The TOPtesi bundle

Claudio Beccari*

v.5.91 — 2015/12/18

Contents

Abstract

This file describes the TOPtesi bundle; it is a set of files designed to typeset a university final report that in Italian is generally called "tesi"; it was originally developed at the Technical University of Turin (Politecnico di Torino) but it was adapted for typesetting theses in any Italian university. Well... since the Erasmus student mobility is very extended and many Italian students participate in the so called *double degree* programs, their theses, or whatever they are called in other countries, may be typeset so as to comply also with the host university rules; therefore this set of files has the ambition to be suited for typesetting theses in any university in the world... This ambition can't be fulfilled, though, because of the complexity of the title page (and possibly of the legal page) requirements. This version is experimentally compliant with the X͟ΗLᴬTᴇX and the LuaLᴬTᴇX programs. Up to now the few conflicts that have been spotted have been solved with suitable corrections or additions. The most important feature with X͟ΗLᴬTᴇX is that the option pdfa cannot be used any more; this is due to the fact that the typesetting engine X͟ΗTᴇX does not produce directly any PDF output but a modified, extended DVI output, that is immediately converted into a PDF file through xdvipdfmx, a special version of the conversion program. Another minor X͟ΗLᴬTᴇX feature is that it cannot fully exploit the typesetting facilities of the microtype package, but the wide choice of OpenType fonts replaces almost completely such missing features.

---

*E-mail: `claudio dot beccari at gmail dot com`

1

1 Introduction

The TOPtesi bundle contains a certain number of files.

1. A class file `toptesi.cls` to be used as the main document class; the main document may be any of a certain number of reports that in Italy are called with various names: "monografia", "monografia di laurea", "tesi di laurea", "tesi di laurea specialistica", "tesi di laurea magistrale", "tesi di dottorato", "dissertazione di dottorato", and so on. All these documents have in common the fact that they all conclude a period of university education. Moreover since they may conclude a double degree university program, they may receive foreign names such as, for example, "Projet de fin d'études", "Master thesis", and the like.

2. An extension package `toptesi.sty` that contains most of the code for the real typesetting; it might be used as an extension to other classes, although this use is seldom necessary.

3. A second extension package `topfront.sty` that contains the commands and the typesetting macros for the title page; this file may be used as an independent extension package to be added to, say, the report class file for typesetting just the title page; this file might be used as a template for setting up the title page fixed information in languages different from Italian. This package is part of the bundle, but it is automatically loaded only if the user did not specify the class option `noTOPfront` or its alias `usefrontespizio`; if this option is specified the user is free to use whatever other external package s/he prefers in order to typeset the tile page; in this case we suggest to use the package `frontespizio` and we address the user to its documentation. It is important to recall that if the `topfront` package has to be loaded, is is only at the execution of the `\begin{document}` command. There fore no title page commands of any kind (those defined by `topfront`) can be used in the preamble.

4. A third extension package `topcoman.sty` that defines a certain number of user commands suitable for typesetting technical matters.

5. Previous versions of this bundle contained also the logos of a set of universities; these logos are not distributed anymore because of legal constraints. Every one who is working on his/her degree course final report must retrieve the logo of his/her university, but s/he should pay attention to use it according to the rules and limitations of the university.

6. A documentation file `toptesi-it-xetex.pdf` written in Italian where every feature is explained in detail; essential information is given in this English documentation. The source file of the documentation `toptesi-it-xetex.tex` may be used as a sample or template for typesetting one's thesis with Xe-LaTeX.

The above files are complemented with a configuration file that any user may customise at will; this customisation file makes it easy to configure the bundle so as to make it suitable for another language; in facts the `babel` and `polyglossia` packages contain localisations for many languages, but such localisations deal with the standard infix LaTeX names and phrases and do not cope with the thesis title page requirements. This customisation is usable only if no `noTOPfront` or `usefrontespizio` option had been declared to the class.

TOPtesi was specifically conceived for typesetting theses with the LaTeX markup, and initially it was using the `tex` typesetting engine; later on this engine was substantially substituted by the `pdftex` one, which was capable of direct output of PDF files. Since about 2006 the typesetting engine X∃TEX is available; the most important feature of this engine is its capability of using OpentType fonts, among which those that are available to the operating system of the specific platform where the document is being typeset. For what concerns theses this might be of essential importance when the thesis deals with specific languages that use different scripts (Greek, Russian, Bulgarian, Chinese, Japanese, Korean, Hebrew, Arabic, Farsi, Thai, and so on).

This version of TOPtesi has been tested with X∃LATEX. Some conflicts have been spotted and solved; may be there are still hidden ones, so that feedback from users is particularly welcome. The only main drawback still present when running X∃LATEX is the fact that this program cannot still *directly* produce the output file in PDF format, although it automatically transforms its specific output file into PDF format. This implies that the specific `pdftex` features required to produce a PDF/A compliant output PDF file suitable for long term archiving cannot be used. But with some attention the PDF file output by X∃LATEX may be converted to PDF/A by using the program `ghostscript`.

2   User commands

The `toptesi.cls` is basically an extension of the standard class `report.cls`; it redefines the page typesetting grid, the headers and the footers, and the title page layout and commands. `toptesi.cls` does not set such crazy settings as "double spaced" text; it is intended to typeset the thesis with the quality of a LaTeX well typeset document, not as a typewriter written one.

Theses very often are full of specialised material: formulas, diagrams and pictures, texts written in non Latin alphabets, special symbols for philological markup, and the like; a common typewriter would not be suitable and the quality of the contents requires professional typesetting; this is why I strongly believe that instructions on typesetting styles that refer to the "gone by times" of mechanical typewriters should be banned.

At the same time the class allows to typeset theses on any paper format; nowadays, in facts, Universities are requesting theses in smaller formats than A4 or letter paper. With smaller paper sizes the layout changes automatically, but the title page might require some more attention. The default paper size is A4, but the user can set any paper size among those accepted by the `report` standard

class. If the `noTOPfront` option was specified and the package `frontepsizio` was loaded by the user, it is necessary to remind that it can easily create pretty nice title pages, but by default it is preset to use A4 paper where it can typeset the tytle page in two styles, one so called "standard" and the other called "elements", that mimics that of the famous book *Elements of typographic style* by Robert Bringhurst; this is so at least with `frontespizio`'s version 1.4a of 2011/09/21 or earlier. It is possible to use `frontespizio` to use a different paper size, but it's necessary to make use of certain customisation commands described in its documentation.

Therefore if you want to typeset your thesis while typesetting the title page by means of the functionalities of package `frontespizio` you have to follow carefully how to use that package's options in the proper way as to bypass the two fixed styles it can produce. See also the Italian documentation contained in the document `toptesi-it-xetex.pdf`.

Most new commands refer themselves to the information that should be typeset in the title page; some class options specify special stylistic page details; the rest is simple and traditional LaTeX mark-up as it is implemented in the LaTeX kernel and in the report class. If X∃LaTeX has to be used, some essential preamble specific commands are to be used, but essentially the body of the thesis has the same mark-up.

2.1 Class options

The class accepts all the options accepted by the `report` document class plus the ones defined here:

chapterbib Allows to typeset a list of references at the end of each chapter and the bibliography items are numbered with a *chapter.item* indication. This implies a manual build up of each end-of-chapter bibliography by means off a specific `thebibliography` environment. This works may be avoided if the user relies on the various packages already available in any complete TeX system distribution and the facilities offered by packages such as *biblatex* and sorting end formatting engines such as `biber`.

classica Specifies a general modification of certain details that are supposed to be more adequate for theses in humanities; specifically this option lets old style numbers to be used for certain numerical pieces of information; some variations are also introduced in the title page.

cucitura In two sided printing it is better to move the typesetting grid towards the outer edges so as to cope with the thesis binding that is generally not made up by sewing together a number of signatures; the default outer displacement is fixed to 7mm, but it can be customised by means of a proper entry in the configuration file or by an explicit command in the preamble.

14pt Extends the normal size choice to 14 points; it is appreciated in various fields of humanities, but I would discourage this use in a technical thesis, where

10 point perhaps is too small, but 11 point or 12 point typesetting may be adequate.

autoretitolo This option modifies the left hand (even numbered) pages in two side typesetting; normally the even numbered page headings contain the chapter title, while the odd numbered ones contain the current section title. If the `classica` option has been specified, then with this option it is possible to have even numbered headings contain the author's name and the thesis title, while the odd numbered ones contain the chapter title. Since the thesis title might be too long to fit in the header together with the author's name, the `\title` macro as been modified so as to accept an optional short title, similarly to the ordinary sectioning commands.

oldstyle Also this option works only if `classica` had already been specified; it typesets several numerical data with the old style numbers.

noTOPfront allows the user to load a different package for creating and typesetting the title page; of course if this option is specified, the native package `topfront` is not loaded, and the commands described in the next section are not available. On the opposite, commands available with the chosen package become available, while possible package conflicts are eliminated at the root.

usefrontespizio is basically equivalent to the previous option; both options set up the `\ifTOPfront` switch that is more mnemonic than the long (actually not defined) `\iffrontespizio` one; if it was defined, `\iffrontespizio` would be true when `\ifTOPfront` is false.

2.2 Title page commands

Note: Skip this section if you decide to load an external package for the title page. Notice also that if you want to load an external package for typesetting the title page, you should load it *after* specifying the input encoding you use for your text files.

The user must specify a certain number of commands in order to have the title page contain all the required information. It must be specified that most of these commands may be used in the configuration file so as to avoid repeating the same data for different "final reports"; well, a university student might write a bachelor's "monografia", then a master thesis and finally a doctoral dissertation; why should s/he repeat his/her name, the name of the institution, and so on?

All the user commands for the title page redefine default values or strings; therefore if none of the required information is given, the default values and strings are typeset, possibly with hilarious results...

Since most users are supposed to be Italian, the user commands are mostly in Italian; the following description gives their names and meanings; every command receives one argument; only the command `\title` accepts an optional argument according to the usual LaTeX syntax:

$$\backslash command \, \texttt{[}opt\_arg\texttt{]}\texttt{\{}req\_arg\texttt{\}}$$

fontespizio and **frontespizio*** are two environments that typeset the title page with two opposite styles: with the university logo in the header vs the university logo in the lower part of the front page. The commands described in the following items shall be used either in the configuration file or within the body of these environments.

\frontespizio Even if the above environments are recommended, the command **\frontespizio**, defined in the previous versions of the TOPtesi bundle, is still usable. It typesets the title page according to the actual status of the boolean **\topTPTlogos**; the user, before issuing the **\frontespizio** command, may set this boolean by means of the **\topTPTlogostrue** or **\topTPTlogosfalse**, or with either one of the commands **\booltrue{topTPTlogos}** and **\boolfalse{topTPTlogos}**.

\monografia sets the bachelor's report style and retrieves its title

\titolo gets the master or PhD thesis title and an optional thesis short title

\sottotitolo gets the thesis subtitle if any

\materia gets the name of the subject the thesis deals with

\Materia alias for **\materia**

\direttore gets the name of the Doctoral School Director

\coordinatore gets the name of the Doctoral School Coordinator

\QualificaDirettore gets the phrase that describes the director or coordinator official position; by using the command **\direttore** the default phrase "Direttore della Scuola di Dottorato" is printed above the "director's" name; if **\coordinatore** is used the default phrase "Coordinatore della Scuola di Dottorato" is printed instead. If neither one is applicable or a description in another language is required, this macro is available for specifying such position.

\relatore gets the name of the thesis first supervisor

\secondorelatore gets the name of the second supervisor

\terzorelatore gets the name of the third supervisor; it is assumed that the number of supervisors never exceeds three.

\tutore gets the name of the doctorate tutor; there is no difference with regards to the **\relatore**, but the default phrase "Tutore" is printed above this person's name.

\TutorName gets the phrase that describes the tutor position, possibly in a different language.

\AdvisorName gets the string that qualifies the supervisor(s); the default string is "Relatore:" or "Relatori:" for the plural; in another language this command is used to define the string, say, "Supervisors:" if the thesis has been supervised by more than one person.

\CoAdvisorName gets the string that qualifies the co-supervisor(s); the default string is "Correlatore:" or "Correlatori:" in the plural; this command may be used to define the string, say, "Corapporteur:" in a French Projet de fin d'études.

\candidato gets the name of the male author

\candidata gets the name of the female author

\secondocandidato gets the name of the second male author

\secondacandidata gets the name of the second female author

\terzocandidato gets the name of the third male author

\terzacandidata gets the name of the third female author; most often the thesis author is just one person; but there are some institutions where group final works are accepted; it is assumed that the group does not contain more than three authors. The specification of the gender allows the software to determine the correct labelling phrase in the proper gender and proper number. For different languages there might be no difference in gender but there is a difference in the plural ending.

\CandidateName gets the string that describes the student status in a foreign language or even in Italian; the default string is "Candidato:" (with colons) adjusted to masculine or feminine, singular or plural; with the option `classica` the string becomes "Laureando:"; in other languages it is necessary to specify this string in the proper gender and number.

\sedutadilaurea gets the date of the final exam, or presentation, or defence of the thesis; if this date is omitted the default date is the current month and year in Italian.

\esamedidottorato an alias for **\sedutadilaurea** to be used for doctoral dissertations.

\ciclodidottorato gets the roman numeral that specifies the doctoral cycle.

\CycleName redefines the string that expresses the name of the doctoral cycle; by default this is "ciclo" but this command is useful to set the name in a different language.

\corsodilaurea gets the proper name of the degree course; the phrase that describes the degree course is specified, if necessary, with the next command; with this one you specify just, say, "Electrical Engineering"

`\CorsoDiLaureaIn` gets the generic name of the degree course, for example "Bachelor Degree in␣"

`\TesiDiLaurea` gets the generic phrase that describes the thesis; by default it is "Tesi di Laurea"; in English one might set it to "Master Thesis".

`\NomeMonografia` gets the phrase that describes the bachelor's report; by default it is "Monografia di Laurea". In some Italian universities it might be called "Tesi di Laurea", so that the master thesis should be given another qualification, for example "Tesi di Laurea Magistrale".

`\NomeDissertazione` gets the phrase that describes the doctoral thesis; by default it is "Tesi di Dottorato".

`\InName` infix strings often require adjusting of the prepositions; this macro gets the preposition that stands for "in" (the default). In German it might become "auf".

`\NomeAnnoAccademico` defines the infix string that stands for "Academic year". This macro is defined only if the option `classica` is in force; after all the command `\annoaccademico` is defined only with that option.

`\logosede` specifies the name of the file or the files that contain the university logos; no default is defined; rather a warning message is issued if no name is given or the file is missing, but typesetting goes on without the inclusion of any logo. A list of logos can be specified, useful when a thesis is carried on in a multiple University environment such as, for example, in a double degree Erasmus program; or under the Erasmus Mundus program. The "string" of logos is scaled properly so that they may fit in `\textwidth`.

`\setbindingcorrection` sets up the length to displace the text block to the external margin so as to have a wider internal margin to accommodate the binding correction. Its argument is not optional and is used to modify the default correction of 7 mm. Notice that 7 mm is already a large displacement; most often than not the binding correction is unnecessary.

`\retrofrontespizio` with its argument, made up of one or several paragraphs, defines what should be printed on the verso of the title page, generally named "copyright page"; if this command is specified with an empty text or if it is not used at all, no copyright page is assumed.

Since the infix strings are all memorised into control sequences and for each of them it is possible to use a defining command, all strings can be modified at will, so that there is no difficulty to localise the package in another language; this comes particularly handy for the Erasmus students on double degree programs.

As a final remark notice that the commands for typesetting the title page and the copyright page are contained in the package `topfront.sty`, which can be used as an autonomous extension to the `report` document class. One could easily typeset either just the title page with a separate TEX source file so as to test the

completeness of the commands and coherence of the configuration file, or for just printing the isolated title and copyright pages (if any).

2.3   Typesetting commands

The TOPtesi bundle and the `toptesi` document class accept all LaTeX commands provided by the LaTeX `kernel`, the `report` document class, and the `graphicx` extension package, besides those provided by the `babel` package. If the source thesis file is being typeset by means of X∃LaTeX the `babel` package is not loaded; in its place the `polyglossia` package is loaded that should implement in X∃LaTeX most of the functionality provided by `babel` in LaTeX. "Most" means that not all the functionality is available, therefore it's better to consult the documentation of `polyglossia` before using its built in commands.

   With this respect it must be underlined that the Italian and English languages are specified by default, the Italian one being the main language. An initial specification of `\selectlanguage{english}` sets the English language as the default. Should a student typeset the thesis in French by means of `pdflatex`, it would be necessary to specify the option `french` among the *class* options, and then start the document by specifying `\selectlanguage{french}`. But the user should pay attention to use `babel` in the proper way.

