The STEX3 Package Collection *

Michael Kohlhase, Dennis Müller FAU Erlangen-Nürnberg

http://kwarc.info/

2022-08-08

Abstract

STEX is a collection of LATEX packages that allow to markup documents semantically without leaving the document format.

Running 'pdflatex' over sTeX-annotated documents formats them into normal-looking PDF. But sTeX also comes with a conversion pipeline into semantically annotated HTML5, which can host semantic added-value services that make the documents active (i.e. interactive and user-adaptive) and essentially turning LATEX into a document format for (mathematical) knowledge management (MKM). STEX augments LATEX with

- semantic macros that denote and distinguish between mathematical concepts, operators, etc. independent of their notational presentation,
- a powerful module system that allows for authoring and importing individual fragments containing document text and/or semantic macros, independent of

 and without hard coding – directory paths relative to the current document,
- a mechanism for exporting STEX documents to (modular) XHTML, preserving all the semantic information for semantically informed knowledge management services

This is the full documentation of STFX. It consists of four parts:

- Part I is a general manual for the STEX package and associated software. It is primarily directed at end-users who want to use STEX to author semantically enriched documents.
- Part II documents the macros provided by the STEX package. It is primarily directed
 at package authors who want to build on STEX, but can also serve as a reference
 manual for end-users.
- Part III documents additional packages that build on STEX, primarily its module system. These are not part of the STEX package itself, but useful additions enabled by STEX package functionality.
- Part IV is the detailled documentation of the STFX package implementation.

^{*}Version 3.2 (last revised 2022-08-08)

Contents

| Ι | $\mathbf{M}_{\mathbf{i}}$ | Manual | | 1 | | | |
|---|---------------------------|------------|--|----|--|--|--|
| 1 | What is STEX? | | | 2 | | | |
| 2 | Qui | Quickstart | | | | | |
| | 2.1 | Setup |) | 3 | | | |
| | | 2.1.1 | Minimal Setup for the PDF-only Workflow | 3 | | | |
| | | 2.1.2 | GIT-based Setup for the STEX Development Version | 3 | | | |
| | | 2.1.3 | STEX Archives (Manual Setup) | 4 | | | |
| | | 2.1.4 | The STEX IDE | 4 | | | |
| | | 2.1.5 | Manual Setup for Active Documents and Knowledge Management | | | | |
| | | | Services | 4 | | | |
| | 2.2 | A Fir | est STEX Document | 5 | | | |
| | | 2.2.1 | OMDoc/xhtml Conversion | 8 | | | |
| | | 2.2.2 | MMT/OMDoc Conversion | 9 | | | |
| | ~ | | | | | | |
| 3 | | - ~ | TEX Content | 10 | | | |
| | 3.1 | | Knowledge is Organized in STEX | 10 | | | |
| | 3.2 | | Archives | 11 | | | |
| | | 3.2.1 | The Local MathHub-Directory | 11 | | | |
| | | 3.2.2 | The Structure of SI _E X Archives | 12 | | | |
| | | 3.2.3 | MANIFEST.MF-Files | 12 | | | |
| | | 3.2.4 | Using Files in STEX Archives Directly | 13 | | | |
| | 3.3 | | ule, Symbol and Notation Declarations | 14 | | | |
| | | 3.3.1 | The smodule-Environment | 14 | | | |
| | | 3.3.2 | Declaring New Symbols and Notations | 16 | | | |
| | | | Operator Notations | 19 | | | |
| | | 3.3.3 | Argument Modes | 20 | | | |
| | | | Mode-b Arguments | 20 | | | |
| | | | Mode-a Arguments | 21 | | | |
| | | | Mode-B Arguments | 22 | | | |
| | | 3.3.4 | Type and Definiens Components | 23 | | | |
| | | 3.3.5 | Precedences and Automated Bracketing | 24 | | | |
| | | 3.3.6 | Variables | 26 | | | |
| | | 3.3.7 | Variable Sequences | 27 | | | |
| | 3.4 | Modu | ile Inheritance and Structures | 28 | | | |
| | | 3.4.1 | Multilinguality and Translations | 28 | | | |
| | | 3.4.2 | Simple Inheritance and Namespaces | 29 | | | |
| | | 3.4.3 | The mathstructure Environment | 31 | | | |
| | | 3.4.4 | The copymodule Environment | 34 | | | |
| | | 3.4.5 | The interpretmodule Environment | 35 | | | |
| | 3.5 | Primi | itive Symbols (The STEX Metatheory) | 36 | | | |
| 4 | Usi | ng eTra | X Symbols | 37 | | | |
| 1 | 4.1 | | ref and its variants | 37 | | | |
| | 4.1 | | ing Un Text and On-the-Fly Notations | 38 | | | |

| 5 | STE | X Stat | ements | 42 | | | | |
|---|-----|----------------|--|--------------|--|--|--|--|
| | 5.1 | | itions, Theorems, Examples, Paragraphs | 42 | | | | |
| | 5.2 | Proof | fs | 45 | | | | |
| | 5.3 | High | lighting and Presentation Customizations | 50 | | | | |
| 6 | Cro | $_{ m ss}$ Ref | erences | 52 | | | | |
| | | | | | | | | |
| 7 | | | l Packages | 54 | | | | |
| | 7.1 | | input: Treating TIKZ code as images | 54 | | | | |
| | 7.2 | | ular Document Structuring | 55 | | | | |
| | | 7.2.1 | Introduction | 55 | | | | |
| | | 7.2.2 | Package Options | 55 | | | | |
| | | 7.2.3 | Document Fragments | 55 | | | | |
| | | 7.2.4 | Ending Documents Prematurely | 57 | | | | |
| | 7.0 | 7.2.5 | Global Document Variables | 57 | | | | |
| | 7.3 | | s and Course Notes | 57 | | | | |
| | | 7.3.1 | Introduction | 57 | | | | |
| | | 7.3.2 | Package Options | 58 | | | | |
| | | 7.3.3 | Notes and Slides | 58 | | | | |
| | | 7.3.4 | Customizing Header and Footer Lines | 59 | | | | |
| | | 7.3.5 | Frame Images | 60 | | | | |
| | 7 4 | 7.3.6 | Excursions | 61 | | | | |
| | 7.4 | - | esenting Problems and Solutions | 62 | | | | |
| | | 7.4.1 | Introduction | 62 | | | | |
| | | 7.4.2 $7.4.3$ | Problems and Solutions | 62 | | | | |
| | | 1.4.5 | Markup for Added-Value Services | 64 | | | | |
| | | | Multiple Choice Blocks | 64 | | | | |
| | | 7 4 4 | Filling-In Concrete Solutions | 65 | | | | |
| | 7.5 | 7.4.4 | Including Problems | 66 67 | | | | |
| | 1.5 | 7.5.1 | eworks, Quizzes and Exams | 67 | | | | |
| | | 7.5.1 $7.5.2$ | Introduction | 67 | | | | |
| | | 7.5.2 $7.5.3$ | Package Options | 67 | | | | |
| | | 7.5.3 $7.5.4$ | 0 | 68 | | | | |
| | | 7.5.4 $7.5.5$ | Including Assignments | 68 | | | | |
| | | 1.0.0 | Typesetting Exams | 00 | | | | |
| Π | D | ocum | entation | 70 | | | | |
| 0 | -00 | v D | | F 1 | | | | |
| 8 | _~ | X-Bas | | 71 71 | | | | |
| | 8.1 | Macr 8.1.1 | os and Environments | 71 | | | | |
| | | 8.1.2 | | 72 | | | | |
| | | 8.1.2 | Babel Languages | 72 | | | | |
| | | | | | | | | |
| 9 | ~ - | - | thHub | 73 | | | | |
| | 9.1 | | ros and Environments | 73 | | | | |
| | | 9.1.1 | Files, Paths, URIs | 73 | | | | |
| | | 9.1.2 | MathHub Archives | 74 75 | | | | |
| | | u i 3 | Light Content in Archives | 75 | | | | |

| 10 ST _E X-References | | 76 |
|---|------|-----------|
| 10.1 Macros and Environments | | 76 |
| 10.1.1 Setting Reference Targets | | 76 |
| 10.1.2 Using References | | 77 |
| 10.1.2 Comg fleterences | | • • • |
| 11 STEX-Modules | | 78 |
| 11.1 Macros and Environments | | 78 |
| 11.1.1 The smodule environment | | 80 |
| 11.1.1 The smodule environment | | 80 |
| 12 STEX-Module Inheritance | | 82 |
| 12.1 Macros and Environments | | 82 |
| | | |
| 12.1.1 SMS Mode | | 82 |
| 12.1.2 Imports and Inheritance | | 83 |
| 19 cm-V Cl -l- | | 05 |
| 13 ST _E X-Symbols | | 85 |
| 13.1 Macros and Environments | | 85 |
| 14 - M V M | | 0,7 |
| 14 STEX-Terms | | 87 |
| 14.1 Macros and Environments | | 87 |
| 1F TO V Ct. 1 I D 1 | | 00 |
| 15 STEX-Structural Features | | 89 |
| 15.1 Macros and Environments | | 89 |
| 15.1.1 Structures | | 89 |
| | | |
| 16 STEX-Statements | | 90 |
| 16.1 Macros and Environments | | 90 |
| | | |
| 17 STEX-Proofs: Structural Markup for Proofs | | 91 |
| 40 77.75 13 | | |
| 18 STEX-Metatheory | | 92 |
| 18.1 Symbols | | 92 |
| | | |
| | | |
| III Extensions | | 93 |
| | | |
| 19 Tikzinput: Treating TIKZ code as images | | 94 |
| 19.1 Macros and Environments | | 94 |
| | | |
| 20 document-structure: Semantic Markup for Open Mathematical De | ocu- | • |
| ments in L ^A T _E X | | 95 |
| | | |
| 21 NotesSlides – Slides and Course Notes | | 96 |
| | | |
| 22 problem.sty: An Infrastructure for formatting Problems | | 97 |
| | _ | |
| 23 hwexam.sty/cls: An Infrastructure for formatting Assignments and | Ex- | |
| ams | | 98 |
| | | |
| | | |
| IV Implementation | | 99 |

| 24 | сТъХ | -Basics Implementation | 100 |
|-----------|------|-------------------------------------|-----|
| | 24.1 | The ST-XDocument Class | |
| | 24.2 | Preliminaries | |
| | 24.3 | Messages and logging | |
| | 24.4 | HTML Annotations | |
| | 24.5 | Babel Languages | |
| | 24.6 | Persistence | |
| | 24.7 | Auxiliary Methods | |
| 25 | сТъХ | -MathHub Implementation | 110 |
| _0 | 25.1 | Generic Path Handling | |
| | 25.2 | PWD and kpsewhich | |
| | 25.3 | File Hooks and Tracking | |
| | 25.4 | MathHub Repositories | |
| | | Using Content in Archives | |
| 26 | сТъХ | -References Implementation | 124 |
| -0 | 26.1 | Document URIs and URLs | |
| | 26.2 | Setting Reference Targets | |
| | 26.3 | Using References | |
| | | | |
| 27 | STEX | -Modules Implementation | 136 |
| | 27.1 | The smodule environment | |
| | 27.2 | Invoking modules | 146 |
| 28 | STEX | -Module Inheritance Implementation | 148 |
| | 28.1 | SMS Mode | 148 |
| | 28.2 | Inheritance | 152 |
| 29 | сТъХ | -Symbols Implementation | 158 |
| | 29.1 | Symbol Declarations | |
| | 29.2 | Notations | |
| | | Variables | |
| 30 | сТъХ | -Terms Implementation | 183 |
| 00 | 30.1 | Symbol Invocations | |
| | 30.2 | Terms | |
| | 30.3 | Notation Components | |
| | 30.4 | Variables | |
| | 30.5 | Sequences | |
| 31 | «TeX | -Structural Features Implementation | 202 |
| | 31.1 | Imports with modification | 203 |
| | 31.2 | The feature environment | |
| | 31.3 | Structure | |
| 32 | «TeX | -Statements Implementation | 222 |
| | 32.1 | Definitions | 222 |
| | 32.2 | Assertions | 228 |
| | 32.3 | Examples | _ |
| | 32.4 | Logical Paragraphs | _ |
| | | | |

| 33 | | Implementation Proofs | 239 239 |
|-----------|--------------|------------------------------------|----------------|
| 34 | STE | G-Others Implementation | 248 |
| 35 | STE | K-Metatheory Implementation | 250 |
| 36 | Tikz | input Implementation | 253 |
| 37 | docı | ument-structure.sty Implementation | 256 |
| | 37.1 | Package Options | |
| | 37.2 | Document Structure | |
| | 37.3 | Front and Backmatter | |
| | 37.4 | Global Variables | 263 |
| 38 | Note | ${ m esSlides-Implementation}$ | 264 |
| | 38.1 | Class and Package Options | |
| | 38.2 | Notes and Slides | |
| | 38.3 | Header and Footer Lines | |
| | 38.4 | Frame Images | |
| | 38.5 | Sectioning | |
| | 38.6 | Excursions | 276 |
| 39 | The | Implementation | 278 |
| | 39.1 | Package Options | 278 |
| | 39.2 | Problems and Solutions | 279 |
| | 39.3 | Marup for Added Value Services | 286 |
| | 39.4 | Multiple Choice Blocks | 286 |
| | 39.5 | Filling in Concrete Solutions | 287 |
| | 39.6 | Including Problems | 287 |
| | 39.7 | Reporting Metadata | 289 |
| 40 | Imp | lementation: The hwexam Package | 291 |
| | $40.\bar{1}$ | Package Options | 291 |
| | 40.2 | Assignments | 292 |
| | 40.3 | Including Assignments | 295 |
| | 40.4 | Typesetting Exams | 296 |
| | 40.5 | Leftovers | 298 |
| 41 | Refe | erences | 299 |

Part I Manual



Boxes like this one contain implementation details that are mostly relevant for more advanced use cases, might be useful to know when debugging, or might be good to know to better understand how something works. They can easily be skipped on a first read.



 $\begin{array}{l} \overset{\longleftarrow}{M} \xrightarrow{\longrightarrow} \text{Boxes like this one explain how some STeX concept relates to the MMT/OMDoc} \\ \overset{\longleftarrow}{M} \xrightarrow{\longrightarrow} \text{system, philosophy or language; see [MMT; Koh06] for introductions.} \end{array}$

Chapter 1

What is STEX?

Formal systems for mathematics (such as interactive theorem provers) have the potential to significantly increase both the accessibility of published knowledge, as well as the confidence in its veracity, by rendering the precise semantics of statements machine actionable. This allows for a plurality of added-value services, from semantic search up to verification and automated theorem proving. Unfortunately, their usefulness is hidden behind severe barriers to accessibility; primarily related to their surface languages reminiscent of programming languages and very unlike informal standards of presentation.

STEX minimizes this gap between informal and formal mathematics by integrating formal methods into established and widespread authoring workflows, primarily LATEX, via non-intrusive semantic annotations of arbitrary informal document fragments. That way formal knowledge management services become available for informal documents, accessible via an IDE for authors and via generated *active* documents for readers, while remaining fully compatible with existing authoring workflows and publishing systems.

Additionally, an extensible library of reusable document fragments is being developed, that serve as reference targets for global disambiguation, intermediaries for content exchange between systems and other services.

Every component of the system is designed modularly and extensibly, and thus lay the groundwork for a potential full integration of interactive theorem proving systems into established informal document authoring workflows.

The general ST_EX workflow combines functionalities provided by several pieces of software:

- $\bullet\,$ The STEX package collection to use semantic annotations in LATEX documents,
- RusTeX [RT] to convert tex sources to (semantically enriched) xhtml,
- The MMT system [MMT], that extracts semantic information from the thus generated xhtml and provides semantically informed added value services. Notably, MMT integrates the RusTeX system already.

Chapter 2

Quickstart

2.1 Setup

There are two ways of using STEX: as a

- 1. way of writing LATEX more modularly (object-oriented Math) for creating PDF documents or
- 2. foundation for authoring active documents in HTML5 instrumented with knowledge management services.

Both are legitimate and useful. The first requires a significantly smaller tool-chain, so we describe it first. The second requires a much more substantial (and experimental) toolchain of knowledge management systems. Both workflows profit from an integrated development environment (IDE), which (also) automates setup as far as possible (see subsection 2.1.4).

2.1.1 Minimal Setup for the PDF-only Workflow

In the best of all worlds, there is no setup, as you already have a new version of TEXLive on your system as a LATEX enthusiast. If not now is the time to install it; see [TL]. You can usually update TEXLive via a package manager or the TEXLive manager tlmgr.

Alternatively, you can install ST_EX from CTAN, the Comprehensive T_EX Archive Network; see [ST] for details.

2.1.2 GIT-based Setup for the STFX Development Version

If you want use the latest and greatest STEX packages that have not even been released to CTAN, then you can directly clone them from the STEX development repository [sTeX] by the following command-line instructions:

```
cd <stexdir>
git clone https://github.com/slatex/sTeX.git
```

and keep it updated by pulling updates via git pull in the cloned STEX directory. Then update your TEXINPUTS environment variable, e.g. by placing the following line in your .bashrc:

```
export TEXINPUTS="$(TEXINPUTS):<sTeXDIR>//:"
```

2.1.3 STEX Archives (Manual Setup)

Writing semantically annotated STEX becomes much easier, if we can use well-designed libraries of already annotated content. STEX provides such libraries as STEX archives—i.e. GIT repositories at https://gl.mathhub.info—most prominently the SMGLoM libraries at https://gl.mathhub.info/smglom.

To do so, we set up a **local MathHub** by creating a MathHub directory <mhdir>. Every STEX archive as an **archive path** <apath> and a name <archive>. We can clone the STEX archive by the following command-line instructions:

```
cd <mhdir>/<apath>
git clone https://gl.mathhub.info/smglom/<archive>.git
```

Note that STEX archives often depend on other archives, thus you should be prepared to clone these as well – e.g. if pdflatex reports missing files. To make sure that STEX too knows where to find its archives, we need to set a global system variable MATHHUB, that points to your local MathHub-directory (see section 3.2).

export MATHHUB="<mhdir>",

2.1.4 The STEX IDE

We are currently working on an STEX IDE as an STEX plugin for VScode; see [SIa]. It will feature a setup procedure that automates the setup described above (and below). For additional functionality see the (now obsolete) plugin for STEX 1 [SLS; SIb].

2.1.5 Manual Setup for Active Documents and Knowledge Management Services

Foregoing on the STEX IDE, we will need several additional (on top of the minimal setup above) pieces of software; namely:

- The Mmt System available here. We recommend following the setup routine documented here.
 - Following the setup routine (Step 3) will entail designating a MathHub-directory on your local file system, where the MMT system will look for STEX/MMT content archives.
- STEX Archives If we only care about LATEX and generating pdfs, we do not technically need MMT at all; however, we still need the MATHHUB system variable to be set. Furthermore, MMT can make downloading content archives we might want to use significantly easier, since it makes sure that all dependencies of (often highly interrelated) STEX archives are cloned as well.
 - Once set up, we can run mmt in a shell and download an archive along with all of its dependencies like this: lmh install <name-of-repository>, or a whole group of archives; for example, lmh install smglom will download all smglom archives.
- RusTeX The Mmt system will also set up RusTeX for you, which is used to generate (semantically annotated) xhtml from tex sources. In lieu of using Mmt, you can also download and use RusTeX directly here.

2.2 A First STEX Document

Having set everything up, we can write a first STEX document. As an example, we will use the smglom/calculus and smglom/arithmetics archives, which should be present in the designated MathHub-folder, and write a small fragment defining the *geometric series*:

```
1 \documentclass{article}
 2 \usepackage{stex,xcolor,stexthm}
4 \begin{document}
5 \begin{smodule}{GeometricSeries}
       \importmodule(smglom/calculus){series}
      \importmodule[smglom/arithmetics]{realarith}
8
9
      \symdef{geometricSeries}[name=geometric-series]{\comp{S}}
      \begin{sdefinition} [for=geometricSeries]
11
          The \definame{geometricSeries} is the \symname{?series}
12
13
          \[\defeq{\geometricSeries}{\definiens{
              \displaystyle \inf \{ \sup \{ svar\{n\} \} \} \} 
                  \realdivide[frac]{1}{
16
                      \realpower{2}{\svar{n}}
17
              }}
18
          }}.\]
19
      \end{sdefinition}
20
21
      \begin{sassertion} [name=geometricSeriesConverges, type=theorem]
      The \symname{geometricSeries} \symname{converges} towards $1$.
      \end{sassertion}
24 \end{smodule}
25 \end{document}
```

Compiling this document with pdflatex should yield the output

Definition 0.1. The **geometric series** is the series

$$S := \sum_{n=1}^{\infty} \frac{1}{2^n}.$$

Theorem 0.2. The geometric series converges towards 1.

Move your cursor over the various highlighted parts of the document – depending on your pdf viewer, this should yield some interesting (but possibly for now cryptic) information.

Remark 2.2.1:

Note that all of the highlighting, tooltips, coloring and the environment headers come from stexthm – by default, the amount of additional packages loaded is kept to a minimum and all the presentations can be customized, see ??.

Let's investigate this document in detail to understand the respective parts of the ST_FX markup infrastructure:

```
smodule \begin{smodule}{GeometricSeries}
...
\end{smodule}
```

First, we open a new *module* called GeometricSeries. The main purpose of the smodule environment is to group the contents and associate it with a *globally unique* identifier (URI), which is computed from the name GeometricSeries and the document context.

(Depending on your pdf viewer), the URI should pop up in a tooltip if you hover over the word **geometric series**.

\importmodule

```
\importmodule[smglom/calculus]{series}
\importmodule[smglom/arithmetics]{realarith}
```

Next, we *import* two modules — series from the STEX archive smglom/calculus, and realarith from the STEX archive smglom/arithmetics. If we investigate these archives, we find the files series.en.tex and realarith.en.tex (respectively) in their respective source-folders, which contain the statements \begin{smodule}{series} and \begin{smodule}{realarith} (respectively).

The \importmodule-statements make all STEX symbols and associated semantic macros (e.g. \infinitesum, \realdivide, \realpower) in the imported module available to the current module GeometricSeries. The module GeometricSeries "exports" all of these symbols to all modules imports it via an \importmodule (GeometricSeries) instruction. Additionally it exports the local symbol \geometricSeries.

\usemodule

If we only want to *use* the content of some module Foo, e.g. in remarks or examples, but none of the symbols in our current module actually *depend* on the content of Foo, we can use \usemodule instead – like \importmodule, this will make the module content available, but will *not* export it to other modules.

\symdef

```
\symdef{GeometricSeries}[name=geometric-series]{\comp{S}}
```

Next, we introduce a new symbol with name geometric-series and assign it the semantic macro \geometricSeries. \symdef also immediately assigns this symbol a notation, namely S.

\comp

The macro \comp marks the S in the notation as a notational component, as opposed to e.g. arguments to \geometricSeries. It is the notational components that get highlighted and associated with the corresponding symbol (i.e. in this case geometricSeries). Since \geometricSeries takes no arguments, we can wrap the whole notation in a \comp.

```
\begin{sdefinition} [for=geometricSeries]
...
\end{sdefinition}
\begin{sassertion} [name=geometricSeriesConverges, type=theorem]
...
\end{sassertion}
```

What follows are two STEX-statements (e.g. definitions, theorems, examples, proofs, ...). These are semantically marked-up variants of the usual environments, which take additional optional arguments (e.g. for=, type=, name=). Since many LATEX templates predefine environments like definition or theorem with different syntax, we use sdefinition, sassertion, sexample etc. instead. You can customize these environments to e.g. simply wrap around some predefined theorem-environment. That way, we can still use sassertion to provide semantic information, while being fully compatible with (and using the document presentation of) predefined environments.

In our case, the stexthm-package patches e.g. \begin{sassertion}[type=theorem] to use a theorem-environment defined (as usual) using the amsthm package.

\symname

... is the \symname{?series}

The \symname-command prints the name of a symbol, highlights it (based on customizable settings) and associates the text printed with the corresponding symbol.

Note that the argument of \symref can be an imported symbol (here the series symbol is imported from the series module). STEX tries to determine the full symbol URI from the argument. If there are name clashes in or with the imported symbols, the name of the exporting module can be prepended to the symbol name before the ? character.

If you hover over the word series in the pdf output, you should see a tooltip showing the full URI of the symbol used.

\symref

The \symname-command is a special case of the more general \symref-command, which allows customizing the precise text associated with a symbol. \symref takes two arguments: the first ist the symbol name (or macro name), and the second a variant verbalization of the symbol, e.g. an inflection variant, a different language or a synonym. In our example \symname{?series} abbreviates \symref{?series}{series}.

\definame \definiendum

```
The \definame{geometricSeries} ...
```

The sdefinition-environment provides two additional macros, \definame and \definiendum which behave similarly to \symname and \symref, but explicitly mark the symbols as being defined in this environment, to allow for special highlighting.

```
\[\defeq{\geometricSeries}{\definiens{
   \infinitesum{\svar{n}}{1}{
      \realdivide[frac]{1}{
      \realpower{2}{\svar{n}}
   }}
}}.\]
```

The next snippet – set in a math environment – uses several semantic macros imported from (or recursively via) series and realarithmetics, such as \defeq , \infinitesum , etc. In math mode, using a semantic macro inserts its (default) definition. A semantic macro can have several notations – in that case, we can explicitly choose a specific notation by providing its identifier as an optional argument; e.g. $\realdivide[frac]{a}{b}$ will use the explicit notation named $\frac{frac}{frac}$ of the semantic macro \realdivide , which yields $\frac{a}{b}$ instead of $\frac{a}{b}$.

\svar

The \svar{n} command marks up the n as a variable with name n and notation n.

\definiens

The sdefinition-environment additionally provides the \definiens-command, which allows for explicitly marking up its argument as the definiens of the symbol currently being defined.

2.2.1 OMDoc/xhtml Conversion

So, if we run pdflatex on our document, then STEX yields pretty colors and tooltips¹. But STEX becomes a lot more powerful if we additionally convert our document to xhtml while preserving all the STEX markup in the result.

TODO VSCode Plugin

Using RusTeX [RT], we can convert the document to xhtml using the command rustex -i /path/to/file.tex -o /path/to/outfile.xhtml. Investigating the resulting file, we notice additional semantic information resulting from our usage of semantic macros, \symref etc. Below is the (abbreviated) snippet inside our \definiens block:

```
<mrow resource="" property="stex:definiens">
  <mrow resource="...?series?infinitesum" property="stex:OMBIND">
  <munderover displaystyle="true">
   <mo resource="...?series?infinitesum" property="stex:comp">∑</mo>
    <mrow resource="1" property="stex:arg">
     <mi resource="var://n" property="stex:OMV">n</mi>
    </mrow>
    <mo resource="...?series?infinitesum" property="stex:comp">=</mo>
    <mi resource="2" property="stex:arg">1</mi>
   </mrow>
   <mi resource="...?series?infinitesum" property="stex:comp">\infty</pi>
  </munderover>
  <mrow resource="3" property="stex:arg">
<mrow resource="3" property="stex:arg">
<mfrac resource="...?realarith?division#frac#" property="stex:OMA">
    <mi resource="1" property="stex:arg">1</mi>
<mrow resource="2" property="stex:arg">
<msup resource="...realarith?exponentiation" property="stex:OMA">

       <mi resource="1" property="stex:arg">2</mi>
<mrow resource="2" property="stex:arg">
        <mi resource="var://n" property="stex:OMV">n</mi>
       </mrow>
      </msup>
    </mrow>
   </mfrac>
  </mrow>
</mrow>
</mrow>
```

...containing all the semantic information. The MMT system can extract from this the following OpenMath snippet:

```
<OMBIND>
<OMID name="...?series?infinitesum"/>
<OMV name="n"/>
```

^{1...}and hyperlinks for symbols, and indices, and allows reusing document fragments modularly, and...

```
<OMLIT name="1"/>
<OMA>
  <OMS name="...?realarith?division"/>
  <OMLIT name="1"/>
  <OMA>
       <OMS name="...realarith?exponentiation"/>
       <OMLIT name="2"/>
       <OMV name="n"/>
       </OMA>
  </OMA>
</OMBIND>
```

...giving us the full semantics of the snippet, allowing for a plurality of knowledge management services – in particular when serving the xhtml.

Remark 2.2.2:

Note that the html when opened in a browser will look slightly different than the pdf when it comes to highlighting semantic content – that is because naturally html allows for much more powerful features than pdf does. Consequently, the html is intended to be served by a system like MMT, which can pick up on the semantic information and offer much more powerful highlighting, linking and similar features, and being customizable by readers rather than being prescribed by an author.

Additionally, not all browsers (most notably Chrome) support MATHML natively, and might require additional external JavaScript libraries such as MathJax to render mathematical formulas properly.

2.2.2 Mmt/OMDoc Conversion

Another way to convert our document to actual MMT/OMDOC is to put it in an STEX archive (see ??) and have MMT take care of everything.

Assuming the above file is source/demo.tex in an STEX archive MyTest, you can run MMT and do build MyTest stex-omdoc demo.tex to convert the document to both xhtml (which you will find in xhtml/demo.xhtml in the archive) and formal MMT/OMDoc, which you can subsequently view in the MMT browser (see https://uniformal.github.io//doc/applications/server.html#the-mmt-web-site for details).

Chapter 3

Creating STeX Content

We can use STEX by simply including the package with \usepackage{stex}, or - primarily for individual fragments to be included in other documents - by using the STEX document class with \documentclass{stex} which combines the standalone document class with the stex package.

Both the stex package and document class offer the following options:

lang $(\langle language \rangle *)$ Languages to load with the babel package.

mathhub ($\langle directory \rangle$) MathHub folder to search for repositories – this is not necessary if the MATHHUB system variable is set.

writesms (\langle boolean \rangle) with this package option, STEX will write the contents of all external modules imported via \importmodule or \usemodule into a file \jobname.sms (analogously to the table of contents .toc-file).

usems (\langle boolean \rangle) subsequently tells STEX to read the generated sms-file at the beginning of the document. This allows for e.g. collaborating on documents without all authors having to have all used archives and modules available – one author can load the modules with writesms, and the rest can use the the modules with usesms. Furthermore, the sms file can be submitted alongside a tex-file, effectively making it "standalone".

image $(\langle boolean \rangle)$ passed on to tikzinput.

debug $(\langle log\text{-}prefix\rangle *)$ Logs debugging information with the given prefixes to the terminal, or all if all is given. Largely irrelevant for the majority of users.

3.1 How Knowledge is Organized in STEX

STFX content is organized on multiple levels:

- 1. STEX archives (see section 3.2) contain individual .tex-files.
- 2. These may contain ST_EX modules, introduced via $\begin{smodule}{\bf Smodule}{\bf Smod$

- 3. Modules contain STEX symbol declarations, introduced via \symdecl{symbolname}, \symdef{symbolname} and some other constructions. Most symbols have a notation that can be used via a semantic macro \symbolname generated by symbol declarations.
- 4. STFX expressions finally are built up from usages of semantic macros.



- STEX archives are simultaneously MMT archives, and the same directory structure is consequently used.
- STEX modules correspond to OMDoc/MMT theories. \importmodules (and similar constructions) induce MMT includes and other theory morphisms, thus giving rise to a theory graph in the OMDoc sense [RK13].
- Symbol declarations induce OMDoc/Mmt constants, with optional (formal) type and definiens components.
- Finally, STEX expressions are converted to OMDoc/MMT terms, which use the abstract syntax (and XML encoding) of OPENMATH [Bus+04].

3.2 STEX Archives

3.2.1 The Local MathHub-Directory

\userodule, \importmodule, \inputref etc. allow for including content modularly without having to specify absolute paths, which would differ between users and machines. Instead, STEX uses archives that determine the global namespaces for symbols and statements and make it possible for STEX to find content referenced via such URIs.

All STEX archives need to exist in the local MathHub-directory. STEX knows where this folder is via one of four 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. Finally, if all else 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.

3.2.2 The Structure of STEX Archives

An STEX archive group/name is stored in the directory /path/to/mathhub/group/name; e.g. assuming your local MathHub-directory is set as /user/foo/MathHub, then in order for the smglom/calculus-archive to be found by the STEX system, it needs to be in /user/foo/MathHub/smglom/calculus.

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 where STEX will look for files included via \\libinput.

Additionally a *group* of archives group/name may have an additional archive group/meta-inf. If this meta-inf-archive has a /lib-subdirectory, it too will be searched by \libinput from all tex files in any archive in the group/*-group.

We recommend the following additional directory structure in the **source**-folder of an ST_EX archive:

- /source/mod/ individual STEX modules, containing symbol declarations, notations, and \begin{sparagraph} [type=symdoc,for=...] environments for "encyclopaedic" symbol documentations
- /source/def/ definitions
- /source/ex/ examples
- /source/thm/ theorems, lemmata and proofs; preferably proofs in separate files to allow for multiple proofs for the same statement
- /source/snip/ individual text snippets such as remarks, explanations etc.
- /source/frag/ individual document fragments, ideally only \inputrefing snippets, definitions, examples etc. in some desirable order
- /source/tikz/ tikz images, as individual .tex-files
- /source/PIC/ image files.

3.2.3 MANIFEST.MF-Files

The MANIFEST.MF in the META-INF-directory consists of key-value-pairs, informing STEX (and associated software) of various properties of an archive. For example, the MANIFEST.MF of the smglom/calculus-archive looks like this:

teaser: Terminology for the mathematical study of change.

description: desc.html

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

id: The name of the archive, including its group (e.g. smglom/calculus),

source-base or

ns: The namespace from which all symbol and module URIs in this repository are formed, see (TODO),

narration-base: The namespace from which all document URIs in this repository are formed, see (TODO),

url-base: The URL that is formed as a basis for external references, see (TODO),

dependencies: All archives that this archive depends on. SIEX ignores this field, but MMT can pick up on them to resolve dependencies, e.g. for lmh install.

3.2.4 Using Files in STEX Archives Directly

Several macros provided by STEX allow for directly including files in repositories. These are:

\mhinput

\mhinput [Some/Archive] {some/file} directly inputs the file some/file in the source-folder of Some/Archive.

\inputref

\inputref[Some/Archive]{some/file} behaves like \mhinput, but wraps the input in a \begingroup ... \endgroup. When converting to xhtml, the file is not input at all, and instead an html-annotation is inserted that references the file, e.g. for lazy loading. In the majority of practical cases \inputref is likely to be preferred over \mhinput because it leads to less duplication in the generated xhtml.

\ifinput

Both \minput and \inputref set \iffinput to "true" during input. This allows for selectively including e.g. bibliographies only if the current file is not being currently included in a larger document.

\addmhbibresource

\addmhbibresource [Some/Archive] {some/file} searches for a file like \mhinput does, but calls \addbibresource to the result and looks for the file in the archive root directory directly, rather than the source directory. Typical invocations are

- \addmhbibresource{lib/refs.bib}, which specifies a bibliography in the lib folder in the local archive or
- \addmhbibresource[HW/meta-inf]{lib/refs.bib} in another.

\libinput

\libinput{some/file} searches for a file some/file 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 include all found files in reverse order; e.g. \libinput{preamble} in a .tex-file in smglom/calculus will first input .../smglom/meta-inf/lib/preamble.tex and then ../smglom/calculus/lib/preamble.tex.

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

\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 is found.

Remark 3.2.1:

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

1 \documentclass{stex}
2 \libinput{preamble}
3 \begin{document}
...
5 \iffinputref \else \libinput{postamble} \fi
6 \end{document}

Then the preamble.tex files can take care of loading the generally required pack-
```

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 postamble.tex can e.g. print the bibliography, index etc.

\lambda libusepackage is particularly useful in preamble.tex when we want to use custom packages that are not part of TeXLive. In this case we commit the respective packages in one of the lib folders and use \libusepackage to load them.

3.3 Module, Symbol and Notation Declarations

3.3.1 The smodule-Environment

smodule 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 ${\tt smodule}$ -environment takes several keyword arguments, all of which are optional:

title $(\langle token \ list \rangle)$ to display in customizations.

```
type (\langle string \rangle *) for use in customizations.
```

deprecate $(\langle module \rangle)$ if set, will throw a warning when loaded, urging to use $\langle module \rangle$ instead.

id $(\langle string \rangle)$ for cross-referencing.

ns $(\langle URI \rangle)$ the namespace to use. Should not be used, unless you know precisely what you're doing. If not explicitly set, is computed using $\text{stex_modules_current_namespace:}$.

lang $(\langle language \rangle)$ if not set, computed from the current file name (e.g. foo.en.tex).

sig (\language\rangle) if the current file is a translation of a file with the same base name but a
different language suffix, setting sig=<lamp> 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.

creators ($\langle string \rangle *$) names of the creators.

contributors ($\langle string \rangle *$) names of contributors.

srccite ($\langle string \rangle$) a source citation for the content of this module.

By default, opening a module will produce no output whatsoever, e.g.:

Example 1

Input:

3 \end{smodule}

Output:

Hello World

 $\$ stexpatchmodule

We can customize this behavior either for all modules or only for modules with a specific type using the command \stexpatchmodule[optional-type]{begin-code}{end-code}. Some optional parameters are then available in \smodule*-macros, specifically \smoduletitle, \smoduletype and \smoduleid.

For example:

Example 2

Input:

```
1 \stexpatchmodule[display]
2 {\textbf{Module (\smoduletitle)}\par}
3 {\par\noindent\textbf{End of Module (\smoduletitle)}}
4
5 \begin{smodule}[type=display,title={Some New Module}]{SomeModule2}
6 Hello World
7 \end{smodule}

Output:

Module (Some New Module)
Hello World
End of Module (Some New Module)
```

3.3.2 Declaring New Symbols and Notations

Inside an smodule environment, we can declare new STEX symbols.

\symdecl

The most basic command for doing so is using \symdecl{symbolname}. This introduces a new symbol with name symbolname, arity 0 and semantic macro \symbolname.

The starred variant \symdecl*{symbolname} will declare a symbol, but not introduce a semantic macro. If we don't want to supply a notation (for example to introduce concepts like "abelian", which is not something that has a notation), the starred variant is likely to be what we want.

```
\stackrel{\longleftarrow}{M} \symdecl introduces a new OMDoc/MMT constant in the current mod—\stackrel{\longleftarrow}{M} → ule (=OMDoc/MMT theory). Correspondingly, they get assigned the URI \stackrel{\longleftarrow}{N} \square module-URI>?<constant-name>.
```

Without a semantic macro or a notation, the only meaningful way to reference a symbol is via \symref,\symname etc.

```
Example 3
Input:

1 \symdecl*{foo}
2 Given a \symname{foo}, we can...

Output:

Given a, we can...
```

Obviously, most semantic macros should take actual *arguments*, implying that the symbol we introduce is an *operator* or *function*. We can let \symdecl know the *arity* (i.e. number of arguments) of a symbol like this:

Example 4

Input:

```
1 \symdecl{binarysymbol}[args=2]
2 \symref{binarysymbol}{this} is a symbol taking two arguments.
```

Output:

is a symbol taking two arguments.

.

So far we have gained exactly ... nothing by adding the arity information: we cannot do anything with the arguments in the text.

We will now see what we can gain with more machinery.

\notation

We probably want to supply a notation as well, in which case we can finally actually use the semantic macro in math mode. We can do so using the \notation command, like this:

Example 5

Input:

```
1 \notation{binarysymbol}{\text{First: }#1\text{; Second: }#2}
2 $\binarysymbol{a}{b}$
```

Output:

```
First: a; Second: b
```

.

```
\begin{tabular}{ll} $\leftarrow M \line Applications of semantic macros, such as $$ \binarysymbol{a}{b} \ are translated to $$-M \line MMT/OMDOC as OMA-terms with head <OMS name="...?binarysymbol"/>. $$ Semantic macros with no arguments correspond to OMS directly. $$
```

\comp

For many semantic services e.g. semantic highlighting or **wikification** (linking uservisible notation components to the definition of the respective symbol they come from), we need to specify the notation components. Unfortunately, there is currently no way the STEX engine can infer this by itself, so we have to specify it manually in the notation specification. We can do so with the \comp command.

We can introduce a new notation highlight for \binarysymbol that fixes this flaw, which we can subsequently use with \binarysymbol[highlight]:

Example 6

Input:

```
1 \notation{binarysymbol}[highlight]
2      {\comp{\text{First: }}#1\comp{\text{; Second: }}#2}
3 $\binarysymbol[highlight]{a}{b}$
```

Output:

ab



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 arguments 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 $\addition\{a\}\{b\}$ taking two arguments 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 \addition .



Similarly, a semantic macro can not conceptually be part of the notation of \addition, since a semantic macro represents a distinct mathematical concept with its own semantics, whereas 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), and might want to give those another thought - quite likely, the macro 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.

\symdef

In the vast majority of cases where a symbol declaration should come with a semantic macro, we will want to supply a notation immediately. For that reason, the \symdef command combines the functionality of both \symdecl and \notation with the optional arguments of both:

Example 7

Input:

```
1 \symdef{newbinarysymbol}[h1,args=2]
2     {\comp{\text{1.: }}#1\comp{\text{; 2.: }}#2}
3 $\newbinarysymbol{a}{b}$
```

Output:

ab

We just declared a new symbol newbinarysymbol with args=2 and immediately provided it with a notation with identifier hl. Since hl is the *first* (and so far, only) notation supplied for newbinarysymbol, using \newbinarysymbol without optional argument defaults to this notation.

But one man's meat is another man's poison: it is very subjective what the "default notation" of an operator should be. Different communities have different practices. For instance, the complex unit is written as i in Mathematics and as j in electrical engineering. So to allow modular specification and facilitate re-use of document fragments STEX allows to re-set notation defaults.

\setnotation

The first notation provided will stay the default notation unless explicitly changed — this is enabled by the \setnotation command: \setnotation{symbolname} {notation-id} sets the default notation of \symbolname to notation-id, i.e. henceforth, \symbolname behaves like \symbolname[notation-id] from now on.

Often, a default notation is set right after the corresponding notation is introduced – the starred version \notation* for that reason introduces a new notation and immediately sets it to be the new default notation. So expressed differently, the first \notation for a symbol behaves exactly like \notation*, and \notation*{foo}[bar]{...} behaves exactly like \notation{foo}{bar}.

\textsymdecl

In the less mathematical settings where we want a symbol and semantic macro for some concept with a notation beyond its mere name, but which should also be available in TEX's text mode, the command \textsymdecl is useful. For example, we can declare a symbol openmath with the notation \textsc{OpenMath} using \textsymdecl{openmath}[name=OpenMath]{\textsc{OpenMath}}. The \openmath yields both in text and math mode.

Operator Notations

Once we have a semantic macro with arguments, such as \newbinarysymbol, the semantic macro represents the application of the symbol to a list of arguments. What if we want to refer to the operator itself, though?

We can do so by supplying the \notation (or \symdef) with an operator notation, indicated with the optional argument op=. We can then invoke the operator notation

using \symbolname! [notation-identifier]. Since operator notations never take arguments, we do not need to use \comp in it, the whole notation is wrapped in a \comp automatically:

Example 8

Input:

1 \notation{newbinarysymbol}[ab, op={\text{a:}\cdot\text{; b:}\cdot}]
2 {\comp{\text{a:}}#1\comp{\text{; b:}}#2} \symname{newbinarysymbol} is also
3 occasionally written \$\newbinarysymbol![ab]\$

Output:

is also occasionally written

3.3.3 Argument Modes

The notations so far used <code>simple</code> arguments which we call <code>mode-i</code> arguments. Declaring a new symbol with <code>\symdecl{foo}[args=3]</code> is equivalent to writing <code>\symdecl{foo}[args=iii]</code>, indicating that the semantic macro takes three mode-i arguments. However, there are three more argument modes which we will investigate now, namely mode-b, mode-a and mode-B arguments.

Mode-b Arguments

A mode-b argument represents a variable that is bound by the symbol in its application, making the symbol a binding operator. Typical examples of binding operators are e.g. sums \sum , products \prod , integrals \int , quantifiers like \forall and \exists , that λ -operator, etc.

```
\stackrel{\longleftarrow}{M} → Mode-b arguments behave exactly like mode-i arguments within T_EX, but appli—M → cations of binding operators, i.e. symbols with mode-b arguments, are translated \stackrel{\longleftarrow}{N} to OMBIND-terms in OMDoc/MMT, rather than OMA.
```

For example, we can implement a summation operator binding an index variable and taking lower and upper index bounds and the expression to sum over like this:

Example 9

Input:

```
1 \symdef{summation}[args=biii]
2 {\mathop{\comp{\sum}}_{#1\comp{=}#2}^{#3}#4}
3 $\summation{\svar{x}}{1}{\svar{n}}{\svar{x}}^2$
```

Output:

```
\frac{n}{x1}x^2
```

.

where the variable x is now bound by the \summation-symbol in the expression.

Mode-a Arguments

Mode-a arguments represent a *flexary argument sequence*, i.e. a sequence of arguments of arbitrary length. Formally, operators that take arbitrarily many arguments don't "exist", but in informal mathematics, they are ubiquitous. Mode-a arguments allow us to write e.g. \addition{a,b,c,d,e} rather than having to write something like \addition{a}{\addition{b}{\addition{b}}}!

\notation (and consequently \symdef, too) take one additional argument for each mode-a argument that indicates how to "accumulate" a comma-separated sequence of arguments. This is best demonstrated on an example.

Let's say we want an operator representing quantification over an ascending chain of elements in some set, i.e. $\ascendingchain{S}{a,b,c,d,e}{t}$ should yield $\forall a < sb < sc < sd < se .t.$ The "base"-notation for this operator is simply

 ${\operatorname{1}} \#2\operatorname{2},\$, where #2 represents the full notation fragment *accumulated* from {a,b,c,d,e}.

The additional argument to \notation (or \symdef) takes the same arguments as the base notation and two additional arguments ##1 and ##2 representing successive pairs in the mode-a argument, and accumulates them into #2, i.e. to produce $a <_S b <_S c <_S d <_S e$, we do {##1 \comp{<}_{#1} ##2}:

Example 10

Input:

```
1 \symdef{ascendingchain}[args=iai]
2 {\comp{\forall} #2\comp{.\,}#3}
3 {##1 \comp{<}_{#1} ##2}
5 Tadaa: $\ascendingchain{S}{a,b,c,d,e}{t}$</pre>
```

Output:

```
Tadaa: a_S b_S c_S d_S e t
```

.

If this seems overkill, keep in mind that you will rarely need the single-hash arguments #1,#2 etc. in the a-notation-argument. For a much more representative and simpler example, we can introduce flexary addition via:

```
Example 11
```

```
Input:
```

```
1 \symdef{addition}[args=a]{#1}{##1 \comp{+} ##2}
2 3 Tadaa: $\addition{a,b,c,d,e}$
```

Output:

Tadaa: abcde

The assoc-key We mentioned earlier that "formally", flexary arguments don't really "exist". Indeed, formally, addition is usually defined as a binary operation, quantifiers bind a single variable etc.

Consequently, we can tell STEX (or, rather, MMT/OMDOC) how to "resolve" flexary arguments by providing \symdecl or \symdef with an optional assoc-argument, as in \symdecl{addition}[args=a,assoc=bin]. The possible values for the assoc-key are:

bin: A binary, associative argument, e.g. as in \addition

binl: A binary, left-associative argument, e.g. $a^{b^{c^a}}$, which stands for $((a^b)^c)^d$

binr: A binary, right-associative argument, e.g. as in $A \to B \to C \to D$, which stands for $A \to (B \to (C \to D))$

pre: Successively prefixed, e.g. as in $\forall x, y, z. P$, which stands for $\forall x. \forall y. \forall z. P$

conj: Conjunctive, e.g. as in a = b = c = d or $a, b, c, d \in A$, which stand for $a = d \land b = d \land c = d$ and $a \in A \land b \in A \land c \in A \land d \in A$, respectively

pwconj: Pairwise conjunctive, e.g. as in $a \neq b \neq c \neq d$, which stands for $a \neq b \land a \neq c \land a \neq d \land b \neq c \land b \neq d \land c \neq d$

As before, at the PDF level, this annotation is invisible (and without effect), but at the level of the generated OMDoc/MMT this leads to more semantical expressions.

Mode-B Arguments

Finally, mode-B arguments simply combine the functionality of both a and b - i.e. they represent an arbitrarily long sequence of variables to be bound, e.g. for implementing quantifiers:

Example 12

Output:

```
xyzP
```

22

3.3.4 Type and Definiens Components

\symdecl and \symdef take two more optional arguments. TeX largely ignores them (except for special situations we will talk about later), but MMT can pick up on them for additional services. These are the type and def keys, which expect expressions in math-mode (ideally using semantic macros, of course!)

```
The type and def keys correspond to the type and definiens components of \begin{cal}{l} \begin{
```

The type-key allows us to provide additional information (given the necessary STEX symbols), e.g. for addition on natural numbers:

Example 13

```
Input:
```

```
1 \symdef{Nat}[type=\set]{\comp{\mathbb N}}
2 \symdef{addition}[
3     type=\funtype{\Nat,\Nat}{\Nat},
4     op=+,
5     args=a
6]{#1}{##1 \comp+ ##2}
7
8 \symname{addition} is an operation $\funtype{\Nat,\Nat}{\Nat}$
```

Output:

```
is an operation
```

The def-key allows for declaring symbols as abbreviations:

Example 14

Input:

```
1 \symdef{successor}[
2    type=\funtype{\Nat}{\Nat},
3    def=\fun{\svar{x}}{\addition{\svar{x},1}},
4    op=\mathtt{succ},
5    args=1
6]{\comp{\mathtt{succ(}#1\comp{)}}}
7
8 The \symname{successor} operation $\funtype{\Nat}{\Nat}$
9 is defined as $\fun{\svar{x}}{\addition{\svar{x},1}}$
```

Output:

```
The operation is defined as xx1
```

3.3.5 Precedences and Automated Bracketing

Having done \addition, the obvious next thing to implement is \multiplication. This is straight-forward in theory:

Example 15

```
Input:

1 \symdef{multiplication}[
2    type=\funtype{\Nat,\Nat}{\Nat},
3    op=\cdot,
4    args=a
5 ]{#1}{##1 \comp\cdot ##2}
6
7 \symname{multiplication} is an operation $\funtype{\Nat,\Nat}{\Nat}$
```

Output:

```
is an operation
```

However, if we combine \addition and \multiplication, we notice a problem:

Example 16

```
Input:
```

```
1 \addition{a, \multiplication{b, \addition{c, \multiplication{d,e}}}} \$
```

Output:

```
abcde
```

We all know that $\,$ binds stronger than , so the output abcde does not actually reflect the term we wrote. We can of course insert parentheses manually

Example 17

```
Input:
```

```
1 \addition{a, \multiplication{b, (\addition{c, \multiplication{d,e}}))}} \$
```

Output:

```
ab(cde)
```

but we can also do better by supplying *precedences* and have ST_EX insert parentheses automatically.

For that purpose, $\noindent (and hence \symdef) take an optional argument prec=cpprec>;<argprec1>x...x<argprec n>.$

We will investigate the precise meaning of $\operatorname{\mathsf{copprec}}$ and the $\operatorname{\mathsf{cargprec}}$ s shortly – in the vast majority of cases, it is perfectly sufficient to think of $\operatorname{\mathsf{prec}}$ = taking a single number and having that be the precedence of the notation, where lower precedences (somewhat

counterintuitively) bind stronger than higher precedences. So fixing our notations for \addition and \multiplication, we get:

Example 18

Input:

```
1 \notation{multiplication}[
2    op=\cdot,
3    prec=50
4 ]{#1}{##1 \comp\cdot ##2}
5 \notation{addition}[
6    op=+,
7    prec=100
8 ]{#1}{##1 \comp+ ##2}
9
10 $\addition{a, \multiplication{b, \addition{c, \multiplication{d,e}}}}$
```

Output:

```
ab(cde)
```

Note that the precise numbers used for precedences are pretty arbitrary - what matters is which precedences are higher than which other precedences when used in conjunction.

\infprec \neginfprec

It is occasionally useful to have "infinitely" high or low precedences to enforce or forbid automated bracketing entirely, e.g. for bracket-like notations such as intervals – for those purposes, \infprec and \neginfprec exist (which are implemented as the maximal and minimal integer values accordingly).g

More precisely, each notation takes

- 1. One operator precedence and
- 2. one argument precedence for each argument.

By default, all precedences are 0, unless the symbol takes no argument, in which case the operator precedence is \neginfprec (negative infinity). If we only provide a single number, this is taken as both the operator precedence and all argument precedences.



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 insert parentheses.

When STEX steps into an argument of a semantic macro, it sets p_d to the respective argument precedence of the notation used.

In the example above:

- 1. STEX starts out with $p_d = \$ infprec.
- 2. STeX encounters \addition with $p_{op} = 100$. Since $100 \ge \text{linfprec}$, it inserts no parentheses.
- 3. Next, STEX encounters the two arguments for \addition. Both have no specifically provided argument precedence, so STEX uses $p_d=p_{op}=100$ for both and recurses.

- 4. Next, STEX encounters \multiplication{b,...}, whose notation has $p_{op} = 50$.
- 5. We compare to the current downward precedence p_d set by \addition, arriving at $p_{op} = 50 \ge 100 = p_d$, so SI_EX again inserts no parentheses.



- 6. Since the notation of \multiplication has no explicitly set argument precedences, STEX uses the operator precedence for all arguments of \multiplication, hence sets $p_d = p_{op} = 50$ and recurses.
- 7. Next, STeX encounters the inner \addition{c,...} whose notation has $p_{op}=100.$
- 8. We compare to the current downward precedence p_d set by \multiplication, arriving at $p_{op} = 100 > 50 = p_d$ which finally prompts STFX to insert parentheses, and we proceed as before.

3.3.6 Variables

All symbol and notation declarations require a module with which they are associated, hence the commands \symdecl, \notation, \symdef etc. are disabled outside of smodule-environments.

Variables are different – variables are allowed everywhere, are not exported when the current module (if one exists) is imported (via $\identifont{\sc himportmodule}$ or $\identifont{\sc humbordule}$ and (also unlike symbol declarations) "disappear" at the end of the current $\sc TEX$ group.

\svar

So far, we have always used variables using \sqrt{n} , which marks-up n as a variable with name n. More generally, $\sqrt{\text{texcode}}$ marks-up the arbitrary texcode as representing a variable with name foo.

Of course, this makes it difficult to reuse variables, or introduce "functional" variables with arities > 0, or provide them with a type or definiens.

\vardef

For that, we can use the \vardef command. Its syntax is largely the same as that of \symdef, but unlike symbols, variables have only one notation (TODO: so far?), hence there is only \vardef and no \vardecl.

Example 19

Input:

```
1 \vardef{varf}[
2    name=f,
3    type=\funtype{\Nat}{\Nat},
4    op=f,
5    args=1,
6    prec=0;\neginfprec
7 ]{\comp{f}#1}
8 \vardef{varn}[name=n,type=\Nat]{\comp{n}}
9 \vardef{varx}[name=x,type=\Nat]{\comp{x}}
10
11 Given a function $\varf!:\funtype{\Nat}{\Nat}$,
12 by $\addition{\varf!,\varn}$ we mean the function\rustexBREAK
13 $\fun{\varx}{\varf}{\addition}{\varx},\varn}}$
```

Output:

```
Given a function f:, by fn we mean the function xf(xn)
```

.

(of course, "lifting" addition in the way described in the previous example is an operation that deserves its own symbol rather than abusing \addition, but... well.)

TODO: bind=forall/exists

3.3.7 Variable Sequences

Variable sequences occur quite frequently in informal mathematics, hence they deserve special support. Variable sequences behave like variables in that they disappear at the end of the current TEX group and are not exported from modules, but their declaration is quite different.

\varseq

A variable sequence is introduced via the command \warseq, which takes the usual optional arguments name and type. It then takes a starting index, an end index and a notation for the individual elements of the sequence parametric in an index. Note that both the starting as well as the ending index may be variables.

This is best shown by example:

Example 20

```
Input:
```

```
1 \vardef{varn}[name=n,type=\Nat]{\comp{n}}
2 \varseq{seqa}[name=a,type=\Nat]{1}{\varn}{\comp{a}_{#1}}
3
4 The $i$th index of $\seqa!$ is $\seqa{i}$.
```

Output:

```
The ith index of a_1, \ldots, a_n is a_i.
```

.

Note that the syntax \seqa! now automatically generates a presentation based on the starting and ending index.

TODO: more notations for invoking sequences.

Notably, variable sequences are nicely compatible with a-type arguments, so we can do the following:

Example 21

Input:

 $1 \addition{\seqa}$

Output:

 $a_1...a_n$

Sequences can be *multidimensional* using the args-key, in which case the notation's arity increases and starting and ending indices have to be provided as a comma-separated

Example 22

Input:

```
1 \vardef{varm}[name=m,type=\Nat]{\comp{m}}
2 \varseq{seqa}[
3     name=a,
4     args=2,
5     type=\Nat,
6 ]{1,1}{\varn,\varm}{\comp{a}_{#1}^{#2}}
7
8 $\seqa!$ and $\addition{\seqa}$
```

Output:

```
a_1^1, \ldots, a_n^m and a_1^1, \ldots, a_n^m
```

We can also explicitly provide a "middle" segment to be used, like such:

Example 23

```
Input:

1 \varseq{seqa}[
2     name=a,
3     type=\Nat,
4     args=2,
5     mid={\comp{a}_{\varn}^1,\comp{a}_1^2,\ellipses,\comp{a}_{1}^{\varn}}
6 ]{1,1}{\varn,\varm}{\comp{a}_{\alpha}^{\#1}^{\#2}}
7
8 $\seqa!$ and $\addition{\seqa}$
```

Output:

```
a_1^1, \dots, a_n^1, a_1^2, \dots, a_1^m, \dots, a_n^m \text{ and } a_1^1 \dots a_n^1 a_1^2 \dots a_1^m \dots a_n^m
```

3.4 Module Inheritance and Structures

The STEX features for modular document management are inherited from the OM-Doc/MMT model that organizes knowledge into a graph, where the nodes are theories (called modules in STEX) and the edges are truth-preserving mappings (called theory morphismes in MMT). We have already seen modules/theories above.

Before we get into theory morphisms in STEX we will see a very simple application of modules: managing multilinguality modularly.

3.4.1 Multilinguality and Translations

If we load the STEX document class or package with the option lang=<lang>, STEX will load the appropriate babel language for you – e.g. lang=de will load the babel language

ngerman. Additionally, it makes STEX aware of the current document being set in (in this example) german. This matters for reasons other than mere babel-purposes, though:

Every module is assigned a language. If no STEX package option is set that allows for inferring a language, STEX will check whether the current file name ends in e.g. .en.tex (or .de.tex or .fr.tex, or...) and set the language accordingly. Alternatively, a language can be explicitly assigned via \begin{smodule} [lang=<language>] {Foo}.

```
Technically, each smodule-environment induces two OMDoc/MMT theories: \begin{smodule}[lang=<lang>]{Foo} generates a theory some/namespace?Foo that only contains the "formal" part of the module – i.e. exactly the content—M→ that is exported when using \importmodule.

TAN Additionally, MMT generates a language theory some/namespace/Foo?<lang> that includes some/namespace?Foo and contains all the other document content – variable declarations, includes for each \usenbodule, etc.
```

Notably, the language suffix in a filename is ignored for \usemodule, \importmodule and in generating/computing URIs for modules. This however allows for providing translations for modules between languages without needing to duplicate content:

If a module Foo exists in e.g. english in a file Foo.en.tex, we can provide a file Foo.de.tex right next to it, and write \begin{smodule}[sig=en]{Foo}. The sig-key then signifies, that the "signature" of the module is contained in the english version of the module, which is immediately imported from there, just like \importmodule would.

Additionally to translating the informal content of a module file to different languages, it also allows for customizing notations between languages. For example, the least common multiple of two numbers is often denoted as $\mathtt{lcm}(a,b)$ in english, but is called kleinstes gemeinsames Vielfaches in german and consequently denoted as $\mathtt{kgV}(a,b)$ there.

We can therefore imagine a german version of an lcm-module looking something like this:

```
1 \begin{smodule}[sig=en]{lcm}
2 \notation*{lcm}[de]{\comp{\mathtt{kgV}}(#1,#2)}
3
4 Das \symref{lcm}{kleinste gemeinsame Vielfache}
5 $\lcm{a,b}$ von zwei Zahlen $a,b$ ist...
6 \end{smodule}
```

If we now do \importmodule{lcm} (or \usemodule{lcm}) within a german document, it will also load the content of the german translation, including the de-notation for \lcm.

3.4.2 Simple Inheritance and Namespaces

\importmodule \usemodule

\importmodule[Some/Archive] {path?ModuleName} is only allowed within an smodule-environment and makes the symbols declared in ModuleName available therein. Additionally the symbols of ModuleName will be exported if the current module is imported somewhere else via \importmodule.

\usemodule behaves the same way, but without exporting the content of the used module.

It is worth going into some detail how exactly \importmodule and \usemodule 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 URI.

Ideally, STeX would use arbitrary URIs for modules, with no forced relationships between the logical namespace of a module and the physical location of the file declaring the module – like MMT does things.

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 completenesses 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 archive, the namespace is file://path/to/file.
- If the same statement occurs in a file /path/to/file/bar[. \(\lang\right)\)].tex, the namespace is file://path/to/file/bar.

In other words: outside of 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**-folder is replaced by the archive's namespace URI.

Conversely, here is how namespaces/URIs and file paths are computed in import statements, examplary \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 document 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 document, or otherwise to the module Foo in the archive's top-level namespace. In the latter case, is has to be declared in a file Foo[.\lang\].tex directly in the archive's source-folder.



- 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-folder, respectively.
 - The module Foo must either be declared in the file $\langle top\text{-}directory \rangle / \text{some/path/Foo[.} \langle lang \rangle]$.tex, or in $\langle top\text{-}directory \rangle / \text{some/path[.} \langle lang \rangle]$.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 mathhubdirectory.
- Finally, \importmodule{full://uri?Foo} naturally refers to the module Foo in the namespace full://uri. Since the file this module is declared



in can not be determined directly from the URI, the module must be in memory already, e.g. by being referenced earlier in the same document. Since this is less compatible with a modular development, using full URIs directly is strongly discouraged, unless the module is delared in the current file directly.

\STEXexport

\importmodule and \usemodule import all symbols, notations, semantic macros and (recursively) \importmodules. If you want to additionally export e.g. convenience macros and other (STEX) code from a module, you can use the command \STEXexport{<code>} in your module. Then <code> is executed (both immediately and) every time the current module is opened via \importmodule or \usemodule.

For persistency reasons, everything in an \STEXexport is digested by TeXin the LATeX3-category code scheme. This means that the characters _ and : are considered letters and valid parts of control sequence names, and space characters are ignored entirely. For spaces, use the character ~ instead, and keep in mind, that if you want to use subscripts, you should use \c_math_subscript_token instead of _!



Also note, that **\newcommand** defines macros *globally* and throws an error if the macro already exists, potentially leading to low-level LATEX errors if we put a **\newcommand** in an **\STEXexport** and the **<code>** is executed more than once in a document – which can happen easily.

A safer alternative is to use macro definition principles, that are safe to use even if the macro being defined already exists, and ideally are local to the current T_EX group, such as \def or \let .

3.4.3 The mathstructure Environment

A common occurrence in mathematics is bundling several interrelated "declarations" together into *structures*. For example:

- A monoid is a structure $M \circ e$ with $\circ : M \times M \to M$ and $e \in M$ such that...
- A topological space is a structure $X\mathcal{T}$ where X is a set and \mathcal{T} is a topology on X
- A partial order is a structure $S \leq$ where \leq is a binary relation on S such that...

This phenomenon is important and common enough to warrant special support, in particular because it requires being able to *instantiate* such structures (or, rather, structure *signatures*) in order to talk about (concrete or variable) *particular* monoids, topological spaces, partial orders etc.

mathstructure

The mathstructure environment allows us to do exactly that. It behaves exactly like the smodule environment, but is itself only allowed inside an smodule environment, and allows for instantiation later on.

How this works is again best demonstrated by example:

Example 24

Input:

```
\begin{mathstructure} { monoid}
2
      \symdef{universe}[type=\set]{\comp{U}}}
3
      \symdef{op}[
4
          args=2,
5
          type=\funtype{\universe, \universe}{\universe},
          op=\circ
7
      ]{#1 \comp{\circ} #2}
      \symdef{unit}[type=\universe]{\comp{e}}
9 \end{mathstructure}
11 A \symname{monoid} is...
```

Output:

```
A is...
```

Note that the \symname{monoid} is appropriately highlighted and (depending on your pdf viewer) shows a URI on hovering - implying that the mathstructure environment has generated a symbol monoid for us. It has not generated a semantic macro though, since we can not use the monoid-symbol directly. Instead, we can instantiate it, for example for integers:

Example 25

```
Input:
```

```
1 \searrow mdef{Int}[type=\set]{\comp{\mathbb Z}}
2 \symdef{addition}[
3
     type = \{ Int, Int \} \{ Int \},
     args=2,
     op=+
6 ]{##1 \comp{+} ##2}
7 \symdef{zero}[type=\Int]{\comp{0}}
9 $\mathstruct{\Int,\addition!,\zero}$ is a \symname{monoid}.
```

Output:

```
is a.
```

So far, we have not actually instantiated monoid, but now that we have all the symbols to do so, we can:

Example 26

Input:

```
1 \instantiate{intmonoid}{monoid}{\mathbb{Z}_{+,0}}[
2     universe = Int ,
3     op = addition ,
4     unit = zero
5 ]
6
7 $\intmonoid{\universe}$, $\intmonoid{\unit}$ and $\intmonoid{\unit}$.
8
9 Also: $\intmonoid!$
```

Output:

```
, and ab.
Also:
```

\instantiate

So summarizing: \instantiate takes four arguments: The (macro-)name of the instance, a key-value pair assigning declarations in the corresponding mathstructure to symbols currently in scope, the name of the mathstructure to instantiate, and lastly a notation for the instance itself.

It then generates a semantic macro that takes as argument the name of a declaration in the instantiated **mathstructure** and resolves it to the corresponding instance of that particular declaration.

```
\instantiate and mathstructure make use of the Theories-as-Types paradigm (see [MRK18]):
    mathstructure{<name>} simply creates a nested theory with name

<-M-> <name>-structure. The constant <name> is defined as Mod(<name>-structure)

--M-> - a dependent record type with manifest fields, the fields of which are generated

--T-> from (and correspond to) the constants in <name>-structure.
    \instantiate generates a constant whose definiens is a record term of type

Mod(<name>-structure), with the fields assigned based on the respective key-
value-list.
```

Notably, \instantiate throws an error if not every declaration in the instantiated mathstructure is being assigned.

You might consequently ask what the usefulness of mathstructure even is.

\varinstantiate

The answer is that we can also instantiate a mathstructure with a *variable*. The syntax of \varianstantiate is equivalent to that of \instantiate, but all of the key-value-pairs are optional, and if not explicitly assigned (to a symbol *or* a variable declared with \vardef) inherit their notation from the one in the mathstructure environment.

This allows us to do things like:

Example 27 Input:

```
1 \varinstantiate{varM}{monoid}{M}
2
3 A \symname{monoid} is a structure
4 $\varM!:=\mathstruct{\varM{universe},\varM{op}!,\varM{unit}}$
5 such that
6 $\varM{op}!:\funtype{\varM{universe}},\varM{universe}}{\varM{universe}}$
...

Output:

A is a structure M := Uoe such that o: UUU ...
```

and

Example 28

```
Input:
```

```
1 \varinstantiate{varMb}{monoid}{M_2}[universe = Int]
2
3 Let $\varMb!:=\mathstruct{\varMb{universe},\varMb{op}!,\varMb{unit}}$
4 be a \symname{monoid} on $\Int$ ...
```

Output:

```
Let M_2 := \circ e be a on ...
```

.

We will return to these two example later, when we also know how to handle the axioms of a monoid.

3.4.4 The copymodule Environment

TODO: explain

Given modules:

Example 29

```
Input:
```

```
1 \begin{smodule}{magma}
2 \symdef{universe}{\comp{\mathcal U}}
3 \symdef{operation}[args=2,op=\circ]{#1 \comp\circ #2}
4 \end{smodule}
5 \begin{smodule}{monoid}
6 \importmodule{magma}
7 \symdef{unit}{\comp e}
8 \end{smodule}
9 \begin{smodule}{group}
10 \importmodule{monoid}
11 \symdef{inverse}[args=1]{{#1}^{\comp{-1}}}
12 \end{smodule}
```

Output:

.

We can form a module for *rings* by "cloning" an instance of <code>group</code> (for addition) and <code>monoid</code> (for multiplication), respectively, and "glueing them together" to ensure they share the same universe:

Example 30

Input:

```
\begin{smodule}{ring}
      \begin{copymodule} { group } { addition }
3
          \renamedecl[name=universe] {universe} {runiverse}
          \renamedecl[name=plus]{operation}{rplus}
 4
          \renamedecl[name=zero]{unit}{rzero}
 5
          \renamedecl[name=uminus]{inverse}{ruminus}
 7
      \end{copymodule}
      \notation*{rplus}[plus,op=+,prec=60]{#1 \comp+ #2}
9
      \notation*{rzero}[zero]{\comp0}
10
      \notation*{ruminus}[uminus,op=-]{\comp- #1}
11
      \begin{copymodule} {monoid} {multiplication}
12
          \assign{universe}{\runiverse}
13
          \renamedecl[name=times] { operation } { rtimes }
14
          \renamedecl[name=one] {unit}{rone}
15
      \end{copymodule}
16
      \notation*{rtimes}[cdot,op=\cdot,prec=50]{#1 \comp\cdot #2}
17
      \notation*{rone}[one]{\comp1}
18
      Test: $\rtimes a{\rplus c{\rtimes de}}$
19 \end{smodule}
```

Output:

```
Test: a(cde)
```

.

TODO: explain donotclone

3.4.5 The interpretmodule Environment

TODO: explain

Example 31

```
Input:
```

```
\begin{smodule}{int}
      \symdef{Integers}{\comp{\mathbb Z}}
3
      \symdef{plus}[args=2,op=+]{#1 \comp+ #2}
 4
      \symdef{zero}{\comp0}
 5
6
      \symdef{uminus}[args=1,op=-]{\comp-#1}
      \begin{interpretmodule}{group}{intisgroup}
          \assign{universe}{\Integers}
          \assign{operation}{\plus!}
10
          \assign{unit}{\zero}
11
          \assign{inverse}{\uminus!}
      \end{interpretmodule}
12
13 \end{smodule}
```

Output:

3.5 Primitive Symbols (The STEX Metatheory)

The stex-metatheory package contains STEX symbols so ubiquitous, that it is virtually impossible to describe any flexiformal content without them, or that are required to annotate even the most primitive symbols with meaningful (foundation-independent) "type"-annotations, or required for basic structuring principles (theorems, definitions). As such, it serves as the default meta theory for any STEX module.

We can also see the stex-metatheory as a foundation of mathematics in the sense of [Rab15], albeit an informal one (the ones discussed there are all formal foundations). The state of the stex-metatheory is necessarily incomplete, and will stay so for a long while: It arises as a collection of empirically useful symbols that are collected as more and more mathematics are encoded in STFX and are classified as foundational.

Formal foundations should ideally instantiate these symbols with their formal counterparts, e.g. **isa** corresponds to a typing operation in typed setting, or the \in -operator in set-theoretic contexts; **bind** corresponds to a universal quantifier in (nth-order) logic, or a Π in dependent type theories.

We make this theory part of the STEX collection due to the obiquity of the symbols involved. Note however, that the metatheory is for all practical purposes a "normal" STEX module, and the symbols contained "normal" STEX symbols.

Chapter 4

Using STEX Symbols

Given a symbol declaration \symdecl{symbolname}, we obtain a semantic macro \symbol name. We can use this semantic macro in math mode to use its notation(s), and we can use \symbolname! in math mode to use its operator notation(s). What else can we do?

4.1 \symref and its variants

\symref \symname We have already seen \symname and \symref, the latter being the more general.

\symref{<symbolname>}{<code>} marks-up <code> as referencing <symbolname>. Since quite often, the <code> should be (a variant of) the name of the symbol anyway, we also have \symname{<symbolname>}.

Note that \symname uses the *name* of a symbol, not its macroname. More precisely, \symname will insert the name of the symbol with "-" replaced by spaces. If a symbol does not have an explicit name= given, the two are equal - but for \symname it often makes sense to make the two explicitly distinct. For example:

Example 32

```
Input:
  \symdef{Nat}[
      name=natural-number,
      type=\set
 4]{\mathbb{N}}
 6 A \symname{Nat} is...
Output:
```

A is...

\symname takes two additional optional arguments, pre= and post= that get prepended or appended respectively to the symbol name.

\Symname

Additionally, \Symname behaves exactly like \symname, but will capitalize the first letter of the name:

Example 33

Input:

1 \Symname[post=s]{Nat} are...

Output:

are...

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

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

However, especially in \symname (or if a symbol was introduced using \symdec1* 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 string does *not* correspond to a semantic macro \string and does *not* contain a ?, then STEX checks all symbols currently in scope until it finds one, whose name is string. If string is of the form pre?name, STEX first looks through all modules currently in scope, whose full 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.

4.2 Marking Up Text and On-the-Fly Notations

We can also use semantic macros outside of text mode though, which allows us to annotate arbitrary text fragments.

Let us assume again, that we have $\symdef{addition}[args=2]{\#1 \comp+ \#2}$. Then we can do

Example 34

Input:

Output:

 $n\ m\ \mathbf{is}...$

"...which marks up the text fragment as representing an application of the addition-symbol to two argument n and m.



Note the difference in treating "arguments" between math mode and text mode. In math mode the (in this case two) tokens/groups following the \addition macro are treated as arguments to the addition function, whereas in text mode the group following \addition is taken to be the ad-hoc presentation. We drill in on this now

\arg

In text mode, every semantic macro takes exactly one argument, namely the text-fragment to be annotated. The \arg command is only valid within the argument to a semantic macro and marks up the *individual arguments* for the symbol.

We can also use semantic macros in text mode to invoke an operator itself instead of its application, with the usual syntax using !:

.

Indeed, \symbolname! {<code>} is exactly equivalent to \symref {symbolname} {<code>} (the latter is in fact implemented in terms of the former).

\arg also allows us to switch the order of arguments around and "hide" arguments: For example, $\arg[3]{<code>}$ signifies that <code> represents the *third* argument to the current operator, and $\arg*[i]{<code>}$ signifies that <code> represents the *i*th argument, but it should not produce any output (it is exported in the xhtml however, so that MMT and other systems can pick up on it).

Example 36

Input:

EdN:1

- 1 \addition{\comp{adding}
- 2 \arg[2]{\$\svar{k}\$}
 - \arg*{\$\addition{\svar{n}}{\svar{m}}\$}} yields...

Output:

k yields...

¹ EdNote: MK: I do not understand why we have to/want to give the second arg*; I think this must be elaborated on.

EdN:2

Note that since the second \arg has no explicit argument number, it automatically represents the first not-yet-given argument – i.e. in this case the first one.²

The same syntax can be used in math mod as well. This allows us to spontaneously introduce new notations on the fly. We can activate it using the starred variants of semantic macros:

```
Example 37
Input:

1 Given $\addition{\svar{n}}{\svar{m}}$, then
2 $\addition*{
3 \arg*{\addition{\svar{n}}}{\svar{m}}}
4 \comp{+}
5 \arg{\svar{k}}
6 }$ yields...

