Il pacchetto **esami** per la creazione di prove scritte*

Grazia Messineo, Salvatore Vassallo

Sommario

Viene presentato il pacchetto esami che implementa alcune caratteristiche dei pacchetti exerquiz e probsoln per la produzione collaborativa di database di esercizi e la produzione di temi d'esame.

Abstract

We present the package esami which extends some useful properties of the LaTeX packages exerquiz and probsoln to produce databases of exercises in a collaborative way and to produce exams.

1 Introduzione

In questo lavoro viene presentato un pacchetto da noi elaborato per la gestione delle prove d'esame di Matematica Generale dei corsi della Facoltà di Economia dell'Università Cattolica.

Il pacchetto, pur con alcune modifiche, è in uso da alcuni anni e si è dimostrato stabile ed affidabile, ha permesso di generare senza problemi 60 varianti diverse di uno stesso compito, è in grado di gestire grafici dipendenti da parametri e si è dimostrato flessibile così da essere impiegato in situazioni molto diverse, dalla creazione di prove di verifica nelle scuole secondarie (assegnando ad ogni studente una prova diversa), alla gestione di esami universitari con 20 esercizi.

All'inizio degli anni '90 in Università Cattolica ci si è trovati a dover affrontare il problema di far sostenere prove scritte di matematica ad un numero molto elevato di studenti in aule di capienza insufficiente per impedire "comunicazioni".

Ci si è quindi posti l'obiettivo di creare, con l'ausilio del computer, un numero teoricamente illimitato di varianti di un compito base, varianti in cui gli esercizi dovevano essere "simili", ma con valori numerici diversi, un ordinamento diverso e, nel caso di domande a risposta chiusa, con l'ordinamento delle risposte diverso.

Nel corso degli anni sono stati fatti numerosi tentativi per raggiungere questo obiettivo, usando in tutti i casi programmi a pagamento:

- la creazione di un insieme di macro scritte in *Visual Basic* di Microsoft Word, che estraevano casualmente gli esercizi da un insieme, assegnavano sempre in modo aleatorio dei valori ai parametri presenti negli esercizi, riordinavano domande e risposte. Il metodo usato presentava alcuni problemi e, alla fine, un cambiamento nel linguaggio di programmazione nella suite Office ha reso inutilizzabili tali macro;
- l'uso di Scientific WorkPlace, in particolare del suo generatore di esami con caratteristiche molto simili a quelle desiderate e del modulo che permette di utilizzare un sottoinsieme degli strumenti matematici di Maple o Mupad. I problemi di questa soluzione erano di tre tipi: la formattazione non ottimale dovuta al fatto che il modulo per la creazione degli esami non utilizza LaTeX, ma è più simile al linguaggio RTF, una certa limitatezza nei grafici creati (ad esempio non era possibile inserire parametri) e, soprattutto, il fatto che l'algoritmo per la generazione delle varianti non era noto e quindi poteva accadere facilmente che due compiti "vicini" fossero molto simili tra loro.

2 La filosofia del pacchetto

2.1 La scelta di LaTeX

I problemi sopra esposti hanno spinto a cercare un'altra soluzione e la si è individuata nell'utilizzo completo di LaTeX.

I vantaggi della scelta di LaTeX sono molteplici:

1. la modularità;
2. la possibilità di modificare il codice, controllare a priori l'output delle routine usate, implementare delle proprie macro;
3. le sue capacità di calcolo e grafiche;
4. la possibilità di modificare molti aspetti di formattazione senza modificare il testo;
5. l'output in PDF e quindi la portabilità;
6. la gratuità del software e quindi il risparmio sui costi delle licenze.

L'unico aspetto negativo è che la curva di apprendimento di LaTeX ha una notevole pendenza iniziale e quindi è richiesto agli utilizzatori un periodo di apprendimento abbastanza impegnativo.

Una rapida ricerca sul CTAN mostra che esistono già molti pacchetti e classi dedicati alla stesura di esami, alcuni dei quali sono già stati illustrati su ArsTeXnica (Penna, 2006; Pilu, 2011).

*Il lavoro si inserisce nel progetto di ricerca di interesse d'Ateneo dell'Università Cattolica "Progetto M.In.E.R.Va: Creazione di esercizi di tipo interattivo con percorsi differenziati per il recupero delle carenze e la valorizzazione delle eccellenze in matematica".

2.2 Il pacchetto exerquiz

La scelta iniziale è stata quella di utilizzare il pacchetto exerquiz (Story, 2012) che permette di creare degli esami con domande a risposta chiusa (con la possibilità di variare casualmente l'ordine delle risposte) o aperta; tali esercizi possono essere suddivisi in più parti. All'interno del file con gli esercizi sono presenti anche le soluzioni e tramite un'opzione si può decidere di visualizzarle o meno.

Una delle caratteristiche più importanti del pacchetto è di permettere facilmente l'incorporazione di codice JavaScript nel file PDF di output. Tale funzionalità permette un certa interattività del file PDF come, ad esempio, la correzione automatica degli esercizi: tale caratteristica è stata ampiamente sfruttata in altre occasioni (Messineo e Vassallo, 2012).

