# LaTeX

## Pakiet `blkarray`

Włodzimierz Macewicz

### 1. Wstęp

Tworzenie tabel nie jest najmocniejszą stroną LaTeX-a. Sposób ich tworzenia nie zmienił się od chwili powstania LaTeX209 (ostatnia wersja pochodzi z 1992 r.) i nieco odstaje od możliwości innych, nowych, języków np: HTML. Głównym mankamentem jest konieczność określania szerokości kolumn, w których zawartość rubryk ma być składana tak, jak normalny akapit (niestety, jak do tej pory nie został zaimplementowany algorytm dopasowujący automatycznie szerokość kolumn, tak aby skład tabeli wyglądał dobrze), trudności są również z łączeniem sąsiednich rubryk w pionie (w ramach kolumny) oraz z uzyskaniem ładnych obramowań tabeli (linie podwójne). Pewne mankamenty są rozwiązywalne za pomocą dodatkowych pakietów (`array`, `multirow`, `hhline`, `tabularX`).

W tym tekście chciałbym opisać moje nowe odkrycie – pakiet `blkarray` autorstwa Davida Carlisle, który wnosi trochę świeżego powiewu w dziedzinie składu tabel. Ostatnia wersja pakietu pochodzi z marca 1999 r. i niestety ma kilka błędów, na które natknąłem się tworząc poniższy opis. Wniosek jest jeden; niewiele osób zauważyło istnienie tego pakietu, a jeszcze mniej próbowało go użyć. Wydaje mi się, że warto go wykorzystać. Zauważone błędy zgłosiłem do Autora i otrzymałem informację, że zostaną one wkrótce poprawione. Mam nadzieję, że w momencie wydania Biuletynu pakiet będzie już bez błędów – proszę o pobranie nowej wersji z archiwum CTAN (np. `ftp://ftp.icm.edu.pl/pub/tex/macros/latex/contrib/supported/carlisle`).

Pakiet nie rozwiązuje wszystkich problemów, ale zawiera konstrukcje ułatwiające skład tabel (w szczególności zawiera polecenia równoważne ze zdefiniowanymi w pakietach: `array`, `multirow`, `hhline` – takie „trzy w jednym”). Umożliwia on tworzenie poprawnych obramowań zbudowanych z podwójnych linii (pakiet `hhline`) i linii kropkowanych, nowe typy kolumn (pakiet `array`), tworzenie nowych typów kolumn (pakiet `array`), łączenie rubryk tabeli w pionie (pakiet `multirow`), wstawienie wierszy o innym składzie niż określono to w preambule tabeli, wstawienie materiału nie podlegającego podziałowi na kolumny oraz możliwość definiowania konstrukcji nawiasowych (z dodatkowym opisem), które obejmują kilka wierszy. Oczywiście została zachowana możliwość łączenia rubryk tabeli w poziomie (standardowe polecenie `multicolumn`).

Opis techniczny pakietu znajduje się również w zasobach Wirtualnej Akademii TeX-a. Korzystając z okazji przypominam, że w WA – pod adresem `http://www.ia.pw.edu.pl/~wujek/tex/index.html` – znajduje się wiele adresów stron zawierających dokumentację do TeX-a oraz LaTeX-a; a pod adresem: `http://www.ia.pw.edu.pl/~wujek/tex/latex2e/latex2e.html` opisy pakietów do LaTeX-a (w chwili obecnej dokumentacja w języku polskim do 117 godnych uwagi pakietów).

Zwracam się także z apelem do wszystkich użytkowników LaTeX-a o poświęcenie odrobiny swojego czasu i zrobienie opisu do pakietu (w języku polskim), który jest według Państwa interesujący, a którego właściwości Państwo poznali przy okazji swojej działalności zawodowej lub hobbystycznej.