Output:

Given nm, then k yields...
```

If we take features like \inputref and \mhinput (and the sfragment-environment, see subsection 7.2.1) 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 inputted 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.

The \sref command attempts to do precisely that. Unlike plain \ref, \autoref etc., \sref refers to not just a label, but instead a pair consisting of a label and the document in whose context we want to refer to it. Conversely, every document (i.e. standalone compilable .tex-file) keeps track of the "names" (Definition 3.1 etc.) for every label as determined in the context of the document, and stores them in a dedicated file \jobname.sref. Additionally, every document has a "reference name" (e.g. "the section on Foo"). This allows us to refer to "label x in document D" to yield "Definition 1 in the section on Foo". And of course, STEX can decide based on the current document to either refer to the label by its "full name" or directly as e.g. Definition 3.1 depending on whether the label occurs in the current document anyway (and link to it accordingly).

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

• The *label* of the reference target (e.g. some_definition),

²Ednote: MK: I do not understand this at all.

- (optionally) the *file*/document containing the reference target (e.g. section2). This is not strictly necessary, but allows for additional disambiguation between possibly duplicate labels across files, and
- (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 to provide a title for external references.

\STEXreftitle

For the latter, we can use \TEXreftitle . For example, the full \TEX documentation (stex-doc.tex) declares \TEXreftitle the full \tEX documentation}, whereas the manual (stex-manual.tex) declares \TEXreftitle the \tEX manual}.

\sref

```
\label{lem:context} $$\operatorname{\archive=\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\archive2\arc
```

This command references $\langle label \rangle$ (declared in $\langle file \rangle$ in $\langle archive1 \rangle$). If the object (section, figure, etc.) with that label occurs ultimately 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 $\langle document\text{-}context \rangle$ in $\langle archive2 \rangle$.

For example, the reference to the **sfragment**-environment above will appear as "subsection 7.2.1 (Introduction) in the STEX3 manual" if you are reading this in the package documentation for **stex-references** directly, but as a linked "subsection 7.2.1" in the full documentation or manual. This is achieved using

\sref[file=stex-document-structure] {sec:ds:intro} [in=../stex-manual]. If the optional cite= $\langle citation \rangle$ -argument is provided, \sref will ignore the document title as provided by the document context, and will instead use "in \cite{ $\langle citation \rangle$ }". For a further example, the following:

Part III

will say "Part III" (and link accordingly) in the full documentation, and "Part III (Extensions) in the full STEX3 documentation" everywhere else. This is achieved using \sref[file=../stex-doc] {part:extends}[in=../stex-doc].

\extref

```
\label{lem:context} $$ \operatorname{archive}(archive), file=\langle file \rangle $$ {\langle label \rangle}{archive=\langle archive2 \rangle, in=\langle document-context \rangle, cite=\langle citation \rangle}$$
```

The \extref-command behaves exactly like \sref, but takes required the document context argument and will always use it for generating the document text, regardless of whether the label occurs in the current document.

Chapter 5

STEX Statements

5.1 Definitions, Theorems, Examples, Paragraphs

As mentioned earlier, we can semantically mark-up *statements* such as definitions, theorems, lemmata, examples, etc.

The corresponding environments for that are:

- sdefinition for definitions,
- sassertion for assertions, i.e. propositions that are declared to be *true*, such as theorems, lemmata, axioms,
- sexample for examples and counterexamples, and
- sparagraph for "other" semantic paragraphs, such as comments, remarks, conjectures, etc.

The *presentation* of these environments can be customized to use e.g. predefined theorem-environments, see ?? for details.

All of these environments take optional arguments in the form of key=value-pairs. Common to all of them are the keys id= (for cross-referencing, see ??), type= for customization (see ??) and additional information (e.g. definition principles, "difficulty" etc), as well as title= (for giving the paragraph a title), and finally for=.

The for= key expects a comma-separated list of existing symbols, allowing for e.g. things like

Example 38

```
Input:

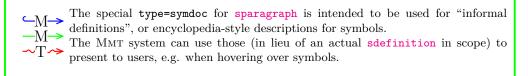
1 \begin{sexample}[
2    id=additionandmultiplication.ex,
3    for={addition,multiplication},
4    type={trivial,boring},
5    title={An Example}
6]
7    $\addition{2,3}$ is $5$, $\multiplication{2,3}$ is $6$.
8 \end{sexample}
```

Output:

Example 5.1.1 (An Example). 23 is 5, 23 is 6.

\definiendum \definame \Definame

sdefinition (and sparagraph with type=symdoc) introduce three new macros: definiendum behaves like symref (and definame/Definame like symname/Symname, respectively), but highlights the referenced symbol as *being defined* in the current definition.



\definiens

Additionally, sdefinition (and sparagraph with type=symdoc) introduces \definiens [<optional symbols which marks up <code> as being the explicit definiens of <optional symbols symbols).

All four statement environments – i.e. sdefinition, sassertion, sexample, and sparagraph – also take an optional parameter name= – if this one is given a value, the environment will generate a *symbol* by that name (but with no semantic macro). Not only does this allow for \symmetry merce et al, it allows us to resume our earlier example for monoids much more nicely:³

Example 39 Input:

EdN:3

 $^{^3{}m EdNote}$: MK: we should reference the example explicitly here.

```
\begin{mathstructure} { monoid}
       \symdef{universe}[type=\set]{\comp{U}}}
 2
 3
       \symdef{op}[
 \tilde{4}
           args=2,
 5
           type=\funtype{\universe,\universe}{\universe},
           op=\circ
 6
7
8
9
      ]{#1 \comp{\circ} #2}
       \symdef{unit}[type=\universe]{\comp{e}}
10
       \begin{sparagraph}[type=symdoc,for=monoid]
           A \definame{monoid} is a structure
11
12
           $\mathstruct{\universe,\op!,\unit}$
13
           where $\op!:\funtype{\universe}{\universe}$ and
14
           $\inset{\unit}{\universe}$ such that
15
\frac{16}{17}
           \begin{sassertion} [name=associative,
               type=axiom,
18
               title=Associativity]
               $\op!$ is associative
19
20
           \end{sassertion}
21
           \begin{sassertion} [name=isunit,
\overline{22}
               type=axiom,
23
               title=Unit]
24
               \displaystyle {\displaystyle \{ \op{\svar}\{x\}}{\unit}}{\svar}\
25
               for all $\inset{\svar{x}}{\universe}$
26
           \end{sassertion}
27
       \end{sparagraph}
   \end{mathstructure}
30 An example for a \symname{monoid} is..
```

Output:

```
A monoid is a structure where: and such that

Axiom 5.1.2 (Associativity). is associative

Axiom 5.1.3 (Unit). xx for all x

An example for a is...
```

٠

EdN:4

The main difference to before⁴ is that the two sassertions now have name= attributes. Thus the mathstructure now contains two additional symbols, namely the axioms for associativity and that e is a unit. Note that both symbols do not represent the mere propositions that e.g. \circ is associative, but the assertion that it is actually true that \circ is associative.

If we now want to instantiate monoid (unless with a variable, of course), we also need to assign associative and neutral to analogous assertions. So the earlier example

```
1 \instantiate{intmonoid}{monoid}{\mathbb{Z}_{+,0}}[
2    universe = Int ,
3    op = addition ,
4    unit = zero
5 ]
```

⁴EDNOTE: MK: reference

...will not work anymore. We now need to give assertions that addition is associative and that zero is a unit with respect to addition.²

5.2 Proofs

The stex-proof package supplies macros and environment that allow to annotate the structure of mathematical proofs in STEX document. This structure can be used by MKM systems for added-value services, either directly from the STEX sources, or after translation.

Its central component is the sproof-environment, whose body consists of:

- *subproofs* via the **subproof**-environment,
- proof steps via the \spfstep, \eqstep \assumption, and \conclude macros, and
- comments, via normal text without special markup.

sproof, subproof and the various proof step macros take the following optional
arguments:

```
id (\langle string \rangle) for referencing,
method (\langle string \rangle) the proof method (e.g. contradiction, induction,...)
```

term $(\langle token \ list \rangle)$ the (ideally semantically-marked up) proposition that is derived/proven by this proof/subproof/proof step.

Additionally, they take one mandatory argument for the document text to be annotated, or (in the case of the environments) as an introductory description of the proof itself. Since the latter often contains the term to be derived as text, alternatively to providing it as an optional argument, the mandatory argument can use the \yield-macro to mark it up in the text.

The sproof and subproof environments additionally take two optional arguments:

for the symbol identifier/name corresponding to the sassertion to be proven. This too subsumes \yield and the term-argument.

hide In the pdf, this only shows the mandatory argument text and hides the body of the environment. In the HTML (as served by MMT), the bodies of all proof and subproof environments are *collapsible*, and hide collapses the body by default.

```
1 \begin{sassertion}[type=theorem,name=sqrt2irr]
2 \conclusion{\irrational{$\arg{\realroot{2}}$ is \comp{irrational}}}.
3 \end{sassertion}
4
5 \begin{sproof}[for=sqrt2irr,method=contradiction]{By contradiction}
6 \assumption{Assume \yield{\rational{$\arg{\realroot{2}}$$ is \comp{rational}}}}
8 \begin{subproof}[method=straightforward]{Then
9 \yield{$\eq{\ratfrac{\intpow{\vara}{2}}{\intpow{\varb}2}}{2}$$
for some $\inset{\vara, \varb}\PosInt$ with
\coprime{$\arg{\vara}, \arg{\varb}$$ \comp{coprime}}}}
```

²Of course, STEX can not check that the assertions are the "correct" ones – but if the assertions (both in monoid as well as those for addition and zero) are properly marked up, MMT can. TODO: should

```
\assumption{By assumption, \yield{there are
                     $\inset{\vara,\varb}\PosInt $ with
14
                     \realroot{2}=\ratfrac{\langle \rangle}{\rangle}}
15
                     \spfstep{wlog, we can assume \coprime{$\arg{\vara},\arg{\varb}$$
                     to be \comp{coprime}}}
16
                             % a comment:
17
                             If not, reduce the fraction until numerator and denominator
18
19
                             are coprime, and let the resulting components be
20
                             $\vara $ and $\varb $
                     \spfstep{Then \yield{$\eq{\intpow{\ratfrac{\vara}{\varb}}2}2$}}
21
22
                     \eqstep{\ratfrac{\intpow{\vara}2}{\intpow{\varb}2}}
23
             \end{subproof}
24
             \begin{subproof}[term=\divides{2}{\vara},method=straightforward]{
25
                     Then $\vara $ is even}
                     \spfstep{Multiplying the equation by $\intpow{\varb}2$ yields
26
                     \ \phi_{\vara}^2_{\inttimes}^2_{\intpow}^2}_{\inttimes}^2_{\intpow}^2}_{\inttimes}^2}_{\intpow}^2_{\intpow}^2}_{\intpow}^2_{\intpow}^2}_{\intpow}^2_{\intpow}^2_{\intpow}^2}_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2}_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{\intpow}^2_{
27
                     \spfstep[term=\divides{2}{\intpow{\vara}2}]{Hence
28
29
                     $\intpow{\vara}2$ is even}
30
                     \conclude[term=\divides{2}{\vara}]{Hence $\vara $ is even as well}
31
                    % another comment:
                    Hint: Think about the prime factorizations of $\vara $ and
32
33
                     $\intpow{\vara}2$
34
             \end{subproof}
35
             \begin{subproof} [term=\divides{2}{\varb}, method=straightforward,] {
36
                     Then $\varb $ is also even}
37
                     \spfstep{Since $\vara $ is even, we have \yield{some $\varc $
38
                        such that \left\{ \left( \frac{2}{\sqrt{s}} \right) \right\}
39
                     \spfstep{Plugging into the above, we get
40
                         \ \left( \frac{1}{2}{\sigma_{\infty}}\right)
41
                             {\left( \sum_{2}{\left( \sum_{v}\right) }\right) }
42
                     \eqstep{\inttimes{4}{\intpow{\vara}2}}
43
                     \spfstep{Dividing both sides by $2$ yields
                          \ \left( \frac{s}{q(\infty_{\varepsilon})^2} \right) 
44
45
                     \spfstep[term=\divides{2}{\intpow{\varb}2}]{Hence
46
                        $\intpow{\varb}2$ is even}
47
                     \conclude[term=\divides{2}{\varb}]{Hence $\varb $ is even}
48
                    % one more comment:
49
                    By the same argument as above
50
             \end{subproof}
51
             \conclude[term=\contradiction]{Contradiction to $\vara,\varb $ being
52
             \symname{coprime}.}
53 \end{sproof}
```

which will produce:

```
Theorem 5.2.1. \sqrt{2} is .

Proof: By contradiction

1. Assume \sqrt{2} is

2. Then (\frac{a^2}{b^2})^2 for some ab with a, b

2.1. By assumption, there are ab with \sqrt{2} = \frac{a}{b}

2.2. wlog, we can assume a, b to be

If not, reduce the fraction until numerator and denominator are coprime, and let the re-
```

```
sulting components be a and b
2.3. Then (\frac{a}{b})^22
= \frac{a^2}{b^2}
3. Then a is even
3.1. Multiplying the equation by b^2 yields a^2 2b^2
3.2. Hence a^2 is even
\Rightarrow Hence a is even as well
 Hint: Think about the prime factorizations of a and a^2
4. Then b is also even
4.1. Since a is even, we have some c such that 2ca
4.2. Plugging into the above, we get (2a)^2 2b^2
= 4a^2
4.3. Dividing both sides by 2 yields b^2 2a^2
4.4. Hence b^2 is even
\Rightarrow Hence b is even
 By the same argument as above
\Rightarrow Contradiction to a, b being.
```

If we mark all subproofs with hide, we will obtain the following instead:

```
Theorem 5.2.2. \sqrt{2} is .

Proof: By contradiction

1. Assume \sqrt{2} is

2. Then \left(\frac{a^2}{b^2}\right)2 for some ab with a, b

3. Then a is even

4. Then b is also even

\Rightarrow Contradiction to a, b being .
```

However, the hidden subproofs will still be shown in the HTML, only in an expandable section which is collapsed by default.

The above style of writing proofs is usually called *structured proofs*. They have a huge advantage over the traditional purely prosaic style, in that (as the name suggests) the actual *structure* of the proof is made explicit, which almost always makes it considerably more comprehensible. We, among many others, encourage the general use of structured proofs.

Alas, most proofs are not written in this style, and we would do users a disservice by insisting on this style. For that reason, the spfblock environment turns all subproofs and proof step macros into presentationally neutral *inline* annotations, as in the induction step of the following example:

```
1 \begin{sproof} [id=simple-proof,method=induction]
2 {We prove that $\sum_{i=1}^n{2i-1}=n^{2}$ by induction over $n$}
```

```
For the induction we have to consider three cases: % <- a comment
     \begin{subproof}{$n=1$}
5
     \spfstep*{then we compute $1=1^2$}
6
     \end{subproof}
7
     \begin{subproof}{$n=2$}
         This case is not really necessary, but we do it for the
9
         fun of it (and to get more intuition).
10
       \spfstep*{We compute $1+3=2^{2}=4$.}
11
     \end{subproof}
12
     \begin{subproof}{\$n>1\$}\begin{spfblock}
13
        \assumption[id=ind-hyp]{
         Now, we assume that the assertion is true for a certain k \leq 1,
14
15
         16
17
18
         We have to show that we can derive the assertion for $n=k+1$ from
         this assumption, i.e. \sum_{i=1}^{k+1}{(2i-1)}=(k+1)^{2}.
19
20
21
       \spfstep{
22
         We obtain \left(\sum_{i=1}^{k+1}{2i-1}\right)
23
           \sum_{i=1}^k{2i-1}+2(k+1)-1}
24
         \spfjust{by \splitsum{\comp{splitting the sum}
25
         \arg*{\{s_{i=1}^{k+1}}{(2i-1)}=(k+1)^{2}}}.
26
27
       \spfstep{
28
         Thus we have \gamma_{i=1}^{k+1}{(2i-1)}=k^2+2k+1}
29
         \spfjust{by \symname{induction-hypothesis}}.
30
31
       \conclude{
32
         We can \spfjust{\simplification{\comp{simplify} the right-hand side
         \arg*{k^2+2k+1}} to
33
34
         {k+1}^2, which proves the assertion.
35
36
     \end{spfblock}\end{subproof}
37
      \conclude{
38
       We have considered all the cases, so we have proven the assertion.
39
40 \end{sproof}
```

This yields the following result:

```
Proof: We prove that \sum_{i=1}^{n} 2i - 1 = n^2 by induction over n For the induction we have to consider three cases:

1. n = 1 then we compute 1 = 1^2

2. n = 2

This case is not really necessary, but we do it for the fun of it (and to get more intuition). We compute 1 + 3 = 2^2 = 4.

3. n > 1

Now, we assume that the assertion is true for a certain k \ge 1, i.e. \sum_{i=1}^{k} (2i - 1) = k^2.

We have to show that we can derive the assertion for n = k+1 from this assumption,
```

i.e. $\sum_{i=1}^{k+1} (2i-1) = (k+1)^2$. We obtain $\sum_{i=1}^{k+1} 2i-1 = \sum_{i=1}^k 2i-1 + 2(k+1) - 1$ by . Thus we have $\sum_{i=1}^{k+1} (2i-1) = k^2 + 2k + 1$ by . We can the right-hand side to $k+1^2$, which proves the assertion.

⇒ We have considered all the cases, so we have proven the assertion.

sproof

The sproof environment is the main container for proofs. It takes an optional KeyVal argument that allows to specify the id (identifier) and for (for which assertion is this a proof) keys. The regular argument of the proof environment contains an introductory comment, that may be used to announce the proof style. The proof environment contains a sequence of spfstep, spfcomment, and spfcases environments that are used to markup the proof steps.

\spfidea

The \spfidea macro allows to give a one-paragraph description of the proof idea.

\spfsketch

For one-line proof sketches, we use the \spfsketch macro, which takes the same optional argument as sproof and another one: a natural language text that sketches the proof.

\spfstep

Regular proof steps are marked up with the \spfstep macro, which takes an optional KeyVal argument for annotations. A proof step usually contains a local assertion (the text of the step) together with some kind of evidence that this can be derived from already established assertions.

\yield

See above

\spfjust

This evidence is marked up with the \spfjust macro in the stex-proofs package. This environment totally invisible to the formatted result; it wraps the text in the proof step that corresponds to the evidence (ideally, a semantically marked-up term).

\assumption

The \assumption macro allows to mark up a (justified) assumption.

\justarg

 ${\tt subproof}$

The subproof environment is used to mark up a subproof. This environment takes an optional KeyVal argument for semantic annotations and a second argument that allows to specify an introductory comment (just like in the proof environment). The method key can be used to give the name of the proof method executed to make this subproof.

\sproofend

Traditionally, the end of a mathematical proof is marked with a little box at the end of the last line of the proof (if there is space and on the end of the next line if there isn't), like so:

The stex-proofs package provides the \sproofend macro for this.

\sProofEndSymbol

If a different symbol for the proof end is to be used (e.g. q.e.d), then this can be obtained by specifying it using the \sProofEndSymbol configuration macro (e.g. by specifying \sProofEndSymbol{q.e.d}).

Some of the proof structuring macros above will insert proof end symbols for subproofs, in most cases, this is desirable to make the proof structure explicit, but sometimes this wastes space (especially, if a proof ends in a case analysis which will supply its own proof end marker). To suppress it locally, just set proofend={} in them or use use \sProofEndSymbol{}.

5.3 Highlighting and Presentation Customizations

The environments starting with s (i.e. smodule, sassertion, sexample, sdefinition, sparagraph and sproof) by default produce no additional output whatsoever (except for the environment content of course). Instead, the document that uses them (whether directly or e.g. via \inputref) can decide how these environments are supposed to look like.

The stexthm package defines some default customizations that can be used, but of course many existing LaTeX templates come with their own definition, theorem and similar environments that authors are supposed (or even required) to use. Their concrete syntax however is usually not compatible with all the additional arguments that STeX allows for semantic information.

Therefore we introduced the separate environments sdefinition etc. instead of using definition directly. We allow authors to specify how these environments should be styled via the commands stexpatch*.

\stexpatchmodule \stexpatchdefinition \stexpatchassertion \stexpatchexample \stexpatchparagraph \stexpatchproof All of these commands take one optional and two proper arguments, i.e. \stexpatch*[<type>] {<begin-code>}{<end-code>}.

After STEX reads and processes the optional arguments for these environments, (some of) their values are stored in the macros \s*<field> (i.e. sexampleid, \sassertionname, etc.). It then checks for all the values <type> in the type=-list, whether an \stexpatch*[<type>] for the current environment has been called. If it finds one, it uses the patches <begin-code> and <end-code> to mark up the current environment. If no patch for (any of) the type(s) is found, it checks whether and \stexpatch* was called without optional argument.

For example, if we want to use a predefined theorem environment for sassertions with type=theorem, we can do

1 \stexpatchassertion[theorem] {\begin{theorem}} {\end{theorem}}

...or, rather, since e.g. theorem-like environments defined using amsthm take an optional title as argument, we can do:

```
1 \stexpatchassertion[theorem]
2 {\ifx\sassertiontitle\@empty
3 \begin{theorem}
4 \else
5 \begin{theorem}[\sassertiontitle]
6 \fi}
7 {\end{theorem}}
```

Or, if we want *all kinds of* **sdefinitions** to use a predefined **definition**-environment irrespective of their **type=**, then we can issue the following customization patch:

```
1 \stexpatchdefinition
2 {\ifx\sdefinitiontitle\@empty
3 \begin{definition}
4 \else
5 \begin{definition}[\sdefinitiontitle]
6 \fi}
7 {\end{definition}}
```

\compemph
\varemph
\symrefemph
\defemph

Apart from the environments, we can control how STEX highlights variables, notation components, \symmets and \definiendums, respectively.

To do so, we simply redefine these four macros. For example, to highlight notation components (i.e. everything in a \comp) in blue, as in this document, we can do \def\compemph#1{\textcolor{blue}{#1}}. By default, \compemph et al do nothing.

\compemph@uri \varemph@uri \symrefemph@uri \defemph@uri For each of the four macros, there exists an additional macro that takes the full URI of the relevant symbol currently being highlighted as a second argument. That allows us to e.g. use pdf tooltips and links. For example, this document uses 5

```
1 \protected\def\symrefemph@uri#1#2{
2 \pdftooltip{
3 \srefsymuri{#2}{\symrefemph{#1}}}
4 }{
5 URI:~\detokenize{#2}
6 }
7 }
```

By default, \compemph@uri is simply defined as \compemph{#1} (analogously for the other three commands).

Chapter 6

Cross References

If we take features like \inputref and \mhinput (and the sfragment-environment, see subsection 7.2.1) 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 inputted 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.

The \sref command attempts to do precisely that. Unlike plain \ref, \autoref etc., \sref refers to not just a label, but instead a pair consisting of a label and the document in whose context we want to refer to it. Conversely, every document (i.e. standalone compilable .tex-file) keeps track of the "names" (Definition 3.1 etc.) for every label as determined in the context of the document, and stores them in a dedicated file \jobname.sref. Additionally, every document has a "reference name" (e.g. "the section on Foo"). This allows us to refer to "label x in document D" to yield "Definition 1 in the section on Foo". And of course, STEX can decide based on the current document to either refer to the label by its "full name" or directly as e.g. Definition 3.1 depending on whether the label occurs in the current document anyway (and link to it accordingly).

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, but allows for additional disambiguation between possibly duplicate labels across files, and
- (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 to provide a title for external references.

\STEXreftitle

For the latter, we can use \STEXreftitle. For example, the full \STEX documentation (stex-doc.tex) declares \STEXreftitle{the full \stex{3} documentation}, whereas the manual (stex-manual.tex) declares \STEXreftitle{the \stex{3} manual}.

\sref

```
\label{lem:context} $$ \left[ \operatorname{archive} \right], file=\left\langle file\right\rangle $$ $$ \left(\left\langle abel\right\rangle\right) \left[\operatorname{archive} \left\langle archive2\right\rangle, \operatorname{in} \left\langle document-context\right\rangle, \operatorname{cite} \left\langle citation\right\rangle $$ $$ $$ $$
```

This command references $\langle label \rangle$ (declared in $\langle file \rangle$ in $\langle archive1 \rangle$). If the object (section, figure, etc.) with that label occurs ultimately 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 $\langle document\text{-}context \rangle$ in $\langle archive2 \rangle$.

For example, the reference to the **sfragment**-environment above will appear as "subsection 7.2.1 (Introduction) in the STEX3 manual" if you are reading this in the package documentation for **stex-references** directly, but as a linked "subsection 7.2.1" in the full documentation or manual. This is achieved using

\sref[file=stex-document-structure] {sec:ds:intro} [in=../stex-manual]. If the optional cite= $\langle citation \rangle$ -argument is provided, \sref will ignore the document title as provided by the document context, and will instead use "in \cite{ $\langle citation \rangle$ }". For a further example, the following:

Part III

will say "Part III" (and link accordingly) in the full documentation, and "Part III (Extensions) in the full STEX3 documentation" everywhere else. This is achieved using \sref[file=../stex-doc] {part:extends}[in=../stex-doc].

\extref

```
\label{lem:context} $$\operatorname{archive=\langle archive1\rangle, file=\langle file\rangle]} $$ {\langle label\rangle}{ archive=\langle archive2\rangle, in=\langle document-context\rangle, cite=\langle citation\rangle}$$
```

The \extref-command behaves exactly like \sref, but takes required the document context argument and will always use it for generating the document text, regardless of whether the label occurs in the current document.

Chapter 7

Additional Packages

7.1 Tikzinput: Treating TIKZ code as images

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 $\tikzinput{\langle file\rangle}$ inputs the TIKZ file $\langle file\rangle$.tex; if not, only the graphicx package is loaded and $\tikzinput{\langle file\rangle}$ loads an image file $\langle file\rangle$. $\langle ext\rangle$ generated from $\langle file\rangle$.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}
3 \usetikzpackage{...}
4 \begin{document}
5 \begin{tikzpicture}
6 ...
7 \end{tikzpicture}
8 \end{document}
```

The standalone class is a minimal LATEX class that when loaded in a document that uses the standalone package: the preamble and the documenat 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 LATEX over it separately, e.g. for generating an image file from it.

\tikzinput \ctikzinput

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[\langle opt \rangle] \{\langle file \rangle\}\$ disregards the optional argument $\langle opt \rangle$ and inputs $\langle file \rangle$. tex via \tikzinput and resizes it to as specified in the width and height keys. If it is, $\tikzinput[\langle opt \rangle] \{\langle file \rangle\}\$ expands to $\tikzinput[\langle opt \rangle] \{\langle file \rangle\}\$.

\ctizkinput is a version of \tikzinput that is centered.

\mhtikzinput \cmhtikzinput \mhtizkinput 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, \cmhtizkinput 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.

7.2 Modular Document Structuring

7.2.1 Introduction

The document-structure package supplies an infrastructure for writing OMDoc documents in IATEX. This includes a simple structure sharing mechanism for STEX that allows to to move from a copy-and-paste document development model to a copy-and-reference model, which conserves space and simplifies document management. The augmented structure can be used by MKM systems for added-value services, either directly from the STEX sources, or after translation.

The document-structure package supplies macros and environments that allow to label document fragments and to reference them later in the same document or in other documents. In essence, this enhances the document-as-trees model to documents-as-directed-acyclic-graphs (DAG) model. This structure can be used by MKM systems for added-value services, either directly from the STEX sources, or after translation. Currently, trans-document referencing provided by this package can only be used in the STEX collection.

DAG models of documents allow to replace the "Copy and Paste" in the source document with a label-and-reference model where document are shared in the document source and the formatter does the copying during document formatting/presentation.

7.2.2 Package Options

The document-structure package accepts the following options:

| $class=\langle name \rangle$ | $load \langle name \rangle.cls instead of article.cls$ |
|---------------------------------|---|
| topsect= $\langle sect \rangle$ | The top-level sectioning level; the default for $\langle sect \rangle$ is section |

7.2.3 Document Fragments

sfragment

The structure of the document is given by nested sfragment environments. In the IATEX route, the sfragment environment is flexibly mapped to sectioning commands, inducing the proper sectioning level from the nesting of sfragment environments. Correspondingly, the sfragment environment takes an optional key/value argument for metadata followed by a regular argument for the (section) title of the sfragment. The optional metadata argument has the keys id for an identifier, creators and contributors for the Dublin Core metadata [DCM03]. The option short allows to give a short title for the generated section. If the title contains semantic macros, we need to give the loadmodules key (it needs no value). For instance we would have

```
1 \begin{smodule}{foo}
2 \symdef{bar}{B^a_r}
3 ...
4 \begin{sfragment}[id=sec.barderiv,loadmodules]
5 {Introducing $\protect\bar$ Derivations}
```

STEX automatically computes the sectioning level, from the nesting of sfragment environments.

But sometimes, we want to skip levels (e.g. to use a \subsection* as an introduction for a chapter).

blindfragment

Therefore the document-structure package provides a variant blindfragment that does not produce markup, but increments the sectioning level and logically groups document parts that belong together, but where traditional document markup relies on convention rather than explicit markup. The blindfragment environment is useful e.g. for creating frontmatter at the correct level. The example below shows a typical setup for the outer document structure of a book with parts and chapters.

```
1 \begin{document}
2 \begin{blindfragment}
3 \begin{blindfragment}
4 \begin{frontmatter}
5 \maketitle\newpage
6 \begin{sfragment}{Preface}
8 \end{sfragment}
9 \clearpage\setcounter{tocdepth}{4}\tableofcontents\clearpage
10 \end{frontmatter}
11 \end{blindfragment}
12 ... <<introductory remarks>>
13 \end{blindfragment}
14 \begin{sfragment}{Introduction}
15 ... <<intro>> ...
16 \end{sfragment}
17 \ldots << more chapters>> \ldots
18 \bibliographystyle{alpha}\bibliography{kwarc}
19 \end{document}
```

Here we use two levels of blindfragment:

- The outer one groups the introductory parts of the book (which we assume to have a sectioning hierarchy topping at the part level). This blindfragment makes sure that the introductory remarks become a "chapter" instead of a "part".
- The inner one groups the frontmatter³ and makes the preface of the book a section-level construct. The frontmatter environment also suppresses numbering as is traditional for prefaces.

\skipfragment

The \skipfragment "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.

 $^{^3}$ We shied away from redefining the **frontmatter** to induce a blindfragment, but this may be the "right" way to go in the future.

\currentsectionlevel \CurrentSectionLevel

The \currentsectionlevel macro supplies the name of the current sectioning level, e.g. "chapter", or "subsection". \CurrentSectionLevel is the capitalized variant. They are useful to write something like "In this \currentsectionlevel, we will..." in an sfragment environment, where we do not know which sectioning level we will end up.

7.2.4 Ending Documents Prematurely

\prematurestop \afterprematurestop For prematurely stopping the formatting of a document, STEX 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 [LMH].

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) <code>courseAcronym</code> and <code>courseTitle</code> instead of the text itself. The variables can then be set in the STEX preamble of the course notes file.

7.2.5 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 \useSGvar

 $\sc SGvar \{\langle vname \rangle\} \{\langle text \rangle\}$ to set the global variable $\langle vname \rangle$ to $\langle text \rangle$ and $\sc SGvar \{\langle vname \rangle\}$ to reference it.

\ifSGvar

With\ifSGvar we can test for the contents of a global variable: the macro call \ifSGvar{ $\langle vname \rangle$ }{ $\langle val \rangle$ }{ $\langle ctext \rangle$ } tests the content of the global variable $\langle vname \rangle$, only if (after expansion) it is equal to $\langle val \rangle$, the conditional text $\langle ctext \rangle$ is formatted.

7.3 Slides and Course Notes

7.3.1 Introduction

The notesslides document class is derived from beamer.cls [Tana], 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.

7.3.2 Package Options

The notesslides class takes a variety of class options:

slides notes The options slides and notes switch between slides mode and notes mode (see subsection 7.3.3).

sectocframes

If the option sectocframes is given, then for the sfragments, special frames with the sfragment title (and number) are generated.

frameimages
fiboxed

If the option frameimages is set, then slide mode also shows the \frameimage-generated frames (see ??). If also the fiboxed option is given, the slides are surrounded by a box.

7.3.3 Notes and Slides

frame Slides are represented with the frame environment just like in the beamer class, see [Tanb] for details.

note 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 IATEX 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}
0 ...
```

```
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 funnily (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}.

nexample, nsproof, nassertion

There are some environments that tend to occur at the top-level of note environments. We make convenience versions of these: e.g. the nparagraph environment is just an sparagraph inside a note environment (but looks nicer in the source, since it avoids one level of source indenting). Similarly, we have the nfragment, ndefinition, nexample, nsproof, and nassertion environments.

7.3.4 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 hight is variable) is



The footer line can be customized. In particular the logos.

\setslidelogo

The default logo provided by the notesslides package is the ST_EX logo it can be customized using $\ensuremath{\texttt{NetSlidelogo}}\{\langle logo \ name \rangle\}$.

\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 Attribuition-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.

7.3.5 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 ST_FX notes.

\frameimage \mhframeimage

In this case we can use $\frac{\langle opt \rangle}{\langle ath \rangle}$, where $\langle opt \rangle$ are the options of $\frac{\langle opt \rangle}{\langle ath \rangle}$ is the file path (extension can be left off like in $\frac{\langle opt \rangle}{\langle ath \rangle}$). 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:

60

7.3.6 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{ $\langle ref \rangle$ }{ $\langle path \rangle$ }{ $\langle text \rangle$ } 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 \printexcursion \excursionref

Here $\activateexcursion{\langle path \rangle}$ augments the \printexcursions macro by a call \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 $\ensuremath{\texttt{\colored}}$ 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= $\langle id \rangle$, intro= $\langle path \rangle$] 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.

7.4 Representing Problems and Solutions

7.4.1 Introduction

The problem package supplies an infrastructure that allows specify problem. Problems are text fragments that come with auxiliary functions: hints, notes, and solutions⁴. 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.

7.4.2 Problems and Solutions

solutions notes hints gnotes pts min boxed test 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.

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

The main environment provided by the problempackage 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 40 Input:

⁴ for the moment multiple choice problems are not supported, but may well be in a future version

```
\documentclass{article}
  \usepackage[solutions,hints,pts,min]{problem}
3 \begin{document}
    \begin{sproblem}[id=elefants,pts=10,min=2,title=Fitting Elefants]
      How many Elefants can you fit into a Volkswagen beetle?
      \begin{hint}
        Think positively, this is simple!
      \end{hint}
      \begin{exnote}
10
        Justify your answer
      \end{exnote}
11
12 \begin{solution} [for=elefants]
13
    Four, two in the front seats, and two in the back.
    \begin{gnote}
      if they do not give the justification deduct 5 pts
16
   \end{gnote}
17 \end{solution}
18 \end{sproblem}
19 \end{document}
```

Output:

Problem 7.4.1 (Fitting Elefants) How many Elefants can you fit into a Volkswagen beetle? Hint: Think positively, this is simple! Note: Justify your answer Solution: Four, two in the front seats, and two in the back. Grading: if they do not give the justification deduct 5 pts

solution

The solution environment can be 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 reference 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, exnote, gnote

The hint and exnote environments can be used in a problem environment to give hints and to make notes that elaborate certain aspects of the problem. The gnote (grading notes) environment can be used to document situations that may arise in grading.

\startsolutions \stopsolutions Sometimes we would like to locally override the solutions option we have given to the package. To turn on solutions we use the \startsolutions, to turn them off, \stopsolutions. These two can be used at any point in the documents.

\ifsolutions

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 \ifsolutions conditional.

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

\mcc

 $\mbox{\colored} \langle keyvals \rangle \mbox{\colored} \langle text \rangle \mbox{\colored}$ takes an optional key/value argument $\langle keyvals \rangle$ for choice metadata and a required argument $\langle text \rangle$ 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 41

```
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=Nooooooooo,feedback=that is for Java]{public static void}
9 \end{mcb}
10 \end{sproblem}
```

Output:

| Problem 7.4.2 (Functions) What is the keyword to introduce a function definition in python? | |
|---|--|
| □ def Correct! | |
| \Box function Wrong! that is for C and $C++$ | |
| ☐ fun Wrong! that is for Standard ML | |
| □ public static void Wrong! that is for Java | |

.

In "exam mode" where disable solutions (here via \stopsolutions)

Example 42

```
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:

```
Problem 7.4.3 (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 $\operatorname{exam}/\operatorname{quiz}$).

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 takes⁶ an a single argument, which contains a concrete solution (i.e. a number, a string, ...), which generates a fill-in-box in test mode:

Example 43 Input: \stopsolutions \begin{sproblem}[id=elefants.fillin,title=Fitting Elefants] How many Elefants can you fit into a Volkswagen beetle? \fillinsol{4} 4 \end{sproblem} Output: Problem 7.4.4 (Fitting Elefants) How many Elefants can you fit into a Volkswagen beetle? and the actual solution in solutions mode: Example 44 Input: \begin{sproblem}[id=elefants.fillin,title=Fitting Elefants] How many Elefants can you fit into a Volkswagen beetle? \fillinsol{4} \end{sproblem} Output: Problem 7.4.5 (Fitting Elefants) How many Elefants can you fit into a Volkswagen beetle?

Obviously, the 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. ⁷

7.4.4 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

EdN:7

 $^{^7{}m EDNotes}$: 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 im 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.

7.5 Homeworks, Quizzes and Exams

7.5.1 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.

7.5.2 Package Options

solutions notes hints gnotes pts min The hwexam package and class take the options solutions, notes, hints, gnotes, pts, 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 LATEX source.

7.5.3 Assignments

assignment number

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 — the ordinal of the assignment environment), title (for the assignment title; this is referenced in the title of the assignment sheet), type (for the assignment type; e.g. "quiz", or "homework"), given (for the date the assignment was given), and due (for the date the assignment is due).

title type given

7.5.4 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.

7.5.5 Typesetting Exams

\testspace \testnewpage \testemptypage \testspace takes an argument that expands to a dimension, and leaves vertical space accordingly. \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.

testheading duration min

reqpts

Finally, 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.

1 \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)

2022-08-08

You have one hour (sharp) for the test;

Write the solutions to the sheet.

The estimated time for solving this exam is 60 minutes, leaving you 0 minutes for revising your exam.

You can reach 40 points if you solve all problems. You will only need 27 points for a perfect score, i.e. 13 points are bonus points.

You have ample time, so take it slow and avoid rushing to mistakes!

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. | 7.4.1 | 7.4.2 | 7.4.3 | 7.4.4 | 7.4.5 | 1.1 | 2.1 | 2.2 | 2.3 | 3.1 | 3.2 | 3.3 | Sum | grade |
| total | 10 | | | | | 4 | 4 | 6 | 6 | 4 | 4 | 2 | 40 | |
| reached | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |

good luck

8

EdN:8

 $^{^8\}mathrm{EdNote}\colon$ MK: The first three "problems" come from the stex examples above, how do we get rid of this?

Part II Documentation

STEX-Basics

This sub package provides general set up code, auxiliary methods and abstractions for xhtml annotations.

8.1 Macros and Environments

\sTeX Both print this STEX logo. \stex

\stex_debug:nn

 $\t (log-prefix) { (message)}$

Logs $\langle message \rangle$, if the package option debug contains $\langle log\text{-}prefix \rangle$.

8.1.1 HTML Annotations

\if@latexml [ATE]

LATEXML conditional for LATEXML

 LATEXX3 conditionals for LATEXML.

 $\stex_if_do_html_p: \star \\ stex_if_do_html: \underline{TF} \star$

Whether to currently produce any HTML annotations (can be false in some advanced structuring environments, for example)

\stex_suppress_html:n

Temporarily disables HTML annotations in its argument code

We have four macros for annotating generated HTML (via LATEXML or $R_{\rm US}T_{\rm E}X)$ with attributes:

Annotates the HTML generated by $\langle content \rangle$ with

behaves like $\stex_annotate:nnn \{\langle property \rangle\} \{\langle resource \rangle\} \{\langle content \rangle\}.$

stex_annotate_env

```
8.1.2 Babel Languages
```

\c_stex_languages_prop
\c_stex_language_abbrevs_prop

Map language abbreviations to their full babel names and vice versa. e.g. \c_stex_languages_prop{en} yields english, and \c_stex_language_abbrevs_prop{english} yields en.

8.1.3 Auxiliary Methods

\stex_deactivate_macro:Nn \stex_reactivate_macro:N

 $\verb|\stex_deactivate_macro:Nn| \langle cs \rangle \{ \langle environments \rangle \}|$

Makes the macro $\langle cs \rangle$ throw an error, indicating that it is only allowed in the context of $\langle environments \rangle$.

 $\scalebox{$\sc s$}$ reactivates it again, i.e. this happens ideally in the $\scalebox{$\sc begin$}$ -code of the associated environments.

\ignorespacesandpars

ignores white space characters and \par control sequences. Expands tokens in the process.

STEX-MathHub

This sub package provides code for handling STEX archives, files, file paths and related methods.

9.1 Macros and Environments

\stex_kpsewhich:n

\stex_kpsewhich:n executes kpsewhich and stores the return in \l_stex_kpsewhich_return_str. This does not require shell escaping.

9.1.1 Files, Paths, URIs

\stex_path_from_string:Nn

 $\stex_path_from_string:Nn \langle path-variable \rangle \{\langle string \rangle\}$

turns the $\langle string \rangle$ into a path by splitting it at /-characters and stores the result in $\langle path-variable \rangle$. Also applies $\text{stex_path_canonicalize:N}$.

\stex_path_to_string:NN \stex_path_to_string:N

The inverse; turns a path into a string and stores it in the second argument variable, or leaves it in the input stream.

\stex_path_canonicalize:N

Canonicalizes the path provided; in particular, resolves . and .. path segments.

\stex_path_if_absolute_p:N *\stex_path_if_absolute:NTF *

Checks whether the path provided is absolute, i.e. starts with an empty segment

\c_stex_pwd_seq
\c_stex_pwd_str
\c_stex_mainfile_seq
\c_stex_mainfile_str

Store the current working directory as path-sequence and string, respectively, and the (heuristically guessed) full path to the main file, based on the PWD and \jobname.

\g_stex_currentfile_seq

The file being currently processed (respecting \input etc.)

\stex_filestack_push:n
\stex_filestack_pop:

Push and pop (repsectively) a file path to the file stack, to keep track of the current file. Are called in hooks file/before and file/after, respectively.

9.1.2 MathHub Archives

\mathhub
\c_stex_mathhub_seq
\c_stex_mathhub_str

We determine the path to the local MathHub folder via one of four means, in order of precedence:

- 1. The mathhub package option, or
- 2. the \mathhub-macro, if it has been defined before the \usepackage{stex}-statement, or
- 3. the MATHHUB system variable, or
- 4. a path specified in ~/.stex/mathhub.path.

In all four cases, \c_stex_mathhub_seq and \c_stex_mathhub_str are set accordingly.

\l_stex_current_repository_prop

Always points to the *current* MathHub repository (if we currently are in one). Has the following fields corresponding to the entries in the MANIFEST.MF-file:

id: The name of the archive, including its group (e.g. smglom/calculus),

ns: The content namespace (for modules and symbols),

narr: the narration namespace (for document references),

docurl: The URL that is used as a basis for external references,

deps: All archives that this archive depends on (currently not in use).

\stex_set_current_repository:n

Sets the current repository to the one with the provided ID. calls __stex_mathhub_-do_manifest:n, so works whether this repository's MANIFEST.MF-file has already been read or not.

\stex_require_repository:n

Calls __stex_mathhub_do_manifest:n iff the corresponding archive property list does not already exist, and adds a corresponding definition to the .sms-file.

\stex_in_repository:nn

 $\stex_in_repository:nn{\langle repository-name \rangle}{\langle code \rangle}$

Change the current repository to $\{\langle repository-name \rangle\}$ (or not, if $\{\langle repository-name \rangle\}$ is empty), and passes its ID on to $\{\langle code \rangle\}$ as #1. Switches back to the previous repository after executing $\{\langle code \rangle\}$.

9.1.3 Using Content in Archives

\mhpath *

 $\mbox{\colored} \mbox{\colored} \mbox{\color$

Expands to the full path of file $\langle filename \rangle$ in repository $\langle archive\text{-}ID \rangle$. Does not check whether the file or the repository exist.

\inputref
\mhinput

 $\input{ref} [\langle archive-ID \rangle] \{\langle filename \rangle\}$

Both \input the file $\langle filename \rangle$ in archive $\langle archive\text{-}ID \rangle$ (relative to the source-subdirectory). \mhinput does so directly. \inputref does so within an \begingroup...\endgroup-block, and skips it in html-mode, inserting a reference to the file instead.

Both also set \ifinputref to true.

\addmhbibresource

 $\input{ref} [\langle archive-ID \rangle] {\langle filename \rangle}$

Adds a .bib-file $\langle filename \rangle$ in archive $\langle archive\text{-}ID \rangle$ (relative to the top-directory of the archive!).

\libinput

 $\left\langle filename \right\rangle$

Inputs $\langle filename \rangle$.tex from the lib folders in the current archive and the meta-infarchive of the current archive group(s) (if existent) in descending order. Throws an error if no file by that name exists in any of the relevant lib-folders.

\libusepackage

 $\label{libusepackage} \label{libusepackage} $$ \left(args \right) \left(filename \right) \right) $$$

Like $\ \$ but looks for .sty-files and calls $\ \$ instead of $\$ input.

Throws an error, if none or more than one suitable package file is found.

\mhgraphics \cmhgraphics

If the graphicx package is loaded, these macros are defined at \begin{document}.

\mhgraphics takes the same arguments as \includegraphics, with the additional optional key mhrepos. It then resolves the file path in \mhgraphics[mhrepos=Foo/Bar]{foo/bar.png} relative to the source-folder of the Foo/Bar-archive.

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

\lstinputmhlisting \clstinputmhlisting Like \mhgraphics, but only defined if the listings-package is loaded, and with \lstinputlisting instead of \includegraphics.

STEX-References

This sub package contains code related to links and cross-references

10.1 Macros and Environments

\STEXreftitle

 $\TEXreftitle{\langle some \ title \rangle}$

Sets the title of the current document to $\langle some\ title \rangle$. A reference to the current document from $some\ other$ document will then be displayed accordingly. e.g. if \STEXreftitle{foo book} is called, then referencing Definition 3.5 in this document in another document will display Definition 3.5 in foo book.

\stex_get_document_uri:

Computes the current document uri from the current archive's narr-field and its location relative to the archive's source-directory. Reference targets are computed from this URI and the reference-id.

\l_stex_current_docns_str

Stores its result in \l_stex_current_docns_str

\stex_get_document_url:

Computes the current URL from the current archive's docurl-field and its location relative to the archive's source-directory. Reference targets are computed from this URL and the reference-id, if this document is only included in SMS mode.

\l_stex_current_docurl_str

Stores its result in \l_stex_current_docurl_str

10.1.1 Setting Reference Targets

\stex_ref_new_doc_target:n

 $\stex_ref_new_doc_target:n{\langle id \rangle}$

Sets a new reference target with id $\langle id \rangle$.

\stex_ref_new_sym_target:n

 $\stex_ref_new_sym_target:n{\langle uri \rangle}$

Sets a new reference target for the symbol $\langle uri \rangle$.

10.1.2 Using References

\sref

 $\left[\left\langle opt-args\right\rangle\right]\left\{\left\langle id\right\rangle\right\}$

References the label with if $\langle id \rangle$. Optional arguments: TODO

\srefsym

 $\verb|\srefsym[\langle opt-args\rangle]{\langle symbol\rangle}|$

Like \sref, but references the *canonical label* for the provided symbol. The canonical target is the last of the following occurring in the document:

- A \definiendum or \definame for $\langle symbol \rangle$,
- The sassertion, sexample or sparagraph with for= $\langle symbol \rangle$ that generated $\langle symbol \rangle$ in the first place, or
- A \sparagraph with type=symdoc and for= $\langle symbol \rangle$.

\srefsymuri

 $\verb|\srefsymuri{|\langle \mathit{URI} \rangle|} {\langle \mathit{text} \rangle}|$

A convenient short-hand for \srefsym[linktext={text}]{URI}, but requires the first argument to be a full URI already. Intended to be used in e.g. \compemph@uri, \defemph@uri, etc.

STEX-Modules

This sub package contains code related to Modules

11.1 Macros and Environments

The content of a module with uri $\langle \langle URI \rangle \rangle$ is stored in four macros. All modifications of these macros are global:

\c_stex_module_<URI>_prop

A property list with the following fields:

name The name of the module,

ns the namespace in field ns,

file the file containing the module, as a sequence of path fragments

lang the module's language,

sig the language of the signature module, if the current file is a translation from some other language,

deprecate if this module is deprecated, the module that replaces it,

meta the metatheory of the module.

\c_stex_module_<URI>_code

The code to execute when this module is activated (i.e. imported), e.g. to set all the semantic macros, notations, etc.

\c_stex_module_<URI>_constants

The names of all constants declared in the module

\c_stex_module_<URI>_constants

The full URIs of all modules imported in this module

\l_stex_current_module_str

\l_stex_current_module_str always contains the URI of the current module (if existent).

\l_stex_all_modules_seq

Stores full URIs for all modules currently in scope.

 $\stex_if_in_module_p: \star$

Conditional for whether we are currently in a module

 $\stex_if_in_module: TF *$

\stex_if_module_exists_p:n *

 $\stex_if_module_exists:n_{\overline{TF}} \star$

Conditional for whether a module with the provided URI is already known.

\stex_add_to_current_module:n \STEXexport

Adds the provided tokens to the _code control sequence of the current module. \stex_add_to_current_module:n is used internally, \STEXexport is intended for

\stex_add_to_current_module:n is used internally, \STEXexport is intended for users and additionally executes the provided code immediately.

\stex_add_constant_to_current_module:n

Adds the declaration with the provided name to the **_constants** control sequence of the current module.

\stex_add_import_to_current_module:n

Adds the module with the provided full URI to the _imports control sequence of the current module.

\stex_collect_imports:n

Iterates over all imports of the provided (full URI of a) module and stores them as a topologically sorted list – including the provided module as the last element – in \l_stex_collect_imports_seq

\stex_do_up_to_module:n

Code that is exported from module (such as symbol declarations) should be local to the current module. For that reason, ideally all symbol declarations and similar commands should be called directly in the module environment, however, that is not always feasible, e.g. in structural features or sparapraphs. \stex_do_up_to_module therefore executes the provided code repeatedly in an \aftergroup up until the group level is equal to that of the innermost smodule environment.

\stex_modules_current_namespace:

Computes the current namespace as follows:

If the current file is .../source/sub/file.tex in some archive with namespace http://some.namespace/foo, then the namespace of is http://some.namespace/foo/sub/file. Otherwise, the namespace is the absolute file path of the current file (i.e. starting with file:///).

The result is stored in \l_stex_module_ns_str. Additionally, the sub path relative to the current repository is stored in \l_stex_module_subpath_str.

11.1.1 The smodule environment

module $\lceil \pmod{module} \lceil \langle options \rangle \rceil \{\langle name \rangle \}$

Opens a new module with name $\langle name \rangle$. Options are:

title $(\langle token \ list \rangle)$ to display in customizations.

type $(\langle string \rangle *)$ for use in customizations.

deprecate $(\langle module \rangle)$ if set, will throw a warning when loaded, urging to use $\langle module \rangle$ instead.

id $(\langle string \rangle)$ for cross-referencing.

ns $(\langle URI \rangle)$ the namespace to use. Should not be used, unless you know precisely what you're doing. If not explicitly set, is computed using $\text{stex_modules_current_namespace:}$.

lang $(\langle language \rangle)$ if not set, computed from the current file name (e.g. foo.en.tex).

sig (\language\rangle) 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.

creators ($\langle string \rangle *$) names of the creators.

contributors ($\langle string \rangle *$) names of contributors.

srccite $(\langle string \rangle)$ a source citation for the content of this module.

\stex_module_setup:nn

 $\stex_module_setup:nn{\langle params \rangle}{\langle name \rangle}$

Sets up a new module with name $\langle name \rangle$ and optional parameters $\langle params \rangle$. In particular, sets \l_stex_current_module_str appropriately.

\stexpatchmodule

\stexpatchmodule $[\langle type \rangle]$ { $\langle begincode \rangle$ } { $\langle endcode \rangle$ }

Customizes the presentation for those smodule-environments with type= $\langle type \rangle$, or all others if no $\langle type \rangle$ is given.

\STEXModule

\STEXModule $\{\langle fragment \rangle\}$

Attempts to find a module whose URI ends with $\langle fragment \rangle$ in the current scope and passes the full URI on to $\text{stex_invoke_module:n.}$

\stex_invoke_module:n

Invoked by \STEXModule. Needs to be followed either by !\macro or ?{ $\langle symbolname \rangle$ }. In the first case, it stores the full URI in \macro; in the second case, it invokes the symbol $\langle symbolname \rangle$ in the selected module.

\stex_activate_module:n

Activate the module with the provided URI; i.e. executes all macro code of the module's <code>_code-</code>macro (does nothing if the module is already activated in the current context) and adds the module to <code>\l_stex_all_modules_seq</code>.

STeX-Module Inheritance

Code related to Module Inheritance, in particular sms mode.

12.1 Macros and Environments

12.1.1 SMS Mode

"SMS Mode" is used when loading modules from external tex files. It deactivates any output and ignores all T_EX commands not explicitly allowed via the following lists – all of which either declare module content or are needed in order to declare module content:

$\g_stex_smsmode_allowedmacros_tl$

Macros that are executed as is; i.e. sms mode continues immediately after. These macros may not take any arguments or otherwise gobble tokens.

 $Initially: \verb|\makeatletter|, \verb|\makeatother|, \verb|\ExplSyntaxOn|, \verb|\ExplSyntaxOff|.$

$\verb|\g_stex_smsmode_allowedmacros_escape_tl|\\$

Macros that are executed and potentially gobble up further tokens. These macros need to make sure, that the very last token they ultimately expand to is \stex_smsmode_do:.

Initially: \symdecl, \notation, \symdef, \importmodule, \STEXexport, \inlineass, \inlinedef, \inlineex, \endinput, \setnotation, \copynotation.

$\g_stex_smsmode_allowedenvs_seq$

The names of environments that should be allowed in SMS mode. The corresponding \begin-statements are treated like the macros in \g_stex_smsmode_allowedmacros_-escape_tl, so \stex_smsmode_do: needs to be the last token in the \begin-code. Since \end-statements take no arguments anyway, those are called directly and sms mode continues afterwards.

 $Initially: \verb|smodule|, copymodule|, interpretmodule|, \verb|sdefinition|, sexample|, \verb|sassertion|, sparagraph|.$

\stex_if_smsmode_p: *
\stex_if_smsmode:TF *

Tests whether SMS mode is currently active.

\stex_file_in_smsmode:nn

 $\stex_in_smsmode:nn {\langle filename \rangle} {\langle code \rangle}$

Executes $\langle code \rangle$ in SMS mode, followed by the content of $\langle filename \rangle$. $\langle code \rangle$ can be used e.g. to set the current repository, and is executed within a new tex group, and the same group as the file content.

\stex_smsmode_do:

Starts gobbling tokens until one is encountered that is allowed in SMS mode.

12.1.2 Imports and Inheritance

\importmodule

 $\verb|\importmodule[\langle archive-ID\rangle]{\langle module-path\rangle}|$

Imports a module by reading it from a file and "activating" it. STEX determines the module and its containing file by passing its arguments on to \stex_import_module_-path:nn.

\usemodule

 $\verb|\importmodule[\langle archive-ID\rangle] {\langle module-path\rangle}|$

Like \importmodule, but does not export its contents; i.e. including the current module will not activate the used module

 $\stex_import_module_uri:nn {\langle archive-ID \rangle} {\langle module-path \rangle}$

Determines the URI of a module by splitting $\langle module\text{-}path \rangle$ into $\langle path \rangle$? $\langle name \rangle$. If $\langle module\text{-}path \rangle$ does not contain a ?-character, we consider it to be the $\langle name \rangle$, and $\langle path \rangle$ to be empty.

If $\langle archive\text{-}ID \rangle$ is empty, it is automatically set to the ID of the current archive (if one exists).

1. If $\langle archive\text{-}ID \rangle$ is empty:

(a) If $\langle path \rangle$ is empty, then $\langle name \rangle$ must have been declared earlier in the same file and retrievable from \g_stex_modules_in_file_seq, or a file with name $\langle name \rangle . \langle lang \rangle$.tex must exist in the same folder, containing a module $\langle name \rangle$.

That module should have the same namespace as the current one.

(b) If $\langle path \rangle$ is not empty, it must point to the relative path of the containing file as well as the namespace.

2. Otherwise:

(a) If $\langle path \rangle$ is empty, then $\langle name \rangle$ must have been declared earlier in the same file and retrievable from \g_stex_modules_in_file_seq, or a file with name $\langle name \rangle$. $\langle lang \rangle$.tex must exist in the top source folder of the archive, containing a module $\langle name \rangle$.

That module should lie directly in the namespace of the archive.

(b) If \(\rangle path \rangle\) is not empty, it must point to the path of the containing file as well as the namespace, relative to the namespace of the archive.

If a module by that namespace exists, it is returned. Otherwise, we call \stex_require_module:nn on the source directory of the archive to find the file.

\l_stex_import_name_str
\l_stex_import_archive_str
\l_stex_import_path_str
\l_stex_import_ns_str

stores the result in these four variables.

 $\stex_import_require_module:nnnn = {\langle ns \rangle} {\langle archive-ID \rangle} {\langle path \rangle} {\langle name \rangle}$

Checks whether a module with URI $\langle ns \rangle$? $\langle name \rangle$ already exists. If not, it looks for a plausible file that declares a module with that URI.

Finally, activates that module by executing its _code-macro.

STEX-Symbols

Code related to symbol declarations and notations

13.1 Macros and Environments

\symdecl

 $\symdecl{\langle macroname \rangle}[\langle args \rangle]$

Declares a new symbol with semantic macro \macroname. Optional arguments are:

- name: An (OMDoc) name. By default equal to $\langle macroname \rangle$.
- type: An (ideally semantic) term, representing a type. Not used by SIEX, but passed on to MMT for semantic services.
- def: An (ideally semantic) term, representing a definiens. Not used by STEX, but passed on to MMT for semantic services.
- local: A boolean (by default false). If set, this declaration will not be added to the module content, i.e. importing the current module will not make this declaration available.
- args: Specifies the "signature" of the semantic macro. Can be either an integer $0 \le n \le 9$, or a (more precise) sequence of the following characters:
 - i a "normal" argument, e.g. \symdecl{plus}[args=ii] allows for \plus{2}{2}.
 - a an *associative* argument; i.e. a sequence of arbitrarily many arguments provided as a comma-separated list, e.g. \symdecl{plus}[args=a] allows for \plus{2,2,2}.
 - b a variable argument. Is treated by STEX like an i-argument, but an application is turned into an OMBind in OMDoc, binding the provided variable in the subsequent arguments of the operator; e.g. \symdecl{forall}[args=bi] allows for \forall{x\in\Nat}{x\geq0}.

\stex_symdecl_do:n

Implements the core functionality of \symdecl, and is called by \symdecl and \symdef. Ultimately stores the symbol $\langle URI \rangle$ in the property list \l_stex_symdecl_ $\langle URI \rangle$ _prop with fields:

- name (string),
- module (string),
- notations (sequence of strings; initially empty),
- local (boolean),
- type (token list),
- args (string of is, as and bs),
- arity (integer string),
- assocs (integer string; number of associative arguments),

\stex_all_symbols:n

Iterates over all currently available symbols. Requires two \seq_map_break: to break fully.

\stex_get_symbol:n

Computes the full URI of a symbol from a macro argument, e.g. the macro name, the macro itself, the full URI...

\notation

 $\notation[\langle args \rangle] \{\langle symbol \rangle\} \{\langle notations^+ \rangle\}$

Introduces a new notation for $\langle symbol \rangle$, see \stex_notation_do:nn

\stex_notation_do:nn

 $\stex_notation_do:nn\{\langle \mathit{URI}\rangle\}\{\langle notations^+\rangle\}$

Implements the core functionality of \notation , and is called by \notation and \symdef .

Ultimately stores the notation in the property list \g_stex_notation_ $\langle \mathit{URI}\rangle$ # $\langle \mathit{variant}\rangle$ # $\langle \mathit{lang}\rangle$ _prop with fields:

- symbol (URI string),
- language (string),
- variant (string),
- opprec (integer string),
- argprecs (sequence of integer strings)

\symdef

 $\symdef[\langle args \rangle] \{\langle symbol \rangle\} \{\langle notations^+ \rangle\}$

Combines \symdecl and \notation by introducing a new symbol and assigning a new notation for it.

ST_EX-Terms

Code related to symbolic expressions, typesetting notations, notation components, etc.

14.1 Macros and Environments

\STEXsymbol

Uses \stex_get_symbol:n to find the symbol denoted by the first argument and passes the result on to \stex_invoke_symbol:n

\symref

 $\symref{\langle symbol \rangle} {\langle text \rangle}$

shortcut for $\STEXsymbol{\langle symbol \rangle}! [\langle text \rangle]$

\stex_invoke_symbol:n

Executes a semantic macro. Outside of math mode or if followed by *, it continues to \stex_term_custom:nn. In math mode, it uses the default or optionally provided notation of the associated symbol.

If followed by !, it will invoke the symbol *itself* rather than its application (and continue to \stex_term_custom:nn), i.e. it allows to refer to \plus![addition] as an operation, rather than \plus[addition of]{some}{terms}.

\STEXInternalTermMathOMSiiii \langle URI \rangle \fragmathOMAiiii \STEXInternalTermMathOMBiiii \STEXInternalTermMathOMBiiii

 $\langle \mathit{URI} \rangle \langle \mathit{fragment} \rangle \langle \mathit{precedence} \rangle \langle \mathit{body} \rangle$

Annotates $\langle body \rangle$ as an OMDoc-term (OMID, OMA or OMBIND, respectively) with head symbol $\langle URI \rangle$, generated by the specific notation $\langle fragment \rangle$ with (upwards) operator precedence $\langle precedence \rangle$. Inserts parentheses according to the current downwards precedence and operator precedence.

Annotates $\langle body \rangle$ as the $\langle int \rangle$ th argument of the current OMA or OMBIND, with (downwards) argument precedence $\langle prec \rangle$.

Annotates $\langle body \rangle$ as the $\langle int \rangle$ th (associative) sequence argument (as comma-separated list of terms) of the current OMA or OMBIND, with (downwards) argument precedence $\langle prec \rangle$ and associative notation $\langle notation \rangle$.

\infprec \neginfprec

Maximal and minimal notation precedences.

\dobrackets

\dobrackets $\{\langle body \rangle\}$

Puts $\langle body \rangle$ in parentheses; scaled if in display mode unscaled otherwise. Uses the current STEX brackets (by default (and)), which can be changed temporarily using \withbrackets.

\withbrackets

\withbrackets $\langle left \rangle \langle right \rangle \{\langle body \rangle\}$

Temporarily (i.e. within $\langle body \rangle$) sets the brackets used by STEX for automated bracketing (by default (and)) to $\langle left \rangle$ and $\langle right \rangle$.

Note that $\langle \mathit{left} \rangle$ and $\langle \mathit{right} \rangle$ need to be allowed after \left and \right in displaymode.

\stex_term_custom:nn

 $\stex_term_custom:nn{\langle \mathit{URI} \rangle}{\langle \mathit{args} \rangle}$

Implements custom one-time notation. Invoked by \stex_invoke_symbol:n in text mode, or if followed by * in math mode, or whenever followed by !.

\comp
\compemph
\compemph@uri
\defemph@uri
\symrefemph
\symrefemph
\rangle
\varemph

\varemph@uri

 $\{\langle args \rangle\}$

Marks $\langle args \rangle$ as a notation component of the current symbol for highlighting, linking, etc.

The precise behavior is governed by \@comp, which takes as additional argument the URI of the current symbol. By default, \@comp adds the URI as a PDF tooltip and colors the highlighted part in blue.

\@defemph behaves like \@comp, and can be similarly redefined, but marks an expression as definiendum (used by \definiendum)

\STEXinvisible

Exports its argument as OMDoc (invisible), but does not produce PDF output. Useful e.g. for semantic macros that take arguments that are not part of the symbolic notation.

\ellipses

TODO

STEX-Structural Features

Code related to structural features

15.1 Macros and Environments

15.1.1 Structures

mathstructure TODO

STEX-Statements

Code related to statements, e.g. definitions, theorems

16.1 Macros and Environments

Declares $\langle text \rangle$ to be a (natural language, encyclopaedic) description of $\{\langle symbols \rangle\}$ (a comma separated list of symbol identifiers).

STEX-Proofs: Structural Markup for Proofs

ST_EX -Metatheory

18.1 Symbols

Part III Extensions

Tikzinput: Treating TIKZ code as images

19.1 Macros and Environments

document-structure: Semantic Markup for Open Mathematical Documents in LATEX

NotesSlides – Slides and Course Notes

problem.sty: An Infrastructure for formatting Problems

hwexam.sty/cls: An
Infrastructure for formatting
Assignments and Exams

 ${\bf Part~IV} \\ {\bf Implementation}$

STEX

-Basics Implementation

24.1 The STEXDocument Class

The stex document class is pretty straight-forward: It largely extends the standalone package and loads the stex package, passing all provided options on to the package.

```
3 %%%%%%%%%%%%%%%
                                                               basics.dtx
                                                                                                             5 \RequirePackage{expl3,13keys2e}
       \ProvidesExplClass{stex}{2022/08/08}{3.2.0}{sTeX document class}
 8 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{stex}}
       \ProcessOptions
       \bool_set_true:N \c_stex_document_class_bool
       \RequirePackage{stex}
       \stex_html_backend:TF {
              \LoadClass{article}
16
17 }{
               \LoadClass[border=1px,varwidth,crop=false]{standalone}
               \setlength\textwidth{15cm}
19
20 }
       \RequirePackage{standalone}
21
22
24 \clist_if_empty:NT \c_stex_languages_clist {
              \seq_get_right:NN \g_stex_currentfile_seq \l_tmpa_str
              \ensuremath{\verb|seq_pop_right:NN||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\verb|l_tmpa_seq||} \ensuremath{\ensuremath{l_tmpa_seq||}} \ensuremath{\ensuremath{l_tmpa_
27
              \exp_args:No \str_if_eq:nnF \l_tmpa_str {tex} {
28
                     \exp_args:No \str_if_eq:nnF \l_tmpa_str {dtx} {
29
                             \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq \l_tmpa_str
```

```
}
31
32
    \seq_pop_left:NN \l_tmpa_seq \l_tmpa_str % <filename>
33
    \seq_if_empty:NF \l_tmpa_seq { %remaining element should be [<something>.]language
      \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
35
      \prop_if_in:NoT \c_stex_languages_prop \l_tmpa_str {
36
        \stex_debug:nn{language} {Language~\l_tmpa_str~
37
          inferred~from~file~name}
38
        \exp_args:NNo \stex_set_language:Nn \l_tmpa_str \l_tmpa_str
39
40
    }
41
42 }
43 (/cls)
```

24.2 Preliminaries

```
44 (*package)
        basics.dtx
                                       48 \RequirePackage{expl3,13keys2e,1txcmds}
          \ProvidesExplPackage{stex}{2022/08/08}{3.2.0}{sTeX package}
        51 \bool_if_exist:NF \c_stex_document_class_bool {
            \verb|\bool_set_false:N \c_stex_document_class_bool|
            \RequirePackage{standalone}
        54 }
        55
          \message{^^J*~This~is~sTeX~version~3.2.0~*^^J}
        58 %\RequirePackage{morewrites}
        Package options:
        61 \keys_define:nn { stex } {
            debug
                      .clist_set:N = \c_stex_debug_clist ,
                      .clist_set:N = \c_stex_languages_clist ,
            lang
                     .tl_set_x:N
                                   = \mathhub ,
            mathhub
                      .bool_set:N
                                   = \c_stex_persist_mode_bool ,
            usesms
            writesms .bool_set:N
                                   = \c_stex_persist_write_mode_bool ,
                                  = \c_tikzinput_image_bool,
            image
                      .bool_set:N
            unknown
                      .code:n
        69 }
        70 \ProcessKeysOptions { stex }
      The STEXlogo:
\sTeX
        71 \RequirePackage{stex-logo} % externalized for backwards-compatibility reasons
       (End definition for \stex and \sTeX. These functions are documented on page 71.)
```

24.3 Messages and logging

```
72 (00=stex_log)
                                Warnings and error messages
                             73 \msg_new:nnn{stex}{error/unknownlanguage}{
                                 Unknown~language:~#1
                             75 }
                             76 \msg_new:nnn{stex}{warning/nomathhub}{
                                 MATHHUB~system~variable~not~found~and~no~
                             77
                                  \detokenize{\mathhub}-value~set!
                             80 \msg_new:nnn{stex}{error/deactivated-macro}{
                                 The~\detokenize{#1}~command~is~only~allowed~in~#2!
                             81
                             82 }
          \stex_debug:nn A simple macro issuing package messages with subpath.
                             83 \cs_new_protected:Nn \stex_debug:nn {
                                  \clist_if_in:NnTF \c_stex_debug_clist { all } {
                                    \msg_set:nnn{stex}{debug / #1}{
                             85
                                      \\Debug~#1:~#2\\
                             86
                             88
                                    \msg_none:nn{stex}{debug / #1}
                             89
                                 }{
                                    \clist_if_in:NnT \c_stex_debug_clist { #1 } {
                             90
                                      \msg_set:nnn{stex}{debug / #1}{
                             91
                                        \\Debug~#1:~#2\\
                             92
                             93
                                      \msg_none:nn{stex}{debug / #1}
                             94
                             95
                                 }
                             96
                           (End definition for \stex_debug:nn. This function is documented on page 71.)
                                Redirecting messages:
                               \verb|\clist_if_in:NnTF \c_stex_debug_clist {all} | \{
                                    \msg_redirect_module:nnn{ stex }{ none }{ term }
                             99
                            100 }{
                                  \clist_map_inline:Nn \c_stex_debug_clist {
                            101
                                    \msg_redirect_name:nnn{ stex }{ debug / #1 }{ term }
                            102
                            104 }
                            106 \stex_debug:nn{log}{debug~mode~on}
                           24.4
                                     HTML Annotations
                            107 (@@=stex_annotate)
     \l_stex_html_arg_tl
                           Used by annotation macros to ensure that the HTML output to annotate is not empty.
\c_stex_html_emptyarg_tl
                            108 \tl_new:N \l_stex_html_arg_tl
                           (End definition for \l_stex_html_arg_tl and \c_stex_html_emptyarg_tl. These variables are docu-
                           mented on page ??.)
```

```
\_stex_html_checkempty:n
                             109 \cs_new_protected:Nn \_stex_html_checkempty:n {
                                  \tl_set:Nn \l_stex_html_arg_tl { #1 }
                                  \tl_if_empty:NT \l_stex_html_arg_tl {
                                    \tl_set_eq:NN \l_stex_html_arg_tl \c_stex_html_emptyarg_tl
                             113
                             114 }
                            (End definition for \_stex_html_checkempty:n. This function is documented on page ??.)
     \stex_if_do_html_p:
                           Whether to (locally) produce HTML output
     \stex_if_do_html: TF
                             115 \bool_new:N \_stex_html_do_output_bool
                             116 \bool_set_true:N \_stex_html_do_output_bool
                                \prg_new_conditional:Nnn \stex_if_do_html: {p,T,F,TF} {
                                  \bool_if:nTF \_stex_html_do_output_bool
                             120
                                    \prg_return_true: \prg_return_false:
                             121 }
                            (End definition for \stex_if_do_html:TF. This function is documented on page 71.)
                          Whether to (locally) produce HTML output
   \stex_suppress_html:n
                             122 \cs_new_protected:Nn \stex_suppress_html:n {
                                  \exp_args:Nne \use:nn {
                                    \bool_set_false:N \_stex_html_do_output_bool
                             124
                             125
                                    #1
                             126
                                    \stex_if_do_html:T {
                             127
                                      \bool_set_true:N \_stex_html_do_output_bool
                             128
                             129
                                    }
                                  }
                             130
                             131 }
                            (End definition for \stex_suppress_html:n. This function is documented on page 71.)
      \stex_annotate:enw
                           depend on the "backend" used (LATEXML, RusTFX, pdflatex).
```

\stex_annotate_invisible:n \stex_annotate_invisible:nnn We define four macros for introducing attributes in the HTML output. The definitions

The pdflatex-macros largely do nothing; the RusTrX-implementations are pretty clear in what they do, the LATEXML-implementations resort to perl bindings.

```
132 \ifcsname if@rustex\endcsname\else
     \expandafter\newif\csname if@rustex\endcsname
     \@rustexfalse
135 \fi
136 \ifcsname if@latexml\endcsname\else
     \expandafter\newif\csname if@latexml\endcsname
137
     \@latexmlfalse
138
139 \fi
140 \tl_if_exist:NF\stex@backend{
    \if@rustex
141
       \def\stex@backend{rustex}
142
143
       \if@latexml
144
         \def\stex@backend{latexml}
       \else
```

```
\cs_if_exist:NTF\HCode{
 147
              \def\stex@backend{tex4ht}
 148
 149
               \def\stex@backend{pdflatex}
 150
 151
         \fi
 152
 153
 154 }
     \input{stex-backend-\stex@backend.cfg}
    \verb|\newif\ifstexhtml|
    \stex_html_backend:TF\stexhtmltrue\stexhtmlfalse
 158
 159
(\mathit{End \ definition \ for \ \ } \texttt{stex\_annotate\_innn} \ , \ \texttt{stex\_annotate\_invisible:nnn}, \ and \ \texttt{stex\_annotate\_invisible:nnn})
These functions are documented on page 72.)
           Babel Languages
24.5
 160 (@@=stex_language)
We store language abbreviations in two (mutually inverse) property lists:
```

\c_stex_languages_prop

```
\c_stex_language_abbrevs_prop
                        161 \exp_args:NNx \prop_const_from_keyval:Nn \c_stex_languages_prop { \tl_to_str:n {
                             en = english ,
                        162
                             de = ngerman ,
                        163
                             ar = arabic ,
                             bg = bulgarian ,
                             ru = russian ,
                             fi = finnish ,
                        167
                             ro = romanian ,
                        168
                             tr = turkish ,
                        169
                             fr = french
                        170
                        171 }}
                        173 \exp_args:NNx \prop_const_from_keyval:Nn \c_stex_language_abbrevs_prop { \tl_to_str:n {
                        174
                             english
                             ngerman
                                        = de ,
                             arabic
                                        = ar ,
                             bulgarian = bg ,
                        177
                                       = ru ,
                        178
                             russian
                                        = fi ,
                             finnish
                        179
                             romanian = ro ,
                        180
                             turkish
                                        = tr ,
                        181
                             french
                                        = fr
                        182
                        183 }}
                        184 % todo: chinese simplified (zhs)
                                    chinese traditional (zht)
```

(End definition for \c_stex_languages_prop and \c_stex_language_abbrevs_prop. These variables are documented on page 72.)

we use the lang-package option to load the corresponding babel languages:

```
186 \cs_new_protected:Nn \stex_set_language:Nn {
    \str_set:Nx \l_tmpa_str {#2}
    \prop_get:NoNT \c_stex_languages_prop \l_tmpa_str #1 {
```

```
\ifx\@onlypreamble\@notprerr
189
         \ltx@ifpackageloaded{babel}{
190
           \exp_args:No \selectlanguage #1
191
         }{}
192
       \else
193
         \exp_args:No \str_if_eq:nnTF #1 {turkish} {
194
           \RequirePackage[#1,shorthands=:!]{babel}
195
         }{
196
           \RequirePackage[#1]{babel}
         }
198
       \fi
199
     }
200
201 }
202
   \clist_if_empty:NF \c_stex_languages_clist {
203
     \bool_set_false:N \l_tmpa_bool
204
     \clist_clear:N \l_tmpa_clist
205
     \clist_map_inline:Nn \c_stex_languages_clist {
206
       \str_set:Nx \l_tmpa_str {#1}
       \str_if_eq:nnT {#1}{tr}{
         \bool_set_true:N \l_tmpa_bool
       \prop_get:NoNTF \c_stex_languages_prop \l_tmpa_str \l_tmpa_str {
211
         \clist_put_right:No \l_tmpa_clist \l_tmpa_str
       } {
         \msg_error:nnx{stex}{error/unknownlanguage}{\l_tmpa_str}
       }
216
     \stex_debug:nn{lang} {Languages:~\clist_use:Nn \l_tmpa_clist {,~} }
217
     \bool_if:NTF \l_tmpa_bool {
       \RequirePackage[\clist_use:Nn \l_tmpa_clist,,shorthands=:!]{babel}
219
220
221
       \RequirePackage[\clist_use:Nn \l_tmpa_clist,]{babel}
     }
223 }
224
   \AtBeginDocument{
225
     \stex_html_backend:T {
226
227
       \seq_get_right:NN \g_stex_currentfile_seq \l_tmpa_str
       \seq_set_split:NnV \l_tmpa_seq . \l_tmpa_str
       \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str % .tex
       \seq_pop_left:NN \l_tmpa_seq \l_tmpa_str % <filename>
       \seq_if_empty:NF \l_tmpa_seq { %remaining element should be language
231
         \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
232
         \stex_debug:nn{basics} {Language~\l_tmpa_str~
           inferred~from~file~name}
234
         \stex_annotate_invisible:nnn{language}{ \l_tmpa_str }{}
235
236
     }
237
238 }
```

24.6 Persistence

```
240 (00=stex_persist)
241 \bool_if:NTF \c_stex_persist_mode_bool {
    \def \stex_persist:n #1 {}
    \def \stex_persist:x #1 {}
243
244 }{
     \bool_if:NTF \c_stex_persist_write_mode_bool {
245
    \iow_new:N \c__stex_persist_iow
246
    \iow_open:Nn \c__stex_persist_iow{\jobname.sms}
247
     \AtEndDocument{
248
      \iow_close:N \c__stex_persist_iow
249
250
     \cs_new_protected:Nn \stex_persist:n {
251
      \tl_set:Nn \l_tmpa_tl { #1 }
252
      \regex_replace_all:nnN { \ } { \~ } \l_tmpa_tl
      \exp_args:NNo \iow_now:Nn \c__stex_persist_iow \l_tmpa_tl
255
256
    \cs_generate_variant:Nn \stex_persist:n {x}
257
258
      \def \stex_persist:n #1 {}
259
      \def \stex_persist:x #1 {}
260
    }
261
262 }
```

24.7 Auxiliary Methods

```
\stex_deactivate_macro:Nn
```

```
263 \cs_new_protected:Nn \stex_deactivate_macro:Nn {
264 \exp_after:wN\let\csname \detokenize{#1} - orig\endcsname#1
265 \def#1{
266 \msg_error:nnnn{stex}{error/deactivated-macro}{\detokenize{#1}}{#2}
267 }
268 }

(End definition for \stex_deactivate_macro:Nn. This function is documented on page 72.)
```

\stex_reactivate_macro:N

```
269 \cs_new_protected:Nn \stex_reactivate_macro:N {
270 \exp_after:wN\let\exp_after:wN#1\csname \detokenize{#1} - orig\endcsname
271 }
```

(End definition for \stex_reactivate_macro:N. This function is documented on page 72.)