Nelle nostre intenzioni un compito era una creatura abbastanza complessa, composta da esercizi a risposta chiusa e a risposta aperta (problemi). Supponendo che i docenti scelgano 10 esercizi (come nel nostro caso), ogni esercizio (sia esso a risposta chiusa o aperta) ha un numero proprio di varianti e ogni variante può dipendere da alcuni parametri. La generazione di un compito prevede che, per ogni versione prodotta (generalmente da 20 a 40 versioni) il pacchetto scelga da ognuno dei file una variante, mescoli i 10 esercizi scelti, nel caso di domande a risposta chiusa mescoli le risposte, scelga i valori dei parametri di ogni esercizio e infine generi un file con le risposte corrette per ogni versione.

Era necessario perciò che il pacchetto exerquiz venisse integrato, poiché tra le caratteristiche volute ha solo la possibilità di permutare casualmente le risposte nelle domande a risposta chiusa. Sia per questo motivo, sia per avere del software "stabile" nel tempo, si è deciso di costruire, partendo da un sottoinsieme di exerquiz, un pacchetto autonomo che abbiamo chiamato esami. Nel corso degli anni (la prima versione del pacchetto è del 2007) ci sono state poi molte intersezioni tra il nostro lavoro e quello dell'autore di exerquiz, con soluzioni a volte simili, a volte diverse.

Il problema di utilizzare parametri all'interno degli esercizi è di facile soluzione: infatti esistono pacchetti che implementano la generazione di numeri casuali e la cosa è possibile anche con un comando nativo di pdfLaTeX (il comando `\pdfuniformdeviate`). La nostra scelta è stata di utilizzare il pacchetto random (Arseneau). Si è deciso poi di utilizzare il pacchetto fp (Mehlich, 1999) per le operazioni matematiche che devono essere effettuate sui parametri perché ha una notazione più intuitiva e ci ha reso possibile implementare altri comandi per semplificare la scrittura degli esercizi.

Il problema della scelta della variante di un esercizio tra quelle proposte è stato risolto adattando e semplificando un comando del pacchetto probsoln (Talbot, 2011): esso è stato creato per la generazione di raccolte di esercizi, scelti tra diverse varianti, da presentare agli studenti in anni accademici diversi.

Abbiamo creato infine un file di configurazione modificabile dall'utente che contiene materiale accessorio (come l'intestazione del compito, le istruzioni per lo studente, ecc.) e dei *template* in cui devono essere inserite variabili che competono direttamente alla generazione del compito, quali il numero di versioni da generare, il numero di esercizi nelle varie parti del compito, la data, ecc..

2.3 Le modifiche al codice

Già dalle prime generazioni di esercizi, il pacchetto esami ha subito evidenziato alcuni limiti.

Il primo limite era intrinseco alla costruzione fatta: era infatti ancora possibile che versioni "vicine" fossero troppo simili. Tale problema nasceva dal fatto che, nel nostro caso, una parte del compito era composta da tre soli esercizi: poiché esistono 6 permutazioni di 3 elementi, la probabilità che versioni adiacenti avessero esercizi simili e nello stesso ordine era di 1/6. Il secondo problema era invece legato alla parametrizzazione degli esercizi: per ogni parametro presente in un esercizio veniva assegnato un intervallo di variazione in cui veniva scelto aleatoriamente il valore. Il seme per la scelta aleatoria viene però fissato all'inizio dell'elaborazione del compito (attualmente dipende dalla data e dal numero della versione del compito); in tale modo parametri con lo stesso intervallo di variazione erano sempre uguali. Il terzo problema era la creazione di una stringa di soluzioni per la correzione del compito e, più in generale, della soluzione degli esercizi: se infatti il pacchetto exerquiz permette facilmente la creazione di una versione del compito con le soluzioni indicate, era di sicuro preferibile avere a disposizione la stringa delle soluzioni alle domande a risposta chiusa. La creazione di tale stringa, visto anche il procedimento di generazione del compito, non si presentava di facile soluzione; inoltre exerquiz scrive le soluzioni delle domande a risposta aperta *verbatim* su un file esterno e le include o no a seconda dell'opzione scelta; tale possibilità ci era preclusa poiché i nostri esercizi sono argomento di un comando e quindi non è possibile usare codice *verbatim*[1].

Un ulteriore problema era invece legato ai file contenenti i sorgenti degli esercizi e, in particolare, alla fase preliminare di controllo della correttezza di tali esercizi: è infatti opportuno avere a disposizione le versioni dei file scritte in forma parametrica, ma non in codice TeX in modo da permettere a chiunque il controllo dei testi.

1. Il pacchetto cprotect risolve tale problema e abbiamo utilizzato tale soluzione in un altro ambito; tale pacchetto è però più recente del nostro e quando l'abbiamo scoperto abbiamo deciso di mantenere il codice che aveva già dimostrato di funzionare bene.

Le nostre soluzioni sono state:

(a) La scelta delle varianti per versioni vicine: come detto con una scelta puramente casuale la probabilità di compiti vicini simili può essere troppo alta (anche se la struttura del compito permette altri "mascheramenti"). Per ovviare a tale inconveniente la nostra soluzione è stata quella di avere sia per la scelta della variante all'interno di un esercizio sia per le permutazioni degli esercizi all'interno del compito, un'assegnazione deterministica da un sottoinsieme delle permutazioni possibili se il numero di "oggetti" tra cui scegliere è piccolo e una scelta del tutto aleatoria altrimenti.

