NEW TEXT ALGORITHMS

Ján KUBEK, Master Degree Programme (5)
Dept. of Computer Systems, FIT, BUT
E-mail: xkubek00@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This project deals with search for new operations above text strings. The aim of this project is to define efective algorithms working with text. Generally used algorithms have reduced possibilities and are very slow in common.

1 ÚVOD

V této práci se budu zabývat novými operacemi v oboru *stringologie*. Uvádím postupy, které vytváří uzavřenou množinu operací, řešící sériové vyhledávání a případnou úpravu hledaného řetězce v textu.

2 DEFINICE

Předpokládám, že čtenář je seznámen s teorií jazyků (viz [1]). Nedeterministický konečný převodník (NKP) M je definován jako šestice

$$M = (Q, \Sigma, \Delta, \delta, q_0, F),$$

kde Q je konečná množina stavů, Σ je konečná vstupní abeceda, Δ je konečná výstupní abeceda, δ je přechodová funkce ve tvaru $\delta \subseteq Q \times (\Sigma \cup \varepsilon) \times Q \times \Delta^*$, $q_0 \in Q$ je počáteční stav a $F \subseteq Q$ je konečná množina koncových stavů. Řetězec w přijímaný NKP M je definován jako $(q_0, w, \varepsilon) \rightarrow^* (q, \varepsilon, y);\ q \in F$. Jazyk $L_{in}(M)$ přijímaný NKP je jazyk $L_{in}(M) = \{w; (q0, w, \varepsilon) \rightarrow^* (q, e, y) \wedge q \in F\}$. Řetězec y generovaný NKP M je definován jako

$$(q_0, w, \varepsilon) \rightarrow^* (q, \varepsilon, y);\ y \in \Delta^*;\ q \in F.$$

Jazyk $L_{out}(M)$ generovaný NKP je jazyk

$$L_{out}(M) = \{y; (q0, w, \varepsilon) \rightarrow^* (q, e, y) \wedge w \in L_{in} \wedge q \in F\}.$$

Autopřevodník je speciální případ konečného převodníku $M = (Q, \Sigma, \Delta, \delta, q_0, F)$, u kterého platí, že $\Sigma = \Delta$.

3 IMPLICITNÍ AUTOPŘEVODNÍKY

Algoritmicky je velice jednoduše možno převést libovolný konečný automat $M = (Q, \Sigma, \delta_A, q_0, F)$ na jeden z jeho *implicitních* autopřevodníků $M_C = (Q, \Sigma, \Sigma, \delta_C, q_0, F)$. Podle úpravy přepisovacích pravidel lze implicitní autopřevodníky rozdělit na čtyři skupiny.

Vyhledávací (přepisovací) autopřevodník $q' \in \delta_A(q,a) \Rightarrow q' \in \delta_C(q,a,a)$ vyhledává řetězec jazyka $L(M_A)$ a kopíruje jej na výstup převodníku.

Mazací autopřevodník $q' \in \delta_A(q,a) \Rightarrow q' \in \delta_C(q,a,\varepsilon)$ vyhledává řetězec jazyka $L(M_A)$, výstupem převodníku je prázdný řetězec, tzn. že tento řetězec vymaže.

Doplňující autopřevodník $q' \in \delta_A(q,a) \Rightarrow q' \in \delta_C(q,\varepsilon,a)$ ignoruje vstupní řetězec, výstupem převodníku je řetězec jazyka $L(M_A)$.

Poslední variantou je prázdný autopřevodník $q' \in \delta_A(q,a) \Rightarrow q' \in \delta_C(q,\varepsilon,\varepsilon)$ ignoruje vstupní řetězec, negeneruje výstupní řetězec. Nemá žádné využití.

4 NOVÉ JAZYKOVÉ OPERACE

V této práci bude využití implicitních autopřevodníků (IA) stěžejní záležitostí. Mějme řetězec w, patřící do jazyka L, $w \in L$, který je vyjádřen konečným automatem M, $M = L(M)$. Na tomto řetězci budeme chtít provádět vyhledávání jiného řetězce (podřetězce), případně přímo najít a nahradit (respektive vymazat) část řetězce.

5 INVERZNÍ REZIDUUM

Operace inverzního rezidua maže ze vstupního řetězce w, definovaného konečným automatem $M_T = (Q_T, \Sigma, \delta_T, q_{T0}, F_T)$ ($w \in L(M_T)$) všechny podřetězce, které odpovídají řetězcům jazyka $L(M_V)$ konečného automatu $M_V = (Q_V, \Sigma, \delta_V, q_{V0}, F_V)$. Výstupem operace inverzního rezidua je tedy řetězec jazyka $L(M_T)$, ze kterého jsou odstraněny řetězce jazyka $L(M_V)$. Jazyk vzniklý touto operací je

$$L_R = \mathit{invreziduum}(L_V, L_T) = \{t_1 t_2 ... t_n; t_1 v_1 t_2 v_2 ... v_{n-1} t_n \in L_T;\ t_1 t_2 ... t_n \in L_T;\ v_1, v_2, ...vn \in L_V\}.$$

Operaci inverzního rezidua vyřešíme vytvořením převodníku $M_C = (Q_C, \Sigma, \Sigma, \delta_C, q_{C0}, F_C)$, který bude vyhledávat a mazat řetězce jazyka L_V ve vstupním řetězci.

