☆

Makra

Odsyłacze

Tomasz Przechlewski

Wstęp

Istnieją systemy takie jak LaTeX (autor: Leslie Lamport), $\mathcal{AMS}$-TeX (Michael Spivak) czy eplain (Karl Berry) umożliwiające automatyczne tworzenie odsyłaczy do różnych elementów dokumentu (tabele, rysunki, rozdziały, punkty, itp.). Zawierają one także wiele innych, gotowych do wykorzystania funkcji, których nie ma formacie plain. Zadaniem tego tekstu nie jest „wyważanie drzwi do lasu" ale raczej pokazanie w jaki sposób takie „zaawansowane" funkcje są implementowane. Przy okazji okaże się, że wcale nie jest to takie trudne jakby się mogło na początku wydawać.

Prezentowany zestaw makr, z racji swojej prostoty, może być bardzo łatwo modyfikowany przez użytkownika w zależności od potrzeb. Jest to podstawowa zaleta stosowania prostej TeX-nologii a nie gotowych formatów. Te ostatnie są skomplikowane, a ich przystosowanie do własnych potrzeb jest z reguły bardzo trudne.

Problem

Odsyłacz to znak (liczba, cyfra, asteryks), umieszczony w składzie przy wyrazie, zwrocie lub fragmencie tekstu, odsyłający do objaśnień zawartych w innym miejscu tekstu. Czytelnik może być odesłany do takich elementów tekstu jak: tabela, rysunek, początek rozdziału, początek punktu, równanie matematyczne, pozycja bibliograficzna, inna strona dokumentu itp. Elementy te są z reguły identyfikowalne poprzez kolejny numer, którym są oznaczone (może to być liczba naturalna, para liczb itp.). Jako odsyłacza używamy wtedy *numeru* elementu do, którego chcemy odesłać czytelnika. Przykłady użycia odsyłaczy to: *patrz tabela 6, porównaj punkt 2.5, z równania (4.5) wynika itd.* Jeżeli w tekście nie stosuje się odesłań, zbędne jest numerowanie jakichkolwiek jego elementów (bo po co?).

Wstawianie do dokumentu TeX-owego numerów rozdziałów, punktów, tabel czy równań oraz używanie tych numerów jako odsyłaczy jest złą praktyką. Należy przyjąć zasadę, że na etapie tworzenia pliku źródłowego *ostateczne* numery tych elementów i odsyłacze do nich *są nam nie znane*. Elementy dokumentu winny być numerowane automatycznie przez TeX-a i w taki sam sposób (tzn. automatycznie) wstawiane odsyłacze. Tylko postępując w ten sposób oszczędzimy sobie wiele czasu podczas pracy nad kolejnymi wersjami dokumentu.

Ideę automatycznego wstawiania odsyłaczy przedstawimy na prostym przykładzie systemu służącego do numerowania równań matematycznych. Niech plik `fermat.tex` zawiera następujący kod:

```
Równanie~(\ref{eq:fermat}) na
s.~\pref{eq:fermat} przedstawia słynne
twierdzenie Fermata:
$$\eqalignno{%
x^n + y^n &= z^n&\elab{eq:fermat}}$$
Historia dowodzenia~(\ref{eq:fermat})
ilustruje znaczenie dostatecznie
szerokich marginesów...
```

po kompilacji powinniśmy otrzymać następujący wynik:

Równanie (1) na s. 1 przedstawia słynne twierdzenie Fermata:

$$x^n + y^n = z^n \tag{1}$$

Historia dowodzenia (1) ilustruje znaczenie dostatecznie szerokich marginesów...

Zauważmy, że odnośniki mogą wskazywać „w tył" (do tekstu już przeczytanego) jak i „w przód" (do tekstu nie przeczytanego) konieczne jest zatem dwukrotne kompilowanie dokumentu do ich prawidłowego wyznaczenia (pierwsza kompilacja) i wstawienia.

Użytkownik posługuje się w tym celu trzema następującymi instrukcjami:

`\elab{`*etykieta*`}` wstawia kolejny numer równania oraz definiuje etykietę, której będziemy używać przy odwołaniach do tego równania,

`\pref{`*etykieta*`}` wstawia numer strony na której znajduje się równanie oznakowane *etykietą*.

`\ref{`*etykieta*`}` wstawia numer równania oznakowanego *etykietą*.

Rozwiązanie

Ogólny schemat działania systemu jest następujący: W czasie pierwszej kompilacji instrukcja `\elab` zwiększa wartość licznika równań o 1, wstawia do dokumentu bieżącą wartość tego licznika oraz przesyła do pliku dodatkowego trzy informacje: bieżący numer strony, numer równania, etykietę. Podczas drugiej kompilacji TEX sprawdza czy ten plik dodatkowy istnieje i jeżeli tak to zostaje on wczytany. Zawarte tam informacje są wykorzystywane przez instrukcje `\ref` i `\pref` do prawidłowego wstawienia odsyłaczy.

