

Edizione con \LaTeX delle opere di Francesco Maurolico

Massimiliano Dominici, Pier Daniele Napolitani

Sommario

Il Progetto Maurolico è nato alcuni anni fa con lo scopo di pubblicare, a stampa e in versione elettronica, l'edizione critica delle opere di Francesco Maurolico (1494–1575). Nell'ambito di questo progetto è stato realizzato un sistema, il \MAURO-TeX , che a partire da un linguaggio di *mark-up* di tipo \LaTeX , consente, attraverso l'uso di appositi strumenti informatici, di ottenere un *output* HTML per la pubblicazione su internet e un *output* in \LaTeX standard come formato intermedio verso PDF e POSTSCRIPT.

\MAURO-TeX è in fase di profonda trasformazione. Da una parte si sta spostando verso l'XML la parte riguardante la codifica dei testi; dall'altra l'avvio della pubblicazione a stampa delle opere offre l'occasione per una profonda revisione delle macro usate dal formato intermedio \LaTeX per generare l'*output* PDF e POSTSCRIPT.

1 Introduzione

Circa otto anni fa cominciava un progetto di edizione delle opere scientifiche di Francesco Maurolico, un importante matematico del Rinascimento (1494–1575). L'edizione, una volta conclusa, consisterà di 10 volumi, per un totale, circa, di 5000 pagine. Essa riunirà circa 250 testi, che vanno da semplici frammenti di appena mezza pagina a un'opera intera divisa in più libri. La produzione dell'autore copre un arco di 60 anni, e tutto il campo della matematica dell'epoca: matematica propriamente detta, ma anche ottica, astronomia, meccanica, gnomonica, ecc. Essa rimaneggia, a volte a partire da alcuni frammenti, nuove edizioni di autori dell'antichità: Euclide, Archimede, Apollonio, Teodosio, Menelao, ecc., che completa con lavori personali di particolare originalità. Scarsamente diffusi durante la vita dell'autore, questi lavori ebbero una sicura influenza sui matematici della fine del XVI secolo e di quello successivo, in Italia, per il tramite del *milieu* matematico gesuita, ma anche nel resto d'Europa.

Il progetto riunisce le competenze di un gruppo di persone, principalmente storici della matematica, che abitano in paesi diversi, lavorano con macchine, strumenti e *software* differenti. Le esigenze del lavoro d'*équipe* hanno spinto all'esplorazione e alla realizzazione di soluzioni informatiche che permettessero a tutti i ricercatori coinvolti di lavorare in maniera uniforme. Malgrado fossero presenti,

all'epoca in cui è partito il progetto, alcuni strumenti alternativi per la gestione di un'edizione critica (EDMAC, *tustep*, *cte*), queste soluzioni, o erano sconosciute ai membri del progetto o non rispondevano alle caratteristiche volute.

Poiché il progetto era partito da un Dipartimento di Matematica (quello dell'Università di Pisa), era naturale che si pensasse a \LaTeX per la realizzazione di un tale sistema di edizione. Il risultato è \MAURO-TeX , un linguaggio di *mark-up* in grado di rispondere a tutte le esigenze di un'edizione critica.

In un primo tempo \MAURO-TeX è stato concepito come un linguaggio capace di padroneggiare le esigenze di codifica di un'edizione critica e, allo stesso tempo, di fornirne una versione *stampabile*, compilando il sorgente con il solo ausilio del file `mauro.sty`. Man mano che il progetto cresceva e si sviluppava, è risultato evidente che ciò non era possibile. Parallelamente a ciò, uno degli obiettivi del Progetto era quello di costruire un sito internet dove pubblicare (in formato HTML) l'intera edizione: a questo scopo è stato sviluppato un *parser bison*¹ in grado di interpretare la grammatica \MAURO-TeX e un programma in C++, `m2hv`, in grado di trasformarla in HTML. A questo punto, le funzioni che venivano aggiunte al linguaggio erano sempre più difficili da implementare in termini di macro \LaTeX , mentre era relativamente semplice inserirle in un programma in C++ che utilizzasse il suddetto *parser*. Così è stato sviluppato `m21v`, l'equivalente per \LaTeX di `m2hv`. Questo programma trasforma il sorgente codificato in \MAURO-TeX in un nuovo file codificato in \LaTeX standard da cui ottenere l'*output* finale in PDF o POSTSCRIPT.

Attualmente è stata progettata, ed è in corso di sviluppo, la trasformazione del linguaggio \MAURO-TeX da un *mark-up* di tipo \LaTeX ad uno di tipo XML. L'XML infatti si presta meglio di \LaTeX ad esprimere la codifica di un testo e ad essere analizzato da opportuni strumenti. Tuttavia \LaTeX rimarrebbe come formato intermedio finalizzato alla produzione di file PDF e POSTSCRIPT.

