Script generated by TTT

Title: Simon: Compilerbau (06.05.2013)

Date: Mon May 06 14:25:55 CEST 2013

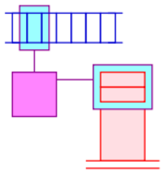
Duration: 83:18 min

Pages: 47

Chapter 2:
Basics of Pushdown Automata

Basics of Pushdown Automata

Languages, specified by context free grammars are accepted by Pushdown Automata:



The pushdown is used e.g. to verify correct nesting of braces.

Example:

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

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

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

Pushdown Automata


Friedrich Bauer


Klaus Samelson

Definition:

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

Pushdown Automata


Friedrich Bauer


Klaus Samelson

Definition:

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

We define computations of pushdown automata with the help of transitions; a particular computation state (the current configuration) is a pair:

$$(\gamma, w) \in Q^* \times T^*$$

consisting of the pushdown content and the remaining input.

... 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

... 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

$$(0, \quad a\,a\,a\,b\,b\,b) \vdash (1\,1, \quad a\,a\,b\,b\,b)$$

... 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

$$\begin{aligned}(0, \quad a\,a\,a\,b\,b\,b) &\vdash (1\,1, \quad a\,a\,b\,b\,b)\\ &\vdash (1\,1\,1, \quad a\,b\,b\,b)\end{aligned}$$

... 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

$$\begin{aligned}(0, \quad a\,a\,a\,b\,b\,b) &\vdash (1\,1, \quad a\,a\,b\,b\,b)\\ &\vdash (1\,1\,1, \quad a\,b\,b\,b)\\ &\vdash (1\,1\,1\,1, \quad b\,b\,b)\end{aligned}$$

... 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

$$\begin{aligned}(0, \quad a\,a\,a\,b\,b\,b) &\vdash (1\,1, \quad a\,a\,b\,b\,b)\\ &\vdash (1\,1\,1, \quad a\,b\,b\,b)\\ &\vdash (1\,1\,1\,1, \quad b\,b\,b)\\ &\vdash (1\,1\,2, \quad b\,b)\end{aligned}$$

... 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

$$\begin{array}{rrl} (0, \; aaabbb) & \vdash & (1\,1, \; aabbb) \\ & \vdash & (1\,1\,1, \; abbb) \\ & \vdash & (1\,1\,1\,1, \; bbb) \\ & \vdash & (1\,1\,2, \; bb) \\ & \vdash & (1\,2, \; b) \end{array}$$

A computation step is characterized by the relation
$\vdash \subseteq (Q^* \times T^*)^2$ with

$$(\alpha\,\gamma, x\,w) \vdash (\alpha\,\gamma', w) \quad \text{for} \quad (\gamma, x, \gamma') \in \delta$$

A computation step is characterized by the relation
$\vdash \subseteq (Q^* \times T^*)^2$ with

$$(\alpha\,\gamma, x\,w) \vdash (\alpha\,\gamma', w) \quad \text{for} \quad (\gamma, x, \gamma') \in \delta$$

Remarks:

- The relation $\vdash$ depends of the pushdown automaton M
- The reflexive and transitive closure of $\vdash$ is called $\vdash^*$
- Then, the language, accepted by M, is

$$\mathcal{L}(M) = \{w \in T^* \mid \exists f \in F : (q_0, w) \vdash^* (f, \epsilon)\}$$

A computation step is characterized by the relation
$\vdash \subseteq (Q^* \times T^*)^2$ with

$$(\alpha\,\gamma, x\,w) \vdash (\alpha\,\gamma', w) \quad \text{for} \quad (\gamma, x, \gamma') \in \delta$$

Remarks:

- The relation $\vdash$ depends of the pushdown automaton M
- The reflexive and transitive closure of $\vdash$ is called $\vdash^*$
- Then, the language, accepted by M, is

$$\mathcal{L}(M) = \{w \in T^* \mid \exists f \in F : (q_0, w) \vdash^* (f, \epsilon)\}$$

We accept with a final state together with empty input.

Deterministic Pushdown Automaton

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

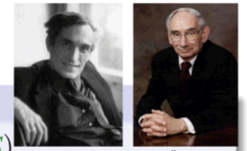
This is exactly the case if for distinct transitions $(\gamma_1, x, \gamma_2)\,,\ (\gamma_1', x', \gamma_2') \in \delta$ we can assume:
Is γ_1 a suffix of γ_1', then $x \neq x' \ \wedge\ x \neq \epsilon \neq x'$ is valid.

... for example:

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

... this obviously holds

Pushdown Automata


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

Construction: Item Pushdown Automaton M_G^L

- Reconstruct a Leftmost derivation.
- Expand nonterminals using a rule.
- Verify successively, that the chosen rule matches the input.

