

# DEEP TOP–DOWN PARSERS

Tomáš HORNÍ, Master Degree Programme (4)  
Dept. of Information Systems, FIT, BUT  
E-mail: xhorni01@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

## ABSTRACT

This text describes some aspects of the classical general top–down parsers formalized by pushdown automata. This generalization consists in allowing them to make expansion deeper in the pushdown. Based on the expansion depth, this paper established an infinite hierarchy of language between the family of context–free and context-sensitive languages.

## 1 ÚVOD

Tento příspěvek pojednává o základních funkcích a vlastnostech automatů s hlubokými zásobníky. Zachycuje jejich podstatu, definici, funkce, základní rozdíly oproti klasickým zásobníkovým automatům a zdůrazňuje některé výhody a důvod, proč je vlastně používat a implementovat.

## 2 ROZBOR

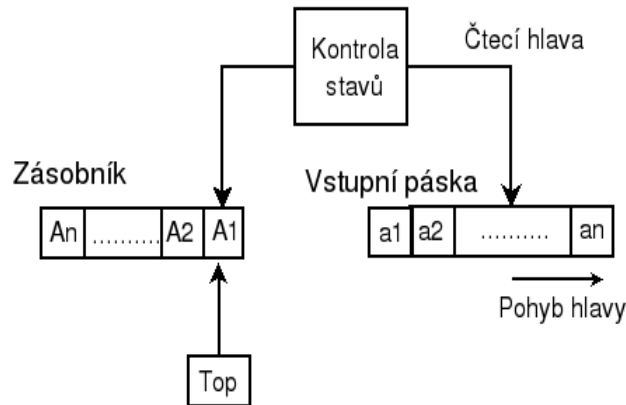
Zásobníkový automat je formální model, schéma je uvedeno níže, který čte data ze vstupní pásky a podle aktuálního symbolu na zásobníku vykonává dvě činnosti. Buď expanduje, tj. přepisuje symbol na vrcholu zásobníku řetězcem symbolů podle gramatiky, nebo symbol z vrcholu zásobníku odebere. To nastane, pokud je aktuální symbol ve vstupním řetězci (na vstupní pásce) shodný se symbolem na vrcholu zásobníku. Automat akceptuje jazyk (případně řetězec z jazyka) tak, že vykoná posloupnost kroků, které vedou k přijetí všech symbolů řetězce na vstupní pásce. Automaty lze využít všude, kde je třeba definovat postupy podle určitých pravidel. Tato pravidla vycházejí z gramatiky jazyka a v praxi určují stavy a chování modelu. Pod slovem automat si lze představit i funkční program, reagující na vstupy a předávající určité výstupy, např. překladač programovacího jazyka. Konkrétně v tomto případě tvoří automat syntaktický analyzátor překladače.

### 2.1 DEFINICE

Automaty s hlubokými zásobníky (DTPD) vycházejí z definice “klasických” zásobníkových automatů. DTPD je uspořádaná sedmice symbolů  $M = \{Q, \Sigma, \Gamma, R, s, S, F\}$ , kde význam jednotlivých symbolů je po řadě následující:  $Q$  – konečná množina stavů

automatu;  $\Sigma$  – vstupní abeceda;  $\Gamma$  – zásobníková abeceda;  $R$  – množina relací přechodů mezi stavy na základě vstupního symbolu a platí  $R \subseteq (I \times Q \times \Gamma - (\Sigma \cup \{\#\}) \times Q \times (\Gamma - \{\#\})^+) \cup (I \times Q \times \{\#\} \times Q \times (\Gamma - \{\#\})^* \{\#\})$ , kde  $\#$  je speciální symbol z množiny  $\Gamma$  označující dno zásobníku,  $I$  značí celá kladná čísla;  $s \in Q$  – počáteční stav automatu;  $S \in \Gamma$  – startovací symbol zásobníku;  $F$  – konečná množina koncových stavů automatu

## 2.2 SCHÉMA



Obr. 1: Schéma zásobníkového automatu

## 2.3 POJMY

- relace přechodu  $R \subseteq (mpAqv)$  jinak též značeno  $mqA \rightarrow pv \in R$  určuje přechod hloubky  $m$  ze stavu  $p$  do stavu  $q$ , kde  $A$  expanduje na řetězec  $v$
- konfigurace automatu  $Q \times \Sigma^* \times (\Gamma - \{\#\})^* \{\#\}$  - např.  $(p, aabbc, Aaa)\#$  - popisuje, v jakém je automat stavu, co má na vstupní pásce a jaký je obsah zásobníku
- odebrání symbolu z vrcholu zásobníku  $x_p \Rightarrow y$ , kde  $x = (p, aw, az)$ ,  $y = (q, w, z)$  a platí  $p, q \in Q$ ,  $a \in \Sigma$ ,  $z \in \Gamma^*$ ,  $w \in \Sigma^*$  je krok, při kterém se přečte symbol  $a$  ze vstupní pásky a je-li souhlasný se symbolem na vrcholu zásobníku, což v našem případě je, tak se symbol z vrcholu odebere
- expanze symbolu na zásobníku  $x_c \Rightarrow y$ , kde  $x = (p, w, A)$ ,  $y = (q, w, z)$ , kde  $mpA \rightarrow qz \in R$  a platí  $p, q \in Q$ ,  $z \in \Gamma^*$ ,  $w \in \Sigma^*$ ,  $A \in \Gamma$  je krok, kdy je přepsán symbol  $A$  na vrcholu zásobníku řetězcem  $z$