1. Due to the way LaTeX classes load the requested files, and to the fact that the `babel` package has already been loaded by the `toptesi` class, the user cannot reload it with a different list of language options; therefore the latter language options *must be specified* as a global class options; so if the thesis has to be typeset, for example, in French, is is necessary to do the following:

   ```
   \documentclass[...,french,...]{toptesi}
   ...
   \begin{document}
   \selectlanguage{french}
   ```

   But if the thesis should be typeset in French by means of X∃LaTeX, then it is perfectly legal to specify the auxiliary language in this way:

   ```
   \documentclass[...]{toptesi}
   \setotherlanguage{french}
   ...
   \begin{document}
   \selectlanguage{french}
   ```

2. With the 2010 TeX system complete distributions, both TeXLive and MiK-TeX, all known (to LaTeX) language hyphenation rules are preloaded; with other less up to date TeX system distributions this might not be true. With both distributions also the language hyphenation rules known to X∃LaTeX are all preloaded. Remember that `babel` and `polyglossia` macros select the language typesetting rules, but hyphenation is activated only if the program format file has been generated with the pertinent language hyphenation

rules. You can check this detail by reading the first dozen lines of your thesis `.log` file; there the list of all language hyphenation rules that are available in the format file is being shown. While (American) English is the default language and almost any *basic* distribution of the TeX system has several preloaded languages, it is more likely that French is preloaded while Italian is not. Complete distributions don't exhibit this flaw.

Should the required language(s) be missing, the user is forced to read his/her distribution instructions, so as to find out how to configure his/her system in order to preload the languages s/he wants to work with, and finally s/he must recreate the format files. The user is invited to carefully investigate on these fine points and to properly configure the system; it would be very upsetting to use fine software for producing a perfectly typeset thesis that, unfortunately, has wrong hyphenation points! Luckily enough, recent distributions of the TeX system have all the known hyphenation rules preloaded; in any case even older distributions have available command-line commands or graphical user interfaces that make it easy to perform the tasks of changing the list of preloaded hyphenation rules and rebuilding all the format files, moving them to the proper places.

The TOPtesi bundle adds very little to the user commands; nevertheless the package `topcoman.sty`, that is part of this bundle and is automatically loaded, defines some useful commands for typesetting technical matters in such a way as to fulfil the ISO regulations. Some of these commands are already defined with the `babel` Italian option, but if your thesis is written in different languages it may happen that such commands are not available any more when you select another language; with the presence of the definitions contained in `topcoman.sty` such useful commands remain available with every language. The `polyglossia` package does not produce any useful additional command for writing in a ISO compliant way, therefore the macros contained in the `topcoman.sty` package come very handy.

The following description specifies these particular commands.

`\DeclareSlantedCapitalGreekLetters` does exactly what its name means: it changes the definitions of the mathematical capital Greek letters so that they are typeset in "italics"; they are in effects taken from the math italic alphabet, instead of the default roman one. This option is useful with LaTeX, while with X∃LaTeX it is unnecessary due to the larger set of math alphabets and math font commands that are available with proper UNICODE math fonts.

`\ensuremath` should be already defined in the LaTeX kernel; should one still be using an obsolete version, this command gets available anyhow.

`\ohm` typesets an upright capital omega even if the capital Greek letters are in italics; another good point is that `\ohm` can be used also in text mode.

`\ped` inserts a subscript in upright type; the ISO regulations require the use of italics for physical or mathematical quantities, and upright type for anything

that is not a variable, from the names of functions (such as sin, cos, log, etc.), to the indices that contain information on something that is not variable. This means that V_i requires an italic index to imply that the object V is the i-th in a set, while, say, $V_{\max}$ indicates the maximum value of the variable V. This command `\ped` may be used in both math and text mode.

`\ap` similarly `\ap` inserts an apex in upright type, in both math and text mode.

`\unit` sets the unit of measure close to the numerical measure value by inserting a non breakable thin space and by setting the units of measure in upright type; this works both in math and text mode. Of course it is necessary to input the `\unit` command without intervening spaces in the source file; it's necessary to typeset, say, `35\unit{km}` and to avoid writing `35␣\unit{km}`. This command, as it is defined, conflicts with the definition of the homonymous command `\unit` as defined by the `unitsx` package, but since this latter package is necessarily input in the preamble, while `topcomand.sty` is read at the execution of `\begin{document}`, the latter file ha provisions to test if `unitsx` has already been loaded, so as not to overwrite the latter package definition and to avoid every possible conflict.

`\micro` sets the decimal prefix μ when typesetting units of measure.

`\gradi` sets the small circle that defines the sexagesimal degrees, for example $35°$; it may be used also for the celsius degrees by writing in the source file, say, `35\unit{\gradi C}` in order to get $35\,°C$; notice the difference with `35\gradi`, that yields $35°$ without any space between the measure and the "degree" little circle; this is what the ISO regulations require; The small space inserted by `\unit` is required when the unit of measure symbol contains at least one letter.

`\gei` inserts the imaginary unit in upright type with the "spelling" used by the technologists: "j". This command may be redefined, of course, but this strange name is due to the fact that nowadays the letter "j" in Italian is called with the English name (much shorter than the traditional Italian name "i lunga") and the indicated spelling "gei" is the phonetic Italian rendering of the English word. The imaginary unit is not a variable, and the ISO regulations require it is typed with an upright serifed font, just as operators are.

`\eu` inserts the Napier number symbol "e" in upright type; since this entity is not a variable, but it is a mathematical constant, the ISO regulations require it to be written in upright type. The ISO regulations require the upright type for "e" and any other mathematical constant, but the electron charge e is typeset in math italics because this is physical "constant", not a mathematical one. X∃TEX allows to typeset upright math Greek letters, so there is the facility to typeset an upright "pi" (the number) to be distinguished from an slanted "pi" (the angle).

`\listing` requires for its argument the name of a file and typesets it in verbatim mode; this command is very useful for typesetting the listings of the programs that were written for the thesis; for best results it is recommended that the source program has lines not longer than 80 characters. There exists an external package `listings`, that the user might want to load, that does a similar work in a more professional and configurable way; this package simpler command might be preferable for its simplicity, but of course it is not compulsory.

All these commands are defined into the separate package `topcoman.sty` that might be used as an independent extension package with any document class.

3   PDF/A archivable format

This section in general does not apply if the thesis is typeset by means of X$_\exists$LAT$_E$X, because the typesetting engine in this case cannot directly produce a PDF formatted output file. Nevertheless even a PDF produced with X$_\exists$LAT$_E$X can be made PDF/A compliant; even better, LuaLAT$_E$X, whose source file is practically the same as that of X$_\exists$LAT$_E$X, may directly produce a PDF/A compliant output file if the user uses a little patch described further on. The patch is integrated into TOPtesi, and the user might be interested only if s/he used a different class to typeset some of the included files.

More details are given in the Italian documentation contained in files `toptesi-it-pdftex`.

Politecnico di Torino as well as many other Italian and foreign universities are moving towards archiving theses in electronic format, specifically in the PDF one. The problem of course is: "Will it be possible to read the archived documents, say, fifty years from now?"

This essential question has been answered by the International Standards Organisation (ISO) that in 2005 published the regulation ISO 19005-1. This regulation defines a PDF variant suitable for archiving, named PDF/A, that has two sub-formats distinguished as PDF/A-1a, and PDF/A-1b. The 'a' sub-format is more exacting, while the 'b' one is less stringent. Recently ISO has published upgrades to these regulations, but here we stick to the older ones, that are a sub set of the newer ones, and for which suitable software is available.

The requirements for the 'a' sub-format imply not only those imposed on the 'b' one, but also that all characters are conforming to UNICODE and that the logical structure of the document be accessible from outside (Tagged PDF). The requirements for the 'b' sub-format are that the document must be reproducible without modifications exactly as it was at the moment of archiving. Both sub-formats must contain *metadata* that are searchable even without decompressing the normally compressed PDF file, and that contain information useful for archive maintenance; among these informations, of course, the PDF/A category the document belongs to, the document title, the authors, and few other optional pieces of information, such as the keywords that ease up library searches.

Since version 1.40, the program `pdflatex` is capable of producing PDF/A-1b compliant files, provided that some attention is put into the manipulation of the source file of the thesis. With the distribution of the 2008 version of the TeX system, the executable `pdflatex` has version number 1.40.9 (or higher) and is capable of producing PDF/A-1b files.

The particular attention needed to avoid problems with the PDF/A compliance is summarised as follows:

1. The preliminary essential requirement is that the `pdflatex` engine used to typeset the thesis be sufficiently recent to support the PDF/A requirements. It's better to have the most recent distribution of the TeX system installed on your PC. Do not try to typeset the thesis with the 'old' LaTeX; you *must* process the input thesis file(s) with `pdflatex` or with xelatex or with lualatex; in the following, no specific check will be made in order to verify if you are actually using `pdflatex`. If you really need to use the 'old' LaTeX, you get a DVI file and you need to transform it with `dvips` into a PS file; at this point you have lost the possibility of exploiting the internal commands of `pdflatex` version 1.40.9 or later. You can still produce a PDF/A final document, but you have to transform it by means of `ghostscript`; read the `ghostscript` (version 9.16 or later) documentation file `ps2pdf.html` in order to find out how to produce the correct PS-to-PDF/A transformation. A procedure similar to this one is available when the thesis is typeset by means of X∃LATeX, and produces valid results provided some caution is exercised when using fonts. The documentation in Italian describes several methods to transform a non compliant PDF/A file into a compliant one, and how to correct the "imperfections" that forbid any valid transformation.

2. As said before, lualatex is not fully compatible with package *pdfx*, because in spite of having a typesetting engine that has a strong resemblance with pdftex it does not contain all its primitives; in particular it misses the `\pdfmdfivesum` primitive. In order to supply a patch it is necessary to load the package *pdftexcmds* that defines some macros that emulate the missing pdftex primitives by means of macros. This patch is already integrated into TOPtesi, so when the user typesets his/her thesis with LuaLaTeX, non "undefined control sequence" errors show up, and their functionality is preserved by means of the above patch.

3. Your up-to-date complete TeX distribution should contain the package pdfx; if it does not, upgrade your TeX system complete distribution.

4. If your thesis main file, the one you run your `pdflatex` on, is named, say, `JohnSmithMasterThesis.tex`, prepare in the same directory another file named `JohnSmithMasterThesis.xmpdata` that contains the *metadata* relative to the thesis; pay attention to follow the stringent syntax described and exemplified in the pdfx documentation. A minimal set of *metadata* example would be the following one:

```
\Title{Experiments in Trichotetratomy}
\Author{John Smith}
```

The keyword list requires a specific format that can be examined in the package documentation.

5. Some mathematical symbol commands obtained from the standard LaTeX set-up and the standard mathematical fonts require some patching that is already included in this `TOPtesi` bundle; but it is not excluded that with other fonts similar patches might be requested. The UNICODE math fonts used by X$_{\text{E}}$LATEX do not require any patch, but unfortunately its PDF byproduct is not PDF/A compliant (it must be converted by means of `ghostscript`). With LuaLATEX there are no problems with fonts.

6. Use only PNG and JPEG images with RGB color profiles.

7. If you include PDF images that contain some text, be sure that the fonts for this text are completely embedded in the included file. Should the PDF file come from an external drawing program be sure to configure that program so that it embeds all the fonts used in the image. If you don't succeed, open the PDF file with the free program `inkscape` and save it back in PDF format; the missing fonts will be replaced with their traced outlines and this will not disturb the PDF/A conformity. Pay attention, because Inkscape might save your file with some specifications for transparency; Transparency specification destroy the PDF/A compliance.

8. Verify your final PDF file with a suitable program and do not give up doing the necessary corrections or modifications while the verification program keeps saying that this or that is not conforming to the PDF/A specification. A suitable program is the Preflight module of Adobe Acrobat Pro XI, but this, although the most authoritative, is a commercial program; probably your university has special facilities for this task.

9. If you use xelatex and you want to convert it by means of `ghostscript`for typesetting your thesis, you have to accompany your master file with another one that contains the metadata and other necessary information. Be sure to call this auxiliary file has the masterfile name glued with `-def.ps`; so if your masterfile is named `JohnSmithMasterThesis.tex`, name this auxiliary file as `JohnSmithMasterThesis-def.ps` and fill in the necessary metadata and other technical information into this file in the lines marked with `% Customize`. Then run the `ghostscript` program with the proper name (different for Windows or UNIX-like platforms) with the proper input parameters and options as the `PS2PD.hhtml` file explains. There are strong chances that if you follow the general recommendations shown here and in the document `toptesi-it-xetex.pdf` file, the output file will be PDF/A compliant.

10. If you really have to typeset your document with xelatex because you have to use Opentype fonts or you have to manage languages in a different way from

what you can do with `babel`, then prefer using lualatex, even if you have to force the compilation; formally LuaLaTeX and package *pdfx* are incompatible (by the end of 2015), but if at the first error you force the continuation of the compilation you may end up with a compliant PDF/A file.

Warning: if you are using Adobe Reader X (or later), this program will open a PDF file beginning with an information header stating that the file PDF/A claims to be compliant; maybe it's true, but do not trust this information too much, at least don't believe that this information is a "certification" of the PDF/A compliance. I have seen files with this comforting information that did not pass the Preflight test!

Up to today the realisation of PDF/A conforming files sets forth several problems that are of great concern for the large Institutions that have thousands of documents a year to archive; it is not a question implied in the free nature of the `pdflatex`, `xelatex`, and lualatex programs, that, on the opposite, according to my experience have a very high rate of success in producing PDF/A compliant documents. If you stick to the default TEX system Type 1 256-glyph fonts, or to the UNICODE encoded otf or ttf fonts and use this patched version of `TOPtesi` you should be able to avoid most problems.

4 Documented code

4.1 The class code

Here begins the usual machinery for stating the required TEX format and for sharing some code between the driver and the class part of the code, since they are supposed to carry the same date and version number, besides the description string.

The class itself is very simple since it requires just the `report` document class and some packages with some default options. All options specified for the `toptesi` class are passed on to the `report` class; the `a4paper` option is made the default, but the user can call the class with any valid LaTeX paper size. According to the LaTeX machinery of option passing, the `\ExecuteOption` command assures that `a4paper` is the default size, but it is not being used if another paper size code is specified to the class. The same holds true for the other class options, except for the encoding code string because their passing machinery does not hold true for the *inputenc* package; therefore this package is not loaded any more as it was in previous versions of the class[1]. On the opposite if another language is specified in the list of *toptesi* class options, this language would be appended to the *babel* package default options and, beware!, it would become the default language.

```
1 \DeclareOption{a4paper}{\PassOptionsToClass{\CurrentOption}{report}}
2 \DeclareOption{titlepage}{\PassOptionsToClass{\CurrentOption}{report}}
3 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{report}}
4 \ExecuteOptions{a4paper,titlepage}
```

---

[1]Thanks to Enrico Gregorio who pointed out this feature.

```
 5 \ProcessOptions\relax
 6 \LoadClass{report}
 7 \RequirePackage{iftex}
 8 \ifPDFTeX
 9   \RequirePackage[english,italian]{babel}
10 \else
11   \RequirePackage{polyglossia}
12   \setmainlanguage{italian}
13   \setotherlanguage{english}
14   \renewcommand*{\iflanguage}[1]{\ifnum\the\language=\csname l@#1\endcsname
15   \expandafter\@firstoftwo\else
16   \expandafter\@secondoftwo\fi}
17 \fi
18 \RequirePackage{toptesi}
19 %
```

4.2   The `toptesi.sty` code

The greatest part of the `toptesi` class code is saved into a separate file partly for
backward compatibility reasons (before version 3.x `toptesi` was just an extension
to the `report` class) and partly because it might be used as a stand alone package,
although, take notice, it might create some conflicts with with the `book` class code
(although with the latest versions of `toptesi` we did not notice any conflict) while
certainly it is incompatible with the `article` class. This package must contain
its own TeX format declaration and might have different version and subversion
numbers compared to the class file and the other invoked packages that are part
of the same bundle.

We start with defining the debugging macros; these trace commands and
macros are the usual ones I use for debugging. I know the `trace` package is
supposed to be much better, but sometimes I use these ones.

```
20 \def\TRON{\tracingcommands \tw@ \tracingmacros \tw@}
21 \def\TROFF{\tracingcommands\z@ \tracingmacros \z@}
22 \let\TROF\TROFF
```

Now we define the specific package options: `classica` and `trieste` are identi-
cal; the use of `trieste` is discouraged, but this option is maintained for backward
compatibility. The option `14pt` is for choosing a normal size of 14pt; the class
option file `size14.clo` is distributed with the *extsizes* package and is already in-
cluded in any complete distribution of the major TeX system ones; in previous
versions of this bundle there was a specific option class file for this large size,
but with the latest complete distributions available, there is no chance that the
existing option file is missing.

