence**  $p_d$  set by `\mult`, arriving at  $p_{op} = 100 > 50 = p_d$  – which finally prompts **STEX** to insert parentheses, and we proceed as before.

`\dobracket` `\dobrackets{(code)}` wraps parentheses around `{(code)}`.

`\withbrackets` `\withbrackets{(left)}{(right)}{(code)}` uses the opening and closing parentheses `(left)` and `(right)` for the next pair of parentheses automatically inserted in `{(code)}`.

#### 7.4.2 Notations for Argument Sequences

The following **macros** can be used in **notations** that take **mode a or B** arguments:

`\argssep` `\argssep{(parameter token)}{(separator)}`

takes the elements of the argument sequence in position `<parameter token>` and separates them by `<separator>`.

Note that the first argument *must* be a parameter token of the form `#k`, and the argument at position `k` of the **notation** has to have **argument mode a or B**.

`\argmap` `\argmap{(parameter token)}{(code)}{(separator)}`

takes the elements of the argument sequence in position `<parameter token>`, applies the code `{(code)}` to each of them (which therefore should use `##1`) and separates them by `<separator>`.

For example, the **notation** `{\argmap{#1}{X^{##1}}{++}}` applied to the argument `{a,b,c}` produces  $X^a + X^b + X^c$ .

`\argarraymap` `TODO13`

#### 7.4.3 Semantic Macros

Assume we have a **semantic macro** `\smacro` taking (exemplary) two arguments. The precise behaviour of `\smacro` depends on whether we are in text or math mode.

**Math Mode** `\smacro!` produces the default **operator notation** of its **symbol**. Without `!`, `\smacro` expects at least two arguments, and `\smacro{a}{b}!` produces the default **notation** of its **symbol**.

If the **symbol** has a **return** code, then `\smacro{a}{b}` continues with executing the **return** code. Otherwise, `\smacro{a}{b}` also simply produces the default **operator notation**.

The starred variants `\smacro*` and `\smacro!*` behave as in *text mode*.

**Text Mode** `\smacro!{\langle arg\rangle}` marks up  $\langle arg \rangle$  similarly to how `\symref{\smacro}{\langle arg\rangle}` would.

Without the `!`, `\smacro` still only takes a single argument, but it is expected, that within  $\langle arg \rangle$ , the arguments for the `symbol` are explicitly marked up. The `\comp` macro is allowed in  $\langle arg \rangle$  to determine the components of  $\langle arg \rangle$  to be highlighted.

`\arg` The `\arg` macro can be used to explicitly mark the arguments of a `semantic macro` in text mode.

By default, they are numbered consecutively; e.g. `\smacro{... \arg{a} ... \arg{b}}` determines `a` and `b` to be the (first and second) arguments.

The starred variant `\arg*` allows for marking up the arguments, but does not produce any output. This can be used to provide arguments that are not mentioned in the text we want to mark up, because they are implicitly obvious or mentioned elsewhere.

If we want to change the order of the arguments, we can provide the precise argument number as an optional argument; e.g. `\smacro{... \arg[2]{a} ... \arg[1]{b}}` determines `b` to be the first and `a` to be the second argument.

An argument number may be used repeatedly, if the corresponding `argument mode` is `a` or `B`.

M → Applications of `semantic macros` with arguments are translated to `MMT/OMDoc` as `OMA-terms` with head `<OMS name="⟨symbol⟩"/>`, or `<OMBIND name="⟨symbol⟩"/>`,  
 M → depending on the absence or presence of `argument mode` `b` or `B` arguments.  
 ~T → Semantic macros with no arguments or invoked with `!` correspond to `OMS` directly.

## 7.5 Simple Inheritance

There are three `macros` that allow for opening a `module`, making its contents available for use:

`\usemodule` `\usemodule{\langle module\rangle}` is allowed anywhere and makes the `module`'s contents available up to the current `TEX` group. This is the right `macro` to use outside of `modules`, or when none of its contents use any of the used `module`'s `symbols` directly (e.g. in `types` or `definientia`).

`\requiremodule` `\requiremodule{\langle module\rangle}` is only allowed in `modules` and makes the required `module`'s contents available within the current `module`. The imported `symbols` can be safely used in `types` and `definientia`, but not in the `return` code of `symbols`, and the imported content is not exported further – i.e. using the current `module` does not also open the required `module`.

`\importmodule` `\importmodule{\langle module\rangle}` is only allowed in `modules` and makes the required `module`'s contents available within the current `module`. The imported `symbols` can be safely used anywhere, and the imported content exported to any `modules` subsequently importing the current one.

$\hookrightarrow M \rightarrow$  In **MMT**, every **document** and every **module** induces an **MMT theory**. **\usemodule**  
 $\dashv M \rightarrow$  induces and **MMT include** in the document **theory**, **\importmodule** and  
 $\rightsquigarrow T \rightsquigarrow \requiremodule$  both induce an **include** in the **module's theory**.

It is worth going into some detail how exactly **\usemodule**, **\importmodule** and **\requiremodule** resolve their arguments to find the desired **module** – which is closely related to the *namespace* generated for a **module**, that is used to generate its **MMT-URI**.

Ideally, **STEX** would allow for arbitrary **MMT-URIs** for **modules**, with no forced relationships between the *logical namespace* of a **module** and the *physical location* of the file declaring it – like **MMT** in fact allows for.

Unfortunately, **TEX** only provides very restricted access to the file system, so we are forced to generate namespaces systematically in such a way that they reflect the physical location of the associated files, so that **STEX** can resolve them accordingly. Largely, users need not concern themselves with namespaces at all, but for completeness sake, we describe how they are constructed:



- If **\begin{smodule}{Foo}** occurs in a file `/path/to/file/Foo[.(lang)].tex` which does not belong to an **math archive**, the namespace is `file://path/to/file`.
- If the same statement occurs in a file `/path/to/file/bar[.(lang)].tex`, the namespace is `file://path/to/file/bar`.

In other words: outside of **math archives**, the namespace corresponds to the file URI with the filename dropped iff it is equal to the **module** name, and ignoring the (optional) language suffix.

If the current file is in an **archive**, the procedure is the same except that the initial segment of the file path up to the **archive**'s **source** directory is replaced by the **archive**'s namespace URI.

Conversely, here is how namespaces/URIs and file paths are computed in import statements, exemplary **\importmodule**:



- **\importmodule{Foo}** outside of an **archive** refers to **module Foo** in the current namespace. Consequently, Foo must have been declared earlier in the same file or, if not, in a file `Foo[.(lang)].tex` in the same directory.
- The same statement *within* an **archive** refers to either the **module Foo** declared earlier in the same file, or otherwise to the **module Foo** in the **archive**'s top-level namespace. In the latter case, it has to be declared in a file `Foo[.(lang)].tex` directly in the **archive**'s **source** directory.
- Similarly, in **\importmodule{some/path?Foo}** the path **some/path** refers to either the sub-directory and relative namespace path of the current directory and namespace outside of an **archive**, or relative to the current **archive**'s top-level namespace and **source** directory, respectively.

The **module Foo** must either be declared in the file `(top-directory)/some/path/Foo[.(lang)].tex`, or in `(top-directory)/some/path[.(lang)].tex` (which are checked in that order).



- Similarly, `\importmodule[Some/Archive]{some/path?Foo}` is resolved like the previous cases, but relative to the archive `Some/Archive` in the `MathHub` directory.

## 7.6 Variables and Sequences

`\vardef` `\vardef{\<name>}[\<options>]{\<notation>}`

Takes the same arguments as `\symdef`, but produces a `variable` rather than a `symbol`. `Variables` definitions are always local to the current `TeX` group and are allowed anywhere (i.e. outside of `modules`).

`\<options>` may include the additional keyword `bind`, in which case the `variable` will be appropriately abstracted away in statements (see also `\varbind`).

Unlike `\symdef`, there is no starred variant `\vardef*` – `variables` always generate a semantic macro.

The semantic macro for a `variable` behaves analogously to that of a `symbol`.

M → `Variables` induces the same `MMT/OMDoc` terms as `symbols` do, except for the head of the term being `<OMV name="..."/>` instead of `<OMS/>`.

`\varnotation` `\varnotation{\<variable>}[\<options>]{\<notation>}`

Takes the exact same arguments as `\notation`, but attaches an additional `notation` to the `variable` `\<variable>` rather than a `symbol`.

`\svar` `\svar[\<name>]{\<text>}`

Semantically marks up `\<text>` as representing a `variable` `\<name>`. The `variable` does not need to have been defined prior. If no `\<name>` is given, `\<text>` will be used as the name.

This is useful in situations like “throwaway expressions” or remarks; e.g.

`$\plus{\svar{n}, \svar{m}}$` means...

`\varseq` `\varseq{\<name>}[\<options>]{\<range>}{\<notation>}`

Declares a new `variable` sequence. The `\<options>` are the same as for `\vardef`. If not provided, `args=1` by default (a 0-ary sequence would just be a normal `variable`).

A `type` (given as `type=`) is interpreted to be the `type` of every element  $a_i$  of the sequence  $a_1, \dots, a_n$  (not of the sequence itself). If the `type` is itself a sequence  $A_1, \dots, A_n$ , the assumption is that its range is the same as the one of the new sequence, and the type of every  $a_i$  in the sequence is  $A_i$ .

`\<range>` needs to be a comma-separated sequence of either `args` many arguments, or `\ellipses`.

The resulting semantic macro is allowed anywhere `STEX` expects an argument mode `a` or `B` argument.

\ellipses Represents ellipses in a range; produces `\ellipses` in math mode.

\seqmap `\seqmap{\langle code \rangle}{\langle sequence \rangle}`

Maps the function  $\langle code \rangle$  (containing #1) over every element of the  $\langle sequence \rangle$ .

Is allowed anywhere STeX expects an **argument mode** a or B argument.

## 7.7 Structures

Mathematical structure bundle interdependent **symbols** together.

**mathstructure** (*env.*)

`\begin{mathstructure}{\langle mname \rangle}[\langle name \rangle, this=\langle code \rangle]` opens a new **mathematical structure** with name  $\langle mname \rangle$  (if provided) or  $\langle name \rangle$  (otherwise), and **semantic macro** `\mname`. It subsequently behaves like the **smodule** environment.

\this

The optional `this=\langle code \rangle` option allows for setting the typesetting of the `\this` macro within a **mathstructure**. In particular, `\this` can be used in **notations** for **symbols** declared in the structure. `\this` can be thought of as representing “the” (current) instance of this **structure**.

**extstructure** (*env.*)

`\begin{extstructure}{\langle mname \rangle}[\langle name \rangle, this=\langle code \rangle]{\langle structs \rangle}` opens a new **mathematical structure** extending the **structures** given in  $\langle structs \rangle$  (a comma-separated list of names).

**extstructure\*** (*env.*)

`\begin{extstructure*}{\langle struct \rangle}` opens a new **mathematical structure** *conservatively* extending the (single) **structure**  $\langle struct \rangle$ . *Conservative* meaning: Every **symbol** newly introduced in this **structure** needs to have a **definiens**. The new **symbols** are attached as fields directly to  $\langle struct \rangle$ .

\usestructure

The `\usestructure` macro behaves like `\usemodule` for **mathematical structures**, making the **symbols** available to use directly.

`\mathstructure` make use of the *Theories-as-Types* paradigm (see [MueRabKoh:tat18]):  
M → `\begin{mathstructure}{\langle name \rangle}` creates a nested **theory** with name  $\langle name \rangle$ -**module**. The **constant**  $\langle name \rangle$  is defined as a *dependent record type with manifest fields*, the fields of which are generated from (and correspond to) the constants in  $\langle name \rangle$ -**module**.

### 7.7.1 Semantic Macros for Structures

Assume we have a **mathematical structure** with **semantic macro** `\struct`:

#### Example 24

```
\begin{mathstructure}{\struct}
\symdef{fielda}{a}
\symdef{fieldb}[args=2]{#1 \maincomp{b} #2}
\symdef{fieldc}[args=2,def=\sn{fieldb}]{#1 \maincomp{c} #2}
```

```

\inliness[name=axiom1]{\conclusion{some axiom}}
\end{mathstructure}
\notation{struct}{StRuCt}

```

- If `\struct` has no `notations`, then `\struct!` produces  $\langle a, b, c \rangle$ . Otherwise, it produces the notation, i.e. `StRuCt`. In both cases, `\struct{}{}` produces  $\langle a, b, c \rangle$  (for reasons that will become clearer in a moment).
- `\struct{}{}` (or `\struct!` in the case where no `notations` are around) can be modified in the following ways:
  - `\struct{}{}[\langle fieldname \rangle, ...]` lets you pick, which precise fields to show, so e.g. `\struct{}{}[fielda, fieldb]` produces  $\langle a, b \rangle$ . By default, `\struct{}{}` shows exactly the fields that have `semantic macros` (which are also used to access the fields in `\struct{\dots}{fieldname}`).
  - `\struct{}{}[\langle id \rangle]` lets you pick the `notation` of the “`mathematical structure`” symbol to use to typeset the `structure`; e.g. `\struct{}{}[parens]` yields  $\langle a, b, c \rangle$ , and (combining both) `\struct{}{}[fielda, fieldb][angle]` yields  $\langle a, b \rangle$ .
- The two arguments in `\struct{first}{second}` represent 1. the term that is to be treated as an instance of `\struct`, and 2. the precise field to invoke. If the first is empty, then there is no instance. If the second is empty, `StTeX` will present all of them (that are not assertions). Hence, `\struct{S}{}{}` yields  $\langle a_S, b_S, c_S \rangle$ , `\struct{S}{fielda}` produces  $a_S$ , `\struct{}{fielda}` produces  $a$ , and `\struct{S}{fieldb}{x}{y}` produces  $x b_S y$ .
- For the sake of completion, `\struct{first}!` simply produces the given argument; e.g. `\struct{S}!` simply produces  $S$ .

More precisely: `\struct{\langle code \rangle}` acts like a “type coercion” of  $\langle code \rangle$  to be an instance of `\struct`.

Of course, it is (occasionally, but) rarely useful to use the `semantic macro` `\struct` by giving it two arguments *manually*; but this is what `StTeX` does when using `\struct` in the `return` of a `symbol` (or `variable`).

Continuing:

- `\struct[comp=\langle code \rangle]{...}{...}` applies  $\langle code \rangle$  (using #1) to every occurrence of `\maincomp` in the `notations` of the fields, as a replacement for the default modification `\#1_{\this}`. For example, `\struct[comp=\#1^{\text{Foo}}]{S}{}{}` produces  $\langle a^{\text{Foo}}, b^{\text{Foo}}, c^{\text{Foo}} \rangle$ , and `\struct[comp=\#1^{\text{\this}}]{S}{fieldb}{x}{y}` yields  $x b_S^{\text{y}}$ .
- `\struct[this=\langle code \rangle]{...}{...}` modifies the way `\this` is being typeset; i.e. the presentation of the first `{...}` argument. For example `\struct[this=T]{S}{}{}` produces  $\langle a, b, c \rangle$  – since `\this` is not being used in the `notations` of the fields. Note that the `this=` and `comp=` variants are (as of yet) mutually incompatible.
- Finally, `\struct[\langle fieldname \rangle=\langle code \rangle]{...}{...}` assigns the field  $\langle fieldname \rangle$  to the term  $\langle code \rangle$ .  $\langle code \rangle$  is subsequently used when using `{...}{}`, but not in fields directly. For example, `\struct[fielda=A]{S}{}{}` produces  $\langle A!, b_S, c_S \rangle$ , but `\struct[fielda=A]{S}{fielda}` produces  $a_S$ .

Note the insertion of ! behind the A – this is to make sure that assignments to [semantic macros](#) that takes arguments don't accidentally eat more than they should.

Also note that multiple assignments can be done in the same pair of [], or chained – i.e. both `$\struct{fielda=A,fieldb=B}...` and `$\struct{fielda=A} [fieldb=B]...` are valid and equivalent. `$\struct{fielda=A,fielda=B}...` however is not – every field may be assigned at most once.

# Chapter 8

## Statements

**STEX** provides four environments to semantically annotate various kinds of statements:

**sdefinition** (*env.*)

The **sdefinition environment** represents (primarily mathematical) *definitions*; in particular for *symbols*. The contents of the environment will be used as *documentation* for any *symbol* that either occurs as a `\definiendum` (or `\definename`) within the **sdefinition**, or that is listed in the optional `for=` argument of the **environment**.

If a `\definiens` occurs, this will be used by **MMT** as the formal definiens for the respective *symbol*.

**sassertion** (*env.*)

The **sassertion environment** represents *assertions*, i.e. *propositions* such as *theorems, lemmata, axioms*, etc. If a `\conclusion` occurs within the **sassertion**, its argument will be used as the formal *statement* of the assertion.

**sexample** (*env.*)

The **sexample environment** represents examples, the `for=` attribute specifies the concept this is an example for. For counterexamples use `style=counterexample`

**sparagraph** (*env.*)

The **sparagraph environment** represents all other kinds of (logical) paragraphs, such as remarks, comments, transitions between topics, recaps, reminders, etc.

All of these take the same arguments:

- `for=(csl)`: a comma-separated list of *symbols*.
- The same optional arguments as `\symdecl`, with `macro=` replacing the name of the *semantic macro*. All of them are only relevant, if either `name=` or `macro=` are provided.

As with `\symdecl`, if no `name` is given, but `macro` is, then the same name is used for both the *semantic macro* and the *symbol* itself.

If `name` is given but `macro` isn't, no *semantic macro* is generated. Subsequently, the newly generated *symbol* is added the `for`-list.

- `style`: see [chapter 9](#).
- `title`: a title to use in various styles (see [chapter 9](#)).
- `id`: a label to use for `\sref`.

\inlineass The macros `\inlineass`, `\inlinedef` and `\inlineex` behave like the `sassertion`, `sdefinition` and `sexample` environments respectively, but take the text to annotate as an argument, rather than as the body of an environment, and do not break paragraphs.

The same macros available in the environments are also available in the argument of the `\inline*` macros.

\varbind `\varbind{\langle cls \rangle}`

retroactively attaches the `bind` option to every `variable` provided (as a comma-separated list).

## 8.1 More on Definitions

In `sdefinition` (and `sparagraph` with `style=symdoc`), the following additional macros are available:

\definiendum The `\definiendum` macro behaves largely like `\symref`, but it uses a dedicated highlighting for `definienda` and adds the referenced `symbol` to the `for=` list of the environment.  
\defname `\defname` is to `\definiendum` as `\symname` is to `\symref`. Analogously, `\Defname` behaves like `\Symname`.

`\defnotation` can be used in math mode to apply the `\definiendum` highlighting to `notations`.

\definiens The `\definiens` macro can be used to semantically annotate the `definiens` in a `sdefinition`.

If the `sdefinition` environment has several elements in its `for` list, an optional argument `\definiens[\langle symbol \rangle]{...}` can be used to tell `STEX` which `symbol`'s definiens this is. By default, the *first* `symbol` in the `for` list is used.

Here is how `MMT` will treat the fragment marked up with `\definiens`:

Firstly, it will attempt to translate its contents into an `MMT/OMDoc` term. This succeeds easily if `\definiens` is some semantic macro (applied to arguments).

Secondly, it will check for all `variables` currently in scope, that were defined with the optional argument `bind`. It then will check, whether a `symbol` is in scope, that has `role=lambda`. If so, it will use that `lambda symbol` to bind all these variables (in the order in which they were defined) in the term. If no `lambda symbol` is found, it will use the `bind symbol` that ships with `STEX`.

The final term will be attached as definiens to the corresponding `MMT` constant, if it was declared in the same `module` as the `\definiens` occurrence.

## 8.2 More on Assertions

\premise  
\conclusion

The `\conclusion` macro can be used to mark up the actual statement within an `sassertion`. The `\premise` macro can be used to additionally mark up *premises*.

If the `sassertion` environment has several elements in its `for` list, an optional argument `\conclusion[⟨symbol⟩]{...}` can be used to tell `STEX` which `symbol`'s statement this is. By default, the *first symbol* in the `for` list is used.

Here is how `MMT` will treat the fragments marked up with `\conclusion` and `\premise`:

Firstly, it will attempt to translate the contents of `\conclusion` into an `MMT/OM-Doc` term  $c$ . This succeeds easily if the `\conclusion` is some semantic macro (applied to arguments).

Secondly, it will collect all fragments marked up with `\premise` and do the same to them ( $p_1, \dots, p_n$ ).

It will then check, whether a `symbol` is in scope, that has `role=implication`. If so, it will use that `implication symbol` to attach all the premises to the conclusion, resulting in  $t := \text{imply}(p_1, \dots, p_n, c)$ .

Next, it will check for all `variables` currently in scope, that were defined with the optional argument `bind`. It then will check, whether a `symbol` is in scope, that has `role forall`. If so, it will use that `forall symbol` to bind all these variables (in the order in which they were defined) in the term  $t$ .

Finally, it will check, whether a `symbol` is in scope, that has `role=judgment`. If so, it will set  $t := \text{judgment}(t)$ .

If no `forall symbol` is found, it will first apply the `judgment symbol` (if existent) and then use the `bind symbol` that ships with `STEX` to bind the `variables`.

The final term will be attached as `type` to the corresponding `MMT constant`, if it was declared in the same `module` as the `\definiens` occurrence.

`sproof (env.)`

TODO<sup>14</sup>

---

<sup>14</sup>TODO: proofs

# Chapter 9

# Customizing Typesetting

There are two kinds of typesetting that can be customized in **STEX**: **symbol** references (**notation** components, `\symref`, **variables**, etc.) are highlighted using a small set of **macros** that can be simply redefined by authors.

Other **macros** and **environments** usually have more complicated “typesetting rules” associated with them – often in the form of other already existing **environments** that should be used.

Lastly, in **HTML** we can provide custom CSS rules in **math archives** that determine the styling of certain **environments**, so that the actual presentation depends on the document in which the fragments are included (e.g. via `\inputref`), rather than the file the fragment is implemented in.

It is generally recommended to implement these customizations in a preamble in the `lib` directory (see [section 5.3](#))

## 9.1 Highlighting Symbol References

---

`\symrefemph`  
`\symrefemph@uri`

`\symrefemph` governs how references via `\symref` (or `\symname`, or their short variants) are highlighted;

Doing `\symref{<symbol>}{<text>}` ultimately expands to `\symrefemph@uri{<text>}{{<symbol URI>}}`, the default implementation of which is just `\symrefemph{<text>}`. The default implementation of `\symrefemph`, in turn, is just `\emph{<text>}`.

If you only want to change e.g. the color of `\symrefs`, you only need to redefine `\symrefemph`, e.g. using

```
\renewcommand\symrefemph[1]{\textcolor{red}{#1}}
```

the `@uri` variant is useful if you want to link somewhere, or show the URI in a tooltip. The **stex-highlighting package** does both, using:

```
\usepackage{pdfcomment}
\protected\def\symrefemph@uri#1#2{%
  \pdftooltip{%
    \srefsymuri{#2}{\symrefemph{#1}}%
  }{%
    URI:~\detokenize{#2}%
  }%
```

```

}
\def\symrefemph#1{%
  \ifcsname textcolor\endcsname
    \textcolor{teal}{#1}%
  \else#1\fi
}

```

---

`\compemph` Like `\symrefemph` and `\symrefemph@uri`, but governs the highlighting of components (marked with `\comp` or `\maincomp`) in `notations`.

---

`\defemph` Like `\symrefemph` and `\symrefemph@uri`, but governs the highlighting of definienda marked with `\definiendum` (or `\definename`).

---

`\varemph` Like `\compemph` and `\compemph@uri`, but governs the highlighting of components (marked with `\comp` or `\maincomp`) in the `notations` of `variables`.

The second argument to `\varemph@uri` is the *name* of the `variable`.

## 9.2 Styling Environments and Macros

A variety of `environments` and `macros` provided by `STEX` are *stylistable* using the `macros` `\stextstyle{name}[\langle style \rangle]{...}`. These *stylistable environments* and `macros` bind various of their parameters to `macros` `\this{parameter}`, which can then be used in the styles.

For example, if we have a `definition environment` that we would want to use to style our `sdefinitions`, we can do (in the simplest case)

```
\stextstyledefinition{\begin{definition}}{\end{definition}}
```

This tells `STEX` to insert `\begin{definition}` at the beginning of every `sdefinition environment`, and `\end{definition}` at the end.

If we have a `environments theorem` and `lemma`, we probably want the `sassertion environment` to use those for theorems and lemma. We can achieve that by doing

```
\stextstyleassertion[theorem]{\begin{theorem}}{\end{theorem}}
\stextstyleassertion[lemma]{\begin{lemma}}{\end{lemma}}
```

Now if we do `\begin{sassertion}[style=theorem]`, it will wrap the `environment` with `\begin{theorem}... \end{theorem}`.

Of course, many such statements might have a title, as e.g. in

**Theorem 9.2.1** (Gödel's First Incompleteness Theorem). ...

In `sassertion`, we can provide that title as optional argument using `title=....` Before calling the styling provided, `sassertion` will store that title in the macro `\thistitle`, which we can use in the styling. For example, we might prefer to pass it on to the `theorem environment`:

```
\stextstyleassertion[theorem]{\ifx\thistitle\empty
  \begin{theorem}\else\begin{theorem}[\thistitle]\fi
  \end{theorem}}
```

---

```
\stexstylemodule           TODO15
```

```
\stexstylecopymodule
\stexstyleinterpretmodule
\stexstylerealization
\stexstylemathstructure
\stexstyleextstructure
\stexstyledefinition
\stexstyleassertion
\stexstyleexample
\stexstyleparagraph
\stexstyleproof
\stexstylesubproof
```

---

Additionally, we can style certain [macros](#), if we want them to produce output. For example, we might (for debuggin or documentation purposes) `\symdecl` to give a short summary of the symbol.

We can achieve that by doing, for example:

```
\stexstylesymdecl[debug]{
    Symbol \thisdeclname~(with arity \thisargs) of type \$\thistype$.
```

in which case

```
\symdecl{foo}[args=2,type=\mathbb{N},style=debug]
```

will yield

Symbol foo (with arity ii) of type N.

---

```
\stexstyleusemodule           TODO16
```

```
\stexstyleimportmodule
\stexstylerequiremodule
\stexstyleassign
\stexstylerenamedecl
\stexstyleassignMorphism
\stexstylecopymod
\stexstyleinterpretmod
\stexstylerealize
\stexstylesymdecl
\stexstyleextsymdecl
\stexstylenotation
\stexstylevarnotation
\stexstylesymdef
\stexstylevardef
\stexstylevarseq
\stexstylepsfesketch
\stexstyleMMTinclude
```

---

### 9.3 Custom CSS for Environments

# Chapter 10

## Additional Packages

### 10.1 NotesSlides Manual

#### 10.1.1 Introduction

The `notesslides` document class is derived from `beamer.cls` [`beamerclass:on`], it adds a “notes version” for course notes that is more suited to printing than the one supplied by `beamer.cls`.

The `notesslides` class takes the notion of a slide frame from Till Tantau’s excellent `beamer` class and adapts its notion of frames for use in the `STEX` and OMDoc. To support semantic course notes, it extends the notion of mixing frames and explanatory text, but rather than treating the frames as images (or integrating their contents into the flowing text), the `notesslides` package displays the slides as such in the course notes to give students a visual anchor into the slide presentation in the course (and to distinguish the different writing styles in slides and course notes).

In practice we want to generate two documents from the same source: the slides for presentation in the lecture and the course notes as a narrative document for home study. To achieve this, the `notesslides` class has two modes: *slides mode* and *notes mode* which are determined by the package option.

#### 10.1.2 Package Options

The `notesslides` class takes a variety of class options:

`slides` The options `slides` and `notes` switch between slides mode and notes mode (see [subsection 10.1.3](#)).

`sectocframes` If the option `sectocframes` is given, then for the `sfragments`, special frames with the `sfragment` title (and number) are generated.

`frameimages` If the option `frameimages` is set, then slide mode also shows the `\frameimage`-generated `fiboxed` frames (see [??](#)). If also the `fiboxed` option is given, the slides are surrounded by a box.

### 10.1.3 Notes and Slides

**frame** (*env.*) Slides are represented with the **frame** environment just like in the **beamer** class, see [Tantau:ugbc] for details.

**note** (*env.*) The **notesslides** class adds the **note** environment for encapsulating the course note fragments.



Note that it is essential to start and end the **notes** environment at the start of the line – in particular, there may not be leading blanks – else L<sup>A</sup>T<sub>E</sub>X becomes confused and throws error messages that are difficult to decipher.

By interleaving the **frame** and **note** environments, we can build course notes as shown here:

```
1 \ifnotes\maketitle\else
2 \frame[noframenumbering]\maketitle\fi
3
4 \begin{note}
5   We start this course with ...
6 \end{note}
7
8 \begin{frame}
9   \frametitle{The first slide}
10 ...
11 \end{frame}
12 \begin{note}
13   ... and more explanatory text
14 \end{note}
15
16 \begin{frame}
17   \frametitle{The second slide}
18 ...
19 \end{frame}
20 ...
```

---

**\ifnotes** Note the use of the **\ifnotes** conditional, which allows different treatment between **notes** and **slides** mode – manually setting **\notestrue** or **\notesfalse** is strongly discouraged however.



We need to give the title frame the **noframenumbering** option so that the frame numbering is kept in sync between the slides and the course notes.



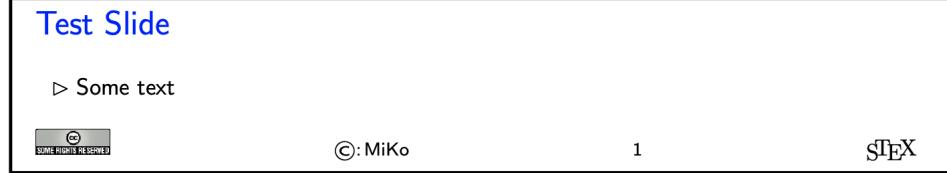
The **beamer** class recommends not to use the **allowframebreaks** option on frames (even though it is very convenient). This holds even more in the **notesslides** case: At least in conjunction with **\newpage**, frame numbering behaves funny (we have tried to fix this, but who knows).

`\inputref*` If we want to transclude a the contents of a file as a note, we can use a new variant `\inputref*` of the `\inputref` macro: `\inputref*{foo}` is equivalent to `\begin{note}\inputref{foo}\end{note}`.

`nparagraph (env.)` There are some environments that tend to occur at the top-level of `note` environments.  
`nparagraph (env.)` We make convenience versions of these: e.g. the `nparagraph` environment is just an  
`ndefinition (env.)` `sparagraph` inside a `note` environment (but looks nicer in the source, since it avoids one  
`nexample (env.)` level of source indenting). Similarly, we have the `nfragment`, `ndefinition`, `nexample`,  
`nsproof (env.)` `nsproof`, and `nassertion` environments.  
`nassertion (env.)`

#### 10.1.4 Customizing Header and Footer Lines

The `notesslides` package and class comes with a simple default theme named `sTeX` that provided by the `beamterthemesTeX`. It is assumed as the default theme for `sTeX`-based notes and slides. The result in `notes` mode (which is like the `slides` version except that the slide height is variable) is



The footer line can be customized. In particular the logos.

`\setslidelogo` The default logo provided by the `notesslides` package is the `sTeX` logo it can be customized using `\setslidelogo{\langle logo name \rangle}`.

`\setsource` The default footer line of the `notesslides` package mentions copyright and licensing. In `notesslides` `\source` stores the author's name as the copyright holder. By default it is the author's name as defined in the `\author` macro in the preamble. `\setsource{\langle name \rangle}` can change the writer's name.

`\setlicensing` For licensing, we use the Creative Commons Attribution-ShareAlike license by default to strengthen the public domain. If package `hyperref` is loaded, then we can attach a hyperlink to the license logo. `\setlicensing[\langle url \rangle]{\langle logo name \rangle}` is used for customization, where `\langle url \rangle` is optional.

#### 10.1.5 Frame Images

Sometimes, we want to integrate slides as images after all – e.g. because we already have a PowerPoint presentation, to which we want to add `sTeX` notes.

---

```
\frameimage  
\mhframeimage
```

In this case we can use `\frameimage[⟨opt⟩]{⟨path⟩}`, where `⟨opt⟩` are the options of `\includegraphics` from the `graphicx` package [CarRah:tpp99] and `⟨path⟩` is the file path (extension can be left off like in `\includegraphics`). We have added the `label` key that allows to give a frame label that can be referenced like a regular `beamer` frame.

The `\mhframeimage` macro is a variant of `\frameimage` with repository support. Instead of writing

```
1 \frameimage{\MathHub{fooMH/bar/source/baz/foobar}}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
1 \mhframeimage[fooMH/bar]{baz/foobar}
```

Note that the `\mhframeimage` form is more semantic, which allows more advanced document management features in `MathHub`.

If `baz/foobar` is the “current module”, i.e. if we are on the `MathHub` path `...MathHub/fooMH/bar...`, then stating the repository in the first optional argument is redundant, so we can just use

```
1 \mhframeimage{baz/foobar}
```

---

```
\textwarning
```

The `\textwarning` macro generates a warning sign: 

### 10.1.6 Ending Documents Prematurely

---

```
\prematurestop  
\afterprematurestop
```

For prematurely stopping the formatting of a document, S<sup>T</sup>E<sub>X</sub> provides the `\prematurestop` macro. It can be used everywhere in a document and ignores all input after that – backing out of the `sfragment` environments as needed. After that – and before the implicit `\end{document}` it calls the internal `\afterprematurestop`, which can be customized to do additional cleanup or e.g. print the bibliography.

`\prematurestop` is useful when one has a driver file, e.g. for a course taught multiple years and wants to generate course notes up to the current point in the lecture. Instead of commenting out the remaining parts, one can just move the `\prematurestop` macro. This is especially useful, if we need the rest of the file for processing, e.g. to generate a theory graph of the whole course with the already-covered parts marked up as an overview over the progress; see `import_graph.py` from the `lmhtools` utilities [lmhtools:github:on].

Text fragments and modules can be made more re-usable by the use of global variables. For instance, the admin section of a course can be made course-independent (and therefore re-usable) by using variables (actually token registers) `courseAcronym` and `courseTitle` instead of the text itself. The variables can then be set in the S<sup>T</sup>E<sub>X</sub> preamble of the course notes file.

### 10.1.7 Global Document Variables

To make document fragments more reusable, we sometimes want to make the content depend on the context. We use **document variables** for that.

---

**\setSGvar** `\setSGvar{<vname>}{<text>}` to set the global variable `<vname>` to `<text>` and **\useSGvar** `\useSGvar{<vname>}` to reference it.

---

**\ifSGvar** With `\ifSGvar` we can test for the contents of a global variable: the macro call `\ifSGvar{<vname>}{<val>}{<ctext>}` tests the content of the global variable `<vname>`, only if (after expansion) it is equal to `<val>`, the conditional text `<ctext>` is formatted.

### 10.1.8 Excursions

In course notes, we sometimes want to point to an “excursion” – material that is either presupposed or tangential to the course at the moment – e.g. in an appendix. The typical setup is the following:

```
1 \excursion{founif}{../fragments/founif.en}
2 {We will cover first-order unification in}
3 ...
4 \begin{appendix}\printexcursions\end{appendix}
```

It generates a paragraph that references the excursion whose source is in the file `../fragments/founif.en.tex` and automatically books the file for the `\printexcursions` command that is used here to put it into the appendix. We will look at the mechanics now.

---

**\excursion** The `\excursion{<ref>}{<path>}{<text>}` is syntactic sugar for

```
1 \begin{nparagraph}[title=Excursion]
2 \activateexcursion{founif}{../ex/founif}
3 We will cover first-order unification in \sref{founif}.
4 \end{nparagraph}
```

---

**\activateexcursion** Here `\activateexcursion{<path>}` augments the `\printexcursions` macro by a call `\inputref{<path>}`. In this way, the `\printexcursions` macro (usually in the appendix) will collect up all excursions that are specified in the main text.

Sometimes, we want to reference – in an excursion – part of another. We can use `\excursionref{<label>}` for that.

---

**\excursiongroup** Finally, we usually want to put the excursions into an `sfragment` environment and add an introduction, therefore we provide the a variant of the `\printexcursions` macro: `\excursiongroup[id=<id>, intro=<path>]` is equivalent to

```
1 \begin{note}
2 \begin{sfragment}[id=<id>]{Excursions}
3 \inputref{<path>}
4 \printexcursions
5 \end{sfragment}
6 \end{note}
```



When option `book` which uses `\pagestyle{headings}` is given and semantic macros are given in the `sfragment` titles, then they sometimes are not defined by the time the heading is formatted. Need to look into how the headings are made. This is a problem of the underlying document-structure package.

Local Variables: mode: latex TeX-master: .../**stex-manual** End:

## 10.2 Problem Manual

### 10.2.1 Introduction

The `problem` package supplies an infrastructure that allows specify problem. Problems are text fragments that come with auxiliary functions: hints, notes, and solutions. Furthermore, we can specify how long the solution to a given problem is estimated to take and how many points will be awarded for a perfect solution.

Finally, the `problem` package facilitates the management of problems in small files, so that problems can be re-used in multiple environment.

### 10.2.2 Problems and Solutions

---

`solutions` The `problem` package takes the options `solutions` (should solutions be output?), `notes` (should the problem notes be presented?), `hints` (do we give the hints?), `gnotes` (do we show grading notes?), `pts` (do we display the points awarded for solving the problem?), `min` (do we display the estimated minutes for problem soling). If theses are specified, then the corresponding auxiliary parts of the problems are output, otherwise, they remain invisible.

`boxed` The `boxed` option specifies that problems should be formatted in framed boxes so that they are more visible in the text. Finally, the `test` option signifies that we are in a test situation, so this option does not show the solutions (of course), but leaves space for the students to solve them.

`problem (env.)` The main environment provided by the `problem` package is (surprise surprise) the `problem` environment. It is used to mark up problems and exercises. The environment takes an optional KeyVal argument with the keys `id` as an identifier that can be reference later, `pts` for the points to be gained from this exercise in homework or quiz situations, `min` for the estimated minutes needed to solve the problem, and finally `title` for an informative title of the problem.

#### Example 25

Input:

```

1 \documentclass{article}
2 \usepackage[solutions,hints,pts,min]{problem}
3 \begin{document}
4 \begin{sproblem}[id=elefants,pts=10,min=2,title=Fitting Elefants]
5 How many Elefants can you fit into a Volkswagen beetle?
6 \begin{hint}
7 Think positively, this is simple!
8 \end{hint}
9 \begin{exnote}
10 Justify your answer
11 \end{exnote}
12 \begin{solution}
13 Four, two in the front seats, and two in the back.
14 \begin{gnote}
15 if they do not give the justification deduct 5 pts
16 \end{gnote}
17 \end{solution}
18 \end{sproblem}
19 \end{document}

```

Output:

### Exercise (Fitting Elefants)

How many Elefants can you fit into a Volkswagen beetle?

*Lösung:* Four, two in the front seats, and two in the back.

**solution (env.)** The **solution** environment can be used to specify a solution to a problem. If the package option **solutions** is set or **\solutionstrue** is set in the text, then the solution will be presented in the output. The **solution** environment takes an optional KeyVal argument with the keys **id** for an identifier that can be referenced for to specify which problem this is a solution for, and **height** that allows to specify the amount of space to be left in test situations (i.e. if the **test** option is set in the **\usepackage** statement).

**hint (env.)** The **hint** and **exnote** environments can be used in a **problem** environment to give hints  
**exnote (env.)** and to make notes that elaborate certain aspects of the problem. The **gnote** (grading  
**gnote (env.)** notes) environment can be used to document situations that may arise in grading.

---

**\start solutions** Sometimes we would like to locally override the **solutions** option we have given to  
**\stop solutions** the package. To turn on solutions we use the **\start solutions**, to turn them off,  
**\stop solutions**. These two can be used at any point in the documents.

---

**\if solutions** Also, sometimes, we want content (e.g. in an exam with master solutions) conditional  
on whether solutions are shown. This can be done with the **\if solutions** conditional.

### 10.2.3 Markup for Added-Value Services

The `problem` package is all about specifying the meaning of the various moving parts of practice/exam problems. The motivation for the additional markup is that we can base added-value services from these, for instance auto-grading and immediate feedback.

The simplest example of this are multiple-choice problems, where the `problem` package allows to annotate answer options with the intended values and possibly feedback that can be delivered to the users in an interactive setting. In this section we will give some infrastructure for these, we expect that this will grow over time.

#### Multiple Choice Blocks

`mcb` (*env.*) Multiple choice blocks can be formatted using the `mcb` environment, in which single choices are marked up with `\mcc` macro.

`\mcc` `\mcc[<keyvals>]{<text>}` takes an optional key/value argument `<keyvals>` for choice metadata and a required argument `<text>` for the proposed answer text. The following keys are supported

- `T` for true answers, `F` for false ones,
- `Ttext` the verdict for true answers, `Ftext` for false ones, and
- `feedback` for a short feedback text given to the student.

What we see when this is formatted to PDF depends on the context. In solutions mode (we start the solutions in the code fragment below) we get

#### Example 26

Input:

```
1 \startSolutions
2 \begin{sproblem}[title=Functions,name=functions1]
3 What is the keyword to introduce a function definition in python?
4 \begin{mcb}
5   \mcc[T]{def}
6   \mcc[F,feedback=that is for C and C++]{function}
7   \mcc[F,feedback=that is for Standard ML]{fun}
8   \mcc[F,Ftext=Noooooooooooo,feedback=that is for Java]{public static void}
9 \end{mcb}
10 \end{sproblem}
```

Output:

### Exercise (Functions)

What is the keyword to introduce a function definition in python?

- def<sup>a</sup>
- function<sup>b</sup>
- fun<sup>c</sup>
- public static void<sup>d</sup>

<sup>a</sup>Korrekt

<sup>b</sup>Falsch

*that is for C and C++*

<sup>c</sup>Falsch

*that is for Standard ML*

<sup>d</sup>Noooooooooo

*that is for Java*

In “exam mode” where disable solutions (here via `\stopsolutions`)

### Example 27

Input:

```
1 \stopsolutions
2 \begin{sproblem}[title=Functions,name=functions1]
3   What is the keyword to introduce a function definition in python?
4   \begin{mcb}
5     \mcc[T]{def}
6     \mcc[F,feedback=that is for C and C++]{function}
7     \mcc[F,feedback=that is for Standard ML]{fun}
8     \mcc[F,Ftext=Noooooooooo,feedback=that is for Java]{public static void}
9   \end{mcb}
10 \end{sproblem}
```

Output:

### Exercise (Functions)

What is the keyword to introduce a function definition in python?

- def
- function
- fun
- public static void

we get the questions without solutions (that is what the students see during the exam/quiz).

### Filling-In Concrete Solutions

The next simplest situation, where we can implement auto-grading is the case where we have fill-in-the-blanks

**\fillinsol** The `\fillinsol` macro takes a single argument, which contains a concrete solution (i.e. a number, a string, ...), which generates a fill-in-box in test mode:

**Example 28**

Input:

```
1 \stopsolutions
2 \begin{sproblem}[id=elefants.fillin,title=Fitting Elefants]
3 How many Elefants can you fit into a Volkswagen beetle? \fillinsol{4}
4 \end{sproblem}
```

**Exercise (Fitting Elefants)**  
and the actual solution in solutions mode:

**Example 29** Many Elefants can you fit into a Volkswagen beetle?

Input:

```
1 \begin{sproblem}[id=elefants.fillin,title=Fitting Elefants]
2 How many Elefants can you fit into a Volkswagen beetle? \fillinsol{4}
3 \end{sproblem}
```

Output:

**Exercise (Fitting Elefants)**

How many Elefants can you fit into a Volkswagen beetle?

If we do not want to leak information about the solution by the size of the blank we can also give `\fillinsol` an optional argument with a size: `\fillinsol[3cm]{12}` makes a box three cm wide.

Obviously, the required argument of `\fillinsol` can be used for auto-grading. For concrete data like numbers, this is immediate, for more complex data like strings “soft comparisons” might be in order. <sup>17</sup>

#### 10.2.4 Including Problems

**\includeproblem** The `\includeproblem` macro can be used to include a problem from another file. It takes an optional KeyVal argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one problem in the include file). The keys `title`, `min`, and `pts` specify the problem title, the estimated minutes for solving the problem and the points to be gained, and their values (if given) overwrite the ones specified in the `problem` environment in the included file.

The sum of the points and estimated minutes (that we specified in the `pts` and `min` keys to the `problem` environment or the `\includeproblem` macro) to the log file and the

<sup>17</sup>For the moment we only assume a single concrete value as correct. In the future we will almost certainly want to extend the functionality to multiple answer classes that allow different feedback like in MCQ. This still needs a bit of design. Also we want to make the formatting of the answer in solutions/test mode configurable.

screen after each run. This is useful in preparing exams, where we want to make sure that the students can indeed solve the problems in an allotted time period.

The `\min` and `\pts` macros allow to specify (i.e. to print to the margin) the distribution of time and reward to parts of a problem, if the `pts` and `pts` options are set. This allows to give students hints about the estimated time and the points to be awarded.

### 10.2.5 Testing and Spacing

The `problem` package is often used by the `hwexam` package, which is used to create homework assignments and exams. Both of these have a “test mode” (invoked by the package option `test`), where certain information –master solutions or feedback – is not shown in the presentation.

```
\testspace      \testspace takes an argument that expands to a dimension, and leaves vertical space accordingly. Specific instances exist: \testsmallspace, \testsmallspace, \testsmallspace \testsmallspace give small (1cm), medium (2cm), and big (3cm) vertical space.  
\testsmallspace \testnewpage makes a new page in test mode, and \testemptypage generates an empty page with the cautionary message that this page was intentionally left empty.  
\testemptypage Local Variables: mode: latex TeX-master: .../stex-manual End:  
LocalWords: solutions,notes,hints,gnotes,pts,min,boxed,test gnotes elefants,pts gnote  
LocalWords: 2,title exnote hint,exnote,gnote ifsolutions mcb keyvals Ttext Ftext Local-  
Words: Functions,name F,feedback Noooooooooo,feedback 2,title includeproblem Local-  
Words: elefants.fillin,title
```

## 10.3 HWExam Manual

### 10.3.1 Introduction

The `hwexam` package and class supplies an infrastructure that allows to format nice-looking assignment sheets by simply including problems from problem files marked up with the `problem` package. It is designed to be compatible with `problems.sty`, and inherits some of the functionality.

### 10.3.2 Package Options

---

`solutions` The `hwexam` package and class take the options `solutions`, `notes`, `hints`, `gnotes`, `pts`, `notes`, `min`, and `boxed` that are just passed on to the `problems` package (cf. its documentation for a description of the intended behavior).

`multiple` Furthermore, the `hwexam` package takes the option `multiple` that allows to combine multiple assignment sheets into a compound document (the assignment sheets are treated as section, there is a table of contents, etc.).

`test` Finally, there is the option `test` that modifies the behavior to facilitate formatting tests. Only in `test` mode, the macros `\testspace`, `\testnewpage`, and `\testemptypage` have an effect: they generate space for the students to solve the given problems. Thus they can be left in the L<sup>A</sup>T<sub>E</sub>X source.

### 10.3.3 Assignments

**assignment** (*env.*) This package supplies the **assignment** environment that groups problems into assignment sheets. It takes an optional KeyVal argument with the keys **number** (for the assignment number; if none is given, 1 is assumed as the default or — in multi-assignment documents **title** — the ordinal of the **assignment** environment), **title** (for the assignment title; this is **type** referenced in the title of the assignment sheet), **type** (for the assignment type; e.g. “quiz”, **given** or “homework”), **given** (for the date the assignment was given), and **due** (for the date **due** the assignment is due).

### 10.3.4 Including Assignments

---

**\inputassignment** The **\inputassignment** macro can be used to input an assignment from another file. It takes an optional KeyVal argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one **assignment** environment in the included file). The keys **number**, **title**, **type**, **given**, and **due** are just as for the **assignment** environment and (if given) overwrite the ones specified in the **assignment** environment in the included file.

### 10.3.5 Typesetting Exams

**testheading** (*env.*) The **\testheading** takes an optional keyword argument where the keys **duration** specifies a string that specifies the duration of the test, **min** specifies the equivalent in number of minutes, and **reqpts** the points that are required for a perfect grade.

```
reqpts1 \title{320101 General Computer Science (Fall 2010)}
2 \begin{testheading}[duration=one hour,min=60,reqpts=27]
3 Good luck to all students!
4 \end{testheading}
```

Will result in

Name:

Matriculation Number:

## 320101 General Computer Science (Fall 2010)

2023-11-24

**You have one hour (sharp) for the test;**

Write the solutions to the sheet.

The estimated time for solving this exam is 23 minutes, leaving you 37 minutes for revising your exam.

You can reach 23 points if you solve all problems. You will only need 27 points for a perfect score, i.e. -4 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

|         | To be used for grading, do not write here |     |     |     |     |     |     |  |     |     |
|---------|-------------------------------------------|-----|-----|-----|-----|-----|-----|--|-----|-----|
| prob.   | 4.1                                       | 1.1 | 1.2 | 2.1 | 2.2 | 2.3 | 2.4 |  | 2.5 | 2.6 |
| total   | 0                                         | 0   | 0   | 10  | 0   | 0   | 0   |  | 0   | 0   |
| reached |                                           |     |     |     |     |     |     |  |     |     |

good luck

18

Local Variables: mode: latex TeX-master: `../stex-manual` End:

LocalWords: hwexam solutions,notes,hints,gnotes,pts,min gnotes testemptypage reqpts LocalWords: inputassignment reqpts hour,min 60,reqpts

### 10.4 Tikzinput Manual

**image** The behavior of the `ikzinput` package is determined by whether the `image` option is given. If it is not, then the `tikz` package is loaded, all other options are passed on to it and `\tikzinput{<file>}` inputs the TIKZ file `<file>.tex`; if not, only the `graphicx` package is loaded and `\tikzinput{<file>}` loads an image file `<file>.ext` generated from `<file>.tex`.

The selective input functionality of the `tikzinput` package assumes that the TIKZ pictures are externalized into a standalone picture file, such as the following one

```
1 \documentclass{standalone}
2 \usepackage{tikz}
```

<sup>18</sup>MK: The first three “problems” come from the stex examples above, how do we get rid of this?

```

3 \usetikzpackage{...}
4 \begin{document}
5   \begin{tikzpicture}
6     ...
7   \end{tikzpicture}
8 \end{document}

```

The `standalone` class is a minimal L<sup>A</sup>T<sub>E</sub>X class that when loaded in a document that uses the `standalone` package: the preamble and the `document` environment are disregarded during loading, so they do not pose any problems. In effect, an `\input` of the file above only sees the `tikzpicture` environment, but the file itself is standalone in the sense that we can run L<sup>A</sup>T<sub>E</sub>X over it separately, e.g. for generating an image file from it.

---

**`\tikzinput`** This is exactly where the `tikzinput` package comes in: it supplies the `\tikzinput` macro, which – depending on the `image` option – either directly inputs the TIKZ picture (source) or tries to load an image file generated from it.

Concretely, if the `image` option is not set for the `tikzinput` package, then `\tikzinput[<opt>]{<file>}` disregards the optional argument `<opt>` and inputs `<file>.tex` via `\input` and resizes it to as specified in the `width` and `height` keys. If it is, `\tikzinput[<opt>]{<file>}` expands to `\includegraphics[<opt>]{<file>}`.

`\ctikzinput` is a version of `\tikzinput` that is centered.

---

**`\mhtikzinput`** `\mhtikzinput` is a variant of `\tikzinput` that treats its file path argument as a relative path in a math archive in analogy to `\inputref`. To give the archive path, we use the `mhrepos=` key. Again, `\cmhtikzinput` is a version of `\mhtikzinput` that is centered.

---

**`\libusetikzlibrary`** Sometimes, we want to supply archive-specific TIKZ libraries in the `lib` folder of the archive or the `meta-inf/lib` of the archive group. Then we need an analogon to `\libinput` for `\usetikzlibrary`. The `stex-tikzinput` package provides the `libusetikzlibrary` for this purpose.

Local Variables: mode: latex TeX-master: .../stex-manual End:

## **Part III**

## **Documentation**

# Chapter 11

## sTeX Developer Manual



Keeping the implementation properly up-to-date is pretty much incompatible with the kinds of workflows systemically enforced in academia. Any of the following may be out of date.

\* indicates fully expandable functions, which can be used within an x-type argument (in plain TeX terms, inside an `\edef`), as well as within an f-type argument. ☆ indicates restricted expandable functions, which can be used within an x-type argument but cannot be fully expanded within an f-type argument. *TF* indicates conditional (if) functions whose variants with T, F and TF argument specifiers expect different “true”/“false” branches.

---

`\stex_debug:nn \stex_debug:nn {⟨prefix⟩} {⟨msg⟩}`

Logs the debug message {⟨msg⟩} under the prefix {⟨prefix⟩}. A message is shown if its prefix is in a list of prefixes given either via the package option `debug=⟨prefixes⟩` or the environment variable `STEX_DEBUG=⟨prefixes⟩`, where the latter overrides the former.

---

`\_stex_do_deprecation:n` TODO `\_stex_do_deprecation:n`

## 11.1 Documents

`\l_stex_docheader_sect` integer register keeping track of the current sectioning level:

- 0 part
- 1 chapter
- 2 section
- 3 subsection
- 4 subsubsection
- 5 paragraph
- > 5 subparagraph

`\setsectionlevel` sets `\l_stex_docheader_sect` to the corresponding integer value.

`\stex_ref_new_doc_target:n` internal variant of `\sreflabel`. If the argument is empty, the label is determined to be `REF<counter>` and `<counter>` is increased.

`\stex_ref_new_symbol:n` registers a new link target for the symbol with the given full uri (as string), using the `url-base`-field of the current archive's manifest file to link the symbol to `<url-base>/symbol?<uri>`.

`\stex_ref_new_sym_target:n` sets a new label for the symbol with the given full uri (as string). If called in sms-mode, defers to `\stex_ref_new_symbol:n`, if not already registered. Otherwise, sets a `\label` for the symbol.

`\stex_ref_new_sym_target:nn` `\stex_ref_new_sym_target:nn {<symbol>} {<target>}`

redirects links for `<symbol>` to the one for the symbol `<target>`. Useful for e.g. symbols elaborated from structural features. Note that this acts as a *default*, in that previous or subsequent calls of `\stex_ref_new_sym_target:n{<symbol>}` are prioritized.

Requires that either `\stex_ref_new_sym_target:n{<target>}` or `\stex_ref_new_sym_target:nn{<target>} {<other-target>}` have been called previously.

## 11.2 Modules

The contents of a module with full URI `<uri>` are represented as four macros:

- `\c_stex_module_<uri>_code`: code to be executed every time the module is activated; e.g. the contents of `\STEXexport`, defines semantic macros and macros for notations, activates dependency modules, etc.

- `\c_stex_module_<uri>_morphisms_prop`: property list containing all module dependencies of this module (see `\stex_module_add_morphism:nnn`).
- `\c_stex_module_<uri>symbols_prop`: property list containing all declarations in this module (see `\stex_module_add_symbol:nnnnnnN`).
- `\c_stex_module_<uri>_notations_prop`: property list containing all notations introduced in this module (see `\stex_add_module_notation:nnnn`).

---

`\l_stex_current_module_str` contains the full URI of the current module.

---

`\l_stex_all_modules_seq` contains the full URIs of all modules currently in scope.

---

`\stex_module_setup:n \stex_module_setup:n {<name>}`

Computes the full URI of a new module with name `<name>` in the current namespace, initializes the four macros above and sets `\l_stex_current_module_str` accordingly. Also takes care of correct naming for nested modules, activates the meta theory and loads the signature module if `sig=` was provided (according to `\l_stex_key_sig_str`).

---

`\stex_close_module:` closes the current module; checks whether we are currently in sms mode, and if so, calls `\stex_persist:n` to write the module contents to the sms-file.

---

`\stex_every_module:n \stex_every_module:n {<code>}`

executes `<code>` every time a new module is opened.

---

`\stex_if_in_module_p: *` tests whether we are currently in a module.  
`\stex_if_in_module:TF *`

---

`\stex_if_module_exists_p:n *`  
`\stex_if_module_exists:nTF *`

tests whether a module with the given full URI exists, in the sense of *has been parsed at some point in the current document*.

---

`\stex_activate_module:n` activates the module with the given full URI *if and only if* it has not already been activated in the current scope.  
`\stex_activate_module:(o|x)`

---

`\stex_execute_in_module:n` executes the provided code, adds it to the current module activation code, and makes sure the macros defined in it are valid in the current module TeX group level.  
`\stex_execute_in_module:x`

This macro is a combination of the following two macros:

---

`\stex_do_up_to_module:n` executes the provided code such that all definitions in it are valid in the current module  
`\stex_do_up_to_module:x` regardless of TeX group level (using `\stex_metagroup_do_in:nn`).

---

`\stex_module_add_code:n` adds the provided code to the module's `\c_stex_module_<uri>_code`-macro.  
`\stex_module_add_code:x`

### 11.3 Symbols

A symbol in **STEX** is represented as a tuple of several components:

---

`\stex_module_add_symbol:nnnnnnnN`

#1 : `<id>`: An *identifier* (possibly empty) that determines its semantic macro name, or e.g. in `mathstructure` its accessor-identifier (if empty, the name is used for that),  
#2 : `<name>` a unique *name*, which in combination with the containing module determines its URI as `<module-URI>?<name>`,  
#3 : `<arity>` a numeric string in the range 0..9,  
#4 : `<args>` a list of argument specifiers of the form `<i><mode>{<argname>}` (the length of `<args>` must be 3·`<arity>`),  
#5 : `<definiens>` (or empty),  
#6 : `<type>` (or empty),  
#7 : `<return code>` (or empty), and  
#8 : `<\invocation_macro>`.

the arguments are stored in the property list `\c_stex_module_<\l_stex_current_module>_symbols_prop` under key `<name>`.

If the identifier `<id>` is non-empty, the macro `\id` is defined as `{\stex_invoke_symbol:<nnnnnnnN>}` with the arguments described there.

**TeXhackers note:** `\invocation_macro` must be `\protected`.

---

`\stex_iterate_symbols:n`  
`\stex_iterate_break:`

`\stex_iterate_symbols:n {<code>}`

`\stex_iterate_break:n`

iterates over all symbols currently in scope and calls `<code>` for all of them with arguments `{<module>}{{<name>}}{<arity>}{<args>}{<definiens>} {<type>} {<return code>} {<\invocation_macro>}`.

Iteration can be stopped prematurely with `\stex_iterate_break:` and can stop and return code with `\stex_iterate_break:n`.

---

`\stex_iterate_symbols:nn`

`\stex_iterate_symbols:nn {<csl>} {<code>}`

iterates over all symbols in the provided `<csl>` of full module URIs. `<code>` receives the same arguments as `\stex_iterate_symbols:n`, but iteration can not be stopped early.

---

```
\stex_get_symbol:n  
\l_stex_get_symbol_mod_str  
\l_stex_get_symbol_name_str  
\l_stex_get_symbol_arity_int  
\l_stex_get_symbol_args_tl  
\l_stex_get_symbol_def_tl  
\l_stex_get_symbol_type_tl  
\l_stex_get_symbol_return_tl  
\l_stex_get_symbol_invoke_cs
```

---

`\stex_get_symbol:n` attempts to find a symbol with the given name or id that is currently in scope. A name may be prefixed with a module name/path separated by `?`.

Throws an error if no such symbol is found; otherwise, sets the listed `\l_stex_get_symbol_...` macros to the components of the found symbol.

---

```
\stex_if_check_terms_p: * whether to typeset declaration components (notations, types, definientia etc.) in a throw-  
\stex_if_check_terms:TF * away box. Is true iff the backend is pdflatex and either the STEX_CHECKTERMS environment variable or checkterms package option is set.
```

---

---

`\stex_check_term:n` typesets the argument in a throwaway box in math mode iff `\stex_if_check_terms:` is true.

Is deactivated in sms-mode.

---

```
\stex_symdecl_do:  
\l_stex_key_name_str  
\l_stex_key_args_str  
\l_stex_key_argnames_clist  
\l_stex_assoc_args_count  
\l_stex_argnames_seq
```

---

`\stex_symdecl_do:` processes the shared (mandatory and optional) arguments of e.g. `\symdecl`, `\symdef`, `\vardef` etc.

Requires the following macros to be set

- `\l_stex_key_name_str`: the name of the symbol,
- `\l_stex_key_args_str`: the `args`-string (e.g. `3` or `ai`)
- `\l_stex_key_argnames_clist`: a list of *names* for the arguments, the length of which should be  $\leq$  the arity of the symbol

and will generate the following macros:

- `\l_stex_get_symbol_arity_int`: the arity of the symbol,
- `\l_stex_key_args_str`: the args string as a definite sequence of argument-mode characters, whose length is the arity of the symbol; e.g. `3` is turned into `iii`,
- `\l_stex_assoc_args_count`: the number of sequence arguments (i.e. `a` or `B` mode),
- `\l_stex_argnames_seq`: the full sequence of argument names; those not provided by `\l_stex_key_argnames_clist` are set to be `$j`, where `j` is the index of the argument,
- `\l_stex_get_symbol_args_tl`: a token list of triples `j{argname}`, where `j` is the index and `m` the respective argument mode character (i.e. `i`, `a`, `b` or `B`).

---

`\_stex_symdecl_check_terms:`

calls `\stex_check_term:n` for the type and definiens stored in `\l_stex_key_type_tl` and `\l_stex_key_def_tl`

---

`\stex_symdecl_top:n \stex_symdecl_top:n {{maybename}}`

checks whether `\l_stex_key_name_str` is empty, and if so, sets it to be `<maybename>`. Then calls `\stex_symdecl_do:` and `\_stex_symdecl_check_terms:`, writes the components to the HTML (if applicable) and adds the symbol to the current module with invocation macro `\stex_invoke_symbol:` and id/macroname `\l_stex_mroname_str`.

Variables work very similar to symbols, except that their declarations are local to the current TeX-group rather than the current module, and they are not exported in modules.

---

`\l_stex_variables_prop` stores all variables currently in scope as a property list with key `<name>` and value `{{id}}{{name}}{{arity}}{{args}}{{definiens}}{{type}}{{return code}}{{\invokation_macro}}`.  
The invocation macro for “normal” variables declared with `\vardef` is `\stex_invoke_symbol:`.

---

`\_stex_vardecl_notation_macro:`

generates the notation macro for a variable, based on the values of the `\l_stex_key-` macros und `\l_stex_notation_macrocode_cs`.

---

`\stex_get_var:n` like `\stex_get_symbol:n`, but attempts to retrieve a variables and throws an error if none is found.

---

`\stex_get_symbol_or_var:n` like `\stex_get_symbol:n`, but first attempts to find a *variable*, and if none is found, defers to `\stex_get_symbol:n`.

## 11.4 Notations

---

`\stex_module_add_notation:nnnnn \stex_module_add_notation:eoexo \stex_module_add_notation:nnnnn`  
 `{{symboluri}}{{id}}{{arity}}{{code}}{{op code}}`

stores the arguments in the property list `\c_stex_module_{\l_stex_current_module}_notations_prop` under key `<symboluri>!<id>` and calls `\stex_set_notation_macro:nnnnn`.

---

```
\stex_set_notation_macro:nnnn \stex_set_notation_macro:nnnn
\stex_set_notation_macro:eoexo {symboluri}{id}{arity}{code}{op code}
```

Declares the following macros:

An active notation for a symbol with uri  $\langle symboluri \rangle$  and id  $\langle id \rangle$  is represented as a macro  $\backslash l\_stex\_notation_{\langle symboluri \rangle}_{\langle id \rangle}\_cs$ , that takes  $\langle arity \rangle$  many argument and expands to  $\langle code \rangle$ , and (if nonempty) an *operator* notation as a macro  $\backslash l\_stex\_notation_{\langle symboluri \rangle}\_op_{\langle id \rangle}\_cs$  that expands to  $\langle op code \rangle$ .

The default notation is represented as the *empty* id. If the correponding macros  $\backslash l\_stex\_notation_{\langle symboluri \rangle}\_cs$ , and  $\backslash l\_stex\_notation_{\langle symboluri \rangle}\_op\_\_cs$  do not yet exist, they are now defined as  $\langle id \rangle$ .

---

```
\stex_iterate_notations:nn \stex_iterate_notations:nn {csl}{code}
```

iterates over all notations in the provided  $\langle csl \rangle$  of full module URIs and calls  $\langle code \rangle$  on each of them with arguments  $\{symboluri\}{id}{arity}{code}{op code}$ .

---

```
\stex_notation_parse_and_then:nw \stex_notation_parse_and_then:nw {code}{notations}
\l_stex_key_prec_str
\l_stex_key_variant_str
\l_stex_notation_macrocode_cs
```

parses a notation (which may consist of multiple braced components, depending on the argument modes) and subsequently calls  $\langle code \rangle$ .

Requires the following macros to be set:

- $\backslash l\_stex\_get\_symbol\_arity\_int$ ,
- $\backslash l\_stex\_get\_symbol\_args\_tl$ ,
- $\backslash l\_stex\_key\_prec\_str$ : The precedence string as provided by a user in the optional *precs*-argument,
- $\backslash l\_stex\_key\_variant\_str$ : the id of the notation variant.

Stores the final notation in the macro  $\backslash l\_stex\_notation\_macrocode\_cs$  taking  $\backslash l\_stex\_get\_symbol\_arity\_int$  many arguments.

Some thing to consider when we generate a notation macro:

- The notation macro generated by the  $\backslash notation$ -command should contain the variant identifier and the precedences, as these are intrinsic parts of the notation.
- It should *not* contain the symbol itself however, so that notations can be copied between symbols.
- Notations as provided by users will largely adhere to the standard L<sup>A</sup>T<sub>E</sub>X category code scheme, and
- notations need to be “persistable” in .deps-files.

We therefore want to augment the simple code provided by a user by various “annotations” that contain the relevant information (such as precedences) and to mark the

argument positions for semantic extractions, but we should adhere to the default L<sup>A</sup>T<sub>E</sub>X category code scheme in doing so.

---

```
\STEXInternalTermMathArgiii
\STEXInternalTermMathAssocArgiiii
\STEXInternalAssocArgMarkerI
\STEXInternalAssocArgMarkerII
\STEXInternalTermMathOMSOrOMViii
\STEXInternalTermMathOMAiii
\STEXInternalTermMathOMBiii
\STEXInternalSymbolAfterInvocationTL
```

---

In OPENMATH/OMDOC, there are (for our purposes) three kinds of expressions that an application of a semantic macro – and hence a notation macro – can represent, each of which corresponds to a macro taking care of the semantic annotations:

- OMS/OMV: a simple symbol (arity 0) (`\STEXInternalTermMathOMSOrOMViii`)
- OMA: an application of a symbol to argument (arity > 0, `\STEXInternalTermMathOMAiii`)
- OMB: a binding application of a symbol that binds/declares one or more variables (argument string contains b or B, `\STEXInternalTermMathOMBiii`).

The arguments are marked with `\STEXInternalTermMathArgiii` (i or b) or `\STEXInternalTermMathAssocArgiiii` (a or B). Finally, the notation is closed with `\STEXInternalSymbolAfterInvocationTL`.

How this works is best explained by example.

### Example 30

Assume we have a symbol and notation:

```
1 \symdecl{someSymbol}[args=iai]
2 \notation{someSymbol}[prec=10;20x30x40,variant=foo]
3 {First: #1; Second: #2; Third: #3; End}
4 {(#1 -- ##1 split ##2 -- #3)}
```

Since the symbol corresponds to an OMA, the whole notation is wrapped in `\STEXInternalTermMathOMAiii`, taking as arguments the variant identifier (foo), the operator precedence (10) and the body of the notation.

The second argument in the notation, being associative, is wrapped in a `\STEXInternalTermMathAssocArgiiii`, taking as arguments the argument number (2), the precedence (30), the T<sub>E</sub>X parameter token (#2) the notation body ((#1 -- ##1 split ##2 -- #3)), and finally the argument mode (a). Additionally, the markers ##1 and ##2 are replaced by `\STEXInternalAssocArgMarkerI` and `\STEXInternalAssocArgMarkerII`, respectively.

Subsequently, the non-sequence parameter tokens are wrapped in `\STEXInternalTermMathArgiii` with arguments mj (where m is the mode und j the index), the precedence (20 or 40 respectively), and the parameter token.

Finally, a `\STEXInternalSymbolAfterInvocationTL` is inserted.

The final expansion of `\l_stex_notation_macrocode_cs` is thus:

```
1 \STEXInternalTermMathOMAiii{foo}{10}{
2   First: \STEXInternalTermMathArgiii{i1}{20}{#1};
3   Second: \STEXInternalTermMathAssocArgiiii{2}{30}{#2}{
4     (\STEXInternalTermMathArgiii{i1}{20}{##1} --
```

```

5      \STEXInternalAssocArgMarkerI split
6      \STEXInternalAssocArgMarkerII --
7      \STEXInternalTermMathArgiii{i3}{40}{##3}
8  }{a};
9  Third: \STEXInternalTermMathArgiii {i3}{40}{#3}; End
10 }\STEXInternalSymbolAfterInvocationTL

```

---

`\stex_notation_top:nw` `\stex_notation_top:nw {<symboluri>}{<code>}`

calls `\stex_notation_parse_and_then:nw{<code>}` and, adds the notation for the symbol with URI `<symboluri>` to the current module and exports it to the HTML (if applicable).

## 11.5 Structural Features

---

`\stex_structural_feature_module:nn` `\stex_structural_feature_module:nn {<name>}{<typeid>}`  
`\stex_structural_feature_module_end:` `\g_stex_last_feature_str`

opens a new module-like structural feature of type `<typeid>` with name `<name>`, which needs to be closed with `\stex_structural_feature_module_end:`.

Its body behaves like a nested module with name `<modulename>/<name>`, the full URI of which is stored in `\g_stex_last_feature_str` for subsequent elaboration.

---

```
\stex_structural_feature_morphism:nnnnn
\stex_structural_feature_morphism_end: \stex_structural_feature_morphism:nnnnn
{\morphisname}{\typeid}{\archive}{\domain}{\annotations}
\l_stex_current_domain_str
\l_stex_feature_name_str
\l_stex_morphism_symbols_prop
\l_stex_morphism_renames_prop
\l_stex_morphism_morphisms_seq
```

---

opens a new morphism-like structural feature of type  $\langle typeid \rangle$  with name  $\langle morphisname \rangle$  and the module  $[\langle archive \rangle] \{ \langle domain \rangle \}$  as domain, which needs to be closed with `\stex_structural_feature_morphism_end:`.

Deactivates `\symdecl`, `\textsymdecl`, `\symdef`, `\notation` and `smodule`, and activates `\assign`, `\assignMorphism` and `\renamedecl`.

Defines the following macros:

- `\l_stex_feature_name_str ={\langle name \rangle}.`
- `\l_stex_current_domain_str` = the full uri of  $\langle domain \rangle$ .
- `\l_stex_morphism_symbols_prop`: This property list is initialized as follows: For every symbol transitively included in  $\langle domain \rangle$  with data  $\langle module \rangle$ ,  $\langle name \rangle$ ,  $\langle id \rangle$ ,  $\langle arity \rangle$ ,  $\langle args \rangle$ ,  $\langle definiens \rangle$ ,  $\langle type \rangle$ , and  $\langle return code \rangle$ , the property list contains an entry with key  $[\langle module \rangle]/[\langle name \rangle]$  and value  $\{\langle id \rangle\}\{\langle arity \rangle\}\{\langle args \rangle\}\{\langle definiens \rangle\}\{\langle type \rangle\}\{\langle return code \rangle\}$ .
- `\l_stex_morphism_renames_prop`: An initially empty property list.
- `\l_stex_morphism_morphisms_seq`: TODO

At `\stex_structural_feature_morphism_end:`, the elaboration is computed from the above data thusly:

For every entry in `\l_stex_morphism_symbols_prop`, a new symbol is created with the values  $\langle arity \rangle$ ,  $\langle args \rangle$ ,  $\langle definiens \rangle$ ,  $\langle type \rangle$  and  $\langle return code \rangle$  from that property list, and either:

- if `\l_stex_morphism_renames_prop` does not contain an entry with key  $\langle module \rangle? \langle name \rangle$ , then the elaborated name is  $\langle morphisname \rangle / \langle name \rangle$  and its  $\langle id \rangle$  is empty (no semantic macro is generated), or
- if `\l_stex_morphism_renames_prop` contains an entry with key  $\langle module \rangle? \langle name \rangle$ , then its value needs to be of the form  $\{\langle id \rangle\} \{\langle name \rangle\}$ , which are used for the elaborated symbol.

All notations of the symbols transitively included in the domain are copied over to their elaborations.

## 11.6 Imports and Morphisms

---

```
\stex_module_add_morphism:nnnn
\stex_module_add_morphism:(nonn|ooox)
```

---

adds a new module morphism (i.e. inheritance, possibly with modification) to the current module

#1 : The name of the morphism (may be empty for e.g. `\importmodule`, but may be named for e.g. `copymodule`),

#2 : the URI of the module being “imported”,

#3 : the “type” of the morphism (e.g. `import` or `copymodule`),

#4 : a list of assignments as pairs  $\{\langle source \rangle\}\{\langle target \rangle\}$  that signify that the symbol with full URI  $\langle source \rangle$  is being mapped or elaborated to the new symbol with name  $\langle target \rangle$  in the current module. May be empty for e.g. `\importmodule`.

The provided arguments are stored in `\c_stex_module_<uri>_morphisms_prop` with key #1 (if non-empty) or [#2] (if #1 is empty) and value {#1}{#2}{#3}{#4}

Import-like statements in STEX are usually given as pairs  $[\langle archive \rangle]\{\langle path \rangle?\langle module \rangle\}$ , which be relative to the current archive and/or file path. For persistence and sms-mode, these pairs first need to be resolved into an *absolute* specification:

---

```
\stex_import_module_uri:nn
\l_stex_import_archive_str
\l_stex_import_name_str
\l_stex_import_uri_str
\l_stex_import_path_str
```

---

`\stex_import_module_uri:nn`  $\{\langle archive \rangle\}\{\langle pathstring \rangle\}$

(where  $\langle archive \rangle$  may be empty) resolves the given arguments into:

- `\l_stex_import_archive_str` the given archive id (in which case `\stex_require_archive:n` is called), or the id of the current archive (if existent and  $\langle archive \rangle$  empty), or empty,
- `\l_stex_import_uri_str` if an archive id is given, or we currently are in an archive, its corresponding namespace; otherwise `{file:}`,
- `\l_stex_import_path_str` the path of the URI relative to the given (or current) archive, or (if not existent) the absolute path of  $\langle pathstring \rangle$  (without a module name) resolved relative to the current file’s parent directory, and
- `\l_stex_import_name_str` the name of the module.

If  $\langle pathstring \rangle$  does not contain the character ?, the whole pathstring is assumed to be the name of the module and the path is empty (or the current file’s parent directory, depending on the above).

---

```
\stex_require_module:nnn
```

---

takes as arguments values `\l_stex_import_archive_str`, `\l_stex_import_path_str` and `\l_stex_import_name_str` as computed by `\stex_import_module_uri:nn` and (optionally loads and) activates the corresponding module (or throws an error if it does not exist).

Most complex morphisms (`copymodule` et al) are implemented as structural features using `\stex_structural_feature_morphism:nnnn`.

---

```
\stex_get_in_morphism:n  
\l_stex_get_symbol_macro_str
```

finds a symbol with the provided name or id in the domain of the current morphism. Sets the same macros as `\stex_get_symbol:n`, and additionally `\l_stex_get_symbol_macro_str` to the  $\langle id \rangle$  of the symbol. Throws an error if no such symbol is found.

## 11.7 Expressions and Semantic Macros

---

```
\_stex_invoke_symbol:nnnnnnN      \l_stex_current_symbol_str  
\l_stex_current_arity_str  
\l_stex_current_args_tl  
\l_stex_current_return_tl  
\l_stex_current_type_tl
```

---

```
\_stex_invoke_symbol:nnnnnnN  
{\module}{\name}{\arity}{\args}{\definiens}  
{\type}  
{\return_code}  
{\invocation_macro}
```

is how a semantic macro is/should be defined. `\_stex_invoke_symbol:nnnnnnN` first checks whether semantic macros are currently allowed, and throws an error if not. Otherwise, it sets the `\comp`-controlled highlighting to `\compemph@uri`, initializes `\STEXInternalSymbolAfterInvocationTL`, defines the following macros for all of its arguments, and subsequently calls the `\invocation_macro`:

- `\l_stex_current_symbol_str ={\<module>}{\<name>}`
- `\l_stex_current_arity_str ={\<arity>}`
- `\l_stex_current_args_tl ={\<args>}`
- `\l_stex_current_type_tl ={\<type>}`
- `\l_stex_current_return_tl ={\<return code>}`

The simplest example for an `\invocation_macro` is `\stex_invoke_symbol`:

---

```
\_stex_invoke_variable:nnnnnnN
```

analogous to `\_stex_invoke_symbol:nnnnnnN`, but for variables; sets the `\comp`-controlled highlighting to `\varemph@uri`.

---

```
\stex_invoke_symbol:
```

branches based on the mode and following characters:

- If math, check next character:
  - ! operator; defer to operator notation
  - else defer to notation and check the value of `\l_stex_current_return_tl ={\<return>}`.
    - \* If  $\langle return \rangle$  is empty, call the notation macro,
    - \* otherwise, call  $\langle return \rangle \{\stex_invoke_symbol!\}$ .
- If text:

---

```
\stex_invoke_sequence_range:  
\stex_invoke_sequence_in:
```

TODO

## 11.8 Optional (Key-Value) Argument Handling

LATEX3 is surprisingly weak when it comes to handling optional (key-val) arguments in such a manner that *only* the freshly set macros are defined, and to modularly build up sets of argument keys. The following macros attempt to fix that:

---

```
\stex_keys_define:nnnn  
\stex_keys_set:nn
```

---

```
\stex_keys_define:nnnn {\langle id\rangle}{\langle setup code\rangle}  
{\langle keyval setup code\rangle}{\langle parents\rangle}
```

Defines a set of keys and their allowed values with identifier **stex**/*id*, that inherits from the sets with identifiers in *parents*.

**\stex\_keys\_set:nn** {\i{id}}{\i{CSL}} first executes *setup code* (e.g. to empty the macros holding the values) and then sets the keys in set *id* with the values provided in *CSL*.

---

**\\_stex\_do\_id:** should be called whenever a macro or environment has a label id, i.e. calls **\stex\_keys\_set:nn**{*id*}{{...}}, after the title has been typeset. Sets a **\label** by calling **\stex\_ref\_new\_doc\_target:n**{*id*}.

### Example 31

If we define a set of keys with:

```
1 \stex_keys_define:nnnn{archive file}{  
2   \str_clear:N \l_stex_key_archive_str  
3   \str_clear:N \l_stex_key_file_str  
4 }{  
5   archive .str_set_x:N = \l_stex_key_archive_str ,  
6   file .str_set_x:N = \l_stex_key_file_str  
7 }{id}
```

then calling **\stex\_keys\_set:nn**{*archive file*}{{*id=foo,file=bar*}} sets **\l\_stex\_key\_file\_str**=*bar*, assures that **\l\_stex\_key\_archive\_str** is empty, and executes the code associated with *id*, i.e. it sets **\l\_stex\_key\_id\_str**=*foo*.

## 11.9 Stylistable Commands and Environments

---

```
\stex_new_stylistable_cmd:nnnn \stex_new_stylistable_cmd:nnnn {<name>} {<arity>} {<code>}
{<default>}
```

Creates a new macro  $\langle name \rangle$  with expansion  $\langle code \rangle$  taking  $\langle arity \rangle$  many arguments, that is customizable in presentation by a user by calling  $\text{\stexstyle}\langle name \rangle$ . On calling  $\text{\stex_style_apply}$ : executes the presentation code provided by a user.

$\langle code \rangle$  should:

- Call  $\text{\stex_keys_set:nn}\{style\} \dots$  (or a keyset inheriting from  $style$ ),
- set macros with prefix  $\backslash this\dots$  that a user might want to use for presentation (e.g.  $\backslash thistitle$ ),
- call  $\text{\stex_style_apply}$ : at some point.

---

```
\stex_new_stylistable_env:nnnnnnn \stex_new_stylistable_env:nnnnnnn {<name>} {<arity>} {<begincode>}
{<endcode>} {<default begin>} {<default end>} {<prefix>}
```

Like  $\text{\stex_new_stylistable_cmd:nnnn}$ , but defines a new environment  $\langle prefix \rangle \langle name \rangle$  stylistable via  $\text{\stexstyle}\langle name \rangle$ . Should call  $\text{\stex_style_apply}$ : twice; once in the  $\langle begincode \rangle$  and once in  $\langle endcode \rangle$ .

---

**\stex\_style\_apply:** Sets  $\backslash thisstyle$  to be the head of the CSL  $\text{\l_stex_key_style_clist}$  and checks whether a style for the current stylistable macro/environment has been defined; if not, executes the code for the default style.

### Example 32

$\text{\importmodule}$  is defined something like the following:

```
1 \stex_new_stylistable_cmd:nnnn{importmodule}{0{} m}{
2 ...
3 \def\thismoduleuri{...}
4 \def\thismodulename{...}
5 \stex_style_apply:
6 ...
7 }{}
```

A user can then customize the output generated by  $\text{\importmodule}$  (by default none) via  $\text{\stexstyleimportmodule}\{...\} \text{\thismodulename}\{...\}$ .

### Example 33

$\text{\smodule}$  does something like

```
1 \stex_new_stylistable_env:nnnnnnn{module}{0{} m} {
2 ...
3 \def\thismoduleuri{...}
4 \def\thismodulename{...}
5 \stex_style_apply:
6 ...
7 }{
8 ...
```

```
9  \stex_style_apply:  
10 ...  
11 }{}{s}
```

which defines the environment name to be `smodule` and generates `\stextylemodule`.

## 11.10 Math Archives

Math archives are represented as L<sup>A</sup>T<sub>E</sub>X3 property lists, the keys/values of which correspond to the entries in the archive's manifest file. The most important fields are

- `id`,
- `narr` the document namespace,
- `ns` the content namespace, and
- `docurl` the document URL base.

---

`\stex_resolve_path_pair:Nnn` `\stex_resolve_path_pair:Nnn {(\target)}{(\archive-id)}{(\pathstring)}`

---

computes the absolute file path of `(pathstring)` relative to the source-folder of `(archive-id)` (if non-empty), or the current archive (if existent) or the parent working directory (otherwise), and stores the result in `\target`.

---

`\l_stex_current_archive_prop`

---

`\l_stex_current_archive_prop` always points to the current math archive or is `\undefined`, if the current file is not part of a math archive.

---

`\c_stex_main_archive_prop` `\c_stex_main_archive_prop` represents the math archive in which the main file resides (if existent).

---

`\stex_require_archive:n` `\stex_require_archive:n {(\id)}`

---

looks for a math archive `(id)` in the MathHub directory, parses its manifest file, creates the corresponding property list `\c_stex_mathhub_{\id}_manifest_prop`, and throws a fatal error if the archive is not found.

If the archive has been found and parsed before, does nothing, so it is cheap and safe to call repeatedly for the same id.

---

`\stex_set_current_archive:n` `\stex_set_current_archive:n {(\id)}`

---

Calls `\stex_require_archive:n{(\id)}` and sets `\l_stex_current_archive_prop`.

---

`\stex_in_archive:nn` `\stex_in_archive:nn {(\opt-id)}{(\code)}`

---

Executes `(code)` in the context of math archive `(opt-id)` (using `\stex_require_archive:n`), i.e. iff `(opt-id)` is non-empty, changes the current archive to the one with id `(opt-id)`, call `(code)` with `(opt-id)` as argument (in #1) and changes it back afterwards.

If `(opt-id)` is empty, `(code)` is called with the id of the *current* math archive as #1, or with #1 empty if there is no current math archive.

## 11.11 SMS-Mode

**STEX** has to extract formal content (i.e. modules and their symbols) from LATEX-files, that may otherwise contain arbitrary code, including macros that may not be defined unless the file is fully processed by TEX. Those modules and symbols also may depend on other modules that have not yet been loaded. The naive way to achieve this, which would be to just suppress output (e.g. by storing it in a box register) and then \input the required file, does not work thanks to TEX's limited *file stack*, which would overflow quickly for modules that have a deeply nested list of dependencies.

To solve those problems, STEX reads dependencies in what we call *sms mode*, which can be summarized thusly:

- In a first pass, we parse the file token by token, ignoring everything other than a select list of macros and environments that introduce dependencies (such as \importmodule and \begin{smodule}[sig=...]). Instead of loading those, we remember them for later.
- After the file has been fully parsed thusly, the dependencies found are loaded, again in sms-mode.
- In a second pass, we parse the file *again* in the same way, but this time execute all macros that are explicitly allowed in sms mode, such as \importmodule, \symdecl, \notation, \symdef, etc.
- all this parsing happens additionally in a \setbox\throwaway\vbox{...}-block to suppress any accidental output.

This means that TEX's input stack never grows by more than +1, but still behaves *as if* the dependencies were loaded recursively, at the detriment of being somewhat slow.

---

\stex\_if\_smsmode\_p: \* tests for whether we are currently in sms-mode.  
\stex\_if\_smsmode:TF \*

---

\stex\_file\_in\_smsmode:nn  
\stex\_file\_in\_smsmode:on    \stex\_file\_in\_smsmode:nn {\<filestring>} {\<setup-code>}  
sets up sms-mode and internal grouping, calls *<setup-code>* and subsequently processes the file *<filestring>* in sms-mode as described above.

### 11.11.1 Second Pass

---

\stex\_sms\_allow:N    \stex\_sms\_allow:N {\<macro>}

registers the \macro-command to be allowed in sms mode.

This only works, if \macro takes no arguments and/or does not touch the subsequent tokens.

For macros taking arguments, we can use

---

**\stex\_sms\_allow\_escape:N**

`\stex_sms_allow_escape:N {<\macro>}`

registers the `\macro`-command to be allowed in sms mode.

If `\macro` is subsequently encountered in sms-mode, parsing is halted and `\macro` can process arguments as desired. It then needs to continue parsing manually though, by calling `\stex_smsmode_do:` as (usually) its last token.

---

**\stex\_sms\_allow\_env:n**

`\stex_sms_allow_env:n {<envname>}`

registers the environment `<envname>` to be allowed in sms mode.

As with `\stex_sms_allow_escape:N`, the `\begin{<envname>}` is escaped, hence the begin-code of the environment needs to call `\stex_smsmode_do:`. Since `\end{<envname>}` never takes arguments, it does not need to be escaped.

---

**\stex\_smsmode\_do:**

continues with sms-mode-style parsing. Does nothing if not in sms-mode, and is therefore safe to be called “just in case”.

### 11.11.2 First Pass

---

**\stex\_sms\_allow\_import:Nn**

---

**\stex\_sms\_allow\_import\_env:nn**

behave like `\stex_sms_allow_escape:N` and `\stex_sms_allow_env:n` respectively, but the macro or environment provided is now allowed in the *first* pass of sms-mode.

This macro should process arguments, add content to `\g_stex_sms_import_code`, and finally call `\stex_smsmode_do:`.

The code provided in the *second* argument is called before the first pass of sms-mode, as to set up functionality for these macros. For example, `\importmodule` provides code that redefines `\importmodule` to store the identified dependency in `\g_stex_sms_import_code` instead of activating it directly.

---

**\g\_stex\_sms\_import\_code**

is built up in the first pass of sms mode and called subsequently; before the second pass.

Code in this token list should load and activate dependencies found in the first pass.

## 11.12 Strings, File Paths, URIs

---

**\stex\_str\_if\_starts\_with\_p:nn \***

`\stex_str_if_starts_with:nn {<first>} {<second>}`

Checks whether the string `<first>` starts with the string `<second>` (i.e. `<second>` is a prefix of `<first>`).

---

**\stex\_str\_if\_ends\_with\_p:nn \***

`\stex_str_if_ends_with:nn {<first>} {<second>}`

Checks whether the string `<first>` ends with the string `<second>` (i.e. `<second>` is a suffix of `<first>`).

### 11.12.1 File Paths

*File paths* are represented as L<sup>A</sup>T<sub>E</sub>X3 sequences. The following methods make sure to

- canonicalize paths, i.e. resolve .. and . segments,
- set all category codes to 12 (other), and
- transform windows file paths containing \ uniformly to /.

---

\stex\_file\_resolve:Nn  
\stex\_file\_resolve:(No|Nx) \stex\_file\_resolve:Nn {(\macro)}{(\string)}

resolves and canonicalizes the file path string *string* and stores the result in \macro.

---

\stex\_file\_set:Nn  
\stex\_file\_set:(No|Nx) \stex\_file\_set:Nn {(\macro)}{(\string)}

represents an already canonicalized file path string as a L<sup>A</sup>T<sub>E</sub>X3 sequence and stores it in \macro.

---

\stex\_if\_file\_absolute\_p:N \*  
\stex\_if\_file\_absolute:NTF \*

\stex\_if\_file\_absolute:N tests whether the given file path (represented as a canonicalized L<sup>A</sup>T<sub>E</sub>X3 sequence) is an absolute file path.

---

\stex\_file\_use:N \* \stex\_file\_use:N expands to a string representation of the given file path.

---

\stex\_if\_file\_starts\_with:NNTF \stex\_if\_file\_starts\_with:NN {(\first)}{(\second)}

tests whether the file path \first is a child of \second. (*Not expandable*)

---

\stex\_file\_split\_off\_ext:NN \stex\_file\_split\_off\_ext:NN {(\target)}{(\source)}

splits off the file extension of \source and stores the resulting file path in \target

---

\stex\_file\_split\_off\_lang:NN \stex\_file\_split\_off\_lang:NN {(\target)}{(\source)}

checks whether the file path \source ends with a language abbreviation (e.g. .en), if so removes it, and stores the result in \target.

The following are primarily used in file hooks, but might occasionally be useful to call manually:

---

\stex\_filestack\_push:n pushes the given file to the file stack, recomputing \g\_stex\_current\_file, the current language, document URI and namespace.

---

\stex\_filestack\_pop: pops the current top entry of the file stack. If the file stack is empty, resets to \c\_stex\_main\_file.

## File Path Constants and Variables

\c\_stex\_pwd\_file store the parent working directory and the absolute path of the main file being processed  
\c\_stex\_main\_file (with guessed file extension .tex).

\c\_stex\_home\_file stores the user's home directory.

\c\_stex\_mathhub\_file stores the user's MathHub directory; its string representation is stored in \mathhub.

\g\_stex\_current\_file always points to the *current* file.

### 11.12.2 URIs

MMT URIs are represented as token lists of the form

`{\_\_stex_path_auth:n{\(authority\)} \_\_stex_path_path:n{\(path\)} \_\_stex_path_module:n{\(modulename\)} \_\_stex_path_name:n{\(declname\)}},`

all of which may be empty. Largely, URIs are used as strings only, but the above representation is used in `\stex_uri_resolve:Nn` to canonicalize URIs when they are computed the first time.

\stex\_map\_uri:Nnnnn `\stex_map_uri:Nnnnn {\(uri\)}{\(authority code\)}{\(path code\)}{\(modulename code\)}{\(declname code\)}`  
executes the provided `\code`s with the components of the `\uri` as arguments.

\stex\_uri\_resolve:Nn behaves analogously to `\stex_file_resolve:Nn`.  
\stex\_uri\_resolve:(No|Nx)

\stex\_uri\_set:Nn behaves analogously to `\stex_file_set:Nn`.  
\stex\_uri\_set:(No|Nx)

\stex\_uri\_use:N \* behaves analogously to `\stex_file_use:N`.

A common usage of URIs is computing the namespace of content elements (modules or documents) from the namespace of a math archive and some relative file path within that archive.

\stex\_uri\_from\_repo\_file:NNNn `\stex_uri_from_repo_file:NNNn {\(target\)}{\(repo_prop\)}{\(filepath\)}{\(ns\_field\)}`

computes the namespace URI from the property list `\repo_prop` of some math archive, the file path `\filepath` and the archive field `\{ns_field\}` (`narr` or `ns`), and stores the result in `\target`.

---

`\stex_uri_from_repo_file_nolang:NNNn`

behaves like `\stex_uri_from_repo_file:NNNn`, but makes sure to split off language abbreviations from the file name (e.g. `.en`).

---

`\stex_uri_from_current_file:Nn`  
`\stex_uri_from_current_file_nolang:Nn`

Special cases for `\stex_uri_from_repo_file[_nolang]:NNNn`, for `\repo_prop=\l_stex_current_archive_prop` and `\filepath=\g_stex_current_file`.

---

`\stex_set_document_uri:` sets the current value of `\l_stex_current_doc_uri` based on the current file and archive.

---

`\stex_set_current_namespace:`

sets the current value of `\l_stex_current_ns_uri` based on the current file and archive.

---

`\stex_uri_add_module:NNn`  
`\stex_uri_add_module:NNo` `\stex_uri_add_module:NNn {\langle target \rangle}{\langle uri \rangle}{\langle name \rangle}`

Checks that URI `\uri` has no module name, adds the provided `\name` and stores the result in `\target`.

### URI Constants and Variables

---

`\l_stex_current_doc_uri` always points to the current document URI.

---

`\l_stex_current_ns_uri` always points to the current content namespace.

## 11.13 Language Handling

---

`\c_stex_languages_prop`  
`\c_stex_language_abrevs_prop`

Property lists converting babel languages to/from their abbreviations; e.g.

- `\prop_item:Nn \c_stex_languages_prop {de}` yields `ngerman`, and
- `\c_stex_language_abrevs_prop {ngerman}` yields `de`.

---

`\l_stex_current_language_str`

always stores the current language.

---

```
\stex_set_language:n  
\stex_set_language:(x|o)
```

```
\stex_set_language:n {⟨abbrev⟩}
```

Sets `\l_stex_current_language_str`, and, if the `babel` package is loaded, calls `\selectlanguage` on the language corresponding to `{⟨abbrev⟩}`.

Note that the package option `lang=` automatically loads the `babel` package.

If `⟨abbrev⟩=tr`, additionally call `\selectlanguage` with the option `shorthands=:!`.

Throws `error/unknownlanguage` if no language with abbreviation `{⟨abbrev⟩}` is known.

---

```
\stex_language_from_file:
```

infers the current language from file ending (e.g. `.en.tex`) and sets it appropriately.

Is called in a file hook, i.e. always switches language when inputting a file `.<lang>.<ext>`.

## 11.14 Inserting Annotations

`STEX` can be used to produce either `HTML` or `PDF`. In `HTML`-mode, multiple macros exist to insert annotations. The same macros are also valid in `PDF` mode but implemented as null operations.

---

```
\stex_suppress_html:n
```

```
\stex_suppress_html:n {⟨code⟩}
```

turns annotations off temporarily in `⟨code⟩` (e.g. as to not generate additional annotations for elaborated declarations, or in sms-mode).

For that to work, code that inserts annotations should use

---

```
\stex_if_do_html_p: *
```

tests whether to generate `HTML` annotations.

```
\stex_if_do_html:TF *
```

---

```
\stex@backend
```

should be set by a backend engine, such that a file `stex-backend-\stex@backend.cfg` exists.

### 11.14.1 Backend macros

Such a backend config file should provide the following:

---

```
\stex_if_html_backend_p: *
```

can be used to determine whether the backend produces `HTML` (e.g. `RuSTEX` or `LATEXML`) or not (e.g. `pdflatex`).

`\ifstexhtml` is set accordingly.

---

```
\stex_annotationnn
```

```
\stex_annotationnn {⟨attr⟩}{⟨value⟩}{⟨code⟩}
```

In `HTML` mode, annotates the output of `⟨code⟩` with the `XML`-attribute `⟨attr⟩="⟨value⟩"`.

In `PDF` mode, just calls `⟨code⟩`.

---

\stex\_annotation\_invisible:n  
\stex\_annotation\_invisible:n

\stex\_annotation\_invisible:n {⟨code⟩}

Should annotate ⟨code⟩ with `shtml:visible="false" style="display:none;"`. In PDF mode, does nothing.

\stex\_annotation\_invisible:nnn combines \stex\_annotation\_invisible:n and \stex\_annotation:nnn.

`stex_annotation_env (env)`    \begin{stex\_annotation\_env}{⟨attr⟩}{⟨value⟩} ⟨code⟩ \end{stex\_annotation\_env}  
should behave like \stex\_annotation:nnn{⟨attr⟩}{⟨value⟩}{⟨code⟩}

---

\stex\_mathml\_intent:nn MathML Intent (TODO)  
\stex\_mathml\_arg:nn

---

## 11.15 Persisting Content from Math Archives in sms-Files

---

\stex\_persist:n  
\stex\_persist:e

\stex\_persist:n {⟨code⟩}

writes ⟨code⟩ to the `\jobname.sms`-file, if `writesms` is active.

**TeXhackers note:** ⟨code⟩ is being read with `expl3` category codes (except for spaces having catcode 10), but not pretokenized; i.e. ⟨code⟩ can safely change the current catcode scheme.

---

\c\_stex\_persist\_mode\_bool  
\c\_stex\_persist\_write\_mode\_bool

whether `usesms` or `writesms` are active.

## 11.16 Utility Methods

---

\stex\_macro\_body:N \*

expands to the *expansion* of the provided macro, including parameter tokens, with the original category codes intact; e.g. if `\def\foo#1{First #1}`, then \stex\_macro\_body:N `\foo` expands to `First #1`.

---

\stex\_macro\_definition:N \*

expands to the token list *defining* the provided macro, including parameters, command attributes (i.e. `\long`, `\protected`), with the original category codes intact; e.g. if `\protected\def\foo#1{First #1}`, then \stex\_macro\_definition:N `\foo` expands to `\protected\def\foo#1{First #1}`.

**TeXhackers note:** Does not work with “higher” parameter tokens, i.e. `##1`, `####1` etc.

---

`\stex_deactivate_macro:Nn` `\stex_reactivate_macro:N`

`\stex_deactivate_macro:Nn {(\macro)} {(msg)}`

Makes `\macro` throw an error message `error/deactivated-macro{(msg)}`, notifying an author that the macro is only allowed in certain environments.

`\stex_reactivate_macro:N` restores the functionality of the macro.

---

`\stex_kpsewhich:Nn`

`\stex_kpsewhich:Nn {(\macro)} {(args)}`

Calls “`kpsewhich args`” and stores the result in `\macro`,

**TeXhackers note:** Does not require `shell-escape`

---

`\stex_get_env:Nn`

`\stex_get_env:Nn {(\macro)} {(envvar)}`

Stores the value of the environment variable `(envvar)` in `\macro`.

---

`\stex_fatal_error:n` `\stex_fatal_error:nn` `\stex_fatal_error:nxx`

Mimic the `\msg_error:-macros`, but make sure that TeX stops processing.

**TeXhackers note:** Calls `\input{non-existent file}`.

---

#### `\stex_ignore_spaces_and_pars:`

As the name suggests, ignores all subsequent spaces and `\pars` until the first non-expandable macro is encountered.

Useful for e.g. ending `\symdecl` and related macros with, so that formatting sources with empty lines does not cause paragraph breaks.

### 11.16.1 Group-like Behaviours

---

`\stex_pseudogroup_with:nn`

`\stex_pseudogroup_with:nn {(\macros)}{(\code)}`

Calls `(code)` and subsequently restores the values of the `(macros)` given.

**TeXhackers note:** Does *not* work recursively!

---

`\stex_pseudogroup:nn`

`\stex_pseudogroup:nn {(\code1)}{(\code2)}`

Expands `(code2)`, and inserts the result after `(code1)`. Works recursively and allows for restoring the values of macros in combination with `\stex_pseudogroup_restore:N`, but *only for macros that take no arguments*:

---

`\stex_pseudogroup_restore:N *` `\stex_pseudogroup_restore:N {(\macro)}`

### Example 34

```
1 \stex_pseudogroup:nn{  
2   something changing \foo  
3   something changing \num  
4 }{  
5   \stex_pseudogroup_restore:N\foo  
6   \int_set:Nn \num {\int_use:N \num}  
7 }
```

restores the values of macro `\foo` and register `\num` after calling the first block.

Commands like `\symdecl` and `\importmodule` that generate (semantic) macros should be local *to the current module*, e.g. `smodule`. For that purpose, we open a new “metagroup” with some identifier (e.g. `\l_stex_current_module_str`) and then execute the relevant code *in the metagroup with that identifier*:

---

```
\stex_metagroup_new:n  
\stex_metagroup_new:o
```

`\stex_metagroup_new:n {⟨id⟩}`

Opens a new metagroup at the current TeX group level with identifier `⟨id⟩`.