### 2. Opis środowiska `blkarray`

Pakiet definiuje środowisko `blockarray` zastępujące jednocześnie środowiska `tabular` i `array` (wewnątrz trybu matematycznego zachowuje się jak `array`, a w trybie tekstowym jak `tabular`). Postać wywołania jest następująca:

\begin{blockarray}{*preambuła*}
*zawartość tabeli*
\end{blockarray}
lub
\begin{blockarray}[*s*]{*preambuła*}
*zawartość tabeli*
\end{blockarray}
gdzie:

- *s* jest parametrem opcjonalnym i określa sposób pozycjonowania tabeli względem sąsiedztwa; może przyjmować wartości:
  - `c` – środek wysokości tabeli na wysokości sąsiadującej linii (wartość domyślna),
  - `b` – dół tabeli równo z linią sąsiednią,
  - `t` – góra tabeli równo z linią sąsiednią,
- *preambuła* definiuje sposób prezentacji (składu) poszczególnych kolumn tabeli oraz określa separatory kolumn; mogą się w niej znaleźć następujące oznaczenia:

<table>
<tr><th>Symbol</th><th>Znaczenie</th></tr>
<tr><td>l</td><td>dosunięcie tekstu do lewej</td></tr>
<tr><td>r</td><td>dosunięcie tekstu do prawej</td></tr>
<tr><td>c</td><td>centrowanie</td></tr>
<tr><td>m{width}</td><td>definiuje kolumnę o szerokości width. Środek rubryki w tej kolumnie (połowa wysokości) pokrywa się ze środkiem rubryki kolumny sąsiedniej. Odpowiednik: \parbox{width}; tekst jest formatowany z wyrównaniem do lewego i prawego marginesu</td></tr>
<tr><td>p{width}</td><td>równoważna: \parbox[t]{width}. Pozycjonowanie w/g szczytu rubryki</td></tr>
<tr><td>b{width}</td><td>równoważna \parbox[b]{width}. Pozycjonowanie w/g dołu rubryki</td></tr>
<tr><td>>{macro}</td><td>może być użyte przed jednym z następujących typów: l, r, c, p, m lub b. Polecenie macro będzie wykonywane zawsze przed rozpoczęciem analizy materiału każdej rubryki w tak określonej kolumnie</td></tr>
<tr><td><{macro}</td><td>może być użyte po jednym z następujących typów: l, r, c, p, m lub b. Polecenie macro będzie wykonywane zawsze po analizie materiału każdej rubryki w tak określonej kolumnie</td></tr>
<tr><td>|</td><td>wstawienie linii pionowej jako separatora kolumn; odległość między kolumnami zostanie zwiększona o szerokość linii (w odróżnieniu od oryginalnej definicji tabeli). Można wstawić dwa znaki | – separatorem będzie linia podwójna (odległość między liniami określa parametr \BAdoublerulesep)</td></tr>
<tr><td>@{macro}</td><td>może być użyte w dowolnym miejscu preambuły zamiast separatora (np.: |). Polecenie macro zostanie wywołane między kolumnami (kasowana jest normalna odległość między kolumnami, może być ono użyte do określenia innej odległości między kolumnami)</td></tr>
<tr><td>!{macro}</td><td>może być użyte w dowolnym miejscu preambuły zamiast separatora (np.: |). Polecenie macro zostanie wywołane między kolumnami (w odróżnieniu od @{...} nie jest kasowana normalna odległość między kolumnami)</td></tr>
<tr><td>&</td><td>znak umieszczony przed symbolem separatora kolumny (|) powoduje, że w przypadku użycia polecenia \BAmulticolumn (s. 33), nie trzeba w nim powtarzać definicji obramowania rubryki z prawej strony (w przypadku użycia makra \BAmulticolumn w pierwszej kolumnie obramowanie z lewej strony musi być redefiniowane). W przypadku podwójnego obramowania (||) znak & należy umieścić tylko przed pierwszym znakiem |</td></tr>
<tr><td>), \} lub ]</td><td>dodatkowa kolumna (o zerowej szerokości) obejmująca odpowiednim nawiasem całą wysokość tabeli (por. przykłady na rys. 2 i 3)</td></tr>
<tr><td>(, \{ lub [</td><td>dodatkowa kolumna (o zerowej szerokości) obejmująca odpowiednim nawiasem całą wysokość tabeli (por. przykłady na rys. 2 i 3)</td></tr>
<tr><td>\Left{tekst}{delimiter}</td><td>tekst i delimiter stanowi kolumnę o zerowej szerokości; konstrukcja służy do wstawienia dodatkowego opisu (składanego antykwą) umieszczonego przed delimiterem (może nim być jeden z nawiasów ([{}])); delimiter obejmuje całą wysokość tabeli</td></tr>
<tr><td>\Right{delimiter}{tekst}</td><td>tekst i delimiter stanowi kolumnę o zerowej szerokości; konstrukcja służy do wstawienia dodatkowego opisu (składanego antykwą) umieszczonego po delimiterze (może nim być jeden z nawiasów ([{}])); delimiter obejmuje całą wysokość tabeli</td></tr>
<tr><td>\BAenum</td><td>dodatkowa kolumna zawierająca numer kolejny wiersza (szerokość kolumny jest ustalona na 2em)</td></tr>
<tr><td>*n{parametry}</td><td>n-krotne powtórzenie specyfikacji np: *{3}{|c} jest równoważne |c|c|c</td></tr>
</table>

