Version 0.1

OpTeX Markup Language Standard

Petr Olšák, 2021

The OpTeX markup language standard (OMLS) declares a list of control sequences used in OpTeX documents including their syntax and sematic. The listed control sequences in OMLS in section 5 in table 5.1 are called *known* and other control sequences are *unknown*.

The main reason for OMLS is to give instructions on how to program convertors from OpTeX documents to another formats (Html, Markdown, LaTeX) or how to interpret the OpTeX document sources in such applications as `texcount` or text editors[1]. These converters and applications are called *cnv-programs* in this document.

We suppose that if you need absolute control over the typography of the document when it is converted to PDF pages, then you use OpTeX itself. If you need to create other formats of the same document then you can use a cnv-program which accepts OMLS. The result is a document without typographical instructions like dimensions of pages, margins, paginations and headers, selection of a font-family, dimensions of the fonts, etc. You can imagine the result of such a conversion as a single Html page where more typographical features can be controlled in a different way, for example by an external CSS file. This is a reason why control sequences like `\fontfam` or `\margins` are ignored by cnv-programs.

Obviously, TeX and OpTeX itself gives possibility to declare various input formats for various purposes. Sometimes (in very special cases) there exists a good reason to declare a different and special input format by TeX macros. But if the source of the document respects the OMLS then it is reasonably transformable to other formats by cnv-programs. We hope that OMLS-ready documents cover a very large set of typical documents used these days.

We suppose that cnv-programs work internally with strings of source lines without tokenization. This is one of the great differences in processing the document directly by OpTeX and using a cnv-program. The second difference is that the expansion process of macros is not implemented in cnv-programs in its full range. We respect that the result of cnv-programs will be different than from processing directly by OpTeX. But this is not a bug, this is the feature. We concentrate on the fixed syntax and sematic given by OMLS of the OpTeX document and we throw behind the head the typographical exactness of the document which can be done only directly by TeX (and it is exactly described in TeXbook, for example).

Table of contents

[1] We suppose advanced editor features: color highlighting, sections/chapters folding, auto-completions, etc.

1 Syntactical rules

The syntactical elements are described as strings here. No TeX's tokenization is taken into account. The rule with a smaller number has precedence.

1. end of line or end of file → ⟨*eol*⟩.
2. `%%:` at beginning of the line → ⟨*cnv-declarator*⟩, see section 4.
3. ⟨*cnv-declarator*⟩⟨*text*⟩⟨*eol*⟩ is interpreted specially.
4. space or a tab-character → ⟨*space*⟩.
5. non-empty sequence of ⟨*space*⟩s → ⟨*spaces*⟩.
6. an empty line or a line only with ⟨*spaces*⟩ → ⟨*empty-line*⟩.
7. a character `a`–`z` or `A`–`Z` → ⟨*letter*⟩.
8. a ⟨*letter*⟩ or `_` → ⟨*specletter*⟩.
9. a character different from previous rule or ⟨*eol*⟩ or ⟨*spaces*⟩ → ⟨*non-specletter*⟩
10. `\`⟨*non-specletter*⟩ → ⟨*singlechar-control-sequence*⟩.
11. a non-empty sequence of ⟨*specletter*⟩s → ⟨*letters-seq*⟩.
12. `\`⟨*letters-seq*⟩⟨*non-specletter*⟩ → ⟨*multiletter-control-sequence*⟩⟨*non-specletter*⟩.
13. `%`⟨*text*⟩⟨*eol*⟩ → ⟨*comment*⟩ and it is completely ignored including ⟨*eol*⟩.
14. ⟨*spaces*⟩ at beginning of the line marks that the line as *indented*.
15. ⟨*spaces*⟩ at beginning of the line → are ignored.
16. ⟨*spaces*⟩⟨*eol*⟩ or ⟨*eol*⟩ → ⟨*spaces*⟩.
17. ⟨*multileter-control-sequence*⟩⟨*spaces*⟩ → ⟨*multileter-control-sequence*⟩.
18. ⟨*multiletter-control-sequence*⟩ or ⟨*singlechar-control-sequence*⟩ → ⟨*control-sequence*⟩.
19. ⟨*spaces*⟩ → ⟨*space*⟩.
20. `~` → non-breakable space.
21. text where all pairs `{` and `}` match at arbitrary level → ⟨*balanced-text*⟩.
22. `{`⟨*balanced-text*⟩`}` can be interpreted as a parameter of a ⟨*control-sequence*⟩, see section 5.
23. The `{` alone not used by previous rule opens a group and the `}` alone closes the same group.
24. There are two main modes: h-mode, v-mode[2]. The document processing starts in v-mode.
25. In v-mode: a ⟨*non-space*⟩ character or a control sequence listed in table 1.1 swithes to h-mode.
26. In h-mode: an ⟨*empty-line*⟩ or a control sequence listed in table 1.2 switches to v-mode.
27. The switching from v-mode to h-mode → a paragraph begins.
28. The switching from h-mode to v-mode → the current paragraph ends.
29. `$`⟨*text*⟩`$` or `$$`⟨*text*⟩`$$` → process ⟨*text*⟩ in math-mode, see section 6.
30. a non-empty sequence of digits with optional `+` or `-` in the front → ⟨*number*⟩.
31. ⟨*number*⟩ with optional dot inside the sequence of digits → ⟨*decimal-number*⟩.
32. a pair of letters listed in the table 1.3 → ⟨*tex-unit*⟩.
33. optional space, i.e. ⟨*space*⟩ or nothing → ⟨*o-space*⟩.
34. ⟨*decimal-number*⟩⟨*o-space*⟩⟨*tex-unit*⟩⟨*o-space*⟩ → ⟨*dimen*⟩.
35. ⟨*control-sequence*⟩ not listed in table 5.1 nor in configuration → ⟨*unknown-control-sequence*⟩.
36. `=`⟨*o-space*⟩ or ⟨*o-space*⟩ → ⟨*o-equal*⟩.
37. ⟨*unknown-control-sequence*⟩⟨*o-equal*⟩⟨*dimen*⟩ → should be completely ignored.
38. ⟨*unknown-control-sequence*⟩⟨*o-equal*⟩⟨*number*⟩ → should be completely ignored.
39. ⟨*unknown-control-sequence*⟩`=`⟨*o-space*⟩`{`⟨*balanced-text*⟩`}` → should be completely ignored.
40. ⟨*unknown-control-sequence*⟩`[`⟨*balanced-text*⟩`]` → should be completely ignored.
41. ⟨*unknown-control-sequence*⟩ → should be ignored.
42. ⟨*control-sequence*⟩s are processed as described in section 5 or by a configuration of the cnv-program.