(b) Al problema della variazione dei parametri si è ovviato da un lato valutando con maggiore attenzione nella stesura degli esercizi l'intervallo di variazione dei parametri, dall'altro inserendo nel codice una routine per far scegliere un numero successivo nella sequenza dei numeri casuali; vi è inoltre la possibilità di una variazione "locale" del seme, per cui se il parametro a è nell'intervallo $[\alpha, \beta]$ si può richiedere di generare il parametro b nello stesso intervallo, ma con un seme diverso (e quindi ottenendo, in generale, un valore diverso). Infine c'è la possibilità di generare numeri casuali in un intervallo che abbia come estremi parametri (già assegnati) o che non assuma valori all'interno un insieme assegnato[2].

(c) La soluzione al problema della creazione della stringa delle soluzioni è stata quella di sfruttare il sistema di riferimenti incrociati `\label` e `\ref` e il file `aux`, mentre per le soluzioni delle domande a risposta aperta si è creato un ambiente solution che scrive normalmente sul file di output se si ha l'opzione solutions mentre cancella il suo contenuto con l'opzione nosolutions.

Della soluzione all'altro problema prospettato (visualizzazione degli esercizi in forma parametrica) ci si occuperà più avanti, presentando i file che il pacchetto usa.

Ci sono poi altri problemi di natura diversa: sono possibili di sicuro notevoli miglioramenti nel codice e nella gestione delle liste; la soluzione da noi implementata per avere un file esterno di soluzioni di facile lettura impedisce di usare domande a risposta chiusa multipla e, infine, poiché si è deciso di inserire ogni esercizio in minipage[3] non sempre la formattazione è ottimale.

2. Il programma cercherà di soddisfare tutte le richieste generando nuovamente il parametro per un certo numero di volte, limitato da una quantità fissata dall'utilizzatore.

3. Alcuni studenti si erano lamentati di non avere visto mezzo esercizio perché diviso su due pagine, per esempio.

I problemi maggiori sono però "umani" e riguardano la maggiore attenzione necessaria nell'elaborazione e nel controllo degli esercizi, soprattutto a risposta chiusa. Non è infrequente infatti che per alcuni valori dei parametri ci possano essere due risposte uguali o incoerenti. Questa maggiore attenzione viene però scontata in una maggiore facilità e rapidità di correzione delle prove.

2.4 Le tipologie di esercizi

Per i nostri scopi iniziali erano necessari solo due tipi di esercizi: i "test", cioè insiemi di esercizi a risposta chiusa, e i "problemi", cioè esercizi a risposta aperta con uno svolgimento articolato, come uno studio di funzione o la discussione e soluzione di un sistema lineare.

In vista della possibilità di usare lo stesso software anche in situazioni diverse si è però deciso fin da subito di non limitare la possibilità di scelta degli esercizi a quelle sopra citate, ma di implementarne altri tipi[4]:

(a) domande a risposta aperta "breve" in cui il risultato viene scritto in un apposito spazio alla fine dell'esercizio;

(b) "riempi lo spazio (*fillin*)": esercizi in cui devono essere riempiti alcuni spazi lasciati bianchi in proposizioni, teoremi, ecc.

(c) domande a risposta aperta "lunga";

(d) tabelle da completare;

(e) "*matching*": esercizi in cui bisogna collegare gli elementi di due liste come, ad esempio, una lista di funzioni con quella delle loro derivate.

Ovviamente tutti questi esercizi possono contenere parametri aleatori, possono essere riordinati casualmente e infine possono avere diverse apparenze "estetiche" (ad esempio lo spazio per la risposta può essere un riquadro, una riga sottolineata, ecc.)

Riuscire ad avere tutte queste caratteristiche, oltre a quelle relative al foglio soluzioni, ha comportato diverse difficoltà e a questo proposito vogliamo ringraziare il GUIT e in particolare il prof. Enrico Gregorio per i preziosi suggerimenti e consigli.

2.5 Il problema della randomizzazione: le nostre scelte

In LaTeX esistono diversi pacchetti che generano numeri casuali (per l'esattezza, pseudocasuali).

Come già detto la nostra scelta è stata quella di affidarci a un metodo "misto". La permutazione delle risposte chiuse ad una domanda è del tutto aleatoria così come la scelta dei parametri. Invece la scelta di una variante di un esercizio tra quelle proposte è aleatoria se il numero di varianti è

4. In particolare alcune tipologie sono nate da precise richieste in ambito di scuola secondaria per la gestione di prove di verifica.

maggiore di un valore prestabilito (attualmente è fissato a 8), mentre negli altri casi viene scelta (deterministicamente sulla base del numero di versione del compito) una permutazione tra 24 possibili permutazioni (6 nel caso le varianti siano 3). Un procedimento analogo viene utilizzato anche per la scelta dell'ordine degli esercizi nel compito.

Nel processo di randomizzazione il seme è stato scelto con una combinazione della data dell'esame e del numero della versione così che compiti "identici" assegnati in date diverse siano differenti. Inoltre nella generazione dei parametri aleatori interviene anche il numero d'ordine dell'esercizio nel compito generato, così che parametri definiti nello stesso intervallo, ma in esercizi diversi, siano differenti.