---

```
\stex_metagroup_do_in:nn  
\stex_metagroup_do_in:nx
```

`\stex_metagroup_do_in:nn {⟨id⟩}{⟨code⟩}`

Executes `⟨code⟩` and adds its content to `\aftergroup` up until the TeX group level of the metagroup with identifier `⟨id⟩`.

# Chapter 12

## Additional Packages

### 12.1 NotesSlides Documentation

TODO

### 12.2 Problem Documentation

TODO

### 12.3 HWExam Documentation

TODO

### 12.4 Tikzinput Documentation

TODO

**Part IV**  
**Implementation**

# Chapter 13

## The sTeX Implementation

### 13.1 Setting up

Setup code for the document class

```
1  <*cls>
2  %%%%%%%%%%%%%%% stex.dtx %%%%%%%%%%%%%%%
3
4  \RequirePackage{expl3, l3keys2e}
5  \ProvidesExplClass{stex}{2023/10/13}{3.4.0}{sTeX document class}
6  \IfFileExists{stex-expl-compat.sty}{
7      \usepackage{stex-expl-compat}
8  }{}
9
10 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{stex}}
11 \ProcessOptions
12
13 \RequirePackage{stex}
14
15 \LoadClass{article}
16 </cls>

    Setup code for the package

17 <*package>
18 \RequirePackage{expl3, l3keys2e, ltxcmds}
19 \ProvidesExplPackage{stex}{2023/10/13}{3.4.0}{sTeX package}
20 \IfFileExists{stex-expl-compat.sty}{
21     \usepackage{stex-expl-compat}
22 }{}
23 \RequirePackage{stex-logo} % externalized for backwards-compatibility reasons
24 \RequirePackage{standalone}
25
26 \message{^^J*~This~is~sTeX~version~3.4.0~*^^J}

    Package options:

27 \keys_define:nn { stex / package } {
28     debug      .str_set_x:N = \c_stex_debug_clist ,
29     lang       .clist_set:N = \c_stex_languages_clist ,
30     mathhub    .tl_set_x:N = \mathhub ,
```

```

31   usesms     .bool_set:N = \c_stex_persist_mode_bool ,
32   writesms   .bool_set:N = \c_stex_persist_write_mode_bool ,
33   checkterms .bool_set:N = \c_stex_check_terms_bool ,
34   image      .bool_set:N = \c_tikzinput_image_bool,
35   nofrontmatter .bool_set:N = \c_stex_no_frontmatter_bool,
36   unknown     .code:n    = {}
37 }
38 \exp_args:NNo \clist_set:Nn \c_stex_debug_clist \c_stex_debug_clist
39 \ProcessKeysOptions { stex / package }

Error messages:
40 \input{stex-en.ldf}

```

## 13.2 Utilities

41 \cs\_set\_eq:NN \stex\_undefined: \undefined

### 13.2.1 Calling kpsewhich and Environment Variables

42 <@@=stex\_envs>

\stex\_kpsewhich:Nn

```

43 \cs_new_protected:Nn \stex_kpsewhich:Nn {\group_begin:
44   \catcode`\|=12
45   \sys_get_shell:nnN { kpsewhich ~ #2 } { } \l_tmpa_tl
46   \tl_gset_eq:NN \l_tmpa_tl \l_tmpa_tl
47   \group_end:
48   \exp_args:NNo\str_set:Nn #1 {\l_tmpa_tl}
49   \tl_trim_spaces:N #1
50 }

```

(End of definition for \stex\_kpsewhich:Nn. This function is documented on page 142.)

\stex\_get\_env:Nn

```

51 \sys_if_platform_windows:T {
52   \cs_new_protected:Nn \stex_get_env:Nn {\group_begin:
53     \escapechar=-1\catcode`\\=12
54     \exp_args:NN \stex_kpsewhich:Nn #1 {-expand-var~\c_percent_str#2\c_percent_str}
55     \exp_args:NNx\use:nn\group_end:{\str_set:Nn \exp_not:N #1 { #1 }
56     }
57   }
58 }
59 }{
60   \cs_new_protected:Nn \stex_get_env:Nn {
61     \stex_kpsewhich:Nn #1 {-var-value~#2}
62   }
63 }

```

(End of definition for \stex\_get\_env:Nn. This function is documented on page 142.)

### 13.2.2 Logging

```
64  <@=stex_debug>
65  \cs_new_protected:Nn \stex_debug:nn {
66      \exp_args:NNo \clist_if_in:NnTF \c_stex_debug_clist { \tl_to_str:n{all} }{
67          \__stex_debug_:nn{#1}{#2}
68      }{
69          \exp_args:NNo \clist_if_in:NnT \c_stex_debug_clist { \tl_to_str:n{#1} }{
70              \__stex_debug_:nn{#1}{#2}
71          }
72      }
73  }
74
75 \cs_new_protected:Nn \__stex_debug_:nn {
76     \msg_set:nnn{stex}{debug / #1}{
77         \\Debug~#1:~#2\\
78     }
79     \msg_none:nn{stex}{debug / #1}
80 }
```

(End of definition for `\stex_debug:nn`. This function is documented on page 120.)

We check an environment variable for debugging and set things up:

```
81 \stex_get_env:Nn\__stex_debug_env_str{STEX_DEBUG}
82 \str_if_empty:NTF\__stex_debug_env_str {
83     \clist_set_eq:NN \l__stex_debug_cl \c_stex_debug_clist
84 }{
85     \clist_set:No \l__stex_debug_cl {\__stex_debug_env_str}
86 }
87 \clist_clear:N \c_stex_debug_clist
88 \clist_map_inline:Nn \l__stex_debug_cl {
89     \exp_args:NNo \clist_put_right:Nn \c_stex_debug_clist
90     { \tl_to_str:n{#1} }
91 }
92
93 \exp_args:NNo \clist_if_in:NnTF \c_stex_debug_clist {\tl_to_str:n{all}} {
94     \msg_redirect_module:nnn{ stex }{ none }{ warning }
95     \stex_debug:nn{all}{Logging~everything!}
96 }{
97     \clist_map_inline:Nn \c_stex_debug_clist {
98         \msg_redirect_name:nnn{ stex }{ debug / #1 }{ warning }
99         \stex_debug:nn{#1}{Logging~#1}
100    }
101 }
```

### 13.2.3 Files and File Paths

```
102 <@=stex_path>
```

```
\stex_file_set:Nn
\stex_file_set:No
\stex_file_set:Nx
103 \cs_new_protected:Nn \stex_file_set:Nn {
104     \str_if_empty:nTF {#2} { \seq_clear:N #1 }{
105         \exp_args:NNno \seq_set_split:Nnn #1 / { \tl_to_str:n{#2} }
106     }
```

```

107 }
108 \cs_generate_variant:Nn \stex_file_set:Nn {No, Nx}

```

(End of definition for `\stex_file_set:Nn`. This function is documented on page 137.)

```

\stex_file_resolve:Nn
\stex_file_resolve:No
\stex_file_resolve:Nx
109 \sys_if_platform_windows:TF{
110   \cs_new_protected:Npn \__stex_path_win_take:w #1#2#3 \__stex_path_ : {
111     \uppercase{ \str_set:Nn \l__stex_path_str{#1}}
112     \str_set:Nx \l__stex_path_drive {\l__stex_path_str #2}
113     \str_set:Nn \l__stex_path_str{#3}
114   }
115   \cs_new_protected:Nn \stex_file_resolve:Nn {
116     \str_set:Nn \l__stex_path_str {#2}
117     \str_clear:N \l__stex_path_win_drive
118     \exp_args:NNo \str_replace_all:Nnn \l__stex_path_str \c_backslash_str /
119     \exp_args:Nx \str_if_eq:nnT {\str_item:Nn \l__stex_path_str 2} : {
120       \exp_after:wN \__stex_path_win_take:w \l__stex_path_str \__stex_path_ :
121     }
122     \stex_file_set:No #1 \l__stex_path_str
123     \__stex_path_canonicalize:N #1
124     \str_if_empty:NF \l__stex_path_win_drive {
125       \seq_pop_left:NN #1 \l__stex_path_str
126       \seq_put_left:No #1 \l__stex_path_win_drive
127     }
128     %\stex_debug:nn{files}{Set~\tl_to_str:n{#1}~to~\stex_file_use:N #1}
129   }
130 }{
131   \cs_new_protected:Nn \stex_file_resolve:Nn {
132     \str_set:Nn \l__stex_path_str {#2}
133     \stex_file_set:No #1 \l__stex_path_str
134     \__stex_path_canonicalize:N #1
135     %\stex_debug:nn{files}{Set~\tl_to_str:n{#1}~to~\stex_file_use:N #1}
136   }
137 }
138 \cs_generate_variant:Nn \stex_file_resolve:Nn {No, Nx}
139
140 \cs_new_protected:Nn \__stex_path_canonicalize:N {
141   \seq_if_empty:NF #1 {
142     \seq_pop:NN #1 \l__stex_path_str
143     \seq_clear:N \l__stex_path_seq
144     \str_if_empty:NTF \l__stex_path_str {
145       \seq_map_function:NN #1 \__stex_path_dodots:n
146       \seq_put_left:Nn \l__stex_path_seq {}
147   }{
148     \seq_push:No #1 \l__stex_path_str
149     \seq_map_function:NN #1 \__stex_path_dodots:n
150   }
151   \seq_set_eq:NN #1 \l__stex_path_seq
152 }
153 }
154 \cs_new_protected:Nn \__stex_path_dodots:n {
155   \str_if_empty:nF{#1}{
```

```

157     \str_if_eq:nnF {#1} {.} {
158         \str_if_eq:nnTF {#1} {..} {
159             \seq_if_empty:NF \l__stex_path_seq {
160                 \seq_pop_right:NN \l__stex_path_seq \l__stex_path_str
161             }
162         }{
163             \seq_put_right:Nn \l__stex_path_seq {#1}
164         }
165     }
166 }
167 }
```

(End of definition for `\stex_file_resolve:Nn`. This function is documented on page 137.)

```

\stex_if_file_absolute_p:N
\stex_if_file_absolute:NTF
168 \sys_if_platform_windows:TF {
169     \prg_new_conditional:Nnn \stex_if_file_absolute:N {p, T, F, TF} {
170         \seq_if_empty:NTF #1 \prg_return_false: {
171             \tl_set:Nx \l__stex_path_maybein_str {\seq_item:Nn #1 1}
172             \exp_args:Nx \tl_if_empty:nTF \l__stex_path_maybein_str \prg_return_true: {
173                 \exp_args:Nx \str_if_eq:nnTF {\str_item:Nn \l__stex_path_maybein_str 2} :
174                     \prg_return_true: \prg_return_false:
175                 }
176             }
177         }
178     }{
179         \prg_new_conditional:Nnn \stex_if_file_absolute:N {p, T, F, TF} {
180             \seq_if_empty:NTF #1 \prg_return_false: {
181                 \exp_args:Nx \tl_if_empty:nTF {\seq_item:Nn #1 1}
182                     \prg_return_true: \prg_return_false:
183                 }
184             }
185     }
```

(End of definition for `\stex_if_file_absolute:NTF`. This function is documented on page 137.)

```

\stex_file_use:N
186 \cs_new:Nn \stex_file_use:N {
187     \seq_use:Nn #1 /
188 }
```

(End of definition for `\stex_file_use:N`. This function is documented on page 137.)

We determine the PWD

```

\c_stex_pwd_file
\c_stex_main_file
189 \sys_if_platform_windows:TF{
190     \stex_get_env:Nn\l__stex_path_str{CD}
191 }{
192     \stex_get_env:Nn\l__stex_path_str{PWD}
193 }
194 \stex_file_resolve:No \c_stex_pwd_file \l__stex_path_str
195 \seq_set_eq:NN \c_stex_main_file \c_stex_pwd_file
196 \seq_put_right:Nx \c_stex_main_file {\jobname\tl_to_str:n{.tex}}
197
198 \stex_debug:nn {files} {PWD:\stex_file_use:N \c_stex_pwd_file}
```

(End of definition for `\c_stex_pwd_file` and `\c_stex_main_file`. These variables are documented on page 138.)

We determine the HOME directory

```
\c_stex_home_file
199 \sys_if_platform_windows:TF{
200   \stex_get_env:Nn \l__stex_path_str {homedrive\c_percent_str\c_percent_str homepath}
201 }{
202   \stex_get_env:Nn \l__stex_path_str {HOME}
203 }
204 \stex_file_resolve:No \c_stex_home_file \l__stex_path_str
```

(End of definition for `\c_stex_home_file`. This variable is documented on page 138.)

### 13.2.4 Languages

```
205 <@=stex_lang>
```

We store language abbreviations in two (mutually inverse) property lists:

```
\c_stex_languages_prop
\c_stex_language_abbrevs_prop
206 \exp_args:NNe \prop_const_from_keyval:Nn \c_stex_languages_prop {
207   \tl_to_str:n {en} = english ,
208   \tl_to_str:n {de} = ngerman ,
209   \tl_to_str:n {ar} = arabic ,
210   \tl_to_str:n {bg} = bulgarian ,
211   \tl_to_str:n {ru} = russian ,
212   \tl_to_str:n {fi} = finnish ,
213   \tl_to_str:n {ro} = romanian ,
214   \tl_to_str:n {tr} = turkish ,
215   \tl_to_str:n {fr} = french
216 }
217
218 \exp_args:NNe \prop_const_from_keyval:Nn \c_stex_language_abbrevs_prop { \tl_to_str:n {
219   english = en ,
220   ngerman = de ,
221   arabic = ar ,
222   bulgarian = bg ,
223   russian = ru ,
224   finnish = fi ,
225   romanian = ro ,
226   turkish = tr ,
227   french = fr
228 }
229 % todo: chinese simplified (zhs)
230 %       chinese traditional (zht)
```

(End of definition for `\c_stex_languages_prop` and `\c_stex_language_abbrevs_prop`. These variables are documented on page 139.)

```
\l_stex_current_language_str
```

```
231 \str_new:N \l_stex_current_language_str
```

(End of definition for `\l_stex_current_language_str`. This variable is documented on page 139.)

we use the lang-package option to load the corresponding babel languages:

```

\stex_set_language:n
\stex_set_language:x 232 \cs_new_protected:Nn \stex_set_language:n {
\stex_set_language:o 233   \str_set:Nn \l_stex_current_language_str { #1 }
234   \prop_get:NoNTF \c_stex_languages_prop \l_stex_current_language_str \l__stex_lang_lang_tl
235   \cs_if_eq:NNTF\@onlypreamble\@notprerr{
236     \ltx@ifpackageloaded{babel}{
237       \exp_args:No\selectlanguage\l_stex_lang_lang_tl
238     }{}
239   }{
240     \ltx@ifpackageloaded{babel}{}{
241       \str_if_eq:nnTF {#1} {tr} {
242         \RequirePackage[turkish,shorthands=:!]{babel}
243       }{
244         \RequirePackage[\l__stex_lang_lang_tl]{babel}
245       }
246     }
247   }
248 }{
249   \msg_error:nnx{stex}{error/unknownlanguage}{#1}
250 }
251 }
252 \cs_generate_variant:Nn \stex_set_language:n {x,o}

```

(End of definition for `\stex_set_language:n`. This function is documented on page 140.)

`\stex_language_from_file:`

```

253 \cs_new_protected:Nn \stex_language_from_file: {
254   \seq_get_right:NN \g_stex_current_file \l_tmpa_str
255   \seq_set_split:NnV \l_tmpa_seq . \l_tmpa_str
256   \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str \% = ".tex/.dtx/.sty"
257   \exp_args:No \str_if_eq:nnF \l_tmpa_str {tex} {
258     \exp_args:No \str_if_eq:nnF \l_tmpa_str {dtx} {
259       \exp_args:No \str_if_eq:nnF \l_tmpa_str {ltx} {
260         \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq \l_tmpa_str
261       }
262     }
263   }
264   \seq_pop_left:NN \l_tmpa_seq \l_tmpa_str \% <filename>
265   \seq_if_empty:NF \l_tmpa_seq { %remaining element should be [<something>.]language
266     \seq_pop_right:NN \l_tmpa_seq \l__stex_lang_str
267     \str_if_eq:NNF \l__stex_lang_str \l_stex_current_language_str {
268       \exp_args:NNo \prop_if_in:NnT \c_stex_languages_prop \l__stex_lang_str {
269         \stex_set_language:o \l__stex_lang_str
270       }
271     }
272     \stex_debug:nn{lang} {Language~\l_stex_current_language_str~
273       inferred~from~file~name}
274   }
275 }

```

(End of definition for `\stex_language_from_file:`. This function is documented on page 140.)

Loading babel:

```

276 \clist_if_empty:NF \c_stex_languages_clist {
277   \bool_set_false:N \l__stex_lang_turkish_bool

```

```

278 \seq_clear:N \l_tmpa_seq
279 \clist_map_inline:Nn \c_stex_languages_clist {
280   \str_set:Nx \l_tmpa_str {\#1}
281   \str_if_eq:nnT {\#1}{tr} {
282     \bool_set_true:N \l__stex_lang_turkish_bool
283   }
284   \prop_get:NoNTF \c_stex_languages_prop \l_tmpa_str \l_tmpa_str {
285     \seq_put_right:No \l_tmpa_seq \l_tmpa_str
286   }
287   \msg_error:nnx{stex}{error/unknownlanguage}{\l_tmpa_str}
288 }
289 \stex_debug:nn[lang] {Languages:~\seq_use:Nn \l_tmpa_seq {,~} }
290 \bool_if:NTF \l__stex_lang_turkish_bool {
291   \exp_args:NNe \use:nn \RequirePackage
292   {[main=\seq_use:Nn \l_tmpa_seq, ,shorthands=:!]}{babel}
293 }
294 \exp_args:NNe \use:nn \RequirePackage
295 {[main=\seq_use:Nn \l_tmpa_seq, ]}{babel}
296 }
297 }
298 }
```

### 13.2.5 Group-like Behaviours

299 ⟨@=stex\_groups⟩

\stex\_pseudogroup:nn

```

\stex_pseudogroup_restore:N
300 \cs_new_protected:Npn \stex_pseudogroup:nn {
301   \exp_args:Nne \use:nn
302 }
303 \cs_new:Nn \stex_pseudogroup_restore:N {
304   \tl_if_exist:NTF #1 {
305     \tl_set:Nn \exp_not:N #1 { \exp_args:No \exp_not:n #1 }
306   }
307   \cs_undefine:N \exp_not:N #1
308 }
309 }
```

(End of definition for \stex\_pseudogroup:nn and \stex\_pseudogroup\_restore:N. These functions are documented on page 142.)

\stex\_pseudogroup\_with:nn

```

310 \cs_new_protected:Nn \stex_pseudogroup_with:nn {
311   \tl_map_inline:nn{\#1} {
312     \cs_set_eq:cN{\_stex_groups_\tl_to_str:n{\#1}}{\#1}
313   }
314   \#2
315   \tl_map_inline:nn{\#1} {
316     \cs_set_eq:Nc{\_stex_groups_\tl_to_str:n{\#1}}{\#1}
317     \cs_undefine:c{\_stex_groups_\tl_to_str:n{\#1}}
318   }
319 }
```

(End of definition for \stex\_pseudogroup\_with:nn. This function is documented on page 142.)

```
\stex_metagroup_new:n List of all currently existing metagroup identifiers
\stex_metagroup_new:o
 320 \seq_new:N \l__stex_groups_ids_seq
start a new metagroup at the current group level with id #1
 321 \cs_new_protected:Nn \stex_metagroup_new:n {
 322   \str_set:cx{\l__stex_groups_#1_int} {\int_use:N\currentgrouplevel}
 323   \seq_put_right:Nn \l__stex_groups_ids_seq {#1}
 324 }
 325 \cs_generate_variant:Nn \stex_metagroup_new:n {o}
```

(End of definition for `\stex_metagroup_new:n`. This function is documented on page 143.)

```
\stex_metagroup_do_in:nn
\stex_metagroup_do_in:nx
 326 \prg_new_conditional:Nnn \__stex_groups_exists:n {TF} {
 327   \str_if_exist:cTF{\l__stex_groups_#1_int}
 328     \prg_return_true: \prg_return_false:
 329 }
 330
 331 \cs_new_protected:Nn \stex_metagroup_do_in:nn {
 332   \__stex_groups_exists:nTF{#1} {
 333     \__stex_groups_do_in:nn{#1}{#2}
 334   }{
 335     \msg_error:nnn{stex}{error/metagroup/missing}{#1}
 336   }
 337 }
 338 \cs_generate_variant:Nn \stex_metagroup_do_in:nn {nx}
 339
 340 \cs_new_protected:Nn \__stex_groups_do_in:nn {
 341   \exp_args:Nnx\stex_debug:nn{metagroup}{adding-to-\detokenize{#1}:^J\tl_to_str:n{#2}}
 342   \tl_set:Nn\__stex_groups_tmp{#2}
 343   \exp_args:Nx \int_compare:nNnF {\use:c{\l__stex_groups_#1_int}}
 344     = \currentgrouplevel {
 345       \tl_if_exist:cTF{\g__stex_groups_#1_\the\currentgrouplevel_content} {
 346         \exp_args:Nno \tl_gput_right:cn{\g__stex_groups_#1_\the\currentgrouplevel_content}
 347       }{
 348         \exp_args:Nno \tl_gset:cn{\g__stex_groups_#1_\the\currentgrouplevel_content}
 349       }\__stex_groups_tmp
 350       \bool_if_exist:cF {\l__stex_groups_\the\currentgrouplevel_bool} {
 351         \group_insert_after:N \__stex_groups_do:
 352         \bool_set_true:c {\l__stex_groups_\the\currentgrouplevel_bool}
 353       }
 354     }
 355   \__stex_groups_tmp
 356 }
 357
 358 \cs_new_protected:Nn \__stex_groups_do: {
 359   \seq_map_inline:Nn \l__stex_groups_ids_seq {
 360     \tl_if_exist:cT{\g__stex_groups_#1_\int_eval:n{\currentgrouplevel+1}_content} {
 361       \exp_args:NNno \exp_args:Nno \__stex_groups_do_in:nn{##1} {
 362         \csname g__stex_groups_##1_\int_eval:n{\currentgrouplevel+1}_content\endcsname
 363       }
 364       \cs_undefine:c{\g__stex_groups_##1_\int_eval:n{\currentgrouplevel+1}_content}
 365     }
 366     \bool_if_exist:cF {\l__stex_groups_\int_eval:n\currentgrouplevel_bool} {
```

```

367     \group_insert_after:N \__stex_groups_do:
368     \bool_set_true:c {l__stex_groups_\int_eval:n\currentgrouplevel _bool}
369   }
370 }
371 }
```

(End of definition for `\stex_metagroup_do_in:nn`. This function is documented on page [143](#).)

### 13.2.6 HTML Annotations

```
372 <@@=stex_annotation>
```

`\stex_if_do_html:TF` Whether to (locally) produce HTML output

```

373 \bool_new:N \_stex_html_do_output_bool
374 \bool_set_true:N \_stex_html_do_output_bool
375
376 \prg_new_conditional:Nnn \stex_if_do_html: {p,T,F,TF} {
377   \bool_if:nTF \_stex_html_do_output_bool
378   \prg_return_true: \prg_return_false:
379 }
```

(End of definition for `\stex_if_do_html:TF`. This function is documented on page [140](#).)

`\stex_suppress_html:n` Temporarily disable HTML output

```

380 \cs_new_protected:Nn \stex_suppress_html:n {
381   \stex_pseudogroup:nn{
382     \bool_set_false:N \_stex_html_do_output_bool
383     #1
384   }{
385     \stex_if_do_html:T {
386       \bool_set_true:N \_stex_html_do_output_bool
387     }
388   }
389 }
```

(End of definition for `\stex_suppress_html:n`. This function is documented on page [140](#).)

We determine the backend:

```

\stex_if_html_backend:p:
\stex_if_html_backend:TF
  \ifstexhtml
    \stex@backend
  \fi
  \stex_get_env:Nn\__stex_annotation_env_str{STEX_FORCE_PDF}
  \exp_args:No \str_if_eq:nnTF \__stex_annotation_env_str {true} {
    \def\stex@backend{pdflatex}
  }{
    \tl_if_exist:NF\stex@backend{
      \if@rustex
        \def\stex@backend{rustex}
      \else
        \cs_if_exist:NTF\HCode{
          \def\stex@backend{tex4ht}
        }{

```

```

406         \def\stex@backend{pdflatex}
407     }
408     \fi
409   }
410 }
411 \input{stex-backend-\stex@backend.cfg}
412 \newif\ifstexhtml
413 \stex_if_html_backend:TF {
414   \stexhtmltrue
415   \bool_set_true:N \_stex_html_do_output_bool
416 }{
417   \stexhtmlfalse
418   \bool_set_false:N \_stex_html_do_output_bool
419 }
420 }
```

(End of definition for `\stex_if_html_backend:TF`, `\ifstexhtml`, and `\stex@backend`. These functions are documented on page 140.)

```

\_stex_annotation_force_break:n
422 \stex_if_html_backend:TF {
423   \cs_new_protected:Nn \_stex_annotation_force_break:n {
424     \stex_annotation_invisible:n{~}
425     #1
426     \stex_annotation_invisible:n{~}
427   }
428 }{
429   \cs_new_protected:Nn \_stex_annotation_force_break:n { #1 }
430 }
```

(End of definition for `\_stex_annotation_force_break:n`. This function is documented on page ??.)

```

\mmlintent
\mmlarg
431 \stex_if_html_backend:TF {
432   \cs_new_protected:Npn \mmlintent #1 #2 {
433     \stex_annotation_nn:mml:intent={#1}{#2}
434   }
435   \cs_new_protected:Npn \mmlarg #1 #2 {
436     \stex_annotation_nn:mml:arg={#1}{#2}
437   }
438 }{
439   \cs_new_protected:Npn \mmlintent #1 #2 { #2 }
440   \cs_new_protected:Npn \mmlarg #1 #2 { #2 }
441 }
```

(End of definition for `\mmlintent` and `\mmlarg`. These functions are documented on page ??.)

### 13.2.7 Auxiliary Methods

```
442 <@@=stex_aux>
```

```

\stex_deactivate_macro:Nn
\stex_reactivate_macro:N
443 \cs_new_protected:Nn \stex_deactivate_macro:Nn {
444   \tl_set_eq:cN{\tl_to_str:n{#1}~~orig}{#1}
```

```

445   \cs_set_protected:Npn#1{
446     \exp_args:Nnno\msg_error:nnnn{stex}{error/deactivated-macro}{#1}{#2}
447   }
448 }
449 \cs_new_protected:Nn \stex_reactivate_macro:N {
450   \exp_after:wN\let\exp_after:wN#1\csname \detokenize{#1}---orig\endcsname
451 }

```

(End of definition for `\stex_deactivate_macro:N` and `\stex_reactivate_macro:N`. These functions are documented on page 142.)

#### `\stex_ignore_spaces_and_pars:`

```

452 \protected\def\stex_ignore_spaces_and_pars:{%
453   \begingroup\catcode13=10\relax
454   \@ifnextchar\par{%
455     \endgroup\expandafter\stex_ignore_spaces_and_pars:\@gobble
456   }{%
457     \endgroup
458   }
459 }

```

(End of definition for `\stex_ignore_spaces_and_pars:`. This function is documented on page 142.)

#### `\stex_keys_define:nnnn`

```

460 \cs_new_nopar:Nn \stex_keys_define:nnnn {
461   \tl_gset:cn {__stex_aux_keys_#1_pre_tl}{#2}
462   \tl_gset:cn {__stex_aux_keys_#1_def_tl}{#3}
463   \tl_if_empty:nF{#4}{%
464     \clist_map_inline:nn{#4}{%
465       \tl_set_eq:Nc \l__stex_aux_tl {__stex_aux_keys_##1_pre_tl}
466       \tl_gput_left:co{__stex_aux_keys_#1_pre_tl} \l__stex_aux_tl
467       \tl_set_eq:Nc \l__stex_aux_tl {__stex_aux_keys_##1_def_tl}
468       \tl_gput_left:cn{__stex_aux_keys_#1_def_tl} ,
469       \tl_gput_left:co{__stex_aux_keys_#1_def_tl} \l__stex_aux_tl
470     }
471   }
472   \tl_set_eq:Nc \l__stex_aux_tl {__stex_aux_keys_#1_def_tl}
473   \exp_args:Nno \keys_define:nn {stex / #1} {\l__stex_aux_tl}
474 }

```

(End of definition for `\stex_keys_define:nnnn`. This function is documented on page 132.)

#### `\stex_keys_set:nn`

```

475 \cs_new_nopar:Nn \stex_keys_set:nn {
476   \use:c{\__stex_aux_keys_#1_pre_tl}
477   \keys_set:nn {stex / #1} { #2 }
478 }

```

(End of definition for `\stex_keys_set:nn`. This function is documented on page 132.)

Some ubiquitous key sets:

```

479 \stex_keys_define:nnnn{archive file}{%
480   \str_clear:N \l_stex_key_archive_str
481   \str_clear:N \l_stex_key_file_str
482 }{%
483   archive .str_set_x:N = \l_stex_key_archive_str ,

```

```

484     file      .str_set_x:N = \l_stex_key_file_str
485 }{}
486
487 \stex_keys_define:nnnn{id}{
488   \str_clear:N \l_stex_key_id_str
489 }{
490   id .str_set_x:N = \l_stex_key_id_str
491 }{}
492
493 \stex_keys_define:nnnn{title}{
494   \tl_clear:N \l_stex_key_title_tl
495 }{
496   title .tl_set:N = \l_stex_key_title_tl
497 }{}
498
499 \stex_keys_define:nnnn{style}{
500   \clist_clear:N \l_stex_key_style_clist
501 }{
502   style .clist_set:N = \l_stex_key_style_clist
503 }{}
504
505 \stex_keys_define:nnnn{deprecate}{
506   \str_clear:N \l_stex_key_deprecate_str
507 }{
508   deprecate     .str_set_x:N = \l_stex_key_deprecate_str
509 }{}
510
511 \stex_keys_define:nnnn{uses}{}{
512   uses .code:n = {
513     \clist_map_inline:nn{#1}{%
514       \stex_if_starts_with:nTF{##1}[{%
515         \_stex_aux_split_at_bracket:w ##1 \_stex_end:
516       }{%
517         \usemodule{##1}
518       }
519     }
520   }
521 }{}
522
523 \cs_new_protected:Npn \_stex_aux_split_at_bracket:w [ #1 ] #2 \_stex_end: {%
524   \usemodule[#1]{#2}
525 }

\stex_do_deprecation:n
526 \cs_new:Nn \stex_do_deprecation:n {
527   \str_if_empty:NF \l_stex_key_deprecate_str {
528     \msg_warning:nnxx{stex}{warning/deprecated}{#1}{\l_stex_key_deprecate_str}
529   }
530 }

(End of definition for \stex_do_deprecation:n. This function is documented on page 120.)
```

```
\stex_do_id:
531 \cs_new_protected:Nn \stex_do_id: {
```

```

532 \stex_if_smsmode:F {
533   \str_if_empty:NF \l_stex_key_id_str {
534     \exp_args:No \stex_ref_new_doc_target:n \l_stex_key_id_str
535   }
536 }
537 }

```

(End of definition for `\_stex\_do\_id`. This function is documented on page 132.)

```

\stex_new_styable_env:nnnnnnn
\stex_new_styable_cmd:nnnn
\stex_style_apply:

538 \cs_new_protected:Nn \stex_new_styable_cmd:nnnn {
539   \exp_after:wN \newcommand \cs:w stexstyle#1 \cs_end:[2] []{
540     \__stex_aux_patch:nnn{#1}{##1}{##2}
541   }
542   \exp_after:wN \NewDocumentCommand\cs:w #1\cs_end:{#2}{
543     \cs_set:Npn \stex_style_apply: {
544       \__stex_aux_apply_patch:n{#1}
545     }
546     #3
547   }
548   \tl_set:cn {__stex_aux_style_#1:} { #4 }
549 }
550
551 \cs_new_protected:Nn \__stex_aux_apply_patch:n {
552   \clist_if_empty:NTF \l_stex_key_style_clist {
553     \tl_clear:N \thisstyle
554     \use:c{__stex_aux_style_#1}
555   }{
556     \clist_get:NN \l_stex_key_style_clist \thisstyle
557     \tl_if_exist:cTF{\__stex_aux_style_#1_\thisstyle :}){
558       \use:c{\__stex_aux_style_#1_\thisstyle :}
559     }{
560       \use:c{\__stex_aux_style_#1:}
561     }
562   }
563 }
564
565 \cs_new_protected:Nn \__stex_aux_patch:nnn {
566   \str_if_empty:nTF {#2} {
567     \tl_set:cn{\__stex_aux_style_#1:}{#3}
568   }{
569     \tl_set:cn{\__stex_aux_style_#1_#2:}{#3}
570   }
571 }
572
573 \cs_new_protected:Nn \stex_new_styable_env:nnnnnnn {
574   \exp_after:wN \newcommand \cs:w stexstyle#1 \cs_end:[3] []{
575     \__stex_aux_patch:nnnn{#1}{##1}{##2}{##3}
576   }
577   \NewDocumentEnvironment{#7#1}{#2}{
578     \cs_set:Npn \stex_style_apply: {
579       \__stex_aux_apply_patch_begin:n{#1}
580     }
581     #3

```

```

582 }{
583   \cs_set:Npn \stex_style_apply: {
584     \__stex_aux_patch_end:n{#1}
585   }
586   #4
587 }
588 \tl_set:cn {__stex_aux_style_#1_start:} { #5 }
589 \tl_set:cn {__stex_aux_style_#1_end:} { #6 }
590 }
591
592 \cs_new_protected:Nn \__stex_aux_patch:nnnn {
593   \str_if_empty:nTF {#2}{%
594     \tl_set:cn{__stex_aux_style_#1_start:}{#3}
595     \tl_set:cn{__stex_aux_style_#1_end:}{#4}
596   }{
597     \tl_set:cn{__stex_aux_style_#1_#2_start:}{#3}
598     \tl_set:cn{__stex_aux_style_#1_#2_end:}{#4}
599   }
600 }
601
602 \cs_new_protected:Nn \__stex_aux_apply_patch_begin:n {
603   \clist_if_empty:NTF \l_stex_key_style_clist {
604     \tl_clear:N \thisstyle
605     \use:c{__stex_aux_style_#1_start:}
606   }{
607     \clist_get:NN \l_stex_key_style_clist \thisstyle
608     \stex_debug:nn{styling}{dominant-style:~\thisstyle}
609     \tl_if_exist:cTF{__stex_aux_style_#1_\thisstyle_start:}{%
610       \use:c{__stex_aux_style_#1_\thisstyle_start:}
611     }{
612       \use:c{__stex_aux_style_#1_start:}
613     }
614   }
615 }
616
617 \cs_new_protected:Nn \__stex_aux_apply_patch_end:n {
618   \tl_if_empty:NTF \thisstyle {
619     \use:c{__stex_aux_style_#1_end:}
620   }{
621     \tl_if_exist:cTF{__stex_aux_style_#1_\thisstyle_end:}{%
622       \use:c{__stex_aux_style_#1_\thisstyle_end:}
623     }{
624       \use:c{__stex_aux_style_#1_end:}
625     }
626   }
627 }

```

(End of definition for `\stex_new_stylable_env:nnnnnnn`, `\stex_new_stylable_cmd:nnnn`, and `\stex_style_apply:..`. These functions are documented on page 133.)

`\stex_str_if_ends_with_p:nn`

`\stex_str_if_ends_with:nnTF`

```

628 \prg_new_conditional:Nnn \stex_str_if_ends_with:nn {p,T,F,TF} {
629   \exp_args:Ne \str_if_eq:nnTF {
630     \str_range:nnn{#1}{-}\str_count:n{#2}{-1}

```

```

631   }{#2}\prg_return_true: \prg_return_false:
632 }

```

(End of definition for `\stex_if_starts_with:nnTF`. This function is documented on page 136.)

### `\stex_if_starts_with:pnn`

```

\stex_if_starts_with:nnTF
633 \prg_new_conditional:Nnn \stex_if_starts_with:nn {p,T,F,TF} {
634   \exp_args:Ne \str_if_eq:nnTF {
635     \str_range:nnn{#1}{1}{\str_count:n{#2}}
636   }{#2}\prg_return_true: \prg_return_false:
637 }

```

(End of definition for `\stex_if_starts_with:nnTF`. This function is documented on page 136.)

### `\stex_macro_body:N`

```

\stex_macro_body:N
638 \cs_new:Npn \__stex_aux_start:#1\__stex_aux_end: {\exp_not:n{#1}}
639 \cs_new_protected:Nn \__stex_aux_end: {}
640 \cs_new:Nn \stex_macro_body:N {
641   \exp_args:Nne\use:nn{\exp_after:wN \__stex_aux_start: #1} {
642     \__stex_aux_args:e {\cs_parameter_spec:N #1}\__stex_aux_end:
643   }
644 }
645
646 \cs_new:Nn \__stex_aux_args:n {
647   \tl_if_empty:nF{#1} {
648     {##\exp_args:Ne \tl_head:n {\tl_tail:n {#1}}}
649     \__stex_aux_args:e {\exp_args:Ne\tl_tail:n{\tl_tail:n{#1}}}
650   }
651 }
652 \cs_generate_variant:Nn \__stex_aux_args:n {e}

```

(End of definition for `\stex_macro_body:N`. This function is documented on page 141.)

### `\stex_macro_definition:N`

```

\stex_macro_definition:N
653 \cs_new:Nn \stex_macro_definition:N {
654   \__stex_aux_prefix:e {\cs_prefix_spec:N #1}
655   \def\exp_not:N #1
656   \__stex_aux_params:e {\cs_parameter_spec:N #1}
657   {
658     \stex_macro_body:N #1
659   }
660 }
661
662 \cs_new:Nn \__stex_aux_prefix:n {
663   \tl_if_empty:nF{#1} {
664     \str_if_eq:eeTF {
665       \tl_range:nnn{#1}{1}{10}~
666     }{\tl_to_str:n{\protected}}{
667       \protected
668       \__stex_aux_prefix_long:e {
669         \str_range:nnn{#1}{11}{-1}
670       }
671     }{
672       \__stex_aux_prefix_long:n {#1}
673     }

```

```

674     }
675   }
676 \cs_generate_variant:Nn \__stex_aux_prefix:n {e}
677
678 \cs_new:Nn \__stex_aux_prefix_long:n {
679   \tl_if_empty:nF{#1}{
680     \str_if_eq:eeT {
681       \tl_range:nnn{#1}{1}{10}~
682     }{\tl_to_str:n{\long}}{\long}
683   }
684 }
685 \cs_generate_variant:Nn \__stex_aux_prefix_long:n {e}
686
687 \cs_new:Nn \__stex_aux_params:n {
688   \tl_if_empty:nF{#1}{
689     \exp_args:NNe \str_if_eq:VnTF \c_hash_str {\tl_head:n{#1}}{
690       #####
691     }{
692       \tl_head:n{#1}
693     }
694     \__stex_aux_params:e {\tl_tail:n{#1}}
695   }
696 }
697 \cs_generate_variant:Nn \__stex_aux_params:n {e}

```

(End of definition for `\stex_macro_definition:N`. This function is documented on page [141](#).)

### 13.2.8 Persistence

```
698 <@=stex_persist>
```

We check the environment variables:

```

699 \stex_get_env:Nn\__stex_persist_env_str{STEX_USESMS}
700 \str_if_empty:NF\__stex_persist_env_str{
701   \exp_args:No \str_if_eq:nnF \__stex_persist_env_str{false}{
702     \bool_set_true:N \c_stex_persist_mode_bool
703   }
704 }
705 \stex_get_env:Nn\__stex_persist_env_str{STEX_WRITESMS}
706 \str_if_empty:NF\__stex_persist_env_str{
707   \exp_args:No \str_if_eq:nnF \__stex_persist_env_str{false}{
708     \bool_set_true:N \c_stex_persist_write_mode_bool
709   }
710 }
```

```

\stex_persist:n
\stex_persist:e
711 \iow_new:N \c__stex_persist_sms_iow
712
713 \bool_if:NTF \c_stex_persist_write_mode_bool {
714   \stex_if_html_backend:TF{
715     \cs_new:Npn \stex_persist:n #1 {}
716     \cs_new:Npn \stex_persist:e #1 {}
717   }{
718     \cs_new_protected:Nn \stex_persist:n {
719       \iow_now:Nn \c__stex_persist_sms_iow {#1}
```

```

720      }
721      \cs_generate_variant:Nn \stex_persist:n {e}
722  }
723 }{
724   \cs_new:Npn \stex_persist:n #1 {}
725   \cs_new:Npn \stex_persist:e #1 {}
726 }

```

(End of definition for `\stex_persist:n`. This function is documented on page 141.)

Is called at the end of the .sty-file:

```

727 \cs_new_protected:Nn \__stex_persist_load_file:n {
728   \file_if_exist:nT{#1} {
729     \group_begin:
730     \cs_set:Npn \stex_persist:n ##1 {}
731     \cs_set:Npn \stex_persist:e ##1 {}
732     \stex_debug:nn{persist}{restoring~from~sms~file}
733     \catcode`\ =10\relax
734     \cs:w @ @ input \cs_end:#1\relax
735   \group_end:
736 }
737 }
738 }

739 \cs_new_protected:Nn \__stex_persist_write_only: {
740   \iow_open:Nn \c__stex_persist_sms_iow {\jobname.sms}
741   \AtEndDocument{ \iow_close:N \c__stex_persist_sms_iow }
742 }

743 \cs_new_protected:Nn \__stex_persist_read_and_write: {
744   \file_if_exist:nTF{\jobname sms} {
745     \ior_open:Nn \g_tmpa_ior {\jobname sms}
746     \ior_open:Nn \g_tmpa_iow {\jobname sms2}
747     \ior_str_map_inline:Nn \g_tmpa_ior {
748       \iow_now:Nn \g_tmpa_iow {##1}
749     }
750     \iow_close:N \g_tmpa_iow
751     \ior_close:N \g_tmpa_iow
752     \__stex_persist_write_only:
753     \ior_open:Nn \g_tmpa_ior {\jobname sms2}
754     \ior_str_map_inline:Nn \g_tmpa_iow {
755       \iow_now:Nn \c__stex_persist_sms_iow {##1}
756     }
757     \ior_close:N \g_tmpa_iow
758     \__stex_persist_load_file:n{\jobname sms2}
759   } \__stex_persist_write_only:
760 }

761 }

762 }

763 \cs_new_protected:Nn \_stex_persist_read_now: {
764   \bool_if:NTF \c_stex_persist_mode_bool {
765     \bool_if:NTF \c_stex_persist_write_mode_bool {
766       \__stex_persist_read_and_write:
767     }
768   }
769   \__stex_persist_load_file:n{\jobname sms}
770 }

```

```

771     }{
772         \bool_if:NT \c_stex_persist_write_mode_bool \__stex_persist_write_only:
773     }
774 }

```

### 13.2.9 Files, Paths and URIs

```
775 <@=stex_path>
```

`stex_if_file_starts_with:NNTF`

```

776 \prg_new_protected_conditional:Nnn \stex_if_file_starts_with:NN {T,F,TF} {
777     \seq_set_eq:NN \l__stex_path_a_seq #1
778     \seq_set_eq:NN \l__stex_path_b_seq #2
779     \tl_clear:N \l__stex_path_return_tl
780     \bool_while_do:nn{
781         \bool_not_p:n{
782             \bool_lazy_any_p:n{
783                 {\seq_if_empty_p:N \l__stex_path_a_seq}
784                 {\seq_if_empty_p:N \l__stex_path_b_seq}
785                 {\bool_not_p:n{\tl_if_empty_p:N \l__stex_path_return_tl}}
786             }
787         }
788     }{
789         \seq_pop_left:NN \l__stex_path_a_seq \l__stex_path_a_tl
790         \seq_pop_left:NN \l__stex_path_b_seq \l__stex_path_b_tl
791         \str_if_eq:NNF \l__stex_path_a_tl \l__stex_path_b_tl {
792             \tl_set:Nn \l__stex_path_return_tl {\prg_return_false:}
793         }
794     }
795     \tl_if_empty:NTF \l__stex_path_return_tl {
796         \seq_if_empty:NTF \l__stex_path_b_seq \prg_return_true: \prg_return_false:
797     } \l__stex_path_return_tl
798 }

```

(End of definition for `\stex_if_file_starts_with:NNTF`. This function is documented on page 137.)

`\stex_file_split_off_ext:NN`  
`\stex_file_split_off_lang:NN`

```

799 \cs_new_protected:Nn \stex_file_split_off_ext:NN {
800     \seq_set_eq:NN #1 #2
801     \seq_pop_right:NN #1 \l__stex_path_str
802     \seq_set_split:NnV \l__stex_path_seq . \l__stex_path_str
803     \seq_pop_right:NN \l__stex_path_seq \l__stex_path_str
804     \seq_put_right:Nx #1 {\seq_use:Nn \l__stex_path_seq .}
805 }
806 \cs_new_protected:Nn \stex_file_split_off_lang:NN {
807     \seq_set_eq:NN #1 #2
808     \seq_pop_right:NN #1 \l__stex_path_str
809     \seq_set_split:NnV \l__stex_path_seq . \l__stex_path_str
810     \seq_pop_right:NN \l__stex_path_seq \l__stex_path_str
811
812     \seq_pop_right:NN \l__stex_path_seq \l__stex_path_str
813     \exp_args:NNo \prop_if_in:NnF \c_stex_languages_prop \l__stex_path_str {
814         \seq_put_right:No \l__stex_path_seq \l__stex_path_str
815     }

```

```

816     \seq_put_right:Nx #1 {\seq_use:Nn \l_stex_path_seq .}
817 }
818 }
```

(End of definition for `\stex_file_split_off_ext:NN` and `\stex_file_split_off_lang:NN`. These functions are documented on page 137.)

URIs:

### `\stex_map_uri:Nnnnn`

```

819 \cs_set_protected:Nn \__stex_path_auth:n {
820     \msg_error:nnnx{stex}{error/misused-uri}{\tl_to_str:n{#1}}
821 }
822 \cs_set_eq:NN \__stex_path_path:n \__stex_path_auth:n
823 \cs_set_eq:NN \__stex_path_module:n \__stex_path_auth:n
824 \cs_set_eq:NN \__stex_path_name:n \__stex_path_auth:n
825
826 \cs_set_protected:Nn \stex_map_uri:Nnnnn{
827     \stex_pseudogroup_with:nn{\__stex_path_auth:n\__stex_path_path:n\__stex_path_module:n\__st
828         \cs_set:Npn \__stex_path_auth:n ##1 {#2}
829         \cs_set:Npn \__stex_path_path:n ##1 {#3}
830         \cs_set:Npn \__stex_path_module:n ##1 {#4}
831         \cs_set:Npn \__stex_path_name:n ##1 {#5}
832         #1
833     }
834 }
```

(End of definition for `\stex_map_uri:Nnnnn`. This function is documented on page 138.)

### `\stex_uri_set:Nn`

`\stex_uri_set:Nn`

`\stex_uri_set:Nx`

```

835 \str_set:Nx\__stex_path_colonslash{\cColonStr}
836 \cs_new_protected:Nn \__stex_path_uri_set:NnN {
837     \str_if_empty:nTF {#2} {
838         \msg_error:nnxx{stex}{error/invalid-uri}{\tl_to_str:n{#2}}{empty}
839     }{
840         \exp_args:NNNo \exp_args:NNno \seq_set_split:Nnn #1 \__stex_path_colonslash { \tl_to_str
841         \seq_pop_left:NN #1 \l_stex_path_auth_str
842         \seq_if_empty:NTF #1 {
843             \msg_error:nnxx{stex}{error/invalid-uri}{\tl_to_str:n{#2}}{missing-authority}
844         }{
845             \exp_args:NNnx \seq_set_split:Nnn #1 ? {\exp_args:NNo \seq_use:Nn #1 \__stex_path_col
846             \seq_pop_left:NN #1 \l_stex_path_path
847             #3 \l_stex_path_path \l_stex_path_path
848             \seq_if_empty:NTF #1 {
849                 \exp_args:NNo \__stex_path_uri_set:Nnxnn #1 \l_stex_path_auth_str
850                 {\stex_file_use:N \l_stex_path_path} {} {}
851             }{
852                 \seq_pop_left:NN #1 \l_stex_path_mod
853                 \seq_if_empty:NTF #1 {
854                     \exp_args:NNo \__stex_path_uri_set:Nnxon #1 \l_stex_path_auth_str
855                     {\stex_file_use:N \l_stex_path_path} \l_stex_path_mod {}
856                 }{
857                     \seq_pop_left:NN #1 \l_stex_path_name
858                     \seq_if_empty:NTF #1 {
859                         \exp_args:NNo \__stex_path_uri_set:Nnxon #2 \l_stex_path_auth_str
860                         {\stex_file_use:N \l_stex_path_path} \l_stex_path_mod \l_stex_path_name

```

```

861         }{
862             \msg_error:nxxx{stex}{error/invalid-uri}{\tl_to_str:n{#2}}{too-many~?s}
863         }
864     }
865   }
866 }
867 \stex_debug:nn{uris}{Set~\tl_to_str:n{#1}~to~\stex_uri_use:N #1}
868 }
869 }
870 \cs_new_protected:Nn \__stex_path_uri_set:Nnnnn{
871     \tl_set:Nn #1 {
872         \__stex_path_auth:n{ #2 }
873         \__stex_path_path:n{ #3 }
874         \__stex_path_module:n{ #4 }
875         \__stex_path_name:n{ #5 }
876     }
877 }
878 }
879 \cs_generate_variant:Nn\__stex_path_uri_set:Nnnnn {Nnxnn,Nnxon,Nnxoo}
880
881 \cs_new_protected:Nn \stex_uri_set:Nn {
882     \__stex_path_uri_set:NnN #1 {#2} \stex_file_set:Nn
883 }
884 \cs_generate_variant:Nn \stex_uri_set:Nn {No, Nx}

```

(End of definition for `\stex_uri_set:Nn`. This function is documented on page 138.)

**\stex\_uri\_resolve:Nn**  
`\stex_uri_resolve:No`  
`\stex_uri_resolve:Nx`

```

885 \cs_new_protected:Nn \stex_uri_resolve:Nn {
886     \__stex_path_uri_set:NnN #1 {#2} \stex_file_resolve:Nn
887 }
888 \cs_generate_variant:Nn \stex_uri_resolve:Nn {No, Nx}

```

(End of definition for `\stex_uri_resolve:Nn`. This function is documented on page 138.)

**\stex\_uri\_use:N**

```

889 \cs_new:Npn \__stex_path_uri_use:w \__stex_path_auth:n #1 \__stex_path_path:n #2 \__stex_path_
890     #1\c_colon_str/ #2 \tl_if_empty:nF { #3 }{ ? #3
891         \tl_if_empty:nF { #4 }{ ? #4 } }
892 }
893 \cs_new:Nn \stex_uri_use:N {
894     \exp_args:Ne \cs_if_eq:NNTF { \tl_head:N #1 } \__stex_path_auth:n {
895         \exp_after:wN \__stex_path_uri_use:w #1
896     }{
897         \msg_error:nnnn{stex}{error/invalid-uri}{#1}{Not-a-URI}
898     }
899 }

```

(End of definition for `\stex_uri_use:N`. This function is documented on page 138.)

**\stex\_uri\_from\_repo\_file:NNNn**  
`\stex_uri_from_repo_file_nolang:NNNn`

```

900 \cs_new_protected:Npn \stex_uri_from_repo_file_nolang:NNNn {
901     \__stex_path_from_repo_file:NNNNn \stex_file_split_off_lang:NN
902 }
903 \cs_new_protected:Npn \stex_uri_from_repo_file:NNNn {

```

```

904     \__stex_path_from_repo_file:NNNNn \stex_file_split_off_ext:NN
905 }
906
907 \cs_new_protected:Nn \__stex_path_from_repo_file:NNNNn {
908     #1 \l_stex_path_file #4
909     \prop_if_exist:NTF #3 {
910         \str_clear:N \l_stex_path_uri
911         \prop_get:NnNF #3 {#5} \l_stex_path_uri {
912             \prop_get:NnNF #3 {ns} \l_stex_path_uri {
913                 \__stex_path_uri_set:Nnxnn #2 {file}
914                 {\stex_file_use:N \l_stex_path_file} {} {}
915             }
916         }
917         \str_if_empty:NF \l_stex_path_uri {\__stex_path_relativize:N #2}
918     }{
919         \exp_args:NNx \__stex_path_uri_set:Nnxnn #2 {\tl_to_str:n{file}}
920         {\stex_file_use:N \l_stex_path_file} {} {}
921     }
922 }
923
924 \cs_new_protected:Nn \__stex_path_relativize:N {
925     \seq_set_eq:NN \l_stex_path_seq \l_stex_path_file
926     \stex_file_set:Nx \l_stex_path_path {\prop_item:Nn \l_stex_current_archive_prop {path}}
927     \seq_map_inline:Nn \l_stex_path_path { % id
928         \seq_pop_left:NN \l_stex_path_seq \l_stex_path_tl
929     }
930     \seq_pop_left:NN \l_stex_path_seq \l_stex_path_tl % source
931
932     \stex_uri_set:Nx #1 { \l_stex_path_uri / \stex_file_use:N \l_stex_path_seq }
933 }

```

(End of definition for `\stex_uri_from_repo_file:NNNn` and `\stex_uri_from_repo_file_nolang:NNNn`.  
These functions are documented on page [138](#).)

```

\stex_uri_from_current_file:Nn
\stex_uri_from_current_file_nolang:Nn
934 \cs_new_protected:Nn \stex_uri_from_current_file:Nn {
935     \stex_debug:nn{docuri}{Here:~\stex_file_use:N \g_stex_current_file}
936     \stex_uri_from_repo_file:NNNn #1 \l_stex_current_archive_prop
937     \g_stex_current_file {#2}
938     \stex_debug:nn{docuri}{resolved:~\stex_uri_use:N #1}
939 }
940 \cs_new_protected:Nn \stex_uri_from_current_file_nolang:Nn {
941     \stex_uri_from_repo_file_nolang:NNNn #1 \l_stex_current_archive_prop
942     \g_stex_current_file {#2}
943 }

```

(End of definition for `\stex_uri_from_current_file:Nn` and `\stex_uri_from_current_file_nolang:Nn`.  
These functions are documented on page [139](#).)

```

\stex_uri_add_module:NNn
\stex_uri_add_module:NNo
944 \cs_new_protected:Nn \stex_uri_add_module:NNn {
945     \exp_args:Ne \cs_if_eq:NNTF { \tl_head:N #2 } \__stex_path_auth:n {
946         \stex_pseudogroup_with:nn
947             {\__stex_path_auth:n\__stex_path_path:n\__stex_path_module:n\__stex_path_name:n}
948             {

```

```

949      \cs_set:Npn \__stex_path_module:n ##1 {
950          \tl_if_empty:nTF{##1} {
951              \exp_not:N \__stex_path_module:n {#3}
952          }{
953              \msg_error:nnn{stex}{error/invalid-dpath}{#2}
954          }
955      }
956      \cs_set:Npn \__stex_path_name:n ##1 {
957          \tl_if_empty:nTF{##1} {
958              \exp_not:N \__stex_path_name:n {}
959          }{
960              \msg_error:nnn{stex}{error/invalid-dpath}{#2}
961          }
962      }
963      \tl_set:Nx #1 {#2}
964  }
965  }{
966      \msg_error:nnnn{stex}{error/invalid-uri}{#2}{Not~a~URI}
967  }
968 }
969 \cs_generate_variant:Nn \stex_uri_add_module:NNn {NNo}

```

(End of definition for `\stex_uri_add_module:NNn`. This function is documented on page 139.)

### `\l_stex_current_doc_uri`

```
970 \tl_new:N \l_stex_current_doc_uri
```

(End of definition for `\l_stex_current_doc_uri`. This variable is documented on page 139.)

### `\stex_set_document_uri:`

```

971 \cs_new_protected:Nn \stex_set_document_uri: {
972     \stex_uri_from_current_file:Nn \l_stex_current_doc_uri {narr}
973     \%stex_debug:nn{sref}{Document~URI:~\stex_uri_use:N \l_stex_current_doc_uri}
974 }

```

(End of definition for `\stex_set_document_uri:`. This function is documented on page 139.)

### `\stex_set_current_namespace:`

```

975 \cs_new_protected:Nn \stex_set_current_namespace: {
976     \stex_uri_from_current_file_nolang:Nn \l_stex_current_ns_uri {source-base}
977     \%stex_debug:nn{modules}{Namespace~URI:~\stex_uri_use:N \l_stex_current_ns_uri}
978 }

```

(End of definition for `\stex_set_current_namespace:`. This function is documented on page 139.)

## 13.2.10 File Hooks

keeps track of file changes:

```

979 \seq_gclear_new:N \g__stex_path_stack
980 \seq_gclear_new:N \g_stex_current_file

```

```

\stex_filestack_push:n
 981 \cs_new_protected:Nn \stex_filestack_push:n {
 982   \stex_if_ends_with:nnTF {#1}{.tex} {
 983     \stex_file_resolve:No \g_stex_current_file {#1}
 984   }{
 985     \stex_file_resolve:No \g_stex_current_file {#1.tex}
 986   }
 987   \stex_if_file_absolute:NF \g_stex_current_file {
 988     \stex_file_resolve:Nx \g_stex_current_file {
 989       \stex_file_use:N \c_stex_pwd_file / \stex_file_use:N \g_stex_current_file
 990     }
 991   }
 992   \seq_gset_eq:NN \g_stex_current_file \g_stex_current_file
 993   \exp_args:NNx \seq_gpush:Nn \g_stex_path_stack {\stex_file_use:N \g_stex_current_file}
 994   \stex_every_file:
 995 }
 996
 997 \cs_new_protected:Nn \stex_every_file: {
 998   \stex_set_document_uri:
 999   \stex_language_from_file:
1000   \stex_set_current_namespace:
1001 }
1002 %\AtBeginDocument{\stex_every_file:}

(End of definition for \stex_filestack_push:n. This function is documented on page 137.)

```

#### \stex\_filestack\_pop:

```

1003 \cs_new_protected:Nn \stex_filestack_pop: {
1004   \seq_if_empty:NF \g_stex_path_stack {
1005     \seq_gpop>NN \g_stex_path_stack \l_stex_path_str
1006   }
1007   \seq_if_empty:NTF \g_stex_path_stack {
1008     \seq_gset_eq:NN \g_stex_current_file \c_stex_main_file
1009   }{
1010     \seq_get>NN \g_stex_path_stack \l_stex_path_str
1011     \exp_args:NNo \stex_file_set:Nn \g_stex_current_file \l_stex_path_str
1012     \seq_gset_eq:NN \g_stex_current_file \g_stex_current_file
1013   }
1014   \stex_every_file:
1015 }

(End of definition for \stex_filestack_pop:. This function is documented on page 137.)

```

Hooks for the current file:

```

1016 \cs_new_protected:Nn \stex_input_with_hooks:n {
1017   \tl_set:Nn \l_stex_path_do_hooks_pre_tl {
1018     \tl_gset:Nn \l_stex_path_do_hooks_pre_tl {}
1019     \stex_debug:nn{HERE}{Hook-for~#1^J\meaning\CurrentFilePath^J\CurrentFile}
1020     \tl_if_empty:NTF\CurrentFilePath{
1021       \exp_args:No \stex_filestack_push:n {\CurrentFile}
1022     }{
1023       \exp_args:Ne \stex_filestack_push:n { \CurrentFilePath / \CurrentFile }
1024     }
1025   }
1026   \input{#1}

```

```

1027     \stex_debug:nn{HERE}{Hook~end~for~#1}
1028     \stex_filestack_pop:
1029 }
1030 \tl_new:N \l__stex_path_do_hooks_pre_tl {}
1031
1032 \AddToHook{file/before}{
1033     \l__stex_path_do_hooks_pre_tl
1034 }
1035 \%AddToHook{file/after}{ \stex_filestack_pop: }

```

### 13.3 Math Archives

`\mathhub` The path to the mathhub directory. If the `\mathhub`-macro is not set, we query `kpsewhich` for the MATHHUB system variable.

`\c_stex_mathhub_file`

```

1036 <@=stex_mathhub>
1037 \str_if_empty:NTF\mathhub{
1038     \stex_get_env:Nn \l__stex_mathhub_str {MATHHUB}
1039     \str_if_empty:NTF \l__stex_mathhub_str {
1040         \ior_open:NnTF \g_tmpa_ior{\stex_file_use:N \c_stex_home_file/.stex/mathhub.path}{
1041             \group_begin:
1042                 \escapechar=-1\catcode`\\=12
1043                 \ior_str_get:NN \g_tmpa_ior \l__stex_mathhub_str
1044                 \str_gset_eq:NN \l__stex_mathhub_str \l__stex_mathhub_str
1045             \group_end:
1046             \ior_close:N \g_tmpa_ior
1047             \stex_debug:nn{mathhub}{MathHub-directory-determined-from-home-directory}
1048         }{
1049             \str_clear:N \l__stex_mathhub_str
1050         }
1051     }{
1052         \stex_debug:nn{mathhub}{MathHub-directory-determined-from-environment-variable}
1053     }
1054 }{
1055     \str_set_eq:NN \l__stex_mathhub_str \mathhub
1056 }
1057
1058 \seq_new:N \c_stex_mathhub_files
1059
1060 \str_if_empty:NTF \l__stex_mathhub_str {
1061     \msg_warning:nn{stex}{warning/nomathhub}
1062     \exp_args:NNN \stex_file_set:Nn \c_stex_mathhub_file {\stex_file_use:N \c_stex_home_file \c_stex_mathhub_file}
1063     \seq_push:Nx \c_stex_mathhub_files {\stex_file_use:N \c_stex_mathhub_file}
1064 }{
1065     \stex_debug:nn{mathhub}{mathhubs:\l__stex_mathhub_str}
1066     \clist_set:No \l_tmpa_clist \l__stex_mathhub_str
1067     \clist_map_inline:Nn \l_tmpa_clist {
1068         \stex_debug:nn{mathhub}{processing:#1}
1069         \stex_file_resolve:Nn \c_stex_mathhub_file {#1}
1070         \stex_if_file_absolute:NF \c_stex_mathhub_file {
1071             \exp_args:NNN \stex_file_resolve:Nn \c_stex_mathhub_file {
1072                 \stex_file_use:N \c_stex_main_file / ... / #1
1073             }

```

```

1074     }
1075     \stex_debug:nn{mathhub}{yields:~\stex_file_use:N \c_stex_mathhub_file}
1076     \seq_push:Nx \c_stex_mathhub_files {\stex_file_use:N \c_stex_mathhub_file}
1077   }
1078 }
1079
1080 \exp_args:NNe \str_set:Nn \mathhub {\stex_file_use:N \c_stex_mathhub_file}
1081 \stex_debug:nn{mathhub}{MATHHUB~DIRS: \seq_use:Nn \c_stex_mathhub_files {,~}}

```

(End of definition for `\mathhub`, `\c_stex_mathhub_file`, and `\c_stex_mathhub_files`. These variables are documented on page ??.)

### `\l_stex_current_archive`

#### `\stex_set_current_archive:n`

```

1082 \cs_new_protected:Nn \stex_set_current_archive:n {
1083   \stex_require_archive:n { #1 }
1084   \stex_debug:nn{mathhub}{switching-to-archive-#1}
1085   \prop_set_eq:Nc \l_stex_current_archive_prop {
1086     c_stex_mathhub_#1_manifest_prop
1087   }
1088 }

```

(End of definition for `\l_stex_current_archive` and `\stex_set_current_archive:n`. These variables are documented on page ??.)

#### `\stex_in_archive:nn`

```

1089 \cs_new_protected:Nn \stex_in_archive:nn {
1090   \cs_if_exist:NF \l_stex_mathhub_cs {
1091     \cs_set:Npn \l_stex_mathhub_cs ##1 {}
1092   }
1093   \stex_pseudogroup:nn{
1094     \cs_set:Npn \l_stex_mathhub_cs ##1 {#2}
1095     \tl_if_empty:nTF{#1} {
1096       \prop_if_exist:NTF \l_stex_current_archive_prop {
1097         \exp_args:Ne \l_stex_mathhub_cs {\prop_item:Nn \l_stex_current_archive_prop { id }}
1098       }{
1099         \l_stex_mathhub_cs {}
1100       }
1101     }{
1102       \stex_set_current_archive:n{#1}
1103       \l_stex_mathhub_cs {#1}
1104     }
1105   }{
1106     \stex_pseudogroup_restore:N \l_stex_current_archive_prop
1107     \cs_set:Npn \exp_not:N \l_stex_mathhub_cs ##1 {
1108       \exp_args:No \exp_not:n {\l_stex_mathhub_cs {##1}}
1109     }
1110   }
1111 }

```

(End of definition for `\stex_in_archive:nn`. This function is documented on page 134.)

#### `\stex_require_archive:n`

#### `\stex_require_archive:o`

```

1112 \cs_new_protected:Nn \stex_require_archive:n {
1113   \prop_if_exist:cF { c_stex_mathhub_#1_manifest_prop } {
1114     \stex_debug:nn{mathhub}{Opening~archive:~#1}

```

```

1115     \__stex_mathhub_do_manifest:n { #1 }
1116   }
1117 }
1118 \cs_generate_variant:Nn \stex_require_archive:n {o}

```

(End of definition for `\stex_require_archive:n`. This function is documented on page 134.)

Code for finding and parsing manifest files:

```

1119 \str_new:N \l__stex_mathhub_manifest_str
1120 \cs_new_protected:Nn \__stex_mathhub_do_manifest:n {
1121   \str_clear:N \l__stex_mathhub_manifest_str
1122   \seq_map_inline:Nn \c_stex_mathhub_files {
1123     \str_if_empty:NT \l__stex_mathhub_manifest_str {
1124       \__stex_mathhub_find_manifest:n {##1 / #1}
1125     }
1126   }
1127   \str_if_empty:NT \l__stex_mathhub_manifest_str {
1128     \msg_fatal:nneee{stex}{error/noarchive}
1129     {#1}{\seq_use:Nn \c_stex_mathhub_files {,-}}
1130   }
1131   \__stex_mathhub_parse_manifest:n{#1}
1132 }
1133
1134 \cs_new_protected:Nn \__stex_mathhub_find_manifest:n {
1135   \str_set:Nn \l__stex_mathhub_top_str {#1}
1136   \__stex_mathhub_check_manifest:n {#1 / MANIFEST.MF}
1137   \str_if_empty:NT \l__stex_mathhub_manifest_str {
1138     \__stex_mathhub_check_manifest:n {#1 / META-INF / MANIFEST.MF}
1139     \str_if_empty:NT \l__stex_mathhub_manifest_str {
1140       \__stex_mathhub_check_manifest:n {#1 / meta-inf / MANIFEST.MF}
1141     }
1142   }
1143 }
1144
1145 \cs_new_protected:Nn \__stex_mathhub_check_manifest:n {
1146   \stex_debug:nn{mathhub}{Checking ~ #1}
1147   \file_if_exist:nT { #1 } {
1148     \str_set:Nn \l__stex_mathhub_manifest_str { #1 }
1149   }
1150 }
1151
1152 \cs_generate_variant:Nn \__stex_mathhub_find_manifest:n {e}
1153
1154 \ior_new:N \c__stex_mathhub_manifest_ior
1155 \cs_new_protected:Nn \__stex_mathhub_parse_manifest:n {
1156   \ior_open:Nn \c__stex_mathhub_manifest_ior \l__stex_mathhub_manifest_str
1157   \prop_clear:N \l__stex_mathhub_prop
1158   \prop_put:Nno \l__stex_mathhub_prop { path }{ \l__stex_mathhub_top_str }
1159   \ior_map_inline:Nn \c__stex_mathhub_manifest_ior {
1160     \exp_args:NNNo \exp_args:NNNx
1161     \seq_set_split:Nnn \l__stex_mathhub_seq \c_colon_str {\tl_to_str:n{##1}}
1162     \seq_pop_left:NNT \l__stex_mathhub_seq \l__stex_mathhub_key {
1163       \exp_args:NNNo \str_set:Nn \l__stex_mathhub_key \l__stex_mathhub_key
1164       \str_set:Nx \l__stex_mathhub_val {\seq_use:Nn \l__stex_mathhub_seq :}
1165       \str_case:Nn \l__stex_mathhub_key {

```

```

1166     {id}          {\prop_put:Nno \l__stex_mathhub_prop { id }      \l__stex_mathhub_
1167     {narration-base} {\prop_put:Nno \l__stex_mathhub_prop { narr }    \l__stex_mathhub_
1168     {url-base}       {\prop_put:Nno \l__stex_mathhub_prop { docurl }   \l__stex_mathhub_
1169     {source-base}    {\prop_put:Nno \l__stex_mathhub_prop { ns }      \l__stex_mathhub_
1170     {ns}            {\prop_put:Nno \l__stex_mathhub_prop { ns }      \l__stex_mathhub_
1171   }
1172 }
1173 }
1174 \ior_close:N \c__stex_mathhub_manifest_ior
1175 \prop_gset_eq:cN { c_stex_mathhub_#1_manifest_prop } \l__stex_mathhub_prop
1176 \stex_debug:nn{mathhub}{Result:~\prop_to_keyval:N \l__stex_mathhub_prop}
1177 \stex_persist:e {
1178   \prop_gset_from_keyval:cn {c_stex_mathhub_#1_manifest_prop} {
1179     \prop_to_keyval:N \l__stex_mathhub_prop
1180   }
1181 }
1182 }

```

Current MathHub archive

```

\c_stex_main_archive_prop
\l_stex_current_archive_prop
1183 \seq_map_inline:Nn \c_stex_mathhub_files {
1184   \exp_args:Ne \stex_if_starts_with:nnT {\stex_file_use:N \c_stex_pwd_file} {#1} {
1185     \seq_set_eq:NN \l_tmpa_seq \c_stex_pwd_file
1186     \bool_set_true:N \l_tmpa_bool
1187     \bool_while_do:Nn \l_tmpa_bool {
1188       \seq_if_empty:NTF \l_tmpa_seq {
1189         \bool_set_false:N \l_tmpa_bool
1190       }{
1191         \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
1192         \str_if_eq:onT \l_tmpa_str {source} {
1193           \bool_set_false:N \l_tmpa_bool
1194         }
1195       }
1196     }
1197     \seq_if_empty:NF \l_tmpa_seq {
1198       \__stex_mathhub_find_manifest:ef { \stex_file_use:N \l_tmpa_seq }
1199       \str_if_empty:NF \l__stex_mathhub_manifest_str {
1200         \__stex_mathhub_parse_manifest:n { main }
1201         \prop_set_eq:NN \c_stex_main_archive_prop \c_stex_mathhub_main_manifest_prop
1202         \cs_undefine:N \c_stex_mathhub_main_manifest_prop
1203         \prop_get:NnN \c_stex_main_archive_prop {id}
1204           \l_tmpa_str
1205         \prop_set_eq:cN { c_stex_mathhub_\l_tmpa_str _manifest_prop }
1206           \c_stex_main_archive_prop
1207         \exp_args:No \stex_set_current_archive:n { \l_tmpa_str }
1208         \stex_debug:nn{mathhub}{Current~archive:~}
1209           \prop_item:Nn \l_stex_current_archive_prop {id}
1210         }
1211         \bool_if:NT \c_stex_persist_write_mode_bool {
1212           \tl_put_right:Nx \stex_persist_read_now: {
1213             \stex_persist:n {
1214               \prop_gset_from_keyval:cn {c_stex_mathhub_\l_tmpa_str _manifest_prop} {
1215                 \prop_to_keyval:N \c_stex_main_archive_prop
1216               }
1217             }
1218           }
1219         }
1220       }
1221     }
1222   }
1223 }
```

```

1217     \prop_gset_eq:Nc \exp_not:N \l_stex_current_archive_prop {
1218         c_stex_mathhub_\l_tmpa_str _manifest_prop
1219     }
1220     \prop_gset_eq:Nc \exp_not:N \c_stex_main_archive_prop {
1221         c_stex_mathhub_\l_tmpa_str _manifest_prop
1222     }
1223     }
1224   }
1225   }
1226   }
1227   }
1228 }
1229 }

1230 \prop_if_exist:NF \l_stex_current_archive_prop {
1231     \stex_debug:nn{mathhub}{Not~currently~in~the~MathHub~directory}
1232 }
1233 }

(End of definition for \c_stex_main_archive_prop and \l_stex_current_archive_prop. These variables are documented on page 134.)

```

## 13.4 Documents

### 13.4.1 Title

Stores the title, if it exists:

```

1235 <@=stex_doc>
1236 \tl_new:N \g__stex_doc_title_tl

```

\stexdoctitle Initial definition, will be changed at begin document:

```

1237 \cs_new_protected:Npn \stexdoctitle #1 {
1238     \tl_gset:Nn \g__stex_doc_title_tl { #1 }
1239     \global\def\stexdoctitle##1{}
1240 }

```

At begin document, we switch to:

```

1241 \cs_new_protected:Nn \__stex_doc_set_title:n {
1242     \stex_if_smsmode:F{
1243         \global\def\stexdoctitle##1{}
1244         \stex_debug:nn{title}{Setting~title~to:\tl_to_str:n##1}
1245         \tl_gset:Nn \g__stex_doc_title_tl { #1 }
1246         \__stex_doc_title_html:
1247     }
1248 }

```

Hooks, changes and HTML:

```

1249 \cs_new_protected:Nn \__stex_doc_title_html: {
1250     \stex_if_do_html:T{\stex_if_html_backend:T{
1251         \stex_annotate_invisible:nn{shtml:doctitle={}}{ \hbox{\g__stex_doc_title_tl} }
1252     }}
1253 }
1254
1255 \AtBeginDocument {

```

```

1256 \tl_if_empty:NTF \g__stex_doc_title_tl {
1257   \cs_set_eq:NN \stexdoctitle \__stex_doc_set_title:n
1258 }{
1259   \stex_debug:nn{title}{Setting~title~to:\exp_args:No\tl_to_str:n\g__stex_doc_title_tl}
1260   \global\def\stexdoctitle#1{}
1261   \__stex_doc_title_html:
1262 }
1263
1264 \cs_set_eq:NN \__stex_doc_maketitle: \maketitle
1265 \global\protected\def\maketitle{
1266   \tl_if_empty:N \@title {
1267     \exp_args:No \stexdoctitle \@title
1268   }
1269   \__stex_doc_maketitle:
1270 }
1271 }

```

(End of definition for `\stexdoctitle`. This function is documented on page ??.)

### 13.4.2 Sectioning

```

1272 \int_new:N \l_stex_docheader_sect
1273
1274 \tl_set:Nn \stex_current_section_level {document}
1275
1276 \cs_set_protected:Npn \currentsectionlevel {
1277   \stex_if_do_html:TF{
1278     \stex_annotation:nn{shtml:currentsectionlevel={},shtml:capitalize=false}{}}
1279   }{
1280     \stex_current_section_level
1281   }
1282 \tl_if_exist:NT\xspace\xspace
1283 }
1284
1285 \cs_set_protected:Npn \Currentsectionlevel {
1286   \stex_if_do_html:TF{
1287     \stex_annotation:nn{shtml:currentsectionlevel={},shtml:capitalize=true}{}}
1288   }{
1289     \exp_args:No \stex_capitalize:n \stex_current_section_level
1290   }
1291 \tl_if_exist:NT\xspace\xspace
1292 }
1293
1294 \stex_if_html_backend:TF {
1295   \cs_new_protected:Nn \_sfragment_do_level:nn {
1296     \stexdoctitle{#2}
1297     \par
1298     \begin{stex_annotation_env}{shtml:section={\int_use:N \l_stex_docheader_sect}}
1299       \noindent\stex_annotation:nn{shtml:sectiontitle={}{}{
1300         \stex_annotation_force_break:n{#2}
1301       }\par
1302     }
1303     \cs_new_protected:Nn \_sfragment_end: {
1304       \end{stex_annotation_env}

```

```

1305     }
1306   }{
1307     \cs_new_protected:Nn \_sfragment_do_level:nn {
1308       \stextdoctitle{#2}
1309       \tl_if_empty:NTF \l_stex_key_short_tl {
1310         \use:c{#1}
1311       }{
1312         \exp_args:Nnx \use:nn{\use:c{#1}}{\exp_args:No \exp_not:n \l_stex_key_short_tl}
1313       }{#2}
1314       \int_incr:N \l_stex_docheader_sect
1315       \tl_set:Nn \stex_current_section_level{#1}
1316     }
1317     \cs_new_protected:Nn \_sfragment_end: {}
1318   }
1319
1320
1321 \cs_new_protected:Npn \__stex_doc_do_section:n {
1322   \int_case:nnF \l_stex_docheader_sect {
1323     {0}{\cs_if_exist:NTF \thepart {\_sfragment_do_level:nn{part}}{
1324       \int_incr:N \l_stex_docheader_sect
1325       \__stex_doc_do_section:n
1326     }}
1327     {1}{\cs_if_exist:NTF \thechapter {\_sfragment_do_level:nn{chapter}}{
1328       \int_incr:N \l_stex_docheader_sect
1329       \__stex_doc_do_section:n
1330     }}
1331     {2}{\_sfragment_do_level:nn{section}}
1332     {3}{\_sfragment_do_level:nn{subsection}}
1333     {4}{\_sfragment_do_level:nn{subsubsection}}
1334     {5}{\_sfragment_do_level:nn{paragraph}}
1335   }{\_sfragment_do_level:nn{subparagraph}}
1336 }
1337
1338 \stex_keys_define:nnnn{ sfragment }{
1339   \tl_clear:N \l_stex_key_short_tl
1340 }
1341   short .tl_set:N = \l_stex_key_short_tl
1342 }{id}
1343
1344 \NewDocumentEnvironment{sfragment}{ O{} m} {
1345   \stex_keys_set:nn{sfragment}{#1}
1346   \__stex_doc_do_section:n{#2}
1347   \stex_do_id:
1348 }
1349   \_sfragment_end:
1350 }
1351
1352 \%int_incr:N \l_stex_docheader_sect
1353 \NewDocumentEnvironment{blindfragment}{}{
1354   \__stex_doc_skip_section:
1355 }
1356   \stex_if_html_backend:T{
1357     \stex_annotation_invisible:n{~}
1358     \end{stex_annotation_env}

```

```

1359     }
1360 }
1361
1362
1363 \cs_new_protected:Nn \__stex_doc_skip_section_i: {
1364     \int_case:nn \l_stex_docheader_sect {
1365         {0}{\cs_if_exist:NF \thechapter {
1366             \int_incr:N \l_stex_docheader_sect \__stex_doc_skip_section_i:
1367         }}
1368         {1}{\cs_if_exist:NF \thechapter {
1369             \int_incr:N \l_stex_docheader_sect \__stex_doc_skip_section_i:
1370         }}
1371     }
1372     \int_incr:N \l_stex_docheader_sect
1373 }
1374
1375 \stex_if_html_backend:TF {
1376     \cs_new_protected:Nn \__stex_doc_skip_section: {
1377         \__stex_doc_skip_section_i:
1378         \begin{stex_annotation_env}{shtml:skipsection=\int_use:N \l_stex_docheader_sect}
1379         \stex_annotation_invisible:n{~}
1380     }
1381 }
1382 \{
1383     \cs_set_eq:NN \__stex_doc_skip_section: \__stex_doc_skip_section_i:
1384 }
1385
1386 \cs_new_protected:Nn \__stex_doc_skip_fragment:n {
1387     \stepcounter{\#1}
1388 }
1389
1390 \cs_new_protected:Npn \skipfragment {
1391     \int_case:nnF \l_stex_docheader_sect {
1392         {0}{\cs_if_exist:NTF \thechapter {\__stex_doc_skip_fragment:n{part}}{
1393             \int_incr:N \l_stex_docheader_sect
1394             \skipfragment
1395         }}
1396         {1}{\cs_if_exist:NTF \thechapter {\__stex_doc_skip_fragment:n{chapter}}{
1397             \int_incr:N \l_stex_docheader_sect
1398             \skipfragment
1399         }}
1400         {2}{\__stex_doc_skip_fragment:n{section}}
1401         {3}{\__stex_doc_skip_fragment:n{subsection}}
1402         {4}{\__stex_doc_skip_fragment:n{subsubsection}}
1403         {5}{\__stex_doc_skip_fragment:n{paragraph}}
1404     }{\__stex_doc_skip_fragment:n{subparagraph}}
1405 }
1406
\setsectionlevel
1407 \cs_new_protected:Npn \setsectionlevel #1 {
1408     \str_case:nnF{\#1}{
1409         {part}{\int_set:Nn \l_stex_docheader_sect 0}
1410         {chapter}{\int_set:Nn \l_stex_docheader_sect 1}
1411         {section}{\int_set:Nn \l_stex_docheader_sect 2}

```

```

1411 {subsection}{\int_set:Nn \l_stex_docheader_sect 3}
1412 {subsubsection}{\int_set:Nn \l_stex_docheader_sect 4}
1413 {paragraph}{\int_set:Nn \l_stex_docheader_sect 5}
1414 }{
1415   \int_set:Nn \l_stex_docheader_sect 6
1416 }
1417 \cs_if_eq:NNTF\onlypreamble\notprerr{
1418   \stex_annotation_invisible:nn{shml:sectionlevel={\int_use:N\l_stex_docheader_sect}}{}}
1419 }{}
1420 }
1421
1422 \stex_if_html_backend:T{
1423   \cs_new_protected:Nn \__stex_doc_check_topsect: {
1424     \int_case:nnF \l_stex_docheader_sect {
1425       {0}{\cs_if_exist:NTF \thechapter {
1426         \stex_annotation_invisible:nn{shml:sectionlevel=0}{}}
1427       }{
1428         \int_incr:N \l_stex_docheader_sect
1429         \__stex_doc_check_topsect:
1430       }
1431       {1}{\cs_if_exist:NTF \thechapter {
1432         \stex_annotation_invisible:nn{shml:sectionlevel=1}{}}
1433       }{
1434         \int_incr:N \l_stex_docheader_sect
1435         \__stex_doc_check_topsect:
1436       }
1437     }{
1438       \stex_annotation_invisible:nn{shml:sectionlevel={\int_use:N\l_stex_docheader_sect}}{}}
1439     }
1440   }
1441   \AtBeginDocument{\__stex_doc_check_topsect:}
1442 }
1443
1444 \AtBeginDocument{
1445   \bool_if:NF \c_stex_no_frontmatter_bool {
1446     \cs_if_exist:NTF\frontmatter{
1447       \let\__stex_doc_orig_frontmatter\frontmatter
1448       \let\frontmatter\relax
1449     }{
1450       \tl_set:Nn\__stex_doc_orig_frontmatter{
1451         \clearpage
1452         \%@\mainmatterfalse
1453         \pagenumbering{roman}
1454       }
1455     }
1456     \cs_if_exist:NTF\backmatter{
1457       \let\__stex_doc_orig_backmatter\backmatter
1458       \let\backmatter\relax
1459     }{
1460       \tl_set:Nn\__stex_doc_orig_backmatter{
1461         \clearpage
1462         \%@\mainmatterfalse
1463         \pagenumbering{roman}
1464     }

```

```

1465 }
1466 \newenvironment{frontmatter}{
1467   \__stex_doc_orig_frontmatter
1468 }{
1469   \cs_if_exist:N\mainmatter{
1470     \mainmatter
1471   }{
1472     \clearpage
1473     \%@\mainmattertrue
1474     \pagenumbering{arabic}
1475   }
1476 }
1477 \newenvironment{backmatter}{
1478   \__stex_doc_orig_backmatter
1479 }{
1480   \cs_if_exist:N\mainmatter{
1481     \mainmatter
1482   }{
1483     \clearpage
1484     \%@\mainmattertrue
1485     \pagenumbering{arabic}
1486   }
1487 }
1488 }
1489 }

```

(End of definition for `\setsectionlevel`. This function is documented on page 82.)

### 13.4.3 References

```
1490 <@=stex_refs>
```

References are stored in the file `\jobname.sref`, to enable cross-referencing external documents.

```

1491 \iow_new:N \c__stex_refs_iow
1492 \AtBeginDocument{\iow_open:Nn \c__stex_refs_iow {\jobname.sref}}
1493 \AtEndDocument{\iow_close:N \c__stex_refs_iow}

```

The following macros are written to the `.aux`-file, and hence use L<sup>A</sup>T<sub>E</sub>X2e character code scheme:

```

1494 \cs_new_protected:Npn \STEXInternalSrefRestoreTarget #1#2#3#4#5 {}
1495 
1496 \cs_new_protected:Npn \STEXInternalSetSrefSymURL #1 #2 {
1497   \str_gset:cn{g_stex_sref_sym_}\tl_to_str:n{#1}_target}{#2}
1498 }
1499

```

```

\stex_ref_new_doc_target:n
  \sreflabel
    1500 \seq_new:N \g__stex_refs_files_seq
    1501 \int_new:N \l__stex_refs_unnamed_counter_int
    1502 
    1503 \cs_new_protected:Nn \stex_ref_new_id:n {
    1504   \str_if_empty:nTF {#1} {
    1505     \int_gincr:N \l__stex_refs_unnamed_counter_int
    1506     \str_set:Nx \l__stex_refs_str {REF\int_use:N \l__stex_refs_unnamed_counter_int}

```

```

1507    }{
1508      \str_set:Nn \l__stex_refs_str {\#1}
1509    }
1510    \str_set:Nx \l_stex_ref_url_str {\stex_uri_use:N \l_stex_current_doc_uri ? \l__stex_refs_s
1511  }
1512
1513 \cs_new_protected:Nn \stex_ref_new_doc_target:n {
1514   \_stex_ref_new_id:n{\#1}
1515   \%stex_uri_add_module:NNo \l_stex_refs_uri \l_stex_current_doc_uri \l__stex_refs_str
1516   \%stex_debug:nn{sref}{New-document-target:~\stex_uri_use:N \l_stex_refs_uri}
1517   \_stex_refs_add_doc_ref:xo {\stex_uri_use:N \l_stex_current_doc_uri} \l__stex_refs_str
1518   \stex_if_smsmode:F {
1519     \iow_now:Nx \c__stex_refs_iow {
1520       \STEXInternalSrefRestoreTarget
1521       {\stex_uri_use:N \l_stex_current_doc_uri}
1522       {\l__stex_refs_str}
1523       {\@currentcounter}
1524       {\@currentlabel}
1525       {
1526         \tl_if_exist:NT\@currentlabelname{
1527           \exp_args:No\exp_not:n\@currentlabelname
1528         }
1529       }
1530     }
1531     \exp_args:Nx \label {sref@\l_stex_ref_url_str}
1532     \stex_if_do_html:T {
1533       \pdfdest name "sref@\l_stex_ref_url_str" xyz\relax
1534     }
1535   }
1536 }
1537 \NewDocumentCommand \sreflabel {m} {\stex_ref_new_doc_target:n {\#1}}
1538
1539 \cs_new_protected:Nn \_stex_refs_add_doc_ref:nn {
1540   \seq_if_in:NnTF \g__stex_refs_files_seq {\#1} {
1541     \seq_if_in:cnF \g__stex_refs_#1_seq{\#2} {
1542       \seq_gput_left:cn{\g__stex_refs_#1_seq}{\#2}
1543     }
1544   }{
1545     \seq_gput_right:Nn \g__stex_refs_files_seq {\#1}
1546     \seq_new:c{\g__stex_refs_#1_seq}
1547     \seq_gput_left:cn{\g__stex_refs_#1_seq}{\#2}
1548   }
1549 }
1550 \cs_generate_variant:Nn \_stex_refs_add_doc_ref:nn {xo,xx}

```

(End of definition for `\stex_ref_new_doc_target:n` and `\sreflabel`. These functions are documented on page [121](#).)

**\sref** Optional arguments:

```

\extref
1551 \stex_keys_define:nnnn{sref / 1}{}{
1552   % TODO get rid of this
1553   fallback .code:n = {},
1554   pre .code:n = {},
1555   post .code:n = {}

```

```

1556 }{archive file}
1557 \stex_keys_define:nnnn{sref / 2}{}{}{archive file, title}
1558
1559 \str_new:N \l__stex_refs_default_archive_str
1560 \str_new:N \l__stex_refs_default_file_str
1561 \tl_new:N \l__stex_refs_default_title_tl
1562
1563 \cs_set_protected:Nn \__stex_refs_set_keys_b:n {
1564     \tl_if_empty:nTF{#1}{%
1565         \str_set_eq:NN \l_stex_key_archive_str \l__stex_refs_default_archive_str
1566         \str_set_eq:NN \l_stex_key_file_str \l__stex_refs_default_file_str
1567         \tl_set_eq:NN \l_stex_key_title_tl \l__stex_refs_default_title_tl
1568     }{%
1569         \stex_keys_set:nn{ sref / 2 }{ #1 }
1570     }
1571 }
1572
1573 \newcommand\srefsetin[3][]{%
1574     \str_set:Nx \l__stex_refs_default_archive_str {#1}
1575     \str_set:Nx \l__stex_refs_default_file_str {#2}
1576     \tl_set:Nn \l__stex_refs_default_title_tl {#3}
1577 }
1578

```

Auxiliary methods:

```

1579 \cs_new_protected:Nn \__stex_refs_find_uri:n {
1580     \str_clear:N \l__stex_refs_uri_str
1581     \stex_debug:nn{sref}{%
1582         File:~\l_stex_key_file_str^^J
1583         Repo:\l_stex_key_archive_str
1584     }
1585     \str_if_empty:NTF \l_stex_key_file_str {
1586         \stex_debug:nn{sref}{Empty.~Checking~current~file~for~#1}
1587         \seq_if_exist:cT{g__stex_refs_\stex_uri_use:N \l_stex_current_doc_uri _seq}{%
1588             \exp_args:Nnx \__stex_refs_find_uri_in_file:nnn{#1}
1589             {\stex_uri_use:N \l_stex_current_doc_uri}\seq_map_break:
1590         }
1591         \str_if_empty:NT \l__stex_refs_uri_str {
1592             \seq_map_inline:Nn \g__stex_refs_files_seq {
1593                 \__stex_refs_find_uri_in_file:nnn{#1}{##1}{\seq_map_break:n{\seq_map_break:}}
1594             }
1595         }{
1596             \str_if_empty:NTF \l_stex_key_archive_str {
1597                 \prop_if_exist:NTF \l_stex_current_archive_prop {
1598                     \__stex_refs_find_uri_in_prop_file:N \l_stex_current_archive_prop
1599                 }{
1600                     \stex_file_resolve:Nx \l__stex_refs_file
1601                     { \stex_file_use:N \g_stex_current_file / .. / \l_stex_key_file_str }
1602                     \str_set:Nx \l__stex_refs_uri_str { file:/ \stex_file_use:N \l__stex_refs_file }
1603                 }
1604             }{
1605                 \stex_require_archive:o \l_stex_key_archive_str
1606                 \prop_set_eq:Nc \l_stex_refs_prop { c_stex_mathhub_\l_stex_key_archive_str _manifest_ }
1607             }
1608         }
1609     }
1610 }
1611

```

```

1608      \__stex_refs_find_uri_in_prop_file:N \l__stex_refs_prop
1609    }
1610  }
1611 }
1612
1613 \cs_new_protected:Nn \__stex_refs_find_uri_in_prop_file:N {
1614   \str_set:Nx \l__stex_refs_uri_str {
1615     \prop_item:Nn #1 {path} /
1616     source / \l_stex_key_file_str .sref
1617   }
1618   \stex_file_resolve:No \l__stex_refs_file \l__stex_refs_uri_str
1619   \stex_uri_from_repo_file>NNNn \l__stex_refs_uri #1
1620   \l__stex_refs_file {narr}
1621   \str_set:Nx \l__stex_refs_uri_str {\stex_uri_use:N \l__stex_refs_uri}
1622 }
1623
1624 \cs_new_protected:Nn \__stex_refs_find_uri_in_file:nnn {
1625   \stex_debug:nn{sref}{Checking~file~#2}
1626   \seq_map_inline:cn{g__stex_refs_#2_seq}{
1627     \str_if_eq:nnT{#1}{##1}{
1628       \str_set:Nx \l__stex_refs_uri_str {\stex_uri_use:N \l_stex_current_doc_uri}
1629       \stex_debug:nn{sref}{Found.}
1630       #3
1631     }
1632   }
1633 }

```

Doing the actual referencing:

```

1634 \cs_new_protected:Nn \__stex_refs_do_autoref:n {
1635   \cs_if_exist:cTF{autoref}{
1636     \exp_args:Nx\autoref{sref@#1}
1637   }{
1638     \exp_args:Nx\ref{sref@#1}
1639   }
1640 }
1641
1642 \cs_new_protected:Nn \__stex_refs_do_sref:nn {
1643   \str_if_empty:NTF \l__stex_refs_uri_str {
1644     \str_if_empty:NTF \l_stex_key_file_str {
1645       \stex_debug:nn{sref}{autoref~on~\stex_uri_use:N \l_stex_current_doc_uri?#1}
1646       \exp_args:Ne \__stex_refs_do_autoref:n{\stex_uri_use:N \l_stex_current_doc_uri ? #1}
1647     }{
1648       \stex_debug:nn{sref}{srefin~on~#1}
1649       \__stex_refs_set_keys_b:n{ #2 }
1650       \__stex_refs_do_sref_in:n{#1}
1651     }
1652   }{
1653     \exp_args:NNo \seq_if_in:NnTF \g__stex_refs_files_seq \l__stex_refs_uri_str {
1654       \stex_debug:nn{sref}{Using~ref~file~\l__stex_refs_uri_str}
1655       \exp_args:Nnx \seq_if_in:cnTF{g_stex_ref}\l__stex_refs_uri_str _seq}{\detokenize{#1}}{
1656         \stex_debug:nn{sref}{Reference~found~in~ref~files;~autoref~on~\l__stex_refs_uri_str}
1657         \__stex_refs_do_autoref:n{\l__stex_refs_uri_str?#1}
1658       }{
1659         \str_if_empty:NTF \l_stex_key_file_str {

```

```

1660     \stex_debug:nn{sref}{in~empty;~autoref~on~\l_stex_refs_uri_str?#1}
1661     \__stex_refs_do_autoref:n{\l_stex_refs_uri_str?#1}
1662 }
1663     \stex_debug:nn{sref}{in~non~empty;~srefin~on~\l_stex_refs_uri_str?#1}
1664     \__stex_refs_set_keys_b:n{ #2 }
1665     \__stex_refs_do_sref_in:n{#1}
1666 }
1667 }
1668 }
1669     \stex_debug:nn{sref}{No~ref~file~found~for~\l_stex_refs_uri_str}
1670     \str_if_empty:NTF \l_stex_key_file_str {
1671         \stex_debug:nn{sref}{in~empty;~autoref~on~\l_stex_refs_uri_str?#1}
1672         \__stex_refs_do_autoref:n{\l_stex_refs_uri_str?#1}
1673 }
1674     \stex_debug:nn{sref}{in~non~empty;~srefin~on~\l_stex_refs_uri_str?#1}
1675     \__stex_refs_set_keys_b:n{ #2 }
1676     \__stex_refs_do_sref_in:n{#1}
1677 }
1678 }
1679 }
1680 }
1681 \cs_new_protected:Nn \__stex_refs_do_sref_in:n {
1682     \stex_debug:nn{sref}{In: \l_stex_key_file_str^~JRepo:\l_stex_key_archive_str}
1683     \stex_debug:nn{sref}{URI: \l_stex_refs_uri_str?#1}
1684     \tl_if_exist:cTF{r@sref@\l_stex_refs_uri_str?#1}{
1685         \__stex_refs_do_autoref:n{\l_stex_refs_uri_str?#1}
1686 }
1687 }
1688 \%msg_warning:nnn{stex}{warning/smsmissing}{<filename>}
1689 \group_begin:\catcode13=9\relax\catcode10=9\relax
1690     \str_if_empty:NTF \l_stex_key_archive_str {
1691         \prop_if_exist:NTF \l_stex_current_archive_prop {
1692             \str_set:Nx \l_stex_refs_file_str {
1693                 \prop_item:Nn \l_stex_current_archive_prop { path }
1694                 / source / \l_stex_key_file_str .sref
1695             }
1696         }
1697         \str_set:Nx \l_stex_refs_file_str {
1698             \stex_file_use:N \g_stex_current_file / .. / \l_stex_key_file_str .sref
1699         }
1700     }
1701 }
1702     \stex_require_archive:o \l_stex_key_archive_str
1703     \str_set:Nx \l_stex_refs_file_str {
1704         \prop_item:cn {c_stex_mathhub_ \l_stex_key_archive_str _manifest_prop} {path}
1705         / source / \l_stex_key_file_str .sref
1706     }
1707 }
1708 \stex_file_resolve:No \l_stex_refs_file \l_stex_refs_file_str
1709 \str_set:Nx \l_stex_refs_file_str {\stex_file_use:N \l_stex_refs_file }
1710 \stex_debug:nn{sref}{File: \l_stex_refs_file_str }
1711 \exp_args:NNNx \exp_args:No \str_if_eq:nnTF \l_stex_refs_file_str {\stex_file_use:N\c
1712     \__stex_refs_do_autoref:n{
1713         \str_if_empty:NF\l_stex_refs_uri_str{\l_stex_refs_uri_str?}#1

```

```

1714     }
1715   }{
1716     \exp_args:No \IfFileExists \l__stex_refs_file_str {
1717       \tl_clear:N \l__stex_refs_return_tl
1718       \str_set:Nn \l__stex_refs_id_str {\#1}
1719       \let\STEXInternalSrefRestoreTarget\__stex_refs_restore_target:nnnn
1720       \use:c{@ @ input}{\l__stex_refs_file_str}
1721       \exp_args:No \tl_if_empty:nTF \l__stex_refs_return_tl {
1722         \exp_args:Nnno \msg_warning:nnn{stex}{warning/smslabelmissing}\l__stex_refs_file_
1723         \__stex_refs_do_autoref:n{
1724           \str_if_empty:NF\l__stex_refs_uri_str{\l__stex_refs_uri_str?}\#1
1725         }
1726       }{
1727         \l__stex_refs_return_tl
1728       }
1729     }{
1730       \exp_args:Nnno \msg_warning:nnn{stex}{warning/smsmissing}\l__stex_refs_file_str
1731       \__stex_refs_do_autoref:n{
1732         \str_if_empty:NF\l__stex_refs_uri_str{\l__stex_refs_uri_str?}\#1
1733       }
1734     }
1735   }
1736   \group_end:
1737 }
1738 }
1739
1740 \cs_new_protected:Nn \__stex_refs_do_return:nnnn {
1741   \tl_set:Nn \l__stex_refs_return_tl {
1742     \stex_annotation:nn{shtml:sref={\#4},shtml:srefin={\l__stex_refs_file_str}}{
1743       \use:c{\#3autorefname}~\#1\tl_if_empty:nF{\#2}{~(\#2)}
1744       \tl_if_empty:NF\l_stex_key_title_tl{
1745         {}~in~\l_stex_key_title_tl
1746       }
1747     }
1748   }
1749 }
1750
1751 \cs_new_protected:Nn \__stex_refs_restore_target:nnnn {
1752   \str_if_empty:NTF \l__stex_refs_uri_str {
1753     \exp_args:No \str_if_eq:nnT \l__stex_refs_id_str {\#2}){
1754       \__stex_refs_do_return:nnnn{\#4}{\#5}{\#3}{\#1?\#2}
1755     }
1756   }{
1757     \stex_debug:nn{sref}{\l__stex_refs_uri_str{}~ == ~ \#1 ~ ?}
1758     \exp_args:No \str_if_eq:nnT \l__stex_refs_uri_str {\#1}){
1759       \stex_debug:nn{sref}{\l__stex_refs_id_str{}~ == ~ \#2 ~ ?}
1760       \exp_args:No \str_if_eq:nnT \l__stex_refs_id_str {\#2}){
1761         \stex_debug:nn{sref}{success!}
1762         \__stex_refs_do_return:nnnn{\#4}{\#5}{\#3}{\#1?\#2}
1763         \endinput
1764       }
1765     }
1766   }
1767 }

```

The actual macros:

```

1768 \NewDocumentCommand \sref { O{} m O{} }{
1769   \stex_keys_set:nn { sref / 1 }{ #1 }
1770   \__stex_refs_find_uri:n { #2 }
1771   \__stex_refs_do_sref:nn{#2}{#3}
1772 }
1773 \NewDocumentCommand \extref { O{} m m }{
1774   \stex_keys_set:nn { sref / 1 }{ #1 }
1775   \__stex_refs_find_uri:n { #2 }
1776   \__stex_refs_set_keys_b:n{ #3 }
1777   \str_if_empty:NT \l_stex_key_file_str {
1778     \msg_error:nn{stex}{error/extrefmissing}
1779   }
1780   \__stex_refs_do_sref_in:n{#2}
1781 }
```

(End of definition for `\sref` and `\extref`. These functions are documented on page 84.)

```

\stex_ref_new_sym_target:n
1782 \cs_new_protected:Nn \stex_ref_new_symbol:n {
1783   \cs_if_exist:cF{r@sref@sym@\tl_to_str:n{#1}}{
1784     \__stex_refs_new_symbol:n{#1}
1785   }
1786 }
1787
1788 \cs_new_protected:Nn \_stex_sref_do_aux:n {
1789   #1 \iow_now:Nn \Oauxout {#1}
1790 }
1791
1792 \cs_new_protected:Nn \__stex_refs_new_symbol:n {
1793   \prop_if_exist:NTF \l_stex_current_archive_prop {
1794     \prop_get:NnNTF \l_stex_current_archive_prop {docurl} \l__stex_refs_str {
1795       \exp_args:Ne \_stex_sref_do_aux:n {
1796         \STEXInternalSetSrefSymURL{#1}{\l__stex_refs_str / symbol? #1}
1797       }
1798     }{
1799       \_stex_sref_do_aux:n { \STEXInternalSetSrefSymURL{#1}{} }
1800     }
1801   }{
1802     \_stex_sref_do_aux:n { \STEXInternalSetSrefSymURL{#1}{} }
1803   }
1804 }
1805
1806 \cs_new_protected:Nn \stex_ref_new_sym_target:n {
1807   \cs_if_exist:NT \hypertarget{
1808     \exp_args:Ne \hypertarget{\tl_to_str:n{sref@sym@ #1}}{}
1809     \str_gset:cx{\tl_to_str:n{r@sref@sym@ #1}}{\tl_to_str:n{sref@sym@ #1}}
1810   }
1811 }
1812
1813 \cs_new_protected:Nn \stex_ref_new_sym_target:nn {
1814   \str_if_eq:nnTF{#1}{#2}{
1815     \stex_ref_new_sym_target:n{#1}
1816   }
```

```

1817     \str_gset:cn{g_stex_sref_sym_ #1 _label}{#2}
1818 }
1819 }
```

(End of definition for `\stex_ref_new_sym_target:n`. This function is documented on page 121.)