`chapterbib` allows to set a list of references at the end of each chapter.

For the options some boolean variables are defined and the option definitions
set some of them to the value "true".

```
23 \newif\if@utoretitolo \@utoretitolofalse
24 \newif\if@ldstyle \@ldstylefalse
25 \newif\if@xivpt \@xivptfalse
```

```
26 \newif\ifT@Pfrontespizio \T@Pfrontespiziofalse
27 \newif\ifTOPfront \TOPfronttrue
```

A binding correction is established; its default value is parametrised to the paper dimensions, even if this correction should actually not depend on the paper size; the paper flexibility at the spine margin should be independent from its width; nevertheless large sizes allow for larger default corrections. In any case the user can override this setting by using the specific command \setbindingcorrection. The recommendation to the user is to not exaggerate with this correction. The options usefrontespizio and noTOPfront are synonymous; the latter should be preferred to the former one.

```
28 \newlength\T@Pbinding\setlength\T@Pbinding{7mm}
29 \def\setbindingcorrection#1{\T@Pbinding=#1}
30 \newif\if@binding \@bindingfalse
31 \newif\ifchapterbibliography \chapterbibliographyfalse
32 \newif\ifclassica \classicafalse
33 \DeclareOption{cucitura}{\@bindingtrue}
34 \DeclareOption{14pt}{\@xivpttrue}
35 \DeclareOption{chapterbib}{\chapterbibliographytrue}
36 \DeclareOption{trieste}{\classicatrue}% Just for backwards compatibility
37 \DeclareOption{classica}{\classicatrue}
38 \DeclareOption{autoretitolo}{\ifclassica\@utoretitolotrue\fi}
39 \DeclareOption{oldstyle}{\ifclassica\@ldstyletrue\fi}
40 \DeclareOption{pdfa}{\PackageWarning{toptesi}{Option pdfa ignored.\MessageBreak
41 Read documentation toptesi-it-*.pdf, because package pdfx has changed}}
42 \DeclareOption{usefrontespizio}{\T@Pfrontespiziotrue\TOPfrontfalse}
43 \DeclareOption{noTOPfront}{\T@Pfrontespiziotrue\TOPfrontfalse}
44 %
45 \ProcessOptions\relax
```

The graphicx package is loaded by default; it is required for inserting the university logo(s); if the user forgets that this package has already been loaded nothing dramatic happens, because the \usepackage and \RequirePackage macros perform the necessary tests in order to avoid reloading the same packages again and again. In order to produce PDF/A compliant files the necessary patches are very simple, but must be done directly into a personal copy of the *pdfx* package; see the Italian documentation about this point. This notice is done here because this would have been a convenient point to introduce the initial single patch I found; but this patch by itself is not sufficient.

```
46 \RequirePackage{graphicx}
47 \RequirePackage{etoolbox}
48 \if@xivpt\input{size14.clo}\fi
```

The \textheight is parametrised to the paper height and adjusted so as to contain an integer number of normal text lines. A new dimension is defined to hold the actual value of the inner/spine margin.

```
49 \newlength\interno
50 \textheight 0.7\paperheight
51 \setlength{\textheight}{\dimexpr\textheight*\baselineskip/\baselineskip+\topskip}
```

```
52 %\divide\textheight by \baselineskip
53 %\multiply\textheight by \baselineskip
54 %\advance\textheight by \topskip
```

The inner margin is parametrised to the paper width, but a small correction is made if the extra size of 14pt is chosen. Also `\footskip` is parametrised to the paper height in a slightly different way when the large `14pt` size is chosen.

```
55 \ifx\f@size\@xivpt
56     \setlength\interno{\dimexpr\paperwidth/6}
57     \footskip=1,5\baselineskip
58 \else
59     \setlength\interno{\dimexpr\paperwidth/7}
60     \footskip=2\baselineskip
61 \fi
```

The convenience of holding the spine margin widthin a dimensional register becomes really useful now in order to define the other text-body grid-dimensions. Without binding correction the inner and outer margin are chosen equal, but the grid is moved outwards if the binding correction option is specified.

```
62 \textwidth=\dimexpr\paperwidth-2\interno\relax
63 \oddsidemargin=\dimexpr\interno-1in\relax
64 \evensidemargin=\oddsidemargin
65 \marginparwidth=\dimexpr\interno-2.5\marginparsep
66 \AtBeginDocument{%
67 \if@binding
68  \PackageInfo{TOPtesi}{Margin width recalculation}
69  \PackageInfo{TOPtesi}{Before:\MessageBreak
70  oddsidemargin\space\space \the\oddsidemargin\MessageBreak
71  evensidemargin\space \the\evensidemargin}
72  \advance\oddsidemargin \T@Pbinding
73  \advance\evensidemargin -\T@Pbinding
74  \advance\marginparwidth -\T@Pbinding
75  \PackageInfo{TOPtesi}{After:\MessageBreak
76  oddsidemargin\space\space \the\oddsidemargin\MessageBreak
77  evensidemargin\space \the\evensidemargin}
78 \fi}
79
```

We now establish the page style. We start by setting to "empty" the tokens that keep the left and the right marks; we define a box so as to set the headers inside this box; we redefine also the plain page styles; it is actually a leftover from the previous versions when the page number was set at the foot in bold face, but we leave it here without the bold face specification, so that in future versions folios may be redefined in a common way with the other page styles. Notice that in all page styles folios are always in the footers. By defining `\lapagina` to be equivalent to `\thepage` we can later on redefine `\lapagina` the way we like; we actually do so with the option `classica`.

```
80 \def\lapagina{\thepage}
81 \mark{{}{}}
82 \newbox\@intesta
```

```
83 %
84 \def\ps@plain{\let\@mkboth\@gobbletwo
85 \def\@oddfoot{\null\hfill {\scshape\lapagina}\hfill \null}\def\@oddhead{}
86 \def\@evenhead{}\let\@evenfoot\@oddfoot}
```

Other page styles are defined in a different way according to the choice of one side or two side printing. In any case the header is set without the uppercasing that is done in all the default document classes, and it is underlined at a fixed distance from the base line. If the chapter or section heading exceed the \textwidth a warning is issued so as to invite the user to exploit the sectioning commands optional short argument.

```
87 \if@twoside
88     \def\ps@headings{\let\@mkboth\markboth%
89     \def\@oddfoot{\null\hfill {\scshape\lapagina} \hfill\null}
90     \let\@evenfoot\@oddfoot
91     %
92     \def\@evenhead{\setbox\@intesta\hbox{\footnotesize\slshape
93         \leftmark}%
94     \ifdim\wd\@intesta>\textwidth \headWarn{\chapter}\fi%
95     \underline{\makebox[\textwidth]{\footnotesize\slshape
96         \strut\leftmark}}}%
97     \def\@oddhead{\setbox\@intesta\hbox{\footnotesize\slshape
98         \rightmark}%
99     \ifdim\wd\@intesta>\textwidth \headWarn{\section}\fi%
100    \underline{\makebox[\textwidth]{\footnotesize\slshape
101        \strut\rightmark}}}%
102    \def\chaptermark##1{%
103        \markboth{\thechapter\ -- ##1}{\thechapter\ -- ##1}}
104    \def\sectionmark##1{\markright{\ifnum\c@secnumdepth>\z@
105    \thesection\ -- \fi ##1}}}
106 \else
107    \def\ps@headings{\let\@mkboth\markboth
108    \def\@oddfoot{\null\hfill {\scshape\lapagina} \hfill\null}
109    \def\@evenfoot{}
110    \def\@oddhead{\setbox\@intesta\hbox{\footnotesize\slshape
111        \rightmark}%
112    \ifdim\wd\@intesta>\textwidth \headWarn{\chapter}\fi%
113    \underline{\makebox[\textwidth]{\footnotesize\slshape
114        \strut\rightmark}}}%
115    \def\chaptermark##1{\markright{\thechapter\ -- ##1}}}
116 \fi
117
118 \def\headWarn#1{\PackageWarning{toptesi}{%
119    THE HEADING IS TOO LONG\MessageBreak
120    Use the optional argument of command \string#1\MessageBreak
121    See the LaTeX Handbook (1994) on section C.4.1\MessageBreak}}
```

It is also necessary to redefine the format of the unnumbered chapter entries in the table of contents so as to have page numbers in small caps.

```
122 \renewcommand*\l@chapter[2]{%
```

```
123    \ifnum \c@tocdepth >\m@ne
124      \addpenalty{-\@highpenalty}%
125      \vskip 1.0em \@plus\p@
126      \setlength\@tempdima{1.5em}%
127      \begingroup
128        \parindent \z@ \rightskip \@pnumwidth
129        \parfillskip -\@pnumwidth
130        \leavevmode \bfseries
131        \advance\leftskip\@tempdima
132        \hskip -\leftskip
133        #1\nobreak\hfil \nobreak
134        \hb@xt@\@pnumwidth{\hss\ifPDFTeX\normalfont\fi\scshape{#2}}\par
135        \penalty\@highpenalty
136      \endgroup
137    \fi}
```

The various tables of contents or figures or tables require some boolean variables to be defined; in facts, although the ISO regulations require that every technical report contains the list of figures and/or tables, in Italy theses rarely contain these lists; after all: is a thesis a technical report? We require also some other boolean variables to handle the difference between front matter and main matter; this differences are already defined in the `book` document class, but not in the `report` one.

```
138 \newif\iffigurespage
139 \newif\iftablespage
140 \newif\ifnumeriromani
141 \newif\iffrontmatter
```

The `\frontmatter` and `\mainmatter` commands are defined and at the beginning of the document the default situation of front matter is established.

```
142 \def\frontmatter{\clearpage\ps@plain\pagenumbering{roman}%
143      \numeriromanitrue\frontmattertrue\@openrightfalse\c@secnumdepth=-2}
144 \def\mainmatter{\if@twoside\@openrighttrue\fi
145      \numeriromanifalse\frontmatterfalse\c@secnumdepth=2
146      \clearpage\ps@headings\pagenumbering{arabic}%
147      }
148 \AtBeginDocument{\frontmatter}
```

The main matter is automatically established with the first `\chapter` command issued by the user; this means that every command that starts a section at the "chapter" level within the front matter must be executed without an explicit call to `\chapter`.

By default we set to false the boolean variables that control the typesetting of the list of figures and the list of tables.

```
149 \figurespagefalse
150 \tablespagefalse
```

Before going further on, we redefine the `\cleardoublepage` command so that it uses by default the plain page style, but it can be set to any other style.

```
151 \let\ps@blank\ps@plain
```

```
152 \newcommand*\blankpagestyle[1]{%
153     \expandafter\let\expandafter\ps@blank\csname ps@#1\endcsname}
154 \renewcommand\cleardoublepage[1][blank]{\clearpage\ifodd\value{page}\else
155     \if@twoside\if@openright
156 \clearpage\null\thispagestyle{#1}\clearpage\fi\fi\fi}
```

We have to define the front matter sectioning names \sommario and \ringraziamenti
so as to remain in the front matter.

```
157 \def\sommario{%
158  \iffrontmatter\else\frontmattertrue\fi
159  \chapter*{\summaryname}%
160  \addcontentsline{toc}{chapter}{\summaryname}%
161           }
162 %
163 \def\ringraziamenti{%
164  \iffrontmatter\else\frontmattertrue\fi
165  \chapter*{\acknowledgename}%
166           }
```

The strings \summaryname and \acknowledgename are not defined in any language
option to babel. Default definitions are given below, but the user must define new
names for localising the package in a language different from Italian and English.

We have to modify the \chapter and \part commands so that as the user
first issues one of these commands the typesetting style is switched to that of the
main matter. Actually it is not necessary to redefine the whole commands, but
just those of the unstarred versions. In facts the starred chapter commands to
typeset the Summary and the Acknowledgements chapters, as defined above, are
regularly typeset in the style of the front matter, opening on any page (even or
odd) and numbered with small caps roman numerals.

```
167 \def\@chapter[#1]#2{\iffrontmatter\mainmatter\fi
168                 \ifnum \c@secnumdepth >\m@ne
169                     \refstepcounter{chapter}%
170                     \typeout{\@chapapp\space\thechapter.}%
171                     \addcontentsline{toc}{chapter}%
172                             {\protect\numberline{\thechapter}#1}%
173                 \else
174                   \addcontentsline{toc}{chapter}{#1}%
175                 \fi
176                 \chaptermark{#1}%
177                 \addtocontents{lof}{\protect\addvspace{10\p@}}%
178                 \addtocontents{lot}{\protect\addvspace{10\p@}}%
179                 \if@twocolumn
180                   \@topnewpage[\@makechapterhead{#2}]%
181                 \else
182                   \@makechapterhead{#2}%
183                   \@afterheading
184                 \fi}
185 %
186 \def\@part[#1]#2{\iffrontmatter\mainmatter\fi
187     \ifnum \c@secnumdepth >-2\relax
```

```
188        \refstepcounter{part}%
189        \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
190    \else
191        \addcontentsline{toc}{part}{#1}%
192    \fi
193    \markboth{}{}%
194    {\centering
195      \interlinepenalty \@M
196      \normalfont
197      \ifnum \c@secnumdepth >-2\relax
198        \huge\bfseries \partname\nobreakspace\thepart
199        \par
200        \vskip 20\p@
201      \fi
202      \Huge \bfseries #2\par}%
203    \@endpart}
```

At the same time we have to make sure that `\tableofcontents`, `\listoftables` and `\listoffigures` do not exit from the front matter style. We assume these commands are issued while in front matter, the default at the begin document step, so we have to avoid to use starred or un-starred `\chapter` commands.

```
204 \renewcommand\tableofcontents{%
205  \chapter*{\contentsname}%
206  \@mkboth{\contentsname}{\contentsname}%
207  \@starttoc{toc}%
208  \clearpage
209  \if@restonecol\twocolumn\fi
210      }
211 \renewcommand\listoffigures{%
212      \chapter*{\listfigurename}
213      \@mkboth{\listfigurename}{\listfigurename}%
214      \addcontentsline{toc}{chapter}{\listfigurename}
215      \@starttoc{lof}%
216      \clearpage
217      \if@restonecol\twocolumn\fi
218      }
219 \renewcommand\listoftables{%
220      \chapter*{\listtablename}%
221      \addcontentsline{toc}{chapter}{\listtablename}
222      \@mkboth{\listtablename}{\listtablename}%
223      \@starttoc{lot}%
224      \clearpage
225      \if@restonecol\twocolumn\fi
226      }
```

We need to define `\indici` that typesets the table of contents and, optionally, the lists of tables and/or figures while assuring that the front matter style is used for typesetting.

```
227 \def\indici{%
228  \iffrontmatter\else\frontmattertrue\fi
```

```
229            \tableofcontents
230            \iftablespage
231                    {\addvspace{10pt}
232                    \let\saveaddvspace=\addvspace
233                    \def\addvspace##1{}
234                    \listoftables
235                    \let\addvspace=\saveaddvspace}
236            \fi
237            \iffigurespace
238                    {\addvspace{10pt}
239                    \let\saveaddvspace=\addvspace
240                    \def\addvspace##1{}
241                    \listoffigures
242                    \let\addvspace=\saveaddvspace}
243            \fi
244 \ifbool{@twoside}{\clearpage\thispagestyle{empty}\null\clearpage}{}}
```

Command `\onecolumn` is not actually necessary; it simply overrides the possible misused option `twocolumn` in the opening document class statement; no thesis should be typeset in two columns.

```
245 \onecolumn
```

Here come some declarations for vertical justification and for avoiding an hyphenated word at the bottom of a page

```
246 \if@twoside
247     \flushbottom
248 \else
249     \ifx\@xivpt\f@size
250         \raggedbottom
251     \else
252         \flushbottom
253     \fi
254 \fi
255 \brokenpenalty=10000
```

Here comes a questionable command and/or environment; good typesetting requires the baseline skip to be proportioned to the font size, generally it is some 10–20% larger than the font size. In some reasonable instances a larger or a smaller baseline skip might be required; the LaTeX kernel allows to use the `\linespread` command; in the previous versions of this bundle a command `\interlinea` and an environment `interlinea` were defined so as to allow setting the line spread factor. The experience has shown that some students tend to use this command so as to typeset a poor and thin thesis on more pages. Well, every instrument can be judiciously or maliciously used; this is one of those double sided instruments.

```
256 \def\interlinea#1{\linespread{#1}\selectfont}
257 \def\endinterlinea{\par}
```

But whatever might be the current line spread factor within figures and tables we reset this factor to the unit value; floating bodies do not belong to this or that section of text where a different spread factor might be reasonable.

```
258 \def\@floatboxreset {%
259         \reset@font
260         \linespread{1}%
261         \normalsize
262         \@setminipage
263 }
```

Since we are at it we define the floating bodies placing parameters; not only the values "here", "top of the page" and "bottom of the page", that we set as the default ones leaving to the user to explicitly specify the "page of floats", but also the numerical and geometrical parameters that control the float placements. These geometrical parameters are critical and everybody has his/her own ideas of what are the best values for them. According to my experience these parameters work quite well but I would not suggest them for every kind of typewritten document. In particular the zero value for the text fraction appears strange, but students have the tendency to create large figures (more than large tables) and these tend to clog the figures queue. A 100% space for the top of page figures and a requirement of 0% text allows large floats to exit the queue provided they do not exceed the text height.

```
264 \def\fps@figure{htb} \def\fps@table{htb}
265 %
266 \setcounter{topnumber}{2}
267 \def\topfraction{1}
268 \setcounter{bottomnumber}{1}
269 \def\bottomfraction{.5}
270 \setcounter{totalnumber}{2}
271 \def\textfraction{0}
272 \def\floatpagefraction{0}
273 \setcounter{dbltopnumber}{2}
274 \def\dbltopfraction{1}
275 \def\dblfloatpagefraction{0}
```

One command that the default definition does not satisfy me very much is the \caption command; actually it is the internal \@makecaption macro that performs the job. The point is that I prefer a narrower justified caption rather than a caption where the last line is just a short word or the right segment of an hyphenated word. If one sets the \finalhyphendemerits parameter to an incredibly high value, one might succeed in avoiding hyphenation in the last word of a caption; but this might lead to a very loose typesetting of the caption paragraph, especially if the horizontal box that contains it hardly exceeds the caption width. I initially set the caption width (a new length) to the overall text width diminished by 3em; then if the caption text, inclusive of the caption type string and number, is shorter than the text width it is typeset as centred text; if it exceeds the text width it is set as a justified paragraph whose line width equals the established caption width; but in any case the last line of the paragraph is measured and if it is shorter than one third of the caption width, this width is shortened a little bit and the paragraph is set again with this shortened caption width; in order to be sure that one iteration is sufficient, the caption width shrinking must be computed according to

the number of lines the paragraph occupies.

