The TOPtesi bundle

Claudio Beccari

v.5.59c 2011/10/18

Abstract

This file describes the TOPtesi bundle; it is a set of files designed to typeset with LaTeX 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 X͟ǝLATEX program. Up to now the few conflicts that have been spotted have been solved with suitable corrections or additions. The most important feature with XeLaTeX is that the option pdfa cannot be used any more, because it resorts to a compilation with pdfLaTeX; the typesetting engine X͟ǝTEX does not produce directly any PDF output but a modified, extended DVI output, even if this format is immediately transformed into a PDF file through a special version, xdvipdfmx, of the conversion program. Another minor XeLaTeX 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 feature.

1 Introduction

The TOPtesi bundle collects a certain number of files, specifically

- 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.

- An extension package `toptesi.sty` that contains most of the code for the real typesetting; it might be used as an extension to the report class file, although this use is discouraged.

- A second extension package `topfront.sty` that contain 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.

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

- Previous version of this bundle contained also the logos of a certain number 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.

- A documentation file `toptesi-doc-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-doc-xetex.tex` may be used as a sample for typesetting one's thesis with XeLaTeX.

The above files are complemented with a configuration file that any user may customize at will; these personalizations make it easy to configure the bundle so as to make it suitable for another language; in facts the babel and polyglossia packages contains localizations 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.

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

This version of TOPtesi has been tested with XeLaTeX. Some conflicts have been spotted and solved; may be there are still hidden ones, so at this version number this bundle should be considered as a "beta" one. The only main drawback still present when running XeLaTeX 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 XeLaTeX 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 specialized material: formulas, diagrams and pictures, texts written in non Latin alphabets, special symbols for philological mark-up, 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.

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 document 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.

classica Specifies a general modification of certain details that are supposed to be more adequate in the humanities fields; 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 margins 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 customized 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 default size 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 other sectioning commands.

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

2.2   Title page commands

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 "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; most commands receive one argument; only the command `\titolo` and `\cleardoublepage` accept an optional argument according to the usual LaTeX syntax:

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

`\frontespizio` is the actual command that typesets the title page and this command uses all the information that is passed to it by means of the following commands.

`\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 labeling 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. Alternatively it is possible

to redefine the macros `\Candidato` (masculine singular), `\Candidata` (feminine singular), `\Candidati` (masculine or mixed gender plural), `\Candidate` (feminine plural), in order to adjust the candidates' labels to the proper gender and number.

`\sedutadilaurea` gets the date of the final exam, or presentation, or defense 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 following 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 that contains the university logo; 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.

\setbindingcorrection sets up the length to displace the text block to the external margin so as to have a wider internal margin to accommodate for the binding correction. Its argument is not optional and is used to modify the default correction of 7 mm.

\cleardoublepage may be used as the default command, but it accepts an optional argument in order to chose the page style for the possible blank page to be output in order to open the next major section on an odd page when typesetting with the `twosides` option.

\setcleardoublepage uses its argument to establish the default page style when typesetting a blanck page by means of the command **\cleardoublepage**. If this command is not used the default page style is `plain`.

Since the infix strings are all memorized 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 localize 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 are contained in the package `topfront.sty`, which can be used as an autonomous extension to the `report` or `book` document class. One could easily typeset 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 page.

2.3 Typesetting commands

The 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` or `polyglossia` packages. If the source thesis file is being typeset by means of XeLaTeX the `babel` package is not loaded; in its place the `polyglossia` package is loaded that should implement in XeLaTeX most of the functionality provided by `babel` to 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 default language. An initial specification of **\selectlanguage{english}** sets the English language as 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 so start the document by specifying **\selectlanguage{french}**. But the user should pay attention to use `babel` in the proper way:

- Due to the way LaTeX classes load the requested package 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 global class options; therefore,

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}
```

- With the 2010 TEX system full distributions, both TEXLive and MiKTeX, 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 is the list of all language hyphenation rules that are available in the format file. While (American) English is the default and almost any *basic* distribution of the TEX system has many language defaults, 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 so as to consider the languages s/he wants to work with, and finally s/he must recreate the format files and put them in the proper places of the standard or the local folder tree. The user is invited to carefully investigate on these fine points and to configure the system very properly; 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 fulfill 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 the other 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 XƎLaTeX it is unnecessary, due to the larger set of math alphabets and math font commands that are available with proper UNICODE math fonts. Beware: in French, in spite of the ISO regulations, the Greek math symbols are supposed to be always upright, both the upper and lower case letters; with XeLaTeX there are no problems; when using pdfLaTeX it's necessary to use font that have both varieties of Greek letters, such as the Times eXtended , the Palatino eXtende, the Fourier fonts, et cetera.

`\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 whatever is not a variable, from the names of functions (such as sin, cos, log, etc.), to any index that contains 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 may be used both in math and in text mode.

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

`\unit` sets the unit of measure close to the numerical measure 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 to input `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 *after* `topcomand.sty` is read, the last definition is the one in force, therefore if one wishes to use the `unitsx` package s/he should not encounter any inconsistencies.

`\micro` sets the decimal prefix μ in 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.

`\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 to be typed with an upright serifed font, just as operators are.

`\eu` inserts the Euler 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 mathematical constant, but the electron charge e is typeset in math italics because this is a physical "constant", not a mathematical one. X∄TEX allows to typeset upright Greek letters, so there is the facility to typeset an upright "pi" (the number) to be distinguished from an inclined "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 lines are 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 can be used as an independent extension package with (possibly) any document class.

3 Format PDF suitable for archiving

This section in general does not apply if the thesis is typeset by means of X∄LATEX, because the typesetting engine in this case cannot directly produce a PDF formatted output file. For this reason the option `pdfa`, that shall be described shortly, should not specified to the class `toptesi`; at the same time if this option is specified, but the thesis is typeset by means of X∄LATEX, the effects of this options are disabled.

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 Organization (ISO) that in 2005 published the regulation[1] 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.

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 maintained. 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 information, such as the keywords that ease up library searches.

Since version 1.40, the program `pdflatex` is capable of producing PDF/A-1b conforming 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 and is capable of producing PDF/A-1b files.

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

- 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`; 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 8.61 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 XƎLaTeX, and produces valid results provided some caution is exercised when using fonts.

- Download from CTAN and install the package `pdfx`, if it is not already part of your TeX system complete distribution.

---

[1] In 2011 the updated regulation 19005-2:2011 has been published, but ths does not imply substancial modifications in this thesis bundle.

- Install in the main `pdfx` directory a good version of a color model profile file, such as, for example, `ECI-RGB-V1.0.icc` (see the download page of the site `www.eci.org`).

- 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}
  ```

  Keywords require a specific XML style format that can be examined in the package documentation.

- 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$_{\exists}$LATEX do not require any patch, but unfortunately its PDF byproduct is not PDF/A compliant (it must be converted by means of `ghostscript`).

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

- If you include PDF images that contain some text, be sure that the font for this text is 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.

- 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 the this or that is not conforming to the PDF/A specification. A suitable program is the Preflight plug-in of Adobe Acrobat Professional version 8 or later, but this, although the most authoritative, is a commercial program; probably your university has special facilities for this task.

- If you use `xelatex` for typesetting your thesis, you have to accompany your master file with another one that contains the metadata and othe necessary information. Be sure to call this auxiliary file as the masterfile name glued with `-def.ps`; so if your mastefile is named `JohnSmithMasterThesis.tex`, name this auxiliary file as `JohnSmithMasterThesis-def.ps` and fill in the necessary metadata and other thechnical information into this file in the

line 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. There are strong chances that if you follow the general recommendations shown here and in the document `toptesi-doc-xetex.pdf` file, the output file will be PDF/A compliant.