Table 1.1 List of control sequences which switch from v-mode to h-mode.

`\`⟨*space*⟩, `\hfil`, `\hfill`, `\hskip`, `\hss`, `\indent`, `\leavevmode`, `\noindent`, `\quad`, `\qquad`, `\vrule`.

Table 1.2 List of control sequences which switch from h-mode to v-mode.

`\begblock`, `\begitems`, `\begmulti`, `\begtt`, `\bib`, `\bigskip`, `\bye`, `\caption`, `\chap`, `\cskip`, `\end`, `\endblock`, `\enditems`, `\endmulti` `\hrule`, `\medskip`, `\par`, `\sec`, `\secc`, `\secl`, `\smallskip`, `\vfil`, `\vskip`.

[2] this is great simplification of real TeX modes.

Table 1.3 List of TEX units.

`bp`, `cc`, `cm`, `dd`, `em`, `ex`, `in`, `mm`, `pc`, `pt`, `sp`.

Examples

- `\%` is ⟨*control-sequence*⟩ by rules 10, 18. It does not start comment, because rule 10 has precedence before rule 13.
- `\%`⟨*space*⟩: the ⟨*space*⟩ is kept, but `\foo`⟨*space*⟩: the space is removed by rule 17.
- `wordA`⟨*spaces*⟩⟨*eol*⟩⟨*spaces*⟩`wordB` is `wordA`⟨*space*⟩`wordB` by rules 15, 16, and 19.
- `\kern-3pt` should be ignored, because `\kern` is ⟨*unknown-control-sequence*⟩ and rule 37 is applied.
- `\vskip42mm` should finalize paragraph in h-mode by rule 26 and then it is ignored by rule 37 because `\vskip` is ⟨*unknown-control-sequence*⟩ not listed in table 5.1.
- `\typosize[12/16]` is ignored by rule 40.
- `\foo{text}` is `{text}` (i.e. `text` in a group) by rules 41 and 23.

2 Scanning parameters

If a control sequence listed in rules above or in the section 5 has a parameter, the parameter is scanned as a string with interpretation only those rules which are needed to determine the boundary of the parameters. For example `\tit` ⟨*title*⟩⟨*eol*⟩ applies only rule 1 during scanning the parameter, i.e. ⟨*title*⟩ is a string telemetered by the end of the line or the end of the file. Or `\fnote{`⟨*balanced-text*⟩`}` applies only rule 21.