In order to count the number of lines the caption paragraph occupies it is necessary to recall that the first line occupies a vertical space that equals \topskip while the other lines occupy a vertical space equal to \baselineskip; the latter one is generally larger than \topskip therefore the integer division of the height of the vertical box divided by the \baselineskip is truncated to count a line less than the true value. In our case if the caption does not stay in one line, its text is typeset in a vertical box with a line spread of 0.95 so as to make the caption a little more compact than the regular text. The baseline skip is a little shorter than the regular one, but it should still exceed the default \topskip; when we first typeset the caption in a vertical box we strip off the last line and we must remember the presence of this line in our arithmetics.

If the length of the stripped last line is longer that one third of the caption width, then the vertical box is recomposed by restacking the individual lines, but if this last line is shorter than one third of the caption width, this width is recomputed in this way: let $N-1$ be the number of lines obtained with the integer division, i.e. one line less than those actually contained in the vertical box. Let x be the initial caption width and x_2 the last line width; let y be the new caption width; then the total length of the caption of width x is $Nx + x_2$ and this should be distributed over $N+1$ lines; if we obtained the new caption width y by dividing the total length by $N + 1$ we should be able to typeset the whole caption with all the lines of equal length. This does not actually take place because the new lines do not necessarily contain the same amount of inter word space, some words might have been hyphenated in a different way, and so on. Moreover we do not want a caption that barely exceeds the length of \captionwidth to be retyped into a two line caption width that is about one half of the width of the other captions. Therefore we allow for some white space in the last line by computing the new caption width with the following formula

$$y = \frac{(N + 0.5)x + x_2}{N + 1}$$

```
276 \newdimen\captionwidth
277 \long\def\@makecaption#1#2{%
278    \begingroup
279    \small \parskip\z@ \parindent\z@
280    \finalhyphendemerits 100000\relax
281    \linespread{0.95}\selectfont
282    \vskip \abovecaptionskip
283    \captionwidth=\hsize
284    \advance\captionwidth-3em
285    \setbox0 \hbox{#1.\quad#2}%
286    \ifdim\wd0>\hsize
287       \setbox1 \vbox{\hsize=\captionwidth
288       \unhbox0\par\global\setbox2\lastbox}%
289       \setbox2\hbox{\unhbox2}%
290       \ifdim\wd2<0.333333\captionwidth
291          \count255=\ht1 \advance\count255 \dp1
```

```
292        \divide\count255\baselineskip
293        \advance\count255\@ne
294        \@tempdima=\wd2
295        \advance\@tempdima \count255\captionwidth
296        \advance\@tempdima 0.5\captionwidth
297        \advance \count255\@ne
298        \divide \@tempdima \count255
299        \captionwidth=\@tempdima
300        \setbox0 \vbox{\hsize\captionwidth
301        #1.\quad#2}
302      \else
303        \setbox2\hbox to\captionwidth{\unhbox2 \hfill}%
304        \setbox0\vbox{\unvbox1\box2}%
305      \fi
306    \fi
307    \makebox[\hsize]{\box0}%
308    \endgroup
309 }
```

The option `chapterbib` requires a redefinition of the `thebibliogrpahy` environment in case a separate reference list is required for every chapter. The point is that for this task the reference key must contain also the chapter number; the rest is simply a redefinition of the environment that behaves differently according to the chosen option. In any case the bibliography goes to the table of contents as an unnumbered chapter or section.

```
310 \def\redef@bibitem{\def\@bibitem##1{\item\if@filesw
311        \immediate\write\@auxout
312        {\string\bibcite{##1}{\thechapter.\the\c@enumi}}\fi\ignorespaces}}
313 %
314 \def\thebibliography#1{%
315 \ifchapterbibliography\section*{\bibname}\relax
316      \if@twoside\markright{\bibname}\fi
317      \addcontentsline{toc}{section}{\bibname}\relax
318      \redef@bibitem
319      \list{[\thechapter.\arabic{enumi}]}{%
320      \settowidth\labelwidth{[\thechapter.#1]}\leftmargin\labelwidth
321      \advance\leftmargin\labelsep\itemsep\z@ plus 1pt\parsep\z@
322      \usecounter{enumi}}
323 \else
324      \chapter*{\bibname}\relax
325      \@mkboth{\bibname}{\bibname}\relax
326      \addcontentsline{toc}{chapter}{\bibname}\relax
327      \list{[\arabic{enumi}]}{\settowidth\labelwidth{[#1]}%
328      \leftmargin\labelwidth
329      \advance\leftmargin\labelsep\itemsep\z@ plus 1pt\parsep\z@
330      \usecounter{enumi}}
331 \fi
332 \def\newblock{\hskip .11em plus .33em minus -.07em}
333 \sloppy
334 \sfcode`\.=1000\relax}
```

```
335
336 \let\endthebibliography=\endlist
```

For what regards footnotes nothing is changed except resetting the line spread to one, in case the current value is different.

```
337 \long\def\@footnotetext#1{\insert\footins{\linespread{1}\footnotesize
338      \interlinepenalty\interfootnotelinepenalty
339      \splittopskip\footnotesep
340      \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
341      \hsize\columnwidth \@parboxrestore
342      \edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
343      \@makefntext{\rule{\z@}{\footnotesep}\ignorespaces#1\strut}}}
```

These last heterogeneous definitions are partly important and partly residues of the good old times of MS-DOS v.3 when a Ctrl-Z character would be placed at the end of files. It's a long time that such DOS version is not being used, but some old time files might still be around.

The cryptic code that redefines the comma in math mode establishes that this character is a normal math character, instead of a math punctuation mark. Actually the code that defines the mathematical active comma is a new addition that lets the comma perform correctly in its double function (decimal separator and punctuation mark). The only point where this code fails is when a list of numbers is typeset: when a numeric list must be typeset, such as, for example, $\forall i = 0, 1, 2, 3, n$, in the source code a space *must* be inserted after every punctuating comma while no space follows a decimal comma: for example $\forall␣i=0,␣1,␣2,␣3,n$. The space before the n is not necessary (but it wouldn't hurt) because n is not recognised as a digit, therefore the "intelligent" comma inserts the necessary space by itself.

```
344  \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
345  \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}
346  \AtBeginDocument{\mathcode'\,=\string"8000}
347  {\catcode '\,=\active \gdef,{\futurelet\let@token\m@thcomma}}
348 \def\m@thcomma{\let\@tempB\virgola
349 \@tfor\@tempA:=0123456789\do{%
350 \expandafter\ifx\@tempA\let@token\let\@tempB\virgoladecimale
351 \@break@tfor\fi}\@tempB}
352 %
353 \catcode'\^^Z=10
354 \topmargin 0pt
```

The TOPtesi bundle contains two new chapter like sections activated with the commands \sommario and \ringraziamenti respectively. The infix strings that start these sections depend on the used language.

Because of this it is necessary to extend the list of infix string definitions provided the \captions⟨language⟩ macros defined by the language description files of babel or of polyglossia; for this purpose we define a macro for adding new items for these two new sectioning commands. This macro receives three arguments: the first is the babel language name, the second is the string for the summary name, and finally the third is the string for the acknowledgements name;

everything is contained within a group and only the relevant captions macro is globally redefined. The token register '0' is normally for scratch usage, but the fact that its value is restored upon exiting the group provides the necessary protection against an involuntary reassignment to this register. At the same time if a specific language option was not specified, a warning message is issued, but compilation goes on any way without the sectioning string names. For being sure no other undefined error messages are issued, the `\summaryname` and `\acknowledgename` are let to `\empty`.

```
355 \providecommand*\summaryname{}
356 \providecommand*\acknowledgename{}
357 \newcommand*\ExtendCaptions[3]{{%
358 \@ifundefined{captions#1}{%
359 \PackageWarning{toptesi}{Language option #1 not specified\MessageBreak
360 Skipping any redefinition\MessageBreak}%
361 }{%
362 \expandafter\let\expandafter\@tempA\csname captions#1\endcsname
363  \toks0=\expandafter{\@tempA%
364     \def\summaryname{#2}%
365     \def\acknowledgename{#3}}%
366  \expandafter\xdef\csname captions#1\endcsname{\the\toks0}%
367 }}}%
```

For Italian and English there are no problems; we provide immediately these extensions by means of the newly available macro:

```
368 \ExtendCaptions{italian}{Sommario}{Ringraziamenti}
369 \ExtendCaptions{english}{Summary}{Acknowledgements}
```

In facts the `babel` options for Italian and English have already been loaded by default; therefore both caption macros `\captionsitalian` and `\captionsenglish` are already defined and can be freely extended. For any other language the corresponding language option must be entered in the class opening statement, otherwise a warning is issued but compilation is not stopped. Therefore if, for example, the user wants to write the thesis in Spanish, the thesis main file shall start like this:

```
\documentclass[...,spanish]{toptesi}
\ExtendCaptions{spanish}{Resumen}{Agradecimientos}
...
\begin{document}
\selectlanguahe{spanish}
...
```

and the rest of the thesis will be typeset correctly in Spanish. Remember that Italian is the main language and nothing is necessary to set up the Italian defaults. If English is desired, then after `\begin{document}` is is necessary to specify the default language; for ease of use the following macros are defined so they can be used instead of the lengthy `babel` command; after the beginning of the document it is then possible to specify `\inglese` or `\english`, and the default language

turns into English. These shorthand commands can be alternated so as to switch from one language to the other; nevertheless remember that there are more correct ways to switch languages with the `babel` commands without changing the infix strings.

```
370 \def\italiano{\selectlanguage{italian}}%
371 \def\english{\selectlanguage{english}}%
372 \let\inglese\english
```

At the beginning of the document the following commands are executed; the general macro `\italiano` sets the summary and the acknowledgements names in Italian, as the main language; and the `\@chapapp` macro is redefined so as to agree with the default language. If a different default language is desired, we recall it again, it is necessary to do the following:

1. specify `\english` *after* the `\begin{document}` statement, if English is supposed to be the main language, or

2. specify the language name, other than Italian or English, among the class options; use the `\ExtendCaptions` macro for extending the list of sectioning commands infix strings as explained above; specify with `\selectlanguage` the new language as the default one *after* the `\begin{document}` statement;

3. if XeLaTeX is being used for typesetting it suffices to specify in the preamble the name of the other language to be used, by means of the `setotherlanguage` command, and to define the summary and acknowledgments names in the same way as with pdfLATEX; the same must be done at the beginning of the document to declare the new language a the default one.

```
373 \AtBeginDocument{%
374     \italiano
375     \renewcommand\@chapapp{\chaptername}%
376 }
```

At last the subsidiary and independent packages `topcoman` and `topfront` are requested for input; the former is loaded only when the `documentclass` option `noTOPfront` has NOT been declared. For using XeLaTeX as the typesetting engine it's necessary to load such packages at the "begin document" step, so that all the other settings, especially fonts, are already established. Even if the past experience has shown that loading these packages does not need to be deferred al the "begin document" level with pdfLATEX, it does not hurt to delay their loading operation with both engines. This delayed loading implies that the commands defined in `topfront` and in `topcomand` cannot be used in the preamble.

```
377 \AtBeginDocument{%
378 \unless\ifT@Pfrontespizio
379         \RequirePackage{topfront}
380     \fi
381 \RequirePackage{topcoman}%
382 }
```

Last but not least, here come the specifications for the PDF/A-1b format. First of all the patches to the macros `\not` and `\mapstochar` that produce problems with the format, because they have a declared width of 0 pt; this is no problem for `latex` or `pdflatex`, but it is a problem for the PDF/A format. Therefore these commands must be replaced by equivalent ones that do not use zero-width glyphs. For `\not`, another slash can be used, but in order to have it the right size in all math typesetting modes it is necessary to have a different command for every mode; this is achieved with the `\mathchoice` primitive as such:

```
383 \ifPDFTeX
384 \renewcommand*\not{\mathrel{\mathchoice%
385 {\rlap{$\displaystyle\mkern2.5mu\mathnormal{/}$}}%
386 {\rlap{$\textstyle\mkern2.5mu\mathnormal{/}$}}%
387 {\rlap{$\scriptstyle\mkern2.5mu\mathnormal{/}$}}%
388 {\rlap{$\scriptscriptstyle\mkern2.5mu\mathnormal{/}$}}%
389 }}
```

Actually the zero-width property of the slash must be simulated with a zero-width box but within this box it is necessary to specify the typesetting style of the math mode material.

A similar trick is used to patch the `\mapstochar` command but no other glyph was found suitable for substituting the original one; therefore we had to make it up with the `picture` environment:

```
390 \renewcommand\mapstochar{\mathrel{\mathchoice
391 {\displaystyle\unitlength=0.0555\fontdimen6\font
392     \picture(0,0)\put(1,1.5){\linethickness{0.72\unitlength}%
393     \line(0,1){6}}\endpicture}
394 {\textstyle\unitlength=0.0555\fontdimen6\font
395     \picture(0,0)\put(1,1.5){\linethickness{0.72\unitlength}%
396     \line(0,1){6}}\endpicture}
397 {\scriptstyle\unitlength=0.0555\fontdimen6\font
398     \picture(0,0)\put(1,0.55){\linethickness{0.61\unitlength}%
399     \line(0,1){5}}\endpicture}
400 {\scriptscriptstyle\unitlength=0.0555\fontdimen6\font
401     \picture(0,0)\put(1,0.2){\linethickness{0.56\unitlength}%
402     \line(0,1){4}}\endpicture}
403 }}\fi
```

The above patches are introduced irrespective of using `pdflatex` for producing a PDF/A compliant file; one might need to produce a traditional PDF file, or even a DVI file, to be converted to the PS format, in order to possibly further transform it to PDF/A by means of `ghostscript` (see the details in the `ps2pdf.html` file belonging to the `ghostscript` documentationx). The patches are avoided if XeLaTeX is being used with the UNICODE math definitions.

4.3   The `classica` option

As mentioned above, the `classica` option was devised in order to cope with theses in humanities and the specifications came from Paolo Ciacchi, a student of the

University of Trieste, who was preparing a master thesis in classical Greek philology. The necessity of the large normal size derived from the necessity of having clear mark-up signs among the myriad signs philologists use, that sometimes I believe that the philological mark-up requires typesetting environments similar to the mathematical ones: in facts the material to be typeset does not form linear sequences, as in plain text, but bi-dimensional structures as in mathematics.