In quest'ottica, la preparazione, da poco iniziata, dell'edizione a stampa delle opere di Maurolico offre quindi l'occasione, soprattutto in questa pri-

1. Un *parser* è un programma in grado di analizzare un flusso di dati (letto ad esempio da un file o da tastiera) e di interpretarlo in base ad una grammatica formale, per costruirne un albero degli elementi.

ma fase, di individuare quale forma dovrà avere l'output LATEX del sistema MAURO-TEX così trasformato (o del suo eventuale successore). Una particolare cura verrà posta nell'assicurarsi che questa forma si mantenga comunque flessibile ed in grado di collaborare con i più importanti pacchetti dedicati alla gestione di edizioni critiche.

2 Il MAURO-TEX

2.1 Testo critico e apparati

L'edizione critica deve prendere in considerazione, innanzitutto, le diverse modalità con cui ci pervengono i testi degli autori: sotto forma, a volte, del solo manoscritto, altre volte copiati in alcune centinaia di esemplari, permettendo così la creazione di una *tradizione* testuale. A partire dal XV secolo i testi possono essere stati stampati. L'insieme dei testi in mano all'editore costituiscono i *testimoni* dell'opera.

I testimoni sono generalmente lontani dall'essere identici. L'autore stesso può aver modificato il testo nel tentativo di migliorarlo. Se il testo è stato copiato, il copista può aver introdotto degli errori. L'esperienza mostra che questo tipo di errori è piuttosto frequente e può introdurre (a causa della dimenticanza di parole, del salto di intere righe di testo) delle modifiche che stravolgono il senso del testo. Se sono presenti note marginali, spesso di mano diversa da quella dell'autore, il copista può decidere di integrarle senza darne alcun avviso. Inoltre bisogna tener conto di difficoltà più materiali: le ingiurie del tempo, lacerazioni della carta, fogli andati perduti, l'inchiostro che si cancella, macchie di umidità che rendono illeggibile il testo. Infine, il numero degli errori aumenta in maniera esponenziale con l'aumentare dei *testimoni* disponibili.

L'editore ha il dovere di presentare, a partire da tutti i testimoni, un *testo critico*, ovvero un testo che rappresenti i diversi testimoni e che egli ritiene dia conto della volontà dell'autore. Il testo critico è accompagnato da un *apparato critico* che deve dare al lettore la possibilità di ricostruire a) il percorso attraverso il quale l'editore è giunto al testo critico, a partire dai testimoni; b) il contenuto di ogni singolo testimone. L'editore deve segnalare in apparato tutte le *varianti* tra i testimoni: parole e frasi differenti, diverse disposizioni del testo, note marginali, aggiunte in interlinea, in margine, lacune, ecc. Inoltre deve segnalare i propri interventi sul testo: le parole che ha dovuto sopprimere o aggiungere o correggere per arrivare a dare al testo critico un senso compiuto. Infine, deve segnalare i cambiamenti di scrittura e d'inchiostro che indicano l'intervento di diverse mani (o anche della stessa in tempi diversi).

Nel quadro di un'edizione che presenti una tradizione molto ricca, l'editore deve anche provvedere

a individuare i legami di parentela e filiazione dei vari testimoni, il cosiddetto *stemma codicum*. Questo lavoro può essere parte integrante dell'edizione stessa, in quanto, una volta individuata, per un testimone, la sicura derivazione da un altro, l'editore può decidere di non tenerne più conto ai fini dell'edizione, con una evidente semplificazione del lavoro.

Le edizioni scientifiche presentano delle particolarità proprie rispetto alle edizioni classiche. I testi scientifici presentano una forte strutturazione di cui l'editore deve rendere conto. I testi antichi rimandano a numerosi riferimenti interni ed esterni. Le centinaia di riferimenti agli *Elementi* di Euclide (la Bibbia dei matematici e il loro riferimento principale nel corso di molti secoli), sono costantemente citati, esclusivamente per mezzo di un numero, in tutti i lavori matematici, fino all'avvento dei matematici moderni. Queste citazioni sono estremamente importanti per lo storico della matematica e devono essere inventariate, classificate e rese per esteso. Inoltre, la quantità di figure e di numeri riveste particolare importanza nei testi scientifici, ed è fonte di numerosi errori dei copisti anch'essa. Infine una difficoltà supplementare è data dalla notazione matematica stessa, così differente da quella dei matematici moderni.

2.2 Il linguaggio

Il MAURO-TEX è un linguaggio di *mark-up* che permette di costruire, passo per passo, un'edizione critica. L'utente trascrive il testo dei vari testimoni, riunendoli in un unico file LATEX. Il linguaggio permette di trattare tutti i casi che si presentano nel corso di un'edizione che comprenda uno o più testimoni. Per ogni occasione che dia luogo ad una o più varianti nei testimoni, egli segnala consecutivamente il loro contenuto nei diversi campi di una apposita macro ($\backslash\mathbb{V}\{\}$). Qualche esempio basterà a dare un'idea del meccanismo. Si consideri il caso di tre testimoni che presentino le rispettive lezioni:

- A: Sit data gratia, sit datus cubus...
- B: Sit data latio, sit datus cubus...
- C: Sit data ratio, sit datus cubus...