If the parameter is in the format `{`⟨*something*⟩`}` then the ⟨*something*⟩ is always meant as ⟨*balanced-text*⟩. We don't specify the type ⟨*balanced-text*⟩ explicitly here, so we refer to `\fnote{`⟨*text*⟩`}` and not `\fnote{`⟨*balanced-text*⟩`}`, for example.

The spaces before the scanned parameter are optional and they are ignored. Spaces inside `{...}` are not ignored.

If the parameter is in the format `{`⟨*something*⟩`}` and the first non-space character is not `{` then the parameter is this first non-space character or a ⟨*control-sequence*⟩ if the first non-space character is `\`. For example `\fnote a` is equal to `\fnote{a}`. The difference from this rule is given for `\input`, `\verbinput`,`\inspic` and `\inkinspic` control sequences in section 5.

If the parameter is scanned as a string already then all syntactical rules are applied when it is used. For example:

```
\tit    This is    a title
```

the parameter is scanned as `This is    a title` and the rules 5 and 19 are applied when it is used.

3 Declaration and text parts of the document

A typical OpTEX document has two parts. A declaration part, where macros are defined by `\def` and friends, fonts and sizes are declared, etc. This part should be ignored by cnv-programs. The second part includes the document text with a markup using well-declared control sequences. The cnv-program has to interpret the second part.

So, the cnv-program starts in declaration-skipping mode and it switches to the text mode later.

When cnv-program is in declaration-skipping mode then all indented lines are ignored. And lines which begins by `}` or by a ⟨*control-sequence*⟩ not listed in table 3.1 are ignored too.

If the line begins by a character other than `}` or by a ⟨*control-sequence*⟩ listed in table 3.1 then cnv-program switches to text mode. All texts are interpreted from this line including this one.

User can say explicitly where the second part of the document starts by `%%:text` given at beginning of a line. Moreover, if `%%:decl` is given at beginning of a line, then all text between `%%:decl` and `%%:text` is ignored, only other possible ⟨*cnv-declarator*⟩s are processed here. It means that the declaration part and the text part of the document can be simply determined by the pair `%%:decl` and `%%:text`.

Table 3.1 List of control sequences which start the text mode.

`\address`, `\begblock`, `\begitems`, `\begmulti`, `\begtt`, `\bf`, `\bi`, `\bib`, `\caption`, `\cite`, `\clipinoval`, `\clipincircle`, `\ecite`, `\fnote`, `\frame`, `\hfil`, `\hfill`, `\ii`, `\iid`, `\incircle`, `\inkinspic`, `\inoval`, `\inspic`, `\it`, `\LaTeX`, `\LuaTeX`, `\maketoc`, `\mnote`, `\OpTeX`, `\putpic`, `\puttext`, `\rcite`, `\rm`, `\rotbox`, `\sec`, `\secc`, `\secl`, `\table`, `\TeX`, `\tit`, `\usebib`, `\verbinput`.

4 The `%%:` declarators

The `%%:` declarators are ignored when the document is processed by TeX but they can give instructions to cnv-programs. The `%%:` must be placed at beginning of the line. The list of `%%:` declarators follows:

- `%%:decl` – the following text is ignored until `%%:text` occurs. Only other `%%:` declarators are interpreted.
- `%%:text` – the following text must be interpreted in text mode.
- `%%:to` ⟨*format*⟩ ⟨*config-file*⟩ – if the cnv-program converts to the ⟨*format*⟩ then it has to use the ⟨*config-file*⟩. For example:

  ```
  %%:to html html-mydocument.cfg
  %%:to markdown markdown-mydocument.cfg
  %%:to latex preamble-mydocument.cfg
  ```

  The language of the config files are not a part of this standard, we suppose something to be natural for used cnv-program. The config file should give additional rules for interpreting control sequences not listed in the section 5 (see rule 35). For example a tool for defining a behavior of unknown control sequences can be here. These definitions can depend on the converted document and on the output format and they can be given in the configuration files.
- `%%:app` ⟨*application*⟩ ⟨*config-file*⟩ – behaves like `%%;to` but the cnv-program name instead output format is given here.
- `%%:do` ⟨*format-or-app*⟩ ⟨*action*⟩ – does an ⟨*action*⟩ if ⟨*format-or-app*⟩ is output format or used cnv-program name. The ⟨*action*⟩ syntax depends on used cnv-program and it typicaly does a change in its configuration or give a command to it.
- `%%:skip` ⟨*formats-or-apps*⟩ – ignores all following lines until another `%%:` occurs if the output format or application name is included in the ⟨*formats-or-apps*⟩ space-separated list. Example:

  ```
  %%:skip html markdown
  This text is not interpreted when Html or Markdown output is generated.
  %%:
  ```