The other requirements involve the title page and I agreed to implement them, since they are likely to be useful for other universities as well; the effort for localising this bundle makes this point less stringent compared to the previous versions, but there are other layout fine points that cannot be solved with a simple substitution of infix strings.

The singular and plural masculine and feminine strings for "candidate" are redefined. For other languages the configuration file comes in handy.

```
404 \ifTOPfront
405 \ifclassica
406 \def\Candidato{Laureando}
407 \def\Candidata{Laureanda}
408 \def\Candidati{Laureandi}
409 \def\Candidate{Laureande}
410 \fi\fi
```

Since theses in humanities may end up to occupy several volumes (classically "tomo" in Italian means "volume", although the latter word is valid also in Italian with the same meaning but appears to be neglected by the humanists; in English "tome" indicates a "large book"; in Italian the meaning nuance of "tomo" is a little different because it is used to indicate the volumes into which a large printed work is divided; the humanists, as usual, know well their words and use them properly!) a mechanism is set up to create a different title page for each volume; we need a volume counter and a command to start each volume. The localisation allows to change the infix string that is going to be printed in the title page. But when several tomes are involved, instead of using the \frontespizio macro or frontespizio environment, use the \tomo macro, that provides to stepping up the volume counter before actually printing the new tome title page.

```
411 \newcounter{tomo}
412 \newcommand*{\tomo}{\clearpage\stepcounter{tomo}\boolfalse{topTPTlogos}%
413     \fr@ntespizio}
```

Folios as well are to be redefined and also the page styles require a redefinition so as to being able to use old style numbers. The macro \lapagina (it's not a case the this macro name is the direct translation of "the page") contains the folio number; if it is in roman numerals nothing happens, but if the old style numbers are required the folio is surrounded by the proper macros so as to expand the folio macro before typesetting it in old style. The page style does not need an actual redefinition, because the original definition in file toptesi.sty already uses \lapagina as the typeset folio indicator.

```
414 \renewcommand*{\lapagina}{%
415     \ifnumeriromani
416         \thepage
```

```
417      \else
418          \if@ldstyle
419              \expandafter\oldstylenums\expandafter{\thepage}%
420          \else
421              \thepage
422          \fi
423      \fi
424 }
```

But we actually have to redefine the page style for the new headings because the `classica` option changes completely the left and the right headers depending on the option `autoretitolo`; in this case the left header contains the candidate's name and a short version of the thesis title, while the right heading contains the chapter (short) title; if this option is not in force, headings appear as usual. The redefinition of the `headings` page style is made only if this option is in force. In case of two side printing where the left head and the right head are different, the left heading contains the author name and the short title name; if there are other authors the first author name is printed followed by "et al."; if the over all header string exceeds the text width, a message is printed so as to induce the user to use the optional `\titolo` argument, the one that is supposed to contain the short title. In the right heading there is the chapter title; again if the header turns out to bee too wide, a message is issued to the user.

More complicated things are done when typesetting on one side; in facts the author name (possibly followed by "et al.") and the short thesis title are typeset on the left of the only header while the chapter title is typeset on the right; in the unlikely situation where these two elements separated by at least 2em of white space do not exceed the text width, they are printed, but if they do, as it is likely to happen, then my suggestion would be not to use the `autoretitolo` option; but if the user absolutely wants this layout, then the author's name, possibly followed by "et al.", and the short thesis title are set in a vertical box; the same happens for the chapter title in another box; these texts are set with ragged margins, and eventually these boxes are set one next to the other with an intermediate glob of infinitely stretchable glue, and finally set in the header box with a rule underneath the two of them. The result might be barely acceptable if both vertical boxes do not exceed two lines and no words have to be hyphenated, but in general I believe it is an ugly layout; the user is warned!

```
425 \if@utoretitolo
426     \if@twoside
427 %
428          \renewcommand*{\ps@headings}{\let\@mkboth\markboth%
429          \def\@oddfoot{\null \hfill \textbf{\lapagina} \hfill \null}%
430          \let\@evenfoot\@oddfoot
431          \def\@evenhead{%
432              \setbox\@intesta\hbox{\ifPDFTeX\latintext\fi
433              \footnotesize\strut\textsc{%
434              \@author\ifx\@secondauthor\empty\else\ et al.\fi: \@stitle}%
435              }%
436          \ifdim\wd\@intesta>\textwidth\headwrn{\titolo}\fi
```

```
437            \underline{\makebox[\textwidth]{\box\@intesta}}}%
438        \def\@oddhead{\ifPDFTeX\latintext\fi
439            \setbox\@intesta\hbox{%
440                \footnotesize\strut\textsl{\rightmark}}%
441        \ifdim\wd\@intesta>\textwidth \headWarn{\section}\fi%
442        \underline{\makebox[\textwidth]{\box\@intesta}}}%
443        \def\chaptermark##1{\markright{\thechapter\ -- ##1}}%
444        \def\sectionmark##1{}}%
445    \else
446        \renewcommand*{\ps@headings}{\let\@mkboth\markboth
447            \def\@oddfoot{\null \hfill \textbf{\lapagina}\hfill \null}%
448            \let\@evenfoot\empty\let\@evenhead\empty
449            \def\@oddhead{{\setbox\z@\hbox{\ifPDFTeX\latintext\fi\footnotesize
450            \textsc{%
451            \@author\ifx\@secondauthor\empty\else\ et al.\fi: \@stitle}}%
452            \setbox\tw@\hbox{\ifPDFTeX\latintext\fi\footnotesize\textsl{\rightmark}}%
453            \dimen@=\dimexpr2em + \wd\z@ + \wd\tw@\relax
454            \ifdim\dimen@<\textwidth \relax
455            \else
456              \setbox\z@\vbox{\hsize.48\textwidth\parindent\z@\raggedright
457              \ifPDFTeX\latintext\fi\footnotesize\textsc{%
458              \@author\ifx\@secondauthor\empty\else\ et al.\fi: \@stitle
459              }}%
460              \setbox\tw@\vbox{\hsize.48\textwidth\parindent\z@\raggedleft
461                \ifPDFTeX\latintext\fi\footnotesize\textsl{\rightmark}}%
462            \fi
463            \setbox\@intesta\vbox to\z@{%
464                \vss\hbox to\textwidth{\strut\box\z@\hfill\box\tw@}}%
465            \underline{\box\@intesta}}}%
466            \def\chaptermark##1{\markright{\thechapter\ -- ##1}}}
467    \fi
468 \fi
```

Here `\annoaccademico` is defined to typesets the infix string corresponding to "Anno accademico" followed by the year range in old style numbers (irrespective of the `oldstyle` option); localisation commands are provided so as to set a different string, possibly through the configuration file. In order to set an adequate en-dash between the old style numbers a new definition is given that takes care to set the dash at a height above the base line that copes with the specific shape of the old style numbers.

```
469 \newcommand*\NomeAnnoAccademico[1]{\gdef\AnnoAccademico{#1}}
470 \@ifundefined{AnnoAccademico}{\gdef\AnnoAccademico{Anno accademico}}{}
471 %
472 \def\annoaccademico#1{\ifclassica
473   \def\@submitdate{{\large\textsc{\AnnoAccademico}} {\Large\s@tanno#1!}}}
474   \else
475   \PackageWarning{toptesi}{\string\annoaccademico\space is usable only
476    when the\MessageBreak
477    'classica' option is in force}%
478   \def\@submitdate{\AnnoAccademico\ #1 --- Needs 'classica' option}
```

```
479   \fi}
480 \def\s@tanno#1-#2!{\oldstylenums{#1\EnDash#2}}
481
482 \def\EnDash{{\settowidth{\dimen@}{\large\scshape I}%
483 \setbox\tw@\hbox{2}\dimen\tw@.5\ht\tw@\advance\dimen\tw@-.5\dp\tw@
484 \dimen4\dimen\tw@\advance\dimen4by.0385ex\relax
485 \advance\dimen\tw@-.0385ex\relax
486 \makebox[1.5\dimen@]{%
487     \vrule\@width\dimen@\@height\dimen4\@depth-\dimen\tw@}}}
```

The footnote rule separator is also longer that the default one. Dealing with notes the footnote separator is also changed as well as the footnote skip. But the humanists like to have also unnumbered notes within numbered ones, as if they were two separate sets; of course specialised extension modules, such as, for example, the `eledmac` Package, are available on CTAN, but as a poor man solution the command `\nota` was introduced that inserts a note with a symbol as a note mark; the symbol *must* be a mathematical symbol as the dagger or the double dagger; by default it is the asterisk. At the same time the default definition of the mathematical asterisk is that of a binary operator; I have experienced that the note symbol gets a better positioning if it is used as an ordinary symbol through `\mathord`. This is why its math code is redefined. A final unusual request was to be able to put a blank unnumbered note, as a (rather wide) note separator. The command `\NoteWhiteLine` has to be used at the end of the text of the preceding note.

```
488 \renewcommand\footnoterule{%
489   \kern-6\p@
490   \hrule\@width.4\columnwidth
491   \kern5.6\p@}
492 \setlength\footnotesep{12\p@}
493 \setlength{\skip\footins}{24\p@ \@plus 4\p@ \@minus 2\p@}
494 \newcommand*\nota[1][\mathord{*}]{%
495     \xdef\@thefnmark{\ensuremath{\m@th#1}}\@footnotemark\@footnotetext
496 }
497 \newcommand*{\NoteWhiteLine}{\par\vspace*{-.3\baselineskip}}
```

The humanists asked me to create some other simple macros: one for skipping a whole page, without header and footer; another to compose a dedication page; a third one for typesetting a page with one or more witty sentences. The first request has a trivial solution, but at least `\paginavuota` is much shorter to type in that its expansion.

The environments `dedica` for the dedication, and `citazioni` for the witty sentences are almost identical; both typeset their content with a reduced text width, half of the normal one; this column is typeset on the right of the page. The `dedica` environment is supposed to be used in the front matter, while the witty sentence environment may be used anywhere. The dedication is typeset in `\Large` font size and in italics; if the author wants a different size and/or a different shape s/he must specify it at the beginning of the dedication. The `citazioni` environment typesets its material with the default font shape series and size, so the author has to specify any change s/he desires. These three commands, though,

are available irrespective of the `classica` option, so that they can be used also for theses outside the humanities fields.

```
498 \newcommand*\paginavuota{\clearpage\thispagestyle{empty}\null\clearpage}
499 %
500 \newenvironment{dedica}{\clearpage
501     \if@twoside
502         \ifodd\c@page\else\thispagestyle{empty}\null\clearpage\fi
503     \fi
504     \thispagestyle{empty}%
505     \list{}{\labelwidth\z@
506     \leftmargin.5\textwidth
507     \parindent\z@
508     \raggedright\LARGE\itshape}\item[]
509 }{%
510     \endlist\clearpage
511 }
512 %
513 \newenvironment{citazioni}{%
514     \clearpage\thispagestyle{empty}
515     \list{}{\labelwidth\z@
516     \leftmargin.5\textwidth
517     \parindent\z@
518     \raggedright}\item[]
519 }{%
520     \endlist\clearpage
521 }
```

4.4 The `topfront.sty` code

This file may input by `toptesi`, version 5.x provided that the `noTOPfront` is not specified among the `\documentclass` options, but it can be used as an independent extension module with (hopefully) any document class.

It contains all the definition for the composition of just the title page along the style requirements of `toptesi`, version 5.x. It makes use of an optional configuration file where the user can define a lot of default information and all the infix language dependent strings that are peculiar to this title page. Among other things it can typeset also the copyright page on the verso of the title page if the `\retrofrontespizio` macro argument is not empty.

This file specifies that it requires the LaTeX 2_ε format and identifies itself. Since this module package might be used to extend the performances of other classes, this package does not specify any input encoding, on the assumptions that the calling class already provided this information.

In oder to use `topfront` as a stand-alone extension package with other classes it is necessary to verify if the `classica` option corresponds to a valid setting of `\ifclassica`; since this test is defined in the `toptesi` class, here we need to verify its existence and, in case, to set its value to `false`. This implies two points: we need a powerful macro package to test a "switch", and when this `topfront`

module is used as a stand-alone one, the settings of the `classica` options are not available.

```
522 \@ifpackageloaded{etoolbox}{}{\RequirePackage{etoolbox}}
523 \ifcsundef{ifclassica}{%
524     \newif\ifclassica
525     \classicafalse
526 }{}
```

For the title page we need a special style, in order to put some information in the header and some other in the footer, without actually changing the page layout, except for centring the grid horizontally in the page. The headers, due to a specific request of Politecnico di Torino, is to have the university logo(s) in the header; other universities maintain their logo(s) in the lower part of the page as it was done all the time in the past. We need some device to switch positions to the logos, without actually changing the page layout. Since the logo(s) are sort of large, the header must smash the header contents, so as to avoid any modification of the position and size of the other parts of the page. The `TPT@logobox` box is being defined later on and the `\logosede` command takes care of filling it up. Besides these devices, the `\frontepsizio` command and the `frontespizio` environments (with or without asterisk) produce either layout depending on the state of the boolean `topTPTlogos`.

```
527 \def\headstrut{\vrule \@depth4\p@ \@height\z@ \@width\z@}
528 \def\ps@titlepage{\let\@mkboth\@gobbletwo
529     \def\@oddfoot{\vbox to 0.05\paperheight{\vss
530         \hbox to\hsize{\hfil{\Large{\@submitdate}}\hfil}}}%
531     \let\@evenfoot\@oddfoot
532     \def\@oddhead{
533         \vbox to\headheight{\vss\iftopTPTlogos
534             \hbox to\textwidth{%
535                 \headstrut\hfil
536                 \raisebox{3\baselineskip}{\usebox\TPT@logobox}\hfil\null%
537             }\fi
538             \ifcsvoid{@ateneo}{}{\vskip\smallskipamount
539             \hbox to\textwidth{\hss\LARGE\MakeUppercase{\@ateneo}\hss}}
540             \vss
541         }%
542     }%
543     \let\@evenhead\@oddhead
544     \def\chaptermark##1{}\def\sectionmark##1{}%
545 }
```

Similarly a different title page style for typesetting the logos in the lower half of the page is defined; since it is the only style usable with the `classica` option, we call it the `classica` page style:

```
546         \def\ps@classica{\let\@mkboth\markboth
547         \def\@oddhead{\vbox{%
548         \hbox to \hsize{\hfill {\LARGE\MakeUppercase{\@ateneo}}\hfill}%
549         \ifclassica
550             \hbox to \hsize{\hfil\vrule\@width\z@
```

```
551        \@height2ex\vrule\@height1.4\p@\@depth-\p@\@width50mm\hfil}%
552        \fi
553        }}%
554        \def\@oddfoot{\vbox to \dimexpr\paperheight/20\relax{\vss
555        \ifclassica
556           \hbox to \hsize{\hfil\raisebox{-.3ex}[\z@][\z@]{%
557           \vrule\@height-2.6\p@\@depth3\p@\@width\dimexpr\textwidth/3}\hfil}%
558        \fi
559        \hbox to\hsize{\hfill{\Large{\@submitdate}}\hfill}}%
560        }%
561        \let\@evenhead\@oddhead
562        \let\@evenfoot\@oddfoot
563        }%
```

The title page information depends on the type of "thesis" that is being typeset. The following commands specify the kind of information that is going to be typeset. Some boolean variables are automatically set by the commands in order to change some formatting depending on the kind of thesis. For languages that distinguish feminine from masculine adjectives or qualifications some automatic machinery is set up in order to format some infix strings in a way that copes with the singular or plural forms; in particular when there is a multitude of authors (maximum three) of different gender the adjectives or qualifications are set masculine plurals, while when there is just one author or the authors are of the same gender the adjectives or qualifications are set according to number and gender. All this is done by setting or resetting the truth value associated to the boolean variable `femminile`. The boolean variable `dottorato` controls the PhD thesis format, while the other variable `laureatriennale` controls the formatting of the bachelor's degree report. All other theses are treated as master theses, and in all cases the appropriate infix string is typeset in the title page:

```
564 \newif\iffemminile
565 \newif\ifdottorato \dottoratofalse
566 \newif\iflaureatriennale \laureatriennalefalse
```

The thesis title is specified by means of the following commands; `\monografia`, the name of the bachelors degree final report, sets also the corresponding boolean variables and redefines the command `\titolo` so as to avoid duplications and inconsistencies; of course something might still be inconsistent if the commands are given in the wrong order.

