Zásobníkové automaty s omezeným obsahem zásobníku

Ing. Zbyněk Křivka

FIT VUT Brno,
Božetěchova 2, 612 66 Brno
krivka@fit.vutbr.cz

Vedoucí práce: Doc. RNDr. Alexander Meduna, CSc.

Abstrakt: Prezentovaný článek zavádí nový pojem zásobníkového automatu s omezeným obsahem zásobníku, který vychází z modifikace klasického zásobníkového automatu. Obsah zásobníku je omezován tzv. omezujícím jazykem, na jehož klasifikaci v Chomského hierarchii závisí výsledná mocnost takovéhoto automatu. Ukazuji, že pokud je tento omezující jazyk regulární (např. reprezentován konečným automatem), tak systém nepřekročí mocnost bezkontextových jazyků, což je demonstrováno mimo jiné i rigorózním důkazem. V poslední části je prezentován krátký příklad a diskuse využití modifikované metody LR syntaktické analýzy pro navržený systém.

Klíčová slova: zásobníkový automat, bezkontextový jazyk, konečný automat, regulární jazyk, omezování zásobníku, třídy jazyků Chomského hierarchie

1 Úvod

V této práci zavádím nový pojem - zásobníkové automaty s omezeným obsahem zásobníku, u kterých diskutuji závislost jejich mocnosti na třídě jazyků, do které spadá omezující jazyk. Hlavní část práce je věnována důkazu, že pokud je omezující jazyk regulární, tak výsledná mocnost automatu zůstává nezměněna, tedy přijímané jazyky spadají do třídy bezkontextových jazyků (CFL).

Kromě kapitoly 2 se základními pojmy je veškerý obsah tohoto článku výsledkem mé vlastní výzkumné činnosti založené na podnětných odborných konzultacích mého školitele a vychází z části mé diplomové práce [4].

2 Základní pojmy

Než začnu s dalším výkladem, tak bude vhodné pro zopakování doplnit ještě několik základních pojmů teoretické informatiky. Kromě vlastních znalostí jsem čerpal především z [4], [4] a [4], kde naleznete další podrobnosti, vysvětlivky a příklady.

Definice 1: Konečný automat je pětice $M_{FA} = (Q, \Sigma, R, S, F)$, kde Q je konečná množina stavů, Σ je konečná vstupní abeceda, $R \subseteq Q \times \Sigma \rightarrow Q$ je konečná množina pravidel tvaru $pa \rightarrow q$, kde $p, q \in Q$ a $a \in \Sigma$, $S \in Q$ je počáteční stav a $F \subseteq Q$ je množina konečných stavů.