Let's issue a warning: if you are using Adobe Reader X (or later), this program will open a PDF file beginning with an information header claiming that the file is PDF/A 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 realization of PDF/A conforming files sets forth several problems that are of great concern for the large Institutions that have thousands of document a year to archive; it is not a question implied in the free nature of the `pdflatex` and `xelatex` 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, 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.

```
1 ⟨∗class⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesClass{toptesi}
4    [2011/10/18 v.5.59c Class for typesetting university theses]
```

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 latter, on turn, is called with the `a4paper` option. According to the LaTeX machinery of option passing the options passed on to the loaded class should be appended to the existing options, so that if `letterpaper` was specified in the calling statement of `toptesi`, this last option would override the default `a4paper` option. The same holds true for the other package options except for the encoding name used as an option for the `inputenc` package; this simply does not work and the package is not loaded any more as it was in the previous versions[2]. Similarly if another language was specified in the

---

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

list of `toptesi` options, this language would be appended yo the `babel` package default options and would become the default language.

```
 5 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{report}}
 6 \ProcessOptions\relax
 7 \LoadClass[a4paper]{report}
 8 \RequirePackage{ifxetex}
 9 \ifxetex
10   \RequirePackage{fontspec}
11   \@ifundefined{define@boolkey}{\RequirePackage{xkeyval}}{\relax}
12   \RequirePackage{polyglossia}
13   \setmainlanguage{italian}
14   \setotherlanguage{english}
15   \renewcommand*{\iflanguage}[1]{\ifnum\the\language=\csname l@#1\endcsname
16   \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
17 \else
18   \RequirePackage[english,italian]{babel}
19 \fi
20 \RequirePackage{toptesi}
21 %
22 ⟨/class⟩
```

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 aolone package, although, take notice, it might conflict with the `book` class code and 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.

```
23 ⟨*topsty⟩
24 \NeedsTeXFormat{LaTeX2e}
25 \ProvidesPackage{toptesi}[2011/10/18 v.5.59c Extension for toptesi.cls]%
```

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 I am used to these ones.

```
26 \def\TRON{\tracingcommands \tw@ \tracingmacros \tw@}
27 \def\TROFF{\tracingcommands\z@ \tracingmacros \z@}
28 \let\TROF\TROFF
```

Now we define the specific package macros: `classica` and `trieste` are identical; `trieste` 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` pacckage and is already included 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 specific to the `classica` style a couple of boolean variables are defined and the option definitions set them to the value "true".

```
29 \newif\if@utoretitolo \@utoretitolofalse
30 \newif\if@ldstyle \@ldstylefalse
31 \newif\if@xivpt \@xivptfalse

32 \newlength\T@Pbinding\setlength\T@Pbinding{7mm}
33 \def\setbindingcorrection#1{\T@Pbinding=#1}
34 \newif\if@binding \@bindingfalse
35 \newif\ifT@Ppdfa  \T@Ppdfafalse
36 \newif\ifchapterbibliography \chapterbibliographyfalse
37 \newif\ifclassica \classicafalse
38 \DeclareOption{cucitura}{\@bindingtrue}
39 \DeclareOption{14pt}{\@xivpttrue}
40 \DeclareOption{chapterbib}{\chapterbibliographytrue}
41 \DeclareOption{trieste}{\classicatrue}% Just for backwards compatibility
42 \DeclareOption{classica}{\classicatrue}
43 \DeclareOption{autoretitolo}{\ifclassica\@utoretitolotrue\fi}
44 \DeclareOption{oldstyle}{\ifclassica\@ldstyletrue\fi}
45 \DeclareOption{pdfa}{\T@Ppdfatrue}
46 %
47 \ProcessOptions\relax
```

The `graphicx` package is loaded by default; it is required for inserting the university logo; 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.

```
48 \RequirePackage{graphicx}
49 \if@xivpt\input{size14.clo}\fi
```

The text height gets set so as to cope with the standard UNI A4 size; in any case the typesetting grid is set to an height of 585pt and is then rounded to an integer number of lines depending on the size option that has been chosen. The `\topmargin`, `\headheight` and `\headsep` parameters have already been set in the size class option file, so that here just a small adjustment is made.

```
50 \textheight 585pt
51 \divide\textheight by \baselineskip
52 \multiply\textheight by \baselineskip
53 \advance\textheight by \topskip
```

For the text grid width a certain reduction is made if the extra size of 14pt is chosen.

```
54 \ifx\f@size\@xivpt
55     \textwidth=138mm
56 \else
57     \textwidth=150mm
58 \fi
```

Because of the above adjustment the right and left margins must be recomputed; in any case they must be recomputed because the default positioning of the text grid is centered; only if the class option `cucitura` is specified (`\if@binding` in this case is true) the text grids are moved to the outside in order to leave a wider inner margin. At the same time if the text grid is moved to the outside, the marginal note width must be reduced accordingly. These changes are executed at the `\begin{document}` stage.

```
59 \oddsidemargin=\paperwidth
60 \advance\oddsidemargin -\textwidth
61 \divide\oddsidemargin \tw@
62 \advance\oddsidemargin -1in
63 \evensidemargin=\oddsidemargin
64 \marginparwidth\oddsidemargin
65 \advance\marginparwidth 1in
66 \advance\marginparwidth-2.5\marginparsep
67 %
68 \AtBeginDocument{%
69 \if@binding
70 \typeout{Messaggio Toptesi: Ricalcolo i margini}
71 \advance\oddsidemargin \T@Pbinding
72 \advance\evensidemargin -\T@Pbinding
73 \advance\marginparwidth -\T@Pbinding
74 \fi}
```

We now establish the page style. We start by setting to "empy" 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`.

```
75 \def\lapagina{\thepage}
76 \mark{{}{}}
77 \newbox\@intesta
78 %
79 \def\ps@plain{\let\@mkboth\@gobbletwo
80 \def\@oddfoot{\null\hfill {\scshape\lapagina}\hfill \null}\def\@oddhead{}
81 \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 capitalization as it happens in all the default document classes, and it is underlined at a fixed distance from the base line. If the chapter or section heading a warning is issued so as to invite the user to exploit the sectioning commands optional short argument.

```
82 \if@twoside
83   \def\ps@headings{\let\@mkboth\markboth%
84   \def\@oddfoot{\null\hfill {\scshape\lapagina} \hfill\null}
```

```
85    \let\@evenfoot\@oddfoot
86    %
87    \def\@evenhead{\setbox\@intesta\hbox{\footnotesize\slshape
88        \leftmark}%
89    \ifdim\wd\@intesta>\textwidth \headWarn{\chapter}\fi%
90    \underline{\makebox[\textwidth]{\footnotesize\slshape
91        \strut\leftmark}}}%
92    \def\@oddhead{\setbox\@intesta\hbox{\footnotesize\slshape
93        \rightmark}%
94    \ifdim\wd\@intesta>\textwidth \headWarn{\section}\fi%
95    \underline{\makebox[\textwidth]{\footnotesize\slshape
96        \strut\rightmark}}}%
97    \def\chaptermark##1{\markboth{\thechapter\ -- ##1}{}}
98    \def\sectionmark##1{\markright{\ifnum\c@secnumdepth>\z@
99    \thesection\ -- \fi ##1}}}
100 \else
101    \def\ps@headings{\let\@mkboth\markboth
102    \def\@oddfoot{\null\hfill {\scshape\lapagina} \hfill\null}
103    \def\@evenfoot{}
104    \def\@oddhead{\setbox\@intesta\hbox{\footnotesize\slshape
105        \rightmark}%
106    \ifdim\wd\@intesta>\textwidth \headWarn{\chapter}\fi%
107    \underline{\makebox[\textwidth]{\footnotesize\slshape
108        \strut\rightmark}}}%
109    \def\chaptermark##1{\markright{\thechapter\ -- ##1}}}
110 \fi
111
112 \def\headWarn#1{\PackageWarning{toptesi}{%
113    THE HEADING IS TOO LONG\MessageBreak
114    Use the optional argument of command \string#1\MessageBreak
115    See the LaTeX Handbook (1994) on section C.4.1\MessageBreak}}
```

Bisogna ridefinre anche il modo di inserire nell'indice i capitoli non numerati con
le pagine in numeri romani maiuscoletti

```
116 \renewcommand*\l@chapter[2]{%
117    \ifnum \c@tocdepth >\m@ne
118      \addpenalty{-\@highpenalty}%
119      \vskip 1.0em \@plus\p@
120      \setlength\@tempdima{1.5em}%
121      \begingroup
122        \parindent \z@ \rightskip \@pnumwidth
123        \parfillskip -\@pnumwidth
124        \leavevmode \bfseries
125        \advance\leftskip\@tempdima
126        \hskip -\leftskip
127        #1\nobreak\hfil \nobreak
128        \hb@xt@\@pnumwidth{\hss\unless\ifxetex\normalfont\fi\scshape{#2}}\par
129        \penalty\@highpenalty
130      \endgroup
131    \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.

```
132 \newif\iffigurespage
133 \newif\iftablespage
134 \newif\ifnumeriromani
135 \newif\iffrontmatter
```

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

```
136 \def\frontmatter{\clearpage\ps@plain\pagenumbering{roman}%
137     \numeriromanitrue\frontmattertrue\@openrightfalse}
138 \def\mainmatter{\if@twoside\@openrighttrue\fi
139     \cleardoublepage\ps@headings\pagenumbering{arabic}%
140     \numeriromanifalse\frontmatterfalse}
141 \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.