La trascrizione si effettua nella seguente maniera:

```
Sit data \mathbb{V}{A :gratia}{B :latio}
{C :ratio}}, sit datus cubus ...
```

E l'output è il seguente:

Sit data gratia^a, sit datus cubus

a. gratia A latio B ratio C

Per definizione, il primo campo della macro $\backslash\mathbb{V}\{\}$ contiene il testo critico; gli altri campi sono destinati a riempire l'apparato. Una semplice

permutazione dei campi basta a modificare il testo critico e, di conseguenza, l'apparato. Per trattare le differenti situazioni che si incontrano nel corso di un'edizione critica, l'utente ha a sua disposizione un certo numero di scorciatoie finalizzate a riempire l'apparato critico con abbreviazioni e argomenti standard della filologia. Supponiamo che nel testimone *C* la parola *gratia*, cancellata con un tratto di penna, segua la parola *ratio*. Il trascrittore utilizza la macro predefinita `\POSTDEL` e scrive:

```
Sit data \VV{C:\POSTDEL{gratia}:ratio}
{A:gratia}{B:latio}}, sit datus cubus ...
```

E ottiene:

Sit data ratio^a, sit datus cubus

a. post ratio del. gratia C gratia A latio B

Se la parola *ratio* appare come aggiunta in interlinea:

```
Sit data \VV{C:\INTERL:ratio}
{A:gratia}{B:latio}}, sit datus cubus ...
```

ottenendo

Sit data ratio^a, sit datus cubus

a. ratio in interl. C gratia A latio B

Naturalmente è possibile riempire l'apparato critico con tutto ciò che sembra più opportuno all'editore, soprattutto nei casi non espressamente previsti dal linguaggio. Ad esempio, se si desidera segnalare che la parola *ratio* è scritta con inchiostro rosso, si può ricorrere alla macro `\DES{}`, il cui argomento è libero.:

```
Sit data
\VV{C:\DES{rubro atramento}:ratio}
{A:gratia}{B:latio}}, sit datus cubus ...
```

E si avrà

Sit data ratio^a, sit datus cubus

a. ratio rubro atramento C gratia A latio B

Per tutti i casi ricorrenti è stata prevista una scorciatoia del tipo di quelle viste sopra: omissioni, lacune, parole illeggibili, integrazioni in margine e in interlinea, correzioni del copista, congetture e correzioni dell'editore, ecc. È stata prevista una codifica dei nomi dei testimoni in modo da poter descrivere le diverse mani che hanno lavorato sul manoscritto. Le varianti cosiddette "banali" (errori di ortografia e refusi) sono codificate tramite la macro `\VB{}`, che ha la stessa sintassi di `\VV{}`. L'uso vuole che queste varianti vengano segnalate nella codifica del testo ma non riportate nell'apparato, in modo da non appesantirlo inutilmente.

Il *mark-up* sembra sufficientemente completo per poter realizzare una vera e propria classificazione delle varianti utili all'editore e allo storico, e altri elementi di codifica permettono di descrivere formalmente i testimoni. Il linguaggio fornisce inoltre una serie di macro più elementari per il trattamento dei cambi di foglio, delle date, dei titoli, di citazioni e riferimenti, degli enunciati delle proposizioni, ecc.

3 Il lavoro sull'edizione a stampa

3.1 Specifiche e linee guida

Abbiamo già fatto notare che il lavoro di revisione delle macro utilizzate dall'*output* L^AT_EX di *MAURO-T_EX*, ha un duplice obiettivo: quello immediato di avere una edizione a stampa conforme ai criteri espressi dall'editore, e quella, sperimentale, di determinare la forma futura del suddetto *output*, una volta che *MAURO-T_EX* (o ciò che prenderà il suo posto) verrà trasformato in linguaggio di *mark-up* puro e semplice.

Finora il lavoro si è limitato alla parte concernente l'apparato critico vero e proprio, che rappresenta comunque il nucleo centrale del *MAURO-T_EX*, e, anche limitatamente a ciò, non è ancora concluso.

Piuttosto che implementare da zero un ennesimo pacchetto per edizioni critiche, cosa che non è tra gli scopi di questo lavoro, si è pensato di utilizzare un pacchetto già esistente e la scelta è caduta su *ledmac* (WILSON (2005)), che può vantare, rispetto ai concorrenti, *ednotes* (LÜCK (2006)) e *bigfoot* (KASTRUP (2006)), un livello di confidenza maggiore (*ledmac* è l'erede di *EDMAC*, che viene usato ormai da una decina di anni nella produzione di edizioni critiche) e una documentazione più estesa e organica.

L'impostazione di fondo, però, che si è tenuta presente nell'elaborare le macro, è stata quella di mantenere un livello di astrazione maggiore possibile. L'idea sarebbe quella di avere, a conclusione del lavoro, un *front-end* in grado di lavorare all'occorrenza con diversi *back-end*. In parole povere l'eventuale classe *maurolico*, risultato finale di questo esperimento, dovrebbe essere in grado di lavorare, internamente, con diversi pacchetti dedicati alle edizioni critiche, quali *ledmac*, *ednotes* o *bigfoot*, lasciando all'utente la possibilità di selezionare, tramite una semplice opzione della classe, quale di questi pacchetti usare.