Definice 2: Bezkontextová gramatika G = (N, T, P, S*)*, kde N a T jsou disjunktní abecedy (neterminály a terminály), $S \in N$ a P je konečná množina pravidel tvaru $A \rightarrow \gamma$, kde $A \in N$ a $\gamma \in (N \cup T)^*$. Jestliže $A \rightarrow x \in P$ a $u, v \in T^*$, pak $uAv \Rightarrow uxv$ $[A \rightarrow x]$ nebo zkráceně $uAv \Rightarrow uxv$. Standardním způsobem rozšiřme operaci přímého derivačního kroku $\Rightarrow$ na $\Rightarrow^n$ pro $n \geq 0$. Pak na základě $\Rightarrow^n$ definujme $\Rightarrow^+$ (tranzitivní uzávěr operace) a $\Rightarrow^*$ (reflexivní a tranzitivní uzávěr). Jazyk gramatiky G, L(G), je definován jako $L(G) = \{ w \mid S \Rightarrow^* w,\ w \in T^* \}$.
Definice 3: Nechť G = (N, T, P, S) je bezkontextová gramatika. G reprezentuje **lineární gramatiku**, jestliže pro každé její pravidlo tvaru $A \rightarrow x$, platí $x \in T^* (N \cup \{\varepsilon\})T^*$, $A \in N$.
Definice 4: Jazyk L je **lineární** právě když L = L(G), kde G je lineární gramatika.
Definice 5: Nechť G = (N, T, P, S) je lineární gramatika. G nazveme **regulární gramatikou**, jestliže pro každé její pravidlo $A \rightarrow x \in P$ platí $x \in T(N \cup \{\varepsilon\})$.
Definice 6: Jazyk L je **regulární** (REG) právě když L = L(G), kde G je regulární gramatika.
Definice 7: Zásobníkový automat je sedmice $M_{PDA} = (Q, \Sigma, \Gamma, R, s, S, F)$, kde Q je konečná množina stavů, Σ je vstupní abeceda, Γ je zásobníková abeceda, $R \subseteq \Gamma \times Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \Gamma^* \times Q$ je konečná množina pravidel, $s \in Q$ je počáteční stav, $S \in \Gamma$ je počáteční symbol na zásobníku, $F \subseteq Q$ je konečná množina koncových stavů. Q, Σ, Ω jsou po dvojicích navzájem disjunktní. Pravidla zapisujeme ve tvaru $Apa \rightarrow wq$ (případně použiji alternativní tvar $(A, p, a) \rightarrow (w, q)$), kde $A \in \Gamma$, $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $w \in \Gamma^*$. Pokud je p aktuální stav, a aktuální vstupní symbol, A je na vrcholu zásobníku a $Apa \rightarrow wq \in R$, pak tento zásobníkový automat M_{PDA} může přečíst symbol a, změnit vrchol zásobníku z A na w a nakonec změnit stav z p na q.
Definice 8: Konfigurace zásobníkového automatu je trojice $(\alpha, p, w) \in \Gamma^* \times Q \times \Sigma^*$, kde p označuje aktuální stav, w zatím nezpracovaný řetězec na vstupu a α celý obsah zásobníku. **Přechodem** potom nazýváme binární relaci $\vdash \subseteq \Gamma^* \times Q \times \Sigma^* \rightarrow \Gamma^* \times Q \times \Sigma^*$, pro kterou platí, že $(z\gamma, p, aw) \vdash (\beta\gamma, q, w) \Leftrightarrow (z, p, a) \rightarrow (\beta, q) \in R$, *kde* $z \in \Gamma$, $\gamma, \beta \in \Gamma^*$, $p, q \in Q$, $a \in \Sigma$, $w \in \Sigma^*$.

Poznámka: Pokud je a rovno ε, tak to znamená, že ze vstupu nebyl přečten žádný symbol.

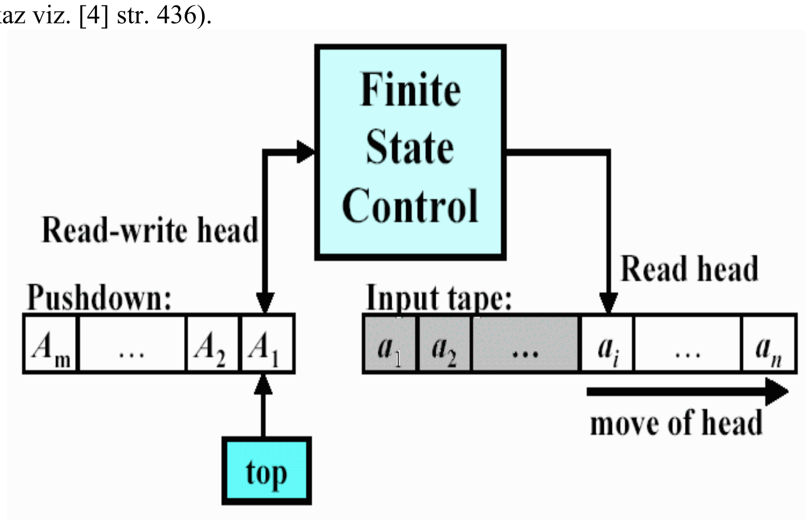
Standardním způsobem rozšířím relaci $\vdash$ na $\vdash^n$ pro $n \geq 0$. Pak na základě $\vdash^n$ definujme $\vdash^+$ a $\vdash^*$.
Definice 9: Typy přijímaných jazyků zásobníkovými automaty:
Nechť $M_{PDA} = (Q, \Sigma, \Gamma, R, s, S, F)$ je zásobníkový automat, pak jsou definovány tyto jazyky:

