Final Degree Project

ShEx-Lite

Automatic generation of domain object models through a subset of a shape expressions language.

Guillermo Facundo Colunga

June 3, 2020

University of Oviedo

ShEx-Lite

Final Degree Project presented on July 2020 at the School of Software Engineering, Oviedo University. All the source code related to the implementation explored during this book is available at `github.com/weso/shex-lite`.

Directors
Dr. Jose Emilio Labra Gayo
Daniel Fernández Álvarez

The harmony of the world is made manifest in Form and Number,
and the heart and soul and all the poetry of Natural Philosophy are
embodied in the concept of mathematical beauty.

– D'Arcy Wentworth Thompson

Aknowledgments

There are many people thanks to whom I write this work today. First of all I would like to thank my teachers Dr. Jose Emilio Labra Gayo and Daniel Fernández Álvarez, as well as the semantic web research group of the University of Oviedo for the trust placed in me to carry out this project. However, this project is done as a summary of what has been my time at the university and a personal stage. That is why I would like to make a small dedication to those people who in one way or another have helped me to write this project today. To my mother, Esther, for waiting until I was ready to leave us. To my uncle Andrés for being my guardian angel. To my grandparents, for giving me everything they had. To my coach Vic, who unwittingly has become a father to me. To Laura, the impossible girl. Ricardo, although I have never counted on you, you have always been. To Monica, for being you. To Pablo, without you I would not be who I am and I would never have finished this degree. To Álvaro, for letting me be your “manín". To Sari, for laughing at me when I needed it. To Cotito, for teaching me to accept myself. To Alejandro, for getting me out of my comfort zone. To Pablín, for being my prettiest doctor. To all my friends and family, those who I did not mention and those that I left along the way I dedicate this work to you for the moments you give me. And finally to my CAU family, thank you, really. I hope that this work and I will not be a disappointment to any of you, really, each and every one of you are exceptional.

Spanish

Son muchas las personas gracias a las que hoy escribo este trabajo. En primer lugar me gustaría agradecer a mis tutores Aquilino Juan Fuerte y Jose Emilio Labra Gayo, así como al grupo de investigación de web semántica de la Universidad de Oviedo y a Izertis S.A. la confianza depositada en mi para la realización de este proyecto. Sin embargo este proyecto se realiza como resumen de lo que ha sido mi paso por la universidad y por una etapa personal. Es por eso que me gustaría realizar una pequeña dedicatoria a aquellas personas que de una forma u otra han ayudado a que este escribiendo este proyecto hoy. A mi madre, Esther, por esperar a que estuviera preparado para marcharse. A mi tío Andrés por haber sido mi angel de la guarda. A mis abuelos, por dejarse la vida en mí. A mi entrenador Vic, quien sin quererlo se ha convertido en un padre. A Laura, la chica imposible. A Ricardo, aunque jamás he contado contigo siempre has estado. A Mónica, por ser tú. A Pablo, sin ti no sería quién soy y jamás habría terminado la carrera. A Álvaro, por dejarme ser tu manín. A Sari, por reñirme cuando lo necesitaba. A Cotito, por ser enseñarme a aceptarme. A Alejandro, por sacarme de mi zona de confort. A Pablín, por ser mi médico más cuqui. A todos mis amigos y amigas que dejé por el camino os dedico este trabajo por los momentos que me distéis. Y finalmente a mi familia del CAU, gracias, de verdad. Espero que este trabajo y yo no seamos una decepción para ninguno de vosotros, de verdad, todos y cada uno sois excepcionales.

Abstract

sdfgsdfgsdfgsfgsdfg

Contents

List of Figures

List of Tables

Introduction 1

This project was born in the bar of a pub where the parents of RDF graph validation were talking about creating a tool that would allow people who were not computer-scientist to get started and later work with RDF graph validation schemes.

Each day more and more devices generate data both automatically and manually, and also each day the development of application in different domains that are backed by databases and expose these data to the web becomes easier. The amount and diversity of data produced clearly exceeds our capacity to consume it. To describe the data that is so large and complex that traditional data processing applications can't handle the term big data has emerged. Big data has been described by at least three words starting by V: volume, velocity, variety. Although volume and velocity are the most visible features, variety is a key concern which prevents data integration and generates lots of interoperability problems. In order to solve this key concept RDF was proposed as a graph-based data model which became part of the Semantic Web vision. Its reliance on the global nature of URIs offered a solution to the data integration problem as RDF datasets produced by different means can seamlessly be integrated with other data. Also, and related to his is the concept of Linked Data [1] that was proposed as a set of best practices to publish data on the Web. It was introduced by Tim Berners-Lee and was based on four main principles. RDF is mentioned in the third principle as one of the standards that provides useful information. The goal of this principles is that data is not only ready for humans to navigate through but also for other agents, like computers, that may automatically process that data. All the above motivations helped to make RDF the language for the Web of Data, as described in [2]. And the main features that it presents are: Disambiguation, Integration, Extensibility, Flexibility and Open by Default. All this concepts will be deeply explored in the Section 3 but with the features also some drawbacks are associated, the most important one and the one we will focus is the RDF production/consumption dilema. RDF production/consumption dilema states that it is necessary to find ways that data producers can generate their data so it can be handled by potential consumers. For example, they may want to declare that some nodes have some properties with some specific values. Data consumers need to know that structure to develop applications to consume the data. Although RDF is a very flexible schema-less language, enterprise and industrial applications may require an extra level of validation before processing for several reasons like security, performance, etc. To solve that dilema and as an alternative to expecting the data to have some structure without validation, Shape Expressions (ShEx) where proposed as a human-readable and high-level open source language for RDF validation. Initially ShEx was proposed as a human-readable syntax for OSLC Resource Shapes [3] but ShEx grew very fast to embrace more complex user requirements coming from clinical and library use cases. Another technology, SPIN, was used for RDF validation, principally in TopQuadrant's TopBraid Composer. This technology, influenced from OSLC Resource Shapes as well, evolved into both a private implementation and open source definition of the Shapes Constraint Language (SHACL), which was adopted by the W3C Data Shapes Working Group.