\ignorespacesandpars

```
272 \protected\def\ignorespacesandpars{
273    \begingroup\catcode13=10\relax
274    \@ifnextchar\par{
275     \endgroup\expandafter\ignorespacesandpars\@gobble
276    }{
277     \endgroup
278    }
279 }
```

```
\cs_new_protected:Nn \stex_copy_control_sequence:NNN {
281
    \tl_set:Nx \_tmp_args_tl {\cs_argument_spec:N #2}
282
    \exp_args:NNo \tl_remove_all:Nn \_tmp_args_tl \c_hash_str
283
    \int_set:Nn \l_tmpa_int {\tl_count:N \_tmp_args_tl}
284
285
    \tl_clear:N \_tmp_args_tl
286
    \int_step_inline:nn \l_tmpa_int {
287
       \tl_put_right:Nx \_tmp_args_tl {{\exp_not:n{###}\exp_not:n{##1}}}
289
290
    \tl_set:Nn #3 {\cs_generate_from_arg_count:NNnn #1 \cs_set:Npn}
291
     \tl_put_right:Nx #3 { {\int_use:N \l_tmpa_int}{
292
         \exp_after:wN\exp_after:wN\exp_after:wN \exp_not:n
293
        \exp_after:wN\exp_after:wN\exp_after:wN {
294
           \exp_after:wN #2 \_tmp_args_tl
295
296
    }}
297
298 }
  \cs_generate_variant:Nn \stex_copy_control_sequence:NNN {cNN}
  \cs_generate_variant:Nn \stex_copy_control_sequence:NNN {NcN}
  \cs_generate_variant:Nn \stex_copy_control_sequence:NNN {ccN}
301
302
  \cs_new_protected:Nn \stex_copy_control_sequence_ii:NNN {
303
    \tl_set:Nx \_tmp_args_tl {\cs_argument_spec:N #2}
304
     \exp_args:NNo \tl_remove_all:Nn \_tmp_args_tl \c_hash_str
305
    \int_set:Nn \l_tmpa_int {\tl_count:N \_tmp_args_tl}
306
307
    \tl_clear:N \_tmp_args_tl
308
    \int_step_inline:nn \l_tmpa_int {
      310
311
312
    \edef \_tmp_args_tl {
313
       \exp_after:wN\exp_after:wN\exp_after:wN \exp_not:n
314
       \exp_after:wN\exp_after:wN\exp_after:wN {
315
         \exp_after:wN #2 \_tmp_args_tl
316
317
318
    }
     \exp_after:wN \def \exp_after:wN \_tmp_args_tl
     \exp_after:wN ##\exp_after:wN 1 \exp_after:wN ##\exp_after:wN 2
321
    \exp_after:wN { \_tmp_args_tl }
322
323
     \edef \_tmp_args_tl {
324
       \exp_after:wN \exp_not:n \exp_after:wN {
325
         \_tmp_args_tl {####1}{####2}
326
327
    }
328
329
330
    \tl_set:Nn #3 {\cs_generate_from_arg_count:NNnn #1 \cs_set:Npn}
331
     \tl_put_right:Nx #3 { {\int_use:N \l_tmpa_int}{
332
      \exp_after:wN\exp_not:n\exp_after:wN{\_tmp_args_tl}
    }}
333
```

```
334 }
            335
            336 \cs_generate_variant:Nn \stex_copy_control_sequence_ii:NNN {cNN}
            337 \cs_generate_variant:Nn \stex_copy_control_sequence_ii:NNN {NcN}
               \cs_generate_variant:Nn \stex_copy_control_sequence_ii:NNN {ccN}
           (End definition for \ignorespacesandpars. This function is documented on page 72.)
\MMTrule
               \NewDocumentCommand \MMTrule {m m}{
                  \seq_set_split:Nnn \l_tmpa_seq , {#2}
            340
                  \int_zero:N \l_tmpa_int
            341
                  \stex_annotate_invisible:nnn{mmtrule}{scala://#1}{
                    \seq_if_empty:NF \l_tmpa_seq {
            343
                      $\seq_map_inline:Nn \l_tmpa_seq {
                        \int_incr:N \l_tmpa_int
            345
                        \label{lem:nnn} $$ \operatorname{stex\_annotate:nnn}_{arg}_i\in \mathbb{N} \leq \mathbb{N} + \mathbb{q}_{int}^{\#1} $$
            346
                      }$
            347
            348
                 }
            349
            350 }
            351
               \NewDocumentCommand \MMTinclude {m}{
                  \stex_annotate_invisible:nnn{import}{#1}{}
            353
            354 }
            355
               \tl_new:N \g_stex_document_title
            356
               \cs_new_protected:Npn \STEXtitle #1 {
                 \tl_if_empty:NT \g_stex_document_title {
            358
                    \tl_gset:Nn \g_stex_document_title { #1 }
            359
            360
            361 }
            362
               \cs_new_protected:Nn \stex_document_title:n {
            363
                 \tl_if_empty:NT \g_stex_document_title {
                    \tl_gset:Nn \g_stex_document_title { #1 }
                    \stex_annotate_invisible:n{\noindent
                      \stex_annotate:nnn{doctitle}{}{ #1 }
            367
                    \par}
                 }
            368
            369 }
               \AtBeginDocument {
            370
                 \let \STEXtitle \stex_document_title:n
            371
                 \tl_if_empty:NF \g_stex_document_title {
            372
                    \stex_annotate_invisible:n{\noindent
            373
                      \stex_annotate:nnn{doctitle}{}{ \g_stex_document_title }
            374
            375
                 }
            376
                 \let\_stex_maketitle:\maketitle
            377
                  \def\maketitle{
            378
                    \tl_if_empty:NF \@title {
            379
                      \exp_args:No \stex_document_title:n \@title
            380
            381
                    \_stex_maketitle:
            382
```

383

```
384 }
385
386 \cs_new_protected:Nn \stex_par: {
387  \mode_if_vertical:F{
388   \if@minipage\else\if@nobreak\else\par\fi\fi
389  }
390 }
391
392 \(\frac{package}\)
(End definition for \MMTrule. This function is documented on page ??.)
```

Chapter 25

STEX -MathHub Implementation

```
393 (*package)
394
mathhub.dtx
                                397 (@@=stex_path)
   Warnings and error messages
  \msg_new:nnn{stex}{error/norepository}{
    No~archive~#1~found~in~#2
400 }
401 \msg_new:nnn{stex}{error/notinarchive}{
    Not~currently~in~an~archive,~but~\detokenize{#1}~
402
    needs~one!
403
404 }
405 \msg_new:nnn{stex}{error/nofile}{
    \detokenize{#1}~could~not~find~file~#2
406
408 \msg_new:nnn{stex}{error/twofiles}{
    \detokenize{#1}~found~two~candidates~for~#2
410 }
```

25.1 Generic Path Handling

We treat paths as LATEX3-sequences (of the individual path segments, i.e. separated by a /-character) unix-style; i.e. a path is absolute if the sequence starts with an empty entry.

\stex_path_from_string:Nn

```
411 \cs_new_protected:Nn \stex_path_from_string:Nn {
412 \stex_debug:nn{files}{#2}
413 \str_set:Nx \l_tmpa_str { #2 }
414 \str_if_empty:NTF \l_tmpa_str {
415 \seq_clear:N #1
416 }{
417 \exp_args:NNNo \seq_set_split:Nnn #1 / { \l_tmpa_str }
418 \sys_if_platform_windows:T{
```

```
\seq_clear:N \l_tmpa_tl
                              419
                                        \seq_map_inline:Nn #1 {
                              420
                                          \seq_set_split:Nnn \l_tmpb_tl \c_backslash_str { ##1 }
                              421
                                          \seq_concat:NNN \l_tmpa_tl \l_tmpa_tl \l_tmpb_tl
                              422
                              423
                                        \seq_set_eq:NN #1 \l_tmpa_tl
                              424
                              425
                                      \stex_path_canonicalize:N #1
                              426
                                   }
                              427
                                    \stex_debug:nn{files}{Yields: \stex_path_to_string:N#1}
                              428
                              429 }
                              430
                             (End definition for \stex_path_from_string:Nn. This function is documented on page 73.)
  \stex_path_to_string:NN
   \stex_path_to_string:N
                              431 \cs_new_protected:Nn \stex_path_to_string:NN {
                                    \exp_args:NNe \str_set:Nn #2 { \seq_use:Nn #1 / }
                              432
                              433 }
                              434
                                 \cs_new:Nn \stex_path_to_string:N {
                              435
                                    \seq_use:Nn #1 /
                              436
                              437 }
                             (End definition for \stex_path_to_string:NN and \stex_path_to_string:N. These functions are doc-
                             umented on page 73.)
                             . and ..., respectively.
    \c__stex_path_dot_str
     \c__stex_path_up_str
                              438 \str_const:Nn \c__stex_path_dot_str {.}
                              439 \str_const:Nn \c__stex_path_up_str {..}
                             (End definition for \c_stex_path_dot_str and \c_stex_path_up_str.)
                             Canonicalizes the path provided; in particular, resolves . and . . path segments.
\stex_path_canonicalize:N
                                 \cs_new_protected:Nn \stex_path_canonicalize:N {
                                    \seq_if_empty:NF #1 {
                              441
                                      \seq_clear:N \l_tmpa_seq
                              442
                                      \seq_get_left:NN #1 \l_tmpa_tl
                              443
                                      \str_if_empty:NT \l_tmpa_tl {
                              444
                                        \seq_put_right:Nn \l_tmpa_seq {}
                              445
                                      }
                              446
                                      \seq_map_inline:Nn #1 {
                                        \str_set:Nn \l_tmpa_tl { ##1 }
                                        \str_if_eq:NNF \l_tmpa_tl \c__stex_path_dot_str {
                              449
                                          \str_if_eq:NNTF \l_tmpa_tl \c__stex_path_up_str {
                              450
                                            \seq_if_empty:NTF \l_tmpa_seq {
                              451
                              452
                                               \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq {
                              453
                                                 \c__stex_path_up_str
                              454
                              455
                                               \seq_get_right:NN \l_tmpa_seq \l_tmpa_tl
                              456
                                               \str_if_eq:NNTF \l_tmpa_tl \c__stex_path_up_str {
                              457
                                                 \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq {
                                                   \c__stex_path_up_str
```

```
}{
                                 461
                                                    \seq_pop_right:NN \l_tmpa_seq \l_tmpb_tl
                                 462
                                 463
                                               }
                                 464
                                             }{
                                                \str_if_empty:NF \l_tmpa_tl {
                                 466
                                                  \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq { \l_tmpa_tl }
                                                }
                                 469
                                             }
                                           }
                                 470
                                        }
                                 471
                                         \seq_gset_eq:NN #1 \l_tmpa_seq
                                 472
                                      }
                                 473
                                 474 }
                                (End definition for \stex_path_canonicalize:N. This function is documented on page 73.)
\stex_path_if_absolute_p:N
\stex_path_if_absolute:NTF
                                    \prg_new_conditional:Nnn \stex_path_if_absolute:N {p, T, F, TF} {
                                       \seq_if_empty:NTF #1 {
                                 476
                                         \prg_return_false:
                                 477
                                 478
                                 479
                                         \seq_get_left:NN #1 \l_tmpa_tl
                                 480
                                         \sys_if_platform_windows:TF{
                                           \str_if_in:NnTF \l_tmpa_tl {:}{
                                             \prg_return_true:
                                 482
                                           }{
                                 483
                                 181
                                             \prg_return_false:
                                           }
                                 485
                                        }{
                                 486
                                           \str_if_empty:NTF \l_tmpa_tl {
                                 487
                                             \prg_return_true:
                                 488
                                 489
                                              \prg_return_false:
                                           }
                                 492
                                        }
                                 493
                                      }
                                 494 }
                                (End definition for \stex_path_if_absolute:NTF. This function is documented on page 73.)
```

}

460

25.2 PWD and kpsewhich

\stex_kpsewhich:n

```
495 \str_new:N\l_stex_kpsewhich_return_str
496 \cs_new_protected:Nn \stex_kpsewhich:n {\begingroup
497 \catcode'\ =12
498 \sys_get_shell:nnN { kpsewhich ~ #1 } { } \l_tmpa_tl
499 \tl_gset_eq:NN \l_tmpa_tl \l_tmpa_tl
500 \endgroup
501 \exp_args:NNo\str_set:Nn\l_stex_kpsewhich_return_str{\l_tmpa_tl}
502 \tl_trim_spaces:N \l_stex_kpsewhich_return_str
503 }
```

```
We determine the PWD

\c_stex_pwd_seq
\c_stex_pwd_str

504 \sys_if_platform_windows:TF{
505    \begingroup\escapechar=-1\catcode'\\=12
506    \exp_args:Nx\stex_kpsewhich:n{-expand-var~\c_percent_str CD\c_percent_str}
507    \exp_args:NNx\str_replace_all:Nnn\l_stex_kpsewhich_return_str{\c_backslash_str}/
508    \exp_args:Nnx\use:nn{\endgroup}{\str_set:Nn\exp_not:N\l_stex_kpsewhich_return_str{\l_stex_509}}{
509    \frac{510}{510}    \stex_kpsewhich:n{-var-value-PWD}
511    }
512
513 \stex_path_from_string:Nn\c_stex_pwd_seq\l_stex_kpsewhich_return_str
```

(End definition for \stex_kpsewhich:n. This function is documented on page 73.)

(End definition for \c_stex_pwd_seq and \c_stex_pwd_str. These variables are documented on page 73.)

25.3 File Hooks and Tracking

514 \stex_path_to_string:NN\c_stex_pwd_seq\c_stex_pwd_str
515 \stex_debug:nn {mathhub} {PWD:~\str_use:N\c_stex_pwd_str}

```
516 (@@=stex_files)
```

527

We introduce hooks for file inputs that keep track of the absolute paths of files used. This will be useful to keep track of modules, their archives, namespaces etc.

Note that the absolute paths are only accurate in \input-statements for paths relative to the PWD, so they shouldn't be relied upon in any other setting than for STEX-purposes.

```
keeps track of file changes
   \g__stex_files_stack
                            517 \seq_gclear_new:N\g__stex_files_stack
                           (End\ definition\ for\ \g_stex_files_stack.)
   \c_stex_mainfile_seq
   \c_stex_mainfile_str
                            \str_set:Nx \c_stex_mainfile_str {\c_stex_pwd_str/\jobname.tex}
                            519 \stex_path_from_string:Nn \c_stex_mainfile_seq
                                 \c_stex_mainfile_str
                           (End definition for \c_stex_mainfile_seq and \c_stex_mainfile_str. These variables are documented
                           on page 73.)
\g_stex_currentfile_seq
                            521 \seq_gclear_new:N\g_stex_currentfile_seq
                           (End definition for \g_stex_currentfile_seq. This variable is documented on page 74.)
 \stex_filestack_push:n
                            522 \cs_new_protected:Nn \stex_filestack_push:n {
                                 \stex_path_from_string:Nn\g_stex_currentfile_seq{#1}
                            523
                                 \stex_path_if_absolute:NF\g_stex_currentfile_seq{
                            524
                                   \stex_path_from_string: Nn\g_stex_currentfile_seq{
                            525
                                     \c_stex_pwd_str/#1
                            526
```

```
528
                              \seq_gset_eq:NN\g_stex_currentfile_seq\g_stex_currentfile_seq
                         529
                              \exp_args:NNo\seq_gpush:Nn\g__stex_files_stack\g_stex_currentfile_seq
                         530
                              \stex_get_document_uri:
                         531
                         532 }
                        (End definition for \stex_filestack_push:n. This function is documented on page 74.)
\stex_filestack_pop:
                            \cs_new_protected:Nn \stex_filestack_pop: {
                               \seq_if_empty:NF\g__stex_files_stack{
                         534
                                 \seq_gpop:NN\g__stex_files_stack\l_tmpa_seq
                         535
                         536
                               \seq_if_empty:NTF\g__stex_files_stack{
                         537
                                 \verb|\seq_gset_eq:NN\g_stex_currentfile_seq\c_stex_mainfile_seq| \\
                         539
                                 \seq_get:NN\g__stex_files_stack\l_tmpa_seq
                         540
                                 \seq_gset_eq:NN\g_stex_currentfile_seq\l_tmpa_seq
                         541
                         542
                               \stex_get_document_uri:
                         543
                         544 }
                        (End definition for \stex_filestack_pop:. This function is documented on page 74.)
                             Hooks for the current file:
                         545 \AddToHook{file/before}{
                              \tl_if_empty:NTF\CurrentFilePath{
                                 \stex_filestack_push:n{\CurrentFile}
                         547
                         548
                                 \stex_filestack_push:n{\CurrentFilePath/\CurrentFile}
                         549
                         550
                         551 }
                         552 \AddToHook{file/after}{
                              \stex_filestack_pop:
                         554 }
```

25.4 MathHub Repositories

```
555 (@@=stex_mathhub)
```

567

\mathhub \c_stex_mathhub_seq \c_stex_mathhub_str The path to the mathhub directory. If the \mathhub-macro is not set, we query kpsewhich for the MATHHUB system variable.

```
\str_if_empty:NTF\mathhub{
556
     \sys_if_platform_windows:TF{
557
       \begingroup\escapechar=-1\catcode'\\=12
558
       \exp_args:Nx\stex_kpsewhich:n{-expand-var~\c_percent_str MATHHUB\c_percent_str}
559
       \exp_args:NNx\str_replace_all:Nnn\l_stex_kpsewhich_return_str{\c_backslash_str}/
       \exp_args:NNx\str_if_eq:onT\l_stex_kpsewhich_return_str{\c_percent_str MATHHUB\c_percent
       \exp_args:Nnx\use:nn{\endgroup}{\str_set:Nn\exp_not:N\l_stex_kpsewhich_return_str{\l_ste
562
563
     }{
       \stex_kpsewhich:n{-var-value~MATHHUB}
564
565
     \str_set_eq:NN\c_stex_mathhub_str\l_stex_kpsewhich_return_str
566
```

```
\str_if_empty:NT \c_stex_mathhub_str {
 568
        \sys_if_platform_windows:TF{
 569
          \verb|\begingroup\escapechar=-1\catcode'\=12|
 570
          \exp_args:Nx\stex_kpsewhich:n{-var-value~HOME}
 571
          \exp_args:NNx\str_replace_all:Nnn\l_stex_kpsewhich_return_str{\c_backslash_str}/
 572
          \exp_args:Nnx\use:nn{\endgroup}{\str_set:Nn\exp_not:N\l_stex_kpsewhich_return_str{\l_s
 573
        }{
 574
          \stex_kpsewhich:n{-var-value~HOME}
 575
        }
 576
        \ior_open:NnT \g_tmpa_ior{\l_stex_kpsewhich_return_str / .stex / mathhub.path}{
 577
 578
          \begingroup\escapechar=-1\catcode'\\=12
          \ior_str_get:NN \g_tmpa_ior \l_tmpa_str
 579
          \sys_if_platform_windows:T{
 580
             \exp_args:NNx\str_replace_all:Nnn\l_tmpa_str{\c_backslash_str}/
 581
 582
          \str_gset_eq:NN \c_stex_mathhub_str\l_tmpa_str
 583
 584
          \ior_close:N \g_tmpa_ior
 585
      \str_if_empty:NTF\c_stex_mathhub_str{
        \msg_warning:nn{stex}{warning/nomathhub}
 589
      }{
 590
        \stex_debug:nn{mathhub}{MathHub:~\str_use:N\c_stex_mathhub_str}
 591
        \exp_args:NNo \stex_path_from_string:Nn\c_stex_mathhub_seq\c_stex_mathhub_str
 592
      }
 593
 594 }{
      \stex_path_from_string:Nn \c_stex_mathhub_seq \mathhub
 595
      \stex_path_if_absolute:NF \c_stex_mathhub_seq {
 596
 597
        \exp_args:NNx \stex_path_from_string:Nn \c_stex_mathhub_seq {
 598
          \c_stex_pwd_str/\mathhub
        }
 599
      }
 600
      \stex_path_to_string:NN\c_stex_mathhub_seq\c_stex_mathhub_str
 601
      \stex_debug:nn{mathhub} {MathHub:~\str_use:N\c_stex_mathhub_str}
 602
 603 }
(End definition for \mathhub, \c_stex_mathhub_seq, and \c_stex_mathhub_str. These variables are
documented on page 74.)
the corresponding manifest file
```

\ stex mathhub do manifest:n

Checks whether the manifest for archive #1 already exists, and if not, finds and parses

```
\cs_new_protected: Nn \__stex_mathhub_do_manifest:n {
     \prop_if_exist:cF {c_stex_mathhub_#1_manifest_prop} {
605
       \str_set:Nx \l_tmpa_str { #1 }
606
       \prop_new:c { c_stex_mathhub_#1_manifest_prop }
607
       \seq_set_split:NnV \l_tmpa_seq / \l_tmpa_str
608
       \seq_concat:NNN \l_tmpa_seq \c_stex_mathhub_seq \l_tmpa_seq
609
       \__stex_mathhub_find_manifest:N \l_tmpa_seq
610
       \seq_if_empty:NTF \l__stex_mathhub_manifest_file_seq {
611
         \msg_error:nnxx{stex}{error/norepository}{#1}{
612
           \stex_path_to_string:N \c_stex_mathhub_str
613
         \input{Fatal~Error!}
```

```
} {
                                                                   616
                                                                                          \exp_args:No \__stex_mathhub_parse_manifest:n { \l_tmpa_str }
                                                                   617
                                                                                    }
                                                                   618
                                                                               }
                                                                   619
                                                                  620 }
                                                                (End\ definition\ for\ \verb|\__stex_mathhub_do_manifest:n.|)
\l stex mathhub manifest file seq
                                                                  621 \seq_new:N\l__stex_mathhub_manifest_file_seq
                                                                (End\ definition\ for\ \verb|\l_stex_mathhub_manifest_file_seq.|)
                                                               Attempts to find the MANIFEST.MF in some file path and stores its path in \l__stex_-
     \_stex_mathhub_find_manifest:N
                                                               mathhub_manifest_file_seq:
                                                                   \mbox{\em 622 } \mbox{\em cs_new\_protected:} \mbox{\em Nn } \mbox{\em L_stex_mathhub\_find_manifest:} \mbox{\em N} \mbox{\em {\em 622 } \mbox{\em 622 } \mbox
                                                                               \seq_set_eq:NN\l_tmpa_seq #1
                                                                   623
                                                                               \bool_set_true:N\l_tmpa_bool
                                                                   624
                                                                               \bool_while_do:Nn \l_tmpa_bool {
                                                                   625
                                                                                    \seq_if_empty:NTF \l_tmpa_seq {
                                                                   626
                                                                                         \bool_set_false:N\l_tmpa_bool
                                                                   627
                                                                   628
                                                                   629
                                                                                         \file_if_exist:nTF{
                                                                                              \stex_path_to_string:N\l_tmpa_seq/MANIFEST.MF
                                                                   631
                                                                                         }{
                                                                                              \seq_put_right:Nn\l_tmpa_seq{MANIFEST.MF}
                                                                   632
                                                                                              \bool_set_false:N\l_tmpa_bool
                                                                   633
                                                                                         }{
                                                                   634
                                                                                              \file_if_exist:nTF{
                                                                   635
                                                                                                    \stex_path_to_string:N\l_tmpa_seq/META-INF/MANIFEST.MF
                                                                   636
                                                                   637
                                                                                                    \seq_put_right:Nn\l_tmpa_seq{META-INF}
                                                                   638
                                                                                                    \seq_put_right:Nn\l_tmpa_seq{MANIFEST.MF}
                                                                                                    \bool_set_false:N\l_tmpa_bool
                                                                                              }{
                                                                                                    \file_if_exist:nTF{
                                                                                                        \verb|\stex_path_to_string:N\l_tmpa_seq/meta-inf/MANIFEST.MF| \\
                                                                   643
                                                                                                   }{
                                                                   644
                                                                                                         \seq_put_right:Nn\l_tmpa_seq{meta-inf}
                                                                   645
                                                                                                         \seq_put_right:Nn\l_tmpa_seq{MANIFEST.MF}
                                                                   646
                                                                                                         \bool_set_false:N\l_tmpa_bool
                                                                   647
                                                                   648
                                                                                                         \seq_pop_right:NN\l_tmpa_seq\l_tmpa_tl
                                                                   649
                                                                                                    }
                                                                                        }
                                                                                    }
                                                                   653
                                                                               655
                                                                  656 }
                                                                (End\ definition\ for\ \verb|\__stex_mathhub_find_manifest:N.)
         \c stex mathhub manifest ior File variable used for MANIFEST-files
```

657 \ior_new:N \c__stex_mathhub_manifest_ior

 $(End\ definition\ for\ \verb|\c_stex_mathhub_manifest_ior.|)$

```
\ stex mathhub parse manifest:n Stores the entries in manifest file in the corresponding property list:
```

```
658 \cs_new_protected:Nn \__stex_mathhub_parse_manifest:n {
      \seq_set_eq:NN \l_tmpa_seq \l__stex_mathhub_manifest_file_seq
 659
      \ior_open:Nn \c__stex_mathhub_manifest_ior {\stex_path_to_string:N \l_tmpa_seq}
 660
      \ior_map_inline:Nn \c__stex_mathhub_manifest_ior {
 661
        \str_set:Nn \l_tmpa_str {##1}
 662
        \exp_args:NNoo \seq_set_split:Nnn
 663
            \l_tmpb_seq \c_colon_str \l_tmpa_str
 664
        \seq_pop_left:NNTF \l_tmpb_seq \l_tmpa_tl {
 665
          \exp_args:NNe \str_set:Nn \l_tmpb_tl {
            \exp_args:NNo \seq_use:Nn \l_tmpb_seq \c_colon_str
          }
          \exp_args:No \str_case:nnTF \l_tmpa_tl {
            {id} {
 670
              \prop_gput:cno { c_stex_mathhub_#1_manifest_prop }
 671
                 { id } \l_tmpb_tl
 672
 673
            {narration-base} {
 674
              \prop_gput:cno { c_stex_mathhub_#1_manifest_prop }
 675
                 { narr } \l_tmpb_tl
 676
            {url-base} {
 679
              \prop_gput:cno { c_stex_mathhub_#1_manifest_prop }
 680
                 { docurl } \l_tmpb_tl
 681
            {source-base} {
 682
               \prop_gput:cno { c_stex_mathhub_#1_manifest_prop }
 683
                 { ns } \l_tmpb_tl
 684
 685
            {ns} {
 686
              \prop_gput:cno { c_stex_mathhub_#1_manifest_prop }
                 { ns } \l_tmpb_tl
            {dependencies} {
               \prop_gput:cno { c_stex_mathhub_#1_manifest_prop }
 691
                 { deps } \l_tmpb_tl
 692
 693
          }{}{}
 694
        }{}
 695
 696
      \ior_close:N \c__stex_mathhub_manifest_ior
 697
      \stex_persist:x {
        \prop_set_from_keyval:cn{ c_stex_mathhub_#1_manifest_prop }{
          \exp_after:wN \prop_to_keyval:N \csname c_stex_mathhub_#1_manifest_prop\endcsname
 700
        }
 701
      }
 702
 703 }
(End\ definition\ for\ \_\_stex\_mathhub\_parse\_manifest:n.)
```

704 \cs_new_protected:Nn \stex_set_current_repository:n {

```
\prop_set_eq:Nc \l_stex_current_repository_prop {
                              706
                                     c_stex_mathhub_#1_manifest_prop
                              707
                              708
                              709 }
                             (End definition for \stex_set_current_repository:n. This function is documented on page 74.)
\stex_require_repository:n
                              710 \cs_new_protected:Nn \stex_require_repository:n {
                                   \prop_if_exist:cF { c_stex_mathhub_#1_manifest_prop } {
                                     \stex_debug:nn{mathhub}{Opening~archive:~#1}
                                     \__stex_mathhub_do_manifest:n { #1 }
                              713
                              714
                                   7
                              715 }
                             (End definition for \stex_require_repository:n. This function is documented on page 74.)
     716 %\prop_new:N \l_stex_current_repository_prop
                                 \bool_if:NF \c_stex_persist_mode_bool {
                                   \__stex_mathhub_find_manifest:N \c_stex_pwd_seq
                              718
                                   \seq_if_empty:NTF \l__stex_mathhub_manifest_file_seq {
                              719
                                     \stex_debug:nn{mathhub}{Not~currently~in~a~MathHub~repository}
                              720
                                   } {
                              721
                                     \__stex_mathhub_parse_manifest:n { main }
                                     \prop_get:NnN \c_stex_mathhub_main_manifest_prop {id}
                              723
                                       \l_tmpa_str
                              724
                                     \prop_set_eq:cN { c_stex_mathhub_\l_tmpa_str _manifest_prop }
                              725
                                       \c_stex_mathhub_main_manifest_prop
                              726
                                     \exp_args:Nx \stex_set_current_repository:n { \l_tmpa_str }
                                     \stex_debug:nn{mathhub}{Current~repository:~
                              728
                                        \prop_item: Nn \l_stex_current_repository_prop {id}
                              729
                              730
                              731
                              732 }
                             (End definition for \l_stex_current_repository_prop. This variable is documented on page 74.)
                             Executes the code in the second argument in the context of the repository whose ID is
    \stex_in_repository:nn
                             provided as the first argument.
                                 \cs_new_protected:Nn \stex_in_repository:nn {
                              733
                                   \str_set:Nx \l_tmpa_str { #1 }
                                   \cs_set:Npn \l_tmpa_cs ##1 { #2 }
                              735
                                   \str_if_empty:NTF \l_tmpa_str {
                              736
                                     \prop_if_exist:NTF \l_stex_current_repository_prop {
```

738

740 741

742

743 744

745

}{

}{

\l_tmpa_cs{}

\exp_args:Ne \l_tmpa_cs{

\stex_require_repository:n { #1 }

\prop_item:Nn \l_stex_current_repository_prop { id }

\stex_debug:nn{mathhub}{do~in~current~repository:~\prop_item:Nn \l_stex_current_reposi

```
\stex_debug:nn{mathhub}{in~repository:~\l_tmpa_str}
746
       \stex_require_repository:n \l_tmpa_str
747
       \str_set:Nx \l_tmpa_str { #1 }
748
       \exp_args:Nne \use:nn {
749
         \stex_set_current_repository:n \l_tmpa_str
750
         \exp_args:Nx \l_tmpa_cs{\l_tmpa_str}
751
752
         \stex_debug:nn{mathhub}{switching~back~to:~
753
           \prop_if_exist:NTF \l_stex_current_repository_prop {
754
             \prop_item:Nn \l_stex_current_repository_prop { id }:~
755
             \meaning\l_stex_current_repository_prop
           }{
757
             no~repository
758
           }
759
760
         \prop_if_exist:NTF \l_stex_current_repository_prop {
761
          \stex_set_current_repository:n {
762
           \prop_item: Nn \l_stex_current_repository_prop { id }
763
          }
         }{
           \let\exp_not:N\l_stex_current_repository_prop\exp_not:N\undefined
767
768
    }
769
770 }
```

(End definition for \stex_in_repository:nn. This function is documented on page 74.)

25.5 Using Content in Archives

```
\mhpath
                                                                        \def \mhpath #1 #2 {
                                                           771
                                                                                   \exp_args:Ne \tl_if_empty:nTF{#1}{
                                                                                            \c_stex_mathhub_str /
                                                           773
                                                           774
                                                                                                      \prop_item:Nn \l_stex_current_repository_prop { id }
                                                           775
                                                                                                      / source / #2
                                                           776
                                                                                            \c_stex_mathhub_str / #1 / source / #2
                                                           778
                                                           779 }
                                                      (End definition for \mhpath. This function is documented on page 75.)
\inputref
     \mhinput
                                                           780 \newif \ifinputref \inputreffalse
                                                           781
                                                                        \verb|\cs_new_protected:Nn \ | \_stex_mathhub_mhinput:nn \{ | \cs_new_protected | \cs_new_
                                                           782
                                                                                  \stex_in_repository:nn {#1} {
                                                           783
                                                                                            \ifinputref
                                                           784
                                                                                                      \input{ \c_stex_mathhub_str / ##1 / source / #2 }
                                                           785
                                                           786
                                                                                                      \inputreftrue
                                                           787
                                                                                                      \input{ \c_stex_mathhub_str / ##1 / source / #2 }
```

```
\inputreffalse
 789
        \fi
 790
      }
 791
 792 }
    \NewDocumentCommand \mhinput { O{} m}{
 793
      \_stex_mathhub_mhinput:nn{ #1 }{ #2 }
 794
 795
 796
    \cs_new_protected:Nn \__stex_mathhub_inputref:nn {
 797
      \stex_in_repository:nn {#1} {
 798
        \stex_html_backend:TF {
 799
           \str_clear:N \l_tmpa_str
 800
           \prop_get:NnNF \l_stex_current_repository_prop { narr } \l_tmpa_str {
 801
             \prop_get:NnNF \l_stex_current_repository_prop { ns } \l_tmpa_str {}
 802
 803
 804
          \tl_if_empty:nTF{ ##1 }{
 805
             \IfFileExists{#2}{
 806
               \stex_annotate_invisible:nnn{inputref}{
                 \l_tmpa_str / #2
               }{}
             }{
 810
               \int \int d^2 t dt
 811
             }
 812
          }{
 813
             \IfFileExists{ \c_stex_mathhub_str / ##1 / source / #2 }{
 814
               \stex_annotate_invisible:nnn{inputref}{
 815
                 \l_tmpa_str / #2
 816
               }{}
 817
             }{
               \input{ \c_stex_mathhub_str / ##1 / source / #2 }
 819
 820
             }
          }
 821
 822
        }{
 823
           \begingroup
 824
             \inputreftrue
 825
             \tl_if_empty:nTF{ ##1 }{
 826
 827
               \int \inf\{\#2\}
             }{
               \input{ \c_stex_mathhub_str / ##1 / source / #2 }
             }
          \endgroup
 831
        }
 832
      }
 833
 834 }
    \NewDocumentCommand \inputref { O{} m}{
 835
      \__stex_mathhub_inputref:nn{ #1 }{ #2 }
 836
837 }
(End definition for \inputref and \mhinput. These functions are documented on page 75.)
```

\addmhbibresource

```
\mbox{\tt 838} \ \mbox{\tt cs_new\_protected:Nn \ \_stex_mathhub\_mhbibresource:nn} \ \{
```

```
\stex_in_repository:nn {#1} {
                  830
                         \addbibresource{ \c_stex_mathhub_str / ##1 / #2 }
                  840
                  841
                  842 }
                     \newcommand\addmhbibresource[2][]{
                  843
                       \__stex_mathhub_mhbibresource:nn{ #1 }{ #2 }
                 (End definition for \addmhbibresource. This function is documented on page 75.)
     \libinput
                     \cs_new_protected:Npn \libinput #1 {
                       \prop_if_exist:NF \l_stex_current_repository_prop {
                  847
                         \msg_error:nnn{stex}{error/notinarchive}\libinput
                  848
                  849
                       \prop_get:NnNF \l_stex_current_repository_prop {id} \l_tmpa_str {
                  850
                         \msg_error:nnn{stex}{error/notinarchive}\libinput
                  851
                  852
                       \seq_clear:N \l__stex_mathhub_libinput_files_seq
                  853
                       \seq_set_eq:NN \l_tmpa_seq \c_stex_mathhub_seq
                  854
                       \seq_set_split:NnV \l_tmpb_seq / \l_tmpa_str
                  855
                  856
                       \bool_while_do:nn { ! \seq_if_empty_p:N \l_tmpb_seq }{
                  857
                         \str_set:Nx \l_tmpa_str {\stex_path_to_string:N \l_tmpa_seq / meta-inf / lib / #1.tex}
                  858
                         \IfFileExists{ \l_tmpa_str }{
                  859
                           \seq_put_right:No \l__stex_mathhub_libinput_files_seq \l_tmpa_str
                  860
                  861
                         \seq_pop_left:NN \l_tmpb_seq \l_tmpa_str
                  862
                  863
                         \seq_put_right:No \l_tmpa_seq \l_tmpa_str
                       \str_set:Nx \l_tmpa_str {\stex_path_to_string:N \l_tmpa_seq / lib / #1.tex}
                       \IfFileExists{ \l_tmpa_str }{
                  867
                         \seq_put_right:No \l__stex_mathhub_libinput_files_seq \l_tmpa_str
                  868
                  869
                  870
                       \seq_if_empty:NTF \l__stex_mathhub_libinput_files_seq {
                  871
                         \msg_error:nnxx{stex}{error/nofile}{\exp_not:N\libinput}{#1.tex}
                  872
                  873
                  874
                         \seq_map_inline: Nn \l__stex_mathhub_libinput_files_seq {
                  875
                           \input{ ##1 }
                  876
                         }
                       }
                  877
                  878 }
                 (End definition for \libinput. This function is documented on page 75.)
\libusepackage
                     \NewDocumentCommand \libusepackage {0{} m} {
                       \prop_if_exist:NF \l_stex_current_repository_prop {
                         \msg_error:nnn{stex}{error/notinarchive}\libusepackage
                  881
                  882
                       \prop_get:NnNF \l_stex_current_repository_prop {id} \l_tmpa_str {
                  883
                         \msg_error:nnn{stex}{error/notinarchive}\libusepackage
                  884
                  885
```

```
\seq_set_eq:NN \l_tmpa_seq \c_stex_mathhub_seq
                      887
                           \seq_set_split:NnV \l_tmpb_seq / \l_tmpa_str
                      888
                      889
                            \bool_while_do:nn { ! \seq_if_empty_p:N \l_tmpb_seq }{
                      890
                              \str_set:Nx \l_tmpa_str {\stex_path_to_string:N \l_tmpa_seq / meta-inf / lib / #2}
                      891
                              \IfFileExists{ \l_tmpa_str.sty }{
                      892
                                \seq_put_right:No \l__stex_mathhub_libinput_files_seq \l_tmpa_str
                      893
                      894
                              \seq_pop_left:NN \l_tmpb_seq \l_tmpa_str
                      895
                              \seq_put_right:No \l_tmpa_seq \l_tmpa_str
                           }
                      897
                      898
                            \str_set:Nx \l_tmpa_str {\stex_path_to_string:N \l_tmpa_seq / lib / #2}
                      899
                            \IfFileExists{ \l_tmpa_str.sty }{
                      900
                              \seq_put_right:No \l__stex_mathhub_libinput_files_seq \l_tmpa_str
                      901
                      902
                      903
                           \seq_if_empty:NTF \l__stex_mathhub_libinput_files_seq {
                              \msg_error:nnxx{stex}{error/nofile}{\exp_not:N\libusepackage}{#2.sty}
                       905
                       906
                              \int_compare:nNnTF {\seq_count:N \l__stex_mathhub_libinput_files_seq} = 1 {
                      907
                                \seq_map_inline: Nn \l__stex_mathhub_libinput_files_seq {
                      908
                                  \usepackage[#1]{ ##1 }
                      909
                      910
                      911
                                \msg_error:nnxx{stex}{error/twofiles}{\exp_not:N\libusepackage}{#2.sty}
                      912
                             }
                      913
                           }
                      914
                      915 }
                     (End definition for \libusepackage. This function is documented on page 75.)
        \mhgraphics
       \cmhgraphics
                      916
                         \AddToHook{begindocument}{
                      917
                         \ltx@ifpackageloaded{graphicx}{
                      918
                              \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
                              \providecommand\mhgraphics[2][]{%
                      920
                                \def\Gin@mhrepos{}\setkeys{Gin}{#1}%
                      921
                                \includegraphics[#1]{\mhpath\Gin@mhrepos{#2}}}
                      922
                              923
                           }{}
                      924
                     (End definition for \mhgraphics and \cmhgraphics. These functions are documented on page 75.)
\lstinputmhlisting
\clstinputmhlisting
                      925 \ltx@ifpackageloaded{listings}{
                              \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
                              \newcommand\lstinputmhlisting[2][]{%
                      927
                                \def\lst@mhrepos{}\setkeys{lst}{#1}%
                      928
                                \lstinputlisting[#1]{\mhpath\lst@mhrepos{#2}}}
                      929
                              \newcommand\clstinputmhlisting[2][]{\begin{center}\lstinputmhlisting[#1]{#2}\end{center}
                      930
                           }{}
                      931
                      932 }
```

\seq_clear:N \l__stex_mathhub_libinput_files_seq

886

```
933
934 </package>
```

(End definition for \lstinputmhlisting and \clstinputmhlisting. These functions are documented on page 75.)

Chapter 26

$ST_{E}X$

-References Implementation

```
935 (*package)
               939 (@@=stex_refs)
                   Warnings and error messages
                940 \msg_new:nnn{stex}{error/extrefmissing}{
                    Missing~in~or~cite~value~for~\detokenize{\extref}!
               942 }
               943 \msg_new:nnn{stex}{warning/smsmissing}{
                    .sref~file~#1~doesn't~exist!
               944
               945 }
                946 \msg_new:nnn{stex}{warning/smslabelmissing}{
                    No~label~#2~in~.sref~file~#1!
                   References are stored in the file \jobname.sref, to enable cross-referencing external
               949 \iow_new:N \c__stex_refs_refs_iow
               950 \AtBeginDocument{
                    \iow_open:Nn \c__stex_refs_refs_iow {\jobname.sref}
                953 \AtEndDocument{
               954 \iow_close:N \c__stex_refs_refs_iow
\STEXreftitle
               956 \str_set:Nn \g__stex_refs_title_tl {Unnamed~Document}
               958 \NewDocumentCommand \STEXreftitle { m } {
                    \tl_gset:Nx \g__stex_refs_title_tl { #1 }
                    \iow_now:Nx \c__stex_refs_refs_iow {
                960
                      \STEXInternalSrefDocumentName { #1 }
               961
               962
               963 }
```

26.1 Document URIs and URLs

```
\l_stex_current_docns_str
                              964 \str_new:N \l_stex_current_docns_str
                             (End definition for \l_stex_current_docns_str. This variable is documented on page 76.)
  \stex_get_document_uri:
                                 \cs_new_protected:Nn \stex_get_document_uri: {
                                   \seq_set_eq:NN \l_tmpa_seq \g_stex_currentfile_seq
                                   \seq_pop_right:NN \l_tmpa_seq \l_tmpb_str
                                   \exp_args:NNno \seq_set_split:Nnn \l_tmpb_seq . \l_tmpb_str
                                   \seq_get_left:NN \l_tmpb_seq \l_tmpb_str
                              970
                                   \seq_put_right:No \l_tmpa_seq \l_tmpb_str
                              971
                                   \str_clear:N \l_tmpa_str
                              972
                                   \prop_if_exist:NT \l_stex_current_repository_prop {
                              973
                                     \prop_get:NnNF \l_stex_current_repository_prop { narr } \l_tmpa_str {
                              974
                                        \prop_get:NnNF \l_stex_current_repository_prop { ns } \l_tmpa_str {}
                              975
                              976
                                   }
                              977
                                   \str_if_empty:NTF \l_tmpa_str {
                                     \str_set:Nx \l_stex_current_docns_str {
                              980
                                       file:/\stex_path_to_string:N \l_tmpa_seq
                              981
                              982
                                   }{
                              983
                                     \bool_set_true:N \l_tmpa_bool
                              984
                                     \bool_while_do:Nn \l_tmpa_bool {
                              985
                                       \seq_pop_left:NN \l_tmpa_seq \l_tmpb_str
                              986
                                       \exp_args:No \str_case:nnTF { \l_tmpb_str } {
                              987
                                          {source} { \bool_set_false:N \l_tmpa_bool }
                                       }{}{
                                          \seq_if_empty:NT \l_tmpa_seq {
                                            \bool_set_false:N \l_tmpa_bool
                              991
                              992
                                       }
                              993
                                     }
                              994
                              995
                                     \seq_if_empty:NTF \l_tmpa_seq {
                              996
                                       \str_gset_eq:NN \l_stex_current_docns_str \l_tmpa_str
                              997
                                       \str_gset:Nx \l_stex_current_docns_str {
                                          \l_tmpa_str/\stex_path_to_string:N \l_tmpa_seq
                             1001
                                     }
                             1002
                             1003
                                   %\stex_get_document_url:
                             1004
                             1005 }
                             (End definition for \stex_get_document_uri: This function is documented on page 76.)
```

```
\l_stex_current_docurl_str
                               1006 \str_new:N \l_stex_current_docurl_str
                              (End definition for \l_stex_current_docurl_str. This variable is documented on page 76.)
   \stex_get_document_url:
                                   \cs_new_protected:Nn \stex_get_document_url: {
                                     \seq_set_eq:NN \l_tmpa_seq \g_stex_currentfile_seq
                               1008
                                     \seq_pop_right:NN \l_tmpa_seq \l_tmpb_str
                               1009
                                     \exp_args:NNno \seq_set_split:Nnn \l_tmpb_seq . \l_tmpb_str
                               1010
                                     \seq_get_left:NN \l_tmpb_seq \l_tmpb_str
                               1011
                                     \seq_put_right:No \l_tmpa_seq \l_tmpb_str
                               1012
                               1013
                                     \str_clear:N \l_tmpa_str
                               1015
                                     \prop_if_exist:NT \l_stex_current_repository_prop {
                                       \prop_get:NnNF \l_stex_current_repository_prop { docurl } \l_tmpa_str {
                               1016
                               1017
                                         \prop_get:NnNF \l_stex_current_repository_prop { narr } \l_tmpa_str {
                                           \prop_get:NnNF \l_stex_current_repository_prop { ns } \l_tmpa_str {}
                               1018
                               1019
                                       }
                               1020
                               1021
                               1022
                                     \str_if_empty:NTF \l_tmpa_str {
                               1023
                                       \str_set:Nx \l_stex_current_docurl_str {
                               1024
                                         file:/\stex_path_to_string:N \l_tmpa_seq
                                       }
                                     }{
                               1027
                                       \bool_set_true:N \l_tmpa_bool
                               1028
                                       \bool_while_do:Nn \l_tmpa_bool {
                               1029
                                         \seq_pop_left:NN \l_tmpa_seq \l_tmpb_str
                               1030
                                         \exp_args:No \str_case:nnTF { \l_tmpb_str } {
                               1031
                                           {source} { \bool_set_false:N \l_tmpa_bool }
                               1032
                               1033
                                           \seq_if_empty:NT \l_tmpa_seq {
                               1034
                                             \bool_set_false:N \l_tmpa_bool
                                         }
                               1037
                                       }
                               1038
                               1039
                                       \seq_if_empty:NTF \l_tmpa_seq {
                               1040
                                         \str_set_eq:NN \l_stex_current_docurl_str \l_tmpa_str
                               1041
                               1042
                                         \str_set:Nx \l_stex_current_docurl_str {
                               1043
                                           \l_tmpa_str/\stex_path_to_string:N \l_tmpa_seq
                               1044
                               1045
                                       }
                                     }
                               1047
                               1048 }
```

(End definition for \stex_get_document_url:. This function is documented on page 76.)

26.2 Setting Reference Targets

```
1049 \str_const:Nn \c__stex_refs_url_str{URL}
                               1050 \str_const:Nn \c__stex_refs_ref_str{REF}
                               1051 \str_new:N \l__stex_refs_curr_label_str
                               1052 % @currentlabel -> number
                               1053 % @currentlabelname -> title
                               1054 % @currentHref -> name.number <- id of some kind
                               1055 % @currentcounter <- name/id
                               1056 % \#autorefname <- "Section"
                               1057 % \theH# -> \arabic{section}
                               _{1058} % \the# -> number
                               1059 % \hyper@makecurrent{#}
                               1060 \int_new:N \l__stex_refs_unnamed_counter_int
                                   Restoring references from .sref-files
      \STEXInternalSrefRestoreTarget
                               1061 \cs_new_protected:Npn \STEXInternalSrefDocumentName #1 {}
                               1062 \cs_new_protected:Npn \STEXInternalSrefRestoreTarget #1#2#3#4#5 {}
                              (End definition for \STEXInternalSrefRestoreTarget. This function is documented on page ??.)
\stex_ref_new_doc_target:n
                                  \seq_new:N \g_stex_ref_files_seq
                               1063
                               1064
                                  \cs_new_protected:Nn \stex_ref_new_doc_target:n {
                               1065
                                    %\stex_get_document_uri:
                               1066
                                     \str_clear:N \l__stex_refs_curr_label_str
                               1067
                                     \str_set:Nx \l_tmpa_str { #1 }
                               1068
                                     \str_if_empty:NT \l_tmpa_str {
                               1069
                                       \int_gincr:N \l__stex_refs_unnamed_counter_int
                               1070
                               1071
                                       \str_set:Nx \l_tmpa_str {REF\int_use:N \l__stex_refs_unnamed_counter_int}
                               1072
                                    \str_set:Nx \l__stex_refs_curr_label_str {
                               1074
                                       \l_stex_current_docns_str?\l_tmpa_str
                               1075
                               1076
                                    \exp_args:Noo \STEXInternalAuxAddDocRef\l_stex_current_docns_str\l_tmpa_str
                               1077
                               1078
                                    %\seq_if_exist:cF{g__stex_refs_labels_\l_tmpa_str _seq}{
                               1079
                                        \seq_new:c {g__stex_refs_labels_\l_tmpa_str _seq}
                               1080
                               1081
                                    %\seq_if_in:coF{g__stex_refs_labels_\l_tmpa_str _seq}\l__stex_refs_curr_label_str {
                               1082
                                        \seq_gput_right:co{g__stex_refs_labels_\l_tmpa_str _seq}\l__stex_refs_curr_label_str
                               1083
                               1084
                               1085
                               1086
                                     \stex_if_smsmode:TF {
                               1087
                                       %\stex_get_document_url:
                               1088
                                       %\str_gset_eq:cN {sref_url_\l__stex_refs_curr_label_str _str}\l_stex_current_docurl_str
                               1089
                                       %\str_gset_eq:cN {sref_\l__stex_refs_curr_label_str _type}\c__stex_refs_url_str
                               1090
                               1091
                                       \iow_now:Nx \c__stex_refs_refs_iow {
                               1092
                                         \STEXInternalSrefRestoreTarget
                               1093
                                           {\l_stex_current_docns_str}
                                           {\l_tmpa_str}
```

```
{\@currentcounter}
                                            {\@currentlabel}
                               1097
                                            {\exp_args:No\unexpanded\@currentlabelname}
                               1098
                               1099
                                       %\iow_now:Nx \c__stex_refs_refs_iow {
                               1100
                                       % {\l_stex_current_docns_str?\l_tmpa_str}~=~{{\use:c{\@currentcounter autorefname}~\@cu
                                       \exp_args:Nx\label{sref_\l_stex_refs_curr_label_str}
                                       \immediate\write\@auxout{\STEXInternalAuxAddDocRef{\1_stex_current_docns_str}{\1_tmpa_st
                                       %\str_gset:cx {sref_\l__stex_refs_curr_label_str _type}\c__stex_refs_ref_str
                               1105
                               1106 }
                                   \NewDocumentCommand \slabel {m} {\stex_ref_new_doc_target:n {#1}}
                              (End definition for \stex_ref_new_doc_target:n. This function is documented on page 76.)
                                   The following is used to set the necessary macros in the .aux-file.
                                   \cs_new_protected:Npn \STEXInternalAuxAddDocRef #1 #2 {
                               1108
                                     \exp_args:NNx \seq_if_in:NnTF \g_stex_ref_files_seq {\detokenize{#1}} {
                               1109
                                       \exp_args:\nx \seq_if_in:cnF{g_stex_ref_ #1 _seq}{\detokenize{#2}}{
                                         \exp_args:Nnx \seq_gput_left:cn{g_stex_ref_ #1 _seq}{\detokenize{#2}}
                               1112
                               1113
                                     }{
                                         \exp_args:NNx \seq_gput_right:Nn \g_stex_ref_files_seq {\detokenize{#1}}
                               1114
                                         %\seq_if_exist:cF{g_stex_ref_ #1 _seq}{
                               1115
                                            \seq_new:c{g_stex_ref_ #1 _seq} % <- seq_new throws errors??
                               1116
                               1117
                                         \exp_args:Nnx \seq_gput_left:cn{g_stex_ref_ #1 _seq}{\detokenize{#2}}
                               1118
                               1119
                               1120
                                     %\str_set:Nn \l_tmpa_str {#1?#2}
                               1121
                                     %\str_gset_eq:cN{sref_#1?#2_type}\c__stex_refs_ref_str
                                     %\seq_if_exist:cF{g__stex_refs_labels_#2_seq}{
                                     % \seq_new:c {g__stex_refs_labels_#2_seq}
                               1124
                               1125
                                     \label{lem:cofg_stex_refs_labels_#2_seq}\labels_$$\sharp 2_seq}\labels_$$
                               1126
                                        \label{lem:cog_stex_refs_labels_#2_seq} $$\co{g_stex_refs_labels_#2_seq} \le $$\co{g_stex_refs_labels_#2_seq} $$
                                     %
                                     %}
                               1128
                               1129 }
                              To avoid resetting the same macros when the .aux-file is read at the end of the document:
                               1130 \AtEndDocument{
                                     \def\STEXInternalAuxAddDocRef#1 #2 {}{}
                               1132 }
\stex_ref_new_sym_target:n
                                  \cs_new_protected:Nn \stex_ref_new_sym_target:n {
                               1133
                               1135 %
                                      \stex_if_smsmode:TF {
                                        \str_if_exist:cF{sref_sym_#1_type}{
                               1136 %
                               1137 %
                                          \stex_get_document_url:
                                          \str_gset_eq:cN {sref_sym_url_#1_str}\l_stex_current_docurl_str
                               1138 %
                               1139 %
                                           \str_gset_eq:cN {sref_sym_#1_type}\c__stex_refs_url_str
                               1140 %
                               1141 %
```

\str_if_empty:NF \l__stex_refs_curr_label_str {

1142 %

(End definition for \stex_ref_new_sym_target:n. This function is documented on page 76.)

26.3 Using References

\sref Optional arguments:

```
\keys_define:nn { stex / sref / 1 } {
      archive
                   .str_set_x:N = \l__stex_refs_repo_str,
1154
                .str_set_x:N = \l__stex_refs_file_str
1155
1156 }
    \cs_new_protected:Nn \__stex_refs_args_i:n {
      \str_clear:N \l__stex_refs_repo_str
1158
      \str_clear:N \l__stex_refs_file_str
1159
      \keys_set:nn { stex / sref / 1 } { #1 }
1160
1161 }
_{\mbox{\scriptsize 1162}} \ \mbox{\scriptsize keys\_define:nn} \ \{ \ \mbox{\scriptsize stex} \ / \ \mbox{\scriptsize sref} \ / \ 2 \ \} \ \{
             .str_set_x:N = \l__stex_refs_in_str,
1163
                   .str_set_x:N = \l__stex_refs_repob_str,
      archive
1164
                .str_set_x:N = \l__stex_refs_cite_str
1165
1166 }
1167
    \cs_new_protected:Nn \__stex_refs_args_ii:n {
      \str_clear:N \l__stex_refs_in_str
1168
      \str_clear:N \l__stex_refs_cite_str
      \str_clear:N \l__stex_refs_repob_str
      \keys_set:nn { stex / sref / 2 } { #1 }
1172 }
The actual macro:
1173 \NewDocumentCommand \sref { O() m O()}{
      \__stex_refs_args_i:n{#1}
1174
      \__stex_refs_args_ii:n{#3}
1175
      \str_clear:N \l__stex_refs_uri_str
1176
      \__stex_refs_find_uri:n{#2}
      \__stex_refs_do_sref:n{#2}
1178
1179 }
    \NewDocumentCommand \extref { O{} m m}{
1180
      \__stex_refs_args_i:n{#1}
       \_stex_refs_args_ii:n{#3}
1182
      \str_if_empty:NT \l__stex_refs_in_str {
         \msg_error:nn{stex}{error/extrefmissing}
1184
1185
      \str_clear:N \l__stex_refs_uri_str
1186
      \__stex_refs_find_uri:n{#2}
1187
      \__stex_refs_do_sref_in:n{#2}
1188
```

```
1189
1190
        \cs_new_protected:Nn \__stex_refs_find_uri:n {
1191
            \stex_debug:nn{sref}{File: \l__stex_refs_file_str^^JRepo:\l__stex_refs_repo_str}
1192
            \str_if_empty:NTF \l__stex_refs_file_str {
                \seq_if_exist:cT{g_stex_ref_\l_stex_current_docns_str _seq}{
1194
                     \seq_map_inline:cn{g_stex_ref_\l_stex_current_docns_str _seq}{
1195
                         \str_if_eq:nnT{#1}{##1}{
1196
                             \str_set_eq:NN \l__stex_refs_uri_str \l_stex_current_docns_str
                             \seq_map_break:
1199
                        }
                    }
1200
1201
                \str_if_empty:NF \l__stex_refs_uri_str {
1202
                    \seq_map_inline: Nn \g_stex_ref_files_seq {
1203
                         \seq_map_inline:cn{g_stex_ref_##1_seq}{
1204
                             \str_if_eq:nnT{#1}{####1}{
1205
                                  \str_set:Nn \l__stex_refs_uri_str {##1}
1206
                                  \seq_map_break:n{\seq_map_break:}
                            }
                        }
                    }
               }
                \str_if_empty:NTF \l__stex_refs_repo_str {
1213
                    \prop_if_exist:NTF \l_stex_current_repository_prop {
1214
                         \prop_get:NnN \l_stex_current_repository_prop { ns } \l__stex_refs_uri_str
1215
1216
                         \str_set:Nx \l__stex_refs_uri_str {\l__stex_refs_uri_str / \l__stex_refs_file_str}
                         \stex_path_from_string:\n\l_tmpb_seq \l__stex_refs_uri_str
1217
                         \str_set:Nx \l__stex_refs_uri_str {\stex_path_to_string:N \l_tmpb_seq}
                    }{
1219
                         \stex_path_from_string:Nn \l_tmpb_seq {
1221
                             \stex_path_to_string:N \g_stex_currentfile_seq/ .. / \l__stex_refs_file_str
                         \str_set:Nx \l__stex_refs_uri_str {file:/\stex_path_to_string:N \l_tmpb_seq}
1223
                    }
1224
1225
                     \stex_require_repository:n \l__stex_refs_repo_str
1226
1227
                     \prop_get:cnN { c_stex_mathhub_\l__stex_refs_repo_str _manifest_prop } { ns } \l__stex
                     \str_set:Nx \l__stex_refs_uri_str {\l__stex_refs_uri_str / \l__stex_refs_file_str}
                    \stex_path_from_string:\n\\l_tmpb_seq \l__stex_refs_uri_str
                     \str_set:Nx \l__stex_refs_uri_str {\stex_path_to_string:N \l_tmpb_seq}
               }
           }
       }
1234
        \cs_new_protected:Nn \__stex_refs_do_autoref:n{
1235
            \cs_if_exist:cTF{autoref}{
1236
1237
                  \exp_args:Nx\autoref{sref_#1}
1238
             }{
                   \ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath{\ensuremath}\amb}\amb}\amb}}}}}}}}}}}}}}
1240
             }
1241 }
```

1242

```
\cs_new_protected:Nn \__stex_refs_do_sref:n {
      \str_if_empty:NTF \l__stex_refs_uri_str {
1244
        \str_if_empty:NTF \l__stex_refs_in_str {
1245
            __stex_refs_do_autoref:n{#1}
1246
1247
             _stex_refs_do_sref_in:n{#1}
1248
        }
1249
      }{
1250
        \exp_args:NNo \seq_if_in:NnTF \g_stex_ref_files_seq \l__stex_refs_uri_str {
          \exp_args:Nnx \seq_if_in:cnTF{g_stex_ref_\l__stex_refs_uri_str _seq}{\detokenize{#1}}{
1252
1253
            \__stex_refs_do_autoref:n{\l__stex_refs_uri_str?#1}
          }{
1254
            \str_if_empty:NTF \l__stex_refs_in_str {
1255
               \__stex_refs_do_autoref:n{#1}
1256
1257
               \__stex_refs_do_sref_in:n{#1}
1258
1259
          }
1260
          \str_if_empty:NTF \l__stex_refs_in_str {
            \__stex_refs_do_autoref:n{#1}
          }{
1265
             \_\_stex_refs_do_sref_in:n{#1}
1266
        }
1267
      }
1268
1269 }
1270
    \cs_new_protected:Nn \__stex_refs_restore_target:nnnnn {
1271
      \str_if_empty:NTF \l__stex_refs_uri_str {
1273
        \exp_args:No \str_if_eq:nnT \l__stex_refs_id_str {#2}{
1274
          \tl_set:Nn \l__stex_refs_return_tl {
            \label{lem:lempty:nF} $$ \sup : c{\#3autorefname}^{\#4}\tl_if_empty:nF{\#5}{^{(\#5)}}^{n} $$
1275
            \str_if_empty:NTF \l__stex_refs_cite_str{\tl_if_empty:nTF\l__stex_refs_docname_tl{
1276
1277
            }\l__stex_refs_docname_tl}{\cite{\l__stex_refs_cite_str}}
1278
1279
        }
1280
1281
      }{
        \stex_debug:nn{sref}{\l__stex_refs_uri_str{}~ == ~ #1 ~ ?}
        \exp_args:No \str_if_eq:nnT \l__stex_refs_uri_str {#1}{
          \stex_debug:nn{sref}{\l__stex_refs_id_str~ == ~ #2 ~ ?}
          \exp_args:No \str_if_eq:nnT \l__stex_refs_id_str {#2}{
1285
            \stex_debug:nn{sref}{success!}
1286
            \tl_set:Nn \l_stex_refs_return_tl {
1287
               \c : c{\#3autorefname}^{\#4}\tl_if_empty:nF{\#5}{^{(\#5)}}^{in}
1288
               \str_if_empty:NTF \l__stex_refs_cite_str{\tl_if_empty:nTF\l__stex_refs_docname_tl{
1289
1290
               }\l__stex_refs_docname_tl}{\cite{\l__stex_refs_cite_str}}
1291
            }
1292
            \endinput
1294
          }
1295
        }
```

}

1296

```
1297
   \cs_new_protected: Nn \__stex_refs_document_name:n {
     \tl_set:Nn \l__stex_refs_docname_tl {#1}
1299
1300 }
   % TODO cite
1301
   \cs_new_protected:Nn \__stex_refs_do_sref_in:n {
1302
     \stex_debug:nn{sref}{In: \l__stex_refs_in_str^^JRepo:\l__stex_refs_repo_str}
1303
     \stex_debug:nn{sref}{URI: \l__stex_refs_uri_str?#1}
1304
     %\msg_warning:nnn{stex}{warning/smsmissing}{<filename>}
     \begingroup\catcode13=9\relax\catcode10=9\relax
       \str_if_empty:NTF \l__stex_refs_repob_str {
1307
          \prop_if_exist:NTF \l_stex_current_repository_prop {
1308
            \str_set:Nx \l_tmpa_str {
1309
              \c_stex_mathhub_str /
              \prop_item:Nn \l_stex_current_repository_prop { id }
              / source / \l_stex_refs_in_str .sref
         }{
1314
            \str_set:Nx \l_tmpa_str {
              \stex_path_to_string:N \g_stex_currentfile_seq/ .. / \l__stex_refs_in_str . sref
1317
         }
1318
       }{
1319
          \str_set:Nx \l_tmpa_str {
            \c_stex_mathhub_str / \l__stex_refs_repob_str
1321
            / source / \l_stex_refs_in_str . sref
1322
         }
1323
       }
1324
       \stex_path_from_string:Nn \l_tmpb_seq \l_tmpa_str
1325
       \stex_path_to_string:NN \l_tmpb_seq \l_tmpa_str
1327
       \stex_debug:nn{sref}{File: \l_tmpa_str}
       \exp_args:No \IfFileExists \l_tmpa_str {
1328
1329
          \tl_clear:N \l__stex_refs_docname_tl
          \tl_clear:N \l__stex_refs_return_tl
1330
          \str_set:Nn \l__stex_refs_id_str {#1}
          \let\STEXInternalSrefDocumentName\__stex_refs_document_name:n
          \let\STEXInternalSrefRestoreTarget\__stex_refs_restore_target:nnnnn
          \use:c{@ @ input}{\l_tmpa_str}
1334
1335
          \exp_args:No \tl_if_empty:nTF \l__stex_refs_return_tl {
            \exp_args:Nnno \msg_warning:nnnn{stex}{warning/smslabelmissing}\l_tmpa_str{#1}
              _stex_refs_do_autoref:n{
              \str_if_empty:NF\l__stex_refs_uri_str{\l__stex_refs_uri_str?}#1
           }
1330
         }{
1340
              __stex_refs_return_tl
1341
1342
       }{
1343
          \exp_args:Nnno \msg_warning:nnn{stex}{warning/smsmissing}\l_tmpa_str
1344
          \__stex_refs_do_autoref:n{
1345
1346
            \str_if_empty:NF\l__stex_refs_uri_str{\l__stex_refs_uri_str?}#1
1347
1348
       }
1349
     \endgroup
1350 }
```

```
1351
    % \__stex_refs_args:n { #1 }
1352
    % \str_if_empty:NTF \l__stex_refs_indocument_str {
1353
         \str_set:Nx \l_tmpa_str { #2 }
1354
         \exp_args:NNno \seq_set_split:Nnn \l_tmpa_seq ? \l_tmpa_str
1355
         \int_compare:nNnTF {\seq_count:N \l_tmpa_seq} = 1 {
1356
           \seq_if_exist:cTF{g__stex_refs_labels_\l_tmpa_str _seq}{
1357
    %
             \seq_get_left:cNF {g__stex_refs_labels_\l_tmpa_str _seq} \l_tmpa_str {
1358
    %
               \str_clear:N \l_tmpa_str
    %
    %
           }{
    %
             \str_clear:N \l_tmpa_str
1362
           }
    %
1363
         }{
1364
    %
    %
           \seq_pop_left:NN \l_tmpa_seq \l_tmpb_str
1365
           \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
    %
1366
          \int_set:Nn \l_tmpa_int { \exp_args:Ne \str_count:n {\l_tmpb_str?\l_tmpa_str} }
1367
           \seq_if_exist:cTF{g__stex_refs_labels_\l_tmpa_str _seq}{
1368
    %
             \str_set_eq:NN \l_tmpc_str \l_tmpa_str
             \str_clear:N \l_tmpa_str
1371
    %
             \seq_map_inline:cn {g__stex_refs_labels_\l_tmpc_str _seq} {
               \str_if_eq:eeT { \l_tmpb_str?\l_tmpc_str }{
1372
    %
                  \str_range:nnn { ##1 }{ -\l_tmpa_int}{ -1 }
1373
    %
               }{
1374
    %
    %
                  \seq_map_break:n {
    %
                    \str_set:Nn \l_tmpa_str { ##1 }
1376
1377
    %
               }
1378
    %
    %
             }
1379
    %
           }{
1381
    %
             \str_clear:N \l_tmpa_str
           }
1382
    %
1383
    %
         \str_if_empty:NTF \l_tmpa_str {
1384
    %
           \tl_if_empty:NTF \l__stex_refs_linktext_tl \l__stex_refs_fallback_tl \l__stex_refs_li
    %
1385
    %
1386
    %
           \str_if_eq:cNTF {sref_\l_tmpa_str _type} \c__stex_refs_ref_str {
1387
1388
    %
             \tl_if_empty:NTF \l__stex_refs_linktext_tl {
1389
               \cs_if_exist:cTF{autoref}{
    %
                  \l__stex_refs_pre_tl\exp_args:Nx\autoref{sref_\l_tmpa_str}\l__stex_refs_post_tl
    %
               }{
    %
                  \l__stex_refs_pre_tl\exp_args:Nx\ref{sref_\l_tmpa_str}\l__stex_refs_post_tl
               }
1393
    %
             }{
1394
    %
               \ltx@ifpackageloaded{hyperref}{
    %
1395
    %
                  \hyperref[sref_\l_tmpa_str]\l__stex_refs_linktext_tl
1396
    %
1397
    %
                  \label{locality} $$ l_stex_refs_linktext_tl $$
1398
               }
    %
1399
    %
             }
1400
    %
           }{
1402
    %
             \ltx@ifpackageloaded{hyperref}{
               \href{\use:c{sref_url_\l_tmpa_str _str}}{\tl_if_empty:NTF \l__stex_refs_linktext_
1403
    %
    %
             }{
1404
```

```
}
                                     %
                          1407
                                    %
                                               }
                          1408
                                    % }{
                          1409
                                             % TODO
                          1410
                                   % }
                          1411
                          1412 %}
                         (End definition for \sref. This function is documented on page 77.)
\srefsym
                          1413 \NewDocumentCommand \srefsym { O{} m}{
                                        \stex_get_symbol:n { #2 }
                                        \__stex_refs_sym_aux:nn{#1}{\l_stex_get_symbol_uri_str}
                          1416 }
                          1417
                                   \cs_new_protected:Nn \__stex_refs_sym_aux:nn {
                          1418
                          1419
                          1420 %
                                          \str_if_exist:cTF {sref_sym_#2 _label_str }{
                                               \sref[#1]{\use:c{sref_sym_#2 _label_str}}
                          1421 %
                          1422 %
                          1423 %
                                               \__stex_refs_args:n { #1 }
                          1424 %
                                               \str_if_empty:NTF \l__stex_refs_indocument_str {
                          1425 %
                                                    \tl_if_exist:cTF{sref_sym_#2 _type}{
                          1426 %
                                                         % doc uri in \l_tmpb_str
                          1427 %
                                                         \str_set:Nx \l_tmpa_str {\use:c{sref_sym_#2 _type}}
                          1428 %
                                                         \str_if_eq:NNTF \l_tmpa_str \c__stex_refs_ref_str {
                                                              % reference
                          1429 %
                          1430 %
                                                              \tl_if_empty:NTF \l__stex_refs_linktext_tl {
                          1431 %
                                                                    \cs_if_exist:cTF{autoref}{
                          1432 %
                                                                        1433 %
                                                                         \l__stex_refs_pre_tl\ref{sref_sym_#2}\l__stex_refs_post_tl
                                                                   }
                          1435
                                  %
                                                              }{
                          1436 %
                          1437 %
                                                                    \ltx@ifpackageloaded{hyperref}{
                          1438 %
                                                                        \hyperref[sref_sym_#2]\l__stex_refs_linktext_tl
                                                                   }{
                          1439 %
                                                                         \label{local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_loc
                          1440 %
                          1441 %
                                                              }
                          1442 %
                          1443 %
                                                         }{
                                                              % URL
                                                              \ltx@ifpackageloaded{hyperref}{
                          1446 %
                                                                    \href{\use:c{sref_sym_url_#2 _str}}{\tl_if_empty:NTF \l__stex_refs_linktext_tl
                          1447 %
                                                              }{
                                                                    \tl_if_empty:NTF \l__stex_refs_linktext_tl \l__stex_refs_fallback_tl \l__stex_r
                          1448 %
                          1449 %
                                                              }
                          1450 %
                                                         }
                          1451 %
                                                    }{
                          1452 %
                                                          \tl_if_empty:NTF \l__stex_refs_linktext_tl \l__stex_refs_fallback_tl \l__stex_refs_
                          1453 %
```

\tl_if_empty:NTF \l__stex_refs_linktext_tl \l__stex_refs_fallback_tl \l__stex_ref

%

1406 %

1405

1454 %

}{

```
1455 % % TODO
1456 % }
1457 % }
1458 }

(End definition for \srefsym. This function is documented on page 77.)

\srefsymuri

1459 \cs_new_protected:Npn \srefsymuri #1 #2 {
1460 \__stex_refs_sym_aux:nn{linktext={#2}}{#1}}
1461 }

(End definition for \srefsymuri. This function is documented on page 77.)

1462 \( /package \)
```

Chapter 27

STEX -Modules Implementation

```
1463 (*package)
                              1464
                              modules.dtx
                                                                <@@=stex_modules>
                                  Warnings and error messages
                                 \msg_new:nnn{stex}{error/unknownmodule}{
                                   No~module~#1~found
                              1471 \msg_new:nnn{stex}{error/syntax}{
                                   Syntax~error:~#1
                              1472
                              1473 }
                              1474 \msg_new:nnn{stex}{error/siglanguage}{
                                   Module~#1~declares~signature~#2,~but~does~not~
                              1475
                                   declare~its~language
                              1476
                                 \msg_new:nnn{stex}{warning/deprecated}{
                                   #1~is~deprecated;~please~use~#2~instead!
                              1480 }
                              1482 \msg_new:nnn{stex}{error/conflictingmodules}{
                                   Conflicting~imports~for~module~#1
                              1484 }
                             The current module:
\l_stex_current_module_str
                              1485 \str_new:N \l_stex_current_module_str
                             (End definition for \l_stex_current_module_str. This variable is documented on page 79.)
                             Stores all available modules
   \l_stex_all_modules_seq
                              1486 \seq_new:N \l_stex_all_modules_seq
                             (End definition for \l_stex_all_modules_seq. This variable is documented on page 79.)
```

```
\stex_if_in_module_p:
     \stex_if_in_module: <u>TF</u>
                                1487 \prg_new_conditional:Nnn \stex_if_in_module: {p, T, F, TF} {
                                     \str_if_empty:NTF \l_stex_current_module_str
                                1488
                                        \prg_return_false: \prg_return_true:
                                1489
                                1490 }
                               (End definition for \stex_if_in_module:TF. This function is documented on page 79.)
\stex_if_module_exists_p:n
\stex_if_module_exists:nTF
                                   \prg_new_conditional:Nnn \stex_if_module_exists:n {p, T, F, TF} {
                                      \prop_if_exist:cTF { c_stex_module_#1_prop }
                                        \prg_return_true: \prg_return_false:
                                1494 }
                               (End definition for \stex if module exists:nTF. This function is documented on page 79.)
                               Only allowed within modules:
       \stex add to current module:n
                \STEXexport
                                1495 \cs_new_protected:Nn \stex_execute_in_module:n { \stex_if_in_module:T {
                                      \stex_add_to_current_module:n { #1 }
                                1496
                                      \stex_do_up_to_module:n { #1 }
                                1497
                                1499
                                   \cs_generate_variant:Nn \stex_execute_in_module:n {x}
                                   \cs_new_protected:Nn \stex_add_to_current_module:n {
                                1502
                                      \tl_gput_right:cn {c_stex_module_\l_stex_current_module_str _code} { #1 }
                               1503 }
                                   \cs_generate_variant:Nn \stex_add_to_current_module:n {x}
                                1504
                                   \cs_new_protected:Npn \STEXexport {
                                1505
                                      \ExplSyntaxOn
                                1506
                                      \__stex_modules_export:n
                                1507
                               1508 }
                                   \cs_new_protected:Nn \__stex_modules_export:n {
                                      \ignorespacesandpars#1\ExplSyntaxOff
                                      \stex_add_to_current_module:n { \ignorespacesandpars#1}
                                1511
                                      \stex_smsmode_do:
                                1512
                                1513 }
                                1514 \let \stex_module_export_helper:n \use:n
                                1515 \stex_deactivate_macro:Nn \STEXexport {module~environments}
                               (End definition for \stex_add_to_current_module:n and \STEXexport. These functions are documented
                               on page 79.)
\stex add constant to current module:n
                                1516 \cs_new_protected:Nn \stex_add_constant_to_current_module:n {
                                      \str_set:Nx \l_tmpa_str { #1 }
                                      \seq_gput_right:co {c_stex_module_\l_stex_current_module_str _constants} { \l_tmpa_str }
                                1518
                               1519 }
                               (End definition for \stex_add_constant_to_current_module:n. This function is documented on page
                               79.)
  \stex_add_import_to_current_module:n
                                {\tt 1520 \ \backslash cs\_new\_protected: Nn \ \backslash stex\_add\_import\_to\_current\_module:n \ \{}
                                      \str_set:Nx \l_tmpa_str { #1 }
                                1521
                                      \exp_args:Nno
                                1522
```

```
\seq_if_in:cnF{c_stex_module_\l_stex_current_module_str _imports}\l_tmpa_str{
                                   \seq_gput_right:co{c_stex_module_\l_stex_current_module_str _imports}\l_tmpa_str
                           1524
                           1525
                           1526 }
                           (End definition for \stex_add_import_to_current_module:n. This function is documented on page 79.)
\stex_collect_imports:n
                              \cs_new_protected:Nn \stex_collect_imports:n {
                                 \seq_clear:N \l_stex_collect_imports_seq
                           1528
                                 \__stex_modules_collect_imports:n {#1}
                           1529
                           1530 }
                               \cs_new_protected:Nn \__stex_modules_collect_imports:n {
                           1531
                                 \seq_map_inline:cn {c_stex_module_#1_imports} {
                           1532
                                   \seq_if_in:NnF \l_stex_collect_imports_seq { ##1 } {
                           1533
                                     \__stex_modules_collect_imports:n { ##1 }
                           1534
                           1535
                           1536
                                 \seq_if_in:NnF \l_stex_collect_imports_seq { #1 } {
                           1537
                                   \seq_put_right:Nx \l_stex_collect_imports_seq { #1 }
                           1538
                           1539
                           1540 }
                           (End definition for \stex_collect_imports:n. This function is documented on page 79.)
\stex_do_up_to_module:n
                              \int_new:N \l__stex_modules_group_depth_int
                               \cs_new_protected:Nn \stex_do_up_to_module:n {
                                 \int_compare:nNnTF \l__stex_modules_group_depth_int = \currentgrouplevel {
                           1544
                                   #1
                                 }{
                           1545
                                   #1
                           1546
                                   \expandafter \tl_gset:Nn
                           1547
                                   \csname l_stex_modules_aftergroup_\l_stex_current_module_str _tl
                           1548
                                   \expandafter\expandafter\expandafter\endcsname
                           1549
                                   \expandafter\expandafter\expandafter { \csname
                                     l__stex_modules_aftergroup_\l_stex_current_module_str _tl\endcsname #1 }
                           1551
                                   \aftergroup\__stex_modules_aftergroup_do:
                           1553
                           1554 }
                               \cs_generate_variant:Nn \stex_do_up_to_module:n {x}
                               \cs_new_protected: Nn \__stex_modules_aftergroup_do: {
                           1556
                                 \stex_debug:nn{aftergroup}{\cs_meaning:c{
                           1557
                                   l_stex_modules_aftergroup_\l_stex_current_module_str _tl
                           1558
                                 }}
                           1559
                                 \int_compare:nNnTF \1 _stex_modules_group_depth_int = \currentgrouplevel {
                           1560
                                   \use:c{l__stex_modules_aftergroup_\l_stex_current_module_str _tl}
                           1561
                                   \tl_gclear:c{l__stex_modules_aftergroup_\l_stex_current_module_str _tl}
                           1562
                           1563
                                   \use:c{l__stex_modules_aftergroup_\l_stex_current_module_str _tl}
                           1565
                                   \aftergroup\__stex_modules_aftergroup_do:
                                 }
                           1566
                           1567
                               \cs_new_protected:Nn \_stex_reset_up_to_module:n {
                           1568
                                 \expandafter\let\csname l__stex_modules_aftergroup_#1_tl\endcsname\undefined
```

```
1570 }
```

(End definition for \stex_do_up_to_module:n. This function is documented on page 79.)

