

Script generated by TTT

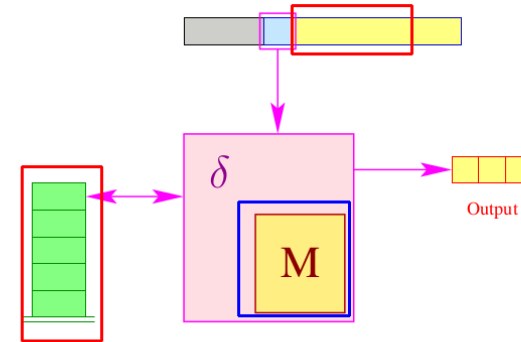
Title: Petter: Compilerbau (09.05.2016)

Date: Mon May 09 14:26:58 CEST 2016

Duration: 90:07 min

Pages: 31

## 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.

91/282

## 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

## 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

### Definition:

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

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



Philip Lewis



Richard Stearns

92/282

92/282

## Topdown Parsing

### Example 1:

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

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

93/282

## Topdown Parsing

### Example 1:

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

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

### Example 2:

$S \rightarrow \text{if } ( E ) S \text{ else } S \mid$   
 $\quad \text{if } ( E ) S \mid$   
 $\quad \text{while } ( E ) S \mid$   
 $E \rightarrow \text{id}$

$S$   
 $\text{if } ( \text{id} ) \text{if } ( \text{id} ) \text{else}$   
 ... is not  $LL(k)$  for any  $k > 0$ .

93/282

## Lookahead Sets

### Definition: $\text{First}_1$ -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:  $S \rightarrow \epsilon \mid aSb$

$\text{First}_1(S)$
$\epsilon$
$a$
$aabb$
$aaabbb$
...

94/282

## Lookahead Sets

### Arithmetics:

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

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

$\odot$  being 1 – concatenation

95/282

## Lookahead Sets

### Arithmetics:

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

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

$\odot$  being 1 – concatenation

### Definition: 1-concatenation

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

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

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

95/282

## Lookahead Sets

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

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

Idea: Treat  $\epsilon$  separately:  $\text{First}_1(A) = F_\epsilon(A) \cup \{\epsilon \mid A \rightarrow^* \epsilon\}$

- Let  $\text{empty}(X) = \text{true}$  iff  $X \rightarrow^* \epsilon$ .
- $F_\epsilon(X_1 \dots X_m) = \bigcup_{i=1}^j F_\epsilon(X_i)$  if  $\bigwedge_{i=1}^{j-1} \text{empty}(X_i)$



96/282

## Lookahead Sets

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

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

Idea: Treat  $\epsilon$  separately:  $\text{First}_1(A) = F_\epsilon(A) \cup \{\epsilon \mid A \rightarrow^* \epsilon\}$

- Let  $\text{empty}(X) = \text{true}$  iff  $X \rightarrow^* \epsilon$ .
- $F_\epsilon(X_1 \dots X_m) = \bigcup_{i=1}^j F_\epsilon(X_i)$  if  $\bigwedge_{i=1}^{j-1} \text{empty}(X_i)$

We characterize the  $\epsilon$ -free  $\text{First}_1$ -sets with an inequality system:

$$\begin{aligned} F_\epsilon(a) &= \{a\} & \text{if } a \in T \\ F_\epsilon(A) &\supseteq F_\epsilon(X_j) & \text{if } A \rightarrow X_1 \dots X_m \in P, \\ & & \bigwedge_{i=1}^{j-1} \text{empty}(X_i) \end{aligned}$$

96/282

## Lookahead Sets

for example...

$$\begin{array}{lcl} E & \rightarrow & E+T \quad | \quad T \\ T & \rightarrow & T * F \quad | \quad F \\ F & \rightarrow & (E) \quad | \quad \text{name} \quad | \quad \text{int} \end{array}$$

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

... we obtain:

$$\begin{array}{lcl} F_\epsilon(S') & \supseteq & F_\epsilon(E) \quad F_\epsilon(E) \supseteq F_\epsilon(E) \\ F_\epsilon(E) & \supseteq & F_\epsilon(T) \quad F_\epsilon(T) \supseteq F_\epsilon(T) \\ F_\epsilon(T) & \supseteq & F_\epsilon(F) \quad F_\epsilon(F) \supseteq \{(\text{ , name, int})\} \end{array}$$