### \srefsym

```

1820 \NewDocumentCommand \srefsym { m m }{
1821   \stex_get_symbol:n { #1 }
1822   \exp_args:Ne
1823   \__stex_refs_sym_aux:nn{\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
1824 }
1825
1826 \cs_new_protected:Npn \__stex_refs_do_internal_link:nn #1 {
1827   \cs_if_exist:NTF \hyperlink {
1828     \hyperlink{#1}
1829   }\use:n
1830 }
1831
1832 \cs_new_protected:Npn \__stex_refs_do_url_link:nn {
1833   \cs_if_exist:NTF \href \href \use_i:nn
1834 }
1835
1836 \cs_new_protected:Npn \__stex_refs_sym_aux:nn #1 {
1837   \cs_if_exist:cTF{\tl_to_str:n{r@sref@sym@#1}}{
1838     \exp_args:Ne \__stex_refs_do_internal_link:nn{\tl_to_str:n{sref@sym@#1}}
1839   }{
1840     \str_if_exist:cTF{g_stex_sref_sym_#1_label}{
1841       \exp_args:Ne \__stex_refs_sym_aux:nn{\use:c{g_stex_sref_sym_#1_label}}
1842     }{
1843       \str_if_empty:cTF{g_stex_sref_sym_#1_target}{
1844         \exp_args:Ne \__stex_refs_do_internal_link:nn{\tl_to_str:n{sref@sym@#1}}
1845       }{
1846         \exp_args:Ne \__stex_refs_do_url_link:nn{\use:c{g_stex_sref_sym_#1_target}}
1847       }
1848     }
1849   }
1850 }
```

(End of definition for `\srefsym`. This function is documented on page 89.)

### \srefsymuri

```

1851 \cs_new_protected:Npn \srefsymuri #1 {
1852   \__stex_refs_sym_aux:nn{#1}
1853 }
```

(End of definition for `\srefsymuri`. This function is documented on page 89.)

#### 13.4.4 Inputs

```
1854 <@=stex_inputs>
```

`\stex_resolve_path_pair:Nnn`  
`\stex_resolve_path_pair:Nxx`

```

1855 \cs_new_protected:Nn \stex_resolve_path_pair:Nnn {
1856   \stex_debug:nn{resolving~path}{#3~in~[#2]}
```

```

1857 \str_if_empty:nTF{#2}{
1858   \prop_if_exist:NTF \l_stex_current_archive_prop {
1859     \str_set:Nx #1 {
1860       \prop_item:Nn \l_stex_current_archive_prop { path } / source /
1861       #3}
1862       \stex_debug:nn{resolving~path}{In~current~archive~}
1863       \prop_item:Nn \l_stex_current_archive_prop { id }
1864       ;~result:~#1}
1865   }{
1866     \str_set:Nx #1 {\stex_file_use:N \c_stex_pwd_file / .. / #3 }
1867     \stex_debug:nn{resolving~path}{No~current~archive;~result:~#1}
1868   }
1869 }{
1870   \stex_require_archive:n { #2 }
1871   \str_set:Nx #1 {
1872     \prop_item:cn {c_stex_mathhub_#2_manifest_prop} { path } / source /
1873     #3}
1874     \stex_debug:nn{resolving~path}{result:~#1}
1875   }
1876 }
1877 \cs_generate_variant:Nn \stex_resolve_path_pair:Nnn {Nxx}

```

(End of definition for \stex\_resolve\_path\_pair:Nnn. This function is documented on page 134.)

```

\inputref
\mhinput
\ifinputref
1878 \newif \ifinputref \inputreffalse
1879
1880 \cs_new_protected:Nn \__stex_inputs_mhinput:nn {
1881   \stex_in_archive:nn {#1} {
1882     \ifinputref
1883       \stex_input_with_hooks:n{ \prop_item:Nn \l_stex_current_archive_prop {path} / source /
1884     \else
1885       \inputreftrue
1886       \stex_input_with_hooks:n{ \prop_item:Nn \l_stex_current_archive_prop {path} / source /
1887       \inputreffalse
1888     \fi
1889   }
1890 }
1891
1892 \NewDocumentCommand \mhinput { O{} m }{
1893   \exp_args:NNx\exp_args:Nnx\__stex_inputs_mhinput:nn{ \tl_to_str:n{#1} }{ \tl_to_str:n{#2} }
1894 }
1895
1896 \cs_new_protected:Nn \__stex_inputs_inputref_html:nn {
1897   \str_clear:N \l_tmpa_str
1898   \prop_get:NnNF \l_stex_current_archive_prop { narr } \l_tmpa_str {
1899     \prop_get:NnNF \l_stex_current_archive_prop { ns } \l_tmpa_str {}
1900   }
1901   \tl_if_empty:nTF{ #1 }{
1902     \IfFileExists{#2}{
1903       \ifvmode\noindent\fi\stex_annotation_invisible:nn{shtml:inputref={%
1904         \l_tmpa_str / #2
1905       }}{}}
1906   }

```

```

1907     \stex_input_with_hooks:n{#2}
1908   }
1909 }
1910 \IfFileExists{ \prop_item:Nn \l_stex_current_archive_prop { path } / source / #2 }{
1911   \ifvmode\noindent\fi\stex_annotation_invisible:nn{shtml:inputref=}
1912   \l_tmpa_str / #2
1913 }
1914 }
1915 \input{ \prop_item:Nn \l_stex_current_archive_prop { path } / source / #2 }
1916 }
1917 }
1918 }
1919
1920 \cs_new_protected:Nn \__stex_inputs_inputref_pdf:nn {
1921   \begingroup
1922   \inputreftrue
1923   \tl_if_empty:nTF{ #1 }{
1924     \stex_input_with_hooks:n{#2}
1925   }
1926   \stex_input_with_hooks:n{ \prop_item:Nn \l_stex_current_archive_prop { path } / source / #2 }
1927 }
1928 \endgroup
1929 }
1930
1931 \cs_new_protected:Nn \__stex_inputs_inputref:nn {
1932   \stex_in_archive:nn {#1} {
1933     \stex_if_html_backend:TF
1934     \__stex_inputs_inputref_html:nn
1935     \__stex_inputs_inputref_pdf:nn
1936     {##1}{#2}
1937   }
1938 }
1939
1940 \NewDocumentCommand \inputref { O{} m }{
1941   \exp_args:NNx \exp_args:Nnx \__stex_inputs_inputref:nn{ \tl_to_str:n{#1} }{ \tl_to_str:n{#2} }
1942 }

```

(End of definition for `\inputref`, `\mhinput`, and `\ifinputref`. These functions are documented on page 83.)

### \addmhbibresource

```

1943 \cs_new_protected:Nn \__stex_inputs_bibresource:n {
1944   \__stex_inputs_up_archive:nn{#1}{bib}
1945   \seq_if_empty:NTF \l__stex_inputs_libinput_files_seq {
1946     \msg_error:nnxx{stex}{error/nofile}{\exp_not:N\addmhbibresource}{#1.bib}
1947   }
1948   \seq_map_inline:Nn \l__stex_inputs_libinput_files_seq {
1949     \addbibrssource{ ##1 }
1950   }
1951 }
1952 }
1953 \newcommand\addmhbibresource[2][]{%
1954   \tl_if_empty:nTF{#1}{%
1955     \__stex_inputs_bibresource:n{#2}

```

```

1956     }{
1957         \stex_in_archive:nn{#1}{\__stex_inputs_bibresource:n{#2}}
1958     }
1959 }
```

(End of definition for `\addmhbibresource`. This function is documented on page 80.)

### `\IfInputref`

```

1960 \stex_if_html_backend:TF{
1961     \newcommand \IfInputref[2]{
1962         \stex_annotate:nn{shtml:ifinputref=true}{#1}
1963         \stex_annotate:nn{shtml:ifinputref=false}{#2}
1964     }
1965 }{
1966     \newcommand \IfInputref[2]{
1967         \ifinputref #1 \else #2 \fi
1968     }
1969 }
```

(End of definition for `\IfInputref`. This function is documented on page 83.)

### `\libinput`

```

1970 \cs_new_protected:Nn \__stex_inputs_up_archive:nn {
1971     \prop_if_exist:NF \l_stex_current_archive_prop {
1972         \msg_error:nnn{stex}{error/notinarchive}\libinput
1973     }
1974     \prop_get:NnNF \l_stex_current_archive_prop {id} \l_stex_inputs_id_str {
1975         \msg_error:nnn{stex}{error/notinarchive}\libinput
1976     }
1977     \seq_clear:N \l__stex_inputs_libinput_files_seq
1978
1979     \str_set:Nx \l__stex_inputs_path_str { \prop_item:Nn \l_stex_current_archive_prop { path } }
1980     \seq_map_inline:Nn \c_stex_mathhub_files {
1981         \exp_args:No \stex_str_if_starts_with:nnT \l__stex_inputs_path_str {##1} {
1982             \seq_map_break:n { \seq_set_split:Nnn \l__stex_inputs_path_seq / {##1} }
1983         }
1984     }
1985
1986     \seq_set_split:NnV \l__stex_inputs_id_seq / \l__stex_inputs_id_str
1987
1988     \bool_while_do:nn { ! \seq_if_empty_p:N \l__stex_inputs_id_seq }{
1989         \str_set:Nx \l__stex_inputs_path_str {\stex_file_use:N \l__stex_inputs_path_seq / meta-i
1990         \IfFileExists{ \l__stex_inputs_path_str }{
1991             \exp_args:NNo \seq_if_in:NnF \l__stex_inputs_libinput_files_seq \l__stex_inputs_path_s
1992                 \seq_put_right:No \l__stex_inputs_libinput_files_seq \l__stex_inputs_path_str
1993             }
1994         }{}
1995         \seq_pop_left:NN \l__stex_inputs_id_seq \l__stex_inputs_path_str
1996         \seq_put_right:No \l__stex_inputs_path_seq \l__stex_inputs_path_str
1997     }
1998
1999     \str_set:Nx \l__stex_inputs_path_str {\stex_file_use:N \l__stex_inputs_path_seq / lib / #1
2000     \IfFileExists{ \l__stex_inputs_path_str }{
2001         \exp_args:NNo \seq_if_in:NnF \l__stex_inputs_libinput_files_seq \l__stex_inputs_path_s
2002             \seq_put_right:No \l__stex_inputs_libinput_files_seq \l__stex_inputs_path_str
```

```

2003     }
2004   }{}
2005 }
2006
2007 \cs_new_protected:Nn \__stex_inputs_libinput:n {
2008   \__stex_inputs_up_archive:nn{#1}{tex}
2009   \seq_if_empty:NTF \l__stex_inputs_libinput_files_seq {
2010     \msg_error:nnxx{stex}{error/nofile}{\exp_not:N\libinput}{#1.tex}
2011   }{
2012     \seq_map_inline:Nn \l__stex_inputs_libinput_files_seq {
2013       \input{ ##1 }
2014     }
2015   }
2016 }
2017
2018 \newcommand \libinput [2] [] {
2019   \tl_if_empty:nTF{#1} {
2020     \__stex_inputs_libinput:n{#2}
2021   }{
2022     \stex_in_archive:nn{#1}{\__stex_inputs_libinput:n{#2}}
2023   }
2024 }

```

(End of definition for `\libinput`. This function is documented on page 80.)

### \libusepackage

```

2025 \newcommand\libusepackage[2] [] {
2026   \__stex_inputs_up_archive:nn{#2}{sty}
2027   \int_compare:nNnTF {\seq_count:N \l__stex_inputs_libinput_files_seq} = 1 {
2028     \str_set:Nx \l__stex_inputs_tmp_str {\seq_item:Nn \l__stex_inputs_libinput_files_seq 1}
2029     \exp_args:Nne \use:n {\usepackage[#1]} {
2030       \str_range:Nnn\l__stex_inputs_tmp_str 1 {-5}
2031     }
2032   }{
2033     \msg_fatal:nnnn{stex}{error/nofile}{\libusepackage}{#1.sty}
2034   }
2035 }

```

(End of definition for `\libusepackage`. This function is documented on page 80.)

### \mhgraphics

#### \cmhgraphics

#### \lstinputmhlisting

#### \clstinputmhlisting

#### \mhtikzinput

#### \cmhtikzinput

```

2036 \str_new:N \l__stex_inputs_gin_repo_str
2037 \ltx@ifpackageloaded{graphicx}{\use:n}{\AtEndOfPackageFile{graphicx}}{
2038   \define@key{Gin}{archive}{%
2039     \stex_require_archive:n { #1 }
2040     \tl_set:Nx\Gin@mrepos{\prop_item:cn{ c_stex_mathhub_#1 _manifest_prop }{path} / source
2041   }
2042   \providecommand\mhgraphics[2][]{%
2043     \tl_set:Nx\Gin@mrepos{%
2044       \prop_item:Nn \l_stex_current_archive_prop {path} / source /
2045     }
2046     \setkeys{Gin}{#1}
2047     \includegraphics[#1]{ \Gin@mrepos #2 }
2048   }
2049   \providecommand\cmhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}

```

```

2050 }
2051
2052 \ltx@ifpackageloaded{listings}{\use:n}{\AtEndOfPackageFile{listings}}{
2053   \define@key{lst}{archive}{
2054     \stex_require_archive:n { #1 }
2055     \tl_set:Nx\Gin@mrepos{\prop_item:cn{ c_stex_mathhub_#1 _manifest_prop }{path} / source
2056   }
2057   \newcommand\lstinputmhlisting[2][]{%
2058     \def\lst@mrepos{
2059       \prop_item:Nn \l_stex_current_archive_prop {path} / source /
2060     }
2061     \setkeys{lst}{#1}%
2062     \lstinputlisting[#1]{\lst@mrepos #2}
2063   \newcommand\clstinputmhlisting[2][]{\begin{center}\lstinputmhlisting[#1]{#2}\end{center}}
2064 }
2065
2066 \ltx@ifpackageloaded{tikzinput}{\use:n}{\AtEndOfPackageFile{tikzinput}}{
2067   \define@key{Gin}{archive}{
2068     \stex_require_archive:n { #1 }
2069     \tl_set:Nx\Gin@mrepos{\prop_item:cn{ c_stex_mathhub_#1 _manifest_prop }{path} / source
2070   }
2071   \newcommand\mhtikzinput[2][]{%
2072     \str_clear:N \l__stex_inputs_gin_repo_str
2073     \def\Gin@mrepos{
2074       \prop_item:Nn \l_stex_current_archive_prop {path} / source /
2075     }
2076     \setkeys{Gin}{#1}%
2077     \exp_args:No \stex_in_archive:nn \l__stex_inputs_gin_repo_str {
2078       \tikzinput[#1]{\Gin@mrepos #2}
2079     }
2080   }
2081   \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
2082 }

```

(End of definition for `\mhgraphics` and others. These functions are documented on page 83.)

### \libusetikzlibrary

```

2083 \cs_new_protected:Nn \__stex_inputs_usetikzlibrary_i:nn {
2084   \pgfkeys@spdef\pgf@temp{#1}
2085   \expandafter\ifx\csname tikz@library@\pgf@temp @loaded\endcsname\relax%
2086   \expandafter\global\expandafter\let\csname tikz@library@\pgf@temp @loaded\endcsname=\pgfutil@empty
2087   \expandafter\edef\csname tikz@library@#1@atcode\endcsname{\the\catcode`@}
2088   \expandafter\edef\csname tikz@library@#1@barcode\endcsname{\the\catcode`\|}%
2089   \expandafter\edef\csname tikz@library@#1@dollarcode\endcsname{\the\catcode`\$}
2090   \catcode`\@=11
2091   \catcode`\|=12
2092   \catcode`\$=3
2093   \pgfutil@InputIfFileExists{#2}{}{%
2094     \catcode`\@=\csname tikz@library@#1@atcode\endcsname
2095     \catcode`\|= \csname tikz@library@#1@barcode\endcsname
2096     \catcode`\$=\csname tikz@library@#1@dollarcode\endcsname
2097   }
2098
2099 \cs_new_protected:Nn \__stex_inputs_usetikzlibrary:n{

```

```

2100   \__stex_inputs_up_archive:nn{tikzlibrary#1}{code.tex}
2101   \int_compare:nNnTF {\seq_count:N \l__stex_inputs_libinput_files_seq} = 1 {
2102     \exp_args:Nne \__stex_inputs_usetikzlibrary_i:nn{#1}{\seq_item:Nn \l__stex_inputs_lib
2103   }{
2104     \msg_fatal:nnnn{stex}{error/nofile}{\libusetikzlibrary}{tikzlibrary#1.code.tex}
2105   }
2106 }
2107
2108 \newcommand \libusetikzlibrary [2] [] {
2109   \cs_if_exist:N \usetikzlibrary {
2110     \msg_error:nnx{stex}{error/notikz}{\tl_to_str:n{\libusetikzlibrary}}
2111   }
2112   \tl_if_empty:nTF{#1}{
2113     \__stex_inputs_usetikzlibrary:n{#2}
2114   }{
2115     \stex_in_archive:nn{#1}{\__stex_inputs_usetikzlibrary:n{#2}}
2116   }
2117 }

```

(End of definition for `\libusetikzlibrary`. This function is documented on page 118.)

## 13.5 SMS Mode

```
2118 <@=stex_smsmode>
```

Macros and environments allowed in sms mode:

```

2119 \tl_new:N \g__stex_smsmode_allowed_tl
2120 \tl_new:N \g__stex_smsmode_allowed_escape_tl
2121 \seq_new:N \g__stex_smsmode_allowedenvs_seq

```

`\stex_sms_allow:N`

```

2122 \cs_new_protected:Nn \stex_sms_allow:N {
2123   \tl_gput_right:Nn \g__stex_smsmode_allowed_tl {#1}
2124 }
2125

```

```

2126 \cs_new_protected:Nn \stex_sms_allow_escape:N {
2127   \tl_gput_right:Nn \g__stex_smsmode_allowed_escape_tl {#1}
2128 }
2129

```

```

2130 \cs_new_protected:Nn \stex_sms_allow_env:n {
2131   \exp_args:NNx \seq_gput_right:Nn \g__stex_smsmode_allowedenvs_seq {\tl_to_str:n{#1}}
2132 }

```

(End of definition for `\stex_sms_allow:N`, `\stex_sms_allow_escape:N`, and `\stex_sms_allow_env:n`. These functions are documented on page 135.)

Some initial allowed macros:

```

2133 \stex_sms_allow:N \makeatletter
2134 \stex_sms_allow:N \makeatother
2135 \stex_sms_allow:N \ExplSyntaxOn
2136 \stex_sms_allow:N \ExplSyntaxOff
2137 \stex_sms_allow:N \rustexBREAK

```

`\stex_if_smsmode_p:`

`\stex_if_smsmode:TF`

```
2138 \bool_new:N \g__stex_smsmode_bool
```

```

2139 \bool_set_false:N \g__stex_smsmode_bool
2140 \prg_new_conditional:Nnn \stex_if_smsmode: { p, T, F, TF } {
2141   \bool_if:NTF \g__stex_smsmode_bool \prg_return_true: \prg_return_false:
2142 }

```

(End of definition for `\stex_if_smsmode:TF`. This function is documented on page 135.)

`\stex_sms_allow_import:Nn`

```

\stex_sms_allow_import_env:nn
2143 \tl_new:N \g__stex_smsmode_allowed_import_tl
2144 \seq_new:N \g__stex_smsmode_allowed_import_env_seq
2145 \cs_new_protected:Nn \stex_sms_allow_import:Nn {
2146   \tl_gput_right:Nn \g__stex_smsmode_allowed_import_tl {#1}
2147   \tl_gset:cn{\tl_to_str:n{#1}---smsmode} {#2}
2148 }
2149 \cs_new_protected:Nn \stex_sms_allow_import_env:nn {
2150   \exp_args:NNx \seq_gput_right:Nn \g__stex_smsmode_allowed_import_env_seq {\tl_to_str:n{#1}
2151   \tl_gset:cn{\tl_to_str:n{#1}---env---smsmode} {#2}
2152 }
2153
2154 \tl_new:N \g_stex_sms_import_code

```

(End of definition for `\stex_sms_allow_import:Nn` and `\stex_sms_allow_import_env:nn`. These functions are documented on page 136.)

`\stex_file_in_smsmode:nn`

`\stex_file_in_smsmode:on`

```

\stex_file_in_smsmode:on
2155 \cs_new_protected:Nn \__stex_smsmode_in_smsmode:n { \stex_suppress_html:n {
2156   \vbox_set:Nn \l_tmpa_box {
2157     \bool_set_true:N \g__stex_smsmode_bool
2158     \bool_set_false:N \stex_html_do_output_bool
2159     #1
2160   }
2161   \%box_clear:N \l_tmpa_box
2162 } }
2163
2164 \quark_new:N \q__stex_smsmode_break
2165
2166 \cs_new_protected:Nn \__stex_smsmode_start_smsmode:n {
2167   \everyeof{\q__stex_smsmode_break\exp_not:N}
2168   \let\stex_smsmode_do:\__stex_smsmode_smsmode_do:
2169   \exp_after:wN \exp_after:wN \exp_after:wN
2170   \stex_smsmode_do:
2171   \cs:w @ input\cs_end: "#1" \relax
2172 }
2173
2174 \cs_new_protected:Nn \stex_file_in_smsmode:nn {
2175   \seq_gclear:N \l__stex_smsmode_importmodules_seq
2176   \seq_gclear:N \l__stex_smsmode_sigmodules_seq
2177   \tl_clear:N \g_stex_sms_import_code
2178   \group_begin:
2179     \let \l_stex_metatheory_uri \c_stex_default_metatheory
2180     \cs_set:Npn \stex_check_term:n {#1 {}}
2181     \seq_clear:N \l_stex_all_modules_seq
2182     \str_clear:N \l_stex_current_module_str
2183     #2
2184     \stex_filestack_push:n{#1}

```

```

2185   \__stex_smsmode_in_smsmode:n {
2186     \let \__stex_smsmode_do_aux_curr:N \__stex_smsmode_do_aux_imports:N
2187     \tl_map_inline:Nn \g__stex_smsmode_allowed_import_tl {
2188       \use:c{\tl_to_str:n{##1}---smsmode}
2189     }
2190     \seq_map_inline:Nn \g__stex_smsmode_allowed_import_env_seq {
2191       \use:c{\tl_to_str:n{##1}---env---smsmode}
2192     }
2193     \__stex_smsmode_start_smsmode:n{#1}
2194   }
2195   \__stex_smsmode_in_smsmode:n \g_stex_sms_import_code
2196   \__stex_smsmode_in_smsmode:n {
2197     \let \__stex_smsmode_do_aux_curr:N \__stex_smsmode_do_aux_normal:N
2198     \__stex_smsmode_start_smsmode:n{#1}
2199   }
2200   \stex_filestack_pop:
2201 \group_end:
2202 }
2203 \cs_generate_variant:Nn \stex_file_in_smsmode:nn {on}

```

(End of definition for `\stex_file_in_smsmode:nn`. This function is documented on page 135.)

#### `\stex_smsmode_do:`

```

2204 \cs_new_protected:Nn \__stex_smsmode_smsmode_do: {
2205   \%stex_if_smsmode:T {
2206     \__stex_smsmode_do:w
2207   %}
2208 }
2209 \let\stex_smsmode_do:\relax
2210
2211
2212 \cs_new:Nn \__stex_smsmode_check_cs:NNn {
2213   \exp_after:wN\if\exp_after:wN\relax\exp_not:N#3
2214   \exp_after:wN#1\exp_after:wN#3\else
2215   \exp_after:wN#2\fi
2216 }
2217
2218 \cs_new_protected:Npn \__stex_smsmode_do:w #1 {
2219   \exp_args:Nx \tl_if_empty:nTF { \tl_tail:n{ #1 } }{
2220     \__stex_smsmode_check_cs:NNn \__stex_smsmode_do_aux:N \__stex_smsmode_do:w { #1 }
2221   }
2222   \__stex_smsmode_do:w
2223 }
2224 }
2225
2226 \cs_new_protected:Nn \__stex_smsmode_do_aux:N {
2227   \cs_if_eq:NNF #1 \q__stex_smsmode_break {
2228     \__stex_smsmode_do_aux_curr:N #1
2229   }
2230 }
2231
2232 \cs_new_protected:Nn \__stex_smsmode_do_aux_imports:N {
2233   \%stex_debug:nn{sms}{Checking~\tl_to_str:n{#1}~in~import}
2234   \tl_if_in:NnTF \g__stex_smsmode_allowed_import_tl {#1} {

```

```

2235   \stex_debug:nn{sms}{Executing~\tl_to_str:n{#1}~in~import}
2236   #1
2237 }{
2238   \cs_if_eq:NNTF \begin{#1} {
2239     \__stex_smsemode_check_begin:Nn \g__stex_smsemode_allowed_import_env_seq
2240   }{
2241     \cs_if_eq:NNTF \end{#1} {
2242       \__stex_smsemode_check_end:Nn \g__stex_smsemode_allowed_import_env_seq
2243     }{
2244       \__stex_smsemode_do:w
2245     }
2246   }
2247 }
2248 }
2249
2250 \cs_new_protected:Nn \__stex_smsemode_do_aux_normal:N {
2251   % \stex_debug:nn{sms}{Checking~\tl_to_str:n{#1}~in~sms~mode}
2252   \tl_if_in:NnTF \g__stex_smsemode_allowed_tl {#1} {
2253     \stex_debug:nn{sms}{Executing~\tl_to_str:n{#1}}
2254     #1\__stex_smsemode_do:w
2255   }{
2256     \tl_if_in:NnTF \g__stex_smsemode_allowed_escape_tl {#1} {
2257       \stex_debug:nn{sms}{Executing~escaped~\tl_to_str:n{#1}}
2258       #1
2259     }{
2260       \cs_if_eq:NNTF \begin{#1} {
2261         \__stex_smsemode_check_begin:Nn \g__stex_smsemode_allowedenvs_seq
2262       }{
2263         \cs_if_eq:NNTF \end{#1} {
2264           \__stex_smsemode_check_end:Nn \g__stex_smsemode_allowedenvs_seq
2265         }{
2266           \__stex_smsemode_do:w
2267         }
2268       }
2269     }
2270   }
2271 }
2272
2273 \cs_new_protected:Nn \__stex_smsemode_check_begin:Nn {
2274   % \stex_debug:nn{sms}{Checking~environment~#2}
2275   \seq_if_in:NxTF #1 { \tl_to_str:n{#2} }{
2276     \stex_debug:nn{sms}{Environment~#2}
2277     \begin{#2}
2278   }{
2279     \__stex_smsemode_do:w
2280   }
2281 }
2282 \cs_new_protected:Nn \__stex_smsemode_check_end:Nn {
2283   % \stex_debug:nn{sms}{Checking~end~environment~#2}
2284   \seq_if_in:NxTF #1 { \tl_to_str:n{#2} }{
2285     \stex_debug:nn{sms}{End~Environment~#2}
2286     \end{#2}\__stex_smsemode_do:w
2287   }
2288   \%str_if_eq:nnTF{#2}{document} \endinput

```

```

2289     \__stex_smsmode_do:w
2290   }
2291 }
```

(End of definition for `\stex_smsmode_do`. This function is documented on page 136.)

## 13.6 Modules

### 13.6.1 The smodule-environment

```
2292 <@=stex_modules>
```

`\l_stex_current_module_str` The current module:

```
2293 \str_new:N \l_stex_current_module_str
```

(End of definition for `\l_stex_current_module_str`. This variable is documented on page 122.)

`\l_stex_all_modules_seq` Stores all modules currently in scope

```
2294 \seq_new:N \l_stex_all_modules_seq
```

(End of definition for `\l_stex_all_modules_seq`. This variable is documented on page 122.)

`\stex_every_module:n`

```

2295 <@=stex_module_setup>
2296 \tl_clear:N \g_stex_every_module_tl {
2297 }
2298 \cs_new_protected:Nn \stex_every_module:n {
2299   \tl_gput_right:Nn \g_stex_every_module_tl { #1 }
2300 }
```

(End of definition for `\stex_every_module:n`. This function is documented on page 122.)

`\stex_module_setup:n` Sets up a new module:

```

2301 \cs_new_protected:Npn \stex_module_setup:n {
2302   \stex_if_in_module:TF \__stex_module_setup_setup_nested:n \__stex_module_setup_setup_top:n
2303 }
2304 \cs_new_protected:Nn \__stex_module_setup_setup_top:n {
2305   \__stex_module_setup_get_uri_str:n{#1}
2306   \stex_debug:nn{module}{Module~URI:~\l_stex_module_setup_ns_str?#1}
2307   \str_if_empty:NTF \l_stex_key_sig_str
2308   \stex_module_setup_top_nosig:n \__stex_module_setup_setup_top_sig:n {\l_stex_module_setup_ns_str}
2309   \stex_metagroup_new:o \l_stex_current_module_str
2310   \g_stex_every_module_tl
2311   \stex_execute_in_module:x {
2312     \stex_do_deprecation:n{#1}
2313   }
2314   \__stex_module_setup_load_meta:
2315 }
2316
2317 \cs_new_protected:Nn \stex_module_setup_top_nosig:n {
2318   \stex_if_module_exists:nTF{#1} {
2319     \stex_debug:nn{modules}{(already exists)}
2320   }{
2321     \tl_gclear:c{c_stex_module_ #1 _code}
```

```

2323   \prop_gclear:c{c_stex_module_ #1 _morphisms_prop }
2324   \prop_gclear:c{c_stex_module_ #1 _symbols_prop }
2325   \prop_gclear:c{c_stex_module_ #1 _notations_prop }
2326 }
2327 \str_set:Nx \l_stex_current_module_str {#1}
2328 \seq_put_right:No \l_stex_all_modules_seq \l_stex_current_module_str
2329 }

2330
2331 \cs_new_protected:Nn \__stex_module_setup_top_sig:n {
2332   \stex_if_module_exists:nTF{#1} {
2333     \stex_debug:nn{modules}{(already exists)}
2334   }{
2335     \stex_debug:nn{modules}{(needs loading)}
2336     \__stex_module_setup_load_sig:
2337   }
2338 \% \stex_if_smsmode:F { % WHY?
2339   \stex_activate_module:x {
2340     #1
2341   }
2342   %}
2343   \str_set:Nx\l_stex_current_module_str{#1}
2344 }

2345
2346 \cs_new_protected:Nn \__stex_module_setup_load_sig: {
2347   \stex_file_split_off_ext:NN \l_stex_module_setup_sigfile \g_stex_current_file
2348   \stex_file_split_off_lang:NN \l_stex_module_setup_sigfile \l__stex_module_setup_sigfile
2349   \exp_args:Ne \stex_file_in_smsmode:nn {
2350     \stex_file_use:N \l__stex_module_setup_sigfile . \l_stex_key_sig_str . tex
2351   }{}
2352 }

2353
2354 \cs_new_protected:Nn \__stex_module_setup_nested:n {
2355   \exp_after:wN
2356   \__stex_module_setup_split_module:n \l_stex_current_module_str \__stex_module_setup_end:
2357   \stex_debug:nn{module}{Nested-Module-URI:~\l_stex_current_module_str}
2358   \seq_put_right:No \l_stex_all_modules_seq \l_stex_current_module_str
2359   \stex_metagroup_new:o \l_stex_current_module_str
2360 }

2361
2362
2363 \cs_new_protected:Nn \__stex_module_setup_get_uri_str:n {
2364   \str_clear:N \l_stex_module_setup_ns_str
2365   \stex_map_uri:Nnnnn \l_stex_current_ns_uri {
2366     \str_set:Nx \l__stex_module_setup_ns_str{##1\c_colon_str/}
2367   }{
2368     \seq_set_split:Nnn \l__stex_module_setup_seq / {##1}
2369     \seq_pop_right:NN \l__stex_module_setup_seq \l__stex_module_setup_seg
2370     \exp_args:No \str_if_eq:nnF \l__stex_module_setup_seg {#1} {
2371       \seq_put_right:No \l__stex_module_setup_seq \l__stex_module_setup_seg
2372     }
2373     \tl_put_right:Nx \l__stex_module_setup_ns_str {\seq_use:Nn \l__stex_module_setup_seq /}
2374   }{}{ }
2375 }

2376

```

```

2377 \cs_new_protected:Npn \__stex_module_setup_split_module:n #1?#2 \__stex_module_setup_end: #3
2378   \__stex_module_setup_top_nosig:n { #1 ? #2 / #3}
2379 }
2380
2381 \bool_new:N \l_stex_in_meta_bool
2382 \bool_set_false:N \l_stex_in_meta_bool
2383
2384 \cs_new_protected:Nn \__stex_module_setup_load_meta: {
2385   \tl_if_empty:NF \l_stex_metatheory_uri {
2386     \stex_execute_in_module:x{
2387       \stex_pseudogroup_with:nn{ \l_stex_in_meta_bool }{
2388         \stex_activate_module:n { \stex_uri_use:N \l_stex_metatheory_uri }
2389       }
2390     }
2391   }
2392 }
2393
2394 \__stex_modules

```

(End of definition for `\stex_module_setup:n`. This function is documented on page 122.)

### `\stex_close_module:`

```

2395 \cs_new:Nn \stex_close_module: {
2396   \bool_if:NT \c_stex_persist_write_mode_bool \__stex_modules_persist_module:
2397   \stex_debug:nn{module}{
2398     Closing~module~\l_stex_current_module_str^~J
2399     Code:~\expandafter\meaning\csname c_stex_module_\l_stex_current_module_str _code\endcsna
2400     Imports:\exp_after:wN \prop_to_keyval:N \cs:w c_stex_module_\l_stex_current_module_str _Imports
2401     Declarations:\exp_after:wN \prop_to_keyval:N \cs:w c_stex_module_\l_stex_current_module_str _Declarations
2402     Notations:\exp_after:wN \prop_to_keyval:N \cs:w c_stex_module_\l_stex_current_module_str _Notations
2403   }
2404 }
2405
2406 \cs_new_protected:Nn \__stex_modules_persist_module: {
2407   \stex_persist:e {
2408     \__stex_modules_restore_module:nnnn { \l_stex_current_module_str }{
2409       \exp_after:wN \prop_to_keyval:N \cs:w
2410         c_stex_module_\l_stex_current_module_str _morphisms_prop
2411       \cs_end:
2412     }{
2413       \exp_after:wN \prop_to_keyval:N \cs:w
2414         c_stex_module_\l_stex_current_module_str _symbols_prop
2415       \cs_end:
2416     }{
2417       \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:n
2418       \exp_after:wN \exp_after:wN \exp_after:wN
2419       { \cs:w c_stex_module_\l_stex_current_module_str _code \cs_end: }
2420     }{}
2421     \prop_map_function:cN{c_stex_module_\l_stex_current_module_str _notations_prop}
2422       \__stex_modules_persist_nots_i:nn
2423       \exp_not:N \STEXRestoreNotsEnd {}
2424   }
2425 }
2426

```

```

2427 \cs_new_protected:Nn \__stex_modules_restore_module:nnnn {
2428   \prop_gset_from_keyval:c{c_stex_module_ \tl_to_str:n{\#1}_morphisms_prop}{\#2}
2429   \cs_set:Npn \__stex_modules_tl {\#3}
2430   \exp_args:Nno \prop_gset_from_keyval:c{c_stex_module_ \tl_to_str:n{\#1}_symbols_prop}\__stex_
2431   \prop_map_inline:c{c_stex_module_ \tl_to_str:n{\#1}_symbols_prop} {
2432     \stex_ref_new_symbol:n{\#1?##1}
2433   }
2434   \cs_gset:cpn{c_stex_module_ \tl_to_str:n{\#1}_code}{\#4}
2435   \prop_gclear:c{c_stex_module_ \tl_to_str:n{\#1}_notations_prop}
2436   \str_set:Nn \l__stex_modules_restore_mod_str {\#1}
2437   \group_begin:
2438     \catcode`_=8\relax
2439     \catcode`:=12\relax
2440     \__stex_modules_restore_nots:n
2441   }
2442 
2443 \cs_new:Nn \__stex_modules_persist_nots_i:nn {
2444   \exp_not:n{\#2}
2445 }
2446 
2447 \quark_new:N \STEXRestoreNotsEnd
2448 
2449 \cs_new_protected:Nn \__stex_modules_restore_nots:n {
2450   \__stex_modules_restore_nots_i:n
2451 }
2452 
2453 \cs_new_protected:Nn \__stex_modules_restore_nots_i:n {
2454   \tl_if_eq:nnTF{\#1}{\STEXRestoreNotsEnd} {
2455     \group_end:
2456   }{
2457     \__stex_modules_restore_nots_ii:nnnn {\#1}
2458   }
2459 }
2460 
2461 \cs_new_protected:Nn \__stex_modules_restore_nots_ii:nnnn {
2462   \cs_set:Npn \l__stex_modules_tl {{\#4}{\#5}}
2463   \exp_args:NN\use:nn\prop_gput:cnn{
2464     {c_stex_module_ \l__stex_modules_restore_mod_str _notations_prop}
2465     {\tl_to_str:n{\#1!\#2}}{
2466       {\tl_to_str:n{\#1}}{\tl_to_str:n{\#2}}{\#3}
2467       \exp_args:No \exp_not:n \l__stex_modules_tl
2468     }
2469   }
2470   \__stex_modules_restore_nots_i:n
2471 }

```

(End of definition for `\stex_close_module`.. This function is documented on page 122.)

`\l_stex_metatheory_uri`

```
2472 \tl_new:N \l_stex_metatheory_uri
```

(End of definition for `\l_stex_metatheory_uri`. This variable is documented on page ??.)

`\setmetatheory`

```

2473 \cs_new_protected:Nn \__stex_modules_set_metatheory:nn {
2474   \group_begin:
2475     \stex_debug:nn{metatheory}{Setting-metatheory~[#1]#2}
2476     \stex_import_module_uri:nn { #1 } { #2 }
2477     \stex_debug:nn{metatheory}{Here:^^J
2478       \l_stex_import_archive_str^^J
2479       \l_stex_import_path_str^^J
2480       \l_stex_import_name_str^^J
2481     }
2482     \stex_import_require_module:ooo
2483       \l_stex_import_archive_str
2484       \l_stex_import_path_str
2485       \l_stex_import_name_str
2486     \stex_debug:nn{metatheory}{Found:~\l_stex_import_ns_str}
2487     \exp_args:Nne \use:nn {
2488       \group_end: \stex_uri_resolve:Nn \l_stex_metatheory_uri
2489     }{\l_stex_import_ns_str}
2490   }
2491 
2492 \NewDocumentCommand \setmetatheory {O{} m}{
2493   \__stex_modules_set_metatheory:nn { #1 }{ #2 }
2494   \stex_smsmode_do:
2495 }
2496 \stex_sms_allow_escape:N \setmetatheory

```

(End of definition for `\setmetatheory`. This function is documented on page ??.)

Keys and key handling:

```

2497 \stex_keys_define:nnnn{smodule}{
2498   \str_clear:N \l_stex_key_sig_str
2499 }{
2500   meta .code:n = {
2501     \str_if_empty:nTF {#1}{
2502       \tl_clear:N \l_stex_metatheory_uri
2503     }{
2504       \stex_uri_resolve:Nx \l_stex_metatheory_uri { #1 }
2505     }
2506   },
2507   ns .code:n = {
2508     \stex_uri_resolve:Nx \l_stex_current_ns_uri { #1 }
2509   },
2510   lang .code:n = {
2511     \stex_set_language:n { #1 }
2512   },
2513   sig .str_set_x:N = \l_stex_key_sig_str ,
2514   creators .code:n = {} , % todo ?
2515   contributors .code:n = {} , % todo ?
2516   srccite .code:n = {} % todo ?
2517 }{id, title, style, deprecate}

```

`smodule (env.)`

```

2518 \stex_new_stylable_env:nnnnnnn {module} {O{} m}{
2519   \stex_keys_set:nn { smodule }{ #1 }
2520   \tl_set_eq:NN \thistitle \l_stex_key_title_tl
2521   \tl_if_empty:NF \thistitle {

```

```

2522     \exp_args:No \stexdoctitle \thistitle
2523 }
2524 \exp_args:Nx \stex_module_setup:n { \tl_to_str:n{ #2 } }
2525
2526 \stex_if_do_html:T {
2527     \exp_args:Nne \begin{stex_annotation_env} {
2528         shtml:theory={\l_stex_current_module_str},
2529         shtml:language={ \l_stex_current_language_str},
2530         shtml:signature={\l_stex_key_sig_str}
2531         \tl_if_empty:NF \l_stex_metatheory_uri {,
2532             shtml:metatheory={\stex_uri_use:N \l_stex_metatheory_uri}
2533         }
2534     }
2535     \stex_annotation_invisible:n{}
2536 }
2537 \stex_if_smsmode:F {
2538     \str_set_eq:NN \thismoduleuri \l_stex_current_module_str
2539     \tl_set:Nn \thismodulename {#2}
2540     \stex_style_apply:
2541 }
2542 \stex_smsmode_do:
2543 }{
2544     \stex_close_module:
2545     \stex_if_smsmode:F \stex_style_apply:
2546     \stex_if_do_html:T{ \end{stex_annotation_env} }
2547 }{}{}{s}
2548
2549 \stex_sms_allow_env:n{smodule}

```

\stex\_if\_in\_module\_p: Are we currently in a module?

```

\stex_if_in_module:TF
2550 \prg_new_conditional:Nnn \stex_if_in_module: {p, T, F, TF} {
2551     \str_if_empty:NTF \l_stex_current_module_str
2552     \prg_return_false: \prg_return_true:
2553 }

```

(End of definition for \stex\_if\_in\_module:TF. This function is documented on page 122.)

\stex\_if\_module\_exists\_p:n Does a module with this URI exist?

```

\stex_if_module_exists:nTF
2554 \prg_new_conditional:Nnn \stex_if_module_exists:n {p, T, F, TF} {
2555     \tl_if_exist:cTF { c_stex_module_#1_code }
2556     \prg_return_true: \prg_return_false:
2557 }

```

(End of definition for \stex\_if\_module\_exists:nTF. This function is documented on page 122.)

\stex\_do\_up\_to\_module:n Execute code in the current module (i.e. as if the \begin{(smodule)} was the current tex group)

```

2558 \cs_new_protected:Nn \stex_do_up_to_module:n {
2559     \exp_args:No \stex_metagroup_do_in:nn \l_stex_current_module_str {#1}
2560 }
2561 \cs_generate_variant:Nn \stex_do_up_to_module:n {x}

```

(End of definition for \stex\_do\_up\_to\_module:n. This function is documented on page 123.)

```

\stex_module_add_code:n
\stex_module_add_code:x 2562 \cs_new_protected:Nn \stex_module_add_code:n {
2563   \tl_gput_right:cn {c_stex_module_\l_stex_current_module_str _code} { #1 }
2564 }
2565 \cs_generate_variant:Nn \stex_module_add_code:n {x}

```

(End of definition for `\stex_module_add_code:n`. This function is documented on page 123.)

```

\stex_execute_in_module:n
\stex_execute_in_module:x 2566 \cs_new_protected:Nn \stex_execute_in_module:n { \stex_if_in_module:TF {
2567   \stex_module_add_code:n { #1 }
2568   \stex_do_up_to_module:n { #1 }
2569 }{ #1 }}
2570 \cs_generate_variant:Nn \stex_execute_in_module:n {x}

```

(End of definition for `\stex_execute_in_module:n`. This function is documented on page 122.)

### \STEXexport

```

2571 \NewDocumentCommand \STEXexport {} {
2572   \ExplSyntaxOn
2573   \__stex_modules_export:n
2574 }
2575 \cs_new_protected:Nn \__stex_modules_export:n {
2576   \stex_ignore_spaces_and_pars:#1\ExplSyntaxOff
2577   \stex_module_add_code:n { \stex_ignore_spaces_and_pars:#1}
2578   \stex_smsmode_do:
2579 }

```

Only allowed in modules, and allowed (escaped) in sms mode:

```

2580 \stex_deactivate_macro:Nn \STEXexport {module~environments}
2581 \stex_sms_allow_escape:N \STEXexport
2582 \stex_every_module:n {\stex_reactivate_macro:N \STEXexport}

```

(End of definition for `\STEXexport`. This function is documented on page 86.)

### \stex\_module\_add\_morphism:nnnn

\stex\_module\_add\_morphism:nnonn

\stex\_module\_add\_morphism:ooox

```

2583 \cs_new_protected:Nn \stex_module_add_morphism:nnnn {
2584   \exp_args:Nne \prop_gput:cnn{c_stex_module_\l_stex_current_module_str _morphisms_prop}{%
2585     \tl_if_empty:nTF{#1}{[#2]}{#1}
2586   }{#1}{#2}{#3}{#4}
2587 }
2588 \cs_generate_variant:Nn \stex_module_add_morphism:nnnn {nonn,ooox}

```

(End of definition for `\stex_module_add_morphism:nnnn`. This function is documented on page 130.)

```

\stex_module_add_symbol:nnnnnnN
#1 : {⟨Macro name⟩}
#2 : {⟨Name⟩}
#3 : {⟨arity⟩}
#4 : {⟨Arg num⟩}{⟨Arg str⟩}*}
#5 : Definiens
#6 : type
#7 : Return

```

#8 : Command

```
2589 \cs_new_protected:Nn \stex_module_add_symbol:nnnnnnnN {
2590   \stex_debug:nn{declaration}{New~declaration:~\l_stex_current_module_str?#2^^J
2591   Macro:#1^^JArity:#3~(#4)^~J
2592   Def:~\tl_to_str:n{#5}^~J
2593   Type:~\tl_to_str:n{#6}^~J
2594   Returns:~\tl_to_str:n{#7}
2595 }
2596 \%prop_gput:cnx{c_stex_module_\l_stex_current_module_str _symbols_prop}
2597 %{#2}{\exp_not:n{#1}{#2}{#3}{#4}{#5}{#6}{#7}\exp_not:n{#8}}
2598 \prop_gput:cnn{c_stex_module_\l_stex_current_module_str _symbols_prop}
2599 %{#2}{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}
2600 \tl_if_empty:nF{#1}{
2601   \stex_execute_in_module:n {
2602     \__stex_modules_activate_sym:n {#2}
2603   }
2604 }
2605 }
2606
2607 \cs_new_protected:Nn \__stex_modules_activate_sym:n {
2608   \prop_map_inline:cn{c_stex_module_\l_stex_current_module_str _symbols_prop}{
2609     \str_if_eq:nnT{#1}{##1} {
2610       \__stex_modules_activate_i:nnnnnnnn ##2
2611     }
2612   }
2613 }
2614 \cs_new_protected:Nn \__stex_modules_activate_i:nnnnnnnn {
2615   \stex_debug:nn{activating}{#1:\l_stex_current_module_str^~J
2616   \tl_to_str:n{#2}{#3}{#4}{#5}{#6}{#7}{#8}}
2617 }
2618 \cs_set:cp{#1} {
2619   \stex_invoke_symbol:nnnnnnN
2620   {\l_stex_current_module_str}
2621   \exp_not:n{#2}{#3}{#4}{#5}{#6}{#7}{#8}
2622 }
2623 \stex_debug:nn{activating}{done}
2624 \prop_map_break:
2625 }
```

(End of definition for \stex\_module\_add\_symbol:nnnnnnnN. This function is documented on page ??.)

```
\stex_module_add_notation:nnnn #1 : URI
\stex_module_add_notation:eoeeoo #2 : variant
\stex_set_notation_macro:nnnnn #3 : arity
#4 : macro body
#5 : op

2626 \cs_new_protected:Nn \stex_module_add_notation:nnnn {
2627   \stex_debug:nn{notations}{Adding~notation:~^~J
2628   #1~\c_hash_str#2~#3^~J
2629   to~\l_stex_current_module_str
2630 }
2631 \prop_gput:cnn{c_stex_module_\l_stex_current_module_str _notations_prop}
2632 %{#1!#2}{#1}{#2}{#3}{#4}{#5}
```

```

2633   \stex_execute_in_module:n {
2634     \__stex_modules_activate_not:nn{#1}{#2}
2635   }
2636 }
2637 \cs_generate_variant:Nn \stex_module_add_notation:nnnn {eoexo}
2638
2639
2640 \cs_new_protected:Nn \__stex_modules_activate_not:nn {
2641   \prop_map_inline:cn{c_stex_module_\l_stex_current_module_str _notations_prop} {
2642     \str_if_eq:nnT{#1!#2}{##1} {
2643       \prop_map_break:n{\stex_set_notation_macro:nnnn ##2 }
2644     }
2645   }
2646 }

```

(End of definition for `\stex_module_add_notation:nnnn` and `\stex_set_notation_macro:nnnn`. These functions are documented on page 125.)

```

\stex_set_notation_macro:nnnn
\stex_set_notation_macro:eoexo
2647 \cs_new_protected:Nn \stex_set_notation_macro:nnnn {
2648   \tl_set:cn {l_stex_notation_#1#2_cs}{#4}
2649   \cs_if_exist:cF{l_stex_notation_#1__cs} {
2650     \tl_set:cn {l_stex_notation_#1__cs}{#4}
2651   }
2652   \tl_if_empty:nF{#5} {
2653     \tl_set:cn{l_stex_notation_#1_op_#2_cs}{#5}
2654     \cs_if_exist:cF{l_stex_notation_#1_op__cs} {
2655       \cs_set_eq:cc {l_stex_notation_#1_op__cs}{l_stex_notation_#1_op_#2_cs}
2656     }
2657   }
2658 }
2659 \cs_generate_variant:Nn \stex_set_notation_macro:nnnn {eoexo}

```

(End of definition for `\stex_set_notation_macro:nnnn`. This function is documented on page 126.)

```

\stex_activate_module:n
\stex_activate_module:o
\stex_activate_module:x
2660 \cs_new_protected:Nn \stex_activate_module:n {
2661   \seq_if_in:NnF \l_stex_all_modules_seq { #1 } {
2662     \stex_debug:nn{modules}{Activating-module~#1^J\expandafter\meaning\csname c_stex_module
2663     \seq_put_right:Nn \l_stex_all_modules_seq { #1 }
2664     \stex_pseudogroup:nn{
2665       \str_set:Nn \l_stex_current_module_str {#1}
2666       \use:c{ c_stex_module_#1_code }
2667     }{
2668       \stex_pseudogroup_restore:N \l_stex_current_module_str
2669     }
2670   }
2671 }
2672 \cs_generate_variant:Nn \stex_activate_module:n {o,x}

```

(End of definition for `\stex_activate_module:n`. This function is documented on page 122.)

Iterating:

```
2673 <@=stex_iterate>
```

```

\stex_iterate_symbols:n
2674 \cs_new_protected:Nn \stex_iterate_symbols:n {
2675   \stex_pseudogroup_with:nn{_\stex_iterate_sym_cs:nnnnnnnnN}\stex_iterate_break:\stex_iterat
2676   \cs_set:Npn \_\stex_iterate_sym_cs:nnnnnnnnN
2677   ##1 ##2 ##3 ##4 ##5 ##6 ##7 ##8 ##9 { #1 }
2678   \cs_set:Npn \stex_iterate_break: {
2679     \prop_map_break:n{\seq_map_break:}
2680   }
2681   \cs_set:Npn \stex_iterate_break:n ##1 {
2682     \prop_map_break:n{\seq_map_break:n{##1}}
2683   }
2684   \seq_map_inline:Nn \l_stex_all_modules_seq {
2685     \prop_map_inline:cn{c_stex_module_#1_symbols_prop}{%
2686       \_\stex_iterate_sym_cs:nnnnnnnnN {##1} ####2
2687     }
2688   }
2689 }
2690 }

```

(End of definition for `\stex_iterate_symbols:n`. This function is documented on page 123.)

`\stex_iterate_symbols:nn`

```

2691 \cs_new_protected:Nn \stex_iterate_symbols:nn {
2692   \seq_clear:N \l__stex_iterate_mods_seq
2693   \stex_pseudogroup_with:nn{_\stex_iterate_sym_cs:nnnnnnnnN}{%
2694     \cs_set:Npn \_\stex_iterate_sym_cs:nnnnnnnnN
2695     ##1 ##2 ##3 ##4 ##5 ##6 ##7 ##8 ##9 { #2 }
2696     \clist_map_function:nN {#1} \_\stex_iterate_it_decl_i:n
2697   }
2698 }
2699
2700 \cs_new_protected:Nn \_\stex_iterate_it_decl_i:n {
2701   \seq_if_in:NnF \l__stex_iterate_mods_seq {#1} {
2702     \seq_put_left:Nn \l__stex_iterate_mods_seq {#1}
2703     \prop_map_inline:cn{c_stex_module_#1_morphisms_prop}{%
2704       \_\stex_iterate_it_decl_check:nnnn ##2
2705     }
2706     \prop_map_inline:cn{c_stex_module_#1_symbols_prop}{%
2707       \_\stex_iterate_sym_cs:nnnnnnnnN {##1} ##2
2708     }
2709   }
2710 }
2711 \cs_new_protected:Nn \_\stex_iterate_it_decl_check:nnnn {
2712   \tl_if_empty:nT{#1}{%
2713     \_\stex_iterate_it_decl_i:n {#2}
2714   }
2715 }

```

(End of definition for `\stex_iterate_symbols:nn`. This function is documented on page 123.)

`\stex_iterate_notations:nn`

```

2716 \cs_new_protected:Nn \stex_iterate_notations:nn {
2717   \seq_clear:N \l__stex_iterate_mods_seq
2718   \stex_pseudogroup_with:nn{_\stex_iterate_not_cs:nnnn}{%

```

```

2719   \cs_set:Npn \__stex_iterate_not_cs:nnnnn
2720   ##1 ##2 ##3 ##4 ##5 { #2 }
2721   \clist_map_function:nN {#1} \__stex_iterate_it_not_i:n
2722 }
2723 }
2724
2725 \cs_new_protected:Nn \__stex_iterate_it_not_i:n {
2726   \seq_if_in:NnF \l__stex_iterate_mods_seq {#1} {
2727     \seq_put_left:Nn \l__stex_iterate_mods_seq {#1}
2728     \prop_map_inline:cn{c_stex_module_#1_notations_prop} {
2729       \__stex_iterate_not_cs:nnnnn ##2
2730     }
2731     \prop_map_inline:cn{c_stex_module_#1_morphisms_prop} {
2732       \__stex_iterate_it_not_check:nnnn ##2
2733     }
2734   }
2735 }
2736 \cs_new_protected:Nn \__stex_iterate_it_not_check:nnnn {
2737   \tl_if_empty:nT{#1}{%
2738     \__stex_iterate_it_not_i:n {#2}
2739   }
2740 }
```

(End of definition for `\stex_iterate_notations:nn`. This function is documented on page [126](#).)

### `\stex_iterate_morphisms:nn`

```

2741 \cs_new_protected:Nn \stex_iterate_morphisms:nn {
2742   \seq_clear:N \l__stex_iterate_mods_seq
2743   \bool_set_true:N \l__stex_iterate_continue_bool
2744   \cs_set:Npn \__stex_iterate_morphism_cs:nnnn ##1 ##2 ##3 ##4 ##5 {
2745     #2
2746     \bool_if:NT \l__stex_iterate_continue_bool {
2747       \str_if_eq:nnTF{##1}{##2} {
2748         \tl_put_right:Nn \l__stex_iterate_todo_tl {
2749           \__stex_iterate_morphism:nn{##5}{##2}
2750         }
2751       }{
2752         \tl_put_right:Nn \l__stex_iterate_todo_tl {
2753           \__stex_iterate_morphism:nn{##5 / ##1}{##2}
2754         }
2755       }
2756     }
2757   }
2758   \cs_set:Npn \stex_iterate_break:n ##1 {
2759     \bool_set_false:N \l__stex_iterate_continue_bool
2760     \prop_map_break:n{##1}
2761   }
2762   \__stex_iterate_morphism:nn{}{#1}
2763 }
2764
2765 \cs_new_protected:Nn \__stex_iterate_iterate_morphism:nn {
2766   \tl_clear:N \l__stex_iterate_todo_tl
2767   \seq_if_in:NnF \l__stex_iterate_mods_seq {#1 #2} {
2768     \seq_put_right:Nn \l__stex_iterate_mods_seq {#1 #2}
```

```

2769   \prop_map_inline:cn{c_stex_module_#2_morphisms_prop}{
2770     \__stex_iterate_morphism_cs:nnnn ##2 {#1}
2771     % TODO
2772     % ##1: name or [mpath]
2773     % ##2 = {####1}{####2}{####3}{####4}
2774     % ####1 = name
2775     % ####2 = mpath
2776     % ####3 = type
2777     % ####4 = {origname}{newname}*
2778   }
2779   \bool_if:NT \l__stex_iterate_continue_bool \l__stex_iterate_todo_tl
2780 }
2781 }
```

(End of definition for `\stex_iterate_morphisms:nn`. This function is documented on page ??.)

### 13.6.2 Structural Features

```
2782 <@=stex_features>
```

```

\stex_structural_feature_module:nn
\stex_structural_feature_module_end:
2783 \cs_new_protected:Nn \stex_structural_feature_module:nn {
2784   \stex_if_do_html:TF {
2785     \exp_args:Nne \begin{stex_annotation_env} {
2786       shtml:feature-#2={
2787         \l_stex_current_module_str/#1}
2788       \str_if_empty:NF \l_stex_macroname_str {
2789         shtml:macroname={\l_stex_macroname_str}
2790       }
2791     }
2792     \stex_annotation_invisible:n{}
2793   }\group_begin:
2794   \stex_module_setup:n {#1-module}
2795 }
2796
2797 \cs_new_protected:Nn \stex_structural_feature_module_end: {
2798   \tl_gset_eq:NN \g_stex_last_feature_str \l_stex_current_module_str
2799   \stex_close_module:
2800   \stex_if_do_html:TF{
2801     \end{stex_annotation_env}
2802   }\group_end:
2803 }
```

(End of definition for `\stex_structural_feature_module:nn` and `\stex_structural_feature_module_end:`. These functions are documented on page 128.)

```

\stex_structural_feature_morphism:nnn
\stex_structural_feature_morphism_end:
2804 \bool_new:N \l__stex_features_implicit_bool
2805 \cs_new_protected:Nn \stex_structural_feature_morphism:nnnnn {
2806   \str_clear:N \l_stex_current_domain_str
2807   \tl_if_empty:nT{#3}{
2808     \stex_get_mathstructure:n{#4}
2809     \str_set_eq:NN \l_stex_current_domain_str \l_stex_get_structure_module_str
2810   }
2811   \str_if_empty:NT \l_stex_current_domain_str {
```

```

2812   \stex_import_module_uri:nn { #3 }{ #4 }
2813   \group_begin:
2814   \stex_import_require_module:ooo
2815     \l_stex_import_archive_str
2816     \l_stex_import_path_str
2817     \l_stex_import_name_str
2818   \exp_args:Nnx \use:nn \group_end: {
2819     \str_set:Nn \exp_not:N\l_stex_current_domain_str {\l_stex_import_ns_str}
2820   }
2821 }
2822 \tl_if_empty:nTF{#1}{
2823   \bool_set_true:N \l__stex_features_implicit_bool
2824   \str_set:Nx \l_tmpa_str {[ \l_stex_current_domain_str]}
2825 }{
2826   \bool_set_false:N \l__stex_features_implicit_bool
2827   \str_set:Nn \l_tmpa_str {#1}
2828 }
2829
2830 \stex_if_do_html:TF {
2831   \begin{stex_annotation_env} {
2832     \shtml:feature-#2={\l_stex_current_module_str?\l_tmpa_str},
2833     \shtml:domain={\l_stex_current_domain_str}
2834     #5
2835   }
2836   \stex_annotation_invisible:n{}
2837 }\group_begin:
2838 \str_set:Nn \l__stex_features_feature_str {#2}
2839 \str_set_eq:NN \l_stex_feature_name_str \l_tmpa_str
2840 \__stex_features_setup:
2841 \__stex_features_reactivate:
2842 \%^A\stex_activate_module:o \l_stex_current_domain_str
2843 \exp_args:Ne \stex_metagroup_new:n {\l_stex_current_module_str / \l_stex_feature_name_str}
2844 }
2845
2846 \cs_new_protected:Nn \__stex_features_do_for_list: {
2847   \seq_clear:N \l_stex_fors_seq
2848   \clist_map_inline:Nn \l_stex_key_for_clist {
2849     \exp_args:Ne\stex_get_in_morphism:n{\tl_to_str:n{##1}}
2850     \seq_put_right:Nx \l_stex_fors_seq
2851       {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
2852   }
2853 }
2854
2855 \cs_new_protected:Nn \__stex_features_add_definiens:nn {
2856   \__stex_features_set_definiens_macros: #1\__stex_features_break:
2857   \stex_assign_do:n{#2}
2858   #2
2859 }
2860 \cs_new_protected:Npn \__stex_features_set_definiens_macros: #1?#2?#3\__stex_features_break:
2861   \str_set:Nn \l_stex_get_symbol_mod_str {#1?#2}
2862   \str_set:Nn \l_stex_get_symbol_name_str {#3}
2863 \exp_args:Nne\use:nn{\__stex_features_set_definiens_macros_i:nnnnnnn}{%
2864   \prop_item:Nn \l_stex_morphism_symbols_prop {[#1?#2]/[#3]}}
2865 }

```

```

2866 }
2867 \cs_new_protected:Nn \__stex_features_set_definiens_macros_i:nnnnnnn {
2868   \tl_set:Nn \l_stex_get_symbol_def_tl{#4}
2869 }
2870
2871 \cs_new_protected:Nn \stex_structural_feature_morphism_end: {
2872   \str_gset_eq:NN \l_stex_feature_name_str \l_stex_feature_name_str
2873   \str_gset_eq:NN \l_stex_current_domain_str \l_stex_current_domain_str
2874   \seq_gset_eq:NN \l_stex_morphism_symbols_prop \l_stex_morphism_symbols_prop
2875   \seq_gset_eq:NN \l_stex_morphism_renames_prop \l_stex_morphism_renames_prop
2876   \seq_gset_eq:NN \l_stex_morphism_morphisms_seq \l_stex_morphism_morphisms_seq
2877
2878   \bool_if:NT \l__stex_features_implicit_bool {
2879     \stex_execute_in_module:x{
2880       \stex_activate_module:n{\l_stex_current_domain_str}
2881     }
2882   }
2883
2884 \__stex_features_do_elaboration:
2885 \stex_if_do_html:TF{
2886   \end{stex_annotation_env}
2887 }\group_end:
2888 }
2889
2890 \cs_new_protected:Nn \__stex_features_setup: {
2891   \prop_clear:N \l_stex_morphism_symbols_prop
2892   \prop_clear:N \l_stex_morphism_renames_prop
2893   \seq_clear:N \l_stex_morphism_morphisms_seq
2894   \__stex_features_do_decls:
2895   \exp_args:No \__stex_features_do_morphisms:n \l_stex_current_domain_str
2896 }
2897
2898 \cs_new_protected:Nn \__stex_features_rename_all: {
2899 }
2900
2901 \cs_new:Npn \__stex_features_clean:nnw [#1] / #2 \stex_end: {
2902   [#1]/[#2]
2903 }
2904
2905
2906 \cs_new_protected:Nn \__stex_features_do_decls: {
2907   \exp_args:No \stex_iterate_symbols:nn \l_stex_current_domain_str {
2908     \stex_str_if_starts_with:nnTF{##3}[
2909       \exp_args:NNe \prop_put:Nnn \l_stex_morphism_symbols_prop {
2910         \__stex_features_clean:nnw ##3 \stex_end:
2911       }
2912     }{
2913       \prop_put:Nnn \l_stex_morphism_symbols_prop
2914       {[##1]/[##3]}
2915     }{
2916       {[##2}{##4}{##5}{##6}{##7}{##8}##9
2917     }
2918   }
2919 }

```

```

2920 \cs_new_protected:Nn \stex_structural_feature_morphism_check_total: {
2921   \%prop_map_inline:Nn \l_stex_morphism_symbols_prop {
2922     \%__stex_features_total_check: ##1 ##2
2923   }
2924 }
2925 }

2926 \cs_new_protected:Npn \__stex_features_total_check: [#1]/[#2] #3 #4 #5 #6 #7 #8 #9 {
2927   \tl_if_empty:nT{#6} {
2928     \msg_error:nnxx{\stex}{error/needsdefiniens}{#1?#2}{total-morphism}
2929   }
2930 }
2931 }

2932 \cs_new:Npn \__stex_features_split_qm:w #1 ? #2 ? #3 { #3 }
2933 \cs_new_protected:Nn \__stex_features_do_elaboration: {
2934   \stex_debug:nn{morphisms} {
2935     Elaborating:^^J\prop_to_keyval:N \l_stex_morphism_symbols_prop
2936     ^^J
2937     Renamings:^^J
2938     \prop_to_keyval:N \l_stex_morphism_renames_prop
2939   }
2940   \prop_map_inline:Nn \l_stex_morphism_symbols_prop {
2941     \%__stex_features_elab_check: ##1 ##2
2942   }
2943 }

2944 \exp_args:No\stex_iterate_notations:nn\l_stex_current_domain_str{
2945   \prop_get:NnNTF \l_stex_morphism_renames_prop {##1}\l_stex_features_tmp {
2946     \exp_args:Ne \stex_module_add_notation:nnnnn
2947     {\l_stex_current_module_str ? \exp_after:wN \use_i:nn \l__stex_features_tmp}
2948     {##2}{##3}{##4}{##5}
2949   }{
2950     \bool_if:NF \l_stex_features_implicit_bool {
2951       \exp_args:Ne \stex_module_add_notation:nnnnn
2952       {\l_stex_current_module_str ? \l_stex_feature_name_str
2953        / \__stex_features_split_qm:w ##1}{##2}{##3}{##4}{##5}
2954     }
2955   }
2956 }

2957 \stex_module_add_morphism:ooox
2958   \l_stex_feature_name_str
2959   \l_stex_current_domain_str
2960   \l__stex_features_feature_str
2961   {\prop_map_function:NN \l_stex_morphism_renames_prop \__stex_features_rename:nn}
2962 }

2963 \cs_new:Nn \__stex_features_rename:nn{
2964   {#1}{\use_i:nn#2}
2965 }
2966 }

2967 \cs_set_protected:Nn \__stex_features_sym_i_gobble:nnnnnn {}

2968 \cs_new_protected:Npn \__stex_features_elab_check: [#1]/[#2] #3 {
2969   \prop_get:NnNTF \l_stex_morphism_renames_prop {#1?#2}\l_stex_features_tmp {
2970     \stex_debug:nnf{morphisms}{Generating~\l__stex_features_tmp}
2971   }
2972 }
```

```

2974     \exp_after:wN \stex_module_add_symbol:nnnnnnN \l_stex_features_tmp
2975 }{
2976     \bool_if:NTF \l_stex_features_implicit_bool {
2977         \__stex_features_sym_i_gobble:nnnnn
2978         \%stex_debug:nn{morphisms}{Generating~#3:\l_stex_feature_name_str / #2}
2979         \%exp_args:Nno \stex_module_add_symbol:nnnnnnN {#3}{\l_stex_feature_name_str / #2}
2980     }{
2981         \stex_debug:nn{morphisms}{Generating~\l_stex_feature_name_str / #2}
2982         \exp_args:Nno \stex_module_add_symbol:nnnnnnN {}{\l_stex_feature_name_str / #2}
2983     }
2984 }
2985 }
2986
2987 \cs_new_protected:Nn \__stex_features_do_morphisms:n {
2988     \prop_map_inline:cn {c_stex_module_#1_morphisms_prop} {
2989         \__stex_features_do_morph:nnnn ##2
2990     }
2991 }
2992
2993 \cs_new_protected:Nn \__stex_features_do_morph:nnnn {
2994     \tl_if_empty:nF{#3} {
2995         \seq_put_right:Nn \l_stex_morphism_morphisms_seq {{#1}{#2}{#3}}
2996     }
2997     \__stex_features_do_morphisms:n{#2}
2998 }
2999
3000 \cs_new_protected:Npn \__stex_features_reactivate: {
3001     \stex_deactivate_macro:Nn \symdecl {module-environments}
3002     \stex_deactivate_macro:Nn \textsymdecl {module-environments}
3003     \stex_deactivate_macro:Nn \symdef {module-environments}
3004     \stex_deactivate_macro:Nn \notation {module-environments}
3005     \stex_deactivate_macro:Nn \importmodule {module-environments}
3006     \stex_deactivate_macro:Nn \requiremodule {module-environments}
3007     \stex_deactivate_macro:Nn \smodule {outside-of-morphisms}
3008     \stex_reactivate_macro:N \assign
3009     \stex_reactivate_macro:N \assignMorphism
3010     \stex_reactivate_macro:N \renamedecl
3011     \cs_set_eq:NN \stex_do_for_list: \__stex_features_do_for_list:
3012     \cs_set_eq:NN \stex_add_definiens:nn \__stex_features_add_definiens:nn
3013 }

```

(End of definition for \stex\_structural\_feature\_morphism:nnn and \stex\_structural\_feature\_morphism\_end:. These functions are documented on page ??.)

### \stex\_get\_in\_morphism:n

```

3014 \cs_new_protected:Nn \stex_get_in_morphism:n {
3015     \str_clear:N \l_stex_get_symbol_name_str
3016     \prop_map_inline:Nn \l_stex_morphism_symbols_prop {
3017         \exp_args:Nx\__stex_features_get_check:nnnn{\tl_to_str:n{#1}}##1##2
3018     }
3019     \str_if_empty:NT \l_stex_get_symbol_name_str {
3020         \prop_map_inline:Nn \l_stex_morphism_renames_prop {
3021             \__stex_features_renamed_check:nnnn{#1}##1=##2
3022         }

```

```

3023   \str_if_empty:NNT \l_stex_get_symbol_name_str {
3024     \msg_error:n{stex}{error/unknownsymbolin}{#1}{
3025       morphism~\l_stex_feature_name_str
3026     }
3027   }
3028 }
3029 }
3030
3031 \cs_new_protected:Npn \__stex_features_renamed_check:nnnnn #1#2?#3?#4=#5#6 {
3032   \str_if_eq:nnTF{#1}{#5} {
3033     \exp_args:Nnx \use:nn{\__stex_features_check_break:nnnnnnnnn}{#2?#3}{#4} {
3034       \prop_item:Nn \l_stex_morphism_symbols_prop {[#2?#3]/[#4]}
3035     }
3036   } {
3037     \str_if_eq:nnT{#1}{#6} {
3038       \exp_args:Nnx \use:nn{\__stex_features_check_break:nnnnnnnnn}{#2?#3}{#4} {
3039         \prop_item:Nn \l_stex_morphism_symbols_prop {[#2?#3]/[#4]}
3040       }
3041     }
3042   }
3043 }
3044
3045 \cs_new_protected:Npn \__stex_features_get_check:nnnn #1[#2]/[#3]#4 {
3046   \str_if_eq:nnTF{#1}{#3} {
3047     \__stex_features_check_break:nnnnnnnnn{#2}{#3}{#4}
3048   } {
3049     \str_if_eq:nnTF{#1}{#4} {
3050       \__stex_features_check_break:nnnnnnnnn{#2}{#3}{#4}
3051     } {
3052       \use_none:nnnnn
3053     }
3054   }
3055 }
3056
3057 \cs_new_protected:Nn \__stex_features_check_break:nnnnnnnnn {
3058   \prop_map_break:n {
3059     \str_set:Nn \l_stex_get_symbol_mod_str{#1}
3060     \str_set:Nn \l_stex_get_symbol_name_str{#2}
3061     \str_set:Nn \l_stex_get_symbol_macro_str{#3}
3062     \int_set:Nn \l_stex_get_symbol_arity_int {#4}
3063     \tl_set:Nn \l_stex_get_symbol_args_tl {#5}
3064     \tl_set:Nn \l_stex_get_symbol_def_tl {#6}
3065     \tl_set:Nn \l_stex_get_symbol_type_tl {#7}
3066     \tl_set:Nn \l_stex_get_symbol_return_tl {#8}
3067     \tl_set:Nn \l_stex_get_symbol_invoke_cs {#9}
3068   }
3069 }

```

(End of definition for `\stex_get_in_morphism:n`. This function is documented on page 131.)

## 13.7 Inheritance

### 13.7.1 \importmodule/\usemodule

```

3070 <@@=stex_importmodule>

\usemodule

3071 \stex_new_stylable_cmd:nnn {usemodule} { 0{} m } {
3072   \stex_import_module_uri:nn { #1 }{ #2 }
3073   \stex_import_require_module:ooo
3074     \l_stex_import_archive_str
3075     \l_stex_import_path_str
3076     \l_stex_import_name_str
3077   \stex_if_do_html:T {
3078     \hbox{\stex_annotation_invisible:nn
3079       {shtml:usemodule=\l_stex_import_ns_str} {}}
3080   }
3081   \stex_if_smsmode:F{
3082     \group_begin:
3083     \tl_set_eq:NN \thismoduleuri \l_stex_import_ns_str
3084     \tl_set_eq:NN \thismodulename \l_stex_import_name_str
3085     \tl_clear:N \thisstyle
3086     \stex_style_apply:
3087     \group_end:
3088   }
3089 }{}
```

(End of definition for `\usemodule`. This function is documented on page 93.)

### \stex\_import\_module\_uri:nn

```

3090 \cs_new_protected:Nn \stex_import_module_uri:nn {
3091   \stex_debug:nn{importmodule}{URI:~>#1<~>#2<~>}
3092   \exp_args:NNnx \seq_set_split:Nnn \__stex_importmodule_seq ? { \tl_to_str:n{ #2 } }
3093   \seq_pop_right:NN \__stex_importmodule_seq \l_stex_import_name_str
3094   \str_set:Nx \l_stex_import_path_str { \seq_use:Nn \__stex_importmodule_seq ? }
3095   \tl_if_empty:nTF { #1 } {
3096     \stex_debug:nn{importmodule}{No~archive}
3097     \prop_if_exist:NTF \l_stex_current_archive_prop {
3098       \stex_debug:nn{importmodule}{Picking~current~archive}
3099       \str_set:Nx \l_stex_import_archive_str {
3100         \prop_item:Nn \l_stex_current_archive_prop { id }
3101       }
3102     }{
3103       \str_clear:N \l_stex_import_archive_str
3104       \str_set:Nn \l_stex_import_uri_str {file:}
3105       \str_if_empty:NTF \l_stex_import_path_str {
3106         \stex_debug:nn{importmodule}{Empty~Path}
3107         \stex_file_split_off_ext:NN \l__stex_importmodule_path_seq \g_stex_current_file
3108         \stex_file_split_off_lang:NN \l__stex_importmodule_path_seq \l__stex_importmodule_pa
3109         \str_set:Nx \l_stex_import_path_str {
3110           \stex_file_use:N \l__stex_importmodule_path_seq
3111         }
3112     }{
3113       \stex_debug:nn{importmodule}{Resolving~path~\l_stex_import_path_str~relative~to~\ste
3114       \stex_file_resolve:Nx \l__stex_importmodule_seq { \stex_file_use:N \g_stex_current_f
3115       \str_set:Nx \l_stex_import_path_str {
3116         \stex_file_use:N \l__stex_importmodule_seq
3117       }
3118       \stex_debug:nn{importmodule}{...yields~\l_stex_import_path_str}
```

```

3119     }
3120   }
3121 } {
3122   \stex_debug:nn{importmodule}{Archive~#1}
3123   \str_set:Nx \l_stex_import_archive_str { #1 }
3124   \stex_require_archive:o \l_stex_import_archive_str
3125   \str_set:Nx \l_stex_import_uri_str { \prop_item:cnd{ c_stex_mathhub_ } \l_stex_import_archive_str }
3126 }
3127 }
```

(End of definition for `\stex_import_module_uri:nn`. This function is documented on page 130.)

```

\stex_import_require_module:nnn
\stex_import_require_module:ooo
3128 \cs_new_protected:Npn \stex_import_require_module:nnn #1 {
3129   \tl_if_empty:nTF { #1 } {
3130     \str_clear:N \l__stex_importmodule_archive_str
3131     \str_set:Nn \l_stex_import_uri_str {file:{}}
3132     \__stex_importmodule_get_module:nnn {}
3133   }{
3134     \stex_require_archive:n { #1 }
3135     \str_set:Nx \l__stex_importmodule_archive_str {#1}
3136     \str_set:Nx \l_stex_import_uri_str { \prop_item:cnd{ c_stex_mathhub_ } #1 _manifest_prop}
3137     \str_set:Nx \l__stex_importmodule_str { \prop_item:cnd{ c_stex_mathhub_ } #1 _manifest_prop}
3138     \exp_args:No \__stex_importmodule_get_module:nnn \l__stex_importmodule_str
3139   }
3140 }
3141 \cs_generate_variant:Nn \stex_import_require_module:nnn {ooo}
3142
3143 \cs_new_protected:Npn \stex_import_require_module_safe:nnn #1 {
3144   \tl_if_empty:nTF { #1 } {
3145     \str_clear:N \l__stex_importmodule_archive_str
3146     \str_set:Nn \l_stex_import_uri_str {file:{}}
3147     \__stex_importmodule_get_module_safe:nnn {}
3148   }{
3149     \stex_require_archive:n { #1 }
3150     \str_set:Nx \l__stex_importmodule_archive_str {#1}
3151     \str_set:Nx \l_stex_import_uri_str { \prop_item:cnd{ c_stex_mathhub_ } #1 _manifest_prop}
3152     \str_set:Nx \l__stex_importmodule_str { \prop_item:cnd{ c_stex_mathhub_ } #1 _manifest_prop}
3153     \exp_args:No \__stex_importmodule_get_module_safe:nnn \l__stex_importmodule_str
3154   }
3155 }
3156
3157 \cs_new_protected:Nn \__stex_importmodule_get_module_uri:nnn {
3158   \tl_if_empty:nF {#2} {
3159     \str_set:Nx \l_stex_import_uri_str { \l_stex_import_uri_str / #2 }
3160   }
3161   \stex_debug:nn{importmodule}{~>#1<^J>#2<^J>#3<^J>\l_stex_import_uri_str<^J}
3162   Current~file:\stex_file_use:N \g_stex_current_file^J
3163   Current~namespace:\stex_uri_use:N \l_stex_current_ns_uri
3164 }
3165 \stex_if_module_exists:nTF {\l_stex_import_uri_str?#3} {
3166   \str_set:Nx \l_stex_import_ns_str { \l_stex_import_uri_str?#3}
3167 }{
3168   \stex_if_module_exists:nTF{\stex_uri_use:N \l_stex_current_ns_uri ? #3}{
```

```

3169     \str_set:Nx \l_stex_import_ns_str {\stex_uri_use:N \l_stex_current_ns_uri ? #3}
3170     }{
3171         \__stex_importmodule_get_from_file:nnn{#1}{#2}{#3}
3172         \str_set:Nx \l_stex_import_ns_str {\l_stex_import_uri_str?#3}
3173     }
3174 }
3175 }
3176 \cs_new_protected:Nn \__stex_importmodule_get_module_uri_safe:nnn {
3177     \tl_if_empty:nF {#2}){
3178         \str_set:Nx \l_stex_import_uri_str {\l_stex_import_uri_str / #2}
3179     }
3180 \stex_debug:nn{importmodule}{~>#1<^^J>#2<^^J>#3<^^J>\l_stex_import_uri_str<^^J
3181     Current~file:\stex_file_use:N \g_stex_current_file~^J
3182     Current~namespace:\stex_uri_use:N \l_stex_current_ns_uri
3183 }
3184 \stex_if_module_exists:nTF {\l_stex_import_uri_str?#3} {
3185     \str_set:Nx \l_stex_import_ns_str {\l_stex_import_uri_str?#3}
3186 }{
3187     \stex_if_module_exists:nTF{\stex_uri_use:N \l_stex_current_ns_uri ? #3} {
3188         \str_set:Nx \l_stex_import_ns_str {\stex_uri_use:N \l_stex_current_ns_uri ? #3}
3189     }{
3190         \__stex_importmodule_get_from_file_safe:nnn{#1}{#2}{#3}
3191         \str_set:Nx \l_stex_import_ns_str {\l_stex_import_uri_str?#3}
3192     }
3193 }
3194 }
3195
3196 \cs_new_protected:Nn \__stex_importmodule_get_module:nnn {
3197     \stex_debug:nn{importmodule}{Requiring~>[#1]#2?#3<}
3198     \__stex_importmodule_get_module_uri:nnn{#1}{#2}{#3}
3199     \stex_activate_module:o \l_stex_import_ns_str
3200 }
3201
3202 \cs_new_protected:Nn \__stex_importmodule_get_module_safe:nnn {
3203     \stex_debug:nn{importmodule}{Requiring~>[#1]#2?#3<}
3204     \__stex_importmodule_get_module_uri_safe:nnn{#1}{#2}{#3}
3205 }
3206
3207 \cs_new_protected:Nn \__stex_importmodule_get_from_file:nnn {
3208     \stex_file_resolve:Nx \l_stex_importmodule_seq { \tl_if_empty:nF{ #1 }{ #1 / } #2 }
3209     \str_set:Nx \l_stex_importmodule_str {\stex_file_use:N \l_stex_importmodule_seq}
3210     \stex_debug:nn{imports}{Looking~for~\l_stex_import_uri_str?#3...}
3211     \__stex_importmodule_check_file:nn{ /#3.tex }{
3212         \__stex_importmodule_check_file:nn{/#3.\l_stex_current_language_str .tex}{
3213             \__stex_importmodule_check_file:nn{/#3.en.tex}{
3214                 \__stex_importmodule_check_file:nn{.tex}{
3215                     \__stex_importmodule_check_file:nn{.\l_stex_current_language_str .tex}{
3216                         \__stex_importmodule_check_file:nn{.en.tex}{
3217                             \msg_error:nnx{stex}{error/unknownmodule}{\l_stex_import_uri_str?#3}
3218                         }
3219                     }
3220                 }
3221             }
3222         }

```

```

3223 }
3224 \stex_if_smsmode:TF{
3225   \exp_args:NNo \exp_args:Nnx \str_if_eq:nnTF{\l__stex_importmodule_str}{\stex_file_use:N
3226     \stex_debug:nn{imports}{Skipping-current-file}
3227   }{
3228     \__stex_importmodule_load_file:n{#3}
3229   }
3230   }{
3231     \__stex_importmodule_load_file:n{#3}
3232   }
3233 }
3234 \cs_new_protected:Nn \__stex_importmodule_get_from_file_safe:nnn {
3235   \stex_file_resolve:Nx \l__stex_importmodule_seq { \tl_if_empty:nF{ #1 }{ #1 / } #2 }
3236   \str_set:Nx \l__stex_importmodule_str {\stex_file_use:N \l__stex_importmodule_seq}
3237   \stex_debug:nn{imports}{Looking-for-\l_stex_import_uri_str?#3...}
3238   \__stex_importmodule_check_file:nn{ /#3.tex }{
3239     \__stex_importmodule_check_file:nn{/#3.\l_stex_current_language_str .tex}{
3240       \__stex_importmodule_check_file:nn{/#3.en.tex}{
3241         \__stex_importmodule_check_file:nn{.tex}{
3242           \__stex_importmodule_check_file:nn{.\l_stex_current_language_str .tex}{
3243             \__stex_importmodule_check_file:nn{.en.tex}{
3244               }
3245             }
3246           }
3247         }
3248       }
3249     }
3250   \stex_if_smsmode:TF{
3251     \exp_args:NNo \exp_args:Nnx \str_if_eq:nnTF{\l__stex_importmodule_str}{\stex_file_use:N
3252       \stex_debug:nn{imports}{Skipping-current-file}
3253     }{
3254       \IfFileExists{ \l__stex_importmodule_str }{
3255         \__stex_importmodule_load_file:n{#3}
3256       }{}
3257     }
3258   }{
3259     \IfFileExists{ \l__stex_importmodule_str }{
3260       \__stex_importmodule_load_file:n{#3}
3261     }{}
3262   }
3263 }
3264
3265 \cs_new_protected:Nn \__stex_importmodule_load_file:n {
3266   \stex_file_in_smsmode:on \l__stex_importmodule_str {
3267     \str_if_empty:NF \l__stex_importmodule_archive_str {
3268       \stex_set_current_archive:n \l__stex_importmodule_archive_str
3269     }
3270     \stex_debug:nn{modules}{Loading-\l__stex_importmodule_str}
3271   }
3272   \stex_if_module_exists:nF {\l_stex_import_uri_str?#1} {
3273     \msg_error:nnx{stex}{error/unknownmodule}{\l_stex_import_uri_str?#1}
3274   }
3275 }
3276

```

```

3277 \cs_new_protected:Npn \__stex_importmodule_check_file:nn #1 {
3278   \stex_debug:nn{imports}{Checking~\l__stex_importmodule_str #1}
3279   \IfFileExists{ \l__stex_importmodule_str #1 }{
3280     \stex_debug:nn{imports}{Success}
3281     \str_set:Nx \l__stex_importmodule_str { \l__stex_importmodule_str #1 }
3282   }
3283 }

```

(End of definition for `\stex_import_require_module:nnn`. This function is documented on page 130.)

### \importmodule

```

3284 \stex_new_stylable_cmd:nnnn{importmodule} { 0{} m } {
3285   \__stex_importmodule_import_module:nn {#1}{#2}
3286   \stex_smsmode_do:
3287 }
3288 \stex_deactivate_macro:Nn \importmodule {module-environments}
3289
3290 \cs_new_protected:Nn \__stex_importmodule_import_module:nn {
3291   \stex_import_module_uri:nn { #1 }{ #2 }
3292   \stex_import_require_module:ooo
3293     \l_stex_import_archive_str
3294     \l_stex_import_path_str
3295     \l_stex_import_name_str
3296   \stex_execute_in_module:xf
3297     \stex_activate_module:n{\l_stex_import_ns_str}
3298   }
3299   \stex_module_add_morphism:nonn
3300   {}{\l_stex_import_ns_str}{import}{}
3301 \stex_if_do_html:T {
3302   \stex_annotation_invisible:nn
3303   {shtml:import=\l_stex_import_ns_str} {}
3304 }
3305 \stex_if_smsmode:F{
3306   \group_begin:
3307   \tl_set:Nn \thisarchive {#1}
3308   \tl_set_eq:NN \thismoduleuri \l_stex_import_ns_str
3309   \tl_set_eq:NN \thismodulename \l_stex_import_name_str
3310   \tl_clear:N \thisstyle
3311   \stex_style_apply:
3312   \group_end:
3313 }
3314 }
3315
3316 \cs_new_protected:Nn \__stex_importmodule_import_module_presms:nn {
3317   \stex_import_module_uri:nn { #1 }{ #2 }
3318   \tl_gput_right:Nx \g_stex_sms_import_code {
3319     \stex_import_require_module_safe:nnn
3320       {\l_stex_import_archive_str}
3321       {\l_stex_import_path_str}
3322       {\l_stex_import_name_str}
3323   }
3324 }
3325
3326 \stex_sms_allow_escape:N \importmodule

```

```

3327 \stex_every_module:n {\stex_reactivate_macro:N \importmodule}
3328 \stex_sms_allow_import:Nn \importmodule {
3329   \stex_reactivate_macro:N \importmodule
3330   \let \__stex_importmodule_import_module:nn \__stex_importmodule_import_module_presms:nn
3331 }
3332
3333 \stex_new_stylable_cmd:nnnn{requiremodule} { 0{} m } {
3334   \stex_import_module_uri:nn { #1 }{ #2 }
3335   \stex_import_require_module:ooo
3336     \l_stex_import_archive_str
3337     \l_stex_import_path_str
3338     \l_stex_import_name_str
3339   \stex_do_up_to_module:x{
3340     \stex_activate_module:n{\l_stex_import_ns_str}
3341   }
3342   \stex_if_do_html:T {
3343     \stex_annotation_invisible:nn
3344       {shtml:import=\l_stex_import_ns_str} {}
3345   }
3346   \stex_if_smemode:F{
3347     \group_begin:
3348     \tl_set:Nn \thisarchive {#1}
3349     \tl_set_eq:NN \thismoduleuri \l_stex_import_ns_str
3350     \tl_set_eq:NN \thismodulename \l_stex_import_name_str
3351     \tl_clear:N \thisstyle
3352     \stex_style_apply:
3353     \group_end:
3354   }
3355   \stex_smemode_do:
3356 }{}
3357 \stex_deactivate_macro:Nn \requiremodule {module-environments}

```

(End of definition for `\importmodule`. This function is documented on page 93.)

### 13.7.2 Theory Morphisms

```

3358 <@=stex_morphisms>
3359 \stex_new_stylable_cmd:nnnn {assign} { m m }{
3360   \stex_get_in_morphism:n{#1}
3361   \stex_assign_do:n{#2}
3362   \stex_smemode_do:
3363 }{}
3364 \stex_sms_allow_escape:N\assign
3365
3366 \cs_new_protected:Nn \stex_assign_do:n{
3367   \stex_debug:nn{assign}{Assigning~\l_stex_get_symbol_name_str~to~\tl_to_str:n{#1}}
3368   \tl_if_empty:NF \l_stex_get_symbol_def_tl {
3369     \%msg_error:nnxx{stex}{error/symbolalreadydefined}{\l_stex_get_symbol_name_str}{
3370       % morphism~\l_stex_feature_name_str
3371       %}
3372   }
3373   \stex_check_term:n{#1}
3374   \stex_debug:nn{HERE!}{
3375     \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str^~J

```

```

3376     \tl_to_str:n{#1}
3377 }
3378 \stex_if_do_html:T{
3379     \stex_annotation_invisible:nn{shtml:assign={\l_stex_get_symbol_mod_str?\l_stex_get_symbol_}
3380         \stex_annotation_force_break:n{
3381             \mode_if_math:T\hbox{$\stex_annotation:nn{shtml:definiens={}{}{#1}}$}
3382         }
3383     }
3384 }
3385 \exp_args:Ne \stex_metagroup_do_in:nx{
3386     \l_stex_current_module_str / \l_stex_feature_name_str
3387 }{
3388     \prop_put:Nnn \exp_not:N \l_stex_morphism_symbols_prop
3389     {[ \l_stex_get_symbol_mod_str] / [ \l_stex_get_symbol_name_str]}
3390     {
3391         {\l_stex_get_symbol_macro_str}
3392         {\int_use:N \l_stex_get_symbol_arity_int}
3393         {\l_stex_get_symbol_args_t1}
3394         {\exp_not:n{#1}}
3395         {\exp_args:No\exp_not:n\l_stex_get_symbol_type_t1}
3396         {\exp_args:No\exp_not:n\l_stex_get_symbol_return_t1}
3397         {\l_stex_get_symbol_invoke_cs}
3398     }
3399 }
3400 }
3401
3402
3403 \stex_new_stylable_cmd:nnnn {renamedecl} { m 0{} m }{
3404     \stex_get_in_morphism:n{#1}
3405     \stex_renamedecl_do:nn{#2}{#3}
3406     \stex_smsmode_do:
3407 }{}
3408 \stex_sms_allow_escape:N\renamedecl
3409
3410 \cs_new_protected:Nn \stex_renamedecl_do:nn {
3411     \stex_debug:nn{renamedecl}{Renaming~\l_stex_get_symbol_name_str~to~[#1]{#2}}
3412     \stex_if_do_html:T{
3413         \exp_args:Ne \stex_annotation_invisible:nn{
3414             shtml:rename={\l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str},
3415             shtml:macroname={#2}
3416             \str_if_empty:nF{#1}{ ,shtml:to={#1} }
3417     }{}
3418 }
3419 \exp_args:Ne \stex_metagroup_do_in:nx{
3420     \l_stex_current_module_str / \l_stex_feature_name_str
3421 }{
3422     \prop_put:Nnn \exp_not:N \l_stex_morphism_renames_prop
3423     {[ \l_stex_get_symbol_mod_str? \l_stex_get_symbol_name_str]{{#2}{#1}}{
3424         \tl_if_empty:nTF{#1}{\l_stex_feature_name_str/\l_stex_get_symbol_name_str}{#1}
3425     }{#1}
3426 }
3427 }
3428
3429 \stex_new_stylable_cmd:nnnn {assignMorphism} { m m }{

```

```

3430 \str_clear:N \l__stex_morphisms_morphism_dom_str
3431 \stex_iterate_morphisms:nn\l_stex_current_domain_str{
3432   \stex_debug:nn{assignMorphism}{
3433     Checking:~#1~vs:^^J##1^^J##2^^J##3^^J##4
3434   }
3435   \str_if_eq:nnTF{#1}{##1}{
3436     \__stex_morphisms_do_morph_assign:nnn{##1}{##2}{##4}
3437   }{
3438     \stex_str_if_ends_with:nnT{##2}{#1}{
3439       \__stex_morphisms_do_morph_assign:nnn{##1}{##2}{##4}
3440     }
3441   }
3442 }
3443 \str_if_empty:NT \l__stex_morphisms_morphism_dom_str {
3444   \msg_error:nnn{stex}{error/nomorphism}{#1}
3445 }
3446 \bool_set_false:N \l_tmpa_bool
3447 \stex_iterate_morphisms:nn \l_stex_current_module_str {
3448   \stex_debug:nn{assignMorphism}{
3449     Checking:~#2~vs:^^J##1^^J##2^^J##3^^J##4
3450   }
3451   \str_if_eq:nnTF{#2}{##1}{
3452     \stex_debug:nn{assignMorphism}{match!}
3453     \stex_iterate_break:n{
3454       \stex_annotation_invisible:nn{
3455         shtml:assignMorphismFrom={\l__stex_morphisms_morphism_dom_str}
3456         ahtml:assignMorphismTo={\l_stex_current_module_str?##1}
3457       }{}
3458       \bool_set_true:N \l_tmpa_bool
3459     }
3460   }{
3461     \stex_str_if_ends_with:nnT{##2}{#2}{
3462       \stex_debug:nn{assignMorphism}{match!}
3463       \stex_iterate_break:n{
3464         \stex_annotation_invisible:nn{
3465           shtml:assignMorphismFrom={\l__stex_morphisms_morphism_dom_str},
3466           shtml:assignMorphismTo={\l_stex_current_module_str?##1}
3467         }{}
3468         \bool_set_true:N \l_tmpa_bool
3469       }
3470     }
3471   }
3472 }
3473 \bool_if:NF \l_tmpa_bool {
3474   \msg_error:nnn{stex}{error/nomorphism}{#2}
3475 }
3476 }{}}
3477 \cs_new_protected:Nn \__stex_morphisms_do_morph_assign:nnn {
3478   \stex_iterate_break:n{
3479     \str_set:Nx \l__stex_morphisms_morphism_dom_str { \l_stex_current_domain_str ? #1 }
3480     \stex_debug:nn{assignMorphism}{match!}
3481     \stex_iterate_symbols:nn{#2}{
3482       \stex_debug:nn{assignMorphism}{removing~##1?##3}
3483       % TODO: non-trivial assignments

```

```

3484     \prop_remove:Nn \l_stex_morphism_symbols_prop {
3485         [##1]/[##3]
3486     }
3487 }
3488 }
3489 }
3490
3491 \stex_deactivate_macro:Nn \assign {morphism~environments}
3492 \stex_deactivate_macro:Nn \renamedecl {morphism~environments}
3493 \stex_deactivate_macro:Nn \assignMorphism {morphism~environments}
3494 \stex_new_stylable_env:nnnnnnn {copymodule}{m 0{} m}{
3495
3496     \stex_structural_feature_morphism:nnnnn{#1}{morphism}{#2}{#3}{,shtml:total=false}
3497
3498     \stex_if_smsmode:F {
3499         \tl_set:Nn \thiscopyname { #1 }
3500         \tl_set_eq:NN \thismoduleuri \l_stex_current_domain_str
3501         \stex_style_apply:
3502     }
3503     \stex_smsmode_do:
3504 }
3505     \stex_if_smsmode:F {
3506         \stex_style_apply:
3507     }
3508     \stex_structural_feature_morphism_end:
3509 }{}{}{}
3510 \stex_deactivate_macro:Nn \copymodule {module~environments}
3511 \stex_every_module:n {
3512     \stex_reactivate_macro:N \copymodule
3513 }
3514 \stex_sms_allow_env:n{copymodule}
3515
3516 \stex_new_stylable_env:nnnnnnn {interpretmodule}{m 0{} m}{
3517     \stex_structural_feature_morphism:nnnnn{#1}{morphism}{#2}{#3}{,shtml:total=true}
3518     \stex_if_smsmode:F {
3519         \tl_set:Nn \thiscopyname { #1 }
3520         \tl_set_eq:NN \thismoduleuri \l_stex_current_domain_str
3521         \stex_style_apply:
3522     }
3523     \stex_smsmode_do:
3524 }
3525     \stex_structural_feature_morphism_check_total:
3526     \stex_if_smsmode:F {
3527         \stex_style_apply:
3528     }
3529     \stex_structural_feature_morphism_end:
3530 }{}{}{}
3531 \stex_deactivate_macro:Nn \interpretmodule {module~environments}
3532 \stex_every_module:n {
3533     \stex_reactivate_macro:N \interpretmodule
3534 }
3535 \stex_sms_allow_env:n{interpretmodule}
3536 \stex_new_stylable_env:nnnnnnn {realization}{0{} m}{
```

```

3537
3538 \stex_structural_feature_morphism:nnnnn{}{morphism}{#1}{#2}{,shtml:total=true}
3539 \% \stex_execute_in_module:x{
3540 \% \stex_activate_module:n{\l_stex_current_domain_str}
3541 \%}
3542 \stex_if_smsmode:F {
3543   \tl_set:Nn \thiscopyname { #2 }
3544   \tl_set_eq:NN \thismoduleuri \l_stex_current_domain_str
3545   \stex_style_apply:
3546 }
3547 \stex_smsmode_do:
3548 }{
3549   \stex_structural_feature_morphism_check_total:
3550   \stex_if_smsmode:F {
3551     \stex_style_apply:
3552   }
3553   \stex_structural_feature_morphism_end:
3554 }{}{}{}
3555 \stex_deactivate_macro:Nn \realization {module-environments}
3556 \stex_every_module:n {
3557   \stex_reactivate_macro:N \realization
3558 }
3559 \stex_sms_allow_env:n{\realization}

3560 \cs_new_protected:Nn \__stex_morphisms_parse_assign:n {
3561   \str_clear:N \l__stex_morphisms_name_str
3562   \str_clear:N \l__stex_morphisms_newname_str
3563   \tl_clear:N \l__stex_morphisms_ass_tl
3564   \stex_debug:nn{morphisms}{Parsing~\exp_not:n{#1}}
3565   \exp_args:NNN \seq_set_split:Nnn \l__stex_morphisms_seq {\tl_to_str:n@} {#1}
3566   \int_compare:nNnTF {\seq_count:N \l__stex_morphisms_seq} = 1 {
3567     \stex_debug:nnf{morphisms}{No~@}
3568     \seq_pop_left:NN \l__stex_morphisms_seq \l__stex_morphisms_next_tl
3569   }{
3570     \seq_pop_left:NN \l__stex_morphisms_seq \l__stex_morphisms_name_str
3571     \stex_debug:nnf{morphisms}{Name:~\l__stex_morphisms_name_str}
3572     \exp_args:NNN \str_set:Nn \l__stex_morphisms_name_str \l__stex_morphisms_name_str
3573     \tl_set:Nx \l__stex_morphisms_next_tl {\seq_use:Nn \l__stex_morphisms_seq @}
3574   }
3575   \exp_args:NNN \seq_set_split:Nnn \l__stex_morphisms_seq = \l__stex_morphisms_next_tl
3576   \str_if_empty:NTF \l__stex_morphisms_name_str {
3577     \seq_pop_left:NN \l__stex_morphisms_seq \l__stex_morphisms_name_str
3578     \exp_args:NNN \str_set:Nn \l__stex_morphisms_name_str \l__stex_morphisms_name_str
3579     \tl_set:Nx \l__stex_morphisms_ass_tl {\seq_use:Nn \l__stex_morphisms_seq =}
3580   }{
3581     \seq_pop_left:NN \l__stex_morphisms_seq \l__stex_morphisms_newname_str
3582     \exp_args:NNN \str_set:Nn \l__stex_morphisms_newname_str \l__stex_morphisms_newname_str
3583     \tl_set:Nx \l__stex_morphisms_ass_tl {\seq_use:Nn \l__stex_morphisms_seq =}
3584   }
3585   \__stex_morphisms_do_parsed_assign:
3586 }
3587
3588 \cs_new_protected:Nn \__stex_morphisms_do_parsed_assign: {
3589   \exp_args:No \stex_get_in_morphism:n \l__stex_morphisms_name_str
3590   \str_if_empty:NF \l__stex_morphisms_newname_str {

```

```

3591     \exp_after:wN \__stex_morphisms_do_parsed_newname: \l__stex_morphisms_newname_str \__ste
3592   }
3593   \tl_if_empty:NF \l__stex_morphisms_ass_tl {
3594     \exp_args:No \stex_assign_do:n \l__stex_morphisms_ass_tl
3595   }
3596 }
3597
3598 \cs_new_protected:Nn \__stex_morphisms_do_parsed_newname: {
3599   \peek_charcode:NTF [ {
3600     \__stex_morphisms_do_parsed_newname:w
3601   }{
3602     \__stex_morphisms_do_parsed_newname:w []
3603   }
3604 }
3605
3606 \cs_new_protected:Npn \__stex_morphisms_do_parsed_newname:w [#1] #2 \__stex_morphisms_end: {
3607   \stex_renamedecl_do:nn{#1}{#2}
3608 }
3609
3610 \stex_new_stylable_cmd:nnnn{copymod}{m O{} m m}{
3611   \stex_structural_feature_morphism:nnnnn{#1}{morphism}{#2}{#3}{,shtml:total=false}
3612
3613 \clist_map_function:nN{#4}\__stex_morphisms_parse_assign:n
3614
3615 \stex_if_smsmode:F {
3616   \tl_set:Nn \thiscopyname { #1 }
3617   \tl_set_eq:NN \thismoduleuri \l_stex_current_domain_str
3618   \stex_style_apply:
3619 }
3620 \stex_structural_feature_morphism_end:
3621 \stex_smsmode_do:
3622 }{}
3623 \stex_deactivate_macro:Nn \copymod {module~environments}
3624 \stex_every_module:n {
3625   \stex_reactivate_macro:N \copymod
3626 }
3627 \stex_sms_allow_escape:N\copymod
3628
3629
3630 \stex_new_stylable_cmd:nnnn{interpretmod}{m O{} m m}{
3631   \stex_structural_feature_morphism:nnnnn{#1}{morphism}{#2}{#3}{,shtml:total=true}
3632
3633 \clist_map_function:nN{#4}\__stex_morphisms_parse_assign:n
3634
3635 \stex_if_smsmode:F {
3636   \tl_set:Nn \thiscopyname { #1 }
3637   \tl_set_eq:NN \thismoduleuri \l_stex_current_domain_str
3638   \stex_style_apply:
3639 }
3640 \stex_structural_feature_morphism_check_total:
3641 \stex_structural_feature_morphism_end:
3642 \stex_smsmode_do:
3643 }{}
3644 \stex_deactivate_macro:Nn \interpretmod {module~environments}

```

```

3645 \stex_every_module:n {
3646   \stex_reactivate_macro:N \interpretmod
3647 }
3648 \stex_sms_allow_escape:N\interpretmod
3649
3650
3651 \stex_new_stylable_cmd:nnnn{realize}{0{} m m}{
3652   \stex_structural_feature_morphism:nnnnn{}{morphism}{#1}{#2}{,shtml:total=true}
3653
3654 \clist_map_function:nN{#3}\_stex_morphisms_parse_assign:n
3655
3656 \stex_if_smsmode:F {
3657   \tl_set:Nn \thiscopyname { #1 }
3658   \tl_set_eq:NN \thismoduleuri \l_stex_current_domain_str
3659   \stex_style_apply:
3660 }
3661 \stex_structural_feature_morphism_check_total:
3662 \stex_structural_feature_morphism_end:
3663 \stex_smsmode_do:
3664 }{}
3665 \stex_deactivate_macro:Nn \realize {module~environments}
3666 \stex_every_module:n {
3667   \stex_reactivate_macro:N \realize
3668 }
3669 \stex_sms_allow_escape:N\realize

```

## 13.8 Symbols

### 13.8.1 Declarations

```
3670 <@=stex_symdecl>
```

Some setup:

```
\stex_if_check_terms_p:
\stex_if_check_terms:TF
3671 \stex_if_html_backend:TF {
3672   \prg_new_conditional:Nnn \stex_if_check_terms: {p, T, F, TF} {
3673     \prg_return_false:
3674   }
3675 }{
3676   \stex_get_env:Nn\__stex_symdecl_env_str{STEX_CHECKTERMS}
3677   \str_if_empty:NF\__stex_symdecl_env_str{
3678     \exp_args:No \str_if_eq:nnF \__stex_symdecl_env_str{false} {
3679       \bool_set_true:N \c_stex_check_terms_bool
3680     }
3681   }
3682   \bool_if:NTF \c_stex_check_terms_bool {
3683     \prg_new_conditional:Nnn \stex_if_check_terms: {p, T, F, TF} {
3684       \prg_return_true:
3685     }
3686   }{
3687     \prg_new_conditional:Nnn \stex_if_check_terms: {p, T, F, TF} {
3688       \prg_return_false:
3689     }
3690   }
3691 }
```

(End of definition for \stex\_if\_check\_terms:TF. This function is documented on page 124.)