\stex modules compute namespace:nN

Computes the appropriate namespace from the top-level namespace of a repository (#1) and a file path (#2).

157

(End definition for \stex_modules_compute_namespace:nN. This function is documented on page ??.)

\stex_modules_current_namespace:

Computes the current namespace based on the current MathHub repository (if existent) and the current file.

```
1572 \str_new:N \l_stex_module_ns_str
   \str_new:N \l_stex_module_subpath_str
   \cs_new_protected:Nn \__stex_modules_compute_namespace:nN {
      \seq_set_eq:NN \l_tmpa_seq #2
1575
     % split off file extension
1576
      \seq_pop_right:NN \l_tmpa_seq \l_tmpb_str % <- filename
1577
      \exp_args:NNno \seq_set_split:Nnn \l_tmpb_seq . \l_tmpb_str
     \seq_get_left:NN \l_tmpb_seq \l_tmpb_str % <- filename without suffixes
      \seq_put_right:No \l_tmpa_seq \l_tmpb_str % <- file path including name without suffixes
1580
1581
      \bool_set_true:N \l_tmpa_bool
1582
      \bool_while_do:Nn \l_tmpa_bool {
1583
        \seq_pop_left:NN \l_tmpa_seq \l_tmpb_str
1584
        \exp_args:No \str_case:nnTF { \l_tmpb_str } {
1585
          {source} { \bool_set_false:N \l_tmpa_bool }
1586
1587
          \seq_if_empty:NT \l_tmpa_seq {
1588
            \bool_set_false:N \l_tmpa_bool
       }
1591
     }
1592
1593
     \stex_path_to_string:NN \l_tmpa_seq \l_stex_module_subpath_str
1594
     % \l_tmpa_seq <- sub-path relative to archive</pre>
1595
     \str_if_empty:NTF \l_stex_module_subpath_str {
1596
        \str_set:Nx \l_stex_module_ns_str {#1}
1597
1598
        \str_set:Nx \l_stex_module_ns_str {
          #1/\l_stex_module_subpath_str
1601
     }
1602
1603 }
1604
   \cs_new_protected:Nn \stex_modules_current_namespace: {
1605
      \str_clear:N \l_stex_module_subpath_str
1606
      \prop_if_exist:NTF \l_stex_current_repository_prop {
1607
        \prop_get:NnN \l_stex_current_repository_prop { ns } \l_tmpa_str
1608
        \__stex_modules_compute_namespace:nN \l_tmpa_str \g_stex_currentfile_seq
1609
     }{
       % split off file extension
1611
       \seq_set_eq:NN \l_tmpa_seq \g_stex_currentfile_seq
1612
        \seq_pop_right:NN \l_tmpa_seq \l_tmpb_str
1613
```

```
\exp_args:NNno \seq_set_split:Nnn \l_tmpb_seq . \l_tmpb_str
1614
        \seq_get_left:NN \l_tmpb_seq \l_tmpb_str
1615
        \seq_put_right:No \l_tmpa_seq \l_tmpb_str
1616
        \str_set:Nx \l_stex_module_ns_str {
1617
          file:/\stex_path_to_string:N \l_tmpa_seq
1618
1619
     }
1620
1621 }
```

(End definition for \stex_modules_current_namespace: This function is documented on page 80.)

27.1 The smodule environment

smodule arguments:

```
1622 \keys_define:nn { stex / module } {
 1623
      title
                     .tl_set:N
                                  = \smoduletitle ,
                     .str_set_x:N = \smoduletype ,
 1624
      type
                     .str_set_x:N = \smoduleid ,
      id
 1625
                     .str_set_x:N = \l_stex_module_deprecate_str ,
      deprecate
 1626
                     .str_set_x:N = \l_stex_module_ns_str ,
      ns
 1627
      lang
                     .str_set_x:N = \l_stex_module_lang_str ,
 1628
                     .str_set_x:N = \l_stex_module_sig_str ,
      sig
 1629
                     .str_set_x:N = \l_stex_module_creators_str ,
      creators
 1630
      contributors .str_set_x:N = \l_stex_module_contributors_str,
                     .str_set_x:N = \l_stex_module_meta_str ,
      meta
 1632
                     .str_set_x:N = \l_stex_module_srccite_str
 1633
      srccite
1634 }
 1635
    \cs_new_protected:Nn \__stex_modules_args:n {
 1636
      \str_clear:N \smoduletitle
 1637
      \str_clear:N \smoduletype
 1638
      \str_clear:N \smoduleid
 1639
      \str_clear:N \l_stex_module_ns_str
 1640
      \str_clear:N \l_stex_module_deprecate_str
      \str_clear:N \l_stex_module_lang_str
 1642
      \str_clear:N \l_stex_module_sig_str
 1643
      \str_clear:N \l_stex_module_creators_str
 1644
      \verb|\str_clear:N \l_stex_module_contributors_str|\\
 1645
      \str_clear:N \l_stex_module_meta_str
 1646
      \str_clear:N \l_stex_module_srccite_str
 1647
      \keys_set:nn { stex / module } { #1 }
 1648
 1649 }
 1650
 1651 % module parameters here? In the body?
 1652
Sets up a new module property list:
 1653 \cs_new_protected:Nn \stex_module_setup:nn {
```

\stex_module_setup:nn

```
\int_set:Nn \l__stex_modules_group_depth_int {\currentgrouplevel}
1654
     \str_set:Nx \l_stex_module_name_str { #2 }
1655
     \__stex_modules_args:n { #1 }
```

First, we set up the name and namespace of the module. Are we in a nested module?

\stex_if_in_module:TF {

```
% Nested module
1658
        \prop_get:cnN {c_stex_module_\l_stex_current_module_str _prop}
1659
          { ns } \l_stex_module_ns_str
1660
        \str_set:Nx \l_stex_module_name_str {
1661
          \prop_item:cn {c_stex_module_\l_stex_current_module_str _prop}
1662
            { name } / \l_stex_module_name_str
1663
        \str_if_empty:NT \l_stex_module_lang_str {
1665
          \str_set:Nx \l_stex_module_lang_str {
1666
            \prop_item:cn {c_stex_module_\l_stex_current_module_str _prop}
1667
              { lang }
1668
1669
       }
1670
     }{
1671
       % not nested:
1672
1673
        \str_if_empty:NT \l_stex_module_ns_str {
          \stex_modules_current_namespace:
          \exp_args:NNNo \seq_set_split:Nnn \l_tmpa_seq
1676
              / {\l_stex_module_ns_str}
          \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
1677
          \str_if_eq:NNT \l_tmpa_str \l_stex_module_name_str {
1678
            \str_set:Nx \l_stex_module_ns_str {
1679
              \verb|\stex_path_to_string:N \l_tmpa_seq|
1680
1681
         }
1682
        }
1683
     }
1684
    Next, we determine the language of the module:
     \str_if_empty:NT \l_stex_module_lang_str {
1685
1686
        \seq_get_right:NN \g_stex_currentfile_seq \l_tmpa_str
        \seq_set_split:NnV \l_tmpa_seq . \l_tmpa_str
1687
        \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str % .tex
1688
        \exp_args:No \str_if_eq:nnF \l_tmpa_str {tex} {
1689
          \exp_args:No \str_if_eq:nnF \l_tmpa_str {dtx} {
1690
            \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq \l_tmpa_str
1691
         }
        \seq_pop_left:NN \l_tmpa_seq \l_tmpa_str % <filename>
        \seq_if_empty:NF \l_tmpa_seq { %remaining element should be [<something>.]language
          \seq_pop_right:NN \l_tmpa_seq \l_stex_module_lang_str
          \stex_debug:nn{modules} {Language~\l_stex_module_lang_str~
            inferred~from~file~name}
1698
1699
     }
1700
1701
     \stex_if_smsmode:F { \str_if_empty:NF \l_stex_module_lang_str {
1702
       \exp_args:NNo \stex_set_language:Nn \l_tmpa_str \l_stex_module_lang_str
     }}
```

We check if we need to extend a signature module, and set \l_stex_current_-module_prop accordingly:

```
\str_if_empty:NTF \l_stex_module_sig_str {
1705
       \exp_args:Nnx \prop_gset_from_keyval:cn {
1706
         c_stex_module_\l_stex_module_ns_str?\l_stex_module_name_str _prop
1708
         name
                    = \l_stex_module_name_str ,
1709
                    = \l_stex_module_ns_str ,
         file
                    = \exp_not:o { \g_stex_currentfile_seq } ,
         lang
                    = \l_stex_module_lang_str ,
1712
                    = \l_stex_module_sig_str ,
1713
         deprecate = \l_stex_module_deprecate_str ,
1714
                    = \l_stex_module_meta_str
         meta
1715
1716
       \seq_clear:c {c_stex_module_\l_stex_module_ns_str?\l_stex_module_name_str _imports}
       \seq_clear:c {c_stex_module_\l_stex_module_ns_str?\l_stex_module_name_str _constants}
1718
       \seq_clear:c {c_stex_module_\l_stex_module_ns_str?\l_stex_module_name_str _copymodules}
1719
       \tl_clear:c {c_stex_module_\l_stex_module_ns_str?\l_stex_module_name_str _code}
       \str_set:Nx\l_stex_current_module_str{\l_stex_module_ns_str?\l_stex_module_name_str}
    We load the metatheory:
       \str_if_empty:NT \l_stex_module_meta_str {
         \str set:Nx \l stex module meta str {
            \c_stex_metatheory_ns_str ? Metatheory
1724
1725
       }
       \str_if_eq:VnF \l_stex_module_meta_str {NONE} {
         \bool_set_true:N \l_stex_in_meta_bool
         \exp_args:Nx \stex_add_to_current_module:n {
1729
            \bool_set_true:N \l_stex_in_meta_bool
1730
            \stex_activate_module:n {\l_stex_module_meta_str}
            \bool_set_false:N \l_stex_in_meta_bool
1732
          \stex_activate_module:n {\l_stex_module_meta_str}
1734
          \bool_set_false:N \l_stex_in_meta_bool
1736
       \str_if_empty:NT \l_stex_module_lang_str {
         \msg_error:nnxx{stex}{error/siglanguage}{
1739
           \l_stex_module_ns_str?\l_stex_module_name_str
1740
1741
         }{\l_stex_module_sig_str}
1742
       \stex_debug:nn{modules}{Signature~\l_stex_module_sig_str~for~\l_stex_module_ns_str?\l_st
1743
       \stex if module exists:nTF{\l stex module ns str?\l stex module name str}{
1744
         \stex_debug:nn{modules}{(already exists)}
1745
1746
         \stex_debug:nn{modules}{(needs loading)}
1747
         \seq_set_eq:NN \l_tmpa_seq \g_stex_currentfile_seq
1748
         \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
         \seq_set_split:NnV \l_tmpb_seq . \l_tmpa_str
1750
         \seq_pop_right:NN \l_tmpb_seq \l_tmpa_str % .tex
1751
          \seq_pop_left:NN \l_tmpb_seq \l_tmpa_str % <filename>
1752
         \str_set:Nx \l_tmpa_str {
1753
            \stex_path_to_string:N \l_tmpa_seq /
1754
```

```
\IfFileExists \l_tmpa_str {
                                    \exp_args:No \stex_file_in_smsmode:nn { \l_tmpa_str } {
                       1758
                                      \str_clear:N \l_stex_current_module_str
                       1759
                                      \seq_clear:N \l_stex_all_modules_seq
                       1760
                                      \stex_debug:nn{modules}{Loading~signature}
                       1761
                                    }
                       1762
                                  }{
                                    \msg_error:nnx{stex}{error/unknownmodule}{for~signature~\l_tmpa_str}
                                  }
                       1765
                               }
                       1766
                                \stex_if_smsmode:F {
                       1767
                                  \stex_activate_module:n {
                       1768
                                    \l_stex_module_ns_str ? \l_stex_module_name_str
                       1769
                       1770
                        1771
                                \str_set:Nx\l_stex_current_module_str{\l_stex_module_ns_str?\l_stex_module_name_str}
                        1772
                        1773
                              \str_if_empty:NF \l_stex_module_deprecate_str {
                        1774
                                \msg_warning:nnxx{stex}{warning/deprecated}{
                        1775
                                  Module~\l_stex_current_module_str
                       1776
                       1778
                                  \l_stex_module_deprecate_str
                       1779
                       1780
                       1781
                              \seq_put_right:Nx \l_stex_all_modules_seq {
                                \l_stex_module_ns_str ? \l_stex_module_name_str
                       1782
                       1783
                              \tl_clear:c{l__stex_modules_aftergroup_\l_stex_module_ns_str ? \l_stex_module_name_str _tl
                       1784
                       1785 }
                       (End definition for \stex module setup:nn. This function is documented on page 80.)
                      The module environment.
             smodule
\ stex modules begin module:
                       implements \begin{smodule}
                           \cs_new_protected: Nn \__stex_modules_begin_module: {
                             \stex_reactivate_macro:N \STEXexport
                             \stex_reactivate_macro:N \importmodule
                             \stex_reactivate_macro:N \symdecl
                       1789
                              \stex_reactivate_macro:N \notation
                        1790
                             \stex_reactivate_macro:N \symdef
                       1791
                       1792
                              \stex_debug:nn{modules}{
                       1793
                               New~module:\\
                       1794
                               Namespace:~\l_stex_module_ns_str\\
                       1795
                               Name:~\l_stex_module_name_str\\
                       1796
                               Language:~\l_stex_module_lang_str\\
                       1797
                       1798
                               Signature:~\l_stex_module_sig_str\\
                       1799
                               Metatheory:~\l_stex_module_meta_str\\
                       1800
                               File:~\stex_path_to_string:N \g_stex_currentfile_seq
                       1801
```

\l_tmpa_str . \l_stex_module_sig_str .tex

}

1756

```
\stex_if_do_html:T{
                                       \begin{stex_annotate_env} {theory} {
                               1804
                                         \l_stex_module_ns_str ? \l_stex_module_name_str
                               1805
                               1806
                               1807
                                       \stex_annotate_invisible:nnn{header}{} {
                               1808
                                         \stex_annotate:nnn{language}{ \l_stex_module_lang_str }{}
                               1809
                                         \stex_annotate:nnn{signature}{ \l_stex_module_sig_str }{}
                               1810
                                         \str_if_eq:VnF \l_stex_module_meta_str {NONE} {
                                            \stex_annotate:nnn{metatheory}{ \l_stex_module_meta_str }{}
                               1812
                               1813
                                          \str_if_empty:NF \smoduletype {
                               1814
                                            \stex_annotate:nnn{type}{\smoduletype}{}
                               1815
                               1816
                               1817
                               1818
                                     % TODO: Inherit metatheory for nested modules?
                               1819
                               1820 }
                                   \iffalse \end{stex_annotate_env} \fi %^A make syntax highlighting work again
                               (End\ definition\ for\ \verb|\__stex_modules_begin_module:.)
                              implements \end{module}
\__stex_modules_end_module:
                               1822 \cs_new_protected:Nn \__stex_modules_end_module: {
                                     \stex_debug:nn{modules}{Closing~module~\prop_item:cn {c_stex_module_\l_stex_current_module}
                               1823
                                     \_stex_reset_up_to_module:n \l_stex_current_module_str
                               1824
                                     \stex if smsmode:T {
                               1825
                                       \stex_persist:x {
                               1826
                               1827
                                          \prop_set_from_keyval:cn{c_stex_module_\l_stex_current_module_str _prop}{
                                            \exp_after:wN \prop_to_keyval:N \csname c_stex_module_\l_stex_current_module_str _pr
                               1828
                               1829
                                         \seq_set_from_clist:cn{c_stex_module_\l_stex_current_module_str _constants}{
                                            \seq_use:cn{c_stex_module_\l_stex_current_module_str _constants},
                               1831
                               1832
                               1833
                                         \seq_set_from_clist:cn{c_stex_module_\l_stex_current_module_str _imports}{
                                            \seq_use:cn{c_stex_module_\l_stex_current_module_str _imports},
                               1834
                               1835
                                         \tl_set:cn {c_stex_module_\l_stex_current_module_str _code}
                               1836
                               1837
                                       \exp_after:wN \let \exp_after:wN \l_tmpa_tl \csname c_stex_module_\l_stex_current_module
                               1838
                               1839
                                        \exp_after:wN \stex_persist:n \exp_after:wN { \exp_after:wN { \l_tmpa_tl } }
                                     }
                               1840
                               1841 }
                               (End\ definition\ for\ \verb|\__stex_modules_end_module:.)
                                   The core environment
                                   \iffalse \begin{stex_annotate_env} \fi \^^A make syntax highlighting work again
                                   \NewDocumentEnvironment { smodule } { O{} m } {
                               1843
                                     \stex_module_setup:nn{#1}{#2}
                               1844
                               1845
                                     %\par
                                     \stex_if_smsmode:F{
                                       \tl_if_empty:NF \smoduletitle {
                                         \exp_args:No \stex_document_title:n \smoduletitle
                               1848
                               1849
```

```
\tl_clear:N \l_tmpa_tl
                    1850
                             \clist_map_inline:Nn \smoduletype {
                    1851
                               \tl_if_exist:cT {__stex_modules_smodule_##1_start:}{
                    1852
                                 \tl_set:Nn \l_tmpa_tl {\use:c{__stex_modules_smodule_##1_start:}}
                    1853
                    1854
                             }
                    1855
                             \tl_if_empty:NTF \l_tmpa_tl {
                    1856
                               \__stex_modules_smodule_start:
                    1857
                     1859
                               \label{local_local_thm} \label{local_thm} \
                            }
                     1860
                          }
                    1861
                             _stex_modules_begin_module:
                    1862
                           \str_if_empty:NF \smoduleid {
                    1863
                             \stex_ref_new_doc_target:n \smoduleid
                    1864
                    1865
                           \stex_smsmode_do:
                    1866
                          {
                    1867
                           \__stex_modules_end_module:
                           \stex_if_smsmode:F {
                             \end{stex_annotate_env}
                             \clist_set:No \l_tmpa_clist \smoduletype
                    1871
                             \tl_clear:N \l_tmpa_tl
                    1872
                             \clist_map_inline:Nn \l_tmpa_clist {
                    1873
                               \tl_if_exist:cT {__stex_modules_smodule_##1_end:}{
                    1874
                                 \tl_set:Nn \l_tmpa_tl {\use:c{__stex_modules_smodule_##1_end:}}
                    1875
                    1876
                    1877
                             \tl_if_empty:NTF \l_tmpa_tl {
                    1878
                               \__stex_modules_smodule_end:
                            }{
                    1880
                    1881
                               \l_tmpa_tl
                            }
                    1882
                          }
                    1883
                    1884 }
\stexpatchmodule
                        \cs_new_protected:Nn \__stex_modules_smodule_start: {}
                        \cs_new_protected: Nn \__stex_modules_smodule_end: {}
                    1887
                        \newcommand\stexpatchmodule[3][] {
                    1888
                             \str_set:Nx \l_tmpa_str{ #1 }
                    1889
                             \str_if_empty:NTF \l_tmpa_str {
                    1890
                               \tl_set:Nn \__stex_modules_smodule_start: { #2 }
                    1891
                               \tl_set:Nn \__stex_modules_smodule_end: { #3 }
                     1892
                     1893
                               \exp_after:wN \tl_set:Nn \csname __stex_modules_smodule_#1_start:\endcsname{ #2 }
                               \exp_after:wN \tl_set:Nn \csname __stex_modules_smodule_#1_end:\endcsname{ #3 }
                    1897 }
```

(End definition for \stexpatchmodule. This function is documented on page 80.)

27.2 Invoking modules

\STEXModule \stex_invoke_module:n \NewDocumentCommand \STEXModule { m } { \exp_args:NNx \str_set:Nn \l_tmpa_str { #1 } 1899 \int_set:Nn \l_tmpa_int { \str_count:N \l_tmpa_str } 1900 \tl_set:Nn \l_tmpa_tl { 1901 \msg_error:nnx{stex}{error/unknownmodule}{#1} 1902 \seq_map_inline:Nn \l_stex_all_modules_seq { \str_set:Nn \l_tmpb_str { ##1 } 1905 \str_if_eq:eeT { \l_tmpa_str } { 1906 \str_range:Nnn \l_tmpb_str { -\l_tmpa_int } { -1 } 1907 } { 1908 \seq_map_break:n { 1909 \tl_set:Nn \l_tmpa_tl { 1910 \stex_invoke_module:n { ##1 } 1911 1912 } 1914 } 1915 1916 $\label{local_local_thm} \label{local_thm} \$ 1917 } 1918 \cs_new_protected:Nn \stex_invoke_module:n { 1919 \stex_debug:nn{modules}{Invoking~module~#1} 1920 \peek_charcode_remove:NTF ! { 1921 __stex_modules_invoke_uri:nN { #1 } 1922 1923 \peek_charcode_remove:NTF ? { __stex_modules_invoke_symbol:nn { #1 } } { 1926 \msg_error:nnx{stex}{error/syntax}{ 1927 ?~or~!~expected~after~ 1928 \c_backslash_str STEXModule{#1} 1929 1930 1931 } 1932 1933 } \cs_new_protected:Nn __stex_modules_invoke_uri:nN { \str_set:Nn #2 { #1 } 1937 1938 \cs_new_protected:Nn __stex_modules_invoke_symbol:nn { 1939 \stex_invoke_symbol:n{#1?#2} 1940 1941 } (End definition for \STEXModule and \stex_invoke_module:n. These functions are documented on page *80*.) \stex_activate_module:n 1942 \bool_new:N \l_stex_in_meta_bool

1943 \bool_set_false:N \l_stex_in_meta_bool

```
1944 \cs_new_protected:Nn \stex_activate_module:n {
1945   \stex_debug:nn{modules}{Activating~module~#1}
1946   \exp_args:NNx \seq_if_in:NnF \l_stex_all_modules_seq { #1 } {
1947    \seq_put_right:Nx \l_stex_all_modules_seq { #1 }
1948   \use:c{ c_stex_module_#1_code }
1949   }
1950 }

(End definition for \stex_activate_module:n. This function is documented on page 81.)
1951 \(/\package\)
```

Chapter 28

STEX -Module Inheritance Implementation

28.1 SMS Mode

```
\g_stex_smsmode_allowedmacros_tl
\g_stex_smsmode_allowedmacros_escape_tl
\g_stex_smsmode_allowedenvs_seq
```

```
1956 (@@=stex_smsmode)
1957 \tl_new:N \g_stex_smsmode_allowedmacros_tl
1958 \tl_new:N \g_stex_smsmode_allowedmacros_escape_tl
1959 \seq_new:N \g_stex_smsmode_allowedenvs_seq
1961 \tl_set:Nn \g_stex_smsmode_allowedmacros_tl {
     \makeatletter
      \makeatother
     \ExplSyntaxOn
     \ExplSyntaxOff
1965
     \rustexBREAK
1966
1967 }
1968
1969 \tl_set:Nn \g_stex_smsmode_allowedmacros_escape_tl {
1970
     \importmodule
1971
     \notation
     \symdecl
1973
     \STEXexport
1974
     \inlineass
1975
     \inlinedef
1976
     \inlineex
1977
     \endinput
1978
     \setnotation
```

```
\copynotation
                              1980
                                    \assign
                             1981
                                    \renamedec1
                             1982
                                    \donotcopy
                             1983
                                    \instantiate
                             1984
                                    \textsymdecl
                             1985
                             1986
                             1987
                                  \exp_args:NNx \seq_set_from_clist:Nn \g_stex_smsmode_allowedenvs_seq {
                                    \tl_to_str:n {
                             1989
                                      smodule,
                             1990
                                      copymodule,
                             1991
                                      interpretmodule,
                             1992
                                      realization,
                             1993
                                      sdefinition,
                             1994
                                      sexample,
                             1995
                                      sassertion,
                              1996
                                      sparagraph,
                                      mathstructure
                             1999
                                   }
                             2000 }
                             (End definition for \g_stex_smsmode_allowedmacros_t1, \g_stex_smsmode_allowedmacros_escape_t1,
                             and \g_stex_smsmode_allowedenvs_seq. These variables are documented on page 82.)
     \stex_if_smsmode_p:
     \stex_if_smsmode: TF
                             2001 \bool_new:N \g__stex_smsmode_bool
                                 \bool_set_false:N \g__stex_smsmode_bool
                                 \prg_new_conditional:Nnn \stex_if_smsmode: { p, T, F, TF } {
                                   \bool_if:NTF \g__stex_smsmode_bool \prg_return_true: \prg_return_false:
                             2004
                             2005 }
                             (End definition for \stex if smsmode: TF. This function is documented on page 82.)
     \ stex smsmode in smsmode:nn
                                 \cs_new_protected:Nn \__stex_smsmode_in_smsmode:nn { \stex_suppress_html:n {
                                    \vbox_set:Nn \l_tmpa_box {
                             2007
                                      \bool_set_eq:cN { l__stex_smsmode_#1_bool } \g__stex_smsmode_bool
                             2008
                                      \bool_gset_true:N \g__stex_smsmode_bool
                             2009
                             2010
                                      \bool_gset_eq:Nc \g__stex_smsmode_bool { l__stex_smsmode_#1_bool }
                             2011
                             2012
                                    \box_clear:N \l_tmpa_box
                             2013
                             2014 } }
                             (End\ definition\ for\ \_\_stex\_smsmode\_in\_smsmode:nn.)
\stex_file_in_smsmode:nn
                                 \quark_new:N \q__stex_smsmode_break
                                 \NewDocumentCommand \__stex_smsmode_importmodule: { O{} m} {
                             2017
                                    \seq_gput_right:Nn \l__stex_smsmode_importmodules_seq {{#1}{#2}}
                             2018
                                    \stex_smsmode_do:
                             2019
                             2020 }
                             2021
```

```
\cs_new_protected:Nn \__stex_smsmode_module:nn {
     \__stex_modules_args:n{#1}
2023
     \stex_if_in_module:F {
2024
       \str_if_empty:NF \l_stex_module_sig_str {
2025
         \stex_modules_current_namespace:
2026
         \str_set:Nx \l_stex_module_name_str { #2 }
2027
         \stex_if_module_exists:nF{\l_stex_module_ns_str?\l_stex_module_name_str}{
2028
           \seq_set_eq:NN \l_tmpa_seq \g_stex_currentfile_seq
2029
           \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
           \seq_set_split:NnV \l_tmpb_seq . \l_tmpa_str
2031
           \seq_pop_left:NN \l_tmpb_seq \l_tmpa_str % <filename>
2033
           \str_set:Nx \l_tmpa_str {
2034
              \stex_path_to_string:N \l_tmpa_seq /
2035
             \l_tmpa_str . \l_stex_module_sig_str .tex
2036
2037
           \IfFileExists \l_tmpa_str {
2038
              \exp_args:NNx \seq_gput_right:Nn \l__stex_smsmode_sigmodules_seq \l_tmpa_str
2039
              \msg_error:nnx{stex}{error/unknownmodule}{for~signature~\l_tmpa_str}
         }
2043
       }
2044
     }
2045
2046 }
2047
   \prg_new_conditional:Nnn \__stex_smsmode_check_import_pair:nn {T,F,TF} {
2048
     %\stex_debug:nn{import-pair}{\detokenize{{#1}~{#2}}}
2049
     \tl_if_empty:nTF{#1}{
2050
       \prop_if_exist:NTF \l_stex_current_repository_prop
2052
           %\stex_debug:nn{import-pair}{in repository \prop_item:Nn \l_stex_current_repository_
2053
2054
           \prg_return_true:
         } {
2055
           \seq_set_split:Nnn \l_tmpa_seq ? {#2}
2056
           \seq_get_left:NN \l_tmpa_seq \l_tmpa_tl
2057
           \tl_if_empty:NT \l_tmpa_tl {
2058
              \seq_pop_left:NN \l_tmpa_seq \l_tmpa_tl
2059
2060
           %\stex_debug:nn{import-pair}{\seq_use:Nn \l_tmpa_seq,~of~length~\seq_count:N \l_tmpa
           \int_compare:nNnTF {\seq_count:N \l_tmpa_seq} > 1
              \prg_return_true: \prg_return_false:
2065
     }\prg_return_true:
2066
2067
   \cs_new_protected:Nn \stex_file_in_smsmode:nn {
2068
     \stex_filestack_push:n{#1}
2069
     \seq_gclear:N \l__stex_smsmode_importmodules_seq
2070
2071
     \seq_gclear:N \l__stex_smsmode_sigmodules_seq
     % ---- new ------
2073
     \__stex_smsmode_in_smsmode:nn{#1}{
2074
       \let\importmodule\__stex_smsmode_importmodule:
       \let\stex_module_setup:nn\__stex_smsmode_module:nn
2075
```

```
\let\__stex_modules_begin_module:\relax
2076
        \let\__stex_modules_end_module:\relax
2077
        \seq_clear:N \g_stex_smsmode_allowedenvs_seq
2078
        \exp_args:NNx \seq_put_right:Nn \g_stex_smsmode_allowedenvs_seq {\tl_to_str:n{smodule}}
2079
        \tl_clear:N \g_stex_smsmode_allowedmacros_tl
2080
        \tl_clear:N \g_stex_smsmode_allowedmacros_escape_tl
2081
        \tl_put_right:Nn \g_stex_smsmode_allowedmacros_escape_tl {\importmodule}
2082
        \everyeof{\q_stex_smsmode_break\noexpand}
2083
        \expandafter\expandafter\expandafter
        \stex_smsmode_do:
        \csname @ @ input\endcsname "#1"\relax
2087
        \seq_map_inline:Nn \l__stex_smsmode_sigmodules_seq {
2088
          \stex_filestack_push:n{##1}
2089
          \expandafter\expandafter\expandafter
2090
          \stex_smsmode_do:
2091
          \csname @ @ input\endcsname "##1"\relax
2092
          \stex_filestack_pop:
2093
      % ---- new -----
      \__stex_smsmode_in_smsmode:nn{#1} {
2097
2098
        % ---- new ------
2099
        \begingroup
2100
        %\stex_debug:nn{smsmode}{Here:~\seq_use:Nn\l__stex_smsmode_importmodules_seq, }
        \seq_map_inline: Nn \l__stex_smsmode_importmodules_seq {
          \__stex_smsmode_check_import_pair:nnT ##1 { \begingroup
2103
            \stex_import_module_uri:nn ##1
2104
            \stex_import_require_module:nnnn
2106
              \l_stex_import_ns_str
2107
              \l_stex_import_archive_str
2108
              \l_stex_import_path_str
              \l_stex_import_name_str \endgroup
2109
         }
2111
        \endgroup
2112
2113
        \stex_debug:nn{smsmode}{Actually~loading~file~#1}
2114
        % ---- new ------
        \everyeof{\q__stex_smsmode_break\noexpand}
        \expandafter\expandafter\expandafter
        \stex_smsmode_do:
2117
        \csname @ @ input\endcsname "#1"\relax
2118
2119
      \stex_filestack_pop:
2120
2121 }
(End definition for \stex_file_in_smsmode:nn. This function is documented on page 83.)
```

\stex_smsmode_do: is executed on encountering \ in smsmode. It checks whether the corresponding command is allowed and executes or ignores it accordingly:

```
2122 \cs_new_protected:Npn \stex_smsmode_do: {
2123 \stex_if_smsmode:T {
2124 \__stex_smsmode_do:w
```

```
}
2126 }
    \cs_new_protected:Npn \__stex_smsmode_do:w #1 {
2127
      \exp_args:Nx \tl_if_empty:nTF { \tl_tail:n{ #1 }}{
2128
        \expandafter\if\expandafter\relax\noexpand#1
2129
           \expandafter\__stex_smsmode_do_aux:N\expandafter#1
2130
        \else\expandafter\__stex_smsmode_do:w\fi
2131
      }{
2132
2133
         \__stex_smsmode_do:w %#1
2134
2135 }
    \cs_new_protected:Nn \__stex_smsmode_do_aux:N {
2136
      \cs_if_eq:NNF #1 \q__stex_smsmode_break {
        \tl_if_in:NnTF \g_stex_smsmode_allowedmacros_tl {#1} {
2138
          #1\__stex_smsmode_do:w
2139
2140
           \tl_if_in:NnTF \g_stex_smsmode_allowedmacros_escape_tl {#1} {
2141
            #1
2142
          }{
             \cs_if_eq:NNTF \begin #1 {
               \__stex_smsmode_check_begin:n
            }{
2146
               \cs_{if}_{eq}:NNTF \end #1 {
2147
2148
                 \__stex_smsmode_check_end:n
2149
                 \__stex_smsmode_do:w
2150
               }
          }
2153
2154
        }
      }
2155
2156 }
    \cs_new_protected:Nn \__stex_smsmode_check_begin:n {
2158
      \seq_if_in:NxTF \g_stex_smsmode_allowedenvs_seq { \detokenize{#1} }{
2159
        \begin{#1}
2160
2161
         \__stex_smsmode_do:w
2162
2163
2164 }
2165
    \cs_new_protected:Nn \__stex_smsmode_check_end:n {
      \seq_if_in:NxTF \g_stex_smsmode_allowedenvs_seq { \detokenize{#1} }{
2167
        \end{#1}\__stex_smsmode_do:w
2168
        \str_if_eq:nnTF{#1}{document}{\endinput}{\__stex_smsmode_do:w}
2169
2170
2171 }
(End definition for \stex_smsmode_do:. This function is documented on page 83.)
```

28.2 Inheritance

```
2172 \langle @@=stex_importmodule \rangle
```

```
\stex_import_module_uri:nn
```

\l_stex_import_name_str
\l_stex_import_archive_str

\l_stex_import_path_str

\l_stex_import_ns_str

```
2173 \cs_new_protected:Nn \stex_import_module_uri:nn {
       \str_set:Nx \l_stex_import_archive_str { #1 }
 2174
       \str_set:Nn \l_stex_import_path_str { #2 }
 2175
 2176
       \exp_args:NNNo \seq_set_split:Nnn \l_tmpb_seq ? { \l_stex_import_path_str }
 2177
       \seq_pop_right:NN \l_tmpb_seq \l_stex_import_name_str
 2178
       \str_set:Nx \l_stex_import_path_str { \seq_use:Nn \l_tmpb_seq ? }
 2179
 2181
      \stex_modules_current_namespace:
 2182
      \bool_lazy_all:nTF {
         {\str_if_empty_p:N \l_stex_import_archive_str}
 2183
         {\str_if_empty_p:N \l_stex_import_path_str}
 2184
        {\stex_if_module_exists_p:n { \l_stex_module_ns_str ? \l_stex_import_name_str } }
 2185
 2186
         \str_set_eq:NN \l_stex_import_path_str \l_stex_module_subpath_str
 2187
         \str_set_eq:NN \l_stex_import_ns_str \l_stex_module_ns_str
 2188
 2189
         \str_if_empty:NT \l_stex_import_archive_str {
 2190
           \prop_if_exist:NT \l_stex_current_repository_prop {
 2191
             \prop_get:NnN \l_stex_current_repository_prop { id } \l_stex_import_archive_str
 2192
          }
 2193
 2194
         \str_if_empty:NTF \l_stex_import_archive_str {
 2195
           \str_if_empty:NF \l_stex_import_path_str {
 2196
             \stex_path_from_string:Nn \l_tmpb_seq {
 2197
               \l_stex_module_ns_str / .. / \l_stex_import_path_str
 2198
             }
 2199
             \str_set:Nx \l_stex_import_ns_str {\stex_path_to_string:N \l_tmpb_seq}
 2200
             \str_replace_once:Nnn \l_stex_import_ns_str {file://} {file://}
          }
        }{
 2203
           \stex_require_repository:n \l_stex_import_archive_str
 2204
           \prop_get:cnN { c_stex_mathhub_\l_stex_import_archive_str _manifest_prop } { ns }
 2205
             \l_stex_import_ns_str
 2206
           \str_if_empty:NF \l_stex_import_path_str {
 2207
             \str_set:Nx \l_stex_import_ns_str {
 2208
               \l_stex_import_ns_str / \l_stex_import_path_str
 2209
 2210
          }
        }
      }
 2213
 2214 }
(End definition for \stex_import_module_uri:nn. This function is documented on page 84.)
Store the return values of \stex_import_module_uri:nn.
 2215 \str_new:N \l_stex_import_name_str
 2216 \str_new:N \l_stex_import_archive_str
2217 \str_new:N \l_stex_import_path_str
 2218 \str_new:N \l_stex_import_ns_str
```

(End definition for \l_stex_import_name_str and others. These variables are documented on page 84.)

```
\stex_import_require_module:nnnn
                         \{\langle ns \rangle\} \ \{\langle archive-ID \rangle\} \ \{\langle path \rangle\} \ \{\langle name \rangle\}
                              \cs_new_protected:Nn \stex_import_require_module:nnnn {
                                \exp_args:Nx \stex_if_module_exists:nF { #1 ? #4 } {
                                  \stex_debug:nn{requiremodule}{Here:\\~~1:~#1\\~~2:~#2\\~~3:~#3\\~~4:~#4}
                                  \exp_args:NNxx \seq_set_split:Nnn \l_tmpa_seq {\tl_to_str:n{/}} {#4}
                          2224
                                  \seq_get_left:NN \l_tmpa_seq \l_tmpc_str
                          2226
                                  %\stex_debug:nn{requiremodule}{Top~module:\l_tmpc_str}
                          2227
                          2228
                                  % archive
                          2229
                                  \str_set:Nx \l_tmpa_str { #2 }
                          2230
                                  \str_if_empty:NTF \l_tmpa_str {
                                     \seq_set_eq:NN \l_tmpa_seq \g_stex_currentfile_seq
                                     \seq_put_right:Nn \l_tmpa_seq {..}
                                  } {
                          2234
                                     \stex_path_from_string:Nn \l_tmpb_seq { \l_tmpa_str }
                          2235
                                     \seq_concat:NNN \l_tmpa_seq \c_stex_mathhub_seq \l_tmpb_seq
                          2236
                                     \seq_put_right:Nn \l_tmpa_seq { source }
                          2238
                          2239
                                  % path
                          2240
                                  \str_set:Nx \l_tmpb_str { #3 }
                          2241
                                  \str_if_empty:NTF \l_tmpb_str {
                          2242
                                     \str_set:Nx \l_tmpa_str { \stex_path_to_string:N \l_tmpa_seq / \l_tmpc_str }
                                     \ltx@ifpackageloaded{babel} {
                                       \exp_args:NNx \prop_get:NnNF \c_stex_language_abbrevs_prop
                          2246
                                           { \languagename } \l_tmpb_str {
                          2247
                                              \msg_error:nnx{stex}{error/unknownlanguage}{\languagename}
                          2248
                          2249
                                    } {
                          2250
                                       \str_clear:N \l_tmpb_str
                          2253
                                     \stex_debug:nn{modules}{Checking~a1~\l_tmpa_str.\l_tmpb_str.tex}
                                     \IfFileExists{ \l_tmpa_str.\l_tmpb_str.tex }{
                                       \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str.\l_tmpb_str.tex }
                          2256
                                    }{
                          2257
                                       \stex_debug:nn{modules}{Checking~a2~\l_tmpa_str.tex}
                          2258
                                       \IfFileExists{ \l_tmpa_str.tex }{
                          2259
                                         \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str.tex }
                          2260
                                      }{
                          2261
                                         % try english as default
                          2262
                                         \stex_debug:nn{modules}{Checking~a3~\l_tmpa_str.en.tex}
                          2263
                                         \IfFileExists{ \l_tmpa_str.en.tex }{
                                           \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str.en.tex }
                                         ትና
                                           \msg_error:nnx{stex}{error/unknownmodule}{#1?#4}
                          2267
                                         }
                          2268
                                      }
                          2269
```

}

```
} {
          \seq_set_split:NnV \l_tmpb_seq / \l_tmpb_str
         \seq_concat:NNN \l_tmpb_seq \l_tmpa_seq \l_tmpb_seq
2274
2275
         \ltx@ifpackageloaded{babel} {
2276
            \exp_args:NNx \prop_get:NnNF \c_stex_language_abbrevs_prop
2277
                { \languagename } \l_tmpb_str {
2278
                  \msg_error:nnx{stex}{error/unknownlanguage}{\languagename}
2279
         } {
            \str_clear:N \l_tmpb_str
2283
2284
         \stex_path_canonicalize:N \l_tmpb_seq
2285
         \stex_path_to_string:NN \l_tmpb_seq \l_tmpa_str
2286
2287
         \stex_debug:nn{modules}{Checking~b1~\l_tmpa_str/\l_tmpc_str.\l_tmpb_str.tex}
2288
         \IfFileExists{ \l_tmpa_str/\l_tmpc_str.\l_tmpb_str.tex }{
            \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str/\l_tmpc_str.\l_tmpb_str.te
         }{
            \stex_debug:nn{modules}{Checking~b2~\l_tmpa_str/\l_tmpc_str.tex}
           \IfFileExists{ \l_tmpa_str/\l_tmpc_str.tex }{
              \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str/\l_tmpc_str.tex }
           }{
2295
              % try english as default
2296
              \stex_debug:nn{modules}{Checking~b3~\l_tmpa_str/\l_tmpc_str.en.tex}
2297
              \IfFileExists{ \l_tmpa_str/\l_tmpc_str.en.tex }{
2298
                \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str/\l_tmpc_str.en.tex }
2299
             }{
2300
                \stex_debug:nn{modules}{Checking~b4~\l_tmpa_str.\l_tmpb_str.tex}
                \IfFileExists{ \l_tmpa_str.\l_tmpb_str.tex }{
                  \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str.\l_tmpb_str.tex }
               }{
2304
                  \stex_debug:nn{modules}{Checking~b4~\l_tmpa_str.tex}
2305
                  \IfFileExists{ \l_tmpa_str.tex }{
2306
                    \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str.tex }
2307
                  }{
2308
                    % try english as default
2309
                    \stex_debug:nn{modules}{Checking~b5~\l_tmpa_str.en.tex}
                    \IfFileExists{ \l_tmpa_str.en.tex }{
                      \str_gset:Nx \g__stex_importmodule_file_str { \l_tmpa_str.en.tex }
                    }{
                      \msg_error:nnx{stex}{error/unknownmodule}{#1?#4}
2314
                    }
                  }
2316
               }
2317
             }
2318
           }
2319
         }
2321
2323
       \str_if_eq:eeF{\g__stex_importmodule_file_str}{\seq_use:Nn \g_stex_currentfile_seq /}{
2324
         \exp_args:No \stex_file_in_smsmode:nn { \g_stex_importmodule_file_str } {
```

\seq_clear:N \l_stex_all_modules_seq

```
\str_clear:N \l_stex_current_module_str
                             \str_set:Nx \l_tmpb_str { #2 }
                2327
                             \str_if_empty:NF \l_tmpb_str {
                2328
                               \stex_set_current_repository:n { #2 }
                2329
                2330
                             \stex_debug:nn{modules}{Loading~\g__stex_importmodule_file_str}
                2333
                           \stex_if_module_exists:nF { #1 ? #4 } {
                             \msg_error:nnx{stex}{error/unknownmodule}{
                2335
                               #1?#4~(in~file~\g_stex_importmodule_file_str)
                          }
                2338
                2339
                2340
                2341
                       \stex_activate_module:n { #1 ? #4 }
                2342
                2343 }
                (End definition for \stex_import_require_module:nnnn. This function is documented on page 84.)
\importmodule
                    \NewDocumentCommand \importmodule { O{} m } {
                2344
                      \stex_import_module_uri:nn { #1 } { #2 }
                2345
                      \stex_debug:nn{modules}{Importing~module:~
                2346
                         \l_stex_import_ns_str ? \l_stex_import_name_str
                2347
                      \stex_import_require_module:nnnn
                      { \l_stex_import_ns_str } { \l_stex_import_archive_str }
                      { \l_stex_import_path_str } { \l_stex_import_name_str }
                      \stex_if_smsmode:F {
                 2352
                         \stex_annotate_invisible:nnn
                2353
                           {import} {\l_stex_import_ns_str ? \l_stex_import_name_str} {}
                2354
                2355
                      \exp_args:Nx \stex_add_to_current_module:n {
                2356
                         \stex_import_require_module:nnnn
                2357
                         { \l_stex_import_ns_str } { \l_stex_import_archive_str }
                         { \l_stex_import_path_str } { \l_stex_import_name_str }
                2359
                2360
                      \exp_args:Nx \stex_add_import_to_current_module:n {
                2361
                         \l_stex_import_ns_str ? \l_stex_import_name_str
                2362
                2363
                2364
                      \stex_smsmode_do:
                      \ignorespacesandpars
                2365
                2366 }
                    \stex_deactivate_macro:Nn \importmodule {module~environments}
                (End definition for \importmodule. This function is documented on page 83.)
   \usemodule
                    \NewDocumentCommand \usemodule { O{} m } {
                      \stex_if_smsmode:F {
                2369
                         \stex_import_module_uri:nn { #1 } { #2 }
                        \stex_import_require_module:nnnn
                2371
                        { \l_stex_import_ns_str } { \l_stex_import_archive_str }
                2372
```

```
{ \l_stex_import_path_str } { \l_stex_import_name_str }
  2373
                                    \stex_annotate_invisible:nnn
  2374
                                             {usemodule} {\l_stex_import_ns_str ? \l_stex_import_name_str} {}
  2375
  2376
                            \stex_smsmode_do:
  2377
                           \ignorespacesandpars
  2378
  2379 }
(End definition for \uberline \ube
                 \cs_new_protected:Nn \stex_csl_to_imports:Nn {
                           \tl_if_empty:nF{#2}{
  2381
  2382
                                    \clist_set:Nn \l_tmpa_clist {#2}
  2383
                                     \clist_map_inline:Nn \l_tmpa_clist {
                                             \tl_if_head_eq_charcode:nNTF {##1}[{
  2385
                                                      #1 ##1
                                             }{
                                                      #1{##1}
  2387
  2388
  2389
  2390
  2391 }
                   \cs_generate_variant:Nn \stex_csl_to_imports:Nn {No}
  2392
  2393
  2394
  2395 (/package)
```

Chapter 29

STeX -Symbols Implementation

```
2396 (*package)
2397
symbols.dtx
                                 Warnings and error messages
   \msg_new:nnn{stex}{error/wrongargs}{
     args~value~in~symbol~declaration~for~#1~
     needs~to~be~i,~a,~b~or~B,~but~#2~given
2404 \msg_new:nnn{stex}{error/unknownsymbol}{
     No~symbol~#1~found!
2405
2406 }
   \msg_new:nnn{stex}{error/seqlength}{
     Expected~#1~arguments;~got~#2!
2408
2409 }
2410 \msg_new:nnn{stex}{error/unknownnotation}{
     Unknown~notation~#1~for~#2!
2412 }
```

29.1 Symbol Declarations

```
2413 (@@=stex_symdecl)
                      Map over all available symbols
\stex_all_symbols:n
                       2414 \cs_new_protected:Nn \stex_all_symbols:n {
                             \def \__stex_symdecl_all_symbols_cs ##1 {#1}
                       2415
                             \seq_map_inline:Nn \l_stex_all_modules_seq {
                       2416
                               \seq_map_inline:cn{c_stex_module_##1_constants}{
                       2417
                                 \__stex_symdecl_all_symbols_cs{##1?###1}
                       2418
                             }
                       2420
                       2421 }
                       (End definition for \stex_all_symbols:n. This function is documented on page 86.)
```

```
\STEXsymbol
```

```
2422 \NewDocumentCommand \STEXsymbol { m } {
      \stex_get_symbol:n { #1 }
2423
      \exp_args:No
2424
      \stex_invoke_symbol:n { \l_stex_get_symbol_uri_str }
2425
2426 }
(End definition for \STEXsymbol. This function is documented on page 87.)
    symdecl arguments:
2427 \keys_define:nn { stex / symdecl } {
                   .str_set_x:N = \l_stex_symdecl_name_str ;
      name
2428
                   .bool_set:N
                                 = \l_stex_symdecl_local_bool ,
      local
2429
                   .str_set_x:N = \l_stex_symdecl_args_str ,
      args
2430
                   .tl set:N
                                  = \l_stex_symdecl_type_tl ,
      type
2431
      deprecate
                   .str_set_x:N = \l_stex_symdecl_deprecate_str
2432
      align
                   .str_set:N
                                  = \l_stex_symdecl_align_str , % TODO(?)
2433
                                  = \l_stex_symdecl_gfc_str , % TODO(?)
      gfc
                   .str_set:N
2434
      specializes .str_set:N
                                  = \l_stex_symdecl_specializes_str , % TODO(?)
                                  = \l_stex_symdecl_definiens_tl ,
      def
                   .tl_set:N
2437
                   .str_set_x:N = \l_stex_symdecl_reorder_str
      reorder
                   .clist_set:N = \l_stex_symdecl_argnames_clist ,
2438
      argnames
      assoc
                   .choices:nn
2439
          {bin,binl,binr,pre,conj,pwconj}
2440
          {\str_set:Nx \l_stex_symdecl_assoctype_str {\l_keys_choice_tl}}
2441
2442
2443
    \bool_new:N \l_stex_symdecl_make_macro_bool
2444
    \cs_new_protected:Nn \__stex_symdecl_args:n {
      \str_clear:N \l_stex_symdecl_name_str
      \str_clear:N \l_stex_symdecl_args_str
2448
2449
      \str_clear:N \l_stex_symdecl_deprecate_str
      \str_clear:N \l_stex_symdecl_reorder_str
2450
      \str_clear:N \l_stex_symdecl_assoctype_str
2451
      \bool_set_false:N \l_stex_symdecl_local_bool
2452
      \tl_clear:N \l_stex_symdecl_type_tl
2453
      \tl_clear:N \l_stex_symdecl_definiens_tl
2454
      \clist_clear:N \l_stex_symdecl_argnames_clist
2455
      \keys_set:nn { stex / symdecl } { #1 }
2457
2458 }
```

\symdecl Parses the optional arguments and passes them on to \stex_symdecl_do: (so that \symdef can do the same)

```
2459
2460 \NewDocumentCommand \symdecl { s m O{}} {
2461  \__stex_symdecl_args:n { #3 }
2462  \IfBooleanTF #1 {
2463  \bool_set_false:N \l_stex_symdecl_make_macro_bool
2464 } {
2465  \bool_set_true:N \l_stex_symdecl_make_macro_bool
2466 }
2467  \stex_symdecl_do:n { #2 }
```

```
\stex_smsmode_do:
                      2469 }
                      2470
                          \cs_new_protected:Nn \stex_symdecl_do:nn {
                      2471
                            \__stex_symdecl_args:n{#1}
                      2472
                            \bool_set_false:N \l_stex_symdecl_make_macro_bool
                            \stex_symdecl_do:n{#2}
                      2474
                      2475
                      2476
                          \stex_deactivate_macro:Nn \symdecl {module~environments}
                      (End definition for \symdecl. This function is documented on page 85.)
\stex_symdecl_do:n
                      2478 \cs_new_protected:Nn \stex_symdecl_do:n {
                            \stex_if_in_module:F {
                      2479
                              % TODO throw error? some default namespace?
                      2480
                            }
                      2481
                      2482
                            \str_if_empty:NT \l_stex_symdecl_name_str {
                      2483
                              \str_set:Nx \l_stex_symdecl_name_str { #1 }
                      2484
                      2485
                      2486
                            \prop_if_exist:cT { l_stex_symdecl_
                      2487
                                \l_stex_current_module_str ?
                      2488
                                \l_stex_symdecl_name_str
                      2489
                      2490
                            }{
                      2491
                              % TODO throw error (beware of circular dependencies)
                      2492
                      2493
                      2494
                            \prop_clear:N \l_tmpa_prop
                      2495
                            \prop_put:\nx \l_tmpa_prop { module } { \l_stex_current_module_str }
                            \seq_clear:N \l_tmpa_seq
                            \prop_put:Nno \l_tmpa_prop { name } \l_stex_symdecl_name_str
                            \prop_put:Nno \l_tmpa_prop { type } \l_stex_symdecl_type_tl
                            \str_if_empty:NT \l_stex_symdecl_deprecate_str {
                      2501
                              \str_if_empty:NF \l_stex_module_deprecate_str {
                      2502
                                \str_set_eq:NN \l_stex_symdecl_deprecate_str \l_stex_module_deprecate_str
                      2503
                      2504
                      2505
                            \prop_put:Nno \l_tmpa_prop { deprecate } \l_stex_symdecl_deprecate_str
                      2506
                      2507
                            \exp_args:No \stex_add_constant_to_current_module:n {
                      2508
                              \l_stex_symdecl_name_str
                      2510
                      2511
                            % arity/args
                      2512
                            \int_zero:N \l_tmpb_int
                      2513
                      2514
                            \bool_set_true:N \l_tmpa_bool
                      2515
                            \str_map_inline:Nn \l_stex_symdecl_args_str {
                      2516
                              \token_case_meaning:NnF ##1 {
                      2517
```

```
0 {} 1 {} 2 {} 3 {} 4 {} 5 {} 6 {} 7 {} 8 {} 9 {}
2518
          {\tl_to_str:n i} { \bool_set_false:N \l_tmpa_bool }
2519
          {\tl_to_str:n b} { \bool_set_false:N \l_tmpa_bool }
2520
          {\tl_to_str:n a} {
2521
            \bool_set_false:N \l_tmpa_bool
2522
            \int_incr:N \l_tmpb_int
2523
2524
          {\tl_to_str:n B} {
2525
            \bool_set_false:N \l_tmpa_bool
            \int_incr:N \l_tmpb_int
2527
         }
2528
       }{
2529
          \msg_error:nnxx{stex}{error/wrongargs}{
2530
            \l_stex_current_module_str ?
2531
            \l_stex_symdecl_name_str
2532
          }{##1}
2533
2534
     }
2535
     \bool_if:NTF \l_tmpa_bool {
       % possibly numeric
        \str_if_empty:NTF \l_stex_symdecl_args_str {
2539
          \prop_put:Nnn \l_tmpa_prop { args } {}
2540
          \prop_put:Nnn \l_tmpa_prop { arity } { 0 }
2541
2542
          \int_set:Nn \l_tmpa_int { \l_stex_symdecl_args_str }
2543
          \prop_put:Nnx \l_tmpa_prop { arity } { \int_use:N \l_tmpa_int }
2544
          \str_clear:N \l_tmpa_str
2545
          \int_step_inline:nn \l_tmpa_int {
2546
            \str_put_right:Nn \l_tmpa_str i
          }
2548
          \prop_put:Nnx \l_tmpa_prop { args } { \l_tmpa_str }
2549
       }
2550
     } {
2551
        \prop_put:Nnx \l_tmpa_prop { args } { \l_stex_symdecl_args_str }
2552
        \prop_put:Nnx \l_tmpa_prop { arity }
2553
          { \str_count:N \l_stex_symdecl_args_str }
2554
2555
2556
      \prop_put:Nnx \l_tmpa_prop { assocs } { \int_use:N \l_tmpb_int }
     \tl_if_empty:NTF \l_stex_symdecl_definiens_tl {
        \prop_put:Nnx \l_tmpa_prop { defined }{ false }
     }{
2560
        \prop_put:Nnx \l_tmpa_prop { defined }{ true }
2561
     }
2562
2563
     % argnames
2564
2565
     \clist_clear:N \l_tmpa_clist
2566
2567
     \int_step_inline:nn {\prop_item:Nn \l_tmpa_prop {arity}} {
        \clist_if_empty:NTF \l_stex_symdecl_argnames_clist {
2569
          \clist_put_right:Nn \l_tmpa_clist {##1}
       }{
2570
          \clist_pop:NN \l_stex_symdecl_argnames_clist \l_tmpa_tl
2571
```

```
\exp_args:NNx \clist_put_right:Nn \l_tmpa_clist {\c_dollar_str\l_tmpa_tl}
2572
       }
2573
2574
     \prop_put:Nnx \l_tmpa_prop {argnames} {\clist_use:Nn \l_tmpa_clist ,}
2575
2576
     % semantic macro
2577
2578
     \bool_if:NT \l_stex_symdecl_make_macro_bool {
2579
        \exp_args:Nx \stex_do_up_to_module:n {
          \tl_set:cn { #1 } { \stex_invoke_symbol:n {
2581
            \l_stex_current_module_str ? \l_stex_symdecl_name_str
         }}
2583
       }
2584
     }
2585
2586
     \stex_debug:nn{symbols}{New~symbol:~
2587
        \l_stex_current_module_str ? \l_stex_symdecl_name_str^^J
2588
        Type:~\exp_not:o { \l_stex_symdecl_type_tl }^
        Args:~\prop_item:Nn \l_tmpa_prop { args }^^
       Definiens:~\exp_not:o {\l_stex_symdecl_definiens_tl}
2593
     % circular dependencies require this:
2594
     \stex_if_do_html:T {
2595
        \stex_annotate_invisible:nnn {symdecl} {
2596
          \l_stex_current_module_str ? \l_stex_symdecl_name_str
2597
2598
          \tl_if_empty:NF \l_stex_symdecl_type_tl {
2599
            \stex_annotate_invisible:nnn{type}{}{$\l_stex_symdecl_type_tl$}
2600
          \stex_annotate_invisible:nnn{args}{\prop_item:Nn \l_tmpa_prop { args }}{}
          \stex_annotate_invisible:nnn{macroname}{#1}{}
          \tl_if_empty:NF \l_stex_symdecl_definiens_tl {
2604
            \stex_annotate_invisible:nnn{definiens}{}
2605
              {$\l_stex_symdecl_definiens_tl$}
2606
2607
          \str_if_empty:NF \l_stex_symdecl_assoctype_str {
2608
            \stex_annotate_invisible:nnn{assoctype}{\l_stex_symdecl_assoctype_str}{}
2609
2610
          \str_if_empty:NF \l_stex_symdecl_reorder_str {
            \stex_annotate_invisible:nnn{reorderargs}{\l_stex_symdecl_reorder_str}{}
       }
2614
2615
     \prop_if_exist:cF {
2616
       l_stex_symdecl_
2617
        \l_stex_current_module_str ? \l_stex_symdecl_name_str
2618
        _prop
2619
     } {
2620
        \bool_if:NTF \l_stex_symdecl_local_bool \stex_do_up_to_module:x \stex_execute_in_module:
2621
          \__stex_symdecl_restore_symbol:nnnnnnn
            {\l_stex_symdecl_name_str}
2624
            { \prop_item: Nn \l_tmpa_prop {args} }
            { \prop_item: Nn \l_tmpa_prop {arity} }
2625
```

```
{ \prop_item: Nn \l_tmpa_prop {assocs} }
            { \prop_item: Nn \l_tmpa_prop {defined} }
2627
            {\bool_if:NT \l_stex_symdecl_make_macro_bool {#1} }
2628
            {\l_stex_current_module_str}
2629
            { \prop_item:Nn \l_tmpa_prop {argnames} }
2630
       }
2631
     }
2632
2633
    \cs_new_protected:Nn \__stex_symdecl_restore_symbol:nnnnnnnn {
     \prop_clear:N \l_tmpa_prop
     \prop_put:Nnn \l_tmpa_prop { module } { #7 }
2636
     \prop_put:Nnn \l_tmpa_prop { name } { #1}
2637
     \prop_put:Nnn \l_tmpa_prop { args } {#2}
2638
     \prop_put:Nnn \l_tmpa_prop { arity } { #3 }
2639
     \prop_put:Nnn \l_tmpa_prop { assocs } { #4 }
2640
     \prop_put:Nnn \l_tmpa_prop { defined } { #5 }
2641
      \prop_put:Nnn \l_tmpa_prop { argnames } { #8 }
2642
     \tl_if_empty:nF{#6}{
       \tl_set:cx{#6}{\stex_invoke_symbol:n{\detokenize{#7 ? #1}}}
     \prop_set_eq:cN{l_stex_symdecl_ \detokenize{#7 ? #1} _prop}\l_tmpa_prop
2646
     \seq_clear:c{l_stex_symdecl_ \detokenize{#7 ? #1} _notations}
2647
2648
```

(End definition for \stex_symdecl_do:n. This function is documented on page 86.)

\textsymdecl

```
2649
   \keys_define:nn { stex / textsymdecl } {
2650
              .str_set_x:N = \l__stex_symdecl_name_str ,
     name
2651
                            = \l_stex_symdecl_type_tl
     type
              .tl_set:N
2652
2653
2654
   \cs_new_protected:Nn \_stex_textsymdecl_args:n {
      \str_clear:N \l__stex_symdecl_name_str
      \tl_clear:N \l__stex_symdecl_type_tl
      \clist_clear:N \l_stex_symdecl_argnames_clist
      \keys_set:nn { stex / textsymdecl } { #1 }
2659
2660
2661
   \NewDocumentCommand \textsymdecl {m O{} m} {
2662
      \_stex_textsymdecl_args:n { #2 }
2663
      \str_if_empty:NTF \l__stex_symdecl_name_str {
2664
        \__stex_symdecl_args:n{name=#1,#2}
2665
2666
          _stex_symdecl_args:n{#2}
     }
2668
      \bool_set_true:N \l_stex_symdecl_make_macro_bool
2669
      \stex_symdecl_do:n{#1-sym}
2670
      \stex_execute_in_module:n{
2671
        \cs_set_nopar:cpn{#1name}{
2672
          \ifvmode\hbox_unpack:N\c_empty_box\fi
2673
          \ifmmode\hbox{#3}\else#3\fi\xspace
2674
       }
2675
```

```
\ifmmode\csname#1-sym\expandafter\endcsname\else
                      2677
                                \ifvmode\hbox_unpack:N\c_empty_box\fi
                      2678
                                \symref{#1-sym}{#3}\expandafter\xspace
                      2679
                                \fi
                      2680
                              }
                      2681
                           }
                      2682
                            \stex_execute_in_module:x{
                      2683
                              \__stex_notation_restore_notation:nnnnn
                              {\l_stex_current_module_str?\tl_if_empty:NTF\l_stex_symdecl_name_str{#1}\l_stex_symdec
                              {\exp_not:n{\STEXInternalTermMathOMSiiii{\STEXInternalCurrentSymbolStr}{}{\neginfprec}{
                      2687
                                \comp{\hbox{#3}}\STEXInternalSymbolAfterInvokationTL
                      2688
                              }}}
                      2689
                              {}
                      2690
                      2691
                            \stex_smsmode_do:
                      2692
                      2693 }
                     (End definition for \textsymdecl. This function is documented on page 19.)
\stex_get_symbol:n
                         \str_new:N \l_stex_get_symbol_uri_str
                      2694
                      2695
                          \cs_new_protected:Nn \stex_get_symbol:n {
                            \tl_if_head_eq_catcode:nNTF { #1 } \relax {
                              \tl_set:Nn \l_tmpa_tl { #1 }
                      2698
                              \__stex_symdecl_get_symbol_from_cs:
                      2699
                           }{
                      2700
                              % argument is a string
                              % is it a command name?
                              \cs_if_exist:cTF { #1 }{
                                \cs_set_eq:Nc \l_tmpa_tl { #1 }
                      2704
                                \str_set:Nx \l_tmpa_str { \cs_argument_spec:N \l_tmpa_tl }
                                \str_if_empty:NTF \l_tmpa_str {
                                  \exp_args:Nx \cs_if_eq:NNTF {
                                    \tl_head:N \l_tmpa_tl
                                  } \stex_invoke_symbol:n {
                      2709
                                       _stex_symdecl_get_symbol_from_cs:
                      2710
                                  }{
                                       stex_symdecl_get_symbol_from_string:n { #1 }
                      2713
                                }
                                  {
                      2714
                                     stex_symdecl_get_symbol_from_string:n { #1 }
                      2715
                                }
                      2716
                              }{
                      2718
                                % argument is not a command name
                                  __stex_symdecl_get_symbol_from_string:n { #1 }
                      2719
                                % \l_stex_all_symbols_seq
                            \str_if_eq:eeF {
                              \prop_item:cn {
                      2724
                                l_stex_symdecl_\l_stex_get_symbol_uri_str _prop
                      2725
```

\cs_set_nopar:cpn{#1}{

```
}{ deprecate }
2726
     }{}{
2727
        \msg_warning:nnxx{stex}{warning/deprecated}{
2728
          Symbol~\l_stex_get_symbol_uri_str
2729
2730
          \prop_item:cn {l_stex_symdecl_\l_stex_get_symbol_uri_str _prop}{ deprecate }
2731
        }
     }
2733
2734 }
2735
    \cs_new_protected:Nn \__stex_symdecl_get_symbol_from_string:n {
2736
     \tl_set:Nn \l_tmpa_tl {
        \msg_error:nnn{stex}{error/unknownsymbol}{#1}
2738
2739
      \str_set:Nn \l_tmpa_str { #1 }
2740
2741
     %\int_set:Nn \l_tmpa_int { \str_count:N \l_tmpa_str }
2742
2743
     \str_if_in:NnTF \l_tmpa_str ? {
        \exp_args:NNno \seq_set_split:Nnn \l_tmpa_seq ? \l_tmpa_str
        \seq_pop_right:NN \l_tmpa_seq \l_tmpa_str
        \str_set:Nx \l_tmpb_str {\seq_use:Nn \l_tmpa_seq ?}
2747
     }{
2748
        \str_clear:N \l_tmpb_str
2749
2750
      \str_if_empty:NTF \l_tmpb_str {
        \seq_map_inline: Nn \l_stex_all_modules_seq {
          \seq_map_inline:cn{c_stex_module_##1_constants}{
2753
            \exp_args:Nno \str_if_eq:nnT{####1} \l_tmpa_str {
2754
2755
              \seq_map_break:n{\seq_map_break:n{
2756
                \tl_set:Nn \l_tmpa_tl {
                  \str_set:Nn \l_stex_get_symbol_uri_str { ##1 ? ####1 }
                }
2758
              }}
2759
            }
2760
         }
2761
       }
2762
2763
2764
        \int_set:Nn \l_tmpa_int { \str_count:N \l_tmpb_str }
        \seq_map_inline:Nn \l_stex_all_modules_seq {
          \str_if_eq:eeT{ \l_tmpb_str }{ \str_range:nnn {##1}{-\l_tmpa_int}{-1}}{}
            \seq_map_inline:cn{c_stex_module_##1_constants}{
              \exp_args:Nno \str_if_eq:nnT{####1} \l_tmpa_str {
2768
                \seq_map_break:n{\seq_map_break:n{
2769
                  \tl_set:Nn \l_tmpa_tl {
2770
                     \str_set:Nn \l_stex_get_symbol_uri_str { ##1 ? ####1 }
2771
2772
                }}
2773
              }
2774
2775
            }
2776
         }
2777
       }
     }
2778
2779
```

```
\l_tmpa_tl
2781
2782
    \cs_new_protected:Nn \__stex_symdecl_get_symbol_from_cs: {
2783
      \exp_args:NNx \tl_set:Nn \l_tmpa_tl
2784
        { \tl_tail:N \l_tmpa_tl }
2785
      \tl_if_single:NTF \l_tmpa_tl {
2786
        \exp_args:No \tl_if_head_is_group:nTF \l_tmpa_tl {
2787
          \exp_after:wN \str_set:Nn \exp_after:wN
            \l_stex_get_symbol_uri_str \l_tmpa_tl
2789
        }{
          % TODO
2791
          % tail is not a single group
2792
2793
2794
        % TODO
2795
        % tail is not a single group
2796
2797
2798 }
```

(End definition for \stex_get_symbol:n. This function is documented on page 86.)

