Składanie tabel METAPOSTem

Piotr Bolek
P.Bolek@gust.org.pl

Pracę zgłosił: Andrzej Borzyszkowski

Streszczenie

W tekście jest przedstawiony METAPOSTowy zestaw makr, który służy do stosunkowo niekłopotliwego składania tabel. Opisane są problemy związane z implementacją, wynikające m.in. ze specyficznego sposobu integracji METAPOSTa z TEXem. Na kilku przykładach pokazane są także możliwości, jakie daje taki sposób składania tabel.

Wprowadzenie

Tabele można w TEXu składać używając do tego różnego rodzaju narzędzi. Na najbardziej podstawowym poziomie można wykorzystywać TEXowe polecenia wbudowane takie jak `\halign` czy `\valign`. Dają one duże możliwości składania tabel, ale nie zawsze są wygodne w użyciu i nie wszystko można z ich pomocą zrobić. W szczególności wykorzystanie kolorów czy składanie tabel o zadanych wymiarach kolumn, wierszy, albo uzależnianie szerokości kolumn (typu kolumna pierwsza ma być dwa razy szersza niż druga), jest kłopotliwe.

Istnieją różne pakiety makr wyższego poziomu, dostępne np. w LATEXu, pozwalające wygodniej składać tabele i oferujące większe możliwości. Na pewno na uwagę zasługuje także pakiet `tap`, który dzięki wykorzystaniu m.in. możliwości jakie daje integracja TEXa z PostScriptem, pozwala np. kolorować wybrane części tabel.

Wszystkie jednak wspomniane powyżej mechanizmy składu tabel wykorzystują wbudowane niskopoziomowe polecenia `\halign` lub `\valign`.

Zastanawiając się nad tym czym są tabele i jak można je opisywać, dochodzi się do oczywistego wniosku, że tabele to obiekty w dużej części graficzne. Zawartości rubryk, zwłaszcza zawierające więcej tekstu, powinny być składane jako tekst i TEX się do składania zawartości rubryk jak najbardziej nadaje. Ale cała tabela jako taka jest wyrównywanym w pewien uporządkowany sposób zestawem prostokątnych, powiązanych ze sobą rubryk. Rubryki te często powinny być wyróżniane wizualnie: przez podłożenie kolorowego tła lub wzorku albo otoczenie specjalną ramką itp.

Dlatego przyszła mi kiedyś myśl, żeby wykorzystać METAPOSTa (oczywiście w połączeniu z TEXem) do składania tabel. Okazją i motywacją bezpośrednią była pomoc przy składzie książki, nad którą pracowała znajoma. W książce tej były tabele z kolorowanymi rubrykami, które znakomicie nadawały się jako materiał do testowania METAPOSTowych makr do składania tabel.

Idea

Założenie początkowe było takie, aby sposób definiowania tabel był w miarę naturalnym dla kogoś, kto składał wcześniej tabele w LATEXu. Dlatego zawartość tabeli jest definiowana w postaci kolejnych rubryk, wiersz po wierszu.

Zawartość rubryk tabeli z założenia miała być składana TEXem, i to bez konieczności zmuszania użytkownika, który przyzwyczaił się w LATEXu do używania w tabelach konwencji plainowych i odwrotnie.

Kolejnym założeniem było umożliwienie automatycznego ustalania szerokości kolumn przez METAPOSTa. W szczególności, by możliwe było podawanie względnej szerokości kolumn (np. że kolumna piewsza ma być dwa razy szersza niż druga, na podobnej zasadzie jak w HTMLu).

METAPOSTowi oczywiście należało także zostawić kwestie wyrównywania zawartości rubryk, rysowanie ramek, teł i innych elementów graficznych, które będą potrzebne przy składaniu tabeli.

Problemy realizacyjne

Przy realizacji zestawu makr uwzględniającego powyższe założenia oczywiście wystąpiły pewne problemy, z których wspomnieć warto dwa:

- przekazanie wyliczonej przez METAPOSTa szerokości rubryki do TeXa w celu złożenia zawartości rubryki na zadaną szerokość;
- mechanizm łączenia rubryk w pionie, poziomie lub w obu kierunkach na raz.