2.6 La struttura di un esame

Per la generazione di una prova d'esame è necessaria la creazione da parte dell'utente di un certo numero di file:

(a) I file contenenti i testi di ogni esercizio, ognuno con il suo numero di varianti.

(a) Un file "master" per la prova d'esame in cui l'utente interviene solo per indicare la data della prova, il numero di versioni da generare e, ovviamente, gli esercizi da inserire[5].

(b) Un file "master" per le soluzioni, identico al precedente, che genera le soluzioni per ogni versione della prova e la stringa delle risposte alle domande a risposta chiusa.

Vi è inoltre un altro file utilizzato per controllare il testo degli esercizi: per ogni esercizio, stampa tutte le varianti sia in versione parametrica che in una versione numerica. Ciò è stato possibile grazie a un comando che, in dipendenza di un'opzione del pacchetto, calcola o meno le espressioni matematiche presenti nell'esercizio.

Nel corso dell'elaborazione del pacchetto abbiamo anche implementato comandi per la semplificazione di frazioni, di radicali[6], per la formattazione degli esponenti, ecc..

Compito dei docenti è quindi quello di creare un database di esercizi, controllarli e scriverli in LaTeX utilizzando le istruzioni del software. Dopo di che, in previsione di una prova d'esame, vengono scelti gli esercizi e generati i compiti.

Oltre a questi file modificabili dall'utente, sono stati creati altri file che non devono in generale essere modificati dall'utente finale: il file `esami.sty` e un file, modificabile avendo un po' di conoscenza del linguaggio LaTeX, contenente le configurazioni "estetiche" quali il formato della testata, del pié di pagina e le istruzioni per lo studente che appaiono sul foglio dell'esame.

Le motivazioni per questo tipo di struttura sono molteplici:

(a) la creazione e il controllo degli esercizi è del tutto svincolata dalla generazione di un tema d'esame anche se il formato del file è identico: ciò permette la creazione di un vero e proprio database di esercizi che può essere incrementato quando se ne ha la possibilità;

(b) le modifiche "stilistiche" al tema d'esame possono essere effettuate senza bisogno di mettere mano al vero e proprio set di istruzioni;

(c) un utente con poche, se non nulle, conoscenze di LaTeX può essere in grado di generare il tema d'esame una volta che il database è pronto;

(d) poche modifiche al file master permettono di gestire situazioni molto diverse (esami in una o più parti, numero di domande diverso, diverse tipologie di esercizio, ecc.)

3 Il pacchetto

Come detto del pacchetto fanno parte un file di stile, un file di configurazione, due file master, un file per il controllo degli esercizi e, ovviamente, i file che costituiscono il database degli esercizi. Cominciamo da questi ultimi.

3.1 I file degli esercizi

Ogni esercizio (con tutte le sue varianti) va scritto in un file separato. Ogni variante è racchiusa nel comando `\newproblem` che è una versione (molto) modificata dell'analogo comando del pacchetto probsoln, tale comando ha come unico argomento il testo dell'esercizio.

Esercizi a risposta chiusa

Se l'esercizio è a risposta chiusa la sintassi è:

```
\item \PTs{punteggio}
 ... Testo ...
\begin{answers}{numero colonne}
    \bChoices[random]
\Ans0 risposta errata \eAns
\Ans0 risposta errata \eAns
\Ans1 risposta esatta \eAns
    \eFreeze
\Ans0 nessuna delle precedenti \eAns
    \eChoices
\end{answers}
```

nella quale:

- `\item \PTs{punteggio}`

  introduce una domanda con punteggio indicato in `\PTs` (può essere anche un numero decimale e il separatore può essere la virgola a differenza del pacchetto exerquiz);

5. In questo file è possibile modificare anche alcuni parametri che determinano l'aspetto dell'esame, ma per questo è necessaria una (minima) conoscenza di LaTeX.

6. Il risultato non è però ulteriormente semplificabile per cui, ad esempio, $\sqrt{8}/2$ si riduce a $2\sqrt{2}/2$

- `\begin{answers}{numero colonne}`
  `\bChoices[random]`
  `...`
  `\eChoices`
  `\end{answers}`

  introduce le risposte disposte sul numero di colonne specificate. Le risposte vengono permutate in modo casuale solo se è specificata l'opzione `random`;

- `\Ans0` introduce una risposta errata;

- `\Ans1` introduce la risposta esatta[7];

- `\eFreeze` introduce (se si vuole) una o più risposte che non saranno in ordine casuale.

Esercizi a risposta aperta

Se l'esercizio è un problema a risposta aperta verrà inserito nell'ambiente problem o problem* se ha una o, rispettivamente, più parti. La sintassi sarà:

```
\begin{problem} [punteggio]
 ... Testo ...
\begin{solution}[spazio_soluzione]
 ... Soluzione ...
\end{solution}
\end{problem}
```

In `punteggio` va il punteggio dell'esercizio, in `spazio_soluzione` va l'eventuale spazio bianco che deve essere lasciato per la soluzione. Oppure nel caso di esercizio in più parti:

```
\begin{problem*} [punteggio]
....testo....
\begin{parts}
\item \PTs{punteggio della parte}
....testo....
   \begin{solution}[spazio_soluzione]
    .... testo soluzione ......
    \end{solution}
\item \PTs{punteggio della parte}
.....
\end{parts}  \end{problem*}
```

dove con `\PTs{punteggio della parte}` si indica il punteggio della singola parte. Mentre il pacchetto exerquiz ha la possibilità di calcolare automaticamente il punteggio dell'esercizio con l'argomento `[⟨\auto⟩]`, l'introduzione della randomizzazione della scelta degli esercizi in esami ha tolto tale possibilità.

