PARALLEL LANGUAGE OPERATIONS - CONTROLLED TEXT INSERTION

Karel SLANÝ, Master Degree Programme (4)
Dept. of Information Systems, FIT, BUT
E-mail: xslany00@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

A new modern part of formal language theory called stringology researches strings and languages and introduces special language operations on them. These operations usually describe common changes in strings, such as inserting or deleting a substring in many ways. Controlled text insertion is an example of such an operation. This operation inserts a substring into a string after a given pattern. This paper formalizes this operation and presents an algorithm for constructing a finite automaton accepting result of this operation.

1 ÚVOD

Jazykové operace, které mění řetězce pomocí vkládání nebo změny pořadí podřetězců, hrají v moderní teoretické informatice důležitou roli. Jedná se o aplikace v kryptografii přes nejrůznější algoritmy pro zpracování textu až k DNA výpočtům.

Tato práce je zaměřena na operaci vyhledání vzoru v textu a vložení jiného za tento výskyt. Cílem je vytvořit formální model této operace a dokázat, že třída regulárních jazyků je nad touto operací uzavřena, pomocí nalezení algoritmu pro konstrukci automatu, který tento jazyk přijímá.

2 POPIS ALGORITMU

2.1 DEFINICE

Řízené vkládání je operace, která v řetězcích jazyka L_1 vyhledá výskyt vzoru náležejícího do L_2 a za tento výskyt vloží řetězec jazyka L_3. Formálně je výsledek operace řízeného vkládání popsán takto:

$$L_1 \overset{L_2}{\Leftarrow} L_3 = \left\{ u_0v_0w_0u_1v_1w_1 \dots u_kv_kw_ku_{k+1} \;\middle|\; \begin{array}{l} u_0v_0u_1v_1 \dots u_kv_ku_{k+1} \in L_1,\ k \geq 0, \\ v_i \in L_2, w_i \in L_3,\ \forall i \in \langle 0,k \rangle \end{array} \right\}, \quad (1)$$

kde $L_1 = L(M_1)$, $L_2 = L(M_2)$, $L_3 = L(M_3)$.

V dalším textu se omezíme na třídu regulárních jazyků a ukážeme, že je tato třída jazyků vzhledem k operaci řízeného vkládání uzavřená.

Před popisem konstrukce samotného automatu by jsme měli tuto operaci upravit. Definice (1) dovoluje, aby jazyk L_2 obsahoval i prázdný řetězec. To by ale vedlo ke zcela náhodnému vkládání řetězců jazyka L_3 do řetězců L_1. Proto tedy budeme předpokládat, že jazyk L_2 prázdný řetězec neobsahuje.

2.2 ZOBRAZENÍ EPSILON

Pro konstrukci automatu je důležité zobrazení *epsilon*. To zjistí v automatu M_1 výskyt cesty z každého jeho stavu odpovídající cestě z počátečního stavu M_2 do některého jeho koncového stavu.

Zobrazení *epsilon* je pro automaty $M_1 = (Q_1, \Sigma, P_1, s_1, F_1)$ a $M_2 = (Q_2, \Sigma, P_2, s_2, F_2)$ definováno takto:

$$epsilon(p) = \{q : px \vdash^* q, s_2x \vdash^* f, p,q \in Q_1, f \in F_2, x \in \Sigma^*\}.$$

Ze stavu, kterému zobrazení *epsilon* přiřazuje množinu některých stavů z M_1, je možné přijmout řetězec jazyka L_2.

2.3 KONSTRUKCE AUTOMATU

Při konstrukci hledaného automatu se vychází ze tří deterministických konečných automatů, které popisují zpracovávaný text, vyhledávaný a vkládaný vzor.