29.2 Notations

```
2799 (@@=stex_notation)
                notation arguments:
               \keys_define:nn { stex / notation } {
            2801 % lang
                           .tl_set_x:N = \l__stex_notation_lang_str ,
                                        = \l__stex_notation_variant_str ,
                 variant .tl_set_x:N
            2802
                          .str_set_x:N = \l_stex_notation_prec_str,
                 prec
            2803
                          .tl_set:N
                                        = \l__stex_notation_op_tl ,
            2804
                 oр
                                        = \l_stex_notation_primary_bool ,
                 primary .bool_set:N
            2805
                 primary .default:n
                                        = {true} ,
            2806
                           .str_set_x:N = \l__stex_notation_hints_str,
                                        = \str_set:Nx
                 unknown .code:n
                     \l_stex_notation_variant_str \l_keys_key_str
            2809
            2810 }
            2811
               \cs_new_protected:Nn \_stex_notation_args:n {
            2812
                  \str_clear:N \l__stex_notation_lang_str
            2813 %
                  \str_clear:N \l__stex_notation_variant_str
            2814
                  \str_clear:N \l__stex_notation_prec_str
            2815
                  \str_clear:N \l__stex_notation_hints_str
            2816
                  \tl_clear:N \l__stex_notation_op_tl
                 \bool_set_false:N \l__stex_notation_primary_bool
                 \keys_set:nn { stex / notation } { #1 }
            2820
            2821 }
\notation
            2822 \NewDocumentCommand \notation { s m O{}} {
                 \_stex_notation_args:n { #3 }
                 \tl_clear:N \l_stex_symdecl_definiens_tl
```

```
\tl_set:Nn \l_stex_notation_after_do_tl {
                           2826
                                   \__stex_notation_final:
                           2827
                                   \IfBooleanTF#1{
                           2828
                                     \stex_setnotation:n {\l_stex_get_symbol_uri_str}
                           2829
                           2830
                                   \stex_smsmode_do:\ignorespacesandpars
                           2831
                           2832
                                 \stex_notation_do:nnnnn
                                   { \prop_item:cn {l_stex_symdecl_\l_stex_get_symbol_uri_str _prop } { args } }
                           2834
                                   { \prop_item:cn { l_stex_symdecl_\l_stex_get_symbol_uri_str _prop } { arity } }
                                   { \l_stex_notation_variant_str }
                           2836
                                   { \l_stex_notation_prec_str}
                           2837
                           2838 }
                              \stex_deactivate_macro: Nn \notation {module~environments}
                          (End definition for \notation. This function is documented on page 86.)
\stex_notation_do:nnnnn
                              \seq_new:N \l__stex_notation_precedences_seq
                               \tl_new:N \l__stex_notation_opprec_tl
                               \int_new:N \l__stex_notation_currarg_int
                               \tl_new:N \STEXInternalSymbolAfterInvokationTL
                           2844
                               \cs_new_protected:Nn \stex_notation_do:nnnnn {
                                 \let\STEXInternalCurrentSymbolStr\relax
                           2846
                                 \seq_clear:N \l__stex_notation_precedences_seq
                           2847
                                 \tl_clear:N \l__stex_notation_opprec_tl
                           2848
                                 \str_set:Nx \l__stex_notation_args_str { #1 }
                           2849
                                 \str_set:Nx \l__stex_notation_arity_str { #2 }
                           2850
                                 \str_set:Nx \l__stex_notation_suffix_str { #3 }
                           2851
                                 \str_set:Nx \l__stex_notation_prec_str { #4 }
                           2852
                           2853
                                 % precedences
                                 \str_if_empty:NTF \l__stex_notation_prec_str {
                                   \int_compare:nNnTF \l__stex_notation_arity_str = 0 {
                                     \tl_set:No \l__stex_notation_opprec_tl { \neginfprec }
                                  }{
                           2858
                                     \tl_set:Nn \l__stex_notation_opprec_tl { 0 }
                           2859
                                  }
                           2860
                                } {
                           2861
                                   \str_if_eq:onTF \l__stex_notation_prec_str {nobrackets}{
                           2862
                                     \tl_set:No \l__stex_notation_opprec_tl { \neginfprec }
                           2863
                                     \int_step_inline:nn { \l__stex_notation_arity_str } {
                                       \exp_args:NNo
                                       \seq_put_right:Nn \l__stex_notation_precedences_seq { \infprec }
                                     7
                                  }{
                           2868
                                     \seq_set_split:NnV \l_tmpa_seq ; \l__stex_notation_prec_str
                           2869
                                     \seq_pop_left:NNTF \l_tmpa_seq \l_tmpa_str {
                           2870
                                       \tl_set:No \l__stex_notation_opprec_tl { \l_tmpa_str }
                           2871
                                       \seq_pop_left:NNT \l_tmpa_seq \l_tmpa_str {
                           2872
                                         \exp_args:NNNo \exp_args:NNno \seq_set_split:Nnn
                           2873
                                           \l_tmpa_seq {\tl_to_str:n{x} } { \l_tmpa_str }
```

\stex_get_symbol:n { #2 }

```
\seq_map_inline:Nn \l_tmpa_seq {
2875
                \seq_put_right:Nn \l__stex_notation_precedences_seq { ##1 }
2876
              }
2877
            }
2878
         }{
2879
            \int_compare:nNnTF \l__stex_notation_arity_str = 0 {
2880
              \tl_set:No \l__stex_notation_opprec_tl { \infprec }
2881
            }{
              \tl_set:No \l__stex_notation_opprec_tl { 0 }
            }
         }
       }
2886
     }
2887
2888
      \seq_set_eq:NN \l_tmpa_seq \l__stex_notation_precedences_seq
2889
      \int_step_inline:nn { \l__stex_notation_arity_str } {
2890
        \seq_pop_left:NNF \l_tmpa_seq \l_tmpb_str {
2891
          \exp_args:NNo
2892
          \seq_put_right:No \l__stex_notation_precedences_seq {
            \l_stex_notation_opprec_tl
         }
       }
2896
     }
2897
      \tl_clear:N \l_stex_notation_dummyargs_tl
2898
2899
     \int_compare:nNnTF \l__stex_notation_arity_str = 0 {
2900
2901
        \exp_args:NNe
        \cs_set:Npn \l_stex_notation_macrocode_cs {
2902
          \STEXInternalTermMathOMSiiii { \STEXInternalCurrentSymbolStr }
2903
            { \l_stex_notation_suffix_str }
            { \l_stex_notation_opprec_tl }
            { \exp_not:n { #5 } }
2907
        \l_stex_notation_after_do_tl
2908
2909
        \str_if_in:NnTF \l__stex_notation_args_str b {
2910
          \exp_args:Nne \use:nn
2911
2912
2913
          \cs_generate_from_arg_count:NNnn \l_stex_notation_macrocode_cs
          \cs_set:Npn \l__stex_notation_arity_str } { {
            \STEXInternalTermMathOMBiiii { \STEXInternalCurrentSymbolStr }
              { \l_stex_notation_suffix_str }
2917
              { \l_stex_notation_opprec_tl }
              { \exp_not:n { #5 } }
2918
         }}
2919
       }{
2920
          \str_if_in:NnTF \l__stex_notation_args_str B {
2921
            \exp_args:Nne \use:nn
2922
2923
            \cs_generate_from_arg_count:NNnn \l_stex_notation_macrocode_cs
2924
            \cs_set:Npn \l__stex_notation_arity_str } { {
              \STEXInternalTermMathOMBiiii { \STEXInternalCurrentSymbolStr }
2927
                { \l_stex_notation_suffix_str }
                { \l_stex_notation_opprec_tl }
2928
```

```
}{
                               2931
                                           \exp_args:Nne \use:nn
                               2932
                                           {
                               2933
                                           \cs_generate_from_arg_count:NNnn \l_stex_notation_macrocode_cs
                               2934
                                           \cs_set:Npn \l__stex_notation_arity_str } { {
                               2935
                                             \STEXInternalTermMathOMAiiii { \STEXInternalCurrentSymbolStr }
                                               { \l_stex_notation_suffix_str }
                                               { \l_stex_notation_opprec_tl }
                                               { \exp_not:n { #5 } }
                                           } }
                               2940
                                        }
                               2941
                               2942
                               2943
                                       \str_set_eq:NN \l__stex_notation_remaining_args_str \l__stex_notation_args_str
                               2944
                                       \int_zero:N \l__stex_notation_currarg_int
                               2945
                                       \seq_set_eq:NN \l__stex_notation_remaining_precs_seq \l__stex_notation_precedences_seq
                                       }
                               2948
                               2949 }
                              (End definition for \stex notation do:nnnnn. This function is documented on page ??.)
                              Takes care of annotating the arguments in a notation macro
\__stex_notation_arguments:
                                  \cs_new_protected:Nn \__stex_notation_arguments: {
                                     \int_incr:N \l__stex_notation_currarg_int
                               2951
                                     \str_if_empty:NTF \l__stex_notation_remaining_args_str {
                               2952
                               2953
                                       \l_stex_notation_after_do_tl
                                    }{
                               2954
                                       \str_set:Nx \l_tmpa_str { \str_head:N \l__stex_notation_remaining_args_str }
                               2955
                                       \str_set:Nx \l__stex_notation_remaining_args_str { \str_tail:N \l__stex_notation_remaini
                                       \str_if_eq:VnTF \l_tmpa_str a {
                               2957
                                         \__stex_notation_argument_assoc:nn{a}
                               2958
                                      }{
                               2050
                                         \str_if_eq:VnTF \l_tmpa_str B {
                               2960
                                           \__stex_notation_argument_assoc:nn{B}
                               2961
                                         }{
                               2962
                                           \seq_pop_left:NN \l__stex_notation_remaining_precs_seq \l_tmpb_str
                               2963
                                           \tl_put_right:Nx \l_stex_notation_dummyargs_tl {
                               2964
                                             { \STEXInternalTermMathArgiii
                                               { \l_tmpa_str\int_use:N \l__stex_notation_currarg_int }
                                               { \l_tmpb_str }
                                                 ####\int_use:N \l__stex_notation_currarg_int }
                                             }
                               2970
                                              _stex_notation_arguments:
                               2971
                               2972
                                      }
                               2973
                                    }
                               2974
                               2975 }
```

{ \exp_not:n { #5 } }

} }

2930

 $(End\ definition\ for\ __stex_notation_arguments:.)$

```
\__stex_notation_argument_assoc:nn
```

```
2976 \cs_new_protected:Nn \__stex_notation_argument_assoc:nn {
                           2977
                                 \cs_generate_from_arg_count:NNnn \l_tmpa_cs \cs_set:Npn
                           2978
                                   {\l_stex_notation_arity_str}{
                           2979
                           2980
                           2981
                                 \int_zero:N \l_tmpa_int
                           2982
                                 \tl_clear:N \l_tmpa_tl
                                 \str_map_inline:Nn \l__stex_notation_args_str {
                                   \int_incr:N \l_tmpa_int
                                   \tl_put_right:Nx \l_tmpa_tl {
                           2986
                                     \str_if_eq:nnTF {##1}{a}{ {} }}
                           2987
                                       \str_if_eq:nnTF {##1}{B}{ {} }{
                           2988
                                         {\_stex_term_arg:nn{##1\int_use:N \l_tmpa_int}{########### \int_use:N \l_tmpa
                           2989
                           2990
                                     }
                           2991
                                   }
                           2992
                                 }
                                 \exp_after:wN\exp_after:wN\exp_after:wN \def
                                 \exp_after:wN\exp_after:wN\exp_after:wN \l_tmpa_cs
                                 \exp_after:wN\exp_after:wN\exp_after:wN ##
                           2996
                                 \exp_after:wN\exp_after:wN\exp_after:wN 1
                           2997
                                 \exp_after:wN\exp_after:wN\exp_after:wN ##
                           2998
                                 \exp_after:wN\exp_after:wN\exp_after:wN 2
                           2999
                                 \exp_after:wN\exp_after:wN\exp_after:wN {
                           3000
                                   \exp_after:wN \exp_after:wN \exp_after:wN
                           3001
                                   \exp_not:n \exp_after:wN \exp_after:wN \exp_after:wN {
                           3002
                                     \exp_after:wN \l_tmpa_cs \l_tmpa_tl
                           3003
                                   }
                                 }
                                 \seq_pop_left:NN \l__stex_notation_remaining_precs_seq \l_tmpa_str
                           3007
                                 \tl_put_right:Nx \l_stex_notation_dummyargs_tl { {
                           3008
                                   \STEXInternalTermMathAssocArgiiiii
                           3009
                                     { \int_use:N \l__stex_notation_currarg_int }
                           3010
                                     { \l_tmpa_str }
                           3011
                                     { ####\int_use:N \l__stex_notation_currarg_int }
                           3012
                                     { \l_tmpa_cs {####1} {####2} }
                           3013
                                     {#1}
                                 } }
                                 \__stex_notation_arguments:
                           3017
                          (\mathit{End \ definition \ for \ } \verb|\__stex_notation_argument_assoc:nn.)
\__stex_notation_final:
                          Called after processing all notation arguments
                           3018 \cs_new_protected:Nn \__stex_notation_restore_notation:nnnnn {
                                 \cs_generate_from_arg_count:cNnn{stex_notation_\detokenize{#1} \c_hash_str \detokenize{#2}
                           3019
                                 \cs_set_nopar:Npn {#3}{#4}
                           3020
                                 \t! if_empty:nF {#5}{
                           3021
                                   \tl_set:cn{stex_op_notation_\detokenize{#1} \c_hash_str \detokenize{#2}_cs}{ \comp{ #5 }
                           3022
                                 \seq_if_exist:cT { l_stex_symdecl_\detokenize{#1} _notations }{
```

```
\seq_put_right:cx { 1_stex_symdecl_\detokenize{#1} _notations } { \detokenize{#2} }
3026
3027
3028
    \cs_new_protected:Nn \__stex_notation_final: {
3029
3030
      \stex_execute_in_module:x {
3031
        \__stex_notation_restore_notation:nnnnn
3032
          {\l_stex_get_symbol_uri_str}
          {\l_stex_notation_suffix_str}
3034
3035
          {\l_stex_notation_arity_str}
3036
          ₹
            \exp_after:wN \exp_after:wN \exp_after:wN
3037
            \exp_not:n \exp_after:wN \exp_after:wN \exp_after:wN
3038
            { \exp_after:wN \l_stex_notation_macrocode_cs \l_stex_notation_dummyargs_tl \STEXInt
3039
3040
          {\exp_args:No \exp_not:n \l__stex_notation_op_tl }
3041
3042
     \stex_debug:nn{symbols}{
       Notation~\l_stex_notation_suffix_str
        ~for~\l_stex_get_symbol_uri_str^^J
3046
       Operator~precedence:~\l_stex_notation_opprec_tl^^J
3047
        Argument~precedences:~
3048
          \seq_use:\n \l__stex_notation_precedences_seq {,~}^^J
3049
       Notation: \cs_meaning:c {
3050
          stex_notation_ \l_stex_get_symbol_uri_str \c_hash_str
3051
3052
          \l_stex_notation_suffix_str
3053
          _cs
       }
     }
3055
       % HTML annotations
3056
3057
      \stex_if_do_html:T {
        \stex_annotate_invisible:nnn { notation }
3058
        { \l_stex_get_symbol_uri_str } {
3059
          \stex_annotate_invisible:nnn { notationfragment }
3060
            { \l_stex_notation_suffix_str }{}
3061
          \stex_annotate_invisible:nnn { precedence }
3062
3063
            { \l_stex_notation_prec_str }{}
          \int_zero:N \l_tmpa_int
          \str_set_eq:NN \l__stex_notation_remaining_args_str \l__stex_notation_args_str
3067
          \tl_clear:N \l_tmpa_tl
3068
          \int_step_inline:nn { \l__stex_notation_arity_str }{
            \int_incr:N \l_tmpa_int
3069
            \str_set:Nx \l_tmpb_str { \str_head:N \l__stex_notation_remaining_args_str }
3070
            \str_set:Nx \l__stex_notation_remaining_args_str { \str_tail:N \l__stex_notation_rem
3071
            \str_if_eq:VnTF \l_tmpb_str a {
3072
              \tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {
3073
                \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int a}{} ,
3074
                \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int b}{}
              } }
            }{
3077
              \str_if_eq:VnTF \l_tmpb_str B {
3078
```

```
\stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int a}{} ,
               3080
                                  \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int b}{}
                3081
                                } }
               3082
                             }{
               3083
                                \tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {
               3084
                                  \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int}{}
               3085
                                } }
                3086
                              }
                           }
                3088
                         }
                3089
                          \stex_annotate_invisible:nnn { notationcomp }{}{
               3090
                            \str_set:Nx \STEXInternalCurrentSymbolStr {\l_stex_get_symbol_uri_str }
               3091
                            $ \exp_args:Nno \use:nn { \use:c {
               3092
                              stex_notation_ \STEXInternalCurrentSymbolStr
               3093
                              \c_hash_str \l__stex_notation_suffix_str _cs
                3094
                            } { \l_tmpa_tl } $
               3095
                         }
                3096
                         \tl_if_empty:NF \l__stex_notation_op_tl {
                            \stex_annotate_invisible:nnn { notationopcomp }{}{
                              $\l_stex_notation_op_tl$
               3100
                         }
               3101
                       }
               3102
                     }
               3103
               3104 }
               (End definition for \__stex_notation_final:.)
\setnotation
                   \keys_define:nn { stex / setnotation } {
                     lang
                               .tl_set_x:N = \l__stex_notation_lang_str ,
               3106
                     variant .tl_set_x:N = \l__stex_notation_variant_str ,
               3107
               3108
                     unknown .code:n
                                            = \str_set:Nx
               3109
                         \l_stex_notation_variant_str \l_keys_key_str
               3110
               3111
                   \cs_new_protected:Nn \_stex_setnotation_args:n {
               3112
                    % \str_clear:N \l__stex_notation_lang_str
               3113
                     \str_clear:N \l__stex_notation_variant_str
               3114
                     \keys_set:nn { stex / setnotation } { #1 }
               3115
               3116
               3117
                   \cs_new_protected:Nn \__stex_notation_setnotation:nn {
               3118
                     \seq_if_exist:cT{l_stex_symdecl_#1_notations}{
               3119
                       \seq_remove_all:cn { l_stex_symdecl_#1 _notations }{ #2 }
               3120
               3121
                       \seq_put_left:cn { l_stex_symdecl_#1 _notations }{ #2 }
                     }
               3122
               3123
               3124
                   \cs_new_protected:Nn \stex_setnotation:n {
               3125
                     \exp_args:Nnx \seq_if_in:cnTF { l_stex_symdecl_#1 _notations }
               3126
                       { \l_stex_notation_variant_str }{
               3127
                          \stex_execute_in_module:x{ \__stex_notation_setnotation:nn {#1}{\l__stex_notation_vari
               3128
```

\tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {

```
\stex_debug:nn {notations}{
3129
            Setting~default~notation~
3130
            {\l_stex_notation_variant_str }~for~
3131
            #1 \\
3132
            \expandafter\meaning\csname
3133
            l_stex_symdecl_#1 _notations\endcsname
3134
3135
       }{
3136
          \msg_error:nnxx{stex}{unknownnotation}{\l__stex_notation_variant_str}{#1}
3137
3138
3139 }
3140
   \NewDocumentCommand \setnotation {m m} {
3141
      \stex_get_symbol:n { #1 }
3142
      \_stex_setnotation_args:n { #2 }
3143
      \stex_setnotation:n{\l_stex_get_symbol_uri_str}
3144
      \stex_smsmode_do:\ignorespacesandpars
3145
3146 }
   \cs_new_protected:Nn \stex_copy_notations:nn {
3149
     \stex_debug:nn {notations}{
       Copying~notations~from~#2~to~#1\\
3150
        \seq_use:cn{l_stex_symdecl_#2_notations}{,~}
3151
3152
     \tl_clear:N \l_tmpa_tl
3153
      \int_step_inline:nn { \prop_item:cn {l_stex_symdecl_#2_prop}{ arity } } {
3154
        \tl_put_right:Nn \l_tmpa_tl { {####### ##1} }
3155
3156
      \seq_map_inline:cn {l_stex_symdecl_#2_notations}{\begingroup
3157
3158
        \stex_debug:nn{Here}{Here:~##1}
        \cs_set_eq:Nc \l_tmpa_cs { stex_notation_ #2 \c_hash_str ##1 _cs }
3159
        \edef \l_tmpa_tl {
3160
          \exp_after:wN\exp_after:wN\exp_after:wN \exp_not:n
3161
          \exp_after:wN\exp_after:wN\exp_after:wN {
3162
            \exp_after:wN \l_tmpa_cs \l_tmpa_tl
3163
3164
3165
3166
3167
        \exp_after:wN \def \exp_after:wN \l_tmpa_tl
        \exp_after:wN ####\exp_after:wN 1 \exp_after:wN ####\exp_after:wN 2
        \exp_after:wN { \l_tmpa_tl }
        \edef \l_tmpa_tl {
3171
          \exp_after:wN \exp_not:n \exp_after:wN {
3172
            \l_tmpa_tl {####### 1}{###### 2}
3173
3174
       }
3175
3176
        \stex_debug:nn{Here}{Here:~\expandafter\detokenize\expandafter{\l_tmpa_tl}}
3177
3178
3179
        \stex_execute_in_module:x {
3180
          \__stex_notation_restore_notation:nnnnn
3181
            {#1}{##1}
            { \prop_item:cn {l_stex_symdecl_#2_prop}{ arity } }
3182
```

```
{ \exp_after:wN\exp_not:n\exp_after:wN{\l_tmpa_tl} }
          3183
          3184
                        \cs_if_exist:cT{stex_op_notation_ #2\c_hash_str ##1 _cs}{
          3185
                          \exp_args:NNo\exp_args:No\exp_not:n{\csname stex_op_notation_ #2\c_hash_str ##1
          3186
          3187
                      }
          3188
                  }\endgroup
          3189
          3190
          3191 }
          3192
              \NewDocumentCommand \copynotation {m m} {
          3193
                \stex_get_symbol:n { #1 }
          3194
                \str_set_eq:NN \l_tmpa_str \l_stex_get_symbol_uri_str
          3195
                \stex_get_symbol:n { #2 }
          3196
                \exp_args:Noo
          3197
                \stex_copy_notations:nn \l_tmpa_str \l_stex_get_symbol_uri_str
          3198
                \stex_smsmode_do:\ignorespacesandpars
          3199
          3200 }
         (End definition for \setnotation. This function is documented on page 19.)
\symdef
             \keys_define:nn { stex / symdef } {
               name
                        .str_set_x:N = \l_stex_symdecl_name_str ,
                        .bool_set:N = \l_stex_symdecl_local_bool ,
                local
          3204
                        3205
                args
                                     = \l_stex_symdecl_type_tl ,
                        .tl_set:N
          3206
                type
                                      = \l_stex_symdecl_definiens_tl ,
                def
                        .tl_set:N
          3207
               reorder .str_set_x:N = \l_stex_symdecl_reorder_str ,
          3208
                        .tl_set:N
                                     = \l_stex_notation_op_tl ,
              % lang
                         .str_set_x:N = \l__stex_notation_lang_str ,
          3210
                variant .str_set_x:N = \l__stex_notation_variant_str ,
          3211
          3212
                        .str_set_x:N = \l__stex_notation_prec_str ,
                argnames
                            .clist_set:N = \l_stex_symdecl_argnames_clist ,
          3214
                assoc
                        .choices:nn =
          3215
                    {bin,binl,binr,pre,conj,pwconj}
                    {\str_set:Nx \l_stex_symdecl_assoctype_str {\l_keys_choice_tl}},
          3216
                                     = \str set:Nx
                unknown .code:n
          3217
                    \l_stex_notation_variant_str \l_keys_key_str
          3218
             }
          3219
          3220
              \cs_new_protected:Nn \__stex_notation_symdef_args:n {
          3221
                \str_clear:N \l_stex_symdecl_name_str
          3222
                \str_clear:N \l_stex_symdecl_args_str
          3223
                \str_clear:N \l_stex_symdecl_assoctype_str
          3224
                \str_clear:N \l_stex_symdecl_reorder_str
          3225
                \bool_set_false:N \l_stex_symdecl_local_bool
          3226
                \tl_clear:N \l_stex_symdecl_type_tl
          3227
                \tl_clear:N \l_stex_symdecl_definiens_tl
          3228
                \clist_clear:N \l_stex_symdecl_argnames_clist
          3229
              % \str_clear:N \l__stex_notation_lang_str
          3230
                \str_clear:N \l__stex_notation_variant_str
          3231
```

\str_clear:N \l__stex_notation_prec_str

```
\tl_clear:N \l__stex_notation_op_tl
3234
      \keys_set:nn { stex / symdef } { #1 }
3235
3236
3237
    \NewDocumentCommand \symdef { m O{} } {
3238
      \__stex_notation_symdef_args:n { #2 }
3239
      \bool_set_true:N \l_stex_symdecl_make_macro_bool
3240
      \stex_symdecl_do:n { #1 }
3241
      \tl_set:Nn \l_stex_notation_after_do_tl {
3242
3243
        \__stex_notation_final:
        \stex_smsmode_do:\ignorespacesandpars
3244
3245
      \str_set:Nx \l_stex_get_symbol_uri_str {
3246
        \l_stex_current_module_str ? \l_stex_symdecl_name_str
3247
3248
      \exp_args:Nx \stex_notation_do:nnnnn
3249
       { \prop_item:cn {l_stex_symdecl_\l_stex_get_symbol_uri_str _prop } { args } }
3250
         \prop_item:cn { 1_stex_symdecl_\l_stex_get_symbol_uri_str _prop } { arity } }
        { \l_stex_notation_variant_str }
3253
        { \l_stex_notation_prec_str}
3254
3255 \stex_deactivate_macro:Nn \symdef {module~environments}
```

(End definition for \symdef. This function is documented on page 86.)

29.3 Variables

```
<@@=stex_variables>
3257
   \keys_define:nn { stex / vardef } {
3258
              .str_set_x:N = \l__stex_variables_name_str ,
3259
     name
              .str_set_x:N = \l__stex_variables_args_str ,
3260
     args
              .tl_set:N
                             = \l_stex_variables_type_tl ,
     tvpe
3261
     def
              .tl_set:N
                             = \l_stex_variables_def_tl .
3262
                             = \l_stex_variables_op_tl
              .tl_set:N
     qo
3263
              .str_set_x:N = \l__stex_variables_prec_str
     prec
     reorder .str_set_x:N = \l__stex_variables_reorder_str
     argnames
                  .clist_set:N = \l__stex_variables_argnames_clist ,
3266
     assoc
              .choices:nn
3267
          {bin,binl,binr,pre,conj,pwconj}
3268
          {\str_set:Nx \l_stex_variables_assoctype_str {\l_keys_choice_tl}},
3269
              .choices:nn
3270
          {forall, exists}
3271
          {\str_set:Nx \l_stex_variables_bind_str {\l_keys_choice_tl}}
3272
3273
3274
   \cs_new_protected:Nn \__stex_variables_args:n {
     \str_clear:N \l__stex_variables_name_str
     \str_clear:N \l__stex_variables_args_str
3277
     \str_clear:N \l__stex_variables_prec_str
3278
     \verb|\str_clear:N \l|\_stex_variables_assoctype\_str|
3279
     \str_clear:N \l__stex_variables_reorder_str
3280
     \str_clear:N \l__stex_variables_bind_str
3281
```

```
\tl_clear:N \l__stex_variables_type_tl
     \tl_clear:N \l__stex_variables_def_tl
3283
     \tl_clear:N \l__stex_variables_op_tl
3284
     \clist_clear:N \l__stex_variables_argnames_clist
3285
3286
      \keys_set:nn { stex / vardef } { #1 }
3287
3288
3289
    \NewDocumentCommand \__stex_variables_do_simple:nnn { m O{}} {
      \__stex_variables_args:n {#2}
3291
      \str_if_empty:NT \l__stex_variables_name_str {
3292
       \str_set:Nx \l__stex_variables_name_str { #1 }
3293
3294
      \prop_clear:N \l_tmpa_prop
3295
      \prop_put:Nno \l_tmpa_prop { name } \l__stex_variables_name_str
3296
3297
      \int_zero:N \l_tmpb_int
3298
      \bool_set_true:N \l_tmpa_bool
3299
      \str_map_inline:Nn \l__stex_variables_args_str {
        \token_case_meaning:NnF ##1 {
          0 {} 1 {} 2 {} 3 {} 4 {} 5 {} 6 {} 7 {} 8 {} 9 {}
          {\tl_to_str:n i} { \bool_set_false:N \l_tmpa_bool }
3303
          {\tl_to_str:n b} { \bool_set_false:N \l_tmpa_bool }
3304
          {\tl_to_str:n a} {
3305
            \bool_set_false:N \l_tmpa_bool
3306
            \int_incr:N \l_tmpb_int
3307
3308
          {\tl_to_str:n B} {
3309
            \bool_set_false:N \l_tmpa_bool
3310
3311
            \int_incr:N \l_tmpb_int
          }
3312
       }{
3313
          \msg_error:nnxx{stex}{error/wrongargs}{
3314
            variable~\l_stex_variables_name_str
3315
          }{##1}
3316
       }
3317
3318
3319
      \bool_if:NTF \l_tmpa_bool {
       % possibly numeric
        \str_if_empty:NTF \l__stex_variables_args_str {
          \prop_put:Nnn \l_tmpa_prop { args } {}
          \prop_put:Nnn \l_tmpa_prop { arity } { 0 }
3323
       }{
3324
          \int_set:Nn \l_tmpa_int { \l_stex_variables_args_str }
3325
          \prop_put:Nnx \l_tmpa_prop { arity } { \int_use:N \l_tmpa_int }
3326
          \str_clear:N \l_tmpa_str
3327
          \int_step_inline:nn \l_tmpa_int {
3328
            \str_put_right:Nn \l_tmpa_str i
3329
3330
3331
          \str_set_eq:NN \l__stex_variables_args_str \l_tmpa_str
3332
          \prop_put:Nnx \l_tmpa_prop { args } { \l_stex_variables_args_str }
3333
       }
     } {
3334
        \prop_put:Nnx \l_tmpa_prop { args } { \l_stex_variables_args_str }
3335
```

```
\prop_put:Nnx \l_tmpa_prop { arity }
3336
         { \str_count:N \l__stex_variables_args_str }
3337
3338
     \prop_put:Nnx \l_tmpa_prop { assocs } { \int_use:N \l_tmpb_int }
3339
     \tl_set:cx { #1 }{ \stex_invoke_variable:n { \l__stex_variables_name_str } }
3340
3341
     % argnames
3342
3343
     \clist_clear:N \l_tmpa_clist
3344
     \int_step_inline:nn {\prop_item:Nn \l_tmpa_prop {arity}} {
3345
       \clist_if_empty:NTF \l__stex_variables_argnames_clist {
3346
         \clist_put_right:Nn \l_tmpa_clist {##1}
3347
3348
         \clist_pop:NN \l__stex_variables_argnames_clist \l_tmpa_tl
3349
         \exp_args:NNx \clist_put_right:Nn \l_tmpa_clist {\c_dollar_str\l_tmpa_tl}
3350
3351
3352
     \prop_put:Nnx \l_tmpa_prop {argnames} {\clist_use:Nn \l_tmpa_clist ,}
3353
     \prop_set_eq:cN { l_stex_symdecl_var://\l__stex_variables_name_str _prop} \l_tmpa_prop
3356
3357
     \tl_if_empty:NF \l__stex_variables_op_tl {
3358
       \cs_set:cpx {
3359
         stex_var_op_notation_ \l__stex_variables_name_str _cs
3360
       } { \exp_not:N\comp{ \exp_args:No \exp_not:n { \l_stex_variables_op_tl } } }
3361
     }
3362
3363
     \tl_set:Nn \l_stex_notation_after_do_tl {
3364
       \exp_args:Nne \use:nn {
         \cs_generate_from_arg_count:cNnn { stex_var_notation_\l__stex_variables_name_str _cs }
3366
           \cs_set:Npn { \prop_item:Nn \l_tmpa_prop { arity } }
3367
       } {{
3368
         \exp_after:wN \exp_after:wN \exp_after:wN
3369
         \exp_not:n \exp_after:wN \exp_after:wN \exp_after:wN
3370
         { \exp_after:wN \l_stex_notation_macrocode_cs \l_stex_notation_dummyargs_tl \STEXInter
3371
3372
3373
       \stex_if_do_html:T {
3374
         \stex_annotate_invisible:nnn {vardecl}{\l__stex_variables_name_str}{
           \stex_annotate_invisible:nnn { precedence }
             { \l_stex_variables_prec_str }{}
           \tl_if_empty:NF \l__stex_variables_type_tl {\stex_annotate_invisible:nnn{type}{}}{$\l
           3378
           \stex_annotate_invisible:nnn{macroname}{#1}{}
3379
           \tl_if_empty:NF \l__stex_variables_def_tl {
3380
             \stex_annotate_invisible:nnn{definiens}{}
3381
               {\l_stex_variables_def_tl\}
3382
3383
           \str_if_empty:NF \l__stex_variables_assoctype_str {
3384
              \stex_annotate_invisible:nnn{assoctype}{\l__stex_variables_assoctype_str}{}
3385
           \str_if_empty:NF \l__stex_variables_reorder_str {
3388
              \stex_annotate_invisible:nnn{reorderargs}{\l__stex_variables_reorder_str}{}
3380
```

```
\int_zero:N \l_tmpa_int
            \str_set_eq:NN \l__stex_variables_remaining_args_str \l__stex_variables_args_str
3391
            \tl_clear:N \l_tmpa_tl
3392
            \int_step_inline:nn { \prop_item:Nn \l_tmpa_prop { arity } }{
3393
              \int_incr:N \l_tmpa_int
3394
              \str_set:Nx \l_tmpb_str { \str_head:N \l__stex_variables_remaining_args_str }
3395
              \str_set:Nx \l__stex_variables_remaining_args_str { \str_tail:N \l__stex_variables
3396
              \str_if_eq:VnTF \l_tmpb_str a {
3397
                \tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {
                  \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int a}{} ,
                  \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int b}{}
                } }
3401
              }{
3402
                \str_if_eq:VnTF \l_tmpb_str B {
3403
                  \tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {
3404
                    \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int a}{} ,
3405
                    \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int b}{}
3406
                  } }
                }{
                  \tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {
                    \stex_annotate:nnn{argmarker}{\int_use:N \l_tmpa_int}{}
                  } }
3411
                }
3412
              }
3413
           }
3414
            \stex_annotate_invisible:nnn { notationcomp }{}{
3415
              \str_set:Nx \STEXInternalCurrentSymbolStr {var://\l__stex_variables_name_str }
3416
              $ \exp_args:Nno \use:nn { \use:c {
3417
                stex_var_notation_\l__stex_variables_name_str _cs
3418
              } { \l_tmpa_tl } $
            }
3421
            \tl_if_empty:NF \l__stex_variables_op_tl {
3422
              \stex_annotate_invisible:nnn { notationopcomp }{}{
                $\l_stex_variables_op_tl$
3423
3424
           }
3425
         }
3426
          \str_if_empty:NF \l__stex_variables_bind_str {
3427
3428
            \stex_annotate_invisible:nnn {bindtype}{\l__stex_variables_bind_str,\l__stex_variabl
       }\ignorespacesandpars
     }
3431
3432
     \stex_notation_do:nnnnn { \l__stex_variables_args_str } { \prop_item:Nn \l_tmpa_prop { ari
3433
3434 }
3435
   \cs_new:Nn \_stex_reset:N {
3436
     \tl_if_exist:NTF #1 {
3437
        \def \exp_not:N #1 { \exp_args:No \exp_not:n #1 }
3438
        \let \exp_not:N #1 \exp_not:N \undefined
3441
     }
3442 }
```

```
\NewDocumentCommand \__stex_variables_do_complex:nn { m m }{
      \clist_set:Nx \l__stex_variables_names { \tl_to_str:n {#1} }
3445
      \exp_args:Nnx \use:nn {
3446
        % TODO
3447
        \stex_annotate_invisible:nnn {vardecl}{\clist_use:Nn\l__stex_variables_names,}{
3448
3449
        }
3450
     }{
3451
        \_stex_reset:N \varnot
3452
        \_stex_reset:N \vartype
3453
        \_stex_reset:N \vardefi
3454
     }
3455
3456
3457
    \NewDocumentCommand \vardef { s } {
3458
      \IfBooleanTF#1 {
3459
        \__stex_variables_do_complex:nn
3460
3461
        \__stex_variables_do_simple:nnn
3463
3464 }
3465
   \NewDocumentCommand \svar { O{} m }{
3466
      \tl_if_empty:nTF {#1}{
3467
        \str_set:Nn \l_tmpa_str { #2 }
3468
3469
        \str_set:Nn \l_tmpa_str { #1 }
3470
3471
      \_stex_term_omv:nn {
3472
3473
        var://l_tmpa_str
3474
        \exp_args:Nnx \use:nn {
3475
3476
          \def\comp{\_varcomp}
          \str_set:Nx \STEXInternalCurrentSymbolStr { var://\l_tmpa_str }
3477
          \comp{ #2 }
3478
        }{
3479
          \_stex_reset:N \comp
3480
3481
          \_stex_reset:N \STEXInternalCurrentSymbolStr
3482
     }
   }
3485
3486
3487
   \keys_define:nn { stex / varseq } {
3488
              .str_set_x:N = \l__stex_variables_name_str ,
     name
3489
     args
              .int_set:N
                              = \l_stex_variables_args_int ,
3490
              .tl_set:N
                              = \l_stex_variables_type_tl
      type
3491
              .tl_set:N
                              = \l__stex_variables_mid_tl
3492
3493
              .choices:nn
          {forall, exists}
          {\str_set:Nx \l__stex_variables_bind_str {\l_keys_choice_tl}}
3496
3497
```

```
\cs_new_protected:Nn \__stex_variables_seq_args:n {
     \str_clear:N \l__stex_variables_name_str
3499
     \int_set:Nn \l__stex_variables_args_int 1
3500
     \tl_clear:N \l__stex_variables_type_tl
3501
     \str_clear:N \l__stex_variables_bind_str
3502
3503
      \keys_set:nn { stex / varseq } { #1 }
3504
3505
   \NewDocumentCommand \varseq {m O{} m m m}{
3507
      \__stex_variables_seq_args:n { #2 }
3508
      \str_if_empty:NT \l__stex_variables_name_str {
3509
        \str_set:Nx \l__stex_variables_name_str { #1 }
3510
3511
      \prop_clear:N \l_tmpa_prop
3512
      \prop_put:\nx \l_tmpa_prop { arity }{\int_use:\nabla \l__stex_variables_args_int}
3513
3514
      \seq_set_from_clist:Nn \l_tmpa_seq {#3}
3515
      \int_compare:nNnF {\seq_count:N \l_tmpa_seq} = \l__stex_variables_args_int {
        \msg_error:nnxx{stex}{error/seqlength}
3518
          {\int_use:N \l__stex_variables_args_int}
          {\seq_count:N \l_tmpa_seq}
3510
3520
      \seq_set_from_clist:Nn \l_tmpb_seq {#4}
3521
      \int_compare:nNnF {\seq_count:N \l_tmpb_seq} = \l__stex_variables_args_int {
3522
        \msg_error:nnxx{stex}{error/seqlength}
3523
          {\int_use:N \l__stex_variables_args_int}
3524
          {\seq_count:N \l_tmpb_seq}
3525
3526
3527
      \prop_put:Nnn \l_tmpa_prop {starts} {#3}
3528
      \prop_put:Nnn \l_tmpa_prop {ends} {#4}
3529
      \cs_generate_from_arg_count:cNnn {stex_varseq_\l_stex_variables_name_str _cs}
3530
        \cs_set:Npn {\int_use:N \l__stex_variables_args_int} { #5 }
3531
3532
     % argnames
3533
3534
      \clist_clear:N \l_tmpa_clist
3535
     \int_step_inline:nn {\l__stex_variables_args_int} {
3536
          \clist_put_right:Nn \l_tmpa_clist {##1}
      \prop_put:Nnx \l_tmpa_prop {argnames} {\clist_use:Nn \l_tmpa_clist ,}
3540
3541
3542
3543
      \exp_args:NNo \tl_set:No \l_tmpa_tl {\use:c{stex_varseq_\l__stex_variables_name_str _cs}}
3544
     \int_step_inline:nn \l__stex_variables_args_int {
3545
        \tl_put_right:Nx \l_tmpa_tl { \seq_item:Nn \l_tmpa_seq {##1}} }
3546
3547
     \tl_set:Nx \l_tmpa_tl {\exp_args:NNo \exp_args:No \exp_not:n{\l_tmpa_tl}}
3549
     \tl_put_right:Nn \l_tmpa_tl {,\ellipses,}
3550
      \tl_if_empty:NF \l__stex_variables_mid_tl {
```

\tl_put_right:No \l_tmpa_tl \l__stex_variables_mid_tl

```
\tl_put_right:Nn \l_tmpa_tl {,\ellipses,}
3552
     }
3553
     \exp_args:NNo \tl_set:No \l_tmpb_tl {\use:c{stex_varseq_\l_stex_variables_name_str _cs}}
3554
     \int_step_inline:nn \l__stex_variables_args_int {
3555
        \tl_put_right:Nx \l_tmpb_tl { {\seq_item:Nn \l_tmpb_seq {##1}} }
3556
3557
     \tl_set:Nx \l_tmpb_tl {\exp_args:NNo \exp_args:No \exp_not:n{\l_tmpb_tl}}
3558
     \tl_put_right:No \l_tmpa_tl \l_tmpb_tl
3559
3561
     \prop_put:Nno \l_tmpa_prop { notation }\l_tmpa_tl
3562
3563
     \tl_set:cx {#1} {\stex_invoke_sequence:n {\l_stex_variables_name_str}}
3564
3565
     \exp_args:NNo \tl_set:No \l_tmpa_tl {\use:c{stex_varseq_\l__stex_variables_name_str _cs}}
3566
3567
     \int_step_inline:nn \l__stex_variables_args_int {
3568
        \tl_set:Nx \l_tmpa_tl {\exp_args:No \exp_not:n \l_tmpa_tl {
3569
          \STEXInternalTermMathArgiii{i##1}{0}{\exp_not:n{###}##1}
       }}
3571
     }
3572
3573
     \tl_set:Nx \l_tmpa_tl {
3574
        \STEXInternalTermMathOMAiiii { varseq://\l_stex_variables_name_str}{}{0}{
3575
          \exp_args:NO \exp_args:No \exp_not:n {\l_tmpa_tl}
3576
       }
3577
     }
3578
3579
     \tl_set:No \l_tmpa_tl { \exp_after:wN { \l_tmpa_tl \STEXInternalSymbolAfterInvokationTL} }
3580
3581
3582
     \exp_args:Nno \use:nn {
     \cs_generate_from_arg_count:cNnn {stex_varseq_\l__stex_variables_name_str _cs}
3583
3584
        \cs_set:Npn {\int_use:N \l__stex_variables_args_int}}{\l_tmpa_tl}
3585
     \stex_debug:nn{sequences}{New~Sequence:~
3586
        \expandafter\meaning\csname stex_varseq_\l__stex_variables_name_str _cs\endcsname\\~\\
3587
        \prop_to_keyval:N \l_tmpa_prop
3588
3589
     \prop_set_eq:cN {l_stex_symdecl_varseq://\l__stex_variables_name_str _prop}\l_tmpa_prop
3590
     \stex_if_do_html:T{\stex_annotate_invisible:nnn{varseq}{\l__stex_variables_name_str}{
        \tl_if_empty:NF \l__stex_variables_type_tl {
3594
          \stex_annotate:nnn {type}{}{$\l__stex_variables_type_t1$}
3595
       \stex_annotate:nnn {args}{\int_use:N \l__stex_variables_args_int}{}
3596
        \str_if_empty:NF \l__stex_variables_bind_str {
3597
          \stex_annotate:nnn {bindtype}{\l__stex_variables_bind_str}{}
3598
3599
        \stex_annotate:nnn{startindex}{}{$#3$}
3600
       \stex_annotate:nnn{endindex}{}{$#4$}
3601
        \tl_clear:N \l_tmpa_tl
3604
        \int_step_inline:nn \l__stex_variables_args_int {
          \tl_set:Nx \l_tmpa_tl { \l_tmpa_tl {
3605
```

```
\verb|\stex_annotate:nnn{argmarker}{\#1}{}|
3606
          } }
3607
3608
        \stex_annotate_invisible:nnn { notationcomp }{}{
3609
          \str_set:Nx \STEXInternalCurrentSymbolStr {varseq://\l_stex_variables_name_str }
3610
          $ \exp_args:Nno \use:nn { \use:c {
3611
            stex_varseq_\l__stex_variables_name_str _cs
3612
          } { \l_tmpa_tl } $
3613
        \stex_annotate_invisible:nnn { notationopcomp }{}{
3615
          $ \prop_item:Nn \l_tmpa_prop { notation } $
3616
3617
3618
     }}
3619
3620
     \ignorespacesandpars
3621
3622 }
3623
3624 (/package)
```

Chapter 30

STEX -Terms Implementation

```
3625 (*package)
3626
terms.dtx
                               <@@=stex_terms>
    Warnings and error messages
   \msg_new:nnn{stex}{error/nonotation}{
     Symbol~#1~invoked,~but~has~no~notation#2!
3632 }
3633 \msg_new:nnn{stex}{error/notationarg}{
     Error~in~parsing~notation~#1
3634
3635 }
   \msg_new:nnn{stex}{error/noop}{
3636
     Symbol~#1~has~no~operator~notation~for~notation~#2
3637
3638 }
   \msg_new:nnn{stex}{error/notallowed}{
     Symbol~invokation~#1~not~allowed~in~notation~component~of~#2
   \msg_new:nnn{stex}{error/doubleargument}{
     Argument~#1~of~symbol~#2~already~assigned
3644 }
3645 \msg_new:nnn{stex}{error/overarity}{
     Argument~#1~invalid~for~symbol~#2~with~arity~#3
3647 }
3648
```

30.1 Symbol Invocations

```
\stex_invoke_symbol:n Invokes a semantic macro

3649
3650
3651 \bool_new:N \l_stex_allow_semantic_bool
3652 \bool_set_true:N \l_stex_allow_semantic_bool
3653
```

```
\cs_new_protected:Nn \stex_invoke_symbol:n {
      \ifvmode\indent\fi
3655
      \bool_if:NTF \l_stex_allow_semantic_bool {
3656
        \str_if_eq:eeF {
3657
          \prop_item:cn {
3658
            l_stex_symdecl_#1_prop
3659
          }{ deprecate }
3660
        }{}{
3661
          \msg_warning:nnxx{stex}{warning/deprecated}{
            Symbol~#1
          }{
            \prop_item:cn {l_stex_symdecl_#1_prop}{ deprecate }
3665
          }
3666
        }
3667
        \if_mode_math:
3668
          \exp_after:wN \__stex_terms_invoke_math:n
3669
3670
          \exp_after:wN \__stex_terms_invoke_text:n
3671
        \fi: { #1 }
        \msg_error:nnxx{stex}{error/notallowed}{#1}{\STEXInternalCurrentSymbolStr}
3674
     }
3675
3676 }
3677
    \cs_new_protected:Nn \__stex_terms_invoke_text:n {
3678
      \peek_charcode_remove:NTF ! {
3679
        \__stex_terms_invoke_op_custom:nn {#1}
3680
3681
        \__stex_terms_invoke_custom:nn {#1}
3682
3683
     }
3684 }
3685
    \cs_new_protected:Nn \__stex_terms_invoke_math:n {
3686
      \peek_charcode_remove:NTF ! {
3687
        % operator
3688
        \peek_charcode_remove:NTF * {
3689
          % custom op
3690
3691
          \__stex_terms_invoke_op_custom:nn {#1}
3692
        }{
          % op notation
          \peek_charcode:NTF [ {
             \__stex_terms_invoke_op_notation:nw {#1}
3696
               _stex_terms_invoke_op_notation:nw {#1}[]
3697
3698
       }
3699
     }{
3700
        \peek_charcode_remove:NTF * {
3701
          \__stex_terms_invoke_custom:nn {#1}
3702
3703
          % custom
        }{
3705
          % normal
          \peek_charcode:NTF [ {
3706
            \__stex_terms_invoke_notation:nw {#1}
3707
```

```
}{
3708
               stex_terms_invoke_notation:nw {#1}[]
3709
3710
        }
3711
     }
3712
3713
3714
3715
    \cs_new_protected:Nn \__stex_terms_invoke_op_custom:nn {
3716
      \exp_args:Nnx \use:nn {
3717
        \def\comp{\_comp}
3718
        \str_set:Nn \STEXInternalCurrentSymbolStr { #1 }
3719
        \bool_set_false:N \l_stex_allow_semantic_bool
3720
        \stex_mathml_intent:nn{#1}{
3721
          \_stex_term_oms:nnn {#1}{#1 \c_hash_str CUSTOM-}{
3722
            \comp{ #2 }
3723
3724
        }
3725
     }{
3726
        \_stex_reset:N \comp
        \_stex_reset:N \STEXInternalCurrentSymbolStr
3728
        \bool_set_true:N \l_stex_allow_semantic_bool
3729
     }
3730
3731 }
3732
    \keys_define:nn { stex / terms } {
3733
               .tl_set_x:N = \l_stex_notation_lang_str ,
3734
      variant .tl_set_x:N = \l_stex_notation_variant_str ,
3735
      unknown .code:n
                           = \str_set:Nx
3736
3737
          \l_stex_notation_variant_str \l_keys_key_str
3738 }
3739
3740
    \cs_new_protected:Nn \__stex_terms_args:n {
    % \str_clear:N \l_stex_notation_lang_str
3741
      \str_clear:N \l_stex_notation_variant_str
3742
3743
      \keys_set:nn { stex / terms } { #1 }
3744
3745 }
3746
    \cs_new_protected:Nn \stex_find_notation:nn {
      \_stex_terms_args:n { #2 }
      \seq_if_empty:cTF {
        l_stex_symdecl_ #1 _notations
3750
     } {
3751
        \msg_error:nnxx{stex}{error/nonotation}{#1}{s}
3752
     }
3753
        \str_if_empty:NTF \l_stex_notation_variant_str {
3754
          \seq_get_left:cN {l_stex_symdecl_#1_notations}\l_stex_notation_variant_str
3755
3756
3757
          \seq_if_in:cxTF {l_stex_symdecl_#1_notations}{
            \l_stex_notation_variant_str
3759
          }{
          %
             \str_set:Nx \l_stex_notation_variant_str { \l_stex_notation_variant_str \c_hash_str
3760
          }{
3761
```

```
\msg_error:nnxx{stex}{error/nonotation}{#1}{
               \sim\l_stex_notation_variant_str
3763
3764
         }
3765
       }
3766
     }
3767
3768
3769
    cs_new_protected:Npn \__stex_terms_invoke_op_notation:nw #1 [#2] {
3770
     \exp_args:Nnx \use:nn {
3771
3772
        \def\comp{\_comp}
        \str_set:Nn \STEXInternalCurrentSymbolStr { #1 }
3773
        \stex_find_notation:nn { #1 }{ #2 }
3774
        \bool_set_false:N \l_stex_allow_semantic_bool
3775
        \cs_if_exist:cTF {
3776
          stex_op_notation_ #1 \c_hash_str \l_stex_notation_variant_str _cs
3777
3778
          \_stex_term_oms:nnn { #1 }{
3779
            #1 \c_hash_str \l_stex_notation_variant_str
            \use:c{stex_op_notation_ #1 \c_hash_str \l_stex_notation_variant_str _cs}
         }
3784
          \int_compare:nNnTF {\prop_item:cn {l_stex_symdecl_#1_prop}{arity}} = 0{
3785
            \cs_if_exist:cTF {
3786
              stex_notation_ #1 \c_hash_str \l_stex_notation_variant_str _cs
3787
3788
              \tl_set:Nx \STEXInternalSymbolAfterInvokationTL {
3789
                \_stex_reset:N \comp
3790
                \_stex_reset:N \STEXInternalSymbolAfterInvokationTL
                \_stex_reset:N \STEXInternalCurrentSymbolStr
                \bool_set_true:N \l_stex_allow_semantic_bool
              }
3794
              \def\comp{\_comp}
3795
              \str_set:Nn \STEXInternalCurrentSymbolStr { #1 }
3796
              \bool_set_false: N \l_stex_allow_semantic_bool
3797
              \use:c{stex_notation_ #1 \c_hash_str \l_stex_notation_variant_str _cs}
3798
            }{
3799
              \msg_error:nnxx{stex}{error/nonotation}{#1}{
3800
                ~\l_stex_notation_variant_str
            }
          }{
            \msg_error:nnxx{stex}{error/noop}{#1}{\l_stex_notation_variant_str}
3805
          }
3806
       }
3807
     }{
3808
        \_stex_reset:N \comp
3809
        \_stex_reset:N \STEXInternalCurrentSymbolStr
3810
3811
        \bool_set_true:N \l_stex_allow_semantic_bool
     }
3813 }
3814
   \cs_new_protected:Npn \__stex_terms_invoke_notation:nw #1 [#2] {
```

```
\stex_find_notation:nn { #1 }{ #2 }
3816
     \cs_if_exist:cTF {
3817
       stex_notation_ #1 \c_hash_str \l_stex_notation_variant_str _cs
3818
     }{
3819
        \tl_set:Nx \STEXInternalSymbolAfterInvokationTL {
3820
          \_stex_reset:N \comp
3821
          \_stex_reset:N \STEXInternalSymbolAfterInvokationTL
3822
          \_stex_reset:N \STEXInternalCurrentSymbolStr
3823
          \bool_set_true:N \l_stex_allow_semantic_bool
       }
3825
        \def\comp{\_comp}
        \str_set:Nn \STEXInternalCurrentSymbolStr { #1 }
3827
        \bool_set_false:N \l_stex_allow_semantic_bool
3828
        \use:c{stex_notation_ #1 \c_hash_str \l_stex_notation_variant_str _cs}
3829
3830
        \msg_error:nnxx{stex}{error/nonotation}{#1}{
3831
          ~\l_stex_notation_variant_str
3832
3833
     }
3834
   }
3835
   \prop_new:N \l__stex_terms_custom_args_prop
3837
   \clist_new:N \l_stex_argnames_seq
3838
   \seq_new:N \l_stex_terms_tmp_seq
3839
3840
   cs_new_protected:Nn\__stex_terms_custom_comp:n{\bool_set_false:N \l_stex_allow_semantic_boo
3841
3842
3843
   \cs_new_protected:Nn \__stex_terms_invoke_custom:nn {
      \exp_args:Nnx \use:nn {
3844
        \def\comp{\__stex_terms_custom_comp:n}
3846
        \str_set:Nn \STEXInternalCurrentSymbolStr { #1 }
3847
        \prop_clear:N \l__stex_terms_custom_args_prop
3848
        \prop_put:Nnn \l__stex_terms_custom_args_prop {currnum} {1}
        \prop_get:cnN {
3849
          l_stex_symdecl_#1 _prop
3850
       }{ args } \l_tmpa_str
3851
        \exp_args:NNx \seq_set_from_clist:Nn \l_stex_argnames_seq {
3852
3853
          \prop_item:cn {l_stex_symdecl_#1 _prop}{argnames}
3854
        \prop_put:Nno \l__stex_terms_custom_args_prop {args} \l_tmpa_str
        \tl_set:Nn \arg { \__stex_terms_arg: }
        \str_if_empty:NTF \l_tmpa_str {
3858
          \stex_mathml_intent:nn{#1}{
            \_stex_term_oms:nnn {#1}{#1\c_hash_str CUSTOM-}{\ignorespaces#2}
3859
         }
3860
       }{
3861
          \seq_clear:N \l__stex_terms_tmp_seq
3862
          \exp_args:Nx\int_step_inline:nn{\prop_item:cn{l_stex_symdecl_#1 _prop}{arity}}{
3863
            \tl_set:Nx \l_stex_terms_tmp_tl {\seq_item:Nn \l_stex_argnames_seq {##1}}
3864
            \bool_lazy_or:nnT{
              \str_if_eq_p:nn{a}{\left| str_item:Nn\l_tmpa_str{##1} \right|}
            }{
3868
              \str_if_eq_p:nn{B}{\str_item:Nn\l_tmpa_str{##1}}
            }{
3869
```

```
3870
              \tl_put_right:Nn \l__stex_terms_tmp_tl +
           }
3871
            \seq_put_right:No \l__stex_terms_tmp_seq \l__stex_terms_tmp_tl
3872
3873
         \stex_mathml_intent:nn{
3874
           #1[\prop_item:cn {l_stex_symdecl_#1 _prop}{ args }](
3875
              \seq_use:Nn \l__stex_terms_tmp_seq ,
3876
           )
3877
         }{
            \str_if_in:NnTF \l_tmpa_str b {
              \_stex_term_ombind:nnn {#1}{#1\c_hash_str CUSTOM-\l_tmpa_str}{\ignorespaces#2}
           }{
3881
              \str_if_in:NnTF \l_tmpa_str B {
3882
                \_stex_term_ombind:nnn {#1}{#1\c_hash_str CUSTOM-\l_tmpa_str}{\ignorespaces#2}
3883
3884
                \_stex_term_oma:nnn {#1}{#1\c_hash_str CUSTOM-\l_tmpa_str}{\ignorespaces#2}
3885
              }
3886
         }
       \mbox{\ensuremath{\mbox{\%}}}\xspace TODO check that all arguments exist
     }{
3891
       \_stex_reset:N \l_stex_argnames_seq
3892
       \_stex_reset:N \STEXInternalCurrentSymbolStr
3893
       \_stex_reset:N \arg
3894
       \_stex_reset:N \comp
3895
       \_stex_reset:N \l__stex_terms_custom_args_prop
3896
       %\bool_set_true:N \l_stex_allow_semantic_bool
3897
     }
3898
3899 }
3900
   \NewDocumentCommand \__stex_terms_arg: { s O{} m}{
3901
3902
     \tl_if_empty:nTF {#2}{
       \int_set:Nn \l_tmpa_int {\prop_item:Nn \l__stex_terms_custom_args_prop {currnum}}
3903
       \bool_set_true:N \l_tmpa_bool
3904
       \bool_do_while:Nn \l_tmpa_bool {
3905
          \exp_args:NNx \prop_if_in:NnTF \l__stex_terms_custom_args_prop {\int_use:N \l_tmpa_int
3906
            \int_incr:N \l_tmpa_int
3907
         }{
3908
            \bool_set_false:N \l_tmpa_bool
         }
       }
3911
     }{
3912
       \int_set:Nn \l_tmpa_int { #2 }
3913
     }
3914
     \str_set:Nx \l_tmpa_str {\prop_item:Nn \l__stex_terms_custom_args_prop {args} }
3915
     \int_compare:nNnT \l_tmpa_int > {\str_count:N \l_tmpa_str} {
3916
       \msg_error:nnxxx{stex}{error/overarity}
3917
         {\int_use:N \l_tmpa_int}
3918
         {\STEXInternalCurrentSymbolStr}
3919
         {\str_count:N \l_tmpa_str}
3921
3922
     \str_set:Nx \l_tmpa_str {\str_item:Nn \l_tmpa_str \l_tmpa_int}
     3923
```

```
\bool_lazy_any:nF {
                            3924
                                      {\str_if_eq_p:Vn \l_tmpa_str {a}}
                            3925
                                      {\str_if_eq_p:Vn \l_tmpa_str {B}}
                            3926
                                   }{
                            3927
                                      \msg_error:nnxx{stex}{error/doubleargument}
                            3928
                                        {\int_use:N \l_tmpa_int}
                            3929
                                        {\STEXInternalCurrentSymbolStr}
                            3930
                                   }
                            3931
                                 }
                            3932
                                  \exp_args:NNx \prop_put:Nnn \l__stex_terms_custom_args_prop {\int_use:N \l_tmpa_int} {\igr
                            3933
                                  \bool_if:NTF \l_stex_allow_semantic_bool \use_i:nn {
                            3934
                                    \bool_set_true:N \l_stex_allow_semantic_bool
                            3035
                                    \use:nn
                            3936
                            3937
                                 {
                            3938
                                  \stex_mathml_arg:nn{\seq_item:Nn \l_stex_argnames_seq \l_tmpa_int}{
                            3939
                            3940
                                      \stex_annotate_invisible:n { %TODO
                            3941
                                        \exp_args:No \_stex_term_arg:nn {\l_tmpa_str\int_use:N \l_tmpa_int}{\ignorespaces#3}
                                      }
                                   }{ %TODO
                                      \exp_args:No \_stex_term_arg:nn {\l_tmpa_str\int_use:N \l_tmpa_int}{\ignorespaces#3}
                            3945
                            3946
                                 }}
                            3947
                                 {\bool_set_false:N \l_stex_allow_semantic_bool}
                           3948
                           3949 }
                           3950
                           3951
                               \cs_new_protected:Nn \_stex_term_arg:nn {
                            3952
                                  \bool_set_true:N \l_stex_allow_semantic_bool
                                 \stex_annotate:nnn{ arg }{ #1 }{ #2 }
                            3954
                            3955
                                  \bool_set_false:N \l_stex_allow_semantic_bool
                            3956 }
                            3957
                               \cs_new_protected:Npn \STEXInternalTermMathArgiii #1#2#3 {
                            3958
                                  \exp_args:Nnx \use:nn
                            3959
                                    { \int_set:Nn \l__stex_terms_downprec { #2 }
                            3960
                                      \stex_mathml_arg:nn{\seq_item:Nn \l_stex_argnames_seq \l_tmpa_int}{
                            3961
                                        \_stex_term_arg:nn { #1 }{ #3 }
                                      }
                                   { \int_set:Nn \exp_not:N \l__stex_terms_downprec { \int_use:N \l__stex_terms_downprec }
                           3966 }
                           (End definition for \stex_invoke_symbol:n. This function is documented on page 87.)
\STEXInternalTermMathAssocArgiiiii
                               \cs_new_protected:Npn \STEXInternalTermMathAssocArgiiiii #1#2#3#4#5 {
                           3967
                                  \cs_set:Npn \l_tmpa_cs ##1 ##2 { #4 }
                            3968
                                  \tl_set:Nn \l_tmpb_tl {\STEXInternalTermMathArgiii{#5#1}{#2}}
                            3969
                                  \tl_if_empty:nTF { #3 }{
                           3970
                                    \STEXInternalTermMathArgiii{#5#1}{#2}{}
                            3971
                            3972
```

\exp_args:Nx \tl_if_empty:nTF { \tl_tail:n{ #3 }}{

```
\expandafter\if\expandafter\relax\noexpand#3
3974
            \tl_set:Nn \l_tmpa_tl {\__stex_terms_math_assoc_arg_maybe_sequence:Nnn#3{#1}{#5}}
3975
          \else
3976
            \tl_set:Nn \l_tmpa_tl {\__stex_terms_math_assoc_arg_simple:nnn{#1}{#3}{#5}}
3977
          \fi
3978
          \l_tmpa_tl
3979
       }{
3980
          \_\_stex_terms_math_assoc_arg_simple:nnn{#1}{#3}{#5}
3981
     }
3983
3984
3985
   \cs_new_protected:Nn \__stex_terms_math_assoc_arg_maybe_sequence:Nnn {
3986
      \str_set:Nx \l_tmpa_str { \cs_argument_spec:N #1 }
3987
      \str_if_empty:NTF \l_tmpa_str {
3988
        \exp_args:Nx \cs_if_eq:NNTF {
3989
          \tl_head:N #1
3990
       } \stex_invoke_sequence:n {
3991
          \tl_set:Nx \l_tmpa_tl {\tl_tail:N #1}
          \str_set:Nx \l_tmpa_str {\exp_after:wN \use:n \l_tmpa_tl}
          \tl_set:Nx \l_tmpa_tl {\prop_item:cn {l_stex_symdecl_varseq://\l_tmpa_str _prop}{notat
          \exp_args:NNo \seq_set_from_clist:Nn \l_tmpa_seq \l_tmpa_tl
          \tl_set:Nx \l_tmpa_tl {{\exp_not:N \exp_not:n{
            \exp_not:n{\exp_args:Nnx \use:nn} {
3997
              \exp_not:n {
3998
                 \def\comp{\_varcomp}
3999
                \str_set:Nn \STEXInternalCurrentSymbolStr
4000
              } {varseq://l_tmpa_str}
4001
              \exp_not:n{ ##1 }
4002
            }{
              \exp_not:n {
                 \_stex_reset:N \comp
                 \_stex_reset:N \STEXInternalCurrentSymbolStr
4006
              }
4007
            }
4008
          }}}
4009
          \exp_args:Nno \use:n {\seq_set_map:NNn \l_tmpa_seq \l_tmpa_seq} \l_tmpa_tl
4010
          \seq_reverse:N \l_tmpa_seq
4011
4012
          \space{1} \space{1} tmpa_seq l_tmpa_tl
          \seq_map_inline:Nn \l_tmpa_seq {
            \exp_args:NNo \exp_args:NNo \tl_set:No \l_tmpa_tl {
              \exp_args:Nno
              \l_tmpa_cs { ##1 } \l_tmpa_tl
4016
            }
4017
          }
4018
          \tl_set:Nx \l_tmpa_tl {
4019
            \_stex_term_omv:nn {varseq://\l_tmpa_str}{
4020
              \exp_args:No \exp_not:n \l_tmpa_tl
4021
4022
4023
          }
          \exp_args:No\l_tmpb_tl\l_tmpa_tl
4025
       }{
4026
           __stex_terms_math_assoc_arg_simple:nnn{#2} { #1 }{#3}
4027
```

```
4028
           _stex_terms_math_assoc_arg_simple:nnn{#2} { #1 }{#3}
4029
4030
4031
4032
4033
    \cs_new_protected:Nn \__stex_terms_math_assoc_arg_simple:nnn {
4034
      \clist_set:Nn \l_tmpa_clist{ #2 }
4035
      \int_compare:nNnTF { \clist_count:N \l_tmpa_clist } < 2 {</pre>
4036
4037
        \tl_set:Nn \l_tmpa_tl {
          \stex_mathml_arg:nn{\seq_item:Nn \l_stex_argnames_seq #1}{
4038
            \_stex_term_arg:nn{A#3#1}{ #2 } }
4039
4040
     }{
4041
        \clist_reverse:N \l_tmpa_clist
4042
        \clist_pop:NN \l_tmpa_clist \l_tmpa_tl
4043
        \tl_set:Nx \l_tmpa_tl {
4044
          \stex_mathml_arg:nn{\seq_item:Nn \l_stex_argnames_seq #1}{
4045
            \stex_term_arg:nn{A#3#1}{
            \exp_args:No \exp_not:n \l_tmpa_tl
          }
4048
        }}
4049
        \clist_map_inline:Nn \l_tmpa_clist {
4050
          \exp_args:NNo \exp_args:NNo \tl_set:No \l_tmpa_tl {
4051
            \exp_args:Nno
4052
            \l_tmpa_cs {
4053
              \stex_mathml_arg:nn{\seq_item:Nn \l_stex_argnames_seq #1}{
4054
                 \_stex_term_arg:nn{A#3#1}{##1}
4055
              }
4056
4057
            } \l_tmpa_tl
4058
4059
        }
     }
4060
      \exp_args:No\l_tmpb_tl\l_tmpa_tl
4061
4062 }
```

 $(\textit{End definition for } \verb|\STEXInternalTermMathAssocArgiiiii. This function is documented on page \verb|\>88.)|$

30.2 Terms

Precedences:

```
\infprec
\neginfprec
\neginfprec
\lambda \t1_const:Nx \infprec {\int_use:N \c_max_int}

\lambda \t1_const:Nx \neginfprec {-\int_use:N \c_max_int}

\lambd
```

```
(End\ definition\ for\ \verb|\l_stex_terms_left_bracket_str|\ and\ \verb|\l_stex_terms_right_bracket_str|)
\ stex terms maybe brackets:nn
                         Compares precedences and insert brackets accordingly
                             \cs_new_protected:Nn \__stex_terms_maybe_brackets:nn {
                               \bool_if:NTF \l__stex_terms_brackets_done_bool {
                         4070
                                  \bool_set_false:N \l__stex_terms_brackets_done_bool
                         4071
                                  #2
                         4072
                               } {
                          4073
                                  \int_compare:nNnTF { #1 } > \l__stex_terms_downprec {
                         4074
                                    \bool_if:NTF \l_stex_inparray_bool { #2 }{
                          4075
                                      \stex_debug:nn{dobrackets}{\number#1 > \number\l__stex_terms_downprec; \detokenize{#
                          4076
                                      \dobrackets { #2 }
                                 }{ #2 }
                          4079
                               }
                         4080
                         4081 }
                         (End\ definition\ for\ \_\_stex\_terms\_maybe\_brackets:nn.)
          \dobrackets
                         4082 \bool_new:N \l__stex_terms_brackets_done_bool
                             %\RequirePackage{scalerel}
                             \cs_new_protected:Npn \dobrackets #1 {
                         4084
                               \ThisStyle{\if D\moswitch}
                         4085
                                     \exp_args:Nnx \use:nn
                          4086
                                     { \exp_after:wN \left\l__stex_terms_left_bracket_str #1 }
                          4087
                               %
                                     { \exp_not:N\right\l__stex_terms_right_bracket_str }
                          4088
                               %
                                   \else
                                    \exp_args:Nnx \use:nn
                          4091
                                      \bool_set_true:N \l__stex_terms_brackets_done_bool
                          4092
                                      \int_set:Nn \l__stex_terms_downprec \infprec
                          4093
                                      \l__stex_terms_left_bracket_str
                         4094
                                      #1
                         4095
                         4096
                          4097
                                      \bool_set_false:N \l__stex_terms_brackets_done_bool
                          4098
                                      \l_stex_terms_right_bracket_str
                          4099
                                      \int_set:Nn \l__stex_terms_downprec { \int_use:N \l__stex_terms_downprec }
                         4101
                               %\fi}
                         4102
                         4103 }
                         (End definition for \dobrackets. This function is documented on page 88.)
        \withbrackets
                             \cs_new_protected:Npn \withbrackets #1 #2 #3 {
                               \exp_args:Nnx \use:nn
                         4105
                               {
                         4106
                                  \tl_set:Nx \l__stex_terms_left_bracket_str { #1 }
                         4107
                                  \tl_set:Nx \l__stex_terms_right_bracket_str { #2 }
                         4108
                                  #3
                         4109
                         4110
                               }
```

4111

{

```
\tl_set:Nn \exp_not:N \l__stex_terms_left_bracket_str
                                 4112
                                            {\l_stex_terms_left_bracket_str}
                                 4113
                                          \tl_set:Nn \exp_not:N \l__stex_terms_right_bracket_str
                                 4114
                                            {\l_stex_terms_right_bracket_str}
                                 4115
                                 4116
                                 4117 }
                                 (End definition for \withbrackets. This function is documented on page 88.)
               \STEXinvisible
                                 4118 \cs_new_protected:Npn \STEXinvisible #1 {
                                       \stex_annotate_invisible:n { #1 }
                                 4120 }
                                 (End definition for \STEXinvisible. This function is documented on page 88.)
                                     OMDoc terms:
\STEXInternalTermMathOMSiiii
                                     \cs_new_protected:Nn \_stex_term_oms:nnn {
                                       \stex_annotate:nnn{ OMID }{ #2 }{
                                         #3
                                       }
                                 4124
                                 4125 }
                                 4126
                                     \cs_new_protected:Npn \STEXInternalTermMathOMSiiii #1#2#3#4 {
                                 4127
                                       \__stex_terms_maybe_brackets:nn { #3 }{
                                 4128
                                         \stex_mathml_intent:nn{#1} {
                                 4129
                                            \_stex_term_oms:nnn { #1 } { #1\c_hash_str#2 } { #4 }
                                 4130
                                 4131
                                 4132
                                       }
                                 4133 }
                                 (End definition for \STEXInternalTermMathOMSiiii. This function is documented on page 87.)
     \_stex_term_math_omv:nn
                                 4134 \cs_new_protected:Nn \_stex_term_omv:nn {
                                       \stex_annotate:nnn{ OMV }{ #1 }{
                                 4136
                                         #2
                                 4137
                                 4138 }
                                 (End definition for \_stex_term_math_omv:nn. This function is documented on page ??.)
\STEXInternalTermMathOMAiiii
                                     \cs_new_protected:Nn \_stex_term_oma:nnn {
                                       \stex_annotate:nnn{ OMA }{ #2 }{
                                 4140
                                 4141
                                 4143 }
                                 4144
                                 4145 \cs_new_protected:Npn \STEXInternalTermMathOMAiiii #1#2#3#4 {
                                       \exp_args:Nnx \use:nn {
                                 4146
                                          \seq_clear:N \l__stex_terms_tmp_seq
                                 4147
                                          \prop_if_exist:cT{l_stex_symdecl_#1 _prop}{
                                 4148
                                          \exp_args:NNx \seq_set_from_clist:Nn \l_stex_argnames_seq {
                                 4149
```

```
\prop_item:cn {l_stex_symdecl_#1 _prop}{argnames}
4150
        }
4151
        \exp_args:Nx\int_step_inline:nn{\prop_item:cn{l_stex_symdecl_#1 _prop}{arity}}{
4152
           \tl_set:Nx \l_stex_terms_tmp_tl {\seq_item:Nn \l_stex_argnames_seq {##1}}
4153
           \bool_lazy_or:nnT{
4154
             \str_if_eq_p:nn{a}{\str_item:Nn\l_tmpa_str{##1}}
4155
          }{
4156
             \str_if_eq_p:nn{B}{\str_item:Nn\l_tmpa_str{##1}}
4157
          }{
             \tl_put_right:Nn \l__stex_terms_tmp_tl +
4159
          }
4160
           \seq_put_right:No \l__stex_terms_tmp_seq \l__stex_terms_tmp_tl
4161
4162
      }
4163
        _stex_terms_maybe_brackets:nn { #3 }{
4164
        \stex_mathml_intent:nn{
4165
           #1[\prop_item:cn {l_stex_symdecl_#1 _prop}{ args }](
4166
             \seq_use: Nn \l__stex_terms_tmp_seq ,
4167
           \_stex_term_oma:nnn { #1 } { #1\c_hash_str#2 } { #4 }
4171
      }
4172
      }{
4173
          _{	t stex\_reset:N \l_stex\_argnames\_seq}
4174
4175
4176 }
(End definition for \STEXInternalTermMathOMAiiii. This function is documented on page 87.)
```

\STEXInternalTermMathOMBiiii

```
\cs_new_protected:Nn \_stex_term_ombind:nnn {
4177
      \stex_annotate:nnn{ OMBIND }{ #2 }{
4178
4179
       #3
4180
     }
4181
4182
   cs_new_protected:Npn \STEXInternalTermMathOMBiiii #1#2#3#4 {
4183
     \exp_args:Nnx \use:nn {
4184
        \seq_clear:N \l__stex_terms_tmp_seq
4185
        \prop_if_exist:cT{l_stex_symdecl_#1 _prop}{
4186
        \exp_args:NNx \seq_set_from_clist:Nn \l_stex_argnames_seq {
4187
          \prop_item:cn {l_stex_symdecl_#1 _prop}{argnames}
4188
4189
        \exp_args:Nx\int_step_inline:nn{\prop_item:cn{l_stex_symdecl_#1 _prop}{arity}}{
4190
          \tl_set:Nx \l__stex_terms_tmp_tl {\seq_item:Nn \l_stex_argnames_seq {##1}}
          \bool_lazy_or:nnT{
4192
            \str_if_eq_p:nn{a}{\str_item:Nn\l_tmpa_str{##1}}
4193
4194
          }{
            \str_if_eq_p:nn{B}{\str_item:Nn\l_tmpa_str{##1}}
4195
          }{
4196
            \tl_put_right:Nn \l__stex_terms_tmp_tl +
4197
4198
          \seq_put_right:No \l__stex_terms_tmp_seq \l__stex_terms_tmp_tl
4199
```

```
}
           4200
           4201
                    _stex_terms_maybe_brackets:nn { #3 }{
           4202
                    \stex_mathml_intent:nn{
           4203
                      #1[\prop_item:cn {l_stex_symdecl_#1 _prop}{ args }](
           4204
                        \seq_use: Nn \l__stex_terms_tmp_seq ,
           4205
           4206
                   }{
           4207
                      _stex_term_ombind:nnn { #1 } { #1\c_hash_str#2 } { #4 }
           4209
                 }
           4210
                 }{
           4211
                     _stex_reset:N \l_stex_argnames_seq
           4212
                 }
           4213
           4214 }
           (End definition for \STEXInternalTermMathOMBiiii. This function is documented on page 87.)
 \symref
\symname
               \cs_new:Nn \stex_capitalize:n { \uppercase{#1} }
           4215
           4216
               \keys_define:nn { stex / symname } {
           4217
                          .tl_set_x:N
                                           = \l_stex_terms_pre_tl ,
           4218
                          .tl_set_x:N
                                           = \l_stex_terms_post_tl ,
                 root
                          .tl_set_x:N
                                           = \l_stex_terms_root_tl
           4221 }
           4222
               \cs_new_protected:Nn \stex_symname_args:n {
           4223
                 \tl_clear:N \l__stex_terms_post_tl
           4224
                 \tl_clear:N \l__stex_terms_pre_tl
           4225
                 \tl_clear:N \l__stex_terms_root_str
           4226
                 \keys_set:nn { stex / symname } { #1 }
           4227
           4228
           4229
               \NewDocumentCommand \symref { m m }{
                 \let\compemph_uri_prev:\compemph@uri
           4231
                 \let\compemph@uri\symrefemph@uri
                 \STEXsymbol{#1}!{ #2 }
           4233
                 \let\compemph@uri\compemph_uri_prev:
           4234
           4235
           4236
               \NewDocumentCommand \synonym { O{} m m}{
           4237
                 \stex_symname_args:n { #1 }
           4238
                 \let\compemph_uri_prev:\compemph@uri
           4239
                 \let\compemph@uri\symrefemph@uri
           4240
                 % TODO
           4241
           4242
                 \STEXsymbol{#2}!{\l__stex_terms_pre_tl #3 \l__stex_terms_post_tl}
                 \let\compemph@uri\compemph_uri_prev:
           4243
           4244 }
           4245
               \NewDocumentCommand \symname { O{} m }{
           4246
                 \stex_symname_args:n { #1 }
           4247
                 \stex_get_symbol:n { #2 }
           4248
                 \str_set:Nx \l_tmpa_str {
```

```
\prop_item:cn { l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop } { name }
4250
                }
4251
                 \exp_args:NNno \str_replace_all:Nnn \l_tmpa_str {-} {~}
4252
4253
                 \let\compemph_uri_prev:\compemph@uri
4254
                 \let\compemph@uri\symrefemph@uri
4255
                 \exp_args:NNx \use:nn
4256
                 \stex_invoke_symbol:n { { \l_stex_get_symbol_uri_str }!\ifmmode*\fi{
4257
                       \l_stex_terms_pre_tl \l_tmpa_str \l_stex_terms_post_tl
                   } }
4259
                 \let\compemph@uri\compemph_uri_prev:
4260
4261
4262
           \NewDocumentCommand \Symname { O{} m }{
4263
                 \stex_symname_args:n { #1 }
4264
                 \stex_get_symbol:n { #2 }
4265
                 \str_set:Nx \l_tmpa_str {
 4266
                       \prop_item:cn { l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop } { name }
                 \exp_args:NNno \str_replace_all:Nnn \l_tmpa_str {-} {~}
 4270
                 \let\compemph_uri_prev:\compemph@uri
                 \let\compemph@uri\symrefemph@uri
4271
                 \exp_args:NNx \use:nn
4272
                 \stex_invoke_symbol:n { { \l_stex_get_symbol_uri_str }!\ifmmode*\fi{
4273
                       \exp_after:wN \stex_capitalize:n \l_tmpa_str
4274
                              \label{local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_loc
4275
                   } }
4276
                 \let\compemph@uri\compemph_uri_prev:
4277
4278 }
```

(End definition for \symmes and \symmame. These functions are documented on page 87.)

30.3 Notation Components

```
4279 \langle @@=stex_notationcomps \rangle
          \comp
  \compemph@uri
                   4280 \cs_new_protected:Npn \_comp #1 {
      \compemph
                         \str_if_empty:NF \STEXInternalCurrentSymbolStr {
                   4281
                           \stex_html_backend:TF {
       \defemph
                   4282
                             \stex_annotate:nnn { comp }{ \STEXInternalCurrentSymbolStr }{ #1 }
   \defemph@uri
                   4283
    \symrefemph
                   4284
                             \exp_args:Nnx \compemph@uri { #1 } { \STEXInternalCurrentSymbolStr }
                   4285
\symrefemph@uri
                           }
                   4286
       \varemph
                         }
   \varemph@uri
                   4288 }
                      \cs_new_protected:Npn \_varcomp #1 {
                         \str_if_empty:NF \STEXInternalCurrentSymbolStr {
                   4291
                           \stex_html_backend:TF {
                   4292
                             \stex_annotate:nnn { varcomp }{ \STEXInternalCurrentSymbolStr }{ #1 }
                   4293
                   4294
                             \exp_args:Nnx \varemph@uri { #1 } { \STEXInternalCurrentSymbolStr }
                   4295
                   4296
```

```
4298
                4299
                    \def\comp{\_comp}
                4300
                4301
                    \cs_new_protected:Npn \compemph@uri #1 #2 {
                4302
                         \compemph{ #1 }
                4303
                4304
                4305
                4306
                    \cs_new_protected:Npn \compemph #1 {
                4307
                         #1
                4308
                4309 }
                4310
                    \cs_new_protected:Npn \defemph@uri #1 #2 {
                4311
                         \defemph{#1}
                4312
                4313 }
                4314
                    \cs_new_protected:Npn \defemph #1 {
                4315
                         \textbf{#1}
                4316
                4317 }
                4318
                    \cs_new_protected:Npn \symrefemph@uri #1 #2 {
                4319
                         \symrefemph{#1}
                4320
                4321 }
                4322
                    \cs_new_protected:Npn \symrefemph #1 {
                4323
                         \emph{#1}
                4324
                4325 }
                    \cs_new_protected:Npn \varemph@uri #1 #2 {
                4327
                         \varemph{#1}
                4328
                4329 }
                4330
                    \cs_new_protected:Npn \varemph #1 {
                4331
                4332
                4333 }
                (End definition for \comp and others. These functions are documented on page 88.)
   \ellipses
                4334 \NewDocumentCommand \ellipses {} { \ldots }
                (End definition for \ellipses. This function is documented on page 88.)
     \parray
   \prmatrix
                    \bool_new:N \l_stex_inparray_bool
 \parrayline
                    \bool_set_false:N \l_stex_inparray_bool
\parraylineh
                    \NewDocumentCommand \parray { m m } {
 \parraycell
                      \begingroup
                      \bool_set_true:N \l_stex_inparray_bool
                4339
                      \begin{array}{#1}
                4340
                        #2
                4341
                      \end{array}
                4342
                      \endgroup
                4343
```

}

```
4344 }
4345
    \NewDocumentCommand \prmatrix { m } {
4346
      \begingroup
4347
      \bool_set_true:N \l_stex_inparray_bool
4348
      \begin{matrix}
4349
        #1
4350
      \end{matrix}
4351
      \endgroup
4353 }
4354
    \def \maybephline {
4355
      \bool_if:NT \l_stex_inparray_bool {\hline}
4356
4357 }
4358
    \def \parrayline #1 #2 {
4359
      #1 #2 \bool_if:NT \l_stex_inparray_bool {\\}
4360
4361
    \def \pmrow #1 { \parrayline{}{ #1 } }
    \def \parraylineh #1 #2 {
      #1 #2 \bool_if:NT \l_stex_inparray_bool {\\hline}
4366
4367 }
4368
    \def \parraycell #1 {
      #1 \bool_if:NT \l_stex_inparray_bool {&}
4371 }
(End definition for \parray and others. These functions are documented on page ??.)
```

30.4 Variables

```
4372 (@@=stex_variables)
\stex_invoke_variable:n
                           Invokes a variable
                            4373 \cs_new_protected:Nn \stex_invoke_variable:n {
                            4374
                                  \if_mode_math:
                                    \exp_after:wN \__stex_variables_invoke_math:n
                            4375
                            4376
                                    \exp_after:wN \__stex_variables_invoke_text:n
                            4377
                                  \fi: {#1}
                            4378
                            4379 }
                            4380
                                \cs_new_protected:Nn \__stex_variables_invoke_text:n {
                            4381
                                  \peek_charcode_remove:NTF ! {
                            4382
                                    \__stex_variables_invoke_op_custom:nn {#1}
                                    \__stex_variables_invoke_custom:nn {#1}
                                 }
                            4386
                            4387 }
                            4388
                            4389
                            4390 \cs_new_protected:Nn \__stex_variables_invoke_math:n {
```

```
\peek_charcode_remove:NTF ! {
4391
        \peek_charcode_remove:NTF ! {
4392
          \peek_charcode:NTF [ {
4393
            % TODO throw error
4394
4395
               _stex_variables_invoke_op_custom:nn
4396
4397
       }{
4398
             _stex_variables_invoke_op:n { #1 }
       }
4400
4401
     }{
        \peek_charcode_remove:NTF * {
4402
          \__stex_variables_invoke_custom:nn { #1 }
4403
4404
          \__stex_variables_invoke_math_ii:n { #1 }
4405
4406
4407
4408
   \cs_new_protected:Nn \__stex_variables_invoke_op_custom:nn {
4411
      \exp_args:Nnx \use:nn {
        \def\comp{\_varcomp}
4412
        \str_set:Nn \STEXInternalCurrentSymbolStr { var://#1 }
4413
        \bool_set_false:N \l_stex_allow_semantic_bool
4414
        \_stex_term_omv:nn {var://#1}{
4415
          \comp{ #2 }
4416
       }
4417
     }{
4418
        \_stex_reset:N \comp
4419
        \_stex_reset:N \STEXInternalCurrentSymbolStr
        \bool_set_true:N \l_stex_allow_semantic_bool
4421
     }
4422
4423 }
4424
   \cs_new_protected:Nn \__stex_variables_invoke_op:n {
4425
      \cs_if_exist:cTF {
4426
        stex_var_op_notation_ #1 _cs
4427
4428
4429
        \exp_args:Nnx \use:nn {
          \def\comp{\_varcomp}
          \str_set:Nn \STEXInternalCurrentSymbolStr { var://#1 }
          \_stex_term_omv:nn { var://#1 }{
4433
            \use:c{stex_var_op_notation_ #1 _cs }
          }
4434
       }{
4435
          \_stex_reset:N \comp
4436
          \_stex_reset:N \STEXInternalCurrentSymbolStr
4437
       }
4438
     }{
4439
4440
        \int_compare:nNnTF {\prop_item:cn {l_stex_symdecl_var://#1_prop}{arity}} = 0{
          \__stex_variables_invoke_math_ii:n {#1}
       }{
4442
          \msg_error:nnxx{stex}{error/noop}{variable~#1}{}
4443
        }
4444
```

```
}
4445
4446
4447
    \cs_new_protected:Npn \__stex_variables_invoke_math_ii:n #1 {
4448
      \cs_if_exist:cTF {
4449
       stex_var_notation_#1_cs
4450
4451
        \tl_set:Nx \STEXInternalSymbolAfterInvokationTL {
          \_stex_reset:N \comp
          \_stex_reset:N \STEXInternalSymbolAfterInvokationTL
4454
          \_stex_reset:N \STEXInternalCurrentSymbolStr
4455
          \bool_set_true:N \l_stex_allow_semantic_bool
4456
4457
        \def\comp{\_varcomp}
4458
        \str_set:Nn \STEXInternalCurrentSymbolStr { var://#1 }
4459
        \bool_set_false:N \l_stex_allow_semantic_bool
4460
        \use:c{stex_var_notation_#1_cs}
4461
4462
        \msg_error:nnxx{stex}{error/nonotation}{variable~#1}{s}
4465 }
4466
   \cs_new_protected:Nn \__stex_variables_invoke_custom:nn {
4467
      \exp_args:Nnx \use:nn {
4468
        \def\comp{\_varcomp}
4469
        \str_set:Nn \STEXInternalCurrentSymbolStr { var://#1 }
4470
        \prop_clear:N \l__stex_terms_custom_args_prop
4471
        \prop_put:Nnn \l__stex_terms_custom_args_prop {currnum} {1}
4472
        \prop_get:cnN {
4473
          l_stex_symdecl_var://#1 _prop
4475
       }{ args } \l_tmpa_str
        \prop_put:Nno \l__stex_terms_custom_args_prop {args} \l_tmpa_str
4476
        \tl_set:Nn \arg { \__stex_terms_arg: }
4477
        \str_if_empty:NTF \l_tmpa_str {
4478
          \_stex_term_omv:nn {var://#1}{\ignorespaces#2}
4479
       }{
4480
          \str_if_in:NnTF \l_tmpa_str b {
4481
            \_stex_term_ombind:nnn {var://#1}{}\ignorespaces#2}
4482
4483
            \str_if_in:NnTF \l_tmpa_str B {
              \_stex_term_ombind:nnn {var://#1}{}\ignorespaces#2}
            }{
4487
              \_stex_term_oma:nnn {var://#1}{}{\ignorespaces#2}
4488
         }
4489
4490
       % TODO check that all arguments exist
4491
4492
        \_stex_reset:N \STEXInternalCurrentSymbolStr
4493
        \_stex_reset:N \arg
        \_stex_reset:N \comp
        \_stex_reset:N \l__stex_terms_custom_args_prop
4497
       %\bool_set_true:N \l_stex_allow_semantic_bool
     }
4498
```

```
4499 }
```

(End definition for \stex_invoke_variable:n. This function is documented on page ??.)