Altre tipologie di esercizi

Sono state definite alcune nuove tipologie di esercizio:

7. Il meccanismo adottato per riportare in un file esterno le soluzioni ha purtroppo escluso, per ora, la possibilità di avere domande con più di una risposta esatta

fillin Serve per creare esercizi "a riempimento": parte del testo viene lasciata vuota e deve essere riempita dallo studente. Può essere usato anche per creare esercizi a risposta aperta breve. La sintassi è `\fillin[`⟨*tipo*⟩`]{`⟨*larghezza dello spazio*⟩`}{`⟨*risposta*⟩`}` I due parametri obbligatori sono la larghezza dello spazio da lasciare, che deve essere espressa in cm, e la risposta esatta (parola o numero) che lo studente dovrebbe inserire (e che verrà stampata solo nel file delle soluzioni). Il parametro opzionale ⟨*tipo*⟩ definisce come deve essere segnalato lo spazio in cui inserire la risposta: `u` (*underlined*), che è l'opzione di default, fa sì che lo spazio sia evidenziato mediante una riga su cui va inserita la risposta, `e` (*empty*) crea uno spazio vuoto (non evidenziato in alcun modo), `b` (*boxed*) crea uno spazio circondato da un riquadro.

All'interno del comando `\fillin` (nello spazio per la risposta) non è possibile usare i comandi di semplificazione delle frazioni (si veda il paragrafo 3.4).

matching Serve per creare esercizi in cui le parole o formule contenute su due colonne vanno abbinate dallo studente in modo corretto. Le coppie vengono definite dal comando `\pair`: `\pair{`⟨*nome 1*⟩`}{`⟨*nome 2*⟩`}` che va inserito tante volte quanti sono i nomi o le formule da abbinare. L'elenco dei nomi (mischiati in ordine casuale su ciascuna colonna) viene visualizzato con il comando `\matching`. Ad esempio:

```
\pair{Italia}{Roma}
\pair{Germania}{Berlino}
\pair{Grecia}{Atene}
\matching
```

Grecia	Berlino
Italia	Atene
Germania	Roma

Nelle soluzioni viene invece visualizzato l'abbinamento corretto.

tabella serve per creare esercizi con più risposte aperte incolonnate. La sintassi è:

```
\begin{tabella}[num. col. visibili]
{allineamento colonne visibili}
{allineamento colonna nascosta}
 ... & ... \cr
\end{tabella}
```

Il primo parametro (il cui valore di default è 2) indica il numero di colonne della tabella il cui contenuto deve essere visibile anche nel testo dell'esercizio (e non solo nella soluzione), mentre è possibile avere solo una colonna il cui contenuto è invisibile nel testo ma appare nella soluzione. Il secondo parametro consente di scegliere l'allineamento delle

colonne il cui contenuto è sempre visibile e il terzo l'allineamento della colonna invisibile.

Ad esempio, con il codice:

```
\begin{center}
\renewcommand\arraystretch{3}
\begin{tabella}[1]{l}{l}
 \hline
Il dominio della funzione è &
$D_f=(-\infty;2]$\cr
 \hline
L'insieme immagine di $f(x)$ è &
 $Im(f)=(-\infty,0]$ \cr
 \hline
  \end{tabella}
\end{center}
```

si ottiene il seguente risultato (la seconda colonna è visibile solo nelle soluzioni):

Il dominio della funzione è	$D_f = (-\infty; 2]$
L'insieme immagine di $f(x)$ è	$Im(f) = (-\infty, 0]$

I successivi ambienti non definiscono una tipologia vera e propria di esercizi.

problema e problema* Servono per creare esercizi a risposta aperta. Il funzionamento è analogo a problem e problem*, la differenza sta nel fatto di poter eliminare (mediante l'opzione `solutionsonly`) il testo degli esercizi, riportando solo la soluzione.

risposta Serve per creare, in esercizi a risposta aperta lunga, lo spazio per inserire la risposta in un box o su righe. La sintassi del comando è:

```
\begin{risposta}{tipo}{spazio verticale}
 ...
\end{risposta}
```

Il primo parametro (`tipo`) definisce se lo spazio per la risposta deve essere riquadrato (opzione `b`, di default) oppure diviso in righe su cui andrà scritta la risposta (opzione `l`). Il secondo parametro definisce invece l'altezza dello spazio per la risposta (in cm se con opzione `b`, in numero di righe se con opzione `l`).

workarea serve a creare un'area di lavoro, cioè uno spazio bianco sul foglio in cui lo studente possa scrivere. La sintassi è:

```
\begin{solution}{lunghezza}
\end{solution}
\begin{workarea}{larghezza}{lunghezza}
\end{workarea}
```

La lunghezza dello spazio indicato per solution e della workarea dovrebbero essere uguali; in caso contrario il testo presente nella workarea viene posizionato male nello spazio della soluzione, sovrapponendosi al testo dell'esercizio se la sua lunghezza è maggiore di quella della soluzione. La larghezza della workarea è opzionale e di default è uguale alla larghezza del testo. A differenza dell'ambiente solution, nella workarea è possibile inserire testo, grafici, ecc.; ad esempio, è possibile inserire degli assi coordinati per disegnare un grafico.

