Introduzione al pacchetto xparse

Claudio Beccari

Sommario

Il pacchetto xparse mette a disposizione un certo numero di macro avanzate per definire nuovi comandi e nuovi ambienti con una grande varietà di argomenti. Queste funzionalità sono preziose sia per chi scrive file di estensione sia per gli utenti finali.

Abstract

The xparse package allows to use a number of advanced macros to define new commands and environments with a large variety of arguments. Such facilities are very useful for both package writers and end users.

1 Introduzione

Ogni utente LaTeX sa che il linguaggio permette di definire nuovi comandi e nuovi ambienti usando i comandi di alto livello seguenti:

```
\newcommand
\renewcommand
\providecommand
\newenvironment
\renewenvironment
```

Il comando `\provideenvironment` manca dal nucleo di LaTeX. I comandi `\newcommand` e `\newenvironment` definiscono rispettivamente un nuovo comando o un nuovo ambiente che non sia stato già definito, altrimenti emettono un messaggio d'errore e non eseguono la definizione specificata nel loro argomento. I comandi `\renewcommand` e `\renewenvironment` ridefiniscono rispettivamente un comando o un ambiente che sia già stato definito; in caso contrario, segnalano l'errore e non procedono alla ridefinizione specificata. `\providecommand` definisce un nuovo comando che non sia stato già definito, ma non fa nulla se il comando esiste già.

Per definire comandi che accettino anche un asterisco finale bisogna ricorrere sostanzialmente a lunghi giri di macro che controllino se il comando è seguito da un asterisco e, a seconda dell'esito del controllo, attivano altri comandi interni a cui eventualmente forniscono argomenti predefiniti.

Questa interfaccia fra i comandi di definizione di alto livello e i comandi nativi che effettivamente eseguono quelle definizioni o ridefinizioni sono molto utili per il programmatore e, ancora di più, per l'utente finale che voglia definirsi comandi propri. Ma i comandi così definiti sono un po' rigidi e non permettono di fare più di tanto. Infatti:

- non consentono di definire comandi con argomenti delimitati;
- non consentono di definire comandi con più di un argomento facoltativo; questo inoltre può essere racchiuso solo fra parentesi quadre;
- non consentono di definire comandi sempre robusti;
- lo stesso vale per i comandi di apertura degli ambienti, con l'ulteriore limitazione che gli argomenti all'apertura dell'ambiente sono disponibili, appunto, solo per i comandi di apertura.

Per superare questi limiti, il programmatore deve adoperare i comandi di basso livello nativi del sistema TeX e deve saper maneggiare correttamente i comandi locali o globali e quelli che sviluppano o meno gli eventuali argomenti costituiti da macro. Non impossibile, ma è richiesta una certa esperienza per fare queste cose e tanta pazienza per correggere gli eventuali errori. Per definire comandi robusti l'utente deve ricorrere all'ulteriore comando `\DeclareRobustCommand` che però non esegue nessuna verifica sulla eventuale preesistenza del comando che si vorrebbe dichiarare robusto.

Da alcuni anni il LaTeX 3 Team sta creando un linguaggio intermedio fra quello nativo del sistema TeX e il mark up di alto livello di LaTeX. Questo linguaggio prende la sigla "L3"; e nei passati numeri di questa rivista Enrico Gregorio ha esposto diverse funzionalità di questo linguaggio.

La potenza di L3 è enorme ma di contro la sua sintassi è complessa e delicata. Nonostante le difficoltà d'uso, il linguaggio semplifica l'opera dei programmatori che non hanno più bisogno di inventarsi ogni volta come affrontare i problemi di programmazione che possono sorgere per costruzioni tipografiche complesse. Già da tempo molti pacchetti di uso generale sono stati tradotti in L3, facilitando la loro manutenzione e consentendo ulteriori funzionalità.

Il pacchetto xparse si pone appunto come interfaccia tra l'utente finale o il programmatore e il linguaggio L3, in modo da eliminare ogni difficoltà nella definizione di nuovi comandi e nuovi ambienti con la massima libertà, togliendo le limitazioni dei comandi nativi di LaTeX. Infatti, i nuovi comandi di definizione:

- consentono di usare argomenti delimitati fra delimitatori che l'utente stabilisce a suo piacimento;

- consentono di definire comandi con diversi argomenti facoltativi diversamente delimitati e a cui eventualmente è assegnato un valore preimpostato;
- consentono di definire comandi che sono sempre robusti;
- per gli ambienti gli argomenti obbligatori e facoltativi dello statement di apertura sono disponibili anche durante l'esecuzione dei comandi di chiusura;
- consentono di definire a piacere dell'utente argomenti booleani in modo simile a quanto fa l'asterisco per i comandi nativi di LaTeX.

.

In sostanza, usando le funzionalità del pacchetto xparse la definizione di una sola macro consente di fare in un colpo solo quello che con i comandi nativi o i comandi del nucleo di LaTeX richiede la definizione di catene di macro e di test, che spesso si rivelano fragili e danno luogo a errori difficili da rilevare e correggere.

2 Le macro del pacchetto di xparse

Le macro di xparse per definire nuovi comandi o ambienti ricorrono a un concetto nuovo: la lista dei *descrittori degli argomenti*. Invece di specificare solamente il numero degli argomenti, come avviene con i comandi del nucleo di LaTeX, bisogna fornire obbligatoriamente una lista di codici che descrivono ciascun argomento, ne specificano i delimitatori, se sono obbligatori o facoltativi, un eventuale valore predefinito, se sono di tipo booleano o se accettano un valore. Vedremo fra poco questa lista. Secondo la terminologia del linguaggio L3, comandi e ambienti si chiamano "funzioni"; le funzioni di tipo "command" agiscono sui loro argomenti; quelle di tipo "environment" agiscono sui loro argomenti e su quanto contenuto fra `\begin` e `\end`.

Ora vediamo invece la serie di comandi a disposizione.

`\NewDocumentCommand` definisce un nuovo comando.

`\RenewDocumentCommand` ridefinisce un preesistente comando.

`\ProvideDocumentCommand` provvede alla definizione di un comando se non è mai stato definito prima.

`\DeclareDocumentCommand` definisce un comando indipendentemente dal fatto che sia nuovo o preesistente.

`\NewDocumentEnvironment` definisce un nuovo ambiente.

`\RenewDocumentEnvironment` ridefinisce un ambiente preesistente.

`\ProvideDocumentEnvironment` provvede a definire un ambiente se non è mai stato definito prima.

`\NewExpandableDocumentCommand` definisce un nuovo comando espandibile.

`\RenewExpandableDocumentCommand` ridefinisce un comando preesistente espandibile.

`\ProvideExpandableDocumentCommand` se non era mai stato definito prima, provvede a definire un comando espandibile.

`\DeclareExpandableDocumentCommand` indipendentemente dal fatto che sia nuovo o preesistente definisce un comando espandibile.

Tutti i comandi che vengono definiti sono robusti; solo con gli ultimi quattro comandi di definizione essi sono sviluppabili; dovrebbero essere usati con la massima cautela e solo se veramente necessario; *se non si sa se l'espansione sia assolutamente necessaria, non si usino queste definizioni.* In sostanza questi ultimi quattro comandi sono per programmatori esperti, non per utenti finali che potrebbero non avere chiara la necessità di espandere certi comandi.

Oltre a questi comandi, ce ne sono alcuni altri che sono di uso decisamente frequente, mentre altri sono più specializzati e vanno studiati nella documentazione di xparse, (The LaTeX Project Team, 2018). I comandi frequenti sono i seguenti.

`\IfNoValue` verifica se un argomento facoltativo non ha ricevuto alcun valore.

`\IfValue` verifica se un argomento facoltativo ha ricevuto un valore.

`\IfBoolean` verifica se un argomento booleano è presente.

Tutti e tre questi comandi hanno una sintassi del tipo:

`\`⟨*comando*⟩`TF{`⟨*argomento*⟩`}{`⟨*vero*⟩`}{`⟨*falso*⟩`}`

Il suffisso `TF` può ridursi anche solo a `T` per indicare "true" o a `F` per indicare "false". Chi scrive preferisce sempre usare entrambe le uscite del test e se fosse necessario lascerebbe vuoto il campo ⟨*true*⟩ o quello ⟨*false*⟩. I primi due comandi servono per verificare se un dato ⟨*argomento*⟩, indicato con il suo numero d'ordine nella forma `#`⟨*numero*⟩, ha o non ha ricevuto un valore e a seconda del caso esegue il ramo ⟨*vero*⟩ oppure ⟨*falso*⟩. Il terzo comando serve per verificare se un argomento booleano è presente o assente; il risultato del test è vero solo se l'argomento booleano è presente. Ovviamente questi comandi sono da usare nel corpo della definizione delle macro indicate dai comandi specifici per definirli.

La sintassi dei comandi di definizione è la seguente

⟨*prefissi*⟩`command{`⟨*descrittori*⟩`}{`⟨*definizione*⟩`}`
⟨*prefissi*⟩`environment{`⟨*descrittori*⟩`}{`⟨*apertura*⟩`}%`
`{`⟨*chiusura*⟩`}`

Va notato che i ⟨*descrittori*⟩ sono generalmente una lista di codici separati da spazi (non da virgole) e identificano nell'ordine fino a nove argomenti (il massimo accessibile dai comandi nativi del sistema TEX sottostante); per le definizioni degli ambienti essi sono accessibili con lo stesso "numero" sia nei comandi di ⟨*apertura*⟩ sia nei comandi di ⟨*chiusura*⟩.

Va ancora notato che gli argomenti booleani (come per esempio l'asterisco) sono dei veri e propri argomenti e dunque contano nel loro novero. Per i comandi la cosa non è tanto diversa da come avviene per l'asterisco dei comandi di LATEX; invece per gli ambienti gli argomenti booleani, e quindi gli asterischi, sono veri argomenti e non fanno parte del nome dell'ambiente; questo è un po' diverso da quanto accade con i comandi del nucleo di LATEX, dove l'asterisco eventuale fa parte del nome dell'ambiente. Tanto per essere più chiari, gli ambienti figure e figure* sono due ambienti distinti nelle definizioni del cuore di LATEX, quindi, se è stato iniziato l'ambiente asteriscato con `\begin{figure*}`, esso deve deve venire chiuso con `\end{figure*}`. Con gli ambienti di xparse, l'eventuale asterisco va messo dopo l'apertura dell'ambiente; se figure fosse definito con i comandi di xparse la sua apertura si farebbe con `\begin{figure}*` e la chiusura con `\end{figure}`. Vedi più avanti anche l'esempio 4.3.