## 2.4 VLASTNOSTI DTPD

Jednou z klíčových nových vlastností je expanze symbolu v hloubce větší než 1. Hloubkou je míněn stupeň zanoření od vrcholu zásobníku. Doposud bylo možné odebírat či přepisovat symboly pouze na vrcholu zásobníku, což bylo jednoduché pro implementaci, ale svým způsobem omezující. Zavádíme proto definici přechodu  $mpA \rightarrow qz$ , která se liší od původní  $pA \rightarrow qz$  u klasických zásobníkových automatů a značí přechod  $m$ -té hloubky, kde  $m$  je kladné celé číslo. Jestliže  $m \in I$  je kladné celé číslo tak, že automat provádí kroky hloubky  $m$  nebo nižší, říkáme, že je automat hloubky  $m$ , značíme  $_mM$ . Hluboko-zásobníkové automaty

jsou vzhledem k hloubce expanze deterministické. To platí, jestliže pro každý stav  $q \in Q$  je kardinalita množiny hloubek přechodů (počet prvků dané množiny) menší nebo rovna 1. To znamená, že automat může z jednoho stavu vykonávat pouze kroky stejné hloubky.

Dále definujeme 2 typy jazyků přijímaných hluboko-zásobníkovými automaty. Jazyky přijímané vyprázdněním zásobníku  $E(nM) = \{w \in \Sigma^* \mid (s, w, S) \Rightarrow^* (q, \varepsilon, \#) \vee nM, \text{ kde } q \in Q\}$  a jazyky přijímané koncovým stavem  $L(nM) = \{w \in \Sigma^* \mid (s, w, S) \Rightarrow^* (f, \varepsilon, \#) \vee nM, \text{ kde } f \in F\}$ ,  $nM$  je hluboko-zásobníkový automat hloubky  $n$ . U klasických zásobníkových automatů jsme byli zvyklí, že byly omezeny na třídu bezkontextových jazyků, tj. bylo možné sestavit automat pouze pro gramatiku generující bezkontextový jazyk. DTPD tomuto omezení přímo nepodléhají. Jsou schopny přijímat nejen bezkontextové jazyky, ale i jazyky, které jsou nad touto třídou. Konkrétně lze dokázat, že spadají do třídy  $CE = CF \cup (CS - CF)$ , kde CS značí kontextové jazyky a CF bezkontextové. Níže uvedený příklad ukazuje, že automat je schopen přijmout jazyk  $L = \{a^n b^n c^n, n \geq 1\}$ , respektive řetězec  $aabbcc$ , jež náleží  $L$ .

## 2.5 PŘÍKLAD

Předpokládejme DTDP,  ${}_2M = (\{s, q, p\}, \{a, b, c\}, \{A, S, \#\}, R, s, S, \{f\})$ , s následující množinou přechodů:  $\{1sS \rightarrow qAA, 1qA \rightarrow paAb, 1qA \rightarrow pab, 2pA \rightarrow qAc, 2pA \rightarrow fc\}$ . Automat je deterministický vzhledem k hloubce. Postup přijetí řetězce  $aabbcc$ :

$$\begin{aligned} (s, aabbcc, S) &\xrightarrow{\varepsilon} (q, aabbcc, AA) [1sS \rightarrow qAA] \xrightarrow{\varepsilon} (p, aabbcc, aAbA) [1qA \rightarrow paAb] \\ p &\xrightarrow{\varepsilon} (p, abbcc, AbA) \xrightarrow{\varepsilon} (q, abbcc, AbAc) [2pA \rightarrow qAc] \xrightarrow{\varepsilon} (q, abbcc, abbAc) [1qA \rightarrow pab] \\ p &\xrightarrow{\varepsilon} (p, bbcc, bbAc) \xrightarrow{p} (p, bcc, bAc) \xrightarrow{p} (p, cc, Ac) \xrightarrow{\varepsilon} (f, cc, cc) [2pA \rightarrow fc] \xrightarrow{p} (f, c, c) \\ p &\xrightarrow{\varepsilon} (f, \varepsilon, \varepsilon) \end{aligned}$$

Řetězec je automatem přijat,  ${}_2M$  končí v konečném stavu s prázdným zásobníkem. Na tomto příkladě je zřejmé ono rozšíření, jelikož o jazyku  $a^n b^n c^n$  je známo, že není bezkontextový.

## 3 ZÁVĚR

DTDP automaty jsou velkým objevem a dobrým předmětem k dalšímu zkoumání jejich činnosti a možností, ale nesou s sebou i komplikace. Jejich důležitou rolí je implementace při syntaktické analýze shora-dolů a právě zde se objevuje první problém. Není zcela triviální navrhnout syntaktický analyzátor s LL tabulkou, jelikož vzhledem k tomu, že automat je schopen expandovat symboly hlouběji v zásobníku, je třeba brát tuto skutečnost v úvahu a ošetřovat přepisy správných symbolů ve správných hloubkách bez hlášení syntaktických chyb. Ale protože je automat deterministický, nemůže nastat, že by se automat mohl rozhodovat, zda přepíše symbol na vrcholu či někde uvnitř zásobníku. Tato důležitá vlastnost situaci zjednodušuje a umožňuje použít algoritmický proces bez používání tzv. "inteligentního výběru".

## LITERATURA

- [1] Martin, J. C.: Introduction to Languages and the Theory of Computation, McGraw-Hill, New York, 1991
- [2] Meduna, A.: Automata and Languages: Theory and Applications, Springer, London 2000