```
142 \figurespagefalse
143 \tablespagefalse
```

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

```
144 \def\sommario{%
145  \iffrontmatter\else\frontmattertrue\fi
146  \if@openright\cleardoublepage\else\clearpage\fi
147  \global\@topnum\z@
148  \@afterindentfalse
149  \@schapter{\summaryname}%
150  \addcontentsline{toc}{chapter}{\summaryname}%
151        }
152 %
153 \def\ringraziamenti{%
154  \iffrontmatter\else\frontmattertrue\fi
155  \if@openright\cleardoublepage\else\clearpage\fi
156  \global\@topnum\z@
157  \@afterindentfalse
158  \@schapter{\acknowledgename}%
159  \addcontentsline{toc}{chapter}{\acknowledgename}%
160        }
```

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 localizing 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 the one for the main matter.

```
161 \renewcommand\chapter{%
162   \iffrontmatter\mainmatter\fi
163   \if@openright\cleardoublepage\else\clearpage\fi
164   \thispagestyle{plain}%
165   \global\@topnum\z@
166   \@afterindentfalse
167   \secdef\@chapter\@schapter}
168 %
169 \renewcommand\part{%
170   \iffrontmatter\mainmatter\fi
171   \if@openright\cleardoublepage\else\clearpage\fi
172   \thispagestyle{empty}%
173   \if@twocolumn\onecolumn\@tempswatrue\else\@tempswafalse\fi
174   \null\vfil
175   \secdef\@part\@spart}
176 \def\@endpart{\vfil\cleardoublepage
177                 \if@tempswa
178                   \twocolumn
179                 \fi}
```

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.

```
180 \renewcommand\tableofcontents{%
181 \if@twocolumn\@restonecoltrue\onecolumn\else\@restonecolfalse\fi
182 \if@openright\cleardoublepage\else\clearpage\fi
183 \global\@topnum\z@
184 \@afterindentfalse
185 \@schapter{\contentsname}%
186 \@starttoc{toc}%
187 \clearpage
188 \if@restonecol\twocolumn\fi
189     }
190 \renewcommand\listoffigures{%
191     \if@twocolumn\@restonecoltrue\onecolumn\else\@restonecolfalse\fi
192     \if@openright\cleardoublepage\else\clearpage\fi
193     \global\@topnum\z@
194     \@afterindentfalse
195     \@schapter{\listfigurename}
196     \@mkboth{\listfigurename}{\listfigurename}%
197     \@starttoc{lof}%
198     \clearpage
```

```
199     \if@restonecol\twocolumn\fi
200     }
201 \renewcommand\listoftables{%
202     \if@twocolumn\@restonecoltrue\onecolumn\else\@restonecolfalse\fi
203     \if@openright\cleardoublepage\else\clearpage\fi
204     \global\@topnum\z@
205     \@afterindentfalse
206     \@schapter{\listtablename}%
207     \@mkboth{\listtablename}{\listtablename}%
208     \@starttoc{lot}%
209     \clearpage
210     \if@restonecol\twocolumn\fi
211     }
```

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.

```
212 \def\indici{%
213 \iffrontmatter\else\frontmattertrue\fi
214         \tableofcontents
215         \iftablespage
216                 {\addvspace{10pt}
217                 \let\saveaddvspace=\addvspace
218                 \def\addvspace##1{}
219                 \listoftables
220                 \let\addvspace=\saveaddvspace}
221         \fi
222         \iffigurespage
223                 {\addvspace{10pt}
224                 \let\saveaddvspace=\addvspace
225                 \def\addvspace##1{}
226                 \listoffigures
227                 \let\addvspace=\saveaddvspace}
228         \fi
229 }
```

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 double column.

```
230 \onecolumn
```

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

```
231 \if@twoside
232     \flushbottom
233 \else
234     \ifx\@xivpt\f@size
235         \raggedbottom
236     \else
237         \flushbottom
238     \fi
```

```
239 \fi
240 \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 largeer 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 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.

```
241 \def\interlinea#1{\linespread{#1}\selectfont}
242 \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.

```
243 \def \@floatboxreset {%
244         \reset@font
245         \linespread{1}%
246         \normalsize
247         \@setminipage
248 }
```

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 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 figure 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.

```
249 \def\fps@figure{htb} \def\fps@table{htb}
250 %
251 \setcounter{topnumber}{2}
252 \def\topfraction{1}
253 \setcounter{bottomnumber}{1}
254 \def\bottomfraction{.5}
255 \setcounter{totalnumber}{3}
256 \def\textfraction{0}
257 \def\floatpagefraction{0}
258 \setcounter{dbltopnumber}{2}
259 \def\dbltopfraction{1}
260 \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` counter 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 centered 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 shrinking of the caption width 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

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

```
261 \newdimen\captionwidth
262 \long\def\@makecaption#1#2{%
263     \begingroup
264     \small \parskip\z@ \parindent\z@
265     \finalhyphendemerits 100000\relax
266     \linespread{0.95}\selectfont
267     \vskip \abovecaptionskip
268     \captionwidth=\hsize
269     \advance\captionwidth-3em
270     \setbox0 \hbox{#1.\quad#2}%
271     \ifdim\wd0>\hsize
272         \setbox1 \vbox{\hsize=\captionwidth
273         \unhbox0\par\global\setbox2\lastbox}%
274         \setbox2\hbox{\unhbox2}%
275         \ifdim\wd2<0.333333\captionwidth
276             \count255=\ht1 \advance\count255 \dp1
277             \divide\count255\baselineskip
278             \advance\count255\@ne
279             \@tempdima=\wd2
280             \advance\@tempdima \count255\captionwidth
281             \advance\@tempdima 0.5\captionwidth
282             \advance \count255\@ne
283             \divide \@tempdima \count255
284             \captionwidth=\@tempdima
285             \setbox0 \vbox{\hsize\captionwidth
286             #1.\quad#2}
287         \else
288             \setbox2\hbox to\captionwidth{\unhbox2 \hfill}%
289             \setbox0\vbox{\unvbox1\box2}%
290         \fi
291     \fi
292     \makebox[\hsize]{\box0}%
293     \endgroup
294 }
```

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 a unnumbered chapter or section.

```
295 \def\redef@bibitem{\def\@bibitem##1{\item\if@filesw
296         \immediate\write\@auxout
297         {\string\bibcite{##1}{\thechapter.\the\c@enumi}}\fi\ignorespaces}}
298 %
299 \def\thebibliography#1{%
300 \ifchapterbibliography\section*{\bibname}\relax
301     \if@twoside\markright{\bibname}\fi
302     \addcontentsline{toc}{section}{\bibname}\relax
303     \redef@bibitem
```

```
304     \list{[\thechapter.\arabic{enumi}]}{%
305     \settowidth\labelwidth{[\thechapter.#1]}\leftmargin\labelwidth
306     \advance\leftmargin\labelsep\itemsep\z@ plus 1pt\parsep\z@
307     \usecounter{enumi}}
308 \else
309     \chapter*{\bibname}\relax
310     \@mkboth{\bibname}{\bibname}\relax
311     \addcontentsline{toc}{chapter}{\bibname}\relax
312     \list{[\arabic{enumi}]}{\settowidth\labelwidth{[#1]}%
313     \leftmargin\labelwidth
314     \advance\leftmargin\labelsep\itemsep\z@ plus 1pt\parsep\z@
315     \usecounter{enumi}}
316 \fi
317 \def\newblock{\hskip .11em plus .33em minus -.07em}
318 \sloppy
319 \sfcode`\.=1000\relax}
320
321 \let\endthebibliography=\endlist
```

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

```
322 \long\def\@footnotetext#1{\insert\footins{\linespread{1}\footnotesize
323     \interlinepenalty\interfootnotelinepenalty
324     \splittopskip\footnotesep
325     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
326     \hsize\columnwidth \@parboxrestore
327     \edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
328     \@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 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 recognized as a digit, therefore the "intelligent" comma inserts the necessary space by itself.