97/282

## Fast Computation of Lookahead Sets

### Observation:

- The form of each inequality of these systems is:

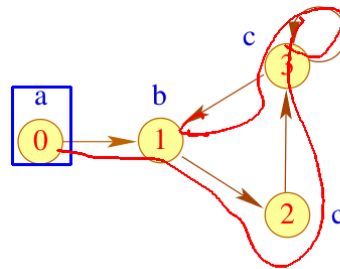
$$x \supseteq y \quad \text{resp.} \quad x \supseteq d$$

for variables  $x, y$  und  $d \in D$ .

- Such systems are called **pure unification problems**
- Such problems can be solved in **linear space/time**.

for example:  $D = 2^{\{a,b,c\}}$

$$\begin{array}{l} x_0 \supseteq \{a\} \\ x_1 \supseteq \{b\} \\ x_2 \supseteq \{c\} \\ x_3 \supseteq \{c\} \end{array} \quad \begin{array}{l} x_1 \supseteq x_0 \\ x_2 \supseteq x_1 \\ x_3 \supseteq x_2 \end{array} \quad \begin{array}{l} x_1 \supseteq x_3 \\ x_2 \supseteq x_1 \\ x_3 \supseteq x_3 \end{array}$$

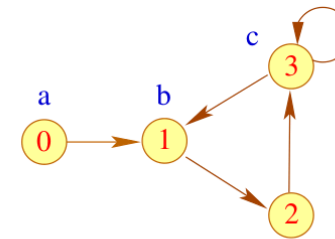


98 / 282

## Fast Computation of Lookahead Sets



Frank DeRemer & Tom Pennello



### Proceeding:

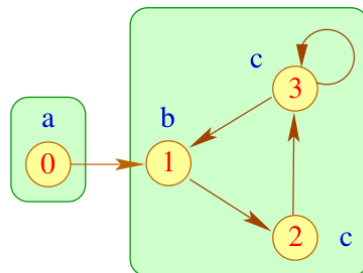
- Create the **Variable Dependency Graph** for the inequality system.

99 / 282

## Fast Computation of Lookahead Sets



Frank DeRemer & Tom Pennello



### Proceeding:

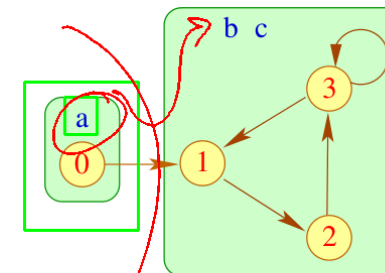
- Create the **Variable Dependency Graph** for the inequality system.
- Whithin a **Strongly Connected Component** ( $\rightarrow$  Tarjan) all variables have the same value

99 / 282

## Fast Computation of Lookahead Sets



Frank DeRemer & Tom Pennello



### Proceeding:

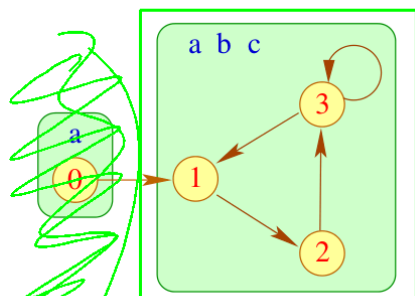
- Create the **Variable Dependency Graph** for the inequality system.
- Whithin a **Strongly Connected Component** ( $\rightarrow$  Tarjan) all variables have the same value
- Is there no ingoing edge for an SCC, its value is computed via the smallest upper bound of all values within the SCC

99 / 282

## Fast Computation of Lookahead Sets



Frank DeRemer & Tom Pennello



Proceeding:

- Create the **Variable Dependency Graph** for the inequality system.
- Whithin a **Strongly Connected Component** ( $\rightarrow$  Tarjan) all variables have the same value
- Is there no ingoing edge for an SCC, its value is computed via the smallest upper bound of all values within the SCC
- In case of ingoing edges, their values are also to be considered for the upper bound