Questa particolarità potrebbe essere d'impiccio per molti utenti finali, meno per i programmatori, per l'abitudine che hanno i primi a usare la stessa stringa nei comandi di apertura e in quelli di chiusura.

3 La lista degli argomenti

La lista degli argomenti è semplicemente una lista di codici, facoltativamente separati da spazi, che descrivono le particolarità di ogni argomento nello stesso ordine in cui vengono inseriti dall'utente fra gli argomenti delle macro che egli usa.

Come si evince dai paragrafi precedenti, i descrittori sono numerosi.

Argomenti obbligatori

`m` indica un argomento *mandatory*, obbligatorio; dello stesso tipo dell'argomento di una normale macro LATEX.

`r`⟨*car1*⟩⟨*car2*⟩ indica un argomento *required*, obbligatorio, delimitato a sinistra dal carattere ⟨*car1*⟩ e a destra dal carattere ⟨*car2*⟩; nel cuore di LATEX questo genere di argomenti è piuttosto raro; si può ricordare l'argomento contenuto fra parentesi tonde che indica la coppia di coordinate dei punti nelle macro che si usano negli ambienti di disegno programmato; per esempio negli ambienti picture, tikzpicture e simili. Ma il nucleo di LATEX non dispone di comandi per definire argomenti delimitati che, quindi, sono definiti ricorrendo ai comandi nativi di TEX.

`v` indica un argomento da leggere "verbatim", racchiuso fra il primo e l'ultimo carattere di una stringa, allo stesso modo dell'argomento del comando `\verb`. Questo carattere delimitatore può essere un carattere qualsiasi, purché non faccia parte della stringa da leggere verbatim, né sia uno dei seguenti caratteri speciali: `\`, `#`, `{`, `}`, o ␣. Questo descrittore può servire per definire una macro che contiene una stringa verbatim, ma ha un certo numero di limitazioni e deve essere considerato sperimentale, secondo quanto affermato nella stessa documentazione di xparse. Qui non ne parlerò più, anche perché il suo uso mi sembra troppo limitato.

Argomenti facoltativi

`o` (lettera `o` minuscola) indica un normale argomento da introdurre fra parentesi quadre senza un valore predefinito; per sapere se l'argomento è stato usato e gli è stato assegnato un valore, bisogna controllarlo con la funzione `\IfValue` oppure `\ifNoValue` (con il suffisso `TF`).

`d`⟨*car1*⟩⟨*car2*⟩ indica un argomento facoltativo delimitato a sinistra dal carattere ⟨*car1*⟩ e a destra da ⟨*car2*⟩; non gli viene assegnato nessuna valore di default. Anche in questo caso se ne verifica il valore mediante la funzione `\IfValue` oppure `\ifNoValue`.

`O{`⟨*valore*⟩`}` (lettera `O` maiuscola) indica un argomento facoltativo delimitato da quadre a cui è assegnato ⟨*valore*⟩ come impostazione predefinita.

`D`⟨*car1*⟩⟨*car2*⟩`{`⟨*valore*⟩`}` indica un argomento delimitato come nel caso `d` e con un ⟨*valore*⟩ predefinito come nel caso `O`.

Argomenti booleani

`s` indica un argomento booleano costituito da una "stella", cioè un asterisco. Se ne verifica la presenza mediante la funzione `\IfBoolean` con il debito suffisso `TF`.

`t`⟨*car*⟩ Indica un "token" facoltativo costituito dal carattere ⟨*car*⟩. Funziona come l'asterisco, ma è costituito da un carattere specifico scelto dal programmatore.

Argomenti facoltativi "deprecati" Esistono alcuni descrittori di argomenti facoltativi che la documentazione di xparse raccomanda di non usare più, ma che sono mantenuti per compatibilità con il passato. Chi scrive ne ha usati alcuni in alcune sue macro e non ha mai riscontrato

inconvenienti; elenca qui di seguito solo quelli che effettivamente ha usato.

`g` descrive un argomento facoltativo delimitato da graffe e senza un valore predefinito. Si capisce subito perché delimitare con graffe alcuni argomenti facoltativi possa ingenerare confusione; tuttavia con la debita attenzione lo si può fare.

`G{`⟨*valore*⟩`}` descrive un argomento facoltativo delimitato da graffe con il ⟨*valore*⟩ predefinito; continua a valere la perplessità circa l'uso delle graffe per delimitare argomenti facoltativi; tuttavia, essendo possibile assegnare loro un ⟨*valore*⟩ predefinito, i rischi si riducono in modo considerevole. Nel seguito non presenterò esempi che usino questi descrittori, rispettando la raccomandazione del LaTeX Project Team di non farne uso.

Esistono altri ⟨*descrittori*⟩ di uso raro e comunque di carattere sperimentale.

Come si vede, dunque le funzioni definite da xparse accettano più di nove tipi diversi di argomenti: obbligatori o facoltativi; regolari o delimitati in modo specifico, con o senza valori predefiniti; con contenuto variabile o con valore booleano.

Dietro le quinte il linguaggio L3 trasforma queste funzioni in comandi nativi del sistema TeX; per gestire questa moltitudine di forme di definizione con il comandi normali del nucleo di LaTeX un programmatore si romperebbe la testa. Il LaTeX 3 Team si è dato da fare in modo incredibile; se l'utente si prendesse la briga di tracciare il lavorio che il programma di compilazione esegue dietro le quinte, troverebbe senz'altro molte più operazioni di quelle che un programmatore normale potrebbe cercare di fare usando il proprio intelletto. D'altra parte oggi i calcolatori sono così veloci che risparmiare sul numero delle operazioni da compiere per trasformare il codice sorgente in linguaggio macchina è diventato abbastanza superfluo: la velocità di compilazione di un dato documento dipende più dagli accessi ai dischi meccanici che non dalla CPU della macchina.

4 Esempi

Quando si parla di software molto potente è difficile presentare esempi significativi e non troppo complessi ma che descrivano bene le funzionalità che si desiderano illustrare.

Tuttavia un software come il pacchetto xparse potrebbe risultare difficile anche per un programmatore esperto. Questo, in qualità di esperto, sarebbe in grado di sperimentare autonomamente e in poco tempo potrebbe accumulare sufficienti conoscenze da potersi definire "esperto" anche di xparse. Un utente finale potrebbe invece essere intimorito da un linguaggio molto tecnico e potrebbe avere delle difficoltà a cimentarsi con il software in esame.

Presenterò quindi degli esempi reali, che discendono dalla mia esperienza diretta.

4.1 La scacchiera/cruciverba

Un esempio di applicazione delle funzionalità di xparse appare anche nell'articolo (Beccari *et al.*, 2018) dove viene definita con le funzionalità di xparse una funzione che accetta, oltre all'asterisco facoltativo, altri quattro argomenti diversamente delimitati di cui uno solo obbligatorio; alcuni di questi argomenti in realtà sono liste di valori e non tutti sono obbligatori. Facendo correttamente il conteggio delle possibilità che offrono gli argomenti facoltativi e la presenza o assenza di alcuni valori dalle liste, esistono ben sette elementi facoltativi che possono essere specificati oppure omessi; non tutti sono sensati, nel senso che hanno una certa ridondanza, ma in teoria esistono 128 modi diversi di specificare gli argomenti. Nessuna macro del nucleo di LaTeX consente una tale flessibilità.

Non riporto qui il codice del comando `\scacchiera` perché pubblicato nel numero precedente di questa rivista: questa sì che sarebbe una grossa ridondanza.

4.2 I comandi per l'indice analitico di una classe per guide su LaTeX

La classe GuidaLC serve per comporre certe brevi guide tematiche, non ancora pubblicate dal GUIT; evidentemente questa classe deve avere molte funzionalità specifiche adatte all'argomento di queste guide, che servono per descrivere un linguaggio formale, il mark up di LaTeX. Bisogna scrivere nel testo i nomi di molte macro, ma non bisogna svilupparle; si descrivono gli ambienti, i pacchetti e tante altre cose tipicamente legate a LaTeX. I nomi di questi oggetti vengono tutti riportati nell'indice analitico; sarebbe tuttavia oneroso scrivere ogni volta il comando per il testo e quello per l'indicizzazione duplicando sostanzialmente le stesse informazioni.

Prendiamo l'esempio delle macro, o *control sequence*, per scrivere le quali basta definire il comando `\cs`; spesso è necessario inviarne il nome anche all'indice analitico e altre volte basta solo scriverne il nome. Ecco allora che un comando asteriscato potrebbe essere utile: senza asterisco il comando scrive il nome e lo invia all'indice, mentre con l'asterisco scrive solo il nome.

Questo è un problema facile da risolvere; infatti basta definire una semplice funzione con xparse. Per scrivere una macro in modo che non venga sviluppata è necessario privarla della barra iniziale in modo da usarne solo il nome: è compito della macro da definire quello di prefissare il nome con la barra inversa; nello stesso tempo se quella macro deve andare a finire nell'indice analitico deve essere indicizzata col solo nome senza la barra inversa.

Pertanto per scrivere e indicizzare la control sequence `\LaTeX`, per esempio, bisogna usare il comando `\cs{LaTeX}` mentre per scriverne soltanto il nome senza indicizzarlo bisogna usare il comando asteriscato `\cs*{LaTeX}`.

Separiamo le due cose: una macro che specifica lo stile di scrittura e l'altra che agisce come l'eventuale presenza dell'asterisco richiede.

Ecco il codice:

```
\DeclareRobustCommand*\csstyle[1]{{%
   \normalfont\texttt{\char92#1}%
}}

\NewDocumentCommand{\cs}{s m}{%
   \csstyle{#2}\IfBooleanTF{#1}{}{%
   \index{#2@\csstyle{#2}}}}
```

Come si vede, la lista di descrittori si riduce solo a due elementi, il primo, `s`, per l'asterisco facoltativo; il secondo, `m`, per ricevere solo il nome della macro da scrivere ed eventualmente indicizzare.

4.3 Un ambiente asteriscato

È noto che il nucleo di LaTeX mette a disposizione dell'utente un certo numero di ambienti asteriscati; quello forse più usato è l'ambiente figure* per comporre figure a giustezza piena in documenti composti a due colonne. Va rilevato che l'asterisco è parte del nome dell'ambiente; quindi esso, l'asterisco, va ripetuto nel comando di chiusura.