3.2 I file master

I file `master` e `master-sol`

Questi due file differiscono solo perché il secondo mostra le soluzioni e contengono le istruzioni per generare effettivamente la prova d'esame. In entrambi è necessario indicare la data (comando `\date`) in formato ⟨*giorno*⟩/⟨*mese*⟩/⟨*anno*⟩ (giorno e mese possono essere scritti in qualsiasi formato: 3/12/2012, 03/7/2013; l'anno deve essere indicato con 4 cifre), il numero di compiti da generare (comando `\numcompiti`) e gli esercizi. Il tema d'esame può essere diviso in più parti e in ognuna si può usare uno o più degli ambienti introdotti e uno o più comandi per la stampa degli esercizi. Nel file possono essere definiti anche il numero massimo di ripetizioni che il comando `\FPsetpar` (si veda 3.4) deve eseguire per cercare di soddisfare le condizioni poste (comando `\maxLoopLimit`) e il meccanismo di calcolo del seme iniziale per la generazione dei numeri casuali (comando `\seme`). È possibile usare i classici meccanismi di sezionamento di LaTeX.

All'interno di questi file gli esercizi a risposta chiusa vengono inseriti nell'ambiente test che ha come argomento opzionale il punteggio (un intero positivo). Ogni gruppo di quesiti è inserito nell'ambiente questions[8].

Le altre tipologie di esercizi non hanno invece alcun ambiente particolare.

Il file `totale-versioni`

Il file `totale-versioni` serve per generare tutte le varianti di un esercizio contenuto in un file ed è usato in fase di costruzione del database degli esercizi con funzione di controllo. Ha sempre l'opzione prova (si veda 3.3); la compilazione genera una versione numerica e usando l'opzione param la versione parametrica degli esercizi. Al suo interno si deve usare un solo comando, `\esercizio{`⟨*nome*⟩`}`, che deve contenere il nome dell'esercizio da compilare. Se si compila la versione parametrica verranno indicati i parametri e i loro intervalli di variazione. Il meccanismo con cui opera il file è simile a quello di `\selectallproblems` del pacchetto probsoln.

3.3 Le opzioni del pacchetto

Il pacchetto esami presenta alcune opzioni:

8. In un ambiente test possono esistere più ambienti questions.

- allowrandomize e norandomize: la prima permette di permutare in modo casuale l'ordine delle risposte nelle domande a risposta chiusa, la seconda di non permutarlo (di default, le risposte vengono permutate in modo casuale);
- shuffle, shufflerandom e noshuffle: la prima permette di permutare in ordine casuale gli esercizi quando il numero è maggiore di 8 e in modo deterministico se minore, la seconda di permutarli casualmente quando il numero è maggiore di 6 e minore altrimenti; la terza di non permutarli (l'opzione di default è shuffle)[9];
- xxxx: fa caricare un file di configurazione (`esami-xxxx.cfg`) con ulteriori informazioni relative a questo compito (per esempio, il nome dell'esame e le istruzioni). Per far leggere un file di configurazione senza modificare il file `esami.sty` è sufficiente scrivere un'opzione sconosciuta (ad esempio zzz) e scrivere il file `esami-zzz.cfg`. È ovviamente possibile anche modificare il file `esami.sty` in modo da avere maggiore flessibilità nella configurazione;
- pointsonright: fa apparire una casella con il punteggio sulla destra dell'esercizio. Se l'opzione non è specificata, la casella non appare;
- nosolutions: questa opzione genera il compito senza soluzioni (default);
- solutions: genera il file delle soluzioni;
- solutionsonly: genera un file con le sole soluzioni, senza testo degli esercizi, usando l'ambiente problema;
- prova: si usa solo con il file `totale-versioni` e serve per visualizzare tutte le varianti di un esercizio. Fa visualizzare automaticamente le risposte esatte negli esercizi a risposta multipla e la soluzione nei problemi da svolgere;
- param: serve per visualizzare i nomi dei parametri casuali anziché il valore numerico attribuito. Stampa anche una frase in coda ad ogni esercizio, nella quale sono contenuti il valore iniziale e finale attribuiti ai parametri usati nell'esercizio. Può essere usata solo insieme all'opzione prova;
- correzione: stampa solo i testi di tutti gli esercizi di un compito, senza soluzioni (funziona solo nel file `totale-versioni`);
- fillb: da usare se nel compito ci sono esercizi di tipo `fillin`, per poter inserire le risposte corrette nella stringa delle soluzioni;
- twocolumns: per scrivere il testo delle domande a risposta chiusa su due colonne;
- sansserif: usa caratteri *sans serif*. Utile soprattutto se si usano le due colonne;
- autopston e autopstoff: caricano il pacchetto auto-pst-pdf e permettono di compilare il file direttamente con pdfLaTeX (anche se contiene codice di pstricks). La prima opzione crea e include direttamente nel documento i file delle immagini, la seconda non crea i file delle immagini (da utilizzare quando il documento non contiene immagini perché rende più veloce la compilazione)[10].

