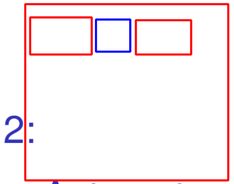
Script generated by TTT

Title: Petter: Compilerbau (03.05.2018)

Date: Thu May 03 14:15:07 CEST 2018

Duration: 85:26 min

Pages: 17



Chapter 2: Basics of Pushdown Automata

Example:

States: $0, 1, 2$
Start state: 0
Final states: $0, 2$

0	a	11
1	a	11
11	b	2
12	b	2

Conventions:

- We do not differentiate between pushdown symbols and states
- The rightmost / upper pushdown symbol represents the state
- Every transition consumes / modifies the upper part of the pushdown

Friedrich Bauer

Klaus Samelson

Definition: Pushdown Automaton

A pushdown automaton (PDA) is a tuple $M = (Q, T, \delta, q_0, F)$ with:

- Q a finite set of states;
- T an input alphabet;
- $q_0 \in Q$ the start state;
- $F \subseteq Q$ the set of final states and
- $\delta \subseteq Q^+ \times (T \cup \{\epsilon\}) \times Q^*$ a finite set of transitions

... for example:

States: $0, 1, 2$
Start state: 0
Final states: $0, 2$

0	a	11
1	a	11
11	b	2
12	b	2

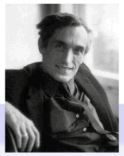
Definition: Deterministic Pushdown Automaton

The pushdown automaton M is deterministic, if every configuration has maximally one successor configuration.

This is exactly the case if for distinct transitions $(\gamma_1, x, \gamma_2)\,,\ (\gamma_1', x', \gamma_2') \in \delta$ we can assume:
Is γ_1 a suffix of γ_1', then $x \neq x' \ \wedge\ x \neq \epsilon \neq x'$ is valid.

Pushdown Automata

CYK




M. Schützenberger A. Öttinger

Theorem:

For each context free grammar $G = (N, T, P, S)$ a pushdown automaton M with $\mathcal{L}(G) = \mathcal{L}(M)$ can be built.

The theorem is so important for us, that we take a look at two constructions for automata, motivated by both of the special derivations:

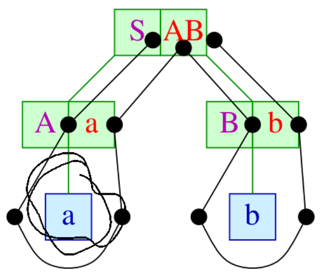
- M_G^L to build Leftmost derivations
- M_G^R to build reverse Rightmost derivations

Chapter 3: Top-down Parsing

Item Pushdown Automaton – Example

Our example:

$$S \rightarrow A\,B \qquad A \rightarrow a \qquad B \rightarrow b$$



Item Pushdown Automaton – Example

We add another rule $S' \rightarrow S\,\$$ for initialising the construction:

Start state: $[S' \rightarrow \bullet\ S\ \$]$
End state: $[S' \rightarrow S\ \bullet\ \$]$
Transition relations:

$[S' \rightarrow \bullet\ S\ \$]$	ϵ	$[S' \rightarrow \bullet\ S\ \$]\,[S \rightarrow \bullet\ A\,B]$
$[S \rightarrow \bullet\ A\,B]$	ϵ	$[S \rightarrow \bullet\ A\,B]\,[A \rightarrow \bullet\ a]$
$[A \rightarrow \bullet\ a]$	a	$[A \rightarrow a\ \bullet]$
$[S \rightarrow \bullet\ A\,B]\,[A \rightarrow a\ \bullet]$	ϵ	$[S \rightarrow A\ \bullet\ B]$
$[S \rightarrow A\ \bullet\ B]$	ϵ	$[S \rightarrow A\ \bullet\ B]\,[B \rightarrow \bullet\ b]$
$[B \rightarrow \bullet\ b]$	b	$[B \rightarrow b\ \bullet]$
$[S \rightarrow A\ \bullet\ B]\,[B \rightarrow b\ \bullet]$	ϵ	$[S \rightarrow A\,B\ \bullet]$
$[S' \rightarrow \bullet\ S\ \$]\,[S \rightarrow A\,B\ \bullet]$	ϵ	$[S' \rightarrow S\ \bullet\ \$]$

Item Pushdown Automaton

The item pushdown automaton M_G^L has three kinds of transitions:

Expansions: $([A \rightarrow \alpha \bullet B\,\beta], \epsilon, [A \rightarrow \alpha \bullet B\,\beta]\,[B \rightarrow \bullet\,\gamma])$ for $A \rightarrow \alpha\,B\,\beta,\ B \rightarrow \gamma\ \in\ P$

Shifts: $([A \rightarrow \alpha \bullet a\,\beta], a, [A \rightarrow \alpha\,a \bullet \beta])$ for $A \rightarrow \alpha\,a\,\beta\ \in\ P$

Reduces: $([A \rightarrow \alpha \bullet B\,\beta]\,[B \rightarrow \gamma \bullet], \epsilon, [A \rightarrow \alpha\,B \bullet \beta])$ for $A \rightarrow \alpha\,B\,\beta,\ B \rightarrow \gamma\ \in\ P$

Items of the form: $[A \rightarrow \alpha\,\bullet]$ are also called complete
The item pushdown automaton shifts the bullet around the derivation tree ...