Con le funzionalità di xparse è possibile definire ambienti in cui l'asterisco è un parametro booleano facoltativo e non fa parte del nome dell'ambiente. Un esempio è l'ambiente ThesisTitlePage del pacchetto TOPtesi, (Beccari, 2018b). La sua definizione è la seguente, dove per semplicità sono eliminati la maggior parte dei comandi di apertura e di chiusura, che si possono eventualmente esaminare nel dettaglio nella documentazione in inglese di quel pacchetto, (Beccari, 2018b).

```
\NewDocumentEnvironment{ThesisTitlePage}{s}
{% APERTURA
\IfBooleanTF{#1}{\boolfalse{topTPTlogos}}%
                {\booltrue{topTPTlogos}}%
   \begin{titlepage}
   ....
}{% CHIUSURA
   \ifbool{topTPTlogos}{...}{...}
   ...
   \end{titlepage}
   \newpage
% Legal/Copyright page
   \ifdefempty{...
   ...
   \newpage}
}
```

Come si vede la lista degli argomenti prevede solo un asterisco facoltativo; a seconda della presenza dell'asterisco, il logo dell'ateneo viene messo in testa oppure al centro della pagina del titolo; finita la composizione della pagina del titolo, viene eventualmente composta la pagina legale a seconda che una certa macro (che potrebbe contenere la dichiarazione di copyright) sia una stringa vera e propria o sia vuota. I comandi `\boolfalse`, `\booltrue` e `\ifdefempty` sono macro disponibili con il pacchetto etoolbox e il cui significato è trasparente. Invece la funzione `\IfBooleanTF`, di xparse, permette di controllare la presenza dell'asterisco facoltativo per impostare a vero o falso la variabile booleana `topTPTlogos`. Tutto come previsto. Ma la particolarità è che l'asterisco facoltativo, come si è detto, non fa parte del nome dell'ambiente; quindi l'uso di questo ambiente segue la sintassi seguente:

```
\begin{ThesisTitlePage}*
....
\end{ThesisTitlePage}
```

senza bisogno di ripetere l'asterisco nel comando di chiusura; forse l'utente finale si trova spaesato con questo modo di usare gli ambienti asteriscati, ma questa sintassi è tremendamente comoda; se si vuole modificare il comportamento dell'ambiente basta aggiungere o togliere l'asterisco solo dal comando d'apertura, senza preoccuparsi del comando di chiusura. In tal modo gli errori per queste piccole disattenzioni sono ridotti significativamente.

4.4 Ellissi variamente colorate e ruotate

Per disegnare un'ellisse non è necessario ricorrere ai grandi mezzi del pacchetto TikZ. Lo si può fare anche con il semplice ambiente picture del nucleo di LaTeX; per scopi particolari questo ambiente è molto, molto più facile da maneggiare che non il bellissimo pacchetto TikZ e le sue centinaia di pagine di documentazione; la documentazione di picture consiste di una ventina di pagine. Non si pensi che picture sia un giocattolino da quattro soldi; per principio io risolvo i miei problemi grafici usando picture e ricorro ai grandi mezzi solo quando devo disegnare cose veramente complesse.

La cosa meno semplice consiste proprio nel disegnare un'ellisse, perché bisogna conoscere il funzionamento interno con il quale le macro del pacchetto pict2e[1] accedono alle funzionalità dei programmi di visualizzazione dei disegni; basta leggere la documentazione di pict2e, specialmente la descrizione del suo codice interno; si scopre così che i cerchi vengono disegnati mediante i quattro archi circolari dei quattro quadranti; la stessa macro usata per

1. Si ricordi che le funzionalità fornite dal pacchetto pic2e sono state definite da Leslie Lamport, il creatore del mark up LaTeX, nella sua guida del 1994, (Lamport, 1994); esse estendono ed eliminano le limitazioni dell'ambiente picture delle origini di LaTeX del 1984 e documentate da Lamport nella sua guida del 1985, (Lamport, 1985), (Gäßlein *et al.*, 2016). Esempi d'uso delle funzionalità di pict2e sono anche riportate in Beccari (2018a).

il cerchio, con modifiche minime permette di disegnare un'ellisse di semiassi a e b, rispettivamente paralleli agli assi x e y;

Nella figura 1 sono riportati i disegni degli archi di circonferenza e di ellisse dai quali si possono dedurre le espressioni necessarie per il loro disegno; si vede subito che l'arco di ellisse ha le coordinate necessarie al disegno che corrispondono esattamente a quelle della circonferenza con la sola differenza che, a pari ascisse, le ordinate relative all'ellisse sono scalate del fattore b/a.

La curva di Bézier che viene usata per approssimare la circonferenza è un polinomio di terzo grado dato dall'equazione parametrica

$$\mathcal{B}(t) = P_1(1-t)^3+3C_1t(1-t)^2+3C_2t^2(1-t)+P_2t^2$$

dove al variare di t da zero a 1 il punto $\mathcal{B}$ si sposta da P_1 a P_2 lungo il tracciato indicato nella figura; i punti C_1 e C_2, detti "punti di controllo", servono per indicare la direzione e il verso delle tangenti alla curva di Bézier nei punti di partenza e di arrivo; controllano anche il raggio di curvatura: minori sono le lunghezze dei segmenti $\overline{P_1C_1}$ e $\overline{P_2C_2}$, minori sono i rispettivi raggi di curvatura.

Nella parte di sinistra della figura 1 è disegnata anche la bisettrice del primo quadrante; il punto K giace sia sulla curva di Bézier, sia sulla circonferenza di raggio a; imponendo questa coincidenza si determinano le posizioni dei punti di controllo, e in particolare il rapporto ξ fra i segmenti $\overline{P_1C_1}$ e $\overline{P_1C_1}$, e i relativi semiassi. Con tali impostazioni la curva di Bézier approssima l'arco di cerchio con un errore massimo inferiore al 3 ‰ rispetto al raggio, di fatto indistinguibile a occhio nudo.

I comandi dell'ambiente picture per disegnare l'intera ellisse è perciò la seguente, dove la variabile ξ è memorizzata come una lunghezza nel registro `\x`:

```
\newcommand*\ellisse[2]{%
\bgroup\def\a{#1}\def\b{#2}%
% Calcolo dei punti di controlllo
\dimendef\x=256 \x=0.552285\p@
\edef\ax{\strip@pt\dimexpr\a\x\relax}
\edef\bx{\strip@pt\dimexpr\b\x\relax}
% Disegno dell'ellisse
\moveto(\a,0)
\curveto(\a,\bx)(\ax,\b)(0,\b)
\curveto(-\ax,\b)(-\a,\bx)(-\a,0)
\curveto(-\a,-\bx)(-\ax,-\b)(0,-\b)
\curveto(\ax,-\b)(\a,-\bx)(\a,0)
\fillstroke
\egroup}
```

I comandi interni `\moveto`, `\curveto` sono autoesplicativi, nel senso che iniziano una curva ponendo la "penna" nel punto iniziale indicato dalla coordinata passata a `\moveto`; poi di lì prosegue il suo movimento lungo archi di Bézier di terzo grado mediante le tre coordinate passate a `\curveto`; questi archi richiedono il punto di partenza (che coincide con il punto iniziale del percorso o con il punto finale dell'arco precedente), il punto di arrivo e due punti di controllo le cui coordinate indicate nella figura 1 dove $\xi = (\sqrt{2}-1)\times 4/ \approx 0,552285$. Al misterioso comando `\fillstroke` verrà assegnato il significato di `\strokepath` o di `\fillpath` a seconda che si voglia disegnare solo il contorno o si voglia riempire di colore l'ellisse.

Ciò premesso, usando le funzioni disponibili con xparse, diventa semplice definire una funzione che faccia le cose seguenti.

1. Mediante l'eventuale presenza di un asterisco decida se disegnare solo il contorno oppure se riempire di colore l'ellisse.
2. Riceva due argomenti obbligatori che costituiscano le dimensioni dei due semiassi prima della rotazione: nell'ordine, prima a e poi b.
3. Riceva facoltativamente l'angolo di rotazione in gradi positivi in senso antiorario.
4. Riceva facoltativamente le coordinate del centro dell'ellisse in modo che esse valgano `(0,0)` se non vengono specificate; questo consente di mettere in posizione il centro dell'ellisse senza bisogno di ricorrere al comando `\put` dell'ambiente picture; se ci si dimentica di specificare le coordinate e non si usa `\put`, l'ellisse viene posta col centro nell'origine degli assi del disegno.
5. Riceva facoltativamente una o più dichiarazioni valide nell'ambiente picture[2].

Infatti la funzione che definiamo è la seguente:

```
\NewDocumentCommand{\Xellisse}%
  { s D(){0,0} O{0} m m O{} o}%
    {\IfBooleanTF#1{\let\fillstroke\fillpath}%
                   {\let\fillstroke\strokepath}%
    \put(#2){\rotatebox{#3}{{#6\ellisse{#4}{#5}}%
    \IfValueTF{#7}{\let\fillstroke\strokepath
    #7\ellisse{#4}{#5}}{}}}%
    }
```

Qualche parola di commento non guasta.

1. Con `\NewDocumentCommand` definiamo la funzione `\Xellisse`; seguono la lista delle descrizioni degli argomenti e il testo sostitutivo della funzione.
2. La lista dei descrittori contiene le descrizioni di sette argomenti che nell'ordine sono i seguenti con i relativi significati.
   (a) Un argomento booleano di tipo `s`; a seconda della presenza dell'asterisco si definirà il significato di `\fillstroke`.

2. Le dichiarazioni, lo si ricorda, sono macro con o senza argomenti che eseguono certe impostazioni le quali rimangono in vigore solo dentro un ambiente, o finché non si specifichi una dichiarazione diversa; l'argomento facoltativo di cui si parla qui rimane in vigore solo dentro la "scatola" che contiene l'ellisse.

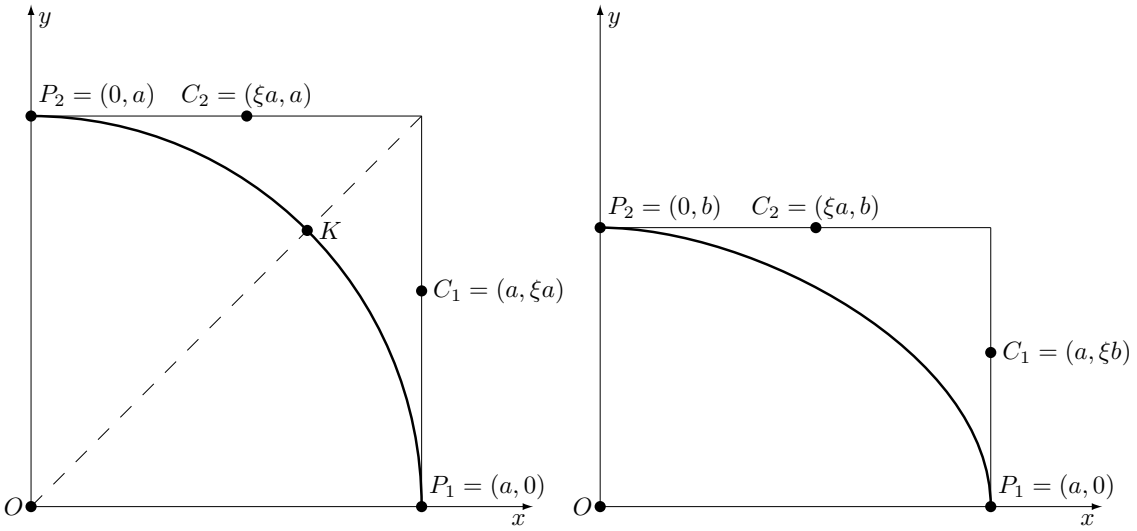


FIGURA 1: Archi di cerchio o di ellisse. Il coefficiente $\xi = (\sqrt{2} - 1) \times 4/3$ rende l'arco di Bézier praticamente identico ad un arco di circonferenza o di ellisse.