`\titolo` accepts an optional argument, the "short title", more or less as the standard sectioning commands; this is due to the fact that with the class option `autoretitolo` the thesis title is written together with the author's name in the even page headings; if the thesis full title is too long it produces overfull headlines with ugly results; a short title may solve the inconvenience. The `\sottotitolo` command is another way to maintain a short title; all the supplementary title information may be typeset in the subtitle.

```
567 \def\monografia#1{\global\laureatriennaletrue
568        \global\dottoratofalse
569        \global\def\titolo##1{\PackageWarning{topfront}%
```

```
570        {Il titolo e' gia' stato impostato con
571            il comando \string\monografia}}%
572        \gdef\@titolo{#1}}
573 \let\@stitle\empty
574 \newcommand*{\titolo}[2][]{%
575 \ifbool{laureatriennale}{%
576 \PackageWarning{topfront}{Il titolo deve essere impostato con
577                              il comando \string\monografia}
578 }
579 {\def\@tempA{#1}\ifdefempty{\@tempA}%
580        {\gdef\@stitle{#2}}
581        {\gdef\@stitle{#1}}%
582        \gdef\@titolo{#2}%
583 }}
584 \def\sottotitolo#1{\gdef\@subtitle{#1}}
```

The \materia or its alias \Materia are used to specify the subject of the thesis; as a silly example a set of commands that reflects this subtle difference might be the following:

```
\materia{Applied Tetratricotomy}
\titolo{The tetratricotomy of blond hair}
\sottotitolo{Accurate measurements of the four fourths
  of tetratricotomised blond hair}
```

and the title page, for example, will contain something like this:

<div align="center">

Master Thesis

in

Applied Tetratricotomy

The tetratricotomy of blond hair

Accurate measurements of the four fourths
of tetratricotomised blond hair

</div>

```
585 \let\@materia\empty
586 \def\Materia#1{\def\@materia{#1}}\let\materia\Materia
```

Things get more complicated for doctoral theses; in general there is no supervisor; at most if a professor is assigned to supervise or control the PhD student's work this may be called in whatever mode but here we assume his name is input with the command \tutore even if "tutor" does not appear as the best choice; in any case in Italian "tutore" does not have the same meaning as the English "tutor". Most Doctoral Schools require to name the School's director or coordinator instead of the tutor. This is why this person's name can be introduced with \direttore or \coordinatore; the actual label printed over this person name is "Direttore" or "Coordinatore" but it can be changed with \QualificaDirettore.

```
587 \newif\ifDirettore \Direttorefalse
588 \def\tutore#1{\gdef\@tutore{#1}}
589 \def\direttore{\Direttoretrue\relatore}%
590 \def\coordinatore{\Direttorefalse\relatore}%
591 \def\QualificaDirettore#1{\gdef\@PhDdirector{#1}}%
```

For "normal" theses we may have from one to three supervisors and from one to three authors; not all universities accept a multitude of supervisors and/or authors of the same thesis, but some do; this is why this bundle accepts up to three names for each category. The `\second...` commands set the plural forms of the labels printed above the name lists. For the candidates there are different commands to input ladies or gentlemen names; according to the masculine (ending in 'o') or feminine (ending in 'a') commands, the appropriate truth values are assigned to the boolean variable `femminile` and the labels are set accordingly.

Notice that in the case of bachelor degree final report no supervisor name is printed even if one or more supervisor names are specified. This must be kept in mind in order to avoid surprises in finding missing information in the title page. Further on, there are suggestions for circumventing this fact.

```
592 \def\relatore#1{\gdef\@principaladviser{#1}}
593 \def\secondorelatore#1{\gdef\@secondadviser{#1}}
594 \def\terzorelatore#1{\gdef\@thirdadviser{#1}}
595 \def\candidato#1{\gdef\@author{#1}\femminilefalse}
596 \def\candidata#1{\gdef\@author{#1}\femminiletrue}
597 \def\secondocandidato#1{\gdef\@secondauthor{#1}\femminilefalse}
598 \def\secondacandidata#1{\gdef\@secondauthor{#1}}
599 \def\terzocandidato#1{\gdef\@thirdauthor{#1}\femminilefalse}
600 \def\terzacandidata#1{\gdef\@thirdauthor{#1}}
```

The next set of macros is used to typeset the "date" of the thesis defence or presentation or whatever is done for the final exam. The macro is sort of complicated because the input format for this "date" may vary from a single year, to a year range, to a month and year specification so that different actions must be taken; if the option `classica` is in force then the formatting of the "date" may be still different. This command is aliased with `\esamendidottorato` which literally means "defence of the doctoral dissertation"; nevertheless both commands refer to a simple date in one of those formats.

`\getseduta` splits the date into its two components, month and year; if the argument is a single string without intervening spaces, the first one is the string itself and the second is empty; this emptiness may be tested and in case the date formatting is modified accordingly. In particular if the string is a single spaceless one, this string is assigned to `\@submitdate`; otherwise a different treatment is made according to the fact that `classica` is in force; if `classica` is not in force the total string, including spaces, is assigned to `\@submitade`. If `classica` is in force, `\s@dutaclassica` is called with the whole string. On turn `\s@dutaclassica` verifies if the date should be typeset with old style numbers or in the usual way; in the latter case the whole string is assigned to `\@submitdate`; in the former one the year part may be either a single year or a year range; this separation is tested by splitting the year part across one dash; if the dash is present the extremes

of the year range are assigned to `\1` and `\2`, otherwise the year part is a single year. If a single year is given this is simply typeset with old style numbers and the appropriate commands are assigned to the `\@submitdate` control sequence. If a year range is given, this year range is also typeset with old style numbers, and the dash is executed with a regular en-dash surrounded with white space.

```
601 \def\sedutadilaurea#1{\getseduta#1 !}
602 \def\getseduta#1 #2!{%
603     \def\@tempA{#2}%
604     \ifx\@tempA\empty
605         \def\@submitdate{#1}%
606     \else
607         \unless\ifclassica
608             \def\@submitdate{#1 #2}%
609         \else
610             \s@dutaclassica#1 #2!%
611         \fi
612     \fi
613 }%
614 \def\s@dutaclassica#1 #2!{%
615     \if@ldstyle
616         \s@paranumeri#2-!%
617         \ifx\2\empty
618             \edef\@submitdate{\noexpand#1 \noexpand\oldstylenums{#2}}%
619         \else
620             \s@paranumeri#2!%
621             \edef\@submitdate{\noexpand#1
622             \noexpand\oldstylenums{\1} -- \noexpand\oldstylenums{\2}}%
623         \fi
624         \let\1\undefined
625         \let\2\undefined
626     \else
627         \def\@submitdate{#1 #2}%
628     \fi
629 }
630 \def\s@paranumeri#1-#2!{\def\1{#1}\def\2{#2}}%
631 \let\esamedidottorato\sedutadilaurea
```

The next macros are used to assign strings to some literal information to be typeset in the title page. `\ciclodidottorato` requires an uppercase roman numeral (in Italy), but it can accept anything that can precede the infix word "cycle". Macros `\corsodilaurea` and `\corsodidottorato` specify the degree course qualification; You would specify just "Elettronica", for example, and the module will write in the title page "Corso di Laurea in Elettronica". The infix part may be changed depending on the default language and the configuration file.

`\scuoladidottorato` gets the name of the PhD School; `\ateneo` gets the generic name of the university; `\nomeateneo` gets the proper name of the university. In Italy Universities are generally named after the city they are in; in large cities where there are several universities, each one of them has a proper name. For example the generic name might be "Università di Roma" and the

proper name might be "La Sapienza". \facolta may receive an optional argument that is the uppercase roman numeral specific of the faculty and a compulsory argument that corresponds to the type of faculty; for example, with \facolta[II]{Ingegneria} the package typesets in the title page "II Facoltà di Ingengeria"; if the optional argument is not specified, no roman numeral is typeset; the infix string Facolt\'a␣di␣ may be changed with the configuration file depending on the default language. It might be necessary to define another name in place of "Facoltà di␣", since with recent bills, the administrative structure of all Italian universities has been changed and the activities formally assigned to the Faculties may be now the responsibility of other structures that may have different names in different universities. If the internal command defined by \FacoltaDi is empty, no name is printed at all and the title page will not have any indication of a particular faculty or other educational structure. Therefore the Degree Course name, specified with \corsodilaurea should be always specified. The alias commands \StrutturaDidattica and \struttura are defined as equivalent commands to \FacoltaDi and \facolta respectively.

Finally \logosede gets the name of the graphic file that contains the information relative to the university logo; it may receive also a comma separated list of logo file names, as it might be necessary when a thesis is developed in a multiple university environment. If such logo file is not available, the user should not specify this command; if the thesis is typeset on smaller paper size than A4 or letter, it would be much better to avoid inserting one or more logos in the title page; this is particularly important when using A5 paper size. Nevertheless this decision is left to the user and this package neither controls this fact nor outputs any warning. If the user uses this command to insert one or more university logo files but some file is not available, the usual graphicx package warning is issued but compilation may go on without the missing logo.

The treatment of one or more logo files requires some extra commands and control sequences suitable to store temporary data or to specify style parameters: one is the name of a save box; another is the default spacing between the logos in the typeset page; the third is the height of the logos. The default spacing may be set with the help of the macro \setlogodistance – notice that the default value is 3em, and if a different distance is desired, it should not be much larger or much smaller than the default one; the default logo height is specified as an optional command to the \logosede command, while the default size is given by the \interno length. This length is specified in the main toptesi package in order to compute the typesetting grid, so that if the topfront module is used by itself with other classes, the existence of the \inteno control sequence is tested and if it is undefined, then and only then it is defined in this module and assigned a default value. The save box name is just for internal workings and does not require any customisation. The analysis and processing to the possible list of logo file names is done through the usual means of the delimited argument extraction of the single names from the list; the "string" of logos is then composed in a save box; as the list has been completely processed, the box is measured; if its width is shorter than the \textwidth it is typeset without further processing; if it's larger, on the opposite, the box gets scaled down so that its width equals the \textwidth.

The `\tutoreaziendale` macro is the last title page addition; several students, who work on their thesis or final project in a company, want to have the company supervisor name printed in the title page; this does not preclude expressing the student's deepest thanks in the acknowledgements section, but it does not harm to name this person also in the title page.

Eventually the `\retrofrontespizio` command, that by default is empty, allows to typeset a copyright page; the argument of this command is in total responsibility to the user who must write it in the thesis main language; the user can specify from zero to several paragraphs, separated by the vertical spaces s/he thinks best; the argument by default is typeset at the bottom of the text block of the copyright page. The user can specify any pertinent space at the bottom of his/her argument, so as to set the text in the position s/he likes best. In order to handle the copyright page in the proper way we need to test if its definition is empty or blank so we need the powerful advanced macros of the package `etoolbox` that has already been loaded by this module or by the main `toptesi` one.

```
632 \def\ciclodidottorato#1{\gdef\@ciclo{#1 \@cyclename}}%
633 \def\corsodilaurea#1{\global\dottoratofalse\gdef\@corso{#1}}
634 \def\corsodidottorato#1{\global\dottoratotrue\global\laureatriennalefalse
635     \gdef\@corso{#1}}
636 \def\scuoladidottorato#1{\global\dottoratotrue\global\laureatriennalefalse
637     \gdef\@phdschool{#1}}
638 \def\ateneo#1{\gdef\@ateneo{#1}}
639 \def\nomeateneo#1{\gdef\@nomeat{\expandafter\uppercase{\expandafter #1}}}
640 \newcommand\facolta[2][]{\gdef\@facname{#2}\gdef\@facnumber{#1}}
641 \let\struttura\facolta
642
643 \newlength{\TPT@logospace}\TPT@logospace=3em\relax
644 \newsavebox{\TPT@logobox}
645 \newdimen\TPT@logoheight
646 \newcommand*\setlogodistance[1]{\TPT@logospace=#1}
647 \providecommand*{\@logosede}{}
648
649 \ifcsundef{interno}{%\
650     \newlength\interno
651     \setlength\interno{\dimexpr\paperwidth/7}}{}
652
653 \newcommand\logosede[2][\interno]{\def\@logosede{#2}\TPT@logoheight=#1\relax
654 \ifcsvoid{@logosede}{\sbox{\TPT@logobox}{}}{\begin{lrbox}{\TPT@logobox}%
655 \expandafter\fillup@TCP@logobox\@logosede,!}}
656
657 \def\fillup@TCP@logobox#1,#2!{%
658 \ifblank{#1}{\end{lrbox}\ifdim\wd\TPT@logobox>\textwidth
659 \sbox\TPT@logobox{\resizebox{\textwidth}{!}{\box\TPT@logobox}}\fi}%
660 {\def\@logosede{#2}%
661 \includegraphics[height=\TPT@logoheight]{#1}\hskip\TPT@logospace
662     \expandafter\fillup@TCP@logobox\@logosede,!}}
663
664 \newcommand\printloghi{\unless\ifvoid\TPT@logobox\usebox{\TPT@logobox}\fi}
```

```
666 \def\tutoreaziendale#1{\gdef\@tutoreaziendale{#1}}
667 \newcommand\retrofrontespizio[1]{\long\gdef\@retrofrontespizio{#1}}
```

The following commands are user commands that modify the infix strings according to the language used and to the specifications of the actual university. All these commands can be put in the configuration file so as to specify what is desired as a default. If these commands are specifically used to redefine something before issuing the `\frontespizio` command, the command that actually typesets the title page, the new definitions override the configuration ones.

`\FacoltaDi` sets or changes the string "Facoltà di" in, say, "Faculty of"

`\DottoratoIn` sets or changes the string "Dottorato in" in, say, "PhD in"

`\CorsoDiLaureaIn` sets or changes the string "Corso di Laurea in" in, say, "Master of Science in"

`\TesiDiLaurea` sets or changes the string "Tesi di Laurea" in, say, "Tesi di Laurea Magistrale"

`\NomeMonografia` sets or changes the string "Monografia di Laurea" in, say, "Tesi di Laurea"

`\NomeDissertazione` sets or changes the string "Dissertazione" in, say, "PhD dissertation"

`\InName` sets or changes the string "in" in, say, "auf"

`\CandidateName` sets or changes the string "Candidato" in, say, "Laureando"

`\AdvisorName` sets or changes the string "Relatore" in, say, "Supervisors"

`\CoAdvisorName` sets or changes the string "Correlatore" in, say, "Corapporteur"

`\NomeTutoreAziendale` sets or changes the string "Supervisore aziendale" in, say, "XYZ Company Supervisor"

`\TutorName` sets or changes the string "Tutore" in, say, "Supervisor"

`\CycleName` sets or changes the string "ciclo" in, say, "cycle"

`\NomePrimoTomo` sets or changes the string "Tomo primo" in, say, "First volume"

`\NomeSecondoTomo` sets or changes the string "Tomo secondo" in, say, "Second volume"

`\NomeTerzoTomo` sets or changes the string "Tomo terzo" in, say, "Third volume"

`\NomeQuartoTomo` sets or changes the string "Tomo quarto" in, say, "Fourth volume"

In the above description the first string is generally the default one, while the second string is just an example of the corresponding string to be set in another language or to be changed in Italian. The last four commands clearly show the difficulty of localising language strings: it is necessary to localise the whole phrase, because of the position of the adjectives.

```
668 \newcommand\FacoltaDi[1]{\gdef\@faculty{#1}}
669 \let\StrutturaDidattica\FacoltaDi
670 \newcommand\DottoratoIn[1]{\gdef\@PhDname{#1}}
671 \newcommand\CorsoDiLaureaIn[1]{\gdef\@laureaname{#1}}
672 \newcommand\TesiDiLaurea[1]{\gdef\@TesiDiLaurea{#1}}
673 \newcommand\NomeMonografia[1]{\gdef\@monografia{#1}}
674 \newcommand\NomeDissertazione[1]{\gdef\@dissertazione{#1}}
675 \newcommand\InName[1]{\gdef\@InName{#1}}
676 \newcommand\CandidateName[1]{\gdef\@nomecandidato{#1}}
677 \newcommand\AdvisorName[1]{\gdef\Relatore{#1}\gdef\Relatori{#1}}
678 \newcommand\CoAdvisorName[1]{\gdef\Correlatore{#1}\gdef\Correlatori{#1}}
679 \newcommand\TutorName[1]{\gdef\Tutore{#1}}
680 \newcommand\NomeTutoreAziendale[1]{\gdef\@tutoreaziendalename{#1}}
681 \newcommand\CycleName[1]{\gdef\@cyclename{#1}}
682 \newcommand\NomePrimoTomo[1]{\gdef\PrimoTomo{#1}}
683 \newcommand\NomeSecondoTomo[1]{\gdef\SecondoTomo{#1}}
684 \newcommand\NomeTerzoTomo[1]{\gdef\TerzoTomo{#1}}
685 \newcommand\NomeQuartoTomo[1]{\gdef\QuartoTomo{#1}}
686 \newcommand\IDN{\\\quad matricola:\space}
```