  If the ⟨*formats-or-apps*⟩ is empty then the `%%:skip` is applied for each cnv-program and for each output.
- `%%:if` ⟨*formats-or-apps*⟩ – processes following lines until another `%%:` only if the output format or application name is included in the ⟨*formats-or-apps*⟩ space-separated list. Other cnv-programs or output formats not mentioned here skip these lines. Note that TeX processes such lines always. But you can use `\ignoreit{`⟨*text*⟩`}` which is processed as `{`⟨*text*⟩`}` by cnv-programs (see rule 41) but it is ignored by OpTeX. Example:

  ```
  %%:if html latex
  \ignoreit{\input{file.tex}}
  %%:
  ```

  The `file.tex` in this example is processed only if LaTeX or Html output is generated.
- `%%:use` – the next single line is fully interpreted unless cnv-program ignores declarations by `%%:decl` or because it is in declaration-skipping mode. Example:

  ```
  %%:use
  \verbchar` \picdir={img/}
  ```

  The example shows how cnv-program is able to read `\verbchar` or `\picdir` settings in the declaration-skipping mode although these control sequences are not listed in table 3.1.
- `%%:quotes` ⟨*qql*⟩ ⟨*qqr*⟩ ⟨*ql*⟩ ⟨*qr*⟩ – declares ⟨*qql*⟩ and ⟨*qqr*⟩ (left and right double quotation marks), ⟨*ql*⟩ and ⟨*qr*⟩ (left and right single quotation mark). They are used when `\"` or `\'` control sequences are processed. Default: unset, i.e. `\"` and `\'` are interpreted as unknown control sequences.

5 List of known control sequences

The phrase "should be" is used very often when the interpretation of control sequences is declared here. It means that this is only a common interpretation, but differences are possible due to the type of the output format and used cnv-program. For example, when we convert to LaTeX then `\-` and `\/` are not ignored but they are re-written without change to the output of the LaTeX source file.

Table 5.1 List of known control sequences alphabeticaly sorted.

\# 5.1 \$ 5.1 \% 5.1 \" 5.1 \' 5.1 \/ 5.1 \- 5.1 \& 5.1 \begblock 5.6 \begitems 5.7 \begmulti 5.9 \begtt 5.8 \bf 5.4 \bi 5.4 \bib 5.10 \Black 5.5 \Blue 5.5 \Brown 5.5 \bslash 5.1 \caption 5.11 \chap 5.3 \cite 5.10 \code 5.8 \cr 5.11 \crl 5.11 \crli 5.11 \crll 5.11 \crlli 5.11 \crlp 5.11 \Cyan 5.5 \def 5.15 \ecite 5.10 \edef 5.15 \em 5.4 \endblock 5.6 \endinput 5.2 \enditems 5.7 \endmulti 5.9 \fC 5.11 \fL 5.11 \fnote 5.12 \fnotemark 5.12 \fnotetext 5.12 \fR 5.11 \fS 5.11 \fX 5.11 \gdef 5.15 \Green 5.5 \ii 5.13 \iid 5.13 \inkinspic 5.2 \input 5.2 \insertoutline 5.15 \inspic 5.2 \it 5.4 \label 5.10 \LaTeX 5.14 \LuaTeX 5.14 \Magenta 5.5 \maketoc 5.10 \mnote 5.12 \mspan 5.11 \noalign 5.11 \notoc 5.10 \OpTeX 5.14 \outlines 5.15 \pgref 5.10 \picdir 5.2 \qquad 5.1 \quad 5.1 \rcite 5.10 \Red 5.5 \ref 5.10 \rm 5.4 \sec 5.3 \secc 5.3 \secl 5.3 \space 5.1 \style 5.7 \table 5.11 \TeX 5.14 \thisoutline 5.15 \tit 5.3 \tskip 5.11 \tt 5.4 \ulink 5.10 \url 5.10 \usebib 5.2 \verbchar 5.8 \verinput 5.2 \vspan 5.11 \White 5.5 \xdef 5.15 \Yellow 5.5

5.1 Character-like control sequences

- \%, \$, \&,\# respectively → should be normal characters %, $, &, # respectively.
- \bslash → normal character \.
- \space, \⟨*space*⟩, \⟨*eol*⟩ → space.
- \, → should be small space or space.
- \quad, \qquad → should be bigger space or more spaces.
- \-, \/ → should be ignored.
- \"⟨*text*⟩" or \'⟨*text*⟩' → ⟨*qql*⟩⟨*text*⟩⟨*qqr*⟩ or ⟨*ql*⟩⟨*text*⟩⟨*qr*⟩, only if %%:quotes are set.