Per tutti i riferimenti al funzionamento specifico di *ledmac* si rimanda ai testi citati in bibliografia.

3.2 Installazione dei font

La prima specifica indicata dall'editore è stata la scelta dei caratteri da utilizzare per la stampa. Questi caratteri, in maggioranza assenti, poiché commerciali, dalle distribuzioni T_EX, hanno richiesto un processo di installazione non banale, in

quanto doveva rispondere a particolari esigenze di armonizzazione e di uso.

L'insieme dei caratteri da utilizzare per l'edizione a stampa delle opere di Francesco Maurolico, è costituito dal *Monotype Dante*, per il testo in tondo e in corsivo; dal *Gill Sans*, per le parti evidenziate in obliquo; dal *Math Garamond*, unico carattere presente in alcune distribuzioni TEX, per i simboli matematici; e dall'*Odyssea* per il testo in greco antico (politonico).

Considerata la complessità dell'installazione, si è ritenuto utile fare uso di un programma come `fontinst` (JEFFREY *et al.* (2004)), che garantisce la possibilità di controllare il processo nei dettagli e di effettuare in piena sicurezza operazioni di trasformazione dei caratteri stessi – quale ad esempio la scalatura del *Gill Sans*, necessaria per armonizzarne le proporzioni rispetto a quelle del *Dante*.

`fontinst` permette di definire all'interno di un file (nel nostro caso `maurofonts-inst.tex`) una serie di istruzioni che regolano la costruzione dei file ausiliari (`.vf` e `.tfm`) necessari a TEX per poter utilizzare il font in questione².

Nel seguito daremo solo qualche cenno sui dettagli dell'installazione, in quei casi in cui si sono riscontrate particolari difficoltà, rimandando, per tutto ciò che è ordinaria amministrazione, alla bibliografia sull'argomento (JEFFREY *et al.* (2004); LEHMAN (2004)).

Corpo del testo

Per il corpo del testo, l'editore ha scelto il *Monotype Dante*, un carattere ideato e disegnato dall'italiano Giovanni Maiderstag sulla base dei caratteri del primo umanesimo incisi da Francesco Griffo per Aldo Manuzio.

Il carattere – disponibile in formato Postscript Type1, nei pesi *regular*, *medium* e *bold*, e corredato di *expert font set*, cioè di un insieme separato di font contenente legature, numeri minuscoli³, numeri e lettere esponenti, ecc. – non ha comportato particolari problemi. `fontinst` è stato creato proprio avendo in mente l'installazione di un font Type1 con codifica Adobe Standard, e questo è il caso, per l'appunto, del *Dante*. Così, è sufficiente l'attenta lettura di un ottimo *tutorial* come *The Font Installation Guide* (LEHMAN (2004)) per ottenere una installazione perfettamente funzionante, al termine della quale l'utente ha a disposizione alcune

2. Questo non è esattamente vero. `fontinst` genera file con estensione `.pl` e `.vpl`, i cosiddetti *Property List* e *Virtual Property List*, versioni ASCII dei *TeX Font Metrics* e *Virtual Fonts* (con estensioni `.tfm` e `.vf`) che TEX richiede per poter utilizzare un font. Questi ultimi si ottengono compilando i primi con `pltotf` e `vptovf`.

3. Numeri “minuscoli”, o “saltellanti”, o, all'inglese “*old style*”, sono quei numeri che, come le lettere minuscole, hanno dimensioni variabili in altezza, caratteristica che li distingue dai numeri “maiuscoli” o “normali”, più comunemente usati, che hanno invece tutti la stessa altezza.

famiglie di font che può sfruttare per determinare l'aspetto tipografico del testo.

Ad esempio, le seguenti linee di codice, inserite nel file `maurofonts.sty`, permettono di usare il *Dante* con i numeri “normali” (caricando il pacchetto con l'opzione `lining`: `\usepackage[lining]{maurofonts}`), o con i numeri “saltellanti” (opzione `oldstyle`: `\usepackage[oldstyle]{maurofonts}`).

```
\RequirePackage{nfsssect}
\DeclareOption{lining}{%
  \renewcommand*{\rmdefault}{mdtx}
}
\DeclareOption{oldstyle}{%
  \renewcommand*{\rmdefault}{mdtj}
}
```

L'uso del *Gill Sans* è limitato all'apparato delle note, dove viene utilizzato per far risaltare i testimoni (indicati di norma con una lettera maiuscola) o alcune notazioni particolari (ad esempio l'indicazione che una particolare parola o frase è stata scritta con un inchiostro diverso da quello utilizzato nel resto del manoscritto: *diverso atramento*).

Anche in questo caso l'installazione è stata lineare e non ha comportato particolari problemi. Il carattere è stato scalato al 90% delle sue dimensioni effettive, per renderlo più armonico al disegno del *Dante*.

Simboli matematici

Per quanto le opere di Francesco Maurolico trattino di argomenti matematici, la notazione simbolica che vi si trova utilizzata è abbastanza semplice e non ha niente a che vedere, in quanto a complessità, con quella che si sarebbe sviluppata nei secoli successivi.