30.5 Sequences

```
<@0=stex_sequences>
4500
4501
   \cs_new_protected: Nn \stex_invoke_sequence:n {
4502
      \peek_charcode_remove:NTF ! {
4503
        \_stex_term_omv:nn {varseq://#1}{
          \exp_args:Nnx \use:nn {
            \def\comp{\_varcomp}
            \str_set:Nn \STEXInternalCurrentSymbolStr {varseq://#1}
4507
            \prop_item:cn{l_stex_symdecl_varseq://#1_prop}{notation}
4508
          }{
4509
            \_stex_reset:N \comp
4510
            \_stex_reset:N \STEXInternalCurrentSymbolStr
4511
4512
       }
4513
4514
        \bool_set_false:N \l_stex_allow_semantic_bool
        \def\comp{\_varcomp}
        \str_set:Nn \STEXInternalCurrentSymbolStr {varseq://#1}
4517
        \tl_set:Nx \STEXInternalSymbolAfterInvokationTL {
4518
          \_stex_reset:N \comp
4519
          \_stex_reset:N \STEXInternalSymbolAfterInvokationTL
4520
          \_stex_reset:N \STEXInternalCurrentSymbolStr
4521
          \bool_set_true:N \l_stex_allow_semantic_bool
4522
4523
        \use:c { stex_varseq_#1_cs }
     }
4526 }
4527  /package
```

Chapter 31

STEX -Structural Features Implementation

```
4528 (*package)
                                  features.dtx
    Warnings and error messages
4532 \msg_new:nnn{stex}{error/copymodule/notallowed}{
     Symbol~#1~can~not~be~assigned~in~copymodule~#2
4534 }
   \msg_new:nnn{stex}{error/interpretmodule/nodefiniens}{
4535
     Symbol~#1~not~assigned~in~interpretmodule~#2
4536
4537 }
4538
   \msg_new:nnn{stex}{error/unknownstructure}{
     No~structure~#1~found!
4543 \msg_new:nnn{stex}{error/unknownfield}{
     No~field~#1~in~instance~#2~found!\\#3
4544
4545 }
4546
4547 \msg_new:nnn{stex}{error/keyval}{
     Invalid~key=value~pair:#1
4548
4550 \msg_new:nnn{stex}{error/instantiate/missing}{
     Assignments~missing~in~instantiate:~#1
4553 \msg_new:nnn{stex}{error/incompatible}{
     Incompatible~signature:~#1~(#2)~and~#3~(#4)
4555 }
4556
```

31.1 Imports with modification

```
<@@=stex_copymodule>
   \cs_new_protected:Nn \stex_get_symbol_in_seq:nn {
     \tl_if_head_eq_catcode:nNTF { #1 } \relax {
        \tl_set:Nn \l_tmpa_tl { #1 }
4560
        \__stex_copymodule_get_symbol_from_cs:
4561
     7.
4562
       % argument is a string
4563
       % is it a command name?
4564
        \cs_if_exist:cTF { #1 }{
4565
          \cs_set_eq:Nc \l_tmpa_tl { #1 }
4566
          \str_set:Nx \l_tmpa_str { \cs_argument_spec:N \l_tmpa_tl }
4567
          \str_if_empty:NTF \l_tmpa_str {
            \exp_args:Nx \cs_if_eq:NNTF {
              \tl_head:N \l_tmpa_tl
            } \stex_invoke_symbol:n {
              \__stex_copymodule_get_symbol_from_cs:n{ #2 }
4572
            }{
4573
               __stex_copymodule_get_symbol_from_string:nn { #1 }{ #2 }
4574
4575
          }
4576
               _stex_copymodule_get_symbol_from_string:nn { #1 }{ #2 }
4577
          }
4578
       }{
4579
          % argument is not a command name
           __stex_copymodule_get_symbol_from_string:nn { #1 }{ #2 }
4581
          % \l_stex_all_symbols_seq
4582
4583
     }
4584
4585 }
4586
   \cs_new_protected:Nn \__stex_copymodule_get_symbol_from_string:nn {
4587
      \str_set:Nn \l_tmpa_str { #1 }
4588
      \bool_set_false:N \l_tmpa_bool
      \bool_if:NF \l_tmpa_bool {
        \tl_set:Nn \l_tmpa_tl {
          \msg_error:nnn{stex}{error/unknownsymbol}{#1}
4593
       \str_set:Nn \l_tmpa_str { #1 }
4594
        \int_set:Nn \l_tmpa_int { \str_count:N \l_tmpa_str }
4595
        \seq_map_inline:Nn #2 {
4596
          \str_set:Nn \l_tmpb_str { ##1 }
4597
          \str_if_eq:eeT { \l_tmpa_str } {
4598
            \str_range:Nnn \l_tmpb_str { -\l_tmpa_int } { -1 }
4599
          } {
            \seq_map_break:n {
              \tl_set:Nn \l_tmpa_tl {
                \str_set:Nn \l_stex_get_symbol_uri_str {
4604
                  ##1
4605
              }
4606
            }
4607
4608
```

```
4609
        \l_tmpa_tl
4610
4611
   }
4612
4613
    \cs_new_protected:Nn \__stex_copymodule_get_symbol_from_cs:n {
4614
      \exp_args:NNx \tl_set:Nn \l_tmpa_tl
4615
        { \tl_tail:N \l_tmpa_tl }
4616
      \tl_if_single:NTF \l_tmpa_tl {
4617
        \exp_args:No \tl_if_head_is_group:nTF \l_tmpa_tl {
4618
          \exp_after:wN \str_set:Nn \exp_after:wN
4619
            \l_stex_get_symbol_uri_str \l_tmpa_tl
4620
          \__stex_copymodule_get_symbol_check:n { #1 }
4621
       }{
4622
          % TODO
4623
          % tail is not a single group
4624
4625
4626
       % TODO
4627
       % tail is not a single group
     }
4629
4630 }
4631
   \cs_new_protected:\n \__stex_copymodule_get_symbol_check:n {
4632
     \exp_args:NNx \seq_if_in:NnF #1 \l_stex_get_symbol_uri_str {
4633
        \msg_error:nnxx{stex}{error/copymodule/notallowed}{\l_stex_get_symbol_uri_str}{
4634
          :~\seq_use:Nn #1 {,~}
4635
4636
     }
4637
4638 }
4639
   \cs_new_protected:Nn \stex_copymodule_start:nnnn {
4640
4641
     % import module
      \stex_import_module_uri:nn { #1 } { #2 }
4642
      \str_set:Nx \l_stex_current_copymodule_name_str {#3}
4643
      \stex_import_require_module:nnnn
4644
        { \l_stex_import_ns_str } { \l_stex_import_archive_str }
4645
4646
        { \l_stex_import_path_str } { \l_stex_import_name_str }
      \stex_collect_imports:n {\l_stex_import_ns_str ?\l_stex_import_name_str }
     \seq_set_eq:NN \l__stex_copymodule_copymodule_modules_seq \l_stex_collect_imports_seq
4650
     % fields
4651
     \seq_clear:N \l__stex_copymodule_copymodule_fields_seq
4652
      \seq_map_inline: Nn \l__stex_copymodule_copymodule_modules_seq {
4653
        \seq_map_inline:cn {c_stex_module_##1_constants}{
4654
          \exp_args:NNx \seq_put_right:Nn \l__stex_copymodule_copymodule_fields_seq {
4655
            ##1 ? ####1
4656
          }
4657
4658
       }
4659
     }
4660
4661
     % setup prop
     \seq_clear:N \l_tmpa_seq
4662
```

```
\exp_args:NNx \prop_set_from_keyval:Nn \l_stex_current_copymodule_prop {
4663
                  = \l_stex_current_copymodule_name_str ,
4664
                  = \l_stex_current_module_str ,
4665
       module
       from
                  = \l_stex_import_ns_str ?\l_stex_import_name_str ,
4666
       includes
                  = \l_tmpa_seq %,
4667
                   = \l_tmpa_seq
        fields
4668
4669
     \stex_debug:nn{copymodule}{#4~for~module~{\l_stex_import_ns_str ?\l_stex_import_name_str}
4670
       as~\l_stex_current_module_str?\l_stex_current_copymodule_name_str}
4671
        \stex_debug:nn{copymodule}{modules:\seq_use:Nn \l__stex_copymodule_copymodule_modules_se
4672
     stex_debug:nn{copymodule}{fields:\seq_use:Nn \l__stex_copymodule_copymodule_fields_seq {,
4673
4674
     \stex_if_do_html:T {
4675
        \begin{stex_annotate_env} {#4} {
4676
          \l_stex_current_module_str?\l_stex_current_copymodule_name_str
4677
4678
        \stex_annotate_invisible:nnn{domain}{\l_stex_import_ns_str ?\l_stex_import_name_str}{}
4679
4680
4681 }
   \cs_new_protected:Nn \stex_copymodule_end:n {
4683
     % apply to every field
4684
     \def \l_tmpa_cs ##1 ##2 {#1}
4685
4686
     \tl_clear:N \__stex_copymodule_module_tl
4687
     \tl_clear:N \__stex_copymodule_exec_tl
4688
4689
     %\prop_get:NnN \l_stex_current_copymodule_prop {fields} \l_tmpa_seq
4690
     \seq_clear:N \__stex_copymodule_fields_seq
4691
     \seq_map_inline:Nn \l__stex_copymodule_copymodule_modules_seq {
4693
        \seq_map_inline:cn {c_stex_module_##1_constants}{
4694
4695
          \tl_clear:N \__stex_copymodule_curr_symbol_tl % <- wrap in current symbol html</pre>
4696
          \l_tmpa_cs{##1}{####1}
4697
4698
          \str_if_exist:cTF {l__stex_copymodule_copymodule_##1?####1_name_str} {
4699
            \str_set_eq:Nc \__stex_copymodule_curr_name_str {l__stex_copymodule_copymodule_##1?#
4700
            \stex_if_do_html:T {
4701
              \tl_put_right:Nx \__stex_copymodule_curr_symbol_tl {
                \stex_annotate_invisible:nnn{alias}{\use:c{l__stex_copymodule_copymodule_##1?###
              }
           }
4705
         }{
4706
            \str_set:Nx \__stex_copymodule_curr_name_str { \l_stex_current_copymodule_name_str /
4707
4708
4709
          \prop_set_eq:Nc \l_tmpa_prop {l_stex_symdecl_ ##1?####1 _prop}
4710
          \prop_put:\nx \l_tmpa_prop { name } \__stex_copymodule_curr_name_str
4711
4712
          \prop_put:Nnx \l_tmpa_prop { module } \l_stex_current_module_str
4713
4714
          \tl_if_exist:cT {l__stex_copymodule_copymodule_##1?####1_def_tl}{
4715
            \stex_if_do_html:T {
              \tl_put_right:Nx \__stex_copymodule_curr_symbol_tl {
4716
```

```
$\stex_annotate_invisible:nnn{definiens}{}{\exp_after:wN \exp_not:N\csname 1__st
             }
4718
           }
4719
            \prop_put:Nnn \l_tmpa_prop { defined } { true }
4720
4721
4722
          \stex_add_constant_to_current_module:n \__stex_copymodule_curr_name_str
4723
          \tl_put_right:Nx \__stex_copymodule_module_tl {
4724
            \seq_clear:c {1_stex_symdecl_ \1_stex_current_module_str ? \__stex_copymodule_curr_r
            \prop_set_from_keyval:cn {
              l_stex_symdecl_\l_stex_current_module_str ? \__stex_copymodule_curr_name_str _prop
            }{
4728
              \prop_to_keyval:N \l_tmpa_prop
4729
4730
         }
4731
4732
          \str_if_exist:cT {l__stex_copymodule_copymodule_##1?###1_macroname_str} {
4733
            \stex_if_do_html:T {
4734
              \tl_put_right:Nx \__stex_copymodule_curr_symbol_tl {
                \stex_annotate_invisible:nnn{macroname}{\use:c{l__stex_copymodule_copymodule_##1
              }
           }
            \tl_put_right:Nx \__stex_copymodule_module_tl {
              \tl_set:cx {\use:c{l__stex_copymodule_copymodule_##1?####1_macroname_str}}{
                \stex_invoke_symbol:n {
4741
                  \l_stex_current_module_str ? \__stex_copymodule_curr_name_str
4742
4743
             }
4744
           }
4745
         }
4747
          \seq_put_right:Nx \__stex_copymodule_fields_seq {\l_stex_current_module_str ? \__stex_
4748
4749
          \tl_put_right:Nx \__stex_copymodule_exec_tl {
4750
            \stex_copy_notations:nn {\l_stex_current_module_str ? \__stex_copymodule_curr_name_s
4751
4752
4753
          \tl_put_right:Nx \__stex_copymodule_exec_tl {
4754
            \stex_if_do_html:TF{
4755
              \stex_annotate_invisible:nnn{assignment} {##1?####1} { \exp_after:wN \exp_not:n \e
           }{
              \exp_after:wN \exp_not:n \exp_after:wN {\__stex_copymodule_curr_symbol_tl}
           }
4759
         }
4760
       }
4761
     }
4762
4763
4764
     \prop_put:Nno \l_stex_current_copymodule_prop {fields} \__stex_copymodule_fields_seq
4765
     \tl_put_left:Nx \__stex_copymodule_module_tl {
4766
        \prop_set_from_keyval:cn {
4768
         l_stex_copymodule_ \l_stex_current_module_str?\l_stex_current_copymodule_name_str _pro
4769
```

\prop_to_keyval:N \l_stex_current_copymodule_prop

```
}
4771
     }
4772
4773
      \seq_gput_right:cx{c_stex_module_\l_stex_current_module_str _copymodules}{
4774
        \l_stex_current_module_str?\l_stex_current_copymodule_name_str
4775
4776
4777
      \exp_args:No \stex_execute_in_module:n \__stex_copymodule_module_tl
4778
4779
      \stex_debug:nn{copymodule}{result:\meaning \__stex_copymodule_module_tl}
      \stex_debug:nn{copymodule}{output:\meaning \__stex_copymodule_exec_tl}
4780
4781
      \__stex_copymodule_exec_tl
4782
      \stex_if_do_html:T {
4783
        \end{stex_annotate_env}
4784
4785
4786 }
4787
    \NewDocumentEnvironment {copymodule} { O{} m m}{
      \stex_copymodule_start:nnnn { #1 }{ #2 }{ #3 }{ copymodule }
      \stex_deactivate_macro:Nn \symdecl {module~environments}
      \stex_deactivate_macro:Nn \symdef {module~environments}
      \stex_deactivate_macro:Nn \notation {module~environments}
4792
      \stex_reactivate_macro:N \assign
4793
      \stex_reactivate_macro:N \renamedecl
4794
      \stex_reactivate_macro:N \donotcopy
4795
      \stex_smsmode_do:
4796
4797 }{
      \stex_copymodule_end:n {}
4798
4799
4800
   \NewDocumentEnvironment {interpretmodule} { O{} m m}{
4801
      \stex_copymodule_start:nnnn { #1 }{ #2 }{ #3 }{ interpretmodule }
4802
      \stex_deactivate_macro:Nn \symdecl {module~environments}
4803
      \stex_deactivate_macro:Nn \symdef {module~environments}
4804
      \stex_deactivate_macro:Nn \notation {module~environments}
4805
      \stex_reactivate_macro:N \assign
4806
      \stex_reactivate_macro:N \renamedecl
4807
      \stex_reactivate_macro:N \donotcopy
      \stex_smsmode_do:
4810 }{
4811
      \stex_copymodule_end:n {
        \tl_if_exist:cF {
4812
          l__stex_copymodule_copymodule_##1?##2_def_tl
4813
        }{
4814
          \str_if_eq:eeF {
4815
            \prop_item:cn{
4816
              l_stex_symdecl_ ##1 ? ##2 _prop }{ defined }
4817
4818
          }{ true }{
            \msg_error:nnxx{stex}{error/interpretmodule/nodefiniens}{
4819
4820
              ##1?##2
            }{\l_stex_current_copymodule_name_str}
4822
4823
       }
     }
4824
```

```
4825
4826
   \iffalse \begin{stex_annotate_env} \fi
4827
   \NewDocumentEnvironment {realization} { O{} m}{
4828
      \stex_copymodule_start:nnnn { #1 }{ #2 }{ #2 }{ realize }
4829
      \stex_deactivate_macro:Nn \symdecl {module~environments}
4830
      \stex_deactivate_macro:Nn \symdef {module~environments}
4831
      \stex_deactivate_macro:Nn \notation {module~environments}
4832
      \stex_reactivate_macro:N \donotcopy
4833
      \stex_reactivate_macro:N \assign
4834
4835
      \stex_smsmode_do:
4836 }{
      \stex_import_module_uri:nn { #1 } { #2 }
4837
      \tl_clear:N \__stex_copymodule_exec_tl
4838
      \tl_set:Nx \__stex_copymodule_module_tl {
4839
        \stex_import_require_module:nnnn
4840
          { \l_stex_import_ns_str } { \l_stex_import_archive_str }
4841
          { \l_stex_import_path_str } { \l_stex_import_name_str }
4842
4843
      \seq_map_inline:Nn \l__stex_copymodule_copymodule_modules_seq {
4845
        \seq_map_inline:cn {c_stex_module_##1_constants}{
4846
          \str_set:Nx \__stex_copymodule_curr_name_str { \l_stex_current_copymodule_name_str / #
4847
          \tl_if_exist:cT {l__stex_copymodule_copymodule_##1?####1_def_tl}{
4848
            \stex_if_do_html:T {
4849
              \tl_put_right:Nx \__stex_copymodule_exec_tl {
4850
                \stex_annotate_invisible:nnn{assignment} {##1?####1} {
4851
                  $\stex_annotate_invisible:nnn{definiens}{}{\exp_after:wN \exp_not:N\csname l__
4852
4853
              }
            }
4855
            \tl_put_right:Nx \__stex_copymodule_module_tl {
4856
4857
              \prop_put:cnn {l_stex_symdecl_##1?####1_prop}{ defined }{ true }
4858
          }
4859
     }}
4860
4861
      \exp_args:No \stex_execute_in_module:n \__stex_copymodule_module_tl
4862
      \__stex_copymodule_exec_tl
      \stex_if_do_html:T {\end{stex_annotate_env}}
4866
4867
   \NewDocumentCommand \donotcopy { m }{
4868
     \str_clear:N \l_stex_import_name_str
4869
     \str_set:Nn \l_tmpa_str { #1 }
4870
      \int_set:Nn \l_tmpa_int { \str_count:N \l_tmpa_str }
4871
      \seq_map_inline:Nn \l_stex_all_modules_seq {
4872
        \str_set:Nn \l_tmpb_str { ##1 }
4873
4874
        \str_if_eq:eeT { \l_tmpa_str } {
4875
          \str_range:Nnn \l_tmpb_str { -\l_tmpa_int } { -1 }
4876
       } {
          \seq_map_break:n {
4877
            \stex_if_do_html:T {
4878
```

```
\stex_if_smsmode:F {
4879
                \stex_annotate_invisible:nnn{donotcopy}{##1}{
4880
                   \stex_annotate:nnn{domain}{##1}{}
4881
4882
              }
4883
            }
4884
            \str_set_eq:NN \l_stex_import_name_str \l_tmpb_str
4885
          }
4886
       }
        \seq_map_inline:cn {c_stex_module_##1_copymodules}{
          \str_set:Nn \l_tmpb_str { ####1 }
          \str_if_eq:eeT { \l_tmpa_str } {
4890
            \str_range:Nnn \l_tmpb_str { -\l_tmpa_int } { -1 }
4891
          } {
4892
            \seq_map_break:n {\seq_map_break:n {
4893
              \stex_if_do_html:T {
4894
                \stex_if_smsmode:F {
4895
                   \stex_annotate_invisible:nnn{donotcopy}{####1}{
                     \stex_annotate:nnn{domain}{
                       \prop_item:cn {l_stex_copymodule_ ####1 _prop}{module}
                    }{}
                  }
                }
4901
              }
              \str_set:Nx \l_stex_import_name_str {
4903
                \prop_item:cn {l_stex_copymodule_ ####1 _prop}{module}
4904
              }
4905
            }}
4906
         }
4907
       }
     }
      \str_if_empty:NTF \l_stex_import_name_str {
4910
       % TODO throw error
4911
     }{
4912
        \stex_collect_imports:n {\l_stex_import_name_str }
4913
        \seq_map_inline:Nn \l_stex_collect_imports_seq {
4914
          \seq_remove_all:Nn \l__stex_copymodule_copymodule_modules_seq { ##1 }
4915
          \seq_map_inline:cn {c_stex_module_##1_constants}{
4916
4917
            \seq_remove_all:Nn \l__stex_copymodule_copymodule_fields_seq { ##1 ? ###1 }
            \bool_lazy_any:nT {
              { \cs_if_exist_p:c {l__stex_copymodule_copymodule_##1?###1_name_str}}
              { \cs_if_exist_p:c {l__stex_copymodule_copymodule_##1?####1_macroname_str}}
4921
              { \cs_if_exist_p:c {l__stex_copymodule_copymodule_##1?####1_def_tl}}
            }{
4922
              % TODO throw error
4923
            }
4924
         }
4925
4926
        \prop_get:NnN \l_stex_current_copymodule_prop { includes } \l_tmpa_seq
4927
4928
        \seq_put_right:Nx \l_tmpa_seq {\l_stex_import_name_str }
        \prop_put:Nno \l_stex_current_copymodule_prop {includes} \l_tmpa_seq
4930
     }
4931
      \stex_smsmode_do:
4932 }
```

```
4933
   \NewDocumentCommand \assign { m m }{
4934
     \stex_get_symbol_in_seq:nn {#1} \l__stex_copymodule_copymodule_fields_seq
4935
     \stex_debug:nn{assign}{defining~{\l_stex_get_symbol_uri_str}~as~\detokenize{#2}}
4936
     \tl_set:cn {l__stex_copymodule_copymodule_\l_stex_get_symbol_uri_str _def_tl}{#2}
4937
     \stex_smsmode_do:
4938
4939
4940
   \keys_define:nn { stex / renamedecl } {
                  .str_set_x:N = \l_stex_renamedecl_name_str
4942
4943 }
   \cs_new_protected:Nn \__stex_copymodule_renamedecl_args:n {
4944
     \str_clear:N \l_stex_renamedecl_name_str
4945
     \keys_set:nn { stex / renamedecl } { #1 }
4946
4947
4948
   \NewDocumentCommand \renamedecl { O{} m m}{
4949
     \__stex_copymodule_renamedecl_args:n { #1 }
4950
     \stex_get_symbol_in_seq:nn {#2} \l__stex_copymodule_copymodule_fields_seq
     \stex_debug:nn{renamedecl}{renaming~{\l_stex_get_symbol_uri_str}~to~#3}
     \str_set:cx {l__stex_copymodule_copymodule_\l_stex_get_symbol_uri_str _macroname_str}{#3}
     \str_if_empty:NTF \l_stex_renamedecl_name_str {
4954
        \tl_set:cx { #3 }{ \stex_invoke_symbol:n {
4955
          \l_stex_get_symbol_uri_str
4956
       } }
4957
     } {
4958
4959
        \str_set:cx {l__stex_copymodule_copymodule_\l_stex_get_symbol_uri_str _name_str}{\l_stex_
        \stex_debug:nn{renamedecl}{@~\l_stex_current_module_str ? \l_stex_renamedecl_name_str}
4960
        \prop_set_eq:cc {l_stex_symdecl_
4961
          \l_stex_current_module_str ? \l_stex_renamedecl_name_str
4963
       }{l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop}
4965
        \seq_set_eq:cc {l_stex_symdecl_
          \l_stex_current_module_str ? \l_stex_renamedecl_name_str
4966
          _notations
4967
       }{l_stex_symdecl_ \l_stex_get_symbol_uri_str _notations}
4968
        \prop_put:cnx {l_stex_symdecl_
4969
          \l_stex_current_module_str ? \l_stex_renamedecl_name_str
4970
4971
          _prop
       }{ name }{ \l_stex_renamedecl_name_str }
        \prop_put:cnx {l_stex_symdecl_
          \l_stex_current_module_str ? \l_stex_renamedecl_name_str
4975
       }{ module }{ \l_stex_current_module_str }
4976
        \exp_args:NNx \seq_put_left:Nn \l__stex_copymodule_copymodule_fields_seq {
4977
          \l_stex_current_module_str ? \l_stex_renamedecl_name_str
4978
4979
        \tl_set:cx { #3 }{ \stex_invoke_symbol:n {
4980
          \l_stex_current_module_str ? \l_stex_renamedecl_name_str
4981
4982
       } }
     }
4984
     \stex_smsmode_do:
4985 }
```

```
4987 \stex_deactivate_macro:Nn \assign {copymodules}
4988 \stex_deactivate_macro:Nn \renamedecl {copymodules}
4989 \stex_deactivate_macro:Nn \donotcopy {copymodules}
4990
4991
```

31.2 The feature environment

structural@feature

```
<@@=stex_features>
4992
   \NewDocumentEnvironment{structural_feature_module}{ m m m }{
     \stex_if_in_module:F {
       \msg_set:nnn{stex}{error/nomodule}{
          Structural~Feature~has~to~occur~in~a~module:\\
4997
          Feature~#2~of~type~#1\\
4998
          In~File:~\stex_path_to_string:N \g_stex_currentfile_seq
4999
5000
        \msg_error:nn{stex}{error/nomodule}
5001
5002
5003
      \str_set_eq:NN \l_stex_feature_parent_str \l_stex_current_module_str
5004
     \stex_module_setup:nn{meta=NONE}{#2 - #1}
5006
5007
     \stex_if_do_html:T {
5008
        \begin{stex_annotate_env}{ feature:#1 }{\l_stex_feature_parent_str ? #2 - #1}
5009
          \stex_annotate_invisible:nnn{header}{}{ #3 }
5010
5011
5012 }{
      \str_gset_eq:NN \l_stex_last_feature_str \l_stex_current_module_str
5013
      \prop_gput:cnn {c_stex_module_ \l_stex_current_module_str _prop}{feature}{#1}
5014
      \stex_debug:nn{features}{
5015
       Feature: \l_stex_last_feature_str
5016
5017
     \stex_if_do_html:T {
5018
        \end{stex_annotate_env}
5019
5020
5021 }
```

31.3 Structure

structure

```
5022 \@@=stex_structures\\
5023 \cs_new_protected:\Nn \stex_add_structure_to_current_module:nn {
5024 \prop_if_exist:cF {c_stex_module_\l_stex_current_module_str_structures}{
5025 \prop_new:c {c_stex_module_\l_stex_current_module_str_structures}}
5026 \\
5027 \prop_gput:cxx{c_stex_module_\l_stex_current_module_str_structures}\\
5028 \{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{
```

```
5031 \keys_define:nn { stex / features / structure } {
                   .str_set_x:N = \l__stex_structures_name_str ,
5032
     name
5033
5034
    \cs_new_protected:Nn \__stex_structures_structure_args:n {
5035
      \str_clear:N \l__stex_structures_name_str
5036
      \keys_set:nn { stex / features / structure } { #1 }
5037
5038
5039
   \NewDocumentEnvironment{mathstructure}{m O{}}{
5040
      \__stex_structures_structure_args:n { #2 }
5041
      \str_if_empty:NT \l__stex_structures_name_str {
5042
        \str_set:Nx \l__stex_structures_name_str { #1 }
5043
5044
      \stex_suppress_html:n {
5045
        \bool_set_true:N \l_stex_symdecl_make_macro_bool
5046
        \exp_args:Nx \stex_symdecl_do:nn {
5047
         name = \l_stex_structures_name_str ,
5048
         def = {\STEXsymbol{module-type}{
            \STEXInternalTermMathOMSiiii {
              \prop_item:cn {c_stex_module_\l_stex_current_module_str _prop}
                { ns } ?
5052
                \prop_item:cn {c_stex_module_\l_stex_current_module_str _prop}
5053
                  { name } / \l_stex_structures_name_str - structure
5054
             }{}{0}{}
5055
         }}
5056
       }{ #1 }
5057
5058
      \exp_args:Nnnx
5059
      \begin{structural_feature_module}{ structure }
5061
        { \l_stex_structures_name_str }{}
      \stex_smsmode_do:
5062
5063 }{
      \end{structural_feature_module}
5064
      \_stex_reset_up_to_module:n \l_stex_last_feature_str
5065
      \exp_args:No \stex_collect_imports:n \l_stex_last_feature_str
5066
      \seq_clear:N \l_tmpa_seq
5067
      \seq_map_inline: Nn \l_stex_collect_imports_seq {
5068
        \seq_map_inline:cn{c_stex_module_##1_constants}{
          \seq_put_right:Nn \l_tmpa_seq { ##1 ? ####1 }
       }
5071
     }
5072
5073
     \exp_args:Nnno
      \prop_gput:cnn {c_stex_module_ \l_stex_last_feature_str _prop}{fields}\l_tmpa_seq
5074
      \stex_debug:nn{structure}{Fields:~\seq_use:Nn \l_tmpa_seq ,}
5075
      \stex_add_structure_to_current_module:nn
5076
        \l__stex_structures_name_str
5077
        \l_stex_last_feature_str
5078
5079
5080
      \stex_execute_in_module:x {
5081
        \tl_set:cn { #1 }{
5082
          \exp_not:N \stex_invoke_structure:nn {\l_stex_current_module_str }{ \l_stex_structure
5083
     }
5084
```

```
5085 }
5086
   \cs_new:Nn \stex_invoke_structure:nn {
5087
     \stex_invoke_symbol:n { #1?#2 }
5088
5089
5090
    \cs_new_protected:Nn \stex_get_structure:n {
5091
     \tl_if_head_eq_catcode:nNTF { #1 } \relax {
5092
       \tl_set:Nn \l_tmpa_tl { #1 }
       \__stex_structures_get_from_cs:
5094
     }{
5095
       \cs_if_exist:cTF { #1 }{
5096
          \cs_set_eq:Nc \l_tmpa_cs { #1 }
5097
          \str_set:Nx \l_tmpa_str {\cs_argument_spec:N \l_tmpa_cs }
5098
          \str_if_empty:NTF \l_tmpa_str {
5099
            \cs_if_eq:NNTF { \tl_head:N \l_tmpa_cs} \stex_invoke_structure:nn {
5100
              \__stex_structures_get_from_cs:
5101
           }{
5102
               .__stex_structures_get_from_string:n { #1 }
         }{
            5106
5107
       }{
5108
            _stex_structures_get_from_string:n { #1 }
5109
5110
     }
5111
5112 }
5113
   \cs_new_protected:Nn \__stex_structures_get_from_cs: {
5115
     \exp_args:NNx \tl_set:Nn \l_tmpa_tl
       { \tl_tail:N \l_tmpa_tl }
5116
5117
     \str_set:Nx \l_tmpa_str {
       \exp_after:wN \use_i:nn \l_tmpa_tl
5118
5119
     \str_set:Nx \l_tmpb_str {
5120
       \exp_after:wN \use_ii:nn \l_tmpa_tl
5121
5122
5123
     \str_set:Nx \l_stex_get_structure_str {
       \l_tmpa_str ? \l_tmpb_str
     \str_set:Nx \l_stex_get_structure_module_str {
5126
5127
       \exp_args:Nno \prop_item:cn {c_stex_module_\l_tmpa_str _structures}{\l_tmpb_str}
5128
   }
5129
5130
    \cs_new_protected:Nn \__stex_structures_get_from_string:n {
5131
     \tl_set:Nn \l_tmpa_tl {
5132
       \msg_error:nnn{stex}{error/unknownstructure}{#1}
5133
5134
5135
     \str_set:Nn \l_tmpa_str { #1 }
5136
     \int_set:Nn \l_tmpa_int { \str_count:N \l_tmpa_str }
5137
     \seq_map_inline: Nn \l_stex_all_modules_seq {
5138
```

```
\prop_if_exist:cT {c_stex_module_##1_structures} {
               5130
                          \prop_map_inline:cn {c_stex_module_##1_structures} {
               5140
                            \str_if_eq:eeT { \l_tmpa_str }{ \str_range:nnn {##1?###1}{-\l_tmpa_int}{-1}}{
               5141
                              \prop_map_break:n{\seq_map_break:n{
               5142
                                \tl_set:Nn \l_tmpa_tl {
               5143
                                  \str_set:Nn \l_stex_get_structure_str {##1?###1}
               5144
                                  \str_set:Nn \l_stex_get_structure_module_str {####2}
               5145
                                }
               5146
                             }}
               5147
                           }
               5148
                         }
               5149
               5150
               5151
                     \l_tmpa_tl
               5152
               5153 }
\instantiate
               5154
                   \keys_define:nn { stex / instantiate } {
               5155
                                  .str_set_x:N = \l__stex_structures_name_str
               5156
                     name
               5157 }
                   \cs_new_protected:Nn \__stex_structures_instantiate_args:n {
               5158
                     \str_clear:N \l__stex_structures_name_str
               5159
                     \keys_set:nn { stex / instantiate } { #1 }
               5160
               5161 }
                   \NewDocumentCommand \instantiate {m O{} m m O{}}{
                     \begingroup
                       \stex_get_structure:n {#3}
               5165
                       \__stex_structures_instantiate_args:n { #2 }
               5166
                       \str_if_empty:NT \l__stex_structures_name_str {
               5167
                         \str_set:Nn \l__stex_structures_name_str { #1 }
               5168
               5169
                       \exp_args:No \stex_activate_module:n \l_stex_get_structure_module_str
               5170
                       \seq_clear:N \l__stex_structures_fields_seq
               5171
                       \exp_args:Nx \stex_collect_imports:n \l_stex_get_structure_module_str
               5172
                       \seq_map_inline: Nn \l_stex_collect_imports_seq {
               5173
                         \seq_map_inline:cn {c_stex_module_##1_constants}{
               5174
                            \seq_put_right:Nx \l__stex_structures_fields_seq { ##1 ? ####1 }
               5175
                         }
               5176
                       }
               5177
               5178
                       \tl_if_empty:nF{#5}{
               5179
                         \seq_set_split:Nnn \l_tmpa_seq , {#5}
               5180
                          \prop_clear:N \l_tmpa_prop
               5181
                          \seq_map_inline:Nn \l_tmpa_seq {
               5182
                            \seq_set_split:Nnn \l_tmpb_seq = { ##1 }
                           \int_compare:nNnF { \seq_count:N \l_tmpb_seq } = 2 {
                              \msg_error:nnn{stex}{error/keyval}{##1}
                           }
               5186
                           \exp_args:Nx \stex_get_symbol_in_seq:nn {\seq_item:Nn \l_tmpb_seq 1} \l__stex_struct
               5187
                           \str_set_eq:NN \l__stex_structures_dom_str \l_stex_get_symbol_uri_str
               5188
                           \exp_args:NNx \seq_remove_all:Nn \l__stex_structures_fields_seq \l_stex_get_symbol_u
               5189
```

5190

\exp_args:Nx \stex_get_symbol:n {\seq_item:Nn \l_tmpb_seq 2}

```
\exp_args:Nxx \str_if_eq:nnF
5191
             5192
             {\prop_item:cn{1_stex_symdecl_\l_stex_get_symbol_uri_str _prop}{args}}{
5193
             \msg_error:nnxxxx{stex}{error/incompatible}
5194
               {\l_stex_structures_dom_str}
5195
               {\prop_item:cn{l_stex_symdecl_\l_stex_structures_dom_str _prop}{args}}
5196
               {\l_stex_get_symbol_uri_str}
5197
               {\prop_item:cn{l_stex_symdecl_\l_stex_get_symbol_uri_str _prop}{args}}
5198
           \prop_put:Nxx \l_tmpa_prop {\seq_item:Nn \l_tmpb_seq 1} \l_stex_get_symbol_uri_str
         }
       }
5202
5203
       \seq_map_inline: Nn \l__stex_structures_fields_seq {
5204
         \str_set:Nx \l_tmpa_str {field:\l__stex_structures_name_str . \prop_item:cn {l_stex_sy
5205
         \stex_debug:nn{instantiate}{Field~\l_tmpa_str :~##1}
5206
         \stex_add_constant_to_current_module:n {\l_tmpa_str}
         \stex_execute_in_module:x {
           \prop_set_from_keyval:cn { l_stex_symdecl_ \l_stex_current_module_str?\l_tmpa_str _p
                    = \l_tmpa_str ,
                    = \prop_item:cn {l_stex_symdecl_##1_prop}{args} ,
5212
             arity = \prop_item:cn {l_stex_symdecl_##1_prop}{arity} ,
5213
             assocs = \prop_item:cn {l_stex_symdecl_##1_prop}{assocs} ,
5214
             argnames = {\prop_item:cn {l_stex_symdecl_##1_prop}{argnames}}
5215
5216
5217
           \seq_clear:c {l_stex_symdecl_\l_stex_current_module_str?\l_tmpa_str _notations}
         }
5218
5219
         \seq_if_empty:cF{l_stex_symdecl_##1_notations}{
           \stex_find_notation:nn{##1}{}
           \stex_execute_in_module:x {
             \seq_put_right:cn {l_stex_symdecl_\l_stex_current_module_str?\l_tmpa_str _notation
5223
           }
5224
5225
           \stex_copy_control_sequence_ii:ccN
5226
             {stex_notation_\l_stex_current_module_str?\l_tmpa_str\c_hash_str \l_stex_notation_
5227
             {stex_notation_##1\c_hash_str \l_stex_notation_variant_str _cs}
5228
             \l tmpa tl
5229
           \exp_args:No \stex_execute_in_module:n \l_tmpa_tl
           \cs_if_exist:cT{stex_op_notation_##1\c_hash_str \l_stex_notation_variant_str _cs}{
5233
             \tl_set_eq:Nc \l_tmpa_cs {stex_op_notation_##1\c_hash_str \l_stex_notation_variant
5234
             \stex_execute_in_module:x {
               \tl set:cn
5236
               {stex_op_notation_\l_stex_current_module_str?\l_tmpa_str\c_hash_str \l_stex_notation_
5237
                 \exp_args:No \exp_not:n \l_tmpa_cs}
5238
             }
5239
           }
5240
         }
5243
         \prop_put:Nxx \l_tmpa_prop {\prop_item:cn {l_stex_symdecl_##1_prop}{name}}{\l_stex_cur
5244
```

```
}
5245
5246
       \stex_execute_in_module:x {
5247
          \prop_set_from_keyval:cn {l_stex_instance_\l_stex_current_module_str?\l__stex_structur
5248
            domain = \l_stex_get_structure_module_str ,
5249
            \prop_to_keyval:N \l_tmpa_prop
5250
         }
5251
         \tl_set:cn{ #1 }{\stex_invoke_instance:n{ \l_stex_current_module_str?\l_stex_structur
       }
5253
       \stex_debug:nn{instantiate}{
5254
         Instance~\l_stex_current_module_str?\l_stex_structures_name_str \\
5255
          \prop_to_keyval:N \l_tmpa_prop
5256
5257
       \exp_args:Nxx \stex_symdecl_do:nn {
5258
         type={\STEXsymbol{module-type}{
5259
            \STEXInternalTermMathOMSiiii {
5260
              \l_stex_get_structure_module_str
5261
            }{}{0}{}
5262
         }}
       }{\l_stex_structures_name_str}
5265 %
         \str_set:Nx \l_stex_get_symbol_uri_str {\l_stex_current_module_str?\l_stex_structures
5266
         \tl_set:Nn \l_stex_notation_after_do_tl {\__stex_notation_final:}
5267
         \t \norm{}{0}{}{\comp{#4}}
5268
    %
5269
       %\exp_args:Nx \notation{\l_stex_structures_name_str}{\comp{#5}}
5270
5271
     \stex_smsmode_do:\ignorespacesandpars
5272
5273 }
5274
   \cs_new_protected:Nn \stex_symbol_or_var:n {
5275
     \cs_if_exist:cTF{#1}{
5276
5277
       \cs_set_eq:Nc \l_tmpa_tl { #1 }
       \str_set:Nx \l_tmpa_str { \cs_argument_spec:N \l_tmpa_tl }
5278
       \str_if_empty:NTF \l_tmpa_str {
5279
          \exp_args:Nx \cs_if_eq:NNTF { \tl_head:N \l_tmpa_tl }
5280
            \stex_invoke_variable:n {
5281
              \bool_set_true:N \l_stex_symbol_or_var_bool
5282
5283
              \bool_set_false:N \l_stex_instance_or_symbol_bool
              \tl_set:Nx \l_tmpa_tl {\tl_tail:N \l_tmpa_tl}
              \tl_set:Nx \l_tmpa_tl {\exp_after:wN \use:n \l_tmpa_tl}
              \str_set:Nx \l_stex_get_symbol_uri_str {
                \exp_after:wN \use:n \l_tmpa_tl
5287
              }
            }{ % TODO \stex_invoke_varinstance:n
              \exp_args:Nx \cs_if_eq:NNTF { \tl_head:N \l_tmpa_tl } \stex_invoke_varinstance:n {
5290
                \bool_set_true: N \l_stex_symbol_or_var_bool
5291
                \bool_set_true:N \l_stex_instance_or_symbol_bool
                \tl_set:Nx \l_tmpa_tl {\tl_tail:N \l_tmpa_tl}
5293
                \tl_set:Nx \l_tmpa_tl {\exp_after:wN \use:n \l_tmpa_tl}
                \str_set:Nx \l_stex_get_symbol_uri_str {
                  \exp_after:wN \use:n \l_tmpa_tl
5297
              }{
5298
```

```
\bool_set_false:N \l_stex_symbol_or_var_bool
                \stex_get_symbol:n{#1}
5300
              }
5301
            }
5302
       }{
5303
             _stex_structures_symbolorvar_from_string:n{ #1 }
5304
5305
     }{
5306
           stex_structures_symbolorvar_from_string:n{ #1 }
5307
     }
5308
5309
5310
    \cs_new_protected:Nn \__stex_structures_symbolorvar_from_string:n {
5311
      \prop_if_exist:cTF {l_stex_symdecl_var://#1 _prop}{
5312
        \bool_set_true:N \l_stex_symbol_or_var_bool
5313
        \str_set:Nn \l_stex_get_symbol_uri_str { #1 }
5314
5315
        \bool_set_false:N \l_stex_symbol_or_var_bool
5316
        \stex_get_symbol:n{#1}
5317
     }
5318
5319 }
5320
   \keys_define:nn { stex / varinstantiate } {
5321
                   .str_set_x:N = \l_stex_structures_name_str,
5322
     name
     bind
                   .choices:nn
5323
          {forall, exists}
5324
          {\str_set:Nx \l_stex_structures_bind_str {\l_keys_choice_tl}}
5325
5326
5327
   \cs_new_protected:Nn \__stex_structures_varinstantiate_args:n {
5329
      \str_clear:N \l__stex_structures_name_str
     \str_clear:N \l__stex_structures_bind_str
5330
      \keys_set:nn { stex / varinstantiate } { #1 }
5331
5332
5333
   \NewDocumentCommand \varinstantiate {m O{} m m O{}}{
5334
      \begingroup
5335
        \stex_get_structure:n {#3}
5336
5337
        \__stex_structures_varinstantiate_args:n { #2 }
        \str_if_empty:NT \l__stex_structures_name_str {
          \str_set:Nn \l__stex_structures_name_str { #1 }
5340
        \stex_if_do_html:TF{
5341
          \stex_annotate:nnn{varinstance}{\l__stex_structures_name_str}
5342
       {\sc }{\sc :n}
5343
        ₹
5344
          \stex_if_do_html:T{
5345
            \stex_annotate_invisible:nnn{domain}{\l_stex_get_structure_module_str}{}
5346
5347
5348
          \seq_clear:N \l__stex_structures_fields_seq
          \exp_args:Nx \stex_collect_imports:n \l_stex_get_structure_module_str
5350
          \seq_map_inline:Nn \l_stex_collect_imports_seq {
            \seq_map_inline:cn {c_stex_module_##1_constants}{
5351
              \seq_put_right:Nx \l__stex_structures_fields_seq { ##1 ? ####1 }
5352
```

```
}
5353
         }
5354
         \exp_args:No \stex_activate_module:n \l_stex_get_structure_module_str
5355
         \prop_clear:N \l_tmpa_prop
5356
          \t: f_empty:nF {#5} {
5357
            \seq_set_split:Nnn \l_tmpa_seq , {#5}
5358
            \seq_map_inline:Nn \l_tmpa_seq {
5359
              \sq_set_split:Nnn \l_tmpb_seq = { ##1 }
              \int_compare:nNnF { \seq_count:N \l_tmpb_seq } = 2 {
                \msg_error:nnn{stex}{error/keyval}{##1}
              }
              \exp_args:Nx \stex_get_symbol_in_seq:nn {\seq_item:Nn \l_tmpb_seq 1} \l__stex_stru
              \str_set_eq:NN \l__stex_structures_dom_str \l_stex_get_symbol_uri_str
5365
5366
              \exp_args:NNx \seq_remove_all:Nn \l__stex_structures_fields_seq \l_stex_get_symbol
              \exp_args:Nx \stex_symbol_or_var:n {\seq_item:Nn \l_tmpb_seq 2}
5367
              \stex_if_do_html:T{
5368
                \stex_annotate:nnn{assign}{\l__stex_structures_dom_str,
5369
                \bool_if:NTF\l_stex_symbol_or_var_bool{var://}{}\l_stex_get_symbol_uri_str}{}
              }
              \bool_if:NTF \l_stex_symbol_or_var_bool {
                \exp_args:Nxx \str_if_eq:nnF
                  {\prop_item:cn{1_stex_symdecl_\l__stex_structures_dom_str _prop}{args}}
                  \label{lem:cnl} $$ {\bf cn{l\_stex\_symdecl\_var://l\_stex\_get\_symbol\_uri\_str\_prop}{args}} $$
                  \msg_error:nnxxxx{stex}{error/incompatible}
                    {\l_stex_structures_dom_str}
5377
                    \label{local_local_local_local_local} $$ {\bf _cn_local_l_stex_structures_dom_str _prop}{args} $$
5378
                    {\l_stex_get_symbol_uri_str}
5379
                    {\prop_item:cn{1_stex_symdecl_var://\l_stex_get_symbol_uri_str _prop}{args}}
5380
5381
                \prop_put:Nxx \l_tmpa_prop {\seq_item:Nn \l_tmpb_seq 1} {\stex_invoke_variable:r
             }{
                \exp_args:Nxx \str_if_eq:nnF
                  {\prop_item:cn{1_stex_symdecl_\l__stex_structures_dom_str _prop}{args}}
5385
                  {\prop_item:cn{l_stex_symdecl_\l_stex_get_symbol_uri_str _prop}{args}}{
5386
                  \msg_error:nnxxxx{stex}{error/incompatible}
5387
                    {\l_stex_structures_dom_str}
5388
                    {\prop_item:cn{l_stex_symdecl_\l__stex_structures_dom_str _prop}{args}}
5389
                    {\l_stex_get_symbol_uri_str}
5390
                    {\prop_item:cn{l_stex_symdecl_\l_stex_get_symbol_uri_str _prop}{args}}
                \prop_put:Nxx \l_tmpa_prop {\seq_item:Nn \l_tmpb_seq 1} {\stex_invoke_symbol:n {
             }
           }
         }
         \tl_gclear:N \g__stex_structures_aftergroup_tl
         \seq_map_inline:Nn \l__stex_structures_fields_seq {
5398
            \str_set:Nx \l_tmpa_str {\l_stex_structures_name_str . \prop_item:cn {l_stex_symdec
5399
            \stex_debug:nn{varinstantiate}{Field~\l_tmpa_str :~##1}
5400
            \seq_if_empty:cF{l_stex_symdecl_##1_notations}{
              \stex_find_notation:nn{##1}{}
              \cs_gset_eq:cc{g__stex_structures_tmpa_\l_tmpa_str _cs}
                {stex_notation_##1\c_hash_str \l_stex_notation_variant_str _cs}
5405
              \stex_debug:nn{varinstantiate}{Notation:~\cs_meaning:c{g__stex_structures_tmpa_\l_
```

\cs_if_exist:cT{stex_op_notation_##1\c_hash_str \l_stex_notation_variant_str _cs}{

```
\cs_gset_eq:cc {g__stex_structures_tmpa_op_\l_tmpa_str _cs}
                  {stex_op_notation_##1\c_hash_str \l_stex_notation_variant_str _cs}
5408
                  \stex_debug:nn{varinstantiate}{Operator~Notation:~\cs_meaning:c{g__stex_struct}
5409
             }
5410
           }
5411
5412
            \exp_args:NNx \tl_gput_right:Nn \g__stex_structures_aftergroup_tl {
5413
              \prop_set_from_keyval:cn { l_stex_symdecl_ var://\l_tmpa_str _prop}{
                       = \l_tmpa_str ,
                       = \prop_item:cn {l_stex_symdecl_##1_prop}{args} ,
                args
                arity = \prop_item:cn {l_stex_symdecl_##1_prop}{arity} ,
5417
                assocs = \prop_item:cn {l_stex_symdecl_##1_prop}{assocs} ,
5418
                argnames = {\prop_item:cn {l_stex_symdecl_##1_prop}{argnames}} ,
5419
              }
5420
              \cs_set_eq:cc {stex_var_notation_\l_tmpa_str _cs}
5421
                {g_stex_structures_tmpa_\l_tmpa_str _cs}
5422
              \cs_set_eq:cc {stex_var_op_notation_\l_tmpa_str _cs}
5423
                {g_stex_structures_tmpa_op_\l_tmpa_str _cs}
5424
            7
            \prop_put:Nxx \l_tmpa_prop {\prop_item:cn {l_stex_symdecl_##1_prop}{name}}{\stex_inv
         }
          \exp_args:NNx \tl_gput_right:Nn \g__stex_structures_aftergroup_tl {
5429
            \prop_set_from_keyval:cn {l_stex_varinstance_\l__stex_structures_name_str _prop }{
              domain = \l_stex_get_structure_module_str ,
5430
              \prop_to_keyval:N \l_tmpa_prop
5431
           }
5432
            \tl_set:cn { #1 }{\stex_invoke_varinstance:n {\l_stex_structures_name_str}}
5433
5434
            \tl_set:cn {l_stex_varinstance_\l__stex_structures_name_str _op_tl}{
              \exp_args:Nnx \exp_not:N \use:nn {
5435
                \str_set:Nn \exp_not:N \STEXInternalCurrentSymbolStr {var://\l__stex_structures_
5437
                \_stex_term_omv:nn {var://\l__stex_structures_name_str}{
                  \exp_not:n{
                    \_varcomp{#4}
5430
                  }
5440
                }
5441
5442
                \exp_not:n{\_stex_reset:N \STEXInternalCurrentSymbolStr}
5443
              }
5444
         }
       }
       \stex_debug:nn{varinstantiate}{\expandafter\detokenize\expandafter{\g__stex_structures_a
5449
       \aftergroup\g__stex_structures_aftergroup_tl
5450
     \endgroup
     \stex_smsmode_do:\ignorespacesandpars
5451
5452 }
5453
    \cs_new_protected:Nn \stex_invoke_instance:n {
5454
     \peek_charcode_remove:NTF ! {
5455
       \stex_invoke_symbol:n{#1}
5456
5457
5458
        \_stex_invoke_instance:nn {#1}
     }
5450
```

5460 }

```
\cs_new_protected:Nn \stex_invoke_varinstance:n {
                               5463
                                     \peek_charcode_remove:NTF ! {
                               5464
                                       \exp_args:Nnx \use:nn {
                               5465
                                         \def\comp{\_varcomp}
                               5466
                                         \use:c{l_stex_varinstance_#1_op_tl}
                                          _stex_reset:N \comp
                                       }
                               5470
                                    }{
                               5471
                                       \_stex_invoke_varinstance:nn {#1}
                               5472
                               5473
                               5474 }
                               5475
                                   \cs_new_protected:Nn \_stex_invoke_instance:nn {
                               5476
                                     \prop_if_in:cnTF {l_stex_instance_ #1 _prop}{#2}{
                               5477
                                       \exp_args:Nx \stex_invoke_symbol:n {\prop_item:cn{1_stex_instance_ #1 _prop}{#2}}
                               5478
                               5479
                                       \prop_set_eq:Nc \l_tmpa_prop{l_stex_instance_ #1 _prop}
                                       \msg_error:nnxxx{stex}{error/unknownfield}{#2}{#1}{
                               5481
                                         \prop_to_keyval:N \l_tmpa_prop
                               5482
                               5483
                                    }
                               5484
                               5485 }
                               5486
                                   \cs_new_protected:Nn \_stex_invoke_varinstance:nn {
                               5487
                                     \prop_if_in:cnTF {l_stex_varinstance_ #1 _prop}{#2}{
                               5488
                                       \prop_get:cnN{l_stex_varinstance_ #1 _prop}{#2}\l_tmpa_tl
                               5489
                                       \l_tmpa_tl
                                    }{
                               5491
                                       \msg_error:nnnnn{stex}{error/unknownfield}{#2}{#1}{}
                               5492
                               5493
                                    }
                               5494 }
                              (End definition for \instantiate. This function is documented on page 33.)
\stex_invoke_structure:nnn
                               5495 % #1: URI of the instance
                               5496 % #2: URI of the instantiated module
                                   \cs_new_protected:Nn \stex_invoke_structure:nnn {
                                     \tl_if_empty:nTF{ #3 }{
                               5498
                                       \prop_set_eq:Nc \l__stex_structures_structure_prop {
                                         c_stex_feature_ #2 _prop
                               5500
                                       }
                               5501
                                       \tl_clear:N \l_tmpa_tl
                                       \prop_get:NnN \l__stex_structures_structure_prop { fields } \l_tmpa_seq
                               5504
                                       \seq_map_inline:Nn \l_tmpa_seq {
                                         \seq_set_split:Nnn \l_tmpb_seq ? { ##1 }
                               5505
                                         \seq_get_right:NN \l_tmpb_seq \l_tmpa_str
                               5506
                                         \cs_if_exist:cT {
                               5507
                                           stex_notation_ #1/\l_tmpa_str \c_hash_str\c_hash_str _cs
                               5508
                               5509
                                           \tl_if_empty:NF \l_tmpa_tl {
                               5510
```

```
\tl_put_right:Nn \l_tmpa_tl {,}
5511
              }
5512
              \tl_put_right:Nx \l_tmpa_tl {
5513
                 5514
5515
            }
5516
         }
5517
          \verb|\exp_args:No \mathstruct \l_tmpa_tl|
5518
          \stex_invoke_symbol:n{#1/#3}
5520
       }
5521
5522 }
(\mathit{End \ definition \ for \ } \texttt{structure:nnn}. \ \mathit{This \ function \ is \ documented \ on \ page \ \ref{eq:condition}.})
_{5523} \langle /package \rangle
```

Chapter 32

STEX

-Statements Implementation

32.1 Definitions

definiendum

```
5531 \keys_define:nn {stex / definiendum }{
          .tl_set:N = \l__stex_statements_definiendum_pre_tl,
                            = \l__stex_statements_definiendum_post_tl,
     post
            .tl_set:N
             .str_set_x:N = \l__stex_statements_definiendum_root_str,
             .str\_set\_x: \mathbb{N} = \\ \\ 1\_stex\_statements\_definiendum\_gfa\_str
5535
5536 }
5537 \cs_new_protected:Nn \__stex_statements_definiendum_args:n {
     \str_clear:N \l__stex_statements_definiendum_root_str
5538
     \tl_clear:N \l__stex_statements_definiendum_post_tl
5539
     \str_clear:N \l__stex_statements_definiendum_gfa_str
     \keys_set:nn { stex / definiendum }{ #1 }
5541
5543 \NewDocumentCommand \definiendum { O{} m m} {
     \__stex_statements_definiendum_args:n { #1 }
     \stex_get_symbol:n { #2 }
5545
     \stex_ref_new_sym_target:n \l_stex_get_symbol_uri_str
5546
     \str_if_empty:NTF \l__stex_statements_definiendum_root_str {
5547
       \tl_if_empty:NTF \l__stex_statements_definiendum_post_tl {
5548
```

```
\tl_set:Nn \l_tmpa_t1 { #3 }
5549
        } {
5550
          \str_set:Nx \l__stex_statements_definiendum_root_str { #3 }
5551
          \tl_set:Nn \l_tmpa_tl {
5552
            \l__stex_statements_definiendum_pre_tl\l__stex_statements_definiendum_root_str\l__st
5553
5554
        }
5555
     } {
5556
        \tl_set:Nn \l_tmpa_tl { #3 }
5557
     }
5558
5559
     % TODO root
5560
      \stex_html_backend:TF {
5561
        \stex_annotate:nnn { definiendum } { \l_stex_get_symbol_uri_str } { \l_tmpa_tl }
5562
5563
        \exp_args:Nnx \defemph@uri { \l_tmpa_tl } { \l_stex_get_symbol_uri_str }
5564
5565
5566 }
   \stex_deactivate_macro: Nn \definiendum {definition~environments}
```

(End definition for definiendum. This function is documented on page 43.)

definame

```
\NewDocumentCommand \definame { O{} m } {
5569
      \__stex_statements_definiendum_args:n { #1 }
5570
     % TODO: root
5571
     \stex_get_symbol:n { #2 }
5572
      \stex_ref_new_sym_target:n \l_stex_get_symbol_uri_str
5573
      \str_set:Nx \l_tmpa_str {
5574
        \prop_item:cn { l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop } { name }
5575
5576
      \str_replace_all:Nnn \l_tmpa_str {-} {~}
5577
      \stex_html_backend:TF {
5578
        \stex_if_do_html:T {
          \stex_annotate:nnn { definiendum } { \l_stex_get_symbol_uri_str } {
            \l_tmpa_str\l__stex_statements_definiendum_post_tl
          }
5582
       }
5583
     } {
5584
        \exp_args:Nnx \defemph@uri {
5585
          \l_tmpa_str\l__stex_statements_definiendum_post_tl
5586
       } { \l_stex_get_symbol_uri_str }
5587
     }
5588
5589
    \stex_deactivate_macro:Nn \definame {definition~environments}
5590
5591
   \NewDocumentCommand \Definame { O{} m } {
5592
      \__stex_statements_definiendum_args:n { #1 }
5593
     \stex_get_symbol:n { #2 }
5594
      \str_set:Nx \l_tmpa_str {
5595
        \prop_item:cn { l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop } { name }
5596
5597
      \exp_args:NNno \str_replace_all:Nnn \l_tmpa_str {-} {~}
5598
```

```
5599
      \stex_ref_new_sym_target:n \l_stex_get_symbol_uri_str
      \stex_html_backend:TF {
5600
        \stex_if_do_html:T {
5601
          \stex_annotate:nnn { definiendum } { \l_stex_get_symbol_uri_str } {
5602
            \exp_after:wN \stex_capitalize:n \l_tmpa_str\l__stex_statements_definiendum_post_tl
5603
5604
       }
5605
     } {
5606
        \exp_args:Nnx \defemph@uri {
          \exp_after:wN \stex_capitalize:n \l_tmpa_str\l__stex_statements_definiendum_post_tl
5609
       } { \l_stex_get_symbol_uri_str }
     }
5610
5611
    \stex_deactivate_macro:Nn \Definame {definition~environments}
5612
5613
   \NewDocumentCommand \premise { m }{
5614
      \noindent\stex_annotate:nnn{ premise }{}{\ignorespaces #1 }
5615
5616
   \NewDocumentCommand \conclusion { m }{
      \noindent\stex_annotate:nnn{ conclusion }{}{\ignorespaces #1 }
5618
5619 }
   \NewDocumentCommand \definiens { O{} m }{
5620
      \str_clear:N \l_stex_get_symbol_uri_str
5621
     \tl_if_empty:nF {#1} {
5622
        \stex_get_symbol:n { #1 }
5623
5624
      \str_if_empty:NT \l_stex_get_symbol_uri_str {
5625
        \int_compare:nNnTF {\clist_count:N \l__stex_statements_sdefinition_for_clist} = 1 {
5626
          \str_set:Nx \l_stex_get_symbol_uri_str {\clist_item:Nn \l__stex_statements_sdefinition
5627
       }{
         % TODO throw error
5629
       }
5630
5631
     }
      \str_if_eq:eeT {\prop_item:cn {l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop}{module}}
5632
        {\l_stex_current_module_str}{
5633
          \str_if_eq:eeF {\prop_item:cn {l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop}{defin
5634
          {true}{
5635
            \prop_put:cnn{l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop}{defined}{true}
5636
5637
            \exp_args:Nx \stex_add_to_current_module:n {
              \prop_put:cnn{l_stex_symdecl_ \l_stex_get_symbol_uri_str _prop}{defined}{true}
         }
     }
5641
      \stex_annotate:nnn{ definiens }{\l_stex_get_symbol_uri_str}{ #2 }
5642
   }
5643
5644
   \NewDocumentCommand \varbindforall {m}{
5645
      \stex_symbol_or_var:n {#1}
5646
      \bool_if:NTF\l_stex_symbol_or_var_bool{
5647
        \stex if do html:T {
5648
          \stex_annotate_invisible:nnn {bindtype}{forall,\l_stex_get_symbol_uri_str}{}
5650
       }
5651
     }{
       % todo throw error
5652
```

```
}
5653
5654
5655
    \stex_deactivate_macro:Nn \premise {definition,~example~or~assertion~environments}
5656
    \stex_deactivate_macro:Nn \conclusion {example~or~assertion~environments}
    \stex_deactivate_macro:Nn \definiens {definition~environments}
    \stex_deactivate_macro:Nn \varbindforall {definition~or~assertion~environments}
(End definition for definame. This function is documented on page 43.)
    \keys_define:nn {stex / sdefinition }{
              .str_set_x:N = \sdefinitiontype,
      type
              .str_set_x:N = \sdefinitionid,
      id
              .str_set_x:N = \sdefinitionname,
5665
      name
              .clist\_set: \verb|N = \l_stex_statements_sdefinition_for_clist|,
      for
5666
      title
              .tl_set:N
                             = \sdefinitiontitle
5667
5668 }
    \cs_new_protected: Nn \__stex_statements_sdefinition_args:n {
5669
      \str_clear:N \sdefinitiontype
5670
      \str_clear:N \sdefinitionid
5671
      \str_clear:N \sdefinitionname
5672
      \clist_clear:N \l__stex_statements_sdefinition_for_clist
5673
      \tl_clear:N \sdefinitiontitle
5674
      \keys_set:nn { stex / sdefinition }{ #1 }
5675
5676
5677
    \NewDocumentEnvironment{sdefinition}{0{}}{
5678
      \__stex_statements_sdefinition_args:n{ #1 }
5679
      \stex_reactivate_macro:N \definiendum
5680
      \stex_reactivate_macro:N \definame
5681
      \stex_reactivate_macro:N \Definame
      \stex_reactivate_macro:N \premise
      \stex_reactivate_macro:N \definiens
      \stex_reactivate_macro:N \varbindforall
      \stex_if_smsmode:F{
5686
        \seq_clear:N \l_tmpb_seq
5687
        \clist_map_inline: Nn \l__stex_statements_sdefinition_for_clist {
5688
          \tl_if_empty:nF{ ##1 }{
5689
            \stex_get_symbol:n { ##1 }
5690
            \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
5691
               \l_stex_get_symbol_uri_str
            }
5693
          }
        }
5695
        \clist_set_from_seq:NN \l__stex_statements_sdefinition_for_clist \l_tmpb_seq
5696
```

sdefinition

5697

5698

5699

5700 5701 \exp_args:Nnnx

\str_if_empty:NF \sdefinitiontype {

\str_if_empty:NF \sdefinitionname {

\begin{stex_annotate_env}{definition}{\seq_use:Nn \l_tmpb_seq {,}}

\stex_annotate_invisible:nnn{typestrings}{\sdefinitiontype}{}

```
\tl_clear:N \l_tmpa_tl
                                \clist_map_inline:Nn \l_tmpa_clist {
                        5707
                                  \tl_if_exist:cT {__stex_statements_sdefinition_##1_start:}{
                        5708
                                    \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sdefinition_##1_start:}}
                        5709
                                  }
                        5710
                                }
                        5711
                                \tl_if_empty:NTF \l_tmpa_tl {
                        5712
                                  \__stex_statements_sdefinition_start:
                        5713
                                }{
                        5714
                                  \l_tmpa_tl
                        5715
                                }
                        5716
                        5717
                              \stex_ref_new_doc_target:n \sdefinitionid
                        5718
                              \stex_smsmode_do:
                        5719
                        5720 }{
                              \stex_suppress_html:n {
                        5721
                                \str_if_empty:NF \sdefinitionname { \stex_symdecl_do:nn{}{\sdefinitionname} }
                        5723
                              \stex_if_smsmode:F {
                        5724
                                \clist_set:No \l_tmpa_clist \sdefinitiontype
                        5725
                                \tl_clear:N \l_tmpa_tl
                        5726
                                \clist_map_inline:Nn \l_tmpa_clist {
                        5727
                                  \tl_if_exist:cT {__stex_statements_sdefinition_##1_end:}{
                        5728
                                    \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sdefinition_##1_end:}}
                        5729
                                  }
                        5730
                        5731
                                \tl_if_empty:NTF \l_tmpa_tl {
                        5733
                                  \__stex_statements_sdefinition_end:
                                }{
                        5734
                        5735
                                  \l_tmpa_tl
                        5736
                                \end{stex_annotate_env}
                        5737
                        5738
                        5739 }
\stexpatchdefinition
                           \cs_new_protected:Nn \__stex_statements_sdefinition_start: {
                              \stex_par:\noindent\titleemph{Definition\tl_if_empty:NF \sdefinitiontitle {
                        5741
                                ~(\sdefinitiontitle)
                        5742
                        5743
                        5744 }
                        5745
                            \cs_new_protected:Nn \__stex_statements_sdefinition_end: {\stex_par:\medskip}
                            \newcommand\stexpatchdefinition[3][] {
                                \str_set:Nx \l_tmpa_str{ #1 }
                                \str_if_empty:NTF \l_tmpa_str {
                                  \tl_set:Nn \__stex_statements_sdefinition_start: { #2 }
                        5750
                                  \tl_set:Nn \__stex_statements_sdefinition_end: { #3 }
                        5751
                                }{
                        5752
                                  \exp_after:wN \tl_set:Nn \csname __stex_statements_sdefinition_#1_start:\endcsname{ #2
                        5753
                                  \exp_after:wN \tl_set:Nn \csname __stex_statements_sdefinition_#1_end:\endcsname{ #3 }
                        5754
```

 $\verb|\stex_annotate_invisible:nnn{statementname}{\sdefinitionname}{}|$

\clist_set:No \l_tmpa_clist \sdefinitiontype

5703

5704

5705

5706

}

```
}
             5755
             5756 }
             (End definition for \stexpatchdefinition. This function is documented on page 50.)
\inlinedef inline:
                 \keys_define:nn {stex / inlinedef }{
             5757
                            .str_set_x:N = \sdefinitiontype,
             5758
                   type
                   id
                            .str_set_x:N = \sdefinitionid,
             5759
                            .clist_set:N = \l__stex_statements_sdefinition_for_clist ,
                   for
             5760
                            .str_set_x:N = \sdefinitionname
                   name
             5761
             5762 }
                 \cs_new_protected:Nn \__stex_statements_inlinedef_args:n {
             5763
                   \str_clear:N \sdefinitiontype
                   \str_clear:N \sdefinitionid
                   \str_clear:N \sdefinitionname
                   \clist_clear:N \l__stex_statements_sdefinition_for_clist
             5767
                   \keys_set:nn { stex / inlinedef }{ #1 }
             5768
             5769 }
                 \NewDocumentCommand \inlinedef { O{} m } {
             5770
                   \begingroup
             5771
                   \__stex_statements_inlinedef_args:n{ #1 }
             5772
                   \stex_reactivate_macro:N \definiendum
             5773
                   \stex_reactivate_macro:N \definame
             5774
                   \stex_reactivate_macro:N \Definame
             5775
                   \stex_reactivate_macro:N \premise
             5776
                   \stex_reactivate_macro:N \definiens
             5777
                   \stex_reactivate_macro:N \varbindforall
             5778
                   \stex_ref_new_doc_target:n \sdefinitionid
             5779
                   \stex_if_smsmode:TF{\stex_suppress_html:n {
             5780
                     \str_if_empty:NF \sdefinitionname { \stex_symdecl_do:nn{}{\sdefinitionname} }
             5781
                   }}{
             5782
                     \seq_clear:N \l_tmpb_seq
             5783
                     \clist_map_inline: Nn \l__stex_statements_sdefinition_for_clist {
                       \tl_if_empty:nF{ ##1 }{
                          \stex_get_symbol:n { ##1 }
                         \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
                            \l_stex_get_symbol_uri_str
             5788
             5789
                       }
             5790
                     }
             5791
                     \clist_set_from_seq:NN \l__stex_statements_sdefinition_for_clist \l_tmpb_seq
             5792
                     \exp_args:Nnx
             5793
                     \stex_annotate:nnn{definition}{\seq_use:Nn \l_tmpb_seq {,}}{
                       \str_if_empty:NF \sdefinitiontype {
                          \stex_annotate_invisible:nnn{typestrings}{\sdefinitiontype}{}
             5796
                       }
             5797
                       #2
             5798
                       \str_if_empty:NF \sdefinitionname {
             5799
                          \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sdefinitionname}}
             5800
                          \stex_annotate_invisible:nnn{statementname}{\sdefinitionname}{}
             5801
             5802
                     }
             5803
```

}

```
5805 \endgroup
5806 \stex_smsmode_do:
5807 }
(End definition for \inlinedef. This function is documented on page ??.)
```

32.2 Assertions

sassertion

```
5808
   \keys_define:nn {stex / sassertion }{
5809
              .str_set_x:N = \sassertiontype,
     type
5810
              .str_set_x:N = \sassertionid,
     id
5811
     title
                             = \sassertiontitle
5812
              .tl_set:N
              .clist_set:N = \l__stex_statements_sassertion_for_clist ,
     for
5813
              .str_set_x:N = \sin sassertionname
5814
5815 }
   \cs_new_protected:Nn \__stex_statements_sassertion_args:n {
5816
     \str_clear:N \sassertiontype
5817
     \str_clear:N \sassertionid
5818
     \str_clear:N \sassertionname
5819
     \clist_clear:N \l__stex_statements_sassertion_for_clist
5820
     \tl_clear:N \sassertiontitle
5821
      \keys_set:nn { stex / sassertion }{ #1 }
5822
5823 }
5824
   %\tl_new:N \g__stex_statements_aftergroup_tl
5825
5826
   \NewDocumentEnvironment{sassertion}{O{}}{
5827
      \__stex_statements_sassertion_args:n{ #1 }
5828
     \stex_reactivate_macro:N \premise
5829
      \stex_reactivate_macro:N \conclusion
5830
      \stex_reactivate_macro:N \varbindforall
5831
      \stex_if_smsmode:F {
5832
        \seq_clear:N \l_tmpb_seq
5833
5834
        \clist_map_inline: Nn \l__stex_statements_sassertion_for_clist {
5835
          \tl_if_empty:nF{ ##1 }{
            \stex_get_symbol:n { ##1 }
            \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
              \l_stex_get_symbol_uri_str
5838
5839
         }
5840
5841
        \exp_args:Nnnx
5842
        \begin{stex_annotate_env}{assertion}{\seq_use:Nn \l_tmpb_seq {,}}
5843
        \str_if_empty:NF \sassertiontype {
5844
          \stex_annotate_invisible:nnn{type}{\sassertiontype}{}
5845
       }
5847
        \str_if_empty:NF \sassertionname {
          \stex_annotate_invisible:nnn{statementname}{\sassertionname}{}
5848
5849
        \clist_set:No \l_tmpa_clist \sassertiontype
5850
       \tl_clear:N \l_tmpa_tl
5851
```

```
\tl_if_exist:cT {__stex_statements_sassertion_##1_start:}{
                                   \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sassertion_##1_start:}}
                       5854
                       5855
                       5856
                               \tl_if_empty:NTF \l_tmpa_tl {
                       5857
                                  \__stex_statements_sassertion_start:
                       5858
                       5859
                                  \l_tmpa_tl
                               }
                       5861
                       5862
                             }
                             \str_if_empty:NTF \sassertionid {
                       5863
                               \str_if_empty:NF \sassertionname {
                       5864
                                 \stex_ref_new_doc_target:n {}
                       5865
                       5866
                             } {
                       5867
                               \stex_ref_new_doc_target:n \sassertionid
                       5868
                             \stex_smsmode_do:
                       5870
                       5871
                             \str_if_empty:NF \sassertionname {
                       5872
                               \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sassertionname}}
                       5873
                               \stex_ref_new_sym_target:n {\l_stex_current_module_str ? \sassertionname}
                       5874
                             }
                       5875
                             \stex_if_smsmode:F {
                       5876
                               \clist_set:No \l_tmpa_clist \sassertiontype
                       5877
                               \tl_clear:N \l_tmpa_tl
                       5878
                               \clist_map_inline:Nn \l_tmpa_clist {
                       5879
                                 \tl_if_exist:cT {__stex_statements_sassertion_##1_end:}{
                       5880
                       5881
                                    \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sassertion_##1_end:}}
                                 }
                       5882
                       5883
                               \tl_if_empty:NTF \l_tmpa_tl {
                       5884
                       5885
                                  \__stex_statements_sassertion_end:
                               }{
                       5886
                                  \l_tmpa_tl
                       5887
                       5888
                       5889
                               \end{stex_annotate_env}
                       5890
                       5891 }
\stexpatchassertion
                       5892
                           \cs_new_protected:Nn \__stex_statements_sassertion_start: {
                       5893
                             \stex_par:\noindent\titleemph{Assertion~\tl_if_empty:NF \sassertiontitle {
                       5894
                               (\sassertiontitle)
                       5895
                           \cs_new_protected:Nn \__stex_statements_sassertion_end: {\stex_par:\medskip}
                       5899
                           \newcommand\stexpatchassertion[3][] {
                       5900
                               \str_set:Nx \l_tmpa_str{ #1 }
                       5901
                               \str_if_empty:NTF \l_tmpa_str {
                       5902
                                 \tl_set:Nn \__stex_statements_sassertion_start: { #2 }
                       5903
```

\clist_map_inline:Nn \l_tmpa_clist {

5852

```
\tl_set:Nn \__stex_statements_sassertion_end: { #3 }
                             5904
                                             }{
                             5905
                                                   \exp_after:wN \tl_set:Nn \csname __stex_statements_sassertion_#1_start:\endcsname{ #2
                             5906
                                                   \exp_after:wN \tl_set:Nn \csname __stex_statements_sassertion_#1_end:\endcsname{ #3 }
                             5907
                             5908
                             5909 }
                            (End definition for \stexpatchassertion. This function is documented on page 50.)
\inlineass
                           inline:
                             5910 \keys_define:nn {stex / inlineass }{
                                                            .str_set_x:N = \sassertiontype,
                             5911
                                         type
                                                            .str_set_x:N = \sassertionid,
                                         id
                             5912
                                                            . \verb|clist_set:N| = \label{eq:loss} = \label{eq:loss} | \label{eq
                                         for
                             5913
                                                            .str_set_x:N = \sassertionname
                                         name
                             5914
                             5915 }
                                     \cs_new_protected:Nn \__stex_statements_inlineass_args:n {
                             5916
                                         \str_clear:N \sassertiontype
                             5917
                                         \str_clear:N \sassertionid
                             5918
                                         \str_clear:N \sassertionname
                             5919
                                         \clist_clear:N \l__stex_statements_sassertion_for_clist
                             5920
                                          \keys_set:nn { stex / inlineass }{ #1 }
                             5921
                             5922 }
                                     \NewDocumentCommand \inlineass { O{} m } {
                             5923
                                          \begingroup
                             5924
                                          \stex_reactivate_macro:N \premise
                             5925
                                          \stex_reactivate_macro:N \conclusion
                             5926
                                          \stex_reactivate_macro:N \varbindforall
                             5927
                                          \__stex_statements_inlineass_args:n{ #1 }
                             5928
                                          \str_if_empty:NTF \sassertionid {
                             5929
                                              \str_if_empty:NF \sassertionname {
                             5930
                                                   \stex_ref_new_doc_target:n {}
                             5931
                             5932
                             5933
                                         } {
                             5934
                                              \stex_ref_new_doc_target:n \sassertionid
                             5935
                                          \stex_if_smsmode:TF{
                             5937
                                              \str_if_empty:NF \sassertionname {
                             5938
                                                   \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sassertionname}}
                             5939
                                                   \stex_ref_new_sym_target:n {\l_stex_current_module_str ? \sassertionname}
                             5940
                             5941
                                         }{
                             5942
                                              \seq_clear:N \l_tmpb_seq
                             5943
                                              \clist_map_inline: Nn \l__stex_statements_sassertion_for_clist {
                             5944
                                                   \tl_if_empty:nF{ ##1 }{
                                                       \stex_get_symbol:n { ##1 }
                             5946
                                                       \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
                             5947
                             5948
                                                            \l_stex_get_symbol_uri_str
                             5949
                                                  }
                             5950
                             5951
                                              \exp_args:Nnx
                             5952
```

\stex_annotate:nnn{assertion}{\seq_use:Nn \l_tmpb_seq {,}}{

```
\str_if_empty:NF \sassertiontype {
5954
            \stex_annotate_invisible:nnn{typestrings}{\sassertiontype}{}
5955
5956
          #2
5957
          \str_if_empty:NF \sassertionname {
5958
            \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sassertionname}}
5959
            \stex_ref_new_sym_target:n {\l_stex_current_module_str ? \sassertionname}
            \stex_annotate_invisible:nnn{statementname}{\sassertionname}{}
       }
5963
5964
      \endgroup
5965
      \stex_smsmode_do:
5966
5967
```

(End definition for \inlineass. This function is documented on page ??.)