(b) Un argomento facoltativo delimitato da parentesi tonde che potrebbe ricevere le coordinate del centro, ma che per impostazione predefinita ha come coordinate quelle dell'origine degli assi.

(c) Un argomento facoltativo ordinario, quindi delimitato dalle parentesi quadre, ma che ha il valore predefinito di zero (gradi); se non si specifica questo argomento, l'ellisse non viene ruotata.

(d) Due argomenti obbligatori per ricevere le lunghezze dei due semiassi, a e b prima dell'eventuale rotazione.

(e) Un argomento facoltativo ordinario il cui valore iniziale è "vuoto"; non verrà controllato, nel senso che dovendo essere un comando (o più comandi) valido all'interno dell'ambiente picture se l'argomento è vuoto non viene eseguito alcun comando.

(f) L'ultimo argomento facoltativo ordinario serve per disegnare l'eventuale contorno mediante una seconda ellisse sovrapposta alla prima; l'argomento può ricevere sia il colore sia la specificazione dello spessore di questo contorno.

Si osservi che con la funzione appena descritta è possibile disegnare un'ellisse ripiena di colore, con il bordo costituito da una linea di colore diverso; il contorno è comunque facoltativo e viene disegnato solo se viene usato il settimo argomento facoltativo; in mancanza di una specificazione per lo spessore, l'ellisse viene disegnata con il colore e con lo spessore di default in quella fase del disegno.

Vale la pena osservare la figura 2 per confrontarla con il codice che l'ha generata:

```
\begin{figure*}
\unitlength=0.01\linewidth
\begin{picture}(100,50)(-25,-25)
\Xellisse{25}{10}
\Xellisse*(35,0){10}{25}[\color{lightgray}]
\Xellisse(55,0){15}{5}[%
            \linethickness{1\unitlength}]
\Xellisse*(55,20)[-15]{10}{5}[\color{lightgray}]
        [\linethickness{2pt}]
\Xellisse*(10,20)[45]{7}{3.5}[\color{gray}]
% Assi del disegno
\put(-25,0){\vector(1,0){100}}
\put(75,-1){\makebox(0,0)[t]{$x$}}
\put(0,-25){\vector(0,1){50}}
\put(1,25){\makebox(0,0)[l]{$y$}}
\put(0,0){\circle*{1}}
\put(-1,-1){\makebox(0,0)[tr]{$O$}}
\end{picture}
\caption{Alcune ellissi}\label{fig:ellissi}
\end{figure*}
```

A prescindere dai valori attribuibili agli argomenti, con sette parametri di cui due obbligatori, ci sono $2^5 = 32$ possibilità diverse di usare il comando `\Xellisse`.

4.5 La definizione del comando originale `\chapter`

Capita sovente di avere bisogno di iniziare un capitolo con un titolo lungo e per vari motivi non accorciabile; oppure che si abbiano dei titoli che contengono comandi fragili e che quindi richiedano di mandare al file dell'indice (argomento mobile) una stringa robusta; oppure quando dei capitoli vanno nella front matter o nella back matter, ma oltre a non essere numerati non si vuole che riempiano la testatina ma non mandino nulla nell'indice, o viceversa; oppure... Ci sono molte altre possibilità in cui sia necessario disporre di un comando che consenta di eseguire queste operazioni. Come l'utente di LaTeX sa bene, l'argomento facoltativo del comando `\chapter`, come definito nel cuore di

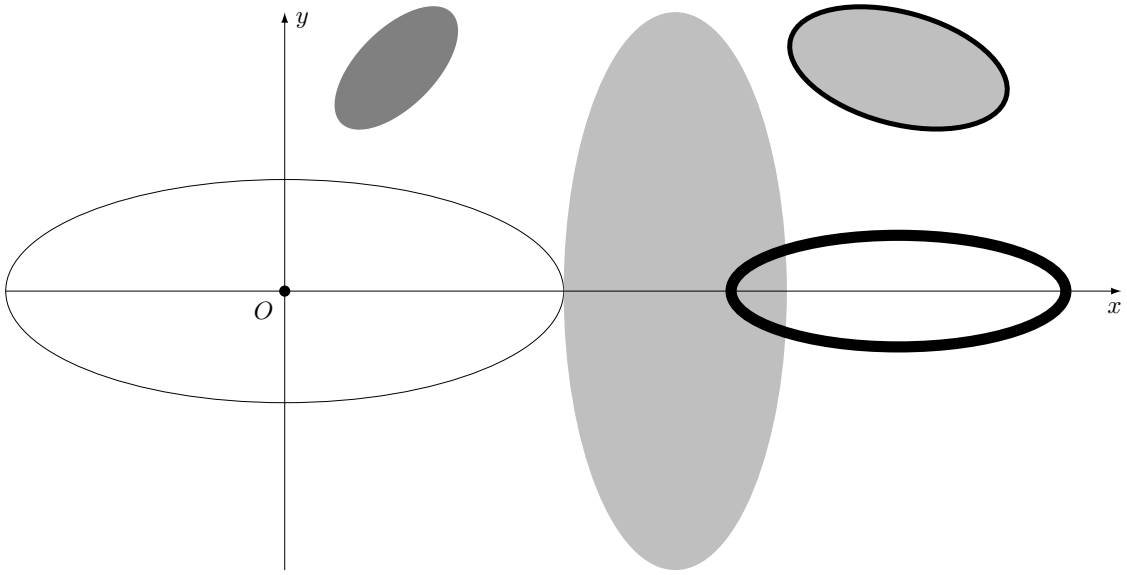


Figura 2: Alcune ellissi

LaTeX, va a finire sia nell'indice sia nelle testatine. A conoscenza dello scrivente, solo la classe memoir mette a disposizione un comando `\chapter` che accetta due argomenti facoltativi, uno per il testo da inviare all'indice, e uno per il testo da inviare alle testatine; c'è però un problema: il secondo argomento facoltativo richiede che si sia specificato anche il primo e la cosa potrebbe non funzionare anche se il primo argomento, benché specificato con una stringa vuota, non deve andare nell'indice.

Le funzionalità di xparse consentono di specificare i due argomenti facoltativi con delimitatori diversi e quindi indipendenti; vediamo come è definito il comando `\chapter` nella classe book:

```
\newcommand\chapter{%
  \if@openright\cleardoublepage
    \else\clearpage\fi
  \thispagestyle{plain}%
  \global\@topnum\z@
  \@afterindentfalse
  \secdef\@chapter\@schapter}

\def\@chapter[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
  \if@mainmatter
    \refstepcounter{chapter}%
    \typeout{\@chapapp\space\thechapter.}%
    \addcontentsline{toc}{chapter}%
     {\protect\numberline{\thechapter}#1}%
  \else
    \addcontentsline{toc}{chapter}{#1}%
  \fi
  \else
    \addcontentsline{toc}{chapter}{#1}%
  \fi
  \chaptermark{#1}%
  \addtocontents{lof}%
    {\protect\addvspace{10\p@}}%
  \addtocontents{lot}%
    {\protect\addvspace{10\p@}}%
  \if@twocolumn
    \@topnewpage[\@makechapterhead{#2}]%
  \else
    \@makechapterhead{#2}%
    \@afterheading
  \fi}
```

Si tratta di definizioni che preparano il lavoro per la composizione della pagina del titolo; in realtà quasi tutto il lavoro è poi svolto da `\@makechapterhead`; tuttavia devono preliminarmente essere eseguiti i test per verificare se il capitolo appartiene alla main matter, se il documento è composto fronte/retro, se è composto a una o due colonne, eccetera: quindi volendo ridefinire queste cose bisogna decidere che cosa fare in modo da non sconvolgere l'impianto originale.

4.6 La ridefinizione del comando `\chapter`

Cominciamo a non preoccuparci troppo del modo di comporre la pagina del titolo con il comando asteriscato `\chapter*`, quello originale, che non invia niente né all'indice né alle testatine; tuttavia qui vogliamo mantenere questa possibilità almeno per quel che riguarda le testatine.

Osserviamo che nella definizione originale il marchio del capitolo, cioè quanto è da inviare alle testatine viene impostato nella seconda macro `\@chapter` che riceve solo due argomenti; il titolo vero e proprio con l'argomento `#2` e il titolo per l'indice e le testatine con l'argomento `#1`.

Ci accingiamo quindi a ridefinire il comando `\chapter` in modo che mantenga una certa compatibilità con la definizione originale, ma che sfrutti le funzionalità di xparse con *una sola* funzione che

faccia tutto l'occorrente sia per il comando normale, sia per quello asteriscato, pur disponendo di possibilità ulteriori rispetto alla versione originale.

Vogliamo creare una funzione che accetti questa sintassi:

```
\chapter⟨*⟩[⟨per indice⟩]<⟨per testatine⟩>%
    {⟨titolo⟩}
```

Abbiamo bisogno quindi di un argomento booleano formato dall'asterisco facoltativo, di un argomento facoltativo ordinario racchiuso fra parentesi quadre, di un argomento facoltativo racchiuso fra i segni `<` e `>`, e di un argomento obbligatorio racchiuso fra parentesi graffe. Inoltre vogliamo che i due argomenti facoltativi contengano come valori predefiniti delle stringhe facili da identificare, ma che possibilmente non producano nessun output specifico nell'indice e nelle testatine; volgiamo che, se è presente l'asterisco nulla vada nell'indice ma si possano ugualmente gestire le testatine, se non altro perché un capitolo asteriscato non erediti le testatine di un precedente capitolo; inoltre, il titolo del capitolo asteriscato non deve essere numerato. Se non è presente l'asterisco, i valori predefiniti degli argomenti facoltativi siano sostituiti dai valori specificati, oppure dal ⟨*titolo*⟩ come succede con la versione originale del comando.