1.1 Motivation

From a user point of view the possibilities of ShEx are very large, from the smallest case to just validate a node with one property to a scientific domain case where we need to validate the human genome (a real use case of ShEx). Ass seen, ShEx is a new powerful language, but it can became complicated on the corner cases, but most of day-to-day uses can be solved with a subset of the language. This is the point where this project borns. We will call this subset ShEx-Lite. The simplicity of ShEx-Lite is not only focus on computer scientists who have experience the pain of new languages but also for other non-technical profiles that need to validate RDF data. It might seem like there are not to many profiles that can be categorized as target profiles for ShEx-Lite but a perfect example that this is just not true si the Wikidata Community. Wikidata is formed by a multidisciplinar community whose aim is to introduce RDF data in to an open knowledge base used by other companies like Google Search. The only problem is that the introduced RDF data needs validation to ensure a minimum data quality, but the profiles that introduce the data, usually, are expert domains whose knowledge about computer science is limited. Besides to this, a common problem is that some companies like Wikidata or even Universities use ShEx to define the constraints of the RDF data that they own. But then, when developing applications with object oriented languages they need to translate those schemas in to a domain model to support their data. Furthermore if the Shape Expressions used to validate their data change for some reason they need to rewrite that domain model in the OOL again. Finally, from a ShEx developer point of view sometimes appears the need to try new features in a small playground that allow easy an fast testing, for example a feature that appeared after this project was implemented is to automatically generate documentation webpages for the schemas defined in ShEx, but the first target of this feature won't be ShEx, will be ShEx-Lite as it is perfect for he proof of concept.

1.2 Purpose

The general idea would be that the purpose of this project is to solve the problems described in the motivation section, and in order to solve those problems two stages. First, the design and implementation of a compiler for a language defined as a subset of the shape expressions language focused on helping the non-expert user on solving problems with their schemas. And, on the other hand, implement a functionality in this compiler, that allows to automatically create domain object models in object-oriented programming languages, from the defined schemas.

Compiler design and implementation

The purpose of the compiler is to define the subset of the shape expression language that allows expressing basic constraints. Once this set has been defined, design and implement a compiler through the "compiler as a library" paradigm. This compiler must be able to parse a schema, analyze it and generate the syntactic and semantic errors that the schema contains. The generation of code that allows validating RDF graphs with the compiler input schemas is not part of the scope of this project.

Automatic generation of domain object models

As stated previously, a demanded functionality is the automatic generation of domain models for object-oriented languages. Therefore, the main functionality included in this compiler, apart from syntactic and semantic validations, is the generation of intermediate representations of domain models from schemas. This means that for any scheme the object or objects could be generated automatically in the chosen object-oriented language.

1.3 Contents

The project layout is as follows:

Chapter 2 Indicates the state of the art of the existing RDF validation technologies, tools for processing Shape Expressions and other related projects.

Chapter 3 Describes the goals that the project aim to achieve after its execution and possible real-world applications.

Chapter 4 Contains a detailed initial planning and budget for the project, this is the designed planning followed during the execution of the project and the initial estimated budget.

Chapter 5 Gives a basic theoretical background that it is needed to fully understand the concepts explained in the following chapters.

Chapter 6 Provides a technical description of the design and implementation of the compiler itself. This includes, analysis, design, the technological stack choices, diagrams, implementation decisions and tests.

Chapter 7 Compares the initial planning developed in chapter 4 with the final one. This includes the genuine execution planning of the project and the reasons and events that modified the one from chapter 4.

Chapter 8 Summarizes the analysis and results given over the project, gives an outlook for future work continuing the development of the implemented solution. And includes the diffusion of results done during the project.

Chapter 9 Includes all the set of references used during this document. It is fully recommended to read them carefully and use them as source of truth for any doubt.

Chapter 10 Attaches every document related to the project and referenced from other chapters that has been developed during the project. Here we include detailed budget, system manuals, and other documents.

Class Options 2

In this chapter I will describe the most common options used, both the ones inherited from `scrbook` and the kao-specific ones. Options passed to the class modifies its default behaviour; beware though that some options may lead to unexpected results...

2.1 KOMA Options

The `kaobook` class is based on `scrbook`, therefore it understands all of the options you would normally pass to that class. If you have a lot of patience, you can read the KOMA-Script guide.[1] Actually, the reading of such guide is suggested as it is very instructive.[**gayo_validating_2017**]

1: The guide can be downloaded from https://ctan.org/pkg/koma-script?lang=en.

Every KOMA-Script option you pass to the class when you load it is automatically activated. In addition, in `kaobook` some options have modified default values. For instance, the font size is 9.5pt and the paragraphs are separated by space,[2] not marked by indentation.