Parametry którymi można regulować pewne wartości konstrukcyjne:

| Polecenie | Opis |
|---|---|
| `\tabcolsep` | połowa szerokości odstępu między kolumnami tabeli (domyślnie `6pt`) |
| `\BAarrayrulewidth` | grubość linii separującej kolumny lub wiersze (o ile są używana – domyślnie `0.4pt`) |
| `\BAdoublerulesep` | odległość między dwoma liniami stanowiącymi separator kolumn lub wierszy (domyślnie `2pt`) |
| `\arraystretch` | „mnożnik” odstępu między wierszami tabeli (np. `\renewcommand{\arraystretch}{1.5}` zwiększa odstęp o 50%) |
| `\BAextrarowheight` | dodatkowy odstęp między wierszami dodawany na górze wiersza (domyślnie `0pt`) |
| `\BAextraheightafterhline` | dodatkowy odstęp po poziomej linii oddzielającej |

**Uwaga:**
W rubrykach tabeli składanych jako akapit (typ: `m`, `p` i `b`) występuje efekt uboczny — blokowane jest przenoszenie wyrazów (*hyphenation*) dla **pierwszego** wyrazu. Efekt możemy wyeliminować wstawiając pustą przestrzeń o długości `0pt` na początku rubryki. Wykorzystując możliwości pakietu możemy to zdefiniować, w preambule tabeli, w następujący sposób:
`>{\hspace{0pt}}p{20mm}`

## 3. Obramowania tabeli

W pakiecie jest zdefiniowane polecenie `\BAhhline` umożliwiające uzyskanie różnych rodzajów linii poziomych. Główną jego przewagą w stosunku do standardowego polecenia `\hline` jest możliwość określenia rodzaju połączenia linii poziomych z pionowymi. Wywołanie polecenia wygląda następująco:

`\BAhhline{`*parametry*`}`

na liście parametrów mogą występować następujące symbole:

| | |
|---|---|
| `=` | podwójna linia o szerokości danej kolumny |
| `"` | podwójna linia przerywana o szerokości danej kolumny |
| `-` | pojedyncza linia o szerokości danej kolumny |
| `.` | pojedyncza linia przerywana o szerokości danej kolumny |
| `~` | brak linii w danej kolumnie (pozwala zastąpić poleceniem `\BAhhline` standardowe polecenie `\cline`) |
| `\|` | linia pionowa przecina (ma priorytet) konstrukcję poziomą; brak tego symbolu widać w przykładzie na rysunku 1, w postaci przerw w liniach poziomych |
| `:` | linia pionowa styka się z podwójną poziomą |
| `#` | skrzyżowanie podwójnych linii |
| `t` | górne wykończenie rogu (lewe lub prawe w zależności od sąsiedztwa) |
| `b` | dolne wykończenie rogu (lewe lub prawe w zależności od sąsiedztwa) |
| `*` | powtórzenie sekwencji; np. `*{3}{==#}` jest równoważne `==#==#==#` |

Możliwości użycia różnych połączeń ilustruje rysunek 1.

## 4. Wstawienie materiału o innej specyfikacji kolumn i delimiterów

Pakiet udostępnia środowiska `block` i `block*`; umożliwiają one wstawienie kolumn, które mogą być inaczej prezentowane niż określa to specyfikacja środowiska `blockarray` (lub innego nadrzędnego), łączenie kolumn sąsiednich oraz zmianę separatorów – jest to „tabela w tabeli”. Postać wywołania jest następująca:

`\begin{block}{`*preambuła*`}`

.