Tuttavia, si è ritenuto comunque opportuno fornire il sistema di caratteri da utilizzare per l'edizione, di un insieme di simboli matematici che coprisse non solo le esigenze più elementari ma fosse adatto anche ad esprimere una notazione più avanzata.

È stato quindi necessario individuare un font matematico che si armonizzasse abbastanza bene con il carattere utilizzato per il corpo del testo, e cioè il *Dante*. Considerando che le caratteristiche morfologiche del *Dante* non sono così dissimili da quelle di un altro carattere umanistico come il *Garamond*, la decisione di utilizzare il font *Math Design URW Garamond*, contenuto nel pacchetto `mathdesign` (PICHAREAU (2004)) di Paul Pichareau e progettato per essere usato insieme all'*URW Garamond*, si è imposta come la più semplice.

A differenza di quanto avviene per il testo, dove LATEX utilizza la codifica T1 a 8 bit, per quanto riguarda la notazione matematica vengono ancora utilizzate delle codifiche a 7 bit (si veda a questo proposito VIETH e HOEKWATER (1999)). In pratica LATEX utilizza quattro famiglie di font, con diversa codifica:

- OT1 (la vecchia codifica per il testo) per estrarre le lettere (generalmente in tondo) usate per comporre gli operatori matematici (come in “sin” o in “log”);
- OML per estrarre le lettere (generalmente in corsivo) usate per le variabili (come in “f” o in “ω”);
- OMS per estrarre i simboli;
- OMX per estrarre simboli aggiuntivi.

In tutti questi casi abbiamo fatto in modo che venissero utilizzati singoli caratteri presi dal *Dante*, ogni volta che fossero presenti, poi da *Odyssea* e infine dal *Math Design URW Garamond*. Questo è possibile richiamando delle apposite procedure nel file `maurofonts-inst.sty` (vedi la sezione 3.2). Per comodità riportiamo qui sotto un esempio che si riferisce unicamente alla codifica OML:

```
\installfont{mdtri7m}{%
  mdtri8r,odysi,mf-gmri7m,mathit}{%
  oml}{OML}{mdt}{m}{it}{}
```

Le istruzioni riportate indicano in sostanza a T_EX di costruire il font virtuale⁴ `mdtri7m` nella codifica OML prendendo dal *Dante* (`mdtri8r`) tutti i caratteri che può, utilizzando poi il font *Odyssea* (`odysi`) per le lettere greche e riempiendo infine quanto manca con *Math Design URW Garamond* (che qui è stato rinominato come `mf-gmri7m`).

Greco

L’installazione del font greco *Odyssea*, si è rivelata la parte più ardua di questa prima fase. In questo caso si sommano due singolarità: una al livello della codifica interna al font, e l’altra al livello del metodo di *input* usato da MAURO-T_EX.

Per quanto riguarda il primo di questi aspetti, il font *Odyssea*, un prodotto commerciale della *Linguist’s Software*, usa una codifica interna poco diffusa, e utilizzata quasi esclusivamente dalla suddetta casa. Questo implica che è difficile trovare un supporto “prefabbricato” a questa codifica per fontinst. Anche la documentazione in materia è scarsa, e di pochissimo aiuto è il file AFM che si può ottenere tramite `ttf2afm`. L’unico aiuto reperibile in rete si trova sul sito della *SIL International* (www.sil.org), da cui è possibile scaricare il file `LingSoftmaps.zip` che contiene la codifica del font *GraecaII*, anch’esso distribuito da *Linguist’s Software* e che presenta sostanzialmente la stessa codifica di *Odyssea*.

4. Un font virtuale (o *virtual font*) è un file binario che viene usato dal driver DVI (o da `pdfetex`, se l’output generato è in PDF) per tracciare effettivamente i caratteri. Tramite un font virtuale è possibile compiere operazioni complesse come la rimappatura di un font, la combinazione di caratteri provenienti da font diversi, o l’applicazione di effetti speciali (scalatura, inclinazione, spaziatura, ecc.).

Allo stesso modo, il metodo di *input* dei caratteri greci utilizzato da MAURO-T_EX è l’*Ibycus*, una variante di *Beta Code*. Quest’ultimo è effettivamente lo standard adottato da una serie di organizzazioni e siti internet (*Thesaurus Linguae Graecae*, *Perseus*, ecc.), ma non lo è affatto per T_EX, dove viene comunemente adottata la codifica LGR. Da ciò segue che la sequenza di caratteri da inserire per ottenere la medesima lettera accentata è diversa nei due casi; e quindi, in generale, nel predisporre l’occorrenza per generare i *virtual fonts* che dovranno essere utilizzati da L^AT_EX, bisognerà costruire in maniera diversa le legature. Ad esempio, mentre, utilizzando il sistema standard, per ottenere “ō” bisognerà digitare la sequenza di caratteri: “~ > w”, lo stesso risultato potrà essere ottenuto in *Ibycus* digitando la sequenza “w (=”. Anche in questo caso non esistono soluzioni “prefabbricate”, ed è stato quindi necessario scrivere da zero i file che contengono le istruzioni da passare a fontinst per costruire le opportune legature.