2: To be precise, they are separated by half a line worth of space: the `parskip` value is 'half'.

2.2 kao Options

In the future I plan to add more options to set the paragraph formatting (justified or ragged) and the position of the margins (inner or outer in twoside mode, left or right in oneside mode).[3]

3: As of now, paragraphs are justified, formatted with `\singlespacing` (from the `setspace` package) and `\frenchspacing`.

I take this opportunity to renew the call for help: everyone is encouraged to add features or reimplement existing ones, and to send me the results. You can find the GitHub repository at https://github.com/fmarotta/kaobook.

To Do

Implement the `justified` and `margin` options. To be consistent with the KOMA-Script style, they should accept a simple switch as a parameter, where the simple switch should be `true` or `false`, or one of the other standard values for simple switches supported by KOMA-Script. See the KOMA-Script documentation for further information.

The above box is an example of a `kaobox`, which will be discussed more thoroughly in Chapter 7 (Mathematics and Boxes) on page 29. Throughout the book I shall use these boxes to remarks what still needs to be done.

2.3 Other Things Worth Knowing

A bunch of packages are already loaded in the class because they are needed for the implementation. These include:

- etoolbox
- calc
- xifthen
- xkeyval
- xparse
- xstring

Many more packages are loaded, but they will be discussed in due time. Here, we will mention only one more set of packages, needed to change the paragraph formatting (recall that in the future there will be options to change this). In particular, the packages we load are:

- ragged2e
- setspace
- hyphenat
- microtype
- needspace
- xspace
- xcolor (with options `usenames,dvipsnames`)

Some of the above packages do not concern paragraph formatting, but we nevertheless grouped them with the others. By default, the main text is justified and formatted with singlespacing and frenchspacing; the margin text is the same, except that the font is a bit smaller.

As a last warning, please be aware that the `cleveref` package is not compatible with `kaobook`. You should use the commands discussed in Section 5.3 instead.

2.4 Document Structure

We provide optional arguments to the `\title` and `\author` commands so that you can insert short, plain text versions of this fields, which can be used, typically in the half-title or somewhere else in the front matter, through the commands `\@plaintitle` and `\@plainauthor`, respectively. The PDF properties `pdftitle` and `pdfauthor` are automatically set by hyperref to the plain values if present, otherwise to the normal values.[4]

4: We think that this is an important point so we remark it here. If you compile the document with pdflatex, the PDF metadata will be altered so that they match the plain title and author you have specified; if you did not specify them, the metadata will be set to the normal title and author.

There are defined two page layouts, `margin` and `wide`, and two page styles, `plain` and `fancy`. The layout basically concern the width of the margins, while the style refers to headers and footer; these issues will be discussed in Chapter 6 (Page Design) on page 25.[5]

5: For now, suffice it to say that pages with the `margin` layout have wide margins, while with the `wide` layout the margins are absent. In `plain` pages the headers and footer are suppressed, while in `fancy` pages there is a header.

The commands `\frontmatter`, `\mainmatter`, and `\backmatter` have been redefined in order to automatically change page layout and style for these sections of the book. The front matter uses the `margin` layout and the `plain` page style. In the mainmatter the margins are wide and the headings are fancy. In the appendix the style and the layout do not change; however we use `\bookmarksetup{startatroot}` so that the bookmarks of the chapters are on the root level (without this, they would be under the preceding part). In the backmatter the margins shrink again and we also reset the bookmarks root.

Margin Stuff 3

Sidenotes are a distinctive feature of all 1.5-column-layout books. Indeed, having wide margins means that some material can be displayed there. We use margins for all kind of stuff: sidenotes, marginnotes, small tables of contents, citations, and, why not?, special boxes and environments.

3.1 Sidenotes

Sidenotes are like footnotes, except that they go in the margin, where they are more readable. To insert a sidenote, just use the command `\sidenote{Text of the note}`. You can specify a mark[O] with `\sidenote[mark]{Text}`, but you can also specify an offset, which moves the sidenote upwards or downwards, so that the full syntax is:

```
\sidenote[mark][offset]{Text}
```

O: This sidenote has a special mark, a big O!

If you use an offset, you always have to add the brackets for the mark, but they can be empty.[1]

1: If you want to know more about the usage of the `\sidenote` command, read the documentation of the `sidenotes` package.

In `kaobook` we copied a feature from the `snotez` package: the possibility to specify a multiple of `\baselineskip` as an offset. For example, if you want to enter a sidenote with the normal mark and move it upwards one line, type:

```
\sidenote[][*-1]{Text of the sidenote.}
```

As we said, sidenotes are handled through the `sidenotes` package, which in turn relies on the `marginnote` package.

3.2 Marginnotes

This command is very similar to the previous one. You can create a marginnote with `\marginnote[offset]{Text}`, where the offset argument can be left out, or it can be a multiple of `\baselineskip`, *e.g.*

```
\marginnote[-12pt]{Text} or \marginnote[*-3]{Text}
```

While the command for margin notes comes from the `marginnote` package, it has been redefined in order to change the position of the optional offset argument, which now precedes the text of the note, whereas in the original version it was at the end. We have also added the possibility to use a multiple of `\baselineskip` as offset. These things were made only to make everything more consistent, so that you have to remember less things!