$\Longrightarrow$ The states are now Items (= rules with a dot):

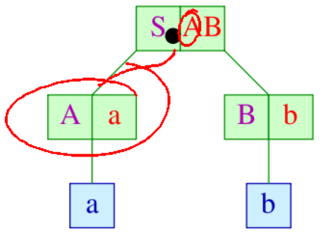
$$[A \to \alpha \bullet \beta]\,, \qquad A \to \alpha\,\beta \ \in\ P$$

The dot marks the spot, how far the rule is already processed

Item Pushdown Automaton – Example

Our example:

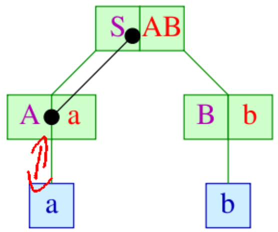
$S \rightarrow AB \quad A \rightarrow a \quad B \rightarrow b$



Item Pushdown Automaton – Example

Our example:

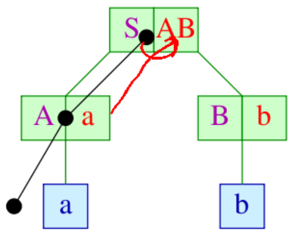
$S \rightarrow AB \quad A \rightarrow a \quad B \rightarrow b$



Item Pushdown Automaton – Example

Our example:

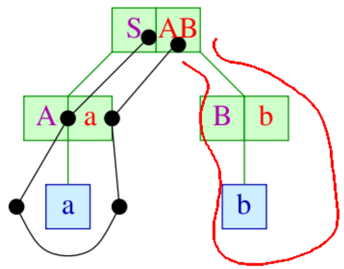
$S \rightarrow AB \quad A \rightarrow a \quad B \rightarrow b$



Item Pushdown Automaton – Example

Our example:

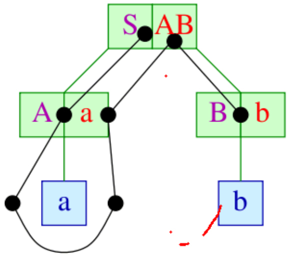
$S \rightarrow AB \quad A \rightarrow a \quad B \rightarrow b$



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:

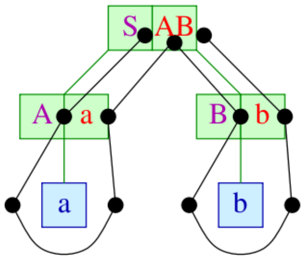
Start state: $[S' \rightarrow \bullet\ S]$
End state: $[S' \rightarrow S \bullet]$
Transition relations:

$[S' \rightarrow \bullet\ S]$	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 – 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: $([A \to \alpha \bullet B\,\beta], \epsilon, [A \to \alpha \bullet B\,\beta]\,[B \to \bullet\,\gamma])$ for $A \to \alpha\,B\,\beta,\ B \to \gamma \in P$

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

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

Items of the form: $[A \to \alpha \bullet]$ are also called complete

The item pushdown automaton shifts the dot once 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 $[A \to \alpha \bullet B\,\beta]$ the following holds:

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

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

Item Pushdown Automaton

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

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

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

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

Items of the form: $[A \to \alpha \bullet]$ are also called complete

The item pushdown automaton shifts the dot once around the derivation tree ...

Item Pushdown Automaton

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

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

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

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

Items of the form: $[A \to \alpha \bullet]$ are also called complete

The item pushdown automaton shifts the dot once around the derivation tree ...

Item Pushdown Automaton

Beispiel: $S \to \epsilon \mid a\,S\,b$

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	$[S' \to S \bullet]$

Conflicts arise between the transitions $(0,1)$ and $(3,4)$, resp..

Topdown Parsing

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

Topdown Parsing

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

Idee 1: GLL Parsing
For each conflict, we create a virtual copy of the complete stack and continue computing in parallel.

Topdown Parsing

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

Idee 1: GLL Parsing
For each conflict, we create a virtual copy of the complete stack and continue computing in parallel.

Idee 2: Recursive Descent & Backtracking
Depth-first search for an appropriate solution.

Topdown Parsing

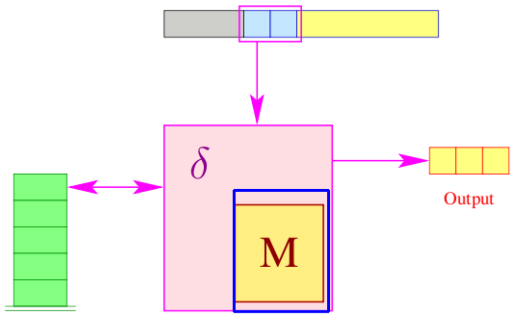
Problem:
Conflicts between the transitions prohibit an implementation of the item pushdown automaton as deterministic pushdown automaton.