1. přijímaný koncovým stavem $L(M)_f = \{ w \mid w \in \Sigma^*, Ssw \vdash^* zf, z \in \Gamma^*, f \in F \}$
2. přijímaný prázdným zásobníkem $L(M)_\varepsilon = \{ w \mid w \in \Sigma^*, Ssw \vdash^* zf, z \in \varepsilon, f \in Q \}$
3. přijímaný koncovým stavem a prázdným zásobníkem $L(M)_{f\varepsilon} = \{ w \mid w \in \Sigma^*, Ssw \vdash^* zf, z \in \varepsilon, f \in F \}$.

Poznámka: Všechny tři typy jazyků mají naprosto stejnou mocnost (tzn. každý z těchto typů jazyků lze algoritmicky převést na libovolný jiný typ jazyka (1 až 3)).
Definice 10: Deterministický zásobníkový automat: Nechť $M_{DPDA} = (Q, \Sigma, \Omega, R, s, S, F)$ je zásobníkový automat. M_{DPDA} je deterministický, pokud pro každé pravidlo

tvaru $Apa \rightarrow wq \in R$ platí, že množina vzniklá rozdílem $R \setminus \{Apa \rightarrow wq\}$ neobsahuje žádné jiné pravidlo tvaru $Apa \rightarrow vo$ nebo $Ap \rightarrow vo$, kde $A \in N$, $a \in \Sigma \cup \{\varepsilon\}$, $v \in \Sigma^*$, $o \in Q$.

Takže v každé konfiguraci umožňuje deterministický zásobníkový automat vybrat nejvýše jedno pravidlo pro přechod do jiné konfigurace. Třída jazyků akceptovaná deterministickými zásobníkovými automaty má menší mocnost než třída bezkontextových jazyků (CFL) (tedy i než nedeterministické zásobníkové automaty) (Důkaz viz. [4] str. 436).



Obr.1. Schéma klasického zásobníkového automatu (převzato z [4])

Pro srovnání s nově definovaným automatem (obr. 2) je zde přiložen obr. 1 znázorňující klasický zásobníkový automat.

Definice 11: Nechť máme řetězec $w \in \Sigma^*$ (Σ je lib. abeceda), $a \in \Sigma$, $w = av$. Pak $suffix(\varepsilon) = \varepsilon$ nebo $suffix(w) = w$ nebo $suffix(w) = suffix(v)$.

3 Zásobníkový automat s omezovaných obsahem zásobníku

Nyní již mám sjednoceny definice obecně používaných pojmů teoretické informatiky nutných pro stanovení nových definic pro popis nově zaváděného modifikovaného zásobníkového automatu, který bude fungovat obdobně jako klasický automat, ale po každém jeho kroku bude prováděna kontrola obsahu zásobníku vůči tzv. omezujícímu jazyku. Klasifikace omezujícího jazyka v třídách Chomského hierarchie jazyků bude ovlivňovat výslednou mocnost takto modifikovaného automatu.

Hlavní myšlenka byla inspirována nedávným článekem [4] ohledně řízených zásobníkových automatů, které sice svou činnost kontrolují poněkud odlišným způsobem, ale některé vlastnosti jsou ve výsledku analogické.

3.1 Nové definice

Nyní již uvedu nově zaváděný systém, který je hlavním předmětem celého článku.

Definice 12:

Mějme zásobníkový automat $M = (Q, \Sigma, \Gamma, R, s, S, F)$ a omezující jazyk $\Xi \subseteq \Gamma^*$. Dále nechť množina možných obsahů zásobníku automatu M při analýze věty w je definována, takto:

$K(M, w) = \{ \gamma \mid M=(Q, \Sigma, \Gamma, R, s, S, F), w\in\Sigma^*, w\in L(M),$ posloupnost přechodů $(S, s, w) \vdash^* (\gamma, q, u)$, že $\exists$ posloupnost přechodů $(\gamma, q, u) \vdash^* (\gamma_F, q_F, \varepsilon), u = suffix(w), q\in Q, q_F \in F, \gamma,\gamma_F \in \Gamma^*\}$.