\stex\_check\_term:n

```
3692 \stex_if_check_terms:TF{
3693   \cs_new_protected:Nn \stex_check_term:n {
3694     \hbox_set:Nn \l_tmpa_box {
3695       \group_begin:
3696         $#1$%
3697       \group_end:
3698     }
3699   }
3700 }{
3701   \cs_new_protected:Nn \stex_check_term:n {}
3702 }
```

(End of definition for \stex\_check\_term:n. This function is documented on page 124.)

symdecl arguments:

```
3703 \stex_keys_define:nnnn{symargs}{
3704   \str_clear:N \l_stex_key_args_str
3705   \str_clear:N \l_stex_key_role_str
3706   \str_clear:N \l_stex_key_reorder_str
3707   \str_clear:N \l_stex_key_assoc_str
3708 }{
3709   args      .str_set:N = \l_stex_key_args_str ,
3710   reorder   .str_set:N = \l_stex_key_reorder_str ,
3711   assoc     .choices:nn  = {bin,binl,binr,pre,conj,pwconj}
3712   {\str_set:Nx \l_stex_key_assoc_str \l_keys_choice_tl},
3713   role      .str_set:N    = \l_stex_key_role_str
3714 }{}
3715
3716 \stex_keys_define:nnnn{decl}){
3717   \str_clear:N \l_stex_key_name_str
3718   \str_clear:N \l_stex_key_args_str
3719   \tl_clear:N \l_stex_key_type_tl
3720   \tl_clear:N \l_stex_key_def_tl
3721   \tl_clear:N \l_stex_key_return_tl
3722   \str_clear:N \l_stex_key_wikidata_str
3723   \clist_clear:N \l_stex_key_argtypes_clist
3724 }{
3725   name      .str_set:N = \l_stex_key_name_str ,
3726
3727   return    .tl_set:N    = \l_stex_key_return_tl ,
3728   argtypes  .clist_set:N = \l_stex_key_argtypes_clist ,
3729   wikidata  .str_set:N    = \l_stex_key_wikidata_str ,
3730
3731   type      .tl_set:N    = \l_stex_key_type_tl ,
3732   def       .tl_set:N    = \l_stex_key_def_tl ,
3733
3734   align     .code:n      = {},
3735   gfc      .code:n      = {}
3736 }{style,deprecate,symargs}
3737 % \_stex_do_deprecation:n{#2}
```

\symdecl

```

3738 \str_new:N \l_stex_macroname_str
3739 \stex_new_stylable_cmd:nnnn {symdecl} { s m O{} } {
3740   \stex_keys_set:nn{decl}{#3}
3741   \IfBooleanTF #1 {
3742     \str_clear:N \l_stex_macroname_str
3743   }{
3744     \str_set:Nx \l_stex_macroname_str { #2 }
3745   }
3746 \stex_symdecl_top:n{#2}

3747
3748 \stex_if_smsmode:F{
3749   \group_begin:
3750   \tl_set:Nx \thisdecluri {\l_stex_current_module_str ? \l_stex_key_name_str}
3751   \tl_set_eq:NN \thisdeclname \l_stex_key_name_str
3752   \tl_set_eq:NN \thistype \l_stex_key_type_tl
3753   \tl_set_eq:NN \thisdefiniens \l_stex_key_def_tl
3754   \tl_set_eq:NN \thisargs \l_stex_key_args_str
3755   \tl_clear:N \thisstyle
3756   \stex_style_apply:
3757   \group_end:
3758 }
3759 \stex_smsmode_do:
3760 }{}
3761 \stex_deactivate_macro:Nn \symdecl {module~environments}
3762 \stex_every_module:n {\stex_reactivate_macro:N \symdecl}
3763 \stex_sms_allow_escape:N \symdecl

```

(End of definition for `\symdecl`. This function is documented on page 87.)

### `\stex_symdecl_top:n`

```

3764 \cs_new_protected:Nn \stex_symdecl_top:n {
3765   \str_if_empty:NT \l_stex_key_name_str {
3766     \str_set:Nx \l_stex_key_name_str { #1 }
3767   }
3768   \stex_symdecl_do:
3769   \stex_symdecl_check_terms:
3770   \__stex_symdecl_add_decl:
3771   \stex_if_do_html:T {
3772     \stex_symdecl_html:
3773   }
3774 }

3775
3776 \cs_new_protected:Nn \__stex_symdecl_add_decl: {
3777   \exp_args:Nnx \use:nn {\stex_module_add_symbol:nnnnnnnN} {
3778     {\l_stex_macroname_str}
3779     {\l_stex_key_name_str}
3780     {\int_use:N \l_stex_get_symbol_arity_int}
3781     {\l_stex_get_symbol_args_tl}
3782     {\tl_if_empty:NF \l_stex_key_def_tl{DEFED} }%{\exp_args:No \exp_not:n \l_stex_key_def_tl
3783     {}%{\exp_args:No \exp_not:n \l_stex_key_type_tl}
3784     {\exp_args:No \exp_not:n \l_stex_key_return_tl}
3785     \stex_invoke_symbol:
3786   }
3787   \exp_args:Ne \stex_ref_new_symbol:n

```

```

3788     {\l_stex_current_module_str?\l_stex_key_name_str}
3789 }
3790
3791 \cs_new:Nn \_stex_return_args:nn {
3792     {\svar{ARGUMENT_#1}\_stex_eat_exclamation_point:}
3793 }
3794
3795 \cs_new_protected:Nn \_stex_symdecl_html: {
3796     \exp_args:Ne \stex_annotation_invisible:nn {
3797         shtml:symdecl = {\l_stex_current_module_str ? \l_stex_key_name_str},
3798         shtml:args = {\l_stex_key_args_str}
3799         \str_if_empty:NF \l_stex_mroname_str {
3800             shtml:macroname={\l_stex_mroname_str}
3801         }
3802         \str_if_empty:NF \l_stex_key_wikidata_str {
3803             shtml:wikidata={\l_stex_key_wikidata_str}
3804         }
3805         \str_if_empty:NF \l_stex_key_assoc_str {
3806             shtml:assocotype={\l_stex_key_assoc_str}
3807         }
3808         \str_if_empty:NF \l_stex_key_reorder_str {
3809             shtml:reorderargs={\l_stex_key_reorder_str}
3810         }
3811         \str_if_empty:NF \l_stex_key_role_str {
3812             shtml:role={\l_stex_key_role_str}
3813         }
3814     }{\hbox\bgroup\_stex_annotation_force_break:n{
3815         \bool_set_true:N \stex_in_invisible_html_bool
3816         \tl_if_empty:NF \l_stex_key_type_tl {
3817             $\\stex_annotation:nn{shtml:type={}{}{\l_stex_key_type_tl}}$}
3818         }
3819         \tl_if_empty:NF \l_stex_key_def_tl {
3820             $\\stex_annotation:nn{shtml:definiens={}{}{\l_stex_key_def_tl}}$}
3821         }
3822         \tl_if_empty:NF \l_stex_key_return_tl{
3823             \exp_args:Nno \use:n{
3824                 \cs_generate_from_arg_count:NNnn \l_stex_symdecl_cs
3825                 \cs_set:Npn \l_stex_get_symbol_arity_int} \l_stex_key_return_tl
3826                 \tl_set:Nx \l_stex_symdecl_args_tl {\l_stex_map_args:N \_stex_return_args:nn}
3827                 $\\stex_annotation:nn{shtml:returnde={}}${
3828                     \exp_after:wN \l_stex_symdecl_cs \l_stex_symdecl_args_tl!
3829                 }$}
3830             }
3831             \clist_if_empty:NF \l_stex_key_argtypes_clist {
3832                 \stex_annotation:nn{shtml:argtypes={}{}{\_stex_annotation_force_break:n{
3833                     \clist_map_inline:Nn \l_stex_key_argtypes_clist {
3834                         $\\stex_annotation:nn{shtml:type={}{}{\##1}}$}
3835                     }
3836                 }}}
3837             }
3838         }\\egroup}
3839 }

```

(End of definition for `\stex_symdecl_top:n`. This function is documented on page 125.)

\stex\_symdecl\_do: Requires the above keys and \l\_stex\_macroname\_str to be set first

```
3840 \cs_new_protected:Nn \stex_symdecl_do: {
3841   \_stex_do_deprecation:n \l_stex_key_name_str
3842   \__stex_symdecl_parse_arity:
3843   \__stex_symdecl_do_args:
3844 }
3845
3846 \int_new:N \l_stex_assoc_args_count
3847
3848 \cs_new_protected:Nn \__stex_symdecl_parse_arity: {
3849   \int_zero:N \l_stex_get_symbol_arity_int
3850   \int_zero:N \l_stex_assoc_args_count
3851   \str_map_inline:Nn \l_stex_key_args_str {
3852     \str_case:nnF ##1 {
3853       0 { \str_map_break: }
3854       1 { \str_map_break:n{
3855         \int_set:Nn \l_stex_get_symbol_arity_int {1}
3856         \str_set:Nn \l_stex_key_args_str {i}
3857       } }
3858       2 { \str_map_break:n{
3859         \int_set:Nn \l_stex_get_symbol_arity_int {2}
3860         \str_set:Nn \l_stex_key_args_str {ii}
3861       } }
3862       3 { \str_map_break:n{
3863         \int_set:Nn \l_stex_get_symbol_arity_int {3}
3864         \str_set:Nn \l_stex_key_args_str {iii}
3865       } }
3866       4 { \str_map_break:n{
3867         \int_set:Nn \l_stex_get_symbol_arity_int {4}
3868         \str_set:Nn \l_stex_key_args_str {iiii}
3869       } }
3870       5 { \str_map_break:n{
3871         \int_set:Nn \l_stex_get_symbol_arity_int {5}
3872         \str_set:Nn \l_stex_key_args_str {iiiii}
3873       } }
3874       6 { \str_map_break:n{
3875         \int_set:Nn \l_stex_get_symbol_arity_int {6}
3876         \str_set:Nn \l_stex_key_args_str {iiiiii}
3877       } }
3878       7 { \str_map_break:n{
3879         \int_set:Nn \l_stex_get_symbol_arity_int {7}
3880         \str_set:Nn \l_stex_key_args_str {iiiiiii}
3881       } }
3882       8 { \str_map_break:n{
3883         \int_set:Nn \l_stex_get_symbol_arity_int {8}
3884         \str_set:Nn \l_stex_key_args_str {iiiiiiii}
3885       } }
3886       9 { \str_map_break:n{
3887         \int_set:Nn \l_stex_get_symbol_arity_int {9}
3888         \str_set:Nn \l_stex_key_args_str {iiiiiiii}
3889       } }
3890       i {\int_incr:N \l_stex_get_symbol_arity_int}
3891       b {\int_incr:N \l_stex_get_symbol_arity_int}
3892       a {\int_incr:N \l_stex_get_symbol_arity_int \int_incr:N \l_stex_assoc_args_count}
```

```

3893     B {\int_incr:N \l_stex_get_symbol_arity_int \int_incr:N \l_stex_assoc_args_count}
3894     }{
3895         \msg_error:n{stex}{error/wrongargs} {
3896             \l_stex_current_module_str ? \l_stex_key_name_str
3897             }{##1}
3898         }
3899     }
3900 }
3901 \cs_new_protected:Nn \__stex_symdecl_do_args: {
3902     \tl_clear:N \l_stex_get_symbol_args_tl
3903     \int_step_inline:nn \l_stex_get_symbol_arity_int {
3904         \tl_put_right:Nn \l_stex_get_symbol_args_tl {##1}
3905         \tl_put_right:Nx \l_stex_get_symbol_args_tl {
3906             \str_item:Nn \l_stex_key_args_str {##1}
3907         }
3908     }
3909 }
3910 }
```

(End of definition for `\stex_symdecl_do:`. This function is documented on page 124.)

### `\_stex_symdecl_check_terms:`

```

3911 \cs_new_protected:Nn \_stex_symdecl_check_terms: {
3912     \stex_check_term:n{
3913         \stex_debug:nn{check_terms}{Checking~type...}
3914         \group_begin:\l_stex_key_type_tl\group_end:
3915         \stex_debug:nn{check_terms}{Checking~definiens...}
3916         \group_begin:\l_stex_key_def_tl\group_end:
3917         \stex_debug:nn{check_terms}{Checking~return...}
3918         \group_begin:\l_stex_key_return_tl!\group_end:
3919         \stex_debug:nn{check_terms}{Checking~argument~types...}
3920         \group_begin:\l_stex_key_argtypes_clist\group_end:
3921     }
3922 }
```

(End of definition for `\_stex_symdecl_check_terms:`. This function is documented on page 125.)

### `\textsymdecl`

```

3923
3924 \stex_keys_define:nnnn{textsymdecl}{
3925     \str_clear:N \l_stex_key_name_str
3926     \tl_clear:N \l_stex_key_type_tl
3927     \tl_clear:N \l_stex_key_def_tl
3928 }{
3929     name      .str_set:N = \l_stex_key_name_str ,
3930     type      .tl_set:N   = \l_stex_key_type_tl ,
3931     def       .tl_set:N   = \l_stex_key_def_tl
3932 }{style,deprecate}
3933
3934 \stex_new_stylable_cmd:nnnn {textsymdecl} {m 0{} m} {
3935     \stex_keys_set:nn{symdef}{}%
3936     \stex_keys_set:nn{textsymdecl}{#2}
3937     \str_set:Nx \l_stex_macroname_str { #1 }
3938     \str_if_empty:NT \l_stex_key_name_str {
3939         \str_set:Nn \l_stex_key_name_str {#1}
```

```

3940 }%{
3941 % \str_set:Nx \l_stex_key_name_str {\l_stex_key_name_str-sym}
3942 %}
3943 \str_set:Nn \l_stex_key_role_str {textsymdecl}
3944
3945 \stex_symdecl_do:
3946 \_stex_symdecl_check_terms:
3947 \exp_args:Nnx \use:nn {\stex_module_add_symbol:nnnnnnnN} {
3948   {\l_stex_macroname_str}
3949   {\l_stex_key_name_str}
3950   {0}{}
3951   {\tl_if_empty:NF \l_stex_key_def_tl{DEFED} }
3952   {}% type
3953   {\use:c{\#1name_nospace}}% return
3954   \stex_invoke_text_symbol:
3955 }
3956 \exp_args:Ne \stex_ref_new_symbol:n
3957   {\l_stex_current_module_str?\l_stex_key_name_str}
3958 \stex_if_do_html:T {
3959   \_stex_symdecl_html:
3960 }
3961
3962 \int_set:Nn \l_stex_get_symbol_arity_int 0
3963 \tl_clear:N \l_stex_key_op_tl
3964 \str_clear:N \l_stex_key_intent_str
3965 \str_clear:N \l_stex_key_prec_str
3966 \str_set_eq:NN \l_stex_get_symbol_mod_str \l_stex_current_module_str
3967 \str_set_eq:NN \l_stex_get_symbol_name_str \l_stex_key_name_str
3968 \stex_notation_parse:n{\hbox{#3}}
3969 \_stex_notation_add:
3970 \stex_if_do_html:T {
3971   \def\comp{\_comp}
3972   \_stex_notation_do_html:n{\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
3973 }
3974 \stex_execute_in_module:x{
3975   \_stex_symdecl_set_textsymdecl_macro:nnn{#1}{\l_stex_current_module_str?\l_stex_key_name}
3976   \exp_not:n{#3}
3977 }
3978
3979 \stex_if_smsmode:F{
3980   \group_begin:
3981   \tl_set:Nx \thisdecluri {\l_stex_current_module_str ? \l_stex_key_name_str}
3982   \tl_set_eq:NN \thisdeclname \l_stex_key_name_str
3983   \tl_clear:N \thisstyle
3984   \stex_style_apply:
3985   \group_end:
3986 }
3987 \stex_smsmode_do:
3988 }{}
3989 \stex_deactivate_macro:Nn \textsymdecl {module-environments}
3990 \stex_every_module:n {\stex_reactivate_macro:N \textsymdecl}
3991 \stex_sms_allow_escape:N \textsymdecl
3992
3993 \cs_new_protected:Nn \_stex_symdecl_set_textsymdecl_macro:nnn {

```

```

3994 \cs_set_protected:cpn{\#1name_nospace}{#3}
3995 \cs_set_protected:cpn{\#1name} {
3996     \mode_if_vertical:T{\hbox_unpack:N\c_empty_box}
3997     \mode_if_math:T\hbox{\let\xspace\relax #3}
3998     \mode_if_math:F{\cs_if_exist:NT\xspace\xspace}
3999 }
4000 }
4001
4002 \cs_new_protected:Nn \stex_invoke_text_symbol: {
4003     \mode_if_vertical:T{\hbox_unpack:N\c_empty_box}
4004     \stex_term_oms_or_omv:nnn{}{}{\maincomp{\let\xspace\relax\l_stex_current_return_tl}}
4005     \group_end:\mode_if_math:F{\cs_if_exist:NT\xspace\xspace}
4006 }

```

(End of definition for `\textsymdecl`. This function is documented on page 88.)

### `\stex_get_symbol:n`

```

4007 \cs_new_protected:Nn \stex_get_symbol:n {
4008     \stex_get_symbol:n{ #1 }
4009     \str_if_empty:NT \l_stex_get_symbol_name_str {
4010         \msg_error:nnn{stex}{error/unknownsymbol}{#1}
4011     }
4012 }
4013
4014 \cs_new_protected:Nn \stex_get_symbol:n {
4015     \str_clear:N \l_stex_get_symbol_mod_str
4016     \str_clear:N \l_stex_get_symbol_name_str
4017     \cs_if_exist:cTF { #1 }{
4018         \cs_set_eq:Nc \l_stex_symdecl_cs { #1 }
4019         % command name
4020         \exp_args:Ne \tl_if_empty:nTF { \cs_argument_spec:N \l_stex_symdecl_cs }{
4021             % ...that takes no arguments
4022             \exp_args:Ne \cs_if_eq:NNTF {\tl_head:N \l_stex_symdecl_cs}
4023                 \stex_invoke_symbol:nnnnnnnn
4024                 \stex_symdecl_get_symbol_from_cs:
4025                 {\stex_symdecl_get_symbol_from_string:n { #1 }}
4026             }{
4027                 \stex_symdecl_get_symbol_from_string:n { #1 }
4028             }
4029         }{
4030             \stex_symdecl_get_symbol_from_string:n { #1 }
4031         }
4032     }
4033
4034 \int_new:N \l_stex_get_symbol_arity_int
4035 \cs_new_protected:Nn \stex_symdecl_get_symbol_from_cs: {
4036     \stex_debug:nn{symbols}{Getting~from~cs...}
4037     \stex_pseudogroup_with:nn{\stex_invoke_symbol:nnnnnnnn}{%
4038         \cs_set:Npn \stex_invoke_symbol:nnnnnnnn ##1 ##2 ##3 ##4 ##5 ##6 ##7 ##8 {
4039             \str_set:Nn \l_stex_get_symbol_mod_str {##1}
4040             \str_set:Nn \l_stex_get_symbol_name_str {##2}
4041             \int_set:Nn \l_stex_get_symbol_arity_int {##3}
4042             \tl_set:Nn \l_stex_get_symbol_args_tl {##4}
4043             \tl_set:Nn \l_stex_get_symbol_def_tl {##5}

```

```

4044     \tl_set:Nn \l_stex_get_symbol_type_tl {##6}
4045     \tl_set:Nn \l_stex_get_symbol_return_tl {##7}
4046     \tl_set:Nn \l_stex_get_symbol_invoke_cs {##8}
4047   }
4048   \l_stex_symdecl_cs
4049 }
4050 }
4051
4052 \cs_new_protected:Nn \__stex_symdecl_get_symbol_from_string:n {
4053   \stex_debug:nn{symbols}{Getting~from~string~#1...}
4054   \seq_set_split:Nnn \l_stex_symdecl_seq ? {#1}
4055   \seq_pop_right:NN \l_stex_symdecl_seq \l_stex_symdecl_name
4056   \seq_if_empty:NTF \l_stex_symdecl_seq {
4057     \exp_args:No \__stex_symdecl_get_from_one_string:n {#1}
4058   }{
4059     \exp_args:NNe \exp_args:Nno \__stex_symdecl_get_symbol_from_modules:nn {
4060       \seq_use:Nn \l_stex_symdecl_seq ?
4061     } \l_stex_symdecl_name
4062   }
4063 }
4064
4065 \cs_new_protected:Nn \__stex_symdecl_sym_from_str_i:nnnn {
4066   \bool_lazy_any:nTF{
4067     {\str_if_eq_p:nn{#2}{#3}}
4068     {\str_if_eq_p:nn{#2}{#4}}
4069     {\str_if_ends_with_p:nn{#4}{/#2}}
4070   }{
4071     \__stex_symdecl_sym_i_finish:nnnnnnN{#1}{#4}
4072   }{
4073     \__stex_symdecl_sym_i_gobble:nnnnn
4074   }
4075 }
4076 \cs_new_protected:Nn \__stex_symdecl_sym_i_gobble:nnnnn {}
4077
4078 \cs_new_protected:Nn \__stex_symdecl_sym_i_finish:nnnnnnN {
4079   \prop_map_break:nf\seq_map_break:nf
4080     \str_set:Nn \l_stex_get_symbol_mod_str {#1}
4081     \str_set:Nn \l_stex_get_symbol_name_str {#2}
4082     \int_set:Nn \l_stex_get_symbol_arity_int {#3}
4083     \tl_set:Nn \l_stex_get_symbol_args_tl {#4}
4084     \tl_set:Nn \l_stex_get_symbol_def_tl {#5}
4085     \tl_set:Nn \l_stex_get_symbol_type_tl {#6}
4086     \tl_set:Nn \l_stex_get_symbol_return_tl {#7}
4087     \tl_set:Nn \l_stex_get_symbol_invoke_cs {#8}
4088   }
4089 }
4090
4091 \cs_new_protected:Nn \__stex_symdecl_get_symbol_from_modules:nn {
4092   \stex_debug:nn{symbols}{Getting~#2~in~#1...}
4093   \seq_map_inline:Nn \l_stex_all_modules_seq {
4094     \stex_str_if_ends_with:nnT{##1}{#1} {
4095       \prop_map_inline:cn{c_stex_module_##1_symbols_prop} {
4096         \__stex_symdecl_sym_from_str_i:nnnn{##1}{#2} ####2
4097       }
4098     }
4099   }
4100 }
```

```

4098     }
4099   }
4100 }
4101
4102 \cs_new_protected:Nn \__stex_symdecl_get_from_one_string:n {
4103   \stex_debug:nn{symbols}{Getting~#1~anywhere...}
4104   \stex_iterate_symbols:n{
4105     \%stex_debug:nn{symbols}{>#1==##2~|~#1==##3<...}
4106     \bool_lazy_any:nT{
4107       {\str_if_eq_p:nn{#1}{##2}}
4108       {\str_if_eq_p:nn{#1}{##3}}
4109       {\stex_str_if_ends_with_p:nn{##3}{/#1}}
4110     }{
4111       \stex_iterate_break:n{
4112         \str_set:Nn \l_stex_get_symbol_mod_str {##1}
4113         \str_set:Nn \l_stex_get_symbol_name_str {##3}
4114         \int_set:Nn \l_stex_get_symbol_arity_int {##4}
4115         \tl_set:Nn \l_stex_get_symbol_args_tl {##5}
4116         \tl_set:Nn \l_stex_get_symbol_def_tl {##6}
4117         \tl_set:Nn \l_stex_get_symbol_type_tl {##7}
4118         \tl_set:Nn \l_stex_get_symbol_return_tl {##8}
4119         \tl_set:Nn \l_stex_get_symbol_invoke_cs {##9}
4120       }
4121     }
4122   }
4123 }

```

(End of definition for `\stex_get_symbol:n`. This function is documented on page 124.)

### 13.8.2 Notations

```

4124 <@=stex_notations>
4125
\__stex_map_args:N
\__stex_map_notation_args:N
4126 \cs_new:Nn \__stex_map_args:N {
4127   \tl_if_empty:NF \l_stex_get_symbol_args_tl {
4128     \exp_after:wN \__stex_notations_map_args_i:w \exp_after:wN
4129     #1 \l_stex_get_symbol_args_tl \__stex_notations_args_end:
4130   }
4131 \cs_new:Npn \__stex_notations_map_args_i:w #1 #2 #3 #4 \__stex_notations_args_end: {
4132   #1 #2 #3
4133   \tl_if_empty:nF{#4}{
4134     \__stex_notations_map_args_i:w #1 #4 \__stex_notations_args_end:
4135   }
4136 }
4137
4138 \cs_new:Nn \__stex_map_notation_args:N {
4139   \tl_if_empty:NF \l_stex_notation_args_tl {
4140     \exp_after:wN \__stex_notations_map_args_ii:w \exp_after:wN
4141     #1 \l_stex_get_symbol_args_tl \__stex_notations_args_end:
4142   }
4143 }
4144 \cs_new:Npn \__stex_notations_map_args_ii:w #1 #2 #3 #4 #5 #6 \__stex_notations_args_end: {
4145   #1 #2 #3 #4 #5

```

```

4146   \tl_if_empty:nF{#6}{
4147     \__stex_notations_map_args_ii:w #1 #6 \__stex_notations_args_end:
4148   }
4149 }
```

(End of definition for `\_stex_map_args:N` and `\_stex_map_notation_args:N`. These functions are documented on page ??.)

notation arguments:

```

4150 \stex_keys_define:nnnn{notation}{
4151   \str_clear:N \l_stex_key_variant_str
4152   \str_clear:N \l_stex_key_prec_str
4153   \str_clear:N \l_stex_key_op_tl
4154   \str_clear:N \l_stex_key_intent_str
4155   \clist_clear:N \l_stex_key_intent_args_clist
4156 }{
4157   variant    .str_set_x:N = \l_stex_key_variant_str ,
4158   prec      .str_set_x:N = \l_stex_key_prec_str ,
4159   op        .tl_set:N     = \l_stex_key_op_tl ,
4160   intent     .str_set:N = \l_stex_key_intent_str ,
4161   argnames   .clist_set:N = \l_stex_key_intent_args_clist ,
4162   unknown    .code:n     = {
4163     \str_if_empty:NTF \l_keys_key_str {
4164       \str_set:Nx \l_stex_key_variant_str {\l_keys_key_tl}
4165     }{
4166       \str_set_eq:NN \l_stex_key_variant_str \l_keys_key_str
4167     }
4168   }
4169 }{style}
```

### \notation

```

4170 \stex_new_stylable_cmd:nnnn {notation} { s m 0{} m} {
4171   \stex_keys_set:nn{notation}{#3}
4172   \stex_get_symbol:n{#2}
4173   \stex_notation_parse:n{#4}
4174   \stex_if_check_terms:T{ \_stex_notation_check: }
4175   \_stex_notation_add:
4176   \stex_if_do_html:T {
4177     \def\comp{\_comp}
4178     \_stex_notation_do_html:nf{\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
4179   }
4180   \IfBooleanTF#1{
4181     \_stex_notation_set_default:n{
4182       \l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str
4183     }
4184   }{}
4185   \stex_if_smsmode:F{
4186     \group_begin:
4187     \__stex_notations_styledefs:
4188     \stex_style_apply:
4189     \group_end:
4190   }
4191   \stex_smsmode_do:
4192 }{}  
4193
```

```

4194 \cs_new_protected:Nn \__stex_notations_styledefs: {
4195   \str_set_eq:NN \thisnotationvariant \l_stex_key_variant_str
4196   \str_set:Nn \thisdeclname \l_stex_get_symbol_name_str
4197   \tl_set:Nx \thisdecluri {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
4198   \def\thisnotation{
4199     $
4200     \tl_set_eq:NN \l_stex_current_symbol_str \thisdecluri
4201     \exp_args:Nne \use:nn{\l_stex_notation_macrocode_cs} {
4202       \__stex_notation_make_args:
4203     }$
4204   }
4205 }
4206
4207 \stex_deactivate_macro:Nn \notation {module~environments}
4208 \stex_every_module:n {\stex_reactivate_macro:N \notation}
4209 \stex_sms_allow_escape:N \notation

```

(End of definition for `\notation`. This function is documented on page 90.)

`\stex_notation_parse:n` requires the above keys, `\l_stex_get_symbol_arity_int`, and `\l_stex_get_symbol_args_tl`

```

4210 \cs_new_protected:Nn \stex_notation_parse:n {
4211   \tl_if_empty:NF \l_stex_key_op_tl {
4212     \tl_set:Nx \l_stex_key_op_tl { \exp_not:N \maincomp {
4213       \exp_args:No \exp_not:n \l_stex_key_op_tl
4214     } }
4215   }
4216   \seq_clear:N \l__stex_notations_precs_seq
4217   \tl_clear:N \l_stex_notation_args_tl
4218   \int_compare:nNnTF \l_stex_get_symbol_arity_int = 0 {
4219     \__stex_notations_const_precs:
4220     \tl_if_empty:NT \l_stex_key_op_tl {
4221       \tl_set:Nn \l_stex_key_op_tl { \maincomp{#1} }
4222     }
4223   }
4224   \__stex_notations_fun_precs:
4225   \str_set:Nn \l__stex_notations_missing_str {#1}
4226   \tl_clear:N \l__stex_notations_missing_tl
4227   \stex_map_args:N \__stex_notations_add_missing_args:nn
4228   \tl_if_empty:NT \l_stex_key_op_tl {
4229     \hbox_set:Nn \l_tmpa_box {
4230       \str_set:Nn \l_stex_current_symbol_str {}
4231       \cs_set:Npn \l_tmpa cs ##1 ##2 ##3 ##4 ##5 ##6 ##7 ##8 ##9 { #1 }
4232       \cs_set:Npn \maincomp ##1 {
4233         \tl_gset:Nn \l_stex_key_op_tl { \maincomp{##1} }
4234         ##1
4235       }
4236       \cs_set:Npn \argsep ##1 ##2 {##1 ##2}
4237       \cs_set:Npn \argmap ##1 ##2 ##3 {##1 ##3}
4238       \cs_set:Npn \argarraymap ##1 ##2 ##3 ##4 {
4239         ##1 ##2
4240       }
4241       \stex_suppress_html:n{$\l_tmpa cs abcdefghj$}
4242     }

```

```

4243     }
4244   }
4245 \exp_args:NNe
4246 \tl_set:Nn \l_stex_notation_macrocode_cs {
4247   \STEXInternalNotation
4248   { \l_stex_key_variant_str }
4249   { \l_stex_notations_oppref_tl }
4250   { \l_stex_key_intent_str }
4251   { \l_stex_notation_args_tl }
4252   {
4253     \int_compare:nNnTF \l_stex_get_symbol_arity_int = 0
4254     { \exp_not:n { \maincomp{ #1 } } }
4255     { \exp_not:n { #1 } \l_stex_notations_missing_tl }
4256   }
4257 }
4258 \stex_debug:nn{notation}{Notation:~\meaning\l_stex_notation_macrocode_cs}
4259 }
4260
4261 \cs_new_protected:Nn \__stex_notations_const_prets: {
4262   \str_if_empty:NTF \l_stex_key_prec_str {
4263     \tl_set:No \l_stex_notations_oppref_tl { \neginfpref }
4264   }{
4265     \str_if_eq:onTF \l_stex_key_prec_str {nobrackets} {
4266       \tl_set:No \l_stex_notations_oppref_tl { \neginfpref }
4267     }{
4268       \tl_set_eq:NN \l_stex_notations_oppref_tl \l_stex_key_prec_str
4269     }
4270   }
4271 }
4272
4273 \cs_new_protected:Nn \__stex_notations_fun_prets: {
4274   \str_if_empty:NTF \l_stex_key_prec_str {
4275     \tl_set:No \l_stex_notations_oppref_tl { \neginfpref }
4276   }{
4277     \str_if_eq:onTF \l_stex_key_prec_str {nobrackets} {
4278       \tl_set:No \l_stex_notations_oppref_tl { \neginfpref }
4279     }{
4280       \tl_set_eq:NN \l_stex_notations_oppref_tl \l_stex_key_prec_str
4281     }
4282   }
4283 \str_if_empty:NTF \l_stex_key_prec_str {
4284   \tl_set:Nn \l_stex_notations_oppref_tl { 0 }
4285   \int_step_inline:nn \l_stex_get_symbol_arity_int {
4286     \seq_put_right:Nn \l_stex_notations_prets_seq {0}
4287   }
4288 }{
4289   \str_if_eq:onTF \l_stex_key_prec_str {nobrackets} {
4290     \stex_debug:nn{notation}{No~brackets}
4291     \tl_set:No \l_stex_notations_oppref_tl { \neginfpref }
4292     \int_step_inline:nn \l_stex_get_symbol_arity_int {
4293       \exp_args:NNo \seq_put_right:Nn \l_stex_notations_prets_seq \infpref
4294     }
4295   }\__stex_notations_parse_prets:
4296 }

```

```

4297   \_\_stex\_notations\_do\_argnames:
4298 }
4299
4300 \cs_new_protected:Nn \_\_stex\_notations\_parse\_precs: {
4301   \stex_debug:nn{notation}{parsing-precedence~\l_stex_key_prec_str}
4302   \seq_set_split:NnV \l_\_stex\_notations_seq ; \l_stex_key_prec_str
4303   \seq_pop_left:NNTF \l_\_stex\_notations_seq \l_\_stex\_notations_str {
4304     \tl_set_eq:NN \l_\_stex\_notations_oppref_tl \l_\_stex\_notations_str
4305     \seq_pop_left:NNT \l_\_stex\_notations_seq \l_\_stex\_notations_str {
4306       \exp_args:NNo \seq_set_split:NnV \l_\_stex\_notations_seq
4307       {\tl_to_str:n{x}} \l_\_stex\_notations_str
4308     }
4309   }{
4310     \tl_set:No \l_\_stex\_notations_oppref_tl { 0 }
4311   }
4312   \int_step_inline:nn \l_stex_get_symbol_arity_int {
4313     \seq_pop_left:NNTF \l_\_stex\_notations_seq \l_\_stex\_notations_str {
4314       \seq_put_right:No \l_\_stex\_notations_precs_seq \l_\_stex\_notations_str
4315     }{
4316       \seq_put_right:No \l_\_stex\_notations_precs_seq \l_\_stex\_notations_oppref_tl
4317     }
4318   }
4319 }
4320
4321 \cs_new_protected:Nn \_\_stex\_notations\_do\_argnames: {
4322   \tl_clear:N \l_stex_notation_args_tl
4323   \stex_map_args:N \_\_stex\_notations\_do\_argname:nn
4324 }
4325
4326 \cs_new_protected:Nn \_\_stex\_notations\_do\_argname:nn {
4327   \clist_if_empty:NTF \l_stex_key_intent_args_clist {
4328     \tl_put_right:Nx \l_stex_notation_args_tl {
4329       #1#2{\seq_item:Nn \l_\_stex\_notations_precs_seq #1} {
4330         \str_if_empty:NF \l_stex_key_intent_str {#1}
4331       }
4332     }
4333   }{
4334     \tl_put_right:Nx \l_stex_notation_args_tl {
4335       #1#2{\seq_item:Nn \l_\_stex\_notations_precs_seq #1} {
4336         {\c_dollar_str\clist_item:Nn \l_stex_key_intent_args_clist 1}
4337       }
4338     \clist_pop:NN \l_stex_key_intent_args_clist \l_tmpa_tl
4339   }
4340 }
4341
4342 \cs_new:Nn \_\_stex\_notations\_add\_missing\_args:nn {
4343   \exp_args:NNe \str_if_in:NnF \l_\_stex\_notations_missing_str {\c_hash_str\c_hash_str#1} {
4344     \tl_put_right:Nn \l_\_stex\_notations_missing_tl{\STEXinvis{## #1}}
4345   }
4346 }

```

(End of definition for `\stex_notation_parse:n`. This function is documented on page ??.)

```

\_\_stex\_notation_check:
\_\_stex\_notation_add:
\_\_stex\_notation_do_html:n
\_\_stex\_notation_make_args:

```

```

4347 \cs_new_protected:Nn \_stex_notation_check: {
4348   \stex_check_term:n{
4349     \str_set:Nn \l_stex_current_symbol_str {test}
4350     \cs_set:Npn \comp ##1 {##1}
4351     \stex_debug:nn{check_terms}{Checking~notation...}
4352     \exp_args:Nne \use:nn{\l_stex_notation_macrocode_cs}{}{
4353       \_stex_notation_make_args:
4354     }
4355   }
4356 }
4357
4358 \cs_new_protected:Nn \_stex_notation_add: {
4359   \stex_module_add_notation:eoeeo{
4360     \l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str
4361   } \l_stex_key_variant_str
4362   {\int_use:N \l_stex_get_symbol_arity_int}
4363   \l_stex_notation_macrocode_cs
4364   \l_stex_key_op_tl
4365 }
4366
4367 \cs_new_protected:Nn \_stex_notation_do_html:n {
4368   \hbox{\stex_annotation_invisible:nn {
4369     \shtml:notation={#1},
4370     \shtml:notationfragment={\l_stex_key_variant_str},
4371     \shtml:precedence={\l_stex_notations_oppref_tl},
4372     \shtml:argprecs={\seq_use:Nn \l_stex_notations_precs_seq ,}
4373   }{
4374     \cs_set_protected:Npn \argsep ##1 ##2 {
4375       \stex_annotation:nn{\shtml:argsep={}}{
4376         ##1 ##2
4377       }
4378     }
4379     \cs_set_protected:Npn \argmap ##1 ##2 ##3 {
4380       \cs_set:Npn \l_stex_notations_map_cs: #####1 { ##2 }
4381       \stex_annotation:nn{\shtml:argmap={}}{
4382         \l_stex_notations_map_cs:{##1} ##3
4383       }
4384     }
4385     \cs_set_protected:Npn \maincomp {
4386       \do_comp:nNn {maincomp}\compemph@uri
4387     }
4388     $
4389     \str_set:Nx \l_stex_current_symbol_str {#1}
4390     \stex_annotation:nn{\shtml:notationcomp={}}{
4391       \exp_args:Nne \use:nn {
4392         \l_stex_notation_macrocode_cs {}
4393       }{
4394         \_stex_map_args:N \l_stex_notations_make_arg_html:nn
4395       }
4396     }
4397     $
4398     \tl_if_empty:NF \l_stex_key_op_tl {
4399       $
4400       \str_set:Nx \l_stex_current_symbol_str {#1}

```

```

4401     \stex_annotate:nn{shml:notationopcomp={}{}{
4402         \stex_term_oms:nnn{\l_stex_key_variant_str}{}{\l_stex_key_op_tl}
4403     }
4404     $
4405   }
4406 }
4407 }
4408
4409 \cs_new:Nn \__stex_notations_make_arg_html:nn {
4410 %   \str_case:nnF #2 {
4411 %     a {{
4412 %       \stex_annotate:nn{shml:argnum=#1a}{x},
4413 %       \stex_annotate:nn{shml:argnum=#1b}{x}
4414 %     }}
4415 %     B {{
4416 %       \stex_annotate:nn{shml:argnum=#1a}{x},
4417 %       \stex_annotate:nn{shml:argnum=#1b}{x}
4418 %     }}
4419 %   }{
4420   {
4421     \stex_annotate:nn{shml:argnum=#1}{x}
4422   }
4423 %
4424 }
4425
4426 \cs_new:Nn \stex_notation_make_args: {
4427   \stex_map_notation_args:N \__stex_notations_make_arg:nnnn
4428 }
4429
4430
4431 \cs_new:Nn \__stex_notations_make_arg:nnnn {
4432   \str_case:nnF #2 {
4433     a {{
4434       a\c_math_subscript_token{#1,1},
4435       a\c_math_subscript_token{#1,2}
4436     }}
4437     B {{
4438       B\c_math_subscript_token{#1,1},
4439       B\c_math_subscript_token{#1,2}
4440     }}
4441   }{
4442     \stex_term_arg:nnnn{#1}{#2}{#3}{#4}
4443     {{#2}\c_math_subscript_token{#1}}
4444   }
4445 }

```

(End of definition for `\stex_notation_check`: and others. These functions are documented on page ??.)

### `\setnotation`

```

\stex_notation_set_default:n
4446 \cs_new_protected:Npn \setnotation #1 #2 {
4447   \stex_get_symbol:n{#1}
4448   \cs_if_exist:cTF{\l_stex_notation_
4449     \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str

```

```

4450      _#2_cs
4451  }{
4452    \tl_set_eq:Nc \l_stex_notation_macrocode cs {l_stex_notation_
4453      \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4454      _#2_cs
4455  }
4456  \cs_if_exist:cTF{l_stex_notation_
4457    \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4458    _op_#2_cs
4459  }{
4460    \tl_set_eq:Nc \l_stex_key_op_tl {l_stex_notation_
4461      \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4462      _op_#2_cs
4463  }
4464  \tl_clear:N \l_stex_key_op_tl
4465  }
4466  \stex_notation_set_default:n{
4467    \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4468  }
4469  \msg_error:nnxx{stex}{unknownnotation}{#2}{
4470    \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4471  }
4472  }
4473  }
4474  }
4475 }
4476
4477 \cs_new_protected:Nn \stex_notation_set_default:n{
4478   \stex_if_in_module:TF{
4479     \stex_module_add_notation:eeeo{#1}{}
4480     {\int_use:N \l_stex_get_symbol_arity_int}
4481     \l_stex_notation_macrocode cs
4482     \l_stex_key_op_tl
4483  }{
4484    \cs_set_eq:cN {l_stex_notation_
4485      \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4486      __cs}\l_stex_notation_macrocode cs
4487    \tl_if_empty:NF \l_stex_key_op_tl {
4488      \cs_set_eq:cN{l_stex_notation_
4489        \l_stex_get_symbol_mod_str?\l_stex_get_symbol_name_str
4490        _op__cs}\l_stex_key_op_tl
4491    }
4492  }
4493 }

```

(End of definition for `\setnotation` and `\stex_notation_set_default:n`. These functions are documented on page 91.)

### \varnotation

```

4494 \stex_new_stylable_cmd:nnnn {varnotation} { s m 0{} m} {
4495   \stex_keys_set:nn{notation}{#3}
4496   \stex_get_var:n{#2}
4497   \str_set_eq:NN \l_stex_key_name_str \l_stex_get_symbol_name_str
4498   \stex_notation_parse:n{#4}

```

```

4499 \stex_if_check_terms:T{ \_stex_notation_check: }
4500 \_stex_vardecl_notation_macro:
4501 \IfBooleanTF#1{
4502   \_stex_notation_set_default:n{\l_stex_get_symbol_name_str}
4503 }{}
4504 \group_begin:
4505 \tl_set_eq:NN \thisvarname \l_stex_get_symbol_name_str
4506 \tl_clear:N \thisstyle
4507 \str_set_eq:NN\thisnotationvariant\l_stex_key_variant_str
4508 \def\thisnotation{
4509   \$\let\l_stex_current_symbol_str\thisvarname
4510   \def\comp{\_varcomp}\exp_args:Nne \use:nn{\l_stex_notation_macrocode_cs}{}
4511   \_stex_notation_make_args:
4512 }$
4513 }
4514 \stex_style_apply:
4515 \group_end:
4516 }{}

```

(End of definition for \varnotation. This function is documented on page 95.)

### \symdef

```

4517 \stex_keys_define:nnnn{symdef}{}{}{decl,notation}
4518
4519 \cs_new_protected:Nn \_stex_symdef_styledefs: {
4520   \tl_set:Nx \thisdecluri {\l_stex_current_module_str ? \l_stex_key_name_str}
4521   \tl_set_eq:NN \thisdeclname \l_stex_key_name_str
4522   \tl_set_eq:NN \thistype \l_stex_key_type_tl
4523   \tl_set_eq:NN \thisdefiniens \l_stex_key_def_tl
4524   \tl_set_eq:NN \thisargs \l_stex_key_args_str
4525   \tl_clear:N \thisstyle
4526   \str_set_eq:NN\thisnotationvariant\l_stex_key_variant_str
4527   \def\thisnotation{
4528     \$\let\l_stex_current_symbol_str\thisdecluri
4529     \def\comp{\_comp}\exp_args:Nne \use:nn{\l_stex_notation_macrocode_cs}{}
4530     \_stex_notation_make_args:
4531   }$
4532 }
4533 }
4534
4535 \stex_new_stylable_cmd:nnnn {symdef} { m 0{} m} {
4536   \stex_keys_set:nn{symdef}{#2}
4537   \str_set:Nx \l_stex_mroname_str { #1 }
4538   \stex_symdecl_top:n{#1}
4539   \stex_debug:nn{symdef}{Doing~\l_stex_current_module_str ? \l_stex_key_name_str}
4540   \str_set_eq:NN \l_stex_get_symbol_mod_str \l_stex_current_module_str
4541   \str_set_eq:NN \l_stex_get_symbol_name_str \l_stex_key_name_str
4542   \stex_notation_parse:n{#3}
4543   \stex_debug:nn{Here!}{\meaning\l_stex_notation_args_tl}
4544   \_stex_notation_check:
4545   \_stex_notation_add:
4546   \stex_if_do_html:T{
4547     \_stex_notation_do_html:n{\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
4548   }

```

```

4549 \stex_if_smsmode:F{
4550   \group_begin:
4551   \_stex_symdef_styledefs:
4552   \stex_style_apply:
4553   \group_end:
4554 }
4555 \stex_smsmode_do:
4556 }{}
4557
4558 \stex_deactivate_macro:Nn \symdef {module~environments}
4559 \stex_every_module:n {\stex_reactivate_macro:N \symdef}
4560 \stex_sms_allow_escape:N \symdef

```

(End of definition for `\symdef`. This function is documented on page 88.)

`\stex_do_default_notation_op:`

```

4561 \cs_new_protected:Nn \stex_do_default_notation: {
4562   \stex_do_default_notation_op:
4563   \tl_if_empty:NTF \l_stex_current_args_tl {
4564     \tl_clear:N \l__stex_notations_args_tl
4565   }{
4566     \__stex_notations_make_name:
4567     \tl_set_eq:NN \l_stex_get_symbol_args_tl \l_stex_current_args_tl
4568     \tl_set:Nx \l__stex_notations_args_tl {
4569       \stex_map_args:N \__stex_notations_augment_arg:nn
4570     }
4571     \tl_put_right:Nn \l_stex_default_notation {\comp{}}
4572     \seq_clear:N \l_tmpa_seq
4573     \int_step_inline:nn \l_stex_current_arity_str {
4574       \seq_put_right:Nn \l_tmpa_seq {#### ##1}
4575     }
4576     \tl_put_right:Nx \l_stex_default_notation {
4577       \seq_use:Nn \l_tmpa_seq {\mathpunct{\comp{},}}}
4578     \tl_put_right:Nn \l_stex_default_notation {\comp{}}
4579   }
4580   \tl_set:Nx \l_stex_default_notation {\STEXInternalNotation{}{0}{}{\l__stex_notations_args_}
4581     \exp_args:No \exp_not:n \l_stex_default_notation
4582   }
4583 }
4584 }
4585
4586 \cs_new:Nn \__stex_notations_augment_arg:nn {
4587   #1#2{0}{}
4588 }
4589
4590 \cs_new_protected:Nn \__stex_notations_make_name: {
4591   \exp_args:NNNNo \seq_set_split:Nnn \l_tmpa_seq ? \l_stex_current_symbol_str
4592   \seq_pop_right:NN \l_tmpa_seq \l__stex_notations_name_str
4593   \exp_args:NNNNo \seq_set_split:Nnn \l_tmpa_seq / \l__stex_notations_name_str
4594   \seq_pop_right:NN \l_tmpa_seq \l__stex_notations_name_str
4595 }
4596
4597 \cs_new_protected:Nn \stex_do_default_notation_op: {
4598   \__stex_notations_make_name:

```

```

4599   \tl_set:Nx \l_stex_default_notation {\exp_not:N \maincomp{ \exp_not:N \mathrm { \l_stex_no
4600 }

(End of definition for \stex_do_default_notation_op:. This function is documented on page ??.)
```

\STEXInternalNotation

```

4601 % 1: variant 2: operator precedence 3: intent 4: arguments 5: code 6: next
4602
4603 \cs_new_protected:Npn \STEXInternalNotation #1 #2 #3 #4 #5 #6 {
4604     \__stex_notations_process_notation:nnnnnn{#1}{#2}{#3}{#4}{#5}{#
4605         \l_stex_notations_code_tl
4606         #6
4607     }
4608 }
4609
4610 \cs_new_protected:Npn \__stex_notations_process_notation:nnnnnn #1 #2 #3 #4 {
4611     \tl_if_empty:nTF{#4} {
4612         \__stex_notations_simple:nnnnn{#1}{#2}{#3}
4613     }{
4614         \__stex_notations_complex:nnnnnn{#1}{#2}{#3}{#4}
4615     }
4616 }
4617
4618 \cs_new_protected:Nn \__stex_notations_simple:nnnnn {
4619     \stex_debug:nn{Notation~code}{\tl_to_str:n{#4}}
4620     \tl_set:Nn \l_stex_notations_code_tl {
4621         \cs_set:Npn \l_stex_notations_cs {
4622             \__stex_maybe_brackets:nn{#2}{#
4623                 \__stex_term_oms_or_omv:nnn{#1}{#3}{#4}
4624             }
4625         }
4626         \l_stex_notations_cs
4627     }
4628     \stex_debug:nn{Here:Notation}{\meaning \l_stex_notations_code_tl }
4629     #5
4630 }
4631
4632 \cs_new_protected:Nn \__stex_notations_complex:nnnnnn {
4633     \stex_debug:nn{Notation~code}{\tl_to_str:n{#5}}
4634     \int_zero:N \l_tmpa_int
4635     \tl_set:Nn \l_stex_notations_pre_tl {\cs_set_eq:NN \__stex_term_oma_or_omb:nnn \__stex_term_
4636     \tl_set:Nn \l_stex_notations_code_tl {
4637         \cs_generate_from_arg_count:NNnn \l_stex_notations_cs \cs_set:Npn \l_tmpa_int
4638         {
4639             \__stex_maybe_brackets:nn{#2}{#
4640                 \__stex_term_oma_or_omb:nnn{#1}{#3}{#
4641                     \bool_set_false:N \l_stex_brackets_dones_bool
4642                     #5
4643                 }
4644             }
4645         }
4646         \l_stex_notations_cs
4647     }
4648     \tl_set:Nn \l_stex_notations_after_tl{
```

```

4649   \exp_args:NNo
4650   \tl_put_left:Nn \l__stex_notations_code_tl \l__stex_notations_pre_tl
4651   \tl_put_left:Nx \l__stex_notations_code_tl {
4652     \int_set:Nn \l_tmpa_int {\int_use:N \l_tmpa_int}
4653   }
4654   \stex_debug:nn{Here:Notation}{\meaning \l__stex_notations_code_tl }
4655   #6
4656 }
4657 \__stex_notations_parse_notation_args:nnnw #4 \__stex_notations_args_end:
4658 }
4659
4660 \cs_new_protected:Npn \__stex_notations_parse_notation_args:nnnw #1 #2 #3 #4 #5 \__stex_no
4661   \tl_if_empty:nTF{#5} {
4662     \__stex_notations_add_last:nnnnn{#1}{#2}{#3}{#4}
4663   }{
4664     \__stex_notations_add_next:nnnnnn{#1}{#2}{#3}{#4}{#5}
4665   }
4666 }
4667
4668 \cs_new_protected:Nn \__stex_notations_add_next:nnnnnn {
4669   \__stex_notations_add:nnnnn{#1}{#2}{#3}{#4}{#6}
4670   \__stex_notations_parse_notation_args:nnnw #5 \__stex_notations_args_end:
4671 }
4672
4673 \cs_new_protected:Nn \__stex_notations_add_last:nnnnnn {
4674   \__stex_notations_add:nnnnn{#1}{#2}{#3}{#4}{#5}
4675   \l__stex_notations_after_tl
4676 }
4677
4678 \cs_new_protected:Nn \__stex_notations_add:nnnnnn {
4679   \int_incr:N \l_tmpa_int
4680   \str_case:nn{#2}{{
4681     i {
4682       \tl_put_right:Nn \l__stex_notations_code_tl {
4683         {\__stex_term_arg:nnnnn{#1}{#2}{#3}{#4}{#5}}
4684       }
4685     }
4686     b {
4687       \tl_set:Nn \l__stex_notations_pre_tl {
4688         \cs_set_eq:NN \__stex_term_oma_or_omb:nnn \__stex_term_omb:nnn
4689       }
4690       \tl_put_right:Nn \l__stex_notations_code_tl {
4691         {\__stex_term_arg:nnnnn{#1}{#2}{#3}{#4}{#5}}
4692       }
4693     }
4694     a {
4695       \tl_put_right:Nn \l__stex_notations_code_tl {
4696         {\__stex_term_arg_aB:nnnnn{#1}{#2}{#3}{#4}{#5}}
4697       }
4698     }
4699     B {
4700       \tl_set:Nn \l__stex_notations_pre_tl {
4701         \cs_set_eq:NN \__stex_term_oma_or_omb:nnn \__stex_term_omb:nnn
4702       }

```

```

4703     \tl_put_right:Nn \l__stex_notations_code_tl {
4704         {\_stex_term_arg_aB:nnnnn{#1}{#2}{#3}{#4}{#5}}
4705     }
4706 }
4707 }
4708 }

```

(End of definition for `\STEXInternalNotation`. This function is documented on page ??.)

## a/B-mode argument handling

### `\argsep`

```

4709 \cs_new_protected:Nn \__stex_notations_check_aB_arg:Nn {
4710     \exp_args:Ne \cs_if_eq:NNF {\tl_head:n{#2}}
4711     \_stex_term_arg_aB:nnnnn {
4712         \msg_error:nnx{\stex}{error/assocarg}{\tl_to_str:n{#1}}
4713     }
4714 }
4715
4716 \cs_new_protected:Npn \argsep #1 #2 {
4717     \__stex_notations_check_aB_arg:Nn\argsep{#1}
4718     \stex_pseudogroup_with:nn{\_stex_term_do_aB_clist:} {
4719         \tl_set:Nn \_stex_term_do_aB_clist: {
4720             \seq_use:Nn \l_stex_aB_args_seq {#2}
4721         }
4722         #1
4723     }
4724 }

```

(End of definition for `\argsep`. This function is documented on page 92.)

### `\argmap`

```

4725 \cs_new_protected:Npn \argmap #1 #2 #3 {
4726     \__stex_notations_check_aB_arg:Nn\argmap{#1}
4727     \stex_pseudogroup_with:nn{
4728         \_stex_term_do_aB_clist:
4729         \__stex_notations_map_cs:
4730     }{
4731         \cs_set:Npn \__stex_notations_map_cs: ##1 { #2 }
4732         \tl_set:Nn \_stex_term_do_aB_clist: {
4733             \seq_clear:N \l_tmpa_seq
4734             \seq_map_inline:Nn \l_stex_aB_args_seq {
4735                 \tl_if_eq:nnTF{##1}{\ellipses} {
4736                     \seq_put_right:Nn \l_tmpa_seq \ellipses
4737                 }{
4738                     \seq_put_right:Nx \l_tmpa_seq {
4739                         \exp_after:wN \exp_not:n \exp_after:wN { \__stex_notations_map_cs: {##1} }
4740                     }
4741                 }
4742             }
4743             \seq_set_eq:NN \l_stex_aB_args_seq \l_tmpa_seq
4744             \seq_use:Nn \l_stex_aB_args_seq {#3}
4745     }
4746     #1

```

```
4747     }
4748 }
```

(End of definition for \argarraymap. This function is documented on page 92.)

### \argarraymap

```
4749 \int_new:N \l__stex_notations_clist_count_int
4750 \cs_new_protected:Npn \argarraymap #1 #2 #3 #4 {
4751   \__stex_notations_check_aB_arg:Nn\argarraymap{#1}
4752   \stex_pseudogroup_with:nnf
4753   \__stex_term_do_aB_clist:
4754   \__stex_notations_map_cs:
4755 }{
4756   \cs_set:Npn \__stex_notations_map_cs: ##1 { #3 }
4757   \int_set:Nn \l__stex_notations_clist_count_int {\exp_args:No\clist_count:n{\tl_to_str:n}
4758   \tl_set:Nn \__stex_term_do_aB_clist: {
4759     \tl_clear:N \l_tmpa_tl
4760     \int_zero:N \l_tmpa_int
4761     \seq_map_inline:Nn \l_stex_aB_args_seq {
4762       \int_incr:N \l_tmpa_int
4763       \int_compare:nNnT \l_tmpa_int > \l__stex_notations_clist_count_int {
4764         \int_set:Nn \l_tmpa_int 1
4765       }
4766       \tl_put_right:Nx \l_tmpa_tl {
4767         \exp_after:wN \exp_not:n \exp_after:wN { \__stex_notations_map_cs: {##1} }
4768         \clist_item:nn{#4}\l_tmpa_int
4769       }
4770     }
4771     \seq_set_eq:NN \l_stex_aB_args_seq \l_tmpa_seq
4772     \begin{array}{#2}
4773       \l_tmpa_tl
4774     \end{array}
4775   }
4776   #1
4777 }
4778 }
```

(End of definition for \argarraymap. This function is documented on page 92.)

### 13.8.3 Variables

```
4779 <@=stex_vars>
```

### \vardef

```
4780 \tl_new:N \l_stex_variables_prop
4781 \bool_new:N \l__stex_vars_bind_bool
4782 \cs_new_protected:Nn \stex_variable:nnnnnnnN {}
4783 \stex_keys_define:nnnn{vardef} {
4784   \bool_set_false:N \l__stex_vars_bind_bool
4785 }{
4786   bind .bool_set:N = \l__stex_vars_bind_bool
4787 }{symdef}
4788
4789 \stex_new_stylable_cmd:nnnn {vardef} { m 0{} m} {
4790   \stex_keys_set:nn{vardef}{#2}
```

```

4791 \str_set:Nx \l_stex_macroname_str { #1 }
4792 \str_if_empty:NT \l_stex_key_name_str {
4793   \str_set:Nx \l_stex_key_name_str { #1 }
4794 }
4795
4796 \stex_symdecl_do:
4797 \_stex_symdecl_check_terms:
4798 \_\_stex_vars\_add:
4799 \_\_stex_vars\_macro:
4800 \stex_if_do_html:T \_\_stex_vars_html:
4801
4802 \int_set:Nn \l_stex_get_symbol_arity_int {\l_stex_get_symbol_arity_int}
4803 \stex_debug:nn{vardef}{Doing~\l_stex_key_name_str}
4804 \tl_set_eq:NN \l_stex_get_symbol_return_tl \l_stex_key_return_tl
4805 \stex_notation_parse:n{#3}
4806 \stex_if_check_terms:T{ \_stex_notation_check: }
4807 \_stex_vardecl_notation_macro:
4808 \stex_if_do_html:T {
4809   \def\comp{\_varcomp}
4810   \_stex_notation_do_html:n \l_stex_key_name_str
4811 }
4812 \group_begin:
4813 \tl_set_eq:NN \thisvarname \l_stex_key_name_str
4814 \tl_clear:N \thisstyle
4815 \str_set_eq:NN \thisnotationvariant \l_stex_key_variant_str
4816 \def\thisnotation{
4817   \$\let\l_stex_current_symbol_str\thisvarname
4818   \def\comp{\_varcomp}\exp_args:Nne \use:nn{\l_stex_notation_macrocode_cs{}}
4819   \_stex_notation_make_args:
4820 }
4821 }
4822 \stex_style_apply:
4823 \group_end:\ignorespaces
4824 }{}
4825
4826 \cs_new_protected:Nn \_\_stex_vars\_add: {
4827   \exp_args:NNNo \exp_args:NNnx
4828   \prop_put:Nnn \l_stex_variables_prop \l_stex_key_name_str {
4829     {\l_stex_macroname_str}
4830     {\l_stex_key_name_str}
4831     {\int_use:N \l_stex_get_symbol_arity_int}
4832     {\l_stex_get_symbol_args_tl}
4833     {\exp_args:No \exp_not:n \l_stex_key_def_tl}
4834     {\exp_args:No \exp_not:n \l_stex_key_type_tl}
4835     {\exp_args:No \exp_not:n \l_stex_key_return_tl}
4836     \stex_invoke_symbol:
4837   }
4838 }
4839
4840 \cs_new_protected:Nn \_\_stex_vars\_macro: {
4841   \tl_set:cx{\l_stex_macroname_str} {
4842     \_stex_invoke_variable:nnnnnnN
4843     {\l_stex_key_name_str}
4844     {\int_use:N \l_stex_get_symbol_arity_int}

```

```

4845     {\l_stex_get_symbol_args_tl}
4846     {\exp_args:No \exp_not:n \l_stex_key_def_tl}
4847     {\exp_args:No \exp_not:n \l_stex_key_type_tl}
4848     {\exp_args:No \exp_not:n \l_stex_key_return_tl}
4849     \stex_invoke_symbol:
4850   }
4851 }
4852
4853 \cs_new_protected:Nn \__stex_vars_html: {
4854   \stex_if_do_html:T {
4855     \hbox\bgroup\exp_args:Ne \stex_annotation_invisible:nn {
4856       \shtml:vardef = {\l_stex_key_name_str},
4857       \shtml:args = {\l_stex_key_args_str}
4858       \str_if_empty:NF \l_stex_macroname_str {
4859         \shtml:macroname={\l_stex_macroname_str}
4860       }
4861       \str_if_empty:NF \l_stex_key_assoc_str {
4862         \shtml:assocotype={\l_stex_key_assoc_str}
4863       }
4864       \str_if_empty:NF \l_stex_key_role_str {
4865         \shtml:role={\l_stex_key_role_str}
4866       }
4867       \str_if_empty:NF \l_stex_key_reorder_str {
4868         \shtml:reorderargs={\l_stex_key_reorder_str}
4869       }
4870     \bool_if:NT \l_stex_vars_bind_bool {
4871       \shtml:bind={}
4872     }
4873   }{
4874     \stex_annotation_force_break:n{
4875       \bool_set_true:N \stex_in_invisible_html_bool
4876       \tl_if_empty:NF \l_stex_key_type_tl {
4877         \stex_annotation:nn{\shtml:type={}}{$\l_stex_key_type_tl$}
4878       }
4879       \tl_if_empty:NF \l_stex_key_def_tl {
4880         \stex_annotation:nn{\shtml:definiens={}}{$\l_stex_key_def_tl$}
4881       }
4882       \tl_if_empty:NF \l_stex_key_return_tl{
4883         \exp_args:Nno \use:n{
4884           \cs_generate_from_arg_count:NNnn \l__stex_vars_cs
4885           \cs_set:Npn \l_stex_get_symbol_arity_int} \l_stex_key_return_tl
4886           \tl_set:Nx \l__stex_vars_args_tl {\l_stex_map_args:N \stex_return_args:nn}
4887           $\stex_annotation:nn{\shtml:returnrtype={}}{%
4888             \exp_after:wN \l__stex_vars_cs \l__stex_vars_args_tl!}$}
4889       }
4890       \tl_if_empty:NF \l_stex_key_argtypes_clist {
4891         \stex_annotation:nn{\shtml:argtypes={}}{
4892           \stex_annotation_force_break:n{
4893             \clist_map_inline:Nn \l_stex_key_argtypes_clist {
4894               $\stex_annotation:nn{\shtml:type={}}{\#\#1}$}
4895             }
4896           }
4897         }
4898   }

```

```

4899      }
4900    }\egroup
4901  }
4902 }

```

(End of definition for `\vardef`. This function is documented on page [95](#).)

#### `\_stex_vardecl_notation_macro:`

```

4903 \cs_new_protected:Nn \_stex_vardecl_notation_macro: {
4904   \tl_set_eq:cN {l_stex_notation_
4905     \l_stex_key_name_str _ 
4906     \l_stex_key_variant_str _cs
4907   }\l_stex_notation_macrocode_cs
4908   \cs_if_exist:cF {l_stex_notation_\l_stex_key_name_str __cs}{
4909     \tl_set_eq:cN{l_stex_notation_\l_stex_key_name_str __cs}
4910     \l_stex_notation_macrocode_cs
4911   }
4912   \tl_if_empty:NF \l_stex_key_op_tl {
4913     \tl_set_eq:cN {l_stex_notation_\l_stex_key_name_str _op_
4914     \l_stex_key_variant_str _cs}\l_stex_key_op_tl
4915     \cs_if_exist:cF {l_stex_notation_\l_stex_key_name_str _op__cs}{
4916       \cs_set_eq:cN{l_stex_notation_\l_stex_key_name_str _op__cs}
4917       \l_stex_key_op_tl
4918     }
4919   }
4920 }

```

(End of definition for `\_stex_vardecl_notation_macro:`. This function is documented on page [125](#).)

#### `\stex_get_symbol_or_var:n`

##### `\stex_get_var:n`

```

4921 \cs_new_protected:Nn \__stex_vars_set_vars:nnnnnnN {
4922   \stex_debug:nn{symbols}{Variable~#1~found}
4923   \cs_set:Npn \_stex_variable:nnnnnnnN ##1 ##2 ##3 ##4 ##5 ##6 ##7 ##8 {}
4924   \str_clear:N \l_stex_get_symbol_mod_str
4925   \str_set:Nn \l_stex_get_symbol_name_str {#1}
4926   \int_set:Nn \l_stex_get_symbol_arity_int {#2}
4927   \tl_set:Nn \l_stex_get_symbol_args_tl {#3}
4928   \tl_set:Nn \l_stex_get_symbol_def_tl {#4}
4929   \tl_set:Nn \l_stex_get_symbol_type_tl {#5}
4930   \tl_set:Nn \l_stex_get_symbol_return_tl {#6}
4931   \tl_set:Nn \l_stex_get_symbol_invoke_cs {#7}
4932 }
4933
4934 \cs_new_protected:Nn \__stex_vars_get_var:n {
4935   \prop_map_inline:Nn \l_stex_variables_prop {
4936     \__stex_vars_check_var:nnnnnnnnN {#1} ##2
4937   }
4938 }
4939
4940 \cs_new_protected:Nn \__stex_vars_check_var:nnnnnnnnN {
4941   \str_if_eq:nnTF{#1}{#2}{

4942     \prop_map_break:n{\__stex_vars_set_vars:nnnnnnN {#3}{#4}{#5}{#6}{#7}{#8}{#9}}
4943   }{
4944     \str_if_eq:nnT{#1}{#3}{

4945       \prop_map_break:n{\__stex_vars_set_vars:nnnnnnN {#3}{#4}{#5}{#6}{#7}{#8}{#9}}
4946     }
4947   }
4948 }

```

```

4946     }
4947   }
4948 }
4949
4950 \cs_new_protected:Nn \stex_get_var:n {
4951   \str_clear:N \l_stex_get_symbol_name_str
4952   \__stex_vars_get_var:n{#1}
4953   \str_if_empty:NT \l_stex_get_symbol_name_str {
4954     \msg_error:nnn{stex}{error/unknownsymbol}{#1}
4955   }
4956 }
4957
4958 \cs_new_protected:Nn \stex_get_symbol_or_var:n {
4959   \str_clear:N \l_stex_get_symbol_name_str
4960   \__stex_vars_get_var:n{#1}
4961   \str_if_empty:NT \l_stex_get_symbol_name_str {
4962     \stex_debug:nn{symbols}{No~variable~#1~found}
4963     \stex_get_symbol:n{#1}
4964   }
4965 }

```

(End of definition for `\stex_get_symbol_or_var:n` and `\stex_get_var:n`. These functions are documented on page 125.)

### \svar

```

4966 \NewDocumentCommand \svar {O{} m}{
4967   \group_begin:
4968   \tl_if_empty:nTF{#1} {
4969     \str_set:Nn \l_stex_current_symbol_str {#2}
4970   }{
4971     \str_set:Nn \l_stex_current_symbol_str {#1}
4972   }
4973   \bool_if:NTF \l_stex_allow_semantic_bool {
4974     \tl_clear:N \l_stex_current_term_tl
4975     \stex_term_omv:nnn{}{}{\l_varcomp{#2}}
4976   }{
4977     \msg_error:nnx{stex}{error/notallowed}{Variable}{\l_stex_current_symbol_str}
4978   }
4979   \group_end:
4980 }

```

(End of definition for `\svar`. This function is documented on page 95.)

## 13.8.4 Sequences

```
4981 <@=stex_seqs>
```

### \varseq

```

4982 \stex_new_stylable_cmd:nnnn {varseq}{m O{} m m} {
4983   \stex_keys_set:nn{symdef}{#2}
4984   \str_set:Nx \l_stex_macroname_str { #1 }
4985   \str_if_empty:NT \l_stex_key_name_str {
4986     \str_set:Nx \l_stex_key_name_str { #1 }
4987   }
4988   \str_if_empty:NT \l_stex_key_args_str {

```

```

4989     \str_set:Nn \l_stex_key_args_str {1}
4990 }
4991 \stex_symdecl_do:
4992
4993 \tl_set_eq:NN \l_stex_get_symbol_return_tl \l_stex_key_return_tl
4994 \clist_set:Nn \l__stex_seqs_range_clist {#3}
4995 \tl_if_empty:NTF \l_stex_key_op_tl {
4996     \stex_notation_parse:n{#4}
4997     \tl_clear:N \l_stex_key_op_tl
4998 }{
4999     \stex_notation_parse:n{#4}
5000 }
5001 \stex_if_do_html:T \__stex_seqs_html:
5002 \stex_if_check_terms:T \__stex_seqs_check_terms:
5003 \__stex_seqs_add:
5004 \__stex_seqs_macro:
5005 \stex_if_check_terms:T \_stex_notation_check:
5006 \_stex_vardecl_notation_macro:
5007 \group_begin:
5008 \tl_set_eq:NN \thisvarname \l_stex_key_name_str
5009 \tl_clear:N \thisstyle
5010 \str_set_eq:NN\thisnotationvariant\l_stex_key_variant_str
5011 \def\thisnotation{
5012     \$\let\l_stex_current_symbol_str\thisvarname
5013         \def\comp{\_varcomp}\exp_args:Nne \use:nn{\l_stex_notation_macrocode_{}}{
5014             \__stex_seqs_make_args:
5015         }$}
5016     }
5017 \stex_style_apply:
5018 \group_end:\ignorespaces
5019 }{}}
5020
5021 \cs_new_protected:Nn \__stex_seqs_add: {
5022     \exp_args:NNNo \exp_args:NNnx
5023     \prop_put:Nnn \l_stex_variables_prop \l_stex_key_name_str {
5024         {\l_stex_mroname_str}
5025         {\l_stex_key_name_str}
5026         {\int_use:N \l_stex_get_symbol_arity_int}
5027         {\l_stex_get_symbol_args_tl}
5028         {\exp_args:No \exp_not:n \l_stex_key_def_tl}
5029         {\exp_args:No \exp_not:n \l__stex_seqs_range_clist}
5030         {\exp_args:No \exp_not:n \l_stex_key_return_tl}
5031         \stex_invoke_sequence:
5032     }
5033 }
5034
5035 \cs_new_protected:Nn \__stex_seqs_macro: {
5036     \tl_set:cx{\l_stex_mroname_str}{%
5037         \_stex_invoke_variable:nnnnnnN
5038         {\l_stex_key_name_str}
5039         {\int_use:N \l_stex_get_symbol_arity_int}
5040         {\l_stex_get_symbol_args_tl}
5041         {\exp_args:No \exp_not:n \l_stex_key_def_tl}
5042         {\exp_args:No \exp_not:n \l__stex_seqs_range_clist}

```

```

5043     {\exp_args:No \exp_not:n \l_stex_key_return_tl}
5044     \stex_invoke_sequence:
5045   }
5046 }
5047
5048 \cs_new_protected:Nn \__stex_seqs_make_args: { \TODO }
5049 \cs_new_protected:Nn \__stex_seqs_check_terms: { \TODO }
5050
5051 \cs_new_protected:Nn \__stex_seqs_html: {
5052     \exp_args:Ne \stex_annotation_invisible:nn {
5053         shtml:varseq = {\l_stex_key_name_str},
5054         shtml:args = {\l_stex_key_args_str}
5055         \str_if_empty:NF \l_stex_macroname_str {,
5056             shtml:macroname={\l_stex_macroname_str}
5057         }
5058         \str_if_empty:NF \l_stex_key_assoc_str {,
5059             shtml:assoctype={\l_stex_key_assoc_str}
5060         }
5061         \str_if_empty:NF \l_stex_key_role_str {,
5062             shtml:role={\l_stex_key_role_str}
5063         }
5064         \str_if_empty:NF \l_stex_key_reorder_str {,
5065             shtml:reorderargs={\l_stex_key_reorder_str}
5066         }
5067     }{\hbox\bgroup
5068         \stex_annotation_force_break:n{
5069             \tl_if_empty:NF \l_stex_key_type_tl {
5070                 \stex_annotation:nn{shtml:type={}{}{$\l_stex_key_type_tl$}}
5071             }
5072             \tl_if_empty:NF \l_stex_key_def_tl {
5073                 \stex_annotation:nn{shtml:definiens={}{}{$\l_stex_key_def_tl$}}
5074             }
5075             \tl_if_empty:NF \l_stex_key_return_tl{
5076                 \exp_args:Nno \use:n{
5077                     \cs_generate_from_arg_count:NNnn \l__stex_seqs_cs
5078                     \cs_set:Npn \l_stex_get_symbol_arity_int} \l_stex_key_return_tl
5079                     \tl_set:Nx \l__stex_seqs_args_tl {\l_stex_map_args:N \stex_return_args:nn}
5080                     \stex_annotation:nn{shtml:returntype={}{}{
5081                         $\exp_after:wN \l__stex_seqs_cs \l__stex_seqs_args_tl!$}
5082                 }
5083                 \tl_if_empty:NF \l_stex_key_argtypes_clist {
5084                     \stex_annotation:nn{shtml:argtypes={}{}{
5085                         \stex_annotation_force_break:n{
5086                             \clist_map_inline:Nn \l_stex_key_argtypes_clist {
5087                                 \stex_annotation:nn{shtml:type={}{}{$##1$}}
5088                             }
5089                         }
5090                     }
5091                 }
5092             }
5093         \egroup}
5094 }

```

(End of definition for `\varseq`. This function is documented on page 95.)

```

\stex_invoke_sequence:
5095 \cs_new_protected:Nn \stex_invoke_sequence: {
5096   \peek_charcode_remove:NTF ! {
5097     \peek_charcode:NTF [ \__stex_seqs_do_op:w { \__stex_seqs_do_op:w [] } ]
5098   }\__stex_seqs_do_first:
5099 }
5100
5101 \cs_new_protected:Npn \__stex_seqs_do_op:w [#1] {
5102   \cs_if_exist:cTF {l_stex_notation_\l_stex_current_symbol_str _op_#1_cs} {
5103     \stex_maybe_brackets:n{ \neginfprefc}{}
5104     \stex_term_oms_or_omv:n{ \neginfprefc{#1}{}}
5105     {\use:c{l_stex_notation_\l_stex_current_symbol_str _op_#1_cs}}
5106   }
5107   \group_end:
5108 }{
5109   \__stex_seqs_get_index_notation:n{ \neginfprefc{#1}}
5110   \peek_charcode:NTF [ \__stex_seqs_doop_range:w { \__stex_seqs_doop_range:w[] } ]
5111 }
5112
5113 \cs_new_protected:Npn \__stex_seqs_doop_range:w [#1] {
5114   \bool_set_true:N \l_stex_allow_semantic_bool
5115   \clist_clear:N \l__stex_seqs_clist
5116   \clist_map_function:NN \l_stex_current_type_tl \__stex_seqs_doop_arg:n
5117     \stex_annotation:n{ \neginfprefc{#1}{OMV, \l_stex_current_symbol_str, \neginfprefc{#1}{}}}
5118   }
5119   \l__stex_seqs_clist
5120 }
5121 \group_end:
5122
5123 \__stex_seqs_doop_arg:n {
5124   \tl_if_eq:nnTF{ \neginfprefc{#1}{\ellipses}}{
5125     \clist_put_right:Nn \l__stex_seqs_clist {
5126       \ellipses
5127     }
5128   }{
5129     \clist_put_right:Nn \l__stex_seqs_clist {
5130       \exp_args:No \str_if_eq:nnTF \l_stex_current_arity_str {1} {
5131         \group_begin:
5132           \l__stex_seqs_cs \group_end: { \neginfprefc{#1}{\l__stex_seqs_clist} }
5133         \group_end:
5134           \l__stex_seqs_cs \group_end: #1
5135         \group_begin:
5136           \l__stex_seqs_cs \group_end: #1
5137         \group_end:
5138       }
5139     }
5140   }
5141 }
5142
5143 \__stex_seqs_doop_arg:n {
5144   \__stex_seqs_get_index_notation:n {
5145

```

```

5148 \cs_if_exist:cTF {l_stex_notation_\l_stex_current_symbol_str _#1_cs} {
5149   \cs_set_eq:Nc \l_stex_seqs_cs {l_stex_notation_\l_stex_current_symbol_str _#1_cs}
5150 }
5151   \stex_do_default_notation:
5152   \cs_set_eq:NN \l_stex_seqs_cs \l_stex_default_notation
5153 }
5154 }
5155
5156
5157 \cs_new:Nn \__stex_seqs_do_first_arg:n {{\exp_not:n{## #1}}}
5158
5159 \cs_new_protected:Nn \__stex_seqs_do_first: {
5160   \exp_args:Nnx \use:nn{
5161     \cs_generate_from_arg_count:NNnn \l_stex_seqs_cs \cs_set:Npn
5162     \l_stex_current_arity_str }{{
5163       \tl_set:Nn \exp_not:N \l_stex_seqs_first_args_tl {
5164         \int_step_function:nN \l_stex_current_arity_str \__stex_seqs_do_first_arg:n
5165       }
5166       \exp_not:N \__stex_seqs_do_first_next:
5167     }}
5168   \l_stex_seqs_cs
5169 }
5170
5171 \cs_new_protected:Nn \__stex_seqs_do_first_next: {
5172   \peekCharCode_remove:NTF ! {
5173     \peekCharCode:NTF [ \__stex_seqs_do_one:w {\__stex_seqs_do_one:w []}]
5174   }{
5175     \peekCharCode:NTF [ \__stex_seqs_do_all:w {\__stex_seqs_do_all:w []}]
5176   }
5177 }
5178
5179 \cs_new_protected:Npn \__stex_seqs_do_one:w [#1] {
5180   \__stex_seqs_get_index_notation:n{#1}
5181   \stex_debug:nn{HERE~seq~one}{\meaning\l_stex_seqs_cs^\J\meaning\l_stex_seqs_first_args_t
5182   \exp_args:Nno\use:nn{\l_stex_seqs_cs\group_end:}\l_stex_seqs_first_args_tl
5183 }
5184
5185 \cs_new_protected:Npn \__stex_seqs_do_all:w [#1] {
5186   \stex_debug:nn{HERE~seq~all}{\meaning\l_stex_seqs_first_args_tl}
5187   \exp_args:Nno\use:nn{\l_stex_invoke_notation:w [#1]}\l_stex_seqs_first_args_tl
5188 }

```

(End of definition for \stex\_invoke\_sequence:. This function is documented on page ??.)

### \seqmap

```

5189 \cs_new_protected:Npn \seqmap #1 #2 {
5190   \symuse{Metatheory?sequence-expression}{\seqmap{#1}{#2}}%\l_tmpa_tl {#2}
5191 }

```

(End of definition for \seqmap. This function is documented on page 96.)

## 13.8.5 Expressions

```
5192 <@=stex_expr>
```

Various variables:

```

5193 \bool_new:N \l_stex_allow_semantic_bool
5194 \bool_set_true:N \l_stex_allow_semantic_bool
5195 \tl_new:N \l_stex_current_term_tl
5196 \tl_set:Nn \l_stex_every_symbol_tl {
5197   \bool_set_false:N \l_stex_allow_semantic_bool
5198 }

```

`\_stex_next_symbol:n`

```

5199 \tl_new:N \l__stex_expr_reset_tl
5200 \cs_new_protected:Nn \_stex_next_symbol:n {
5201   \tl_set:Nx \l_stex_every_symbol_tl {
5202     \tl_gset:Nn \exp_not:N \l__stex_expr_reset_tl {
5203       \tl_set:Nn \exp_not:N \l_stex_every_symbol_tl {
5204         \exp_args:No \exp_not:n \l_stex_every_symbol_tl
5205       }
5206       \tl_gset:Nn \exp_not:N \l__stex_expr_reset_tl {
5207         \exp_args:No \exp_not:n \l__stex_expr_reset_tl
5208       }
5209     }
5210     \tl_set:Nn \exp_not:N \l_stex_every_symbol_tl {
5211       \exp_args:No \exp_not:n \l_stex_every_symbol_tl
5212     }
5213     \exp_not:n{ \aftergroup \l__stex_expr_reset_tl }
5214     \exp_not:N \l_stex_every_symbol_tl
5215     \exp_not:n{ #1 }
5216   }
5217 }
5218 \cs_generate_variant:Nn \_stex_next_symbol:n {e}

```

(End of definition for `\_stex_next_symbol:n`. This function is documented on page ??.)

### \STEXinvisible

```

5219 \cs_new_protected:Npn \STEXinvisible #1 {
5220   \stex_annotation_invisible:n { #1 }
5221 }

```

(End of definition for `\STEXinvisible`. This function is documented on page 81.)

## Invoking Semantic Macros

### `\_stex_invoke_symbol:nnnnnnnN`

```

5222 \cs_new_protected:Nn \_stex_invoke_symbol:nnnnnnnN {
5223   \bool_if:NTF \l_stex_allow_semantic_bool{
5224     \stex_if_html_backend:T{\ifvmode\indent\fi}
5225     \__stex_expr_setup:nnnnnn{\_comp}{#1?#2}{#3}{#4}{#7}{#6}
5226     \cs_set_eq:NN \_stex_term_oms_or_omv:nnn \_stex_term_oms:nnn
5227     \tl_put_right:Nn \l_stex_current_redo_tl{
5228       \cs_set_eq:NN \_stex_term_oms_or_omv:nnn \_stex_term_oms:nnn
5229     }
5230     #8
5231   }{
5232     \msg_error:nnxx{stex}{error/notallowed}{#1?#2}{\l_stex_current_symbol_str}
5233   }
5234 }

```

```

5235 \cs_generate_variant:Nn \stex_invoke_symbol:n{nnnnnnnnN}{ooooooN}
5236
5237 \cs_new_protected:Nn \__stex_expr_setup:n {
5238     \group_begin:
5239     \tl_clear:N \l_stex_return_notation_tl
5240     \tl_set:Nn \l_stex_current_redo_tl {
5241         \let \this \stex_current_this:
5242         \def\comp{\#1}
5243         \def\maincomp{\comp}
5244         \str_set:Nn \l_stex_current_symbol_str {\#2}
5245         \str_set:Nn \l_stex_current_arity_str{ #3 }
5246         \tl_set:Nn \l_stex_current_args_tl{ #4 }
5247         \tl_set:Nn \l_stex_current_return_tl{ #5 }
5248         \tl_set:Nn \l_stex_current_type_tl{ #6 }
5249         \tl_clear:N \l_stex_current_term_tl
5250     }
5251     \tl_put_right:Nx \l_stex_current_redo_tl {
5252         \exp_args:No \exp_not:n \l_stex_every_symbol_tl
5253     }
5254     \l_stex_current_redo_tl
5255 }
```

(End of definition for `\stex_invoke_symbol:n{nnnnnnnnN}`. This function is documented on page ??.)

`\stex_invoke_variable:n{nnnnnn}`

```

5256 \cs_new_protected:Nn \stex_invoke_variable:n{nnnnnnNN} {
5257     \bool_if:NTF \l_stex_allow_semantic_bool{
5258         \stex_if_html_backend:T{\ifvmode\indent\fi}
5259         \__stex_expr_setup:n{nnnnnn{\_varcomp}{#1}{#2}{#3}{#6}{#5}}
5260         \cs_set_eq:NN \stex_term_oms_or_omv:n{nnn} \stex_term_omv:n{nnn}
5261         \tl_put_right:Nn \l_stex_current_redo_tl {
5262             \cs_set_eq:NN \stex_term_oms_or_omv:n{nnn} \stex_term_omv:n{nnn}
5263         }
5264         #7
5265     }{
5266         \msg_error:n{nnxx{stex}{error/notallowed}{#1}{\l_stex_current_symbol_str}}
5267     }
5268 }
```

(End of definition for `\stex_invoke_variable:n{nnnnnnN}`. This function is documented on page ??.)

`\symuse`

```

5269 \cs_new_protected:Npn \symuse #1 {
5270     \stex_get_symbol:n{\#1}
5271     \exp_args:Nno \use:n { \stex_invoke_symbol:ooooooN
5272     \l_stex_get_symbol_mod_str
5273     \l_stex_get_symbol_name_str
5274     {\int_use:N \l_stex_get_symbol_arity_int}
5275     \l_stex_get_symbol_args_tl
5276     \l_stex_get_symbol_def_tl
5277     \l_stex_get_symbol_type_tl
5278     \l_stex_get_symbol_return_tl}
5279     \l_stex_get_symbol_invoke_cs
5280 }
```

(End of definition for `\symuse`. This function is documented on page 89.)

**\stex\_invoke\_symbol:** Top-Level: Check whether text/math mode or custom notation, whether delimited by !, return code etc.

```

5281 \cs_new_protected:Nn \stex_invoke_symbol: {
5282   \stex_debug:nn{expressions}{Invoking-\l_stex_current_symbol_str}
5283   \mode_if_math:TF \__stex_expr_invoke_math: \__stex_expr_invoke_text:
5284 }
5285
5286 \cs_new_protected:Nn \__stex_expr_invoke_text: {
5287   \stex_debug:nn{expressions}{text-mode}
5288   \peek_charcode_remove:NTF ! \__stex_expr_invoke_op_custom:n \__stex_expr_invoke_custom:n
5289 }
5290
5291 \cs_new_protected:Nn \__stex_expr_invoke_math: {
5292   \stex_debug:nn{expressions}{math-mode}
5293   \peek_charcode_remove:NTF !
5294     % operator
5295     \peek_charcode_remove:NTF * \__stex_expr_invoke_op_custom:n {
5296       % op notation
5297       \peek_charcode:NTF [ \__stex_expr_invoke_op_notation:w {
5298         \__stex_expr_invoke_op_notation:w []
5299       }
5300     }
5301   }{
5302     \peek_charcode_remove:NTF * \__stex_expr_invoke_custom:n {
5303       % normal
5304       \peek_charcode:NTF [ \__stex_invoke_notation:w {
5305         \__stex_invoke_notation:w []
5306       }
5307     }
5308   }
5309 }
```

Notations:

```

5310 \cs_new_protected:Npn \stex_invoke_notation:w [#1] {
5311   \stex_debug:nn{expressions}{using-notat~#1-for~\l_stex_current_symbol_str}
5312   \cs_if_exist:cTF{\l_stex_notation_\l_stex_current_symbol_str _#1_cs} {
5313     \tl_if_empty:NTF \l_stex_current_return_tl {
5314       \stex_debug:nn{expressions}{return~empty}
5315       \use:c{\l_stex_notation_\l_stex_current_symbol_str _#1_cs}{\group_end:\stex_eat_exclam}
5316     }{
5317       \stex_debug:nn{expressions}{return?}
5318       \exp_after:wN\exp_after:wN\exp_after:wN
5319       \__stex_expr_invoke_return_maybe:n
5320       \exp_after:wN\exp_after:wN\exp_after:wN
5321       {\cs:w \l_stex_notation_\l_stex_current_symbol_str _#1_cs \cs_end: {}}
5322     }
5323   }{
5324     \stex_do_default_notation:
5325     \tl_if_empty:NTF \l_stex_current_return_tl {
5326       \l_stex_default_notation{\group_end:\stex_eat_exclamation_point:}
5327     }{
5328       \exp_after:wN
```

```

5329     \__stex_expr_invoke_return_maybe:n
5330     \exp_after:wN
5331     {\l_stex_default_notation {}}
5332   }
5333 }
5334 }
5335
5336 \cs_new_protected:Npn \__stex_expr_invoke_op_notation:w [#1] {
5337   \stex_debug:nn{expressions}{op~notation~for~\l_stex_current_symbol_str}
5338   \cs_if_exist:cTF{\l_stex_notation_\l_stex_current_symbol_str _op_#1_cs}{
5339     \__stex_maybe_brackets:nn{\neginfpref}{%
5340       \__stex_term_oms_or_omv:nnn{#1}{}
5341       {\use:c{\l_stex_notation_\l_stex_current_symbol_str _op_#1_cs}}%
5342     }
5343     \group_end:
5344   }{
5345     \int_compare:nNnTF \l_stex_current_arity_str = 0 {
5346       \tl_clear:N \l_stex_current_return_tl
5347       \__stex_invoke_notation:w [#1]
5348     }{
5349       \stex_do_default_notation_op:
5350       \__stex_maybe_brackets:nn{\neginfpref}{%
5351         \__stex_term_oms_or_omv:nnn{#1}{}
5352         {\l_stex_default_notation}
5353       }
5354       \group_end:
5355     }
5356   }
5357 }

```

Return:

```

5358 \cs_new_protected:Nn \__stex_expr_invoke_return_maybe:n {
5359   \tl_clear:N \l_stex_expr_return_args_tl
5360   \tl_set:Nn \l_stex_expr_return_this_tl {#1}
5361   \exp_args:Nnx \use:n {
5362     \cs_generate_from_arg_count:NNnn \__stex_expr_ret_cs
5363     \cs_set:Npn \l_stex_current_arity_str } {
5364       \int_step_function:nN \l_stex_current_arity_str \__stex_expr_return_arg:n
5365       \__stex_expr_invoke_return_next:
5366     }
5367   \__stex_expr_ret_cs
5368 }
5369
5370 \cs_new:Nn \__stex_expr_return_arg:n {
5371   \tl_put_right:Nn \exp_not:N \l_stex_expr_return_args_tl {{#### #1}}
5372 }
5373
5374 \cs_new_protected:Nn \__stex_expr_invoke_return_next: {
5375   \peek_charcode_remove:NTF ! {
5376     \exp_after:wN \l_stex_expr_return_this_tl \l_stex_expr_return_args_tl \group_end:
5377   } \__stex_expr_invoke_return:
5378 }
5379
5380 \cs_new_protected:Nn \__stex_expr_invoke_return: {

```

```

5381 \tl_set:Nx \l__stex_expr_return_this_tl {
5382   \__stex_expr_return_notation:n {
5383     \exp_after:wN \exp_after:wN \exp_after:wN
5384     \exp_not:n \exp_after:wN \exp_after:wN \exp_after:wN {
5385       \exp_after:wN \l__stex_expr_return_this_tl \l__stex_expr_return_args_tl
5386     }
5387   }
5388 }
5389 \stex_debug:nn{return}{Notation:~\meaning\l__stex_expr_return_this_tl}
5390 \tl_put_left:Nx \l__stex_expr_return_this_tl {
5391   \group_begin:\exp_args:No \exp_not:n \l_stex_current_redo_tl
5392 }
5393 \exp_args:Nnx \use:n {
5394   \cs_generate_from_arg_count:NNnn \__stex_expr_ret_cs
5395   \cs_set:Npn \l_stex_current_arity_str } {
5396     \exp_args:No \exp_not:n \l_stex_current_return_tl
5397   }
5398 \stex_debug:nn{return}{
5399   \meaning\__stex_expr_ret_cs^~J
5400   \meaning\l__stex_expr_return_this_tl^~J
5401   \exp_args:No \exp_not:n \l__stex_expr_return_args_tl^~J
5402 }
5403 \exp_args:Nnx \use:nn {
5404   \exp_after:wN \group_end: \__stex_expr_ret_cs
5405 }{
5406   \exp_args:No \exp_not:n \l__stex_expr_return_args_tl
5407   {
5408     \exp_args:No \exp_not:n \l__stex_expr_return_this_tl
5409     \group_end:
5410   }
5411 }
5412 }
5413
5414
5415 \cs_new_protected:Nn \__stex_expr_return_notation:n {
5416   \tl_if_empty:NTF \l_stex_return_notation_tl { #1 }{
5417     \l_stex_return_notation_tl
5418   }
5419 }
5420

```

#### Custom Notations:

```

5421 \cs_new_protected:Nn \__stex_expr_invoke_op_custom:n {
5422   \stex_debug:nn{expressions}{custom~op}
5423   \bool_set_true:N \l_stex_allow_semantic_bool
5424   \stex_term_oms_or_omv:nnn{}{}{\maincomp{#1}}
5425   \group_end:
5426 }
5427
5428 \int_new:N \l_stex_expr_arg_counter_int
5429 \cs_new_protected:Nn \__stex_expr_invoke_custom:n {
5430   \stex_debug:nn{custom}{custom~notation~for~\l_stex_current_symbol_str}
5431   \stex_pseudogroup:nn{
5432     \bool_set_true:N \l_stex_allow_semantic_bool

```

```

5433 \prop_gclear:N \l__stex_expr_customs_prop
5434 \seq_gclear:N \l__stex_expr_customs_seq
5435 \int_gzero:N \l__stex_expr_arg_counter_int
5436 \tl_if_empty:NF \l_stex_current_args_tl {
5437   \exp_after:wN \__stex_expr_add_prop_arg:nw \l_stex_current_args_tl \_stex_args_end:
5438   \cs_set_eq:NN \arg \__stex_expr_arg:n
5439 }
5440 \tl_set_eq:NN \l_stex_get_symbol_args_tl \l_stex_current_args_tl
5441 \cs_set_eq:NN \__stex_expr_do_ab_next:nnn \_stex_term_oma:nnn
5442 \_stex_map_args:N \__stex_expr_check_b:nn
5443 \__stex_expr_do_ab_next:nnn{}{}{#1}
5444 }{
5445   \prop_if_exist:NT \l__stex_expr_customs_prop {
5446     \prop_gset_from_keyval:Nn \exp_not:N \l__stex_expr_customs_prop {
5447       \prop_to_keyval:N \l__stex_expr_customs_prop
5448     }
5449   }
5450   \int_gset:Nn \l__stex_expr_arg_counter_int { \int_use:N \l__stex_expr_arg_counter_int}
5451   \seq_if_exist:NT \l__stex_expr_customs_seq {
5452     \seq_gset_split:Nnn \exp_not:N \l__stex_expr_customs_seq , {
5453       \seq_use:Nn \l__stex_expr_customs_seq ,
5454     }
5455   }
5456 }
5457 % TODO check that all arguments are present
5458 \group_end:
5459 }
5460
5461 \cs_new_protected:Npn \__stex_expr_add_prop_arg:nw #1 #2 #3 \_stex_args_end: {
5462   \prop_gput:Nnn \l__stex_expr_customs_prop {#1} {}
5463   \seq_gput_right:Nn \l__stex_expr_customs_seq {#2}
5464   \tl_if_empty:nF{#3}{\__stex_expr_add_prop_arg:nw #3 \_stex_args_end:}
5465 }
5466
5467 \cs_new:Nn \__stex_expr_check_b:nn {
5468   \str_case:nn #2 {
5469     b {\cs_set_eq:NN \__stex_expr_do_ab_next:nnn \_stex_term_omb:nnn}
5470     B {\cs_set_eq:NN \__stex_expr_do_ab_next:nnn \_stex_term_omb:nnn}
5471   }
5472 }
5473
5474 \NewDocumentCommand \__stex_expr_arg:n { s O{} m } {
5475   \IfBooleanTF #1 {
5476     \stex_annotation_invisible:n{
5477       \__stex_expr_arg_inner:nn{#2}{#3}
5478     }
5479   }{
5480     \__stex_expr_arg_inner:nn{#2}{#3}
5481   }
5482 }
5483
5484 \cs_new_protected:Nn \__stex_expr_arg_inner:nn {
5485   \tl_if_empty:nTF{#1} {
5486     \int_gincr:N \l__stex_expr_arg_counter_int

```

```

5487 \exp_args:Ne \__stex_expr_check:nTF{ \int_use:N \l__stex_expr_arg_counter_int }{
5488   \__stex_expr_arg_do:oon \l_tmpa_tl \l_tmpb_tl
5489 }
5490   \__stex_expr_arg_inner:nn{}}
5491 }{ #2 }
5492 }
5493   \__stex_expr_check:nTF {#1}{%
5494     \__stex_expr_arg_do:oon \l_tmpa_tl \l_tmpb_tl { #2 }
5495   }
5496     \exp_args:No \str_case:nnTF \l_tmpb_tl {
5497       {a}%
5498         \exp_args:NNnx \prop_gput:Nnn \l__stex_expr_customs_prop {#1}{%
5499           \l_tmpa_tl X
5500         }
5501         \tl_set:Nx \l_tmpa_tl { #1 \int_eval:n {\tl_count:N \l_tmpa_tl + 1} }
5502       }
5503       {B}%
5504         \exp_args:NNnx \prop_gput:Nnn \l__stex_expr_customs_prop {#1}{%
5505           \l_tmpa_tl X
5506         }
5507         \tl_set:Nx \l_tmpa_tl { #1 \int_eval:n {\tl_count:N \l_tmpa_tl + 1} }
5508       }
5509     }
5510     \__stex_expr_arg_do:oon \l_tmpa_tl \l_tmpb_tl { #2 }
5511   }
5512   \msg_error:nnxx{stex}{error/invalidarg}{#1}{\l_stex_current_symbol_str}
5513 }
5514 }
5515 }
5516 }
5517
5518 \prg_new_if_empty:NNTF \__stex_expr_check:n {TF} {
5519   \exp_args:NN \prop_get:NnNTF \l__stex_expr_customs_prop {#1} \l_tmpa_tl {
5520     \tl_set:Nx \l_tmpb_tl {\seq_item:Nn \l__stex_expr_customs_seq {#1} }
5521   }
5522   \exp_args:NN \prop_gput:Nnn \l__stex_expr_customs_prop
5523     { #1 }{X}
5524   \exp_args:No \str_case:nnF \l_tmpb_tl {
5525     {a}%
5526       \tl_set:Nx \l_tmpa_tl{ #1 1 }
5527     }
5528     {B}%
5529       \tl_set:Nx \l_tmpa_tl{ #1 1 }
5530     }
5531   }
5532   \tl_set:Nx \l_tmpa_tl{ #1 }
5533 }
5534   \prg_return_true:
5535 }
5536   \prg_return_false:
5537 }
5538 }
5539 \msg_error:nnxx{stex}{error/invalidarg}{#1}{\l_stex_current_symbol_str}
5540 \prg_return_false:

```

```

5541     }
5542 }
5543
5544 % #1 argnum #2 argmode #3 code
5545 \cs_new_protected:Nn \__stex_expr_arg_do:nnn {
5546   \stex_debug:nn{custom}{Doing~argument~#1~of~mode~#2:~\tl_to_str:n{#3}}
5547   \group_begin:
5548     \bool_set_true:N \l_stex_allow_semantic_bool
5549     \stex_term_arg:nnn {#2}{#1}{#3}
5550   \group_end:
5551 }
5552 \cs_generate_variant:Nn \__stex_expr_arg_do:nnn {oon}

```

(End of definition for `\stex_invoke_symbol`. This function is documented on page 131.)