*definicja wierszy*

.

`\end{block}`

```
\begin{blockarray}{||cc||c|c||}
 \BAhhline{|t:==:t:==:t|}
 a & b & c & d\\
 \BAhhline{|:==:|~|~||}
 1 & 2 & 3 & 4\\
 \BAhhline{#==#~|=#}
 i & j & k & l\\
 \BAhhline{||--||--||}
 w & x & y & z\\
 \BAhhline{|:==|:==:|}
 1 & 2 & 3 & 4 \\
 \BAhhline{|:==::=|=:|}
 5 & 6 & 7 & 8 \\
 \BAhhline{|:="::"|=:|}
 9 & 10 & 11 & 12 \\
 \BAhhline{||-|.||.|-||}
 13 & 14 & 15 & 16\\
 \BAhhline{|b:==:b:==:b|}
 \end{blockarray}
```

| | | | |
|---|---|---|---|
| a | b | c | d |
| 1 | 2 | 3 | 4 |
| i | j | k | l |
| w | x | y | z |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

**Rysunek 1**: Przykład ilustruje różne zestawienia połączeń linii poziomych i pionowych

lub

```
\begin{block*}{preambuła}
.
definicja wierszy
.
\end{block*}
```

mogą być one użyte wyłącznie wewnątrz środowiska `blockarray` (można je wielokrotnie zagnieżdżać). *preambuła* (p. s. 27) powinna być zgodna co do liczby kolumn ze specyfikacją kolumn umieszczoną w środowisku nadrzędnym (oprócz dostępnych w niej parametrów, dodatkowo, mogą wystąpić polecenia: `\BAmulticolumn`, definiujące kolumnę obejmującą kilka kolumn środowiska nadrzędnego oraz `\BAmultirow`, które definiuje kolumnę obejmującą wszystkie wiersze zawarte w środowisku `block`). Wszystkie separatory kolumn i kolumny tabeli muszą być redefiniowane. W przypadku użycia konstrukcji nawiasowych w preambule środowiska nadrzędnego wersja „bezgwiazdkowa” umożliwia zdefiniowanie innych konstrukcji nawiasowych (lub rezygnację z nich). W wersji „gwiazdkowej” specyfikacja konstrukcji nawiasowych jest ignorowana (pozostają w mocy specyfikacje środowiska nadrzędnego; różnice najlepiej zilustrują przykłady na rysunkach 2 i 3:

```
\begin{blockarray}{[c|c]c\}}
1111111 & 222222 & 333333 \\
1 & 2 & 3 \\
\begin{block*}{(ll)l\}}
1111 & 2222 & 3333  \\
11 & 22 & 33 \\
\end{block*}
1 & 2 & 3 \\
\end{blockarray}
```

| | | |
|---|---|---|
| 1111111 | 222222 | 333333 |
| 1 | 2 | 3 |
| 1111 | 2222 | 3333 |
| 11 | 22 | 33 |
| 1 | 2 | 3 |

**Rysunek 2**: Środowisko `block` w wersji „gwiazdkowej”; definicje konstrukcji nawiasowych w jego preambule zostały pominięte

```
\begin{blockarray}{[c|c]c\}}
1111111 & 222222 & 333333 \\
1 & 2 & 3 \\
\begin{block}{(ll)l\}}
1111 & 2222 & 3333  \\
11 & 22 & 33 \\ \end{block}
1 & 2 & 3 \\
\end{blockarray}
```

$$\begin{array}{c|cc}
\lceil 1111111 & 222222 \rceil & 333333 \\
\lfloor 1 & 2 \rfloor & 3 \\
\left(\begin{array}{l} 1111 \\ 11 \end{array}\right. & \left.\begin{array}{l} 2222 \\ 22 \end{array}\right) & \begin{array}{l} 3333 \\ 33 \end{array} \\
[\ 1 & 2\ ] & 3
\end{array}$$