99/282

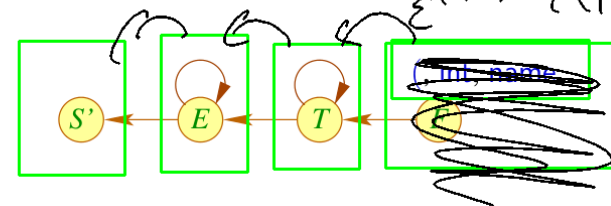
## Fast Computation of Lookahead Sets

... for our example grammar:

$$F_{\epsilon}(T) = \{(\text{int}, \text{name})\}$$

$$F_{\epsilon}(F) = \{(\text{int}, \text{name})\}$$

First<sub>1</sub>:



100/282

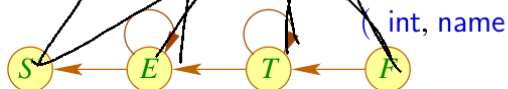
## Fast Computation of Lookahead Sets

$$\text{First}_1(\ ) = F_{\epsilon}(\ )$$

... for our example grammar:

$$F_{\epsilon}(F) = \{(\text{int}, \text{name})\}$$

First<sub>1</sub>:



100/282

## Item Pushdown Automaton as LL(1)-Parser

back to the example:  $S \rightarrow \epsilon \mid a S b$

The transitions in the according Item Pushdown Automaton:

0	$[S' \rightarrow \bullet S]$	$\epsilon$	$[S' \rightarrow \bullet S] [S \rightarrow \bullet]$
1	$[S' \rightarrow \bullet S]$	$\epsilon$	$[S' \rightarrow \bullet S] [S \rightarrow \bullet a S b]$
2	$[S \rightarrow \bullet a S b]$	$a$	$[S \rightarrow a \bullet S b]$
3	$[S \rightarrow a \bullet S b]$	$\epsilon$	$[S \rightarrow a \bullet S b] [S \rightarrow \bullet]$
4	$[S \rightarrow a \bullet S b]$	$\epsilon$	$[S \rightarrow a \bullet S b] [S \rightarrow \bullet a S b]$
5	$[S \rightarrow a \bullet S b] [S \rightarrow \bullet]$	$\epsilon$	$[S \rightarrow a S \bullet]$
6	$[S \rightarrow a \bullet S b] [S \rightarrow a S b \bullet]$	$\epsilon$	$[S \rightarrow a S \bullet]$
7	$[S \rightarrow a S \bullet]$	$b$	$[S \rightarrow a S b \bullet]$
8	$[S' \rightarrow \bullet S] [S \rightarrow \bullet]$	$\epsilon$	$[S' \rightarrow S \bullet]$
9	$[S' \rightarrow \bullet S] [S \rightarrow a S b \bullet]$	$\epsilon$	$[S' \rightarrow S \bullet]$

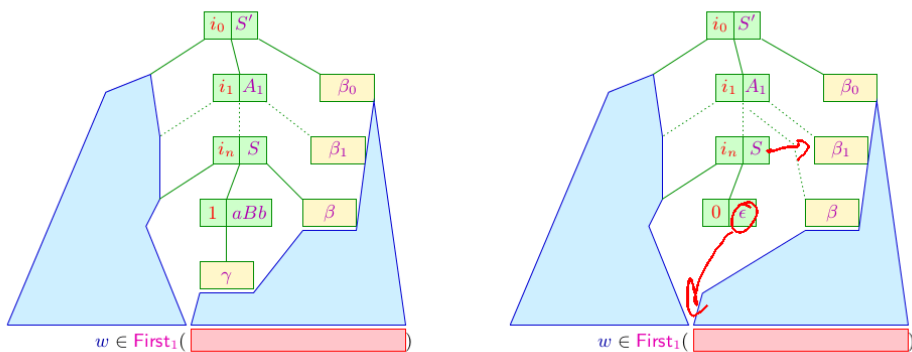
Conflicts arise between transitions (0, 1) or (3, 4) resp..