Przekazywanie szerokości. Ponieważ szerokość rubryk ma być wyliczana METAPOSTem, a skład zawartości robiony przez TeXa, to konieczne jest przekazanie wyliczonej przez METAPOSTa wartości TeXowi. Z pozoru wydaje się to proste, bo przecież standardowo etykiety METAPOSTowe są składane przez TeXa. Zawartość rubryki w ogólnym przypadku powinna być składana w trybie akapitowym na wyliczoną szerokość. Narzucającym się sposobem jest wykorzystanie mechanizmu dynamicznej ewaluacji kodu METAPOSTowego poleceniem `scantokens`. Ale wstawki TeXowe, mimo że traktowane jako obiekty typu `picture`, nie podlegają ewaluacji i użycie `scantokens` nie jest możliwe.

Dlatego w celu rozwiązania tego problemu konieczne była własna implementacja mechanizmu ewaluacji, realizowana przez zapisanie danych poleceniem `write` w zewnętrznym tymczasowym pliku i następnie wczytanie tego pliku poleceniem `input`.

Łączenie rubryk. Mechanizm łączenia rubryk w pionie (między wierszami) lub poziomie (między kolumnami) też sprawia problemy. Makra do składania prostych tabel, niezawierające rubryk łączonych, zrelizować jest nietrudno, bo łatwo jest wyliczyć wymiary wierszy i kolumn korzystając z faktu, że rubryki układają się w regularną siatkę. Uwzględnienie łączenia kolumn zaburza tę miłą regularność i znacznie komplikuje algorytm wyliczania szerokości rubryk. Dlatego zdecydowałem się na obejście tego problemu. W jaki sposób, o tym w dalszej części tekstu.

Przykłady użycia

Użycie METAPOSTa do składania tabel z wykorzystaniem makr `mp-tab` nie jest trudne. Do efektywnego i efektownego ich użycia oczywiście bardzo pomocna jest pewna znajomość METAPOSTa. Ale nawet jeśli ktoś nie używał METAPOSTa wcześniej, powinien sobie poradzić.

Definicja tabeli składa się z kilku części. Najpierw należy podać różne parametry, takie jak: szerokość tabeli, szerokości kolumn, sposób wyrównywania zawartości kolumn, marginesy wewnątrz rubryk itd.

Potem, poleceniem `tab_row`, podaje się zawartość kolejnych wierszy tabeli.

Następnie, po podaniu polecenia `tab_set`, zawartość tabeli jest składana. Poszczególne rubryki są składane przez TeXa w pudełkach o wyliczonych przez METAPOSTa rozmiarach. Wyniki tego procesu są zapamiętywane w wewnętrznych strukturach danych.

Ostatnim etapem jest wizualizacja różnych elementów tabeli. W najprostszym przypadku, używając polecenia `tab_type` i `lines`, uzyskujemy „zwykłą” tabelkę.

Przykład 1. Prostym przykładem jest tabela 1. Poniżej pokazany jest sposób realizacji.

```
tab_init(225pt);
cell_pad(1mm);
cell_align(LLFT);
col_widths(2a,a,a);

tab_row("\bfseries Nazwa",
  "\bfseries pt",
  "\bfseries mm");
tab_row("Brylant",    "3", "1,128");
...
tab_row("Kwadrat",     "48", "18,051");

tab_set;
tab_type;
lines;
```

Wiersz 1 inicjuje tabelę i ustawia jej szerokość na 225 pt.

W wierszu 2 określamy wewnętrzny margines w rubrykach, a w 3 sposób wyrównywania zawartości rubryk.

Wiersz 4 określa liczbę i szerokości kolumn. Szerokości kolumn są tutaj podane względnie. Korzystamy tu z miłej własności METAPOSTa, jaką jest rozwiązywanie równań. Ponieważ suma szerokości kolumn musi być równa szerokości całej tabeli, taka reguła jest „zaszyta" wewnątrz makr pakietu. Szerokości kolumn możemy więc określać względnie, np. określając: „Niech kolumna pierwsza będzie dwa razy szersza do drugiej" i pozostawiając wyliczenie konkretnych wartości METAPOSTowi. Aby to osiągnąć, wystarczy (podając szerokości) użyć zmiennej o nieokreślonej wartości. To właśnie zrobiliśmy w wierszu 4. Kolumna pierwsza będzie dwa razy szersza niż każda z następnych kolumn, a ponieważ kolumn jest w sumie trzy, to będzie zajmowała dokładnie połowę szerokości tabeli. Łatwo sprawdzić, że tak jest rzeczywiście.