**Rysunek 3**: Środowisko `\block` w wersji „bezgwiazdkowej”

Przykład użycia polecenia `\BAmulticolumn` wewnątrz preambuły środowiska `block`

```
\begin{blockarray}{||l|c|c|t{1cm}|c|c||}
\BAhhline{|t:======:t|}
2s & ss &ss &ss &ss&ss\\
\BAhhline{||------||}
\begin{block}{||l|l|\BAmulticolumn{2}{r}|l|l||}
1 & 2 & 3 4 & 5 & 6\\
\end{block}
\BAhhline{||------||}
a & b & c & d & e & f\\
\BAhhline{|b:======:b|}
\end{blockarray}
```

| 2s | ss | ss | ss | ss | ss |
|---|---|---|---|---|---|
| 1 | 2 | 3 4 | | 5 | 6 |
| a | b | c | d | e | f |

Użycie `\BAmultirow` w preambule środowiska `block` wymaga dodatkowego komentarza. Materiał, który ma być składany w bloku o wysokości wszystkich wierszy środowiska `block`, może być wprowadzany w jednym lub kilku wierszach (w tak określonej kolumnie). Pomimo, że jest on podany w kilku wierszach (jedynym tego efektem jest zachowanie podziału na akapity; zapis „od nowej linii” – może to być niewidoczne z powodu braku wcięcia akapitowego), będzie on i tak stanowił całość, która zostanie wypośrodkowana względem wysokości wszystkich wierszy sąsiedniej kolumny.

Proszę zwrócić uwagę, że polecenie `\BAmultirow` działa inaczej niż polecenie `\multirow` zdefiniowane w pakiecie `multirow`).

```
\begin{blockarray}{||l|c|c|t{1cm}|c|c||}
\BAhhline{|t:======:t|}
2s & ss &ss &ss &ss&ss\\
\BAhhline{||------||}
\begin{block}{||l|l|l|l|l|\BAmultirow{1in}||}
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & to jest multirow
                    to jest multirow \\
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & to jest multirow
                    to jest multirow \\
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & \relax\\
\end{block}
\BAhhline{||------||}
a & b & c & d & e & f\\
\BAhhline{|b:======:b|}
\end{blockarray}
```

| 2s | ss | ss | ss | ss | ss |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | to jest multirow<br>to jest multirow<br>to jest multirow<br>to jest multirow |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| a | b | c | d | e | f |

**Rysunek 4**: Użycie `\multirow` w preambule środowiska `block`

**Uwaga!** zawartość kolumn, które nie zawierają materiału do składu, nie może być pusta, muszą one zawierać „coś” np. materiał „pusty” (\relax).

Jeśli wysokość składu materiału przekracza wysokość wszystkich wierszy nie wpływa to na wysokość delimiterów kolumn, pojawia się, niestety, efekt widoczny w poniższym przykładzie.

```
\begin{blockarray}{||l|c|c|t{1cm}|c|c||}
\BAhhline{|t:======:t|}
2s & ss &ss &ss &ss&ss\\
\BAhhline{||------||}
\begin{block}{||l|l|l|l|l|\BAmultirow{1in}||}
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & to jest multirow
    to jest multirow to jest multirow
    to jest multirow to jest multirow \\
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & to jest multirow
    to jest multirow to jest multirow
    to jest multirow to jest multirow \\
1 & 2 & 3 & 4 & 5 & \relax\\
1 & 2 & 3 & 4 & 5 & \relax\\
\end{block}
\BAhhline{||------||}
a & b & c & d & e & f\\
\BAhhline{|b:======:b|}
\end{blockarray}
```

| | | | | | |
|---|---|---|---|---|---|
| 2s | ss | ss | ss | ss | ss |
| 1 | 2 | 3 | 4 | 5 | to jest multirow to jest multirow to jest multirow to jest multirow to jest multirow to jest multirow to jest multirow to jest multirow to jest multirow to jest multirow |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | |
| a | b | c | d | e | f |

## 5. Tworzenie nowych typów kolumn

Za pomocą polecenia \BAnewcolumntype można tworzyć nowe typy kolumn. Składnia polecenia jest następująca:

\BAnewcolumntype{*nazwa*}{*definicja*}

lub

\BAnewcolumntype[*n*]{*nazwa*]}{*definicja*}

gdzie:

