Popis tvaru litery

Jak již bylo uvedeno, je postscriptové písmo vždy realizováno jako slovník, tedy jako seznam pojmenovaných datových struktur. Popis jedné litery (například hvězdičky), může ve slovníku vypadat přibližně takto:

```
asterisk –|  ... kód litery ... |–
```

Jméno `asterisk` je název litery (tedy název příslušné datové struktury) , znaky –| a |– potom ohraničují kód litery, tedy dotyčnou datovou strukturu (ve skutečnosti se nejedná o nějaké syntaktické omezovače, ale o postscriptové operátory, definované v záhlaví písma; často se může jednat i o nějaké jiné znaky podle programu, použitého pro vygenerování písma; tím se zde ale nebudeme podrobněji zabývat).

Kód litery je tvořen jazykem, který je velmi malou podmnožinou jazyka PostScript samotného (ovšem s drobnými rozšířeními). Podobně jako v jazyku PostScript je text jazyka písem typu 1 zapsán polskou notací, tedy nejprve operandy, poté operátor. Jednotlivé operandy jsou přitom kladná nebo záporná celá čísla; operátory jsou výkonné příkazy, které způsobí např. vykreslení úsečky, ale nevracejí žádnou hodnotu (existují ovšem výjimky). Jinými slovy, operandy nelze například sčítat (k tomu by byl potřeba aritmetický operátor, vracející hodnotu), nelze volat běžné matematické funkce (např. sin, log a pod.), ani nelze program větvit nebo snad provádět cykly.

Kód litery je pro úsporu místa kódován tak, že každý operand i operátor jsou uloženy jako jeden nebo dva bajty; operátor `return` je například zapsán jedním bajtem s hodnotou 11. Podobně operandy (což jsou vždy celá čísla) z intervalu –107 až 107 jsou kódovány jedním bajtem. Pouze čísla mimo tento interval, která se vyskytují zřídka, a také málo používané operátory, jsou kódovány dvěma bajty (jednoduše proto, že všechno by se do jednoho bajtu nevešlo). Jinými slovy, kód litery typu 1 **není** zapsán jako text, jak je pravidlem pro jazyk PostScript samotný; proto také, podíváte-li se do souboru s písmem nějakým editorem, neuvidíte žádný operátor typu 1 v textovém tvaru.

Celý slovník, který písmo tvoří, je navíc z komerčních důvodů zašifrován; proto v souboru s písmem nenajdete např. ani výše uvedené jméno `asterisk`.

Kód litery

Kód litery v písmu typu 1 má vždy následující tvar:

```
–| x w hsbw ...cesty... endchar |–
```

Operátor x w hsbw (Horizontal Sidebearing and Width) musí být vždy první a určuje levý okraj litery (x) a šířku kuželky (w); vizte obrázek 1. Operátor `endchar` potom kód litery ukončuje.

Mezi výše uvedenými operátory jsou popsány jednotlivé cesty, které tvoří obrysy litery; tedy například vnější a vnitřní obrys litery A nebo jediný obrys litery f, a podobně. Každá cesta má přitom tvar:

```
x y rmoveto ... kreslicí příkazy ... closepath
```

Operátor `rmoveto` určuje počáteční bod cesty **relativně** ke koncovému bodu předchozí cesty; je tedy ekvivalentem příkazu `x y rmoveto` jazyka Postscript, a znamená relativní posun o vektor [x, y] vůči předchozímu bodu. U první cesty je přitom předchozí bod dán operátorem hsbw. Povšimněte si prosím, že se jedná o **relativní** posun; tak je tomu také ve všech kreslicích příkazech. Příkaz `closepath` potom cestu uzavírá. Poznamenejme, že pro zkrácení kódu má příkaz rmoveto také variantu pro vodorovný posun - x hmoveto (odpovídá zápisu x 0 rmoveto) i pro svislý posun - y vmoveto (odpovídá zápisu 0 y rmoveto).

Kreslicí příkazy

Kreslící příkazy, zmíněné ve výše uvedeném schématu, slouží k popisu jednotlivých cest;

základní dva jsou rlineto a rrcurveto.

