Władca Akapitów

Bogusław Jackowski

TEX posiada niezwykle sprawny algorytm łamania akapitu na wiersze. Niewątpliwą zaletą tego algorytmu – z punktu widzenia profesjonalnych zastosowań – jest jego parametryzacja. Warto wiedzieć jakie parametry i w jaki sposób wpływają na efekt końcowy. Temu zagadnieniu poświęcona jest niniejsza notka.

Podstawową ideą algortytmu jest wyznaczenie potencjalnych miejsc łamania i przypisanie im pewnego kosztu (ang. *demerits*). Po przypisaniu kosztów wszystkim potencjalnym miejscom przełamania TEX wybiera taki ciąg przełamań, by suma kosztów była dlań najmniejsza.

Na ogół o koszcie decyduje stopień ściśnięcia lub rozciągnięcia klejów międzywyrazowych. Odpowiednią wartość dla danego miejsca przełamania TEX wylicza ze wzoru $b = 100f^3$, gdzie f jest czynnikiem określającym ściśnięcie lub rozciągnięcie kleju, a b określa „kiepskość” (ang. *badness*) składu. Na przykład jeśli całkowita rozciągliwość kleju w wierszu wynosi 10pt, a klej został ostatecznie rozciągnięty o 5pt, to $f = 5/10$ i $b = 100 * (5/10)^3 = 12.5$; jeśli klej został rozciągnięty o 10pt, to $b = 100 * (10/10)^3 = 100$. Zauważmy, że kiepskość wynikająca ze ściskania nie może przekroczyć wartości 100, gdyż wielkość ściśnięcia – inaczej niż wielkość rozciągnięcia – nie może przekroczyć wielkości nominalnej.

TEX wyróżnia w tym kontekście cztery klasy wierszy:

klasa 0: jeżeli kiepskość jest większa niż 100, to TEX uznaje dany wiersz za „bardzo rozciągnięty” (*very loose*);

klasa 1: jeżeli kiepskość jest mniejsza niż 100, ale większa od 12, a kleje zostały rozciągnięte, to TEX uznaje dany wiersz za „rozciągnięty” (*loose*);

klasa 2: jeżeli kiepskość nie jest większa od 12, to TEX uznaje go za „przyzwoity” (*decent*);

klasa 3: jeżeli kiepskość jest większa od 12, a kleje zostały ściśnięte, to TEX uznaje dany wiersz za „ściśnięty” (*tight*).

Jaki użytek TEX robi z tej „klasyfikacji wizualnej” – o tym za chwilę.

Nie należy mylić kiepskości z kosztem – kiepskość jest jednym z elementów uwzględnianych przy wyliczaniu kosztu. Podstawowa formuła opisująca sposób obliczania kosztu łamania w danym miejscu jest dość skomplikowana („The TEXbook”, str. 98):

$$d = \begin{cases} (l+b)^2+p^2, & \text{if } 0 \le p < 10000; \\ (l+b)^2-p^2, & \text{if } -10000 < p < 0; \\ (l+b)^2, & \text{if } p \le -10000. \end{cases}$$

W powyższym wzorze b oznacza kiepskość, l – wartość parametru `\linepenalty`, a p – karę (*penalty*) obliczaną jako sumę wartości jednego z parametrów `\exhyphenpenalty` bądź `\hyphenpenalty` oraz ewentualnie kary wstawionej jawnie przez TEX-owca za pomocą polecenia `\penalty`.

Warto zadać sobie trud i przeanalizować powyższą formułę, ale to każdy powinien zrobić „sam na sam ze sobą”. Do dalszych rozważań wystarczy nam spostrzeżenie, że do obliczenia kosztu d wykorzystywane są wartości b, l oraz p podniesione do kwadratu.

Dla ciekawskich parę szczegółów dotyczących parametrów `\exhyphenpenalty` bądź `\hyphenpenalty`: wartość tych parametrów jest wstawiana w miejscu przełamania dyskrecjonalnego (ang. *discretionary break*); jeśli element poprzedzający przełamanie (*pre-break text*) jest niepusty, to uwzględniana jest wartość `\hyphenpenalty`, w przeciwnym razie brana jest pod uwagę wartość `\exhyphenpenalty`.

Ponieważ TEX-owy algorytm przenoszenia wyrazów wstawia przełamania dyskrecjonalne odpowiadające instrukcji `\discretionary{-}{}{}`, więc parametr `\hyphenpenalty` ma wpływ na „powściąganie” TEX-a przed „nadużywaniem” algorytmu dzielenia wyrazów. Natomiast po każdym jawnie użytym znaku dzielenia wyrazów (na ogół jest to znak `-`, ale TEX-owiec może to zmienić, przypisując stosowną wartość parametrowi `\hyphenchar` bądź `\defaulthyphenchar`) lub po ligaturze, której ostatnim elementem jest znak dzielenia wyrazów (np. *endash* lub *emdash*), TEX wstawia niejawnie puste przełamanie dyskrecjonalne `\discretionary{}{}{}`. Oznacza to, że parametr `\exhyphenpenalty` ma wpływ na łamanie wierszy za jawnie użytymi w tekście znakami dzielenia wyrazów.