*nazwa* jest **jednoliterowym** określeniem nowego typu kolumny

*definicja* określa sposób składu zawartości rubryki w danej kolumnie np:

\BAnewcolumntype{c}{>\hfil<{\hfil}}

definiuje kolumnę, w której zawartość rubryki jest centrowana

*n* jest liczbą parametrów nowej specyfikacji kolumny

Przykład definicji nowego typu kolumny z jednym parametrem:

```
\BAnewcolumntype[1]{t}{>{\rightskip=0pt plus 1fil\vtop\BA@startpbox{#1}}<{\BA@endpbox}}
```

definiuje kolumnę analogiczną do typu p, ale łamaną w „chorągiewkę” – parametr określa szerokość składu. Wersja ulepszona, w której również pierwszy wyraz podlega podziałowi (*hyphenation*) wygląda następująco:

```
\BAnewcolumntype[1]{t}{>{\rightskip=0pt plus 1fil\vtop\BA@startpbox{#1}\hskip0pt}<{\BA@endpbox}}
```

Jak widać, tworzenie nowych typów kolumn jest ograniczone do możliwości przedstawionych w opisie preambuły (p. s. 27). Nowa definicja może być jedynie złożeniem zdefiniowanych wcześniej symboli.

## 6. Łączenie rubryk tabeli w kolumnie (w pionie)

Pakiet `blkarray` udostępnia polecenie `\BAmultirow`, umożliwiające łączenie rubryk tabeli w pionie, podobnie jak polecenie `\multirow`. Składnia wywołania jak i działanie polecenia jest jednak nieco inne niż w przypadku polecenie `\multirow`. Postać wywołania jest następująca:

`\BAmultirow{`*szerokość*`}{`*tekst*`}`

Składany materiał obejmuje zawsze **całą** wysokość, zajmowaną przez wiersze danego środowiska (`block` lub `blockarray`).

Jeżeli wewnątrz środowiska, w którym jest użyte polecenie `\BAmultirow` jest wywołane środowisko `block` to *tekst* będzie powielony (sic!), co ilustruje poniższy przykład. Mankament ten zgłosiłem, jako błąd, do Autora pakietu.

```
\begin{blockarray}{||l|c|c|t{1cm}|c|c||}
\BAhhline{|t:======:t|}
2s & ss &ss &ss &ss&ss\\
\BAhhline{||------||}
1 & 2 & 3 & 4 & 5 & \\
1 & 2 & 3 & 4 & 5 &
  \BAmultirow{4cm}{to jest multirow to
        jest multirow} \\
1 & 2 & 3 & 4 & 5 & \\
\begin{block}{||l|l|l|l|l|l||}
1 & 2 & 3 & 4 & 5 & 6\\
\end{block}
1 & 2 & 3 & 4 & 5 & \\
1 & 2 & 3 & 4 & 5 & \\
1 & 2 & 3 & 4 & 5 & \\
\BAhhline{||------||}
a & b & c & d & e & f\\
\BAhhline{|b:======:b|}
\end{blockarray}
```

| 2s | ss | ss | ss | ss | ss |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | to jest multirow to jest |
| 1 | 2 | 3 | 4 | 5 | multirow |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 2 | 3 | 4 | 5 | |
| 1 | 2 | 3 | 4 | 5 | to jest multirow to jest |
| 1 | 2 | 3 | 4 | 5 | multirow |
| a | b | c | d | e | f |

Jeżeli chcemy, aby materiał obejmował jedynie określoną liczbę wierszy w danej kolumnie tabeli (a nie wszystkie wiersze) to należy użyć konstrukcji `block` łącząc ją z poleceniem `\BAmultirow` (por. użycie konstrukcji `\BAmultirow` w preambule środowiska `block` na stronie 31 i rysunku 4).