Příkaz x y rlineto je obdobou stejnojmenného příkazu jazyka PostScript a má za následek vykreslení úsečky z koncového bodu předchozího příkazu ve směru a velikosti vektoru [x, y]. Podobně jako příkaz rmoveto má i příkaz rlineto pro úsporu kódu dvě varianty: x hlineto pro vodorovnou úsečku a y vlineto pro úsečku svislou.

Příkaz x1 y1 x2 y2 x3 y3 rrcurveto vykreslí Beziérovu křivku z koncového bodu předchozího příkazu do bodu, který je posunut o [x1+x2+x3, y1+y2+y3]; křivka je dána dvěma řídícími body, které jsou vůči koncovému bodu předchozího příkazu posunuty o [x1, y1], resp. [x1+x2, y1+y2]. Vizte obrázek 2. Tento příkaz je obdobou příkazu rcurveto jazyka PostScript s tím rozdílem, že jeho souřadnice jsou relativní vůči sobě navzájem, zatímco u příkazu rcurveto jazyka PostScript jsou relativní ke koncovému bodu posledního kreslicího příkazu; jinými slovy příkaz x1 y1 x2 y2 x3 y3 rrcurveto má stejný efekt jako příkaz x1 y1 {x1+x2} {y1+y2} {x1+x2+x3} {y1+y2+y3} rcurveto, pokud za části ve složených závorkách dosadíte příslušné hodnoty.

Podobně jako v případě příkazu rmoveto má i příkaz rrcurveto dvě varianty: hvcurveto pro případ nulového y1 a x3, tedy oblouk s první tangentou vodorovnou a druhou svislou, a vhcurveto pro případ opačný, tedy první tangentu svislou (nulové x1) a druhou vodorovnou (nulové y3).

Výše uvedené příkazy v základě dostačují pro popis tvaru litery; veškeré další operátory již jazyk písem typu 1 nějakým způsobem vylepšují či zdokonalují.

Podprogramy

Podprogramy jsou jednoduchým a poměrně silným nástrojem, jak kód písma zmenšit a tím uspořit místo na disku, paměť v tiskárně i čas při zavádění písma do tiskárny. Hlavní kouzlo podprogramů vyplývá z toho, že všechny kreslicí příkazy mají souřadnice relativní k předchozímu bodu. Jelikož například pravý tah litery **n** i **m** má většinou stejný tvar, je vyjádřen také stejným kódem, a to přesto, že je absolutní y-ová souřadnice jednotlivých bodů křivky rozdílná; dokonce i oblouk litery **a** může mít v některých abecedách stejný tvar jako oblouk litery **g** nebo **p** a tím i stejný kód. Takovýto kód je při generování finálního písma z kódu litery vyjmut a umístěn do společného podprogramu.

Každý podprogram je datová struktura, podobná té, která popisuje literu; všechny podprogramy dohromady potom tvoří pole (tedy nikoli slovník, jako v případě jednotlivých liter), které se standardně nazývá Subrs a je umístěno za záhlavím písma ještě před samotným kódem liter. Jinými slovy, každý podprogram je jedním prvkem pole Subrs. Symbolicky to můžeme vyjádřit zápisem

```
dup n -| ... příkazy... return |
```

Symboly dup, -| a | jsou z našeho hlediska syntaktické omezovače (ač se ve skutečnosti jedná opět o operátory jazyka PostScript, definované v záhlaví písma), n je potom index pole Subrs (tedy pořadové číslo podprogramu) a return je operátor jazyka písma typu 1, který předává řízení volajícímu. V místě, odkud se příslušný podprogram volá, je potom použit operátor n callsubr, který způsobí vyvolání podprogramu číslo n. Je třeba poznamenat, že každý podprogram musí končit operátorem return a že operátor callsubr se může vyskytovat jak přímo v popisu litery, tak i v podprogramu, a to až do deseti úrovní vnoření (počet úrovní vnoření lze zjistit již při konstrukci písma, neboť jazyk typu 1 neobsahuje konstrukce pro větvení; z toho také vyplývá, že není možné rekurentní volání).