Nechť $n \in \{1, 2, 3\}$ značí způsob, jakým zásobníkový automat přijímá jazyk:

- $n = 1$: koncovým stavem,
- $n = 2$: prázdným zásobníkem,
- $n = 3$: koncovým stavem i prázdným zásobníkem.

Pak jazyk $L(M, \Xi, n) = \{ w \mid w \in \Sigma^*, w\in L(M), K(M, w) \subseteq \Xi \}$ je **jazykem přijímaným zásobníkovým automatem M s omezeným obsahem zásobníku jazykem** Ξ, kde L(M) značí jazyk přijímaný automatem M podle definice 9.

Tento systém, resp. jeho jazyk, bude tedy možno alternativně značit $L(\overline{M}, L(M))$, kde $\overline{M}$ je zásobníkový automat typu $n = 3$ a M je konečný automat. Rád bych ještě podotknul, že schopnosti přijímání jazyků pro jednotlivé typy zásobníkového automatu z definice 9 {1, 2, 3} jsou navzájem ekvivalentní.

Třídu jazyků přijímanou tímto novým automatem značme $L(M_{PDA}, REG)$.

Popis definice 12: Množina K(M, w) slouží jako prostředek pro jednodušší definici jazyka $L(M, L(M_{FA}))$, resp. jednoho jeho omezení (týkajícího se obsahů zásobníku). Pro lepší pochopení definice množiny K(M, w) zdůrazňuji, že se skládá z řetězců γ, které nalezneme v konfiguracích zásobníkového automatu M při přijímání věty w (včetně počáteční a koncové konfigurace).

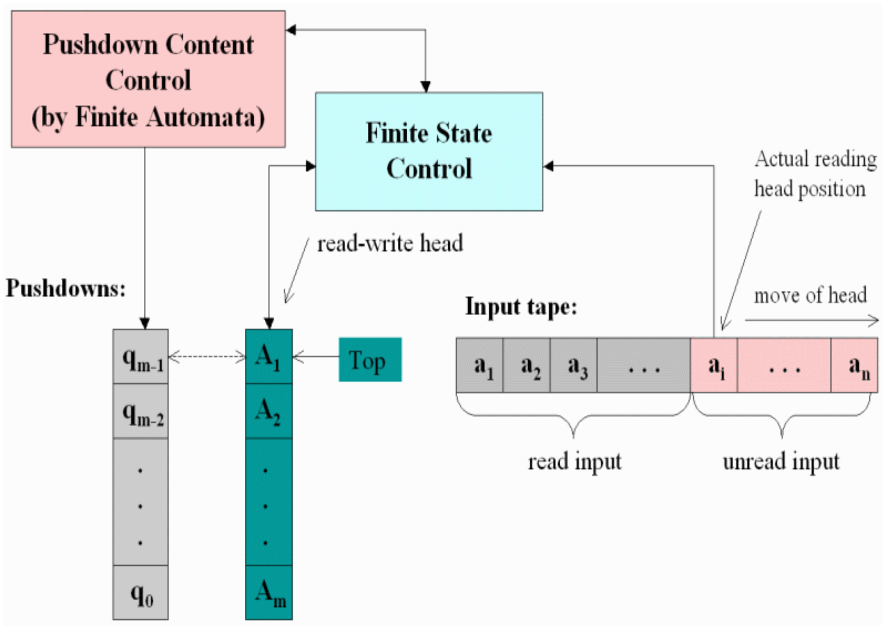
Definice 13: Jako **omezující jazyk** budu označovat jazyk Ξ kontrolující povolený obsah zásobníku. Analogický význam bude mít také pojem **omezující automat** případně **omezující konečný automat**.