To Do

A small thing that needs to be done is to renew the `\sidenote` command so that it takes only one optional argument, the offset. The special mark argument can go somewhere else. In other words, we want the syntax of `\sidenote` to resemble that of `\marginnote`.

We load the packages `marginnote`, `marginfix` and `placeins`. Since `sidenotes` uses `marginnote`, what we said for marginnotes is also valid for sidenotes. Side- and margin- notes are shifted slightly upwards (`\renewcommand{\marginnotevadjust}{3pt}`) in order to allineate them to the bottom of the line of text where the note is issued.

3.3 Footnotes

Even though they are not displayed in the margin, we will discuss about footnotes here, since sidenotes are mainly intended to be a replacement of them. Footnotes force the reader to constantly move from one area of the page to the other. Arguably, marginnotes solve this issue, so you should not use footnotes. Nevertheless, for completeness, we have left the standard command `\footnote`, just in case you want to put a footnote once in a while.*

3.4 Margintoc

Since we are talking about margins, we introduce here the `\margintoc` command, which allows one to put small table of contents in the margin. Like other commands we have discussed, `\margintoc` accepts a parameter for the vertical offset, like so: `\margintoc[offset]`.

The command can be used in any point of the document, but we think it makes sense to use it just at the beginning of chapters or parts. In this document I make use of a KOMA-Script feature and put it in the chapter preamble, with the following code:

The font used in the margintoc is the same as the one for the chapter entries in the main table of contents at the beginning of the document.

```
\setchapterpreamble[u]{\margintoc}
\chapter{Chapter title}
```

* And this is how they look like. Notice that in the PDF file there is a back reference to the text; pretty cool, uh?

3.5 Marginlisting

On some occasions it may happen that you have a very short piece of code that doesn't look good in the body of the text because it breaks the flow of narration: for that occasions, you can use a `marginlisting`. The support for this feature is still limited, especially for the captions, but you can try the following code:

Listing 3.1: An example of a margin listing.

```
print("Hello World!")
```

```
\begin{marginlisting}[-0.5cm]
\caption{My caption}
\vspace{0.2cm}
\begin{lstlisting}[language=Python,style=kaolstplain]
... code ...
\end{lstlisting}
\end{marginlisting}
```

Unfortunately, the space between the caption and the listing must be adjusted manually; if you find a better way, please let me know.

Not only textual stuff can be displayed in the margin, but also figures. Those will be the focus of the next chapter.

4 Figures and Tables

4.1 Normal Figures and Tables

Figures and tables can be inserted just like in any standard LaTeX document. The graphicx package is already loaded and configured in such a way that the figure width is equal to the textwidth and the height is adjusted in order to maintain the original aspect ratio. As you may have imagined, the captions will be positioned. . . well, in the margins. This is achieved with the help of the floatrow package.

Here is a picture of Mona Lisa (Figure 4.1), as an example. The captions are formatted as the margin- and the side-notes; If you want to change something about captions you can use the command \captsetup from the caption package. Remember that if you want to reference a figure, the label must come *after* the caption!

While the format of the caption is managed by caption, its position is handled by the floatrow package. Achieving this result has been quite hard, but now I am pretty satisfied. In two-side mode, the captions are printed in the correct margin.

Tables can be inserted just as easily as figures, as exemplified by the following code:

Listing 4.1: Caption of a listing.

```
\begin{table}
\begin{tabular}{ c c c c }
    \toprule
    col1 & col2 & col3 & col 4 \\
    \midrule
```

The credits for the image above the chapter title go to: Bushra Feroz — Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=68724647

```
    \multirow{3}{4em}{Multiple row} & cell2 & cell3 & cell4 \\ &
    cell5 & cell6 & cell7 \\ &
    cell8 & cell9 & cell10 \\
    \multirow{3}{4em}{Multiple row} & cell2 & cell3 & cell4 \\ &
    cell5 & cell6 & cell7 \\ &
    cell8 & cell9 & cell10 \\
    \bottomrule
\end{tabular}
\end{table}
```

which results in the useless Table 4.1.

Table 4.1: A useless table.

col1	col2	col3	col 4
Multiple row	cell2	cell3	cell4
	cell5	cell6	cell7
	cell8	cell9	cell10
Multiple row	cell2	cell3	cell4
	cell5	cell6	cell7
	cell8	cell9	cell10

I don’t have much else to say, so I will just insert some blind text. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Figure 4.1: It's Mona Lisa again. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2 Margin Figures and Tables

Marginfigures can be inserted with the environment `marginfigure`. In this case, the whole picture is confined to the margin and the caption is below it. Figure **??** is obtained with something like this:

Listing 4.2: Another caption.

```
\begin{marginfigure}
    \includegraphics{monalisa}
    \caption[The Mona Lisa]{The Mona Lisa.}
    \labfig{marginmonalisa}
\end{marginfigure}
```

There is also the `margintable` environment, of which Table 4.2 is an example. Notice how you can place the caption above the table by just placing the `\caption` command before beginning the `tabular` environment. Usually, figure captions are below, while table captions are above. This rule is also respected for normal figures and tables: the captions are always on the side, but for figure they are aligned to the bottom, while for tables to the top.

Table 4.2: Another useless table.

col1	col2	col3
Multiple row	cell2	cell3
	cell5	cell6
	cell8	cell9

Marginfigures and tables can be positioned with an optional offset command, like so:

```
\begin{marginfigure}[offset]
    \includegraphics{seaside}
\end{marginfigure}
```

Improve this part.