3.3 Definizione delle caratteristiche della pagina

Raccolte le indicazioni dell’editore sulle specifiche del layout si è provveduto a trasformarle in parametri per L^AT_EX. Le specifiche erano le seguenti:

- dimensioni della pagina: 17cm × 24cm;
- gabbia del testo: 13cm × 20cm, incluso lo spazio per i titoli correnti;
- corpi dei caratteri: normale 11/13pt, infratesto 10/11pt, note 9/10pt, titoli 12/24pt
- titoli correnti in maiuscoletto spaziato;
- titoli in maiuscolo spaziato.

Per quanto riguarda i parametri della pagina, è stato utilizzato il pacchetto `geometry` (UMEKI (2002)):

```
\RequirePackage{geometry}
\geometry{
  includehead,
  includemp,
  marginparwidth=1.1cm,
  paperwidth=17cm,
  paperheight=24cm,
  width=13cm,
  height=20cm,
  headsep=.2cm
}
```

Poi sono stati ridefiniti i corpi dei caratteri (riportiamo, per brevità, solo la ridefinizione di `\normalsize`):

```
\renewcommand\normalsize{%
  \@setfontsize\normalsize\@xipt{13}%
```

```
\abovedisplayskip 11\p@ \@plus3\p@%
\@minus6\p@
\abovedisplayskip \z@ \@plus3\p@
\belowdisplayshortskip 6.5\p@%
\@plus3.5\p@ \@minus3\p@
\belowdisplayskip \abovedisplayskip
\let\@listi\@listI}
```

E, per l'impostazione delle testatine e piè di pagina, è stato utilizzato il pacchetto fancyhdr (VAN OOSTRUM (2004)):

```
\RequirePackage{fancyhdr}
\fancyhead{}
\fancyfoot{}
\fancyhead[CE]{
\capsselect{maiuscoletto}
\caps{\leftmark}}
\fancyhead[CO]{
\capsselect{maiuscoletto}
\caps{\rightmark}}
\fancyhead[RO,LE]{\thepage}
```

Infine, la spaziatura del maiuscolo e del maiuscoletto è stata ottenuta facendo ricorso ai comandi forniti dal pacchetto soul (FRANZ (2003)):

```
\RequirePackage{soul}
\capsdef{/mdtx//}{%
\scshape}{.08em}{.35em}{.5em}
\capssave{maiuscolo}
\capsdef{/mdtx//}{%
\scshape}{.062em}{.35em}{.5em}
\capssave{maiuscoletto}
```

Tuttavia è in corso di studio la possibilità di abbandonare questa soluzione e sostituirla con la creazione di un font virtuale con le opportune spaziature.

3.4 Apparato

L'apparato critico dell'edizione delle opere di Francesco Maurolico prevede tre livelli di note. Il primo si riferisce alle varianti del testo, il secondo ricostruisce l'apparato delle citazioni, il terzo è riservato agli interventi dell'editore. Quest'ultimo livello non comporta particolari problemi di conversione, essendo l'equivalente di una normale nota a piè di pagina. Il preprocessore, quindi, si limiterà ad esportare letteralmente la macro `\Adnotatio{<arg>}` e sarà poi il particolare *back-end* scelto dall'utente a definire questa macro in maniera significativa. Ad esempio, nel nostro caso, usando `ledmac` come *back-end*, avremo la seguente definizione:

```
\let\Adnotatio\footnoteA
```

L'apparato testuale e quello delle fonti presentano invece una maggiore complessità. Per il momento solo il primo di questi apparati è stato trattato in maniera sufficientemente approfondita. Esaminiamolo nei dettagli.

Nel file sorgente *MAURO-TEX* una variante testuale è marcata in questa maniera:

```
sit rectus: \VV{
{*\ED{conieci}:cum}
{A:\NL}
}
```

Come descritto nel paragrafo 2.2, la macro `\VV` può contenere un numero indefinito di campi delimitati da parentesi graffe: ognuno di questi rappresenta una variante testuale. I sottocampi (delimitati da `:`) in cui ogni variante è organizzata sono tre, di cui uno, il secondo, opzionale, e rappresentano rispettivamente: la fonte manoscritta (`*` sta a significare la lezione del testo critico), eventuali aggiunte dell'editore, e la variante vera e propria. La lezione da inserire nel testo è quella indicata nella prima variante (nel nostro esempio *cum*). La macro `\ED`, contenuta nel secondo campo indica un commento dell'editore e ha come risultato quello di stampare il proprio contenuto opportunamente formattato (nel nostro caso *conieci* verrà stampato in *Gill Sans* obliquo.

Possiamo quindi immaginare di trasformare il codice precedente nel seguente, che ha un aspetto più "*L^AT_EX* friendly":

```
sit rectus: \testocritico{cum}{%
\variante[\ED{conieci}]{*}{cum}
\variante{A}{\NL}
}
```

Resta da dare un significato pratico alle macro `\testocritico`⁵ e `\variante`, in riferimento al nostro *back-end* basato su `ledmac`.