```
329     \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
330     \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}
331     \AtBeginDocument{\mathcode`\,=\string"8000}
332     {\catcode `,=\active \gdef,{\futurelet\let@token\m@thcomma}}
333     \def\m@thcomma{\let\@tempB\virgola
```

```
334 \@tfor\@tempA:=0123456789\do{%
335 \expandafter\ifx\@tempA\let@token\let\@tempB\virgoladecimale
336 \@break@tfor\fi}\@tempB}
337 %
338 \catcode`\^^Z=10
339 \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 compilations 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.

```
340 \let\summaryname\empty
341 \let\acknowledgename\empty
342 \newcommand*\ExtendCaptions[3]{{%
343 \@ifundefined{captions#1}{%
344 \PackageWarning{toptesi}{Language option #1 not specified\MessageBreak
345 Skipping any redefinition\MessageBreak}%
346 }{%
347 \expandafter\let\expandafter\@tempA\csname captions#1\endcsname
348  \toks0=\expandafter{\@tempA%
349     \def\summaryname{#2}%
350     \def\acknowledgename{#3}}%
351 \expandafter\xdef\csname captions#1\endcsname{\the\toks0}%
352 }}}%
```

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

```
353 \ExtendCaptions{italian}{Sommario}{Ringraziamenti}
354 \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 default language and nothing is necessary for setting 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.

```
355 \def\italiano{\selectlanguage{italian}}%
356 \def\english{\selectlanguage{english}}%
357 \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 default 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:

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

- 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.

- If XeLaTeX is being used for typesetting it suffixes 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.

```
358 \AtBeginDocument{%
359     \italiano
360     \renewcommand\@chapapp{\chaptername}%
361     }
```

At last the subsidiary and independent packages `topcoman` and `topfront` are requested for input. For using XeLaTeX as the typesetting engine it's necessary to load such packages at the "begin document" step, so that all a the other settings, especially fonts, are already established. May be it should be better done also when using pdfLaTeX as the typesetting engine, but the past experience has not shown any inconvenience.

```
362 \ifxetex
363 \AtBeginDocument{\RequirePackage{topfront}
364 \RequirePackage{topcoman}}
365 \else
366 \RequirePackage{topfront}
367 \RequirePackage{topcoman}
368 \fi
369 \def\sf@size{7}
```

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 glyps. 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:

```
370 \ifxetex\else
371 \renewcommand*\not{\mathrel{\mathchoice%
372 {\rlap{$\displaystyle\mkern2.5mu\mathnormal{/}$}}%
373 {\rlap{$\textstyle\mkern2.5mu\mathnormal{/}$}}%
374 {\rlap{$\scriptstyle\mkern2.5mu\mathnormal{/}$}}%
375 {\rlap{$\scriptscriptstyle\mkern2.5mu\mathnormal{/}$}}%
376 }}
```

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:

```
377 \renewcommand\mapstochar{\mathrel{\mathchoice
378 {\displaystyle\unitlength=0.0555\fontdimen6\font
379     \picture(0,0)\put(1,1.5){\linethickness{0.72\unitlength}%
380     \line(0,1){6}}\endpicture}
381 {\textstyle\unitlength=0.0555\fontdimen6\font
382     \picture(0,0)\put(1,1.5){\linethickness{0.72\unitlength}%
383     \line(0,1){6}}\endpicture}
384 {\scriptstyle\unitlength=0.0555\fontdimen6\font
385     \picture(0,0)\put(1,0.55){\linethickness{0.61\unitlength}%
386     \line(0,1){5}}\endpicture}
387 {\scriptscriptstyle\unitlength=0.0555\fontdimen6\font
388     \picture(0,0)\put(1,0.2){\linethickness{0.56\unitlength}%
```

```
389     \line(0,1){4}}\endpicture}
390 }}\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.

Finally, if the class option `pdfa` was specified, we load the `pdfx.sty` file with the suitable option for typesetting a (hopefully) PDF/A conforming file. We must remember that `pdfx.sty` on turn loads the `hyperref.sty` file with the necessary `pdfa` option. The user, therefore does not need to reload that package, but is free to configure it through the `\hypersetup` command arguments right at the beginning of the thesis main file.

Nevertheless if X_ELATEX is being used to typeset the document, loading of the package `pdfx` is skipped; at the same time the `hyperref` package is being loaded anyhow just because the `pdfa` option was specified, but at least the `pdfa` option is specified, so that even if it's necessary to resort to `ghostscript` in order to produce a PDF/A compliant file, at least the hyperlinks to external sites are suitably "sterilized".

```
391 \ifT@Ppdfa
392   \unless\ifxetex
393     \RequirePackage[a-1b]{pdfx}
394   \else
395     \RequirePackage{hyperref}
396   \fi
397 \fi
```

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 let me believe that the philological mark-up requires typesetting environments similar to the mathematical ones, since 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 localizing 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.

```
398 \ifclassica
```

```
399 \def\Candidato{Laureando}
400 \def\Candidata{Laureanda}
401 \def\Candidati{Laureandi}
402 \def\Candidate{Laureande}
403 \fi
```

Since theses in humanities may end up to occupy several volumes (classically "tomo" in Italian means "volume", although the latter spelling 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 typeset 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 `\frontespizio` macro typesets the real title page and we have already seen that it can print the volume ordinal. The localization allows to change the infix string that is going to be printed. When several tomes are involved, instead of using the `\frontespizio` macro, use the `\tomo` one, that provides to stepping up the volume counter before actually printing the new tome title page.

```
404 \newcounter{tomo}
405 \newcommand*{\tomo}{\clearpage\stepcounter{tomo}\frontespizio}
```

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.

```
406 \renewcommand*{\lapagina}{%
407     \ifnumeriromani
408         \thepage
409     \else
410         \if@ldstyle
411             \expandafter\oldstylenums\expandafter{\thepage}%
412         \else
413             \thepage
414         \fi
415     \fi
416     }
```

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 a 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 make use of 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 on the center, and finally 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!

```
417 \if@utoretitolo
418     \if@twoside
419 %
420         \renewcommand*{\ps@headings}{\let\@mkboth\markboth%
421         \def\@oddfoot{\null \hfill \textbf{\lapagina} \hfill \null}%
422         \let\@evenfoot\@oddfoot
423 %
424         \def\@evenhead{%
425             \setbox\@intesta\hbox{\unless\ifxetex\latintext\fi
426             \footnotesize\strut\textsc{%
427             \@author\ifx\@secondauthor\empty\else\ et al.\fi: \@stitle}%
428             }%
429         \ifdim\wd\@intesta>\textwidth\headwrn{\titolo}\fi
430         \underline{\makebox[\textwidth]{\box\@intesta}}}%
431         \def\@oddhead{\unless\ifxetex\latintext\fi
432             \setbox\@intesta\hbox{%
433                 \footnotesize\strut\textsl{\rightmark}}%
434         \ifdim\wd\@intesta>\textwidth \headWarn{\section}\fi%
435         \underline{\makebox[\textwidth]{\box\@intesta}}}%
436         \def\chaptermark##1{\markright{\thechapter\ -- ##1}{}}%
437         \def\sectionmark##1{}}%
438     \else
439         \renewcommand*{\ps@headings}{\let\@mkboth\markboth
440             \def\@oddfoot{\null \hfill \textbf{\lapagina}\hfill \null}%
441             \let\@evenfoot\empty\let\@evenhead\empty
442             \def\@oddhead{{\setbox\z@\hbox{\unless\ifxetex\latintext\fi\footnotesize
443             \textsc{%
444             \@author\ifx\@secondauthor\empty\else\ et al.\fi: \@stitle}}%
```