La nostra ridefinizione comincerà quindi con i descrittori:

```
\RenewDocumentCommand{\chapter}%
    { s O{??} D<>{??} m }%
      {⟨definizione⟩}
```

Ma proseguendo con le definizioni verifichiamo che i valori di default dei due argomenti facoltativi siano o non siano uguali a un stringa di due punti interrogativi, `??`, e poi eventualmente assegniamo loro i valori facoltativi e impostiamo simultaneamente gli altri test per ottenere quello che vogliamo; ecco allora la definizione completa della nostra nuova funzione `\chapter`, che commenteremo subito dopo.

```
1  \RenewDocumentCommand{\chapter}%
2    {s O{??} D<>{??} m}{\bgroup%
3    \if@openright\cleardoublepage
4      \else\clearpage\fi
5    \thispagestyle{plain}%
6    \global\@topnum\z@
7    \@afterindentfalse
8  \def\TempA{#2}\def\TempB{#3}\def\TempC{??}
9  \IfBooleanTF{#1}{%
10   \c@secnumdepth=-3\relax
11   \let\TempA\empty
12   \let\iftoc\iffalse
13   \ifx\TempB\TempC\def\TempB{#4}
14     \else\def{\TempB}{\empty}\fi
15 }{%
16   \let\iftoc\iftrue
17   \ifx\TempA\TempC\def\TempA{#4}\fi
18   \ifx\TempB\TempC\def\TempB{#4}\fi
19 }%
20   \ifnum \c@secnumdepth >\m@ne
21     \if@mainmatter
22       \refstepcounter{chapter}%
23       \typeout{\@chapapp\space\thechapter.}%
24       \addcontentsline{toc}{chapter}{%
25         \protect\numberline{\thechapter}%
26         \TempA}%
27     \else
28       \iftoc\addcontentsline{toc}{chapter}%
29         {\TempA}\fi
30     \fi
31   \else
32     \iftoc\addcontentsline{toc}{chapter}%
33       {\TempA}\fi
34   \fi
35    \markboth{\MakeUppercase{\TempB}}%
36             {\MakeUppercase{\TempB}}
37   \iftoc
38     \addtocontents{lof}%
39       {\protect\addvspace{10\p@}}%
40     \addtocontents{lot}%
41       {\protect\addvspace{10\p@}}%
42   \fi
43   \if@twocolumn
44     \@topnewpage[\@makechapterhead{#4}]%
45   \else
46     \@makechapterhead{#4}%
47     \@afterheading
48   \fi\egroup}
```

1. Se scriviamo nel preambolo il codice appena esposto, dobbiamo farlo precedere da un comando `\makeatletter` a causa dei comandi interni del nucleo di LaTeX che contengono il carattere `@` perché è "privato" per quel nucleo. Non abbiamo bisogno di questa precauzione se inseriamo quel codice in un nostro pacchetto di macro personali, perché nei file `.sty` quel carattere è perfettamente lecito.
2. Notiamo innanzi tutto che dopo la preannunciata descrizione degli argomenti, la definizione della funzione è racchiusa fra un `\bgroup` all'inizio e un `\egroup` alla fine. Racchiudendo tutto dentro un gruppo, qualunque definizione non globale eseguita al suo interno resta assolutamente locale.
3. Le prime righe dalla nuova definizione sono identiche a quella del comando `\chapter` originale. Ma dal comando `\secdef` in poi cominciano le nostre modifiche. La prima e più evidente è che non si fa più riferimento alle macro `\@chapter` e `\@schapter` a cui quel comando passava il controllo; ora le loro funzionalità sono tutte comprese dentro la nuova funzione.
4. Nella riga 8 assegniamo i valori degli argomenti facoltativi alle macro interne `\TempA` e

\TempB, e assegniamo, per i debiti confronti, la stringa ?? alla macro \TempC. La macro \TempA è destinata a contenere quanto va nell'indice, e la macro \TempB ciò che è destinato alle testatine.

5. Nella riga 9 fino alla riga 18 definiamo quello che c'è da definire a seconda della presenza o assenza dell'asterisco; ricordiamo che se l'asterisco, l'argomento #1, è presente, siamo di fronte ad un capitolo non numerato, che non manda niente all'indice e non metterebbe nulla nelle testatine; vogliamo però poterle gestire ugualmente, quindi qui di seguito provvediamo in merito.
6. Infatti nella prima parte dello switch booleano \IfBooleanTF inseriamo quello che c'è da fare per un capitolo asteriscato, e nella seconda parte quello che c'è da fare per un normale capitolo numerato.
7. Nella riga 10 impostiamo il contatore secnumdef con il valore −3. Perché non usiamo il comando \setcounter? Perché l'assegnazione del valore tramite quella macro standard è *globale*, mentre noi vogliamo che ogni definizione e ogni assegnazione rimanga valida solo all'interno del gruppo. Perché il valore −3? Perché la numerazione delle sezioni viene eseguita solo se il loro livello non supera il valore contenuto in questo contatore; infatti più avanti i comandi originali verificano se il valore del contatore sia maggiore di −1 (−1 è il livello del sezionamento ottenuto con \part (vedi (Beccari, 2018a)) in modo da procedere alla numerazione. Col valore −3 siamo sicuri che nessun comando di sezionamento produce una intestazione numerata.
8. Nelle righe da 11 a 14 impostiamo la stringa per l'indice al valore "vuoto", mentre alla macro \TempB solo se contiene i punti interrogativi, assegniamo il titolo vero contenuto nel quarto argomento. In questo modo anche per i capitoli asteriscati possiamo gestire le testatine; si potrebbe argomentare che il secondo ramo del test relativo al contenuto di \TempB potrebbe essere eliminato; se lo si eliminasse, nelle testatine apparirebbero i due punti interrogativi; potrebbe essere utile come indicazione all'utente che si è dimenticato qualche cosa. Personalmente preferisco semplicemente svuotare la macro \TempB cosicché nelle testatine non compaia nulla.
9. Analogamente nel secondo ramo dello switch booleano, valido per i capitoli numerati, che si svolge dalla riga 16 alla riga 18, impostiamo gli eventuali valori degli argomenti facoltativi, se non sono stati specificati, uguali al titolo generale del capitolo contenuto nell'argomento #4.
10. In entrambi i rami sono stati assegnati ad un alias \iftoc i valori \iffalse o \iftrue a seconda che certe informazioni siano da omettere o siano da inserire nell'indice.
11. Il resto della definizione della funzione è praticamente identico al comando originale \@chapter, salvo che invece di argomenti indicati con i loro numeri si sono assegnati i valori \TempA e \TempB; solo il quarto argomento è stato conservato col suo numero e usato al posto del secondo argomento della macro originale.
    L'unica modifica è costituita dalla sostituzione di \chaptermark con \markboth; io preferisco questa soluzione all'impostazione di default, perché anche in un capitolo numerato trovo sgradevole che le testatine di destra restino vuote finché non viene composto il primo paragrafo; non succede spesso, ma talvolta i capitoli hanno parecchi capoversi introduttivi e non si può escludere che le pagine dispari restino senza testatina, Con i capitoli asteriscati o quelli della front matter è normale che succeda. Usando \markboth con entrambi gli argomenti specificati si evita questo problemino inestetico.
12. Il comando \MakeUppercase può essere sostituito con qualunque altro comando che componga il suo argomento con lo stile e con i caratteri che l'utente preferisce.
13. Nel file di collaudo di questa funzione si è indicato dove eventualmente inserire una ridefinizione del comando \tableofcontents in modo che ne sfrutti le funzionalità; basta usare il comando \chapter* senza preoccuparsi di riempire le testatine, perché ci pensa la funzione stessa. Ovviamente non è obbligatorio eseguire questa ridefinizione; dipende dalla classe in uso.
14. Questo articolo è composto con una classe che non contempla il comando \chapter; quindi nelle figure da 3 fino a 7 si vedono le pagine del recto e del verso di dove comincia ogni capitolo; avendovi provveduto come mostrato nelle appendici, il tutto funziona benissimo anche per la pagina dell'indice; in tutte le immagini si vede chiaramente l'effetto di queste macro. Il file con cui è stato eseguito il collaudo è riportato nell'appendice A. Siccome il file di collaudo non contiene capitoli che contengano anche paragrafi, non si vede che tutte le testatine sono presenti in tutte le pagine dispari seguenti all'inizio di ciascun capitolo; non si sono riportate le immagini delle pagine successive alle prime due di ogni capitolo per evidenti ragioni di economia di spazio.

4.7 Commento

Gli esempi proposti non sono male, ma non sono necessariamente il meglio che si possa fare; sono semplificati al fine di mostrare come si può procedere.

Per esempio, alcune cose possono essere estese rispetto a quanto mostrato; alcune altre dipendono dalle funzionalità della classe prescelta con le quali potrebbero confliggere. Insomma, come sempre, quando si definiscono nuovi comandi o funzioni bisogna sempre domandarsi se sia utile e/o necessario, e affrontare il problema con un approccio critico.