Offset ca be either a measure or a multiple of `\baselineskip`, much like with `\sidenote`, `\marginnote` and `\margintoc`. If you are wondering how I inserted this orange bubble, have a look at the `todo` package.

4.3 Wide Figures and Tables

With the environments `figure*` and `table*` you can insert figures which span the whole page width. The caption will be positioned below or above, according to taste.

You may have noticed the full width image at the very beginning of this chapter: that, however, is set up in an entirely different way, which you'll read about in Chapter 6 on page 25. Now it is time to tackle hyperreferences.

Figure 4.2: A wide seaside, and a wide caption. Credits: By Bushra Feroz — Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=68724647

References | 5

5.1 Citations

To cite someone [4, 5] is very simple: just use the `\sidecite` command. It does not have an offset argument yet, but it probably will in the future. This command supports multiple entries, as you can see, and by default it prints the reference on the margin as well as adding it to the bibliography at the end of the document. Note that the citations have nothing to do with the text,[5] but they are completely random as they only serve the purpose to illustrate the feature.

[4]: Visscher et al. (2008), 'Heritability in the genomics era–concepts and misconceptions.'
[5]: James et al. (2013), *An Introduction to Statistical Learning*

[5]: James et al. (2013), *An Introduction to Statistical Learning*

For this setup I wrote a separate package, `kaobiblio`, which you can find in the `styles` directory and include in your main tex file. This package accepts all the options that you can pass to `biblatex`, and actually it passes them to `biblatex` under the hood. Moreover, it also defines some commands, like `\sidecite`, and environments that can be used within a `kao` book.[1]

1: For this reason you should always use `kaobiblio` instead of `biblatex`, but the syntax and the options are exactly the same.

As you have seen, the `\sidecite` command will print a citation in the margin. However, this command would be useless without a way to customise the format of the citation, so the `kaobook` provides also the `\formatmargincitation` command. By 'renewing' that command, you can choose which items will be printed in the margins. The best way to understand how it works is to see the actual definition of this command.

```
\newcommand{\formatmargincitation}[1]{
    \parencite{#1}: \citeauthor*{#1} (\citeyear{#1}), \citetitle{#1}\\
}
```

Thus, the `\formatmargincitation` accepts one parameter, which is the citation key, and prints the parencite followed by a colon, then the author, then the year (in brackets), and finally the title.[6] Now, suppose that you wish the margin citation to display the year and the author, followed by the title, and finally a fixed arbitrary string; you would add to your document:

[6]: Battle et al. (2014), 'Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individualsssssssss'

```
\renewcommand{\formatmargincitation}[1]{
    \citeyear{#1}, \citeauthor*{#1}: \citetitle{#1}; very interesting!\\
}
```

The above code results in citations that look like the following.[7] Of course, changing the format is most useful when you also change the default bibliography style. For instance, if you want

2005, Zou et al.: 'Regularization and variable selection via the elastic-net'; very interesting!

to use the 'philosophy-modern' style for your bibliography, you might have something like this in the preamble:

```
\usepackage[style=philosophy-modern]{styles/kaobiblio}
\renewcommand{\formatmargincitation}[1]{
    \sdcite{#1}\\
}
\addbibresource{main.bib}
```

The commands like `\citeyear`, `\parencite` and `\sdcite` are just examples. A full reference of the available commands can be found in this cheatsheet, under the 'Citations' section.

Finally, to compile a document containing citations, you need to use an external tool, which for this class is biber. You need to run the following (assuming that your tex file is called main.tex):

```
$ pdflatex main
$ biber main
$ pdflatex main
```

5.2 Glossaries and Indices

The `kaobook` class loads the packages `glossaries` and `imakeidx`, with which you can add glossaries and indices to your book. For instance, I previously defined some glossary entries and now I am going to use them, like this: computer. `glossaries` also allows you to use acronyms, like the following: this is the full version, Frame per Second (FPS), and this is the short one FPS. These entries will appear in the glossary in the backmatter.

Unless you use Overleaf or some other fancy IDE for LaTeX, you need to run an external command from your terminal in order to compile a document with a glossary. In particular, the commands required are:[2]

2: These are the commands you would run in a UNIX system; I have no idea on how it works in Windows.

```
$ pdflatex main
$ makeglossaries main
$ pdflatex main
```

Note that you need not run `makeglossaries` every time you compile your document, but only when you change the glossary entries.

To create an index, you need to insert the command `\index{subject}` whenever you are talking about 'subject' in the text. For instance, at the start of this paragraph I would write `index{index}`, and an entry would be added to the Index in the backmatter. Check it out!

In theory, you would need to run an external command for the index as well, but luckily the package we suggested, `imakeidx`, can compile the index automatically.

A nomenclature is just a special kind of index; you can find one at the end of this book. To insert a nomenclature, we use the package `nomencl` and add the terms with the command `\nomenclature`. We put then a `\printnomenclature` where we want it to appear.

Also with this package we need to run an external command to compile the document, otherwise the nomenclature will not appear:

```
$ pdflatex main
$ makeindex main.nlo -s nomencl.ist -o main.nls
$ pdflatex main
```

These packages are all loaded in packages.sty, one of the files that come with this class. However, the configuration of the elements is best done in the main.tex file, since each book will have different entries and styles.