Now we can read the configuration file if it exists; in any case what is possibly defined or redefined in the configuration file must not be redefined in the following Lines and this is why everything is subject to the test `\@ifundefined`. Most default definitions are simply "blank"; the others are in Italian. All of them, except the supervisor and candidate strings may be individually redefined in the configuration file or in the preamble. Those that cannot be redefined such as the four "candidate" strings may be actually redefined through the single `\CandidateName` that should be used in a language depended way and with the correct number and gender once for all. The four endings in the Italian strings allow to exercise the correct selection only for Italian; a specific test is made inside the `\frontespizio` command; because of this, the same machinery cannot be used, say, for French but maybe in the future this feature is resolved in a proper way. The same is true for the supervisor and the co-supervisor strings that may be changed once for all with `\AdvisorName` and `\CoAdvisorName`.

```
687 \IfFileExists{\jobname.cfg}{\input{\jobname.cfg}}%
688     {\IfFileExists{toptesi.cfg}{\input{toptesi.cfg}}{}}
689 %
690 \@ifundefined{@cyclename}{\def\@cyclename{ciclo}}{}
691 \@ifundefined{@titolo}{\def\@titolo{}}{}
692 \@ifundefined{@author}{\def\@author{}}{}
693 \@ifundefined{@principaladviser}{\def\@principaladviser{}}{}
694 \@ifundefined{@secondadviser}{\def\@secondadviser{}}{}
695 \@ifundefined{@thirdadviser}{\def\@thirdadviser{}}{}
```

```
696 \ifcsundef{@PhDdirector}{%
697   \ifDirettore\def\@PhDdirector{Direttore del corso di dottorato}\else
698     \def\@PhDdirector{Coordinatore del corso di dottorato}\fi}{}
699 \@ifundefined{@tutore}{\def\@tutore{}}{}
700 \@ifundefined{@secondauthor}{\def\@secondauthor{}}{}
701 \@ifundefined{@thirdauthor}{\def\@thirdauthor{}}{}
702 %
703 \@ifundefined{@nomerelatore}{\def\@nomerelatore{}}{}
704 \@ifundefined{@nomecandidato}{\def\@nomecandidato{}}{}
705 \@ifundefined{Candidato}{\def\Candidato{Candidato}}{}
706 \@ifundefined{Candidata}{\def\Candidata{Candidata}}{}
707 \@ifundefined{Candidati}{\def\Candidati{Candidati}}{}
708 \@ifundefined{Candidate}{\def\Candidate{Candidate}}{}
709 \@ifundefined{Relatore}{\def\Relatore{Relatore}}{}
710 \@ifundefined{Relatori}{\def\Relatori{Relatori}}{}
711 \@ifundefined{Correlatore}{\def\Correlatore{Correlatore}}{}
712 \@ifundefined{Correlatori}{\def\Correlatori{Correlatori}}{}
713 \@ifundefined{Tutore}{\def\Tutore{Tutore}}{}
714 \@ifundefined{@tutoreaziendale}{\def\@tutoreaziendale{}}{}
715 \@ifundefined{@tutoreaziendalename}%
716         {\def\@tutoreaziendalename{Supervisore Aziendale}}{}
717 \@ifundefined{@retrofrontespizio}{\def\@retrofrontespizio{}}{}
718 \@ifundefined{@subtitle}{\def\@subtitle{}}{}
719 %
720 \@ifundefined{@corso}{\def\@corso{}}{}
721 \@ifundefined{@ciclo}{\def\@ciclo{}}{}
722 \@ifundefined{@ateneo}{\def\@ateneo{POLITECNICO DI TORINO}}{}
723 \@ifundefined{@nomeat}{\def\@nomeat{}}{}% Nome proprio dell'ateneo
724 \@ifundefined{@facolta}{\def\@facname{}}{}
725 \@ifundefined{@facnumber}{\def\@facnumber{}}{}
726 \@ifundefined{@faculty}{\def\@faculty{}}{}
727 %
728 \@ifundefined{PrimoTomo}{\def\PrimoTomo{Tomo primo}}{}
729 \@ifundefined{SecondoTomo}{\def\SecondoTomo{Tomo secondo}}{}
730 \@ifundefined{TerzoTomo}{\def\TerzoTomo{Tomo terzo}}{}
731 \@ifundefined{QuartoTomo}{\def\QuartoTomo{Tomo quarto}}{}
```

If the final exam date is not given the default value is the current month and the
current year typeset in Italian; therefore the user is strongly requested to enter
a date either with the \sedutadilaurea or the \esamedidottorato commands.
The university logo command has already been defined empty as its default value.

```
732 \@ifundefined{@submitdate}{\def\@submitdate{\ifcase\the\month\or%
733   Gennaio\or Febbraio\or Marzo\or Aprile\or Maggio\or Giugno\or
734   Luglio\or Agosto\or Settembre\or Ottobre\or Novembre\or Dicembre\fi
735   \space \the\year}}{}
736 %
737 \@ifundefined{@TesiDiLaurea}{\def\@TesiDiLaurea{Tesi di Laurea}}{}
738 \@ifundefined{@phdschool}{\def\@phdschool{SCUOLA DI DOTTORATO}}{}
739 \@ifundefined{@PhDname}{\def\@PhDname{Dottorato in }}{}
740 \@ifundefined{@laureaname}{\def\@laureaname{Corso di Laurea in }}{}
```

```
741 \@ifundefined{@dissertazione}{\def\@dissertazione{Tesi di Dottorato}}{}
742 \@ifundefined{@monografia}{\def\@monografia{Monografia di Laurea}}{}
743 \@ifundefined{@InName}{\def\@InName{in}}{}
```

Finally we have the real macro `\frontespizio` and the corresponding environments, the real macro or environments that actually typesets the title page.

I recommend to use the environments, a new feature of version 5.85. But the legacy command `\frontespizio` is still usable. The principle on which these environments work is that the `frontespizio` environment typesets the title page with logo(s) set in the page header, while the `frontspizio*` environment typesets the logos after the information on the title, the possible sub title and tome information, that is in the lower half of the title page. In order to achieve this result each environment sets the boolean variable `topTPTlogos` to either value `true` (for header logos) or `false` otherwise. The key of the different typesetting style is this boolean-variable state.

Now, since the internal `frontespizio` environment opening command is `\frontespizio` how is it possible to distinguish this opening statement from the homonymous user command? The solution is a little tricky, but after all very simple. the `\begin` command, with which an environment is started, before calling the opening statement defines the internal service macro `\@currenvir` to contain the environment name; this is used by the `\end` statement to control that it is closing the last opened environment. If the `\frontespizio` command is directly used, the `\@currenvir` macro does not contain the name frontespizio; therefore if in the opening environment definition we check the contents of `\@currenvir` against the string `frontespizio` we can decide if the user resorted to the legacy command, or the environment was correctly opened by means of `\begin`. If the user resorted to the legacy command, the service macro `\fr@ontespizio` is called that typesets the title page according to the current status of the boolean `topTPTlogos`, a boolean that has a default value but the user can set at its will. Otherwise the `frontespizio` environment is regularly executed. Notice that the service macro `\fr@ntespizio` tests the state of the boolean `classica` and accordingly uses a different page style.

```
744 \newbool{topTPTlogos} \booltrue{topTPTlogos}
745
746 \newenvironment{frontespizio*}{\boolfalse{topTPTlogos}}{\fr@ntespizio}
747
748 \newenvironment{frontespizio}{%
749     \ifdefstring{\@currenvir}{frontespizio}
750            {\booltrue{topTPTlogos}}{\TPTmaybestar}
751 }{%
752     \fr@ntespizio
753 }
754
```

With the new boolean `AteneoInHead` we can mark these situations where the university common name goes into header; with the `classica` option the title page *must* have this name in the header and if the user forgets to specify one, this module fakes it with a clear message that the university name has been forgotten,

but at the same time this fake message fills up the header position. On the other hand, when the `classica` option has not been specified, the user can use either an empty university name or a specific name. So only when the `\@ateneo` macro remains empty the university name in the header remains really blank; this is good when the university name is part of the university logo (this is the case for Politecnico di Torino, and for many other universities). When no university name is set in the header some little attention in formatting the title page is necessary.

```
755 \newbool{AteneoInHead}\boolfalse{AteneoInHead}
```

In order to use the command `\frontespizio*` as en isolated command instead of the starting command of the `frontespizio*` environment, we have to behave as with the `\frontespizio` command, but we must test for a possible asterisk following the command; for this reason we defined the `\TPTmaybestar` that absorbs une following token: if the token is an asterisk, we set the appropriate settings for the previous behaviour of the isolated command, but if it is not an asterisk we must set it back into the list after finishing the execution of the `\fr@ntespizio` service macro, whose function is to set the title page information at the proper position, but must not contain any spurious material.

```
756 \newcommand\TPTmaybestar[1]{\def\@tempA{#1}%
757 \ifdefstring{\@tempA}{*}%
758   {\boolfalse{topTPTlogos}\booltrue{AteneoInHead}\fr@ntespizio}
759   {\booltrue{topTPTlogos}\fr@ntespizio\@tempA}
760 }
```

We start defining the complex macro `\fr@ntespizio`. We start with a group so that any settings performed by this command remain local; if the title page environments had been used this group would be useless, but if the isolated commands are used, then this group protects the rest of the document from unusual local settings valid only for the title page.

```
761 \def\fr@ntespizio{%
762 \begingroup\par
```

We want also the title page to be set in the middle of the page irrespective of the binding correction; so we assign the average of the two side margins to both of them.

```
763   \oddsidemargin=\dimexpr(\oddsidemargin+\evensidemargin)/2\relax
764   \evensidemargin \oddsidemargin
```

The `\null` command inserts a void horizontal box into the vertical list; it is useful to act as a block against which the vertical glue pushes for setting the subsequent material. The normal font is chosen in case preceding commands did change the font characteristics.

```
765 \null
766 \setcounter{page}{1}%
767 \normalfont
```

Depending on the style of the title page a different `\pagestyle` is set with appropriate switches and settings. If with the classica style a university name is

blank, it is set to an explicit string that the setting of the university name has been forgotten.

```
768 \ifclassica
769     \boolfalse{topTPTlogos}
770     \thispagestyle{classica}
771     \ifcsvoid{@ateneo}{\def\@ateneo{Manca il nome dell'ateneo}
772     }{}
773 \else
774     \thispagestyle{titlepage}
775 \fi
776 \ifcsvoid{@ateneo}{}{\booltrue{AteneoInHead}}
```

The generic university name should already be in the header either in the logo or in the header text; but in spite of this we test if the university generic name macro is void, if it contains something, then we typeset also the generic name; some candidates might obey to university regulations that require the name of the university be at the top, just under the logo. The switch `\ifcsvoid` is true if `@ateneo` is empty or blank, false otherwise; but even with page top logos, no test is made in order to give the possibility to repeat the university name. It is the user responsibility to set an empty value to the `\ateneo` macro so as to avoid repeating the university name possibly already present in the logo itself.

```
777 \ifcsvoid{@ateneo}{%
778     \ifbool{topTPTlogos}
779     {}{\booltrue{AteneoInHead}\def\@ateneo{Manca il nome dell'ateneo}}%
780 }{%
781     \booltrue{AteneoInHead}%
782 }
783
784     \ifbool{AteneoInHead}{}{%
785         {{\centering\LARGE \@ateneo\par}}
786     }
```

If it is non blank the first thing we set on the page is the university proper name and some vertical glue.

```
787     \ifcsvoid{@nomeat}{}
788         {\ifbool{topTPTlogos}{\vspace*{3.5ex}}{\vspace*{-3ex}}%
789         {\centering\Large \@nomeat\par}\vfill}
790
```

Then the faculty name comes next; but for the doctoral school it uses the doctoral school name entered with `\scuoladidottorato`, otherwise it inserts the faculty ordinal number or prefix and name already entered with the optional and required arguments of `\facolta`.

```
791         \begin{center}
792         {\rmfamily\mdseries
793         \ifdottorato
794             \large \@phdschool\par\medskip
795         \else
796             \ifcsvoid{@faculty}{}{%
797             \LARGE\ifx\@facnumber\empty\else\@facnumber\space\fi
```

48

```
798              \@faculty\@facname\par\medskip
799            }
800        \fi
801      }%
```

Further specification: it inserts the field of the PhD research or the degree course name; it inserts a line such as, for example, "Philosophy Degree in Applied Tetratricotomy – XVI cycle" or "Master of Science in Applied Tetratricotomy".

```
802      \ifcsvoid{@corso}{}{{\large
803            \ifdottorato
804                  \@PhDname\@corso\ifx\@ciclo\empty\else~--~\@ciclo\fi
805            \else
806                  \@laureaname\@corso
807            \fi
808            \par}}
809      \end{center}
```

It now centres the name of the report, be it "Doctoral Dissertation" or "Master Thesis" or whatever; in case the command \materia was used, it then centres the discipline which the thesis deals with.

```
810      \vspace{\stretch{0.2}}
811      \begin{center}
812        \LARGE
813        \ifdottorato
814            \@dissertazione%
815        \else
816            \iflaureatriennale
817                  \@monografia%
818            \else
819                  \@TesiDiLaurea%
820            \fi
821        \fi
822        \unless\ifx\empty\@materia
823            \\\@InName\\\@materia
824        \fi
825      \end{center}
```

Next comes the real title entered with \titolo or \monografia and the possible subtitle.

```
826      \vspace{\stretch{0.2}}
827      \begin{center}
828        {\huge\bfseries \baselineskip=0.95em plus 1pt
829              \@titolo \par}
830      \end{center}
```

con l'eventuale sottotitolo:

```
831      \unless\ifx\@subtitle\empty
832            \begin{center}%
833              \large\textrm{\@subtitle}\par
834            \end{center}%
835      \fi
```

If the option `classica` is in force the thesis might be divided in several volumes; theses in humanities apparently are often oversized. In this case the `\tomo` command may be given at the beginning of every volume and the counter `tomo` is stepped up; the volume number is therefore printed in each title page; the infix string may be redefined as it was shown above.

```
836        \ifclassica
837            \ifnum\value{tomo}>\z@
838                \par\bigskip
839                \noindent\makebox[\textwidth]{%
840                \large\textbf{%
841                \ifcase\c@tomo%
842                    \or \PrimoTomo%
843                    \or \SecondoTomo%
844                    \or \TerzoTomo%
845                    \or \QuartoTomo%
846                    \else
847                        \PackageWarning{toptesi}{%
848                        Counter tomo equals \the\c@tomo\MessageBreak
849                        We never considered a thesis might get
850                        divided in more than four volumes}%
851                    \fi}}%
852            \fi
853            \vspace{1em}
854        \fi
855        \par
```

Going down in the title page, now comes the optional insertion of the university logo(s); "optional" in both meanings: one or more university logos are not generally required in a thesis, and in case it depends if the logo(s) have to be put in the header or here. This is a simple task since the `\logosede` already defined the contents to the box `\TPT@logobox`, and this was done either with an explicit command `\logosede` with its argument in the preamble, or with a specific line in the configuration file or within the `frontespizio` environment. If such box `\TPT@logobox` is void, the already defined `\printloghi` macro does not do anything.

```
856        \unless\iftopTPTlogos
857        {\vfill\centering \printloghi\par}\fi
```

The final task is to typeset the possible supervisors' names, the candidates' names and all the rest of the bureaucratical terms. We have to distinguish between a bachelor degree report that is not supposed to have a supervisor, from the doctoral dissertation where we do not indicate the supervisor, but the School Director, and the master thesis where there might be one or more supervisors; with the `classica` option in force no label is printed over the supervisor's name, unless there is a plurality of supervisors.

```
858        \vfill
859        \iflaureatriennale
860            \let\@nomerelatore\empty
861        \else
```

```
862            \ifdottorato
863                    \edef\@nomerelatore{\@PhDdirector}%
864            \else
865                    \ifcsvoid{@principaladviser}{}{%
866                            \def\@nomerelatore{\Relatore}}
867                    \unless\ifclassica
868                            \ifcsvoid{@secondadviser}{}{%
869                                    \def\@nomerelatore{\Relatori}}%
870                    \fi
871            \fi
872      \fi
```