Vstup:
$M_1 = (Q_1, \Sigma, P_1, s_1, F_1)$, $M_2 = (Q_2, \Sigma, P_2, s_2, F_2)$, $M_3 = (Q_3, \Sigma, P_3, s_3, F_3)$
$L_1 = L(M_1)$, $L_2 = L(M_2)$, $L_3 = L(M_3)$

Výstup:
$M = (Q, \Sigma, P, s, F)$, $L(M) = L_1 \overset{L_2}{\Leftarrow} L_3$

Algoritmus:
$Q := Q_1, P := P_1, s := s_1, F := F_1, Q_{new} := \emptyset$

for every p in Q_1
  if $epsilon(p) \neq \emptyset$ **do begin**
    $P := P \setminus \{pa \rightarrow q : q \in Q, a \in \Sigma\}$
    $Q := Q \cup \{[p, \{s_2\}]\}$
    $Q_{new} := Q_{new} \cup \{[p, \{s_2\}]\}$
    $P := P \cup \{p\varepsilon \rightarrow [p, \{s_2\}]\}$
  end

repeat
$Q := Q \cup \{[p_2, N_2] : p_1 a \to p_2 \in P_1, q_1 a \to q_2 \in P_2, q_1 \in N_1, [p_1, N_1] \in Q_{new},$
$N_2 = \{q_2 : q_1 a \to q_2 \in P_2, q_1 \in N_1\} \cup S,$
$S = \{s_2\}$ if $epsilon(p_2) \neq \emptyset, S = \emptyset$ otherwise, $a \in \Sigma\}$
$Q_{new} := Q_{new} \cup \{[p_2, N_2] : p_1 a \to p_2 \in P_1, q_1 a \to q_2 \in P_2, q_1 \in N_1, [p_1, N_1] \in Q_{new},$
$N_2 = \{q_2 : q_1 a \to q_2 \in P_2, q_1 \in N_1\} \cup S,$
$S = \{s_2\}$ if $epsilon(p_2) \neq \emptyset, S = \emptyset$ otherwise, $a \in \Sigma\}$
$P := P \cup \{[p_1, N_1] a \to [p_2, N_2] : [p_1, N_1], [p_2, N_2] \in Q, p_1 a \to p_2 \in P_1,$
$N_2 = \{q_2 : q_1 a \to q_2 \in P_2, q_1 \in N_1\} \cup S, N_1 \cap F_2 = \emptyset,$
$S = \{s_2\}$ if $epsilon(p_2) \neq \emptyset, S = \emptyset$ otherwise, $a \in \Sigma\}$
$P := P \cup \{[p_1, N_1] a \to p_2 : p_1 a \to p_2 \in P_1, q_1 a \to q_2 \notin P_2, q_1 \in N_1, [p_1, N_1] \in Q,$
$N_1 \cap F_2 = \emptyset, a \in \Sigma\}$
$F := F \cup \{[p_1, N_1] : p_1 \in F_1, N_1 \cap F_2 = \emptyset\}$
until Q_{new} does not grow

for every $[p, N]$ in Q_{new}
if $N \cap F_2 \neq \emptyset$ **do begin**
$Q := Q \cup \{[p, r] : r \in Q_3\}$
$P := P \cup \{[p, r_1] a \to [p, r_2] : r_1 a \to r_2 \in P_3, a \in \Sigma\}$
$P := P \cup \{[p, N] \varepsilon \to [p, s_3]\}$
$P := P \cup \{[p, r] \varepsilon \to p : r \in F_3\}$
end

3 ZÁVĚR

Při použití tohoto algoritmu dostáváme jako výsledek rozšířený konečný automat. Ten má stejné schopnosti v přijímání jazyků jako klasický konečný automat. Tento postup zároveň ukazuje, že množina regulárních jazyků je uzavřena nad operací řízeného vkládání.

LITERATURA

[1] Meduna, A., Vítek, M.: New Language Operations in Formal Language Theory, In: Schedae Informaticae, č. 13, Krakov, roč. 2004, s. 123-150. ISSN 0860-0295

[2] Kari, L.: On Insertion and Deletion in Formal Languages, PhD thesis, University of Turuku, Finland, Department of Mathematics, 1991.

[3] Meduna, A.: Automata and Languages: Theory and Applications, Springer, London, 2000. ISBN 1-85233-074-0