Wiersze od 6 do 11 zawierają zawartość kolejnych wierszy tabeli. Zawartość podajemy w postaci napisów (w cudzysłowach), ale będą one przetworzone „w trybie" `btex ... etex`, z wykorzystaniem symulowanego przez zapisanie do pliku mechanizmu ewaluacji, o którym wspomniałem wcześniej.

Wiersz 13 powoduje złożenie tabeli, a wiersze 14 i 15 pokazanie zawartości rubryk i narysowanie ramek.

W naszym pierwszym, prostym przykładzie nie widać nic specjalnego – taką tabelkę łatwiej nawet zrobić używając `\halign`, standardowych LaTeXowych środowisk tabelkowych albo pakietu `tap`. Ale popatrzmy dalej.

Przykład 2. W drugim przykładzie robimy tabelkę z tą samą zawartością, ale nadamy jej bardziej wyrafinowaną formę.

```
tab_init(225pt);
cell_pady(1.15mm);
cell_padx(3mm);
col_widths(2a,a,a);
row_height(1,1.2cm);
row_height(2,1cm);
cell_align(LRT);
col_align(1,LLFT);
col_align(2,S);
cell_align(3,3,M);
```

Nazwa	**pt**	**mm**
Brylant	3	1,128
Diament	4	1,504
Perl	5	1,880
Nonparel	6	2,256
Kolonel	7	2,632
Petit	8	3,008
Borgis	9	3,383
Garamond	10	3,761
Cycero	12	4,513
Średnian	14	5,256
Tercja	16	6,017
Dwugarmond	20	7,521
Półkwadrat	24	9,026
Dwuśrednian	28	10,530
Dwutercja	32	12,034
Konkordans	36	13,538
Kwadrat	48	18,051

Tabela przykładowa nr 1

```
11.
12. tab_row("", "", "");
13. tab_row("", "", "");
14. tab_row("",
15.   "w punktach typograficznych",
16.   "w mm");
17. ...
18. tab_set;
19.
20. evenrows_bg(0.8white);
21. unfill cells_rect(1,1,3,3);
22.
23. tab_type;
24. vlines;
25.
26. draw cells_rect(2,1,3,1);
27. draw cells_rect(3,2,3,3);
28.
29. fill cells_rect(1,1,1,3) withcolor .6white;
30. draw cells_rect(1,1,1,3);
31. multi_cell(1,1,1,3, "\bf Stopnie pisma", M);
32.
33. unfill cells_rect(2,2,2,3);
34. draw cells_rect(2,2,2,3);
35. multi_cell(2,2,2,3, "Wielkość pisma", M);
36.
37. multi_cell(2,1,3,1, "Nazwa", M);
38.
39. drawoptions(withpen pensquare scaled 2);
40. olines;
```

Wiersze 2 i 3 zawierają określenie wewnętrznych marginesów: poziomych i pionowych w rubrykach.

W wierszach 5 i 6 określamy minimalną wysokość wiersza pierwszego i drugiego – jeśli naturalne wysokości tych wierszy będą mniejsze niż podane, to zostaną one rozciągnięte pionowo do podanego rozmiaru. W wierszach 7–10 określamy wyrównywanie zawartości rubryk

Dziwne mogą wydać się wiersze 12 i 13. Są w nich po trzy puste rubryki. Wiersze te służą w tym momencie jedynie jako rezerwacja miejsca dla łączonych w poziomie rubryk obejmujących więcej niż jedną kolumnę. Rubryki te zostaną zdefiniowane później.

Wiersze 14, 15 i 16 opisujące trzeci wiersz tabeli mają pierwszą rubrykę pustą, bo w tym miejscu pojawi się rubryka łączona w pionie i obejmująca wiersz drugi i trzeci. Będzie ona później zdefiniowana.

