

FINITE TRANSDUCER WITH MULTIPLE READ HEADS

David PLUHÁČEK, Bachelor Degree Programme (3)
Dept. of Information Systems, FIT, BUT
E-mail: xpluha02@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This paper deals with some new operations in formal languages theory and their implementation. First, parallel operations are briefly introduced. Following, methods of practical implementation of these operations are discussed. The paper provides definition of a modified finite transducer, a tool can be used to implement parallel operations. It also provides some other related definitions.

1 ÚVOD

V teorii formálních jazyků jsou momentálně diskutovány některé nové jazykové operace. Mezi těmito operacemi se jako zajímavé jeví operace paralelní, kterými se zabývám. Jejich vlastnosti jsou již teoreticky dobře popsány, proto se zaměřuji spíše na zkoumání možností jejich praktické realizace.

2 PARALELNÍ OPERACE

Paralelní operací rozumíme obecně takovou operaci, která probíhá na více místech současně. V teorii formálních jazyků pracujeme nad řetězci, tj. operace probíhá současně na více místech řetězce.

Konkrétními příklady paralelních operací jsou paralelní vkládání a mazání.

2.1 PARALELNÍ VKLÁDÁNÍ

Jde o operaci nad dvěma řetězci, při které je druhý řetězec vložen mezi všechny symboly a před první a za poslední symbol prvního řetězce. Formální definici je možno nalézt v [2]. Operaci značíme operátorem \leq .

Příklad: $ab \leq c = cacbc$

2.2 PARALELNÍ MAZÁNÍ

Opět jde o operaci nad dvěma řetězci (obecně nad jazyky). Paralelní vymazání řetězce v z řetězce u je vymazání všech nepřekrývajících se podřetězců u , které jsou rovny řetězci v . Pokud se podřetězce překrývají, je výsledkem operace více řetězců – všechny možné postupy mazání. Formální definici a podrobnější informace je opět možné nalézt v [2]. Operaci značíme operátorem \Rightarrow .

Příklad: $ababbaba \Rightarrow \{aba\} = \{bb\}$

3 REALIZACE OPERACÍ

Z povahy operací je zřejmé, že k jejich realizaci budeme potřebovat nástroj, schopný číst řetězec, paralelně provádět zvolenou operaci a zapisovat výsledek.

Čtení a zápis řetězce umožňuje jedna z variant konečného automatu, tzv. konečný převodník. Jde o konečný automat, který má navíc zápisovou pásku na kterou může zapisovat výstupní řetězec. Formální definice převodníku a souvisejících pojmů je možné nalézt v [1], kapitola 9.1.

Základní konečný převodník má ale pouze jednu čtecí hlavu, a vstupní řetězec je tak možné číst pouze sekvenčně. Proto k realizaci paralelních operací musíme přistoupit k jeho rozšíření na konečný převodník s obecným počtem čtecích hlav.

3.1 KONEČNÝ PŘEVODNÍK S N ČTECÍMI HLAVAMI

Konečný převodník s n čtecími hlavami je šestice (Q, Σ, R, s, F, H) , kde

- Q je konečná množina stavů
- Σ je abeceda taková, že $\Sigma \cap Q = \emptyset, \Sigma = \Sigma_1 \cup \Sigma_0$, kde Σ_1 je vstupní abeceda, Σ_0 výstupní abeceda.
- $R \subseteq Q((\Sigma_1 \cup \{\varepsilon\})^{| \Sigma_1 \cup \{\varepsilon\} |} \times (Q\Sigma_0^*))$
- $s \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů
- $H = \{h_1 \dots h_n\}$ je uspořádaná množina čtecích hlav

Číslo n je kardinalita množiny H a udává počet čtecích hlav.

Prvky relace R nazveme pravidly, a R pak konečnou množinou pravidel. Každé pravidlo $r \in R$ je tvaru (pu, qv) , kde $p, q \in Q$, $u \in (\Sigma_1 \cup \{\varepsilon\})^{| \Sigma_1 \cup \{\varepsilon\} |} \times (Q\Sigma_0^*)$, $v \in \Sigma_0^*$.

Pravidla tvaru (pu, qv) zapisujeme zkráceně $pu \dashv\vdash qv$.

Řetězec $u = u_1 \dots u_n$ je zřetěžením symbolů přečtených při použití daného pravidla ze vstupních pásek hlavami h_1, \dots, h_n . Mezi každé dva symboly je dále vložen speciální oddělovací symbol $|$. Pokud některá hlava nemá číst vstup, objevují se tedy na daném místě dva symboly $|$.

Rozšíření definice převodníku znamená také nutnost úpravy souvisejících definic,

zejména definice vstupního konečného automatu, konfigurace automatu a převodníku, a přechodu převodníku.

3.2 VSTUPNÍ KONEČNÝ AUTOMAT

Uvažujme konečný převodník $M = (Q, \Sigma, R, s, F, H)$. Konečný automat M_I , součást převodníku M , je definován:

$M_I = (Q, \Sigma_I, R_I, s, F, H)$, kde Σ_I je vstupní abeceda převodníku M a

$$R_I = \{qu \mid \text{--- } p : p, q \in Q, u \in (\Sigma_I \cup \{\varepsilon\})^n, n = \text{card}(H), \exists qu \mid \text{--- } pw \in R, w \in \Sigma_0^*\}$$

3.3 KONFIGURACE AUTOMATU

Konfigurací automatu $M_I = (Q, \Sigma_I, R_I, s, F, H)$ rozumíme řetězec

$$\chi = qu_1 \mid u_2 \mid \dots \mid u_n, \text{ kde } u_1 \dots u_n \in \Sigma_I^*, q \in Q, \mid \text{ je speciální symbol, } \mid \notin \Sigma$$

Řetězce u_1, \dots, u_n udávají obsah vstupních pásek příslušejících hlavám h_1, \dots, h_n , q je aktuální stav automatu.

3.4 KONFIGURACE PŘEVODNÍKU

Konfigurace převodníku $M = (Q, \Sigma, R, s, F, H)$ je řetězec

$$\chi = \chi_I \# w$$

kde $\#$ je speciální symbol, $\# \notin \Sigma, w \in \Sigma_0^*$. χ_I je konfigurace automatu M_I , řetězec w udává obsah výstupní pásky

3.5 PŘECHOD

Mějme konečný převodník $M = (Q, \Sigma, R, s, F, H)$. Dále uvažujeme dvě jeho konfigurace $\chi = \chi_I \# y$ a $\chi' = \chi'_I \# yz$, a pravidlo $r: qa \mid \text{--- } pz$.

Vycházíme z definice přechodu v konečném automatu, jak je uvedena v [1]. Pokud $\chi_I \mid \text{--- } \chi'_I [qa \mid \text{--- } p]$ v konečném automatu M_I , říkáme že převodník M provedl přechod z konfigurace χ do konfigurace χ' podle pravidla r . Symbolický zápis: $\chi \mid \text{--- } \chi' [r]$

4 ZÁVĚR

V tomto článku byly představeny základní nástroje sloužící k realizaci paralelních operací v teorii formálních jazyků. Dalším krokem by měla být snaha o jejich implementaci. Tato oblast teorie FJ je velmi zajímavá a nabízí široké možnosti praktického využití v informatice.

LITERATURA

- [1] Meduna, A.: Automata and Languages: Theory and Applications, London, GB, Springer, 2000, p. 892, ISBN 1-85233-074-0
- [2] Vítek, M.: Diplomová práce: Nové operace v teorii formálních jazyků a jejich užití, Brno, ČR, 2003