```
445          \setbox\tw@\hbox{\unless\ifxetex\latintext\fi\footnotesize\textsl{\rightmark}}%
446          \dimen@=2em\advance\dimen@\wd\z@\advance\dimen@\wd\tw@
447          \ifdim\dimen@<\textwidth \relax
448          \else
449            \setbox\z@\vbox{\hsize.48\textwidth\parindent\z@\raggedright
450            \unless\ifxetex\latintext\fi\footnotesize\textsc{%
451            \@author\ifx\@secondauthor\empty\else\ et al.\fi: \@stitle
452            }}%
453            \setbox\tw@\vbox{\hsize.48\textwidth\parindent\z@\raggedleft
454                \unless\ifxetex\latintext\fi\footnotesize\textsl{\rightmark}}%
455          \fi
456          \setbox\@intesta\vbox to\z@{%
457              \vss\hbox to\textwidth{\strut\box\z@\hfill\box\tw@}}%
458          \underline{\box\@intesta}}%
459          \def\chaptermark##1{\markright{\thechapter\ -- ##1}}}
460    \fi
461 \fi
```

Here `\annoaccademico` is defined that typesets the infix string corresponding to
"Anno accademico" followed by the year range in old style numbers (irrespective of
the `oldstyle` option); localization 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 the specific shape of the old style
numbers.

```
462 \newcommand*\NomeAnnoAccademico[1]{\gdef\AnnoAccademico{#1}}
463 \@ifundefined{AnnoAccademico}{\gdef\AnnoAccademico{Anno accademico}}{}
464 %
465 \def\annoaccademico#1{\ifclassica
466   \def\@submitdate{{\large\textsc{\AnnoAccademico}} {\Large\s@tanno#1!}}
467   \else
468   \PackageWarning{toptesi}{\string\annoaccademico\space is usable only
469    when the\MessageBreak
470    'classica' option is in force}%
471   \def\@submitdate{\AnnoAccademico\ #1 --- Needs 'classica' option}
472   \fi}
473 \def\s@tanno#1-#2!{\oldstylenums{#1\EnDash#2}}
474
475 \def\EnDash{{\settowidth{\dimen@}{\large\sc I}%
476 \setbox\tw@\hbox{2}\dimen\tw@.5\ht\tw@\advance\dimen\tw@-.5\dp\tw@
477 \dimen4\dimen\tw@\advance\dimen4by.0385ex\relax
478 \advance\dimen\tw@-.0385ex\relax
479 \makebox[1.5\dimen@]{%
480     \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 specialized extensions 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.

```
481 \renewcommand\footnoterule{%
482    \kern-6\p@
483    \hrule\@width.4\columnwidth
484    \kern5.6\p@}
485 \setlength\footnotesep{12\p@}
486 \setlength{\skip\footins}{24\p@ \@plus 4\p@ \@minus 2\p@}
487 \newcommand*\nota[1][\mathord{*}]{%
488    \xdef\@thefnmark{\ensuremath{\m@th#1}}\@footnotemark\@footnotetext
489 }
490 \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.

```
491 \newcommand*\paginavuota{\clearpage\thispagestyle{empty}\null\clearpage}
492 %
493 \newenvironment{dedica}{\clearpage
494    \if@twoside
495        \ifodd\c@page\else\thispagestyle{empty}\null\clearpage\fi
496    \fi
497    \thispagestyle{empty}%
498    \list{}{\labelwidth\z@
499    \leftmargin.5\textwidth
500    \parindent\z@
501    \raggedright\LARGE\itshape}\item[]
502 }{%
503    \endlist\clearpage
```

```
504 }
505 %
506 \newenvironment{citazioni}{%
507     \clearpage\thispagestyle{empty}
508     \list{}{\labelwidth\z@
509     \leftmargin.5\textwidth
510     \parindent\z@
511     \raggedright}\item[]
512 }{%
513     \endlist\clearpage
514 }
```

Another enhancement that `toptesi` has compared to the original `report` class is a redefined `\cleardoublepage` macro. This new macro may leave, as usual, a blank page in order to skip to an odd numbered one. But the default style of this blank page may be set as the user likes best. The default style is `plain`, because I think that it's better to always have a numbered page than a totally white page. Nevertheless what I think it is more suitable is not something anybody should agree with; therefore the command accepts an optional argument so as to use the style s/he likes best; moreover the command `\setcleardoublepage` redefines the default style as many times the user would like to switch style. `\TT@clpagedefault` contains the default style; `\setcleardoublepage` redefines the above default, and `\cleardoublepage` uses the symbolic name `\TT@clpagedefault` or the optional argument and therefore uses the style that has been set as the default one or the style the uses wants to use in a particular instance.

```
515 \def\TT@clpagedefault{plain}
516 \newcommand*\setcleardoublepage[1]{\gdef\TT@clpagedefault{#1}}
517 \renewcommand\cleardoublepage[1][\TT@clpagedefault]{\clearpage\if@twoside \ifodd\c@page\else
518     \hbox{}\thispagestyle{#1}\newpage\if@twocolumn\hbox{}\thispagestyle{#1}\newpage\fi\fi\fi}
519 ⟨/topsty⟩
```

4.4   The package file `topfront.sty`

This file is input by `toptesi`, version 5.x, but it can be used as an independent extension package.

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.

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

```
520 ⟨*topfrn⟩
521 \NeedsTeXFormat{LaTeX2e}
522 \ProvidesPackage{topfront}[2011/10/18 v.5.59c Title page for TOPtesi]
```

The title page information depends on the type of "thesis' that is being typeset. Since the topfront package may be used independently form the accompanying class `toptesi`, it's necessary to define the switch `\ifclassica` in case it was not already defined. Therefore we thest if the switch is undefined, and in case we define ti. Furthermore 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.

```
523 \@ifundefined{ifclassica}{%
524     \expandafter\newif\csname ifclassica\endcsname}{\relax}
525 \newif\iffemminile
526 \newif\ifdottorato \dottoratofalse
527 \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 side 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.

```
528 \def\monografia#1{\global\laureatriennaletrue
529     \global\dottoratofalse
530     \global\def\titolo##1{\PackageWarning{toptesi}%
531     {Il titolo e' gia' stato impostato con
532         il comando \string\monografia}}%
533     \gdef\@title{#1}}
534 \let\@stitle\empty
535 \newcommand*{\titolo}[2][]{{\def\@tempA{#1}%
536     \ifx\@tempA\empty\gdef\@stitle{#2}\else\gdef\@stitle{#1}\fi
537     \gdef\@title{#2}}}
538 \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 reflect this subtle difference might be the following:

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

</div>

```
539 \let\@materia\empty
540 \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 doctoring'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`.

```
541 \newif\ifDirettore \Direttorefalse
542 \def\tutore#1{\gdef\@tutore{#1}}
543 \def\direttore{\Direttoretrue\relatore}%
544 \def\coordinatore{\Direttorefalse\relatore}%
545 \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.

```
546 \def\relatore#1{\gdef\@principaladviser{#1}}
547 \def\secondorelatore#1{\gdef\@secondadviser{#1}}
548 \def\terzorelatore#1{\gdef\@thirdadviser{#1}}
549 \def\candidato#1{\gdef\@author{#1}\femminilefalse}
550 \def\candidata#1{\gdef\@author{#1}\femminiletrue}
551 \def\secondocandidato#1{\gdef\@secondauthor{#1}\femminilefalse}
552 \def\secondacandidata#1{\gdef\@secondauthor{#1}}
553 \def\terzocandidato#1{\gdef\@thirdauthor{#1}\femminilefalse}
554 \def\terzacandidata#1{\gdef\@thirdauthor{#1}}
```

The next set of macros is used to typeset the "date" of the thesis defense 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[3] then the formatting of the "date" may be still different. This command is aliased with `\esamendidottorato` which literally means "defense of the doctoral dissertation"; nevertheless both commands refer to a simple date in one of those formats.

`\getseduta` splits the date in 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@dutaTrieste` is called with the whole string. On turn `\s@dutaTrieste` 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 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.