Przejdźmy teraz do szczegółów.

Zamiast definować od razu komendę `\elab` zdefiniujemy najpierw makro `\defreference`, mające dwa parametry, z których pierwszy będzie etykietą dla odsyłacza, a drugi zawierać będzie sam odsyłacz *oraz* numer strony, na której znajduje się odesłanie. Na przykład jeżeli TEX na 44 stronie dokumentu napotkał definicję `\defreference{eq:fermat}{\the\eqnC}`[1]. to jej wykonanie powinno spowodować wysłanie do pliku `fermat.crf` następującej linii (zakładamy, że w chwili wykonywania `\defreference` licznik `\eqnC` był równy 8):

```
\crlab{eq:fermat}{{8}{44}}
```

Co ma oznaczać, że odsyłaczem dla etykiety `eq:fermat` jest 8, a odesłanie wskazuje na stronę 44. Niżej przedstawiona definicja wykonuje zadanie zapisania odpowiedniej linii do pliku `fermat.crf`.

```
1. \def\defreference#1#2{%
2.   \edef\@tmp{\string\crlab
3.   {#1}{{#2}{\noexpand\folio}}}%
4.   \write\crfile\expandafter{\@tmp}}
```

Makro to musi sobie poradzić z podstawowym problemem: zapisania jednocześnie prawidłowego numeru odnośnika i prawidłowego numeru strony na którą ten odnośnik wskazuje. Numer strony nie jest znany *w momencie napotkania instrukcji* `\defreference`. Jest on ustalany *w momencie wykonywania procedury wyjścia (output routine)*. Z drugiej strony odnośnik jest znany i winien być zapisany *natychmiast*. Jeżeli jego rozwinięcie zostanie opóźnione to otrzymany numer będzie bieżącym numerem odnośnika *w czasie wykonywania tej procedury*.

Problem ten jest rozwiązywany w liniach 2–4 makrodefinicji `\defreference`. Linie 2–3 definiują makro `\@tmp`. Zamiast `\def` użyto `\edef` (*expanded definition*) co gwarantuje, że zawartość definicji `\@tmp` zostanie rozwinięta *natychmiast*. Nie ma to znaczenia gdy piszemy `\defreference{foo}{7}`, ale gdy odnośnik jest ustalany automatycznie, np. `\defreference{foo}{\the\eqnC}`, to chodzi nam o bieżącą wartość licznika a nie jego wartość w chwili wykonywania *output routine*.

Sekwencja `\noexpand\folio` spowoduje, że komenda `\folio` (określająca numer strony), nie zostanie rozwinięta przy rozwijaniu zawartości definicji `\@tmp`. Zostanie to opóźnione do czasu rozwijania komendy `\write` podczas wykonywania *output routine*.

W linii 4 zawartość definicji `\@tmp` zostaje wysłana do pliku dodatkowego za pomocą instrukcji `\write`. Konstrukcja:

```
\write\crfile\expandafter{\@tmp}
```

jest prostym przykładem zastosowania instrukcji `\expandafter` w celu zmiany porządku rozwinięcia dwóch żetonów (*tokens*) `{` oraz `\@tmp`. Kiedy TEX napotka konstrukcję `\write\crfile` oczekuje następnie żetonu `{` (por. *The TEXbook* str. 226), a potem ciągu żetonów kończącego się `}`, który zapisuje do pliku. Zapis do pliku jest *opóźniony*, co oznacza, że cały materiał zawarty pomiędzy klamrami `{` i `}` *nie* jest rozwijany w chwili napotkania instrukcji `\write` *ale* umieszczany jako tzw. *whatsit* na głównej liście pionowej (*main vertical list*) i rozwijany później przy wykonywaniu *output routine* (por. *The TEXbook* str. 227).

Jednakże wykonując sekwencję instrukcji z linii 4 TEX napotyka `\expandafter` zamiast `{`. Powoduje to przeczytanie (czyli rozwinięcie) przez TEX-a najpierw makra `\@tmp` a dopiero potem umieszczenie przed rozwiniętym już makrem `\@tmp` żetonu `{`. W efekcie na główną listę pionową, do późniejszego zapisu do pliku `fermat.crf` wędruje sekwencja żetonów tworząca makro `\@tmp` a nie żeton `\@tmp`, który jest od tej chwili gotowy do użycia w następnej instrukcji `\defreference`. Gdyby na główną listę pionową trafiał żeton `\@tmp` rozwijany podczas wykonywania *output routine* to zawartość (*meaning*) wszystkich żetonów byłaby jednakowa i równa zawartości ostatniego zdefiniowanego żetonu `\@tmp` — rezultat całkowicie różny od poprzedniego i raczej przez nas nie oczekiwany!