Note that the `nomencl` package caused problems when the document was compiled, so, to make a long story short, I had to prevent `scrhack` to load the hack-file for `nomencl`. When compiling the document on Overleaf, however, this problem seem to vanish.

This brief section was by no means a complete reference on the subject, therefore you should consult the documentation of the above package to gain a full understanding of how they work.

5.3 Hyperreferences

In this class we provide a handy sub-package to help you referencing the same elements always in the same way, for consistency across the book. First, you can label each element with a specific command. For instance, should you want to label a chapter, you would put `\labch{chapter-title}` right after the `\chapter` directive. This is just a convienence, because `\labch` is actually just an alias to `\label{ch:chapter-title}`, so it spares you the writing of 'ch'. We defined similar commands for many typically labeled elements, including:

- Page: `\labpage`
- Part: `\labpart`
- Chapter: `\labch`
- Section: `\labsec`
- Figure: `\labfig`
- Table: `\labtab`
- Definition: `\labdef`
- Theorem: `\labthm`
- Proposition: `\labprop`
- Lemma: `\lablemma`
- Remark: `\labremark`
- Example: `\labexample`
- Exercise: `\labexercise`

Of course, we have similar commands for referencing those elements. However, since the style of the reference should depend on the context, we provide different commands to reference the same thing. For instance, in some occasions you may want to reference the chapter by name, but other times you want to reference it only by number. In general, there are four reference style, which we call plain, vario, name, and full.

The plain style references only by number. It is accessed, for chapters, with `\refch{chapter-title}` (for other elements, the syntax is analogous). Such a reference results in: Chapter 5.

The vario and name styles rest upon the `varioref` package. Their syntax is `\vrefch{chapter-title}` and `\nrefch{chapter-title}`, and they result in: Chapter 5 on page 19, for the vario style, and: Chapter 5 (References), for the name style. As you can see, the page is referenced in `varioref` style.

The full style references everything. You can use it with `\frefch {chapter-title}` and it looks like this: Chapter 5 (References) on page 19.

Of course, all the other elements have similar commands (*e.g.* for parts you would use `\vrefpart{part-title}` or something like that). However, not all elements implement all the four styles. The commands provided should be enough, but if you want to see what is available or to add the missing ones, have a look at the attached package.

Design and Additional Features

6 Page Design

6.1 Headings

So far, in this document I used two different styles for the chapter headings: one has the chapter name, a rule and, in the margin, the chapter number; the other has an image at the top of the page, and the chapter title is printed in a box (like for this chapter). There is one additional style, which I used only in the appendix (on page 35); there, the chapter title is enclosed in two horizontal rules, and the chapter number (or letter, in the case of the appendix) is above it.[1]

1: To be honest, I do not think that mixing heading styles like this is a wise choice, but in this document I did it only to show you how they look.

Every book is unique, so it makes sense to have different styles from which to choose. Actually, it would be awesome if whenever a kao-user designs a new heading style, he or she added it to the three styles already present, so that it will be available for new users and new books.

The choice of the style is made simple by the \setchapterstyle command. It accepts one option, the name of the style, which can be: 'plain', 'kao', or 'lines'.[2] If instead you want the image style, you have to use the command \setchapterimage, which accepts the path to the image as argument; you can also provide an optional parameter in square brackets to specify the height of the image.

2: Plain is the default LaTeX title style; the other ones are self explanatory.

Let us make some examples. In this book, I begin a normal chapter with the lines:

```
1 \setchapterstyle{kao}
2 \setchapterpreamble[u]{\margintoc}
3 \chapter{Title of the Chapter}
4 \labch{title}
```

In Line 1 I choose the style for the title to be 'kao'. Then, I specify that I want the margin toc. The rest is ordinary administration in LaTeX, except that I use my own `\labch` to label the chapter. Actually, the `\setchapterpreamble` is a standard KOMA-Script one, so I invide you to read about it in the KOMA documentation. Once the chapter style is set, it holds until you change it.[3] Whenever I want to start a chapter with an image, I simply write:

3: The `\margintoc` has to be specified at every chapter. Perhaps in the future this may change; it all depends on how this feature will be welcomed by the users, so keep in touch with me if you have preferences!

```
\setchapterimage[7cm]{path/to/image.png} % Optionally
    specify the height
\setchapterpreamble[u]{\margintoc}
\chapter{Catchy Title} % No need to set a chapter style
\labch{catchy}
```

If you prefer, you can also specify the style at the beginning of the main document, and that style will hold until you change it again.

6.2 Headers & Footers

Headers and footers in KOMA-Script are handled by the `scrlayer-scrpage` package. There are two basic style: 'scrheadings' and 'plain.scrheadings'. The former is used for normal pages, whereas the latter is used in title pages (those where a new chapter starts, for instance) and, at least in this book, in the front matter. At any rate, the style can be changed with the `\pagestyle` command, *e.g.* `\pagestyle{plain .scrheadings}`.

In both stles, the footer is completely empty. In plain.scrheadings, also the header is absent (otherwise it wouldn't be so plain. . .), but in the normal style the design is reminescent of the 'kao' style for chapter titles.

To Do

The `twoside` class option is still unstable and may lead to unexpected behaviours. As always, any help will be greatly appreciated.