3.4 I comandi

I comandi per lavorare con i parametri

Come detto, uno degli obiettivi del pacchetto è l'uso di parametri casuali all'interno degli esercizi. Per i nostri scopi abbiamo definito solo parametri interi, ma nulla vieta di implemetare procedure simili per parametri razionali o in forma decimale. I parametri vengono definiti con il comando `\FPsetpar[`⟨*seme*⟩`]{`⟨*nome-parametro*⟩`}{`⟨*inf*⟩`\}{`⟨*sup*⟩`}[`⟨*valori esclusi*⟩`]`. Il parametro avrà nome `\nome-parametro` e varierà tra ⟨*inf*⟩ e ⟨*sup*⟩ (inclusi). Il parametro opzionale ⟨*seme*⟩ serve a imporre un seme diverso per la generazione del numero casuale. Il valore di default del parametro è `\seme` ed è definito a partire dalla data dell'esame e dal numero della versione. È possibile escludere dalla scelta uno o più valori (parametro ⟨*valori esclusi*⟩). Se i valori sono più di uno devono essere posti tra parentesi graffe e separati tra virgole; tra i valori esclusi ci possono essere anche altri parametri purché già definiti in precedenza (anche il primo e l'ultimo valore possono essere parametri: in questo caso bisogna prestare attenzione al fatto che il primo valore deve essere minore dell'ultimo). Il meccanismo usato per soddisfare tali condizioni consiste nel ripetere la generazione più volte: il numero massimo di ripetizioni è determinato dal parametro `\maxLoopLimit`, il cui valore di default è 10, ma che è definibile anche in fase di compilazione.

Ad esempio, i comandi:

```
\FPsetpar{a}{2}{10}[3]
\FPsetpar{b}{4}{12}[{\a,6}]
```

creano due variabili `\a` (che può assumere un valore casuale tra 2 e 10, escluso 3) e `\b` (che può assumere un valore tra 4, 12, escluso 6 e il valore

9. La scelta di 8 esercizi per l'opzione shuffle è dovuta al limite di 9 parameti dei comandi LaTeX, mentre abbiamo scelto 6 esercizi per l'opzione shufflerandom perché in tal caso la probabilità di esercizi simili vicini è per noi accettabile; tale numero può essere modificato commentando o scommentando una riga di codice nel file `esami.sty`.

10. Per usare auto-pst-pdf è necessario eseguire pdfLaTeX con l'opzione `write18`.

già assegnato ad \a). In questo caso, il seme è quello stabilito nel preambolo del documento.

Sui parametri è possibile eseguire operazioni sia usando i comandi del pacchetto fp sia utilizzando comandi da noi definiti.

Il comando \FPsv[⟨*cifre decimali*⟩]{⟨*operazione su parametri*⟩} serve ad eseguire operazioni su parametri e restituirne il valore numerico (con il numero prescelto di cifre decimali) o stampare l'operazione eseguita (se si sceglie l'opzione param nel pacchetto esami). Ad esempio se $k = 2$: \FPsv{2*k+1} produce 5 o $2 * k + 1$ con l'opzione param; \FPsv[2]{(2*k+1)/2} produce 2.50 o $(2 * k + 1)/2$.

La sintassi delle operazioni è la stessa del pacchetto fp e se il pacchetto usa l'opzione param ogni operazione viene racchiusa tra parentesi tonde per comodità di lettura.

Il comando \FPval{⟨*nome*⟩}{⟨*operazione su parametri*⟩} assegna a ⟨*nome*⟩ il valore dell'operazione arrotondato (è il comando \FPeval di fp modificato) o stampare l'operazione eseguita se si sceglie l'opzione param nel pacchetto esami. Ad esempio il codice:

```
\FPsetpar{k}{1}{3}
\FPval{a}{2*k+1}
\FPsetpar{b}{2}{20}[\a]
```

genera una variabile \b che assume un valore casuale tra 2 e 20, escluso (in questo caso) il valore assegnato a \a. Nella versione parametrica appare invece una stringa di questo tipo (per quanto riguarda la variabile \b):

> Il parametro b varia da 2 a 20. $b \neq (2 * k + 1)$.

Sono stati definiti anche alcuni comandi per la semplificazione di frazioni che possono essere usati anche per la formattazione del testo. Il comando \sempli{⟨*num*⟩}{⟨*den*⟩} serve per semplificare frazioni in cui il numeratore o il denominatore contengono parametri. Il comando \sempli{2*k}{3*k+1} per $k = 1$ genera $\frac{1}{2}$.

Il comando \semplix{⟨*num*⟩}{⟨*den*⟩} serve per semplificare frazioni in cui il numeratore o il denominatore contengono parametri, ma nelle quali il risultato pari ad 1 non deve apparire e il risultato -1 deve apparire come $-$ (ad esempio perché davanti ad una incognita). Il comando può essere utilizzato ponendo il denominatore uguale ad 1 anche per la formattazione di coefficienti. Ad esempio, se il parametro k assume valore 2: \FPsv{k-1}x fornisce $1x$ mentre \semplix{k-1}{1}x dà x.