Definice 14: Již pouze poloformálně doplním, že se pro $L(M, L(M_{FA}))$, kde $M = (Q, \Sigma, \Gamma, R, s, S, F)$ a $M_{FA} = (Q_{FA}, \Gamma, R_{FA}, p_0, F_{FA})$, musí rozšířit definice **konfigurace** (z definice 8) o položku obsahu zásobníku stavů omezujícího automatu M_{FA} na čtveřici $(\alpha, \tau, p, w) \in \Gamma^* \times Q_{FA}^* \times Q \times \Sigma^*$. A následně i definici **přechodu** (opět vycházím z definice 8):

Přechod je binární relace $\vdash \subseteq \Gamma^* \times Q_{FA}^* \times Q \times \Sigma^* \rightarrow \Gamma^* \times Q_{FA}^* \times Q \times \Sigma^*$, pro kterou platí:

$(z\gamma, rr_0\delta, p, aw) \vdash (\beta\gamma, \rho\delta, q, w) \Leftrightarrow (z, p, a)\rightarrow(\beta, q) \in R$ a $b_1r_0\rightarrow r_1, b_2r_1\rightarrow r_2, \ldots, b_nr_{n-1}\rightarrow r_n \in R_{FA}$, kde $\beta = b_1b_2\ldots b_n$, $\rho = r_0r_1r_2\ldots r_n$, $r, r_0, \ldots, r_n \in Q_{FA}$, $\delta, \rho \in Q_{FA}^*$, $z \in \Gamma$, $\gamma, \beta \in \Gamma^*$, $p, q \in Q$, $a \in \Sigma$, $w \in \Sigma^*$. *(PS: pokud* $\beta = \varepsilon$, *tak* $\rho = r_0$*)*

Po rozšíření relace přechodu na její tranzitivní a případně i reflexivní uzávěr lze provádět obecnou syntaktickou analýzu podobně jako u klasického zásobníkového automatu. Počáteční konfigurace bude (S, pp_0, s, w), kde p_0 je počáteční stav omezujícího automatu a $Sp_0\rightarrow p\in R_{FA}$.



Obr.2. Zásobníkový automat s omezeným obsahem zásobníku

Na obrázku je zásobníkový automat s obsahem omezovaným pouze konečným automatem. V budoucím rozvoji tohoto systému plánuji rozšíření schopností omezujícího mechanismu na mocnost zásobníkových automatů (tedy bezkontextových jazyků).

3.2 Věta pro variantu regulárního omezujícího jazyka

Jak už bylo poznamenáno výše, tak mocnost omezujícího jazyka určuje výslednou sílu celého systému, která se pro případ regulárního omezujícího jazyka nezmění, což ukazuje následující věta.

Věta 1: Pro každý konečný automat M a každý zásobníkový automat $\overline{M}$ existuje zásobníkový automat $\widetilde{M}$, že platí $L(\widetilde{M}) = L(\overline{M}, L(M))$, a tedy z pohledu tříd jazyků CFL = L(M_{PDA}, REG).

Důkaz věty 1: Rigorózní důkaz je relativně jednoduchý a zápisem odpovídá algoritmu, který lze pro převod omezeného zásobníkového automatu na obyčejný zásobníkový automat použít. Při vytváření důkazu jsem se inspiroval v článku [4].

Protože převod obyčejného zásobníkového automatu na automat s omezeným obsahem zásobníku je triviální, tak stačí ukázat postup opačný.

Formální důkaz:

Nechť $M = (Q, \Sigma, R, s, F)$ je konečný automat a $\overline{M} = (\overline{Q}, \overline{\Gamma}, \overline{\Sigma}, \overline{R}, \overline{s}, \overline{S}, \overline{F})$ zásobníkový automat. Nyní zkonstruujeme zásobníkový automat $\widetilde{M}$, že bude platit $L(\overline{M}, L(M)) = L(\widetilde{M})$.

Definujme tedy nový zásobníkový automat $\widetilde{M} = (\widetilde{Q}, \widetilde{\Gamma}, \widetilde{\Sigma}, \widetilde{R}, \widetilde{s}, \widetilde{S}, \widetilde{F})$, u něhož položme

$\widetilde{Q} = \overline{Q} \cup \{\widetilde{s}\},\ \widetilde{\Gamma} = \overline{\Gamma} \cup \{\widetilde{S}\} \cup Q,\ \widetilde{\Sigma} = \overline{\Sigma},\ \widetilde{F} = \overline{F}$ a