Similarly the label names for the exam candidates are chosen; in Italian such
names are infix strings that are selected according the gender and the number;
if these labels have to be set in a different language it is necessary to define one
string that has to be selected by the user according to number and gender. The
label for the PhD candidate is left empty.

```
873            \ifdottorato
874                    \let\@nomecandidato\empty
875            \else
876              \iflanguage{italian}{%
877                    \iffemminile
878                            \def\@nomecandidato{\Candidata}%
879                    \else
880                            \def\@nomecandidato{\Candidato}%
881                    \fi
882                    \ifcsvoid{@secondauthor}{}{%
883                            \iffemminile
884                                    \def\@nomecandidato{\Candidate}%
885                            \else
886                                    \def\@nomecandidato{\Candidati}%
887                            \fi}
888              }{}%
889      \fi
```

For the supervisor(s) and of the candidate(s) name(s) a different approach is
used for each one of the three categories of theses. If a bachelor degree report is
dealt with, the name of the *single* candidate is centred and written in caps-and-
small-caps.

```
890          \iflaureatriennale
891                  \begin{center}%
892                          \large\mdseries\textsc{\@author}
893                  \end{center}%
```

For doctoral and master theses two virtual boxes (actually macros) are filled up
so as to align the supervisor name(s) and, in a second virtual box, the candidate
name(s). The label is set in the first line with proper number and gender; in
the second line the first name, and in the subsequent lines, if there are any, the
other names. These virtual boxes actually contain a tabular environment each;
these environments shall be actually typeset when these virtual box macros are

executed. If the `classica` option is in force no label is set over the principal advisor name, but a label is set over the co-advisor name(s). The type size is also a little different for the `classica` option. The `\protect` command is used to protect the names in case they contain accent macros that might be expanded at the wrong moment. The candidate name(s) are typeset in another nested tabular environment of two lines when the user typesets the entry in such a way:

`\candidato{Mario Rossi \IDN 123456}`

where if just the name is used or the command `\IDN` is omitted but the the ID number is entered, the whole string is written in one line without any label before e the ID number; if, on the other side, the `\IDN` command is redefined as not to produce a new line and the ID number is entered, the latter is typeset in a new line after the candidate's name. By default the `\IDN` macro is defined as:

`\newcommand{\IDN}{\\\quad matricola:\space}`

but with `\renewcommand` the user can redefine it as s/he prefers, in particular by changing the Italian name into a proper English one, may be simply 'ID'. It is necessary to recall that the ID number must be entered only if required by the University regulations; it must not be typeset in the title page just because it is possible to do it. If no number is entered, the `\IDN` must not be used.

```
894        \else
895 % For theses of any kind that expect the supervisor and co-supervisor names...
896          \def\BoxRelatori{%
897             \begin{tabular}[t]{l}%
898             \hbox{\ifclassica\else\large\fi
899                  \textbf{\protect\@nomerelatore}}\\[.6ex]
900             \hbox{\large\textrm{\protect\@principaladviser}}%
901             \ifx\@secondadviser\empty \else
902                \ifclassica
903                   \ifx\@thirdadviser\empty
904                      \ifx\@secondadviser\empty\else
905                           \\[1.5ex]\textbf{\Correlatore:}%
906                      \fi
907                   \else
908                      \\[1.5ex]\textbf{\Correlatori:}%
909                   \fi
910                \fi
911                \\[.6ex]\hbox{{\large\textrm{\protect\@secondadviser}}}%
912             \fi
913             \ifx\@thirdadviser\empty \else
914                \\[.6ex] \hbox{{\large\textrm{\protect\@thirdadviser}}}%
915             \fi
916             \end{tabular}%
917          }%
```

A similar approach is taken for the candidate name(s), although for code clarity I prefer to define two secondary macros in order to format the other candidates names and ID in a clearer way.

```
918            \def\print@secondocandidato{\\\relax
919                    \hbox{\large\tabular{@{}l@{}}\@secondauthor\endtabular}}
920            \def\print@terzocandidato{\\\relax
921                    \hbox{\large\tabular{@{}l@{}}\@thirdauthor\endtabular}}
922            \def\BoxCandidati{%
923               \begin{tabular}[t]{l}%
924                  \hbox{\unless\ifclassica\large\fi
925                     \textbf{\protect\@nomecandidato}}\\[.6ex]
926                     \hbox{\large\tabular{@{}l@{}}\@author\endtabular}%
927                     \ifcsvoid{@secondauthor}{}{\print@secondocandidato}%
928                     \ifcsvoid{@thirdauthor}{}{\print@terzocandidato}%
929               \end{tabular}%
930            }%
```

The real typesetting of these name lists takes place now; if the thesis is referred
to the PhD school one type of layout is used, otherwise the default master thesis
layout is used; remember that the bachelor degree case has already taken place. For
the doctoral dissertation the doctoral candidate name is typeset centred in one line
by itself and everything else is set 3em below into a three column table extended
to the \hsize, the first line containing the applicable labels and the second line
containing the true names. The central column is used just for spacing, but it
does not contain anything.

```
931            \ifdottorato
932               \begin{center}\large
933                  \textbf{\@author}\\[3em]
934                  {\normalsize
935                    \begin {tabular*}{\hsize}{@{\extracolsep{\fill}}ccc}
936                       \ifcsvoid{@tutore}{}{\textbf{\Tutore}}
937                    &\relax&
938                       \ifcsvoid{@principaladviser}{}{\textbf{\@nomerelatore}}
939                    \\
940                       \ifcsvoid{@tutore}{}{\@tutore}
941                    &\relax&
942                       \ifcsvoid{@principaladviser}{}{\@principaladviser}
943                    \end{tabular*}
944                  }%
945               \end{center}
946            \else
```

For the master thesis the two virtual boxes are set one besides the other but skewed
to the right or, respectively, to the left of every name so that there is enough space
for the signature. With the classica option in force the two boxes are simply
aligned.

```
947                  \unless\ifclassica
948                     \begin{flushleft}%
949                         \BoxRelatori
950                     \end{flushleft}\par\vspace*{-1.5\baselineskip}
951                     \begin{flushright}%
952                         \BoxCandidati
953                     \end{flushright}%
```

```
954           \else
955             \noindent
956             \makebox[\textwidth]{\BoxRelatori\hfill\BoxCandidati}\par
957           \fi
958         \fi
959       \fi
```

The final item is the optional name of the company supervisor.

```
960 \ifcsvoid{@tutoreaziendale}{}{%
961     \vfill\vfill
962     {\centering \textbf{\@tutoreaziendalename}\\[.6ex]
963     \@tutoreaziendale\par}}
```

All the material now is on the page; we put some more vertical glue and handle
the copyright page; then we close the page sending it to the output file; the final
\endgroup closes the \begingroup that was set at the beginning of this long
macro. In order to handle the copyright page, we test if the internal definition
of the copyright page text is empty; in this case no copyright page should be
output, and a \creadoublepage works fine in both one and two side printing; if
the copyright page text is not empty, after shipping out the title page, we set the
copyright page text flush bottom with the text block, and then we ship out also
this copyright page.

```
964        \par\clearpage
965        \ifcsvoid{@retrofrontespizio}{}%
966        {\null\vfill\thispagestyle{empty}\@retrofrontespizio\par\clearpage}%
967 \endgroup}
```

4.5   A sample configuration file

The following code generates a sample configuration file that the user can change
at will; it can be used as a template for generating a really personal configuration
file. Remember: this template file is named `toptesi.cfg`, but in order to use
it for a specific thesis, whose main file is named JohnSmithMSthesis.tex, the
configuration file must be named JohnSmithMSthesis.cfg.

```
968 %%
969 %%================================================================
970 %% This file is the only file of the TOPtesi bundle that the user
971 %% can modify without restrictions in order to customise the
972 %% contents of this configuration file to his/her needs. The user
973 %% can add or remove lines, comment or uncomment lines, change the
974 %% arguments to the macros, add definitions and so on.
975 %% Use  this file by copying it to another file to be named as the
976 %% thesis main file and with extension .cfg; This bundle will try
977 %% to read "\jobname.cfg"; if this file it does nothing. This means
978 %% that the provided file toptesi.cfg is to be used as a model, not
979 %% to% be used directly.
980 %%================================================================
981 %%
982 \ateneo{Politecnico di Torino}
```

```
983 \facolta{}% nessun nome di default/ no default name
984 \FacoltaDi{}% nessun prefisso per la facolt/no default faculty label
985 %%\DottoratoIn{Corso di dottorato in }
986 \CorsoDiLaureaIn{Corso di Laurea in }
987 \TesiDiLaurea{Tesi di Laurea Magistrale}
988 %%\NomeMonografia{Monografia di Laurea}
989 %%\NomeDissertazione{Tesi di Dottorato}
990 \InName{in}
991 %%\TutorName{Tutore}
992 %%\CycleName{ciclo}
993 %%\retrofrontespizio{Questo testo  soggetto alla Creative Commons Licence}
```

4.6   The `topcoman.sty` code

This file may be used as an independent extension package for the `report` document class, and possibly for other classes

The new command `\DeclareSlantedCapitalGreekLetters` optionally sets the capital Greek letters in math mode with the glyphs taken from the math italic fonts, not from the math roman fonts, as it is by default; some authors prefer to use both symbols with different meanings, so this command lets them do so. This may be useful unless the XeTeX typesetting engine is used; in facts the Unicode math defines specific commands for setting any Latin or Greek mathematical letter in any possible font shape and series.

```
994 \RequirePackage{iftex}
995 \ifPDFTeX
996 \newcommand*\DeclareSlantedCapitalGreekLetters{%
997     \mathchardef\Gamma="7100
998     \mathchardef\Delta="7101
999     \mathchardef\Theta="7102
1000    \mathchardef\Lambda="7103
1001    \mathchardef\Xi="7104
1002    \mathchardef\Pi="7105
1003    \mathchardef\Sigma="7106
1004    \mathchardef\Upsilon="7107
1005    \mathchardef\Phi="7108
1006    \mathchardef\Psi="7109
1007    \mathchardef\Omega="710A
1008 }\fi
```

The `\ensuremath` command is defined in the LaTeX 2$_\varepsilon$ kernel from a certain version on; should the user employ a really old LaTeX 2$_\varepsilon$ implementation, this definition supplies the missing one. Should the `babel` package not be loaded, then we provide the useful command `\textormath` provided by `babel`. We define the text version of the subscript and ensure also that the `textcomp` package is loaded; of course if it is already loaded the `\RequirePackage` command performs the necessary tests and possibly does not load anything.

```
1009 \providecommand*\ensuremath[1]{\ifmmode#1\else$#1$\fi}%
1010 \providecommand*\textormath{}
```

```
1011 \renewcommand*\textormath{\ifmmode\expandafter\@secondoftwo\else
1012     \expandafter\@firstoftwo\fi}
1013 \providecommand*\textsubscript{\raisebox{-0.5ex}}
1014
1015 \ifPDFTeX
1016 \RequirePackage{textcomp}
1017 \fi
```

The following commands may be already defined; should they be missing they are supplied here. Most of them are already defined in the Italian option to the `babel` language if the thesis is typeset with `pdflatex` that loads that package; these commands are not predefined if the thesis is typeset with XƎLᴬTEX or LuaLᴬTEX that do not load the `babel` package; but remember; this package may be used as a stand alone one, without the initial call by the `toptesi` document class, so that the language Italian might be undefined.

```
1018 \providecommand{\ohm}{\textormath{\textohm}{\mathrm{\Omega}}}
1019 \providecommand{\ped}[1]{\textormath{\textsubscript{#1}}{_{\mathrm{#1}}}}
1020 \providecommand{\ap}[1]{\textormath{\textsuperscript{#1}}{^{\mathrm{#1}}}}
1021 \providecommand{\unit}[1]{\ensuremath{{\mathrm{\,#1}}}}
1022 \providecommand{\gei}{\ensuremath{{\mathop{\mathrm{j}}\nolimits}}}
1023 \providecommand{\eu}{\ensuremath{{\mathop{\mathrm{e}}\nolimits}}}
1024 \providecommand{\micro}{\textormath{\textmu}{%
1025   \ifPDFTeX
1026     \ifx\muup\undefined\mu\else\muup\fi
1027   \else
1028     \mathup{\mu}%
1029   \fi
1030 }}
1031 \providecommand{\gradi}{\textormath{\textdegree}{^\circ}}
```

The next set of definitions are used to list a program file. There are finer packages to perform this task in the CTAN archives. This set of macros has the advantage that is very short and light on the computer memory; nevertheless it performs its duty pretty well. The font face is the typewriter one; the font size is chosen so as to allow approximately 80 characters in the text width of the thesis. It respects the possible indentation tabs (ASCII code 9) by inserting the correct amount of typewriter spaces so as to align every line to the correct left margin which is an 'integer multiple of 8' spaces. If the program file contains some form feed ASCII characters (ASCII code 12), this TEX code inserts a new page command. The code may appear strange, and it does because it makes heavy use of the dirty tricks of appendix D of the TEXbook. For what concerns the macro \obeyspaces (already defined in the LᴬTEX 2_ε kernel) we had to change it in order to abolish a conflict with package ABC used to typeset music. The latter package is about 15 years younger than `topcoman`, so it should be this latter package duty to avoid conflicts with pre existing packages; on the opposite we accepted the suggestion of Enrico Gregorio, the author of ABC, whom we thank very much, because his package is much more complicated than this one, so that the conflict is avoided more easily by correcting `topcoman`.

```
1032 \providecommand*{\programmafont}{\ttfamily\footnotesize}
1033 \def\listing#1{\par\begingroup
1034 \programma \input #1 \endgroup}
1035 \def\uncatcodespecials{\def\do##1{\catcode`##1=12}\dospecials}
1036 \def\programma{\programmafont \parindent 0pt
1037 \def\par{\leavevmode\egroup\box0\endgraf}
1038 \catcode`\`=\active \catcode`\^^I=\active \catcode`\^^L=\active
1039 \obeylines \uncatcodespecials \obeyspaces
1040 \begingroup\lccode`~=`\ \lowercase{\endgroup\global\let~}\ %
1041 \everypar{\startbox}}
```

The above code does the whole work, except the alignment to the tab stops and the new page command associated to the form feed ASCII character that are done with the following definitions.

```
1042 \newdimen\tabwidth
1043 \setbox0=\hbox{\programmafont\space}
1044 \tabwidth=8\wd0
1045 \def\startbox{\setbox0=\hbox\bgroup}
1046 %{\obeyspaces\global\let =\space}
1047 {\catcode`\`=\active \gdef`{\relax\lq}}
1048 {\catcode`\^^I=\active
1049 \gdef^^I{\leavevmode\egroup \dimen0=\wd0
1050 \divide\dimen0 by\tabwidth
1051 \multiply\dimen0 by\tabwidth
1052 \advance\dimen0 by\tabwidth
1053 \wd0=\dimen0 \box0 \startbox}}
1054 {\catcode`\^^L=\active \global\let^^L\newpage}
```

Remember that the comma as a decimal separator is required for all languages except English. If you use this module outside TOPtesi, but as an extension of other classes or modules, it's up to you to define an "intelligent comma" macro or to load either the `icomma.sty` or the `nccomma.sty` modules that define such a macro: the `icomma` package defines the comma as a mathematical active character that recognises the subsequent token as a space token so as to insert a punctuation comma; `nccomma` behaves more or less as the macro defined in TOPtesi and recognises if the following token is a digit so as to use a decimal comma.

The following commands are used to write the "lower case" roman numerals with the small-caps font; in order to avoid complications with missing fonts or math environments we make sure to typeset these numerals with script size capitals; this solution is not probably the best one but it works; it typesets these roman numerals with the current font; in TOPtesi roman numerals are used only for folios, but in order to comply with the `hyperref` module, I avoided using this new macro for folios; in other situations there are no problems with the choice of font shapes and series. We need a robust command in order to set the script math size

```
1055 \DeclareRobustCommand*{\simulatedSC}[1]{%
1056 {\mbox{$\relax$}\fontsize{\sf@size}{\f@baselineskip}\selectfont#1}}%
```

A user, Antonio Mele, suggested and requested the possibility of having the figure and table name inserted automatically when the `\ref` command is issued. For single citations the solution works fine, but for range references it does not

work. In Italian the name must be lower case while in other languages, specifically in English, the name has a capital initial. Since this feature might be handy in certain circumstances and annoying in other ones, this feature can be turned on and off at will with the enabling and disabling commands. By default the feature is disabled.

```
1057 \def\ft@figure{\iflanguage{italian}{\MakeLowercase{\figurename}}%
1058          {\figurename}~}
1059 \def\ft@table{\iflanguage{italian}{\MakeLowercase{\tablename}}%
1060          {\tablename}~}
1061 %
1062 \newcommand*\EnableFigTabNames{%
1063          \let\p@figure\ft@figure\let\p@table\ft@table}
1064 \newcommand*\DisableFigTabNames{%
1065          \let\p@figure\empty\let\p@table\empty}
1066 %
1067 \DisableFigTabNames
```