6.3 Table of Contents

Another important part of a book is the table of contents. By default, in `kaobook` there is an entry for everything: list of figures, list of tables, bibliographies, and even the table of contents itself. Not everybody might like this, so we will provide a description of the changes you need to do in order to enable or disable each of these entries. In the following Table 6.1, each item corresponds to a possible entry in the TOC, and its description is the command you need to provide to have such entry. These commands are specified

Entry	Command to Activate
Table of Contents	`\setuptoc{toc}{totoc}`
List of Figs and Tabs	`\PassOptionsToClass{toc=listof}{\@baseclass}`
Bibliography	`\PassOptionsToClass{toc=bibliography}{\@baseclass}`

Table 6.1: Commands to add a particular entry to the table of contents.

in the attached style package,[4] so if you don't want the entries, just comment the corresponding lines.

4: In the same file, you can also choose the titles of these entries.

Of course, some packages, like those for glossaries and indices, will try to add their own entries. In such cases, you have to follow the instructions specific to that package. Here, since we have talked about glossaries and notations in Chapter 5, we will biefly see how to configure them.

In a later section, we will see how you can define your own floating environment, and endow it with an entry in the TOC.

For the `glossaries` package, use the 'toc' option when you load it: `\usepackage[toc]{glossaries}`. For `nomencl`, pass the 'intoc' option at the moment of loading the package. Both `glossaries` and `nomencl` are loaded in the attached 'packages' package.

Additional configuration of the table of contents can be performed through the packages `etoc`, which is loaded because it is needed for the margintocs, or the more traditional `tocbase`. Read the respective documentations if you want to be able to change the default TOC style.[5]

5: (And please, send me a copy of what you have done, I'm so curious!)

6.4 Page Layout

Besides the page style, you can also change the width of the content of a page. This is particularly useful for pages dedicated to part titles, where having the 1.5-column layout might be a little awkward, or for pages where you only put figures, where it is important to exploit all the available space.

In practice, there are two layouts: 'wide' and 'margin'. The former suppresses the margins and allocates the full page for contents, while the latter is the layout used in most of the pages of this book, including this one. The wide layout is also used automatically in the front and back matters.

To change page layout, use the `\pagelayout` command. For example, when I start a new part, I write:

```
\pagelayout{wide}
\addpart{Title of the New Part}
\pagelayout{margin}
```

6.5 Numbers & Counters

In this short section we shall see how dispositions, sidenotes and figures are numbered in the `kaobook` class.

By default, dispositions are numbered up to the section. This is achieved by setting: `\setcounter{secnumdepth}{1}`.

The sidenotes counter is the same across all the document, but if you want it to reset at each chapter, just uncomment the line

```
\counterwithin*{sidenote}{chapter}
```

in the `styles/style.sty` package provided by this class.

Figure and Table numbering is also per-chapter; to change that, use something like:

```
\renewcommand{\thefigure}{\arabic{section}.\arabic{figure}}
```

6.6 White Space

One of the things that I find most hard in LaTeX is to finely tune the white space around objects. There are not fixed rules, each object needs its own adjustment. Here we shall see how some spaces are defined at the moment in this class.

Attention! This section may be incomplete.

Space around figures and tables

```
\renewcommand\FBaskip{.4\topskip}
\renewcommand\FBbskip{\FBaskip}
```

Space around captions

```
\captionsetup{
    aboveskip=6pt,
    belowskip=6pt
}
```

Space around displays (*e.g.* equations)

```
\setlength\abovedisplayskip{6pt plus 2pt minus 4pt}
\setlength\belowdisplayskip{6pt plus 2pt minus 4pt}
\abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
```

Mathematics and Boxes | 7

7.1 Theorems

Despite most people complain at the sight of a book full of equations, mathematics is an important part of many books. Here, we shall illustrate some of the possibilities. We believe that theorems, definitions, remarks and examples should be emphasised with a shaded background; however, the colour should not be to heavy on the eyes, so we have chosen a sort of light yellow.[1]

1: The boxes are all of the same colour here, because we did not want our document to look like Harlequin.

Definition 7.1.1 *Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X, d).*

Definition 7.1.1 is very important. I am not joking, but I have inserted this phrase only to show how to reference definitions. The following statement is repeated over and over in different environments.

Theorem 7.1.1 *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

Proposition 7.1.2 *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

You can even insert footnotes inside the theorem environments; they will be displayed at the bottom of the box.

Lemma 7.1.3 *A finite intersection[a] of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

[a] I'm a footnote

You can safely ignore the content of the theorems. . . I assume that if you are interested in having theorems in your book, you already know something about the classical way to add them. These example should just showcase all the things you can do within this class.

Corollary 7.1.4 (Finite Intersection, Countable Union) *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

Proof. The proof is left to the reader as a trivial exercise. Hint: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. □

Definition 7.1.2 *Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X, d).*

Here is a random equation, just because we can:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$

Example 7.1.1 Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_{d} of all the open subsets of (X, d).

Remark 7.1.1 Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_{d} of all the open subsets of (X, d).

As you may have noticed, definitions, example and remarks have independent counters; theorems, propositions, lemmas and corollaries share the same counter.

Remark 7.1.2 Here is how an integral looks like inline: $\int_a^b x^2 dx$, and here is the same integral displayed in its own paragraph:

$$\int_a^b x^2 dx$$

We provide two files for the theorem styles: plaintheorems.sty, which you should include if you do not want coloured boxes around theorems; and mdftheorems.sty, which is the one used for this document.[2] Of course, you will have to edit these files according to your taste and the general style of the book.

