Moderné regulárne výrazy

Tatiana Tóthová
školiteľ: RNDr. Michal Forišek PhD.

Fakulta matematiky, fyziky a informatiky
Univerzita Komenského

28.mája 2015

Motivácia:

- teoretický model regulárnych výrazov bol v praxi rozšírený o nové konštrukcie

Cieľ :

- preskúmať z hľadiska formálnych jazykov – Chomského hierarchia, vlastnosti triedy jazykov, zložitosť

Definícia

Regex tvoria **znaky** a **metaznaky**

Základné operácie:

- $(e_1)(e_2)$ zreťazenie (bez znaku)
- $(e_1)|(e_2)$ alternácia
- $(e_1)*$ Kleeneho uzáver

Navyše: ľubovoľný znak . , začiatok slova ^, koniec slova $

Ďalšie operácie:

- množiny znakov $[abc], [a-z]$
- komplementy množín znakov $[\hat{}abc], [\hat{}a-z]$
- opakovania $?, \{n\}, \{n,m\}, +$

Regex má konečnú dĺžku!

Zložitejšie konštrukcie (1)

Číslovanie zátvoriek zľava doprava, podľa otváracej zátvorky.
Nečíslujú sa konštrukcie (?...)

Spätné referencie $\backslash k$ – odkazujú sa na k-te zátvorky

$\alpha \underset{k}{(} \beta \underset{k}{)} \gamma \backslash k \ \delta$

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

$$w_k = x_i \dots x_{j-1} = x_l \dots x_{m-1}$$

Zložitejšie konštrukcie (2)

Lookahead (?= . . .) – nazeranie dopredu: $\alpha(?=\beta)\gamma$

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

Lookbehind (?<= . . .) – nazeranie dozadu: $\alpha(?<=\beta)\gamma$

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

Ich **negatívne formy** (?! . . .), (?<! . . .) $\rightarrow$ opačná akceptácia (slová, ktoré sa v danom jazyku nenachádzajú)

Príklady

- $(?{=}(a^m) * \$)(a^{m+1})*$

$$L_1 = \{a^{im(m+1)} \mid i \in \mathbb{N}_0\}$$

iný možný zápis $\rightarrow (a^{m(m+1)})*$

- $(a*)\#(?{=}(\backslash 1) * \$)(a\backslash 1)*$

$$L = \{a^m \# a^{im(m+1)} \mid i \in \mathbb{N}_0\}$$

- $((?{=}(a^{m+1}) * a\{1, m-1\}\$ \mid a^m\$)\ a^m) +$

$$L_2 = \{a^m, a^{2m}, \ldots, a^{m^2}\}$$

Triedy

Regex	základné operácie a znaky . $\widehat{\ }$ \$ ($= \mathcal{R}$)
Eregex	+ spätné referencie
LEregex	+ lookahead, lookbehind
nLEregex	+ negatívny lookahead, negatívny lookbehind

Triedy jazykov: $\mathcal{R}$, $\mathscr{L}_{ERE}$, $\mathscr{L}_{LERE}$, $\mathscr{L}_{nLERE}$

Hierarchia

Veta

Trieda nad $Regex$ s pozitívnym a negatívnym lookaroundom je $\mathcal{R}$.

Hierarchia

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

$\mathscr{L}_{LERE}$ a $\mathscr{L}_{nLERE}$ sú neporovnateľné s $\mathscr{L}_{CF}$

Pumpovacia lema

Veta

Jazyk všetkých platných výpočtov Turingovho stroja patrí do $\mathscr{L}_{LERE}$.

Veta

Nech $\alpha \in LEregex$ nad unárnou abecedou $\Sigma = \{a\}$ taký, že neobsahuje lookahead s $ ani lookbehind s ^ vnútri iterácie $(*)$. Existuje konštanta N taká, že ak $w \in L(\alpha)$ a $|w| > N$, potom existuje dekompozícia $w = xy$ s nasledujúcimi vlastnosťami:

(i) $|y| \geq 1$

(ii) $\exists k \in \mathbb{N},\ k \neq 0;\ \forall j = 1, 2, \ldots : xy^{kj} \in L(\alpha)$

Uzavretosť na zreťazenie

Veta

$\mathscr{L}_{LERE}$ je uzavretá na zreťazenie.