## Argument Handling and Annotating

```

\stex_term_arg:nnnnn
\stex_term_arg:nnn
5553 % 1: argnum 2: argmode 3: precedence 4: argname 5: code
5554 \cs_new_protected:Nn \stex_term_arg:nnnnn {
5555   \group_begin:
5556     \str_clear:N \l_stex_current_symbol_str
5557     \tl_clear:N \l_stex_current_term_tl
5558     \int_set:Nn \l_stex_notation_downprec { #3 }
5559     \bool_set_true:N \l_stex_allow_semantic_bool
5560     \stex_term_arg:nnn {#2}{#1} {
5561       \tl_if_empty:nTF{#4} {
5562         #5
5563       }{
5564         \stex_annotation:nn{mml:arg={#4}}{#5}
5565       }
5566     }
5567   \group_end:
5568 }
5569
5570 \cs_new_protected:Nn \stex_term_arg:nnn {
5571   \stex_annotation:nn{ shtml:arg={#2}, shtml:argmode={#1}}{
5572     \stex_annotation_force_break:n{ #3 }
5573   }
5574 }

```

(End of definition for `\stex_term_arg:nnnnn` and `\stex_term_arg:nnn`. These functions are documented on page ??.)

```

\stex_term_arg_aB:nnnnn
5575 \tl_set:Nn \__stex_expr_do_aB_clist: {
5576   \seq_use:Nn \l_stex_aB_args_seq {
5577     \mathpunct{\comp{,}}}
5578   }
5579 }
5580 \tl_set_eq:NN \stex_term_do_aB_clist: \__stex_expr_do_aB_clist:
5581
5582 \int_new:N \l__stex_expr_count_int
5583 \cs_new_protected:Nn \stex_term_arg_aB:nnnnn {
5584   \tl_if_empty:nTF{#5} {

```

```

5585     \_stex_term_arg:nnnnn{#1}{#2}{#3}{#4}{}
5586   }{
5587     \seq_clear:N \l_stex_aB_args_seq
5588     \int_zero:N \l__stex_expr_count_int
5589     \clist_map_inline:nn{#5} {
5590       \_stex_expr_aB_arg:nnnnn{##1}{#1}{#2}{#3}{#4}
5591     }
5592     \_stex_term_do_aB_clist:
5593   }
5594 }
5595
5596 % 1: code 2: argnum 3: argmode 4: precedence 5: argname
5597 \cs_new_protected:Npn \_stex_expr_aB_arg:nnnnn #1 {
5598   \int_incr:N \l__stex_expr_count_int
5599   \_stex_expr_is_varseq:nTF{#1} {
5600     \exp_after:wN \exp_after:wN \exp_after:wN
5601     \_stex_expr_assoc_seq:nnnnnnn
5602     \exp_after:wN
5603     \_stex_expr_gobble:nnnnnnnn #1 \_stex_expr_end:
5604   }{
5605     \_stex_expr_is_seqmap:nTF{#1} {
5606       \exp_args:NNe \use:nn \_stex_expr_do_seqmap:nnnnnn {\tl_tail:n{#1}}
5607     }{
5608       \_stex_expr_aB_simple_arg:nnnnn{#1}
5609     }
5610   }
5611 }
5612
5613 \cs_new:Npn \_stex_expr_gobble:nnnnnnnn #1 #2 #3 #4 #5 #6 #7 #8 #9 \_stex_expr_end: {
5614   {#2} #3 {#6}
5615 }
5616
5617 \cs_new_protected:Nn \_stex_expr_aB_simple_arg:nnnnn{
5618   \seq_put_right:Nx \l_stex_aB_args_seq {
5619     \_stex_term_arg:nnnnn{#2}\int_use:N\l__stex_expr_count_int}{#3}{#4}{#5}{#6}
5620     \exp_not:n{
5621       \tl_set_eq:NN \_stex_term_do_aB_clist: \_stex_expr_do_aB_clist:
5622       #1
5623     }
5624   }
5625 }
5626 }
5627
5628 \cs_new_protected:Nn \_stex_is_sequentialized:n {
5629   \group_begin: #1 \group_end:
5630 }

Conditionals: Is the argument a sequence variable or a \seqmap?

5631 \prg_new_conditional:Nnn \_stex_expr_is_varseq:n {TF} {
5632   \int_compare:nNnTF {\tl_count:n{#1}} = 1 {
5633     \exp_args:Ne \cs_if_eq:NNTF {\exp_args:No \tl_head:n{#1}}
5634     \_stex_invoke_variable:nnnnnnN {
5635       \exp_args:Ne \cs_if_eq:NNTF {\exp_args:No\tl_item:nn{#1}{8}}
5636       \stex_invoke_sequence:

```

```

5637     \prg_return_true:\prg_return_false:
5638   }\prg_return_false:
5639 }\prg_return_false:
5640 }
5641
5642 \prg_new_conditional:Nnn \__stex_expr_is_seqmap:n {TF} {
5643   \int_compare:nNnTF {\tl_count:n{#1}} = 3 {
5644     \exp_args:Ne \tl_if_eq:nnTF {\tl_head:n{#1}} {\seqmap}
5645     \prg_return_true:\prg_return_false:
5646   }\prg_return_false:
5647 }

Sequence variable:

5648 % 1: name 2: arity 3: clist 4: argnum 5: argmode 6: precedence 7: argname
5649 \cs_new_protected:Nn \__stex_expr_assoc_seq:nnnnnnn {
5650   \group_begin:
5651     \seq_clear:N \l_stex_aB_args_seq
5652     \__stex_expr_assoc_make_seq:nnn{#1}{#3}{#2}
5653   \exp_args:NNe \use:nn \group_end: {
5654     \seq_put_right:Nn \exp_not:N \l_stex_aB_args_seq {
5655       \stex_is_sequentialized:n{
5656         \__stex_term_arg:nnnnn{#4}\int_use:N\__stex_expr_count_int}{#5}{#6}{#7}{#8}{#9}{#10}{#11}{#12}{#13}{#14}{#15}{#16}{#17}{#18}{#19}{#20}{#21}{#22}{#23}{#24}{#25}{#26}{#27}{#28}{#29}{#30}{#31}{#32}{#33}{#34}{#35}{#36}{#37}{#38}{#39}{#40}{#41}{#42}{#43}{#44}{#45}{#46}{#47}{#48}{#49}{#50}{#51}{#52}{#53}{#54}{#55}{#56}{#57}{#58}{#59}{#60}{#61}{#62}{#63}{#64}{#65}{#66}{#67}{#68}{#69}{#70}{#71}{#72}{#73}{#74}{#75}{#76}{#77}{#78}{#79}{#80}{#81}{#82}{#83}{#84}{#85}{#86}{#87}{#88}{#89}{#90}{#91}{#92}{#93}{#94}{#95}{#96}{#97}{#98}{#99}{#100}{#101}{#102}{#103}{#104}{#105}{#106}{#107}{#108}{#109}{#110}{#111}{#112}{#113}{#114}{#115}{#116}{#117}{#118}{#119}{#120}{#121}{#122}{#123}{#124}{#125}{#126}{#127}{#128}{#129}{#130}{#131}{#132}{#133}{#134}{#135}{#136}{#137}{#138}{#139}{#140}{#141}{#142}{#143}{#144}{#145}{#146}{#147}{#148}{#149}{#150}{#151}{#152}{#153}{#154}{#155}{#156}{#157}{#158}{#159}{#160}{#161}{#162}{#163}{#164}{#165}{#166}{#167}{#168}{#169}{#170}{#171}{#172}{#173}{#174}{#175}{#176}{#177}{#178}{#179}{#180}{#181}{#182}{#183}{#184}{#185}{#186}{#187}{#188}{#189}{#190}{#191}{#192}{#193}{#194}{#195}{#196}{#197}{#198}{#199}{#200}{#201}{#202}{#203}{#204}{#205}{#206}{#207}{#208}{#209}{#210}{#211}{#212}{#213}{#214}{#215}{#216}{#217}{#218}{#219}{#220}{#221}{#222}{#223}{#224}{#225}{#226}{#227}{#228}{#229}{#230}{#231}{#232}{#233}{#234}{#235}{#236}{#237}{#238}{#239}{#240}{#241}{#242}{#243}{#244}{#245}{#246}{#247}{#248}{#249}{#250}{#251}{#252}{#253}{#254}{#255}{#256}{#257}{#258}{#259}{#260}{#261}{#262}{#263}{#264}{#265}{#266}{#267}{#268}{#269}{#270}{#271}{#272}{#273}{#274}{#275}{#276}{#277}{#278}{#279}{#280}{#281}{#282}{#283}{#284}{#285}{#286}{#287}{#288}{#289}{#290}{#291}{#292}{#293}{#294}{#295}{#296}{#297}{#298}{#299}{#300}{#301}{#302}{#303}{#304}{#305}{#306}{#307}{#308}{#309}{#3010}{#3011}{#3012}{#3013}{#3014}{#3015}{#3016}{#3017}{#3018}{#3019}{#3020}{#3021}{#3022}{#3023}{#3024}{#3025}{#3026}{#3027}{#3028}{#3029}{#3030}{#3031}{#3032}{#3033}{#3034}{#3035}{#3036}{#3037}{#3038}{#3039}{#3040}{#3041}{#3042}{#3043}{#3044}{#3045}{#3046}{#3047}{#3048}{#3049}{#3050}{#3051}{#3052}{#3053}{#3054}{#3055}{#3056}{#3057}{#3058}{#3059}{#3060}{#3061}{#3062}{#3063}{#3064}{#3065}{#3066}{#3067}{#3068}{#3069}{#3070}{#3071}{#3072}{#3073}{#3074}{#3075}{#3076}{#3077}{#3078}{#3079}{#3080}{#3081}{#3082}{#3083}{#3084}{#3085}{#3086}{#3087}{#3088}{#3089}{#3090}{#3091}{#3092}{#3093}{#3094}{#3095}{#3096}{#3097}{#3098}{#3099}{#30100}{#30101}{#30102}{#30103}{#30104}{#30105}{#30106}{#30107}{#30108}{#30109}{#30110}{#30111}{#30112}{#30113}{#30114}{#30115}{#30116}{#30117}{#30118}{#30119}{#30120}{#30121}{#30122}{#30123}{#30124}{#30125}{#30126}{#30127}{#30128}{#30129}{#30130}{#30131}{#30132}{#30133}{#30134}{#30135}{#30136}{#30137}{#30138}{#30139}{#30140}{#30141}{#30142}{#30143}{#30144}{#30145}{#30146}{#30147}{#30148}{#30149}{#30150}{#30151}{#30152}{#30153}{#30154}{#30155}{#30156}{#30157}{#30158}{#30159}{#30160}{#30161}{#30162}{#30163}{#30164}{#30165}{#30166}{#30167}{#30168}{#30169}{#30170}{#30171}{#30172}{#30173}{#30174}{#30175}{#30176}{#30177}{#30178}{#30179}{#30180}{#30181}{#30182}{#30183}{#30184}{#30185}{#30186}{#30187}{#30188}{#30189}{#30190}{#30191}{#30192}{#30193}{#30194}{#30195}{#30196}{#30197}{#30198}{#30199}{#301000}{#301001}{#301002}{#301003}{#301004}{#301005}{#301006}{#301007}{#301008}{#301009}{#301010}{#301011}{#301012}{#301013}{#301014}{#301015}{#301016}{#301017}{#301018}{#301019}{#301020}{#301021}{#301022}{#301023}{#301024}{#301025}{#301026}{#301027}{#301028}{#301029}{#301030}{#301031}{#301032}{#301033}{#301034}{#301035}{#301036}{#301037}{#301038}{#301039}{#301040}{#301041}{#301042}{#301043}{#301044}{#301045}{#301046}{#301047}{#301048}{#301049}{#301050}{#301051}{#301052}{#301053}{#301054}{#301055}{#301056}{#301057}{#301058}{#301059}{#301060}{#301061}{#301062}{#301063}{#301064}{#301065}{#301066}{#301067}{#301068}{#301069}{#301070}{#301071}{#301072}{#301073}{#301074}{#301075}{#301076}{#301077}{#301078}{#301079}{#301080}{#301081}{#301082}{#301083}{#301084}{#301085}{#301086}{#301087}{#301088}{#301089}{#301090}{#301091}{#301092}{#301093}{#301094}{#301095}{#301096}{#301097}{#301098}{#301099}{#3010100}{#3010101}{#3010102}{#3010103}{#3010104}{#3010105}{#3010106}{#3010107}{#3010108}{#3010109}{#3010110}{#3010111}{#3010112}{#3010113}{#3010114}{#3010115}{#3010116}{#3010117}{#3010118}{#3010119}{#3010120}{#3010121}{#3010122}{#3010123}{#3010124}{#3010125}{#3010126}{#3010127}{#3010128}{#3010129}{#3010130}{#3010131}{#3010132}{#3010133}{#3010134}{#3010135}{#3010136}{#3010137}{#3010138}{#3010139}{#3010140}{#3010141}{#3010142}{#3010143}{#3010144}{#3010145}{#3010146}{#3010147}{#3010148}{#3010149}{#3010150}{#3010151}{#3010152}{#3010153}{#3010154}{#3010155}{#3010156}{#3010157}{#3010158}{#3010159}{#3010160}{#3010161}{#3010162}{#3010163}{#3010164}{#3010165}{#3010166}{#3010167}{#3010168}{#3010169}{#3010170}{#3010171}{#3010172}{#3010173}{#3010174}{#3010175}{#3010176}{#3010177}{#3010178}{#3010179}{#3010180}{#3010181}{#3010182}{#3010183}{#3010184}{#3010185}{#3010186}{#3010187}{#3010188}{#3010189}{#3010190}{#3010191}{#3010192}{#3010193}{#3010194}{#3010195}{#3010196}{#3010197}{#3010198}{#3010199}{#30101910}{#30101911}{#30101912}{#30101913}{#30101914}{#30101915}{#30101916}{#30101917}{#30101918}{#30101919}{#30101920}{#30101921}{#30101922}{#30101923}{#30101924}{#30101925}{#30101926}{#30101927}{#30101928}{#30101929}{#301019210}{#301019211}{#301019212}{#301019213}{#301019214}{#301019215}{#301019216}{#301019217}{#301019218}{#301019219}{#301019220}{#301019221}{#301019222}{#301019223}{#301019224}{#301019225}{#301019226}{#301019227}{#301019228}{#301019229}{#3010192210}{#3010192211}{#3010192212}{#3010192213}{#3010192214}{#3010192215}{#3010192216}{#3010192217}{#3010192218}{#3010192219}{#3010192220}{#3010192221}{#3010192222}{#3010192223}{#3010192224}{#3010192225}{#3010192226}{#3010192227}{#3010192228}{#3010192229}{#30101922210}{#30101922211}{#30101922212}{#30101922213}{#30101922214}{#30101922215}{#30101922216}{#30101922217}{#30101922218}{#30101922219}{#30101922220}{#30101922221}{#30101922222}{#30101922223}{#30101922224}{#30101922225}{#30101922226}{#30101922227}{#30101922228}{#30101922229}{#301019222210}{#301019222211}{#301019222212}{#301019222213}{#301019222214}{#301019222215}{#301019222216}{#301019222217}{#301019222218}{#301019222219}{#301019222220}{#301019222221}{#301019222222}{#301019222223}{#301019222224}{#301019222225}{#301019222226}{#301019222227}{#301019222228}{#301019222229}{#3010192222210}{#3010192222211}{#3010192222212}{#3010192222213}{#3010192222214}{#3010192222215}{#3010192222216}{#3010192222217}{#3010192222218}{#3010192222219}{#3010192222220}{#3010192222221}{#3010192222222}{#3010192222223}{#3010192222224}{#3010192222225}{#3010192222226}{#3010192222227}{#3010192222228}{#3010192222229}{#30101922222210}{#30101922222211}{#30101922222212}{#30101922222213}{#30101922222214}{#30101922222215}{#30101922222216}{#30101922222217}{#30101922222218}{#30101922222219}{#30101922222220}{#30101922222221}{#30101922222222}{#30101922222223}{#30101922222224}{#30101922222225}{#30101922222226}{#30101922222227}{#30101922222228}{#30101922222229}{#301019222222210}{#301019222222211}{#301019222222212}{#301019222222213}{#301019222222214}{#301019222222215}{#301019222222216}{#301019222222217}{#301019222222218}{#301019222222219}{#301019222222220}{#301019222222221}{#301019222222222}{#301019222222223}{#301019222222224}{#301019222222225}{#301019222222226}{#301019222222227}{#301019222222228}{#301019222222229}{#3010192222222210}{#3010192222222211}{#3010192222222212}{#3010192222222213}{#3010192222222214}{#3010192222222215}{#3010192222222216}{#3010192222222217}{#3010192222222218}{#3010192222222219}{#3010192222222220}{#3010192222222221}{#3010192222222222}{#3010192222222223}{#3010192222222224}{#3010192222222225}{#3010192222222226}{#3010192222222227}{#3010192222222228}{#3010192222222229}{#30101922222222210}{#30101922222222211}{#30101922222222212}{#30101922222222213}{#30101922222222214}{#30101922222222215}{#30101922222222216}{#30101922222222217}{#30101922222222218}{#30101922222222219}{#30101922222222220}{#30101922222222221}{#30101922222222222}{#30101922222222223}{#30101922222222224}{#30101922222222225}{#30101922222222226}{#30101922222222227}{#30101922222222228}{#30101922222222229}{#301019222222222210}{#301019222222222211}{#301019222222222212}{#301019222222222213}{#301019222222222214}{#301019222222222215}{#301019222222222216}{#301019222222222217}{#301019222222222218}{#301019222222222219}{#301019222222222220}{#301019222222222221}{#301019222222222222}{#301019222222222223}{#301019222222222224}{#301019222222222225}{#301019222222222226}{#301019222222222227}{#301019222222222228}{#301019222222222229}{#3010192222222222210}{#3010192222222222211}{#3010192222222222212}{#3010192222222222213}{#3010192222222222214}{#3010192222222222215}{#3010192222222222216}{#3010192222222222217}{#3010192222222222218}{#3010192222222222219}{#3010192222222222220}{#3010192222222222221}{#3010192222222222222}{#3010192222222222223}{#3010192222222222224}{#3010192222222222225}{#3010192222222222226}{#3010192222222222227}{#3010192222222222228}{#3010192222222222229}{#30101922222222222210}{#30101922222222222211}{#30101922222222222212}{#30101922222222222213}{#30101922222222222214}{#30101922222222222215}{#30101922222222222216}{#30101922222222222217}{#30101922222222222218}{#30101922222222222219}{#30101922222222222220}{#30101922222222222221}{#30101922222222222222}{#30101922222222222223}{#30101922222222222224}{#30101922222222222225}{#30101922222222222226}{#30101922222222222227}{#30101922222222222228}{#30101922222222222229}{#301019222222222222210}{#301019222222222222211}{#301019222222222222212}{#301019222222222222213}{#301019222222222222214}{#301019222222222222215}{#301019222222222222216}{#301019222222222222217}{#301019222222222222218}{#301019222222222222219}{#301019222222222222220}{#301019222222222222221}{#301019222222222222222}{#301019222222222222223}{#301019222222222222224}{#301019222222222222225}{#301019222222222222226}{#301019222222222222227}{#301019222222222222228}{#301019222222222222229}{#3010192222222222222210}{#3010192222222222222211}{#3010192222222222222212}{#3010192222222222222213}{#3010192222222222222214}{#3010192222222222222215}{#3010192222222222222216}{#3010192222222222222217}{#3010192222222222222218}{#3010192222222222222219}{#3010192222222222222220}{#3010192222222222222221}{#3010192222222222222222}{#3010192222222222222223}{#3010192222222222222224}{#3010192222222222222225}{#3010192222222222222226}{#3010192222222222222227}{#3010192222222222222228}{#3010192222222222222229}{#30101922222222222222210}{#30101922222222222222211}{#30101922222222222222212}{#30101922222222222222213}{#30101922222222222222214}{#30101922222222222222215}{#30101922222222222222216}{#30101922222222222222217}{#30101922222222222222218}{#30101922222222222222219}{#30101922222222222222220}{#30101922222222222222221}{#30101922222222222222222}{#30101922222222222222223}{#30101922222222222222224}{#30101922222222222222225}{#30101922222222222222226}{#30101922222222222222227}{#30101922222222222222228}{#30101922222222222222229}{#301019222222222222222210}{#301019222222222222222211}{#301019222222222222222212}{#301019222222222222222213}{#301019222222222222222214}{#301019222222222222222215}{#301019222222222222222216}{#301019222222222222222217}{#301019222222222222222218}{#301019222222222222222219}{#301019222222222222222220}{#301019222222222222222221}{#301019222222222222222222}{#301019222222222222222223}{#301019222222222222222224}{#301019222222222222222225}{#301019222222222222222226}{#301019222222222222222227}{#301019222222222222222228}{#301019222222222222222229}{#3010192222222222222222210}{#3010192222222222222222211}{#3010192222222222222222212}{#3010192222222222222222213}{#3010192222222222222222214}{#3010192222222222222222215}{#3010192222222222222222216}{#3010192222222222222222217}{#3010192222222222222222218}{#3010192222222222222222219}{#3010192222222222222222220}{#3010192222222222222222221}{#3010192222222222222222222}{#3010192222222222222222223}{#3010192222222222222222224}{#3010192222222222222222225}{#3010192222222222222222226}{#3010192222222222222222227}{#3010192222222222222222228}{#3010192222222222222222229}{#30101922222222222222222210}{#30101922222222222222222211}{#30101922222222222222222212}{#30101922222222222222222213}{#30101922222222222222222214}{#30101922222222222222222215}{#30101922222222222222222216}{#30101922222222222222222217}{#30101922222222222222222218}{#30101922222222222222222219}{#30101922222222222222222220}{#30101922222222222222222221}{#30101922222222222222222222}{#30101922222222222222222223}{#30101922222222222222222224}{#30101922222222222222222225}{#30101922222222222222222226}{#30101922222222222222222227}{#30101922222222222222222228}{#30101922222222222222222229}{#301019222222222222222222210}{#301019222222222222222222211}{#301019222222222222222222212}{#301019222222222222222222213}{#301019222222222222222222214}{#301019222222222222222222215}{#301019222222222222222222216}{#301019222222222222222222217}{#301019222222222222222222218}{#301019222222222222222222219}{#301019222222222222222222220}{#301019222222222222222222221}{#301019222222222222222222222}{#301019222222222222222222223}{#30
```

```

5689 \tl_if_eq:nnTF{##1}{\ellipses}{
5690   \seq_put_right:Nn \l_stex_aB_args_seq { ##1 }
5691 }{
5692   \int_compare:nNnTF {#3} = 1 {
5693     \tl_set:Nn \l__stex_expr_iarg_tl { {##1} }
5694   }{
5695     \tl_set:Nn \l__stex_expr_iarg_tl { ##1 }
5696   }
5697   \seq_put_right:Nx \l_stex_aB_args_seq {
5698     \group_begin:
5699     \exp_not:n {
5700       \tl_set_eq:NN \__stex_term_do_aB_clist: \__stex_expr_do_aB_clist:
5701       \def\comp{\_varcomp}
5702       \str_set:Nn \l_stex_current_symbol_str{#1}
5703     }
5704     \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:n
5705     \exp_after:wN \exp_after:wN \exp_after:wN {
5706       \exp_after:wN \l__stex_expr_cs \exp_after:wN \group_end: \l__stex_expr_iarg_tl
5707     }
5708   }
5709 }
5710 }
5711 }

\seqmap:

5712 % 1: fun 2: clist 3: argnum 4: argmode 5: precedence 6: argname
5713 \cs_new_protected:Nn \__stex_expr_do_seqmap:nnnnnn {
5714   \group_begin:
5715     \cs_set:Npn \l_tmpa_cs ##1 {#1}
5716     \seq_clear:N \l_stex_aB_args_seq
5717     \clist_map_inline:nn{#2} {
5718       \__stex_expr_is_varseq:nTF{##1} {
5719         \exp_after:wN
5720         \__stex_expr_varseq_in_map:nnnnnnnn ##1
5721       }{
5722         \seq_put_right:Nn \l_stex_aB_args_seq {##1}
5723       }
5724     }
5725     \seq_clear:N \l__stex_expr_old_seq
5726     \seq_map_inline:Nn \l_stex_aB_args_seq {
5727       \tl_if_eq:nnTF{##1}{\ellipses} {
5728         \seq_put_right:Nn \l__stex_expr_old_seq {##1}
5729       }
5730       % TODO \stex_is_sequentialized:n
5731     {
5732       \exp_args:NNo \seq_put_right:Nn \l__stex_expr_old_seq {
5733         \l_tmpa_cs {##1}
5734       }
5735     }
5736   }
5737   \seq_set_eq:NN \l_stex_aB_args_seq \l__stex_expr_old_seq
5738 \exp_args:NNe \use:nn \group_end: {
5739   \seq_put_right:Nn \exp_not:N \l_stex_aB_args_seq {
5740     \stex_is_sequentialized:n{

```

```

5741   \_stex_term_arg:nnnnn{#4\int_use:N\l_stex_expr_count_int}{#4}{#5}{#6}{
5742     \bool_set_true:N \l_stex_allow_semantic_bool
5743     \str_set:Nn \exp_not:N \l_stex_current_symbol_str
5744       {\l_stex_current_symbol_str}
5745     \tl_set:Nn \exp_not:N \l_stex_current_term_tl {
5746       \symuse{Metatheory?sequence-map}
5747       {\exp_args:No \exp_not:n \l_tmpa_tl}
5748       {\_stex_term_do_aB_clist:}
5749     }
5750     \_stex_expr_do_headterm:nnn{}{
5751       \_stex_term_do_aB_clist:
5752     }
5753   }
5754 }
5755 }
5756 }
5757 }
5758
5759 \cs_new_protected:Nn \_stex_expr_varseq_in_map:nnnnnnnn {
5760   \_stex_expr_assoc_make_seq:nnn{#2}{#6}{#3}
5761 }

```

(End of definition for `\_stex_term_arg_aB:nnnnn`. This function is documented on page ??.)

## Term HTML Annotations

```

\_stex_term_oms_or_omv:nnn
\_stex_term_oms:nnn
\_stex_term_omv:nnn
5762 \cs_new_protected:Nn \_stex_eat_exclamation_point: {
5763   \peek_charcode_remove:NT ! {
5764     \_stex_eat_exclamation_point:
5765   }
5766 }
5767
5768 \bool_new:N \stex_in_invisible_html_bool
5769 \bool_set_false:N \stex_in_invisible_html_bool
5770 \stex_if_html_backend:TF {
5771   % 1: variant 2: intent 3: code
5772   \cs_new_protected:Nn \_stex_term_oms:nnn {
5773     \tl_if_empty:NTF \l_stex_current_term_tl {
5774       \stex_annotation:nn{
5775         shtml:term=OMID,
5776         shtml:head={\l_stex_current_symbol_str},
5777         shtml:notationid={#1},
5778       }{
5779         \_stex_annotation_force_break:n{#3}
5780       }
5781     }{
5782       \_stex_expr_do_headterm:nnn{#1}{#3}
5783     }
5784   }
5785   \cs_new_protected:Nn \_stex_term_omv:nnn {
5786     \tl_if_empty:NTF \l_stex_current_term_tl {
5787       \stex_annotation:nn{
5788         shtml:term=OMV,

```

```

5789         shtml:head={\l_stex_current_symbol_str},
5790         shtml:notationid={#1}
5791     }{
5792         \_stex_annotation_force_break:n{#3}
5793     }
5794 }{
5795     \_\_stex_expr_do_headterm:nn{#1}{#3}
5796 }
5797 }
5798 \cs_new_protected:Nn \_\_stex_expr_do_headterm:nn {
5799     \bool_if:NTF \stex_in_invisible_html_bool {
5800         {\bool_set_true:N \l_stex_allow_semantic_bool
5801             \ensuremath{\l_stex_current_term_tl}}
5802     }
5803 }{
5804     \stex_annotation:nn{
5805         shtml:term=complex,
5806         shtml:head={\l_stex_current_symbol_str},
5807         shtml:notationid={#1}
5808     }{
5809         \_stex_annotation_force_break:n{
5810             \stex_annotation_invisible:nn{shtml:headterm={}}{
5811                 {\bool_set_true:N \l_stex_allow_semantic_bool
5812                     \ensuremath{\l_stex_current_term_tl}}
5813                 }
5814             }
5815         }
5816         #2
5817     }
5818 }
5819 }
5820 }{
5821     \cs_new_protected:Nn \_stex_term_oms:nnn {#3}
5822     \cs_new_protected:Nn \_stex_term_omv:nnn {#3}
5823     \cs_new_protected:Nn \_\_stex_expr_do_headterm:nn { #2 }
5824 }
5825 \cs_set_eq:NN \_stex_term_oms_or_omv:nnn \_stex_term_oms:nnn

```

(End of definition for `\_stex_term_oms_or_omv:nnn`, `\_stex_term_oms:nnn`, and `\_stex_term_omv:nnn`.  
These functions are documented on page ??.)

```

\_stex_term_oma:nnn
5826 \stex_if_html_backend:TF {
5827     \cs_new_protected:Nn \_stex_term_oma:nnn {
5828         \tl_if_empty:NTF \l_stex_current_term_tl {
5829             \stex_annotation:nn{
5830                 shtml:term=OMA,
5831                 shtml:head={\l_stex_current_symbol_str},
5832                 shtml:notationid={#1}
5833             }{
5834                 \_stex_annotation_force_break:n{#3}
5835             }
5836 }{
5837     \stex_annotation:nn{

```

```

5838     shtml:term=OMA,
5839     shtml:head={\l_stex_current_symbol_str},
5840     shtml:notationid={#1}
5841   }{
5842     \_stex_annotation_force_break:n{
5843       \stex_annotation_invisible:nn{shtml:headterm={}{{
5844         \bool_set_true:N \l_stex_allow_semantic_bool
5845         \l_stex_current_term_tl
5846       }}}
5847       #3
5848     }
5849   }
5850 }
5851 }
5852 }{
5853   \cs_new_protected:Nn \_stex_term_oma:nnn {#3}
5854 }

```

(End of definition for `\_stex_term_oma:nnn`. This function is documented on page ??.)

```

\_stex_term_omb:nnn
5855 \stex_if_html_backend:TF {
5856   \cs_new_protected:Nn \_stex_term_omb:nnn {
5857     \tl_if_empty:NTF \l_stex_current_term_tl {
5858       \stex_annotation:nn{
5859         shtml:term=OMBIND,
5860         shtml:head={\l_stex_current_symbol_str},
5861         shtml:notationid={#1}
5862       }{
5863         \_stex_annotation_force_break:n{#3}
5864       }
5865     }{
5866       \stex_annotation:nn{
5867         shtml:term=OMBIND,
5868         shtml:head={\l_stex_current_symbol_str},
5869         shtml:notationid={#1}
5870       }{
5871         \_stex_annotation_force_break:n{
5872           \stex_annotation_invisible:nn{shtml:headterm={}{{
5873             \bool_set_true:N \l_stex_allow_semantic_bool
5874             \l_stex_current_term_tl
5875           }}}
5876           #3
5877         }
5878       }
5879     }
5880   }{
5881   \cs_new_protected:Nn \_stex_term_omb:nnn { #3 }
5882 }
5883 }

```

(End of definition for `\_stex_term_omb:nnn`. This function is documented on page ??.)

## Automated Bracketing

```
\infprec  
\neginfprec
```

```
5884 \tl_const:Nx \infprec {\int_use:N \c_max_int}  
5885 \tl_const:Nx \neginfprec {-\int_use:N \c_max_int}
```

(End of definition for `\infprec` and `\neginfprec`. These functions are documented on page 91.)

```
\dobrackets
```

```
5886 \int_new:N \l_stex_notation_downprec  
5887 \int_set:Nn \l_stex_notation_downprec \infprec  
5888 \tl_set:Nn \l_stex_expr_left_bracket_str (   
5889 \tl_set:Nn \l_stex_expr_right_bracket_str )  
5890 \bool_new:N \l_stex_brackets_dones_bool  
5891  
5892 \cs_new_protected:Nn \stex_maybe_brackets:nn {  
5893     \bool_if:NTF \l_stex_brackets_dones_bool {  
5894         \bool_set_false:N \l_stex_brackets_dones_bool  
5895         #2  
5896     } {  
5897         \stex_debug:nn{brackets}{#1}\int_eval:n \l_stex_notation_downprec?  
5898         \int_compare:nNnTF { #1 } > \l_stex_notation_downprec {  
5899             \%bool_if:NTF \l_stex_inpararray_bool { #2 }{  
5900                 \dobrackets {  
5901                     #2  
5902                 }  
5903                 %}  
5904             }{  
5905                 #2  
5906             }  
5907         }  
5908     }  
5909  
5910 \%RequirePackage{scalerel}  
5911 \cs_new_protected:Npn \dobrackets #1 {  
5912     \%ThisStyle{\if D\m@switch  
5913         \% \exp_args:Nnx \use:nn  
5914         \% { \exp_after:wN \left\l_stex_expr_left_bracket_str #1 }  
5915         \% { \exp_not:N\right\l_stex_expr_right_bracket_str }  
5916         \% \else  
5917         \group_begin:  
5918             \%stex_pseudogroup_with:nn{\l_stex_brackets_dones_bool\l_stex_notation_downprec}{  
5919                 \bool_set_true:N \l_stex_brackets_dones_bool  
5920                 \%int_set:Nn \l_stex_notation_downprec \infprec  
5921                 \mathopen{\cs_if_exist:NT\l_stex_current_symbol_str\comp  
5922                     \l_stex_expr_left_bracket_str  
5923                 }  
5924                 #1  
5925             \group_end:{}  
5926             \mathclose{\cs_if_exist:NT\l_stex_current_symbol_str\comp  
5927                 \l_stex_expr_right_bracket_str  
5928             }  
5929         \%fi}  
5930 }
```

(End of definition for \dobrackets. This function is documented on page ??.)

### \withbrackets

```
5931 \cs_new_protected:Npn \withbrackets #1 #2 #3 {
5932   \stex_pseudogroup_with:nn{\l__stex_expr_left_bracket_str\l__stex_expr_right_bracket_str}{}
5933   \tl_set:Nn \l__stex_expr_left_bracket_str { #1 }
5934   \tl_set:Nn \l__stex_expr_right_bracket_str { #2 }
5935   #3
5936 }
5937 }
```

(End of definition for \withbrackets. This function is documented on page ??.)

### \dowithbrackets

```
5938 \cs_new_protected:Npn \dowithbrackets #1 #2 #3 {
5939   \withbrackets{#1}{#2}{\dobrackets{#3}}
5940 }
```

(End of definition for \dowithbrackets. This function is documented on page ??.)

## Symname and Variants

```
\symname
\sn      \def\maincomp{\comp}
\sns
\Symname \stex_keys_define:nnnn{symname}{
  \tl_clear:N \l_stex_key_pre_tl
  \tl_clear:N \l_stex_key_post_tl
  \% \tl_clear:N \l_stex_key_proot_tl
}
\symref
\sr
\varref
\varname
\Varname }{}}

\NewDocumentCommand \symref { O{} m m } {
  \group_begin:
  \stex_keys_set:nn{symname}{#1}
  \stex_get_symbol:n{#2}
  \__stex_expr_do_ref:nNn{#3}\symrefemph@uri\_stex_term_oms:nnn
}
\let\sr\symref

\NewDocumentCommand \symname { O{} m } {
  \group_begin:
  \stex_keys_set:nn{symname}{#1}
  \stex_get_symbol:n{#2}
  \__stex_expr_do_ref:nNn{
    \l_stex_key_pre_tl\l_stex_get_symbol_name_str\l_stex_key_post_tl
  }\symrefemph@uri\_stex_term_oms:nnn
}
\let\sn\symname
\protected\def\sns{\symname[post=s]}
```

```

5972 \NewDocumentCommand \Symname { O{} m} {
5973   \group_begin:
5974   \stex_keys_set:nn{symname}{#1}
5975   \stex_get_symbol:n{#2}
5976   \__stex_expr_do_ref:nNn{
5977     \l_stex_key_pre_tl\exp_after:wN\_stex_capitalize:n\l_stex_get_symbol_name_str\l_stex_key
5978   }\symrefemph@uri\_stex_term_oms:nnn
5979 }
5980 \cs_new_protected:Nn \_stex_capitalize:n {
5981   \uppercase{#1}
5982 }
5983 \let\Sn\Symname
5984 \protected\def\Sns{\Symname[post=s]}
5985
5986 \cs_new:Npn \_stex_split_slash: #1/#2/#3\_stex_args_end: {
5987   \tl_if_empty:nTF{#3}{
5988     #2
5989   }{
5990     \_stex_split_slash: #2 / #3 \_stex_args_end:
5991   }
5992 }
5993
5994 \NewDocumentCommand \varref { O{} m m} {
5995   \group_begin:
5996   \stex_keys_set:nn{symname}{#1}
5997   \stex_get_var:n{#2}
5998   \__stex_expr_do_ref:nNn{#3}\varempath@uri{
5999     \str_set_eq:NN \l_stex_current_symbol_str\l_stex_get_symbol_name_str
6000     \def\comp{\_varcomp}
6001     \_stex_term_omv:nnn
6002   }
6003 }
6004
6005 \NewDocumentCommand \varname { O{} m} {
6006   \group_begin:
6007   \stex_keys_set:nn{symname}{#1}
6008   \stex_get_var:n{#2}
6009   \__stex_expr_do_ref:nNn{
6010     \l_stex_key_pre_tl\l_stex_get_symbol_name_str\l_stex_key_post_tl
6011   }\varempath@uri{
6012     \str_set_eq:NN \l_stex_current_symbol_str\l_stex_get_symbol_name_str
6013     \def\comp{\_varcomp}
6014     \_stex_term_omv:nnn
6015   }
6016 }
6017
6018 \NewDocumentCommand \Varname { O{} m} {
6019   \group_begin:
6020   \stex_keys_set:nn{symname}{#1}
6021   \stex_get_var:n{#2}
6022   \__stex_expr_do_ref:nNn{
6023     \l_stex_key_pre_tl\exp_after:wN\_stex_capitalize:n\l_stex_get_symbol_name_str\l_stex_key
6024   }\varempath@uri{
6025     \str_set_eq:NN \l_stex_current_symbol_str\l_stex_get_symbol_name_str

```

```

6026     \def\comp{\_varcomp}
6027     \stex_term_omv:nnn
6028 }
6029 }
6030
6031
6032 \cs_new_protected:Nn \__stex_expr_do_ref:nNn {
6033   \stex_if_html_backend:T{\ifvmode\indent\fi}
6034   \bool_if:NTF \l_stex_allow_semantic_bool{
6035     \str_set:Nx\l_stex_current_symbol_str
6036       {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
6037     \str_if_in:NnT \l_stex_get_symbol_name_str / {
6038       \str_set:Nx \l_stex_get_symbol_name_str {
6039         \exp_after:wn \l_stex_split_slash: \l_stex_get_symbol_name_str
6040         /\_stex_args_end:
6041       }
6042     }
6043     \tl_clear:N \l_stex_current_term_tl
6044     \def\comp{\_comp}
6045     \let\compemph@uri#2
6046     #3{}{}{\comp{#1}}
6047 }{
6048   \msg_error:nnxx{stex}{error/notallowed}{#1}{\l_stex_current_symbol_str}
6049 }
6050 \group_end:
6051 }

```

(End of definition for \symname and others. These functions are documented on page 89.)

## Highlighting

```

6074 }
6075
6076 \cs_new_protected:Npn \_defcomp {
6077     \do_comp:nNn {definiendum}\defemph@uri
6078 }
6079
6080 \cs_set_protected:Npn \comp {}
6081
6082 \cs_new_protected:Npn \compemph@uri #1 #2 {
6083     \compemph{#1 }
6084 }
6085
6086 \cs_new_protected:Npn \compemph #1 {
6087     #1
6088 }
6089
6090 \cs_new_protected:Npn \defemph@uri #1 #2 {
6091     \defemph{#1}
6092 }
6093
6094 \cs_new_protected:Npn \defemph #1 {
6095     \ifmmode\else\expandafter\textbf\fi{#1}
6096 }
6097
6098 \cs_new_protected:Npn \symrefemph@uri #1 #2 {
6099     \symrefemph{#1}
6100 }
6101
6102 \cs_new_protected:Npn \symrefemph #1 {
6103     \emph{#1}
6104 }
6105
6106 \cs_new_protected:Npn \varemph@uri #1 #2 {
6107     \varemph{#1}
6108 }
6109
6110 \cs_new_protected:Npn \varemph #1 {
6111     #1
6112 }

```

(End of definition for `\comp` and others. These functions are documented on page 90.)

## 13.9 Mathematical Structures

```

6113 <@@=stex_structures>
\this
6114 \cs_new_protected:Npn \stex_current_this: {
6115     \bool_set_true:N \l_stex_allow_semantic_bool
6116     \tl_if_empty:NTF \l_stex_current_this_tl {{}}{
6117         \str_set:Nx \l_stex_current_symbol_str {
6118             \l_stex_current_module_str ? \l__stex_structures_name_str
6119         }
6120         \maincomp{\l_stex_current_this_tl}

```

```

6121     }
6122   }
6123 }
6124 \let \this \stex_current_this:

(End of definition for \this. This function is documented on page 96.)

mathstructure (env.)
6125 \stex_new_stylable_env:nnnnnnn {mathstructure}{m 0{}}
6126   \__stex_structures_begin:nn{#1}{#2}
6127   \stex_smsmode_do:
6128 }{
6129   \stex_structural_feature_module_end:
6130   \__stex_structures_do_externals:
6131 }{}{}{}

6132 \stex_keys_define:nnnn{mathstructure}{
6133   \tl_clear:N \l_stex_current_this_tl
6134   \str_clear:N \l__stex_structures_name_str
6135 }{
6136   this .tl_set:N = \l_stex_current_this_tl ,
6137   unknown .code:n = {
6138     \str_if_empty:NTF \l_keys_key_str {
6139       \str_set:Nx \l__stex_structures_name_str {\l_keys_key_tl}
6140     }{
6141       \str_set_eq:NN \l__stex_structures_name_str \l_keys_key_str
6142     }
6143   }
6144 }
6145 }{}

6146 \cs_new_protected:Nn \__stex_structures_begin:nn {
6147   \stex_keys_set:nn {mathstructure}{#2}
6148   \str_if_empty:NT \l__stex_structures_name_str {
6149     \str_set:Nn \l__stex_structures_name_str {#1}
6150   }
6151   \def\comp{\_comp}

6152 \exp_args:Nne \use:nn { \stex_module_add_symbol:nnnnnnnN }
6153   { [#1]{\l__stex_structures_name_str}{0}{}{defed}{\l_stex_current_module_str / \l__stex_structures_name_str-module}}
6154   {}{\stex_invoke_structure:
6155     \str_set:Nx \l_stex_mroname_str {#1}
6156     \stex_execute_in_module:x{
6157       \seq_clear:c{\l_stex_structure_macros_\l_stex_current_module_str / \l__stex_structures_name_str}
6158       \seq_put_right:cn{\l_stex_structure_macros_\l_stex_current_module_str / \l__stex_structures_name_str}{#1}
6159     }
6160   }
6161   \exp_args:No \stex_structural_feature_module:nn
6162     {\l__stex_structures_name_str}{structure}
6163 }

6164 \stex_sms_allow_env:n{mathstructure}
6165 \stex_deactivate_macro:Nn \mathstructure {module-environments}
6166 \stex_every_module:n {\stex_reactivate_macro:N \mathstructure}
6167
6168 \stex_sms_activate_env:n{mathstructure}
6169 \stex_activate_macro:Nn \mathstructure {module-environments}
6170 \stex_every_module:n {\stex_reactivate_macro:N \mathstructure}

```

```

6171 \cs_new_protected:Nn \__stex_structures_do_externals: {
6172   \tl_set:Nn \l__stex_structures_replace_this_tl {####1}
6173   \exp_args:No \stex_iterate_symbols:n{g_stex_last_feature_str} {
6174     \__stex_structures_external_decl:nnnn{##5}{##4}{##3}{##8}
6175   }
6176 }
6177 }
6178
6179 \cs_new_protected:Nn \__stex_structures_external_decl:nnnn {
6180   \%stex_debug:nn{structure}{%
6181     % Generating-external-declaration-\l__stex_structures_name_str/#3-in-
6182     % \l_stex_current_module_str^~J
6183     % \tl_to_str:n{#1}^~J\tl_to_str:n{#2}^~J\tl_to_str:n{#4}
6184   }%
6185   \%tl_set:Nn \l_stex_get_symbol_args_tl {#1}
6186   \%exp_args:Nnx \use:nn {\stex_module_add_symbol:nnnnnnN} {
6187     {}{\l__stex_structures_name_str/#3}\int_eval:n{#2 + 1}}
6188     % {ii}\tl_if_empty:nF{#1}{\stex_map_args:N \__stex_structures_shift_argls:nn}
6189     % {defed}{typed}
6190   \%{#4}\stex_invoke_outer_field:
6191 }
6192
6193 \cs_new:Nn \__stex_structures_shift_argls:nn {
6194   \int_eval:n{#1+1}\#2
6195 }

```

\stex\_get\_mathstructure:n

```

6196 \cs_new_protected:Nn \stex_get_mathstructure:n {
6197   \__stex_get_mathstructure:n{#1}
6198   \str_if_empty:NT \l_stex_get_structure_module_str {
6199     \msg_error:nnn{stex}{error/unknownstructure}{#1}
6200   }
6201 }
6202 \cs_new_protected:Nn \__stex_get_mathstructure:n {
6203   \str_clear:N \l_stex_get_structure_module_str
6204   \__stex_get_symbol:n{#1}
6205   \str_if_empty:NF \l_stex_get_symbol_name_str {
6206     \exp_args:No \tl_if_eq:NNT \l_stex_get_symbol_invoke_cs \stex_invoke_structure: {
6207       \str_set_eq:NN \l_stex_get_structure_module_str \l_stex_get_symbol_type_tl
6208     }
6209   }
6210 }

```

(End of definition for \stex\_get\_mathstructure:n. This function is documented on page ??.)

extstructure (env.)

```

6211 \stex_new_stylable_env:nnnnnn {extstructure}{m 0{} m}{%
6212   \seq_clear:N \l__stex_structures_imports_seq
6213   \clist_map_inline:nn{#3}{%
6214     \__stex_get_mathstructure:n{##1}
6215     \clist_map_inline:Nn \l_stex_get_symbol_type_tl {
6216       \exp_args:Ne \stex_if_module_exists:nT{\tl_to_str:n{####1}}{%
6217         \seq_put_right:Nn \l__stex_structures_imports_seq{####1}
6218       }
6219     }
6220   }

```

```

6219 }
6220 \stex_execute_in_module:x{
6221   \seq_put_right:cn{l_stex_structure_macros_}\l_stex_get_structure_module_str _seq}{#1}
6222 }
6223 }
6224 \__stex_structures_begin:nn{#1}{#2}
6225 \seq_map_inline:Nn\l__stex_structures_imports_seq{
6226   \stex_if_do_html:T {
6227     \hbox{\stex_annotation_invisible:nn
6228       {shtml:import={##1} {}}}
6229   }
6230   \stex_module_add_morphism:nonn
6231     {}{##1}{import}{}
6232   \stex_execute_in_module:x{
6233     \stex_activate_module:n{##1}
6234   }
6235 }
6236 \stex_smsmode_do:
6237 }{
6238   \stex_structural_feature_module_end:
6239   \__stex_structures_do_exernals:
6240 }{}{}{}
6241
6242 \stex_sms_allow_env:n{extstructure}
6243 \stex_deactivate_macro:Nn \extstructure {module-environments}
6244 \stex_every_module:n {
6245   \stex_reactivate_macro:N \extstructure
6246 }
6247
6248 \cs_new:Nn \__stex_structures_extend_structure_i:NnnnnnnN {
6249   \exp_not:n{#1}{#2}{#3}{#4}{#5}{defed}}{\l__stex_structures_extmod_str,#7}\exp_not:n{{#8}{#9}}
6250 }
6251 \cs_new_protected:Nn \__stex_structures_extend_structure:nn {
6252   \stex_debug:nn{ext}{Extending-#1-by-#2}
6253   \str_set:Nn \l__stex_structures_extmod_str{#2}
6254   \tl_set:cx{#1}{
6255     \exp_after:wN \exp_after:wN \exp_after:wN
6256     \__stex_structures_extend_structure_i:NnnnnnnN \cs:w #1 \cs_end:
6257   }
6258 }
6259
6260 \stex_new_stylable_env:nnnnnn {extstructure*}{m}){
6261   \__stex_structures_new_extstruct_name:
6262   \seq_clear:N \l__stex_structures_imports_seq
6263   \stex_get_mathstructure:n{#1}
6264
6265   \clist_map_inline:Nn \l_stex_get_symbol_type_tl {
6266     \exp_args:Ne \stex_if_module_exists:nT{\tl_to_str:n{##1}}{
6267       \seq_put_right:Nn \l__stex_structures_imports_seq{##1}
6268     }
6269   }
6270
6271 \stex_execute_in_module:x{
6272   \seq_map_inline:cn{l_stex_structure_macros_}\l_stex_get_structure_module_str _seq}{

```

```

6273     \exp_not:N \tl_if_exist:cT{####1}{
6274         \__stex_structures_extend_structure:nn{####1}{\l_stex_current_module_str/\l__stex_st
6275     }
6276 }
6277 }
6278
6279 \exp_args:No \__stex_structures_begin:nn\l__stex_structures_exstruct_name_str{}
6280
6281 \seq_map_inline:Nn \l__stex_structures_imports_seq {
6282     \stex_if_do_html:T {
6283         \stex_annotation_invisible:nn
6284             {shtml:import= {##1}} {}
6285     }
6286     \stex_module_add_morphism:nonn
6287         {}{##1}{import}{}
6288     \stex_execute_in_module:x{
6289         \stex_activate_module:n{##1}
6290     }
6291 }
6292
6293 \stex_smsmode_do:
6294 }{
6295     \prop_map_inline:cn{
6296         c_stex_module_ \l_stex_current_module_str _symbols_prop
6297     }{
6298         \__stex_structures_check_def:nnnnnnnn ##2
6299     }
6300     \stex_structural_feature_module_end:
6301     \__stex_structures_do	externals:
6302 }{}{}{}
6303
6304 \stex_sms_allow_env:n{extstructure*}
6305 \exp_after:wN \stex_deactivate_macro:Nn
6306     \cs:w extstructure*\cs_end: {module-environments}
6307 \stex_every_module:n {
6308     \exp_after:wN \stex_reactivate_macro:N \cs:w extstructure*\cs_end:
6309 }
6310
6311 \cs_new_protected:Nn \__stex_structures_check_def:nnnnnnnn {
6312     \tl_if_empty:nT{#5}{
6313         \msg_error:nnnn{stex}{error/needsdefiniens}{#2}{extstructure*}
6314     }
6315 }
6316
6317 \stex_every_module:n{ \str_set:Nn \l__stex_structures_extname_count 0}
6318
6319 \cs_new_protected:Nn \__stex_structures_new_extstruct_name: {
6320     \stex_do_up_to_module:n {
6321         \str_set:Nx \l__stex_structures_extname_count {\int_eval:n{\l__stex_structures_extname_c
6322     }
6323     \str_set:Nx \l__stex_structures_exstruct_name_str {EXTSTRUCT_\l__stex_structures_extname_c
6324 }
6325

```

Invoking structures:

```
6326 \cs_new_protected:Nn \stex_invoke_structure: {
6327   \tl_set:Nn \l__stex_structures_set_comp_tl {\__stex_structures_set_thiscomp:}
6328   \__stex_structures_invoke_top:n {}
6329 }
6330
6331 \cs_new_protected:Nn \__stex_structures_invoke_top:n {
6332   \stex_debug:nn{structure} {
6333     invoking~structure~{\l_stex_current_type_tl}<\tl_to_str:n{#1}>
6334   }
6335   \peekCharCode:NTF [ {
6336     \__stex_structures_merge:nw{#1}
6337   }{
6338     \__stex_structures_invocation_type:n {#1}
6339     \tl_set:Nn \l__stex_structures_this_tl {}
6340     \peekCharCode_remove:NTF ! {
6341       \peekCharCode:NTF [ {
6342         \__stex_structures_maybe_notation:w
6343       }{
6344         \__stex_structures_maybe_notation:w []
6345       }
6346     }{
6347       \__stex_structures_invoke_this:n
6348     }
6349   }
6350 }
6351
6352 \cs_new_protected:Npn \__stex_structures_merge:nw #1 [ #2 ] {
6353   \exp_args:Ne \stex_if_starts_with:nnTF {\tl_to_str:n{#2}}{comp=}{%
6354     \__stex_structures_set_customcomp: #2 \__stex_structures_end:
6355     \__stex_structures_invoke_top:n{#1}
6356   }{
6357     \exp_args:Ne \stex_if_starts_with:nnTF {\tl_to_str:n{#2}}{this=}{%
6358       \__stex_structures_set_thisnotation: #2 \__stex_structures_end:
6359       \__stex_structures_invoke_top:n{#1}
6360     }{
6361       \tl_if_empty:nTF{#1} {
6362         \__stex_structures_invoke_top:n{#2}
6363       }{
6364         \tl_if_empty:nTF{#2} {
6365           \__stex_structures_invoke_top:n{#1}
6366         }{
6367           \__stex_structures_invoke_top:n{#1,#2}
6368         }
6369       }
6370     }
6371   }
6372 }
6373
6374 \cs_new_protected:Npn \__stex_structures_set_thisnotation: this= #1 \__stex_structures_end:
6375   \tl_set:Nn \l_stex_return_notation_tl { \comp{#1} }
6376   \tl_set:Nn \l__stex_structures_set_comp_tl {}
6377 }
6378
```

```

6379 \cs_new_protected:Npn \__stex_structures_set_customcomp: comp= #1 \__stex_structures_end: {
6380   \tl_set:Nn \l__stex_structures_set_comp_tl {
6381     \__stex_structures_set_custom_comp:n{#1}
6382   }
6383   \tl_set:Nn \l_stex_return_notation_tl { \comp{} }
6384 }

```

The structure type:

```

6385 \cs_new_protected:Nn \__stex_structures_invocation_type:n {
6386   \__stex_structures_do_assign_list:n{#1}
6387   \clist_if_empty:NTF \l__stex_structures_fields_clist {
6388     \int_compare:nNnTF {\clist_count:N} \l_stex_current_type_tl}
6389     = 1 {
6390       \tl_set:Nx \l__stex_structures_current_type_tl {
6391         \exp_args:No \exp_not:n \l_stex_current_redo_tl
6392         \stex_term_oms_or_omv:nnn{}{}{}
6393       }
6394     }{
6395       \exp_args:No \__stex_structures_make_type:n \l_stex_current_type_tl
6396     }
6397   }{
6398     \int_compare:nNnTF {\clist_count:N} \l_stex_current_type_tl}
6399     = 1 {
6400       \__stex_structures_make_type:n {}
6401     }{
6402       \exp_args:No \__stex_structures_make_type:n \l_stex_current_type_tl
6403     }
6404   }
6405 }
6406
6407 \cs_new_protected:Nn \__stex_structures_do_assign_list:n {
6408   \clist_clear:N \l__stex_structures_fields_clist
6409   \tl_if_empty:nF {#1} {
6410     \keyval_parse>NNn\TODO\__stex_structures_do_assign:nn{#1}
6411   }
6412 }
6413
6414 \cs_new_protected:Nn \__stex_structures_do_assign:nn {
6415   \clist_put_right:Nn \l__stex_structures_fields_clist {{#1}{#2}}
6416 }
6417
6418 \cs_new_protected:Nn \__stex_structures_make_type:n {
6419   \tl_if_empty:nTF{#1} {
6420     \seq_clear:N \l_tmpa_seq
6421   }{
6422     \seq_set_split:Nnn \l_tmpa_seq ,{#1}
6423     \seq_pop_right:NN \l_tmpa_seq \l_tmpa_tl
6424     \seq_reverse:N \l_tmpa_seq
6425   }
6426   \tl_set:Nx \l__stex_structures_current_type_tl {
6427     \symuse{Metatheory?module~type~merge} {
6428       {
6429         \exp_args:No \exp_not:n \l_stex_current_redo_tl
6430         \stex_term_oms_or_omv:nnn{}{}{}
6431       }

```

```

6432     \seq_map_function:NN \l_tmpa_seq \__stex_structures_make_mod:n
6433     \clist_if_empty:NF \l__stex_structures_fields_clist {
6434         ,\symuse{Metatheory?anonymous~record}{
6435             \exp_args:Ne \tl_tail:n{
6436                 \clist_map_function:NN \l__stex_structures_fields_clist \__stex_structures_make_
6437             }
6438         }
6439     }
6440 }
6441 }
6442 }
6443
6444 \cs_new:Nn \__stex_structures_make_mod:n {
6445     ,\symuse{Metatheory?module-type}{
6446         \stex_annotation:nn{shtml:term=OMMOD,shtml:head={#1}}{}
6447     }
6448 }
6449
6450 \cs_new:Nn \__stex_structures_make_oml:n {
6451     \__stex_structures_make_oml:nn #1
6452 }
6453 \cs_new:Nn \__stex_structures_make_oml:nn {
6454     ,\stex_annotation:nn{
6455         shtml:term=OML,
6456         shtml:head={#1}
6457     }{
6458         \stex_annotation_force_break:n{
6459             \stex_annotation:nn{shtml:definiens={}}{\exp_not:n{#2!}}
6460         }
6461     }
6462 }

```

Insert the structure type as a term:

```

6463 \cs_new:Nn \__stex_structures_current_type: {
6464     \%exp_args:No \exp_not:n \l_stex_current_redo_tl
6465     \tl_set:Nn \exp_not:N \l_stex_current_term_tl {
6466         \exp_args:No\exp_not:n\l__stex_structures_current_type_tl
6467     }
6468     \l_stex_term_oms_or_omv:nnn{}{}{}
6469 }

```

The structure type itself:

```

6470 \cs_new_protected:Npn \__stex_structures_maybe_notation:w [ #1 ] {
6471     \tl_set_eq:NN \l_stex_current_term_tl \l__stex_structures_current_type_tl
6472     \cs_if_exist:cTF{\l_stex_notation_\l_stex_current_symbol_str _#1_cs} {
6473         \use:c{\l_stex_notation_\l_stex_current_symbol_str _#1_cs}\group_end:
6474     }{
6475         \__stex_structures_make_prop:
6476         \__stex_structures_make_prop_assign:
6477         \__stex_structures_present_i:w [#1]
6478     }
6479 }
6480
6481 \cs_new_protected:Nn \__stex_structures_present: {
6482     \peek_charcode:NTF [ {

```

```

6483     \__stex_structures_present_i:w
6484 }{
6485     \__stex_structures_present:nn{}{}
6486 }
6487 }
6488
6489 \cs_new_protected:Npn \__stex_structures_present_i:w [#1] {
6490     \int_compare:nNnTF{\clist_count:n{#1}} = 1 {
6491         \__stex_structures_present:nn{}{#1}
6492 }{
6493     \peek_charcode:NTF [ {
6494         \__stex_structures_present_ii:nw{#1}
6495 }{
6496         \__stex_structures_present:nn{#1}{}
6497 }
6498 }
6499 }
6500
6501 %First: clist, second:notation-id
6502 \cs_new_protected:Npn \__stex_structures_present_ii:nw #1 [#2] {
6503     \__stex_structures_present:nn{#1}{#2}
6504 }
6505
6506 \cs_new_protected:Nn \__stex_structures_present:nn {
6507     \clist_clear:N \l__stex_structures_clist
6508     \tl_if_empty:nTF{#1} {
6509         \cs_set:Npn \l__stex_structures_cs ##1 ##2 ##3 {
6510             \tl_if_empty:nF{##2} {
6511                 \__stex_structures_present_entry:nn {##1}{##3}
6512             }
6513         }
6514 }{
6515         \cs_set:Npn \l__stex_structures_cs ##1 ##2 ##3 {
6516             \exp_args:Ne \clist_if_in:nnT{\tl_to_str:n{#1}}{##1} {
6517                 \__stex_structures_present_entry:nn {##1}{##3}
6518             }
6519         }
6520     }
6521     \prop_map_inline:Nn \l__stex_structures_prop {
6522         \l__stex_structures_cs {##1} ##2
6523     }
6524     \__stex_term_oms_or_omv:nnn{}{}{
6525         \exp_args:Nno \use:n{
6526             \bool_set_true:N \l_stex_allow_semantic_bool
6527             \symuse{Metatheory?mathematical-structure}{#2}
6528         }{\l__stex_structures_clist}
6529     }\group_end:
6530 }
6531
6532 \cs_new_protected:Nn \__stex_structures_present_entry:nn {
6533     \seq_if_in:NnTF \l__stex_structures_assigned_seq {#1} {
6534         \clist_put_right:Nn \l__stex_structures_clist {#2!}
6535 }{
6536     \exp_args:NNe \clist_put_right:Nn \l__stex_structures_clist {

```

```

6537     \_stex_next_symbol:n {
6538         \exp_args:No \exp_not:n \l__stex_structures_set_comp_tl
6539         \tl_set:Nn \exp_not:N \l__stex_structures_this_tl {
6540             \exp_args:No \exp_not:n \l__stex_structures_this_tl
6541         }
6542         \exp_not:n {
6543             \tl_set_eq:NN \this \l__stex_structures_this_tl
6544         }
6545         \tl_set:Nn \exp_not:N \l_stex_return_notation_tl {
6546             \exp_args:No \exp_not:n \l_stex_return_notation_tl
6547         }
6548     }
6549     \exp_not:n{#2!}
6550 }
6551 }
6552 }
6553
6554 \cs_new_protected:Npn \_thiscomp #1 #2 {
6555     {\tl_set:cn{this}{}#1{#2}\c_math_subscript_token{
6556         \group_begin:
6557         \bool_set_true:N \l_stex_allow_semantic_bool
6558         \l__stex_structures_this_tl
6559         \group_end:
6560     }
6561 }
6562 }
6563
6564 \cs_new_protected:Nn \__stex_structures_set_thiscomp: {
6565     \exp_args:Ne \tl_if_eq:NNF {\tl_head:N \maincomp} \_thiscomp {
6566         \edef\maincomp {\_thiscomp\comp}
6567     }
6568 }
6569
6570 \cs_new_protected:Nn \__stex_structures_set_custom_comp:n {
6571     \exp_args:Ne \tl_if_eq:NNF {\tl_head:N \maincomp} \_customthiscomp {
6572         \cs_set_protected:Npx \_customthiscomp ##1 {
6573             \group_begin:
6574                 \bool_set_true:N \l_stex_allow_semantic_bool
6575                 \exp_not:n{
6576                     \cs_set:Npn \l__stex_structures_comp_cs{##1} {
6577                         #1
6578                     }
6579                     \def\maincomp
6580                 }{\comp}
6581                     \exp_not:N\l__stex_structures_comp_cs{\comp{##1}}
6582                     \group_end:
6583                 }
6584                     \def\maincomp {\_customthiscomp}
6585                 }
6586 }
6587
6588 this (of type structure):
6589 \cs_new_protected:Nn \__stex_structures_invoke_this:n {

```

```

6590 \peek_charcode_remove:NTF ! {
6591   \exp_args:Nne\use:nn{
6592     \group_end:\symuse{Metatheory?of~type}[invisible]{#1}
6593   }{
6594     {\__stex_structures_current_type:}
6595   }
6596   }{
6597     \__stex_structures_invoke_maybe_field:nn{#1}
6598   }
6599 }
6600
6601 \cs_new_protected:Nn \__stex_structures_invoke_maybe_field:nn {
6602   \__stex_structures_make_prop:
6603   \__stex_structures_set_this:n{#1}
6604   \tl_if_empty:nTF{#2} {
6605     \__stex_structures_make_prop_assign:
6606     \__stex_structures_present:
6607   }{
6608     \__stex_structures_invoke_field:n{#2}
6609   }
6610 }
6611
6612 \cs_new_protected:Nn \__stex_structures_set_this:n {
6613   \tl_if_empty:nTF{#1} {
6614     \%tl_put_right:Nn \l_stex_current_redo_tl {
6615       \%tl_clear:N \l_stex_structures_this_tl
6616     }
6617   }{
6618     \tl_set:Nx \l_stex_structures_this_tl {{
6619       \bool_set_true:N \l_stex_allow_semantic_bool
6620       \tl_set:Nn \exp_not:N \this {
6621         \exp_args:No \exp_not:n \this
6622       }
6623       \exp_not:n{#1}
6624     }}
6625     \tl_set_eq:NN \this \l_stex_structures_this_tl
6626     \%l_stex_return_notation_tl
6627   }
6628 }
6629
6630 \cs_new_protected:Nn \__stex_structures_get_field_name:n {
6631   \str_set:Nx \l_stex_structures_field_name_str {
6632     \exp_args:Nne \use:n {\exp_after:wN \use_i:nn \use:n}
6633     {\prop_item:Nn \l_stex_structures_prop {#1}}
6634   }
6635   \str_if_empty:NT \l_stex_structures_field_name_str {
6636     \str_set:Nn \l_stex_structures_field_name_str {#1}
6637   }
6638 }
6639
6640 \cs_new_protected:Nn \__stex_structures_invoke_field:n {
6641   \prop_if_in:NnTF \l_stex_structures_prop {#1} {
6642     \__stex_structures_get_field_name:n{#1}
6643     \tl_clear:N \l_stex_structures_more_nextsymbol_tl

```

```

6644 \%exp_args:NNe \seq_if_in:NnF \l__stex_structures_assigned_seq {\tl_to_str:n{#1}}{
6645   \tl_set:Nx \l__stex_structures_more_nextsymbol_tl {
6646     \tl_set:Nn \exp_not:N \l__stex_structures_this_tl {
6647       \exp_args:No \exp_not:n \l__stex_structures_this_tl
6648     }
6649     \exp_not:n {
6650       \tl_set_eq:NN \this \l__stex_structures_this_tl
6651     }
6652     \tl_set:Nn \exp_not:N \l_stex_return_notation_tl {
6653       \exp_args:No \exp_not:n \l_stex_return_notation_tl
6654     }
6655     \exp_args:No \exp_not:n \l__stex_structures_set_comp_tl
6656   }
6657 %}
6658 \exp_args:NNx \use:nn \group_end: {
6659   \stex_next_symbol:n {
6660     \exp_args:No \exp_not:n \l__stex_structures_redo_tl
6661     \tl_set:Nn \exp_not:N \l_stex_current_term_tl {
6662       \symuse{Metatheory?record~field}{%
6663         \symuse{Metatheory?of~type}{%
6664           \exp_args:No \exp_not:n \l__stex_structures_this_tl
6665         }{ \stex_structures_current_type: }
6666       }
6667       \stex_annotation:nn{shtml:term=OML,shtml:head={\l__stex_structures_field_name_str}}
6668     }
6669   }
6670   \exp_args:No \exp_not:n \l__stex_structures_more_nextsymbol_tl
6671 }
6672 \exp_not:N \use_ii:nn
6673 \prop_item:Nn \l__stex_structures_prop {#1}
6674 }
6675 }{
6676   \msg_error:nnn{stex}{error/unknownfield}{#1}
6677 }
6678 }
6679
6680 \cs_new_protected:Nn \stex_structures_make_prop: {
6681   \prop_clear:N \l__stex_structures_prop
6682   \seq_clear:N \l__stex_structures_seq
6683   \seq_clear:N \l__stex_structures_assigned_seq
6684   \tl_clear:N \l__stex_structures_redo_tl
6685   \stex_structures_prop_do_decls:
6686   \stex_structures_prop_do_notations:
6687 }
6688
6689 \cs_new_protected:Nn \stex_structures_make_prop_assign: {
6690   \clist_if_empty:NF \l__stex_structures_fields_clist {
6691     \clist_map_inline:Nn \l__stex_structures_fields_clist {
6692       \stex_structures_make_prop_assign:nn ##1
6693     }
6694   }
6695 }
6696
6697 \cs_new_protected:Nn \stex_structures_make_prop_assign:nn {

```

```

6698 \prop_if_in:NnTF \l__stex_structures_prop {#1}{
6699   \exp_args:NNe \seq_put_right:Nn \l__stex_structures_assigned_seq {\tl_to_str:n{#1}}
6700   \exp_args:Nne \use:nn {\l__stex_structures_make_prop_assign_replace:nnnn {#1}{#2}}
6701   {\prop_item:Nn \l__stex_structures_prop {#1}}
6702 }{
6703   \msg_error:nnn{stex}{error/unknownfieldass}{#1}
6704 }
6705 }
6706 \cs_new_protected:Nn \l__stex_structures_make_prop_assign_replace:nnnn {
6707   \prop_put:Nnn \l__stex_structures_prop {#1}{#3}{#2}
6708   \tl_if_empty:nF{#3}{
6709     \tl_set:cn{#1}{ #2 }
6710     \tl_put_right:Nn \l__stex_structures_redo_tl {
6711       \tl_set:cn{#1}{ #2 }
6712     }
6713   }
6714 }
6715
6716 \cs_new_protected:Nn \l__stex_structures_prop_do_decls: {
6717   \exp_args:No \stex_iterate_symbols:nn \l_stex_current_type_tl {
6718     \tl_if_empty:nTF{##2}[
6719       \l__stex_structures_do_decl_nomacro:nnnnnnnn{##3}
6720     ]{
6721       \l__stex_structures_do_decl:nnnnnnnn{##2}
6722     }
6723     {##1}{##3}{##4}{##5}{##6}{##7}{##8}{##9}
6724   }
6725 }
6726
6727 \cs_new_protected:Nn \l__stex_structures_do_decl_nomacro:nnnnnnnn {
6728   \prop_if_in:NnF \l__stex_structures_prop {#1} {
6729     \seq_put_left:Nx \l__stex_structures_seq {\tl_to_str:n{#2?#3}}
6730     \prop_put:Nnn \l__stex_structures_prop {#1}{}{
6731       }{
6732         \l_stex_invoke_symbol:nnnnnnn
6733         {#2}
6734         {#3}
6735         {#4}{#5}{#6}{#7}{#8}{#9}
6736       }
6737     }
6738   }
6739 }
6740
6741 \cs_new_protected:Nn \l__stex_structures_do_decl:nnnnnnnn {
6742   \prop_if_in:NnF \l__stex_structures_prop {#1} {
6743     \seq_put_left:Nx \l__stex_structures_seq {\tl_to_str:n{#2?#3}}
6744     \prop_put:Nnn \l__stex_structures_prop {#1}{}{
6745       }{
6746         \l_stex_invoke_symbol:nnnnnnn
6747         {#2}
6748         {#3}
6749         {#4}{#5}{#6}{#7}{#8}{#9}
6750       }
6751   }

```

```

6752 }
6753 \%tl_set:cn{#1}%
6754 % \_stex_invoke_symbol:nnnnnnnN
6755 % {#2}{#3}{#4}{#5}{#6}{#7}{#8}#9
6756 %}
6757 \%tl_put_right:Nn \l__stex_structures_redo_tl {
6758 % \tl_set:cn{#1}%
6759 % \_stex_invoke_symbol:nnnnnnnN
6760 % {#2}{#3}{#4}{#5}{#6}{#7}{#8}#9
6761 % }
6762 %}
6763 }
6764
6765 \cs_new_protected:Nn \__stex_structures_prop_do_notations: {
6766   \exp_args:No \stex_iterate_notations:nn\l_stex_current_type_tl{
6767     \exp_args:NNe \seq_if_in:NnT \l__stex_structures_seq {\tl_to_str:n{##1}}{
6768       \tl_put_right:Nn \l__stex_structures_redo_tl {
6769         \cs_if_exist:cF{\l_stex_notation_##1 _##2_cs}{
6770           \tl_set:cn{\l_stex_notation_##1 _##2_cs}{##4}
6771         }
6772         \cs_if_exist:cF{\l_stex_notation_##1 __cs}{
6773           \tl_set:cn{\l_stex_notation_##1 __cs}{##4}
6774         }
6775       }
6776       \cs_if_exist:cF{\l_stex_notation_##1 _##2_cs}{
6777         \tl_set:cn{\l_stex_notation_##1 _##2_cs}{##4}
6778       }
6779       \cs_if_exist:cF{\l_stex_notation_##1 __cs}{
6780         \tl_set:cn{\l_stex_notation_##1 __cs}{##4}
6781       }
6782       \tl_if_empty:nF{##5}{

6783         \tl_put_right:Nn \l__stex_structures_redo_tl {
6784           \cs_if_exist:cF{\l_stex_notation_##1 _op_##2_cs}{

6785             \tl_set:cn{\l_stex_notation_##1 _op_##2_cs}{##5}
6786           }
6787           \cs_if_exist:cF{\l_stex_notation_##1 _op__cs}{

6788             \tl_set:cn{\l_stex_notation_##1 _op__cs}{##5}
6789           }
6790         }
6791         \cs_if_exist:cF{\l_stex_notation_##1 _op_##2_cs}{

6792           \tl_set:cn{\l_stex_notation_##1 _op_##2_cs}{##5}
6793         }
6794         \cs_if_exist:cF{\l_stex_notation_##1 _op__cs}{

6795           \tl_set:cn{\l_stex_notation_##1 _op__cs}{##5}
6796         }
6797       }
6798     }
6799   }
6800 }

\usestructure
6801 \cs_new_protected:Npn \usestructure #1 {
6802   \stex_get_mathstructure:n{ #1 }
6803   \seq_clear:N \l__stex_structures_imports_seq

```

```

6804 \clist_map_inline:Nn \l_stex_get_symbol_type_tl {
6805   \exp_args:Ne \stex_if_module_exists:nT{\tl_to_str:n{##1}}{
6806     \seq_put_right:Nn \l__stex_structures_imports_seq{##1}
6807   }
6808 }
6809 \seq_map_inline:Nn \l__stex_structures_imports_seq {
6810   \stex_if_do_html:T {
6811     \hbox{\stex_annotation_invisible:nn
6812       {shtml:usemodule=##1} {}}
6813   }
6814   \stex_activate_module:n {##1}
6815 }
6816 }

```

(End of definition for `\usestructure`. This function is documented on page 96.)

## 13.10 Statements

```

6817 <@=stex_statements>
6818
6819 \stex_keys_define:nnnn{statement}{
6820   \str_clear:N \l_stex_key_name_str
6821   \str_clear:N \l_stex_key_mroname_str
6822   \clist_clear:N \l_stex_key_for_clist
6823   \str_clear:N \l_stex_key_args_str
6824   \tl_clear:N \l_stex_key_type_tl
6825   \tl_clear:N \l_stex_key_def_tl
6826   \tl_clear:N \l_stex_key_return_tl
6827   \clist_clear:N \l_stex_key_argtypes_clist
6828 }{
6829   name .str_set:N = \l_stex_key_name_str ,
6830   for .clist_set:N = \l_stex_key_for_clist ,
6831   macro .str_set:N = \l_stex_key_mroname_str ,
6832   % start .str_set:N = \l_stex_key_title_str , % TODO remove
6833   type .tl_set:N = \l_stex_key_type_tl ,
6834   judgment .code:n = {},
6835   from .code:n= {}, % TODO remove
6836   to .code:n={} % TODO remove
6837 }{id,title,style,symargs}

\stex_new_statement:nn
6838 \cs_new_protected:Npn \stex_do_for_list: {
6839   \seq_clear:N \l_stex_fors_seq
6840   \clist_map_inline:Nn \l_stex_key_for_clist {
6841     \exp_args:Ne \stex_get_symbol:n{\tl_to_str:n{##1}}
6842     \seq_put_right:Nx \l_stex_fors_seq
6843     {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
6844   }
6845 }
6846
6847 \cs_new_protected:Nn \__stex_statements_setup:nn {
6848   \str_if_empty:NF \l_stex_key_mroname_str {
6849     \str_if_empty:NT \l_stex_key_name_str {
6850       \str_set_eq:NN \l_stex_key_name_str \l_stex_key_mroname_str

```

```

6851     }
6852   }
6853 \_stex_do_for_list:
6854 \str_if_empty:NF \l_stex_key_name_str {
6855   \__stex_statements_force_id:
6856   \seq_put_right:Nx \l_stex_fors_seq {
6857     \l_stex_current_module_str ? \l_stex_key_name_str
6858   }
6859   \str_set_eq:NN \l_stex_mroname_str \l_stex_key_mroname_str
6860   \str_set:Nn \l_stex_key_role_str {#2}
6861   \stex_symdecl_do:
6862   \exp_args:Nnx \use:nn {\stex_module_add_symbol:nnnnnnN} {
6863     {\l_stex_key_mroname_str}{\l_stex_key_name_str}
6864     {\int_use:N \l_stex_get_symbol_arity_int}
6865     {\l_stex_get_symbol_args_t1}
6866     {#1}{\}\stex_invoke_symbol:
6867   }
6868   \stex_if_do_html:T \_stex_symdecl_html:
6869 }
6870 \str_clear:N \l__stex_statements_uri_str
6871 \str_if_empty:NTF \l_stex_key_name_str {
6872   \stex_debug:nn{statement}{no-name}
6873   \int_compare:nNnTF {\seq_count:N \l_stex_fors_seq} = 1 {
6874     \str_set:Nx \l__stex_statements_uri_str {\seq_item:Nn \l_stex_fors_seq 1}
6875     \stex_debug:nn{statement}{for:~\l__stex_statements_uri_str}
6876   }{
6877     \stex_debug:nn{statement}{no-for}
6878   }
6879 }{
6880   \str_set:Nx \l__stex_statements_uri_str {\l_stex_current_module_str ? \l_stex_key_name_s
6881   \stex_debug:nn{statement}{name:~\l__stex_statements_uri_str}
6882 }
6883 }
6884
6885 \cs_new:Nn \__stex_statements_html_keyvals:nn {
6886   shtml:#1={},
6887   shtml:inline={#2},
6888   \seq_if_empty:NF \l_stex_fors_seq {
6889     shtml:fors=\{\seq_use:Nn \l_stex_fors_seq ,\}
6890   }
6891   \str_if_empty:NF \l_stex_key_id_str {
6892     shtml:id=\{\stex_uri_use:N \l_stex_current_doc_uri ? \l_stex_key_id_str\}
6893   }
6894   \clist_if_empty:NF \l_stex_key_style_clist {
6895     shtml:styles=\{\l_stex_key_style_clist\}
6896   }
6897 }
6898
6899 \cs_new_protected:Nn \stex_new_statement:nnn {
6900   \stex_new_stylable_env:nnnnnnn {#1}{0{} }{
6901     \stex_keys_set:nn{statement}{##1}
6902     #3
6903
6904   \stex_if_smsmode:F {

```

```

6905     \exp_args:Nne \begin{stex_annotation_env}{
6906         \_stex_statements_html_keyvals:nn{#1}{false}
6907     }
6908     \tl_set_eq:NN \thistitle \l_stex_key_title_tl
6909     \str_set_eq:NN \thisname \l_stex_key_name_str
6910     \clist_set_eq:NN \thisfor \l_stex_key_for_str
6911     \stex_if_html_backend:TF {
6912         \noindent
6913         \stex_annotation:nn{shtml:statementtitle={}}{\_stex_annotation_force_break:n\l_stex_key_}
6914     }
6915     \stex_style_apply:
6916 }
6917     \_stex_do_id:
6918     \stex_smsmode_do:
6919 }{
6920     \stex_if_smsmode:F {
6921         \stex_if_html_backend:F \stex_style_apply:
6922         \end{stex_annotation_env}
6923     }
6924 }{}{}{s}
6925 \stex_sms_allow_env:n{s#1}
6926
6927 \tl_if_empty:nF{#2}{
6928     \exp_after:wN \NewDocumentCommand \cs:w inline#2\cs_end: { 0{} m}{
6929         \group_begin:
6930             \stex_keys_set:nn{statement}{##1}
6931             #3
6932             \_stex_do_id:
6933             \stex_if_smsmode:F{
6934                 \exp_args:Ne \stex_annotation:nn{\_stex_statements_html_keyvals:nn{#1}{true}}{
6935                     \_stex_annotation_force_break:n##2}
6936                 }
6937             }
6938             \group_end:
6939             \stex_smsmode_do:
6940         }
6941         \exp_after:wN \stex_sms_allow_escape:N\cs:w inline#2\cs_end:
6942     }
6943 }
6944
6945 \cs_new_protected:Nn \__stex_statements_setup_def: {
6946     \stex_if_smsmode:F{
6947         \seq_map_inline:Nn \l_stex_fors_seq {
6948             \stex_ref_new_sym_target:n##1
6949         }
6950     }
6951     \stex_reactivate_macro:N \definiendum
6952     \stex_reactivate_macro:N \defnotation
6953     \stex_reactivate_macro:N \defname
6954     \stex_reactivate_macro:N \Defname
6955     \stex_reactivate_macro:N \varbind
6956 }
6957
6958 \cs_new_protected:Nn \__stex_statements_force_id: {

```

```

6959 \str_if_empty:NT \l_stex_key_id_str {
6960   \_stex_ref_new_id:n{}
6961   \str_set_eq:NN \l_stex_key_id_str \l__stex_refs_str
6962 }
6963 }
6964
6965 \stex_new_statement:nnn{definition}{def}{
6966   \__stex_statements_force_id:
6967   \__stex_statements_setup:nn{}{}
6968   \__stex_statements_setup_def:
6969   \stex_reactivate_macro:N \definiens
6970 }
6971 \stex_new_statement:nnn{assertion}{ass}{
6972   \__stex_statements_setup:nn{}{assertion}
6973   \stex_if_smsmode:F{
6974     \seq_map_inline:Nn \l_stex_fors_seq {
6975       \stex_ref_new_sym_target:n{##1}
6976     }
6977   }
6978   \stex_reactivate_macro:N \varbind
6979   \stex_reactivate_macro:N \conclusion
6980   \stex_reactivate_macro:N \premise
6981   \stex_reactivate_macro:N \definiendum
6982   \stex_reactivate_macro:N \defnotation
6983   \stex_reactivate_macro:N \defname
6984   \stex_reactivate_macro:N \Defname
6985 }
6986 \stex_new_statement:nnn{example}{ex}{\__stex_statements_setup:nn{}{example}}
6987 \stex_new_statement:nnn{paragraph}{}{
6988   \clist_if_in:NnTF \l_stex_key_style_clist {symdoc}{
6989     \__stex_statements_force_id:
6990     \__stex_statements_setup:nn{}{}
6991     \__stex_statements_setup_def:
6992   }{
6993     \__stex_statements_setup:nn{}{}
6994   }
6995 }

```

(End of definition for `\stex_new_statement:nn`. This function is documented on page ??.)

#### definiendum

```

6996 \cs_new_protected:Nn \__stex_statements_do_deref:nn {
6997   \stex_if_html_backend:T{\ifvmode\indent\fi}
6998   \group_begin:
6999   \stex_get_symbol:n{#1}
7000   \bool_if:NTF \l_stex_allow_semantic_bool{
7001     \str_set:Nx \l_stex_current_symbol_str
7002       {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
7003     \str_if_in:NnT \l_stex_get_symbol_name_str / {
7004       \str_set:Nx \l_stex_get_symbol_name_str {
7005         \exp_after:wN \stex_split_slash: \l_stex_get_symbol_name_str
7006         /\_stex_args_end:
7007     }
7008   }

```

```

7009   \exp_args:No \stex_ref_new_sym_target:n \l_stex_current_symbol_str
7010   \def\comp{\_defcomp}
7011   \stex_annotation:nn{shtml:definiendum=\l_stex_current_symbol_str}{\comp{#2}}
7012 }{
7013   \msg_error:nnnx{stex}{error/notallowed}{#1}{\l_stex_current_symbol_str}
7014 }
7015 \group_end:
7016 }
7017
7018 \NewDocumentCommand \defnotation{ m } {
7019   \stex_next_symbol:n { \def\comp{\_defcomp}}#1
7020 }
7021 \stex_deactivate_macro:Nn \defnotation {definition~environments}
7022
7023 \NewDocumentCommand \definiendum { O{} m m} {
7024   \stex_keys_set:nn{symname}{ #1 }
7025   \__stex_statements_do_deref:nn{#2}{#3}
7026 }
7027 \stex_deactivate_macro:Nn \definiendum {definition~environments}
7028
7029 \NewDocumentCommand \defname { O{} m } {
7030   \stex_keys_set:nn{symname}{#1}
7031   \__stex_statements_do_deref:nn{#2}{
7032     \l_stex_key_pre_tl\l_stex_get_symbol_name_str\l_stex_key_post_tl
7033   }
7034 }
7035 \stex_deactivate_macro:Nn \defname {definition~environments}
7036
7037 \NewDocumentCommand \Defname { O{} m } {
7038   \stex_keys_set:nn{symname}{#1}
7039   \__stex_statements_do_deref:nn{#2}{
7040     \l_stex_key_pre_tl\exp_after:wN\l_stex_capitalize:n\l_stex_get_symbol_name_str\l_stex_key
7041   }
7042 }
7043 \stex_deactivate_macro:Nn \Defname {definition~environments}
7044
7045
7046 \NewDocumentCommand \defniens { O{} m }{
7047   \group_begin:
7048   \str_clear:N \l_stex_get_symbol_name_str
7049   \tl_if_empty:nF {#1} {
7050     \stex_get_symbol:n { #1 }
7051     \str_set:Nx \l__stex_statements_uri_str
7052       {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
7053   }
7054   \str_if_empty:NT \l__stex_statements_uri_str {
7055     \msg_error:nn{stex}{error/defniensfor}
7056   }
7057   \stex_debug:nn{defniens}{Checking~\l__stex_statements_uri_str}
7058
7059 \exp_args:No \stex_add_defniens:nn \l__stex_statements_uri_str{#2}
7060
7061 \group_end:
7062 \stex_smsmode_do:

```

```

7063 }
7064 \stex_deactivate_macro:Nn \definiens {definition-environments}
7065 \stex_sms_allow_escape:N \definiens
7066
7067 \cs_new_protected:Nn \stex_add_definiens:nn {
7068   \exp_args:Nno \stex_if_starts_with:nnT{\#1} \l_stex_current_module_str {
7069     \prop_map_inline:cN{c_stex_module}\l_stex_current_module_str _symbols_prop} {
7070       \stex_debug:nn{definiens}{\#1 == \l_stex_current_module_str?##1}
7071       \str_if_eq:nt{\#1}{\l_stex_current_module_str?##1} {
7072         \prop_map_break:n{\stex_add_definiens_inner:nnnnnnnn ##2}
7073       }
7074     }
7075   }
7076 \stex_if_smsmode:F{
7077   \stex_annotation:nn{ shtml:definiens=\#1}{\#2} \%_stex_annotation_force_break:n{ \#2 }
7078 }
7079 }
7080 }
7081 }
7082
7083 \cs_new_protected:Nn \stex_add_definiens_inner:nnnnnnnn {
7084   \stex_debug:nn{definiens}{Adding-definiens-to-\l_stex_current_module_str?#2}
7085   \prop_gput:cnn{c_stex_module}\l_stex_current_module_str _symbols_prop
7086   {\#2}{\#1}{\#2}{\#3}{\#4}{defed}{\#6}{\#7}{\#8}}
7087 }
7088
7089 \NewDocumentCommand \varbind {m} {
7090   \clist_map_inline:nn {\#1} {
7091     \stex_get_var:n {\#\#1}
7092     \stex_if_do_html:T {
7093       \stex_annotation_invisible:nn {shtml:bind=\l_stex_get_symbol_name_str}{}
7094     }
7095   }
7096 }
7097 \stex_deactivate_macro:Nn \varbind {definition-or-assertion-environments}
7098
7099 \NewDocumentCommand \conclusion { O{} m} {
7100   \group_begin:
7101   \str_clear:N \l_stex_get_symbol_name_str
7102   \tl_if_empty:nF {\#1} {
7103     \stex_get_symbol:n { \#1 }
7104     \str_set:Nx \l__stex_statements_uri_str
7105       {\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
7106   }
7107   \str_if_empty:NT \l__stex_statements_uri_str {
7108     \msg_error:nn{stex}{error/conclusionfor}
7109   }
7110   \stex_annotation:nn{ shtml:conclusion=\l__stex_statements_uri_str}{\#2} \%_stex_annotation_force_break:n{ \#2 }
7111 }
7112
7113 \group_end:
7114 }
7115 \stex_deactivate_macro:Nn \conclusion {assertion-environments}
7116

```

```

7117 \NewDocumentCommand \premise {O{} m} {
7118   \tl_if_empty:nF {#1} {
7119     \stex_debug:nn{Here:}{Variable~#1}
7120     \exp_args:Nne\use:nn{\vardef}{{v#1}[name=#1]{#1}}
7121   }
7122   \stex_annotate:nn{shtml:premise={#1}}{#2}
7123 }
7124 \stex_deactivate_macro:Nn \premise {assertion-environments}

```

(End of definition for `definiendum`. This function is documented on page 100.)

## 13.11 Proofs

We first define some keys for the `sproof` environment.

```

7125 <@=stex_proof>
7126 \stex_keys_define:nnnn{ spf }{
7127   \tl_clear:N \l_stex_key_for_clist
7128   \tl_clear:N \l_stex_key_from_tl
7129   \tl_set_eq:NN \l_stex_key_proofend_tl \__stex_proof_proof_box_tl
7130   \tl_clear:N \l_stex_key_continues_tl
7131   \tl_clear:N \l_stex_key_term_tl
7132   \tl_clear:N \l_stex_key_functions_tl
7133   \tl_clear:N \l_stex_key_method_tl
7134     \bool_set_false:N \l_stex_key_hide_bool
7135   }{
7136     for .clist_set:N = \l_stex_key_for_clist ,
7137     from .tl_set:N = \l_stex_key_from_tl ,
7138     proofend .tl_set:N = \l_stex_key_proofend_tl,
7139     continues .tl_set:N = \l_stex_key_continues_tl,
7140     functions .tl_set:N = \l_stex_key_functions_tl,
7141     term .tl_set:N = \l_stex_key_term_tl,
7142     method .tl_set:N = \l_stex_key_method_tl,
7143     hide .bool_set:N = \l_stex_key_hide_bool
7144 }{id,style,title}
7145
7146 \bool_set_true:N \l__stex_proof_inc_counter_bool

```

For proofs, we will have to have deeply nested structures of enumerated list-like environments. However, L<sup>A</sup>T<sub>E</sub>X only allows `enumerate` environments up to nesting depth 4 and general list environments up to listing depth 6. This is not enough for us. Therefore we have decided to go along the route proposed by Leslie Lamport to use a single top-level list with dotted sequences of numbers to identify the position in the proof tree. Unfortunately, we could not use his `pf.sty` package directly, since it does not do automatic numbering, and we have to add keyword arguments all over the place, to accomodate semantic information.

```

7147 \intarray_new:Nn\l__stex_proof_counter_intarray{50}
7148 \cs_new_protected:Npn \__stex_proof_insert_number: {
7149   \int_set:Nn \l_tmpa_int {1}
7150   \bool_while_do:nn {
7151     \int_compare_p:nNn {
7152       \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int
7153     } > 0
7154   }{

```

```

7155     \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int .
7156     \int_incr:N \l_tmpa_int
7157   }
7158 }
7159 \cs_new_protected:Nn \__stex_proof_number_as_string:N {
7160   \str_clear:N #1
7161   \int_set:Nn \l_tmpa_int {1}
7162   \bool_while_do:nn {
7163     \int_compare_p:nNn {
7164       \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int
7165     } > 0
7166   }{
7167     \str_put_right:Nx #1 {\intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int .}
7168     \int_incr:N \l_tmpa_int
7169   }
7170 }
7171
7172 \cs_new_protected:Npn \__stex_proof_inc_counter: {
7173   \int_set:Nn \l_tmpa_int {1}
7174   \bool_while_do:nn {
7175     \int_compare_p:nNn {
7176       \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int
7177     } > 0
7178   }{
7179     \int_incr:N \l_tmpa_int
7180   }
7181   \int_compare:nNf \l_tmpa_int = 1 {
7182     \int_decr:N \l_tmpa_int
7183   }
7184   \intarray_gset:Nnn \l__stex_proof_counter_intarray \l_tmpa_int {
7185     \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int + 1
7186   }
7187 }
7188
7189 \cs_new_protected:Npn \__stex_proof_add_counter: {
7190   \int_set:Nn \l_tmpa_int {1}
7191   \bool_while_do:nn {
7192     \int_compare_p:nNn {
7193       \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int
7194     } > 0
7195   }{
7196     \int_incr:N \l_tmpa_int
7197   }
7198   \intarray_gset:Nnn \l__stex_proof_counter_intarray \l_tmpa_int { 1 }
7199 }
7200
7201 \cs_new_protected:Npn \__stex_proof_remove_counter: {
7202   \int_set:Nn \l_tmpa_int {1}
7203   \bool_while_do:nn {
7204     \int_compare_p:nNn {
7205       \intarray_item:Nn \l__stex_proof_counter_intarray \l_tmpa_int
7206     } > 0
7207   }{
7208     \int_incr:N \l_tmpa_int

```

```

7209   }
7210   \int_decr:N \l_tmpa_int
7211   \intarray_gset:Nnn \l__stex_proof_counter_intarray \l_tmpa_int { 0 }
7212 }

spfsketch
7213 \newenvironment{spfsketchenv}{}{%
7214   \stex_new_stylable_cmd:nnnn{spfsketch}{O{}} m{\par
7215     \begin{spfsketchenv}
7216       \stex_keys_set:nn{spf}{#1}
7217       \stex_do_for_list:
7218       \stex_do_id:
7219       \exp_args:Ne \stex_annotation:nn{%
7220         shtml:proofsketch={%
7221           \seq_if_empty:NF \l_stex_fors_seq {%
7222             \seq_use:Nn \l_stex_fors_seq ,
7223           }%
7224         }%
7225       }{%
7226         \stex_style_apply:
7227         #2
7228       }%
7229     \end{spfsketchenv}
7230   }{%
7231     \noindent\emph{\spfsketchenvautorefname :}-
7232   }

```

(End of definition for `spfsketch`. This function is documented on page ??.)

`\sproofend` This macro places a little box at the end of the line if there is space, or at the end of the next line if there isn't

```

7233 \tl_set:Nn \__stex_proof_box_tl {
7234   \ltx@ifpackageloaded{amssymb}{$\square$}%
7235   \hbox{\vrule\vbox{\hrule width 6 pt\vskip 6pt\hrule}\vrule}%
7236 }
7237 }

7238 \tl_set:Nn \sproofend {
7239   \tl_if_empty:NF \l_stex_key_proofend_tl {%
7240     \hfil\null\nobreak\hfill\l_stex_key_proofend_tl\par\smallskip
7241   }%
7242 }
7243 }


```

(End of definition for `\sproofend`. This function is documented on page ??.)