## 7. Łączenie rubryk tabeli w wierszu (w poziomie)

Podobnie jak w środowisku `tabular` można uzyskać efekt połączenia kilku rubryk tabeli w jedną, obejmującą kilka kolumn; służy do tego polecenie `\BAmulticolumn` – postać wywołania jest następująca:

`\BAmulticolumn{`*n*`}{`*specyfikacja*`}{`*materiał*`}`

*n* – liczba kolumn, którą ma obejmować konstrukcja,
*specyfikacja* – określenie składu kolumny, analogicznie jak w specyfikacji typów kolumn tabeli (por. s. 27). Jeśli `\BAmulticolumn` obejmuje **pierwszą** kolumnę tabeli, *specyfikacja* musi zawierać definicję obramowania z **lewej strony**. Specyfikację obramowania z prawej strony rubryki można zdefiniować na dwa sposoby:

- poprzedzając znakiem `&` definicję separatora (por. definicja preambuły tabeli, s. 28); efekt ilustruje rysunek 5,
- wstawiając separator w *specyfikacja*, podobnie jak w przypadku pierwszej kolumny np:

  `\multicolumn{2}{p{2cm}|}{`*tekst*`}`

*materiał* określa zawartość rubryki i będzie składany zgodnie ze *specyfikacja*.

```
\begin{blockarray}{||l|c|c&|t{1cm}|c|c||}
\BAhhline{|t:======:t|}
2s & ss &ss &ss &ss&ss\\
\BAhhline{||------||}
& \BAmulticolumn{2}{c}{a  b} & c & d & e\\
\BAhhline{||------||}
a & b & c & d & e & f\\
\BAhhline{|b:======:b|}
\end{blockarray}
```

| | | | | | |
|---|---|---|---|---|---|
| 2s | ss | ss | ss | ss | ss |
| | a b | | c | d | e |
| a | b | c | d | e | f |

**Rysunek 5**: Separator z prawej strony rubryki, obejmującej dwie kolumny, jest wymuszony za pomocą znaku &

## 8. Wstawienie materiału nie podlegającego podziałowi na kolumny

W pakiecie jest zdefiniowane polecenie `\BAnoalign` umożliwiające wstawienie tekstu, który nie podlega podziałowi na kolumny – zajmie on szerokość całej tabeli (może być ono użyte również wewnątrz środowiska `block`). Składnia wywołania jest następująca:

`\BAnoalign{tekst}`

lub

`\BAnoalign*{tekst˝`

Polecenie tworzy „dziurę” w normalnej strukturze tabeli. Wersja „gwiazdkowa” nie powoduje przerwania ciągłości nawiasów separujących (definiowanych konstrukcjami: `(`, `{`, `[ i ]`, `}`, `)`, `\Right` i `\Left`) – niestety z powodu błędu wersja ta jest na ogół bezużyteczna. Ciągłość linii pionowych jest zawsze przerywana.

```
\begin{blockarray}{l|c|c|t{1cm}|c|c\}}
2s & ss &ss &ss &ss&ss\\
2s & ss &ss &ss &ss&ss\\
\BAnoalign{ala ma kota, a kot ma Ale
 ala ma kota, a kot ma Ale
 ala ma kota, a kot ma Ale}
a & b & c & d & e & f\\
a & b & c & d & e & f\\
\end{blockarray}
```

| | | | | | |
|---|---|---|---|---|---|
| 2s | ss | ss | ss | ss | ss |
| 2s | ss | ss | ss | ss | ss |

ala ma kota, a kot ma Ale ala ma kota, a kot ma Ale ala ma kota, a kot ma Ale

| | | | | | |
|---|---|---|---|---|---|
| a | b | c | d | e | f |
| a | b | c | d | e | f |

## 9. Przypisy w tabelach

W tabelach definiowanych za pomocą środowiska `blockarray` można definiować przypisy (za pomocą standardowego polecenia `\footnote`) Standardowo przypisy są umieszczane bezpośrednio pod tabelą i są one numerowane za pomocą polecenia `\fnsymbol`. Użycie polecenia `\BAtablenotesfalse` (zmiana wartości zmiennej logicznej) powoduje, że przypis będzie umieszczony na dole strony (tak jak wszystkie inne przypisy i z użyciem standardowego licznika przypisów). W tym wypadku, jeśli środowisko `blockarray` jest umieszczone wewnątrz środowiska `table` lub `figure` przypisy **nie będą prezentowane!**).

Wywołanie polecenia `\BAparfootnotes` powoduje, że przypisy będą składane jako jeden akapit (tylko w przypadku `\BAtablenotestrue`).

◇ Włodzimierz Macewicz
W.Macewicz@ia.pw.edu.pl