Dôkaz. Nech $\alpha, \beta \in LEregex$. Regex pre $L(\alpha)L(\beta)$ je

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

Nový formalizmus

- konfigurácia: $(r_1 \dots \lceil r_i \dots r_n, w_1 \dots \lceil w_j \dots w_m)$
- indexovateľné zátvorky
- krok výpočtu

$$(r_1 \dots \underset{k}{(} \dots \underset{k}{)} \lceil * \dots r_n, w_1 \dots \overset{k}{w_a} \dots \overset{k'}{w_b} \dots \lceil w_j \dots w_m)$$

$$(1) \vdash (r_1 \dots \underset{k}{(} \dots \underset{k}{)} * \lceil \dots r_n, w_1 \dots \overset{k}{w_a} \dots \overset{k'}{w_b} \dots \lceil w_j \dots w_m)$$

$$(2) \vdash (r_1 \dots \underset{k}{(} \lceil \dots \underset{k}{)} * \dots r_n, w_1 \dots w_a \dots w_b \dots \lceil \overset{k}{w_j} \dots w_m)$$

- akceptačný výpočet $(\lceil r, \lceil w) \vdash^* (r \lceil, w \lceil)$
- jazyk

Dĺžka akceptačného výpočtu

Lema

Nech $\alpha \in Eregex$ a $w \in L(\alpha)$. Potom existuje akceptačný výpočet, ktorý má najviac $5 \cdot |\alpha| \cdot |w|$ konfigurácií.

Počet všetkých možných konfigurácií $\alpha \in LEregex$ je

$$O(|\alpha| \cdot |w|^{|\alpha|+2})$$

Kvôli vlastnostiam lookaroundu lepší odhad nemáme.

Priestorová zložitosť

Veta.

$\mathscr{L}_{LERE} \subseteq NSPACE(\log n)$

Dôkaz. Na páske si pamätáme informáciu z poschodových symbolov – vo forme adries

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

Veta

$\mathscr{L}_{nLERE} \subseteq DSPACE(\log^2 n)$

Dôkaz. Idea Savitchovej vety, ale s konfiguráciami z formalizmu. Pre každý negatívny lookaround spúšťame nový Turingov stroj s opačnou akceptáciou.

Regex aj jazyk na vstupe

$r = |regex|,\ w = |word|$

Veta

$L(regex\#word) \in NSPACE(r \log w)$, regex $\in LEregex$

Dôkaz. Vyplýva z $\mathscr{L}_{LERE} \subseteq NSPACE(\log n)$, dĺžka regexu už viac nie je konštanta.

Veta

Nech počet konfigurácií v akceptačnom výpočte je polynomiálny od r a w, potom $L(regex\#word) \in DSPACE(r \log^2 w)$, regex $\in LEregex$

Dôkaz. Idea Savitchovej vety, ale s konfiguráciami z formalizmu.

Na záver

Ďakujem za pozornosť!

Problém s lemou 2

Lema 2

Nech $\alpha \in \mathscr{L}_{LERE},\ s \in L(\alpha),\ r = |\alpha|$ a $w = |s|$. Potom existuje akceptačný výpočet, ktorý má nanajvýš $O(r^2 w^3)$ konfigurácií.

Pripomeňme lemu 1

Lema 1

Nech $\alpha \in \mathscr{L}_{ERE}$ a $w \in L(\alpha)$. Potom existuje akceptačný výpočet, ktorý má najviac $5 \cdot |\alpha| \cdot |w|$ konfigurácií.

Doplnenie na $\mathscr{L}_{LERE}$? Funguje len pre regexy s konštantnou hĺbkou vnorenia lookaroundov.

Počet všetkých možných konfigurácií je $O(|\alpha| \cdot |w|^{|\alpha|+2})$

Komplikovanosť lookaheadu s $ a lookbehindu s ^ vnútri $*$

Napríklad regex

$$((?=\underbrace{\underbrace{(?=(a^m)*\$)}_{a^{km},\ k\in\mathbb{N}}\ \underbrace{(a^{m+1})*a\{1,m-1\}\$ \mid a^m\$}_{\text{vie } a* \text{ okrem } a^{m+1},\ a^{m(m+1)l},\ l\in\mathbb{N}}}_{a^{km} \text{ také, že nevie } a^{m(m+1)l},\ l\in\mathbb{N}})a^m)+$$

generuje konečný jazyk obsahujúci slová

- a^m
- a^{2m}
- ...
- $a^{(m-1)(m+1)}$

Hlavný lookahead je spúšťaný každú iteráciu, teda pre slovo a^{zm} musí matchovať všetky a^{im} pre $i \in \{1, \ldots, z\}$.