Sila a zložitosť moderných regulárnych výrazov

Tatiana Tóthová
školiteľ: RNDr. Michal Forišek, PhD.

Regulárne výrazy vznikli v 60tych rokoch ako ďalší model reprezentujúci regulárne jazyky. Vďaka jednoduchosti a kompaktnosti boli implementované ako vyhľadávací nástroj. V praxi boli postupne rozširované o ďalšie konštrukcie. My sme tento model skúmali z hľadiska teórie formálnych jazykov. Niektoré nové konštrukcie sa dajú prepísať pomocou základných operácií, iné model rozšírili.

Moderné regulárne výrazy (regexy) obsahujú základné operácie regulránych výrazov – zreťazenie, alternácia, Kleeneho $*$. Navyše majú metaznak pre ľubovoľný znak . , začiatok slova $\widehat{}$ a koniec slova \$. Zátvorky regexov sú číslované, zľava doprava podľa otváracej zátvorky.

Zložitejšie konštrukcie rozširujúce model regulárnych výrazov sú:

- **Spätné referencie** ($\backslash k$) sa odkazujú na k-te zátvorky. Pre regex $\alpha \underset{k}{(}\ \beta\ \underset{k}{)}\ \gamma\ \backslash k\ \delta$ výpočet na slove w vyzerá takto:

$$w = \underbrace{x_1 \dots x_{i-1}}_{\alpha} \underbrace{\overbrace{x_i \dots x_{j-1}}^{w_k}}_{\underset{k}{(}\,\beta\,\underset{k}{)}} \underbrace{x_j \dots x_{l-1}}_{\gamma} \underbrace{\overbrace{x_l \dots x_{m-1}}^{w_k}}_{\backslash k} \underbrace{x_m \dots x_n}_{\delta}$$

  pričom platiť: $w_k = x_i \dots x_{j-1} = x_l \dots x_{m-1}$.

- **Lookahead** (nazeranie dopredu (?= . . .)) musí matchovať nejaký prefix od aktuálneho pracovného miesta. Pre regex $\alpha(?{=}\beta)\gamma$:

$$w = \underbrace{x_1 \dots x_{i-1}}_{\alpha} \underbrace{\overbrace{x_i \dots x_j}^{\beta}\, x_{j+1} \dots x_n}_{\gamma}$$

- **Lookbehind** (nazeranie dozadu (?<= . . .)) musí matchovať nejaký sufix od aktuálneho pracovného miesta.

- Operácie lookahead a lookbehind majú aj **negatívne** verzie (pridáva sa !), kde regex vnútri konštrukcie nesmie matchovať žiaden prefix/sufix.

Vlastnosti lookaheadu a lookbehindu

Už len uzavretosť na zreťazenie nie je triviálna, pretože treba zabrániť presahu lookaheadu a lookbehindu vo výpočte na slove z $L(\alpha)L(\beta)$:

$$(?{=}\underset{1}{(}\,\alpha\,\underset{1}{)}\ \underset{k+2}{(}\,\beta\,\underset{k+2}{)}\ \$)\ \alpha'\backslash k{+}2\ (?{<=}\ \widehat{}\backslash 1\ \beta')$$

Zaviedli sme nasledujúce triedy regexov nad množinami operácií:
Regex – základné operácie. *Eregex* – má navyše spätné referencie ($\mathscr{L}_{ERE}$). *LEregex* = *Eregex* + lookahead, lookbehind ($\mathscr{L}_{LERE}$). *nLEregex* = *LEregex* + neg. lookahead, neg. lookbehind ($\mathscr{L}_{nLERE}$).

Ukázali sme, že trieda *Regex* s pozitívnym a negatívnym lookaheadom a lookbehindom je ekvivalentná triede $\mathcal{R}$. Pre zadefinované triedy zároveň platí nasledovná hierarchia:

$$\mathcal{R} \subsetneq \mathscr{L}_{ERE} \subsetneq \mathscr{L}_{LERE} \subseteq \mathscr{L}_{nLERE} \subsetneq \mathscr{L}_{CS}$$

Vidieť, že pokiaľ sú v modeli aj spätné referencie, lookahead a lookbehind ho rozširujú.

Trieda *nLEregex* nie je porovnateľná s $\mathscr{L}_{CF}$.

Priestorová zložitosť

Ukázali sme, že $\mathscr{L}_{LERE} \subseteq NSPACE(\log n)$.

Zo Savitchovej vety vyplýva: $\mathscr{L}_{LERE} \subseteq DSPACE(\log^2 n)$ a výsledok sme rozšírili: $\mathscr{L}_{nLERE} \subseteq DSPACE(\log^2 n)$

Pre problém, kde dostávame na vstup aj regex aj vstupné slovo sme ukázali nasledovné výsledky:

$$L(regex\#word | regex \in LEregex) \in NSPACE(n \log n)$$

Nech regex r je taký, že dĺžka jeho akceptačného výpočtu na slove w sa dá zhora ohraničiť hodnotou $f(|w|)$, potom platí:
$L(r\#w) \in DSPACE(w \cdot \log n \cdot \log(f(n)))$, pričom n je dĺžka vstupu