Nel caso di `\testocritico`, l'operazione è lineare:

```
\newcommand\testocritico[3] []{%
\edtext{#2}{#1\Afootnote{#3\unskip}}}
```

Cioè, la nostra macro è l'equivalente della macro `\edtext` di `ledmac`, utilizzata per generare l'apparato critico, dove il secondo argomento viene direttamente passato ad uno degli apparati in nota, per l'esattezza `\Afootnote`⁶. Rispetto al comportamento di default di `\Afootnote`, questo verrà modificato in modo che non venga riportato in nota il lemma (*cum*), ma solo il numero di riga:

```
\renewcommand{\Afootfmt}[3]{%
\ledsetnormalparstuff
{\notenumfont\printlines#1}|}%
\strut\hspace{3pt}%
% {\select@lemmafnt#1|#2\rbracket}%
```

5. La scelta di utilizzare nomi italiani per le istruzioni è dovuta al fatto che dovendo la futura classe maurolico cooperare con i pacchetti `ledmac` e `ednotes`, era oggettivamente difficile trovare nomi inglesi che rispecchiassero il significato dell'istruzione e non fossero già usati da uno dei due suddetti pacchetti. Tuttavia, essendo ancora il progetto nella sua fase iniziale e di sperimentazione, questa scelta potrebbe essere rivista.

6. L'argomento opzionale serve ad inserire, manualmente, eventuali informazioni da passare a `ledmac` relative al lemma e ai numeri di riga

```
% \enskip#3\strut\par}
#3\strut\par}
```

`\variante`, invece comporta maggiori complicazioni. Ci si aspetterebbe che fosse sufficiente far stampare alla macro i propri argomenti nella sequenza appropriata: lemma, eventuali commenti provenienti dall'argomento opzionale, indicazione della fonte. Questo non è possibile, data la possibilità di avere, come argomento opzionale, macro del tipo `\POSTDEL{<arg>}` che devono stampare il seguente testo: “*post <lemma> del <arg>*”. Dove `<lemma>` è, ovviamente, il secondo argomento obbligatorio della macro `\variante`.

La soluzione sta nel salvare il lemma in una macro interna e interpretare l'argomento opzionale come una “funzione” da applicare a quest'ultima:

```
\newcommand\variante[3][\pm@lemma]{%
  \def\pm@lemma{#3}#1\pm@lemma%
  \ifx*#2\pm@lemmasep\else\ \pm@fonte{#2}\fi%
  \variantsep}
\newcommand\MARG{%
  \pm@lemma\ \pm@varfont{in marg.}%
  \let\pm@lemma\relax}
\newcommand\POSTDEL[1]{%
  \def\pm@lemmasep{\ \pm@varfont{post}}%
  \ \pm@lemma\ \pm@varfont{del} #1%
  \let\pm@lemma\relax}
\newcommand\ANTEDEL[1]{%
  \def\pm@lemmasep{\ \pm@varfont{ante}}%
  \ \pm@lemma\ \pm@varfont{del} #1%
  \let\pm@lemma\relax}
\newcommand\INTERL{%
  \pm@lemma\ \pm@varfont{supra lineam}%
  \let\pm@lemma\relax}
```

Il codice riportato qui sopra, mostra appunto, oltre alla definizione di `\variante`, alcune di queste “funzioni”.

Un'idea visiva, molto sommaria, dei risultati finora ottenuti può essere fornita dalla figura 1.

4 Conclusioni

Riteniamo che quanto abbiamo qui esposto possa avere un interesse che va al di là dell'edizione dell'opera di Maurolico. L'esperienza del *MAURO-TEX* è sicuramente, per molti motivi, limitata alle peculiarità di questa edizione. Tuttavia da essa sono nati almeno due filoni di sviluppo:

1. la creazione di un linguaggio XML capace di supportare un'interfaccia “amichevole” e, soprattutto, capace di adattarsi alle disparate esigenze dell'editore critico in modo flessibile;
2. la preparazione di una serie di strumenti per l'*output* di un'edizione capaci di adattarsi alle disparatissime esigenze non solo dell'editore critico, ma anche della tipografia e della creazione di siti web.

Le esperienze che stiamo attualmente conducendo su entrambi questi fronti ci sembrano confermare le scelte fondamentali del *MAURO-TEX*: individuare correttamente l'informazione rilevante e distinguerla, attraverso un opportuno sistema di *mark-up* dal risultato grafico che si vuole ottenere.

Riferimenti bibliografici

FRANZ, M. (2003). «The soul package». Disponibile su CTAN, `macros/latex/contrib/soul`.

JEFFREY, A., MCDONNELL, R. e HELLSTRÖM, L. (2004). «fontinst: Font installation software for T_EX». Disponibile su CTAN, `fonts/utilities/fontinst/doc/manual`.

KASTRUP, D. (2006). «The bigfoot». Disponibile su CTAN, `macros/latex/contrib/bigfoot`.

LEHMAN, P. (2004). «The font installation guide». Disponibile su CTAN, `info/Type1fonts/fontinstallationguide`.