$\widetilde{R} = \{\widetilde{S}\,\widetilde{s} \to s\,\overline{S}\,\overline{s}\} \cup \{p_1 B_1 p_2 B_2 \ldots p_m B_m p_{m+1} \overline{o} a \to q_1 A_1 q_2 A_2 \ldots q_n A_n q_{n+1} \overline{r} \mid$

$\mid \mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_\mathrm{m} \in Q,\ p_{m+1} \in F,\ m \geq 0,\ B_1, B_2, \ldots, B_m \in \overline{\Gamma},$

$A_1, A_2, \ldots, A_m \in \overline{\Gamma} \cap \Sigma,\ q_1 A_1 \to q_2,\ q_2 A_2 \to q_2, \ldots, q_n A_n \to q_{n+1} \in R,\ q_{n+1} \in F,$

$n \geq 0,\ B_1 B_2 \ldots B_m \overline{o} a \to A_1 A_2 \ldots A_n \overline{r} \in \overline{R},\ \overline{o}, \overline{r} \in \overline{Q}\}.$

Kromě prvního pravidla, které zastupuje inicializační fázi, jsou všechna ostatní prokládána stavy konečného automatu M, který výše definovaným způsobem omezuje přípustný obsah zásobníku. Abychom dosáhli maximální efektivity, tak je nutné si pamatovat všechny stavy ($p_1 B_1 p_2 B_2 \ldots p_m B_m p_{m+1} \overline{o} a \to q_1 A_1 q_2 A_2 \ldots q_n A_n q_{n+1} \overline{r}$), kterými je nutno projít při návratu v M a redukci řetězce na zásobníku. Hlavním požadavkem je, aby po každé operaci se zásobníkem byl poslední stav ($q_{n+1} \in F$) koncový, a tím zajišťoval příslušnost do jazyka L(M).

Rigorózní důkaz o skutečné rovnosti zkonstruovaného automatu $L(\overline{M}, L(M)) = L(\widetilde{M})$ přenechávám na laskavém čtenáři.

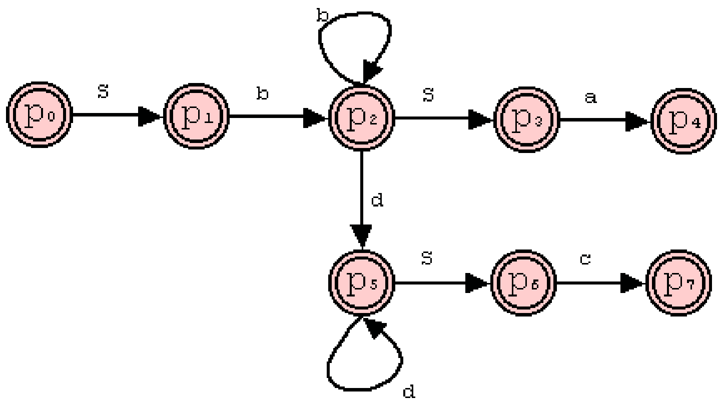
Pro krajní případy jen poznamenejme, že pokud je $m = 0$, pak $p_1 B_1 p_2 B_2 \ldots p_m B_m p_{m+1} = p_1$ a pokud $n = 0$, tak $q_1 A_1 \ldots q_n A_n q_{n+1} \to q_1$.

3.3 Ukázkový příklad

Z nedostatku prostoru pouze uvedu ukázku zadaní automatu pro jazyk L(M, L(M_{FA})) s krátkým popisem.

Automat M je popsán gramatikou G = ({S}, {a, b, c, d}, {S→ε, S→aSbS, S→cSd}, S), omezující automat M_{FA} je zadán graficky na obrázku 3. Poznamenejme, že gramatika G lze na zásobníkový automat převést algoritmicky (viz. [4]).

Samotná gramatika G popisuje jednoduchý závorkový jazyk (písmena a, b označují např. kulaté závorky a c, d hranaté). Pokud do analýzy věty zapojíme i omezující automat, tak zjistíme, že uvnitř hranatých závorek se již nemohou vyskytovat žádné kulaté, přestože to samotná gramatika G nezajišťuje.



Obr.3. Omezující automat M_{FA} pro ukázkový příklad.