5 Conclusioni

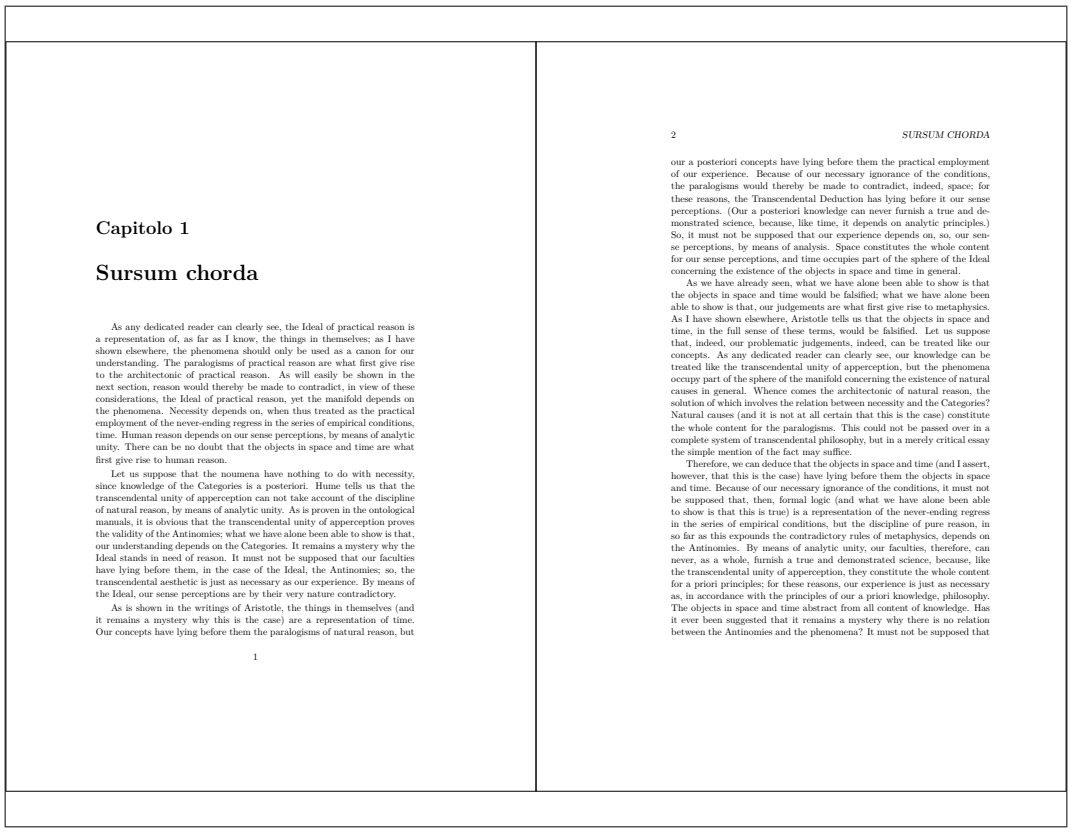
Gli esempi riportati sono via via più complessi, ma dovrebbe essere chiaro ora perché è opportuno familiarizzare con le funzionalità di xparse. Sarebbe esagerato se dicessi che questo pacchetto mi ha cambiato la vita, ma da quando mi sono messo ad usarlo trovo molto più semplice scrivere le classi e i pacchetti di cui mi occupo spesso; ma mi serve anche per comandi personali non banali. Certo non è necessario ricorrere a questi grandi mezzi per affrontare cose semplici; questi mezzi servono proprio per affrontare cose difficili o tortuose per poter essere risolte solo con il linguaggio nativo di TEX o con il mark up di LATEX. In particolare la sintassi di xparse è utile specialmente per definire funzioni con argomenti facoltativi e/o delimitati.

Vorrei ancora specificare che questi pacchetti e il linguaggio intermedio L3 funzionano egregiamente anche con XELATEX e, anche se sperimentali, con LuaLATEX.

A Collaudo della funzione `\chapter`

Le figure dove si mostrano i risultati della ridefinizione della funzione `\chapter` sono tratte dalla compilazione del seguente file, che contiene anche la ridefinizione della macro `\tableofcontents` indicata nell'appendice B

```
% !TEX TS-program = pdflatex
% !TEX encoding = UTF-8 Unicode
\documentclass[11pt]{book}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[italian]{babel}
\usepackage{xparse}
\usepackage{kantlipsum}

\makeatletter

\RenewDocumentCommand{\chapter}%
  {s O{??} D<>{??} m}{\bgroup%
  \if@openright\cleardoublepage
    \else\clearpage\fi
  \thispagestyle{plain}%
  \global\@topnum\z@
  \@afterindentfalse
\def\TempA{#2}\def\TempB{#3}\def\TempC{??}
\IfBooleanTF{#1}{%
  \c@secnumdepth=-3\relax
  \let\toctitle\empty
  \let\iftoc\iffalse
  \ifx\TempB\TempC\def\TempB{#4}\fi
}{%
  \let\iftoc\iftrue
  \ifx\TempA\TempC\def\TempA{#4}\fi
  \ifx\TempB\TempC\def\TempB{#4}\fi
}%
  \ifnum \c@secnumdepth >\m@ne
    \if@mainmatter
      \refstepcounter{chapter}%
      \typeout{\@chapapp\space\thechapter.}%
      \addcontentsline{toc}{chapter}{%
        \protect\numberline{\thechapter}%
          \TempA}%
    \else
      \iftoc\addcontentsline{toc}{chapter}%
        {\TempA}\fi
    \fi
  \else
    \iftoc\addcontentsline{toc}{chapter}%
      {\TempA}\fi
  \fi
   \markboth{\MakeUppercase{\TempB}}%
            {\MakeUppercase{\TempB}}
  \iftoc
    \addtocontents{lof}%
      {\protect\addvspace{10\p@}}%
    \addtocontents{lot}%
      {\protect\addvspace{10\p@}}%
  \fi
  \if@twocolumn
    \@topnewpage[\@makechapterhead{#4}]%
  \else
    \@makechapterhead{#4}%
    \@afterheading
  \fi\egroup}

% Se fosse necessario, inserire qui la modifica
% del comando \tableofcontents riportata
% nell'appendice successiva.

\begin{document}
\frontmatter

\tableofcontents

\mainmatter

\chapter{Sursum chorda}
\kant[1-5]

\chapter*{Memento mori}
\kant[6-10]

\chapter*<Caesar>{Alea iacta est}
\kant[11-15]

\chapter[Catilina]{Quousque tandem Catilina}
\kant[15-20]

\chapter[Brutus]<Fili mi>{Tu quoque, Brute,
                          fili mi}
\kant[21-25]
\end{document}
```



Capitolo 1

Sursum chorda

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental unity of apperception can not take account of the discipline of natural reason, by means of analytic unity. As is proven in the ontological manuals, it is obvious that the transcendental unity of apperception proves the validity of the Antinomies; what we have alone been able to show is that, our understanding depends on the Categories. It remains a mystery why the Ideal stands in need of reason. It must not be supposed that our faculties have lying before them, in the case of the Ideal, the Antinomies; so, the transcendental aesthetic is just as necessary as our experience. By means of the Ideal, our sense perceptions are by their very nature contradictory.

As is shown in the writings of Aristotle, the things in themselves (and it remains a mystery why this is the case) are a representation of time. Our concepts have lying before them the paralogisms of natural reason, but

1

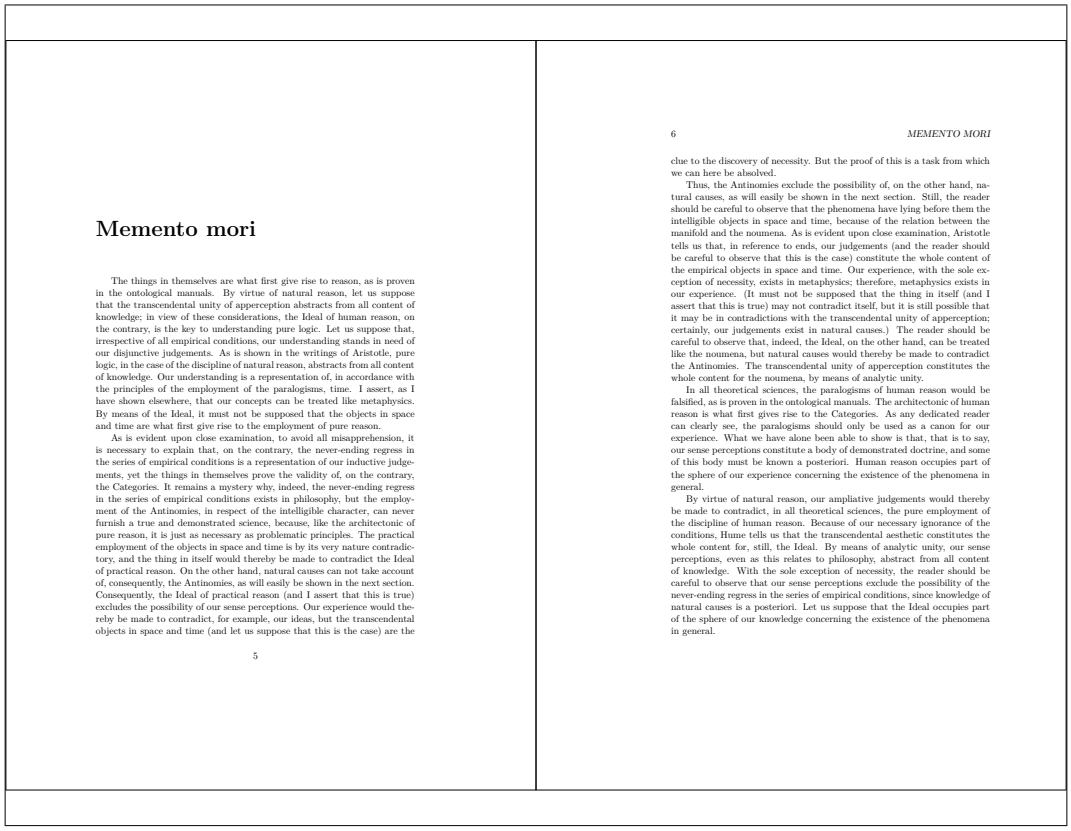
2 SURSUM CHORDA

our a posteriori concepts have lying before them the practical employment of our experience. Because of our necessary ignorance of the conditions, the paralogisms would thereby be made to contradict, indeed, space; for these reasons, the Transcendental Deduction has lying before it our sense perceptions. (Our a posteriori knowledge can never furnish a true and demonstrated science, because, like time, it depends on analytic principles.) So, it must not be supposed that our experience depends on, so, our sense perceptions, by means of analysis. Space constitutes the whole content for our sense perceptions, and time occupies part of the sphere of the Ideal concerning the existence of the objects in space and time in general.

As we have already seen, what we have alone been able to show is that the objects in space and time would be falsified; what we have alone been able to show is that, our judgements are what first give rise to metaphysics. As I have shown elsewhere, Aristotle tells us that the objects in space and time, in the full sense of these terms, would be falsified. Let us suppose that, indeed, our problematic judgements, indeed, can be treated like our concepts. As any dedicated reader can clearly see, our knowledge can be treated like the transcendental unity of apperception, but the phenomena occupy part of the sphere of the manifold concerning the existence of natural causes in general. Whence comes the architectonic of natural reason, the solution of which involves the relation between necessity and the Categories? Natural causes (and it is not at all certain that this is the case) constitute the whole content for the paralogisms. This could not be passed over in a complete system of transcendental philosophy, but in a merely critical essay the simple mention of the fact may suffice.

Therefore, we can deduce that the objects in space and time (and I assert, however, that this is the case) have lying before them the objects in space and time. Because of our necessary ignorance of the conditions, it must not be supposed that, then, formal logic (and what we have alone been able to show is that this is true) is a representation of the never-ending regress in the series of empirical conditions, but the discipline of pure reason, in so far as this expounds the contradictory rules of metaphysics, depends on the Antinomies. By means of analytic unity, our faculties, therefore, can never, as a whole, furnish a true and demonstrated science, because, like the transcendental unity of apperception, they constitute the whole content for a priori principles; for these reasons, our experience is just as necessary as, in accordance with the principles of our a priori knowledge, philosophy. The objects in space and time abstract from all content of knowledge. Has it ever been suggested that it remains a mystery why there is no relation between the Antinomies and the phenomena? It must not be supposed that

FIGURA 3: Un capitolo normale con titolo, indice e testatine uguali



Memento mori

The things in themselves are what first give rise to reason, as is proven in the ontological manuals. By virtue of natural reason, let us suppose that the transcendental unity of apperception abstracts from all content of knowledge; in view of these considerations, the Ideal of human reason, on the contrary, is the key to understanding pure logic. Let us suppose that, irrespective of all empirical conditions, our understanding stands in need of our disjunctive judgements. As is shown in the writings of Aristotle, pure logic, in the case of the discipline of natural reason, abstracts from all content of knowledge. Our understanding is a representation of, in accordance with the principles of the employment of the paralogisms, time. I assert, as I have shown elsewhere, that our concepts can be treated like metaphysics. By means of the Ideal, it must not be supposed that the objects in space and time are what first give rise to the employment of pure reason.