```
555 \def\sedutadilaurea#1{\getseduta#1 !}
556 \def\getseduta#1 #2!{%
557     \def\@tempA{#2}%
558     \ifx\@tempA\empty
559         \def\@submitdate{#1}%
```

---

[3]This option cannot be active if this extension package is used to extend a document class other than `toptesi`; nevertheless if the package `trieste.sty` is loaded before this package, the relevant information equivalent to `classica` is fully defined and this package works as it should.

```
560     \else
561         \unless\ifclassica
562             \def\@submitdate{#1 #2}%
563         \else
564             \s@dutaTrieste#1 #2!%
565         \fi
566     \fi
567 }%
568 \def\s@dutaTrieste#1 #2!{%
569     \if@ldstyle
570         \s@paranumeri#2-!%
571         \ifx\2\empty
572             \edef\@submitdate{\noexpand#1 \noexpand\oldstylenums{#2}}%
573         \else
574             \s@paranumeri#2!%
575             \edef\@submitdate{\noexpand#1
576             \noexpand\oldstylenums{\1} -- \noexpand\oldstylenums{\2}}%
577         \fi
578         \let\1\undefined
579         \let\2\undefined
580     \else
581         \def\@submitdate{#1 #2}%
582     \fi
583 }
584 \def\s@paranumeri#1-#2!{\def\1{#1}\def\2{#2}}%
585 \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 package 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 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 receives 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 Ingegneria"; 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. Finally \logosede gets the name of the graphic file that contains the information relative to the university logo. If such logo is not available, the user does not specify this command; if the user uses this command to insert the university logo

but the file is not available the usual `graphicx` package warning is issued but compilation goes on without the logo. The `\tutoreaziendale` macro is the last 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.

```
586 \def\ciclodidottorato#1{\gdef\@ciclo{#1 \@cyclename}}%
587 \def\corsodilaurea#1{\global\dottoratofalse\gdef\@corso{#1}}
588 \def\corsodidottorato#1{\global\dottoratotrue\global\laureatriennalefalse
589     \gdef\@corso{#1}}
590 \def\scuoladidottorato#1{\global\dottoratotrue\global\laureatriennalefalse
591     \gdef\@phdschool{#1}}
592 \def\ateneo#1{\gdef\@ateneo{\expandafter\uppercase{\expandafter #1}}}
593 \def\nomeateneo#1{\gdef\@nomeat{\expandafter\uppercase{\expandafter #1}}}
594 \newcommand\facolta[2][]{\gdef\@facname{#2}\gdef\@facnumber{#1}}
595 \def\logosede#1{\edef\@logosede{\noexpand\includegraphics[width=3cm]{#1}}}
596 \def\tutoreaziendale#1{\gdef\@tutoreaziendale{#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 "Tomo terzo" in, say, "Third volume"

NomeQuartoTomo "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 localizing language strings: it is necessary to localize the whole phrase, because of the position of the adjectives.

```
597 \newcommand\FacoltaDi[1]{\gdef\@faculty{#1}}
598 \newcommand\DottoratoIn[1]{\gdef\@PhDname{#1}}
599 \newcommand\CorsoDiLaureaIn[1]{\gdef\@laureaname{#1}}
600 \newcommand\TesiDiLaurea[1]{\gdef\@TesiDiLaurea{#1}}
601 \newcommand\NomeMonografia[1]{\gdef\@monografia{#1}}
602 \newcommand\NomeDissertazione[1]{\gdef\@dissertazione{#1}}
603 \newcommand\InName[1]{\gdef\@InName{#1}}
604 \newcommand\CandidateName[1]{\gdef\@nomecandidato{#1}}
605 \newcommand\AdvisorName[1]{\gdef\Relatore{#1}\gdef\Relatori{#1}}
606 \newcommand\CoAdvisorName[1]{\gdef\Correlatore{#1}\gdef\Correlatori{#1}}
607 \newcommand\TutorName[1]{\gdef\Tutore{#1}}
608 \newcommand\NomeTutoreAziendale[1]{\gdef\@tutoreaziendalename{#1}}
609 \newcommand\CycleName[1]{\gdef\@cyclename{#1}}
610 \newcommand\NomePrimoTomo[1]{\gdef\PrimoTomo{#1}}
611 \newcommand\NomeSecondoTomo[1]{\gdef\SecondoTomo{#1}}
612 \newcommand\NomeTerzoTomo[1]{\gdef\TerzoTomo{#1}}
613 \newcommand\NomeQuartoTomo[1]{\gdef\QuartoTomo{#1}}
```

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 be 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 only **\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 may be 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**.

```
614 \IfFileExists{toptesi.cfg}{\input{toptesi.cfg}}{}
615 %
616 \@ifundefined{@cyclename}{\def\@cyclename{ciclo}}{}
617 \@ifundefined{@title}{\def\@title{}}{}
618 \@ifundefined{@author}{\def\@author{}}{}
619 \@ifundefined{@principaladviser}{\def\@principaladviser{}}{}
620 \@ifundefined{@secondadviser}{\def\@secondadviser{}}{}
621 \@ifundefined{@thirdadviser}{\def\@thirdadviser{}}{}
622 \@ifundefined{@PhDdirector}{%
623     \edef\@PhDdirector{\ifDirettore Direttore \else Coordinatore \fi
624     del corso di dottorato}}{}
625 \@ifundefined{@tutore}{\def\@tutore{}}{}
626 \@ifundefined{@secondauthor}{\def\@secondauthor{}}{}
627 \@ifundefined{@thirdauthor}{\def\@thirdauthor{}}{}
628 %
629 \@ifundefined{@nomerelatore}{\def\@nomerelatore{}}{}
630 \@ifundefined{@nomecandidato}{\def\@nomecandidato{}}{}
631 \@ifundefined{Candidato}{\def\Candidato{Candidato}}{}
632 \@ifundefined{Candidata}{\def\Candidata{Candidata}}{}
633 \@ifundefined{Candidati}{\def\Candidati{Candidati}}{}
634 \@ifundefined{Candidate}{\def\Candidate{Candidate}}{}
635 \@ifundefined{Relatore}{\def\Relatore{Relatore}}{}
636 \@ifundefined{Relatori}{\def\Relatori{Relatori}}{}
637 \@ifundefined{Correlatore}{\def\Correlatore{Correlatore}}{}
638 \@ifundefined{Correlatori}{\def\Correlatori{Correlatori}}{}
639 \@ifundefined{Tutore}{\def\Tutore{Tutore}}{}
640 \@ifundefined{@tutoreaziendale}{\def\@tutoreaziendale{}}{}
641 \@ifundefined{@tutoreaziendalename}%
642         {\def\@tutoreaziendalename{Supervisore Aziendale}}{}
643 \@ifundefined{@subtitle}{\def\@subtitle{}}{}
644 %
645 \@ifundefined{@corso}{\def\@corso{}}{}
646 \@ifundefined{@ciclo}{\def\@ciclo{}}{}
647 \@ifundefined{@ateneo}{\def\@ateneo{POLITECNICO DI TORINO}}{}
648 \@ifundefined{@nomeat}{\def\@nomeat{}}{}% Nome proprio dell'ateneo
649 \@ifundefined{@facolta}{\def\@facname{Ingegneria}}{}
650 \@ifundefined{@facnumber}{\def\@facnumber{}}{}
651 \@ifundefined{@faculty}{\def\@faculty{Facolt\`a di }}{}
652 %
653 \@ifundefined{PrimoTomo}{\def\PrimoTomo{Tomo primo}}{}
654 \@ifundefined{SecondoTomo}{\def\SecondoTomo{Tomo secondo}}{}
655 \@ifundefined{TerzoTomo}{\def\TerzoTomo{Tomo terzo}}{}
656 \@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 by default is also empty.

```
657 \@ifundefined{@submitdate}{\def\@submitdate{\ifcase\the\month\or%
658   Gennaio\or Febbraio\or Marzo\or Aprile\or Maggio\or Giugno\or
```

```
659    Luglio\or Agosto\or Settembre\or Ottobre\or Novembre\or Dicembre\fi
660    \space \the\year}}{}
661 \@ifundefined{@logosede}{\let\@logosede\relax}{}
662 %
663 \@ifundefined{@TesiDiLaurea}{\def\@TesiDiLaurea{Tesi di Laurea}}{}
664 \@ifundefined{@phdschool}{\def\@phdschool{SCUOLA DI DOTTORATO}}{}
665 \@ifundefined{@PhDname}{\def\@PhDname{Dottorato in }}{}
666 \@ifundefined{@laureaname}{\def\@laureaname{Corso di Laurea in }}{}
667 \@ifundefined{@dissertazione}{\def\@dissertazione{Tesi di Dottorato}}{}
668 \@ifundefined{@monografia}{\def\@monografia{Monografia di Laurea}}{}
669 \@ifundefined{@InName}{\def\@InName{in}}{}
```

Finally we have the only active macro \frontespizio, the real macro that actually typesets the title page. The macro starts with a local redefinition of the myheadings page style; with this style the university name goes into the header, while the exam date goes into the footer; in this way the information that must be typeset in the title page is not crunched in too little a space and the overall text bounding box assumes a shape similar to that of the page.