Definicje kolejnych wierszy tabeli wyglądają tak samo, jak w przykładzie pierwszym.

Dużo bardziej skomplikowana jest część wizualizacyjna naszej tabeli. W wierszu 20 kolorujemy tło parzystych wierszy, a w 21 kasujemy tła dla pierwszych trzech wierszy zawierających nagłówek tabeli.

Teraz (wiersz 23) możemy już pokazać zawartość rubryk i narysować pionowe linie rozdzielające kolumny (wiersz 24).

Do zrobienia został jeszcze nagłówek tabeli zawierający rubryki łączone w pionie i poziomie. W wierszach 26 i 27 rysujemy część ramek nagłówka. Wreszcie dodajemy rubryki łączone, wypełniając i obrysowując prostokątne obszary obejmujące kilka sąsiednich rubryk: w wierszach 29–31 wszystkie trzy rubryki pierwszego wiersza tabeli, w 33–35 dwie ostatnie rubryki wiersza drugiego.

W wierszu 37 umieszczamy zawartość rubryki z pierwszej kolumny obejmującej drugi i trzeci wiersz tabeli.

Na koniec obrysowujemy całą tabelę grubszą ramką, wykorzystując w tym celu standardowe METAPOSTowe polecenie `drawoptions`, zmieniające domyślne atrybuty rysowania.

Przykład 3. Jeśli rubryka zawiera więcej tekstu, a nie chcemy w pliku wejściowym zostawiać bardzo długich wierszy, możemy wykorzystać standardowy METAPOSTowy mechanizm sklejania ciągów znaków, tak jak w poniższym fragmencie kodu. Przy sklejaniu napisów należy pamiętać o umieszczeniu dodatkowej

Stopnie pisma		
Nazwa	Wielkość pisma	
	w punktach typograficznych	w mm
Brylant	3	1,128
Diament	4	1,504
Perl	5	1,880
Nonparel	6	2,256
Kolonel	7	2,632
Petit	8	3,008
Borgis	9	3,383
Garamond	10	3,761
Cycero	12	4,513
Średnian	14	5,256
Tercja	16	6,017
Dwugarmond	20	7,521
Półkwadrat	24	9,026
Dwuśrednian	28	10,530
Dwutercja	32	12,034
Konkordans	36	13,538
Kwadrat	48	18,051

Tabela przykładowa nr 2

spacji przed cudzysłowem otwierającym kontynowany tekst, albo na końcu ciągu, do którego będziemy coś przyklejać.

```
tab_row("\texttt{min_row_height}",
  "1 -- wysokość wiersza",
  "Określenie minimalnej wysokości wierszy."
  & " Wszystkie wiersze, których naturalna"
  & " wysokość będzie mniejsza zostaną "
```

ULFT NW TL LT	TOP N T	URT NE TR RT
W LFT L	C M	E RT R
SW BL LB LLFT	B S BOT	SE LR RL LRT

```
& "rozciągnięte do podanego rozmiaru.");
```

TEXnikalia

Na koniec nieco istotnych informacji dotyczących składania tabel z wykorzystaniem pakietu `mp-tab`.

TEX a LATEX. Jak wspomniałem, zawartości rubryk można składać TEXem albo LATEXem w zależności od upodobań. Oczywiście METAPOST musi wiedzieć, jaką wersję TEXa ma wywoływać do składania tekstów. Jeśli używamy plaina, to wywołujemy METAPOSTa bez żadnych dodatkowych opcji i włączamy plik, a makra tabelkowe włączamy wywołując: `input mp-tab-plain.mp`.

Jeśli chcemy do składania tabel użyć LATEXa, to wywołujemy METAPOSTa z opcją `--tex=platex` albo ustawiamy zmienną środowiska `TEX=platex`, a w pliku źródłowym na początku wczytujemy plik `mp-tab-ltx.mp`.

Określanie sposobu wyrównywania zawartości rubryk. Wartości jakie można podać jako argumenty makr `cell_align`, `row_align`, `col_align`, są zebrane w tabeli obok.

Spis makr pakietu. Zestawienie makra używanych do opisywania tabel jest podane w tabeli na końcu tekstu.