`\stexcommentfont`

```

7244 \cs_new_protected:Npn \stexcommentfont {
7245   \small\itshape
7246 }


```

(End of definition for `\stexcommentfont`. This function is documented on page ??.)

```

sproof (env.)
7247 \cs_new_protected:Nn \__stex_proof_start_list:n {
7248   \begin{list}{}{%
7249     \setlength\topsep{0pt}
7250     \setlength\parsesep{0pt}
7251     \setlength\rightmargin{0pt}
7252   }\item[#1]
7253 }
7254 \cs_new_protected:Nn \__stex_proof_end_list: {
7255   \end{list}
7256 }
7257
7258 \cs_new_protected:Nn \__stex_proof_html: {
7259   \stex_annotation_invisible:n{\hbox{
7260     \tl_if_empty:NF \l_stex_key_term_tl {
7261       $\stex_annotation:nn{shtml:proofterm={}}{\l_stex_key_term_tl}$
7262     }
7263     \tl_if_empty:NF \l_stex_key_method_tl {
7264       \stex_annotation:nn{shtml:proofmethod={}}{\l_stex_key_method_tl}
7265     }
7266   }}
7267 }
7268
7269 \cs_new_protected:Nn \__stex_proof_html_env:n {
7270   \exp_args:Nne \begin{stex_annotation_env}{%
7271     shtml:#1=%
7272       \seq_if_empty:NF \l_stex_fors_seq {%
7273         \seq_use:Nn \l_stex_fors_seq ,%
7274       }%
7275     }%
7276     \bool_if:NT \l_stex_key_hide_bool {%
7277       shtml:proofhide=true
7278     }%
7279   }%
7280   \__stex_proof_html:
7281 }
7282
7283 \bool_set_false:N \l_stex_proof_in_spfblock_bool
7284 \cs_new_protected:Nn \__stex_proof_begin_proof:nn {\par
7285   \intarray_gzero:N \l_stex_proof_counter_intarray
7286   \intarray_gset:Nnn \l_stex_proof_counter_intarray 1 1
7287   \stex_keys_set:nn{spfsteps}{#1}
7288   \stex_do_for_list:
7289   \stex_if_do_html:T {
7290     \__stex_proof_html_env:n{proof}
7291   }
7292   \seq_map_inline:Nn \l_stex_fors_seq {
7293     \stex_debug:nn{definiens}{Adding-definiens-to~##1}
7294     \stex_add_definiens:nn {##1}{\STEXinvisibl{proven}}
7295   }
7296   \stex_style_apply:
7297   \stex_do_id:
7298   \stex_reactivate_macro:N \subproof
7299   \stex_reactivate_macro:N \spfstep

```

```

7300  \stex_reactivate_macro:N \conclude
7301  \stex_reactivate_macro:N \assumption
7302  \stex_reactivate_macro:N \eqstep
7303  \stex_reactivate_macro:N \yield
7304  \stex_reactivate_macro:N \spfblock
7305  \stex_reactivate_macro:N \spfjust
7306  \stex_annotate:nn{shtml:prooftitle={}}{#2}
7307  \stex_if_do_html:T{
7308      \begin{stex_annotate_env}{shtml:proofbody={}}
7309  }
7310 }
7311 \stex_new_stylable_env:nnnnnnn{proof}{0{} m}{
7312     \__stex_proof_begin_proof:nn{#1}{#2}
7313     \bool_set_true:N\l__stex_proof_in_spfblock_bool\__stex_proof_start_list:n{}
7314     \group_begin:\stexcommentfont
7315 }{
7316     \stex_style_apply:
7317     \stex_if_do_html:T{\end{stex_annotate_env}\end{stex_annotate_env}}
7318 }{
7319     \emph{\sproofautorefname :}-
7320 }{
7321     \sproofend
7322 }{s}
7323 \AddToHook{env/sproof/end}{
7324     \bool_if:NT\l__stex_proof_in_spfblock_bool {
7325         \group_end:\__stex_proof_end_list:
7326     }
7327 }
7328
7329 \stex_new_stylable_env:nnnnnnn{proof*}{0{}>{
7330     \__stex_proof_begin_proof:nn{#1}{}
7331     \bool_set_false:N\l__stex_proof_in_spfblock_bool
7332 }{
7333     \stex_style_apply:
7334     \stex_if_do_html:T{\end{stex_annotate_env}\end{stex_annotate_env}}
7335 }{
7336     \emph{Proof:}-
7337 }{
7338     \sproofend
7339 }{s}

subproof (env.)
7340 \str_set_eq:NN \subproofautorefname \spfstepautorefname
7341 \stex_new_stylable_env:nnnnnnn{subproof}{s 0{} m}{\par
7342     \stex_keys_set:nn{spf}{#2}
7343     \stex_do_for_list:
7344     \stex_if_do_html:T {
7345         \__stex_proof_html_env:n{subproof}
7346     }
7347     \seq_map_inline:Nn \l_stex_fors_seq {
7348         \stex_debug:nn{definiens}{Adding-definiens-to-##1}
7349         \stex_add_definiens:nn {##1}{\STEXinvisibl{proven}}
7350     }
7351

```

```

7352 \IfBooleanTF #1 {
7353   \stex_style_apply:
7354   \str_if_empty:NF \l_stex_key_id_str {
7355     \__stex_proof_number_as_string:N \@currentlabel
7356     \str_set:Nx \@currentHref{subproof.\@currentlabel}
7357     \__stex_do_id:
7358   }
7359   \bool_set_false:N \l__stex_proof_in_spfblock_bool
7360   \stex_annotation:nn{shtml:prooftitle={}}{#3}
7361 }{
7362   \bool_if:NTF \l__stex_proof_in_spfblock_bool {
7363     \str_if_empty:NF \l_stex_key_id_str {
7364       \__stex_proof_number_as_string:N \@currentlabel
7365       \str_set:Nx \@currentHref{subproof.\@currentlabel}
7366       \__stex_do_id:
7367     }
7368     \__stex_proof_start_list:n\__stex_proof_insert_number:
7369     \stex_annotation:nn{shtml:prooftitle={}}{#3}
7370     \__stex_proof_add_counter:
7371     \stex_style_apply:
7372   }{
7373     \stex_annotation:nn{shtml:prooftitle={}}{#3}
7374     \stex_style_apply:
7375     \__stex_do_id:
7376   }
7377 }
7378 \stex_if_do_html:T{
7379   \begin{stex_annotation_env}{shtml:proofbody={}}
7380 }
7381 \bool_if:NT \l__stex_proof_in_spfblock_bool {\group_begin:\stexcommentfont}
7382 }{
7383   \stex_style_apply:
7384   \bool_if:NT \l__stex_proof_in_spfblock_bool \__stex_proof_inc_counter:
7385   \stex_if_do_html:T{\end{stex_annotation_env}}
7386   \bool_if:NT \l__stex_proof_in_spfblock_bool \__stex_proof_end_list:
7387   \stex_if_do_html:T{\end{stex_annotation_env}}
7388   \aftergroup \__stex_proof_inblock_restore:
7389 }{}{}{}
7390 \AddToHook{env/subproof/before}{
7391   \bool_if:NT \l__stex_proof_in_spfblock_bool \group_end:
7392 }
7393 \AddToHook{env/subproof/end}{
7394   \bool_if:NT \l__stex_proof_in_spfblock_bool {
7395     \group_end:\__stex_proof_remove_counter:
7396     \%__stex_proof_end_list:
7397   }
7398 }
7399 \stex_deactivate_macro:Nn \subproof {sproof~environments}
7400
7401 \cs_new_protected:Nn \__stex_proof_inblock_restore: {
7402   \bool_if:NT \l__stex_proof_in_spfblock_bool {
7403     \group_begin:\stexcommentfont
7404   }
7405 }

```

```

\spfstep
\conclude
\assumption
    \have
\eqstep
7406 \stex_keys_define:nnnn { spfsteps } {
7407     \clist_clear:N \l_stex_key_for_clist
7408     \str_clear:N \l_stex_key_name_str
7409     \tl_clear:N \l_stex_key_method_tl
7410     \tl_clear:N \l_stex_key_term_tl
7411 }
7412 \{
7413     for .clist_set:N = \l_stex_key_for_clist ,
7414     method .tl_set:N = \l_stex_key_method_tl,
7415     term .tl_set:N = \l_stex_key_term_tl,
7416     name .str_set_x:N = \l_stex_key_name_str
7417     % todo: style=inline
7418 }{id,style,title}
7419
7420 \newenvironment{spfstepenv}{
7421     \str_set_eq:NN \spfstepenvautorefname \spfstepautorefname
7422 }{}
7423
7424 \cs_new_protected:Nn \__stex_proof_step_html:nn {
7425     \stex_if_do_html:TF{
7426         \exp_args:Ne \stex_annotation:nn{
7427             shtml:spf#1={
7428                 \seq_if_empty:NF \l_stex_fors_seq {
7429                     \seq_use:Nn \l_stex_fors_seq ,
7430                 }
7431             }
7432             \str_if_empty:NF \l_stex_key_name_str {
7433                 shtml:stepname={\l_stex_key_name_str}
7434             }
7435         }{
7436             \__stex_proof_html:
7437             #2
7438         }
7439     }{ #2 }
7440 }
7441
7442 \cs_new_protected:Nn \__stex_proof_make_step_macro:Nnnnn {
7443     \NewDocumentCommand #1 { s O{} +m } {
7444         \bool_if:NT \l__stex_proof_in_spfblock_bool \group_end:
7445         \stex_keys_set:nn{spfsteps}{##2}
7446         \str_if_empty:NF \l_stex_key_name_str {
7447             \stex_debug:nn{Here:}{Variable-\l_stex_key_name_str}
7448             \exp_args:Nne\use:nn{\vardef}{\v\l_stex_key_name_str}[name=\l_stex_key_name_str]{\l_stex_key_name_str}
7449         }
7450
7451         \begin{spfstepenv}
7452             \str_if_empty:NF \l_stex_key_id_str {
7453                 \__stex_proof_number_as_string:N \@currentlabel
7454                 \str_set:Nx \@currentHref{spfstep.\@currentlabel}
7455                 \stex_do_id:
7456             }
7457
7458             \bool_if:NTF \l__stex_proof_in_spfblock_bool {

```

```

7459     \IfBooleanTF ##1 {
7460         \__stex_proof_step_html:nn{#2}{##3}
7461     }{
7462         \__stex_proof_step_html:nn{#2}{\__stex_proof_start_list:n{#3} ##3 \__stex_proof_end_}
7463         #5
7464     }
7465     \end{spfstepenv}
7466     \group_begin:\stexcommentfont
7467 }{
7468     \__stex_proof_step_html:nn{#2}{##3}
7469     \end{spfstepenv}
7470 }
7471 }
7472 \stex_deactivate_macro:Nn #1 {sproof~environments}
7473 }
7474
7475 \__stex_proof_make_step_macro:Nnnnn \assumption {assumption} \__stex_proof_insert_number: {}
7476 \__stex_proof_make_step_macro:Nnnnn \conclude {conclusion} {${\Rightarrow}{}$} {} {}
7477 \__stex_proof_make_step_macro:Nnnnn \spfstep {step} \__stex_proof_insert_number: {} \__stex_
7478
7479 \NewDocumentCommand \eqstep {s m} {
7480     \bool_if:NTF \l__stex_proof_in_spfblock_bool {
7481         \group_end:
7482         \IfBooleanTF #1 {
7483             \__stex_proof_step_html:nn{eqstep}{$= #2$}
7484         }{
7485             \__stex_proof_step_html:nn{eqstep}{\__stex_proof_start_list:n{$=$} $#2$ \__stex_proof_}
7486         }
7487         \group_begin:\stexcommentfont
7488     }{
7489         \__stex_proof_step_html:nn{eqstep}{$= #2$}
7490     }
7491 }
7492 \stex_deactivate_macro:Nn \eqstep {sproof~environments}
7493
7494 \NewDocumentCommand \yield {+m} {
7495     \stex_annotation:nn{shtml:proofterm={}}{ #1 }
7496 }
7497 \stex_deactivate_macro:Nn \yield {sproof~environments}
7498
7499 \NewDocumentEnvironment{spfblock}{}{
7500     \bool_set_false:N \l__stex_proof_in_spfblock_bool
7501 }{
7502     \aftergroup\__stex_proof_inblock_restore:
7503 }
7504 \stex_deactivate_macro:Nn \spfblock {sproof~environments}
7505 \AddToHook{env/spfblock/before} {
7506     \bool_if:NT \l__stex_proof_in_spfblock_bool \group_end:
7507 }
7508
7509 \newcommand\spfjust[1]{
7510     \stex_annotation:nn{spfjust={}}{ #1 }
7511 }
7512

```

7513 \stex\_deactivate\_macro:Nn \spfjust {sproof~environments}

(End of definition for `\spfstep` and others. These functions are documented on page ??.)

## 13.12 Metatheory

```

7514 <@=stex_meta>
7515 \group_begin:
7516   \cs_set:Npn \__stex_modules_persist_module: {}
7517   \cs_set:Npn \stex_check_term:n #1 {}
7518   \cs_set:Npn \stex_sref_do_aux:n #1 { #1 }
7519   \bool_set_false:N \stex_html_do_output_bool
7520   \bool_set_false:N \c_stex_check_terms_bool
7521   \stex_uri_resolve:Nn \l_stex_current_ns_uri {http://mathhub.info/sTeX/meta}
7522   \stex_module_setup:n{Metatheory}

7523
7524 \symdef{of~type}[args=ii,invisible]{#1}
7525 \notation{of~type}[colon]{#1 \mathbin{\comp{:}} #2}

7526
7527 \symdef{apply}[args=ia,prec=0;\infprec x\infprec]{#1\mathopen{\comp{} #2 \mathclose{\comp{}}}}
7528 \notation{apply}[\lambda]{#1\mathrel{\lambda}\mathrel{\argsep{#2}{;}}}
7529 \notation{apply}[\infixop]{\argsep{#2}{\mathbin{#1}}}
7530 \notation{apply}[\infixrel]{\argsep{#2}{\mathrel{#1}}}

7531 %
7532 % structures
7533 \symdef{module~type}[args=i,op=\mathhtt{MOD}]
7534 {\mathopen{\comp{\mathhtt{MOD}}}\#1\mathclose{\comp{}}}}
7535 \symdef{module~type~merge}[args=a,op=\oplus]
7536 {\argsep{#1}{\mathbin{\comp{\oplus}}}}
7537 \symdef{anonymous~record}[args=a]
7538 {\mathopen{\comp{[}}\#1\mathclose{\comp{]}}}}
7539 \symdef{record~field}[args=2]{#1\comp{.}\#2}
7540 \symdecl*{record~type}

7541 \symdecl{mathstruct}[name=mathematical-structure,args=a] % TODO
7542 \notation{mathstruct}[angle,prec=nobrackets]
7543 {\mathopen{\comp{\langle}}\#1\mathclose{\comp{\rangle}}}
7544 \notation{mathstruct}[parens,prec=nobrackets]
7545 {\mathopen{\comp{)}}\#1\mathclose{\comp{}}}

7546 %
7547 % sequences
7548 \symdef{ellipses}[ldots]{\ldots}
7549 \symdef{sequence~expression}[comma,args=a]{#1}
7550 \symdef{sequence~type}[args=1]{#1^{\comp{\ast}}}
7551 \symdef{sequence~map}[args=ia]{
7552   \comp{\mathit{map}}\mathopen{\comp{}}\#1\mathclose{\mathpunct{\comp{,}}}
7553   \#2\mathclose{\comp{}}}
7554 }

7555 \iffalse
7556 % binder (\forall, \Pi, \lambda etc.)
7557 \symdef{pibind}[name=dependent function type,prec=nobrackets,
7558 op=(\cdot)\mathrel{\to}\mathrel{\cdot},args=Bi,assoc=pre]
7559 {\argmap{#1}{\mathopen{\comp{}}\#1\mathclose{\comp{}}}}
7560
7561 
```

```

7562     }{\mathbin{\backslash comp{\backslash to}}} \mathbin{\backslash comp{\backslash to}} #2}
7563 \notation{pibind}[forall]{\comp\forall #1\mathpunct{\comp.} #2}
7564 \notation{pibind}[Pi]{\mathop{\backslash comp\prod}\c_math_subscript_token{#1}#2}
7565
7566 \symdef{mapbind}[name=lambda,mapsto,prec=nobrackets,op=\mapsto,args=Bi,assoc=pre]
7567   { #1 \mathrel{\backslash comp\mapsto} #2}
7568 \notation{mapbind}[lambda,prec=nobrackets,op=\lambda]
7569   {\comp\lambda #1 \mathpunct{\comp.} #2}
7570 \fi
7571 \symdecl{bind}[args=Bi,assoc=pre]
7572 \notation{bind}[depfun,prec=nobrackets,op=(\cdotdot)\; \backslash to\; \cdotdot]
7573   {\mathopen{\backslash comp{}} #1 \mathclose{\backslash comp{}}\mathbin{\backslash comp{\backslash to}}} #2}
7574 \notation{bind}[forall]{\comp\forall #1.\;#2}
7575 \notation{bind}[Pi]{\mathop{\backslash comp\prod}\c_math_subscript_token{#1}#2}
7576
7577 \symdef{implicit-bind}[args=Bi,assoc=pre]{\mathopen{\backslash comp{}} #1 \mathclose{\backslash comp{}}\c_math}
7578
7579 \symdecl*{integer-literal}
7580 \notation{integer-literal}{\mathbb{Z}}
7581
7582 \symdecl*{ordinal}
7583 \notation{ordinal}{\mathit{Ord}}
7584
7585 % propositions
7586 \symdef{prop}[name=proposition]{\mathit{Prop}}
7587 \symdef{judgment-holds}[args=i,role=judgment]{\comp\vdash ;#1}
7588
7589 % any object
7590 \symdef{object}{\mathit{Obj}}
7591
7592 % TODO DELETE
7593 \symdef{aseqdots}[args=a,prec=nobrackets]
7594   { #1\comp{, \ldots} }%{##1\comp,##2}
7595 \symdef{aseqfromto}[args=ai,prec=nobrackets]
7596   { #1\comp{, \ldots, }#2 }%{##1\comp,##2}
7597 \symdef{aseqfromtovia}[args=aии,prec=nobrackets]
7598   { #1\comp{, \ldots, }#2\comp{, \ldots, }#3 }%{##1\comp,##2}
7599
7600
7601 \stex_close_module:
7602 \stex_uri_add_module:NNN \l_stex_metatheory_uri \l_stex_current_ns_uri {Metatheory}
7603 \global \let \l_stex_metatheory_uri \l_stex_metatheory_uri
7604 \global \let \c_stex_default_metatheory \l_stex_metatheory_uri
7605 \group_end:

```

### 13.13 MMT Interfaces

```

7606 <@@=todo>
7607 \cs_new_protected:Npn \MSC #1 {}
\MMTinclude
7608 \stex_new_stylable_cmd:nnnn{MMTinclude}{m}{%
7609   \stex_annotation_invisible:nn{shtml:import={#1}}{}%
7610 }{}%

```

```

7611 \stex_deactivate_macro:Nn \MMTinclude {module~environments}
7612 \stex_every_module:n {\stex_reactivate_macro:N \MMTinclude}
```

(End of definition for \MMTinclude. This function is documented on page ??.)

### \MMTrule

```

7613 \NewDocumentCommand \MMTrule {m m} {
7614   \tl_if_empty:nTF{#2}{\seq_clear:N \l_tmpa_seq} {
7615     \seq_set_split:Nnn \l_tmpa_seq , {#2}
7616   }
7617   \int_zero:N \l_tmpa_int
7618   \stex_annotation_invisible:n{
7619     $
7620     \stex_annotation:nn{shtml:rule={scala://#1}}{
7621       \stex_annotation_force_break:n{
7622         \seq_if_empty:NF \l_tmpa_seq {
7623           \seq_map_inline:Nn \l_tmpa_seq {
7624             \int_incr:N \l_tmpa_int
7625             \stex_annotation:nn{
7626               shtml:argmode=i,
7627               shtml:arg={\int_use:N \l_tmpa_int}
7628             }{ ##1 }
7629           }
7630         }
7631       }
7632     }$}
7633   }
7634 }
7635 \stex_deactivate_macro:Nn \MMTrule {module~environments}
7636 \stex_every_module:n{\stex_reactivate_macro:N \MMTrule}
```

(End of definition for \MMTrule. This function is documented on page ??.)

### mmtinterface (env.)

```

7637 \NewDocumentEnvironment { mmtinterface } { O{} m m } {
7638   \stex_module_setup_top_nosig:n { #3 }
7639   \str_set_eq:NN \l__todo_mmt_module_str \l_stex_current_module_str
7640   \str_clear:N \l_stex_current_module_str
7641   \stex_keys_set:nn { smodule }{ #1 }
7642   \stex_module_setup:n{ #2 }
7643   \str_set_eq:NN \l__todo_stex_module_str \l_stex_current_module_str
7644   \stex_debug:nn{mmt}{Interface~\l__todo_stex_module_str^~Jfor~\l__todo_mmt_module_str}
7645
7646 \stex_if_do_html:T {
7647   \exp_args:Nne \begin{stex_annotation_env} {
7648     shtml:theory={\l_stex_current_module_str},
7649     shtml:language={ \l_stex_current_language_str},
7650     shtml:signature={}
7651     \tl_if_empty:NF \l_stex_metatheory_uri {
7652       shtml:metatheory={\stex_uri_use:N \l_stex_metatheory_uri}
7653     }
7654   }
7655   \stex_annotation_invisible:n{}
7656   \stex_annotation_invisible:nn
7657     {shtml:import=\l__todo_mmt_module_str} {}
```

```

7658 }
7659 \stex_module_add_code:x{
7660   \stex_activate_module:n{ \l_todo_mmt_module_str }
7661 }
7662 \stex_module_add_morphism:nonn
7663   {}{\l_todo_mmt_module_str}{import}{}
7664 \stex_reactivate_macro:N \mmtdef
7665 \stex_smsmode_do:
7666 }{
7667   \str_set_eq:NN \l_stex_current_module_str \l_todo_mmt_module_str
7668   \stex_close_module:
7669   \str_set_eq:NN \l_stex_current_module_str \l_todo_stex_module_str
7670   \stex_close_module:
7671   \stex_if_do_html:T { \end{stex_annotation_env} }
7672 }
7673 \stex_sms_allow_env:n{mmtinterface}

\mmtdef

7674 \NewDocumentCommand \mmtdef {m O{} m} {
7675   \stex_keys_set:nn{symdef}{#2}
7676   \str_set:Nx \l_stex_macroname_str { #1 }
7677   \str_if_empty:NT \l_stex_key_name_str {
7678     \str_set:Nx \l_stex_key_name_str { #1 }
7679   }
7680   \stex_symdecl_do:
7681
7682   \str_set_eq:NN \l_stex_current_module_str \l_todo_mmt_module_str
7683   \cs_set_eq:NN \l_todo_old_metagroup_cd \stex_metagroup_do_in:nn
7684   \cs_set_protected:Npn \stex_metagroup_do_in:nn ##1 ##2 {##2}
7685   \exp_args:Nnx \use:nn {\stex_module_add_symbol:nnnnnnN}{
7686     {\l_stex_macroname_str}
7687     {\l_stex_key_name_str}
7688     {\int_use:N \l_stex_get_symbol_arity_int}
7689     {\l_stex_get_symbol_args_tl}
7690     {}
7691     {}
7692     {}
7693     \stex_invoke_symbol:
7694   }
7695   \cs_set_eq:NN \stex_metagroup_do_in:nn \l_todo_old_metagroup_cd
7696   \str_set_eq:NN \l_stex_current_module_str \l_todo_stex_module_str
7697
7698   \str_set_eq:NN \l_stex_get_symbol_mod_str \l_todo_mmt_module_str
7699   \str_set_eq:NN \l_stex_get_symbol_name_str \l_stex_key_name_str
7700   \stex_notation_parse:n{#3}
7701   \stex_notation_check:
7702   \stex_notation_add:
7703   \stex_if_do_html:T{
7704     \stex_notation_do_html:n{\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str}
7705   }
7706   \stex_smsmode_do:
7707 }
7708 \stex_deactivate_macro:Nn \mmtdef {mmtinterface~environments}
7709 \stex_sms_allow_escape:N \mmtdef

```

(End of definition for \mmtdef. This function is documented on page ??.)

#### VoLL-KI Annotations

```
7710 \newcommand\precondition[2]{  
7711   \str_clear:N \l_stex_get_symbol_name_str  
7712   \stex_get_symbol:n{#2}  
7713   \str_if_empty:NTF \l_stex_get_symbol_name_str{  
    \errmessage{Unknown-symbol~#2}  
  }  
7716 \str_case:nnTF {#1}{  
  {remember}{}  
  {understand}{}  
  {analyze}{}  
  {evaluate}{}  
  {apply}{}  
  {create}{}  
}  
7723 \stex_annotation_invisible:nn{  
  shtml:preconditionsymbol={\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str},  
  shtml:preconditiondimension={#1}  
}  
7728 }{\errmessage{Unknown~cognitive~dimension~#1}}  
7729 }  
7730 }  
7731 \newcommand\objective[2]{  
7732   \str_clear:N \l_stex_get_symbol_name_str  
7733   \stex_get_symbol:n{#2}  
7734   \str_if_empty:NTF \l_stex_get_symbol_name_str{  
    \errmessage{Unknown-symbol~#2}  
  }  
7737 \str_case:nnTF {#1}{  
  {remember}{}  
  {understand}{}  
  {analyze}{}  
  {evaluate}{}  
  {apply}{}  
  {create}{}  
}  
7744 \stex_annotation_invisible:nn{  
  shtml:objectivesymbol={\l_stex_get_symbol_mod_str ? \l_stex_get_symbol_name_str},  
  shtml:objectivedimension={#1}  
}  
7749 }{\errmessage{Unknown~cognitive~dimension~#1}}  
7750 }  
7751 }  
7752 \seq_if_empty:NT \g_stex_current_file {  
7753   \seq_gset_eq:NN \g_stex_current_file \c_stex_main_file  
7754 }  
7755 \stex_persist_read_now:  
7756 \stex_every_file:  
7757 \cs_new_protected:Nn \__todo_newlabel:n {  
7758   \exp_args:Ne\__todo_old_newlabel:{\tl_to_str:n{#1}}  
7759 }  
7760 \AtBeginDocument{
```

```
7761 \iow_now:Nn \auxout {
7762   \ExplSyntaxOn
7763   \let\__todo_old_newlabel:\newlabel
7764   \let\newlabel\__todo_newlabel:n
7765   \ExplSyntaxOff
7766 }
7767 }
7768 
```

# Chapter 14

## Additional Packages

### 14.1 Implementation: The `notesslides` Package

#### 14.1.1 Class and Package Options

We define some Package Options and switches for the `notesslides` class and activate them by passing them on to `beamer.cls` and `omdoc.cls` and the `notesslides` package. We pass the `nontheorem` option to the `statements` package when we are not in notes mode, since the `beamer` package has its own (overlay-aware) theorem environments.

```
7769 <*cls>
7770 <@=notesslides>
7771 \ProvidesExplClass{notesslides}{2023/10/13}{3.4.0}{notesslides Class}
7772 \RequirePackage{13keys2e}
7773
7774 \str_const:Nn \c__notesslides_class_str {article}
7775
7776 \keys_define:nn{notesslides / cls}{
7777   class .str_set_x:N = \c__notesslides_class_str,
7778   notes .bool_set:N = \c__notesslides_notes_bool ,
7779   slides .code:n    = { \bool_set_false:N \c__notesslides_notes_bool },
7780   %docopt .str_set_x:N = \c__notesslides_docopt_str,
7781   unknown .code:n   = {
7782     \PassOptionsToClass{\CurrentOption}{beamer}
7783     \PassOptionsToClass{\CurrentOption}{\c__notesslides_class_str}
7784     \PassOptionsToPackage{\CurrentOption}{notesslides}
7785     \PassOptionsToPackage{\CurrentOption}{stex}
7786   }
7787 }
7788 \ProcessKeysOptions{ notesslides / cls }
7789
7790 \RequirePackage{stex}
7791 \stex_if_html_backend:T {
7792   \bool_set_true:N \c__notesslides_notes_bool
7793 }
7794
7795 \bool_if:NTF \c__notesslides_notes_bool {
7796   \PassOptionsToPackage{notes=true}{notesslides}
7797   \message{notesslides.cls:-Formatting-document-in-notes-mode}
```

```

7798 }{
7799   \PassOptionsToPackage{notes=false}{notesslides}
7800   \message{notesslides.cls:~Formatting~document~in~slides~mode}
7801 }
7802
7803 \bool_if:NTF \c_notesslides_notes_bool {
7804   \LoadClass{\c__notesslides_class_str}
7805 }{
7806   \LoadClass[10pt,notheorems,xcolor={dvipsnames,svgnames}]{beamer}
7807   \%newcounter{Item}
7808   \%newcounter{paragraph}
7809   \%newcounter{ subparagraph}
7810   \%newcounter{Hfootnote}
7811 }
7812 \RequirePackage{notesslides}
7813 
```

now we do the same for the `notesslides` package.

```

7814 <*package>
7815 \ProvidesExplPackage{notesslides}{2023/10/13}{3.4.0}{notesslides Package}
7816 \RequirePackage{l3keys2e}
7817
7818 \keys_define:nn{notesslides / pkg}{
7819   notes          .bool_set:N  = \c_notesslides_notes_bool ,
7820   slides         .code:n    = { \bool_set_false:N \c_notesslides_notes_bool },
7821   sectocframes  .bool_set:N  = \c_notesslides_sectocframes_bool ,
7822   topsect        .str_set_x:N = \c_notesslides_topsect_str,
7823   unknown        .code:n    = {
7824     \PassOptionsToPackage{\CurrentOption}{stex}
7825     \PassOptionsToPackage{\CurrentOption}{tikzinput}
7826   }
7827 }
7828 \ProcessKeysOptions{ notesslides / pkg }
7829
7830 \RequirePackage{stex}
7831 \stex_if_html_backend:T {
7832   \bool_set_true:N\c_notesslides_notes_bool
7833 }
7834
7835 \cs_set:Npn \sectiontitleemph #1 {
7836   \textbf{\Large #1}
7837 }
7838
7839 \newif\ifnotes
7840 \bool_if:NTF \c_notesslides_notes_bool {
7841   \notestrue
7842   \PassOptionsToPackage{usenames,dvipsnames,svgnames}{xcolor}
7843   \RequirePackage[noamsthm,hyperref]{beamerarticle}
7844   \RequirePackage{mdframed}
7845   \str_if_empty:NTF \c_notesslides_topsect_str{
7846     \%setsectionlevel{section}
7847   }{
7848     \exp_args:No \setsectionlevel \c_notesslides_topsect_str
7849   }

```

```

7850 }{
7851   \notesfalse
7852
7853   \cs_new_protected:Nn \__notesslides_do_sectocframes: {
7854     \cs_set_protected:Nn \__notesslides_do_label:n {
7855       \str_case:nnF{##1} {
7856         {part} {
7857           \tl_set:Nx\l__notesslides_num{\the part}
7858           \tl_set:cx{@ @ label}{%
7859             \cs_if_exist:NTF\parttitlename{\exp_not:N\parttitlename}{\exp_not:N\partname}{-}}
7860         }
7861         {chapter} {
7862           \tl_set:Nx\l__notesslides_num{\the chapter}
7863           \tl_set:cx{@ @ label}{%
7864             \cs_if_exist:NTF\chapertitlename{\exp_not:N\chapertitlename}{\exp_not:N\chaptername}{-}}
7865         }
7866         {section} {
7867           \tl_set:Nx\l__notesslides_num{\cs_if_exist:NT\thechapter{\the chapter.}\thesection.}
7868           \tl_set:cx{@ @ label}{\l__notesslides_num\quad}
7869         }
7870         {subsection} {
7871           \tl_set:Nx\l__notesslides_num{\cs_if_exist:NT\thechapter{\the chapter.}\thesubsection.}
7872           \tl_set:cx{@ @ label}{\l__notesslides_num\quad}
7873         }
7874         {subsubsection} {
7875           \tl_set:Nx\l__notesslides_num{\cs_if_exist:NT\thechapter{\the chapter.}\thesubsubsection.}
7876           \tl_set:cx{@ @ label}{\l__notesslides_num\quad}
7877         }
7878         {paragraph} {
7879           \tl_set:Nx\l__notesslides_num{\cs_if_exist:NT\thechapter{\the chapter.}\thesubparagraph.}
7880           \tl_set:cx{@ @ label}{\l__notesslides_num\quad}
7881         }
7882       }{
7883         \tl_set:Nx\l__notesslides_num{\cs_if_exist:NT\thechapter{\the chapter.}\thesubsubsubsection.}
7884         \tl_set:cx{@ @ label}{\l__notesslides_num\quad}
7885       }
7886     }
7887     \cs_set_protected:Nn \_sfragment_do_level:nn {
7888       \tl_if_exist:cT{c@##1}{\stepcounter{##1}}
7889       \addcontentsline{toc}{##1}{\protect\numberline{\use:c{the##1}}##2}
7890       \__notesslides_do_label:n{##1}
7891       \pdfbookmark[\int_use:N \l_stex_docheader_sect]{\l__notesslides_num##2}{##1.\l__note}
7892       \begin{frame}[noframenumbering]
7893         \vfill\centering
7894         \sectiontitleemph{%
7895           \use:c{@ @ label} ##2
7896         }
7897         \end{frame}
7898         \int_incr:N \l_stex_docheader_sect
7899         \tl_set:Nn \stex_current_section_level{##1}
7900     }
7901   }
7902
7903 \AtBeginDocument{

```

```

7904 \str_if_empty:NTF \c_notesslides_topsect_str {
7905   \setsectionlevel{section}
7906 } {
7907   \exp_args:No \setsectionlevel \c_notesslides_topsect_str
7908   \exp_args:No \str_if_eq:nnTF \c_notesslides_topsect_str {chapter} {
7909     \__notesslides_define_chapter:
7910   }{
7911     \exp_args:No \str_if_eq:nnT \c_notesslides_topsect_str {part} {
7912       \__notesslides_define_chapter:
7913       \__notesslides_define_part:
7914     }
7915   }
7916 }
7917 }
7918
7919 \bool_if:NT \c_notesslides_sectocframes_bool {
7920   \__notesslides_do_sectocframes:
7921 }
7922 }

7923 \cs_new_protected:Nn \__notesslides_define_chapter: {
7924   \cs_if_exist:NF \chaptername {
7925     \cs_set_protected:Npn \chaptername {Chapter}
7926   }
7927   \cs_if_exist:NF \chapter {
7928     \cs_set_protected:Npn \chapter {INVALID}
7929   }
7930   \cs_if_exist:NF \c@chapter {
7931     \newcounter{chapter}\counterwithin*{section}{chapter}
7932   }
7933 }
7934 }

7935 \cs_new_protected:Nn \__notesslides_define_part: {
7936   \cs_if_exist:NF \partname {
7937     \cs_set_protected:Npn \partname {Part}
7938   }
7939   \cs_if_exist:NF \part {
7940     \cs_set_protected:Npn \part {INVALID}
7941   }
7942   \cs_if_exist:NF \c@part {
7943     \newcounter{part}\counterwithin*{chapter}{part}
7944   }
7945 }
7946 }

```

**\prematurestop** We initialize \afterprematurestop, and provide \prematurestop@endsfragment which looks up \sfragment@level and recursively ends enough {sfragment}s.

```

7947 \def \c_notesslides_document_str{document}
7948 \newcommand\afterprematurestop{}
7949 \def\prematurestop@endsfragment{
7950   \unless\ifx\@currenvir\c_notesslides_document_str
7951     \expandafter\expandafter\expandafter\end\expandafter\expandafter\expandafter{\expandafter
7952       \expandafter\prematurestop@endsfragment
7953     \fi
7954 }

```

```

7955 \providecommand\prematurestop{
7956   \stex_if_html_backend:F{
7957     \message{Stopping-sTeX-processing-prematurely}
7958     \prematurestop@endsfragment
7959     \afterprematurestop
7960     \end{document}
7961   }
7962 }
```

(End of definition for `\prematurestop`. This function is documented on page 108.)

### 14.1.2 Notes and Slides

For the notes case, we also provide the `\usetheme` macro that would otherwise come from the the `beamer` class.

```

7963 \bool_if:NT \c_notesslides_notes_bool {
7964   \renewcommand\usetheme[2] [] {\usepackage[#1]{beamertheme#2}}
7965 }
7966 \NewDocumentCommand \libusetheme {O{} m} {
7967   \libusepackage[#1]{beamertheme#2}
7968 }
```

We define the sizes of slides in the notes. Somehow, we cannot get by with the same here.

```

7969 \newlength{\slidewidth}\setlength{\slidewidth}{13.5cm}
7970 \newlength{\slideheight}\setlength{\slideheight}{9cm}
```

We first set up the slide boxes in `notes` mode. We set up sizes and provide a box register for the frames and a counter for the slides.

```

7971 \ifnotes
7972
7973 \newlength{\slideframewidth}
7974 \setlength{\slideframewidth}{1.5pt}
```

`frame (env.)` We first define the keys.

```

7975 \cs_new_protected:Nn \__notesslides_do_yes_param:Nn {
7976   \exp_args:Nx \str_if_eq:nnTF { \str_uppercase:n{ #2 } }{ yes }{
7977     \bool_set_true:N #1
7978   }{
7979     \bool_set_false:N #1
7980   }
7981 }
7982
7983 \stex_keys_define:nnnn{notesslides / frame}{
7984   \str_clear:N \l__notesslides_frame_label_str
7985   \bool_set_true:N \l__notesslides_frame_allowframebreaks_bool
7986   \bool_set_true:N \l__notesslides_frame_allowdisplaybreaks_bool
7987   \bool_set_true:N \l__notesslides_frame_fragile_bool
7988   \bool_set_true:N \l__notesslides_frame_shrink_bool
7989   \bool_set_true:N \l__notesslides_frame_squeeze_bool
7990   \bool_set_true:N \l__notesslides_frame_t_bool
7991 }
7992   label .str_set_x:N = \l__notesslides_frame_label_str,
7993   allowframebreaks .code:n = {
```

```

7994     \__notesslides_do_yes_param:Nn \l__notesslides_frame_allowframebreaks_bool { #1 }
7995 },
7996 allowdisplaybreaks .code:n      = {
7997     \__notesslides_do_yes_param:Nn \l__notesslides_frame_allowdisplaybreaks_bool { #1 }
7998 },
7999 fragile          .code:n      = {
8000     \__notesslides_do_yes_param:Nn \l__notesslides_frame_fragile_bool { #1 }
8001 },
8002 shrink           .code:n      = {
8003     \__notesslides_do_yes_param:Nn \l__notesslides_frame_shrink_bool { #1 }
8004 },
8005 squeeze          .code:n      = {
8006     \__notesslides_do_yes_param:Nn \l__notesslides_frame_squeeze_bool { #1 }
8007 },
8008 t                .code:n      = {
8009     \__notesslides_do_yes_param:Nn \l__notesslides_frame_t_bool { #1 }
8010 },
8011 unknown         .code:n      = {}
8012 }{}
```

We redefine the `itemize` environment so that it looks more like the one in `beamer`.

```

8013 \cs_new_protected:Nn \__notesslides_setup_itemize: {
8014     \def\itemize@level{outer}
8015     \def\itemize@outer{outer}
8016     \def\itemize@inner{inner}
8017     \%newcommand\metakeys@show@keys[2]{\marginnote{{\scriptsize ##2}}}
8018     \renewenvironment{itemize} {
8019         \ifx\itemize@level\itemize@outer
8020             \def\itemize@label{$\rhd$}
8021         \fi
8022         \ifx\itemize@level\itemize@inner
8023             \def\itemize@label{$\scriptstyle\rhd$}
8024         \fi
8025         \begin{list}
8026             {\itemize@label}
8027             {\setlength{\labelsep}{.3em}
8028              \setlength{\labelwidth}{.5em}
8029              \setlength{\leftmargin}{1.5em}
8030            }
8031             \edef\itemize@level{\itemize@inner}
8032         }{
8033             \end{list}
8034         }
8035     }
```

We create the box with the `mdframed` environment from the equinymous package.

```

8036 \stex_if_html_backend:TF {
8037     \cs_new_protected:Nn \__notesslides_frame_box_begin: {
8038         \vbox\bgroup
8039         \begin{stex_annotation_env}[shtml:frame={}]
8040             \mdf@patchamsthm\notesslidesfont
8041         }
8042     \cs_new_protected:Nn \__notesslides_frame_box_end: {
8043         %^A \notesslides@slidelabel
```

```

8044     \medskip\par\noindent\tiny\notesslidesfooter
8045     \end{stex_annotation_env}\egroup
8046   }
8047 }{
8048   \cs_new_protected:Nn \__notesslides_frame_box_begin: {
8049     \begin{mdframed}[
8050       linewidth=\slideframewidth,
8051       skipabove=1ex,
8052       skipbelow=1ex,
8053       userdefinedwidth=\slidewidth,
8054       align=center
8055     ]\notesslidesfont
8056   }
8057   \cs_new_protected:Nn \__notesslides_frame_box_end: {
8058     \medskip\par\noindent\tiny\notesslidesfooter%^^A\notesslides@slidelabel
8059     \end{mdframed}
8060   }
8061 }

```

We define the environment, read them, and construct the slide number and label.

```

8062 \renewenvironment{frame}[1][]{
8063   \stex_keys_set:nn{notesslides / frame}{#1}
8064   \stepcounter{framenumber}
8065   \renewcommand{\newpage}{\addtocounter{framenumber}{1}}
8066   \def\@currentlabel{\theframenumber}
8067   \str_if_empty:NF \l__notesslides_frame_label_str {
8068     \label{\l__notesslides_frame_label_str}
8069   }
8070   \__notesslides_setup_itemize:
8071   \__notesslides_frame_box_begin:
8072 }{
8073   \__notesslides_frame_box_end:
8074 }

```

Now, we need to redefine the frametitle (we are still in course notes mode).

```

\frametitle
8075 \renewcommand{\frametitle}[1]{
8076   \stexdoctitle { #1 }
8077   \notesslidestitlenameph{#1}\medskip
8078 }

```

(End of definition for `\frametitle`. This function is documented on page ??.)

```

\pause
8079 \newcommand{\pause}{}

```

(End of definition for `\pause`. This function is documented on page ??.)

We redefine the `columns` and `column` environments:

```

8080 \renewenvironment{columns}[1][]{
8081   \par\noindent
8082   \begin{minipage}
8083     \slidewidth\centering\leavevmode
8084 %   \stex_if_html_backend:T{

```

```

8085 %     \cs_if_exist:NT \rustex_if:T {
8086 %         \rustex_if:T {\par
8087 %             \rustex_direct_HTML:n{<table><tr><td>}
8088 %         }
8089 %     }
8090 % }
8091 }{
8092 % \stex_if_html_backend:T{
8093 %     \cs_if_exist:NT \rustex_if:T {
8094 %         \rustex_if:T {\par
8095 %             \rustex_direct_HTML:n{</td></tr></table>}
8096 %         }
8097 %     }
8098 % }
8099 \end{minipage}\par\noindent
8100 }
8101 \newsavebox\columnbox
8102 \renewenvironment<>{column}[2] []{
8103     \begin{lrbox}{\columnbox}
8104 % \stex_if_html_backend:T{
8105 %     \cs_if_exist:NT \rustex_if:T {
8106 %         \rustex_if:T {\par
8107 %             \rustex_direct_HTML:n{</td><td>}
8108 %         }
8109 %     }
8110 % }
8111 \begin{minipage}{#2}
8112 }{
8113 \end{minipage}
8114 % \stex_if_html_backend:T{
8115 %     \cs_if_exist:NT \rustex_if:T {
8116 %         \rustex_if:T {\par
8117 %             \rustex_direct_HTML:n{</td><td>}
8118 %         }
8119 %     }
8120 % }
8121 \end{lrbox}\usebox\columnbox
8122 }

8123 \fi

```

#### 14.1.3 Environment and Macro Patches

The `note` environment is used to leave out text in the `slides` mode. It does not have a counterpart in OMDoc. So for course notes, we define the `note` environment to be a no-operation otherwise we declare the `note` environment to produce no output.

```

8124 \bool_if:NTF \c_notesslides_notes_bool {
8125     \renewenvironment{note}{\ignorespaces}{}
8126 }{
8127     \renewenvironment{note}{\setbox \l_tmpa_box\vbox\bgroup\egroup}
8128 }

```

For other environments we introduce variants prefixed with `n`, which are excluded in `slides` mode.

```

8129 \cs_new_protected:Nn \__notesslides_notes_env:n {
8130   \bool_if:NTF \c_notesslides_notes_bool {
8131     \newenvironment{#1}{#2}{#3}{#4}
8132   }{
8133     \newenvironment{#1}{#2{
8134       \cs_set:Npn \__notesslides_eat: #####1 \end #####2 {
8135         \str_if_eq:nnTF{#1}{#####2} {
8136           \end{#1}
8137         }{
8138           \__notesslides_eat:
8139         }
8140       }
8141       \__notesslides_eat:
8142       \%setbox\l_tmpa_box\vbox\bgroup#3
8143     }{
8144       %#4\egroup
8145     }
8146   }
8147 }

8148 \__notesslides_notes_env:n{nparagraph}{[1]}{\begin{sparagraph}{#1}}{\end{sparagraph}}
8149 \__notesslides_notes_env:n{nfragment}{[2]}{\begin{sfragment}{#1}{#2}}{\end{sfragment}}
8150 \__notesslides_notes_env:n{ndefinition}{[1]}{\begin{sdefinition}{#1}}{\end{sdefinition}}
8151 \__notesslides_notes_env:n{nassertion}{[1]}{\begin{sassertion}{#1}}{\end{sassertion}}
8152 \__notesslides_notes_env:n{nproof}{[2]}{\begin{sproof}{#1}{#2}}{\end{sproof}}
8153 \__notesslides_notes_env:n{nexample}{[1]}{\begin{sexample}{#1}}{\end{sexample}}
8154 \__notesslides_notes_env:n{nparagrapheq}{[1]}{\begin{sparagrapheq}{#1}}{\end{sparagrapheq}}
8155 \RequirePackage{graphicx}

8156 \NewDocumentCommand\frameimage{s O{} m}{

8157   \IfBooleanTF #1 {
8158     \begin{frame}[plain]
8159   }{
8160     \begin{frame}
8161   }
8162   \bool_if:NTF \c_notesslides_notes_bool {
8163     \slidewidth=\dimexpr\slidewidth-(2\slideframewidth)\relax
8164   }{
8165     \slidewidth=\textwidth\relax
8166   }
8167   \def\Gin@ewidth{}\setkeys{Gin}{#2}
8168   \tl_if_empty:NTF \Gin@ewidth {
8169     \mhgraphics[width=\slidewidth,#2]{#3}
8170   }{
8171     \mhgraphics[#2]{#3}
8172   }
8173   \end{frame}
8174 }

8175 }

8176 }

8177 \cs_set_eq:NN \__notesslides_inputref:\inputref
8178 \cs_set_protected:Npn \inputref{@ifstar\ninputref\__notesslides_inputref:}

```

hacking inputref:

\inputref\*

```

8177 \cs_set_eq:NN \__notesslides_inputref:\inputref
8178 \cs_set_protected:Npn \inputref{@ifstar\ninputref\__notesslides_inputref:}

```

```

8179 \bool_if:NTF \c_notesslides_notes_bool {
8180   \newcommand\nininputref[2][]{%
8181     \__notesslides_inputref:[#1]{#2}%
8182   }%
8183 }{%
8184   \newcommand\nininputref[2]{}%
8185 }

```

(End of definition for `\inputref*`. This function is documented on page 107.)

#### 14.1.4 Styling Across Notes/Slides

```

8186 \def\notesslidestitleemph#1{%
8187   {\Large\bf\sf#1}%
8188   \vskip0.1\baselineskip
8189   \leaders\vrule width \textwidth
8190   \vskip0.4pt%
8191   \nointerlineskip
8192 }%
8193
8194 \def\notesslidesfooter{}%
8195
8196 \let\notesslidesfont\sffamily

```

#### 14.1.5 Beamer Compatibility

All of this should be removed and made part of a template

```

8197
8198 \bool_if:NT \c_notesslides_notes_bool {%
8199   \def\author{\@dblarg\ns@author}%
8200   \long\def\ns@author[#1]#2{%
8201     \tl_if_empty:nTF{#1}{%
8202       \def\beamer@shortauthor{#2}%
8203     }{%
8204       \def\beamer@shortauthor{#1}%
8205     }%
8206     \def\@author{#2}%
8207   }%
8208   \def\title{\@dblarg\ns@title}%
8209   \long\def\ns@title[#1]#2{%
8210     \tl_if_empty:nTF{#1}{%
8211       \def\beamer@shorttitle{#2}%
8212     }{%
8213       \def\beamer@shorttitle{#1}%
8214     }%
8215     \def\@title{#2}%
8216     \stexdoctitle{#2}%
8217   }%
8218   \def\insertshortauthor{%
8219     \hbox\bgroup\def\\{}\cs_if_exist:NT\beamer@shortauthor\beamer@shortauthor\egroup
8220   }%
8221   \def\insertshorttitle{%
8222     \hbox\bgroup\def\\{}\cs_if_exist:NT\beamer@shorttitle\beamer@shorttitle\egroup
8223   }%
8224   \stex_if_html_backend:TF{%

```

```

8225     \def\insertframenumber{\stex_annotate:nn{shml:framenumber={}}{}}
8226   }{
8227     \def\insertframenumber{\@arabic\c@framenumber}
8228   }
8229   \def\insertshortdate{\today}
8230 }

```

### 14.1.6 TODO Excursions

**\excursion**

The excursion macros are very simple, we define a new internal macro `\excursionref` and use it in `\excursion`, which is just an `\inputref` that checks if the new macro is defined before formatting the file in the argument.

```

8231 \gdef\printexcursions{}
8232 \newcommand\excursionref[2]{% label, text
8233   \bool_if:NT \c_notesslides_notes_bool {
8234     \begin{sparagraph}[title=Excursion]
8235       #2 \sref[fallback=the appendix]{#1}.
8236     \end{sparagraph}
8237   }
8238 }
8239 \newcommand\activate@excursion[2][]{%
8240   \tl_gput_right:Nn\printexcursions{\inputref[#1]{#2}}
8241 }
8242 \newcommand\excursion[4][]{% repos, label, path, text
8243   \bool_if:NT \c_notesslides_notes_bool {
8244     \activate@excursion[#1]{#3}
8245     \excursionref{#2}{#4}
8246   }
8247 }

```

(End of definition for `\excursion`. This function is documented on page 109.)

**\excursiongroup**

```

8248 \keys_define:nn{notesslides / excursiongroup }{
8249   id      .str_set_x:N = \l__notesslides_excursion_id_str,
8250   intro    .tl_set:N    = \l__notesslides_excursion_intro_tl,
8251   archive  .str_set_x:N = \l__notesslides_excursion_mrepos_str
8252 }
8253 \cs_new_protected:Nn \__notesslides_excursion_args:n {
8254   \tl_clear:N \l__notesslides_excursion_intro_tl
8255   \str_clear:N \l__notesslides_excursion_id_str
8256   \str_clear:N \l__notesslides_excursion_mrepos_str
8257   \keys_set:nn {notesslides / excursiongroup }{ #1 }
8258 }
8259 \newcommand\excursiongroup[1][]{%
8260   \__notesslides_excursion_args:n{ #1 }
8261   \tl_if_empty:NF\printexcursions
8262   {\IfInputref{}{\begin{note}
8263     \begin{sfragment}{Excursions}% TODO pass on id
8264     \ifdefempty\l__notesslides_excursion_intro_tl{}{
8265       \exp_args:NNe \use:nn \inputref{[\l__notesslides_excursion_mrepos_str]}{
8266         \l__notesslides_excursion_intro_tl
8267       }
8268     }
8269   }

```

```

8269     \printexcursions%
8270     \end{sfragment}
8271     \end{note}}}
8272 }
8273 \ifcsname beameritemnestingprefix\endcsname\else\def\beameritemnestingprefix{}\fi
(End of definition for \excursiongroup. This function is documented on page 109.)
8274 \prop_new:N \g__notesslides_variables_prop
8275 \cs_set_protected:Npn \setSGvar #1 #2 {
8276     \prop_gput:Nnn \g__notesslides_variables_prop {#1}{#2}
8277 }
8278 \cs_set_protected:Npn \useSGvar #1 {
8279     \prop_item:Nn \g__notesslides_variables_prop {#1}
8280 }
8281 \cs_set_protected:Npn \ifSGvar #1 #2 #3 {
8282     \prop_get:NnNF \g__notesslides_variables_prop {#1} \l__notesslides_tmp {
8283         \PackageError{document-structure}
8284         {The sTeX Global variable #1 is undefined}
8285         {set it with \protect\setSGvar}\TODO better error
8286     }
8287     \tl_if_eq:NnT \l__notesslides_tmp {#2}{ #3 }
8288 }
8289
8290
8291 </package>

```

## 14.2 Implementation: The problem Package

### 14.2.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. They all come with their own conditionals that are set by the options.

```

8292 <*package>
8293 <@=problems>
8294 \ProvidesExplPackage{problem}{2023/10/13}{3.4.0}{Semantic Markup for Problems}
8295 \RequirePackage{l3keys2e}
8296
8297 \keys_define:nn { problem / pkg }{
8298     notes      .default:n   = { true },
8299     notes      .bool_set:N  = \c__problems_notes_bool,
8300     gnotes     .default:n   = { true },
8301     gnotes     .bool_set:N  = \c__problems_gnotes_bool,
8302     hints      .default:n   = { true },
8303     hints      .bool_set:N  = \c__problems_hints_bool,
8304     solutions   .default:n   = { true },
8305     solutions   .bool_set:N  = \c__problems_solutions_bool,
8306     pts        .default:n   = { true },
8307     pts        .bool_set:N  = \c__problems_pts_bool,
8308     min        .default:n   = { true },
8309     min        .bool_set:N  = \c__problems_min_bool,
8310     %boxed    .default:n   = { true },
8311     %boxed    .bool_set:N  = \c__problems_boxed_bool,

```

```

8312 test .default:n = { true },
8313 test .bool_set:N = \c__problems_test_bool,
8314 unknown .code:n = {
8315   \PassOptionsToPackage{\CurrentOption}{stex}
8316 }
8317 }
8318 \newif\ifsolutions
8319
8320 \ProcessKeysOptions{ problem / pkg }
8321 \bool_if:NTF \c__problems_solutions_bool {
8322   \solutionstrue
8323 }{
8324   \solutionsfalse
8325 }
8326 \newif\ifintest
8327 \bool_if:NTF \c__problems_test_bool {
8328   \intesttrue
8329 }{
8330   \intestfalse
8331 }
8332
8333 \RequirePackage{stex}

```

\problem@kw@\* For multilinguality, we define internal macros for keywords that can be specialized in \*.ldf files.

```

8334 \AddToHook{begindocument}{
8335   \ExplSyntaxOn\makeatletter
8336   \input{problem-english.ldf}
8337   \ltx@ifpackageloaded{babel}{
8338     \clist_set:Nx \l_tmpa_clist {\exp_args:No \tl_to_str:n \bbl@loaded}
8339       \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{ngerman}}{
8340         \input{problem-ngerman.ldf}
8341       }
8342       \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{finnish}}{
8343         \input{problem-finnish.ldf}
8344       }
8345       \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{french}}{
8346         \input{problem-french.ldf}
8347       }
8348       \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{russian}}{
8349         \input{problem-russian.ldf}
8350       }
8351   }{}}
8352   \makeatother\ExplSyntaxOff
8353 }

```

(End of definition for \problem@kw@\*. This function is documented on page ??.)

### 14.2.2 Problems and Solutions

We now prepare the KeyVal support for problems. The key macros just set appropriate internal macros.

```
8354 \bool_new:N \l_stex_key_autogradable_bool
```

```

8355 \stex_keys_define:nnnn{ problem }{
8356   \tl_set:Nn \l_stex_key_pts_tl 0
8357   \tl_set:Nn \l_stex_key_min_tl 0
8358   \str_clear:N \l_stex_key_name_str
8359   \str_clear:N \l_stex_key_mhrepos_str
8360   \bool_set_false:N \l_stex_key_autogradable_bool
8361 }{
8362   pts .tl_set:N = \l_stex_key_pts_tl,
8363   min .tl_set:N = \l_stex_key_min_tl,
8364   name .str_set:N = \l_stex_key_name_str,
8365   autogradable .bool_set:N = \l_stex_key_autogradable_bool,
8366   archive .code:n = {},
8367   %archive .str_set:N = \l_stex_key_mhrepos_str,
8368   creators .code:n = {}
8369   %imports .tl_set:N = \l__problems_prob_imports_tl,
8370   %refnum .int_set:N = \l__problems_prob_refnum_int,
8371 }{id,title,style,uses}

```

Then we set up a counter for problems.

```
\numberproblemsin
8372 \newcounter{sproblem}[section]
8373 \newcommand\numberproblemsin[1]{
8374   \@addtoreset{sproblem}{#1}
8375   \def\thesproblem{\arabic{#1}.\arabic{sproblem}}
8376 }
8377 \numberproblemsin[section]
8378 %\def\theplainproblem{\arabic{sproblem}}
8379 %\def\thesproblem{\thesection.\theplainproblem}
```

(End of definition for `\numberproblemsin`. This function is documented on page ??.)

```
sproblem (env.)
8380 \cs_new:Nn \__problems_activate_macros: {
8381   \stex_reactivate_macro:N \solution
8382   \stex_reactivate_macro:N \mcb
8383   \stex_reactivate_macro:N \scb
8384   \stex_reactivate_macro:N \fillinsol
8385   \stex_reactivate_macro:N \hint
8386   \stex_reactivate_macro:N \exnote
8387   \stex_reactivate_macro:N \gnote
8388 }
8389 \newcounter{pts}
8390 \newcounter{min}
8391 \bool_new:N \l__problems_in_problem_bool
8392 \bool_new:N \l__problems_has_pts_bool
8393 \bool_new:N \l__problems_has_min_bool
8394 \bool_set_false:N \l__problems_in_problem_bool
8395 \stex_new_stylable_env:nnnnnn {problem} {0{}}
8396 \bool_if:NF \l__problems_in_problem_bool {
8397   \bool_if:NT \l__problems_in_problem_bool {
8398     \msg_error:nn{stex}{error/nestedproblem}
8399   }
8400 \cs_if_exist:NTF \l_problem_inputproblem_keys_tl {
8401   \tl_put_left:Nn \l_problem_inputproblem_keys_tl {#1,}
```

```

8402   \exp_args:Nno \stex_keys_set:nn{problem}{
8403     \l_problem_inputproblem_keys_tl
8404   }
8405 }{
8406   \stex_keys_set:nn{problem}{#1}
8407 }
8408 \refstepcounter{sproblem}
8409
8410 \stex_if_do_html:T {
8411   \str_if_empty:NT \l_stex_key_name_str {
8412     \stex_file_split_off_lang:NN \l__problems_path_seq \g_stex_current_file
8413     \seq_get_right:NN \l__problems_path_seq \l_stex_key_name_str
8414   }
8415   \exp_args:Nne \begin{stex_annotation_env} {
8416     shtml:problem={\l_stex_key_name_str},
8417     shtml:language={ \l_stex_current_language_str},
8418     shtml:autogradable={\bool_if:NTF \l_stex_key_autogradable_bool {true}{false}}
8419   }
8420   \tl_if_empty:NF \l_stex_key_title_tl {
8421     \exp_args:No \stexdoctitle \l_stex_key_title_tl
8422   }
8423   \stex_annotation_invisible:nn{
8424     shtml:problempoints={\l_stex_key_pts_tl}
8425   }{ \l_stex_key_pts_tl }
8426 }
8427 \tl_set_eq:NN \thistitle \l_stex_key_title_tl
8428
8429
8430 \bool_set_true:N \l__problems_in_problem_bool
8431 \tl_set_eq:NN \l__problems_pts_tl \l_stex_key_pts_tl
8432 \tl_set_eq:NN \l__problems_min_tl \l_stex_key_min_tl
8433 \tl_if_eq:NnTF \l__problems_pts_tl {0}
8434   {\bool_set_false:N \l__problems_has_pts_bool}
8435   {\bool_set_true:N \l__problems_has_pts_bool}
8436 \tl_if_eq:NnTF \l__problems_min_tl {0}
8437   {\bool_set_false:N \l__problems_has_min_bool}
8438   {\bool_set_true:N \l__problems_has_min_bool}
8439 \int_gzero:N \g__problems_subproblem_int
8440
8441 \stex_style_apply:
8442 \stex_do_id:
8443 \__problems_activate_macros:
8444 }{
8445   \addtocounter{pts}{\l__problems_pts_tl}
8446   \addtocounter{min}{\l__problems_min_tl}
8447   \__problems_record_problem:
8448   \stex_style_apply:
8449   \stex_if_do_html:T{ \end{stex_annotation_env} }
8450 }{
8451   \par\noindent\problemheader
8452   \bool_if:NT \c__problems_pts_bool {
8453     \tl_if_eq:NnF \l__problems_pts_tl {0}{
8454       \marginpar{\l__problems_pts_tl{}-\problem@kw@pts\smallskip}
8455     }

```

```

8456   }
8457   \bool_if:NT \c__problems_min_bool {
8458     \tl_if_eq:NnF \l__problems_min_tl {0} {
8459       \marginpar{\l__problems_min_tl{}-\problem@kw@minutes\smallskip}
8460     }
8461   }
8462   \par
8463   \stex_ignore_spaces_and_pars:
8464 }{
8465   \par\bigskip
8466 % \bool_if:NT \c__problems_test_bool \pagebreak
8467 }{s}

8468 \tl_set:Nn \problemheader {
8469   \stex_if_do_html:TF{
8470     \tl_if_empty:NF \thistitle {
8471       \stex_annotate:nn{shtml:problemtitle={}}{\textbf{\thistitle}}
8472     }
8473   }{
8474     \textbf{\sproblemautorefname{}\thesproblem}
8475     \tl_if_empty:NF \thistitle {
8476       {~}(\thistitle)
8477     }
8478   }
8479 }
8480 }
8481 }
8482
8483 \cs_new_protected:Nn \__problems_record_problem: {
8484   \exp_args:NNN \iow_now:Nn \auxout {
8485     \problem@restore {\thesproblem}{\l__problems_pts_tl}{\l__problems_min_tl}
8486   }
8487 }
8488
8489 \cs_new_protected:Npn \problem@restore #1 #2 #3 {}

subproblem (env.)
8490 \int_new:N \g__problems_subproblem_int
8491
8492 \stex_new_stylable_env:nnnnnnn {subproblem} {0}{{
8493   \stex_keys_set:nn{problem}{#1}
8494   \bool_if:NF \l__problems_in_problem_bool{
8495     \ifstexhtml\else
8496       \par{\bfseries WARNING~subproblem~to~be~used~in~some~problem}\par
8497     \fi
8498     \__problems_activate_macros:
8499     \bool_set_true:N \l__problems_in_problem_bool
8500     \tl_set:Nn \l__problems_pts_tl{0}
8501     \tl_set:Nn \l__problems_min_tl{0}
8502   }
8503   \str_if_empty:NT \l_stex_key_name_str {
8504     \stex_file_split_off_lang:NN \l__problems_path_seq \g_stex_current_file
8505     \seq_get_right:NN \l__problems_path_seq \l_stex_key_name_str
8506   }
8507   \stex_if_do_html:T {

```

```

8508   \str_if_empty:NT \l_stex_key_name_str {
8509     \stex_file_split_off_lang:NN \l__problems_path_seq \g_stex_current_file
8510     \seq_get_right:NN \l__problems_path_seq \l_stex_key_name_str
8511   }
8512   \exp_args:Nne \begin{stex_annotation_env} {
8513     shtml:subproblem={\l_stex_key_name_str},
8514     shtml:language={\l_stex_current_language_str},
8515     shtml:autogradable={\bool_if:NTF \l_stex_key_autogradable_bool {true}{false}}
8516   }
8517   \stex_annotation_invisible:n}
8518   \tl_if_empty:NF \l_stex_key_title_tl {
8519     \exp_args:No \stexdoctitle \l_stex_key_title_tl
8520   }
8521 }
8522 \int_gincr:N \g__problems_subproblem_int
8523 \bool_if:NF \l__problems_has_pts_bool {
8524   \tl_gset:Nx \l__problems_pts_tl {\int_eval:n {\l__problems_pts_tl + \l_stex_key_pts_tl}}
8525 }
8526 \bool_if:NF \l__problems_has_min_bool {
8527   \tl_gset:Nx \l__problems_min_tl {\int_eval:n {\l__problems_min_tl + \l_stex_key_min_tl}}
8528 }
8529 \stex_if_smsmode:F \stex_style_apply:
8530 }{
8531   \stex_if_smsmode:F \stex_style_apply:
8532   \stex_if_do_html:T{ \end{stex_annotation_env} }
8533 }{
8534   \begin{list}{}{
8535     \setlength\topsep{0pt}
8536     \setlength\parsep{0pt}
8537     \setlength\rightmargin{0pt}
8538   }\item[\int_use:N \g__problems_subproblem_int .]
8539   \bool_if:NT \c__problems_pts_bool {
8540     \bool_if:NF \l__problems_has_pts_bool {
8541       \marginpar{\smallskip\l_stex_key_pts_tl{}~\problem@kw@pts}
8542     }
8543   }
8544   \bool_if:NT \c__problems_min_bool {
8545     \bool_if:NF \l__problems_has_min_bool{
8546       \marginpar{\smallskip\l_stex_key_min_tl{}~\problem@kw@minutes}
8547     }
8548   }
8549 }{
8550   \end{list}
8551 }{}}

\includeproblem
8552 \stex_keys_define:nnnn{ includeproblem }{
8553   \str_clear:N \l_stex_key_mhrepos_str
8554 }{
8555   archive .str_set:N      = \l_stex_key_mhrepos_str,
8556   unknown .code:n = {}
8557 }{}}
8558
8559 \NewDocumentCommand\includeproblem{O{} m}{}

```

```

8560   \group_begin:
8561   \tl_set:Nn \l_problem_inputproblem_keys_tl {#1}
8562   \stex_keys_set:nn{includeproblem}{#1}
8563   \exp_args:Nno \use:nn{\inputref[]}\l_stex_key_mhrepos_str]{#2}
8564   \group_end:
8565 }
8566

```

(End of definition for `\includeproblem`. This function is documented on page 114.)

`solution (env.)`

```

8567   \int_new:N \g_problem_id_counter
8568   \dim_new:N \l_stex_key_testsphere_dim
8569   \stex_keys_define:nnnn{ solution }{
8570     \str_clear:N \l_stex_key_answerclass_str
8571     \dim_zero:N \l_stex_key_testsphere_dim
8572   }{
8573     testspace .dim_set:N = \l_stex_key_testsphere_dim,
8574     answerclass .str_set:N = \l_stex_key_answerclass_str
8575   }{id,title,style}
8576
8577   \cs_new_protected:Nn \__problems_solution_start:n {
8578     \stex_keys_set:nn{ solution }{#1}
8579     \str_if_empty:NT \l_stex_key_id_str {
8580       \int_gincr:N \g_problem_id_counter
8581       \str_set:Nx \l_stex_key_id_str {
8582         SOLUTION_\int_use:N \g_problem_id_counter
8583       }
8584     }
8585     \stex_if_do_html:T{
8586       \begin{stex_annotation_env}{
8587         shtml:solution=\l_stex_key_id_str,
8588         shtml:answerclass={\l_stex_key_answerclass_str}
8589       }
8590     }
8591     \stex_style_apply:
8592   }
8593
8594   \stex_new_stylable_env:nnnnnnn { solution }{ 0{} }{
8595     \stex_if_do_html:TF{
8596       \__problems_solution_start:n{#1}
8597     }{
8598       \ifsolutions
8599         \__problems_solution_start:n{#1}
8600       \else
8601         \stex_keys_set:nn{ solution }{#1}
8602         \testspace{\l_stex_key_testsphere_dim}
8603         \setbox\l_tmpa_box\vbox\bgroup
8604       \fi
8605     }
8606   }{
8607     \stex_if_do_html:TF{
8608       \stex_style_apply:
8609     \end{stex_annotation_env}

```

```

8610    }{
8611      \ifsolutions
8612        \stex_style_apply:
8613        \stex_if_do_html:T{
8614          \end{stex_annotation_env}
8615        }
8616      \else
8617        \egroup
8618      \fi
8619    }
8620  }{
8621    \par\smallskip\rule[.3em]{\linewidth}{0.4pt}\newline\smallskip
8622    \noindent\emph{\problem@kw@solution\tl_if_empty:N \l_stex_key_title_tl{
8623      \l_stex_key_title_tl \str_if_empty:N \l_stex_key_answerclass_str {
8624        (Answer Class: \l_stex_key_answerclass_str)
8625      }
8626    } :~}
8627  }{
8628    \par\rule[.3em]{\linewidth}{0.4pt}\newline
8629  }{}
8630
8631 \stex_deactivate_macro:Nn \solution {sproblem~environments}

\start solutions
\stop solutions
8632 \cs_new_protected:Npn \start solutions{
8633   \global\solutionstrue
8634 }
8635 \cs_new_protected:Npn \stop solutions{
8636   \global\solutionsfalse
8637 }

(End of definition for \start solutions and \stop solutions. These functions are documented on page
111.)
```

**hint (env.)**

```

8638
8639 \stex_keys_define:nnnn{ problemenv }{}{}{id,title,style}
8640
8641 \cs_new_protected:Nn \__problems_hint_start:n {
8642   \stex_keys_set:nn{ problemenv }{#1}
8643   \str_if_empty:NT \l_stex_key_id_str {
8644     \int_gincr:N \g_problem_id_counter
8645     \str_set:Nx \l_stex_key_id_str {
8646       HINT_\int_use:N \g_problem_id_counter
8647     }
8648   }
8649   \stex_if_do_html:T{
8650     \begin{stex_annotation_env}{
8651       shtml:problemhint=\l_stex_key_id_str
8652     }
8653   }
8654   \stex_style_apply:
8655 }
```

```

8657 \stex_new_stylable_env:nnnnnnn { hint }{ 0{} }{
8658   \stex_if_do_html:TF{
8659     \__problems_hint_start:n{#1}
8660   }{
8661     \bool_if:NTF \c__problems_hints_bool {
8662       \__problems_hint_start:n{#1}
8663     }{
8664       \setbox\l_tmpa_box\vbox\bgroup
8665     }
8666   }
8667 }{
8668   \stex_if_do_html:TF{
8669     \stex_style_apply:
8670     \end{stex_annotation_env}
8671   }{
8672     \bool_if:NTF \c__problems_hints_bool {
8673       \stex_style_apply:
8674       \stex_if_do_html:T{
8675         \end{stex_annotation_env}
8676       }
8677     }{
8678       \egroup
8679     }
8680   }
8681 }{
8682   \par\smallskip\rule[.3em]{\linewidth}{0.4pt}\newline\smallskip
8683   \noindent\emph{\problem@kw@hint\tl_if_empty:N \l_stex_key_title_tl{
8684     {~}\l_stex_key_title_tl
8685   } :~}
8686 }{
8687   \par\rule[.3em]{\linewidth}{0.4pt}\newline
8688 }{}
8689 \stex_deactivate_macro:Nn \hint {sproblem~environments}

exnote (env.)
8690 \cs_new_protected:Nn \__problems_exnote_start:n {
8691   \stex_keys_set:nn{ problemenv }{#1}
8692   \str_if_empty:NT \l_stex_key_id_str {
8693     \int_gincr:N \g_problem_id_counter
8694     \str_set:Nx \l_stex_key_id_str {
8695       EXNOTE_\int_use:N \g_problem_id_counter
8696     }
8697   }
8698   \stex_if_do_html:T{
8699     \begin{stex_annotation_env}[
8700       shtml:problemnote=\l_stex_key_id_str
8701     ]
8702   }
8703   \stex_style_apply:
8704 }
8705
8706 \stex_new_stylable_env:nnnnnnn { exnote }{ 0{} }{
8707   \stex_if_do_html:TF{
8708     \__problems_exnote_start:n{#1}

```

```

8709  }{
8710    \bool_if:NTF \c__problems_notes_bool {
8711      \__problems_exnote_start:n{#1}
8712    }{
8713      \setbox\l_tmpa_box\vbox\bgroup
8714    }
8715  }
8716 }{
8717   \stex_if_do_html:TF{
8718     \stex_style_apply:
8719     \end{stex_annotation_env}
8720   }{
8721     \bool_if:NTF \c__problems_notes_bool {
8722       \stex_style_apply:
8723       \stex_if_do_html:T{
8724         \end{stex_annotation_env}
8725       }
8726     }{
8727       \egroup
8728     }
8729   }
8730 }{
8731   \par\smallskip\rule[.3em]{\linewidth}{0.4pt}\newline\smallskip
8732   \noindent\emph{\problem@kw@note\tl_if_empty:N \l_stex_key_title_tl{
8733     \l_stex_key_title_tl
8734   }} :~}
8735 }{
8736   \par\rule[.3em]{\linewidth}{0.4pt}\newline
8737 }{}}
8738 \stex_deactivate_macro:Nn \exnote {sproblem~environments}

gnote (env.)
8739 \int_new:N \l__problemsансcls_int
8740
8741 \cs_new_protected:Nn \__problems_gnote_start:n {
8742   \stex_keys_set:nn{ \problemenv }{#1}
8743   \str_if_empty:NT \l_stex_key_id_str {
8744     \int_gincr:N \g_problem_id_counter
8745     \str_set:Nx \l_stex_key_id_str {
8746       GNOTE_\int_use:N \g_problem_id_counter
8747     }
8748   }
8749
8750   \stex_if_do_html:T{
8751     \begin{stex_annotation_env} {
8752       \shtml:problemgnote=\l_stex_key_id_str
8753     }
8754   }
8755   \stex_style_apply:
8756 }
8757
8758 \stex_new_stylable_env:nnnnnnn { gnote }{ 0{} }{
8759   \stex_if_do_html:TF{
8760     \__problems_gnote_start:n{#1}

```

```

8761   }{
8762     \bool_if:NTF \c__problems_gnotes_bool {
8763       \__problems_gnote_start:n{#1}
8764     }{
8765       \setbox\l_tmpa_box\vbox\bgroup
8766     }
8767   }
8768   \stex_reactivate_macro:N \anscls
8769 }{
8770   \stex_if_do_html:TF{
8771     \stex_style_apply:
8772     \end{stex_annotation_env}
8773 }{
8774   \bool_if:NTF \c__problems_gnotes_bool {
8775     \stex_style_apply:
8776     \stex_if_do_html:T{
8777       \end{stex_annotation_env}
8778     }
8779   }{
8780     \egroup
8781   }
8782 }
8783 }{
8784   \par\smallskip\rule[.3em]{\linewidth}{0.4pt}\newline\smallskip
8785   \noindent\emph{\problem@kw@grading\str_if_empty:N \l_stex_key_title_tl{
8786     {~}\l_stex_key_title_tl
8787   } :~}
8788 }{
8789   \par\rule[.3em]{\linewidth}{0.4pt}\newline
8790 }{}}
8791 \stex_deactivate_macro:Nn \gnote {sproblem~environments}
8792
8793
8794 \stex_keys_define:nnnn{ anscls }{
8795   \str_clear:N \l_stex_key_pts_str
8796   \tl_clear:N \l_stex_key_feedback_tl
8797 }{
8798   pts .str_set:N = \l_stex_key_pts_str,
8799   feedback .tl_set:N = \l_stex_key_feedback_str,
8800   update .code:n = {}
8801 }{id}
8802 \newcommand \anscls [2] [] {
8803   \stex_keys_set:nn{ anscls }{#1}
8804   \str_if_empty:NT \l_stex_key_id_str {
8805     \int_incr:N \l__problems_anscls_int
8806     \str_set:Nx \l_stex_key_id_str {
8807       AC\int_use:N \l__problems_anscls_int
8808     }
8809   }
8810   \begin{list}{}{
8811     \setlength\topsep{0pt}
8812     \setlength\parsep{0pt}
8813     \setlength\rightmargin{0pt}
8814   }\item[\l_stex_key_id_str]

```

```

8815 \stex_if_do_html:TF{
8816   \stex_annotate:nn{
8817     shtml:answerclass={\l_stex_key_id_str}
8818     \str_if_empty:NF \l_stex_key_pts_str{
8819       ,shtml:answerclass-pts={\l_stex_key_pts_str}
8820     }
8821     \str_if_empty:NF \l_stex_key_feedback_str{
8822       ,shtml:answerclass-feedback={\l_stex_key_feedback_str}
8823     }
8824   }{
8825     #2
8826   }
8827   \str_if_empty:NF \l_stex_key_pts_str {\par
8828     ~ \problem@kw@points:~\l_stex_key_pts_str
8829   }
8830   \str_if_empty:NF \l_stex_key_feedback_str {\par
8831     ~ \problem@kw@feedback :~\l_stex_key_feedback_str
8832   }
8833 \end{list}
8834 }
8835 \stex_deactivate_macro:Nn \anscls {gnote-environments}

```

The margin pars are reader-visible, so we need to translate

```

8837 \def\pts#1{
8838   \bool_if:NT \c__problems_pts_bool {
8839     \stex_annotate:nn{shtml:problempoints={#1}}{\marginpar{#1~\problem@kw@pts}}
8840   }\hbox_unpack:N\c_empty_box
8841 }
8842 \def\min#1{
8843   \bool_if:NT \c__problems_min_bool {
8844     \stex_annotate:nn{shtml:problemminutes={}}{\marginpar{#1~\problem@kw@minutes}}
8845   }\hbox_unpack:N\c_empty_box
8846 }

```

*mcb (env.)*

```

8847 \stex_new_stylable_env:nnnnnnn{mcb}{0{}){
8848   \stex_keys_set:nn{style}{#1}
8849   \stex_if_do_html:T{
8850     \tl_set:Nn\problem_mcc_box_tl{}
8851     \exp_args:Nne \begin{stex_annotate_env}{
8852       shtml:multiple-choice-block={}
8853       \clist_if_empty:NF \l_stex_key_style_clist ,,
8854         shtml:styles={\l_stex_key_style_clist}
8855       }
8856     }
8857   }
8858 \stex_deactivate_macro:Nn \mcb {sproblem~environments}
8859 \stex_deactivate_macro:Nn \scb {sproblem~environments}
8860 \stex_deactivate_macro:Nn \solution {sproblem~environments}
8861 \stex_deactivate_macro:Nn \hint {sproblem~environments}
8862 \stex_deactivate_macro:Nn \exnote {sproblem~environments}
8863 \stex_deactivate_macro:Nn \gnote {sproblem~environments}
8864 \stex_reactivate_macro:N \mcc

```

```

8865 \cs_set:Nn \__problems_mccline:n{
8866   \begin{list}{}{%
8867     \setlength\topsep{0pt}
8868     \setlength\parsep{0pt}
8869     \setlength\rightmargin{0pt}
8870   }\item[\problem_mcc_box_tl] ##1 \end{list}
8871 }
8872 \stex_style_apply:
8873 }{
8874   \stex_style_apply:
8875   \stex_if_do_html:T{
8876     \end{stex_annotation_env}
8877   }
8878 }{\par}{}
8879 \stexstylemcb[inline]{
8880   \cs_set:Nn \__problems_mccline:n{\problem_mcc_box_tl{~} #1}
8881 }{}
8882
8883 \stex_deactivate_macro:Nn \mcb {sproblem-environments}
we define the keys for the mcc macro
8884 \cs_new_protected:Nn \__problems_do_yes_param:Nn {
8885   \exp_args:Nx \str_if_eq:nnTF { \str_lowercase:n{ #2 } }{ yes }{
8886     \bool_set_true:N #1
8887   }{
8888     \bool_set_false:N #1
8889   }
8890 }
8891 \stex_keys_define:nnnn{mcc}{

8892   \tl_clear:N \l_stex_key_feedback_tl
8893   \bool_set_false:N \l_stex_key_T_bool
8894   \tl_clear:N \l_stex_key_Ttext_tl
8895   \tl_clear:N \l_stex_key_Ftext_tl
8896 }{
8897   feedback .tl_set:N      = \l_stex_key_feedback_tl ,
8898   T       .code:n      = {\bool_set_true:N \l_stex_key_T_bool} ,
8899   F       .code:n      = {\bool_set_false:N \l_stex_key_T_bool} ,
8900   Ttext  .tl_set:N      = \l_stex_key_Ttext_tl ,
8901   Ftext  .tl_set:N      = \l_stex_key_Ftext_tl ,
8902 }{id}
8903

\mcc
8904 \tl_set:Nn \problem_mcc_box_tl {
8905   \ltx@ifpackageloaded{amssymb}{$\square$}%
8906   \hbox{\vrule\vbox{\hrule width 6 pt\vskip 6pt\hrule}\vrule}
8907 }
8908 }
8909 \newcommand\mcc[2][]{%
8910   \stex_keys_set:nn{mcc}{#1}
8911   \tl_set:Nn \l_tmpa_tl{%
8912     \bool_if:NTF \l_stex_key_T_bool {
8913       \tl_if_empty:NTF \l_stex_key_Ttext_tl \problem@kw@correct \l_stex_key_Ttext_tl
8914     }%

```

```

8915     \tl_if_empty:NNTF \l_stex_key_Ftext_tl \problem@kw@wrong \l_stex_key_Ftext_tl
8916   }
8917   \tl_if_empty:NF \l_stex_key_feedback_tl {
8918     \\\emph{\l_stex_key_feedback_tl}
8919   }
8920 }

8921 \__problems_mccline:n{
8922   \stex_if_do_html:TF{
8923     \stex_annotate:nn{shtml:mcc={
8924       \bool_if:NTF \l_stex_key_T_bool {true}{false}
8925     }}{
8926       #2\stex_annotate:nn{shtml:mcc-solution={}{}\l_tmpa_tl}
8927     }
8928   }{
8929     #2\ifsolutions\footnote{\l_tmpa_tl}\fi
8930   }
8931 }
8932 }
8933 }
8934 \stex_deactivate_macro:Nn \mcc {mcb~environments}

```

(End of definition for `\mcc`. This function is documented on page 112.)

`scb (env.)`

```

8935 \tl_set:Nn\problem_scc_box_tl{$\bigcirc$}
8936
8937 \cs_new_protected:Nn \__problems_maybe_inline:n {
8938   \clist_if_in:NnTF \l_stex_key_style_clist {inline} {
8939     \let\__problems_oldpar:\stex_par:
8940     \cs_set:Nn\stex_par:{}}
8941   #1
8942   \let\stex_par:\__problems_oldpar:
8943 }{#1}
8944 }
8945
8946 \stex_new_stylable_env:nnnnnnn{scb}{0}{}
8947   \stex_keys_set:nn{style}{#1}
8948   \stex_if_do_html:T{
8949     \__problems_maybe_inline:n{
8950       \exp_args:Nne\begin{stex_annotate_env}{}{
8951         shtml:single-choice-block={}
8952         \clist_if_empty:NF \l_stex_key_style_clist ,,
8953           shtml:styles={\l_stex_key_style_clist}
8954         }
8955       }
8956     }
8957   }
8958   \tl_set:Nn\problem_scc_box_tl{}
8959 }
8960 \stex_deactivate_macro:Nn \mcb {sproblem~environments}
8961 \stex_deactivate_macro:Nn \scb {sproblem~environments}
8962 \stex_deactivate_macro:Nn \solution {sproblem~environments}
8963 \stex_deactivate_macro:Nn \hint {sproblem~environments}
8964 \stex_deactivate_macro:Nn \exnote {sproblem~environments}

```

```

8965 \stex_deactivate_macro:Nn \gnote {sproblem~environments}
8966 \stex_reactivate_macro:N \scc
8967 \cs_set:Nn \__problems_sccline:n{
8968   \begin{list}{}{
8969     \setlength\topsep{0pt}
8970     \setlength\parsep{0pt}
8971     \setlength\rightmargin{0pt}
8972   }\item[\problem_scc_box_tl] ##1 \end{list}
8973 }
8974 \stex_style_apply:
8975 }{
8976   \stex_style_apply:
8977   \stex_if_do_html:T{
8978     \__problems_maybe_inline:n { \end{stex_annotation_env} }
8979   }
8980 }{\par}{}}}
8981 \stexstylecb[inline]{
8982   \cs_set:Nn \__problems_sccline:n{\problem_scc_box_tl{~} #1}
8983 }{}
8984
8985 \stex_deactivate_macro:Nn \scc {sproblem~environments}

\scc

8986 \newcommand\scc[2][]{%
8987   \stex_keys_set:nn{mcc}{#1}
8988   \tl_set:Nn \l_tmpa_tl{%
8989     \bool_if:NTF \l_stex_key_T_bool {
8990       \tl_if_empty:NTF \l_stex_key_Ttext_tl \problem@kw@correct \l_stex_key_Ttext_tl
8991     }%
8992     \tl_if_empty:NTF \l_stex_key_Ftext_tl \problem@kw@wrong \l_stex_key_Ftext_tl
8993   }%
8994   \tl_if_empty:NF \l_stex_key_feedback_tl {
8995     \\\emph{\l_stex_key_feedback_tl}
8996   }%
8997 }%
8998 }

8999 \__problems_sccline:n{
9000   \stex_if_do_html:TF{
9001     \stex_annotation:nn{shtml:scc={%
9002       \bool_if:NTF \l_stex_key_T_bool {true}{false}
9003     }}%
9004     \#2\stex_annotation:nn{shtml:scc-solution={}}{\l_tmpa_tl}
9005   }%
9006   \#2\if\l_tmpa_tl\fi
9007   \#2\if\l_tmpa_tl\fi
9008   \#2\if\l_tmpa_tl\fi
9009   \#2\if\l_tmpa_tl\fi
9010   \#2\if\l_tmpa_tl\fi
9011 }%
9012 \stex_deactivate_macro:Nn \scc {scc~environments}

9013
9014
9015 \newcommand\yesTnoF{
9016   \begin{scb}[style=inline]

```

```

9017     \scc[T]{yes}~\scc[F]{no}
9018     \end{scb}
9019 }
9020 \newcommand\yesFnoT{
9021   \begin{scb}[style=inline]
9022     \scc[F]{yes}~\scc[T]{no}
9023   \end{scb}
9024 }
9025 \newcommand\trueTfalseF{
9026   \begin{scb}[style=inline]
9027     \scc[T]{true}~\scc[F]{false}
9028   \end{scb}
9029 }
9030 \newcommand\trueFfalseT{
9031   \begin{scb}[style=inline]
9032     \scc[F]{true}~\scc[T]{false}
9033   \end{scb}
9034 }

```

(End of definition for `\scc`. This function is documented on page ??.)

### \fillinsol

```

9035 \stex_keys_define:nnnn{fillinsol}{
9036   \tl_clear:N \l__problems_fillin_solution_tl
9037   \dim_zero:N \l_stex_key_testspace_dim
9038 }{
9039   testspace .dim_set:N = \l_stex_key_testspace_dim,
9040   exact .code:n = {\__problems_parse_fillin_arg:nnnn{exact}\#1 },
9041   numrange .code:n = {\__problems_parse_fillin_arg:nnnn{numrange}\#1 },
9042   regex .code:n = {\__problems_parse_fillin_arg:nnnn{regex}\#1 }
9043 }{}
9044
9045 \cs_new:Nn \__problems_parse_fillin_arg:nnnn {
9046   \stex_if_do_html:TF{
9047     \tl_set:Nn \l_tmpa_tl{\#3}
9048     \tl_set:Nn \l_tmpb_tl{T}
9049     \stex_annotation_invisible:nn{
9050       \shtml:fillin-case=\{#1\},
9051       \shtml:fillin-case-value=\{#2\},
9052       \shtml:fillin-case-verdict={
9053         \tl_if_eq:NNTF\l_tmpa_tl\l_tmpb_tl{true}{false}
9054       },
9055     }{\#4}
9056   }{
9057     \tl_put_right:Nn \l__problems_fillin_solution_tl {
9058       \#1{\~} & \#2{\~} & \#3{\~} & \#4{\~} \\
9059     }
9060   }
9061 }
9062
9063 \newcommand\fillinsol[2][]{%
9064   \quad
9065   \mode_if_math:TF{
9066     \hbox{\__problems_fillinsol:nn{\#1}{$\#2$}}%

```

```

9067    }{
9068      \__problems_fillinsol:nn{#1}{#2}
9069    }
9070    \quad
9071  }
9072 \cs_new_protected:Nn \__problems_fillinsol:nn {
9073   \stex_keys_set:nn{fillinsol}{#1}
9074   \stex_if_do_html:TF{
9075     \stex_annotation:nn{shtml:fillinsol={}}{\_stex_annotation_force_break:n{
9076       #2
9077       \l__problems_fillin_solution_tl
9078     } }
9079  }{
9080    \ifsolutions
9081      \textcolor{red}{\fbox{#2}}
9082      \tl_if_empty:NF \l__problems_fillin_solution_tl {
9083        \footnote{
9084          \halign{ ##\hfil & ##\hfil &##\hfil&##\hfil \cr
9085            \textbf{type} &\textbf{case} &\textbf{verdict} &\textbf{feedback} \cr
9086            \l__problems_fillin_solution_tl
9087          }
9088        }
9089      }
9090    \else
9091      \fbox{\dim_compare:nNnTF\l_stex_key_testspace_dim={0pt}{
9092        \phantom{\huge{#2}}
9093      }{
9094        \hspace{\l_stex_key_testspace_dim}
9095      }}
9096    \fi
9097  }
9098 }
9099 \stex_deactivate_macro:Nn \fillinsol {sproblem~environments}

```

(End of definition for `\fillinsol`. This function is documented on page 114.)

`\testemptypage`

```

9100 \newcommand\testemptypage[1][]{%
9101 \bool_if:NT \c__problems_test_bool {\vfill\begin{center}\hwexam@kw@testemptypage\end{center}}
9102 }

```

(End of definition for `\testemptypage`. This function is documented on page ??.)

`\testspace`

```

9103 \newcommand\testspace[1]{\bool_if:NT \c__problems_test_bool {\vspace*{#1}}}
9104 \newcommand\testsmallspace{\testspace{1cm}}
9105 \newcommand\testmedspace{\testspace{2cm}}
9106 \newcommand\testbigspace{\testspace{3cm}}

```

(End of definition for `\testspace`. This function is documented on page ??.)

`\testnewpage`

```

9107 \newcommand\testnewpage{\bool_if:NT \c__problems_test_bool {\newpage}}

```

(End of definition for `\testnewpage`. This function is documented on page ??.)

```

9108 
```

## 14.3 Implementation: The `hwexam` Package

### 14.3.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. Some come with their own conditionals that are set by the options, the rest is just passed on to the `problems` package.

```
9109 <*package>
9110 \ProvidesExplPackage{hwexam}{2023/10/13}{3.4.0}{homework assignments and exams}
9111 \RequirePackage{l3keys2e}
9112
9113 \keys_define:nn {hwexam / pkg} {
9114   multiple .default:n = { false },
9115   multiple .bool_set:N = \c_hwexam_multiple_bool,
9116   unknown .code:n = {
9117     \PassOptionsToPackage{\CurrentOption}{problem}
9118   }
9119 }
9120 \ProcessKeysOptions{ hwexam /pkg }
9121 \RequirePackage{problem}
```

`\hwexam_kw_*` For multilinguality, we define internal macros for keywords that can be specialized in `*.ldf` files.

```
9122 \AddToHook{begindocument}{
9123 \ExplSyntaxOn\makeatletter
9124 \input{hwexam-english.ldf}
9125 \ltx@ifpackageloaded[babel]{
9126 \clist_set:Nx \l_tmpa_clist {\exp_args:No \tl_to_str:n \bb@loaded}
9127 \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{ngerman}}{
9128 \input{hwexam-ngerman.ldf}
9129 }
9130 \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{finnish}}{
9131 \input{hwexam-finnish.ldf}
9132 }
9133 \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{french}}{
9134 \input{hwexam-french.ldf}
9135 }
9136 \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{russian}}{
9137 \input{hwexam-russian.ldf}
9138 }
9139 }{ }
9140 \makeatother\ExplSyntaxOff
9141 }
```

(End of definition for `\hwexam_kw_*`. This function is documented on page ??.)

### 14.3.2 Assignments

Then we set up a counter for problems and make the problem counter inherited from `problem.sty` depend on it. Furthermore, we specialize the `\prob@label` macro to take the assignment counter into account.

```

assignment (env.)
9142 \stex_keys_define:nnnn{ assignment }{
9143   \tl_clear:N \l_stex_key_number_tl
9144 \tl_clear:N \l_stex_key_given_tl
9145 \tl_clear:N \l_stex_key_due_tl
9146 }{
9147   number .tl_set:N = \l_stex_key_number_tl,
9148   given .tl_set:N = \l_stex_key_given_tl,
9149   due .tl_set:N = \l_stex_key_due_tl,
9150 unknown .code:n = {}
9151 }{id,title,style}
9152
9153 \newcounter{assignment}
9154 \stex_new_stylable_env:nnnnnnn {assignment}{0}{}{
9155   \cs_if_exist:NTF \l_hwexam_includeassignment_keys_tl {
9156     \tl_put_left:Nn \l_hwexam_includeassignment_keys_tl {\#1,}
9157     \exp_args:Nno \stex_keys_set:nn{assignment}{
9158       \l_hwexam_includeassignment_keys_tl
9159     }
9160   }{
9161     \stex_keys_set:nn{assignment}{\#1}
9162   }
9163 \tl_if_empty:NF \l_stex_key_number_tl {
9164 \global\setcounter{assignment}{\int_eval:n{\l_stex_key_number_tl-1}}
9165 }
9166 \global\refstepcounter{assignment}
9167 \setcounter{sproblem}{0}
9168 \def\thesproblem{\theassignment.\arabic{sproblem}}
9169 \stex_style_apply:
9170 \_stex_do_id:
9171 }{
9172 \stex_style_apply:
9173 }{
9174 \par\begin{center}
9175 \textbf{\Large\assignmentautorefname~\theassignment}
9176 \tl_if_empty:NF \l_stex_key_title_tl {
9177 {\~}---\l_stex_key_title_tl
9178 }
9179 }\par\smallskip
9180 \textbf{
9181 \tl_if_empty:NF \l_stex_key_given_tl {
9182 \hwexam@kw@given :~\l_stex_key_given_tl\quad
9183 }
9184 \tl_if_empty:NF \l_stex_key_due_tl {
9185 \hwexam@kw@due :~\l_stex_key_due_tl\quad
9186 }
9187 }
9188 \end{center}
9189 \par\bigskip
9190 }{
9191 \par\pagebreak
9192 }{}}

\includeassignment

```

```

9193 \NewDocumentCommand\includeassignment{O{} m}{
9194   \group_begin:
9195   \tl_set:Nn \l_hwexam_includeassignment_keys_tl {#1}
9196   \stex_keys_set:nn{includeproblem}{#1}
9197   \exp_args:Nno \use:nn{\inputref[]}\l_stex_key_mhrepos_str]{#2}
9198   \group_end:
9199 }
```

(End of definition for `\includeassignment`. This function is documented on page ??.)

Restoring information about problems:

```

9200 \prop_new:N \c_@@_problems_prop
9201 \tl_set:Nn \c_@@_total_mins_tl {0}
9202 \tl_set:Nn \c_@@_total_pts_tl {0}
9203 \int_new:N \c_@@_total_problems_int
9204 \cs_set_protected:Npn \problem@restore #1 #2 #3 {
9205   \int_gincr:N \c_@@_total_problems_int
9206   \prop_gput:Nnn \c_@@_problems_prop {#1}{#2}{#3}
9207   \tl_gset:Nx \c_@@_total_pts_tl { \int_eval:n { \c_@@_total_pts_tl + #2 } }
9208   \tl_gset:Nx \c_@@_total_mins_tl { \int_eval:n { \c_@@_total_mins_tl + #2 } }
9209 }
```

`\correction@table` This macro generates the correction table

```

9210 \newcommand\correction@table{
9211   \int_compare:nNnT \c_@@_total_problems_int = 0 {
9212     \int_incr:N \c_@@_total_problems_int
9213     \prop_put:Nnn \c_@@_problems_prop {\sim}{\sim}{\sim}
9214   }
9215   \tl_clear:N \l_tmpa_tl
9216   \tl_clear:N \l_tmpb_tl
9217   \tl_clear:N \l_tmpc_tl
9218   \prop_map_inline:Nn \c_@@_problems_prop {
9219     \tl_put_right:Nn \l_tmpa_tl { ##1 & }
9220     \tl_put_right:Nx \l_tmpb_tl { \use_i:nn ##2 & }
9221     \tl_put_right:Nn \l_tmpc_tl { & }
9222   }
9223   \resizebox{\textwidth}{!}{%
9224     \exp_args:Nne \begin{tabular}{|l|*{\int_use:N \c_@@_total_problems_int}{c|}c||l|}\hline
9225       &\exp_args:Ne \multicolumn{\int_eval:n{ \c_@@_total_problems_int + 1}}{c||}{}
9226       {\footnotesize\hwexam@kw@forgrading} &\hline
9227       \hwexam@kw@probs & \l_tmpa_tl \hwexam@kw@sum & \hwexam@kw@grade\hline
9228       \hwexam@kw@pts & \l_tmpb_tl \c_@@_total_pts_tl & \hline
9229       \hwexam@kw@reached & \l_tmpc_tl & [.7cm]\hline
9230     \end{tabular}}}
```

(End of definition for `\correction@table`. This function is documented on page ??.)

`\testheading`

```

9231 \def\hwexamheader{\input{hwexam-default.header}}
9232
9233 \def\hwexamminutes{
9234   \tl_if_empty:NTF \hwexam@duration {
9235     \hwexam@min}~\hwexam@minutes@kw
9236   }{
9237   \hwexam@duration
```

```

9238 }
9239 }
9240
9241 \stex_keys_define:nnnn{ hwexam / testheading }{
9242 \tl_clear:N \hwexam@min
9243 \tl_clear:N \hwexam@duration
9244 \tl_clear:N \hwexam@reqpts
9245 \tl_clear:N \hwexam@tools
9246 }{
9247 min .tl_set:N = \hwexam@min,
9248 duration .tl_set:N = \hwexam@duration,
9249 reqpts .tl_set:N = \hwexam@reqpts,
9250 tools .tl_set:N = \hwexam@tools
9251 }{}}
9252
9253 \newenvironment{testheading}[1][]{%
9254 \stex_keys_set:nn { hwexam / testheading}{#1}%
9255
9256 \tl_set_eq:NN \hwexam@totalpts \c_@@_total_pts_tl
9257 \tl_set_eq:NN \hwexam@totalmin \c_@@_total_mins_tl
9258 \tl_set:Nx \hwexam@checktime {\int_eval:n { \hwexam@min - \hwexam@totalmin }}%
9259
9260 \newif\if@bonuspoints
9261 \tl_if_empty:NTF \hwexam@reqpts {
9262 \@bonuspointsfalse
9263 }{
9264 \tl_set:Nx \hwexam@bonuspts {
9265 \int_eval:n{\hwexam@totalpts - \hwexam@reqpts}
9266 }
9267 \@bonuspointstrue
9268 }
9269
9270 \makeatletter\hwexamheader\makeatother
9271 }{
9272 \newpage
9273 }

```

(End of definition for `\testheading`. This function is documented on page ??.)