LÜCK, U. (2006). «ednotes: for critical editions with L^AT_EX». Disponibile su CTAN, `macros/latex/contrib/ednotes`.

PICHAUREAU, P. (2004). «The mathdesign package». Disponibile su CTAN, `fonts/mathdesign`.

UMEKI, H. (2002). «The geometry package». Disponibile su CTAN, `macros/latex/contrib/geometry`.

VAN OOSTRUM, P. (2004). «Page layout in L^AT_EX». Disponibile su CTAN, `macros/latex/contrib/fancyhdr`.

VIETH, U. e HOEKWATER, T. (1999). «Surviving the tex font encoding mess». Disponibile su CTAN, `fonts/utilities/fontinst/doc/talks`.

WILSON, P. (2005). «ledmac: A presumptuous attempt to port edmac, tabmac, edstanza, to L^AT_EX». Disponibile su CTAN, `macros/latex/contrib/ledmac`.

▷ Massimiliano Dominici
mlgdominici@interfree.it

▷ Pier Daniele Napolitani
Dipartimento di Matematica
Università di Pisa
napolita@mail.dm.unipi.it

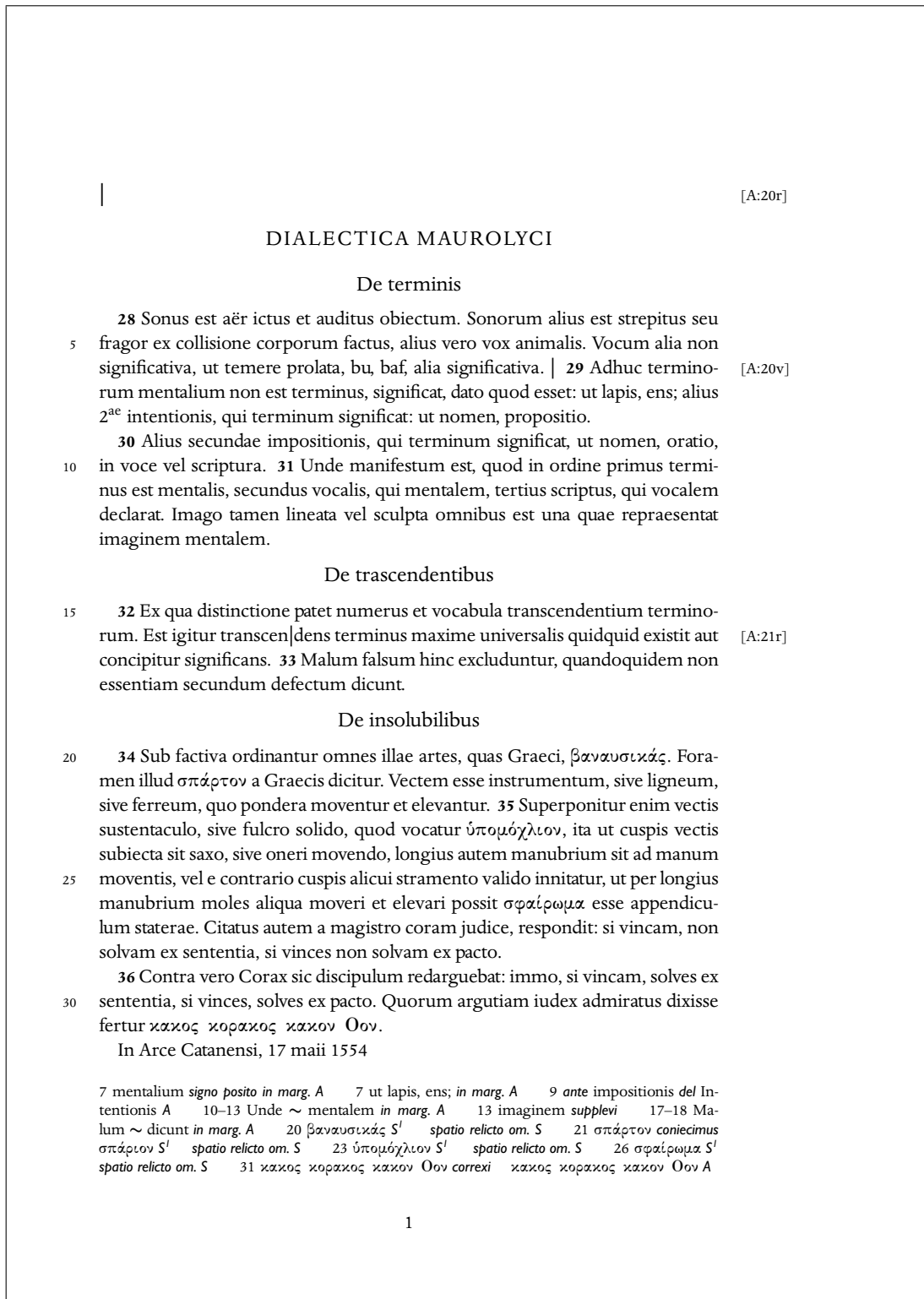


Figura 1: Una pagina dell'edizione a stampa delle opere di Francesco Maurolico