Il comando \esempli{⟨*num*⟩}{⟨*den*⟩} serve per semplificare delle frazioni in cui il numeratore o il denominatore contengono parametri, ma nelle quali il risultato pari ad 1 non deve apparire, ma deve apparire esplicitamente il valore -1 (ad esempio per gli esponenti). Il comando può essere utilizzato ponendo il denominatore uguale ad 1 anche per la formattazione di esponenti. Ad esempio, se il parametro k assume valore 2: x^{\FPsv{k-1}} fornisce x^1 mentre x^{\esempli{k-1}{1}} dà x.

Il comando \sempliz{⟨*num*⟩}{⟨*den*⟩} serve per semplificare frazioni in cui il numeratore o il denominatore contengono parametri e che possono assumere il valore 0 (ad esempio nelle risposte.)[11]

Il comando \simpsqrt{⟨*ind*⟩}{⟨*rad*⟩} serve per "portar fuori" da una radice tutti i fattori possibili, su tali fattori non è possibile eseguire altre operazioni (semplificazioni, ecc.). Il primo parametro obbligatorio, ⟨*ind*⟩ è l'indice della radice e può essere un numero o un parametro; il secondo, ⟨*rad*⟩ il radicando e può essere un numero, un parametro o un'operazione su parametri. Ad esempio con $a = 2$ e $b = 1$, \simpsqrt{2}{a^2+4*b} fornisce $2\sqrt{2}$.

I comandi per la gestione degli esercizi e delle liste

I comandi per la gestione degli esercizi e delle liste rientrano in un'unica categoria in quanto hanno lo stesso tipo di funzionamento: dato un elenco di nomi, riordinare tale elenco in modo casuale ed estrarre poi un certo numero di elementi.

Il comando fondamentale per la gestione degli esercizi è \esercizi{⟨*file 1*⟩, ⟨*file 2*⟩, ..., ⟨*file n*⟩} che sceglie un esercizio casuale da ognuno dei file specificati, riordina casualmente gli n esercizi scelti e li manda in output.

Il comando \estrai[⟨*m*⟩]{⟨*lista*⟩}{⟨*nome*⟩} estrae $n - m$ elementi da ⟨*lista*⟩; gli elementi estratti avranno nome \nomei, \nomeii, e via di seguito. Ad esempio con il comando \estrai[1]{insiemi,logica,potenze}{alpha} si scelgono 2 elementi dell'insieme e questi due elementi si chiameranno \alphai e \alphaii. Se questi nomi corrispondono a nomi di file di esercizi è possibile inserire nella lista di file argomento del comando \esercizi anche \alphai e \alphaii.

Il comando \estraialfa{⟨*n*⟩}{⟨*lista*⟩}{⟨*nome*⟩} estrae ⟨*n*⟩ oggetti casuali da ⟨*lista*⟩, mantenendo l'ordine fissato. Gli elementi estratti si chiameranno \nomei, \nomeii e via discorrendo. Ad esempio con il comando \estraialfa[2]{a,b,c,d}{alpha} si scelgono 2 elementi dell'insieme, mantenendo l'ordine alfabetico e questi due elementi si chiameranno \alphai e \alphaii.

Infine il comando \estraies[⟨*m*⟩]{⟨*lista*⟩} funziona come il comando \estrai, ma esclusivamente su una lista di esercizi; una volta effettuata l'estrazione, manda gli elementi estratti in output come il comando \esercizi.

Tali comandi permettono ulteriori meccanismi aleatori di scelta degli esercizi: ad esempio, se il database contiene 20 esercizi su un dato argomento e se ne vogliono inserire nel compito 4, un comando

11. Con gli altri comandi di semplificazione il risultato 0 dà luogo a un errore e al blocco della compilazione.

permette la scelta casuale dei 4 esercizi dell'insieme da cui poi estrarrà una variante.

È possibile sia inserire tutti gli esercizi con un unico comando, sia usare più comandi `\esercizi` o `\estraies`. Questa possibilità è utile ad esempio se si vuole che ci siano esercizi da due sottoinsiemi diversi (ad esempio 5 esercizi sui limiti scelti tra 7 e 3 esercizi sugli asintoti scelti tra 5) o se si vuole, per esigenze di impaginazione, avere alcuni esercizi su due colonne ed altri su una colonna.

Riferimenti bibliografici

Arseneau, D. «Il pacchetto random». `CTAN:/macros/generic/misc/random.tex`.

Mehlich, M. (1999). «Il pacchetto fp». `CTAN:/macros/latex/contrib/fp`.

Messineo, G. e Vassallo, S. (2012). «LaTeX per la creazione di temi d'esame. Seconda parte: esami online». Contributi di ricerca in Econometria e Matematica 125, Università Cattolica del S. Cuore, Milano.

Penna, S. (2006). «Test interattivi di matematica e fisica on-line: il LaTeX come strumento di sviluppo». ArsTEXnica, **2**, pp. 65–70.

Pilu, A. (2011). «La creazione di una prova d'esame». ArsTEXnica, **11**, pp. 5–14.

Story, D. P. (2012). «Exerquiz & AcroTeX». URL `http://www.acrotex.net/`.

Talbot, N. L. (2011). «Il pacchetto probsoln». `CTAN:macros/latex/contrib/probsoln`.

▷ Grazia Messineo
Università Cattolica Milano – ITC "G. Falcone" Corsico
`grazia dot messineo at unicatt dot it`

▷ Salvatore Vassallo
Università Cattolica Milano
`salvatore dot vassallo at unicatt dot it`