9274 ⟨/package⟩

### 14.3.3 Leftovers

at some point, we may want to reactivate the logos font, then we use

```

here we define the logos that characterize the assignment
\font\bierfont=../assignments/bierglas
\font\denkerfont=../assignments/denker
\font\uhrfont=../assignments/uhr
\font\warnschildfont=../assignments/achtung

```

```

\newcommand\bierglas{{\bierfont\char65}}
\newcommand\denker{{\denkerfont\char65}}
\newcommand\uhr{{\uhrfont\char65}}

```

```

\newcommand\warnschild{{\warnschildfont\char 65}}
\newcommand\hardA{\warnschild}
\newcommand\longA{\uhr}
\newcommand\thinkA{\denker}
\newcommand\discussA{\bierglas}

```

## 14.4 Tikzinput Implementation

```

9275 <@=tikzinput>
9276 <*package>
9277
9278 %%%%%%%%%%%%%% tikzinput.dtx %%%%%%%%%%%%%%
9279
9280 \ProvidesExplPackage{tikzinput}{2023/10/13}{3.4.0}{tikzinput package}
9281 \RequirePackage{l3keys2e}
9282
9283 \keys_define:nn { tikzinput } {
9284   image .bool_set:N = \c_tikzinput_image_bool,
9285   image .default:n = false ,
9286   unknown .code:n = {}
9287 }
9288
9289 \ProcessKeysOptions { tikzinput }
9290
9291 \bool_if:NTF \c_tikzinput_image_bool {
9292   \RequirePackage{graphicx}
9293
9294   \providecommand\usetikzlibrary[]{}
9295   \newcommand\tikzinput[2][]{\includegraphics[#1]{#2}}
9296 }
9297   \RequirePackage{tikz}
9298   \RequirePackage{standalone}
9299
9300 \newcommand \tikzinput [2] [] {
9301   \setkeys{Gin}{#1}
9302   \ifx \Gin@ewidth \Gin@exclamation
9303     \ifx \Gin@eheight \Gin@exclamation
9304       \input { #2 }
9305     \else
9306       \resizebox{!}{\Gin@eheight }{
9307         \input { #2 }
9308       }
9309     \fi
9310   \else
9311     \ifx \Gin@eheight \Gin@exclamation
9312       \resizebox{ \Gin@ewidth }{! }{
9313         \input { #2 }
9314       }
9315     \else
9316       \resizebox{ \Gin@ewidth }{ \Gin@eheight }{
9317         \input { #2 }
9318       }
9319 
```

```
9319     \fi
9320   \fi
9321 }
9322 }
9323
9324 \newcommand \ctikzinput [2] [] {
9325   \begin{center}
9326     \tikzinput [#1] {#2}
9327   \end{center}
9328 }
9329 </package>
```

# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

| Symbols                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \\$ .....                       | 2089, 2092, 2096                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \; .....                        | <u>7528</u> , <u>7559</u> , <u>7572</u> , <u>7574</u> , <u>7587</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| @@ commands:                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \c_@@_problems_prop .....       | <u>9200</u> , <u>9206</u> , <u>9213</u> , <u>9218</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| \c_@@_total_mins_tl .           | <u>9201</u> , <u>9208</u> , <u>9257</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \c_@@_total_problems_int .....  | <u>9203</u> , <u>9205</u> , <u>9211</u> , <u>9212</u> , <u>9224</u> , <u>9225</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| \c_@@_total_pts_tl .....        | <u>9202</u> , <u>9207</u> , <u>9228</u> , <u>9256</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| \\ .....                        | <u>53</u> , <u>77</u> ,<br><u>1042</u> , <u>7859</u> , <u>7864</u> , <u>8219</u> , <u>8222</u> , <u>8918</u> ,<br><u>8996</u> , <u>9058</u> , <u>9226</u> , <u>9227</u> , <u>9228</u> , <u>9229</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| \{ .....                        | <u>7577</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \} .....                        | <u>7577</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \_ .....                        | <u>44</u> , <u>734</u> , <u>7891</u> , <u>9101</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| \_comp .....                    | <u>3971</u> ,<br><u>4177</u> , <u>4529</u> , <u>5225</u> , <u>6044</u> , <u>6068</u> , <u>6152</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| \_customthiscomp .....          | <u>6571</u> , <u>6572</u> , <u>6584</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \_defcomp .....                 | <u>6076</u> , <u>7010</u> , <u>7019</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \_stex_html_do_output_bool .... | <u>373</u> ,<br><u>374</u> , <u>377</u> , <u>382</u> , <u>386</u> , <u>417</u> , <u>420</u> , <u>2158</u> , <u>7519</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \_thiscomp .....                | <u>6554</u> , <u>6565</u> , <u>6566</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \_varcomp .....                 | <u>4510</u> , <u>4809</u> , <u>4818</u> , <u>4975</u> , <u>5013</u> ,<br><u>5259</u> , <u>5701</u> , <u>6000</u> , <u>6013</u> , <u>6026</u> , <u>6072</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| \  .....                        | <u>2088</u> , <u>2091</u> , <u>2095</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>A</b>                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \activateexcursion .....        | <u>109</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| \addbibresource .....           | <u>80</u> , <u>1949</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \addcontentsline .....          | <u>7889</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \addmhbibresource .....         | <u>80</u> , <u>1943</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \addtocounter .....             | <u>8065</u> , <u>8445</u> , <u>8446</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \AddToHook .....                | <u>1032</u> , <u>1035</u> ,<br><u>7323</u> , <u>7390</u> , <u>7393</u> , <u>7505</u> , <u>8334</u> , <u>9122</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \aftergroup .....               | <u>143</u> , <u>5213</u> , <u>7388</u> , <u>7502</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| \afterprematurestop .....       | <u>108</u> , <u>7948</u> , <u>7959</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \anscls .....                   | <u>8768</u> , <u>8802</u> , <u>8836</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \apply .....                    | <u>88</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| \arabic .....                   | <u>8375</u> , <u>8378</u> , <u>9168</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \arg .....                      | <u>46</u> , <u>93</u> , <u>5438</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| \argarraymap .....              | <u>92</u> , <u>4238</u> , <u>4749</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| \argmap .....                   | <u>92</u> , <u>4237</u> , <u>4379</u> , <u>4725</u> , <u>7560</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| \argsep .....                   | <u>40</u> , <u>92</u> , <u>4236</u> ,<br><u>4374</u> , <u>4709</u> , <u>7528</u> , <u>7529</u> , <u>7530</u> , <u>7536</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>B</b>                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \assign .....                   | <u>64</u> , <u>129</u> , <u>3008</u> , <u>3364</u> , <u>3491</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \assignment (env.) .....        | <u>73</u> , <u>116</u> , <u>9142</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| \assignmentautorefname .....    | <u>9175</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \assignMorphism .....           | <u>129</u> , <u>3009</u> , <u>3493</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \assumption .....               | <u>7301</u> , <u>7406</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| \ast .....                      | <u>7551</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \AtBeginDocument .....          | <u>1002</u> ,<br><u>1255</u> , <u>1441</u> , <u>1444</u> , <u>1492</u> , <u>7760</u> , <u>7903</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| \AtEndDocument .....            | <u>742</u> , <u>1493</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \AtEndOfFile .....              | <u>2037</u> , <u>2052</u> , <u>2066</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \author .....                   | <u>8199</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \autoref .....                  | <u>84</u> , <u>1636</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>B</b>                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \backmatter .....               | <u>1456</u> , <u>1457</u> , <u>1458</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \baselineskip .....             | <u>8188</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \beameritemnestingprefix .....  | <u>8273</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \begin .....                    | <u>96</u> , <u>135</u> , <u>136</u> ,<br><u>141</u> , <u>201</u> , <u>1298</u> , <u>1378</u> , <u>2049</u> , <u>2063</u> ,<br><u>2081</u> , <u>2238</u> , <u>2260</u> , <u>2277</u> , <u>2527</u> , <u>2785</u> ,<br><u>2831</u> , <u>4772</u> , <u>6905</u> , <u>7215</u> , <u>7248</u> , <u>7270</u> ,<br><u>7308</u> , <u>7379</u> , <u>7451</u> , <u>7647</u> , <u>7892</u> , <u>8025</u> ,<br><u>8039</u> , <u>8049</u> , <u>8082</u> , <u>8103</u> , <u>8111</u> , <u>8149</u> ,<br><u>8150</u> , <u>8151</u> , <u>8152</u> , <u>8153</u> , <u>8154</u> , <u>8160</u> ,<br><u>8162</u> , <u>8234</u> , <u>8262</u> , <u>8263</u> , <u>8415</u> , <u>8512</u> ,<br><u>8534</u> , <u>8586</u> , <u>8650</u> , <u>8699</u> , <u>8751</u> , <u>8810</u> ,<br><u>8851</u> , <u>8866</u> , <u>8951</u> , <u>8968</u> , <u>9016</u> , <u>9021</u> ,<br><u>9026</u> , <u>9031</u> , <u>9101</u> , <u>9174</u> , <u>9224</u> , <u>9325</u> |
| \begingroup .....               | <u>453</u> , <u>1921</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \bf .....                       | <u>8187</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \bfseries .....                 | <u>8496</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \bgroup .....                   | <u>3814</u> , <u>4855</u> , <u>5067</u> , <u>8038</u> , <u>8127</u> , <u>8142</u> ,<br><u>8219</u> , <u>8222</u> , <u>8603</u> , <u>8664</u> , <u>8713</u> , <u>8765</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \bigcirc .....                  | <u>8936</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \bigskip .....                  | <u>8465</u> , <u>9189</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| \blindfragment (env.) .....     | <u>82</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| bool commands:                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \bool_if:NTF .....              | <u>291</u> , <u>713</u> , <u>765</u> ,<br><u>766</u> , <u>772</u> , <u>1211</u> , <u>1445</u> , <u>2141</u> , <u>2396</u> ,<br><u>2746</u> , <u>2779</u> , <u>2878</u> , <u>2950</u> , <u>2976</u> , <u>3473</u> ,<br><u>3682</u> , <u>4870</u> , <u>4973</u> , <u>5223</u> , <u>5257</u> , <u>5799</u> ,<br><u>5893</u> , <u>5899</u> , <u>6034</u> , <u>7000</u> , <u>7276</u> , <u>7324</u> ,<br><u>7362</u> , <u>7381</u> , <u>7384</u> , <u>7386</u> , <u>7391</u> , <u>7394</u> ,<br><u>7402</u> , <u>7444</u> , <u>7458</u> , <u>7480</u> , <u>7506</u> , <u>7795</u> ,<br><u>7803</u> , <u>7840</u> , <u>7919</u> , <u>7963</u> , <u>8124</u> , <u>8130</u> ,                                                                                                                                                                                                                                                                   |

8164, 8179, 8198, 8233, 8243, 8321,  
 8327, 8397, 8418, 8452, 8457, 8466,  
 8494, 8515, 8523, 8526, 8539, 8540,  
 8544, 8545, 8661, 8672, 8710, 8721,  
 8762, 8774, 8838, 8843, 8912, 8925,  
 8990, 9003, 9101, 9103, 9107, 9291  
 $\backslash$ bool\_if:nTF ..... 377  
 $\backslash$ bool\_if\_exist:NTF ..... 350, 366  
 $\backslash$ bool\_lazy\_any:nTF ..... 4066, 4106  
 $\backslash$ bool\_lazy\_any\_p:n ..... 782  
 $\backslash$ bool\_new:N ..... 373, 2138, 2381, 2804, 4781, 5193,  
 5768, 5890, 8354, 8392, 8393, 8394  
 $\backslash$ bool\_not\_p:n ..... 781, 785  
 $\backslash$ bool\_set\_false:N .. 277, 382, 420,  
 1189, 1193, 2139, 2158, 2382, 2759,  
 2826, 3446, 4641, 4784, 5197, 5769,  
 5894, 7134, 7283, 7331, 7359, 7500,  
 7519, 7520, 7779, 7820, 7979, 8360,  
 8395, 8434, 8437, 8888, 8893, 8899  
 $\backslash$ bool\_set\_true:N ... 282, 352, 368,  
 374, 386, 417, 702, 708, 1186, 2157,  
 2743, 2823, 3458, 3468, 3679, 3815,  
 4875, 5115, 5194, 5423, 5432, 5548,  
 5559, 5657, 5742, 5800, 5811, 5844,  
 5873, 5919, 6115, 6526, 6557, 6574,  
 6619, 7146, 7313, 7792, 7832, 7977,  
 7985, 7986, 7987, 7988, 7989, 7990,  
 8430, 8435, 8438, 8499, 8886, 8898  
 $\backslash$ bool\_while\_do:Nn ..... 1187  
 $\backslash$ bool\_while\_do:nn ..... 780, 1988, 7150, 7162, 7174, 7191, 7203  
 $\backslash$ l\_tmpa\_bool ..... 1186, 1187,  
 1189, 1193, 3446, 3458, 3468, 3473  
 box commands:  
 $\backslash$ box\_clear:N ..... 2161  
 $\backslash$ c\_empty\_box .. 3996, 4003, 8840, 8845  
 $\backslash$ l\_tmpa\_box 2156, 2161, 3694, 4229,  
 8127, 8142, 8603, 8664, 8713, 8765  
boxed ..... 110

**C**

$\backslash$ catcode ..... 44, 53, 453, 734, 1042,  
 1689, 2087, 2088, 2089, 2090, 2091,  
 2092, 2094, 2095, 2096, 2438, 2439  
 $\backslash$ cdot ..... 7559, 7572  
 $\backslash$ centering ..... 7893, 8083  
 $\backslash$ chapter ..... 27, 82, 7928, 7929  
 $\backslash$ chaptername ..... 7864, 7925, 7926  
 $\backslash$ chaptertitlename ..... 7864  
 $\backslash$ circ ..... 54  
 $\backslash$ clearpage ..... 1451, 1461, 1472, 1483

clist commands:

$\backslash$ clist\_clear:N 87, 500, 3723, 4155,  
 5116, 6408, 6507, 6822, 6827, 7408  
 $\backslash$ clist\_count:N ..... 6388, 6398  
 $\backslash$ clist\_count:n ..... 4757, 6490  
 $\backslash$ clist\_get>NN ..... 556, 607  
 $\backslash$ clist\_if\_empty:NTF ..... 276, 552, 603, 3831, 4327,  
 6387, 6433, 6690, 6894, 8853, 8953  
 $\backslash$ clist\_if\_in:NnTF ..... 66, 69, 93, 6988, 8339, 8342, 8345,  
 8348, 8939, 9127, 9130, 9133, 9136  
 $\backslash$ clist\_if\_in:nnTF ..... 6516  
 $\backslash$ clist\_item:Nn ..... 4336  
 $\backslash$ clist\_item:nn ..... 4768  
 $\backslash$ clist\_map\_function:NN ... 5117, 6436  
 $\backslash$ clist\_map\_function:nN ..... 2696, 2721, 3613, 3633, 3654  
 $\backslash$ clist\_map\_inline:Nn ..... 88,  
 97, 279, 1067, 2848, 3833, 4893,  
 5086, 6215, 6265, 6691, 6804, 6840  
 $\backslash$ clist\_map\_inline:nn ..... 464, 513, 5589, 5688, 5717, 6213, 7090  
 $\backslash$ clist\_pop:NN ..... 4338  
 $\backslash$ clist\_put\_right:Nn ..... 89, 5130, 5134, 6415, 6534, 6536  
 $\backslash$ clist\_set:Nn ..... 38, 85, 1066, 4994, 8338, 9126  
 $\backslash$ clist\_set\_eq:NN ..... 83, 6910  
 $\backslash$ l\_tmpa\_clist ..... 1066, 1067, 8338, 8339, 8342, 8345,  
 8348, 9126, 9127, 9130, 9133, 9136  
 $\backslash$ clstinputmhlisting ..... 83, 2036  
 $\backslash$ cmhgraphics ..... 83, 2036  
 $\backslash$ cmhtikzinput ..... 118, 2036  
 $\backslash$ columnbox ..... 8101, 8103, 8121  
 $\backslash$ comp ..... 32, 33, 36, 46, 90, 93,  
 103, 131, 3971, 4177, 4350, 4510,  
 4529, 4571, 4577, 4579, 4809, 4818,  
 5013, 5242, 5243, 5577, 5701, 5921,  
 5926, 5941, 6000, 6013, 6026, 6044,  
 6046, 6053, 6152, 6375, 6383, 6566,  
 6580, 6581, 7010, 7011, 7019, 7525,  
 7527, 7534, 7536, 7538, 7539, 7544,  
 7546, 7551, 7553, 7554, 7561, 7562,  
 7563, 7564, 7567, 7569, 7573, 7574,  
 7575, 7577, 7587, 7594, 7596, 7598  
 $\backslash$ compemph ..... 103, 6053  
 $\backslash$ conclude ..... 7300, 7406  
 $\backslash$ conclusion 50–52, 99, 101, 6979, 7099, 7115  
 $\backslash$ copymod ..... 62, 63, 3623, 3625, 3627  
 $\backslash$ copymodule ..... 3510, 3512  
 $\backslash$ counterwithin ..... 7932, 7944  
 $\backslash$ cr ..... 9084, 9085

**cs** commands:

\cs:w . . . . . 539, 542, 574, 735, 2171,  
2400, 2401, 2402, 2409, 2413, 2419,  
5321, 6256, 6306, 6308, 6928, 6941  
\cs\_argument\_spec:N . . . . . 4020  
\cs\_end: . . . . . 539, 542, 574, 735, 2171,  
2400, 2401, 2402, 2411, 2415, 2419,  
5321, 6256, 6306, 6308, 6928, 6941  
\cs\_generate\_from\_arg\_count:NNnn  
. . . . . 3824,  
4637, 4884, 5077, 5161, 5362, 5394  
\cs\_generate\_variant:Nn . . . . .  
. . . . . 108, 138, 252, 325, 338,  
652, 676, 685, 697, 721, 879, 884,  
888, 969, 1118, 1152, 1550, 1877,  
2203, 2561, 2565, 2570, 2588, 2637,  
2659, 2672, 3141, 5218, 5235, 5552  
\cs\_gset:Npn . . . . . 2434  
\cs\_if\_eq:NNTF . . . . . 235,  
894, 945, 1417, 2227, 2238, 2241,  
2260, 2263, 4022, 4710, 5633, 5635  
\cs\_if\_exist:NTF . . . . . 403,  
1090, 1323, 1327, 1365, 1368, 1392,  
1396, 1425, 1431, 1446, 1456, 1469,  
1480, 1635, 1783, 1807, 1827, 1833,  
1837, 2109, 2649, 2654, 3998, 4005,  
4017, 4448, 4456, 4908, 4915, 5102,  
5148, 5312, 5338, 5682, 5921, 5926,  
6472, 6769, 6772, 6776, 6779, 6784,  
6787, 6791, 6794, 7859, 7864, 7867,  
7871, 7875, 7879, 7883, 7925, 7928,  
7931, 7937, 7940, 7943, 8085, 8093,  
8105, 8115, 8219, 8222, 8400, 9155  
\cs\_new:Nn . . . . . 186, 303, 526,  
640, 646, 653, 662, 678, 687, 893,  
2212, 2395, 2443, 2964, 3791, 4125,  
4138, 4342, 4409, 4426, 4431, 4586,  
5157, 5370, 5467, 6193, 6248, 6444,  
6450, 6453, 6463, 6885, 8380, 9045  
\cs\_new:Npn 638, 715, 716, 724, 725,  
889, 2902, 2933, 4131, 4144, 5613, 5986  
\cs\_new\_nopar:Nn . . . . . 460, 475  
\cs\_new\_protected:Nn . . . . . 43,  
52, 60, 65, 75, 103, 115, 131, 140,  
155, 232, 253, 310, 321, 331, 340,  
358, 380, 423, 429, 443, 449, 531,  
538, 551, 565, 573, 592, 602, 617,  
639, 718, 728, 740, 745, 764, 799,  
806, 836, 871, 881, 885, 907, 924,  
934, 940, 944, 971, 975, 981, 997,  
1003, 1016, 1082, 1089, 1112, 1120,  
1134, 1145, 1155, 1241, 1249, 1295,  
1303, 1307, 1317, 1363, 1376, 1386,  
1423, 1503, 1513, 1539, 1579, 1613,  
1624, 1634, 1642, 1682, 1740, 1751,  
1782, 1788, 1792, 1806, 1813, 1855,  
1880, 1896, 1920, 1931, 1943, 1970,  
2007, 2083, 2099, 2122, 2126, 2130,  
2145, 2149, 2155, 2166, 2174, 2204,  
2226, 2232, 2250, 2273, 2282, 2298,  
2305, 2318, 2331, 2346, 2354, 2363,  
2384, 2406, 2427, 2449, 2453, 2461,  
2473, 2558, 2562, 2566, 2575, 2583,  
2589, 2607, 2614, 2626, 2640, 2647,  
2660, 2674, 2691, 2700, 2711, 2716,  
2725, 2736, 2741, 2765, 2783, 2797,  
2805, 2846, 2855, 2867, 2871, 2890,  
2898, 2906, 2921, 2934, 2987, 2993,  
3014, 3057, 3090, 3157, 3176, 3196,  
3202, 3207, 3234, 3265, 3290, 3316,  
3366, 3410, 3477, 3560, 3588, 3598,  
3693, 3701, 3764, 3776, 3795, 3840,  
3848, 3902, 3911, 3993, 4002, 4007,  
4014, 4035, 4052, 4065, 4076, 4078,  
4091, 4102, 4194, 4210, 4261, 4273,  
4300, 4321, 4326, 4347, 4358, 4367,  
4477, 4519, 4561, 4590, 4597, 4618,  
4632, 4668, 4673, 4678, 4709, 4782,  
4826, 4840, 4853, 4903, 4921, 4934,  
4940, 4950, 4958, 5021, 5035, 5048,  
5049, 5051, 5095, 5128, 5147, 5159,  
5171, 5200, 5222, 5237, 5256, 5281,  
5286, 5291, 5358, 5374, 5380, 5415,  
5421, 5429, 5484, 5545, 5554, 5570,  
5583, 5617, 5628, 5649, 5681, 5713,  
5759, 5762, 5772, 5785, 5798, 5821,  
5822, 5823, 5827, 5853, 5856, 5882,  
5892, 5980, 6032, 6053, 6147, 6172,  
6179, 6196, 6202, 6251, 6311, 6319,  
6326, 6331, 6385, 6407, 6414, 6418,  
6481, 6506, 6532, 6564, 6570, 6589,  
6601, 6612, 6630, 6640, 6680, 6689,  
6697, 6706, 6716, 6727, 6741, 6765,  
6847, 6899, 6945, 6958, 6996, 7067,  
7083, 7159, 7247, 7254, 7258, 7269,  
7284, 7401, 7424, 7442, 7757, 7853,  
7924, 7936, 7975, 8013, 8037, 8042,  
8048, 8057, 8129, 8253, 8483, 8577,  
8641, 8690, 8741, 8884, 8938, 9072  
\cs\_new\_protected:Npn . . . . .  
. . . . . 110, 300, 432, 435,  
439, 440, 523, 900, 903, 1237, 1321,  
1390, 1406, 1494, 1496, 1826, 1832,  
1836, 1851, 2218, 2301, 2377, 2860,  
2927, 2971, 3000, 3031, 3045, 3128,  
3143, 3277, 3606, 4446, 4603, 4610,  
4660, 4716, 4725, 4750, 5101, 5114,  
5179, 5185, 5189, 5219, 5269, 5310,

|                                             |                                   |  |
|---------------------------------------------|-----------------------------------|--|
| \csname .....                               | 362                               |  |
| 391, 450, 2085, 2086, 2087, 2088,           |                                   |  |
| 2089, 2094, 2095, 2096, 2399, 2662          |                                   |  |
| \ctikzinput .....                           | 118, 9324                         |  |
| \CurrentFile .....                          | 1019, 1021, 1023                  |  |
| \CurrentFilePath .....                      | 1019, 1020, 1023                  |  |
| \currentgrouplevel .....                    | 322, 344, 345, 346,               |  |
| 348, 350, 352, 360, 362, 364, 366, 368      |                                   |  |
| \CurrentOption .....                        | 10, 7782, 7783,                   |  |
| 7784, 7785, 7824, 7825, 8315, 9117          |                                   |  |
| \Currentsectionlevel .....                  | 82, 1285                          |  |
| \currentsectionlevel .....                  | 82, 1276                          |  |
| <b>D</b>                                    |                                   |  |
| \DeclareOption .....                        | 10                                |  |
| \def ...                                    | 91, 141, 397, 401, 404, 406, 452, |  |
| 655, 1239, 1243, 1260, 1265, 2058,          |                                   |  |
| 2073, 3971, 4177, 4198, 4508, 4510,         |                                   |  |
| 4527, 4529, 4809, 4816, 4818, 5011,         |                                   |  |
| 5013, 5242, 5243, 5701, 5941, 5970,         |                                   |  |
| 5336, 5461, 5597, 5911, 5931, 5938,         |                                   |  |
| 6068, 6072, 6076, 6082, 6086, 6090,         |                                   |  |
| 6094, 6098, 6102, 6106, 6110, 6114,         |                                   |  |
| 6352, 6374, 6379, 6470, 6489, 6502,         |                                   |  |
| 6554, 6801, 6838, 7148, 7172, 7189,         |                                   |  |
| 7201, 7244, 7607, 8489, 8632, 8635          |                                   |  |
| \cs_parameter_spec:N .....                  | 642, 656                          |  |
| \cs_prefix_spec:N .....                     | 654                               |  |
| \cs_set:Nn .....                            | 8865, 8880, 8941, 8967, 8982      |  |
| \cs_set:Npn .....                           | 543, 578, 583,                    |  |
| 731, 732, 828, 829, 830, 831, 949,          |                                   |  |
| 956, 1091, 1094, 1107, 2180, 2429,          |                                   |  |
| 2462, 2676, 2678, 2681, 2694, 2719,         |                                   |  |
| 2744, 2758, 3825, 4038, 4231, 4232,         |                                   |  |
| 4236, 4237, 4238, 4350, 4380, 4621,         |                                   |  |
| 4637, 4731, 4756, 4885, 4923, 5078,         |                                   |  |
| 5161, 5363, 5395, 5715, 6509, 6515,         |                                   |  |
| 6576, 7516, 7517, 7518, 7835, 8134          |                                   |  |
| \cs_set:Npx .....                           | 2618                              |  |
| \cs_set_eq:NN .....                         | 41, 312, 316, 822, 823,           |  |
| 824, 1257, 1264, 1382, 2655, 3011,          |                                   |  |
| 3012, 4018, 4484, 4488, 4635, 4688,         |                                   |  |
| 4701, 4916, 5149, 5152, 5226, 5228,         |                                   |  |
| 5260, 5262, 5438, 5441, 5469, 5470,         |                                   |  |
| 5683, 5686, 5825, 7683, 7695, 8177          |                                   |  |
| \cs_set_protected:Nn .....                  |                                   |  |
| ... 819, 826, 1563, 2969, 7854, 7887        |                                   |  |
| \cs_set_protected:Npn .....                 | 445,                              |  |
| 1276, 1285, 3994, 3995, 4374, 4379,         |                                   |  |
| 4385, 6080, 7684, 7926, 7929, 7938,         |                                   |  |
| 7941, 8178, 8275, 8278, 8281, 9204          |                                   |  |
| \cs_set_protected:Npx .....                 | 6572                              |  |
| \cs_undefine:N .....                        | 307, 317, 364, 1202               |  |
| \l_tmpa_cs .....                            | 4231, 4241, 5715, 5733            |  |
| <b>E</b>                                    |                                   |  |
| \edef ...                                   | 120, 2087, 2088, 2089, 6566, 8031 |  |
| \egroup .....                               |                                   |  |
| 3838, 4900, 5093, 8045, 8127, 8144,         |                                   |  |
| 8219, 8222, 8617, 8678, 8727, 8780          |                                   |  |
| \eject .....                                | 9101                              |  |
| \ellipses .....                             | 95,                               |  |
| 96, 4735, 4736, 5129, 5131, 5689, 5727      |                                   |  |
| \else .....                                 | 390, 402, 1884,                   |  |
| 1967, 2214, 5916, 6095, 8273, 8495,         |                                   |  |
| 8600, 8616, 9090, 9305, 9310, 9315          |                                   |  |
| \emph {17, 19, 102, 6103, 7231, 7319, 7336, |                                   |  |
| 8622, 8683, 8732, 8785, 8918, 8996          |                                   |  |
| \end .....                                  | 136, 141, 1304,                   |  |
| 1358, 2049, 2063, 2081, 2241, 2263,         |                                   |  |
| 2286, 2546, 2801, 2886, 4774, 6922,         |                                   |  |
| 7229, 7255, 7317, 7334, 7385, 7387,         |                                   |  |
| 7465, 7469, 7671, 7897, 7951, 7960,         |                                   |  |
| 8033, 8045, 8059, 8099, 8113, 8121,         |                                   |  |

8134, 8136, 8149, 8150, 8151, 8152,  
 8153, 8154, 8175, 8236, 8270, 8271,  
 8449, 8532, 8550, 8609, 8614, 8670,  
 8675, 8719, 8724, 8772, 8777, 8834,  
 8870, 8876, 8972, 8978, 9018, 9023,  
 9028, 9033, 9101, 9188, 9230, 9327  
`\endcsname` . . . . . 362, 390, 391,  
 450, 2085, 2086, 2087, 2088, 2089,  
 2094, 2095, 2096, 2399, 2662, 8273  
`\endgroup` . . . . . 455, 457, 1928  
`\endinput` . . . . . 1763, 2288  
`\ensuremath` . . . . . 5801, 5812  
 environments:  
     `assignment` . . . . . 73, 116, [9142](#)  
     `blindfragment` . . . . . 82  
     `exnote` . . . . . 1, [8690](#)  
     `extstructure` . . . . . 96, [6211](#)  
     `extstructure*` . . . . . 96  
     `frame` . . . . . 1, [7975](#)  
     `gnote` . . . . . 1, [8739](#)  
     `hint` . . . . . 1, [8638](#)  
     `mathstructure` . . . . . 96, [6125](#)  
     `mcb` . . . . . 1, [8847](#)  
     `mmtinterface` . . . . . [7637](#)  
     `nassertion` . . . . . 1  
     `ndefinition` . . . . . 1  
     `nexample` . . . . . 1  
     `note` . . . . . 1  
     `nparagraph` . . . . . 1, 1  
     `nsproof` . . . . . 1  
     `problem` . . . . . 1  
     `sassertion` . . . . . 99  
     `scb` . . . . . [8935](#)  
     `sdefinition` . . . . . 99  
     `sexample` . . . . . 99  
     `sfragment` . . . . . 82  
     `smodule` . . . . . 86, [2518](#)  
     `solution` . . . . . 1, [8567](#)  
     `sparagraph` . . . . . 99  
     `sproblem` . . . . . 8380  
     `sproof` . . . . . 101, [7247](#)  
     `stex_annotation_env` . . . . . 141  
     `subproblem` . . . . . 8490  
     `subproof` . . . . . 7340  
     `testheading` . . . . . 74, 116  
`\eq` . . . . . 32, 42  
`\eqstep` . . . . . 7302, [7406](#)  
`\equal` . . . . . 31  
`\errmessage` . . . . . 7714, 7728, 7735, 7749  
`\escapechar` . . . . . 53, 1042  
`\everyeof` . . . . . 2167  
`\excursion` . . . . . 109, [8231](#)  
`\excursiongroup` . . . . . 109, [8248](#)  
`\excursionref` . . . . . 109, 8232, 8245  
`\exnote` (env.) . . . . . 1, [8690](#)  
`\exnote` . . . . . 8386, 8738, 8862, 8964  
 exp commands:  
     `\exp_after:wN` . . . . . 120, 450,  
         539, 542, 574, 641, 895, 2169, 2213,  
         2214, 2215, 2355, 2400, 2401, 2402,  
         2409, 2413, 2417, 2418, 2947, 2974,  
         3591, 3828, 4127, 4140, 4739, 4767,  
         4888, 5081, 5318, 5320, 5328, 5330,  
         5376, 5383, 5384, 5385, 5404, 5437,  
         5600, 5602, 5704, 5705, 5706, 5719,  
         5914, 5977, 6023, 6039, 6255, 6305,  
         6308, 6632, 6928, 6941, 7005, 7040  
`\exp_args:NNe` . . . . . 629, 634, 648, 649,  
 894, 945, 1023, 1097, 1184, 1646,  
 1795, 1808, 1822, 1838, 1841, 1844,  
 1846, 2349, 2843, 2849, 2946, 2951,  
 3385, 3413, 3419, 3787, 3796, 3956,  
 4020, 4022, 4710, 4855, 5052, 5487,  
 5633, 5635, 5644, 6216, 6266, 6353,  
 6357, 6435, 6516, 6565, 6571, 6805,  
 6841, 6934, 7219, 7426, 7758, 9225  
`\exp_args:NNe` . . . . . 54,  
 206, 218, 292, 295, 689, 1062, 1071,  
 1080, 2463, 2909, 3565, 4059, 4245,  
 4343, 5519, 5522, 5606, 5653, 5738,  
 6536, 6644, 6699, 6767, 8265, 8484  
`\exp_args:Nne` . . . . .  
     . 301, 641, 2029, 2102, 2487, 2527,  
     2584, 2785, 2863, 4201, 4352, 4391,  
     4510, 4529, 4818, 5013, 6154, 6591,  
     6632, 6700, 6905, 7120, 7270, 7448,  
     7647, 8415, 8512, 8851, 8951, 9224  
`\exp_args:NNNo` . . . . .  
     840, 1160, 3575, 4591, 4593, 4827, 5022  
`\exp_args:NNno` . . . . . 105, 361, 840  
`\exp_args:Nnno` . . . . . 446, 1722, 1730  
`\exp_args:NNNx` . . . . . 1160, 1711  
`\exp_args:NNnx` . . . . .  
     . 845, 3092, 4827, 5022, 5498, 5504  
`\exp_args:NNo` . . . . . 38,  
 48, 66, 69, 89, 93, 118, 260, 268,  
 813, 845, 849, 854, 859, 1011, 1163,  
 1653, 1991, 2001, 3225, 3251, 3572,  
 3578, 3582, 4293, 4306, 4649, 5732  
`\exp_args:Nno` . . . . . 346, 348, 361,  
 473, 2430, 2979, 2982, 3823, 4059,  
 4883, 5076, 5182, 5187, 5271, 6062,  
 6525, 7068, 8402, 8563, 9157, 9197  
`\exp_args:NNx` . . . 55, 919, 993, 1893,  
 1941, 2131, 2150, 6658, 8339, 8342,  
 8345, 8348, 9127, 9130, 9133, 9136  
`\exp_args:Nnx` . . . 341, 1312, 1588,  
 1655, 1893, 1941, 2818, 3033, 3038,

|                              |                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              | <b>F</b>                                                                                                                                                                                                                                                                                                                                                                         |
| \fbox . . . . .              | 9081, 9091                                                                                                                                                                                                                                                                                                                                                                       |
| \fi . . . . .                | 393, 408, 1888, 1903,<br>1911, 1967, 2215, 5224, 5258, 5929,<br>6033, 6095, 6997, 7570, 7953, 8021,<br>8024, 8123, 8273, 8497, 8604, 8618,<br>8930, 9008, 9096, 9309, 9319, 9320                                                                                                                                                                                                 |
| fibboxed . . . . .           | 105                                                                                                                                                                                                                                                                                                                                                                              |
| file commands:               |                                                                                                                                                                                                                                                                                                                                                                                  |
| \file_if_exist:nTF . . . . . | 729, 746, 1147                                                                                                                                                                                                                                                                                                                                                                   |
| \filepath . . . . .          | 138, 139                                                                                                                                                                                                                                                                                                                                                                         |
| \fillinsol . . . . .         | 114, 8384, 9035                                                                                                                                                                                                                                                                                                                                                                  |
| \first . . . . .             | 137                                                                                                                                                                                                                                                                                                                                                                              |
| \fn . . . . .                | 54                                                                                                                                                                                                                                                                                                                                                                               |
| \foo . . . . .               | 15, 54, 88, 141, 143                                                                                                                                                                                                                                                                                                                                                             |
| \fooname . . . . .           | 15                                                                                                                                                                                                                                                                                                                                                                               |
| \footnote . . . . .          | 8930, 9008, 9083                                                                                                                                                                                                                                                                                                                                                                 |
| \footnotesize . . . . .      | 9226                                                                                                                                                                                                                                                                                                                                                                             |
| \foral . . . . .             | 46, 51                                                                                                                                                                                                                                                                                                                                                                           |
| \forallall . . . . .         | 46, 7557, 7563, 7574                                                                                                                                                                                                                                                                                                                                                             |
| frame (env.) . . . . .       | 1, 7975                                                                                                                                                                                                                                                                                                                                                                          |
| \frameimage . . . . .        | 108, 8158                                                                                                                                                                                                                                                                                                                                                                        |
| frameimages . . . . .        | 105                                                                                                                                                                                                                                                                                                                                                                              |
| \frametitle . . . . .        | 8075                                                                                                                                                                                                                                                                                                                                                                             |
| \frontmatter . . . . .       | 1446, 1447, 1448                                                                                                                                                                                                                                                                                                                                                                 |
| \fun . . . . .               | 45                                                                                                                                                                                                                                                                                                                                                                               |
| \funspace . . . . .          | 37                                                                                                                                                                                                                                                                                                                                                                               |
|                              | <b>G</b>                                                                                                                                                                                                                                                                                                                                                                         |
| \gdef . . . . .              | 8231                                                                                                                                                                                                                                                                                                                                                                             |
| \given . . . . .             | 73, 116                                                                                                                                                                                                                                                                                                                                                                          |
| \global . . . . .            | 1239, 1243, 1260, 1265, 2086,<br>7603, 7604, 8633, 8636, 9164, 9166                                                                                                                                                                                                                                                                                                              |
| \gnote (env.) . . . . .      | 1, 8739                                                                                                                                                                                                                                                                                                                                                                          |
| \gnote . . . . .             | 8387, 8791, 8863, 8965                                                                                                                                                                                                                                                                                                                                                           |
| \gnotes . . . . .            | 73, 110, 115                                                                                                                                                                                                                                                                                                                                                                     |
| group commands:              |                                                                                                                                                                                                                                                                                                                                                                                  |
| \group_begin: . . . . .      | 43, 52, 730, 1041,<br>1689, 2178, 2437, 2474, 2793, 2813,<br>2837, 3082, 3306, 3347, 3695, 3749,<br>3914, 3916, 3918, 3920, 3980, 4186,<br>4504, 4550, 4812, 4967, 5007, 5136,<br>5140, 5238, 5391, 5547, 5555, 5629,<br>5650, 5698, 5714, 5917, 5954, 5962,<br>5973, 5995, 6006, 6019, 6556, 6573,<br>6929, 6998, 7047, 7100, 7314, 7381,<br>7403, 7466, 7487, 7515, 8560, 9194 |
| \group_end: . . . . .        | 47, 55, 736,<br>1045, 1736, 2201, 2455, 2488, 2802,<br>2818, 2887, 3087, 3312, 3353, 3697,<br>3757, 3914, 3916, 3918, 3920, 3985,<br>4005, 4189, 4515, 4553, 4823, 4979,<br>5018, 5107, 5125, 5137, 5141, 5182,<br>5315, 5326, 5343, 5354, 5376, 5404,                                                                                                                           |

|                                                                                                                   |                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| \group_insert_after:N                                                                                             | 351, 367                                                                                                                                  |
| <b>H</b>                                                                                                          |                                                                                                                                           |
| \halign                                                                                                           | 9084                                                                                                                                      |
| \have                                                                                                             | 7406                                                                                                                                      |
| \hbox                                                                                                             | 1251, 3078, 3381, 3814, 3968,<br>3997, 4368, 4855, 5067, 6227, 6811,<br>7235, 7259, 8219, 8222, 8906, 9066                                |
| hbox commands:                                                                                                    |                                                                                                                                           |
| \hbox_set:Nn                                                                                                      | 3694, 4229                                                                                                                                |
| \hbox_unpack:N                                                                                                    | 3996, 4003, 8840, 8845                                                                                                                    |
| \HCode                                                                                                            | 403                                                                                                                                       |
| \hfil                                                                                                             | 7241, 9084                                                                                                                                |
| \hfill                                                                                                            | 7241                                                                                                                                      |
| hint (env.)                                                                                                       | 1, 8638                                                                                                                                   |
| \hint                                                                                                             | 8385, 8689, 8861, 8963                                                                                                                    |
| hints                                                                                                             | 73, 110, 115                                                                                                                              |
| \hline                                                                                                            | 9224, 9226, 9227, 9228, 9229                                                                                                              |
| \href                                                                                                             | 1833                                                                                                                                      |
| \hrule                                                                                                            | 7235, 8906                                                                                                                                |
| \hspace                                                                                                           | 9094                                                                                                                                      |
| \HTML                                                                                                             | 15, 26                                                                                                                                    |
| \huge                                                                                                             | 9092                                                                                                                                      |
| hwexam commands:                                                                                                  |                                                                                                                                           |
| \l_hwexam_includeassignment_-<br>keys_tl                                                                          | 9155, 9156, 9158, 9195                                                                                                                    |
| \hwexam_kw_*                                                                                                      | 9122                                                                                                                                      |
| \c_hwexam_multiple_bool                                                                                           | 9115                                                                                                                                      |
| \hwexamheader                                                                                                     | 9231, 9270                                                                                                                                |
| \hwexamminutes                                                                                                    | 9233                                                                                                                                      |
| \hyperlink                                                                                                        | 1827, 1828                                                                                                                                |
| \hypertarget                                                                                                      | 1807, 1808                                                                                                                                |
| <b>I</b>                                                                                                          |                                                                                                                                           |
| \id                                                                                                               | 123                                                                                                                                       |
| \if                                                                                                               | 2213, 5912                                                                                                                                |
| \IfBooleanTF                                                                                                      | 3741, 4180,<br>4501, 5475, 7352, 7459, 7482, 8159                                                                                         |
| \ifcsname                                                                                                         | 390, 8273                                                                                                                                 |
| \ifdefempty                                                                                                       | 8264                                                                                                                                      |
| \iffalse                                                                                                          | 7556                                                                                                                                      |
| \IfFileExists                                                                                                     | 6, 20, 1716, 1902,<br>1910, 1990, 2000, 3254, 3259, 3279                                                                                  |
| \IfInputref                                                                                                       | 28, 83, 1960, 8262                                                                                                                        |
| \ifinputref                                                                                                       | 28, 83, 1878, 1967                                                                                                                        |
| \ifintest                                                                                                         | 8326                                                                                                                                      |
| \ifmmode                                                                                                          | 6095                                                                                                                                      |
| \ifnotes                                                                                                          | 106, 7839, 7971                                                                                                                           |
| \ifSGvar                                                                                                          | 109, 8281                                                                                                                                 |
| \ifsolutions                                                                                                      | 111, 8318, 8598, 8611, 8930, 9008, 9080                                                                                                   |
| \ifstexhtml                                                                                                       | 28, 81, 140, 390, 8495                                                                                                                    |
| \ifvmode                                                                                                          | 1903, 1911, 5224, 5258, 6033, 6997                                                                                                        |
| \ifx                                                                                                              | 2085, 7950, 8019, 8022, 9302, 9303, 9311                                                                                                  |
| \ignorespaces                                                                                                     | 4823, 5018, 8125                                                                                                                          |
| \image                                                                                                            | 117                                                                                                                                       |
| \imply                                                                                                            | 51                                                                                                                                        |
| \importmodule                                                                                                     | 38, 49, 55, 93–95,<br>130, 133, 135, 136, 143, 212, 3005, 3284                                                                            |
| \includeassignment                                                                                                | 9193                                                                                                                                      |
| \includegraphics                                                                                                  | 83, 2047, 9295                                                                                                                            |
| \includeproblem                                                                                                   | 72, 114, 8552                                                                                                                             |
| \indent                                                                                                           | 5224, 5258, 6033, 6997                                                                                                                    |
| \infpref                                                                                                          | 90, 91, 4293, 5884, 5887, 5920, 7527                                                                                                      |
| \inline*                                                                                                          | 100                                                                                                                                       |
| \inlineass                                                                                                        | 50, 55, 58, 100                                                                                                                           |
| \inlinedef                                                                                                        | 50, 100                                                                                                                                   |
| \inlineex                                                                                                         | 100                                                                                                                                       |
| \input                                                                                                            | 27, 80, 135, 40, 412,<br>1026, 1915, 2013, 8336, 8340, 8343,<br>8346, 8349, 9124, 9128, 9131, 9134,<br>9137, 9231, 9304, 9307, 9313, 9317 |
| \inputassignment                                                                                                  | 74, 116                                                                                                                                   |
| \inputref                                                                                                         | 27, 28, 82, 83, 102, 1878,<br>8177, 8178, 8240, 8265, 8563, 9197                                                                          |
| \inputref*                                                                                                        | 107, 8177                                                                                                                                 |
| \inputreffalse                                                                                                    | 1878, 1887                                                                                                                                |
| \inputreftrue                                                                                                     | 1885, 1922                                                                                                                                |
| \insertframenumber                                                                                                | 8225, 8227                                                                                                                                |
| \insertshortauthor                                                                                                | 8218                                                                                                                                      |
| \insertshortdate                                                                                                  | 8229                                                                                                                                      |
| \insertshorttitle                                                                                                 | 8221                                                                                                                                      |
| \inset                                                                                                            | 45                                                                                                                                        |
| int commands:                                                                                                     |                                                                                                                                           |
| \int_case:nn                                                                                                      | 1364                                                                                                                                      |
| \int_case:nnTF                                                                                                    | 1322, 1391, 1424                                                                                                                          |
| \int_compare:nNnTF                                                                                                |                                                                                                                                           |
| 343, 2027, 2101, 3566, 4218, 4253,<br>4763, 5345, 5632, 5643, 5692, 5898,<br>6388, 6398, 6490, 6873, 7181, 9211   |                                                                                                                                           |
| \int_compare_p:nNn                                                                                                |                                                                                                                                           |
| 7151, 7163, 7175, 7192, 7204                                                                                      |                                                                                                                                           |
| \int_decr:N                                                                                                       | 7182, 7210                                                                                                                                |
| \int_eval:n                                                                                                       |                                                                                                                                           |
| 360, 362, 364, 366, 368, 5501, 5507,<br>5897, 6187, 6194, 6321, 8524, 8527,<br>9164, 9207, 9208, 9225, 9258, 9265 |                                                                                                                                           |
| \int_gincr:N                                                                                                      | 1505, 5486,<br>8522, 8580, 8644, 8693, 8744, 9205                                                                                         |
| \int_gset:Nn                                                                                                      | 5450                                                                                                                                      |
| \int_gzero:N                                                                                                      | 5435, 8439                                                                                                                                |
| \int_incr:N                                                                                                       | 1314, 1324, 1328,<br>1352, 1366, 1369, 1372, 1393, 1397,                                                                                  |

|                                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \int_new:N . . . . .             | 1428, 1434, 3890, 3891, 3892, 3893, 4679, 4762, 5598, 7156, 7168, 7179, 7196, 7208, 7624, 7898, 8805, 9212                                                                                                             | iow commands:                                                                                                                                                                                                |
| \int_step_function:nN . . . . .  | 5164, 5364                                                                                                                                                                                                             | \iow_close:N . . . . . 742, 752, 1493                                                                                                                                                                        |
| \int_step_inline:nn . . . . .    | 3904, 4285, 4292, 4312, 4573                                                                                                                                                                                           | \iow_new:N . . . . . 711, 1491                                                                                                                                                                               |
| \int_use:N . . . . .             | 322, 1298, 1378, 1418, 1438, 1506, 3392, 3780, 4362, 4480, 4652, 4831, 4844, 5026, 5039, 5274, 5450, 5487, 5619, 5656, 5741, 5884, 5885, 6864, 7627, 7688, 7891, 8538, 8582, 8646, 8695, 8746, 8807, 9224              | \iow_now:Nn . . . . . 719, 750, 757, 1519, 1789, 7761, 8484                                                                                                                                                  |
| \int_zero:N . . . . .            | 3849, 3850, 4634, 4760, 5588, 7617                                                                                                                                                                                     | \iow_open:Nn . . . . . 741, 748, 1492                                                                                                                                                                        |
| \c_max_int . . . . .             | 5884, 5885                                                                                                                                                                                                             | \g_tmpa_iow . . . . . 748, 750, 752                                                                                                                                                                          |
| \l_tmpa_int . . . . .            | 4634, 4637, 4652, 4679, 4760, 4762, 4763, 4764, 4768, 7149, 7152, 7155, 7156, 7161, 7164, 7167, 7168, 7173, 7176, 7179, 7181, 7182, 7184, 7185, 7190, 7193, 7196, 7198, 7202, 7205, 7208, 7210, 7211, 7617, 7624, 7627 | \isassociative . . . . . 49                                                                                                                                                                                  |
| intarray commands:               |                                                                                                                                                                                                                        | \iscommutative . . . . . 49                                                                                                                                                                                  |
| \intarray_gset:Nnn . . . . .     | 7184, 7198, 7211, 7286                                                                                                                                                                                                 | \item . . . . . 7252, 8538, 8814, 8870, 8972                                                                                                                                                                 |
| \intarray_gzero:N . . . . .      | 7285                                                                                                                                                                                                                   | \itshape . . . . . 7245                                                                                                                                                                                      |
| \intarray_item:Nn . . . . .      | 7152, 7155, 7164, 7167, 7185, 7193, 7205                                                                                                                                                                               |                                                                                                                                                                                                              |
| \intarray_new:Nn . . . . .       | 7147                                                                                                                                                                                                                   | <b>J</b>                                                                                                                                                                                                     |
| \interpretmod . . . . .          | 3644, 3646, 3648                                                                                                                                                                                                       | \jobname . . . . . 77, 141, 196, 741, 746, 747, 748, 755, 760, 769, 1492, 1711                                                                                                                               |
| \interpretmodule . . . . .       | 3531, 3533                                                                                                                                                                                                             | \join . . . . . 63                                                                                                                                                                                           |
| \intestfalse . . . . .           | 8330                                                                                                                                                                                                                   |                                                                                                                                                                                                              |
| \intesttrue . . . . .            | 8328                                                                                                                                                                                                                   | <b>K</b>                                                                                                                                                                                                     |
| invokation commands:             |                                                                                                                                                                                                                        | keys commands:                                                                                                                                                                                               |
| \invokation_macro . . . . .      | 123, 125, 131                                                                                                                                                                                                          | \l_keys_choice_tl . . . . . 3712                                                                                                                                                                             |
| ior commands:                    |                                                                                                                                                                                                                        | \keys_define:nn . . . . . 27, 473, 7776, 7818, 8248, 8297, 9113, 9283                                                                                                                                        |
| \ior_close:N . . . . .           | 753, 759, 1046, 1174                                                                                                                                                                                                   | \l_keys_key_str . . . . . 4163, 4166, 6139, 6142                                                                                                                                                             |
| \ior_map_inline:Nn . . . . .     | 1159                                                                                                                                                                                                                   | \l_keys_key_tl . . . . . 4164, 6140                                                                                                                                                                          |
| \ior_new:N . . . . .             | 1154                                                                                                                                                                                                                   | \keys_set:nn . . . . . 477, 8257                                                                                                                                                                             |
| \ior_open:Nn . . . . .           | 747, 755, 1156                                                                                                                                                                                                         | keyval commands:                                                                                                                                                                                             |
| \ior_open:NnTF . . . . .         | 1040                                                                                                                                                                                                                   | \keyval_parse>NNn . . . . . 6410                                                                                                                                                                             |
| \ior_str_get:NN . . . . .        | 1043                                                                                                                                                                                                                   |                                                                                                                                                                                                              |
| \ior_str_map_inline:Nn . . . . . | 749, 756                                                                                                                                                                                                               | <b>L</b>                                                                                                                                                                                                     |
| \g_tmpa_ior . . . . .            | 747, 749, 753, 755, 756, 759, 1040, 1043, 1046                                                                                                                                                                         | \label . . . . . 84, 121, 132, 1531, 8068                                                                                                                                                                    |
|                                  |                                                                                                                                                                                                                        | \labelsep . . . . . 8027                                                                                                                                                                                     |
|                                  |                                                                                                                                                                                                                        | \labelwidth . . . . . 8028                                                                                                                                                                                   |
|                                  |                                                                                                                                                                                                                        | \lambda . . . . . 7557, 7568, 7569                                                                                                                                                                           |
|                                  |                                                                                                                                                                                                                        | \langle . . . . . 7544                                                                                                                                                                                       |
|                                  |                                                                                                                                                                                                                        | \Large . . . . . 7836, 8187, 9175                                                                                                                                                                            |
|                                  |                                                                                                                                                                                                                        | \LaTeX . . . . . 23                                                                                                                                                                                          |
|                                  |                                                                                                                                                                                                                        | \latex . . . . . 23                                                                                                                                                                                          |
|                                  |                                                                                                                                                                                                                        | \ldots . . . . . 7549, 7594, 7596, 7598                                                                                                                                                                      |
|                                  |                                                                                                                                                                                                                        | \leaders . . . . . 8189                                                                                                                                                                                      |
|                                  |                                                                                                                                                                                                                        | \leavevmode . . . . . 8083                                                                                                                                                                                   |
|                                  |                                                                                                                                                                                                                        | \left . . . . . 5914                                                                                                                                                                                         |
|                                  |                                                                                                                                                                                                                        | \leftmargin . . . . . 8029                                                                                                                                                                                   |
|                                  |                                                                                                                                                                                                                        | \let . . . . . 450, 1447, 1448, 1457, 1458, 1719, 2086, 2168, 2179, 2186, 2197, 2209, 3330, 3997, 4004, 4509, 4528, 4817, 5012, 5241, 5959, 5969, 5983, 6045, 6124, 7603, 7604, 7763, 7764, 8196, 8940, 8943 |
|                                  |                                                                                                                                                                                                                        | \libinput . . . . . 20, 80, 1970                                                                                                                                                                             |
|                                  |                                                                                                                                                                                                                        | \libusepackage . . . . . 80, 81, 2025, 7967                                                                                                                                                                  |
|                                  |                                                                                                                                                                                                                        | \libusetheme . . . . . 7966                                                                                                                                                                                  |
|                                  |                                                                                                                                                                                                                        | \libusetikzlibrary . . . . . 80, 118, 2083                                                                                                                                                                   |
|                                  |                                                                                                                                                                                                                        | \linewidth . . . . . 8621, 8628, 8682, 8687, 8731, 8736, 8784, 8789                                                                                                                                          |

|                        |                                                                                                            |                          |                                                                                                                                                                                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \LoadClass             | 15, 7804, 7806                                                                                             | \MMTinclude              | 7608                                                                                                                                                                                                                                                                 |
| \long                  | 141, 682, 8200, 8209                                                                                       | mmtinterface (env.)      | 7637                                                                                                                                                                                                                                                                 |
| \lstinputlisting       | 83, 2062                                                                                                   | \MMTrule                 | 7613                                                                                                                                                                                                                                                                 |
| \lstinputmhlisting     | 83, 2036                                                                                                   | \mname                   | 87, 96                                                                                                                                                                                                                                                               |
| mode commands:         |                                                                                                            |                          |                                                                                                                                                                                                                                                                      |
| \mode_if_math:TF       | 3381, 3997, 3998, 4005, 5283, 9065                                                                         | \mode_if_vertical:TF     | 3996, 4003                                                                                                                                                                                                                                                           |
| \MSC                   |                                                                                                            | \MSC                     | 7607                                                                                                                                                                                                                                                                 |
| msg commands:          |                                                                                                            |                          |                                                                                                                                                                                                                                                                      |
| \msg_error             |                                                                                                            | \msg_error               | 142                                                                                                                                                                                                                                                                  |
| \msg_error:nn          | 1778, 7055, 7108, 8398                                                                                     | \msg_error:nnn           |                                                                                                                                                                                                                                                                      |
| \msg_error:nnn         | 249, 287, 335, 820, 953, 960, 1972, 1975, 2110, 3217, 3273, 3444, 3474, 4010, 4712, 4954, 6199, 6676, 6703 | \msg_error:nnnn          | 446, 838, 843, 862, 897, 966, 1946, 2010, 2929, 3024, 3369, 3895, 4471, 4977, 5232, 5266, 5512, 5539, 6048, 6313, 7013                                                                                                                                               |
| \msg_fatal:nnnn        | 1128, 2033, 2104                                                                                           | \msg_fatal:nnnn          |                                                                                                                                                                                                                                                                      |
| \msg_none:nn           |                                                                                                            | \msg_redirect_module:nnn | 94                                                                                                                                                                                                                                                                   |
| \msg_redirect_name:nnn |                                                                                                            | \msg_set:nnn             | 98                                                                                                                                                                                                                                                                   |
| \msg_set:nnn           |                                                                                                            | \msg_warning:nn          | 76                                                                                                                                                                                                                                                                   |
| \msg_warning:nn        |                                                                                                            | \msg_warning:nnn         | 1061                                                                                                                                                                                                                                                                 |
| \msg_warning:nnn       |                                                                                                            | \msg_warning:nnnn        | 1688, 1730                                                                                                                                                                                                                                                           |
| \mult                  | 40, 41, 91, 92                                                                                             | \mult                    | 528, 1722                                                                                                                                                                                                                                                            |
| \multicolumn           |                                                                                                            | \multicolumn             | 9225                                                                                                                                                                                                                                                                 |
| \multiple              |                                                                                                            | \multiple                | 73, 115                                                                                                                                                                                                                                                              |
| N                      |                                                                                                            |                          |                                                                                                                                                                                                                                                                      |
| nassertion (env.)      |                                                                                                            | nassertion (env.)        | 1                                                                                                                                                                                                                                                                    |
| \Nat                   |                                                                                                            | \Nat                     | 36                                                                                                                                                                                                                                                                   |
| ndefinition (env.)     |                                                                                                            | ndefinition (env.)       | 1                                                                                                                                                                                                                                                                    |
| \neginfpref            | 41, 90, 91, 4263, 4266, 4275, 4278, 4291, 5103, 5339, 5350, 5884                                           | \newcommand              | 539, 574, 1573, 1953, 1961, 1966, 2018, 2025, 2057, 2063, 2071, 2081, 2108, 7510, 7710, 7731, 7948, 8017, 8079, 8180, 8184, 8232, 8239, 8242, 8259, 8373, 8802, 8909, 8987, 9015, 9020, 9025, 9030, 9063, 9100, 9103, 9104, 9105, 9106, 9107, 9210, 9295, 9300, 9324 |
| \newcounter            | 7807, 7808, 7809, 7810, 7932, 7944, 8372, 8390, 8391, 9153                                                 | \newcounter              |                                                                                                                                                                                                                                                                      |
| \NewDocumentCommand    |                                                                                                            | \NewDocumentCommand      | 542, 1537, 1768, 1773, 1820, 1892, 1940, 2492, 2571, 4966, 5474, 5953, 5961, 5972, 5994, 6005, 6018, 6928, 7018, 7023, 7029, 7037, 7046,                                                                                                                             |

7089, 7099, 7117, 7443, 7479, 7494,  
 7613, 7674, 7966, 8158, 8559, 9193  
`\NewDocumentEnvironment` .....  
     ..... 577, 1344, 1353, 7499, 7637  
`\newenvironment` ..... 1466,  
     1477, 7213, 7420, 8131, 8133, 9253  
`\newif` 391, 414, 1878, 7839, 8318, 8326, 9260  
`\newlabel` ..... 7763, 7764  
`\newlength` ..... 7969, 7970, 7973  
`\newline` ..... 8621, 8628,  
     8682, 8687, 8731, 8736, 8784, 8789  
`\newpage` ..... 8065, 9107, 9272  
`\newsavebox` ..... 8101  
`\example` (env.) ..... 1  
`\inputref` ..... 8178, 8180, 8184  
`\nobreak` ..... 7241  
`\noindent` ..... 1299, 1903,  
     1911, 6912, 7231, 8044, 8058, 8081,  
     8099, 8451, 8622, 8683, 8732, 8785  
`\nointerlineskip` ..... 8191  
`\notation` .. 32, 33, 41, 88, 90, 95, 126,  
     129, 135, 3004, 4170, 7525, 7528,  
     7529, 7530, 7543, 7545, 7563, 7564,  
     7568, 7572, 7574, 7575, 7580, 7583  
`note` (env.) ..... 1  
`notes` ..... 73, 105, 110, 115  
`\notesfalse` ..... 7851  
`notesslides` commands:  
     `\c_notesslides_notes_bool` ....  
         ... 7778, 7779, 7792, 7795, 7803,  
         7819, 7820, 7832, 7840, 7963, 8124,  
         8130, 8164, 8179, 8198, 8233, 8243  
     `\c_notesslides_sectocframes_bool`  
         ..... 7821, 7919  
     `\c_notesslides_topsect_str` 7822,  
         7845, 7848, 7904, 7907, 7908, 7911  
`notesslides` internal commands:  
     `\c_notesslides_class_str` ....  
         ... 7774, 7777, 7783, 7804  
     `\_notesslides_define_chapter:` ...  
         ... 7909, 7912, 7924  
     `\_notesslides_define_part:` ...  
         ... 7913, 7936  
     `\_notesslides_do_label:n` 7854, 7890  
     `\_notesslides_do_sectocframes:` ...  
         ... 7853, 7920  
     `\_notesslides_do_yes_param:Nn` ...  
         ... 7975,  
         7994, 7997, 8000, 8003, 8006, 8009  
`\c__notesslides_docopt_str` ... 7780  
`\c__notesslides_document_str` ...  
     ... 7947, 7950  
`\_notesslides_eat:` .. 8134, 8138, 8141  
`\_notesslides_excursion_args:n` .  
     ..... 8253, 8260  
`\l__notesslides_excursion_id_str`  
     ..... 8249, 8255  
`\l__notesslides_excursion_intro_-`  
     `tl` ..... 8250, 8254, 8264, 8266  
`\l__notesslides_excursion_-`  
     `mhrepos_str` ..... 8251, 8256, 8265  
`\l__notesslides_frame_allowdisplaybreaks_-`  
     `bool` ..... 7986, 7997  
`\l__notesslides_frame_allowframebreaks_-`  
     `bool` ..... 7985, 7994  
`\_notesslides_frame_box_begin:` ..  
     ..... 8037, 8048, 8071  
`\_notesslides_frame_box_end:` ...  
     ..... 8042, 8057, 8073  
`\l__notesslides_frame_fragile_-`  
     `bool` ..... 7987, 8000  
`\l__notesslides_frame_label_str` .  
     ..... 7984, 7992, 8067, 8068  
`\l__notesslides_frame_shrink_-`  
     `bool` ..... 7988, 8003  
`\l__notesslides_frame_squeeze_-`  
     `bool` ..... 7989, 8006  
`\l__notesslides_frame_t_bool` ...  
     ..... 7990, 8009  
`\_notesslides_inputref:` ...  
     ..... 8177, 8178, 8181  
`\_notesslides_notes_env:nnnn` ...  
     ..... 8129,  
     8149, 8150, 8151, 8152, 8153, 8154  
`\l__notesslides_num` .....  
     ..... 7857, 7859, 7862,  
     7864, 7867, 7868, 7871, 7872, 7875,  
     7876, 7879, 7880, 7883, 7884, 7891  
`\_notesslides_setup_itemize:` ...  
     ..... 8013, 8070  
`\l__notesslides_tmp` .... 8282, 8287  
`\g__notesslides_variables_prop` ..  
     ... 8274, 8276, 8279, 8282  
`\notesslidesfont` ..... 8040, 8055, 8196  
`\notesslidesfooter` ..... 8044, 8058, 8194  
`\notesslidestitleemph` ..... 8077, 8186  
`\notesttrue` ..... 7841  
`nparagraph` (env.) ..... 1, 1  
`nsproof` (env.) ..... 1  
`\null` ..... 7241  
`\num` ..... 143  
`\number` ..... 73, 116  
`\numberline` ..... 7889  
`\numberproblemsin` ..... 8372

## O

`\objective` ..... 7731

\OMDoc ..... 26  
\omdoc ..... 15  
\oplus ..... 7535, 7536

**P**

\PackageError ..... 8283  
\pagebreak ..... 8466, 9191  
\pagenumbering ... 1453, 1463, 1474, 1485  
\par .. 142, 454, 1297, 1301, 7214, 7241,  
    7284, 7341, 8044, 8058, 8081, 8086,  
    8094, 8099, 8106, 8116, 8451, 8462,  
    8465, 8496, 8621, 8628, 8682, 8687,  
    8731, 8736, 8784, 8789, 8828, 8831,  
    8878, 8980, 9174, 9179, 9189, 9191  
\paragraph ..... 27, 82  
\parsep ..... 7250, 8536, 8812, 8868, 8970  
\part ..... 27, 82, 7940, 7941  
\partname ..... 7859, 7937, 7938  
\parttitlename ..... 7859  
\PassOptionsToClass ..... 7782, 7783  
\PassOptionsToPackage .....  
    10, 7784, 7785, 7796,  
    7799, 7824, 7825, 7842, 8315, 9117  
\pause ..... 8079  
\PDF ..... 15, 26  
\pdfbookmark ..... 7891  
\pdfdest ..... 1533  
peek commands:  
    \peek\_charcode:NTF .....  
        ... 3599, 5097, 5110, 5173, 5175,  
        5297, 5304, 6335, 6341, 6482, 6493  
    \peek\_charcode\_remove:NTF .....  
        ... 5096, 5172, 5288, 5293,  
        5295, 5302, 5375, 5763, 6340, 6590  
\phantom ..... 9092  
\Pi ..... 7557  
\plus ..... 39–41, 44, 90–92  
\precondition ..... 7710  
\prematurestop ..... 108, 7947  
\premise ..... 52, 101, 6980, 7117, 7124  
prg commands:  
    \prg\_new\_conditional:Nnn .....  
        ... 169, 179, 326,  
        376, 628, 633, 2140, 2550, 2554,  
        3672, 3683, 3687, 5518, 5631, 5642  
    \prg\_new\_protected\_conditional:Nnn ..... 776  
    \prg\_return\_false: .. 170, 174, 180,  
        182, 328, 378, 631, 636, 792, 796,  
        2141, 2552, 2556, 3673, 3688, 5536,  
        5540, 5637, 5638, 5639, 5645, 5646  
    \prg\_return\_true: ..... 172, 174,  
        182, 328, 378, 631, 636, 796, 2141,  
        2552, 2556, 3684, 5534, 5637, 5645  
\printexcursion ..... 109  
\printexcursions .. 8231, 8240, 8261, 8269  
problem (env.) ..... 1  
problem commands:  
    \g\_problem\_id\_counter .....  
        ... 8567, 8580, 8582,  
        8644, 8646, 8693, 8695, 8744, 8746  
    \l\_problem\_inputproblem\_keys\_tl .....  
        ... 8400, 8401, 8403, 8561  
    \problem\_mcc\_box\_tl .....  
        ... 8850, 8870, 8880, 8904  
    \problem\_scc\_box\_tl .....  
        ... 8936, 8958, 8972, 8982  
problem@kw@points commands:  
    \problem@kw@points: ..... 8829  
\problemheader ..... 8451, 8469  
problems internal commands:  
    \\_\_problems\_activate\_macros: ...  
        ... 8380, 8443, 8498  
    \l\_\_problems\_anscls\_int .....  
        ... 8739, 8805, 8807  
    \c\_\_problems\_boxed\_bool ..... 8311  
    \\_\_problems\_do\_yes\_param:Nn .. 8884  
    \\_\_problems\_exnote\_start:n .....  
        ... 8690, 8708, 8711  
    \l\_\_problems\_fillin\_solution\_tl .....  
        ... 9036, 9057, 9077, 9082, 9086  
    \\_\_problems\_fillinsol:nn .....  
        ... 9066, 9068, 9072  
    \\_\_problems\_gnote\_start:n .....  
        ... 8741, 8760, 8763  
    \c\_\_problems\_gnotes\_bool .....  
        ... 8301, 8762, 8774  
    \l\_\_problems\_has\_min\_bool .....  
        ... 8394, 8437, 8438, 8526, 8545  
    \l\_\_problems\_has\_pts\_bool .....  
        ... 8393, 8434, 8435, 8523, 8540  
    \\_\_problems\_hint\_start:n .....  
        ... 8641, 8659, 8662  
    \c\_\_problems\_hints\_bool .....  
        ... 8303, 8661, 8672  
    \l\_\_problems\_in\_problem\_bool ...  
        ... 8392, 8395, 8397, 8430, 8494, 8499  
    \\_\_problems\_maybe\_inline:n .....  
        ... 8938, 8950, 8978  
    \\_\_problems\_mccline:n 8865, 8880, 8922  
    \c\_\_problems\_min\_bool .....  
        ... 8309, 8457, 8544, 8843  
    \l\_\_problems\_min\_tl ... 8432, 8436,  
        8446, 8458, 8459, 8485, 8501, 8527  
    \c\_\_problems\_notes\_bool .....  
        ... 8299, 8710, 8721  
    \\_\_problems\_oldpar: ..... 8940, 8943

```

\__problems_parse_fillin_-  

    arg:nnnn ... 9040, 9041, 9042, 9045  

\l__problems_path_seq .....  

    .. 8412, 8413, 8504, 8505, 8509, 8510  

\l__problems_prob_imports_tl . 8369  

\l__problems_prob_refnum_int . 8370  

\c__problems_pts_bool .....  

    ..... 8307, 8452, 8539, 8838  

\l__problems_pts_tl .. 8431, 8433,  

    8445, 8453, 8454, 8485, 8500, 8524  

\__problems_record_problem: ....  

    ..... 8447, 8483  

\__problems_sccline:n 8967, 8982, 9000  

\__problems_solution_start:n ...  

    ..... 8577, 8596, 8599  

\c__problems_solutions_bool ...  

    ..... 8305, 8321  

\g__problems_subproblem_int ...  

    ..... 8439, 8490, 8522, 8538  

\c__problems_test_bool .....  

    .. 8313, 8327, 8466, 9101, 9103, 9107  

\ProcessKeysOptions .....  

    .... 39, 7788, 7828, 8320, 9120, 9289  

\ProcessOptions ..... 11  

\prod ..... 7564, 7575  

\prop ..... 42  

prop commands:  

    \prop_clear:N . 1157, 2891, 2892, 6681  

    \prop_const_from_keyval:Nn . 206, 218  

    \prop_gclear:N .....  

        ..... 2323, 2324, 2325, 2435, 5433  

    \prop_get:NnN ..... 1203  

    \prop_get:NnNTF .....  

        .. 234, 284, 911, 912, 1794, 1898,  

        1899, 1974, 2945, 2972, 5519, 8282  

    \prop_gput:Nnn .....  

        2463, 2584, 2596, 2598, 2631, 5462,  

        5498, 5504, 5522, 7085, 8276, 9206  

    \prop_gset_eq:NN .... 1175, 1217, 1220  

    \prop_gset_from_keyval:Nn .....  

        ..... 1178, 1214, 2428, 2430, 5446  

    \prop_if_exist:NTF .....  

        .... 909, 1096, 1113, 1231, 1598,  

        1691, 1793, 1858, 1971, 3097, 5445  

    \prop_if_in:NnTF .....  

        .... 268, 813, 6641, 6698, 6728, 6742  

    \prop_item:Nn 926, 1097, 1209, 1615,  

        1693, 1704, 1860, 1863, 1872, 1883,  

        1886, 1910, 1915, 1926, 1979, 2040,  

        2044, 2055, 2059, 2069, 2074, 2864,  

        3034, 3039, 3100, 3125, 3136, 3137,  

        3151, 3152, 6633, 6673, 6701, 8279  

    \prop_map_break: ..... 2624  

\prop_map_break:n 2643, 2679, 2682,  

    2760, 3058, 4079, 4942, 4945, 7072  

\prop_map_function:NN .... 2421, 2961  

\prop_map_inline:Nn .. 2431, 2608,  

    2641, 2685, 2703, 2706, 2728, 2731,  

    2769, 2922, 2941, 2988, 3016, 3020,  

    4095, 4935, 6295, 6521, 7069, 9218  

\prop_new:N ..... 8274, 9200  

\prop_put:Nnn 1158, 1166, 1167, 1168,  

    1169, 1170, 2909, 2913, 3388, 3422,  

    4828, 5023, 6707, 6730, 6744, 9213  

\prop_remove:N ..... 3484  

\prop_set_eq:NN 1085, 1201, 1205, 1607  

\prop_to_keyval:N .....  

    ... 1176, 1179, 1215, 2400, 2401,  

    2402, 2409, 2413, 2936, 2939, 5447  

\protect ..... 7889, 8285  

\protected ..... 123,  

    141, 452, 666, 667, 1265, 5970, 5984  

\providecommand .. 2042, 2049, 7955, 9294  

\ProvidesExplClass ..... 5, 7771  

\ProvidesExplPackage .....  

    ..... 19, 7815, 8294, 9110, 9280  

\pts ..... 8837  

pts ..... 73, 110, 115

```

## Q

```

\quad ..... 7868, 7872, 7876,  

    7880, 7884, 9064, 9070, 9182, 9185  

quark commands:  

    \quark_new:N ..... 2164, 2447  

quark internal commands:  

    \q__stex_smemode_break .....  

        ..... 2164, 2167, 2227

```

## R

```

\rangle ..... 7544  

\realization ..... 3555, 3557  

\realize ..... 63, 64, 3665, 3667, 3669  

\ref ..... 84, 1638  

\refstepcounter ..... 8408, 9166  

\relax ..... 453, 734, 735, 1448, 1458,  

    1533, 1689, 2085, 2171, 2209, 2213,  

    2438, 2439, 3997, 4004, 8165, 8167  

\renamedecl .... 64, 129, 3010, 3408, 3492  

\renewcommand ..... 7964, 8065, 8075  

\renewenvironment .....  

    .. 8018, 8062, 8080, 8102, 8125, 8127  

repo commands:  

    \repo_prop ..... 138, 139  

\reqpts ..... 74, 116  

\requiremodule ..... 93, 94, 3006, 3357  

\RequirePackage .....  

    ... 4, 13, 18, 23, 24, 242, 244, 292,

```

295, 5910, 7772, 7790, 7812, 7816,  
 7830, 7843, 7844, 8156, 8295, 8333,  
 9111, 9121, 9281, 9292, 9297, 9298  
`\resizebox` ..... 9223, 9306, 9312, 9316  
`\rhd` ..... 8020, 8023  
`\right` ..... 5915  
`\Rrightarrow` ..... 7476  
`\rightmargin` . 7251, 8537, 8813, 8869, 8971  
`\rule` ..... 8621, 8628,  
     8682, 8687, 8731, 8736, 8784, 8789  
 rustex commands:  
     `\rustex_direct_HTML:N` .....  
         ..... 8087, 8095, 8107, 8117  
     `\rustex_if:TF` ..... 8085, 8086,  
         8093, 8094, 8105, 8106, 8115, 8116  
`\rustexBREAK` ..... 2137

**S**

`sassertion (env.)` ..... 99  
`scb (env.)` ..... 8935  
`\scb` ..... 8383, 8859, 8961, 8985  
`\scc` ..... 8966, 8986  
`\scriptsize` ..... 8017  
`\scriptstyle` ..... 8023  
`sdefinition (env.)` ..... 99  
`\second` ..... 137  
`\section` ..... 26, 27, 82  
`\sectontitleemph` ..... 7835, 7894  
`sectocframes` ..... 105  
`\selectlanguage` ..... 140, 237  
 seq commands:  
     `\seq_clear:N` ..... 104, 143,  
         278, 1977, 2181, 2692, 2717, 2742,  
         2847, 2893, 4216, 4572, 4733, 5587,  
         5651, 5716, 5725, 6161, 6212, 6262,  
         6420, 6682, 6683, 6803, 6839, 7614  
     `\seq_count:N` .. 2027, 2101, 3566, 6873  
     `\seq_gclear:N` ..... 2175, 2176, 5434  
     `\seq_gclear_new:N` ..... 979, 980  
     `\seq_get:NN` ..... 1010  
     `\seq_get_right:NN` 254, 8413, 8505, 8510  
     `\seq_gpop:NN` ..... 1005  
     `\seq_gpush:Nn` ..... 993  
     `\seq_gput_left:Nn` ..... 1542, 1547  
     `\seq_gput_right:Nn` .....  
         ..... 1545, 2131, 2150, 5463  
     `\seq_gset_eq:NN` .....  
         992, 1008, 1012, 2874, 2875, 2876, 7753  
     `\seq_gset_split:Nnn` ..... 5452  
     `\seq_if_empty:NTF` 141, 159, 170, 180,  
         265, 796, 842, 848, 853, 858, 1004,  
         1007, 1188, 1197, 1945, 2009, 4056,  
         6888, 7221, 7272, 7428, 7622, 7752  
     `\seq_if_empty_p:N` .... 783, 784, 1988

`\seq_if_exist:NTF` ..... 1587, 5451  
`\seq_if_in:NnTF` . 1540, 1541, 1653,  
     1655, 1991, 2001, 2275, 2284, 2661,  
     2701, 2726, 2767, 6533, 6644, 6767  
`\seq_item:Nn` ..... 171,  
     181, 2028, 2102, 4329, 4335, 5520, 6874  
`\seq_map_break:` .... 1589, 1593, 2679  
`\seq_map_break:n` 1593, 1982, 2682, 4079  
`\seq_map_function:NN` . 145, 149, 6432  
`\seq_map_inline:Nn` .....  
     . 359, 927, 1122, 1183, 1592, 1626,  
     1948, 1980, 2012, 2190, 2684, 4093,  
     4734, 4761, 5726, 6225, 6272, 6281,  
     6809, 6947, 6974, 7292, 7347, 7623  
`\seq_new:N` .....  
     . 320, 1058, 1500, 1546, 2121, 2144, 2294  
`\seq_pop>NN` ..... 142  
`\seq_pop_left:NN` ..... 125,  
     264, 789, 790, 841, 846, 852, 857,  
     928, 930, 1995, 3568, 3570, 3577, 3581  
`\seq_pop_left:NNTF` .....  
     . 1162, 4303, 4305, 4313  
`\seq_pop_right:NN` ..... 160, 256,  
     266, 801, 803, 808, 810, 812, 1191,  
     2369, 3093, 4055, 4592, 4594, 6423  
`\seq_push:Nn` ..... 148, 1063, 1076  
`\seq_put_left:Nn` .....  
     . 126, 146, 2702, 2727, 6729, 6743  
`\seq_put_right:Nn` .....  
     . 163, 196, 260, 285, 323,  
     804, 814, 817, 1992, 1996, 2002,  
     2328, 2358, 2371, 2663, 2768, 2850,  
     2995, 4286, 4293, 4314, 4316, 4574,  
     4736, 4738, 5618, 5654, 5690, 5697,  
     5722, 5728, 5732, 5739, 6162, 6217,  
     6221, 6267, 6699, 6806, 6842, 6856  
`\seq_reverse:N` ..... 6424  
`\seq_set_eq:NN` . 151, 195, 777, 778,  
     800, 807, 925, 1185, 4743, 4771, 5737  
`\seq_set_split:Nnn` ..... 105,  
     255, 802, 809, 840, 845, 1161, 1982,  
     1986, 2368, 3092, 3565, 3575, 4054,  
     4302, 4306, 4591, 4593, 6422, 7615  
`\seq_use:Nn` ..... 187,  
     290, 293, 296, 804, 817, 845, 1081,  
     1129, 1164, 2373, 3094, 3573, 3579,  
     3583, 4060, 4372, 4577, 4720, 4744,  
     5453, 5576, 6889, 7222, 7273, 7429  
`\l_tmpa_seq` .....  
     . 255, 256, 260, 264, 265, 266,  
     278, 285, 290, 293, 296, 1185, 1188,  
     1191, 1197, 1198, 4572, 4574, 4577,  
     4591, 4592, 4593, 4594, 4733, 4736,

4738, 4743, 4771, 6420, 6422, 6423,  
 6424, 6432, 7614, 7615, 7622, 7623  
`\seqmap` ..... 96, 5189, 5644  
`\setbox` ..... 135, 8127, 8142, 8603, 8664, 8713, 8765  
`\setcounter` ..... 9164, 9167  
`\setkeys` .... 2046, 2061, 2076, 8169, 9301  
`\setlength` ..... 7249, 7250, 7251,  
 7969, 7970, 7974, 8027, 8028, 8029,  
 8535, 8536, 8537, 8811, 8812, 8813,  
 8867, 8868, 8869, 8969, 8970, 8971  
`\setlicensing` ..... 107  
`\setmetatheory` ..... 2473  
`\setnotation` ..... 33, 91, 4446  
`\setsectionlevel` ..... 27,  
 82, 121, 1406, 7846, 7848, 7905, 7907  
`\setSGvar` ..... 109, 8275, 8285  
`\setslidelogo` ..... 107  
`\setsource` ..... 107  
`sexample` (env.) ..... 99  
`\sf` ..... 8187  
`\sffamily` ..... 8196  
`sfragment` (env.) ..... 82  
 sfragment commands:  
     `\_sfragment_do_level:nn` .....  
         ..... 1295, 1307, 1323, 1327,  
         1331, 1332, 1333, 1334, 1335, 7887  
     `\_sfragment_end:` ..... 1303, 1317, 1349  
`\skipfragment` ..... 82, 1390, 1394, 1398  
`\slideframewidth` .. 7973, 7974, 8050, 8165  
`\slideheight` ..... 7970  
`slides` ..... 105  
`\slidewidth` .....  
 .. 7969, 8053, 8083, 8165, 8167, 8171  
`\smacro` ..... 92, 93  
`\small` ..... 7245  
`\smallskip` ... 7241, 8454, 8459, 8541,  
 8546, 8621, 8682, 8731, 8784, 9179  
`smodule` (env.) ..... 86, 2518  
`\smodule` ..... 3007  
`\Sn` ..... 20, 89, 5941  
`\sn` ..... 20, 24, 89, 5941  
`\Sns` ..... 20, 89, 5941  
`\sns` ..... 20, 89, 5941  
`\solution` ..... 73  
`solution` (env.) ..... 1, 8567  
`\solution` ..... 8381, 8631, 8860, 8962  
`solutions` ..... 73, 110, 115  
`\solutionsfalse` ..... 8324, 8636  
`\solutionstrue` ..... 8322, 8633  
`\source` ..... 137  
`sparagraph` (env.) ..... 99  
`\spfblock` ..... 7304, 7504  
`\spfjust` ..... 7305, 7510, 7513  
`\spfsketch` ..... 7213  
`\spfsketchenvautorefname` ..... 7231  
`\spfstep` ..... 7299, 7406  
`\spfstepautorefname` ..... 7340, 7421  
`\spfstepenvautorefname` ..... 7421  
`sproblem` (env.) ..... 8380  
`\sproblemautorefname` ..... 8475  
`sproof` (env.) ..... 101, 7247  
`\sproofautorefname` ..... 7319  
`\sproofend` ..... 7233, 7321, 7338  
`\square` ..... 7234, 8905  
`\sr` ..... 20, 24, 88, 5941  
`\sref` ..... 84, 85, 99, 1551, 8235  
`\sreflabel` ..... 84, 86, 121, 1500  
`\srefsetin` ..... 84, 1573  
`\srefsym` ..... 89, 1820  
`\srefsymuri` ..... 89, 1851  
`\startsolutions` ..... 111, 8632  
`\stepcounter` ..... 1387, 7888, 8064  
`\sTeX` ..... 14, 15, 81  
`\stex` ..... 15, 24, 81  
 stex commands:  
     `\l_stex_ab_args_seq` .....  
         4720, 4734, 4743, 4744, 4761, 4771,  
         5576, 5587, 5618, 5651, 5654, 5690,  
         5697, 5716, 5722, 5726, 5737, 5739  
`\stex_activate_module:n` .....  
 .. 122, 2339, 2388, 2660,  
 2660, 2672, 2842, 2880, 3199, 3297,  
 3340, 3540, 6233, 6289, 6814, 7660  
`\stex_add_definiens:nn` .....  
 .. 3012, 7059, 7067, 7294, 7349  
`\stex_add_definiens_inner:nnnnnnn`  
 ..... 7072, 7083  
`\stex_add_module_notation:nnnn` 122  
`\l_stex_all_modules_seq` .....  
 .. 122, 2181, 2294,  
 2328, 2358, 2661, 2663, 2684, 4093  
`\l_stex_allow_semantic_bool` ....  
 4973, 5115, 5193, 5194, 5197, 5223,  
 5257, 5423, 5432, 5548, 5559, 5657,  
 5742, 5800, 5811, 5844, 5873, 6034,  
 6115, 6526, 6557, 6574, 6619, 7000  
`\stex_annotation:nn` .....  
 .. 433, 436, 1278, 1287, 1299,  
 1742, 1962, 1963, 3381, 3817, 3820,  
 3827, 3832, 3834, 4375, 4381, 4390,  
 4401, 4412, 4413, 4416, 4417, 4421,  
 4877, 4880, 4887, 4891, 4894, 5070,  
 5073, 5080, 5084, 5087, 5118, 5564,  
 5571, 5665, 5774, 5787, 5804, 5829,  
 5837, 5858, 5866, 6060, 6446, 6454,  
 6459, 6667, 6913, 6934, 7011, 7077,  
 7110, 7122, 7219, 7261, 7264, 7306,

7360, 7369, 7373, 7426, 7495, 7511,  
 7620, 7625, 8225, 8472, 8816, 8839,  
 8844, 8924, 8927, 9002, 9005, 9075  
`\stex_annotationnn` ..... 140, 141  
`\stex_annotation_force_break:n` ...  
       ..... 422, 423, 429,  
           1300, 3380, 3814, 3832, 4874, 4892,  
           5068, 5085, 5572, 5670, 5779, 5792,  
           5809, 5834, 5842, 5863, 5871, 6458,  
           6913, 6935, 7078, 7111, 7621, 9075  
`\stex_annotation_invisible:n` .....  
       ..... 141, 424,  
           426, 1357, 1379, 2535, 2792, 2836,  
           5220, 5476, 7259, 7618, 7655, 8517  
`\stex_annotation_invisible:nn` 1251,  
   1418, 1426, 1432, 1438, 1903, 1911,  
   3078, 3302, 3343, 3379, 3413, 3454,  
   3464, 3796, 4368, 4855, 5052, 5810,  
   5843, 5872, 6227, 6283, 6811, 7093,  
   7609, 7656, 7724, 7745, 8423, 9049  
`\stex_annotation_invisible:nnn` .. 141  
`\l_stex_argnames_seq` ..... 124  
`\stex_args_end:` ..... 5437,  
   5461, 5464, 5986, 5990, 6040, 7006  
`\stex_assign_do:n` .....  
       ..... 2857, 3361, 3366, 3594  
`\l_stex_assoc_args_count` .....  
       ..... 124, 3846, 3850, 3892, 3893  
`\l_stex_brackets_dones_bool` ...  
       .. 4641, 5890, 5893, 5894, 5918, 5919  
`\stex_capitalize:n` .....  
       ..... 1289, 5977, 5980, 6023, 7040  
`\stex_check_term:n` .....  
       ..... 124, 125, 2180, 3373,  
           3692, 3693, 3701, 3912, 4348, 7517  
`\c_stex_check_terms_bool` .....  
       ..... 33, 3679, 3682, 7520  
`\stex_close_module:` ... 122, 2395,  
   2395, 2544, 2799, 7601, 7668, 7670  
`\l_stex_current_archive` ..... 1082  
`\l_stex_current_archive_prop` ...  
       ..... 134, 139, 926, 936,  
           941, 1085, 1096, 1097, 1106, 1183,  
           1598, 1599, 1691, 1693, 1793, 1794,  
           1858, 1860, 1863, 1883, 1886, 1898,  
           1899, 1910, 1915, 1926, 1971, 1974,  
           1979, 2044, 2059, 2074, 3097, 3100  
`\l_stex_current_args_tl` .....  
       131, 4563, 4567, 5246, 5436, 5437, 5440  
`\l_stex_current_arity_str` .....  
       ..... 131, 4573, 5135, 5162,  
           5164, 5245, 5345, 5363, 5364, 5395  
`\l_stex_current_doc_uri` 139, 970,  
   972, 973, 1510, 1515, 1517, 1521,  
       1587, 1589, 1628, 1645, 1646, 6892  
`\l_stex_current_domain_str` ....  
       ... 129, 2806, 2809, 2811, 2819,  
           2824, 2833, 2842, 2873, 2880, 2895,  
           2907, 2944, 2959, 3431, 3479, 3500,  
           3520, 3540, 3544, 3617, 3637, 3658  
`\g_stex_current_file` .....  
       .. 137-139, 254, 935, 937, 942, 980,  
           983, 985, 987, 988, 989, 992, 993,  
           1008, 1011, 1012, 1602, 1698, 2347,  
           3107, 3113, 3114, 3162, 3181, 3225,  
           3251, 7752, 7753, 8412, 8504, 8509  
`\l_stex_current_language_str` ...  
       ..... 139, 140,  
           231, 233, 234, 267, 272, 2529, 3212,  
           3215, 3239, 3242, 7649, 8417, 8514  
`\l_stex_current_module` ..... 123, 125  
`\l_stex_current_module_str` ....  
       .. 122, 143, 2182, 2293, 2310, 2327,  
           2328, 2343, 2356, 2357, 2358, 2359,  
           2398, 2399, 2400, 2401, 2402, 2408,  
           2410, 2414, 2419, 2421, 2528, 2538,  
           2551, 2559, 2563, 2584, 2590, 2596,  
           2598, 2608, 2615, 2620, 2629, 2631,  
           2641, 2665, 2668, 2787, 2798, 2832,  
           2843, 2947, 2952, 3386, 3420, 3447,  
           3456, 3466, 3750, 3788, 3797, 3896,  
           3957, 3966, 3975, 3981, 4520, 4539,  
           4540, 6118, 6156, 6161, 6162, 6182,  
           6274, 6296, 6857, 6880, 7068, 7069,  
           7070, 7071, 7084, 7085, 7639, 7640,  
           7643, 7648, 7667, 7669, 7682, 7696  
`\l_stex_current_ns_uri` ..... 139,  
   976, 977, 2365, 2508, 3163, 3168,  
   3169, 3182, 3187, 3188, 7521, 7602  
`\l_stex_current_redo_tl` .....  
       ..... 5227, 5240, 5251, 5254,  
           5261, 5391, 6391, 6429, 6464, 6614  
`\l_stex_current_return_tl` .....  
       131, 4004, 5247, 5313, 5325, 5346, 5396  
`\stex_current_section_level` ....  
       ..... 1274, 1280, 1289, 1315, 7899  
`\l_stex_current_symbol_str` ....  
       ..... 131, 4200, 4230,  
           4349, 4389, 4400, 4509, 4528, 4591,  
           4817, 4969, 4971, 4977, 5012, 5102,  
           5105, 5120, 5148, 5149, 5232, 5244,  
           5266, 5282, 5311, 5312, 5315, 5321,  
           5337, 5338, 5341, 5430, 5512, 5539,  
           5556, 5658, 5659, 5702, 5743, 5744,  
           5776, 5789, 5806, 5831, 5839, 5860,  
           5868, 5921, 5926, 5999, 6012, 6025,  
           6035, 6048, 6056, 6060, 6062, 6117,  
           6472, 6473, 7001, 7009, 7011, 7013

```

\l_stex_current_term_tl 4974, 5195,
5249, 5557, 5660, 5661, 5662, 5745,
5773, 5786, 5801, 5812, 5828, 5845,
5857, 5874, 6043, 6465, 6471, 6661
\stex_current_this: . 5241, 6114, 6124
\l_stex_current_this_tl .....
..... 6116, 6120, 6134, 6137
\l_stex_current_type_tl .....
..... 131, 5117, 5248, 6333,
6388, 6395, 6398, 6402, 6717, 6766
\stex_deactivate_macro:Nn .....
..... 142, 443,
443, 2580, 3001, 3002, 3003, 3004,
3005, 3006, 3007, 3288, 3357, 3491,
3492, 3493, 3510, 3531, 3555, 3623,
3644, 3665, 3761, 3989, 4207, 4558,
6169, 6243, 6305, 7021, 7027, 7035,
7043, 7064, 7097, 7115, 7124, 7399,
7472, 7492, 7497, 7504, 7513, 7611,
7635, 7708, 8631, 8689, 8738, 8791,
8836, 8858, 8859, 8860, 8861, 8862,
8863, 8883, 8934, 8960, 8961, 8962,
8963, 8964, 8965, 8985, 9012, 9099
\stex_debug:nn .....
..... 120, 65,
65, 95, 99, 128, 135, 198, 272, 290,
341, 608, 733, 868, 935, 938, 973,
977, 1019, 1027, 1047, 1052, 1065,
1068, 1075, 1081, 1084, 1114, 1146,
1176, 1208, 1232, 1244, 1259, 1516,
1581, 1586, 1625, 1629, 1645, 1648,
1654, 1656, 1660, 1663, 1669, 1671,
1674, 1683, 1684, 1710, 1757, 1759,
1761, 1856, 1862, 1867, 1874, 2233,
2235, 2251, 2253, 2257, 2274, 2276,
2283, 2285, 2307, 2320, 2333, 2335,
2357, 2397, 2475, 2477, 2486, 2590,
2615, 2623, 2627, 2662, 2935, 2973,
2978, 2981, 3091, 3096, 3098, 3106,
3113, 3118, 3122, 3161, 3180, 3197,
3203, 3210, 3226, 3237, 3252, 3270,
3278, 3280, 3367, 3374, 3411, 3432,
3448, 3452, 3462, 3480, 3482, 3564,
3567, 3571, 3913, 3915, 3917, 3919,
4036, 4053, 4092, 4103, 4105, 4258,
4290, 4301, 4351, 4539, 4543, 4619,
4628, 4633, 4654, 4803, 4922, 4962,
5181, 5186, 5282, 5287, 5292, 5311,
5314, 5317, 5337, 5389, 5398, 5422,
5430, 5546, 5897, 6180, 6252, 6332,
6872, 6875, 6877, 6881, 7057, 7070,
7084, 7119, 7293, 7348, 7447, 7644
\c_stex_debug_clist .....
.... 28, 38, 66, 69, 83, 87, 89, 93, 97
\c_stex_default_metatheory 2179, 7604
\l_stex_default_notation .....
.... 4571, 4576, 4579, 4581, 4582,
4599, 5152, 5326, 5331, 5352, 5686
\stex_do_default_notation: .....
..... 4561, 5151, 5324, 5685
\stex_do_default_notation_op: .....
..... 4561, 4562, 4597, 5349
\stex_do_deprecation:n .....
.... 120, 526, 526, 2313, 3737, 3841
\stex_do_for_list: .....
... 3011, 6838, 6853, 7217, 7288, 7343
\stex_do_id: .....
.... 132, 531,
531, 1347, 6917, 6932, 7218, 7297,
7357, 7366, 7375, 7455, 8442, 9170
\stex_do_up_to_module:n .....
.... 123, 2558, 2558, 2561, 2568, 3339, 6320
\l_stex_docheader_sect .....
.... 121, 1272, 1298, 1314, 1322, 1324,
1328, 1352, 1364, 1366, 1369, 1372,
1378, 1391, 1393, 1397, 1408, 1409,
1410, 1411, 1412, 1413, 1415, 1418,
1424, 1428, 1434, 1438, 7891, 7898
\stex_eat_exclamation_point: ...
..... 3792, 5315, 5326, 5762, 5764
\stex_end: .....
.... 515, 523, 2902, 2910
\stex_every_file: .....
..... 994, 997, 1002, 1014, 7756
\stex_every_module:n .....
..... 122, 2295, 2298,
2582, 3327, 3511, 3532, 3556, 3624,
3645, 3666, 3762, 3990, 4208, 4559,
6170, 6244, 6307, 6317, 7612, 7636
\g_stex_every_module_tl .....
..... 2296, 2299, 2311
\l_stex_every_symbol_tl 5196, 5201,
5203, 5204, 5210, 5211, 5214, 5252
\stex_execute_in_module:n .....
.... 122, 2312, 2386, 2566, 2566,
2570, 2601, 2633, 2879, 3296, 3539,
3974, 6160, 6220, 6232, 6271, 6288
\stex_fatal_error:n .....
.... 142
\stex_fatal_error:nnn .....
.... 142
\l_stex_feature_name_str .....
..... 129, 2839, 2843,
2872, 2952, 2958, 2978, 2979, 2981,
2982, 3025, 3370, 3386, 3420, 3424
\stex_file_in_smsmode:nn .....
... 135, 2155, 2174, 2203, 2349, 3266
\stex_file_resolve:Nn .....
.... 137, 138, 109, 115, 131, 138, 194,
204, 886, 983, 985, 988, 1069, 1071,
1601, 1618, 1708, 3114, 3208, 3235
\stex_file_set:Nn .. 137, 138, 103,
103, 108, 122, 133, 882, 926, 1011, 1062

```

```

\stex_file_split_off_ext:NN . . .
    ..... 137, 799, 799, 904, 2347, 3107
\stex_file_split_off_lang:NN . . .
    ..... 137, 799,
    806, 901, 2348, 3108, 8412, 8504, 8509
\stex_file_use:N . . .
    137, 138, 128, 135, 186, 186, 198,
    850, 855, 860, 914, 920, 932, 935,
    989, 993, 1040, 1062, 1063, 1072,
    1075, 1076, 1080, 1184, 1198, 1602,
    1603, 1698, 1709, 1711, 1866, 1989,
    1999, 2350, 3110, 3113, 3114, 3116,
    3162, 3181, 3209, 3225, 3236, 3251
\stex_filestack_pop: . . .
    ... 137, 1003, 1003, 1028, 1035, 2200
\stex_filestack_push:n . . .
    ..... 137, 981, 981, 1021, 1023, 2184
\l_stex_fors_seq . . .
    2847,
    2850, 6839, 6842, 6856, 6873, 6874,
    6888, 6889, 6947, 6974, 7221, 7222,
    7272, 7273, 7292, 7347, 7428, 7429
\stex_get_env:Nn . . .
    ..... 142, 51, 52, 60, 81, 190,
    192, 200, 202, 395, 699, 705, 1038, 3676
\stex_get_in_morphism:n . . .
    131, 2849, 3014, 3014, 3360, 3404, 3589
\_stex_get_mathstructure:n . . .
    ..... 2808, 6197, 6202
\stex_get_mathstructure:n . . .
    ..... 6196, 6196, 6214, 6263, 6802
\l_stex_get_structure_module_str
    .. 2809, 6198, 6203, 6207, 6221, 6272
\_stex_get_symbol:n . 4008, 4014, 6204
\stex_get_symbol:n . . .
    124,
    125, 131, 1821, 4007, 4007, 4172,
    4447, 4963, 5270, 5956, 5964, 5975,
    6841, 6999, 7050, 7103, 7712, 7733
\l_stex_get_symbol_args_tl . . .
    ..... 124, 126,
    235, 3063, 3393, 3781, 3903, 3905,
    3906, 4042, 4083, 4115, 4126, 4128,
    4141, 4567, 4832, 4845, 4927, 5027,
    5040, 5275, 5440, 6185, 6865, 7689
\l_stex_get_symbol_arity_int . . .
    ..... 124, 126, 235,
    3062, 3392, 3780, 3825, 3849, 3855,
    3859, 3863, 3867, 3871, 3875, 3879,
    3883, 3887, 3890, 3891, 3892, 3893,
    3904, 3962, 4034, 4041, 4082, 4114,
    4218, 4253, 4285, 4292, 4312, 4362,
    4480, 4802, 4831, 4844, 4885, 4926,
    5026, 5039, 5078, 5274, 6864, 7688
\l_stex_get_symbol_def_tl . . .
    ..... 124, 2868, 3064,

```

```

8410, 8449, 8470, 8507, 8532, 8585,
8595, 8607, 8613, 8649, 8658, 8668,
8674, 8698, 8707, 8717, 8723, 8750,
8759, 8770, 8776, 8815, 8849, 8875,
8923, 8949, 8977, 9001, 9046, 9074
\stex_if_do_html_p: ..... 140
\stex_if_file_absolute:N 137, 169, 179
\stex_if_file_absolute:NTF .....
..... 137, 168, 987, 1070
\stex_if_file_absolute_p:N . 137, 168
\stex_if_file_starts_with:NN 137, 776
\stex_if_file_starts_with:NNTF ..
..... 137, 776
\stex_if_html_backend:TF .....
..... 140, 390, 415,
422, 431, 714, 1250, 1294, 1356,
1375, 1422, 1933, 1960, 3671, 5224,
5258, 5770, 5826, 5855, 6033, 6059,
6911, 6921, 6997, 7791, 7831, 7956,
8036, 8084, 8092, 8104, 8114, 8224
\stex_if_html_backend_p: ... 140, 390
\stex_if_in_module: ..... 2550
\stex_if_in_module:TF .....
..... 122, 2302, 2550, 2566, 4478
\stex_if_in_module_p: ... 122, 2550
\stex_if_module_exists:n .... 2554
\stex_if_module_exists:nTF .....
..... 122, 2319, 2332, 2554, 3165, 3168,
3184, 3187, 3272, 6216, 6266, 6805
\stex_if_module_exists_p:n 122, 2554
\stex_if_smsmode: ..... 2140
\stex_if_smsmode:TF .....
..... 135, 532, 1242, 1518, 2138,
2205, 2338, 2537, 2545, 3081, 3224,
3250, 3305, 3346, 3498, 3505, 3518,
3526, 3542, 3550, 3615, 3635, 3656,
3748, 3979, 4185, 4549, 6904, 6920,
6933, 6946, 6973, 7076, 8529, 8531
\stex_if_smsmode_p: ..... 135, 2138
\stex_ignore_spaces_and_pars: ...
..... 142, 452, 452, 455, 2576, 2577, 8463
\l_stex_import_archive_str . 130,
2478, 2483, 2815, 3074, 3099, 3103,
3123, 3124, 3125, 3293, 3320, 3336
\stex_import_module_uri:nn .....
..... 130, 2476, 2812,
3072, 3090, 3090, 3291, 3317, 3334
\l_stex_import_name_str .....
..... 130, 2480, 2485, 2817, 3076, 3084,
3093, 3295, 3309, 3322, 3338, 3350
\l_stex_import_ns_str .... 2486,
2489, 2819, 3079, 3083, 3166, 3169,
3172, 3185, 3188, 3191, 3199, 3297,
3300, 3303, 3308, 3340, 3344, 3349
\l_stex_import_path_str .....
..... 130, 2479, 2484,
2816, 3075, 3094, 3105, 3109, 3113,
3114, 3115, 3118, 3294, 3321, 3337
\stex_import_require_module:nnn .
..... 130, 2482, 2814,
3073, 3128, 3128, 3141, 3292, 3335
\stex_import_require_module_-
safe:nnn ..... 3143, 3319
\l_stex_import_uri_str .....
..... 130, 3104, 3125, 3131,
3136, 3146, 3151, 3159, 3161, 3165,
3166, 3172, 3178, 3180, 3184, 3185,
3191, 3210, 3217, 3237, 3272, 3273
\stex_in_archive:nn 134, 1089, 1089,
1881, 1932, 1957, 2022, 2077, 2115
\stex_in_invisible_html_bool ...
..... 3815, 4875, 5768, 5769, 5799
\l_stex_in_meta_bool 2381, 2382, 2387
\l_stex_inpararray_bool ..... 5899
\stex_input_with_hooks:n .....
.. 1016, 1883, 1886, 1907, 1924, 1926
\stex_invoke_notation:w .....
..... 5187, 5304, 5305, 5310, 5347
\stex_invoke_outer_field: ... 6190
\stex_invoke_sequence: .....
..... 5031, 5044, 5095, 5095, 5636
\stex_invoke_sequence_in: ... 132
\stex_invoke_sequence_range: ... 132
\stex_invoke_structure: .....
..... 6158, 6206, 6326
\stex_invoke_symbol ..... 131
\stex_invoke_symbol: .....
..... 125, 131, 3785,
4836, 4849, 5281, 5281, 6866, 7693
\stex_invoke_symbol:nnnnnnnnN ...
..... 123, 131,
2619, 4023, 4037, 4038, 5222, 5222,
5235, 5271, 6732, 6746, 6754, 6759
\stex_invoke_symbol:nnnnnnnnN\l_-
stex_current_symbol_str ... 131
\stex_invoke_text_symbol: 3954, 4002
\stex_invoke_variable:nnnnnn . 5256
\stex_invoke_variable:nnnnnnN ...
..... 131, 4842, 5037, 5256, 5634
\stex_is_sequentialized:n .....
..... 5628, 5655, 5730, 5740
\stex_iterate_break: 123, 2675, 2678
\stex_iterate_break:n .. 123, 2675,
2681, 2758, 3453, 3463, 3478, 4111
\stex_iterate_morphisms:nn .....
..... 2741, 2741, 3431, 3447
\stex_iterate_notations:nn .....
..... 126, 2716, 2716, 2944, 6766