Ostateczny koszt łamania w danym miejscu wyznaczany jest przez dodanie do wartości wyznaczonej na podstawie powyższej formuły kosztów określonych za pomocą parametrów `\adjdemerits`, `\doublehyphendemerits` oraz `\finalhyphendemerits`:

- wartość parametru `\adjdemerits` jest dodawana wówczas, gdy poprzedni wiersz należy do innej klasy (w sensie podanej wyżej klasyfikacji wizualnej) niż wiersz bieżący;
- wartość parametru `\doublehyphendemerits` jest dodawana wówczas, gdy zarówno poprzedni jak i bieżący wiersz kończy się podzielonym słowem;

– wartość parametru \finalhyphendemerits jest dodawana wówczas, gdy bieżący wiersz kończy się podzielonym słowem, a jest to przedostatni wiersz akapitu.

Koszty dodatkowe określane przez powyższe parametry związane są nie tyle z konkretnym miejscem przełamania, ile z określonym ciągiem przełamań, dlatego nie zostały uwzględnione w sposób jawny w formule. Wartości tych parametrów są dodawane do sumarycznego kosztu „jak leci”, tzn. bez podnoszenia do kwadratu, co oznacza, że ich wartości należy podawać w jednostkach odpowiadających wartościom kar podniesionym do kwadratu, jeżeli parametry te mają mieć zauważalny wpływ na łamanie akapitu.

W ten sposób TEX poszukuje optymalnego przełamania akapitu.

Omówienie co się dzieje, jeśli TEX nie znajdzie akceptowalnego ciągu przełamań, to równie długa historia. Na szczęście w praktyce jest to mniej istotne, dlatego pozwalam sobie pominąć ten temat.

Parę uwag natury ogólnej

Na ogół w dokumentacjach TEX-owych trudno o opis wyczerpujący i precyzyjny. Tak jest i w tym przypadku – pozostała spora lista nie omówionych tematów związanych z parametryzacją algorytmu łamania akapitu.

Uważny czytelnik zapewne podejrzliwie odniósł się do określenia „potencjalne miejsca przełamania” – cóż miałoby to oznaczać? Otóż TEX – dla ułatwienia sobie życia – nie rozpatruje w ogóle niektórych miejsc, tych mianowicie, dla których $p \geqslant 10000$ lub $b > t$, gdzie t oznacza wartość parametru \tolerance. Z kolei miejsca, dla których $p \leqslant -10000$, wyznaczają obowiązkowe miejsca przełamań; koszt dla tych miejsc wyznaczany jest z uproszczonej formuły nie uwzględniającej wartości p jako zbędnej, mianowicie $(l+b)^2$ (p. wyżej).

Jest jeszcze parę parametrów wpływających na sposób przełamania akapitu, z których niezwykle przydatny przy nadawaniu składanemu tekstowi ostatecznego kształtu jest parametr \looseness. Wartością tego parametru jest liczba całkowita. Powoduje on, że TEX stara się wydłużyć lub skrócić akapit o zadaną liczbę wierszy. Dokładniej: niech n oznacza liczbę wierszy, jaką by TEX utworzył składając akapit bez uwzględnienia wartości parametru \looseness i niech k oznacza aktualną wartość tego parametru; TEX próbuje znaleźć taki ciąg przełamań, by liczba wierszy była możliwie bliska $n + k$ i by jednocześnie kiepskość żadnego z wierszy nie przekroczyła wartości parametru \tolerance. (Uwaga: instrukcja \par powoduje nadanie parametrowi \looseness wartości zero.)

Na sposób łamania akapitu ma ponadto wpływ parametr \pretolerance określający najwyższą dopuszczalną kiepskość przełamania akapitu bez dzielenia wyrazów, parametry \lefthyphenmin oraz \righthyphenmin określające długość fragmentów przenoszonych słów, parametry \spaceskip, \xspaceskip oraz \spacefactor wpływające na wielkość odstępów międzywyrazowych, parametr \hfuzz określający dopuszczalną wielkość przekroczenia prawego marginesu, parametr \uchyph pozwalający zablokować dzielenie wyrazów zaczynających się dużą literą, oraz...

No właśnie, co jeszcze można dorzucić do tej kolekcji?

Odpowiedź:

Najpierw małe uzasadnienie, po co komu taka informacja. Ano, czasem się zdarza, że „nic się nie zmieniło, a składa się inaczej”. Oczywiście taka sytuacja jest niemożliwa, po prostu zmienił się któryś z rzadko używanych parametrów. Dlatego potrzebna może się okazać wiedza o tym, jakie parametry mogą coś napsocić. Są to: bieżący font, ustawienia matematyki (\thinmuskip, \medmuskip, \thickmuskip, \binoppenalty, \relpenalty), \emergencystretch, \everypar, \parfillskip, \rightskip, \leftskip, \parindent, \parshape, \hangindent i \hangafter, \language, \hyphenation, \uccode, \lccode, \overfullrule (włączanie tego ostatniego parametru jest może trochę naciągane). Czy coś jeszcze?

Domyślne (plainowe) ustawienia omawianych parametrów

```
\linepenalty=10           \doublehyphendemerits=10000
\hyphenpenalty=50         \finalhyphendemerits=5000
\exhyphenpenalty=50       \adjdemerits=10000
\looseness=0              \lefthyphenmin=2
\tolerance=200            \righthyphenmin=3
\pretolerance=100         \defaulthyphenchar=`\-
\spaceskip=0pt            \uchyph=1
\xspaceskip=0pt           \hfuzz=0.1pt
```

Podziękowania

Dziękuję Włodkowi Macewiczowi za skłonienie mnie do nauczenia się paru ciekawych rzeczy.

◇ Bogusław Jackowski
B.Jackowski@GUST.ORG.PL