5.2 Input files

- \input {⟨*file-name*⟩} or \input ⟨*file-name*⟩⟨*space*⟩ should redirect the input to given file. At the end of the input-ed file or at \endinput the reading of the current file continues. The file is read from the current directory, but the ⟨*file-name*⟩ should include the full path to the file or relative path starting from the current directory. First, the file ⟨*file-name*⟩.tex is tried to read and if it doesn't exist then the file ⟨*file-name*⟩ is read.
- \picdir ⟨*o-equal*⟩{⟨*text*⟩} saves ⟨*text*⟩ to ⟨*picdir-value*⟩. By default, ⟨*picdir-value*⟩ is empty.
- \inspic {⟨*file-name*⟩} or \inspic ⟨*file-name*⟩⟨*space*⟩ should include the given picture from the ⟨*picdir-value*⟩⟨*file-name*⟩.
- \inkinspic {⟨*file-name*⟩} or \inkinspic ⟨*file-name*⟩⟨*space*⟩ behaves like \inspic.
- \verinput ⟨*ignore*⟩ (⟨*lines*⟩) ⟨*file-name*⟩⟨*space*⟩ should include the ⟨*file-name*⟩ (only given lines) as a verbatim text, i. e. without any syntactical interpretation.
- \usebib/⟨*letter*⟩ (⟨*style*⟩) ⟨*file-names*⟩ should use files from ⟨*file-names*⟩ to generate the list of bib references. The ⟨*file-names*⟩ is comma separated list (the .bib extension is appended to each such file name). Only cited bib records should be used in this bib-references, i.e. their label must be used in a \cite[⟨*labels*⟩] or \rcite[⟨*labels*⟩] or \ecite[⟨*label*⟩]. What bib fields are used in bib records depends on the cnv-program and on its configuration. Maybe, simple cnv-programs should generate nothing here.

5.3 Titles

- \tit ⟨*title*⟩⟨*eol*⟩ should be a title of the document.
- \chap ⟨*title*⟩⟨*eol*⟩ or \chap [⟨*label*⟩] ⟨*title*⟩⟨*eof*⟩ is the title of first level.
- \sec ⟨*title*⟩⟨*eol*⟩ or \sec [⟨*label*⟩] ⟨*title*⟩⟨*eof*⟩ is the title of second level.
- \secc ⟨*title*⟩⟨*eol*⟩ or \secc [⟨*label*⟩] ⟨*title*⟩⟨*eof*⟩ is the title of third level.
- \secl⟨*level*⟩ ⟨*title*⟩⟨*eol*⟩ is the title of given level.

5.4 Fonts

\rm selects upright normal font (it is selected by default), \it selects italic, \bf selects upright bold font, \bi selects bold italic and \tt selects a monospaced font, \em select italic (if upright is current) or upright (if italic is current). The font sizes or other font features are typically ignored by cnv-programs. The actual font selection is closed at the end of the current group. The groups are given:

- explicitly by { and } characters (which are not delimiters of parameters of known control sequences),
- implicitly: parameters of control sequences listed in table 5.2 are processed in a group and blocks of text enclosed by \begitems...\enditems, \begblock...\endblock, \begmulti...\endmulti are processed in a group. Each item in \table is in a group.

Table 5.2 Parameters of following control sequences are processed in a group.

`\caption`, `\chap`, `\fnote`, `\mnote`, `\sec`, `\secc`, `\tit`.

5.5 Colors

`\Red`, `\Green`, `\Blue`, `\Cyan`, `\Magenta`, `\Yellow`, `\White`, `\Black`, and `\Brown` should select the given color of the font. The selection is closed at the end of the current group (like font selectors).

5.6 Blockquotes

The blockquote[3] is opened by `\begblock` and closed by `\endblock`. Blockquotes can contain multiple paragraphs and can contain nested blockquotes.

5.7 Lists

The list is opened by `\begitems` and closed by `\enditems`. The `*` starts a new item in this environment. Nested lists are allowed.

The type of items (ordered/unordered) is given by `\style` ⟨*character*⟩, see section 1.4.5 in the OpTEX manual. Default type is unordered (bullets are used).

5.8 Code blocks (verbatim texts)

Code blocks are inline verbatim or display verbatim.