Idee 1: GLL Parsing
For each conflict, we create a virtual copy of the complete stack and continue computing in parallel.

Idee 2: Recursive Descent & Backtracking
Depth-first search for an appropriate solution.

Idee 3: Recursive Descent & Lookahead
Conflicts are resolved by considering a lookup of the next input symbol.

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

Idee:

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

Topdown Parsing

Idee:

- Emanate from the item pushdown automaton
- Consider the next 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:

$$S \to \text{if } (E)\ S \text{ else } S \mid \text{while } (E)\ S \mid E;$$
$$E \to \text{id}$$

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

Topdown Parsing

Example 1:

$$S \to \text{if } (E)\ S \text{ else } S \mid \text{while } (E)\ S \mid E;$$
$$E \to \text{id}$$

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

Example 2:

$$S \to \text{if } (E)\ S \text{ else } S \mid \text{if } (E)\ S \mid \text{while } (E)\ S \mid E;$$
$$E \to \text{id}$$

... is not $LL(k)$ for any $k > 0$.

Lookahead Sets

Definition:
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:

ϵ
$a\,b$
$a\,a\,b\,b$
$a\,a\,a\,b\,b\,b$

Lookahead Sets

Definition:
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:

ϵ
$a\,b$
$a\,a\,b\,b$
$a\,a\,a\,b\,b\,b$

the prefixes of length 1

Lookahead Sets

Arithmetics:
$\text{First}_1(\_)$ is compatible with union and concatenation:

$$\begin{array}{lll} \text{First}_1(\emptyset) & = & \emptyset \\ \text{First}_1(L_1 \cup L_2) & = & \text{First}_1(L_1) \cup \text{First}_1(L_2) \\ \text{First}_1(L_1 \cdot L_2) & = & \text{First}_1(\text{First}_1(L_1) \cdot \text{First}_1(L_2)) \\ & := & \text{First}_1(L_1) \odot \text{First}_1(L_2) \end{array}$$

$1-$ concatenation

Observation:
Let $L_1, L_2 \subseteq T \cup \{\epsilon\}$ with $L_1 \neq \emptyset \neq L_2$. Then:

$$L_1 \odot L_2 = \begin{cases} L_1 & \text{if } \epsilon \notin L_1 \\ (L_1 \backslash \{\epsilon\}) \cup L_2 & \text{otherwise} \end{cases}$$

If all rules of G are productive, then all sets $\text{First}_1(A)$ are non-empty.

Lookahead Sets

For $\alpha \in (N \cup T)^*$ we are interested in the set:

$$\text{First}_1(\alpha) = \text{First}_1(\{w \in T^* \mid \alpha \to^* w\})$$

Idea: Treat ϵ separately: F_ϵ

- Let $\text{empty}(X) = \text{true}$ iff $X \to^* \epsilon$.
- $F_\epsilon(X_1 \ldots X_m) = \bigcup_{i=1}^{j} F_\epsilon(X_i)$ if $\text{empty}(X_1) \wedge \ldots \wedge \text{empty}(X_{j-1})$

Lookahead Sets

For $\alpha \in (N \cup T)^*$ we are interested in the set:

$$\text{First}_1(\alpha) = \text{First}_1(\{w \in T^* \mid \alpha \to^* w\})$$

Idea: Treat ϵ separately: F_ϵ

- Let $\text{empty}(X) = \text{true}$ iff $X \to^* \epsilon$.
- $F_\epsilon(X_1 \ldots X_m) = \bigcup_{i=1}^{j} F_\epsilon(X_i)$ if $\text{empty}(X_1) \wedge \ldots \wedge \text{empty}(X_{j-1})$

We characterize the ϵ-free First_1-sets with an inequality system:

$$\begin{array}{llll} F_\epsilon(a) & = & \{a\} & \text{if } a \in T \\ F_\epsilon(A) & \supseteq & F_\epsilon(X_j) & \text{if } A \to X_1 \ldots X_m \in P, \\ & & & \text{empty}(X_1) \wedge \ldots \wedge \text{empty}(X_{j-1}) \end{array}$$

Lookahead Sets

for example...

$$\begin{array}{lllll} E & \to & E+T & \mid & T \\ T & \to & T*F & \mid & F \\ F & \to & (\,E\,) & \mid & \text{name} \mid \text{int} \end{array}$$

with $\text{empty}(E) = \text{empty}(T) = \text{empty}(F) = \text{false}$