```

```

\stex_iterate_symbols:n .....
    ..... 123, 2674, 2674, 4104
\stex_iterate_symbols:nn .....
    123, 2691, 2691, 2907, 3481, 6174, 6717
\l_stex_key_answerclass_str .....
    ..... 8570, 8574, 8588, 8623, 8624
\l_stex_key_archive_str .....
    .. 132, 480, 483, 1565, 1583, 1597,
        1606, 1607, 1683, 1690, 1702, 1704
\l_stex_key_argnames_clist .....
    124
\l_stex_key_args_str .....
    .... 124, 3704, 3709, 3718, 3754,
        3798, 3851, 3856, 3860, 3864, 3868,
        3872, 3876, 3880, 3884, 3888, 3907,
        4524, 4857, 4988, 4989, 5054, 6823
\l_stex_key_argtypes_clist .....
    ..... 3723, 3728, 3831, 3833,
        3920, 4890, 4893, 5083, 5086, 6827
\l_stex_key_assoc_str 3707, 3712,
    3805, 3806, 4861, 4862, 5058, 5059
\l_stex_key_autogradable_bool ...
    ..... 8354, 8360, 8365, 8418, 8515
\l_stex_key_continues_tl . 7130, 7139
\l_stex_key_def_tl 125, 3720, 3732,
    3753, 3782, 3819, 3820, 3916, 3927,
    3931, 3951, 4523, 4833, 4846, 4879,
    4880, 5028, 5041, 5072, 5073, 6825
\l_stex_key_deprecate_str .....
    ..... 506, 508, 527, 528
\l_stex_key_due_tl .....
    ..... 9145, 9149, 9184, 9185
\l_stex_key_feedback_str .....
    ..... 8799, 8821, 8822, 8831, 8832
\l_stex_key_feedback_tl ... 8796,
    8892, 8897, 8917, 8918, 8995, 8996
\l_stex_key_file_str .....
    ..... 132, 481, 484, 1566,
        1582, 1585, 1602, 1616, 1644, 1659,
        1670, 1683, 1694, 1698, 1705, 1777
\l_stex_key_for_clist 2848, 6822,
    6830, 6840, 7127, 7136, 7408, 7413
\l_stex_key_for_str .....
    6910
\l_stex_key_from_tl .....
    7128, 7137
\l_stex_key_Ftext_tl .....
    ..... 8895, 8901, 8915, 8993
\l_stex_key_functions_tl . 7132, 7140
\l_stex_key_given_tl .....
    ..... 9144, 9148, 9181, 9182
\l_stex_key_hide_bool 7134, 7143, 7276
\l_stex_key_id_str .....
    ..... 132, 488, 490, 533,
        534, 6891, 6892, 6959, 6961, 7354,
        7363, 7452, 8579, 8581, 8587, 8643,
        8645, 8651, 8692, 8694, 8700, 8743,
        8745, 8752, 8804, 8806, 8814, 8817
\l_stex_key_intent_args_clist .....
    ..... 4155, 4161, 4327, 4336, 4338
\l_stex_key_intent_str .....
    ..... 3964, 4154, 4160, 4250, 4330
\l_stex_key_macroname_str .....
    .. 6821, 6831, 6848, 6850, 6859, 6863
\l_stex_key_method_tl .....
    .. 7133, 7142, 7263, 7264, 7410, 7414
\l_stex_key_mrepos_str .....
    .. 8359, 8367, 8553, 8555, 8563, 9197
\l_stex_key_min_tl .....
    ..... 8357, 8363, 8432, 8527, 8546
\l_stex_key_name_str .....
    ..... 124, 125, 3717, 3725,
        3750, 3751, 3765, 3766, 3779, 3788,
        3797, 3841, 3896, 3925, 3929, 3938,
        3939, 3941, 3949, 3957, 3967, 3975,
        3981, 3982, 4497, 4520, 4521, 4539,
        4541, 4792, 4793, 4803, 4810, 4813,
        4828, 4830, 4843, 4856, 4905, 4908,
        4909, 4913, 4915, 4916, 4985, 4986,
        5008, 5023, 5025, 5038, 5053, 6820,
        6829, 6849, 6850, 6854, 6857, 6863,
        6871, 6880, 6909, 7409, 7416, 7432,
        7433, 7446, 7447, 7448, 7677, 7678,
        7687, 7699, 8358, 8364, 8411, 8413,
        8416, 8503, 8505, 8508, 8510, 8513
\l_stex_key_number_tl .....
    ..... 9143, 9147, 9163, 9164
\l_stex_key_op_tl .....
    ... 3963, 4153, 4159, 4211, 4212,
        4213, 4220, 4221, 4228, 4233, 4364,
        4398, 4402, 4460, 4465, 4482, 4487,
        4490, 4912, 4914, 4917, 4995, 4997
\l_stex_key_post_tl .. 5945, 5949,
    5966, 5977, 6010, 6023, 7032, 7040
\l_stex_key_pre_tl .. 5944, 5948,
    5966, 5977, 6010, 6023, 7032, 7040
\l_stex_key_prec_str .. 126, 3965,
    4152, 4158, 4262, 4265, 4268, 4274,
    4277, 4280, 4283, 4289, 4301, 4302
\l_stex_key_proofend_tl .....
    ..... 7129, 7138, 7240, 7241
\l_stex_key_proot_tl .....
    ..... 5946
\l_stex_key_pts_str .....
    .. 8795, 8798, 8818, 8819, 8828, 8829
\l_stex_key_pts_tl .....
    ..... 8356,
        8362, 8424, 8425, 8431, 8524, 8541
\l_stex_key_reorder_str 3706, 3710,
    3808, 3809, 4867, 4868, 5064, 5065
\l_stex_key_return_tl .....
    ... 3721, 3727, 3784, 3822, 3825,

```

3918, 4804, 4835, 4848, 4882, 4885,  
 4993, 5030, 5043, 5075, 5078, 6826  
`\l_stex_key_role_str` .....  
     ..... 3705, 3713, 3811, 3812,  
         3943, 4864, 4865, 5061, 5062, 6860  
`\l_stex_key_root_tl` ..... 5950  
`\l_stex_key_short_tl` .....  
     ..... 1309, 1312, 1339, 1341  
`\l_stex_key_sig_str` .....  
     ... 122, 2308, 2350, 2498, 2513, 2530  
`\l_stex_key_style_clist` 133, 500,  
 502, 552, 556, 603, 607, 6894, 6895,  
 6988, 8853, 8854, 8939, 8953, 8954  
`\l_stex_key_T_bool` ..... 8893,  
 8898, 8899, 8912, 8925, 8990, 9003  
`\l_stex_key_term_tl` .....  
     .. 7131, 7141, 7260, 7261, 7411, 7415  
`\l_stex_key_testspace_dim` .....  
     ..... 8568, 8571,  
 8573, 8602, 9037, 9039, 9091, 9094  
`\l_stex_key_title_str` ..... 6832  
`\l_stex_key_title_tl` .....  
     .... 494, 496, 1567, 1744, 1745,  
 2520, 6908, 6913, 8420, 8421, 8427,  
 8518, 8519, 8622, 8623, 8683, 8684,  
 8732, 8733, 8785, 8786, 9176, 9177  
`\l_stex_key_Ttext_tl` .....  
     ..... 8894, 8900, 8913, 8991  
`\l_stex_key_type_tl` ..... 125,  
 3719, 3731, 3752, 3783, 3816, 3817,  
 3914, 3926, 3930, 4522, 4834, 4847,  
 4876, 4877, 5069, 5070, 6824, 6833  
`\l_stex_key_variant_str` .....  
     .... 126, 4151, 4157, 4164,  
 4166, 4195, 4248, 4361, 4370, 4402,  
 4507, 4526, 4815, 4906, 4914, 5010  
`\l_stex_key_wikidata_str` .....  
     ..... 3722, 3729, 3802, 3803  
`\stex_keys_define:nnnn` .....  
     ..... 132, 460, 460,  
 479, 487, 493, 499, 505, 511, 1338,  
 1551, 1557, 2497, 3703, 3716, 3924,  
 4150, 4517, 4783, 5943, 6133, 6819,  
 7126, 7407, 7983, 8355, 8552, 8569,  
 8639, 8794, 8891, 9035, 9142, 9241  
`\stex_keys_set:nn` ..... 132,  
 133, 475, 475, 1345, 1569, 1769,  
 1774, 2519, 3740, 3935, 3936, 4171,  
 4495, 4536, 4790, 4983, 5955, 5963,  
 5974, 5996, 6007, 6020, 6148, 6901,  
 6930, 7024, 7030, 7038, 7216, 7287,  
 7342, 7445, 7641, 7675, 8063, 8402,  
 8406, 8493, 8562, 8578, 8601, 8642,  
     8691, 8742, 8803, 8848, 8910, 8948,  
 8988, 9073, 9157, 9161, 9196, 9254  
`\stex_kpsewhich:Nn` 142, 43, 43, 54, 61  
`\c_stex_language_abbrevs_prop` ...  
     ..... 139, 206  
`\stex_language_from_file:` .....  
     ..... 140, 253, 253, 999  
`\c_stex_languages_clist` . 29, 276, 279  
`\c_stex_languages_prop` .....  
     ..... 139, 206, 234, 268, 284, 813  
`\g_stex_last_feature_str` .....  
     ..... 128, 2798, 6174  
`\stex_macro_body:N` . 141, 638, 640, 658  
`\stex_macro_definition:N` 141, 653, 653  
`\l_stex_macroname_str` .....  
     ..... 125, 228, 2788,  
 2789, 3738, 3742, 3744, 3778, 3799,  
 3800, 3937, 3948, 4537, 4791, 4829,  
 4841, 4858, 4859, 4984, 5024, 5036,  
 5055, 5056, 6159, 6859, 7676, 7686  
`\c_stex_main_archive_prop` . 134, 1183  
`\c_stex_main_file` .....  
     .... 137, 138, 189, 1008, 1072, 7753  
`\_stex_map_args:N` .....  
     ... 3826, 4125, 4125, 4227, 4323,  
 4394, 4569, 4886, 5079, 5442, 6188  
`\_stex_map_notation_args:N` .....  
     ..... 4125, 4138, 4427  
`\stex_map_uri:Nnnnn` 138, 819, 826, 2365  
`\c_stex_mathhub_file` .... 138, 1037  
`\c_stex_mathhub_files` .....  
     ..... 1037, 1122, 1129, 1183, 1980  
`\c_stex_mathhub_main_manifest_-`  
 prop ..... 1201, 1202  
`\stex_mathml_arg:nn` ..... 141  
`\stex_mathml_intent:nn` ..... 141  
`\_stex_maybe_brackets:nn` .....  
     .. 4622, 4639, 5103, 5339, 5350, 5892  
`\stex_metagroup_do_in:nn` .....  
     .... 123, 143, 326, 331,  
 338, 2559, 3385, 3419, 7683, 7684, 7695  
`\stex_metagroup_new:n` .....  
     . 143, 320, 321, 325, 2310, 2359, 2843  
`\l_stex_metatheory_uri` 2179, 2385,  
 2388, 2472, 2488, 2502, 2504, 2531,  
 2532, 7602, 7603, 7604, 7651, 7652  
`\c_stex_module_` ..... 123, 125  
`\stex_module_add_code:n` .....  
     123, 2562, 2562, 2565, 2567, 2577, 7659  
`\stex_module_add_morphism:nnn` .. 122  
`\stex_module_add_morphism:nnnn` ..  
     ..... 130, 2583, 2583,  
 2588, 2957, 3299, 6230, 6286, 7662

\stex\_module\_add\_notation:nnnn .  
..... 125, 2626,  
2626, 2637, 2946, 2951, 4359, 4479  
\stex\_module\_add\_symbol:nnnnnnN .  
..... 122, 2589  
\stex\_module\_add\_symbol:nnnnnnnN  
.... 123, 2589, 2974, 2979, 2982,  
3777, 3947, 6154, 6186, 6862, 7685  
\stex\_module\_setup:n ..  
122, 2301, 2301, 2524, 2794, 7522, 7642  
\\_stex\_module\_setup\_top\_nosig:n .  
..... 2309, 2318, 2378, 7638  
\l\_stex\_morphism\_morphisms\_seq ..  
.... 129, 2876, 2893, 2995  
\l\_stex\_morphism\_renames\_prop ...  
.... 129, 2875, 2892,  
2939, 2945, 2961, 2972, 3020, 3422  
\l\_stex\_morphism\_symbols\_prop ...  
.... 129, 2864,  
2874, 2891, 2909, 2913, 2922, 2936,  
2941, 3016, 3034, 3039, 3388, 3484  
\stex\_new\_statement:nn ..... 6838  
\stex\_new\_statement:nnn ..  
.... 6899, 6965, 6971, 6986, 6987  
\stex\_new\_stylable\_cmd:nnnn ..  
.... 133, 538, 538,  
3071, 3284, 3333, 3359, 3403, 3429,  
3610, 3630, 3651, 3739, 3934, 4170,  
4494, 4535, 4789, 4982, 7214, 7608  
\stex\_new\_stylable\_env:nnnnnnn ..  
.... 133, 538, 573, 2518, 3494,  
3516, 3536, 6125, 6211, 6260, 6900,  
7311, 7329, 7341, 8396, 8492, 8594,  
8657, 8706, 8758, 8847, 8947, 9154  
\\_stex\_next\_symbol:n ..  
.. 5199, 5200, 5218, 6537, 6659, 7019  
\c\_stex\_no\_frontmatter\_bool 35, 1445  
\l\_stex\_notation ..... 126  
\\_stex\_notation\_add: ..  
.. 3969, 4175, 4347, 4358, 4545, 7702  
\l\_stex\_notation\_args\_t1 .. 4139,  
4217, 4251, 4322, 4328, 4334, 4543  
\\_stex\_notation\_check: 4174, 4347,  
4347, 4499, 4544, 4806, 5005, 7701  
\\_stex\_notation\_do\_html:n .. 3972,  
4178, 4347, 4367, 4547, 4810, 7704  
\l\_stex\_notation\_downprec .. 5558,  
5886, 5887, 5897, 5898, 5918, 5920  
\l\_stex\_notation\_macrocode\_cs ..  
.... 125–127, 4201, 4246, 4258,  
4352, 4363, 4392, 4452, 4481, 4486,  
4510, 4529, 4818, 4907, 4910, 5013  
\\_stex\_notation\_make\_args: 4202,  
4347, 4353, 4426, 4511, 4530, 4819  
\stex\_notation\_parse:n .....  
..... 3968, 4173, 4210, 4210,  
4498, 4542, 4805, 4996, 4999, 7700  
\stex\_notation\_parse\_and\_then:nw  
..... 126, 128  
\\_stex\_notation\_set\_default:n ..  
.... 4181, 4446, 4467, 4477, 4502  
\stex\_notation\_top:nw ..... 128  
\stex\_par: ..... 8940, 8941, 8943  
\stex\_persist:n ..  
122, 141, 711, 715, 716, 718, 721,  
724, 725, 731, 732, 1177, 1213, 2407  
\c\_stex\_persist\_mode\_bool ..  
.... 141, 31, 702, 765  
\\_stex\_persist\_read\_now: ..  
.... 764, 1212, 7755  
\c\_stex\_persist\_write\_mode\_bool ..  
141, 32, 708, 713, 766, 772, 1211, 2396  
\stex\_pseudogroup:nn ..... 142,  
143, 300, 300, 381, 1093, 2664, 5431  
\stex\_pseudogroup\_restore:N ..  
.... 142, 143, 300, 303, 1106, 2668  
\stex\_pseudogroup\_with:nn ..  
.... 142, 310, 310, 827,  
946, 2387, 2675, 2693, 2718, 4037,  
4718, 4727, 4752, 5918, 5932, 6054  
\c\_stex\_pwd\_file ..  
138, 189, 989, 1184, 1185, 1711, 1866  
\stex\_reactivate\_macro:N 142, 443,  
449, 2582, 3008, 3009, 3010, 3327,  
3329, 3512, 3533, 3557, 3625, 3646,  
3667, 3762, 3990, 4208, 4559, 6170,  
6245, 6308, 6951, 6952, 6953, 6954,  
6955, 6969, 6978, 6979, 6980, 6981,  
6982, 6983, 6984, 7298, 7299, 7300,  
7301, 7302, 7303, 7304, 7305, 7612,  
7636, 7664, 8381, 8382, 8383, 8384,  
8385, 8386, 8387, 8768, 8864, 8966  
\stex\_ref\_new\_doc\_target:n ..  
.... 121, 132, 534, 1500, 1513, 1537  
\\_stex\_ref\_new\_id:n .. 1503, 1514, 6960  
\stex\_ref\_new\_sym\_target:n ..  
121, 1782, 1806, 1815, 6948, 6975, 7009  
\stex\_ref\_new\_sym\_target:nn 121, 1813  
\stex\_ref\_new\_symbol:n ..  
.... 121, 1782, 2432, 3787, 3956  
\l\_stex\_ref\_url\_str .. 1510, 1531, 1533  
\\_stex\_renamedecl\_do:nn ..  
.... 3405, 3410, 3607  
\stex\_require\_archive:n ..  
.... 130, 134, 1083,  
1112, 1112, 1118, 1606, 1702, 1870,  
2039, 2054, 2068, 3124, 3134, 3149

```

\stex_resolve_path_pair:Nnn . . .
    ..... 134, 1855, 1855, 1877
\_stex_return_args:nn . . .
    ..... 3791, 3826, 4886, 5079
\l_stex_return_notation_tl . . .
    ..... 5239, 5416, 5417, 6375,
        6383, 6545, 6546, 6626, 6652, 6653
\stex_set_current_archive:n . . .
    ... 134, 1082, 1082, 1102, 1207, 3268
\stex_set_current_namespace: . . .
    ..... 139, 975, 975, 1000
\stex_set_document_uri: . . .
    ..... 139, 971, 971, 998
\stex_set_language:n . . .
    ..... 140, 232, 232, 252, 269, 2511
\stex_set_notation_macro:nnnnn . .
    125, 126, 2626, 2643, 2647, 2647, 2659
\stex_sms_allow:N . . .
    135, 2122,
        2122, 2133, 2134, 2135, 2136, 2137
\stex_sms_allow_env:n . . .
    136, 2122, 2130, 2549, 3514, 3535,
        3559, 6168, 6242, 6304, 6925, 7673
\stex_sms_allow_escape:N . . .
    136, 2122, 2126, 2496, 2581, 3326,
        3364, 3408, 3627, 3648, 3669, 3763,
        3991, 4209, 4560, 6941, 7065, 7709
\stex_sms_allow_import:Nn . . .
    ..... 136, 2143, 2145, 3328
\stex_sms_allow_import_env:nn . .
    ..... 136, 2143, 2149
\g_stex_sms_import_code . . .
    ..... 136, 2154, 2177, 2195, 3318
\stex_smsmode_do: . . .
    136, 2168, 2170, 2204, 2209, 2494,
        2542, 2578, 3286, 3355, 3362, 3406,
        3503, 3523, 3547, 3621, 3642, 3663,
        3759, 3987, 4191, 4555, 6127, 6236,
        6293, 6918, 6939, 7062, 7665, 7706
\stex_split_slash: . . .
    ..... 5986, 5990, 6039, 7005
\_stex_sref_do_aux:n . . .
    ..... 1788, 1795, 1799, 1802, 7518
\stex_str_if_ends_with:nn .. 136, 628
\stex_str_if_ends_with:nnTF . .
    ..... 136, 628, 982, 3438, 3461, 4094
\stex_str_if_ends_with_p:nn . .
    ..... 136, 628, 4069, 4109
\stex_str_if_starts_with:nn 136, 633
\stex_str_if_starts_with:nnTF . .
    ..... 136, 514,
        633, 1184, 1981, 2908, 6353, 6357, 7068
\stex_str_if_starts_with_p:nn . .
    ..... 136, 633
\stex_structural_feature_-_
    module:nn .. 128, 2783, 2783, 6164
\stex_structural_feature_module_-
    end: 128, 2783, 2797, 6129, 6238, 6300
\stex_structural_feature_-_
    morphism:nn .. 2804
\stex_structural_feature_-_
    morphism:nnnn .. 130
\stex_structural_feature_-_
    morphism:nnnn .. 129, 2805,
        3496, 3517, 3538, 3611, 3631, 3652
\stex_structural_feature_-_
    morphism_check_total: . . .
        ..... 2921, 3525, 3549, 3640, 3661
\stex_structural_feature_-_
    morphism_end: . 129, 2804, 2871,
        3508, 3529, 3553, 3620, 3641, 3662
\stex_style_apply: . . .
    ..... 133, 134, 538, 543, 578,
        583, 2540, 2545, 3086, 3311, 3352,
        3501, 3506, 3521, 3527, 3545, 3551,
        3618, 3638, 3659, 3756, 3984, 4188,
        4514, 4552, 4822, 5017, 6915, 6921,
        7226, 7296, 7316, 7333, 7353, 7371,
        7374, 7383, 8441, 8448, 8529, 8531,
        8591, 8608, 8612, 8654, 8669, 8673,
        8703, 8718, 8722, 8755, 8771, 8775,
        8872, 8874, 8974, 8976, 9169, 9172
\stex_suppress_html:n . . .
    ..... 140, 380, 380, 2155, 4241
\_stex_symdecl_check_terms: . . .
    ... 125, 3769, 3911, 3911, 3946, 4797
\stex_symdecl_do: . . .
    ..... 124, 125, 3768, 3840,
        3840, 3945, 4796, 4991, 6861, 7680
\stex_symdecl_html: . . .
    ..... 3772, 3795, 3959, 6868
\stex_symdecl_top:n . . .
    ..... 125, 3746, 3764, 3764, 4538
\stex_symdef_styledefs: . 4519, 4551
\stex_term_arg:nnn . . .
    ..... 5549, 5553, 5560, 5570
\stex_term_arg:nnnn .. 4442, 4683, 4691,
    5553, 5554, 5585, 5619, 5656, 5741
\stex_term_arg_aB:nnnn .. 4696, 4704, 4711, 5575, 5583
\stex_term_do_aB_clist: .. 4718,
    4719, 4728, 4732, 4753, 4758, 5580,
    5592, 5621, 5671, 5700, 5748, 5751
\stex_term_oma:nnn . . .
    ..... 4635, 5441, 5826, 5827, 5853
\stex_term_oma_or_omb:nnn . . .
    ..... 4635, 4640, 4688, 4701

```

```

\stex_term_omb:nnn ..... 4688,
    4701, 5469, 5470, 5855, 5856, 5882
\stex_term_oms:nnn .....
    ..... 4402, 5226, 5228, 5762,
    5772, 5821, 5825, 5957, 5967, 5978
\stex_term_oms_or_omv:nnn .....
    ..... 4004, 4623, 5104, 5226,
    5228, 5260, 5262, 5340, 5351, 5424,
    5762, 5825, 6392, 6430, 6468, 6524
\stex_term_omv:nnn .....
    ..... 4975, 5260, 5262,
    5762, 5785, 5822, 6001, 6014, 6027
\stex_undefined: ..... 41
\stex_uri_add_module:NNn .....
    ..... 139, 944, 944, 969, 1515, 7602
\stex_uri_from_current_file:Nn ..
    ..... 139, 934, 934, 972
\stex_uri_from_current_file_-
    nolang:Nn ..... 139, 934, 940, 976
\stex_uri_from_repo_file ..... 139
\stex_uri_from_repo_file:NNNn ...
    ..... 138, 139, 900, 903, 936, 1619
\stex_uri_from_repo_file_-
    nolang:NNNn ..... 139, 900, 900, 941
\stex_uri_resolve:Nn ..... 138,
    885, 885, 888, 2488, 2504, 2508, 7521
\stex_uri_set:Mn 138, 835, 881, 884, 932
\stex_uri_use:N ..... 138, 868,
    889, 893, 938, 973, 977, 1510, 1516,
    1517, 1521, 1587, 1589, 1621, 1628,
    1645, 1646, 2388, 2532, 3163, 3168,
    3169, 3182, 3187, 3188, 6892, 7652
\stex_vardecl_notation_macro: ...
    ... 125, 4500, 4807, 4903, 4903, 5006
\stex_variable:nnnnnnnN . 4782, 4923
\l_stex_variables_prop .....
    ..... 125, 4780, 4828, 4935, 5023
stex internal commands:
\stex_annotation_env_str ... 395, 396
\stex_aux_apply_patch:n .. 544, 551
\stex_aux_apply_patch_begin:n ..
    ..... 579, 602
\stex_aux_apply_patch_end:n ...
    ..... 584, 617
\stex_aux_args:n . 642, 646, 649, 652
\stex_aux_end: ..... 638, 639, 642
\stex_aux_params:n 656, 687, 694, 697
\stex_aux_patch:nnn ..... 540, 565
\stex_aux_patch:nnnn ..... 575, 592
\stex_aux_prefix:n .. 654, 662, 676
\stex_aux_prefix_long:n .....
    ..... 668, 672, 678, 685
\stex_aux_split_at_bracket:w ..
    ..... 515, 523
\stex_aux_start: ..... 638, 641
\l_stex_aux_t1 .....
    ..... 465, 466, 467, 469, 472, 473
\stex_debug_nn ..... 67, 70, 75
\l_stex_debug_cl ..... 83, 85, 88
\stex_debug_env_str ..... 81, 82, 85
\stex_doc_check_topsect: .....
    ..... 1423, 1429, 1435, 1441
\stex_doc_do_section:n .....
    ..... 1321, 1325, 1329, 1346
\stex_doc_maketitle: ... 1264, 1269
\stex_doc_orig_backmatter .....
    ..... 1457, 1460, 1478
\stex_doc_orig_frontmatter .....
    ..... 1447, 1450, 1467
\stex_doc_set_title:n .. 1241, 1257
\stex_doc_skip_fragment:n .....
    ..... 1386, 1392,
    1396, 1400, 1401, 1402, 1403, 1404
\stex_doc_skip_section: .....
    ..... 1354, 1376, 1382
\stex_doc_skip_section_i: .....
    ..... 1363, 1366, 1369, 1377, 1382
\stex_doc_title_html: .....
    ..... 1246, 1249, 1261
\g_stex_doc_title_t1 .....
    ... 1236, 1238, 1245, 1251, 1256, 1259
\stex_expr_aB_arg:nnnnn 5590, 5597
\stex_expr_aB_simple_arg:nnnnn
    ..... 5608, 5617
\stex_expr_add_prop_arg:nw ...
    ..... 5437, 5461, 5464
\stex_expr_arg:n ..... 5438, 5474
\l_stex_expr_arg_counter_int ...
    ..... 5428, 5435, 5450, 5486, 5487
\stex_expr_arg_do:nnn .....
    ..... 5488, 5494, 5510, 5545, 5552
\stex_expr_arg_inner:nn .....
    ..... 5477, 5480, 5484, 5490
\stex_expr_assoc_make_seq:nnn .
    ..... 5652, 5681, 5760
\stex_expr_assoc_seq:nnnnnnn ..
    ..... 5601, 5649
\stex_expr_check:n ..... 5518
\stex_expr_check:nTF ... 5487, 5493
\stex_expr_check_b:nn .. 5442, 5467
\l_stex_expr_count_int .....
    ... 5582, 5588, 5598, 5619, 5656, 5741
\l_stex_expr_cs ... 5683, 5686, 5706
\l_stex_expr_customs_prop .....
    ..... 5433, 5445, 5446,
    5447, 5462, 5498, 5504, 5519, 5522
\l_stex_expr_customs_seq .....
    ... 5434, 5451, 5452, 5453, 5463, 5520

```

```

\__stex_expr_do_aB_clist: .....
..... 5575, 5580, 5621, 5700
\__stex_expr_do_ab_next:nnn ...
..... 5441, 5443, 5469, 5470
\__stex_expr_do_headterm:nn ...
..... 5750, 5782, 5795, 5798, 5823
\__stex_expr_do_ref:nNn ... 5957,
5965, 5976, 5998, 6009, 6022, 6032
\__stex_expr_do_seqmap:nnnnnn ...
..... 5606, 5713
\__stex_expr_end: ..... 5603, 5613
\__stex_expr_gobble:nnnnnnn ...
..... 5603, 5613
\l__stex_expr_iarg_t1 5693, 5695, 5706
\__stex_expr_invoke_custom:n ...
..... 5288, 5302, 5429
\__stex_expr_invoke_math: 5283, 5291
\__stex_expr_invoke_op_custom:n ...
..... 5288, 5295, 5421
\__stex_expr_invoke_op_notation:w
..... 5297, 5298, 5336
\__stex_expr_invoke_return: ...
..... 5377, 5380
\__stex_expr_invoke_return_-
maybe:n ...
..... 5319, 5329, 5358
\__stex_expr_invoke_return_next:
..... 5365, 5374
\__stex_expr_invoke_text: 5283, 5286
\__stex_expr_is_seqmap:n ...
..... 5642
\__stex_expr_is_seqmap:nTF ...
..... 5605
\__stex_expr_is_varseq:n ...
..... 5631
\__stex_expr_is_varseq:nTF 5599, 5718
\l__stex_expr_left_bracket_str ...
..... 5888, 5914, 5922, 5932, 5933
\l__stex_expr_old_seq ...
..... 5725, 5728, 5732, 5737
\l__stex_expr_reset_t1 ...
..... 5199, 5202, 5206, 5207, 5213
\__stex_expr_ret_cs ...
..... 5362, 5367, 5394, 5399, 5404
\__stex_expr_return_arg:n 5364, 5370
\l__stex_expr_return_args_t1 ...
.. 5359, 5371, 5376, 5385, 5401, 5406
\__stex_expr_return_notation:n ...
..... 5382, 5415
\l__stex_expr_return_this_t1 ...
..... 5360, 5376,
5381, 5385, 5389, 5390, 5400, 5408
\l__stex_expr_right_bracket_str ...
..... 5889, 5915, 5927, 5932, 5934
\__stex_expr_setup:nnnnnn ...
..... 5225, 5237, 5259
\__stex_expr_varseq_in_map:nnnnnnnn
..... 5720, 5759
\__stex_features_add_definiens:nn
..... 2855, 3012
\__stex_features_break: ... 2856, 2860
\__stex_features_check_break:nnnnnnnn
..... 3033, 3038, 3047, 3050, 3057
\__stex_features_clean:nnw 2902, 2910
\__stex_features_do_decls: 2894, 2906
\__stex_features_do_elaboration:
..... 2884, 2934
\__stex_features_do_for_list: ...
..... 2846, 3011
\__stex_features_do_morph:nnnn ...
..... 2989, 2993
\__stex_features_do_morphisms:n ...
..... 2895, 2987, 2997
\__stex_features_elab_check: ...
..... 2942, 2971
\l__stex_features_feature_str ...
..... 2838, 2960
\__stex_features_get_check:nnnn ...
..... 3017, 3045
\l__stex_features_implicit_bool ...
.. 2804, 2823, 2826, 2878, 2950, 2976
\__stex_features_reactivate: ...
..... 2841, 3000
\__stex_features_rename:nn 2961, 2964
\__stex_features_rename_all: .. 2898
\__stex_features_renamed_-
check:nnnn ...
..... 3021, 3031
\__stex_features_set_definiens_-
macros: ...
..... 2856, 2860
\__stex_features_set_definiens_-
macros_i:nnnnnn ...
..... 2863, 2867
\__stex_features_setup: ... 2840, 2890
\__stex_features_split_qm:w ...
..... 2933, 2953
\__stex_features_sym_i_gobble:nnnnnn
..... 2969, 2977
\l__stex_features_tmp ...
..... 2945, 2947, 2972, 2973, 2974
\__stex_features_total_check: ...
..... 2923, 2927
\__stex_groups_do: ... 351, 358, 367
\__stex_groups_do_in:nn 333, 340, 361
\__stex_groups_exists:n ...
..... 326
\__stex_groups_exists:nTF ...
..... 332
\l__stex_groups_ids_seq 320, 323, 359
\__stex_groups_tmp ...
..... 342, 349, 355
\l__stex_importmodule_archive_-
str 3130, 3135, 3145, 3150, 3267, 3268
\__stex_importmodule_check_-
file:nn ...
..... 3211,
3212, 3213, 3214, 3215, 3216, 3238,
3239, 3240, 3241, 3242, 3243, 3277

```

```

\__stex_importmodule_get_from_-
    file:nnn ..... 3171, 3207
\__stex_importmodule_get_from_-
    file_safe:nnn ..... 3190, 3234
\__stex_importmodule_get_-
    module:nnn ..... 3132, 3138, 3196
\__stex_importmodule_get_module_-
    safe:nnn ..... 3147, 3153, 3202
\__stex_importmodule_get_module_-
    uri:nnn ..... 3157, 3198
\__stex_importmodule_get_module_-
    uri_safe:nnn ..... 3176, 3204
\__stex_importmodule_import_-
    module:nn ..... 3285, 3290, 3330
\__stex_importmodule_import_-
    module_presms:nn ..... 3316, 3330
\__stex_importmodule_load_file:n
    ..... 3228, 3231, 3255, 3260, 3265
\l__stex_importmodule_path_seq ..
    ..... 3107, 3108, 3110
\__stex_importmodule_seq .....
    ..... 3092, 3093, 3094
\l__stex_importmodule_seq .....
    .. 3114, 3116, 3208, 3209, 3235, 3236
\l__stex_importmodule_str .....
    ..... 3137, 3138, 3152,
    3153, 3209, 3225, 3236, 3251, 3254,
    3259, 3266, 3270, 3278, 3279, 3281
\__stex_inputs_bibresource:n ...
    ..... 1943, 1955, 1957
\l__stex_inputs_gin_repo_str ...
    ..... 2036, 2072, 2077
\l__stex_inputs_id_seq .....
    ..... 1986, 1988, 1995
\l__stex_inputs_id_str ... 1974, 1986
\__stex_inputs_inputref:nn 1931, 1941
\__stex_inputs_inputref_html:nn .
    ..... 1896, 1934
\__stex_inputs_inputref_pdf:nn ..
    ..... 1920, 1935
\__stex_inputs_libinput:n .....
    ..... 2007, 2020, 2022
\l__stex_inputs_libinput_files_-
    seq ..... 1945,
    1948, 1977, 1991, 1992, 2001, 2002,
    2009, 2012, 2027, 2028, 2101, 2102
\__stex_inputs_mhinput:nn 1880, 1893
\l__stex_inputs_path_seq .....
    ..... 1982, 1989, 1996, 1999
\l__stex_inputs_path_str .....
    1979, 1981, 1989, 1990, 1991, 1992,
    1995, 1996, 1999, 2000, 2001, 2002
\l__stex_inputs_tmp_str .. 2028, 2030
\__stex_inputs_up_archive:nn ...
    ..... 1944, 1970, 2008, 2026, 2100
\__stex_inputs_usetikzlibrary:n .
    ..... 2099, 2113, 2115
\__stex_inputs_usetikzlibrary_-
    i:nn ..... 2083, 2102
\l__stex_iterate_continue_bool ..
    ..... 2743, 2746, 2759, 2779
\__stex_iterate_it_decl_check:nnnn
    ..... 2704, 2711
\__stex_iterate_it_decl_i:n .....
    ..... 2696, 2700, 2713
\__stex_iterate_it_not_check:nnnn
    ..... 2732, 2736
\__stex_iterate_it_not_i:n .....
    ..... 2721, 2725, 2738
\__stex_iterate_iterate_morphism:nn
    ..... 2749, 2753, 2762, 2765
\l__stex_iterate_mods_seq .....
    ..... 2692, 2701, 2702,
    2717, 2726, 2727, 2742, 2767, 2768
\__stex_iterate_morphism_cs:nnnn
    ..... 2744, 2770
\__stex_iterate_not_cs:nnnnn ...
    ..... 2718, 2719, 2729
\__stex_iterate_sym_cs:nnnnnnnn
    .. 2675, 2676, 2686, 2693, 2694, 2707
\l__stex_iterate_todo_tl .....
    ..... 2748, 2752, 2766, 2779
\l__stex_lang_lang_tl . 234, 237, 244
\l__stex_lang_str .. 266, 267, 268, 269
\l__stex_lang_turkish_bool .....
    ..... 277, 282, 291
\__stex_mathhub_check_manifest:n
    ..... 1136, 1138, 1140, 1145
\l__stex_mathhub_cs .. 1090, 1091,
    1094, 1097, 1099, 1103, 1107, 1108
\__stex_mathhub_do_manifest:n ...
    ..... 1115, 1120
\__stex_mathhub_find_manifest:n .
    ..... 1124, 1134, 1152, 1198
\l__stex_mathhub_key 1162, 1163, 1165
\c__stex_mathhub_manifest_ior ...
    ..... 1154, 1156, 1159, 1174
\l__stex_mathhub_manifest_str ...
    ..... 1119, 1121, 1123,
    1127, 1137, 1139, 1148, 1156, 1199
\__stex_mathhub_parse_manifest:n
    ..... 1131, 1155, 1200
\l__stex_mathhub_prop .....
    ..... 1157, 1158, 1166, 1167,
    1168, 1169, 1170, 1175, 1176, 1179
\l__stex_mathhub_seq 1161, 1162, 1164

```

```

\l__stex_mathhub_str ..... 1038, 1039, 1043,
..... 1044, 1049, 1055, 1060, 1065, 1066
\l__stex_mathhub_top_str . 1135, 1158
\l__stex_mathhub_val ..... 1164, 1166, 1167, 1168, 1169, 1170
\__stex_module_setup_end: 2356, 2377
\__stex_module_setup_get_uri_-
str:n ..... 2306, 2363
\__stex_module_setup_load_meta: .
..... 2315, 2384
\__stex_module_setup_load_sig: ...
..... 2336, 2346
\l__stex_module_setup_ns_str ...
..... 2307, 2309, 2364, 2366, 2373
\l__stex_module_setup_seg .....
..... 2369, 2370, 2371
\l__stex_module_setup_seq .....
..... 2368, 2369, 2371, 2373
\__stex_module_setup_setup_-
nested:n ..... 2302, 2354
\__stex_module_setup_setup_top:n
..... 2302, 2305
\__stex_module_setup_setup_top_-
sig:n ..... 2309, 2331
\l__stex_module_setup_sigfile ...
..... 2347, 2348, 2350
\__stex_module_setup_split_-
module:n ..... 2356, 2377
\__stex_modules_activate_-
i:nnnnnnn .. 2610, 2614
\__stex_modules_activate_not:nn ..
..... 2634, 2640
\__stex_modules_activate_sym:n ..
..... 2602, 2607
\__stex_modules_export:n . 2573, 2575
\__stex_modules_persist_module: .
..... 2396, 2406, 7516
\__stex_modules_persist_nots_-
i:nn ..... 2422, 2443
\l__stex_modules_restore_mod_str
..... 2436, 2464
\__stex_modules_restore_module:nnnn
..... 2408, 2427
\__stex_modules_restore_nots:n ..
..... 2440, 2449
\__stex_modules_restore_nots_i:n
..... 2450, 2453, 2470
\__stex_modules_restore_nots_-
ii:nnnn .. 2457, 2461
\__stex_modules_set_metatheory:nn
..... 2473, 2493
\__stex_modules_tl ..... 2429, 2430
\l__stex_modules_tl ..... 2462, 2467
\l__stex_morphisms_ass_tl .....
..... 3563, 3579, 3583, 3593, 3594
\__stex_morphisms_do_morph_-
assign:nnn ..... 3436, 3439, 3477
\__stex_morphisms_do_parsed_-
assign: ..... 3585, 3588
\__stex_morphisms_do_parsed_-
newname: ..... 3591, 3598
\__stex_morphisms_do_parsed_-
newname:w ..... 3600, 3602, 3606
\__stex_morphisms_end: ... 3591, 3606
\l__stex_morphisms_morphism_dom_-
str ... 3430, 3443, 3455, 3465, 3479
\l__stex_morphisms_name_str ...
..... 3561, 3570,
..... 3571, 3572, 3576, 3577, 3578, 3589
\l__stex_morphisms_newname_str ...
..... 3562, 3581, 3582, 3590, 3591
\l__stex_morphisms_next_tl .....
..... 3568, 3573, 3575
\__stex_morphisms_parse_assign:n
..... 3560, 3613, 3633, 3654
\l__stex_morphisms_seq .....
..... 3565, 3566, 3568, 3570,
..... 3573, 3575, 3577, 3579, 3581, 3583
\__stex_notations_add:nnnnn ...
..... 4669, 4674, 4678
\__stex_notations_add_last:nnnnn
..... 4662, 4673
\__stex_notations_add_missing_-
args:nn ..... 4227, 4342
\__stex_notations_add_next:nnnnnn
..... 4664, 4668
\l__stex_notations_after_tl ...
..... 4648, 4675
\__stex_notations_args_end: ...
..... 4128, 4131, 4134,
..... 4141, 4144, 4147, 4657, 4660, 4670
\l__stex_notations_args_tl ...
..... 4564, 4568, 4581
\__stex_notations_augment_arg:nn
..... 4569, 4586
\__stex_notations_check_aB_-
arg:Nn ..... 4709, 4717, 4726, 4751
\l__stex_notations_clist_count_-
int ..... 4749, 4757, 4763
\l__stex_notations_code_tl ...
... 4605, 4620, 4628, 4636, 4650,
..... 4651, 4654, 4682, 4690, 4695, 4703
\__stex_notations_complex:nnnnnn
..... 4614, 4632
\__stex_notations_const_precs: ...
..... 4219, 4261

```

```

\l__stex_notations_cs . . . . .
    ..... 4621, 4626, 4637, 4646
\__stex_notations_do_argname:nn .
    ..... 4323, 4326
\__stex_notations_do_argnames: .
    ..... 4297, 4321
\__stex_notations_fun_precs: .
    ..... 4224, 4273
\__stex_notations_make_arg:nnnn .
    ..... 4427, 4431
\__stex_notations_make_arg_-
    html:nn ..... 4394, 4409
\__stex_notations_make_name: .
    ..... 4566, 4590, 4598
\__stex_notations_map_args_i:w ..
    ..... 4127, 4131, 4134
\__stex_notations_map_args_ii:w .
    ..... 4140, 4144, 4147
\__stex_notations_map_cs: .
    ..... 4380, 4382,
        4729, 4731, 4739, 4754, 4756, 4767
\l__stex_notations_missing_str: .
    ..... 4225, 4343
\l__stex_notations_missing_tl: .
    ..... 4226, 4255, 4344
\l__stex_notations_name_str: .
    ..... 4592, 4593, 4594, 4599
\l__stex_notations_opprec_tl 4249,
    4263, 4266, 4268, 4275, 4278, 4280,
    4284, 4291, 4304, 4310, 4316, 4371
\__stex_notations_parse_notation_-
    args:nnnnw ..... 4657, 4660, 4670
\__stex_notations_parse_precs: .
    ..... 4295, 4300
\l__stex_notations_pre_tl: .
    ..... 4635, 4650, 4687, 4700
\l__stex_notations_precs_seq: .
    ..... 4216, 4286,
        4293, 4314, 4316, 4329, 4335, 4372
\__stex_notations_process_-
    notation:nmmnnn ..... 4604, 4610
\l__stex_notations_seq: .
    ..... 4302, 4303, 4305, 4306, 4313
\__stex_notations_simple:nnnnn: .
    ..... 4612, 4618
\l__stex_notations_str: .
    .. 4303, 4304, 4305, 4307, 4313, 4314
\__stex_notations_styledefs: .
    ..... 4187, 4194
\__stex_path_: .
    ..... 110, 120
\l__stex_path_a_seq: .
    ..... 777, 783, 789
\l__stex_path_a_tl: .
    ..... 789, 791
\__stex_path_auth:n: .
    ..... 138, 819, 822, 823,
        824, 827, 828, 873, 889, 894, 945, 947
\l__stex_path_auth_str: .
    ..... 841, 849, 854, 859
\l__stex_path_b_seq: .
    ..... 778, 784, 790, 796
\l__stex_path_b_tl: .
    ..... 790, 791
\__stex_path_canonicalize:N: .
    ..... 123, 134, 140
\__stex_path_colonslash: .
    ..... 835, 840, 845
\l__stex_path_do_hooks_pre_tl: .
    ..... 1017, 1018, 1030, 1033
\__stex_path_dodots:n: .
    ..... 145, 149, 155
\l__stex_path_file: .
    ..... 908, 914, 920, 925
\__stex_path_from_repo_file:NNNNn: .
    ..... 901, 904, 907
\l__stex_path_maybewin_str: .
    ..... 171, 172, 173
\l__stex_path_mod: .
    ..... 852, 855, 860
\__stex_path_module:n: .
    ..... 138,
        823, 827, 830, 875, 889, 947, 949, 951
\l__stex_path_name: .
    ..... 857, 860
\__stex_path_name:n: .
    ..... 138,
        824, 827, 831, 876, 889, 947, 956, 958
\l__stex_path_path: .
    ..... 846, 847, 850, 855, 860, 926, 927
\__stex_path_path:n: .
    ..... 138, 822, 827, 829, 874, 889, 947
\__stex_path_relativize:N: .
    ..... 917, 924
\l__stex_path_return_tl: .
    ..... 779, 785, 792, 795, 797
\l__stex_path_seq: .
    ..... 143, 146, 151,
        159, 160, 163, 802, 803, 804, 809,
        810, 812, 814, 817, 925, 928, 930, 932
\g__stex_path_stack: .
    ..... 979, 993, 1004, 1005, 1007, 1010
\l__stex_path_str: .
    ..... 111, 112, 113, 116, 118,
        119, 120, 122, 125, 132, 133, 142,
        144, 148, 160, 190, 192, 194, 200,
        202, 204, 801, 802, 803, 808, 809,
        810, 812, 813, 814, 1005, 1010, 1011
\l__stex_path_t1: .
    ..... 928, 930
\l__stex_path_uri: .
    ..... 910, 911, 912, 917, 932
\__stex_path_uri_set:NnN: .
    ..... 836, 882, 886
\__stex_path_uri_set:Nnnnn: .
    ..... 849, 854, 859, 871, 879, 913, 919
\__stex_path_uri_use:w: .
    ..... 889, 895
\l__stex_path_win_drive: .
    ..... 112, 117, 124, 126
\__stex_path_win_take:w: .
    ..... 110, 120
\__stex_persist_env_str: .
    ..... 699, 700, 701, 705, 706, 707

```

```

\__stex_persist_load_file:n . . .
    ..... 728, 760, 769
\__stex_persist_read_and_write: . .
    ..... 745, 767
\c__stex_persist_sms_iow . . .
    ..... 711, 719, 741, 742, 757
\__stex_persist_write_only: . .
    ..... 740, 754, 761, 772
\__stex_proof_add_counter: 7189, 7370
\__stex_proof_begin_proof:nn . .
    ..... 7284, 7312, 7330
\l__stex_proof_counter_intarray .
    ..... 7147, 7152,
    7155, 7164, 7167, 7176, 7184, 7185,
    7193, 7198, 7205, 7211, 7285, 7286
\__stex_proof_end_list: . .
    .. 7254, 7325, 7386, 7396, 7462, 7485
\__stex_proof_html: . 7258, 7280, 7436
\__stex_proof_html_env:n . .
    ..... 7269, 7290, 7345
\l__stex_proof_in_spfblock_bool .
    ... 7283, 7313, 7324, 7331, 7359,
    7362, 7381, 7384, 7386, 7391, 7394,
    7402, 7444, 7458, 7480, 7500, 7506
\__stex_proof_inblock_restore: . .
    ..... 7388, 7401, 7502
\__stex_proof_inc_counter: . .
    ..... 7172, 7384, 7475, 7477
\l__stex_proof_inc_counter_bool 7146
\__stex_proof_insert_number: . .
    ..... 7148, 7368, 7475, 7477
\__stex_proof_make_step_macro:Nnnnn
    ..... 7442, 7475, 7476, 7477
\__stex_proof_number_as_string:N
    ..... 7159, 7355, 7364, 7453
\__stex_proof_proof_box_tl 7129, 7233
\__stex_proof_remove_counter: . .
    ..... 7201, 7395
\__stex_proof_start_list:n . .
    ..... 7247, 7313, 7368, 7462, 7485
\__stex_proof_step_html:nn 7424,
    7460, 7462, 7468, 7483, 7485, 7489
\__stex_refs_add_doc_ref:nn . .
    ..... 1517, 1539, 1550
\l__stex_refs_default_archive_
    str . . .
    ..... 1559, 1565, 1574
\l__stex_refs_default_file_str ..
    ..... 1560, 1566, 1575
\l__stex_refs_default_title_tl ..
    ..... 1561, 1567, 1576
\__stex_refs_do_autoref:n . .
    ..... 1634, 1646, 1657,
    1661, 1672, 1686, 1712, 1723, 1731
\__stex_refs_do_internal_link:nn
    ..... 1826, 1838, 1844
\__stex_refs_do_return:nnnn . .
    ..... 1740, 1754, 1762
\__stex_refs_do_sref:nn .. 1642, 1771
\__stex_refs_do_sref_in:n . .
    ..... 1650, 1665, 1676, 1682, 1780
\__stex_refs_do_url_link:nn . .
    ..... 1832, 1846
\l__stex_refs_file . .
    .. 1601, 1603, 1618, 1620, 1708, 1709
\l__stex_refs_file_str . .
    1692, 1697, 1703, 1708, 1709, 1710,
    1711, 1716, 1720, 1722, 1730, 1742
\g__stex_refs_files_seq . .
    ..... 1500, 1540, 1545, 1592, 1653
\__stex_refs_find_uri:n . .
    ..... 1579, 1770, 1775
\__stex_refs_find_uri_in_-
    file:nnn . . .
    ..... 1588, 1593, 1624
\__stex_refs_find_uri_in_prop_-
    file:N . . .
    ..... 1599, 1608, 1613
\l__stex_refs_id_str . .
    ..... 1718, 1753, 1759, 1760
\c__stex_refs_iow . .
    ..... 1491, 1492, 1493, 1519
\__stex_refs_new_symbol:n 1784, 1792
\l__stex_refs_prop . .
    .. 1607, 1608
\__stex_refs_restore_target:nnnnn
    ..... 1719, 1751
\l__stex_refs_return_t1 . .
    ..... 1717, 1721, 1727, 1741
\__stex_refs_set_keys_b:n . .
    ..... 1563, 1649, 1664, 1675, 1776
\l__stex_refs_str 1506, 1508, 1510,
    1515, 1517, 1522, 1794, 1796, 6961
\__stex_refs_sym_aux:nn . .
    ..... 1823, 1836, 1841, 1852
\l__stex_refs_unnamed_counter_-
    int . . .
    ..... 1501, 1505, 1506
\l__stex_refs_uri . .
    ..... 1515, 1516, 1619, 1621
\l__stex_refs_uri_str . .
    ... 1580, 1591, 1603, 1614, 1618,
    1621, 1628, 1643, 1653, 1654, 1655,
    1656, 1657, 1660, 1661, 1663, 1669,
    1671, 1672, 1674, 1684, 1685, 1686,
    1713, 1724, 1732, 1752, 1757, 1758
\__stex_seqs_add: . . .
    ..... 5003, 5021
\l__stex_seqs_args_t1 . .
    .. 5079, 5081
\__stex_seqs_check_terms: 5002, 5049
\l__stex_seqs_clist . .
    ..... 5116, 5123, 5130, 5134

```

```

\l__stex_seqs_cs ..... 5077, 5081, 5137, 5141,
      5149, 5152, 5161, 5168, 5181, 5182
\l__stex_seqs_do_all:w ... 5175, 5185
\l__stex_seqs_do_first: ... 5098, 5159
\l__stex_seqs_do_first_arg:n ...
      ..... 5157, 5164
\l__stex_seqs_do_first_next: ...
      ..... 5166, 5171
\l__stex_seqs_do_one:w ... 5173, 5179
\l__stex_seqs_do_op:w ... 5097, 5101
\l__stex_seqs_doop_arg:n .. 5117, 5128
\l__stex_seqs_doop_range:w 5110, 5114
\l__stex_seqs_first_args_tl ...
      ..... 5163, 5181, 5182, 5186, 5187
\l__stex_seqs_get_index_notation:n
      ..... 5109, 5147, 5180
\l__stex_seqs_html: ..... 5001, 5051
\l__stex_seqs_macro: ..... 5004, 5035
\l__stex_seqs_make_args: .. 5014, 5048
\l__stex_seqs_range_clist ...
      ..... 4994, 5029, 5042
\g__stex_smsmode_allowed_escape_-
  tl ..... 2120, 2127, 2256
\g__stex_smsmode_allowed_import_-
  env_seq 2144, 2150, 2190, 2239, 2242
\g__stex_smsmode_allowed_import_-
  tl ..... 2143, 2146, 2187, 2234
\g__stex_smsmode_allowed_tl ...
      ..... 2119, 2123, 2252
\g__stex_smsmode_allowedenvs_seq
      ..... 2121, 2131, 2261, 2264
\g__stex_smsmode_bool ...
      ..... 2138, 2139, 2141, 2157
\l__stex_smsmode_check_begin:Nn ..
      ..... 2239, 2261, 2273
\l__stex_smsmode_check_cs:NNn ...
      ..... 2212, 2220
\l__stex_smsmode_check_end:Nn ...
      ..... 2242, 2264, 2282
\l__stex_smsmode_do:w ...
      ..... 2206, 2218, 2220, 2222,
      2244, 2254, 2266, 2279, 2286, 2289
\l__stex_smsmode_do_aux:N .. 2220, 2226
\l__stex_smsmode_do_aux_curr:N ...
      ..... 2186, 2197, 2228
\l__stex_smsmode_do_aux_imports:N
      ..... 2186, 2232
\l__stex_smsmode_do_aux_normal:N ..
      ..... 2197, 2250
\l__stex_smsmode_importmodules_-
  seq ..... 2175
\l__stex_smsmode_in_smsmode:n ...
      ..... 2155, 2185, 2195, 2196
\l__stex_smsmode_sigmodules_seq 2176
\l__stex_smsmode_smsmode_do: ...
      ..... 2168, 2204
\l__stex_smsmode_start_smsmode:n .
      ..... 2166, 2193, 2198
\l__stex_statements_do_defref:nn .
      ..... 6996, 7025, 7031, 7039
\l__stex_statements_force_id: ...
      ..... 6855, 6958, 6966, 6989
\l__stex_statements_html_keyvals:nn
      ..... 6885, 6906, 6934
\l__stex_statements_setup:nn ...
      .. 6847, 6967, 6972, 6986, 6990, 6993
\l__stex_statements_setup_def: ...
      ..... 6945, 6968, 6991
\l__stex_statements_uri_str ...
      6870, 6874, 6875, 6880, 6881, 7051,
      7054, 7057, 7059, 7104, 7107, 7110
\l__stex_structures_assigned_seq
      ..... 6533, 6644, 6683, 6699
\l__stex_structures_begin:nn ...
      ..... 6126, 6147, 6224, 6279
\l__stex_structures_check_-
  def:nnnnnnnn ..... 6298, 6311
\l__stex_structures_clist ...
      ..... 6507, 6528, 6534, 6536
\l__stex_structures_comp_cs ...
      ..... 6576, 6581
\l__stex_structures_cs ...
      ..... 6509, 6515, 6522
\l__stex_structures_current_type:
      ..... 6463, 6594, 6665
\l__stex_structures_current_-
  type_tl .... 6390, 6426, 6466, 6471
\l__stex_structures_do_assign:nn .
      ..... 6410, 6414
\l__stex_structures_do_assign_-
  list:n ..... 6386, 6407
\l__stex_structures_do_decl:nnnnnnnn
      ..... 6721, 6741
\l__stex_structures_do_decl_-
  nomacro:nnnnnnnn ..... 6719, 6727
\l__stex_structures_do_externals:
      ..... 6130, 6172, 6239, 6301
\l__stex_structures_end: ...
      ..... 6354, 6358, 6374, 6379
\l__stex_structures_exstruct_-
  name_str ..... 6274, 6279, 6323
\l__stex_structures_extend_-
  structure:nn ..... 6251, 6274
\l__stex_structures_extend_-
  structure_i:NnnnnnnnN .. 6248, 6256
\l__stex_structures_external_-
  decl:nnnn ..... 6175, 6179

```

```

\l__stex_structures_extmod_str . .
    ..... 6249, 6253
\l__stex_structures_extname_-
    count ..... 6317, 6321, 6323
\l__stex_structures_field_name_-
    str ..... 6631, 6635, 6636, 6667
\l__stex_structures_fields_clist
    ..... 6387,
        6408, 6415, 6433, 6436, 6690, 6691
\l__stex_structures_get_field_-
    name:n ..... 6630, 6642
\l__stex_structures_imports_seq .
    ..... 6212, 6217, 6225,
        6262, 6267, 6281, 6803, 6806, 6809
\l__stex_structures_invocation_-
    type:n ..... 6338, 6385
\l__stex_structures_invoke_-
    field:n ..... 6608, 6640
\l__stex_structures_invoke_maybe_-
    field:nn ..... 6597, 6601
\l__stex_structures_invoke_this:n
    ..... 6347, 6589
\l__stex_structures_invoke_top:n .
    ..... 6328,
        6331, 6355, 6359, 6362, 6365, 6367
\l__stex_structures_make_mod:n ...
    ..... 6432, 6444
\l__stex_structures_make_oml:n ...
    ..... 6436, 6450
\l__stex_structures_make_oml:nn ..
    ..... 6451, 6453
\l__stex_structures_make_prop: ...
    ..... 6475, 6602, 6680
\l__stex_structures_make_prop_-
    assign: ..... 6476, 6605, 6689
\l__stex_structures_make_prop_-
    assign:nn ..... 6692, 6697
\l__stex_structures_make_prop_-
    assign_replace:nnnn ... 6700, 6706
\l__stex_structures_make_type:n ..
    ..... 6395, 6400, 6402, 6418
\l__stex_structures_maybe_-
    notation:w ..... 6342, 6344, 6470
\l__stex_structures_merge:nw ...
    ..... 6336, 6352
\l__stex_structures_more_-
    nextsymbol_tl ... 6643, 6645, 6670
\l__stex_structures_name_str 6118,
    6135, 6140, 6142, 6149, 6150, 6155,
    6156, 6161, 6162, 6165, 6181, 6187
\l__stex_structures_new_extstruct_-
    name: ..... 6261, 6319
\l__stex_structures_present: ...
    ..... 6481, 6606
\l__stex_structures_present:nn ...
    ..... 6485, 6491, 6496, 6503, 6506
\l__stex_structures_present_-
    entry:nn ..... 6511, 6517, 6532
\l__stex_structures_present_i:w ...
    ..... 6477, 6483, 6489
\l__stex_structures_present_ii:nw
    ..... 6494, 6502
\l__stex_structures_prop .....
    6521, 6633, 6641, 6673, 6681, 6698,
        6701, 6707, 6728, 6730, 6742, 6744
\l__stex_structures_prop_do_-
    decls: ..... 6685, 6716
\l__stex_structures_prop_do_-
    notations: ..... 6686, 6765
\l__stex_structures_redo_t1 ...
    .. 6660, 6684, 6710, 6757, 6768, 6783
\l__stex_structures_replace_-
    this_t1 ..... 6173
\l__stex_structures_seq .....
    ..... 6682, 6729, 6743, 6767
\l__stex_structures_set_comp_t1 .
    ..... 6327, 6376, 6380, 6538, 6655
\l__stex_structures_set_custom_-
    comp:n ..... 6381, 6570
\l__stex_structures_set_customcomp:
    ..... 6354, 6379
\l__stex_structures_set_this:n ...
    ..... 6603, 6612
\l__stex_structures_set_thiscomp:
    ..... 6327, 6564
\l__stex_structures_set_thisnotation:
    ..... 6358, 6374
\l__stex_structures_shift_-
    argls:nn ..... 6188, 6193
\l__stex_structures_this_t1 ...
    6339, 6539, 6540, 6543, 6558, 6615,
        6618, 6625, 6646, 6647, 6650, 6664
\l__stex_symdecl_add_decl: 3770, 3776
\l__stex_symdecl_args_t1 . 3826, 3828
\l__stex_symdecl_cs .....
    .. 3824, 3828, 4018, 4020, 4022, 4048
\l__stex_symdecl_do_args: . 3843, 3902
\l__stex_symdecl_env_str .....
    ..... 3676, 3677, 3678
\l__stex_symdecl_get_from_one_-
    string:n ..... 4057, 4102
\l__stex_symdecl_get_symbol_from_-
    cs: ..... 4024, 4035
\l__stex_symdecl_get_symbol_from_-
    modules:nn ..... 4059, 4091
\l__stex_symdecl_get_symbol_from_-
    string:n ... 4025, 4027, 4030, 4052
\l__stex_symdecl_name ... 4055, 4061

```

|                                                         |                                                   |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                        |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
|---------------------------------------------------------|---------------------------------------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-----|--------------------------|----------------------------|------------------------|----------------------|-------------------------------|------------------|-------------------------|------|----------------------|-----|-----------------------|-----------------|-------------------------------|-----|--------------------------|---------|----------------------------|-----|------------------------|------|------------------------------------|-----|------------------------|------------|--|--|--------------------------|---------------------------------------------------------|--|--|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|-------------------------|------|--|--|------------------------|----------|--|--|------------------------|------------------|
| \_stex_symdecl_parse_arity: . . . . .                   | 3842, 3848                                        | \stexstylecopymod . . . . .                | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \l\_stex_symdecl_seq . . . . .                          | 4054, 4055, 4056, 4060                            | \stexstylecopymodule . . . . .             | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_symdecl_set_textsymdecl_-<br>macro:nnn . . . . . | 3975, 3993                                        | \stexstyledefinition . . . . .             | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_symdecl_sym_from_str_-<br>i:nnnn . . . . .       | 4065, 4096                                        | \stexstyleexample . . . . .                | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_symdecl_sym_i_finish:nnnnnnN<br>. . . . .        | 4071, 4078                                        | \stexstyleextstructure . . . . .           | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_symdecl_sym_i_gobble:nnnnnn<br>. . . . .         | 4073, 4076                                        | \stexstyleimportmodule . . . . .           | 104, 133                                                                                                                                                                                                                                                                                                                                                                                                               |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_vars_add: . . . . .                              | 4798, 4826                                        | \stexstyleinterpretmod . . . . .           | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \l\_stex_vars_args_t1 . . . . .                         | 4886, 4888                                        | \stexstyleinterpretmodule . . . . .        | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \l\_stex_vars_bind_bool . . . . .                       | 4781, 4784, 4786, 4870                            | \stexstylemathstructure . . . . .          | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_vars_check_var:nnnnnnnnN<br>. . . . .            | 4936, 4940                                        | \stexstylemcb . . . . .                    | 8879                                                                                                                                                                                                                                                                                                                                                                                                                   |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \l\_stex_vars_cs . . . . .                              | 4884, 4888                                        | \stexstyleMMTinclude . . . . .             | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_vars_get_var:n . . . . .                         | 4934, 4952, 4960                                  | \stexstylemodule . . . . .                 | 104, 134                                                                                                                                                                                                                                                                                                                                                                                                               |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_vars_html: . . . . .                             | 4800, 4853                                        | \stexstylenotation . . . . .               | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_vars_macro: . . . . .                            | 4799, 4840                                        | \stexstyleparagraph . . . . .              | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \_stex_vars_set_vars:nnnnnn . . . . .                   | 4921, 4942, 4945                                  | \stexstyleproof . . . . .                  | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \sTeX/ComputerScience/Software . . . . .                | 15                                                | \stexstylerealization . . . . .            | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| stex_annotation_env (env.) . . . . .                    | 141                                               | \stexstylerealize . . . . .                | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexcommentfont . . . . .                              | 7244, 7314, 7381, 7403, 7466, 7487                | \stexstylerenamedecl . . . . .             | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexdoctitle . . . . .                                 | 1237, 1296,<br>1308, 2522, 8076, 8216, 8421, 8519 | \stexstylerequiremodule . . . . .          | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXexport . . . . .                                   | 49, 86, 87, 121, 2571                             | \stexstylesc . . . . .                     | 8981                                                                                                                                                                                                                                                                                                                                                                                                                   |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexhtmlfalse . . . . .                                | 419                                               | \stexstylepsfsketch . . . . .              | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexhtmltrue . . . . .                                 | 416                                               | \stexstylesubproof . . . . .               | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalAssocArgMarkerI . . . . .                  | 127                                               | \stexstylesymdecl . . . . .                | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalAssocArgMarkerII . . . . .                 | 127                                               | \stexstylesymdef . . . . .                 | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalNotation . . . . .                         | 4247, 4581, 4601                                  | \stexstyletextsymdecl . . . . .            | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalSetSrefSymURL . . . . .                    | 1496, 1796, 1799, 1802                            | \stexstyleusemodule . . . . .              | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalSrefRestoreTarget . . . . .                | 1494, 1520, 1719                                  | \stexstylevardef . . . . .                 | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalSymbolAfterInvokationTL<br>. . . . .       | 127, 131                                          | \stexstylevarnotation . . . . .            | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalTermMathArgiii . . . . .                   | 127                                               | \stexstylevarseq . . . . .                 | 104                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalTermMathAssocArgiiii . . . . .             | 127                                               | \stopsolutions . . . . .                   | 111, 8632                                                                                                                                                                                                                                                                                                                                                                                                              |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalTermMathOMAiii . . . . .                   | 127                                               | str commands:                              |                                                                                                                                                                                                                                                                                                                                                                                                                        |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXInternalTermMathOMBiii . . . . .                   | 127                                               | \STEXInternalTermMathOMSOrOMViii . . . . . | 127                                                                                                                                                                                                                                                                                                                                                                                                                    | \c_backslash_str . . . . . | 118 | \STEXinvisible . . . . . | 81, 4344, 5219, 7294, 7349 | \c_colon_str . . . . . | 835, 890, 1161, 2366 | \STEXRestoreNotsEnd . . . . . | 2423, 2447, 2454 | \c_dollar_str . . . . . | 4336 | \stexstyle . . . . . | 133 | \c_hash_str . . . . . | 689, 2628, 4343 | \stexstyleassertion . . . . . | 104 | \c_percent_str . . . . . | 54, 200 | \stexstyleassign . . . . . | 104 | \str_case:Nn . . . . . | 1165 | \stexstyleassignMorphism . . . . . | 104 | \str_case:nn . . . . . | 4680, 5468 |  |  | \str_case:mnTF . . . . . | 1407, 3852, 4410,<br>4432, 5496, 5524, 7716, 7737, 7855 |  |  | \str_clear:N . . . . . | 117, 480, 481, 488,<br>506, 910, 1049, 1121, 1580, 1897,<br>2072, 2182, 2364, 2498, 2806, 3015,<br>3103, 3130, 3145, 3430, 3561, 3562,<br>3704, 3705, 3706, 3707, 3717, 3718,<br>3722, 3742, 3925, 3964, 3965, 4015,<br>4016, 4151, 4152, 4153, 4154, 4924,<br>4951, 4959, 5556, 6135, 6203, 6820,<br>6821, 6823, 6870, 7048, 7101, 7160,<br>7409, 7640, 7711, 7732, 7984, 8255,<br>8256, 8358, 8359, 8553, 8570, 8795 |  |  | \str_const:Nn . . . . . | 7774 |  |  | \str_count:n . . . . . | 630, 635 |  |  | \str_gset:Nn . . . . . | 1497, 1809, 1817 |
| \STEXInternalTermMathOMSOrOMViii . . . . .              | 127                                               | \c_backslash_str . . . . .                 | 118                                                                                                                                                                                                                                                                                                                                                                                                                    |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXinvisible . . . . .                                | 81, 4344, 5219, 7294, 7349                        | \c_colon_str . . . . .                     | 835, 890, 1161, 2366                                                                                                                                                                                                                                                                                                                                                                                                   |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \STEXRestoreNotsEnd . . . . .                           | 2423, 2447, 2454                                  | \c_dollar_str . . . . .                    | 4336                                                                                                                                                                                                                                                                                                                                                                                                                   |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexstyle . . . . .                                    | 133                                               | \c_hash_str . . . . .                      | 689, 2628, 4343                                                                                                                                                                                                                                                                                                                                                                                                        |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexstyleassertion . . . . .                           | 104                                               | \c_percent_str . . . . .                   | 54, 200                                                                                                                                                                                                                                                                                                                                                                                                                |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexstyleassign . . . . .                              | 104                                               | \str_case:Nn . . . . .                     | 1165                                                                                                                                                                                                                                                                                                                                                                                                                   |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
| \stexstyleassignMorphism . . . . .                      | 104                                               | \str_case:nn . . . . .                     | 4680, 5468                                                                                                                                                                                                                                                                                                                                                                                                             |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
|                                                         |                                                   | \str_case:mnTF . . . . .                   | 1407, 3852, 4410,<br>4432, 5496, 5524, 7716, 7737, 7855                                                                                                                                                                                                                                                                                                                                                                |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
|                                                         |                                                   | \str_clear:N . . . . .                     | 117, 480, 481, 488,<br>506, 910, 1049, 1121, 1580, 1897,<br>2072, 2182, 2364, 2498, 2806, 3015,<br>3103, 3130, 3145, 3430, 3561, 3562,<br>3704, 3705, 3706, 3707, 3717, 3718,<br>3722, 3742, 3925, 3964, 3965, 4015,<br>4016, 4151, 4152, 4153, 4154, 4924,<br>4951, 4959, 5556, 6135, 6203, 6820,<br>6821, 6823, 6870, 7048, 7101, 7160,<br>7409, 7640, 7711, 7732, 7984, 8255,<br>8256, 8358, 8359, 8553, 8570, 8795 |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
|                                                         |                                                   | \str_const:Nn . . . . .                    | 7774                                                                                                                                                                                                                                                                                                                                                                                                                   |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
|                                                         |                                                   | \str_count:n . . . . .                     | 630, 635                                                                                                                                                                                                                                                                                                                                                                                                               |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |
|                                                         |                                                   | \str_gset:Nn . . . . .                     | 1497, 1809, 1817                                                                                                                                                                                                                                                                                                                                                                                                       |                            |     |                          |                            |                        |                      |                               |                  |                         |      |                      |     |                       |                 |                               |     |                          |         |                            |     |                        |      |                                    |     |                        |            |  |  |                          |                                                         |  |  |                        |                                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |                         |      |  |  |                        |          |  |  |                        |                  |

\str\_gset\_eq:NN . . . . . 1044, 2872, 2873  
\str\_if\_empty:NFT . . . . .  
  . . . . . 82, 124, 144, 527, 533,  
  700, 706, 917, 1037, 1039, 1060,  
  1123, 1127, 1137, 1139, 1199, 1585,  
  1591, 1597, 1643, 1644, 1659, 1670,  
  1690, 1713, 1724, 1732, 1752, 1777,  
  1843, 2308, 2551, 2788, 2811, 3019,  
  3023, 3105, 3267, 3443, 3576, 3590,  
  3677, 3765, 3799, 3802, 3805, 3808,  
  3811, 3938, 4009, 4163, 4262, 4274,  
  4283, 4330, 4792, 4858, 4861, 4864,  
  4867, 4953, 4961, 4985, 4988, 5055,  
  5058, 5061, 5064, 6056, 6139, 6149,  
  6198, 6205, 6635, 6848, 6849, 6854,  
  6871, 6891, 6959, 7054, 7107, 7354,  
  7363, 7432, 7446, 7452, 7677, 7713,  
  7734, 7845, 7904, 8067, 8411, 8503,  
  8508, 8579, 8623, 8643, 8692, 8743,  
  8785, 8804, 8818, 8821, 8828, 8831  
\str\_if\_empty:nTF . . . . . 104, 156,  
  566, 593, 837, 1504, 1857, 2501, 3416  
\str\_if\_eq:NNTF . . . . . 267, 791  
\str\_if\_eq:nnTF . . . . .  
  . 119, 157, 158, 173, 241, 257, 258,  
  259, 281, 396, 629, 634, 664, 680,  
  689, 701, 707, 1192, 1627, 1711,  
  1753, 1758, 1760, 1814, 2288, 2370,  
  2609, 2642, 2747, 3032, 3037, 3046,  
  3049, 3225, 3251, 3435, 3451, 3678,  
  4265, 4277, 4289, 4941, 4944, 5135,  
  7071, 7908, 7911, 7976, 8135, 8885  
\str\_if\_eq\_p:nn 4067, 4068, 4107, 4108  
\str\_if\_exist:NFT . . . . . 327, 1840  
\str\_if\_in:NnTF . . . . . 4343, 6037, 7003  
\str\_item:Nn . . . . . 119, 173, 3907  
\str\_lowercase:n . . . . . 8885  
\str\_map\_break: . . . . . 3853  
\str\_map\_break:n . . . . . 3854, 3858, 3862,  
  3866, 3870, 3874, 3878, 3882, 3886  
\str\_map\_inline:Nn . . . . . 3851  
\str\_new:N . . . . .  
  231, 1119, 1559, 1560, 2036, 2293, 3738  
\str\_put\_right:Nn . . . . . 7167  
\str\_range:Nnn . . . . . 2030  
\str\_range:nnn . . . . . 630, 635, 669  
\str\_replace\_all:Nnn . . . . . 118  
\str\_set:Nn . . . . . 48, 56,  
  111, 112, 113, 116, 132, 233, 280,  
  322, 835, 1080, 1135, 1148, 1163,  
  1164, 1506, 1508, 1510, 1574, 1575,  
  1603, 1614, 1621, 1628, 1692, 1697,  
  1703, 1709, 1718, 1859, 1866, 1871,  
  1979, 1989, 1999, 2028, 2327, 2343,  
  2366, 2436, 2665, 2819, 2824, 2827,  
  2838, 2861, 2862, 3059, 3060, 3061,  
  3094, 3099, 3104, 3109, 3115, 3123,  
  3125, 3131, 3135, 3136, 3137, 3146,  
  3150, 3151, 3152, 3159, 3166, 3169,  
  3172, 3178, 3185, 3188, 3191, 3209,  
  3236, 3281, 3479, 3572, 3578, 3582,  
  3712, 3744, 3766, 3856, 3860, 3864,  
  3868, 3872, 3876, 3880, 3884, 3888,  
  3937, 3939, 3941, 3943, 4039, 4040,  
  4080, 4081, 4112, 4113, 4164, 4196,  
  4225, 4230, 4349, 4389, 4400, 4537,  
  4791, 4793, 4925, 4969, 4971, 4984,  
  4986, 4989, 5244, 5245, 5658, 5702,  
  5743, 6035, 6038, 6117, 6140, 6150,  
  6159, 6253, 6317, 6321, 6323, 6631,  
  6636, 6860, 6874, 6880, 7001, 7004,  
  7051, 7104, 7356, 7365, 7454, 7676,  
  7678, 8581, 8645, 8694, 8745, 8806  
\str\_set\_eq:NN . . . . .  
  1055, 1565, 1566, 2538, 2809, 2839,  
  3966, 3967, 4166, 4195, 4497, 4507,  
  4526, 4540, 4541, 4815, 5010, 5999,  
  6012, 6025, 6142, 6207, 6850, 6859,  
  6909, 6961, 7340, 7421, 7639, 7643,  
  7667, 7669, 7682, 7696, 7698, 7699  
\str\_uppercase:n . . . . . 7976  
\l\_tmpa\_str . . . . . 254, 255, 256, 257,  
  258, 259, 260, 264, 280, 284, 285,  
  287, 1191, 1192, 1204, 1205, 1207,  
  1214, 1218, 1221, 1897, 1898, 1899,  
  1904, 1912, 2824, 2827, 2832, 2839  
\subparagraph . . . . . 27, 82  
subproblem (env.) . . . . . 8490  
subproof (env.) . . . . . 7340  
\subproof . . . . . 7298, 7399  
\subproofautorefname . . . . . 7340  
\subsection . . . . . 26, 27, 82  
\subsubsection . . . . . 27, 82  
\svar . . . . . 95, 3792, 4966  
\symbol . . . . . 89  
\symdecl . . . . . 23,  
  24, 31, 32, 39, 40, 47, 87-89, 99,  
  104, 124, 129, 135, 142, 143, 3001,  
  3738, 7540, 7542, 7571, 7579, 7582  
\symdef . . . . . 32, 33, 35, 40, 41, 88,  
  95, 124, 129, 135, 3003, 4517, 7524,  
  7527, 7533, 7535, 7537, 7539, 7549,  
  7550, 7551, 7552, 7558, 7566, 7577,  
  7586, 7587, 7590, 7593, 7595, 7597  
\Symname . . . . . 89, 100, 5941  
\symname 19, 20, 24, 62, 89, 90, 100, 102, 5941  
\symref . . . . . 19,  
  20, 24, 48, 88, 89, 93, 100, 102, 5941

|                                                                               |                                                                       |
|-------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| \symrefemph . . . . .                                                         | 102, 103, 6053                                                        |
| \symuse . . . . .                                                             | 48, 89, 5190, 5269, 5746, 6427,<br>6434, 6445, 6527, 6592, 6662, 6663 |
| sys commands:                                                                 |                                                                       |
| \sys_get_shell:nnN . . . . .                                                  | 45                                                                    |
| \sys_if_platform_windows:TF . . . . .                                         | 51, 109, 168, 189, 199                                                |
| T                                                                             |                                                                       |
| \target . . . . .                                                             | 134, 137–139                                                          |
| \test . . . . .                                                               | 73, 115                                                               |
| test . . . . .                                                                | 110                                                                   |
| \testbigspace . . . . .                                                       | 9106                                                                  |
| \testemptypage . . . . .                                                      | 73, 115                                                               |
| \testemptypage . . . . .                                                      | 9100                                                                  |
| testheading (env.) . . . . .                                                  | 74, 116                                                               |
| \testheading . . . . .                                                        | 9231                                                                  |
| \testmedspace . . . . .                                                       | 9105                                                                  |
| \testnewpage . . . . .                                                        | 73, 115                                                               |
| \testnewpage . . . . .                                                        | 9107                                                                  |
| \testsmallspace . . . . .                                                     | 73, 73, 73, 115, 115, 115                                             |
| \testsmallspace . . . . .                                                     | 9104                                                                  |
| \testspace . . . . .                                                          | 73, 115                                                               |
| \testspace . . . . .                                                          | 8602, 9103                                                            |
| \TeX . . . . .                                                                | 23, 24                                                                |
| \tex . . . . .                                                                | 23, 24                                                                |
| T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>C</sub> commands: |                                                                       |
| \@ . . . . .                                                                  | 2087, 2090, 2094                                                      |
| \@addtoreset . . . . .                                                        | 8374                                                                  |
| \@arabic . . . . .                                                            | 8227                                                                  |
| \@author . . . . .                                                            | 8206                                                                  |
| \@auxout . . . . .                                                            | 1789, 7761, 8484                                                      |
| \@bonuspointsfalse . . . . .                                                  | 9262                                                                  |
| \@bonuspointstrue . . . . .                                                   | 9267                                                                  |
| \@currentHref . . . . .                                                       | 7356, 7365, 7454                                                      |
| \@currentcounter . . . . .                                                    | 1523                                                                  |
| \@currentlabel . . . . .                                                      | 1524, 7355,<br>7356, 7364, 7365, 7453, 7454, 8066                     |
| \@currentlabelname . . . . .                                                  | 1526, 1527                                                            |
| \@currenvir . . . . .                                                         | 7950, 7951                                                            |
| \@dblarg . . . . .                                                            | 8199, 8208                                                            |
| \@gobble . . . . .                                                            | 455                                                                   |
| \@ifnextchar . . . . .                                                        | 454                                                                   |
| \@ifstar . . . . .                                                            | 8178                                                                  |
| \@mainmatterfalse . . . . .                                                   | 1452, 1462                                                            |
| \@mainmattertrue . . . . .                                                    | 1473, 1484                                                            |
| \@notprerr . . . . .                                                          | 235, 1417                                                             |
| \@onlypreamble . . . . .                                                      | 235, 1417                                                             |
| \@rusttexfalse . . . . .                                                      | 392                                                                   |
| \@title . . . . .                                                             | 1266, 1267, 8215                                                      |
| \activate@excursion . . . . .                                                 | 8239, 8244                                                            |
| \bbl@loaded . . . . .                                                         | 8338, 9126                                                            |
| \beamer@shortauthor . . . . .                                                 | 8202, 8204, 8219                                                      |
| \beamer@shorttitle . . . . .                                                  | 8211, 8213, 8222                                                      |
| \c@chapter . . . . .                                                          | 7931                                                                  |
| \c@framenumber . . . . .                                                      | 8227                                                                  |
| \c@part . . . . .                                                             | 7943                                                                  |
| \compemph@uri . . . . .                                                       | 103, 131, 4386, 6045, 6053                                            |
| \correction@table . . . . .                                                   | 9210                                                                  |
| \defemph@uri . . . . .                                                        | 103, 6053                                                             |
| \define@key . . . . .                                                         | 2038, 2053, 2067                                                      |
| \Gin@eheight . . . . .                                                        | 9303, 9306, 9311, 9316                                                |
| \Gin@ewidth . . . . .                                                         | 8169, 8170, 9302, 9312, 9316                                          |
| \Gin@exclamation . . . . .                                                    | 9302, 9303, 9311                                                      |
| \Gin@mrepos . . . . .                                                         | 2040,<br>2043, 2047, 2055, 2069, 2073, 2078                           |
| \hwexam@bonuspts . . . . .                                                    | 9264                                                                  |
| \hwexam@checktime . . . . .                                                   | 9258                                                                  |
| \hwexam@duration . . . . .                                                    | 9234, 9237, 9243, 9248                                                |
| \hwexam@kw@due . . . . .                                                      | 9185                                                                  |
| \hwexam@kw@forgrading . . . . .                                               | 9226                                                                  |
| \hwexam@kw@given . . . . .                                                    | 9182                                                                  |
| \hwexam@kw@grade . . . . .                                                    | 9227                                                                  |
| \hwexam@kw@probs . . . . .                                                    | 9227                                                                  |
| \hwexam@kw@pts . . . . .                                                      | 9228                                                                  |
| \hwexam@kw@reached . . . . .                                                  | 9229                                                                  |
| \hwexam@kw@sum . . . . .                                                      | 9227                                                                  |
| \hwexam@kw@testemptypage . . . . .                                            | 9101                                                                  |
| \hwexam@min . . . . .                                                         | 9235, 9242, 9247, 9258                                                |
| \hwexam@minutes@kw . . . . .                                                  | 9235                                                                  |
| \hwexam@reqpts . . . . .                                                      | 9244, 9249, 9261, 9265                                                |
| \hwexam@tools . . . . .                                                       | 9245, 9250                                                            |
| \hwexam@totalmin . . . . .                                                    | 9257, 9258                                                            |
| \hwexam@totalpts . . . . .                                                    | 9256, 9265                                                            |
| \if@bonuspoints . . . . .                                                     | 9260                                                                  |
| \if@rustex . . . . .                                                          | 400                                                                   |
| \itemize@inner . . . . .                                                      | 8016, 8022, 8031                                                      |
| \itemize@label . . . . .                                                      | 8020, 8023, 8026                                                      |
| \itemize@level . . . . .                                                      | 8014, 8019, 8022, 8031                                                |
| \itemize@outer . . . . .                                                      | 8015, 8019                                                            |
| \lst@mrepos . . . . .                                                         | 2058, 2062                                                            |
| \ltx@ifpackageloaded . . . . .                                                | 236, 240, 2037,<br>2052, 2066, 7234, 8337, 8905, 9125                 |
| \m@switch . . . . .                                                           | 5912                                                                  |
| \mdf@patchamsthdm . . . . .                                                   | 8040                                                                  |
| \metakeys@show@keys . . . . .                                                 | 8017                                                                  |
| \notesslides@slidelabel . . . . .                                             | 8043, 8058                                                            |
| \ns@author . . . . .                                                          | 8199, 8200                                                            |
| \ns@title . . . . .                                                           | 8208, 8209                                                            |
| \pgf@temp . . . . .                                                           | 2084, 2085, 2086                                                      |
| \pgfkeys@spdef . . . . .                                                      | 2084                                                                  |
| \pgfutil@empty . . . . .                                                      | 2086                                                                  |
| \pgfutil@InputIfFileExists . . . . .                                          | 2093                                                                  |
| \prematrestop@endsfragment . . . . .                                          | 7949, 7952, 7958                                                      |
| \problem@kw@* . . . . .                                                       | 8334                                                                  |
| \problem@kw@correct . . . . .                                                 | 8913, 8991                                                            |
| \problem@kw@feedback . . . . .                                                | 8832                                                                  |

```

\problem@kw@grading ..... 8785
\problem@kw@hint ..... 8683
\problem@kw@minutes . 8459, 8546, 8844
\problem@kw@note ..... 8732
\problem@kw@pts .... 8454, 8541, 8839
\problem@kw@solution ..... 8622
\problem@kw@wrong ..... 8915, 8993
\problem@restore .... 8485, 8489, 9204
\stex@backend ..... 140, 390
\symrefemph@uri .....
.... 102, 103, 5957, 5967, 5978, 6053
\varname@uri .....
.... 103, 131, 5998, 6011, 6024, 6053
\texname ..... 24
\text ..... 20
\textbf ..... 19, 6095,
.... 7836, 8472, 8475, 9085, 9175, 9180
\textcolor ..... 9081
\textsymdecl .. 23, 24, 88, 129, 3002, 3923
\textwarning ..... 108
\textwidth ..... 8167, 8189, 9223
\the 345, 346, 348, 350, 352, 2087, 2088, 2089
\theassignment ..... 9168, 9175
\thechapter ... 1327, 1368, 1396, 1431,
.... 7862, 7867, 7871, 7875, 7879, 7883
\theframenumber ..... 8066
\theparagraph ..... 7879, 7883
\thepart ... 1323, 1365, 1392, 1425, 7857
\theplainsproblem ..... 8378, 8379
\thesection ..... .
.... 7867, 7871, 7875, 7879, 7883, 8379
\thesproblem . 8375, 8379, 8475, 8485, 9168
\thesubparagraph ..... 7883
\thesubsection ... 7871, 7875, 7879, 7883
\thesubsubsection ..... 7875, 7879, 7883
\this ..... 57–59, 96, 97, 5241,
.... 6114, 6543, 6620, 6621, 6625, 6650
\thisarchive ..... 3307, 3348
\thisargs ..... 3754, 4524
\thiscopyname ..... .
.... 3499, 3519, 3543, 3616, 3636, 3657
\thisdeclname ... 3751, 3982, 4196, 4521
\thisdecluri ..... .
.... 3750, 3981, 4197, 4200, 4520, 4528
\thisdefiniens ..... 3753, 4523
\thisfor ..... 6910
\thismodulename 133, 2539, 3084, 3309, 3350
\thismoduleuri ..... .
.... 133, 2538, 3083, 3308, 3349,
.... 3500, 3520, 3544, 3617, 3637, 3658
\thisname ..... 6909
\thisnotation 4198, 4508, 4527, 4816, 5011
\thisnotationvariant ..... .
.... 4195, 4507, 4526, 4815, 5010
\ThisStyle ..... 5912
>thisstyle ..... 133, 553, 556,
.... 557, 558, 604, 607, 608, 609, 610,
.... 618, 621, 622, 3085, 3310, 3351,
.... 3755, 3983, 4506, 4525, 4814, 5009
\thistitle 103, 133, 2520, 2521, 2522,
.... 6908, 8427, 8471, 8472, 8476, 8477
\thistype ..... 3752, 4522
>thisvarname ..... .
.... 4505, 4509, 4813, 4817, 5008, 5012
\throwaway ..... 135
\tikzinput ... 118, 2078, 9295, 9300, 9326
tikzinput commands:
\c_tikzinput_image_bool 34, 9284, 9291
\tiny ..... 8044, 8058
\ttitle ..... 73, 116
\ttitle ..... 8208
tl commands:
\tl_clear:N ..... 494, 553, 604,
.... 779, 1339, 1717, 2177, 2296, 2502,
.... 2766, 3085, 3310, 3351, 3563, 3719,
.... 3720, 3721, 3755, 3903, 3926, 3927,
.... 3963, 3983, 4217, 4226, 4322, 4465,
.... 4506, 4525, 4564, 4759, 4814, 4974,
.... 4997, 5009, 5239, 5249, 5346, 5359,
.... 5557, 5944, 5945, 5946, 6043, 6134,
.... 6615, 6643, 6684, 6824, 6825, 6826,
.... 7127, 7128, 7130, 7131, 7132, 7133,
.... 7410, 7411, 8254, 8796, 8892, 8894,
.... 8895, 9036, 9143, 9144, 9145, 9215,
.... 9216, 9217, 9242, 9243, 9244, 9245
\tl_const:Nn ..... 5884, 5885
\tl_count:N ..... 5501, 5507
\tl_count:n ..... 5632, 5643
\tl_gclear:N ..... 2322
\tl_gput_left:Nn ..... 466, 468, 469
\tl_gput_right:Nn ..... 346, 2123,
.... 2127, 2146, 2299, 2563, 3318, 8240
\tl_gset:Nn ..... 348, 461, 462,
.... 1018, 1238, 1245, 2147, 2151, 4233,
.... 5202, 5206, 8524, 8527, 9207, 9208
\tl_gset_eq:NN ..... 46, 2798
\tl_head:N .. 894, 945, 4022, 6565, 6571
\tl_head:n ..... .
.... 648, 689, 692, 4710, 5633, 5644
\tl_if_empty:NTF ..... .
.... 618, 795, 1020, 1256, 1266, 1309,
.... 1744, 2385, 2521, 2531, 3368, 3593,
.... 3782, 3816, 3819, 3822, 3951, 4126,
.... 4139, 4211, 4220, 4228, 4398, 4487,
.... 4563, 4876, 4879, 4882, 4890, 4912,
.... 4995, 5069, 5072, 5075, 5083, 5313,
.... 5325, 5416, 5436, 5521, 5660, 5773,
.... 5786, 5828, 5857, 6116, 7240, 7260,

```

7263, 7651, 8170, 8261, 8420, 8471,  
 8476, 8518, 8622, 8683, 8732, 8913,  
 8915, 8917, 8991, 8993, 8995, 9082,  
 9163, 9176, 9181, 9184, 9234, 9261  
 $\backslash$ tl\_if\_empty:nTF . . . . . 172, 181,  
 463, 647, 663, 679, 688, 890, 891,  
 950, 957, 1095, 1564, 1721, 1743,  
 1901, 1923, 1954, 2019, 2112, 2219,  
 2585, 2600, 2652, 2712, 2737, 2807,  
 2822, 2928, 2994, 3095, 3129, 3144,  
 3158, 3177, 3208, 3235, 3424, 4020,  
 4133, 4146, 4611, 4661, 4968, 5464,  
 5485, 5561, 5584, 5987, 6188, 6312,  
 6361, 6364, 6409, 6419, 6508, 6510,  
 6604, 6613, 6708, 6718, 6782, 6927,  
 7049, 7102, 7118, 7614, 8201, 8210  
 $\backslash$ tl\_if\_empty\_p:N . . . . . 785  
 $\backslash$ tl\_if\_eq:NNTF 6206, 6565, 6571, 9053  
 $\backslash$ tl\_if\_eq:NnTF . . . . . 8287, 8433, 8436, 8453, 8458  
 $\backslash$ tl\_if\_eq:nnTF . . . . . 2454, 4735, 5129, 5644, 5689, 5727  
 $\backslash$ tl\_if\_exist:NTF . . . . . 304,  
 345, 360, 399, 557, 609, 621, 1282,  
 1291, 1526, 1685, 2555, 6273, 7888  
 $\backslash$ tl\_if\_in:NnTF . . . . . 2234, 2252, 2256  
 $\backslash$ tl\_item:nn . . . . . 5635  
 $\backslash$ tl\_map\_inline:Nn . . . . . 2187  
 $\backslash$ tl\_map\_inline:nn . . . . . 311, 315  
 $\backslash$ tl\_new:N . . . . . 970, 1030, 1236, 1561, 2119, 2120,  
 2143, 2154, 2472, 4780, 5195, 5199  
 $\backslash$ tl\_put\_left:Nn . . . . . 4650, 4651, 5390, 8401, 9156  
 $\backslash$ tl\_put\_right:Nn . . . . . 1212, 2373, 2748, 2752, 3905, 3906,  
 4328, 4334, 4344, 4571, 4576, 4579,  
 4682, 4690, 4695, 4703, 4766, 5227,  
 5251, 5261, 5371, 6614, 6710, 6757,  
 6768, 6783, 9057, 9219, 9220, 9221  
 $\backslash$ tl\_range:nnn . . . . . 665, 681  
 $\backslash$ tl\_set:Nn . . . . . 171, 305, 342, 548, 567, 569,  
 588, 589, 594, 595, 597, 598, 792,  
 872, 963, 1017, 1274, 1315, 1450,  
 1460, 1576, 1741, 2040, 2043, 2055,  
 2069, 2539, 2648, 2650, 2653, 2868,  
 3063, 3064, 3065, 3066, 3067, 3307,  
 3348, 3499, 3519, 3543, 3573, 3579,  
 3583, 3616, 3636, 3657, 3750, 3826,  
 3981, 4042, 4043, 4044, 4045, 4046,  
 4083, 4084, 4085, 4086, 4087, 4115,  
 4116, 4117, 4118, 4119, 4197, 4212,  
 4221, 4246, 4263, 4266, 4275, 4278,  
 4284, 4291, 4310, 4520, 4568, 4581,  
 4599, 4620, 4635, 4636, 4648, 4687,  
 4700, 4719, 4732, 4758, 4841, 4886,  
 4927, 4928, 4929, 4930, 4931, 5036,  
 5079, 5163, 5196, 5201, 5203, 5210,  
 5240, 5246, 5247, 5248, 5360, 5381,  
 5501, 5507, 5520, 5526, 5529, 5532,  
 5575, 5661, 5693, 5695, 5745, 5888,  
 5889, 5933, 5934, 6173, 6185, 6254,  
 6327, 6339, 6375, 6376, 6380, 6383,  
 6390, 6426, 6465, 6539, 6545, 6555,  
 6618, 6620, 6645, 6646, 6652, 6661,  
 6709, 6711, 6753, 6758, 6770, 6773,  
 6777, 6780, 6785, 6788, 6792, 6795,  
 7233, 7239, 7857, 7858, 7862, 7863,  
 7867, 7868, 7871, 7872, 7875, 7876,  
 7879, 7880, 7883, 7884, 7899, 8356,  
 8357, 8469, 8500, 8501, 8561, 8850,  
 8904, 8911, 8936, 8958, 8989, 9047,  
 9048, 9195, 9201, 9202, 9258, 9264  
 $\backslash$ tl\_set\_eq:NN . . . . . 444, 465, 467,  
 472, 1567, 2520, 3083, 3084, 3308,  
 3309, 3349, 3350, 3500, 3520, 3544,  
 3617, 3637, 3658, 3751, 3752, 3753,  
 3754, 3982, 4200, 4268, 4280, 4304,  
 4452, 4460, 4505, 4521, 4522, 4523,  
 4524, 4567, 4804, 4813, 4904, 4909,  
 4913, 4993, 5008, 5440, 5580, 5621,  
 5700, 6471, 6543, 6625, 6650, 6908,  
 7129, 8427, 8431, 8432, 9256, 9257  
 $\backslash$ tl\_tail:n . . . . . 648, 649, 694, 2219, 5606, 6435  
 $\backslash$ tl\_to\_str:n 66, 69, 90, 93, 105, 128,  
 135, 196, 207, 208, 209, 210, 211,  
 212, 213, 214, 215, 218, 312, 316,  
 317, 341, 444, 666, 682, 820, 838,  
 840, 843, 862, 868, 919, 1062, 1161,  
 1244, 1259, 1497, 1783, 1808, 1809,  
 1837, 1838, 1844, 1893, 1941, 2110,  
 2131, 2147, 2150, 2151, 2188, 2191,  
 2233, 2235, 2251, 2253, 2257, 2275,  
 2284, 2428, 2430, 2431, 2434, 2435,  
 2465, 2466, 2524, 2592, 2593, 2594,  
 2616, 2849, 3017, 3092, 3367, 3376,  
 3565, 4307, 4619, 4633, 4712, 4757,  
 5546, 6183, 6216, 6266, 6333, 6353,  
 6357, 6516, 6644, 6699, 6729, 6743,  
 6767, 6805, 6841, 7758, 8338, 9126  
 $\backslash$ tl\_trim\_spaces:N . . . . . 49  
 $\backslash$ l\_tmpa\_t1 . . . . . 45, 46, 48,  
 4338, 4759, 4766, 4773, 5190, 5488,  
 5494, 5499, 5501, 5505, 5507, 5510,  
 5519, 5521, 5526, 5529, 5532, 5747,  
 6423, 8911, 8927, 8930, 8989, 9005,

|                            |                                                                |
|----------------------------|----------------------------------------------------------------|
| \use_i:nn . . . . .        | 6632, 9220                                                     |
| \use_ii:nn . . . . .       | 1833, 2947, 2965, 6672                                         |
| \use_none:nnnnnn . . . . . | 3052                                                           |
| \usebox . . . . .          | 8121                                                           |
| \usemodule . . . . .       | 15, 18, 22, 23, 26,<br>37, 49, 93, 94, 96, 212, 517, 524, 3071 |
| \usepackage . . . . .      | 7, 21, 2029, 7964                                              |
| \useSGvar . . . . .        | 109, 8278                                                      |
| \usestructure . . . . .    | 96, 6801                                                       |
| \usetHEME . . . . .        | 7964                                                           |
| \usetikzlibrary . . . . .  | 80, 2109, 9294                                                 |
| <b>V</b>                   |                                                                |
| \varbind . . . . .         | 47,<br>52, 95, 100, 6955, 6978, 7089, 7097                     |
| \vardef . . . . .          | 35,<br>47, 52, 95, 124, 125, 4780, 7120, 7448                  |
| \varempH . . . . .         | 103, 6053                                                      |
| \Varname . . . . .         | 5941                                                           |
| \varname . . . . .         | 5941                                                           |
| \varnotation . . . . .     | 95, 4494                                                       |
| \varref . . . . .          | 5941                                                           |
| \varseq . . . . .          | 43, 95, 4982                                                   |
| \vbox . . . . .            | 135, 7235, 8038, 8127,<br>8142, 8603, 8664, 8713, 8765, 8906   |
| \vbox commands:            |                                                                |
| \vbox_set:Nn . . . . .     | 2156                                                           |
| \vdash . . . . .           | 7587                                                           |
| \vfill . . . . .           | 7893, 9101                                                     |
| \vM . . . . .              | 54                                                             |
| \vMs . . . . .             | 55                                                             |
| \vn . . . . .              | 43                                                             |
| \vop . . . . .             | 45, 47                                                         |
| \vrule . . . . .           | 7235, 8189, 8906                                               |
| \vskip . . . . .           | 7235, 8188, 8190, 8906                                         |
| \vspace . . . . .          | 9103                                                           |
| <b>W</b>                   |                                                                |
| \wff . . . . .             | 20                                                             |
| \withbrackets . . . . .    | 92, 5931, 5939                                                 |
| <b>X</b>                   |                                                                |
| \xspace .                  | 1282, 1291, 3997, 3998, 4004, 4005                             |
| <b>Y</b>                   |                                                                |
| \yesFnoT . . . . .         | 9020                                                           |
| \yesTnoF . . . . .         | 9015                                                           |
| \yield . . . . .           | 7303, 7494, 7497                                               |