- `\begtt` ⟨*ignored*⟩⟨*eol*⟩⟨*text*⟩`\endtt`⟨*ignored*⟩⟨*eol*⟩ processes ⟨*text*⟩ in "display verbatim mode", i. e. there are no special characters, each character is processed without special interpretation, the ⟨*eol*⟩s are end of lines. The text at the same line after `\begtt` and after `\endtt` (if any) is ignored.
- `\verbchar` ⟨*character*⟩ declares ⟨*verbchar*⟩. By default, it is undeclared.
  New `\verbchar` ⟨*character*⟩ rewrites previous setting. The setting is local in the group.
- Inline verbatim is enclosed in the pairs of ⟨*verbchar*⟩s. The text between two ⟨*verbchar*⟩s is processed without special interpretation. Only possibly ⟨*eol*⟩s are replaced by space. Example:

  ```
  %%:use
  \verbchar`
  Now, `$this %text   ~\` is processed as inline verbatim.
  ```

  gives: Now, `$this %text   ~\` is processed as inline verbatim.
- `\code{`⟨*text*⟩`}` processes ⟨*text*⟩ like inline verbatim, but all `\`⟨*character*⟩ are processed as ⟨*character*⟩, specially `\{` and `\}` are `{` and `}` but without taking them into ⟨*balanced-text*⟩ rule, `\\` is `\` but do not create a ⟨*control-sequence*⟩, etc.

5.9 Multicolumns

`\begmulti` ⟨*number*⟩⟨*space*⟩ opens the group and `\endmulti` closes the group. If output format allows multi-columns then the text enclosed between `\begmulti` ⟨*number*⟩⟨*space*⟩ and matching `\endmulti` should be printed in ⟨*number*⟩ balanced columns.

5.10 Links

- `\url{`⟨*text*⟩`}` creates ⟨*text*⟩ as an external link which points to ⟨*text*⟩. The `\`⟨*character*⟩ is interpreted as ⟨*character*⟩ in ⟨*text*⟩ with one exception: `\|` is ignored.
- `\ulink[`⟨*url*⟩`]{`⟨*text*⟩`}` creates ⟨*text*⟩ as an external link which points to ⟨*url*⟩. The `\`⟨*character*⟩ is interpreted as ⟨*character*⟩ only in ⟨*url*⟩ parameter.
- `\label[`⟨*label*⟩`]` sets the ⟨*label*⟩. First following occurence of `\chap`, `\sec`, `\secc`, `\caption` or `\eqmark` sets this ⟨*label*⟩ as bounded to its position in the document.
- `\ref[`⟨*label*⟩`]` should create an internal link to the position given by `\chap`, `\sec`, `\secc`, `\caption` or `\eqmark`, if `[`⟨*label*⟩`]` is used as the parameter of this control sequence else if ⟨*label*⟩ is bounded here by previous `\label[`⟨*label*⟩`]` The visual aspect of the link is not declared by OMLS because we don't suppose that the chapters, sections, equations, etc. are automatically numbered by exactly the

[3] The terminology is borrowed from Markdown.

same way as in OpTEX. A recommendation should be: create a simple sequence of numbers over all internal links.

- `\pgref[`⟨*label*⟩`]` should be replaced by `??` if cnv-program generates single-page output (like Html, Markdown). Unfortunately, we get the irrelevant phrases in the output: "`The problem is shown at the page ??`". Users can declare `%%:skip` for such cases.
- `\bib[`⟨*label*⟩`]` or `\bib[`⟨*label*⟩`]`⟨*o-space*⟩`=`⟨*o-space*⟩`{`⟨*ignored*⟩`}` should open the new bib record. More exactly, it closes previous paragraph (if h-mode is current) and opens new h-mode. Then prints an auto-generated reference number in `[...]`. Following text is interpreted as a bib record until the h-mode ends.
- `\cite[`⟨*labels*⟩`]` should create internal links to corresponding bib records generated by `\usebib` or by `\bib`. The ⟨*labels*⟩ is a comma-separated list of labels used in `.bib` files or in `\bib` commands to indicate the corresponding bib record. The labels should be replaced by auto-generated reference numbers used in bib records. All reference numbers created by single `\cite` should be enclosed by single `[...]`. These numbers are internal links. If the cnv-program does not implement this complicated bib machinery then `\cite[`⟨*labels*⟩`]` should print only `[`⟨*labels*⟩`]`.
- `\rcite[`⟨*labels*⟩`]` does the same as `\cite[`⟨*labels*⟩`]` but doesn't print outer `[...]`.
- `\ecite[`⟨*label*⟩`]{`⟨*text*⟩`}` creates the link to corresponding bib record, ⟨*text*⟩ is is hyperlinked (no the auto-generated reference number).
- `\maketoc` should create a list of titles from all `\chap`, `\sec` and `\secc` used in the document if they are not preceded by `\notoc`. All lines in this list should include internal links to the position where the corresponding title is used.