32.3 Examples

sexample

```
\keys_define:nn {stex / sexample }{
5969
              .str_set_x:N = \exampletype,
     type
5970
              .str_set_x:N = \sexampleid,
5971
             .tl_set:N
                             = \sexampletitle,
5972
              .str_set_x:N = \sexamplename ,
5973
5974
              .clist_set:N = \l__stex_statements_sexample_for_clist,
5975 }
   \cs_new_protected:Nn \__stex_statements_sexample_args:n {
     \str_clear:N \sexampletype
5977
     \str_clear:N \sexampleid
5978
     \str_clear:N \sexamplename
5979
     \tl_clear:N \sexampletitle
5980
      \clist_clear:N \l__stex_statements_sexample_for_clist
5981
      \keys_set:nn { stex / sexample }{ #1 }
5982
5983
   \NewDocumentEnvironment{sexample}{0{}}{
      \__stex_statements_sexample_args:n{ #1 }
      \stex_reactivate_macro:N \premise
5987
     \stex_reactivate_macro:N \conclusion
5988
      \stex_if_smsmode:F {
5989
        \seq_clear:N \l_tmpb_seq
5990
        \clist_map_inline: Nn \l__stex_statements_sexample_for_clist {
5991
          \t! \int_{empty:nF{ \#1 }{}}
5992
            \stex_get_symbol:n { ##1 }
5993
            \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
              \l_stex_get_symbol_uri_str
         }
5997
       }
5998
        \exp_args:Nnnx
5999
        \begin{stex_annotate_env}{example}{\seq_use:Nn \l_tmpb_seq {,}}
6000
```

```
\stex_annotate_invisible:nnn{typestrings}{\sexampletype}{}
                     6002
                     6003
                             \str_if_empty:NF \sexamplename {
                     6004
                               \stex_annotate_invisible:nnn{statementname}{\sexamplename}{}
                     6005
                     6006
                             \clist_set:No \l_tmpa_clist \sexampletype
                     6007
                             \tl_clear:N \l_tmpa_tl
                     6008
                             \clist_map_inline:Nn \l_tmpa_clist {
                               \tl_if_exist:cT {__stex_statements_sexample_##1_start:}{
                                 \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sexample_##1_start:}}
                     6011
                               }
                     6012
                     6013
                             \tl_if_empty:NTF \l_tmpa_tl {
                     6014
                               \__stex_statements_sexample_start:
                     6015
                     6016
                               \l_tmpa_tl
                     6017
                             }
                     6018
                           \str_if_empty:NF \sexampleid {
                             \stex_ref_new_doc_target:n \sexampleid
                     6022
                     6023
                           \stex_smsmode_do:
                     6024 }{
                           \str_if_empty:NF \sexamplename {
                     6025
                             \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sexamplename}}
                     6026
                     6027
                           \stex_if_smsmode:F {
                     6028
                             \clist_set:No \l_tmpa_clist \sexampletype
                     6029
                             \tl_clear:N \l_tmpa_tl
                     6031
                             \clist_map_inline:Nn \l_tmpa_clist {
                               \tl_if_exist:cT {__stex_statements_sexample_##1_end:}{
                     6032
                     6033
                                 \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sexample_##1_end:}}
                     6034
                     6035
                             \tl_if_empty:NTF \l_tmpa_tl {
                     6036
                               \__stex_statements_sexample_end:
                     6037
                             }{
                     6038
                     6039
                               \l_tmpa_tl
                             \end{stex_annotate_env}
                     6042
                     6043 }
\stexpatchexample
                     6044
                         \cs_new_protected:Nn \__stex_statements_sexample_start: {
                           \stex_par:\noindent\titleemph{Example~\tl_if_empty:NF \sexampletitle {
                             (\sexampletitle)
                          }~}
                     6048
                     6049 }
                        \cs_new_protected:\n \__stex_statements_sexample_end: {\stex_par:\medskip}
                     6050
                     6051
                        \newcommand\stexpatchexample[3][] {
```

\str_if_empty:NF \sexampletype {

```
\str_set:Nx \l_tmpa_str{ #1 }
            6053
                    \str_if_empty:NTF \l_tmpa_str {
            6054
                      \tl_set:Nn \__stex_statements_sexample_start: { #2 }
            6055
                      \tl_set:Nn \__stex_statements_sexample_end: { #3 }
            6056
            6057
                       \exp_after:wN \tl_set:Nn \csname __stex_statements_sexample_#1_start:\endcsname{ #2 }
            6058
                      \exp_after:wN \tl_set:Nn \csname __stex_statements_sexample_#1_end:\endcsname{ #3 }
            6059
            6060
            (End definition for \stexpatchexample. This function is documented on page 50.)
\inlineex
          inline:
                \keys_define:nn {stex / inlineex }{
                          .str_set_x:N = \sexampletype,
                  type
                           .str_set_x:N = \sexampleid,
            6064
                  id
                           .clist_set:N = \l__stex_statements_sexample_for_clist ,
                  for
            6065
                           .str_set_x:N = \sexamplename
                  name
            6066
            6067 }
                \cs_new_protected:Nn \__stex_statements_inlineex_args:n {
            6068
                  \str_clear:N \sexampletype
            6069
                  \str_clear:N \sexampleid
            6070
                  \str_clear:N \sexamplename
            6071
                  \clist_clear:N \l__stex_statements_sexample_for_clist
            6072
                  \keys_set:nn { stex / inlineex }{ #1 }
            6073
            6074 }
                \NewDocumentCommand \inlineex { O{} m } {
            6075
                  \begingroup
            6076
                  \stex_reactivate_macro:N \premise
            6077
                  \stex_reactivate_macro:N \conclusion
            6078
                  \__stex_statements_inlineex_args:n{ #1 }
            6079
                  \str_if_empty:NF \sexampleid {
            6080
                    \stex_ref_new_doc_target:n \sexampleid
            6081
                  \stex_if_smsmode:TF{
                    \str_if_empty:NF \sexamplename {
                      \stex_suppress_html:n{\stex_symdecl_do:nn{}{\examplename}}
                    }
            6086
                  }{
            6087
                    \seq_clear:N \l_tmpb_seq
            6088
                    \clist_map_inline: Nn \l__stex_statements_sexample_for_clist {
            6089
                      \tl_if_empty:nF{ ##1 }{
            6090
                        \stex_get_symbol:n { ##1 }
            6091
                        \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
                           \l_stex_get_symbol_uri_str
                      }
            6096
            6097
                    \exp_args:Nnx
                    \stex_annotate:nnn{example}{\seq_use:Nn \l_tmpb_seq {,}}{
            6098
                      \str_if_empty:NF \sexampletype {
            6099
                        \stex_annotate_invisible:nnn{typestrings}{\sexampletype}{}
            6100
                      }
            6101
```

#2

(End definition for $\$ inlineex. This function is documented on page $\ref{eq:condition}$.)

32.4 Logical Paragraphs

sparagraph

```
6112 \keys_define:nn { stex / sparagraph} {
     id
              .str_set_x:N
                              = \sparagraphid ,
6113
                              = \l_stex_sparagraph_title_tl ,
6114
     title
              .tl_set:N
                             = \sparagraphtype ,
              .str_set_x:N
6115
     type
              .clist_set:N = \l__stex_statements_sparagraph_for_clist ,
6116
     for
                              = \sparagraphfrom ,
              .tl_set:N
     from
6117
              .tl_set:N
                              = \sparagraphto ,
6118
     to
                              = \l_stex_sparagraph_start_tl ,
     start
             .tl_set:N
6119
              .str_set:N
                              = \sparagraphname ,
6120
      imports .tl_set:N
                              = \l__stex_statements_sparagraph_imports_tl
6121
6122 }
6123
   \cs_new_protected:Nn \stex_sparagraph_args:n {
6124
      \tl_clear:N \l_stex_sparagraph_title_tl
6125
      \tl_clear:N \sparagraphfrom
6126
      \tl_clear:N \sparagraphto
6127
      \tl_clear:N \l_stex_sparagraph_start_tl
6128
      \tl_clear:N \l__stex_statements_sparagraph_imports_tl
6129
      \str_clear:N \sparagraphid
6130
      \str_clear:N \sparagraphtype
6131
6132
      \clist_clear:N \l__stex_statements_sparagraph_for_clist
6133
      \str_clear:N \sparagraphname
      \keys_set:nn { stex / sparagraph }{ #1 }
6135 }
   \newif\if@in@omtext\@in@omtextfalse
6136
6137
   \NewDocumentEnvironment {sparagraph} { O{} } {
6138
      \stex_sparagraph_args:n { #1 }
6139
      \tl_if_empty:NTF \l_stex_sparagraph_start_tl {
6140
        \tl_set_eq:NN \sparagraphtitle \l_stex_sparagraph_title_tl
6141
6142
        \tl_set_eq:NN \sparagraphtitle \l_stex_sparagraph_start_tl
6143
6144
6145
      \@in@omtexttrue
6146
      \stex_if_smsmode:F {
6147
        \seq_clear:N \l_tmpb_seq
        \clist_map_inline:Nn \l__stex_statements_sparagraph_for_clist {
6148
          \tilde{f}_{empty:nF{ ##1 }{ }}
6149
```

```
\stex_get_symbol:n { ##1 }
6150
            \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
6151
6152
              \l_stex_get_symbol_uri_str
6153
         }
6154
       }
6155
        \exp_args:Nnnx
6156
        \begin{stex_annotate_env}{paragraph}{\seq_use:Nn \l_tmpb_seq {,}}
6157
        \str_if_empty:NF \sparagraphtype {
          \stex_annotate_invisible:nnn{typestrings}{\sparagraphtype}{}
6159
        \str_if_empty:NF \sparagraphfrom {
6161
          \stex_annotate_invisible:nnn{from}{\sparagraphfrom}{}
6162
6163
        \str_if_empty:NF \sparagraphto {
6164
          \stex_annotate_invisible:nnn{to}{\sparagraphto}{}
6165
6166
        \str_if_empty:NF \sparagraphname {
6167
          \stex_annotate_invisible:nnn{statementname}{\sparagraphname}{}
        \clist_set:No \l_tmpa_clist \sparagraphtype
        \tl_clear:N \l_tmpa_tl
6171
6172
        \clist_map_inline:Nn \sparagraphtype {
          \tl_if_exist:cT {__stex_statements_sparagraph_##1_start:}{
6173
            \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sparagraph_##1_start:}}
6174
          }
6175
6176
        \stex_csl_to_imports:No \usemodule \l__stex_statements_sparagraph_imports_tl
6177
        \tl_if_empty:NTF \l_tmpa_tl {
6178
          \__stex_statements_sparagraph_start:
       }{
6180
6181
          \l_tmpa_tl
       }
6182
6183
      \clist_set:No \l_tmpa_clist \sparagraphtype
6184
      \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\tl_to_str:n{symdoc}}
6185
6186
        \stex_reactivate_macro:N \definiendum
6187
6188
        \stex_reactivate_macro:N \definame
        \stex_reactivate_macro:N \Definame
        \stex_reactivate_macro:N \premise
        \stex_reactivate_macro:N \definiens
6192
      \str_if_empty:NTF \sparagraphid {
6193
        \str_if_empty:NTF \sparagraphname {
6194
          \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\tl_to_str:n{symdoc}}{
6195
            \stex_ref_new_doc_target:n {}
6196
6197
       } {
6198
          \stex_ref_new_doc_target:n {}
6199
     } {
        \stex_ref_new_doc_target:n \sparagraphid
6202
6203
```

```
6211
                             }
                       6212
                       6213
                             \stex_smsmode_do:
                       6214
                             \ignorespacesandpars
                       6215
                             \str_if_empty:NF \sparagraphname {
                       6216
                               \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sparagraphname}}
                       6217
                               \stex_ref_new_sym_target:n {\l_stex_current_module_str ? \sparagraphname}
                       6218
                       6219
                             \stex_if_smsmode:F {
                       6220
                               \clist_set:No \l_tmpa_clist \sparagraphtype
                       6221
                               \tl_clear:N \l_tmpa_tl
                               \clist_map_inline:Nn \l_tmpa_clist {
                                 \tl_if_exist:cT {__stex_statements_sparagraph_##1_end:}{
                                   \tl_set:Nn \l_tmpa_tl {\use:c{__stex_statements_sparagraph_##1_end:}}
                       6225
                       6226
                       6227
                               \tl_if_empty:NTF \l_tmpa_tl {
                       6228
                                 \__stex_statements_sparagraph_end:
                       6229
                               }{
                       6230
                       6231
                                 \l_tmpa_tl
                               }
                       6232
                       6233
                               \end{stex_annotate_env}
                             }
                       6234
                       6235 }
\stexpatchparagraph
                       6236
                           \cs_new_protected:Nn \__stex_statements_sparagraph_start: {
                       6237
                             \stex_par:\noindent\tl_if_empty:NTF \l_stex_sparagraph_start_tl {
                               \tl_if_empty:NF \l_stex_sparagraph_title_tl {
                       6239
                                 \titleemph{\l_stex_sparagraph_title_tl}:~
                       6240
                               }
                       6241
                             ትና
                       6242
                               \titleemph{\l_stex_sparagraph_start_tl}~
                       6243
                       6244
                       6245 }
                           \cs_new_protected:Nn \__stex_statements_sparagraph_end: {\stex_par:\medskip}
                       6246
                       6247
                           \newcommand\stexpatchparagraph[3][] {
                               \str_set:Nx \l_tmpa_str{ #1 }
                               \str_if_empty:NTF \l_tmpa_str {
                                 \tl_set:Nn \__stex_statements_sparagraph_start: { #2 }
                       6251
                                 \tl_set:Nn \__stex_statements_sparagraph_end: { #3 }
                       6252
                               }{
                       6253
                                 \exp_after:wN \tl_set:Nn \csname __stex_statements_sparagraph_#1_start:\endcsname{ #2
                       6254
                                 \exp_after:wN \tl_set:Nn \csname __stex_statements_sparagraph_#1_end:\endcsname{ #3 }
                       6255
```

\exp_args:NNx

}

\tl_if_empty:nF{ ##1 }{

\stex_get_symbol:n { ##1 }

6205

6206

6207

6208

6209 6210 \clist_if_in:NnT \l_tmpa_clist {\tl_to_str:n{symdoc}}{

\clist_map_inline: Nn \l__stex_statements_sparagraph_for_clist {

\stex_ref_new_sym_target:n \l_stex_get_symbol_uri_str

```
}
6256
6257
6258
    \keys_define:nn { stex / inlinepara} {
6259
              .str_set_x:N
                              = \sparagraphid ,
6260
              .str_set_x:N
                              = \sparagraphtype ,
     type
6261
              .clist_set:N
                              = \l_stex_statements_sparagraph_for_clist ,
6262
              .tl_set:N
                              = \sparagraphfrom ,
6263
     to
              .tl_set:N
                              = \sparagraphto ,
              .str_set:N
                              = \sparagraphname
     name
6265
6266 }
    \cs_new_protected:Nn \__stex_statements_inlinepara_args:n {
6267
      \tl_clear:N \sparagraphfrom
6268
      \tl_clear:N \sparagraphto
6269
      \str_clear:N \sparagraphid
6270
      \str_clear:N \sparagraphtype
6271
      \clist_clear:N \l__stex_statements_sparagraph_for_clist
6272
      \str_clear:N \sparagraphname
6273
      \keys_set:nn { stex / inlinepara }{ #1 }
6274
6275 }
   \NewDocumentCommand \inlinepara { O{} m } {
6276
6277
     \begingroup
      \__stex_statements_inlinepara_args:n{ #1 }
6278
     \clist_set:No \l_tmpa_clist \sparagraphtype
6279
      \str_if_empty:NTF \sparagraphid {
6280
        \str_if_empty:NTF \sparagraphname {
6281
          \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\tl_to_str:n{symdoc}}{
6282
            \stex_ref_new_doc_target:n {}
6283
6284
       } {
          \stex_ref_new_doc_target:n {}
6286
       }
6287
     } {
6288
        \stex_ref_new_doc_target:n \sparagraphid
6289
6290
      \stex_if_smsmode:TF{
6291
        \str_if_empty:NF \sparagraphname {
6292
          \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sparagraphname}}
6293
6294
          \stex_ref_new_sym_target:n {\l_stex_current_module_str ? \sparagraphname}
       }
     }{
        \seq_clear:N \l_tmpb_seq
6298
        \clist_map_inline:Nn \l__stex_statements_sparagraph_for_clist {
          \tl_if_empty:nF{ ##1 }{
6299
            \stex_get_symbol:n { ##1 }
6300
            \exp_args:NNo \seq_put_right:Nn \l_tmpb_seq {
6301
              \l_stex_get_symbol_uri_str
6302
6303
         }
6304
        }
6305
        \exp_args:Nnx
6307
        \stex_annotate:nnn{paragraph}{\seq_use:Nn \l_tmpb_seq {,}}{
6308
          \str_if_empty:NF \sparagraphtype {
            \stex_annotate_invisible:nnn{typestrings}{\sparagraphtype}{}
6309
```

```
6310
           \str_if_empty:NF \sparagraphfrom {
6311
             \stex_annotate_invisible:nnn{from}{\sparagraphfrom}{}
6312
6313
           \str_if_empty:NF \sparagraphto {
6314
             \stex_annotate_invisible:nnn{to}{\sparagraphto}{}
6315
6316
           \str_if_empty:NF \sparagraphname {
6317
             \stex_suppress_html:n{\stex_symdecl_do:nn{}{\sparagraphname}}
             \verb|\statementname|{\statementname}|{\statementname}| \\
6319
             \stex_ref_new_sym_target:n {\l_stex_current_module_str ? \sparagraphname}
6320
          }
6321
           \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\tl_to_str:n{symdoc}}{
6322
             \clist_map_inline:Nn \l_tmpb_seq {
6323
               \stex_ref_new_sym_target:n {##1}
6324
6325
          }
6326
          #2
6327
        }
      \endgroup
6330
      \stex_smsmode_do:
6331
6332 }
6333
(End definition for \stexpatchparagraph. This function is documented on page 50.)
6334 //package>
```

Chapter 33

The Implementation

33.1 Proofs

We first define some keys for the **proof** environment.

```
6340 \keys_define:nn { stex / spf } {
                 .str_set_x:N = \spfid,
6341
     for
                 .clist_set:N = \l__stex_sproof_spf_for_clist ,
     from
                .tl_set:N
                               = \l_stex_sproof_spf_from_tl ,
     proofend .tl_set:N
                                = \l_stex_sproof_spf_proofend_tl,
     type
               .str_set_x:N = \spftype,
                                = \spftitle,
6346
     title
                 .tl\_set:N
                                = \l__stex_sproof_spf_continues_tl,
     continues
                .tl_set:N
6347
                .tl_set:N
                               = \l_stex_sproof_spf_functions_tl,
     functions
6348
                .tl_set:N
     term
                                = \l__stex_sproof_spf_term_tl,
6349
                                = \l_stex_sproof_spf_method_tl,
     method
                 .tl_set:N
6350
                 .bool_set:N = \l__stex_sproof_spf_hide_bool
6351
6352 }
6353 \cs_new_protected:Nn \__stex_sproof_spf_args:n {
6354 \str_clear:N \spfid
6355 \tl_clear:N \l__stex_sproof_spf_for_tl
6356 \tl_clear:N \l__stex_sproof_spf_from_tl
6357 \tl_set:Nn \l__stex_sproof_spf_proofend_tl {\sproof@box}
6358 \str_clear:N \spftype
6359 \tl_clear:N \spftitle
6360 \tl_clear:N \l__stex_sproof_spf_continues_tl
6361 \tl_clear:N \l__stex_sproof_spf_term_tl
6362 \tl_clear:N \l__stex_sproof_spf_functions_tl
6363 \tl_clear:N \l__stex_sproof_spf_method_tl
     \bool_set_false:N \l__stex_sproof_spf_hide_bool
6365 \keys_set:nn { stex / spf }{ #1 }
6367 \bool_set_true:N \l__stex_sproof_inc_counter_bool
```

\c__stex_sproof_flow_str

We define this macro, so that we can test whether the display key has the value flow \$\str_set:Nn\c_stex_sproof_flow_str{inline}\$

(End definition for \c__stex_sproof_flow_str.)

For proofs, we will have to have deeply nested structures of enumerated list-like environments. However, LATEX 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 accommodate semantic information.

```
\intarray_new:Nn\l__stex_sproof_counter_intarray{50}
    \cs_new_protected:Npn \sproofnumber {
      \int_set:Nn \l_tmpa_int {1}
6371
6372
      \bool_while_do:nn {
6373
        \int_compare_p:nNn {
          \intarray_item: Nn \l__stex_sproof_counter_intarray \l_tmpa_int
6374
        } > 0
6375
6376
        \intarray_item: Nn \l__stex_sproof_counter_intarray \l_tmpa_int .
6377
        \int_incr:N \l_tmpa_int
6378
6379
   }
6380
    \cs_new_protected:Npn \__stex_sproof_inc_counter: {
      \int_set:Nn \l_tmpa_int {1}
6382
      \bool_while_do:nn {
6383
        \int_compare_p:nNn {
6384
          \intarray_item:Nn \l__stex_sproof_counter_intarray \l_tmpa_int
6385
        } > 0
6386
     }{
6387
        \int_incr:N \l_tmpa_int
6388
6389
      \int_compare:nNnF \l_tmpa_int = 1 {
        \int_decr:N \l_tmpa_int
6391
6392
      \intarray_gset:Nnn \l__stex_sproof_counter_intarray \l_tmpa_int {
6393
        \intarray_item: Nn \l__stex_sproof_counter_intarray \l_tmpa_int + 1
6394
     }
6395
6396
6397
    \cs_new_protected:Npn \__stex_sproof_add_counter: {
6398
      \int_set:Nn \l_tmpa_int {1}
6399
      \bool_while_do:nn {
6400
        \int_compare_p:nNn {
          \intarray_item: Nn \l__stex_sproof_counter_intarray \l_tmpa_int
        } > 0
6403
     }{
        \int_incr:N \l_tmpa_int
6405
6406
      \intarray_gset:Nnn \l_stex_sproof_counter_intarray \l_tmpa_int { 1 }
6407
6408 }
6409
```

```
\cs_new_protected:Npn \__stex_sproof_remove_counter: {
                \int_set:Nn \l_tmpa_int {1}
           6411
                \bool_while_do:nn {
           6412
                   \int_compare_p:nNn {
           6413
                     \intarray_item: Nn \l__stex_sproof_counter_intarray \l_tmpa_int
           6414
           6415
                }{
           6416
                   \int_incr:N \l_tmpa_int
           6417
           6418
                \int_decr:N \l_tmpa_int
           6419
                \intarray_gset:Nnn \l__stex_sproof_counter_intarray \l_tmpa_int { 0 }
           6420
           6421
          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
              \def\sproof@box{
                \hbox{\vrule\vbox{\hrule width 6 pt\vskip 6pt\hrule}\vrule}
           6423
           6424 }
              \def\sproofend{
                \tl_if_empty:NF \l__stex_sproof_spf_proofend_tl {
                   6428
                }
           6429 }
          (End definition for \sproofend. This function is documented on page 50.)
spf@*@kw
           6430 \def\spf@proofsketch@kw{Proof~Sketch}
           6431 \def\spf@proof@kw{Proof}
           6432 \def\spf@step@kw{Step}
          (End definition for spf@*@kw. This function is documented on page ??.)
               For the other languages, we set up triggers
              \AddToHook{begindocument}{
                \ltx@ifpackageloaded{babel}{
           6434
                   \makeatletter
           6435
                   \clist_set:Nx \l_tmpa_clist {\bbl@loaded}
           6437
                   \clist_if_in:NnT \l_tmpa_clist {ngerman}{
                     \input{sproof-ngerman.ldf}
           6438
                  }
           6439
                   \clist_if_in:NnT \l_tmpa_clist {finnish}{
           6440
                     \input{sproof-finnish.ldf}
           6441
           6442
                   \clist_if_in:NnT \l_tmpa_clist {french}{
           6443
                     \input{sproof-french.ldf}
           6444
           6445
                   \clist_if_in:NnT \l_tmpa_clist {russian}{
                     \input{sproof-russian.ldf}
                  }
           6448
                   \makeatother
           6449
                }{}
           6450
           6451 }
```

spfsketch

6452 \newcommand\spfsketch[2][]{

```
\begingroup
                           6453
                                  \let \premise \stex_proof_premise:
                           6454
                                  \__stex_sproof_spf_args:n{#1}
                           6455
                                  \stex_if_smsmode:TF {
                           6456
                                    \str_if_empty:NF \spfid {
                           6457
                           6458
                                      \stex_ref_new_doc_target:n \spfid
                                    }
                                 }{
                           6460
                                    \seq_clear:N \l_tmpa_seq
                           6461
                                    \clist_map_inline:Nn \l__stex_sproof_spf_for_clist {
                           6462
                                      \tl_if_empty:nF{ ##1 }{
                           6463
                                        \stex_get_symbol:n { ##1 }
                           6464
                                        \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq {
                           6465
                                           \l_stex_get_symbol_uri_str
                           6466
                           6467
                                      }
                           6468
                                    }
                                    \exp_args:Nnx
                           6471
                                    \stex_annotate:nnn{proofsketch}{\seq_use:Nn \l_tmpa_seq {,}}{
                           6472
                                      \str_if_empty:NF \spftype {
                                        \stex_annotate_invisible:nnn{type}{\spftype}{}
                           6473
                           6474
                                      \clist_set:No \l_tmpa_clist \spftype
                           6475
                                      \tl_set:Nn \l_tmpa_tl {
                           6476
                                        \titleemph{
                           6477
                                           \tl_if_empty:NTF \spftitle {
                           6478
                                             \spf@proofsketch@kw
                           6479
                                          }{
                                             \spftitle
                                           }
                           6482
                                        }:~
                           6483
                                      }
                           6484
                                      \clist_map_inline:Nn \l_tmpa_clist {
                           6485
                                        \exp_args:No \str_if_eq:nnT \c__stex_sproof_flow_str {##1} {
                           6486
                                           \tl_clear:N \l_tmpa_tl
                           6487
                                        }
                           6488
                                      }
                           6489
                                      \str_if_empty:NF \spfid {
                                        \stex_ref_new_doc_target:n \spfid
                                      \l_tmpa_tl #2 \sproofend
                           6493
                                    }
                           6494
                                 }
                           6495
                                  \endgroup
                           6496
                                  \stex_smsmode_do:
                           6497
                           6498 }
                           (End definition for spfsketch. This function is documented on page 49.)
  \ stex sproof maybe comment:
\ stex sproof maybe comment end:
                           6500 \bool_set_false:N \l__stex_sproof_in_spfblock_bool
  \_stex_sproof_start_comment:
```

```
6501
                        \cs_new_protected:Nn \__stex_sproof_maybe_comment: {
                    6502
                          \bool_if:NF \l__stex_sproof_in_spfblock_bool {
                    6503
                            \par \setbox \l_tmpa_box \vbox \bgroup \everypar{\__stex_sproof_start_comment:}
                    6504
                    6505
                    6506
                        \cs_new_protected:Nn \__stex_sproof_maybe_comment_end: {
                    6507
                          \bool_if:NF \l__stex_sproof_in_spfblock_bool { \egroup }
                    6508
                        \cs_new_protected: Nn \__stex_sproof_start_comment: {
                          \csname @ @ par\endcsname\egroup\item[]\bgroup\stexcommentfont
                    6511
                    6512
                    6513
                   (End definition for \__stex_sproof_maybe_comment:, \__stex_sproof_maybe_comment_end:, and \__-
                   \verb|stex_sproof_start_comment:.||
\stexcommentfont
                    6514 \cs_new_protected:Npn \stexcommentfont {
                    6515
                          \small\itshape
                    6516 }
                   (End definition for \stexcommentfont. This function is documented on page ??.)
                   In this environment, we initialize the proof depth counter \count10 to 10, and set up
           sproof
                   the description environment that will take the proof steps. At the end of the proof, we
                   position the proof end into the last line.
                        \cs_new_protected:\n\__stex_sproof_start_env:nnn {
                    6517
                    6518
                          \seq_clear:N \l_tmpa_seq
                    6519
                          \clist_map_inline:Nn \l__stex_sproof_spf_for_clist {
                            \tl_if_empty:nF{ ##1 }{
                    6520
                              \stex_get_symbol:n { ##1 }
                              \exp_args:NNo \seq_put_right:Nn \l_tmpa_seq {
                                \l_stex_get_symbol_uri_str
                    6523
                    6524
                            }
                    6525
                          }
                    6526
                          \exp_args:Nnnx
                    6527
                          \begin{stex_annotate_env}{#1}{\seq_use:Nn \l_tmpa_seq {,}}
                    6528
                          \str_if_empty:NF \spftype {
                    6529
                            \stex_annotate_invisible:nnn{type}{\spftype}{}
                    6530
                    6531
                    6532
                          #3 {~\stex_annotate:nnn{spftitle}{}{#2}}
                    6533
                          \str_if_empty:NF \spfid {
                    6534
                            \stex_ref_new_doc_target:n \spfid
                    6535
                          \begin{stex_annotate_env}{spfbody}{\bool_if:NTF \l__stex_sproof_spf_hide_bool {false}{true}
                    6536
                          \bool_if:NT \l__stex_sproof_spf_hide_bool{
                    6537
                            \stex_html_backend:F{\setbox\l_tmpa_box\vbox\bgroup}
                    6538
                    6539
                          \begin{list}{}{
                            \setlength\topsep{0pt}
                    6541
                            \setlength\parsep{0pt}
                    6542
                            \setlength\rightmargin{0pt}
```

```
6544
6545
     }\__stex_sproof_maybe_comment:
6546
    \cs_new_protected:Nn \__stex_sproof_end_env:n {
6547
      \stex_if_smsmode:F{
6548
        \__stex_sproof_maybe_comment_end:
6549
        \end{list}
6550
        \bool_if:NT \l__stex_sproof_spf_hide_bool{
6551
          \stex_html_backend:F{\egroup}
6553
        \clist_set:No \l_tmpa_clist \spftype
6554
       #1
6555
        \end{stex_annotate_env}
6556
        \end{stex_annotate_env}
6557
6558
6559
    \NewDocumentEnvironment{sproof}{s O{} m}{
6560
     \intarray_gzero:N \l__stex_sproof_counter_intarray
6561
      \intarray_gset:Nnn \l__stex_sproof_counter_intarray 1 1
      \stex_reactivate_macro:N \yield
      \stex_reactivate_macro:N \eqstep
      \stex_reactivate_macro:N \assumption
6565
      \stex_reactivate_macro:N \conclude
6566
      \stex_reactivate_macro:N \spfstep
6567
      \__stex_sproof_spf_args:n{#2}
6568
      \stex_if_smsmode:TF {
6569
        \str_if_empty:NF \spfid {
6570
          \stex_ref_new_doc_target:n \spfid
6571
       }
6572
     }{
6573
        \__stex_sproof_start_env:nnn{sproof}{#3}{
6574
          \clist_set:No \l_tmpa_clist \spftype
6575
          \tl_clear:N \l_tmpa_tl
6576
          \clist_map_inline:Nn \l_tmpa_clist {
6577
            \tl_if_exist:cT {__stex_sproof_sproof_##1_start:}{
6578
              \tl_set:Nn \l_tmpa_tl {\use:c{__stex_sproof_sproof_##1_start:}}
6579
6580
            \exp_args:No \str_if_eq:nnT \c__stex_sproof_flow_str {##1} {
6581
6582
              \tl_set:Nn \l_tmpa_tl {\use:n{}}
          }
          \tl_if_empty:NTF \l_tmpa_tl {
            \__stex_sproof_sproof_start:
6586
          }{
6587
            \l_tmpa_tl
6588
6589
       }
6590
6591
      \stex_smsmode_do:
6592
   }{\__stex_sproof_end_env:n{
6593
     \tl_clear:N \l_tmpa_tl
6595
      \clist_map_inline:Nn \l_tmpa_clist {
        \tl_if_exist:cT {__stex_sproof_sproof_##1_end:}{
6596
          \tl_set:Nn \l_tmpa_tl {\use:c{__stex_sproof_sproof_##1_end:}}
6597
```

```
}
              6599
                    \tl_if_empty:NTF \l_tmpa_tl {
              6600
                      \__stex_sproof_sproof_end:
              6601
              6602
                      \l_tmpa_tl
              6603
              6604
                 }}
              6605
                  \NewDocumentEnvironment{subproof}{s O{} m}{
                    \__stex_sproof_spf_args:n{#2}
              6607
                    \stex_if_smsmode:TF {
                      \str_if_empty:NF \spfid {
              6609
                        \stex_ref_new_doc_target:n \spfid
              6610
              6611
              6612
                        _stex_sproof_start_env:nnn{subproof}{\item[\sproofnumber]\ignorespacesandpars #3}{}
              6613
              6614
                    \__stex_sproof_add_counter:
              6615
                    \stex_smsmode_do:
                   {\__stex_sproof_remove_counter:\__stex_sproof_end_env:n{}
                    \bool_if:NT \l__stex_sproof_inc_counter_bool {
                      \_\_stex_sproof_inc_counter:
              6619
              6620
              6621
                    \aftergroup\__stex_sproof_maybe_comment:
              6622 }
                  \AddToHook{env/subproof/before}{\__stex_sproof_maybe_comment_end:}
              6623
              6624
                  \cs_new_protected:Nn \__stex_sproof_sproof_start: {
              6625
                    \par\noindent\titleemph{
              6626
                      \tl_if_empty:NTF \spftype {
                        \spf@proof@kw
                     }{
              6630
                        \spftype
                     }
              6631
                   }:
              6632
              6633
                  \cs_new_protected:Nn \__stex_sproof_sproof_end: {\sproofend}
              6634
              6635
              6636
                  \newcommand\stexpatchproof[3][] {
                    \str_set:Nx \l_tmpa_str{ #1 }
                    \str_if_empty:NTF \l_tmpa_str {
                      \tl_set:Nn \__stex_sproof_sproof_start: { #2 }
              6640
                      \tl_set:Nn \__stex_sproof_sproof_end: { #3 }
                   }{
              6641
                      \exp_after:wN \tl_set:Nn \csname __stex_sproof_sproof_#1_start:\endcsname{ #2 }
              6642
                      \exp_after:wN \tl_set:Nn \csname __stex_sproof_sproof_#1_end:\endcsname{ #3 }
              6643
              6644
              6645 }
     \pstep
  \conclude
\assumption
                  \keys_define:nn { stex / spfsteps } {
              6647
                                .str_set_x:N = \spfstepid,
      \have
                   id
              6648
                                for
    \eqstep
              6649
```

```
6650
     type
                   .str_set_x:N = \spftype,
                                 = \spftitle,
                   .tl_set:N
6651
     title
                                 = \l__stex_sproof_spf_method_tl,
                   .tl set:N
6652
     method
                   .tl_set:N
                                 = \l_stex_sproof_spf_term_tl
6653
     term
6654 }
    \cs_new_protected:Nn \__stex_sproof_spfstep_args:n {
6655
    \str_clear:N \spfstepid
   \clist_clear:N \l__stex_sproof_spf_for_clist
   \str_clear:N \spftype
   \tl_clear:N \l__stex_sproof_spf_method_tl
   \tl_clear:N \l__stex_sproof_spf_term_tl
     %\bool_set_false:N \l__stex_sproof_inc_counter_bool
   \keys_set:nn { stex / spfsteps }{ #1 }
6662
6663
6664
    \cs_new_protected:Nn \__stex_sproof_make_step_macro:Nnnnn {
6665
      \NewDocumentCommand #1 {s O{} +m} {
6666
        \__stex_sproof_maybe_comment_end:
6667
        \__stex_sproof_spfstep_args:n{##2}
        \stex_annotate:nnn{spfstep}{#2}{
          \tl_if_empty:NF \l__stex_sproof_spf_term_tl {
6671
            \stex_annotate_invisible:nnn{spfyield}{}\$\l__stex_sproof_spf_term_tl$}
6672
6673
          \bool_if:NTF \l__stex_sproof_in_spfblock_bool {
6674
            #4
6675
          }{
6676
            \item[\IfBooleanTF ##1 {}{#3}]
6677
          }
6678
          \ignorespacesandpars ##3
6680
        \bool_if:NF \l__stex_sproof_in_spfblock_bool { \IfBooleanTF ##1 {}{ #5 } }
6681
6682
        \__stex_sproof_maybe_comment:
6683
      \stex_deactivate_macro:Nn #1 {sproof~environments}
6684
6685
6686
    \__stex_sproof_make_step_macro:Nnnnn \assumption {assumption} \sproofnumber {} \__stex_sproo
6687
    \__stex_sproof_make_step_macro:Nnnnn \conclude {conclusion} {$\Rightarrow$} {} {}
6688
    \__stex_sproof_make_step_macro:Nnnnn \spfstep {} \sproofnumber {} \__stex_sproof_inc_counter
    \NewDocumentCommand \eqstep {s m}{
      \__stex_sproof_maybe_comment_end:
     \bool_if:NTF \l__stex_sproof_in_spfblock_bool {
6693
        $=$
6694
     }{
6695
        \item[$=$]
6696
6697
     $\stex_annotate:nnn{spfstep}{eq}{ #2 }$
6698
      \__stex_sproof_maybe_comment:
6699
6701
   \stex_deactivate_macro:Nn \eqstep {sproof~environments}
6702
   \NewDocumentCommand \yield {+m}{
```

```
\stex_annotate:nnn{spfyield}{}{ #1 }
           6705 }
               \stex_deactivate_macro:Nn \yield {sproof~environments}
            6706
            6707
               \NewDocumentEnvironment{spfblock}{}{
            6708
                  \item[]
            6709
                  \bool_set_true:N \l__stex_sproof_in_spfblock_bool
            6710
            6711 }{
                  \aftergroup\__stex_sproof_maybe_comment:
           6713
               \AddToHook{env/spfblock/before}{\__stex_sproof_maybe_comment_end:}
           6714
           6715
           (End definition for \pstep and others. These functions are documented on page ??.)
\spfidea
            _{6716} \NewDocumentCommand\spfidea{0{} +m}{
                 \__stex_sproof_spf_args:n{#1}
           6717
                 \titleemph{
           6718
                    \tl_if_empty:NTF \spftype {Proof~Idea}{
            6719
                      \spftype
           6720
                    }:
            6721
            6722
                 }~#2
           6723
                  \sproofend
            6724 }
           (End definition for \spfidea. This function is documented on page 49.)
            6725 \newcommand\spfjust[1]{
           6726
            6727 }
            6728 (/package)
                Some auxiliary code, and clean up to be executed at the end of the package.
```

STEX -Others Implementation

```
6729 (*package)
6730
others.dtx
                                  <@@=stex_others>
    Warnings and error messages
      % None
Math subject classifier
6735 \NewDocumentCommand \MSC {m} {
      % TODO
6737 }
(End definition for \MSC. This function is documented on page ??.)
    Patching tikzinput, if loaded
6738 \@ifpackageloaded{tikzinput}{
      \RequirePackage{stex-tikzinput}
    \bool_if:NT \c_stex_persist_mode_bool {
      \let\__stex_notation_restore_notation_old:nnnnn
        \__stex_notation_restore_notation:nnnnn
      \def\__stex_notation_restore_notation_new:nnnnn#1#2#3#4#5{
6745
        \__stex_notation_restore_notation_old:nnnnn{#1}{#2}{#3}{#4}{#5}
6746
        \ExplSyntaxOn
6747
6748
      \def\__stex_notation_restore_notation:nnnnn{
6749
        \ExplSyntaxOff
        \catcode'~10
        \__stex_notation_restore_notation_new:nnnnn
6753
      \input{\jobname.sms}
6754
      \let\__stex_notation_restore_notation:nnnnn
6755
        \__stex_notation_restore_notation_old:nnnnn
6756
      \prop_if_exist:NT\c_stex_mathhub_main_manifest_prop{
```

% dummy variable

STEX

-Metatheory Implementation

```
6768 (*package)
   <@@=stex_modules>
6769
6770
metatheory.dtx
                                  6773 \str_const:Nn \c_stex_metatheory_ns_str {http://mathhub.info/sTeX/meta}
6774 \begingroup
6775 \stex_module_setup:nn{
    ns=\c_stex_metatheory_ns_str,
    meta=NONE
6778 }{Metatheory}
6779 \stex_reactivate_macro:N \symdecl
6780 \stex_reactivate_macro:N \notation
6781 \stex_reactivate_macro:N \symdef
6782 \ExplSyntaxOff
   \csname stex_suppress_html:n\endcsname{
     % is-a (a:A, a \in A, a is an A, etc.)
     \symdecl{isa}[args=ai]
     \notation{isa}[typed,op=:]{#1 \comp{:} #2}{##1 \comp, ##2}
     \notation{isa}[in]{#1 \comp\in #2}{##1 \comp, ##2}
     \notation{isa}[pred]{#2\\comp(#1 \comp)}{##1 \comp, ##2}
6788
6789
     % bind (\forall, \Pi, \lambda etc.)
6790
     \symdecl{bind}[args=Bi,assoc=pre]
6791
     \notation{bind}[depfun,prec=nobrackets,op={(\cdot)\;\cdot}]{\comp( #1 \comp{)\;\to\;}
6792
     \notation{bind}[forall]{\comp\forall #1.\;#2}{##1 \comp, ##2}
     \notation{bind}[Pi]{\comp\prod_{#1}#2}{##1 \comp, ##2}
     % implicit bind
     \symdecl{implicitbind}[args=Bi,assoc=pre]
6797
     6798
     \notation{implicitbind}[depfun,prec=nobrackets]{\comp( #1 \comp{)\;\to_I\;} #2}{##1 \comp,
6799
     \notation{implicitbind}[Pi]{\comp\prod^I_{#1}#2}{##1\comp,##2}
6800
6801
```

```
\symdecl{dummyvar}
     \notation{dummyvar}[underscore]{\comp\_}
     \notation{dummyvar}[dot]{\comp\cdot}
     \notation{dummyvar}[dash]{\comp{{\rm --}}}
6806
6807
     %fromto (function space, Hom-set, implication etc.)
6808
     \symdecl{fromto}[args=ai]
6809
     \notation{fromto}[xarrow]{#1 \comp\to #2}{##1 \comp\times ##2}
6810
     \notation{fromto}[arrow]{#1 \comp\to #2}{##1 \comp\to ##2}
6811
6812
     % mapto (lambda etc.)
6813
     %\symdecl{mapto}[args=Bi]
6814
     %\notation{mapto}[mapsto]{#1 \comp\mapsto #2}{#1 \comp, #2}
6815
     %\notation{mapto}[lambda]{\comp\lambda #1 \comp.\; #2}{#1 \comp, #2}
6816
     %\notation{mapto}[lambdau]{\comp\lambda_{#1} \comp.\; #2}{#1 \comp, #2}
6817
6818
     % function/operator application
6819
     \symdecl{apply}[args=ia]
6820
     \notation{apply}[prec=0;0x\infprec,parens,op=\cdot(\cdot)]{#1 \comp( #2 \comp)}{##1 \comp,
     \notation{apply}[prec=0;0x\infprec,lambda]{#1 \; #2 }{##1 \; ##2}
     % collection of propositions/booleans/truth values
6824
     \symdecl{prop}[name=proposition]
6825
     \notation{prop}[prop]{\comp{{\rm prop}}}}
6826
     \notation{prop}[BOOL]{\comp{{\rm BOOL}}}
6827
6828
     \symdecl{judgmentholds}[args=1]
6829
     \notation{judgmentholds}[vdash,op=\vdash]{\comp\vdash\; #1}
6830
6831
6832
     % sequences
     \symdecl{seqtype}[args=1]
6833
     \notation{seqtype}[kleene]{#1^{\comp\ast}}
6834
6835
     \symdecl{seqexpr}[args=a]
6836
     \notation{seqexpr}[angle,prec=nobrackets]{\comp\langle #1\comp\rangle}{##1\comp,##2}
6837
6838
     \symdef{seqmap}[args=abi,setlike]{\comp\{#3 \comp| #2\comp\in \dobrackets{#1} \comp\}}{##1
6839
     \symdef{seqprepend}[args=ia]{#1 \comp{::} #2}{##1 \comp, ##2}
6840
     \symdef{seqappend}[args=ai]{#1 \comp{::} #2}{##1 \comp, ##2}
     \symdef{seqfoldleft}[args=iabbi]{ \comp{foldl}\dobrackets{#1,#2}\dobrackets{#3\comp,#4\com
     symdef{seqfoldright}[args=iabbi,op=foldr]{ \comp{foldr}\dobrackets{#1,#2}\dobrackets{#3\c
     \symdef{seqhead}[args=a]{\comp{head}\dobrackets{#1}}{##1 \comp, ##2}
     \symdef{seqtail}[args=a]{\comp{tail}\dobrackets{#1}}{##1 \comp, ##2}
6845
     \symdef{seqlast}[args=a]{\comp{last}\dobrackets{#1}}{##1 \comp, ##2}
6846
     \symdef{seqinit}[args=a]{\comp{tail}\dobrackets{#1}}{##1 \comp, ##2}
6847
6848
     \symdef{sequence-index}[args=2,li,prec=nobrackets]{{#1}_{#2}}
6849
     \notation{sequence-index}[ui,prec=nobrackets]{{#1}^{#2}}
6850
6851
6852
     \symdef{aseqdots}[args=a,prec=nobrackets]{#1\comp{,\ellipses}}{##1\comp,##2}
6853
     \symdef{aseqfromto}[args=ai,prec=nobrackets]{#1\comp{,\ellipses,}#2}{##1\comp,##2}
6854
     \symdef{aseqfromtovia}[args=aii,prec=nobrackets]{#1\comp{,\ellipses,}#2\comp{,\ellipses,}#
6855
```

% nat literals

6856

```
\symdef{natliteral}{\comp{\mathtt{Ord}}}
6857
6858
     % letin (''let'', local definitions, variable substitution)
6859
     \symdecl{letin}[args=bii]
6860
     \notation{letin}[let]_{\comp{{\rm let}}\; \#1\comp{=} \#2\; \comp{{\rm in}}\; \#3}
6861
     \notation{letin}[subst]{#3 \comp[ #1 \comp/ #2 \comp]}
6862
     \notation{letin}[frac]{#3 \comp[ \frac{#2}{#1} \comp]}
6863
6864
     % structures
     \symdecl*{module-type}[args=1]
     \notation{module-type}{\comp{\mathtt{MOD}}} #1}
6867
     \symdecl{mathstruct}[name=mathematical-structure,args=a] % TODO
6868
     \notation{mathstruct}[angle,prec=nobrackets]{\comp\langle #1 \comp\rangle}{##1 \comp, ##2}
6869
6870
     % objects
6871
     \symdecl{object}
6872
     \notation{object}{\comp{\mathtt{OBJECT}}}
6873
6874
6875 }
   % The following are abbreviations in the sTeX corpus that are left over from earlier
6877
   \mbox{\ensuremath{\mbox{\%}}}\xspace developments. They will eventually be phased out.
6878
6879
     \ExplSyntaxOn
6880
     \stex_add_to_current_module:n{
6881
       6882
       \def\nappui#1#2#3#4{\apply{#1}{\nasequi{#2}{#3}{#4}}}
6883
       \def\livar{\csname sequence-index\endcsname[li]}
6884
       \def\uivar{\csname sequence-index\endcsname[ui]}
6885
       \label{livar} $$ \left( \frac{1}{\#2} \right)^{\#1}{\#3}} 
       \def\nasequi#1#2#3{\aseqfromto{\uivar{#1}{#2}}{\uivar{#1}{#3}}}
6887
6889 \__stex_modules_end_module:
6890 \endgroup
6891 (/package)
```

Tikzinput Implementation

```
<@@=tikzinput>
   \langle *package \rangle
6894
tikzinput.dtx
                                     \ProvidesExplPackage{tikzinput}{2022/08/08}{3.2.0}{tikzinput package}
   \RequirePackage{13keys2e}
6899
   \keys_define:nn { tikzinput } {
            .bool_set:N = \c_tikzinput_image_bool,
            .default:n
                            = false ,
     unknown .code:n
                              = {}
6904 }
6905
   \ProcessKeysOptions { tikzinput }
6906
6907
   \bool_if:NTF \c_tikzinput_image_bool {
6908
     \RequirePackage{graphicx}
6909
6910
     \providecommand\usetikzlibrary[]{}
6911
     \newcommand\tikzinput[2][]{\includegraphics[#1]{#2}}
6913 }{
     \RequirePackage{tikz}
6914
     \RequirePackage{standalone}
6915
6916
     \newcommand \tikzinput [2] [] {
6917
       \setkeys{Gin}{#1}
6918
       \ifx \Gin@ewidth \Gin@exclamation
6919
         \ifx \Gin@eheight \Gin@exclamation
6920
           \input { #2 }
6921
         \else
           \resizebox{!}{ \Gin@eheight }{
              \input { #2 }
           }
6925
         \fi
6926
       \else
6927
         \ifx \Gin@eheight \Gin@exclamation
6928
           \resizebox{ \Gin@ewidth }{!}{
6929
```

```
\input { #2 }
6930
                           }
6931
                       \else
6932
                            \resizebox{ \Gin@ewidth }{ \Gin@eheight }{
6933
                                  \input { #2 }
6934
6935
                      \fi
6936
                  \fi
6937
             }
6938
6939
6940
         \newcommand \ctikzinput [2] [] {
6941
             \begin{center}
6942
                  \tikzinput [#1] {#2}
6943
             \end{center}
6944
6945
6946
        \@ifpackageloaded{stex}{
             \RequirePackage{stex-tikzinput}
6949 }{}
        ⟨/package⟩
6951
        ⟨*stex⟩
6952
        \ProvidesExplPackage{stex-tikzinput}{2022/08/08}{3.2.0}{stex-tikzinput}
        \RequirePackage{stex}
        \RequirePackage{tikzinput}
6956
         \newcommand\mhtikzinput[2][]{%
6957
             \def\Gin@mhrepos{}\setkeys{Gin}{#1}%
6958
             \stex_in_repository:nn\Gin@mhrepos{
6959
                  \tikzinput[#1]{\mhpath{##1}{#2}}
6960
6961
6962
         \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
6963
         \cs_new_protected:Nn \__tikzinput_usetikzlibrary:nn {
             \pgfkeys@spdef\pgf@temp{#1}
             \expandafter\ifx\csname tikz@library@\pgf@temp @loaded\endcsname\relax%
             \verb|\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\e
             \expandafter\edef\csname tikz@library@#1@atcode\endcsname{\the\catcode'\@}
             \expandafter\edef\csname tikz@library@#1@barcode\endcsname{\the\catcode'\|}
6970
             \expandafter\edef\csname tikz@library@#1@dollarcode\endcsname{\the\catcode'\$}
6971
             \catcode'\@=11
6972
             \catcode'\|=12
6973
             \catcode'\$=3
6974
             \pgfutil@InputIfFileExists{#2}{}{}
             \catcode'\@=\csname tikz@library@#1@atcode\endcsname
6977
             \catcode'\|=\csname tikz@library@#1@barcode\endcsname
             \catcode'\$=\csname tikz@library@#1@dollarcode\endcsname
6978
6979
6980
6981
       \newcommand\libusetikzlibrary[1]{
```

```
\prop_if_exist:NF \l_stex_current_repository_prop {
6983
       \msg_error:nnn{stex}{error/notinarchive}\libusetikzlibrary
6984
6985
     \prop_get:NnNF \l_stex_current_repository_prop {id} \l_tmpa_str {
6986
        \msg_error:nnn{stex}{error/notinarchive}\libusetikzlibrary
6987
6988
     \seq_clear:N \l__tikzinput_libinput_files_seq
6989
     \seq_set_eq:NN \l_tmpa_seq \c_stex_mathhub_seq
6990
     \seq_set_split:NnV \l_tmpb_seq / \l_tmpa_str
     \bool_while_do:nn { ! \seq_if_empty_p:N \l_tmpb_seq }{
6993
        \str_set:Nx \l_tmpa_str {\stex_path_to_string:N \l_tmpa_seq / meta-inf / lib / tikzlibra
6994
        \IfFileExists{ \l_tmpa_str }{
6995
          \seq_put_right:No \l__tikzinput_libinput_files_seq \l_tmpa_str
6996
6997
        \seq_pop_left:NN \l_tmpb_seq \l_tmpa_str
6998
        \seq_put_right:No \l_tmpa_seq \l_tmpa_str
6999
7000
     \str_set:Nx \l_tmpa_str {\stex_path_to_string:N \l_tmpa_seq / lib / tikzlibrary #1 .code.t
     \IfFileExists{ \l_tmpa_str }{
       \seq_put_right:No \l__tikzinput_libinput_files_seq \l_tmpa_str
7004
7005
7006
     \seq_if_empty:NTF \l__tikzinput_libinput_files_seq {
7007
        \msg_error:nnxx{stex}{error/nofile}{\exp_not:N\libusetikzlibrary}{tikzlibrary #1 .code.t
7008
7009
        \int_compare:nNnTF {\seq_count:N \l__tikzinput_libinput_files_seq} = 1 {
7010
          \seq_map_inline: Nn \l__tikzinput_libinput_files_seq {
7011
            \__tikzinput_usetikzlibrary:nn{#1}{ ##1 }
         }
7013
7014
          \msg_error:nnxx{stex}{error/twofiles}{\exp_not:N\libusetikzlibrary}{tikzlibrary #1 .cc
7015
7016
     }
7017
7018 }
7019 (/stex)
```

document-structure.sty Implementation

```
7020 (*package)
7021 (@@=document_structure)
7022 \ProvidesExplPackage{document-structure}{2022/08/08}{3.2.0}{Modular Document Structure}
7023 \RequirePackage{13keys2e}
```

37.1 Package Options

We declare some switches which will modify the behavior according to the package options. Generally, an option xxx will just set the appropriate switches to true (otherwise they stay false).

```
7024
7025 \keys_define:nn{ document-structure }{
     class .str_set_x:N = \c_document_structure_class_str,
                .str_set_x:N = \c_document_structure_topsect_str,
     unknown
                .code:n
                          = {
       \PassOptionsToClass{\CurrentOption}{stex}
       \PassOptionsToClass{\CurrentOption}{tikzinput}
7031
      showignores .bool_set:N
                               = \c_document_structure_showignores_bool,
7032 %
7033 }
7034 \ProcessKeysOptions{ document-structure }
   \str_if_empty:NT \c_document_structure_class_str {
7035
     \str_set:Nn \c_document_structure_class_str {article}
7036
   \str_if_empty:NT \c_document_structure_topsect_str {
     \str_set:Nn \c_document_structure_topsect_str {section}
7039
7040 }
```

Then we need to set up the packages by requiring the **sref** package to be loaded, and set up triggers for other languages

```
7041 \RequirePackage{xspace}
7042 \RequirePackage{comment}
7043 \RequirePackage{stex}
7044 \AddToHook{begindocument}{
```

\section@level

Finally, we set the \section@level macro that governs sectioning. The default is two (corresponding to the article class), then we set the defaults for the standard classes book and report and then we take care of the levels passed in via the topsect option.

```
\int_new:N \l_document_structure_section_level_int
   \str_case:VnF \c_document_structure_topsect_str {
7053
     {part}{
7054
        \int_set:Nn \l_document_structure_section_level_int {0}
7055
7056
     {chapter}{
7057
        \int_set:Nn \l_document_structure_section_level_int {1}
7058
7059
7060 }{
      \str_case:VnF \c_document_structure_class_str {
7061
7062
        {book}{
          \int_set:Nn \l_document_structure_section_level_int {0}
7063
7064
        {report}{
7065
          \int_set:Nn \l_document_structure_section_level_int {0}
7066
7067
7068
        \int_set:Nn \l_document_structure_section_level_int {2}
     }
7070
7071 }
```

37.2 Document Structure

The structure of the document is given by the sfragment environment. The hierarchy is adjusted automatically according to the LATEX class in effect.

\currentsectionlevel

EdN:9

For the \currentsectionlevel and \Currentsectionlevel macros we use an internal macro \current@section@level that only contains the keyword (no markup). We initialize it with "document" as a default. In the generated OMDoc, we only generate a text element of class omdoc_currentsectionlevel, wich will be instantiated by CSS later. 9

```
7072 \def\current@section@level{document}%
7073 \newcommand\currentsectionlevel{\lowercase\expandafter{\current@section@level}\xspace}%
7074 \newcommand\Currentsectionlevel{\expandafter\MakeUppercase\current@section@level\xspace}%
```

 $(End\ definition\ for\ \verb|\currentsection| evel.\ This\ function\ is\ documented\ on\ page\ {\bf 57.})$

\skipfragment

```
7075 \cs_new_protected:Npn \skipfragment {
```

 $^{^9{}m EDNote}$: MK: we may have to experiment with the more powerful uppercasing macro from mfirstuc.sty once we internationalize.

```
\ifcase\l_document_structure_section_level_int
                           \or\stepcounter{part}
                     7077
                           \or\stepcounter{chapter}
                     7078
                           \or\stepcounter{section}
                     7079
                           \or\stepcounter{subsection}
                     7080
                           \or\stepcounter{subsubsection}
                     7081
                           \or\stepcounter{paragraph}
                     7082
                           \or\stepcounter{subparagraph}
                           \fi
                     7085 }
                    (End definition for \skipfragment. This function is documented on page 56.)
   blindfragment
                         \newcommand\at@begin@blindsfragment[1]{}
                         \newenvironment{blindfragment}
                     7088 {
                           \int_incr:N\l_document_structure_section_level_int
                     7089
                           \at@begin@blindsfragment\l_document_structure_section_level_int
                     7090
                     7091 }{}
                    convenience macro: \sfragment@nonum{\langle level \rangle}{\langle title \rangle} makes an unnumbered section-
\sfragment@nonum
                    ing with title \langle title \rangle at level \langle level \rangle.
                     7092 \newcommand\sfragment@nonum[2]{
                           \ifx\hyper@anchor\@undefined\else\phantomsection\fi
                           \label{line-problem} $$ \addcontentsline{toc}{\#1}{\#2}\Onameuse{\#1}*{\#2}$
                     7095 }
                    (End definition for \sfragment@nonum. This function is documented on page ??.)
                    convenience macro: \sfragment@nonum{\langle level\rangle}{\langle title\rangle} makes numbered sectioning
  \sfragment@num
                    with title \langle title \rangle at level \langle level \rangle. We have to check the short key was given in the
                    sfragment environment and - if it is use it. But how to do that depends on whether
                    the rdfmeta package has been loaded. In the end we call \sref@label@id to enable
                    crossreferencing.
                         \newcommand\sfragment@num[2]{
                           \tl_if_empty:NTF \l__document_structure_sfragment_short_tl {
                     7097
                             \@nameuse{#1}{#2}
                     7098
                     7099
                             \cs_if_exist:NTF\rdfmeta@sectioning{
                     7100
                                \@nameuse{rdfmeta@#1@old}[\1__document_structure_sfragment_short_t1]{#2}
                     7101
                                \@nameuse{#1}[\l__document_structure_sfragment_short_tl]{#2}
                     7103
                           }
                     7106 %\sref@label@id@arg{\omdoc@sect@name~\@nameuse{the#1}}\sfragment@id
                    (End definition for \sfragment@num. This function is documented on page ??.)
        sfragment
                     7108 \keys_define:nn { document-structure / sfragment }{
                                           .str_set_x:N = \l__document_structure_sfragment_id_str,
                     7109
                                           .str_set_x:N = \l__document_structure_sfragment_date_str,
                           date
                     7110
```

```
.clist_set:N = \l__document_structure_sfragment_creators_clist,
     creators
                    .clist_set:N = \l__document_structure_sfragment_contributors_clist,
7112
     contributors
                                  = \l__document_structure_sfragment_srccite_tl,
                    .tl set:N
7113
     srccite
                    .tl_set:N
                                  = \l__document_structure_sfragment_type_tl,
7114
     type
     short
                    .tl_set:N
                                  = \l__document_structure_sfragment_short_tl,
7115
                                  = \l__document_structure_sfragment_intro_tl,
                    .tl_set:N
7116
                                  = \l__document_structure_sfragment_imports_tl,
     imports
                    .tl set:N
     loadmodules
                    .bool_set:N
                                 = \l__document_structure_sfragment_loadmodules_bool
7118
7119 }
    \cs_new_protected:Nn \__document_structure_sfragment_args:n {
7120
      \str_clear:N \l__document_structure_sfragment_id_str
7121
      \str_clear:N \l__document_structure_sfragment_date_str
      \clist_clear:N \l__document_structure_sfragment_creators_clist
7123
      \clist_clear:N \l__document_structure_sfragment_contributors_clist
7124
      \tl_clear:N \l__document_structure_sfragment_srccite_tl
7125
      \tl_clear:N \l__document_structure_sfragment_type_tl
7126
      \tl_clear:N \l__document_structure_sfragment_short_tl
      \tl_clear:N \l__document_structure_sfragment_imports_tl
7128
      \tl_clear:N \l__document_structure_sfragment_intro_tl
      \bool_set_false:N \l__document_structure_sfragment_loadmodules_bool
      \keys_set:nn { document-structure / sfragment } { #1 }
7131
7132 }
```

\at@begin@sfragment

we define a switch for numbering lines and a hook for the beginning of groups: The \at@begin@sfragment macro allows customization. It is run at the beginning of the sfragment, i.e. after the section heading.

```
7133 \newif\if@mainmatter\@mainmattertrue
7134 \newcommand\at@begin@sfragment[3][]{}
```

Then we define a helper macro that takes care of the sectioning magic. It comes with its own key/value interface for customization.

```
\keys_define:nn { document-structure / sectioning }{
              .str_set_x:N = \l__document_structure_sect_name_str
     name
7136
              .str_set_x:N = \l__document_structure_sect_ref_str
     ref
     clear
              .bool_set:N
                             = \l__document_structure_sect_clear_bool ,
7138
     clear
              .default:n
                             = {true}
7139
                             = \l__document_structure_sect_num_bool
              .bool_set:N
7140
              .default:n
                             = {true}
7141
7142 }
    \cs_new_protected:Nn \__document_structure_sect_args:n {
7143
      \str_clear:N \l__document_structure_sect_name_str
7144
      \str_clear:N \l__document_structure_sect_ref_str
7145
      \bool_set_false:N \l__document_structure_sect_clear_bool
7146
      \bool_set_false:N \l__document_structure_sect_num_bool
7147
      \keys_set:nn { document-structure / sectioning } { #1 }
7148
7149 }
    \newcommand\omdoc@sectioning[3][]{
7150
      \__document_structure_sect_args:n {#1 }
      \let\omdoc@sect@name\l__document_structure_sect_name_str
      \bool_if:NT \l__document_structure_sect_clear_bool { \cleardoublepage }
      \if@mainmatter% numbering not overridden by frontmatter, etc.
7154
        \bool_if:NTF \l__document_structure_sect_num_bool {
          \sfragment@num{#2}{#3}
7156
       }{
```

```
7158  \sfragment@nonum{#2}{#3}
7159  }
7160  \def\current@section@level{\omdoc@sect@name}
7161  \else
7162  \sfragment@nonum{#2}{#3}
7163  \fi
7164 }% if@mainmatter
```

and another one, if redefines the \addtocontentsline macro of LATEX to import the respective macros. It takes as an argument a list of module names.

```
7165 \newcommand\sfragment@redefine@addtocontents[1]{%
7166 %\edef\__document_structureimport{#1}%
7167 %\@for\@I:=\__document_structureimport\do{%
7168 %\edef\@path{\csname module@\@I @path\endcsname}%
7169 %\@ifundefined{tf@toc}\relax%
7170 % {\protected@write\tf@toc{}{\string\@requiremodules{\@path}}}
7171 %\ifx\hyper@anchor\@undefined% hyperref.sty loaded?
7172 %\def\addcontentsline##1##2##3{%
7173 %\addtocontents{##1}{\protect\contentsline{##2}{\string\withusedmodules{#1}{\#3}}{\thepage}}
7174 %\else% hyperref.sty not loaded
7175 %\def\addcontentsline##1##2##3{%
7176 %\addtocontents{##1}{\protect\contentsline{##2}{\string\withusedmodules{#1}{\#3}}{\thepage}}
7177 %\fi
7178 }\% hyperref.sty loaded?
```

now the sfragment environment itself. This takes care of the table of contents via the helper macro above and then selects the appropriate sectioning command from article.cls. It also registeres the current level of sfragments in the \sfragment@level counter.

```
7179 \newenvironment{sfragment}[2][]% keys, title
7180 {
7181 \__document_structure_sfragment_args:n { #1 }%\sref@target%
```

If the loadmodules key is set on \begin{sfragment}, we redefine the \addcontetsline macro that determines how the sectioning commands below construct the entries for the table of contents.

```
7182  \stex_csl_to_imports:No \usemodule \l__document_structure_sfragment_imports_tl
7183
7184  \bool_if:NT \l__document_structure_sfragment_loadmodules_bool {
7185   \sfragment@redefine@addtocontents{
7186    %\@ifundefined{module@id}\used@modules%
7187    %{\@ifundefined{module@idoule@id @path}{\used@modules}\module@id}
7188   }
7189 }
```

now we only need to construct the right sectioning depending on the value of \section@level.

```
7190
7191 \stex_document_title:n { #2 }
7192
7193 \int_incr:N\l_document_structure_section_level_int
7194 \ifcase\l_document_structure_section_level_int
7195 \or\omdoc@sectioning[name=\omdoc@part@kw,clear,num]{part}{#2}
7196 \or\omdoc@sectioning[name=\omdoc@chapter@kw,clear,num]{chapter}{#2}
7197 \or\omdoc@sectioning[name=\omdoc@section@kw,num]{section}{#2}
7198 \or\omdoc@sectioning[name=\omdoc@subsection@kw,num]{subsection}{#2}
```

```
\or\omdoc@sectioning[name=\omdoc@subsubsection@kw,num]{subsubsection}{#2}
7199
       \or\omdoc@sectioning[name=\omdoc@paragraph@kw,ref=this \omdoc@paragraph@kw]{paragraph}{#
7200
       \or\omdoc@sectioning[name=\omdoc@subparagraph@kw,ref=this \omdoc@subparagraph@kw]{paragr
7201
     \fi
7202
     \at@begin@sfragment[#1]\l_document_structure_section_level_int{#2}
7203
     \str_if_empty:NF \l__document_structure_sfragment_id_str {
7204
       \stex_ref_new_doc_target:n\l__document_structure_sfragment_id_str
7205
7207 }% for customization
7208 {}
    and finally, we localize the sections
   \newcommand\omdoc@part@kw{Part}
   \newcommand\omdoc@chapter@kw{Chapter}
   \newcommand\omdoc@section@kw{Section}
   \newcommand\omdoc@subsection@kw{Subsection}
   \newcommand\omdoc@subsubsection@kw{Subsubsection}
   \newcommand\omdoc@paragraph@kw{paragraph}
7215 \newcommand\omdoc@subparagraph@kw{subparagraph}
```

37.3 Front and Backmatter

Index markup is provided by the omtext package [Kohlhase:smmtf:git], so in the document-structure package we only need to supply the corresponding \printindex command, if it is not already defined

\printindex

```
7216 \providecommand\printindex{\IfFileExists{\jobname.ind}}{\input{\jobname.ind}}{}} (End definition for \printindex. This function is documented on page ??.)
```

some classes (e.g. book.cls) already have \frontmatter, \mainmatter, and \backmatter macros. As we want to define frontmatter and backmatter environments, we save their behavior (possibly defining it) in orig@*matter macros and make them undefined (so that we can define the environments).

```
\cs_if_exist:NTF\frontmatter{
      \let\__document_structure_orig_frontmatter\frontmatter
      \let\frontmatter\relax
7219
7220 }{
      \tl_set:Nn\__document_structure_orig_frontmatter{
7222
        \clearpage
        \@mainmatterfalse
        \pagenumbering{roman}
7224
7225
7226 }
    \cs_if_exist:NTF\backmatter{
7227
      \let\__document_structure_orig_backmatter\backmatter
7228
      \let\backmatter\relax
7229
7230 }{
      \tl_set:Nn\__document_structure_orig_backmatter{
7232
        \clearpage
        \@mainmatterfalse
7233
        \pagenumbering{roman}
7234
7235
```

```
7236 }
```

Using these, we can now define the frontmatter and backmatter environments

frontmatter we use the \orig@frontmatter macro defined above and \mainmatter if it exists, otherwise we define it.

```
\newenvironment{frontmatter}{
      \__document_structure_orig_frontmatter
7238
7239 }{
      \cs if exist:NTF\mainmatter{
7240
        \mainmatter
7241
7242
        \clearpage
7243
        \@mainmattertrue
        \pagenumbering{arabic}
      }
7246
7247 }
```

backmatter As backmatter is at the end of the document, we do nothing for \endbackmatter.

```
\newenvironment{backmatter}{
      \__document_structure_orig_backmatter
7249
7250 }{
      \cs_if_exist:NTF\mainmatter{
7251
        \mainmatter
7252
7253
        \clearpage
7254
        \@mainmattertrue
7255
        \pagenumbering{arabic}
7256
7257
7258 }
```

finally, we make sure that page numbering is a rabic and we have main matter as the default $\,$

7259 \@mainmattertrue\pagenumbering{arabic}

\prematurestop

7273 }

We initialize \afterprematurestop, and provide \prematurestop@endsfragment which looks up \sfragment@level and recursively ends enough {sfragment}s.

```
\def \c__document_structure_document_str{document}
   \newcommand\afterprematurestop{}
   \def\prematurestop@endsfragment{
     \unless\ifx\@currenvir\c__document_structure_document_str
       \expandafter\expandafter\expandafter\end\expandafter\expandafter\expandafter{\expandafter}
7265
       \expandafter\prematurestop@endsfragment
     \fi
7266
7267 }
   \providecommand\prematurestop{
7268
     \message{Stopping~sTeX~processing~prematurely}
7269
     \prematurestop@endsfragment
7270
     \afterprematurestop
7271
7272
     \end{document}
```

(End definition for \prematurestop. This function is documented on page 57.)