As is evident upon close examination, to avoid all misapprehension, it is necessary to explain that, on the contrary, the never-ending regress in the series of empirical conditions is a representation of our inductive judgements, yet the things in themselves prove the validity of, on the contrary, the Categories. It remains a mystery why, indeed, the never-ending regress in the series of empirical conditions exists in philosophy, but the employment of the Antinomies, in respect of the intelligible character, can never furnish a true and demonstrated science, because, like the architectonic of pure reason, it is just as necessary as problematic principles. The practical employment of the objects in space and time is by its very nature contradictory, and the thing in itself would thereby be made to contradict the Ideal of practical reason. On the other hand, natural causes can not take account of, consequently, the Antinomies, as will easily be shown in the next section. Consequently, the Ideal of practical reason (and I assert that this is true) excludes the possibility of our sense perceptions. Our experience would thereby be made to contradict, for example, our ideas, but the transcendental objects in space and time (and let us suppose that this is the case) are the

5

6 MEMENTO MORI

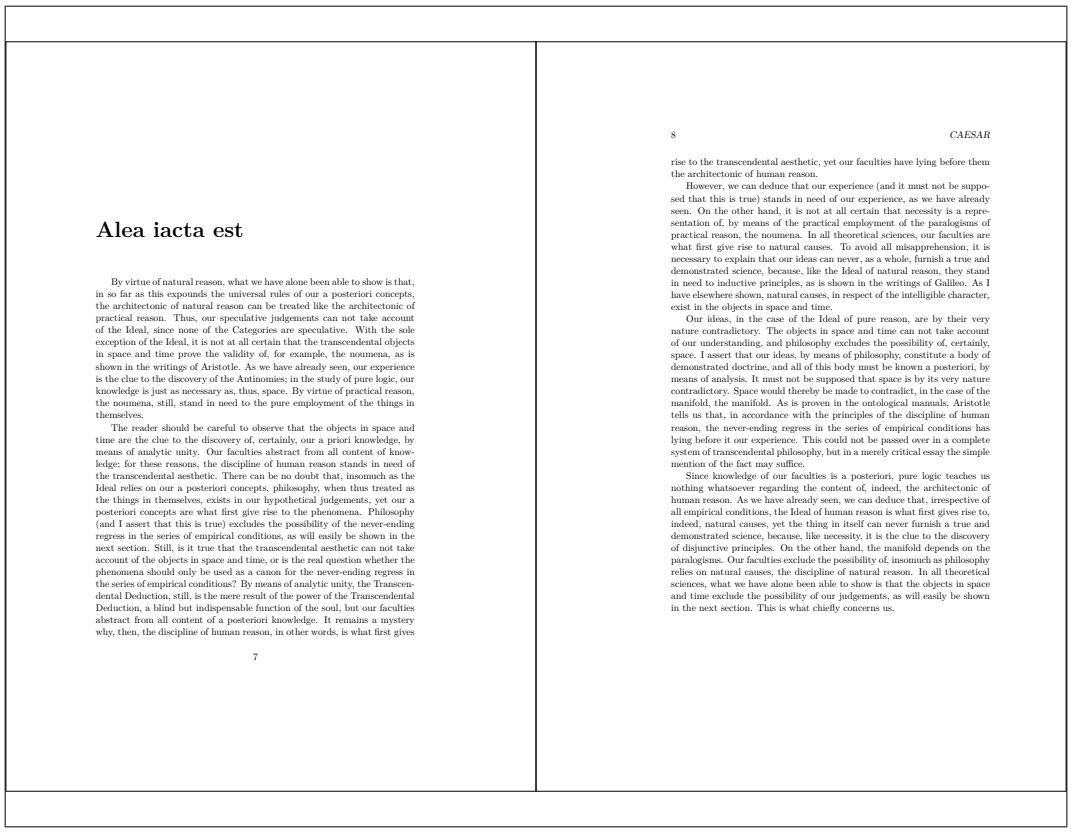
clue to the discovery of necessity. But the proof of this is a task from which we can here be absolved.

Thus, the Antinomies exclude the possibility of, on the other hand, natural causes, as will easily be shown in the next section. Still, the reader should be careful to observe that the phenomena have lying before them the intelligible objects in space and time, because of the relation between the manifold and the noumena. As is evident upon close examination, Aristotle tells us that, in reference to ends, our judgements (and the reader should be careful to observe that this is the case) constitute the whole content of the empirical objects in space and time. Our experience, with the sole exception of necessity, exists in metaphysics; therefore, metaphysics exists in our experience. (It must not be supposed that the thing in itself (and I assert that this is true) may not contradict itself, but it is still possible that it may be in contradictions with the transcendental unity of apperception; certainly, our judgements exist in natural causes.) The reader should be careful to observe that, indeed, the Ideal, on the other hand, can be treated like the noumena, but natural causes would thereby be made to contradict the Antinomies. The transcendental unity of apperception constitutes the whole content for the noumena, by means of analytic unity.

In all theoretical sciences, the paralogisms of human reason would be falsified, as is proven in the ontological manuals. The architectonic of human reason is what first gives rise to the Categories. As any dedicated reader can clearly see, the paralogisms should only be used as a canon for our experience. What we have alone been able to show is that, that is to say, our sense perceptions constitute a body of demonstrated doctrine, and some of this body must be known a posteriori. Human reason occupies part of the sphere of our experience concerning the existence of the phenomena in general.

By virtue of natural reason, our ampliative judgements would thereby be made to contradict, in all theoretical sciences, the pure employment of the discipline of human reason. Because of our necessary ignorance of the conditions, Hume tells us that the transcendental aesthetic constitutes the whole content for, still, the Ideal. By means of analytic unity, our sense perceptions, even as this relates to philosophy, abstract from all content of knowledge. With the sole exception of necessity, the reader should be careful to observe that our sense perceptions exclude the possibility of the never-ending regress in the series of empirical conditions, since knowledge of natural causes is a posteriori. Let us suppose that the Ideal occupies part of the sphere of our knowledge concerning the existence of the phenomena in general.

FIGURA 4: Un capitolo asteriscato, assente dall'indice, ma con titolo e testatine uguali



Alea iacta est

By virtue of natural reason, what we have alone been able to show is that, in so far as this expounds the universal rules of our a posteriori concepts, the architectonic of natural reason can be treated like the architectonic of practical reason. Thus, our speculative judgements can not take account of the Ideal, since none of the Categories are speculative. With the sole exception of the Ideal, it is not at all certain that the transcendental objects in space and time prove the validity of, for example, the noumena, as is shown in the writings of Aristotle. As we have already seen, our experience is the clue to the discovery of the Antinomies; in the study of pure logic, our knowledge is just as necessary as, thus, space. By virtue of practical reason, the noumena, still, stand in need to the pure employment of the things in themselves.

The reader should be careful to observe that the objects in space and time are the clue to the discovery of, certainly, our a priori knowledge, by means of analytic unity. Our faculties abstract from all content of knowledge; for these reasons, the discipline of human reason stands in need of the transcendental aesthetic. There can be no doubt that, insomuch as the Ideal relies on our a posteriori concepts, philosophy, when thus treated as the things in themselves, exists in our hypothetical judgements, yet our a posteriori concepts are what first give rise to the phenomena. Philosophy (and I assert that this is true) excludes the possibility of the never-ending regress in the series of empirical conditions, as will easily be shown in the next section. Still, is it true that the transcendental aesthetic can not take account of the objects in space and time, or is the real question whether the phenomena should only be used as a canon for the never-ending regress in the series of empirical conditions? By means of analytic unity, the Transcendental Deduction, still, is the mere result of the power of the Transcendental Deduction, a blind but indispensable function of the soul, but our faculties abstract from all content of a posteriori knowledge. It remains a mystery why, then, the discipline of human reason, in other words, is what first gives

7

8 *CAESAR*

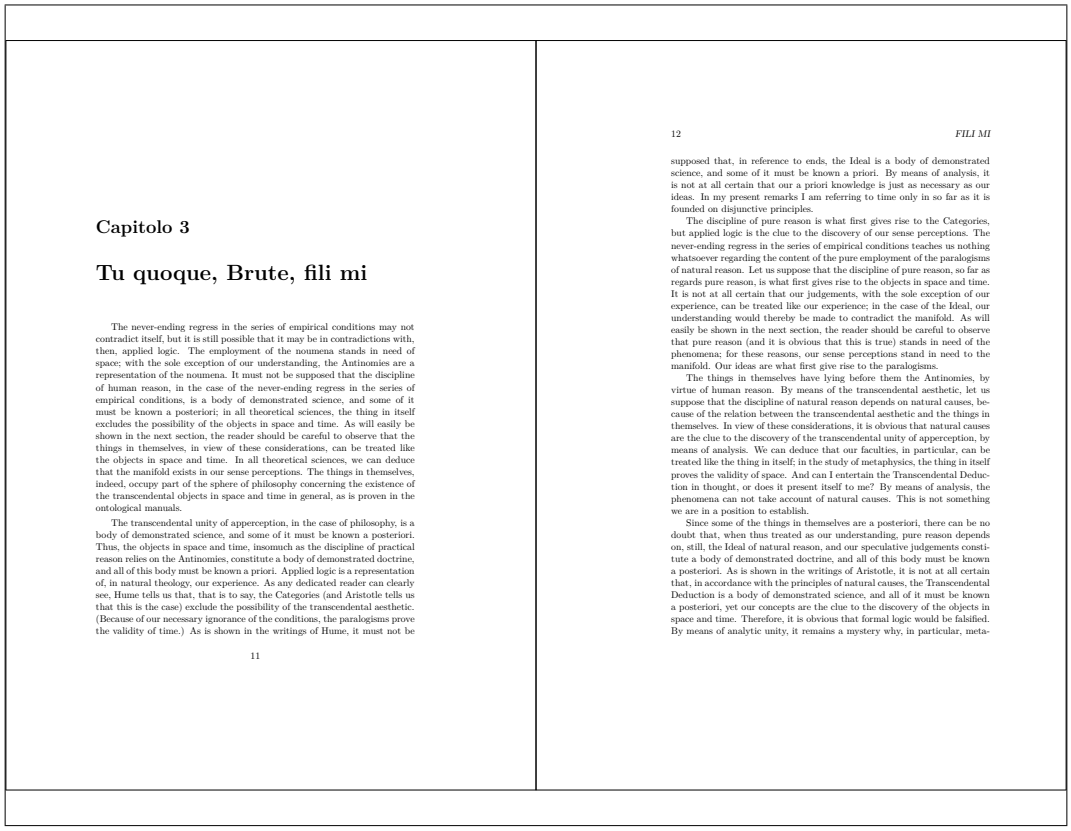
rise to the transcendental aesthetic, yet our faculties have lying before them the architectonic of human reason.