Makro `\elab` można zdefiniować następująco:

```
5. \newcount\eqnC
6. \def\elab#1{\global\advance\eqnC 1
7. \defreference{#1}{\the\eqnC}%
8. (\the\eqnC)}
```

1: Występujące w poniższym opisie nazwy i konstrukcje prędzej lub później zostaną wyjaśnione.

Po pierwszej kompilacji plik `fermat.crf` zawiera informacje o wszystkich odsyłaczach, które wykorzystujemy przy powtórnej kompilacji dokumentu. W tym celu najpierw zdefiniujemy komendę `\crlab`. Jak widać wyżej, posiada ona dwa parametry, z których pierwszy zawiera etykietę odsyłacza a drugi łącznie odsyłacz oraz numer strony, na której odesłanie się znajduje. Zarówno odsyłacz, jak i numer strony zawarte są w parze nawiasów klamrowych.

```
9. \def\crlab#1#2{%
10.  \global\expandafter
11.  \def\csname #1\endcsname{#2}}
```

Wykonanie makra `\crlab{eq:fermat}{{8}{44}}` spowoduje utworzenie nowego makra o nazwie `eq:fermat` rozwijającego się dokładnie do `{8}{44}`. Wykorzystanie konstrukcji `\csname...\endcsname` umożliwia definiowanie etykiet zawierających znaki o dowolnych „egzotycznych” kodach, np. `&`, `:`, `#`, itd. Wręcz wskazane jest umieszczenie takich znaków, co zapobiegnie niezamierzonej zmianie znaczenia „normalnych” makr o przypadkowo identycznej nazwie.

Teraz możemy zdefiniować instrukcję `\ref`. Makro to powinno wstawiać odsyłacz a pomijać numer strony. Kopiujemy w tym celu pomysłowe rozwiązanie tego problemu z formatu LaTeX, w którym znowu w roli głównej występuje instrukcja `\expandafter`:

```
12. \def\@car#1#2{#1}
13. \def\ref#1{%
14.  \edef\@tempa{\csname #1\endcsname}
15.  \expandafter\@car\@tempa}
```

Makrodefinicja `\ref` ma jeden argument — etykietę odnośnika. W linii 14 tworzona jest instrukcja `\@tempa`, której zawartością jest wykonanie makrodefinicji o nazwie tożsamej z nazwą etykiety. W następnej linii najpierw rozwijana jest instrukcja `\@tempa`, co oznacza rozwinięcie jej zawartości do postaci {*odnośnik*}{*strona*}. Następnie rozwijane jest makro `\@car`, które z dwóch swoich parametrów wstawia pierwszy a pomija drugi. Proste!

Skonstruowane w analogiczny sposób makro `\pref` wstawia numer strony a pomija odnośnik:

```
16. \def\@cdr#1#2{#2}
17. \def\pref#1{%
18.  \edef\@tempa{\csname #1\endcsname}
19.  \expandafter\@cdr\@tempa}
```

Teraz określmy wreszcie plik, z którego *pobierane* będą odnośniki a następnie otwórzmy go do czytania:

```
20. \newread\crfile
21. \openin\crfile=\jobname.crf
22. \input \jobname.crf
```

Powyższy kod ma jeden poważny minus. Mianowicie gdyby z jakichś względów plik `fermat.crf` nie istniał (w pierwszej kompilacji dokumentu na pewno go nie będzie) to wtedy próba wykonania linii `\input \jobname.crf` spowoduje błąd `I can't find file fermat.crf`. Lepiej zabezpieczyć się na tę okoliczność używając komendy `\ifeof`. Tak więc w powyższym fragmencie kodu ostatnią linię należy zastąpić przez:

```
22. \ifeof\crfile \else
23.   \input \jobname.crf \fi
```

Wreszcie pozostaje do zdefiniowania plik do którego będą *wysyłane* informacje o odesłaniach:

```
24. \newwrite\crfile
25. \openout\crfile=\jobname.crf
```

I te 25 linii kodu pokazane wyżej wystarczą dla TeX-a do prawidłowego wstawienia odpowiednich odsyłaczy. Wystarczą TeX-owi ale nie TeX-owcowi, który z pewnością popełniać będzie błędy. Dlatego powyższe makra należy rozbudować o obsługę błędów i ostrzeżeń. W szczególności należy zadbać o ostrzeganie użytkownika o:

- odwołaniu się do niezdefiniowanej etykiety. W tym przypadku system powinien wysłać komunikat do pliku `.log` i na ekran a także oznakować brakujące odnośniki w składanym dokumencie,
- dwukrotnym (lub wielokrotnym) zdefiniowanie tej samej etykiety. W tym przypadku system powinien wysłać odpowiedni komunikat do pliku `.log` i na ekran.