37.4 Global Variables

```
set a global variable
\setSGvar
            7274 \RequirePackage{etoolbox}
            7275 \newcommand\setSGvar[1]{\@namedef{sTeX@Gvar@#1}}
            (End definition for \setSGvar. This function is documented on page 57.)
\useSGvar
           use a global variable
            7276 \newrobustcmd\useSGvar[1]{%
                  \@ifundefined{sTeX@Gvar@#1}
            7278
                  {\PackageError{document-structure}
            7279
                    {The sTeX Global variable #1 is undefined}
                    {set it with \protect\setSGvar}}
            7281 \@nameuse{sTeX@Gvar@#1}}
            (End definition for \useSGvar. This function is documented on page 57.)
 \ifSGvar execute something conditionally based on the state of the global variable.
            7282 \newrobustcmd\ifSGvar[3]{\def\0test{#2}\%
                  \@ifundefined{sTeX@Gvar@#1}
                  {\PackageError{document-structure}
            7284
                    {The sTeX Global variable #1 is undefined}
            7285
                    {set it with \protect\setSGvar}}
            7286
                  {\expandafter\ifx\csname sTeX@Gvar@#1\endcsname\@test #3\fi}}
            7287
            (End definition for \ifSGvar. This function is documented on page 57.)
```

NotesSlides – Implementation

38.1 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.

```
7288 (*cls)
7289 (@@=notesslides)
7290 \ProvidesExplClass{notesslides}{2022/08/08}{3.2.0}{notesslides Class}
7291 \RequirePackage{13keys2e}
7292
7293 \keys_define:nn{notesslides / cls}{
              .str_set_x:N = \c_notesslides_class_str_s
7294
              .bool_set:N = \c_notesslides_notes_bool_set:N = \c_notesslides_notes_bool_set.
7295
                        = { \bool_set_false: N \c__notesslides_notes_bool },
     slides
              .code:n
7296
     docopt .str_set_x:N = \c_notesslides_docopt_str,
                         = {
     unknown .code:n
        \PassOptionsToPackage{\CurrentOption}{document-structure}
        \PassOptionsToClass{\CurrentOption}{beamer}
7300
        \PassOptionsToPackage{\CurrentOption}{notesslides}
7301
        \PassOptionsToPackage{\CurrentOption}{stex}
7302
7303
7304 }
   \ProcessKeysOptions{ notesslides / cls }
7305
7306
   \str_if_empty:NF \c__notesslides_class_str {
      \PassOptionsToPackage{class=\c_notesslides_class_str}{document-structure}
   \exp_args:No \str_if_eq:nnT\c__notesslides_class_str{book}{
7311
      \PassOptionsToPackage{defaulttopsect=part}{notesslides}
7312
7313
7314 \exp_args:No \str_if_eq:nnT\c__notesslides_class_str{report}{
      \PassOptionsToPackage{defaulttopsect=part}{notesslides}
7315
7316 }
7318 \RequirePackage{stex}
```

```
7319 \stex_html_backend:T {
      \bool_set_true:N\c__notesslides_notes_bool
7321
7322
    \bool_if:NTF \c__notesslides_notes_bool {
7323
      \PassOptionsToPackage{notes=true}{notesslides}
7324
      \message{notesslides.cls:~Formatting~course~materials~in~notes~mode}
7325
7326 }{
      \PassOptionsToPackage{notes=false}{notesslides}
      \message{notesslides.cls:~Formatting~course~materials~in~slides~mode}
7329
   \langle / cls \rangle
7330
now we do the same for the notesslides package.
    \ProvidesExplPackage{notesslides}{2022/08/08}{3.2.0}{notesslides Package}
    \RequirePackage{13keys2e}
7333
7334
    \keys_define:nn{notesslides / pkg}{
7335
                      .str_set_x:N = \c_notesslides_topsect_str,
      7337
                      .bool_set:N
                                    = \c__notesslides_notes_bool ,
7338
      notes
      slides
                      .code:n
                                     = { \bool_set_false:N \c__notesslides_notes_bool },
7330
                      .bool set:N
                                     = \c__notesslides_sectocframes_bool ,
      sectocframes
7340
                      .bool set:N
                                     = \c_notesslides_frameimages_bool ,
      frameimages
7341
      fiboxed
                      .bool set:N
                                    = \c__notesslides_fiboxed_bool
7342
      noproblems
                      .bool_set:N
                                     = \c_notesslides_noproblems_bool;
7343
      unknown
                      .code:n
7344
        \PassOptionsToClass{\CurrentOption}{stex}
        \PassOptionsToClass{\CurrentOption}{tikzinput}
7347
7348
    \ProcessKeysOptions{ notesslides / pkg }
7350
    \RequirePackage{stex}
7351
    \stex html backend:T {
      \bool_set_true:N\c__notesslides_notes_bool
7353
7354
7355
    \newif\ifnotes
    \bool_if:NTF \c__notesslides_notes_bool {
      \notestrue
7359
7360
      \notesfalse
7361
we give ourselves a macro \@ctopsect that needs only be evaluated once, so that the
\ifdefstring conditionals work below.
7363 \str_if_empty:NTF \c__notesslides_topsect_str {
      \str_set_eq:NN \__notesslidestopsect \c__notesslides_defaulttopsec_str
7365 }{
      \str_set_eq:NN \__notesslidestopsect \c__notesslides_topsect_str
7366
7367
7368 \PassOptionsToPackage{topsect=\_notesslidestopsect}{document-structure}
```

```
7369 (/package)
```

Depending on the options, we either load the article-based document-structure or the beamer class (and set some counters).

```
\bool_if:NTF \c__notesslides_notes_bool {
      \str_if_empty:NT \c__notesslides_class_str {
        \str_set:Nn \c__notesslides_class_str {article}
7374
      \verb|\exp_after:wN| LoadClass | exp_after:wN[\c_notesslides_docopt_str]| \\
7375
        {\c_notesslides\_class\_str}
7376
7377 }{
      \LoadClass[10pt,notheorems,xcolor={dvipsnames,svgnames}]{beamer}
7378
      \newcounter{Item}
7379
      \newcounter{paragraph}
7380
      \newcounter{subparagraph}
7381
      \newcounter{Hfootnote}
7382
   \RequirePackage{document-structure}
```

now it only remains to load the notesslides package that does all the rest.

```
7385 \RequirePackage{notesslides}
7386 (/cls)
```

In notes mode, we also have to make the beamer-specific things available to article via the beamerarticle package. We use options to avoid loading theorem-like environments, since we want to use our own from the STEX packages. The first batch of packages we want are loaded on notesslides.sty. These are the general ones, we will load the STFX-specific ones after we have done some work (e.g. defined the counters m*). Only the stex-logo package is already needed now for the default theme.

```
⟨*package⟩
   \bool if:NT \c notesslides notes bool {
7388
    \RequirePackage{a4wide}
7389
    \RequirePackage{marginnote}
7390
    \PassOptionsToPackage{usenames, dvipsnames, svgnames}{xcolor}
7391
    \RequirePackage{mdframed}
    \RequirePackage[noxcolor,noamsthm]{beamerarticle}
    7395
7396 \RequirePackage{stex-tikzinput}
  \RequirePackage{comment}
  \RequirePackage{url}
  \RequirePackage{graphicx}
  \RequirePackage{pgf}
```

38.2Notes and Slides

\RequirePackage{bookmark}

For the lecture notes cases, we also provide the \usetheme macro that would otherwise come from the beamer class.

```
7402 \bool_if:NT \c__notesslides_notes_bool {
     \renewcommand\usetheme[2][]{\usepackage[#1]{beamertheme#2}}
7404 }
```

```
7405 \NewDocumentCommand \libusetheme {0{} m} {
7406 \libusepackage[#1]{beamertheme#2}
7407 }
7408
```

We define the sizes of slides in the notes. Somehow, we cannot get by with the same here.

```
7409 \newcounter{slide}
7410 \newlength{\slidewidth}\setlength{\slidewidth}{13.5cm}
7411 \newlength{\slideheight}\setlength{\slideheight}{9cm}
```

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 as a comment via the comment package.

```
7412 \bool_if:NTF \c__notesslides_notes_bool {
7413 \renewenvironment{note}{\ignorespaces}{}
7414 }{
7415 \excludecomment{note}
7416 }
```

We first set up the slide boxes in article mode. We set up sizes and provide a box register for the frames and a counter for the slides.

frame We first define the keys.

```
\cs_new_protected:Nn \__notesslides_do_yes_param:Nn {
       \exp_args:Nx \str_if_eq:nnTF { \str_uppercase:n{ #2 } }{ yes }{
7421
         \bool_set_true:N #1
7422
       }{
7423
         \bool_set_false:N #1
7424
       }
7425
7426
     \keys_define:nn{notesslides / frame}{
7427
                           7428
7429
       allowframebreaks
                           .code:n
                                         = {
         \_notesslides_do_yes_param:Nn \_notesslides_frame_allowframebreaks_bool { #1 }
7431
       allowdisplaybreaks .code:n
7432
         \__notesslides_do_yes_param:Nn \l__notesslides_frame_allowdisplaybreaks_bool { #1 }
7433
       },
7434
       fragile
                           .code:n
                                         = {
7435
         \__notesslides_do_yes_param:Nn \l__notesslides_frame_fragile_bool { #1 }
7436
7437
7438
         \__notesslides_do_yes_param:Nn \l__notesslides_frame_shrink_bool { #1 }
7439
       },
       squeeze
                            .code:n
                                         = {
7442
         \__notesslides_do_yes_param:Nn \l__notesslides_frame_squeeze_bool { #1 }
7443
       t
                                         = {
7444
                           .code:n
```

```
},
    7446
                                                                                                                                                                        = {}
                                                                                              .code:n
    7447
                                        unknown
    7448
                                \cs_new_protected:Nn \__notesslides_frame_args:n {
   7449
                                          \str_clear:N \l__notesslides_frame_label_str
   7450
                                          \bool_set_true:N \l__notesslides_frame_allowframebreaks_bool
   7451
                                          \bool_set_true:N \l__notesslides_frame_allowdisplaybreaks_bool
    7452
                                          \bool_set_true:N \l__notesslides_frame_fragile_bool
                                          \bool_set_true:N \l__notesslides_frame_shrink_bool
                                          \verb|\bool_set_true:N \l| \_notesslides\_frame\_squeeze\_bool|
    7455
                                          \verb|\bool_set_true:N \l| = notesslides_frame_t_bool|
    7456
                                           \keys_set:nn { notesslides / frame }{ #1 }
    7457
    7458
We define the environment, read them, and construct the slide number and label.
                                \renewenvironment{frame}[1][]{
    7/150
                                           \__notesslides_frame_args:n{#1}
    7460
                                          \sffamily
   7461
                                          \stepcounter{slide}
    7462
                                          \def\@currentlabel{\theslide}
    7463
                                          \str if empty:NF \l notesslides frame label str {
    7464
                                                      \label{\l_notesslides_frame_label_str}
    7465
We redefine the itemize environment so that it looks more like the one in beamer.
                                          \def\itemize@level{outer}
   7467
                                          \def\itemize@outer{outer}
    7468
                                           \def\itemize@inner{inner}
    7469
                                           \renewcommand\newpage{\addtocounter{framenumber}{1}}
                                          %\newcommand\metakeys@show@keys[2]{\marginnote{{\scriptsize ##2}}}
                                           \renewenvironment{itemize}{
                                                      \ifx\itemize@level\itemize@outer
    7473
                                                                \def\itemize@label{$\rhd$}
    7474
                                                     \fi
    7475
                                                     \ifx\itemize@level\itemize@inner
    7476
                                                               \def\itemize@label{$\scriptstyle\rhd$}
   7477
                                                     \fi
   7478
                                                      \begin{list}
   7479
                                                     {\itemize@label}
    7480
                                                     {\left\langle \cdot \right\rangle }{\left\langle 
                                                          \setlength{\labelwidth}{.5em}
                                                          \setlength{\leftmargin}{1.5em}
    7484
                                                     \edef\itemize@level{\itemize@inner}
    7485
                                         }{
    7486
                                                      \end{list}
   7487
    7488
We create the box with the mdframed environment from the equinymous package.
                                          \stex_html_backend:TF {
    7489
                                                      \begin{stex_annotate_env}{frame}{}\vbox\bgroup
   7490
                                                                \mdf@patchamsthm
   7491
                                         7-{
   7492
                                                      \begin{mdframed}[linewidth=\slideframewidth,skipabove=1ex,skipbelow=1ex,userdefinedwid
   7493
```

_notesslides_do_yes_param:Nn \l__notesslides_frame_t_bool { #1 }

7445

```
}
               7494
               7495
                       \stex_html_backend:TF {
               7496
                          \verb|\miko@slidelabel\egroup\end{stex\_annotate\_env}|
               7497
                       }{\medskip\miko@slidelabel\end{mdframed}}
               7498
               7499
                   Now, we need to redefine the frametitle (we are still in course notes mode).
\frametitle
                     \renewcommand{\frametitle}[1]{
               7500
                       \stex_document_title:n { #1 }
               7501
                       {\Large\bf\sf\color{blue}{#1}}\medskip
               7503
               7504 }
              (End definition for \frametitle. This function is documented on page ??.)
              10
     \pause
               7505 \bool_if:NT \c__notesslides_notes_bool {
                     \newcommand\pause{}
              (End definition for \parbox{\color{$\sim$}} This function is documented on page \parbox{\color{$\sim$}}.)
 nparagraph
               7508 \bool_if:NTF \c__notesslides_notes_bool {
                     \newenvironment{nparagraph}[1][]{\begin{sparagraph}[#1]}{\end{sparagraph}}
                     \excludecomment{nparagraph}
               7512 }
  nfragment
               7513 \bool_if:NTF \c__notesslides_notes_bool {
                     \newenvironment{nfragment}[2][]{\begin{sfragment}[#1]{#2}}{\end{sfragment}}
                     \excludecomment{nfragment}
               7517 }
ndefinition
               7518 \bool_if:NTF \c__notesslides_notes_bool {
                     \newenvironment{ndefinition}[1][]{\begin{sdefinition}[#1]}{\end{sdefinition}}}
                     \excludecomment{ndefinition}
               7522 }
 nassertion
               7523 \bool_if:NTF \c__notesslides_notes_bool {
                     \newenvironment{nassertion}[1][]{\begin{sassertion}[#1]}{\end{sassertion}}
                     \excludecomment{nassertion}
               7527 }
```

EdN:10

 $^{10}\mathrm{EdNote}$: MK: fake it in notes mode for now

```
nsproof
                 7528 \bool_if:NTF \c__notesslides_notes_bool {
                       7530 }{
                       \excludecomment{nproof}
                 7531
                 7532 }
      nexample
                 7533 \bool_if:NTF \c__notesslides_notes_bool {
                       \newenvironment{nexample}[1][]{\begin{sexample}[#1]}{\end{sexample}}
                 7535 }{
                       \excludecomment{nexample}
                 7536
                 7537 }
                We customize the hooks for in \inputref.
\inputref@*skip
                 7538 \def\inputref@preskip{\smallskip}
                 7539 \def\inputref@postskip{\medskip}
                 (End definition for \inputref@*skip. This function is documented on page ??.)
    \inputref*
                 7540 \let\orig@inputref\inputref
                 7541 \def\inputref{\@ifstar\ninputref\orig@inputref}
                 7542 \newcommand\ninputref[2][]{
                       \bool_if:NT \c__notesslides_notes_bool {
                         \orig@inputref[#1]{#2}
                 7546
                 (End definition for \inputref*. This function is documented on page 59.)
```

38.3 Header and Footer Lines

Now, we set up the infrastructure for the footer line of the slides, we use boxes for the logos, so that they are only loaded once, that considerably speeds up processing.

\setslidelogo

The default logo is the SIEX logo. Customization can be done by $\setslidelogo\{\langle logo name \rangle\}$.

```
7547 \newlength{\slidelogoheight}
7548
   \RequirePackage{graphicx}
7549
7550
7551 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
   \providecommand\mhgraphics[2][]{
7552
      \def\Gin@mhrepos{}\setkeys{Gin}{#1}
7553
      \includegraphics[#1]{\mhpath\Gin@mhrepos{#2}}
7554
7555 }
7557 \bool_if:NTF \c__notesslides_notes_bool {
     \setlength{\slidelogoheight}{.4cm}
7558
7559 }{
     \setlength{\slidelogoheight}{.25cm}
7560
7561 }
```

```
\ifcsname slidelogo\endcsname\else
      \newsavebox{\slidelogo}
7563
      \slidelogo{\sIidelogo}{\sTeX}
7564
7565
    \newrobustcmd{\setslidelogo}[2][]{
7566
      \tl_if_empty:nTF{#1}{
7567
        \sbox{\slidelogo}{\includegraphics[height=\slidelogoheight]{#2}}
7568
7569
        \sbox{\slidelogo}{\mhgraphics[height=\slidelogoheight,mhrepos=#1]{#2}}
7571
7572 }
```

(End definition for \setslidelogo. This function is documented on page 60.)

\author In notes mode, we redefine the \author macro so that it does not disregard the optional argument (as beamerarticle does). We want to use it to set the source later.

```
7573 \bool_if:NT \c__notesslides_notes_bool {
7574 \def\author{\@dblarg\ns@author}
7575 \long\def\ns@author[#1]#2{%
7576 \def\c__notesslides_shortauthor{#1}%
7577 \def\@author{#2}
7578 }
7579 }
```

(End definition for \author. This function is documented on page ??.)

\setsource

\source stores the writer's name. By default it is *Michael Kohlhase* since he is the main user and designer of this package. \setsource $\{\langle name \rangle\}$ can change the writer's name.

```
7580 \newrobustcmd{\setsource}[1]{\def\source{#1}}
```

(End definition for \setsource. This function is documented on page 60.)

\setlicensing

Now, we set up the copyright and licensing. By default we use the Creative Commons Attribuition-ShareAlike license to strengthen the public domain. If package hyperref is loaded, then we can attach a hyperlink to the license logo. $\ensuremath{\mbox{setlicensing}}[\langle url \rangle] \{\langle logo\ name \rangle\}$ is used for customization, where $\langle url \rangle$ is optional.

```
\def\copyrightnotice{%
7581
     \footnotesize\copyright :\hspace{.3ex}%
7582
     \ifcsname source\endcsname\source\else%
7583
     \ifcsname c_notesslides_shortauthor\endcsname\c_notesslides_shortauthor\else%
7584
     \PackageWarning{notesslides}{Author/Source~undefined~in~copyright~notice}%
7585
     ?source/author?\fi%
     \{fi\}
   \newsavebox{\cclogo}
   \sbox{\cclogo}{\includegraphics[height=\slidelogoheight]{stex-cc_somerights}}
   \newif\ifcchref\cchreffalse
   \AtBeginDocument{
7591
     \@ifpackageloaded{hyperref}{\cchreftrue}{\cchreffalse}
7592
7593 }
   \def\licensing{
7594
     \ifcchref
7595
        \href{http://creativecommons.org/licenses/by-sa/2.5/}{\usebox{\cclogo}}
7596
       {\usebox{\cclogo}}
7598
```

```
7600 }
                   \newrobustcmd{\setlicensing}[2][]{
               7601
                      \left( \frac{41}{41} \right)
               7602
                      \sbox{\cclogo}{\includegraphics[height=\slidelogoheight]{#2}}
                7603
                      \int (Qurl \end y)
                7604
                        \def\licensing{{\usebox{\cclogo}}}
                7605
                      \else
                7606
                        \def\licensing{
                          \ifcchref
                7608
                           \href{#1}{\usebox{\cclogo}}
                           \else
                7610
                          {\usebox{\cclogo}}
                7611
                           \fi
                7612
                7613
                      \fi
               7614
               7615 }
               (End definition for \setlicensing. This function is documented on page 60.)
\slidelabel Now, we set up the slide label for the article mode. 11
                   \newrobustcmd\miko@slidelabel{
                      \vbox to \slidelogoheight{
                        \vss\hbox to \slidewidth
                        {\consing\hfill\copyright notice\hfill\arabic{slide}\hfill\usebox{\slidelogo}}}
                7619
                7620
               7621 }
               (End definition for \slidelabel. This function is documented on page ??.)
```

38.4 Frame Images

\fi

7599

EdN:11

\frameimage We have to make sure that the width is overwritten, for that we check the \Gin@ewidth macro from the graphicx package. We also add the label key.

```
\def\Gin@mhrepos{}
   \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
   \define@key{Gin}{label}{\def\@currentlabel{\arabic{slide}}\label{#1}}
   \newrobustcmd\frameimage[2][]{
     \stepcounter{slide}
7626
     \bool_if:NT \c__notesslides_frameimages_bool {
7627
       \def\Gin@ewidth{}\setkeys{Gin}{#1}
7628
       \bool_if:NF \c__notesslides_notes_bool { \vfill }
7629
       \begin{center}
7630
          \bool_if:NTF \c__notesslides_fiboxed_bool {
            fbox{
              \int Gin@ewidth\end{array}
7633
                \ifx\Gin@mhrepos\@empty
7634
                  \mhgraphics[width=\slidewidth,#1]{#2}
7635
                \else
7636
                  \mhgraphics[width=\slidewidth,#1,mhrepos=\Gin@mhrepos]{#2}
7637
7638
              \else% Gin@ewidth empty
```

 $^{^{11}\}mathrm{EdNote}\colon$ see that we can use the themes for the slides some day. This is all fake.

```
\ifx\Gin@mhrepos\@empty
                   \mhgraphics[#1]{#2}
7641
                 \else
7642
                   \mhgraphics[#1,mhrepos=\Gin@mhrepos]{#2}
7643
                 \fi
7644
               \fi% Gin@ewidth empty
            }
          }{
            \int Gin@ewidth\end{array}
              \ifx\Gin@mhrepos\@empty
                 \mhgraphics[width=\slidewidth,#1]{#2}
7651
                 \mhgraphics[width=\slidewidth, #1, mhrepos=\Gin@mhrepos]{#2}
7652
7653
               \ifx\Gin@mhrepos\@empty
7654
                 \mhgraphics[#1]{#2}
7655
7656
                 \mhgraphics[#1,mhrepos=\Gin@mhrepos]{#2}
              \fi
            \fi% Gin@ewidth empty
          }
         \end{center}
7661
        \par\strut\hfill{\footnotesize Slide \arabic{slide}}%
7662
        \bool_if:NF \c__notesslides_notes_bool { \vfill }
7663
7664
7665 } % ifmks@sty@frameimages
```

(End definition for \frameimage. This function is documented on page 60.)

38.5 Sectioning

If the sectocframes option is set, then we make section frames. We first define counters for part and chapter, which beamer.cls does not have and we make the section counter which it does dependent on chapter.

```
\stex_html_backend:F {
      \bool_if:NT \c__notesslides_sectocframes_bool {
        \str_if_eq:VnTF \__notesslidestopsect{part}{
7668
          \newcounter{chapter}\counterwithin*{section}{chapter}
7669
        }{
7670
          \verb|\str_if_eq:VnT\__notesslidestopsect{chapter}| \{
7671
            \newcounter{chapter}\counterwithin*{section}{chapter}
7672
7673
7674
7675
     }
7676 }
```

\section@level

We set the \section@level counter that governs sectioning according to the class options. We also introduce the sectioning counters accordingly.

\section@level

```
7677 \def\part@prefix{}
7678 \@ifpackageloaded{document-structure}{}{
7679 \str_case:VnF \__notesslidestopsect {
```

```
{part}{
          \int_set:Nn \l_document_structure_section_level_int {0}
7681
          \def\thesection{\arabic{chapter}.\arabic{section}}
7682
          \def\part@prefix{\arabic{chapter}.}
7683
7684
        {chapter}{
7685
          \int_set:Nn \l_document_structure_section_level_int {1}
7686
          \def\thesection{\arabic{chapter}.\arabic{section}}
7687
          \def\part@prefix{\arabic{chapter}.}
     7-{
        \int_set:Nn \l_document_structure_section_level_int {2}
7691
        \def\part@prefix{}
7692
7693
7694
7695
   \bool_if:NF \c__notesslides_notes_bool { % only in slides
```

(End definition for \section@level. This function is documented on page ??.)

The new counters are used in the sfragment environment that choses the LATEX sectioning macros according to \section@level.

sfragment

```
7697
     \renewenvironment{sfragment}[2][]{
       \__document_structure_sfragment_args:n { #1 }
7698
       \int_incr:N \l_document_structure_section_level_int
7699
       \bool_if:NT \c__notesslides_sectocframes_bool {
7700
          \stepcounter{slide}
7701
          \begin{frame} [noframenumbering]
7702
          \vfill\Large\centering
7703
7704
            \ifcase\l_document_structure_section_level_int\or
              \stepcounter{part}
              \def\__notesslideslabel{{\omdoc@part@kw}~\Roman{part}}
              \label{line} $$ \addcontentsline{toc}{part}{\protect\numberline{\thepart}$\#2}$
7708
              \pdfbookmark[0]{\thepart\ #2}{part.\thepart}
7709
              \def\currentsectionlevel{\omdoc@part@kw}
            \or
              \stepcounter{chapter}
              \def\__notesslideslabel{{\omdoc@chapter@kw}~\arabic{chapter}}
7713
7714
              \addcontentsline{toc}{chapter}{\protect\numberline{\thechapter}#2}
              \pdfbookmark[1]{\thechapter\ #2}{chapter.\cs_if_exist:cT{thepart}\thepart.\thechap
              \def\currentsectionlevel{\omdoc@chapter@kw}
            \or
              \stepcounter{section}
7718
              \def\__notesslideslabel{\part@prefix\arabic{section}}
7719
              \addcontentsline{toc}{section}{\protect\numberline{\thesection}#2}
              \pdfbookmark[2]{\cs_if_exist:cT{thechapter}{\thechapter.}\thesection\ #2}
              \{section.\cs_if_exist:cT\{thepart\}\{\thepart\}.\cs_if_exist:cT\{thechapter\}\{\thechapter\}\}
              \def\currentsectionlevel{\omdoc@section@kw}
              \stepcounter{subsection}
              \def\__notesslideslabel{\part@prefix\arabic{section}.\arabic{subsection}}
```

```
\{subsection.\cs_if_exist:cT\{thepart\}\{thepart\}.\cs_if_exist:cT\{thechapter\}\{thechapter\}\}
7729
                                                    \def\currentsectionlevel{\omdoc@subsection@kw}
7730
                                            \or
                                                     \stepcounter{subsubsection}
                                                     \def\__notesslideslabel{\part@prefix\arabic{section}.\arabic{subsection}.\arabic{s}
                                                     \addcontentsline{toc}{subsubsection}{\protect\numberline{\thesubsubsection}#2}
7734
                                                     \protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\pro
                                                     {subsubsection.\cs_if_exist:cT{thepart}{\thepart}.\cs_if_exist:cT{thechapter}{\the
                                                     \def\currentsectionlevel{\omdoc@subsubsection@kw}
                                                     \stepcounter{paragraph}
7730
                                                     7740
                                                     \label{lem:line} $$ \addcontentsline{toc}{paragraph}_{\protect\numberline{theparagraph}$\#2}$ $$
7741
                                                     \protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\pro
7742
                                                     {paragraph.\cs_if_exist:cT{thepart}{\thepart}.\cs_if_exist:cT{thechapter}{\thechap
7743
                                                     \def\currentsectionlevel{\omdoc@paragraph@kw}
7744
                                              \else
                                                     \def\__notesslideslabel{}
                                                     \def\currentsectionlevel{\omdoc@paragraph@kw}
                                             \fi% end ifcase
                                             \_{notesslideslabel\quad\ #2\%}
7749
                                    }%
7750
                                     \vfill%
7751
                                     \end{frame}%
7753
7754
                             \str_if_empty:NF \l__document_structure_sfragment_id_str {
7755
                                     \stex_ref_new_doc_target:n\l__document_structure_sfragment_id_str
7757
                    }{}
7758 }
```

We set up a beamer template for theorems like ams style, but without a block environment.

```
7759 \def\inserttheorembodyfont{\normalfont}
7760 %\bool_if:NF \c__notesslides_notes_bool {
7761 % \defbeamertemplate{theorem begin}{miko}
7762 % {\inserttheoremheadfont\inserttheoremname\inserttheoremnumber
7763 % \inserttheoremaddition\@empty\else\ (\inserttheoremaddition)\fi%
7764 % \inserttheorempunctuation\inserttheorembodyfont\xspace}
7765 % \defbeamertemplate{theorem end}{miko}{}
and we set it as the default one.
```

7766 % \setbeamertemplate{theorems}[miko]

The following fixes an error I do not understand, this has something to do with beamer compatibility, which has similar definitions but only up to 1.

```
7767 % \expandafter\def\csname Parent2\endcsname{}
7768 %}
7769
7770 \AddToHook{begindocument}{ % this does not work for some reasone
7771 \setbeamertemplate{theorems}[ams style]
7772 }
7773 \bool_if:NT \c__notesslides_notes_bool {
7774 \renewenvironment{columns}[1][]{%
```

```
\par\noindent%
7775
        \begin{minipage}%
7776
        \slidewidth\centering\leavevmode%
      }{%
7778
        \end{minipage}\par\noindent%
7779
      3%
7780
      \newsavebox\columnbox%
7781
      \renewenvironment<>{column}[2][]{%
7782
        \begin{lrbox}{\columnbox}\begin{minipage}{#2}\%
      }{%
7784
        \end{minipage}\end{lrbox}\usebox\columnbox%
7785
      }%
7786
7787
    \bool if:NTF \c notesslides noproblems bool {
7788
      \newenvironment{problems}{}{}
7789
7790
   }{
      \excludecomment{problems}
7792 }
```

38.6 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.

```
\gdef\printexcursions{}
                       \newcommand\excursionref[2]{% label, text
                         \bool_if:NT \c__notesslides_notes_bool {
                   7795
                           \begin{sparagraph}[title=Excursion]
                   7796
                             #2 \sr [fallback=the appendix]{#1}.
                   7797
                           \end{sparagraph}
                   7798
                   7799
                   7800
                   7801
                       \newcommand\activate@excursion[2][]{
                         \gappto\printexcursions{\inputref[#1]{#2}}
                   7802
                   7803
                       \newcommand\excursion[4][]{% repos, label, path, text
                         \verb|\bool_if:NT \c_notesslides_notes_bool| \{
                   7805
                           \activate@excursion[#1]{#3}\excursionref{#2}{#4}
                   7806
                   7807
                   7808 }
                  (End definition for \excursion. This function is documented on page 61.)
\excursiongroup
                      \keys_define:nn{notesslides / excursiongroup }{
                   7809
                         id
                                    .str_set_x:N = \l__notesslides_excursion_id_str,
                   7810
                                                   = \l__notesslides_excursion_intro_tl,
                         intro
                                   .tl_set:N
                   7811
                                   .str_set_x:N = \l__notesslides_excursion_mhrepos_str
                   7812
                        mhrepos
                   7813
                      \cs_new_protected:Nn \__notesslides_excursion_args:n {
                         \tl_clear:N \l__notesslides_excursion_intro_tl
                   7815
                         \str_clear:N \l__notesslides_excursion_id_str
                   7816
```

```
\verb|\str_clear:N| l\_notesslides_excursion_mhrepos\_str|
7817
                        \keys_set:nn {notesslides / excursiongroup }{ #1 }
7818
7819 }
                \newcommand\excursiongroup[1][]{
7820
                         \__notesslides_excursion_args:n{ #1 }
7821
                        \iftime for the following the following the following the following the following following the following the following following the following following the following following following the following fo
7822
                        {\begin{note}
7823
                                 \begin{sfragment}[#1]{Excursions}%
 7824
                                         \verb|\input ref[\l_notesslides_excursion_mhrepos_str]| \{
                                                           \verb|\label{loss}| 1\_notesslides\_excursion\_intro\_tl|
  7828
                                         }
 7829
                                          \printexcursions%
 7830
                                 \end{sfragment}
7831
                        \end{note}}
7832
7833 }
7834 \ifcsname beameritemnestingprefix\endcsname\else\def\beameritemnestingprefix{}\fi
7835 (/package)
```

(End definition for $\ensuremath{\char{\color{location}}}$ function is documented on page 61.)

The Implementation

39.1 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.

```
7836 (*package)
7837 (@@=problems)
7838 \ProvidesExplPackage{problem}{2022/08/08}{3.2.0}{Semantic Markup for Problems}
7839 \RequirePackage{13keys2e}
7840 \RequirePackage{amssymb}% for \Box
7841
7842 \keys_define:nn { problem / pkg }{
    notes .default:n = { true },
              .bool_set:N = \c__problems_notes_bool,
    notes
    gnotes .default:n
                            = { true },
    gnotes .bool_set:N = \c__problems_gnotes_bool,
              .default:n
                            = { true },
    hints
7847
            .bool_set:N = \c_problems_hints_bool,
    hints
7848
    solutions .default:n
                            = { true },
7849
    solutions .bool_set:N = \c__problems_solutions_bool,
7850
   pts .default:n
                            = { true },
7851
            .bool_set:N = \c__problems_pts_bool,
.default:n = { true },
   pts
7852
             .bool_set:N = \c_problems_min_bool,
    min
    boxed .default:n
                             = { true },
     boxed .bool_set:N = \c_problems_boxed_bool,
               .code:n
     unknown
       \PassOptionsToPackage{\CurrentOption}{stex}
7858
7859
7860 }
   \newif\ifsolutions
7861
7863 \ProcessKeysOptions{ problem / pkg }
7864 \bool_if:NTF \c__problems_solutions_bool {
     \solutionstrue
7866 }{
    \solutionsfalse
```

```
7868 }
 7869 \RequirePackage{stex}
    Then we make sure that the necessary packages are loaded (in the right versions).
 7870 \RequirePackage{comment}
    The next package relies on the LATEX3 kernel, which LATEXMLonly partially sup-
ports. As it is purely presentational, we only load it when the boxed option is given and
we run LaTeXML.
7871 \bool_if:NT \c__problems_boxed_bool { \RequirePackage{mdframed} }
For multilinguality, we define internal macros for keywords that can be specialized in
7872 \def\prob@problem@kw{Problem}
    \def\prob@solution@kw{Solution}
    \def\prob@hint@kw{Hint}
    \def\prob@note@kw{Note}
    \def\prob@gnote@kw{Grading}
 7877 \def\prob@pt@kw{pt}
7878 \def\prob@min@kw{min}
7879 \def\prob@correct@kw{Correct}
7880 \def\prob@wrong@kw{Wrong}
(End definition for \prob@*@kw. This function is documented on page ??.)
    For the other languages, we set up triggers
    \AddToHook{begindocument}{
      \ltx@ifpackageloaded{babel}{
 7882
           \makeatletter
           \clist_set:Nx \l_tmpa_clist {\bbl@loaded}
           \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{ngerman}}{
             \input{problem-ngerman.ldf}
 7886
 7887
           \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{finnish}}{
 7888
             \input{problem-finnish.ldf}
 7889
 7890
           \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{french}}{
```

Problems and Solutions 39.2

\makeatother

\input{problem-french.ldf}

\input{problem-russian.ldf}

\prob@*@kw

7891

7892

7896

7897

7898 7899 } }{}

We now prepare the KeyVal support for problems. The key macros just set appropriate internal macros.

\exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{russian}}{

```
7900 \keys_define:nn{ problem / problem }{
               .str_set_x:N = \label{eq:str_set_x} = \label{eq:str_set_x} 
     id
7901
     pts
               .tl set:N
                               = \l__problems_prob_pts_tl,
7902
     min
               .tl set:N
                               = \1_problems_prob_min_tl,
7903
```

```
.tl_set:N
                                                 = \l__problems_prob_type_tl,
                    7905
                          type
                                                 = \l__problems_prob_imports_tl,
                          imports .tl_set:N
                    7906
                                  .str_set_x:N = \l__problems_prob_name_str,
                    7907
                                  .int_set:N
                                                 = \l_problems_prob_refnum_int
                          refnum
                    7908
                    7909
                        \cs_new_protected:Nn \__problems_prob_args:n {
                    7910
                          \str_clear:N \l__problems_prob_id_str
                    7911
                          \str_clear:N \l__problems_prob_name_str
                          \verb|\tl_clear:N \l_problems_prob_pts_tl|
                    7913
                          \tl_clear:N \l__problems_prob_min_tl
                    7914
                          \verb|\tl_clear:N \l_problems_prob_title_tl|
                    7915
                          \tl_clear:N \l__problems_prob_type_tl
                    7916
                          \verb|\tl_clear:N \l_problems_prob_imports_tl|\\
                    7917
                          7918
                          \keys_set:nn { problem / problem }{ #1 }
                    7919
                          \int_compare:nNnT \l__problems_prob_refnum_int = 0 {
                    7920
                            \label{lems_prob_refnum_int} \
                    7921
                    7922
                    7923 }
                        Then we set up a counter for problems.
\numberproblemsin
                        \newcounter{sproblem}[section]
                        \newcommand\numberproblemsin[1]{\@addtoreset{sproblem}{#1}}
                        \def\theplainsproblem{\arabic{sproblem}}
                        \def\thesproblem{\thesection.\theplainsproblem}
                    (End definition for \numberproblemsin. This function is documented on page ??.)
                   We provide the macro \prob@label to redefine later to get context involved.
      \prob@label
                    7928 \newcommand\prob@label[1]{\thesection.#1}
                    (End definition for \prob@label. This function is documented on page ??.)
                   We consolidate the problem number into a reusable internal macro
     \prob@number
                        \newcommand\prob@number{
                    7929
                          \int_if_exist:NTF \l__problems_inclprob_refnum_int {
                    7930
                            \prob@label{\int_use:N \l__problems_inclprob_refnum_int }
                    7931
                    7932
                    7933
                            \int_if_exist:NTF \l__problems_prob_refnum_int {
                              \prob@label{\int_use:N \l__problems_prob_refnum_int }
                    7934
                    7935
                                 \prob@label\theplainsproblem
                    7937
                    7938
                    7939 }
                        \def\sproblemautorefname{\prob@problem@kw}
```

= \l__problems_prob_title_tl,

title

7904

.tl_set:N

\prob@title We consolidate the problem title into a reusable internal macro as well. \prob@title takes three arguments the first is the fallback when no title is given at all, the second and third go around the title, if one is given.

(End definition for \prob@number. This function is documented on page ??.)

(End definition for \prob@title. This function is documented on page ??.)

With these the problem header is a one-liner

\prob@heading

We consolidate the problem header line into a separate internal macro that can be reused in various settings.

(End definition for \prob@heading. This function is documented on page ??.)

With this in place, we can now define the **problem** environment. It comes in two shapes, depending on whether we are in boxed mode or not. In both cases we increment the problem number and output the points and minutes (depending) on whether the respective options are set.

sproblem

```
\newenvironment{sproblem}[1][]{
     \__problems_prob_args:n{#1}%\sref@target%
7957
7958
     \@in@omtexttrue% we are in a statement (for inline definitions)
     \verb|\refstepcounter{sproblem}| \verb|\record@problem||
     \def\current@section@level{\prob@problem@kw}
     \str_if_empty:NT \l__problems_prob_name_str {
7962
       \seq_get_right:NN \g_stex_currentfile_seq \l_tmpa_str
7963
       7964
       7965
7966
7967
     \stex_if_do_html:T{
7968
       \tl_if_empty:NF \l__problems_prob_title_tl {
7969
         \exp_args:No \stex_document_title:n \l__problems_prob_title_tl
7971
     }
7972
7973
     \exp_args:Nno\stex_module_setup:nn{type=problem}\l_problems_prob_name_str
7974
7975
     \stex_reactivate_macro:N \STEXexport
7976
     \stex_reactivate_macro:N \importmodule
7977
     \stex_reactivate_macro:N \symdecl
7978
     \stex_reactivate_macro:N \notation
7979
     \stex_reactivate_macro:N \symdef
```

```
7981
     \stex_if_do_html:T{
7982
        \begin{stex_annotate_env} {problem} {
7983
          \l_stex_module_ns_str ? \l_stex_module_name_str
7984
7985
7986
        \stex_annotate_invisible:nnn{header}{} {
7987
          \stex_annotate:nnn{language}{ \l_stex_module_lang_str }{}
          \stex_annotate:nnn{signature}{ \l_stex_module_sig_str }{}
          \str_if_eq:VnF \l_stex_module_meta_str {NONE} {
            \stex_annotate:nnn{metatheory}{ \l_stex_module_meta_str }{}
7992
       }
7993
     }
7994
7995
     \stex_csl_to_imports:No \importmodule \l__problems_prob_imports_tl
7996
7997
7998
     \verb|\tl_if_exist:NTF \ | \_problems_inclprob_type_tl \ \{
        \tl_set_eq:NN \sproblemtype \l__problems_inclprob_type_tl
     }{
        \tl_set_eq:NN \sproblemtype \l__problems_prob_type_tl
8002
8003
     \verb|\str_if_exist:NTF \l_problems_inclprob_id_str \{|
8004
        \verb|\str_set_eq:NN \sproblemid \l_problems_inclprob_id_str|\\
8005
8006
        \str_set_eq:NN \sproblemid \l__problems_prob_id_str
8007
8008
8009
     \stex_if_smsmode:F {
8011
        \clist_set:No \l_tmpa_clist \sproblemtype
8012
        \t! clear: N \l_tmpa_tl
8013
        \clist_map_inline:Nn \l_tmpa_clist {
8014
          \tl_if_exist:cT {__problems_sproblem_##1_start:}{
8015
            \tl_set:Nn \l_tmpa_tl {\use:c{__problems_sproblem_##1_start:}}
8016
8017
8018
8019
        \tl_if_empty:NTF \l_tmpa_tl {
          \__problems_sproblem_start:
          \l_tmpa_t1
       }
8023
8024
     \verb|\stex_ref_new_doc_target:n \sproblemid|
8025
     \stex_if_smsmode:TF \stex_smsmode_do: \ignorespacesandpars
8026
8027
      \__stex_modules_end_module:
8028
     \stex_if_smsmode:F{
8029
        \clist_set:No \l_tmpa_clist \sproblemtype
8030
        \tl_clear:N \l_tmpa_tl
8032
        \clist_map_inline:Nn \l_tmpa_clist {
          \verb|\tl_if_exist:cT {\_problems_sproblem_\#1_end:}{|} 
8033
            8034
```

```
8036
                             \tl_if_empty:NTF \l_tmpa_tl {
                    8037
                                \_\_problems\_sproblem\_end:
                    8038
                    8039
                                \label{local_local_thm} \label{local_thm} $$1_tmpa_t1$
                     8040
                    8041
                    8042
                           \stex_if_do_html:T{
                             \end{stex_annotate_env}
                     8044
                     8045
                     8046
                           \smallskip
                    8047
                    8048 }
                    8049
                         \seq_put_right:Nx\g_stex_smsmode_allowedenvs_seq{\tl_to_str:n{sproblem}}
                    8050
                    8051
                    8052
                         \cs_new_protected:Nn \__problems_sproblem_start: {
                           \par\noindent\textbf\prob@heading\show@pts\show@min\\\ignorespacesandpars
                     8056
                         \cs_new_protected:Nn \__problems_sproblem_end: {\par\smallskip}
                    8057
                    8058
                         \newcommand\stexpatchproblem[3][] {
                    8059
                             \str_set:Nx \l_tmpa_str{ #1 }
                    8060
                             \str_if_empty:NTF \l_tmpa_str {
                     8061
                                \tl_set:Nn \__problems_sproblem_start: { #2 }
                     8062
                                \tl_set:Nn \__problems_sproblem_end: { #3 }
                     8063
                             }{
                                \exp_after:wN \tl_set:Nn \csname __problems_sproblem_#1_start:\endcsname{ #2 }
                     8065
                                \exp_after:wN \t1_set:Nn \csname __problems_sproblem_#1_end:\endcsname{ #3 }
                    8067
                    8068
                    8069
                    8070
                        \bool_if:NT \c__problems_boxed_bool {
                    8071
                    8072
                           \surroundwithmdframed{problem}
                   This macro records information about the problems in the *.aux file.
\record@problem
                         \def\record@problem{
                    8074
                           \protected@write\@auxout{}
                    8075
                           {
                    8076
                             \string\@problem{\prob@number}
                    8077
                     8078
                                \tl_if_exist:NTF \l__problems_inclprob_pts_tl {
                                  \label{local_problems_inclprob_pts_tl} $$ l_problems_inclprob_pts_tl $$
                                  \verb|\l_problems_prob_pts_t|
                     8082
                     8083
                             }%
                    8084
                             {
                    8085
                                \tl_if_exist:NTF \l__problems_inclprob_min_tl {
                    8086
```

}

8035

```
8087 \l__problems_inclprob_min_tl

8088 }{

8089 \l__problems_prob_min_tl

8090 }

8091 }

8092 }

8093 }
```

(End definition for \record@problem. This function is documented on page ??.)

\Oproblem This macro acts on a problem's record in the *.aux file. It does not have any functionality here, but can be redefined elsewhere (e.g. in the assignment package).

```
8094 \def\@problem#1#2#3{}
```

(End definition for \Oproblem. This function is documented on page ??.)

solution

The solution environment is similar to the problem environment, only that it is independent of the boxed mode. It also has it's own keys that we need to define first.

```
\keys_define:nn { problem / solution }{
                   id
8096
     for
                   .str_set_x:N = \label{eq:solution_for_str} ,
8097
     type
                   .str_set_x:N = \\l_problems_solution_type_str ,
8098
                   .tl_set:N
                                = \l__problems_solution_title_tl
8099
8100
   \cs_new_protected:Nn \__problems_solution_args:n {
8101
     \str_clear:N \l__problems_solution_id_str
     \verb|\str_clear:N \l_problems_solution_type_str|\\
8103
     \tr_clear: N \l_problems_solution_for_str
8104
     \tl_clear:N \l__problems_solution_title_tl
     \keys_set:nn { problem / solution }{ #1 }
8106
8107 }
```

\startsolutions

for the \startsolutions macro we use the \specialcomment macro from the comment package. Note that we use the \@startsolution macro in the start codes, that parses the optional argument.

```
\box new:N \setminus l problems solution box
8108
              \newenvironment{solution}[1][]{
8109
                      \__problems_solution_args:n{#1}
8110
                      \stex_html_backend:TF{
8111
                              \stex_if_do_html:T{
8112
                                      \begin{stex_annotate_env}{solution}{}
8113
                                              \str_if_empty:NF \l__problems_solution_type_str {
8114
                                                      \stex_annotate_invisible:nnn{typestrings}{\sexampletype}{}
8115
8116
                                              8117
8118
8119
                              \setbox\l__problems_solution_box\vbox\bgroup
8120
                                      \par\smallskip\hrule\smallskip
8121
                                      \label{lem:lemble_loss} $$ \operatorname{loss}_{solution}_{tl_if_empty:NF\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_title_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_problems_solution_tl^{-}(\l_
8122
8123
8124 }{
                      \stex_html_backend:TF{
8125
                              \stex if do html:T{
8126
                                      \end{stex_annotate_env}
8127
```

```
8129
                                                                   \smallskip\hrule
                                              8130
                                                                   \egroup
                                              8131
                                                                   \bool_if:NT \c__problems_solutions_bool {
                                              8132
                                                                         \box\l_problems_solution_box
                                              8133
                                              8134
                                              8135
                                              8136
                                              8137
                                                        \newcommand\startsolutions{
                                              8138
                                                              \verb|\bool_set_true:N \ \verb|\c_problems_solutions_bool||
                                              8139
                                                              \solutionstrue
                                              8140
                                                                \specialcomment{solution}{\@startsolution}{
                                              8141 %
                                                                       \bool_if:NF \c__problems_boxed_bool {
                                              8142 %
                                              8143 %
                                                                            \hrule\medskip
                                              8144 %
                                              8145
                                                       %
                                                                      \end{small}%
                                                               }
                                              8146
                                                       %
                                                                \bool_if:NT \c__problems_boxed_bool {
                                              8148 %
                                                                      \surroundwithmdframed{solution}
                                              8149 %
                                                               }
                                              8150 }
                                             (End definition for \startsolutions. This function is documented on page 63.)
\stopsolutions
                                              {\tt 8151} \ {\tt newcommand \ top solutions \{ bool\_set\_false: N \ c\_problems\_solutions\_bool \ \ solutions false \} \% \ extends on the solution of the solution o
                                             (End definition for \stopsolutions. This function is documented on page 63.)
                      exnote
                                                        \bool_if:NTF \c__problems_notes_bool {
                                              8152
                                                              \newenvironment{exnote}[1][]{
                                              8153
                                                                   \par\smallskip\hrule\smallskip
                                              8154
                                                                   \noindent\textbf{\prob@note@kw :~ }\small
                                              8155
                                              8156
                                                                    \smallskip\hrule
                                              8157
                                              8158
                                              8159 }{
                                                              \excludecomment{exnote}
                                              8161 }
                           hint
                                                         \bool_if:NTF \c__problems_notes_bool {
                                              8162
                                                              \newenvironment{hint}[1][]{
                                              8163
                                              8164
                                                                   \par\smallskip\hrule\smallskip
                                              8165
                                                                   \noindent\textbf{\prob@hint@kw :~ }\small
                                              8166
                                                             }{
                                                                   \smallskip\hrule
                                              8167
                                              8168
                                                              \newenvironment{exhint}[1][]{
                                              8169
                                                                   \par\smallskip\hrule\smallskip
                                              8170
                                                                   \noindent\textbf{\prob@hint@kw :~ }\small
                                              8171
```

}

8128

```
\smallskip\hrule
         8173
         8174
         8175 }{
               \excludecomment{hint}
         8176
               \excludecomment{exhint}
         8178 }
gnote
             \verb|\bool_if:NTF| \verb|\c_problems_notes_bool| \{
               \newenvironment{gnote}[1][]{
         8180
                 \par\smallskip\hrule\smallskip
         8181
                 \noindent\textbf{\prob@gnote@kw :~ }\small
         8182
         8183
                 \smallskip\hrule
         8184
               \excludecomment{gnote}
         8188 }
```

39.3 Marup for Added Value Services

39.4 Multiple Choice Blocks

EdN:12

```
12
mcb
          \newenvironment{mcb}{
            \begin{enumerate}
            \end{enumerate}
      8193 }
      we define the keys for the mcc macro
          \cs_new_protected:Nn \__problems_do_yes_param:Nn {
            \exp_args:Nx \str_if_eq:nnTF { \str_lowercase:n{ #2 } }{ yes }{
               \bool_set_true:N #1
               \bool_set_false:N #1
      8198
      8199
      8200 }
          \keys_define:nn { problem / mcc }{
      8201
                       .str\_set\_x:N = \label{eq:local_str} .str\_set\_x:N = \label{eq:local_str} ,
      8202
            feedback .tl_set:N
                                       = \l__problems_mcc_feedback_tl ,
      8203
                       .default:n
                                       = { false } ,
      8204
                       .bool_set:N
                                       = \l__problems_mcc_t_bool ,
                       .default:n
                                       = { false } ,
                                       = \label{local_problems_mcc_f_bool} ,
                       .bool_set:N
                                       = \1_problems_mcc_Ttext_tl
            Ttext
                       .tl_set:N
                                       = \l__problems_mcc_Ftext_tl
                       .tl_set:N
      8209
            Ftext
      8210 }
      8211 \cs_new_protected:Nn \l__problems_mcc_args:n {
```

 $^{^{12}\}mathrm{EdNote}\colon\operatorname{MK:}$ maybe import something better here from a dedicated MC package

```
\str_clear:N \l__problems_mcc_id_str
                                        \tl_clear:N \l__problems_mcc_feedback_tl
                     8213
                                        \bool_set_false:N \l__problems_mcc_t_bool
                      8214
                                        \verb|\bool_set_false:N \l| \_problems_mcc_f\_bool|
                     8215
                                        \tl_clear:N \l__problems_mcc_Ttext_tl
                     8216
                                        \tl_clear:N \l__problems_mcc_Ftext_tl
                     8217
                                        \str_clear:N \l__problems_mcc_id_str
                     8218
                                        \keys_set:nn { problem / mcc }{ #1 }
                     8220 }
\mcc
                                 \def\mccTrueText{\textbf{\prob@correct@kw!~}}
                                  \def\mccFalseText{\textbf{\prob@wrong@kw!~}}
                                  \newcommand\mcc[2][]{
                                        \l__problems_mcc_args:n{ #1 }
                     8224
                                        \left[ \mathbb{S} \right] #2
                                        \bool_if:NT \c__problems_solutions_bool{
                                              11
                                              \verb|\bool_if:NT \l|\_problems_mcc_t_bool| \{
                      8228
                                                     \verb|\tl_if_empty:NTF|l_problems_mcc_Ttext_tl| mccTrueText|l_problems_mcc_Ttext_tl| mccTrueText_tl| mcc
                      8229
                     8230
                                              \bool_if:NT \l__problems_mcc_f_bool {
                      8231
                                                      \t l_if_empty:NTF \ l_problems_mcc_Ttext_tl \ mccFalseText \ l_problems_mcc_Ftext_tl
                      8232
                      8233
                                              \tl_if_empty:NF \l__problems_mcc_feedback_tl {
                                                      \emph{\l__problems_mcc_feedback_tl}
                     8237
                     8238 } %solutions
```

39.5 Filling in Concrete Solutions

(End definition for \mcc. This function is documented on page 64.)

\includeproblem This is embarrasingly simple, but can grow over time.

```
8239 \newcommand\fillinsol[1]{\quad%
8240 \ifsolutions\textcolor{red}{#1!}\else%
8241 \fbox{\phantom{\huge{#1}}}%
8242 \fi}
```

(End definition for \includeproblem. This function is documented on page 66.)

39.6 Including Problems

\includeproblem

The \includeproblem command is essentially a glorified \input statement, it sets some internal macros first that overwrite the local points. Importantly, it resets the inclprob keys after the input.

```
= \l__problems_inclprob_title_tl,
              .tl_set:N
     title.
              .int_set:N
                             = \l_problems_inclprob_refnum_int,
8249
     refnum
                             = \l__problems_inclprob_type_tl,
              .tl set:N
8250
     type
     mhrepos .str_set_x:N = \l__problems_inclprob_mhrepos_str
8251
8252 }
    \cs_new_protected:Nn \__problems_inclprob_args:n {
8253
      \str_clear:N \l__problems_prob_id_str
8254
      \tl_clear:N \l__problems_inclprob_pts_tl
8255
      \tl_clear:N \l__problems_inclprob_min_tl
      \tl_clear:N \l__problems_inclprob_title_tl
8257
      \tl_clear:N \l__problems_inclprob_type_tl
8258
      \int_zero_new:N \l__problems_inclprob_refnum_int
8250
      \verb|\str_clear:N \l_problems_inclprob_mhrepos_str|\\
8260
      \keys_set:nn { problem / inclproblem }{ #1 }
8261
      \tl_if_empty:NT \l__problems_inclprob_pts_tl {
8262
        \let\l__problems_inclprob_pts_tl\undefined
8263
8264
      \tl_if_empty:NT \l__problems_inclprob_min_tl {
8265
        \let\l__problems_inclprob_min_tl\undefined
      \tl_if_empty:NT \l__problems_inclprob_title_tl {
8268
        8269
8270
      \tl_if_empty:NT \l__problems_inclprob_type_tl {
8271
        \let\l__problems_inclprob_type_tl\undefined
8272
8273
      \int_compare:nNnT \l__problems_inclprob_refnum_int = 0 {
8274
        \let\l__problems_inclprob_refnum_int\undefined
8275
8276
8277
8278
    \cs_new_protected:Nn \__problems_inclprob_clear: {
8279
8280
      \label{lems_inclprob_id_str} \
      \label{lem:lems_inclprob_pts_tl} $$ \left( \sum_{j=1}^{n} \frac{1}{j} \right) = \frac{1}{n} . $$
8281
      \left( 1_{problems_inclprob_min_t1 \right) 
8282
      \let\l__problems_inclprob_title_tl\undefined
8283
      \let\l__problems_inclprob_type_tl\undefined
8284
      \let\l__problems_inclprob_refnum_int\undefined
8285
8286
      \label{lems_inclprob_mhrepos_str} \
8287
    \__problems_inclprob_clear:
8290
    \newcommand\includeproblem[2][]{
      \__problems_inclprob_args:n{ #1 }
8291
      \exp_args:No \stex_in_repository:nn\l__problems_inclprob_mhrepos_str{
8292
        \stex html backend:TF {
8293
          \str_clear:N \l_tmpa_str
8294
          \prop_get:NnNF \l_stex_current_repository_prop { narr } \l_tmpa_str {
8295
            \prop_get:NnNF \l_stex_current_repository_prop { ns } \l_tmpa_str {}
8296
8297
          \stex_annotate_invisible:nnn{includeproblem}{
8299
            \1_tmpa_str / #2
          }{}
8300
        }{
8301
```

```
8302
             \begingroup
                \inputreftrue
8303
                \tl_if_empty:nTF{ ##1 }{
8304
                   \displaystyle \begin{array}{l} \ \ \ \ \ \end{array}
8305
8306
                   \input{ \c_stex_mathhub_str / ##1 / source / #2 }
8307
                }
8308
             \endgroup
8309
8311
           _problems_inclprob_clear:
8313 }
```

(End definition for \includeproblem. This function is documented on page 66.)

39.7 Reporting Metadata

For messages it is OK to have them in English as the whole documentation is, and we can therefore assume authors can deal with it.

```
\AddToHook{enddocument}{
8314
      \bool_if:NT \c__problems_pts_bool {
8315
        \message{Total:~\arabic{pts}~points}
8316
8317
      \bool_if:NT \c__problems_min_bool {
8318
        \message{Total:~\arabic{min}~minutes}
8320
8321 }
    The margin pars are reader-visible, so we need to translate
8322
      \bool_if:NT \c__problems_pts_bool {
8323
        \marginpar{#1~\prob@pt@kw}
8324
8325
    \def\min#1{
      \bool_if:NT \c__problems_min_bool {
8328
        \marginpar{#1~\prob@min@kw}
8329
8330
8331 }
```

\show@pts The \show@pts shows the points: if no points are given from the outside and also no points are given locally do nothing, else show and add. If there are outside points then we show them in the margin.

```
\newcounter{pts}
   \def\show@pts{
8333
    \tl_if_exist:NTF \l__problems_inclprob_pts_tl {
8334
      \bool_if:NT \c__problems_pts_bool {
8335
        \addtocounter{pts}{\l__problems_inclprob_pts_tl}
8337
8338
    }{
8339
      \tl_if_exist:NT \l__problems_prob_pts_tl {
8340
        \bool_if:NT \c__problems_pts_bool {
8341
```

```
\verb|\tl_if_empty:NT\l_problems_prob_pts_tl||
             8342
                             \tl_set:Nn \l__problems_prob_pts_t1 {0}
             8343
             8344
                          8345
                           \verb| add to counter {pts}{ | l\_problems\_prob\_pts\_t1}|
             8346
             8347
             8348
             8349
             8350 }
            (End definition for \show@pts. This function is documented on page ??.)
                 and now the same for the minutes
\show@min
                 \newcounter{min}
             8351
                 \def\show@min{
             8352
                   \tl_if_exist:NTF \l__problems_inclprob_min_tl {
             8353
                      \bool_if:NT \c__problems_min_bool {
             8354
                        \label{lem:lems_inclprob_pts_tl} $$ \max\{l_problems_inclprob_pts_tl\ min\}$$
             8355
                        \addtocounter{min}{\l__problems_inclprob_min_tl}
             8357
                   }{
             8358
                      \verb|\tl_if_exist:NT \l_problems_prob_min_tl| \{
             8359
                        \verb|\bool_if:NT \c__problems_min_bool| \{
             8360
                          \verb|\tl_if_empty:NT\l__problems_prob_min_tl| \{
             8361
                             \tl_set:Nn \l__problems_prob_min_t1 {0}
             8362
             8363
                          \label{lems_prob_min_tl} $$\max\{l_problems_prob_min_tl\ min\}$$
             8364
                           \addtocounter{min}{\l_problems_prob_min_tl}
             8365
             8369 }
             8370 \langle /package \rangle
            (End definition for \show@min. This function is documented on page ??.)
```

Chapter 40

Implementation: The hwexam Package

40.1 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.

```
8371 (*package)
8372 \ProvidesExplPackage{hwexam}{2022/08/08}{3.2.0}{homework assignments and exams}
8373 \RequirePackage{l3keys2e}
8374
8375 \newif\iftest\testfalse
8376 \DeclareOption{test}{\testtrue}
8377 \newif\ifmultiple\multiplefalse
8378 \DeclareOption{multiple}{\multipletrue}
8379 \DeclareOption{multiple}{\multipletrue}
8379 \DeclareOption{lang}{\PassOptionsToPackage{\CurrentOption}{problem}}
8380 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
8381 \ProcessOptions
Then we make sure that the necessary packages are loaded (in the right versions).
8382 \RequirePackage{keyval}[1997/11/10]
8383 \RequirePackage{problem}
```

\hwexam@*@kw

For multilinguality, we define internal macros for keywords that can be specialized in *.ldf files.

 ${\it 8394} \ \ \texttt{Newcommand} \ \texttt{correctionQforgradingQkw{To-be-used-for-grading,-do-not-write-here}}$

```
(End definition for \hwexam@*@kw. This function is documented on page ??.)
   For the other languages, we set up triggers
8395 \AddToHook{begindocument}{
8396 \ltx@ifpackageloaded{babel}{
8397 \makeatletter
8398 \clist_set:Nx \l_tmpa_clist {\bbl@loaded}
\input{hwexam-ngerman.ldf}
8400
8401 }
\label{lem:lem:lem:nt} $$ \exp_{args:NNx \ clist_if_in:NnT \ l_tmpa_clist {\ detokenize{finnish}}} $$
     \input{hwexam-finnish.ldf}
8405 \exp_args:NNx \clist_if_in:NnT \l_tmpa_clist {\detokenize{french}}{
     \input{hwexam-french.ldf}
8407
\input{hwexam-russian.ldf}
8409
8410 }
8411 \makeatother
8412 }{}
8413 }
8414
```

40.2 Assignments

8429 \str_clear:N \l_@@_assign_id_str

8437 }

8430 \int_set:Nn \l_@@_assign_number_int {-1} 8431 \tl_clear:N \l_@@_assign_title_tl 8432 \tl_clear:N \l_@@_assign_type_tl 8433 \tl_clear:N \l_@@_assign_given_tl 8434 \tl_clear:N \l_@@_assign_due_tl

8435 \bool_set_false:N \l_@@_assign_loadmodules_bool 8436 \keys_set:nn { hwexam / assignment }{ #1 }

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.

```
8415 \newcounter{assignment}
8416 %\numberproblemsin{assignment}
We will prepare the keyval support for the assignment environment.
8417 \keys_define:nn { hwexam / assignment } {
8418 id .str_set_x:N = \l_@@_assign_id_str,
8419 number .int_set:N = \l_@@_assign_number_int,
8420 title .tl_set:N = \l_@@_assign_title_tl,
8421 type .tl_set:N = \l_@@_assign_type_tl,
8422 given .tl_set:N = \l_@@_assign_given_tl,
8423 due .tl_set:N = \l_@@_assign_due_tl,
8424 loadmodules .code:n = {
8425 \bool_set_true:N \l_@@_assign_loadmodules_bool
8426 }
8427 }
8428 \cs_new_protected:Nn \_@@_assignment_args:n {
```

The next three macros are intermediate functions that handle the case gracefully, where the respective token registers are undefined.

The \given@due macro prints information about the given and due status of the assignment. Its arguments specify the brackets.

```
8438 \newcommand\given@due[2]{
8439 \bool_lazy_all:nF {
8440 {\tl_if_empty_p:V \l_@@_inclassign_given_tl}
8441 \{\tl_if_empty_p: V \l_00_assign_given_tl\}
8442 {\tilde{p}:V l_0@_inclassign_due_tl}
8443 {\tl_if_empty_p:V \l_@@_assign_due_tl}
8444 }{ #1 }
8445
8446 \tl_if_empty:NTF \l_@@_inclassign_given_tl {
   \tl if empty:NF \l @@ assign given tl {
   \hwexam@given@kw\xspace\l_@@_assign_given_tl
8450 }{
   \hwexam@given@kw\xspace\l_@@_inclassign_given_tl
8452
8453
8454 \bool_lazy_or:nnF {
8455 \bool_lazy_and_p:nn {
8456 \tl_if_empty_p:V \l_@@_inclassign_due_tl
8457 }{
8458
   \tl_if_empty_p:V \l_@@_assign_due_tl
8460 }{
8461 \bool_lazy_and_p:nn {
   \tl_if_empty_p:V \l_@@_inclassign_due_tl
8464 \tl_if_empty_p:V \l_@@_assign_due_tl
8465 }
8466 }{ ,~ }
8467
   \tl_if_empty:NTF \l_@@_inclassign_due_tl {
   \tl_if_empty:NF \l_@@_assign_due_tl {
   \hwexam@due@kw\xspace \l_@@_assign_due_tl
8471 }
   \hwexam@due@kw\xspace \l_@@_inclassign_due_tl
8474
8475
8476 \bool_lazy_all:nF {
8477 { \t = mpty_p:V \leq 0_inclassign_given_tl }
8478 { \t_if_empty_p:V \l_@@_assign_given_tl }
8479 { \tl_if_empty_p:V \l_@@_inclassign_due_tl }
8480 { \tl_if_empty_p:V \l_@@_assign_due_tl }
8481 }{ #2 }
8482 }
```

\assignment@title This macro prints the title of an assignment, the local title is overwritten, if there is one from the \inputassignment. \assignment@title takes three arguments the first is the

fallback when no title is given at all, the second and third go around the title, if one is given.

```
8483 \newcommand\assignmentOtitle[3]{
8484 \tl_if_empty:NTF \l_QQ_inclassign_title_tl {
8485 \tl_if_empty:NTF \l_QQ_assign_title_tl {
8486 #1
8487 }{
8488 #2\l_QQ_assign_title_tl#3
8489 }
8490 }{
8491 #2\l_QQ_inclassign_title_tl#3
8492 }
8493 }
```

(End definition for \assignment@title. This function is documented on page ??.)

\assignment@number

Like \assignment@title only for the number, and no around part.

```
8494 \newcommand\assignment@number{
8495 \int_compare:nNnTF \l_@@_inclassign_number_int = {-1} {
8496 \int_compare:nNnTF \l_@@_assign_number_int = {-1} {
8497 \arabic{assignment}
8498 } {
8499 \int_use:N \l_@@_assign_number_int
8500 }
8501 }{
8502 \int_use:N \l_@@_inclassign_number_int
8503 }
8504 }
```

 $(\mathit{End \ definition \ for \ } \verb|\assignment@number|. \ \mathit{This \ function \ is \ documented \ on \ page \ \ref{eq:condition}.)}$

With them, we can define the central assignment environment. This has two forms (separated by \ifmultiple) in one we make a title block for an assignment sheet, and in the other we make a section heading and add it to the table of contents. We first define an assignment counter

 ${\tt assignment}$

For the assignment environment we delegate the work to the Cassignment environment that depends on whether multiple option is given.

```
8505 \newenvironment{assignment}[1][]{
8506 \_@@_assignment_args:n { #1 }
8507 %\sref@target
8508 \int_compare:nNnTF \l_@@_assign_number_int = {-1} {
8509 \global\stepcounter{assignment}
8510 }{
8511 \global\setcounter{assignment}{\int_use:N\l_@@_assign_number_int}
8512 }
8513 \setcounter{sproblem}{0}
8514 \renewcommand\prob@label[1]{\assignment@number.##1}
8515 \def\current@section@level{\document@hwexamtype}
8516 %\sref@label@id{\document@hwexamtype \thesection}
8517 \begin{@assignment}
8518 }{
8519 \end{@assignment}
8520 }
```

In the multi-assignment case we just use the omdoc environment for suitable sectioning.

```
8521 \def\ass@title{
8522 {\protect\document@hwexamtype}~\arabic{assignment}
% assignment@title{}{\;(){})\;} -- \given@due{}{}
8524
8525 \ifmultiple
8526 \newenvironment{@assignment}{
8527 \bool_if:NTF \l_@@_assign_loadmodules_bool {
8528 \begin{sfragment}[loadmodules]{\ass@title}
8530 \begin{sfragment}{\ass@title}
8531 }
8532 }{
8533 \end{sfragment}
8534 }
for the single-page case we make a title block from the same components.
8536 \newenvironment{@assignment}{
8537 \begin{center}\bf
8538 \Large\@title\strut\\
8539 \document@hwexamtype~\arabic{assignment}\assignment@title{\;}{:\;}{\\}
8540 \large\given@due{--\;}{\;--}
8541 \end{center}
8542 }{}
8543 \fi% multiple
```

40.3 Including Assignments

\in*assignment

This macro is essentially a glorified \include statement, it just sets some internal macros first that overwrite the local points Importantly, it resets the inclassig keys after the input.

```
8544 \keys_define:nn { hwexam / inclassignment } {
8545 %id .str_set_x:N = \l_@@_assign_id_str,
8546 number .int_set:N = \lambda[@@_inclassign_number_int,
8547 title .tl_set:N = \l_@@_inclassign_title_tl,
ss48 type .tl_set:N = \l_@@_inclassign_type_tl,
8549 given .tl set:N = \label{eq:N} = \label{eq:N} 00 inclassign given tl,
8550 due .tl_set:N = \l_@@_inclassign_due_tl,
8551 mhrepos .str_set_x:N = \l_@@_inclassign_mhrepos_str
8552 }
8553 \cs_new_protected:Nn \_@@_inclassignment_args:n {
8554 \int_set:Nn \l_@@_inclassign_number_int {-1}
8556 \tl_clear:N \l_@@_inclassign_type_tl
8557 \tl_clear:N \l_@@_inclassign_given_tl
8558 \tl_clear:N \l_@@_inclassign_due_tl
8559 \str_clear:N \l_@@_inclassign_mhrepos_str
8560 \keys_set:nn { hwexam / inclassignment }{ #1 }
8561
8562
   \ @@ inclassignment args:n {}
8564 \newcommand\inputassignment[2][]{
```

```
8565 \_@@_inclassignment_args:n { #1 }
8566 \str_if_empty:NTF \l_@@_inclassign_mhrepos_str {
8567 \input{#2}
8568 }{
8569 \stex_in_repository:nn{\l_@@_inclassign_mhrepos_str}{
8570 \input{\mhpath{\l_@@_inclassign_mhrepos_str}{#2}}
8571 }
8572 }
8573 \_@@_inclassignment_args:n {}
8574 }
8574 \newcommand\includeassignment[2][]{
8576 \newcommand\includeassignment[#1]{#2}
8577 \inputassignment[#1]{#2}
8578 }
(End definition for \in*assignment. This function is documented on page ??.)
```

40.4 Typesetting Exams

```
\quizheading
```

```
8579 \ExplSyntaxOff
8580 \newcommand\quizheading[1]{%
8581 \def\@tas{#1}%
8582 \large\noindent NAME: \hspace{8cm} MAILBOX:\\[2ex]%
8583 \ifx\@tas\@empty\else%
8584 \noindent TA:~\@for\@I:=\@tas\do{{\Large$\Box$}\@I\hspace*{1em}}\\[2ex]%
8585 \fi%
8586 }
8587 \ExplSyntaxOn
(End definition for \quizheading. This function is documented on page ??.)
```

\testheading

```
\def\hwexamheader{\input{hwexam-default.header}}
8589
8590
   \def\hwexamminutes{
   \tl_if_empty:NTF \testheading@duration {
   {\testheading@min}~\hwexam@minutes@kw
   \testheading@duration
8597 }
8598
_{\it 8599} \keys_define:nn { hwexam / testheading } {
8600 min .tl_set:N = \testheading@min,
8601 duration .tl_set:N = \testheading@duration,
8602 reqpts .tl_set:N = \testheading@reqpts,
s603 tools .tl_set:N = \testheading@tools
8604 }
8605 \cs_new_protected:Nn \_@@_testheading_args:n {
8606 \tl_clear:N \testheading@min
8607 \tl_clear:N \testheading@duration
```

```
8613 \_@@_testheading_args:n{ #1 }
                                       8614 \newcount\check@time\check@time=\testheading@min
                                       8615 \advance\check@time by -\theassignment@totalmin
                                        8616 \newif\if@bonuspoints
                                       8617 \tl_if_empty:NTF \testheading@reqpts {
                                       8618 \@bonuspointsfalse
                                       8619 }{
                                       8620 \newcount\bonus@pts
                                       8621 \bonus@pts=\theassignment@totalpts
                                       8622 \advance\bonus@pts by -\testheading@reqpts
                                                \edef\bonus@pts{\the\bonus@pts}
                                                \@bonuspointstrue
                                        8625
                                               \edef\check@time{\the\check@time}
                                       8628 \makeatletter\hwexamheader\makeatother
                                       8629 }{
                                       8630 \newpage
                                       8631 }
                                      (End definition for \testheading. This function is documented on page ??.)
         \testspace
                                       % \newcommand\testspace[1] {\left(\frac{1}{\left(\frac{1}{1}\right)}\right)}
                                      (End definition for \testspace. This function is documented on page ??.)
    \testnewpage
                                       8633 \newcommand\testnewpage{\iftest\newpage\fi}
                                      (End definition for \testnewpage. This function is documented on page ??.)
\testemptypage
                                       %634 \newcommand\testemptypage[1][]{\iftest\begin{center}\hwexam@testemptypage@kw\end{center}\vfi
                                      (End definition for \testemptypage. This function is documented on page ??.)
            \@problem
                                     This macro acts on a problem's record in the *.aux file. Here we redefine it (it was
                                      defined to do nothing in problem.sty) to generate the correction table.
                                       8635 (@@=problems)
                                       8636 \renewcommand\@problem[3]{
                                       8637 \stepcounter{assignment@probs}
                                       8638 \def\__problemspts{#2}
                                       8639 \ifx\__problemspts\@empty\else
                                       8640 \addtocounter{assignment@totalpts}{#2}
                                       8641 \fi
                                       \label{lem:bound} $$  \def\_problemsmin{#3} ifx\_problemsmin\\@empty\\else\\add to counter{assignment @totalmin}{#3} ifx\_problemsmin\\add to counter{assignment @totalmin}{*3} ifx\_problemsmin\\add to counter{assignment @to
                                       8643 \xdef\correction@probs{\correction@probs & #1}%
                                       8644 \xdef\correction@pts{\correction@pts & #2}
                                       8645 \xdef\correction@reached{\correction@reached &}
```

 $8608 \ \text{tl_clear:N } \ \text{testheading@reqpts}$ 8609 \tl_clear:N \testheading@tools

8612 \newenvironment{testheading}[1][]{

8611 }

8610 \keys_set:nn { hwexam / testheading }{ #1 }

```
8646 }
                 8647 (@@=hwexam)
                (End definition for \Cproblem. This function is documented on page ??.)
\correction@table This macro generates the correction table
                 8648 \newcounter{assignment@probs}
                 8649 \newcounter{assignment@totalpts}
                 8650 \newcounter{assignment@totalmin}
                 8651 \def\correction@probs{\correction@probs@kw}
                 8652 \def\correction@pts(\correction@pts@kw)
                 8653 \def\correction@reached{\correction@reached@kw}
                 8654 \stepcounter{assignment@probs}
                 8655 \newcommand\correction@table{
                 8656 \resizebox{\textwidth}{!}{%
                 8659 {\footnotesize\correction@forgrading@kw} &\\\hline
                 8661 \correction@pts &\theassignment@totalpts & \\\hline
                 8662 \correction@reached & & \\[.7cm]\hline
                 8663 \end{tabular}}}
                 8664 (/package)
                (End definition for \correction@table. This function is documented on page ??.)
```

40.5 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\char65}} \newcommand\hardA{\warnschildfont\char65}} \newcommand\hardA{\warnschild} \newcommand\hardA{\warnschild} \newcommand\hardA{\uhr} \newcommand\hardA{\uhr} \newcommand\hardA{\uhr} \newcommand\discussA{\uhrganignments}} \newcommand\discussA{\uhrganignments}
```

Chapter 41

References

EdN:13

13

- [Bus+04] Stephen Buswell et al. The Open Math Standard, Version 2.0. Tech. rep. The OpenMath Society, 2004. URL: http://www.openmath.org/standard/om20.
- [CR99] David Carlisle and Sebastian Rathz. The graphicxl package. Part of the TEX distribution. The Comprehensive TEX Archive Network. 1999. URL: https://www.tug.org/texlive/devsrc/Master/texmf-dist/doc/latex/graphics/graphicx.pdf.
- [DCM03] The DCMI Usage Board. *DCMI Metadata Terms*. DCMI Recommendation. Dublin Core Metadata Initiative, 2003. URL: http://dublincore.org/documents/dcmi-terms/.
- [Koh06] Michael Kohlhase. OMDoc An open markup format for mathematical documents [Version 1.2]. LNAI 4180. Springer Verlag, Aug. 2006. URL: http://omdoc.org/pubs/omdoc1.2.pdf.
- [LMH] LMH Scripts. URL: https://github.com/sLaTeX/lmhtools.
- [MMT] MMT Language and System for the Uniform Representation of Knowledge. Project web site. URL: https://uniformal.github.io/ (visited on 01/15/2019).
- [MRK18] Dennis Müller, Florian Rabe, and Michael Kohlhase. "Theories as Types". In: 9th International Joint Conference on Automated Reasoning. Ed. by Didier Galmiche, Stephan Schulz, and Roberto Sebastiani. Springer Verlag, 2018. URL: https://kwarc.info/kohlhase/papers/ijcar18-records.pdf.
- [Rab15] Florian Rabe. "The Future of Logic: Foundation-Independence". In: *Logica Universalis* 10.1 (2015). 10.1007/s11787-015-0132-x; Winner of the Contest "The Future of Logic" at the World Congress on Universal Logic, pp. 1–20.
- [RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: Information & Computation 0.230 (2013), pp. 1–54. URL: https://kwarc.info/frabe/Research/mmt.pdf.
- [RT] sLaTeX/RusTeX. URL: https://github.com/sLaTeX/RusTeX (visited on 04/22/2022).

 $^{^{13}\}mathrm{EdNote}$: we need an un-numbered version sfragment*

- [SIa] sLaTeX/sTeX-IDE. URL: https://github.com/slatex/sTeX-IDE (visited on 04/22/2022).
- [SIb] sLaTeX/stexls-vscode-plugin. URL: https://github.com/slatex/stexls-vscode-plugin (visited on 04/22/2022).
- [SLS] sLaTeX/stexls. URL: https://github.com/slatex/stexls (visited on 04/22/2022).
- [ST] sTeX An Infrastructure for Semantic Preloading of LaTeX Documents. URL: https://ctan.org/pkg/stex (visited on 04/22/2022).
- [sTeX] sTeX: A semantic Extension of TeX/LaTeX. URL: https://github.com/sLaTeX/sTeX (visited on 05/11/2020).
- [Tana] Till Tantau. beamer A LaTeX class for producing presentations and slides. URL: http://ctan.org/pkg/beamer (visited on 01/07/2014).
- [Tanb] Till Tantau. User Guide to the Beamer Class. URL: http://ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf.
- [TL] TeX Live. URL: http://www.tug.org/texlive/ (visited on 12/11/2012).