5.11 Tables

- `\caption/`⟨*letter*⟩ opens a caption. More exactly, if current mode is h-mode, then switch to v-mode first (i.e. close the current paragraph). Then switch from v-mode to h-mode and put a ⟨*caption-head*⟩. The following text should be interpreted as the caption text until h-mode ends. If ⟨*letter*⟩ is `t` then ⟨*caption-head*⟩ is `Table` followed by an auto-generated number. If the ⟨*letter*⟩ is `f` then ⟨*caption-head*⟩ is `Figure` followed by an auto-generated number. The configuration of cnv-programs shoud allow different words than default `Table` or `Figure` (for example, if another language is used).
- `\table`⟨*ignored*⟩`{`⟨*declaration*⟩`}{`⟨*data*⟩`}` should create a table. The fidelity rate of created output depends heavily on the used cnv-program and the output format. We don't suppose that all aspects of OpTEX tables are implemented. The following features are listed in their priority of implementation.
  - Items in ⟨*data*⟩ are separated by `&`. The last item in each row is separated by end-row separator: `\cr`, `\crl`, `\crll`, `\crli`, `\crlli`, or `\crlp{`⟨*list*⟩`}`. If there are n columns in the table then we have $n-1$ `&` separators and one end-row separator for each table row. There can be an optional end-row separator at the first item in ⟨*data*⟩ and it should be ignored. The ⟨*data*⟩ can end without the end-row separator, it should be added here.
  - Spaces around `&` and before end-row separator are ignored.
  - `\noalign{`⟨*text*⟩`}` and `\tskip` ⟨*dimen*⟩ should be ignored.
  - Colum declarators in ⟨*declaration*⟩ should be interpreted: `l` (left aligned), `c` (center aligned), `r` (right aligned) or `p{`⟨*ignored*⟩`}` (paragraph-like item).
  - The ⟨*number*⟩⟨*letter*⟩ or ⟨*number*⟩`{`⟨*text*⟩`}` should be interpreted as ⟨*number*⟩-times repeated ⟨*letter*⟩ or ⟨*text*⟩ in ⟨*declaration*⟩.
  - `\vspan`⟨*decimal-number*⟩`{`⟨*text*⟩`}` should be an item with only ⟨*text*⟩.
  - `\mspan`⟨*number*⟩`[`⟨*decl*⟩`]{`⟨*text*⟩`}` should create ⟨*text*⟩ as an item which spans over ⟨*number*⟩ columns. It is used instead ⟨*number*⟩ real items, i.e. ⟨*number*⟩−1 separators `&` aren't used here.
  - Rules in the table should be interpreted, i.e. `|` in ⟨*declaration*⟩ should be not ignored and various end-row separators should be distinguished (see OpTEX manual, section 1.4.6).
  - The aligning of `p{`⟨*p-data*⟩`}` columns should be interpreded by reading `\fL`, `\fR`, `\fC`, `\fS` and `\fX` control sequences in ⟨*p-data*⟩. See OpTEX manual, section 1.4.6.

5.12 Footnotes, marginal notes

- `\fnote{`⟨*text*⟩`}` should create a footnote link as an auto-generated reference number. The ⟨*text*⟩ should be appended at the end of the document labeled by this reference number. Another alternative (for Html output, for example): the mouse over the reference number riases an auto-popup ⟨*text*⟩.
- `\fnotemark`⟨*number*⟩ should create a pointer to a next declared footnote as auto-generated reference number. The ⟨*text*⟩ of the footnote is declared after that by `\fnotetext{`⟨*text*⟩`}`. Exact behavior:

if the reference number of the last processed `\fnote` is n then `\fnotemark`⟨*number*⟩ uses the value $n+$ ⟨*number*⟩ as its reference number. A set of `\fnotemark`s is followed by an equally large set of `\fnotetext{`⟨*text*⟩`}`. First one have reference number $n+1$, second $n+2$, etc. When whole set of `\fnotetext{`⟨*text*⟩`}` is processed then n is reset to the last reference number used here.
- `\mnote` ⟨*ignored*⟩`{`⟨*text*⟩`}` should be created as auto-popup ⟨*text*⟩ or the ⟨*text*⟩ is inserted at the margin.

5.13 Tagging for index

We don't suppose that the cnv-program can generate the alphabetically sorted index. So, the markup for creating the index should be ignored:

- `\ii` ⟨*word*⟩⟨*space*⟩ (where ⟨*word*⟩ is a sequence of non-space characters) should be ignored.
- `\iid` ⟨*word*⟩⟨*space*⟩ is ⟨*word*⟩⟨*space*⟩ by default, but if `,` or `.` follows after ⟨*space*⟩ then the ⟨*space*⟩ is removed.