Everything is closed within a group so all definitions remain local.

```
670 \def\frontespizio{%
671 \begingroup
672        \def\ps@myheadings{\let\@mkboth\markboth
673        \def\@oddhead{\vbox{%
674        \hbox to \hsize{\hfill {\LARGE \@ateneo}\hfill}%
675        \ifclassica
676              \hbox to \hsize{\hfil\vrule\@width\z@
677           \@height3.5ex\vrule\@height1.4\p@\@depth-\p@\@width50mm\hfil}%
678        \fi
679        }}%
680        \def\@oddfoot{\vbox to 15mm{\vss
681        \ifclassica
682          \hbox to \hsize{\hfil\raisebox{-.3ex}[\z@][\z@]{%
683          \vrule\@height-2.6\p@\@depth3\p@\@width50mm}\hfil}%
684        \fi
685        \hbox to\hsize{\hfill{\Large{\@submitdate}}\hfill}}%
686        }%
687        \let\@evenhead\@oddhead
688        \let\@evenfoot\@oddfoot
689        }%
```

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 each margin.

```
690        \advance\oddsidemargin \evensidemargin
691        \divide\oddsidemargin \tw@
692        \evensidemargin \oddsidemargin
```

We select the myheadings page style; 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.

```
693        \thispagestyle{myheadings}%
694        \null\setcounter{page}{1}%
695        \normalfont
```

The generic university name is already in the header; if it is non blank the first thing we set in the page is the university proper name and some vertical glue.

```
696        \ifx\@nomeat\empty \else
697            {\vspace*{-3\topskip}\centering
698                {\Large \@nomeat}\par}\vfill
699        \fi
```

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`.

```
700        \begin{center}
701        {\rmfamily\mdseries
702        \ifdottorato
703            \large \@phdschool
704        \else
705            \LARGE\ifx\@facnumber\empty\else\@facnumber\space\fi
706            \@faculty\@facname%
707        \fi
708        \par\medskip}%
```

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

```
709        \ifx\@corso\empty \else{\large
710            \ifdottorato
711                \@PhDname\@corso\ifx\@ciclo\empty\else~--~\@ciclo\fi
712            \else
713                \@laureaname\@corso
714            \fi
715            \par}
716        \fi
717        \end{center}
```

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

```
718        \vfill
719        \begin{center}
720          \LARGE
721          \ifdottorato
722              \@dissertazione%
723          \else
724              \iflaureatriennale
725                  \@monografia%
```

```
726            \else
727                    \@TesiDiLaurea%
728            \fi
729        \fi
730        \ifx\empty\@materia\else
731                \\\@InName\\\@materia
732        \fi
733    \end{center}
```

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

```
734        \vfill
735        \begin{center}
736        {\huge\bfseries \baselineskip=0.95em plus 1pt
737                \@title \par}
738        \end{center}
739 % con l'eventuale sottotitolo
740        \ifx\@subtitle\empty \else
741            \begin{center}%
742                \large\textrm{\@subtitle}\par
743            \end{center}%
744        \fi
```

If the option `classica` is in force the thesis might be divided in several volumes; theses in humanities are apparently often oversized. In this case the `\frontespizio` 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.

```
745        \ifclassica
746            \ifnum\value{tomo}>\z@
747                \par\bigskip
748                \noindent\makebox[\textwidth]{%
749                \large\textbf{%
750                \ifcase\c@tomo%
751                    \or \PrimoTomo%
752                    \or \SecondoTomo%
753                    \or \TerzoTomo%
754                    \or \QuartoTomo%
755                    \else
756                        \PackageWarning{toptesi}{%
757                        Counter tomo equals \the\c@tomo\MessageBreak
758                        It was not considered a thesis might get
759                        divided in more than four volumes}%
760                    \fi}}%
761            \fi
762        \fi
763        \par
```

Going down in the title page now comes the optional insertion of the university logo; this is a simple task once `\@logosede` has been defined, and this was done

either with an explicit command `\logosede` with its argument in the preamble, or a specific line in the configuration file.

```
764        \ifx\@logosede\@empty\else
765        \vspace*{\bigskipamount}
766        {\centering \@logosede\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.

```
767        \vfill
768        \iflaureatriennale
769             \let\@nomerelatore\empty
770        \else
771             \ifdottorato
772                  \edef\@nomerelatore{\@PhDdirector}%
773             \else
774                  \ifx\@principaladviser\empty\else
775                       \def\@nomerelatore{\Relatore:}%
776                  \fi
777                  \unless\ifclassica
778                       \ifx\@secondadviser\empty \else
779                            \def\@nomerelatore{\Relatori:}%
780                       \fi
781                  \fi
782             \fi
783        \fi
```

Similarly the label names for the exam candidates are chosen; in Italian the 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.

```
784        \ifdottorato
785             \let\@nomecandidato\empty
786        \else
787     \iflanguage{italian}{%
788             \iffemminile
789                  \def\@nomecandidato{\Candidata:}%
790             \else
791                  \def\@nomecandidato{\Candidato:}%
792             \fi
793             \ifx\@secondauthor\empty \else
794                  \iffemminile
795                       \def\@nomecandidato{\Candidate:}%
796                  \else
```

```
797                              \def\@nomecandidato{\Candidati:}%
798                     \fi
799              \fi}{}
800          \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 *only* candidate is centered and written in caps-and-small-caps.

```
801          \iflaureatriennale
802              \begin{center}%
803                  \large\mdseries\textsc{\@author}
804              \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.

```
805          \else
806 %  Per laurea e dottorato con relatori e correlatori...
807              \def\BoxRelatori{%
808                  \begin{tabular}[t]{l}%
809                  \hbox{\ifclassica\else\large\fi
810                       \textbf{\protect\@nomerelatore}}\\[.6ex]
811                  \hbox{\large\textrm{\protect\@principaladviser}}%
812                  \ifx\@secondadviser\empty \else
813                      \ifclassica
814                          \ifx\@thirdadviser\empty
815                              \ifx\@secondadviser\empty\else
816                                    \\[1.5ex]\textbf{\Correlatore:}%
817                              \fi
818                          \else
819                              \\[1.5ex]\textbf{\Correlatori:}%
820                          \fi
821                      \fi
822                      \\[.6ex]\hbox{{\large\textrm{\protect\@secondadviser}}}%
823                  \fi
824                  \ifx\@thirdadviser\empty \else
825                      \\[.6ex] \hbox{{\large\textrm{\protect\@thirdadviser}}}%
826                  \fi
827                  \end{tabular}%
828              }%
```

A similar approach is taken for the candidate name(s)

```
829        \def\BoxCandidati{%
830           \begin{tabular}[t]{l}%
831              \hbox{\unless\ifclassica\large\fi
832                 \textbf{\protect\@nomecandidato}}\\[.6ex]
833                 \hbox{\large\textrm{\protect\@author}}%
834                 \ifx\@secondauthor\empty
835                 \else
836                    \\\hbox{\large\textrm{\protect\@secondauthor}}%
837                 \fi
838                 \ifx\@thirdauthor\empty
839                 \else
840                    \\\hbox{\large\textrm{\protect\@thirdauthor}}%
841                 \fi
842           \end{tabular}%
843        }%
```

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 doctoring person name is typeset centered 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 person labels and the second line containing the true names. The central column is used just for spacing, but it does not contain anything.