3.4 Poznámka k syntaktické analýze

Pro ukázkový příklad automatu byl naprogramován experimentální syntaktický analyzátor založený na LR syntaktické analýze (metoda zdola nahoru) a to pouze pro regulární omezující jazyk. Program je součástí příloh diplomové práce [4].

Výsledek využití LR metody však nebyl příliš uspokojivý a v budoucnu budou muset být prověřeny i jiné metody, které budou pro praktickou analýzu použitelnější (např. rekurzivní sestup, tedy metoda shora dolů). Nebo vytvořit naprosto nové specializované řešení.

Algoritmus LR metody, který musel být mimo jiné také upraven pro zajištění dodatečných omezujících kontrol obsahů zásobníku, totiž pracuje po jednotlivých vstupních terminálech a tak musí být omezující automat v některých (včetně ukázkového příkladu) případech podstatně složitější nebo se neobejdeme bez zkomplikování náročnější bezkontextové části systému (gramatika zastupující zásobníkový automat). V praxi je metoda vhodná pouze pro případy, kdy omezujícím jazykem je tzv. prefixový jazyk.

4 Závěr

Celý systém zatím nemá dostatečně demonstrováno případné praktické využití, ale již nyní je jasné, že mnohem zajímavější budou až budoucí kapitoly věnující se silnějším omezujícím jazykům než regulárním. Mocnost takovýchto systémů bezpochyby převýší třídu bezkontextových jazyků a dokonce existuje hypotéza o dosažení třídy typu 0 (tedy rekurzivně spočetné jazyky) pouhým využitím lineárního omezovacího jazyka.

V diplomové práci, ze které této článek vzešel, je podrobněji popsán důvod (i s příkladem), proč není pro tento nový druh automatu vhodná metoda LR syntaktické analýzy.

Hlavní výhodou tohoto systému je přenesení jednodušších částí z bezkontextového mechanismu na regulární. Bohužel na druhou stranu je velkým neduhem režie na přepínání mezi oběma režimy - nutnost přidání kontrol na správnost obsahů zásobníku a také paměť pro pomocný zásobník stavů konečného automatu. A tak ve výsledku nemá systém s regulárním omezujícím jazykem zatím valný praktický význam (např. pro psaní syntaktických analyzátorů v překladačích). Práce má však zajímavý teoretický výsledek v dokázané větě 1.

To vše ale nemusí platit pro složitější variantu využívající lineární nebo dokonce bezkontextový omezující jazyk, kdy mocnost celého systému rapidně roste a porovnání s některými nedeterministickými metodami může vyznít podstatně kladněji. Tato pasáž je však teprve ve stádiu zkoumání.

Jeden ze základních obecnějších postřehů nejen tohoto článku je přístup, jak zvyšovat mocnost zkoumaného systému pomocí zavádění dodatečných restriktivních mechanizmů (ovlivňujících inkluzi zpracovávané věty do jazyka) k systému původnímu.

Literatura

1. Češka, M.: *Gramatiky a jazyky*, FEI VUT Brno, 1992.
2. Křivka, Z.: *Syntaktická analýza založená na multigenerování* [diplomová práce], FIT VUT Brno, 2004.
3. Meduna, A.: *Automata and Languages*. Springer, London, 2000, ISBN 1-85233-074-0, (anglicky).
4. Meduna, A.: *Základy překladačů* [texty k přednáškám], FIT VUT Brno, 2001.
5. Meduna, A., Kolář D.: Regulated Pushdown Automata. *Acta Cybernetica*, roč. 2000, č. 4, US, ISSN 0324-721X, 653-664, (anglicky).

Annotation:

Pushdown automata with restricted pushdown contents

This paper introduces new term – pushdown automata with restricted pushdown contents. The article concentrates on restricting by regular language (represented by finite automata). There is also a proof that this system has the same power as context-free languages. A simple example of such automata with few paragraphs describing practical problems of analyzing sentences with bottom-up parsing methods (eg. LR parsing) is presented at the end. The most general knowledge is that powerness of any system can be increased by using some restriction mechanism. This follows from the fact that language Σ^* (where Σ is an alphabeth of terminals) is regular language.