Ponieważ w przedstawionych wyżej makrach używamy znaku `@` w nazwach komend, powinny zostać one zawarte pomiędzy liniami:

```
\catcode`@=11
...
\catcode`@=12
```

Makra

Prezentowany poniżej zestaw makr jest dostępny na komputerze `sunrise.pg.gda.pl` w katalogu

TeX/GUST/bulletin/05 lub halina.univ.gda.pl w katalogu gust/bulletin/05 — plik tp-crf.tex. W porównaniu do przedstawionych już makrodefinicji dodano następujące ważniejsze komendy:

\nocrwarns Ostrzeżenia o błędach nie są wyświetlane na ekranie (przydatne na wstępnym etapie pracy nad dokumentem),

\nocrfile Dodatkowy plik nie jest odtwarzany,

\makecrfile Dodatkowy plik jest tworzony,

\crstatistics Wyświetlenie sumarycznej informacji o użytych odsyłaczach. Przedefiniowana komenda \bye wywołuje to makro.

```
%% --------------------------------
%% Cross-reference generic macros
%% Tomasz Przechlewski
%% Date: 02.01.1995
%% --------------------------------
\catcode`@=11
\def\@crwrn#1{\if@crwrns\immediate
\write16{#1}\fi}
\def\@markmissingcr{{\bf ??}\@marginmarker}
\def\@marginmarker{\vadjust{\vbox to0pt{%
\kern-.77\normalbaselineskip
\hbox{{\it\kern\hsize\kern15pt?}}\vss}}}

\newif\if@crwrns
\global\@crwrnstrue % default
\def\nocrfile{\global\@crfilefalse}
\def\nocrwrns{\global\@crwrnsfalse}

\def\@car#1#2{#1}
\def\@cdr#1#2{#2}

\long\def\@ifundefined#1#2#3{%
 \expandafter\ifx\csname
 #1\endcsname\relax#2\else#3\fi}

\def\namedef#1{\expandafter
  \def\csname #1\endcsname}

\def\newlabel#1#2{\@ifundefined{#1}{}%
{\@crwrn{-> WARNING: multiple label #1}}%
\global\namedef{#1}{#2}}
\newread\crfile

\openin\crfile=\jobname.crf
\ifeof\crfile
  \@crwrn{-> WARNING: CR-FILE UNDEFINED!!}
 \else
  \@crwrn{READING REFS FROM \jobname.crf}
  \input \jobname.crf
\fi
\closein\crfile

\def\makecrfile{%
  \openout\crfile=\jobname.crf}
\def\nocrfile{\@crwrn{-> WARNING:
        CR-FILE not created}
 \def\crfile{-1}}

\def\ref#1{\@nextcrf\@ifundefined{#1}{%
 \@markmissingcr
 \@crwrn{undefined cr -> \string#1}}%
 {\edef\@tempa{\csname #1\endcsname}
 \expandafter \@car\@tempa}}

\def\pageref#1{\@nextpcrf
 \@ifundefined{#1}{\@markmissingcr
 \@crwrn{undefined cr -> \string#1}}%
 {\edef\@tempa{\csname #1\endcsname}%
 \expandafter \@cdr\@tempa}}

\def\defreference#1#2{\@nextdrf%
 \edef\save{\string\newlabel{#1}%
 {{#2}{\noexpand\folio}}}%
 \write\crfile\expandafter{\save}}

\newcount\@crfC\newcount\@pcrfC
\newcount\@dcrfC
\def\@nextdrf{\global\advance\@dcrfC1}
\def\@nextcrf{\global\advance\@crfC1}
\def\@nextpcrf{\global\advance\@pcrfC1}
\def\crstatistics{%
\@crwrn{==============================}
\@crwrn{= REFERENCE STATISTICS =======}
\@crwrn{= refs defined.... \the\@dcrfC}
\@crwrn{= refs used....... \the\@crfC}
\@crwrn{= page refs used.. \the\@pcrfC}
\@crwrn{==============================}}
\outer\def\bye{\crstatistics\end}
\catcode`@=12
\endinput
```

Bibliografia

[1] Knuth D. E., *The TeXbook*, Addison-Wesley, Reading MA: 1986.

[2] Salomon D., Macros for Indexing and Table-of Contents Preparation, *TUGboat*, 10(3): s. 394–400.