Item Pushdown Automaton

Discussion:

- The expansions of a computation form a leftmost derivation
- Unfortunately, the expansions are chosen nondeterministically
- For proving correctness of the construction, we show that for every Item $[A \rightarrow \alpha \bullet B\,\beta]$ the following holds:

$$([A \rightarrow \alpha \bullet B\,\beta],\ w) \vdash^* ([A \rightarrow \alpha\,B \bullet \beta],\ \epsilon) \qquad \text{iff} \qquad B \rightarrow^* w$$

- LL-Parsing is based on the item pushdown automaton and tries to make the expansions deterministic ...

Item Pushdown Automaton

Example: $S' \to S\,\$ \qquad S \to \epsilon \mid a\,S\,b$

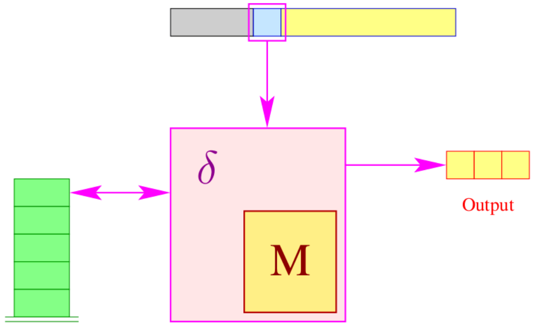
The transitions of the according Item Pushdown Automaton:

0	$[S' \to \bullet\, S\,\$]$	ϵ	$[S' \to \bullet\, S\,\$]\,[S \to \bullet]$
1	$[S' \to \bullet\, S\,\$]$	ϵ	$[S' \to \bullet\, S\,\$]\,[S \to \bullet\, a\,S\,b]$
2	$[S \to \bullet\, a\,S\,b]$	a	$[S \to a \bullet S\,b]$
3	$[S \to a \bullet S\,b]$	ϵ	$[S \to a \bullet S\,b]\,[S \to \bullet]$
4	$[S \to a \bullet S\,b]$	ϵ	$[S \to a \bullet S\,b]\,[S \to \bullet\, a\,S\,b]$
5	$[S \to a \bullet S\,b]\,[S \to \bullet]$	ϵ	$[S \to a\,S \bullet b]$
6	$[S \to a \bullet S\,b]\,[S \to a\,S\,b \bullet]$	ϵ	$[S \to a\,S \bullet b]$
7	$[S \to a\,S \bullet b]$	b	$[S \to a\,S\,b \bullet]$
8	$[S' \to \bullet\, S\,\$]\,[S \to \bullet]$	ϵ	$[S' \to S \bullet\, \$]$
9	$[S' \to \bullet\, S\,\$]\,[S \to a\,S\,b \bullet]$	ϵ	$[S' \to S \bullet\, \$]$

Topdown Parsing

Problem:
Conflicts between the transitions prohibit an implementation of the item pushdown automaton as deterministic pushdown automaton.

Structure of the $LL(1)$-Parser:



- The parser accesses a frame of length 1 of the input;
- it corresponds to an item pushdown automaton, essentially;
- table $M[q, w]$ contains the rule of choice.

Topdown Parsing

Idea:
- Emanate from the item pushdown automaton
- Consider the next input symbol to determine the appropriate rule for the next expansion
- A grammar is called $LL(1)$ if a unique choice is always possible


Philip Lewis


Richard Stearns

Definition:
A reduced grammar is called $LL(1)$, if for each two distinct rules $A \to \alpha\,,\ \ A \to \alpha' \ \in\ P$ and each derivation $S \to_L^* u\,A\,\beta$ with $u \in T^*$ the following is valid:

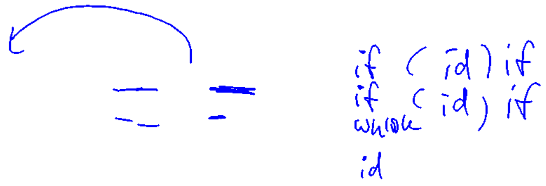
$$\text{First}_1(\alpha\,\beta) \ \cap\ \text{First}_1(\alpha'\,\beta) = \emptyset$$

Topdown Parsing

Example 1:

$$
\begin{array}{lll}
S & \to & \text{if} \ (\ E\ )\ S\ \text{else}\ S \quad | \\
  &   & \text{while} \ (\ E\ )\ S \quad | \\
  &   & E\ ; \\
E & \to & \text{id}
\end{array}
$$

is $LL(1)$, since $\text{First}_1(E) = \{\text{id}\}$



Lookahead Sets

Definition: First_1-Sets

For a set $L \subseteq T^*$ we define:

$$\text{First}_1(L) \;=\; \{\epsilon \mid \epsilon \in L\} \cup \{u \in T \mid \exists v \in T^* : \; uv \in L\}$$

Example: $S \to \epsilon \quad | \quad a\,S\,b$

$\text{First}_1(S)$
ϵ
$a\,b$
$a\,a\,b\,b$
$a\,a\,a\,b\,b\,b$
$\dots$