Podsumowanie

Realizacja pakietu makr pozwalającego składać tabele z wykorzystaniem TEXa i METAPOSTa jest kolejną prezentacją możliwości jakie daje łączenie tych dwóch narzędzi.

Z drugiej strony, wynikające z zaszłości historycznych problemy, które trzeba rozwiązywać (czy raczej obchodzić) przy okazji zaprzęgania do współpracy tych programów, trochę dokuczają.

Polecenie	Argumenty	Znaczenie
Parametry		
tab_defs	lista ciągów znaków	Globalne definicje lub ustawienia parametrów dla TEXa, decydują o sposobie formatowania zawartości rubryk tabeli. Mogą tu być zdefiniowane makra których będziemy używać do formatowania tekstu w rubrykach.
cell_pad cell_padx cell_pady	1 – odstęp	Marginesy wewnątrz rubryk. cell_pad ustawia oba marginesy na tę samą wartość, dwa pozostałe służą do ustawienia marginesu poziomego i pionowego
tab_init	1 – szerokość tabeli	Zainicjowanie makr, ustawienie szerokości tabeli. Można nie podać szerokości tabeli (wywołaniu należy podać puste nawiasy), ale wtedy trzeba jawnie podać szerokości wszystkich kolumn w poleceniu col_widths
col_widths	tyle ile kolumn – szerokości poszczególnych kolumn	Określenie liczby i szerokości kolumn. Szerokości mogą być podawane względnie. Jeśli w tab_row podaliśmy jawnie szerokość tabeli, to szerokość co najmniej jednej kolumny musi być pozostawiona do wyliczenia Metapostowi.
min_row_height	1 – wysokość wiersza	Określenie minimalnej wysokości wierszy. Wszystkie wiersze, których naturalna wysokość będzie mniejsza zostaną rozciągnięte do podanego rozmiaru.
row_height	2 – nr i wysokość wiersza	Określenie minimalnej wysokości podanego wiersza.
cell_align	1 – domyślny sposób wyrównywania dla wszystkich rubryk 3 – nr wiersza, nr kolumny, sposób wyrównania konkretnej rubryki	Określenie sposobu wyrównywania zawartości rubryk.
col_align	2 – nr kolumny i sposób wyrównywania	Sposób wyrównywania zawartości w podanej kolumnie.
row_align	2 – nr wiersza i sposób wyrównywania	Sposób wyrównywania zawartości w podanym wierszu.
Zawartość		
tab_row	tyle ile kolumn w tabeli	Zawartość kolejnych rubryk w wierszu tabeli
multi_cell	6 – wiersz początkowy, kolumna początkowa, wiersz końcowy, kolumna końcowa, zawartość, wyrównanie	Umieszczenie tekstu w polu obejmującym kilka rubryk pozwalające uzyskiwać efekty typu „multicolumn”, „multirow” albo oba jednocześnie (nie można użyć przed użyciem tab_set)
Wizualizacja		
tab_set	brak	Złożenie tabeli, nie ma żadnego efektu wizualnego, powoduje tylko zapisanie wewnętrznych stuktur danych, które będą wykorzystane później do wizualizacji poszczególnych elementów tabeli
hlines vlines ihlines ivlines ohlines ovlines ilines olines lines	brak	Narysowanie linii poziomych lub pionowych rozdzielających rubryki. Wersje z „i” rysują tylko linie wewnętrzne, z „o” – tylko zewnętrzne.
row_bg col_bg	2 – numer wiersza lub kolumny, kolor	Wypełnienie podanego wiersza lub kolumny podanym kolorem
oddrows_bg evenrows_bg oddcols_bg evenrows_bg	1 – kolor	Wypełnienie parzystych lub nieparzystych wierszy lub kolumn podanym kolorem
checker	2 – kolory	Wypełnienie tabeli „w szachownicę” podanymi kolorami
cell_rect	2 – nr wiersza, nr kolumny	Ścieżka (prostokąt) podanej rubryki
cells_rect	4 – nr wiersza początkowego, nr kolumny początkowej, nr wiersza końcowego, nr kolumny końcowej	Ścieżka (prostokąt) obejmujący podany zakres rubryk
tab_type	brak	Wydrukowanie zawartości rubryk tabeli.