2: The plain one is not showed, but actually it is exactly the same as this one, only without the yellow boxes.

7.2 Boxes & Custom Environments [3]

Say you want to insert a special section, an optional content or just something you want to emphasise. We think that nothing works better than a box in these cases. We used `mdframed` to construct the ones shown below. You can create and modify such environments by editing the provided file environments.sty.

3: Notice that in the table of contents and in the header, the name of this section is 'Boxes & Environments'; we achieved this with the optional argument of the `section` command.

Title of the box

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

If you set up a counter, you can even create your own numbered environment.

Comment 7.2.1

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.3 Experiments

It is possible to wrap marginnotes inside boxes, too. Audacious readers are encouraged to try their own experiments and let me

title of margin note

Margin note inside a kaobox. (Actually, kaobox inside a marginnote!)

know the outcomes.

I believe that many other special things are possible with the `kaobook` class. During its development, I struggled to keep it as flexible as possible, so that new features could be added without too great an effort. Therefore, I hope that you can find the optimal way to express yourselves in writing a book, report or thesis with this class, and I am eager to see the outcomes of any experiment that you may try.

Appendix

A
Heading on Level 0 (chapter)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1 Heading on Level 1 (section)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 2 (subsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 3 (subsubsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 4 (paragraph) Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.2 Lists

Example for list (itemize)

- First item in a list
- Second item in a list
- Third item in a list
- Fourth item in a list
- Fifth item in a list

Example for list (4*itemize)

- First item in a list
  - First item in a list
    - First item in a list
      - First item in a list
      - Second item in a list
    - Second item in a list
  - Second item in a list
- Second item in a list

Example for list (enumerate)

1. First item in a list
2. Second item in a list
3. Third item in a list
4. Fourth item in a list
5. Fifth item in a list

Example for list (4*enumerate)

1. First item in a list
    a) First item in a list
        i. First item in a list
            A. First item in a list
            B. Second item in a list
        ii. Second item in a list
    b) Second item in a list
2. Second item in a list

Example for list (description)

First item in a list
Second item in a list
Third item in a list
Fourth item in a list
Fifth item in a list

Example for list (4*description)

- **First** item in a list
    - **First** item in a list
        - **First** item in a list
            - **First** item in a list
            - **Second** item in a list
        - **Second** item in a list
    - **Second** item in a list
- **Second** item in a list

Bibliography

Here are the references in citation order.

[1] Tom Heath and Christian Bizer. 'Linked data: Evolving the web into a global data space'. In: *Synthesis lectures on the semantic web: theory and technology* 1.1 (2011), pp. 1–136 (cited on page 1).

[2] Jose Emilio Labra Gayo et al. 'Validating RDF data'. In: *Synthesis Lectures on Semantic Web: Theory and Technology* 7.1 (2017), pp. 1–328 (cited on page 1).

[3] Arthur G Ryman, Arnaud Le Hors, and Steve Speicher. 'OSLC Resource Shape: A language for defining constraints on Linked Data.' In: *LDOW* 996 (2013) (cited on page 2).

[4] Peter M Visscher, William G Hill, and Naomi R Wray. 'Heritability in the genomics era–concepts and misconceptions.' In: *Nat. Rev. Genet.* 9.4 (2008), pp. 255–266. doi: 10.1038/nrg2322 (cited on page 19).

[5] Gareth James et al. *An Introduction to Statistical Learning*. 2013 (cited on page 19).

[6] Alexis Battle et al. 'Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individualsssssssss'. In: *Genome Res.* 24.1 (2014), pp. 14–24. doi: 10.1101/gr.155192.113 (cited on page 19).

[7] Hui Zou and Trevor Hastie. 'Regularization and variable selection via the elastic-net'. In: *J. R. Stat. Soc.* 67.2 (2005), pp. 301–320. doi: 10.1111/j.1467-9868.2005.00503.x (cited on page 19).

Notation

The next list describes several symbols that will be later used within the body of the document.

c	Speed of light in a vacuum inertial frame
h	Planck constant

Greek Letters with Pronounciation

Character	Name	Character	Name
α	alpha *AL-fuh*	ν	nu *NEW*
β	beta *BAY-tuh*	ξ, Ξ	xi *KSIGH*
γ, Γ	gamma *GAM-muh*	o	omicron *OM-uh-CRON*
δ, Δ	delta *DEL-tuh*	π, Π	pi *PIE*
ϵ	epsilon *EP-suh-lon*	ρ	rho *ROW*
ζ	zeta *ZAY-tuh*	σ, Σ	sigma *SIG-muh*
η	eta *AY-tuh*	τ	tau *TOW (as in cow)*
θ, Θ	theta *THAY-tuh*	υ, Υ	upsilon *OOP-suh-LON*
ι	iota *eye-OH-tuh*	ϕ, Φ	phi *FEE, or FI (as in hi)*
κ	kappa *KAP-uh*	χ	chi *KI (as in hi)*
λ, Λ	lambda *LAM-duh*	ψ, Ψ	psi *SIGH, or PSIGH*
μ	mu *MEW*	ω, Ω	omega *oh-MAY-guh*

Capitals shown are the ones that differ from Roman capitals.

Special Terms

C

computer is a programmable machine that receives input, stores and manipulates data, and provides output in a useful format. 20

F

FPS Frame per Second. 20

T

TOC Table of Contents. 26, 27

Alphabetical Index