Neformálně se převodník M_C pro funkci $\mathit{invreziduum}(M_V, M_T)$ vytvoří převedením automatu M_V na mazací implicitní autopřevodník a automatu M_T na přepisovací implicitní autopřevodník. Kartézským součinem stavů těchto implicitních autopřevodníků (vzorec 1) vznikne *kartézské převodníkové pole*, ve kterém jsou zavedeny přechody autopřevodníku M'_V (vzorec 5). Kartézské převodníkové pole (KPP) je pak propojeno s přepisovacím autopřevodníkem M'_T přechody ε/ε, vždy ze stavu převodníku M'_T do odpovídajícího startovního stavu převodníku M'_V v KPP (vzorec 3) a z koncového (resp. koncových, má-li jich tento více) stavu převodníku M'_V zpět do odpovídajícího stavu převodníku M'_T (vzorec 4). Startovací stav (resp. koncové stavy) převodníku M_C odpovídají startovacím (resp. koncovým) stavům převodníku M'_T (vzorec 7).

$$Q := \{Q_T \cup \langle Q_T \times Q_V \rangle\} \tag{1}$$

$$\delta_{C1} := \{q_T a \to q'_T a;\ q_T a \to q'_T \in \delta_T;\ q, q' \in Q_T;\ a \in \Sigma\} \quad (2)$$

$$\delta_{C2} := \{q_T \varepsilon \to \langle q_T, q_{V0} \rangle \varepsilon;\ q_T \in Q_T\} \quad (3)$$

$$\delta_{C3} := \{\langle q_T, q_V \rangle \varepsilon \to q_T \varepsilon;\ q_T \in Q_T;\ q_V \in F_V\} \quad (4)$$

$$\delta_{C4} := \{\langle q_T, q_V \rangle a \to \langle q_T, q'_V \rangle \varepsilon;\ q_V a \to q'_V \in \delta_V;\ q_T \in Q_T;\ q_V, q'_V \in Q_V;\ a \in \Sigma\} \quad (5)$$

$$\delta_C := \delta_{C1} \cup \delta_{C2} \cup \delta_{C3} \cup \delta_{C4} \quad (6)$$

$$q_{C0} = q_{T0};\ F_C := F_T \quad (7)$$

6 PŘÍMÉ REZIDUUM

Operace přímého rezidua maže vstupního řetězec w, který je definován konečným automatem $M_T = (Q_T, \Sigma, \delta_T, q_{T0}, F_T)$ kromě podřetězců, které odpovídají řetězcům jazyka $L(M_V)$ konečného automatu $M_V = (Q_V, \Sigma, \delta_V, q_{V0}, F_V)$. Výstupem operace přímého rezidua je tedy kontatenace řetězců jazyka $L(M_V)$. Princip konstrukce konečného převodníku pro tuto operaci je analogií předchozího, proto ji zde nebudu uvádět.

7 MOŽNOSTI KARTÉZSKÉHO PŘEVODNÍKOVÉHO POLE

V obou uvedených operacích využíváme kartézské převodníkové pole přidané k základnímu převodníku, jehož přijímaný jazyk L_T odpovídá jazyku textu, který je zpracováván. Nabízí se otázka, kam až sahají možnosti převodníků s KPP, tedy jaké operace lze s využitím KPP vytvořit. Uvažujme tedy, že převodník pro průchod textem M'_T může být pouze přepisovacím a nebo mazacím autopřevodníkem (tzn. přechody budou typu a/a, resp. a/ε). Převodník vyhledávaného textu M'_V nám ale nabízí za možnosti všechny druhy autopřevodníků. Vzniká tím prostor pro využití doplňujícího a prázdného autopřevodníku. Tyto označuji jako M'_D. Některé operace vzniknou vhodnou konkatenací převodníků M'_V (vyhledávání nebo mazání) a M'_D (náhrada).

Veškeré možné operace převodníku s kartézským převodníkovým polem jsou určeny kombinacemi možných druhů převodníků M'_T, M'_Va M'_D.

8 ZÁVĚR

Použitím kartézského převodníkového pole vzniká prostor pro nové jazykové operace, které jsou popsány v mé práci. Jde o operace *Nahrazení* a *Doplnění prefixu/suffixu*.

Využití operací, které jsem v mé práci definoval je víceméně zřejmé – v moderních textových editorech. Jedinou překážkou přímé aplikace je fakt, že vytvořené převodníky nejsou deterministické, tzn. že podle aktuálního stavu a vstupního symbolu není možno určit který stav bude následovat, je možno pouze určit množinu následujících stavů.

REFERENCE

[1] Meduna, A.: Automata and Languages: Theory and Applications, Springer, London, 2000.