5.14 Logos

`\TeX`, `\LuaTeX`, `\OpTeX`, `\LaTeX`, respectively → `TeX`, `LuaTeX`, `OpTeX`, `LaTeX`, respectively. If `/` follows these control sequences then it is ignored. Other logos should be declared in the configuration of the cnv-program.

5.15 What should be ignored

- `\def`⟨*text*⟩`{`⟨*text*⟩`}` and other variants with `\gdef`, `\edef`, `\xdef`. Reason: user can insert a special definition in the `%%:text` part of the document in order to solve a problem. But cnv-program should ignore it.
- `\outlines{`⟨*text*⟩`}`, `\insertoutline{`⟨*text*⟩`}` and `\thisoutline{`⟨*text*⟩`}`.

6 Math mode processing

There are two math modes, inline: `$`⟨*formula*⟩`$` or display: `$$`⟨*formula*⟩`$$`. We suppose that ⟨*formula*⟩ will be processed by MathJax or similar software. So, the ⟨*formula*⟩ should be kept without changes in cnv-program output, but there are exceptions:

- `$`⟨*sign*⟩⟨*decimal-number*⟩`$` should be transformed to non-math text ⟨*sign*⟩⟨*decimal-number*⟩. The ⟨*sign*⟩ is `+` or `-` or nothing. The ⟨*decimal-number*⟩ can include `,` (comma) instead `.` (period). If the ⟨*sign*⟩ is `-` (character U+002D) then it must be converted to `−` (character U+2212, math minus).
- `{\bbchar` ⟨*text*⟩`}` should be converted to `\mathbb{`⟨*text*⟩`}`, because MathJax doesn't understand `\bbchar`. `$\bbchar` ⟨*text*⟩`$` should be converted to `$\mathbb{`⟨*text*⟩`}$`.
- `{\frak` ⟨*text*⟩`}` or `$\frak` ⟨*text*⟩`$` → `\mathfrak{`⟨*text*⟩`}` or `$\mathfrak{`⟨*text*⟩`}$`.
- `{\script` ⟨*text*⟩`}` or `$\script` ⟨*text*⟩`$` → `\mathscr{`⟨*text*⟩`}` or `$\mathscr{`⟨*text*⟩`}$`.
- `\eqmark`, `\eqmark[`⟨*label*⟩`]` → `\leqno (`⟨*num*⟩`)`, where ⟨*num*⟩ is auto-generated number. Or it should be completely ignored.
- It is strongly recommended to allow users to declare another replacement rules over ⟨*formula*⟩ in configuration files of cnv-programs.

7 Notes on various conversions

The cnv-programs behave differently because output formats have different concepts and are intended for different purposes. The following sections mention the main differences for typical conversions.

7.1 To Html

The Html document is one single page. If we want to split the document to more pages, then we can use `%%:do html` ⟨*action*⟩ in positions where the document should be split (the syntax of ⟨*action*⟩ depends on the used cnv-program.)

The design of Html pages should be declared by CSS files.

Pictures inserted by `\inspic` should be inserted into the Html page as `<img src="...">`, so the picture files must be installed in the webserver to render the Html page correctly in a web browser.

The `\maketoc` used in the OpTEX document means that the Table of contents should be here. It should be generated by the cnv-program, two-pass processing is needed because the cnv-program needs to read all titles of `\chap`, `\sec` and `\secc`. Another approach is to read the `.ref` file produced by OpTEX.

7.2 To Markdown

Markdown source file is "one-page document" too. The principles a re similar to conversion to Html.

7.3 From Markdown

The cnv-program which does such a reverse conversion should be very useful for people they prefer even clearer markup than in OpTeX.

7.4 To LaTeX

LaTeX gives the same main feature as OpTeX: it can generate PDF output from a source file. But different macros, a different concept, and a different markup are used.

The main reason for such conversion is that journal editors require LaTeX source file, but user want to prepare their real document in more comfortable (OpTeX) markup and with more simple macros.

This conversion should keep almost all features. For example, the `\pgref[`⟨*label*⟩`]` is converted to `\pageref{`⟨*label*⟩`}`. The cnv-program needs not to generate a Table of contents, bibliography references, etc. Only appropriate LaTeX markup must be used.

The contents of generated LaTeX preamble should be configurable.

7.5 From LaTeX

The reverse conversion from LaTeX to OpTeX can be usable for people, they want to switch to creating their documents in OpTeX.

The cnv-program must solve additional problem in this type of conversion. There are various math LaTeX environments, they must be converted to plain TeX syntax of math typesetting. For example:
`\begin{array}...\end{array}` → `\matrix{...}`,
`\begin{align*}...\end{align*}` → `$$\eqalign{...}$$`
`\frac{...}{...}` → `{...\over...}`.