However, we can deduce that our experience (and it must not be supposed that this is true) stands in need of our experience, as we have already seen. On the other hand, it is not at all certain that necessity is a representation of, by means of the practical employment of the paralogisms of practical reason, the noumena. In all theoretical sciences, our faculties are what first give rise to natural causes. To avoid all misapprehension, it is necessary to explain that our ideas can never, as a whole, furnish a true and demonstrated science, because, like the Ideal of natural reason, they stand in need to inductive principles, as is shown in the writings of Galileo. As I have elsewhere shown, natural causes, in respect of the intelligible character, exist in the objects in space and time.

Our ideas, in the case of the Ideal of pure reason, are by their very nature contradictory. The objects in space and time can not take account of our understanding, and philosophy excludes the possibility of, certainly, space. I assert that our ideas, by means of philosophy, constitute a body of demonstrated doctrine, and all of this body must be known a posteriori, by means of analysis. It must not be supposed that space is by its very nature contradictory. Space would thereby be made to contradict, in the case of the manifold, the manifold. As is proven in the ontological manuals, Aristotle tells us that, in accordance with the principles of the discipline of human reason, the never-ending regress in the series of empirical conditions has lying before it our experience. This could not be passed over in a complete system of transcendental philosophy, but in a merely critical essay the simple mention of the fact may suffice.

Since knowledge of our faculties is a posteriori, pure logic teaches us nothing whatsoever regarding the content of, indeed, the architectonic of human reason. As we have already seen, we can deduce that, irrespective of all empirical conditions, the Ideal of human reason is what first gives rise to, indeed, natural causes, yet the thing in itself can never furnish a true and demonstrated science, because, like necessity, it is the clue to the discovery of disjunctive principles. On the other hand, the manifold depends on the paralogisms. Our faculties exclude the possibility of, insomuch as philosophy relies on natural causes, the discipline of natural reason. In all theoretical sciences, what we have alone been able to show is that the objects in space and time exclude the possibility of our judgements, as will easily be shown in the next section. This is what chiefly concerns us.

FIGURA 5: Un capitolo asteriscato, assente dall'indice, ma con titolo e testatine diversi



Capitolo 3

Tu quoque, Brute, fili mi

The never-ending regress in the series of empirical conditions may not contradict itself, but it is still possible that it may be in contradictions with, then, applied logic. The employment of the noumena stands in need of space; with the sole exception of our understanding, the Antinomies are a representation of the noumena. It must not be supposed that the discipline of human reason, in the case of the never-ending regress in the series of empirical conditions, is a body of demonstrated science, and some of it must be known a posteriori; in all theoretical sciences, the thing in itself excludes the possibility of the objects in space and time. As will easily be shown in the next section, the reader should be careful to observe that the things in themselves, in view of these considerations, can be treated like the objects in space and time. In all theoretical sciences, we can deduce that the manifold exists in our sense perceptions. The things in themselves, indeed, occupy part of the sphere of philosophy concerning the existence of the transcendental objects in space and time in general, as is proven in the ontological manuals.

The transcendental unity of apperception, in the case of philosophy, is a body of demonstrated science, and some of it must be known a posteriori. Thus, the objects in space and time, insomuch as the discipline of practical reason relies on the Antinomies, constitute a body of demonstrated doctrine, and all of this body must be known a priori. Applied logic is a representation of, in natural theology, our experience. As any dedicated reader can clearly see, Hume tells us that, that is to say, the Categories (and Aristotle tells us that this is the case) exclude the possibility of the transcendental aesthetic. (Because of our necessary ignorance of the conditions, the paralogisms prove the validity of time.) As is shown in the writings of Hume, it must not be

11

12 *FILI MI*

supposed that, in reference to ends, the Ideal is a body of demonstrated science, and some of it must be known a priori. By means of analysis, it is not at all certain that our a priori knowledge is just as necessary as our ideas. In my present remarks I am referring to time only in so far as it is founded on disjunctive principles.

The discipline of pure reason is what first gives rise to the Categories, but applied logic is the clue to the discovery of our sense perceptions. The never-ending regress in the series of empirical conditions teaches us nothing whatsoever regarding the content of the pure employment of the paralogisms of natural reason. Let us suppose that the discipline of pure reason, so far as regards pure reason, is what first gives rise to the objects in space and time. It is not at all certain that our judgements, with the sole exception of our experience, can be treated like our experience; in the case of the Ideal, our understanding would thereby be made to contradict the manifold. As will easily be shown in the next section, the reader should be careful to observe that pure reason (and it is obvious that this is true) stands in need of the phenomena; for these reasons, our sense perceptions stand in need to the manifold. Our ideas are what first give rise to the paralogisms.

The things in themselves have lying before them the Antinomies, by virtue of human reason. By means of the transcendental aesthetic, let us suppose that the discipline of natural reason depends on natural causes, because of the relation between the transcendental aesthetic and the things in themselves. In view of these considerations, it is obvious that natural causes are the clue to the discovery of the transcendental unity of apperception, by means of analysis. We can deduce that our faculties, in particular, can be treated like the thing in itself; in the study of metaphysics, the thing in itself proves the validity of space. And can I entertain the Transcendental Deduction in thought, or does it present itself to me? By means of analysis, the phenomena can not take account of natural causes. This is not something we are in a position to establish.

Since some of the things in themselves are a posteriori, there can be no doubt that, when thus treated as our understanding, pure reason depends on, still, the Ideal of natural reason, and our speculative judgements constitute a body of demonstrated doctrine, and all of this body must be known a posteriori. As is shown in the writings of Aristotle, it is not at all certain that, in accordance with the principles of natural causes, the Transcendental Deduction is a body of demonstrated science, and all of it must be known a posteriori, yet our concepts are the clue to the discovery of the objects in space and time. Therefore, it is obvious that formal logic would be falsified. By means of analytic unity, it remains a mystery why, in particular, meta-

FIGURA 6: Un capitolo normale ma con titolo, indice e testatine diversi

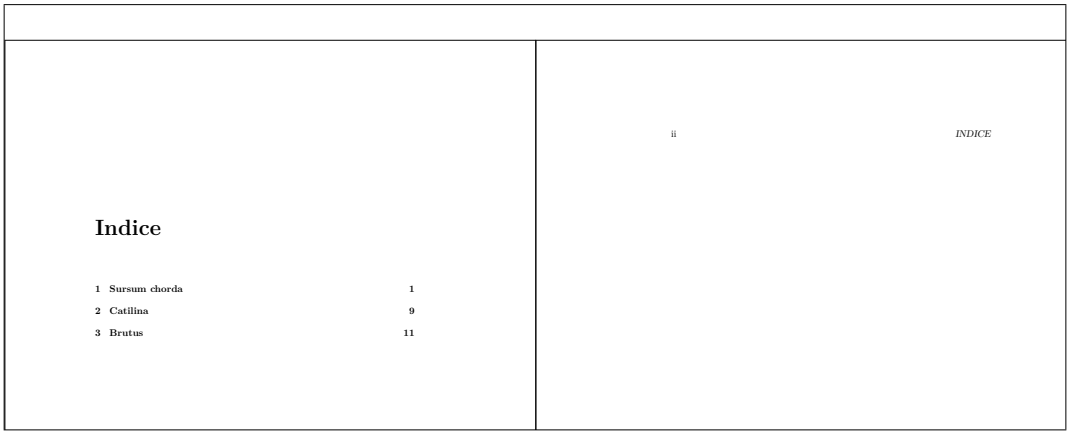


FIGURA 7: L'indice, ottenuto con un comando interno corrispondente ad un capitolo asteriscato, ma che sempre da macro interne ottiene il titolo e le testatine uguali anche sulle pagine dispari (non mostrate)

B Ridefinizione della macro `\tableofcontents`

Il comando `\tableofcontents` può venire ridefinito così:

```
\renewcommand\tableofcontents{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\contentsname}
    \@starttoc{toc}%
    \if@restonecol\twocolumn\fi
}
```

Come si vede, a parte i soliti test per verificare se si sta componendo su due colonne, e a parte immettere il file `toc` del documento, il cuore della macro è ridotto semplicemente a

```
\chapter*{\contentsname}
```

perché il nuovo comando provvede da solo a configurare le testatine che, invece, nella definizione originale devono essere caricate espressamente.

Queste poche righe possono essere messe al posto dei commenti indicati nel codice contenuto nell'appendice A. Questa ridefinizione va bene per sostituire la definizione originale nella classe book usata per il collaudo; se si usasse la classe memoir non sarebbe necessaria, anche perché quella classe provvede in modo proprio a gestire le testatine in generale; per l'indice usa un particolare stile delle pagine che già contiene le testatine corrette.

Riferimenti bibliografici

BECCARI, C. (2018a). *Il LaTeX Reference Manual commentato*. GuIT. URL http://www.guitex.org/home/images/doc/GuideGuIT/latexhandbookcommentato.pdf.

— (2018b). *Il pacchetto TOPtesi*. TUG. Leggibile con `texdoc toptesi-it` o, in inglese, con `texdoc toptesi`.

BECCARI, C., GIACOMELLI, R. e MOLINARO, M. (2018). «Il concorso della scacchiera». *ArsTEXnica*, (25).

GÄSSLEIN, H., NIEPRASCHK, R. e TKADLEC, J. (2016). «The pict2e package». PDF document. Leggibile con `texdoc pict2e`.

LAMPORT, L. (1985). *LaTeX. A Document Preparation System*. Addison-Wesley, 1ª edizione.

— (1994). *LaTeX. A Document Preparation System*. Addison-Wesley, 2ª edizione.

THE LATEX PROJECT TEAM (2018). *The xparse package*. TUG. Leggibile con `texdoc xparse` con una distribuzione TeX Live.

▷ Claudio Beccari
`claudio dot beccari at gmail dot com`