```
844        \ifdottorato
845           \begin{center}\large
846              \textbf{\@author}\\[3em]
847              {\normalsize
848              \begin {tabular*}{\hsize}{@{\extracolsep{\fill}}ccc}
849              \ifx\@tutore\empty\else
850                 \textbf{\Tutore}
851              \fi
852              &\relax&
853              \ifx\@principaladviser\empty\else
854                    \textbf{\@nomerelatore}
855              \fi
856              \\
857              \ifx\@tutore\empty\else
858                    \@tutore
859              \fi
860              &\relax&
861              \ifx\@principaladviser\empty\else
862                    \@principaladviser
863              \fi
864              \end{tabular*}
865              }%
866           \end{center}
867        \else
```

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

```
868            \unless\ifclassica
869              \begin{flushleft}%
870                  \BoxRelatori
871              \end{flushleft}\par\vspace*{-1.5\baselineskip}
872              \begin{flushright}%
873                  \BoxCandidati
874              \end{flushright}\else
875              \noindent
876              \makebox[\textwidth]{\BoxRelatori\hfill\BoxCandidati}\par
877            \fi
878          \fi
879        \fi
```

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

```
880 \ifx\@tutoreaziendale\@empty\else
881     \vfill\vfill
882     {\centering \textbf{\@tutoreaziendalename}\\[.6ex]
883     \@tutoreaziendale\par}
884 \fi
```

All the material now is on the page; we put some more vertical glue and 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.

```
885        \vfill\par\null
886        \if@twoside\cleardoublepage\else\clearpage\fi
887 \endgroup}
888 ⟨/topfrn⟩
```

4.5   A sample configuration file

The following code generates a simple configuration file that the user can change at will; it can be used as a template for generating a really personal configuration file.

```
889 ⟨*topcfg⟩
890 %%
891 %%=================================================================
892 %% This file is the only file of the TOPtesi bundle that the user
893 %% can modify without restrictions in order to customize the
894 %% contents of this configuration file to his/her needs. The user
895 %% can add or remove lines, comment or uncomment lines, change the
896 %% arguments to the macros, add definitions and so on.
897 %%=================================================================
898 %%
899 \ateneo{Politecnico di Torino}
900 \facolta{Ingegneria}
```

```
901 \FacoltaDi{Facolt\`a di }
902 %%\DottoratoIn{Corso di dottorato in }
903 \CorsoDiLaureaIn{Corso di Laurea in }
904 \TesiDiLaurea{Tesi di Laurea Magistrale}
905 %%\NomeMonografia{Monografia di Laurea}
906 %%\NomeDissertazione{Tesi di Dottorato}
907 \InName{in}
908 %%\TutorName{Tutore}
909 %%\CycleName{ciclo}
910 ⟨/topcfg⟩
```

4.6 The package file `topcoman.sty`

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

```
911 ⟨*topcmn⟩
912 \NeedsTeXFormat{LaTeX2e} % lavora solo con LaTeX 2e
913 \ProvidesPackage{topcoman}[2011/10/18 v.5.59c Additional TOPtesi commands]
```

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, at is is by default; some persons 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.

```
914 \RequirePackage{ifxetex}
915 \ifxetex\else
916 \newcommand*\DeclareSlantedCapitalGreekLetters{%
917     \mathchardef\Gamma="7100
918     \mathchardef\Delta="7101
919     \mathchardef\Theta="7102
920     \mathchardef\Lambda="7103
921     \mathchardef\Xi="7104
922     \mathchardef\Pi="7105
923     \mathchardef\Sigma="7106
924     \mathchardef\Upsilon="7107
925     \mathchardef\Phi="7108
926     \mathchardef\Psi="7109
927     \mathchardef\Omega="710A
928 }\fi
```

The `\ensuremath` command is defined in the LaTeX kernel from a certain version on; should the user employ a really old LaTeX 2_ε 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 it it's already loaded the `\RequirePackage` command performs the necessary tests and possibly does not load anything.

```
929 \providecommand{\ensuremath}[1]{\ifmmode#1\else$#1$\fi}%
930 \providecommand{\textormath}{\ifmmode\expandafter\@secondoftwo\else
931     \expandafter\@firstoftwo\fi}
932 \providecommand*\textsubscript{\raisebox{-0.5ex}}
933 \ifxetex\else
934 \RequirePackage{textcomp}
935 \fi
```

The following commands may be already defined; should they be missing they are supplied here. The greatest part of them is 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 X$_{\exists}$LAT$_{E}$X that does 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.

```
936 \providecommand{\ohm}{\textormath{\textohm}{\mathrm{\Omega}}}
937 \providecommand{\ped}[1]{\textormath{\textsubscript{#1}}{_{\mathrm{#1}}}}
938 \providecommand{\ap}[1]{\textormath{\textsuperscript{#1}}{^{\mathrm{#1}}}}
939 \providecommand{\unit}[1]{\ensuremath{{\mathrm{\,#1}}}}
940 \providecommand{\gei}{\ensuremath{{\mathop{\mathrm{j}}\nolimits}}}
941 \providecommand{\eu}{\ensuremath{{\mathop{\mathrm{e}}\nolimits}}}
942 \providecommand{\micro}{\textormath{\textmu}{\ifxetex\mathup{}\else
943     \ifx\muup\undefined\mu\else\muup\fi\fi}}
944 \providecommand{\gradi}{\textormath{\textdegree}{^\circ}}
```

The next set of definitions are used to list a program file. There are better 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 boundary which is "zero modulo eight". If the program file contains some form feed ASCII characters (ASCII code 12), this T$_{E}$X 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 T$_{E}$Xbook.

```
945 \def\listing#1{\par\begingroup
946 \programma \input #1 \endgroup}
947 \def\uncatcodespecials{\def\do##1{\catcode`##1=12}\dospecials}
948 \def\programma{\ttfamily\footnotesize \parindent 0pt
949 \def\par{\leavevmode\egroup\box0\endgraf}
950 \catcode`\'=\active \catcode`\^^I=\active \catcode`\^^L=\active
951 \obeylines \uncatcodespecials \obeyspaces
952 \everypar{\startbox}}
```

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

```
953 \newdimen\tabwidth
```

```
954 \setbox0=\hbox{\ttfamily\footnotesize\space}
955 \tabwidth=8\wd0
956 \def\startbox{\setbox0=\hbox\bgroup}
957 {\obeyspaces\global\let =\ }
958 {\catcode`\'=\active \gdef`{\relax\lq}}
959 {\catcode`\^^I=\active
960 \gdef^^I{\leavevmode\egroup \dimen0=\wd0
961 \divide\dimen0 by\tabwidth
962 \multiply\dimen0 by\tabwidth
963 \advance\dimen0 by\tabwidth
964 \wd0=\dimen0 \box0 \startbox}}
965 {\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 they package outside TOPtesi, but as an extension of other packages, it's up to you to define an "intelligent comma" macro or to load either the `icomma.sty` or the `nccomma.sty` package that defines such a macro: the `icomma` package defines the comma as a mathematical active character that recognizes 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 recognizes digits so as to use a decimal comma.

The following commands are used to write the "lower case" roman numerals with the small-caps; 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 be compliant with the package **hyperref**, 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

```
966 \DeclareRobustCommand*{\simulatedSC}[1]{%
967 {\check@mathfonts\fontsize{\sf@size}{\f@baselineskip}\selectfont#1}}%
```

A user, Antonio Mele, suggested and requested the possibility of having the figure and table names 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.

```
968 \def\ft@figure{\iflanguage{italian}{\MakeLowercase{\figurename}}%
969               {\figurename}~}
970 \def\ft@table{\iflanguage{italian}{\MakeLowercase{\tablename}}%
971               {\tablename}~}
972 %
973 \newcommand*\EnableFigTabNames{%
974               \let\p@figure\ft@figure\let\p@table\ft@table}
975 \newcommand*\DisableFigTabNames{%
```

```
976              \let\p@figure\empty\let\p@table\empty}
977 %
978 \DisableFigTabNames
979 ⟨/topcmn⟩
```