101/282

## Item Pushdown Automaton as LL(1)-Parser

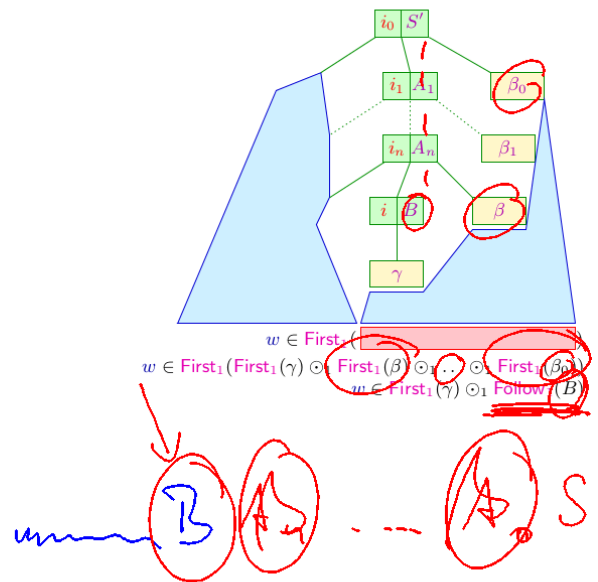
... in detail:  $S \rightarrow \epsilon^0 \mid a S b^1$

$\text{First}_1(\text{input})$	$\epsilon$	$a$	$b$
$S$	$?$	$?$	$?$



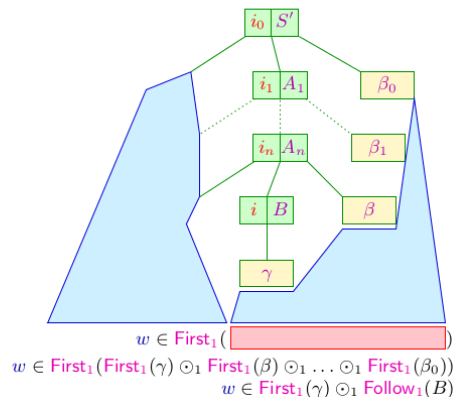
102/282

## Item Pushdown Automaton as LL(1)-Parser



103/282

## Item Pushdown Automaton as LL(1)-Parser



Inequality system for  $\text{Follow}_1(B) = \text{First}_1(\beta) \circ_1 \dots \circ_1 \text{First}_1(\beta_0)$

$$\begin{aligned} \text{Follow}_1(S) &\supseteq \{\epsilon\} \\ \text{Follow}_1(B) &\supseteq F_\epsilon(X_j) \quad \text{if } A \rightarrow \alpha B X_1 \dots X_m \in P, \\ &\quad \text{empty}(X_1) \wedge \dots \wedge \text{empty}(X_{j-1}) \\ \text{Follow}_1(B) &\supseteq \text{Follow}_1(A) \quad \text{if } A \rightarrow \alpha B X_1 \dots X_m \in P, \\ &\quad \text{empty}(X_1) \wedge \dots \wedge \text{empty}(X_m) \end{aligned}$$

104/282

## Item Pushdown Automaton as LL(1)-Parser

Is  $G$  an  $LL(1)$ -grammar, we can index a lookahead-table with items and nonterminals:

### LL(1)-Lookahead Table

We set  $M[B][w] = i$  with  $B \rightarrow \gamma^k$  if  $w \in \text{First}_1(\gamma) \circ_1 \text{Follow}_1(B)$

... for example:  $S \rightarrow \epsilon^0 \mid a S b^1$

104/282

## Item Pushdown Automaton as LL(1)-Parser

Is  $G$  an  $LL(1)$ -grammar, we can index a lookahead-table with items and nonterminals:

### LL(1)-Lookahead Table

We set  $M[B, w] = i$  with  $B \rightarrow \gamma^i$  if  $w \in \text{First}_1(\gamma) \odot_1 \text{Follow}_1(B)$

... for example:  $S \rightarrow \epsilon^0 \mid aSb^1$

$$\text{First}_1(S) = \{\epsilon, a\}$$

104/282

## Item Pushdown Automaton as LL(1)-Parser

Is  $G$  an  $LL(1)$ -grammar, we can index a lookahead-table with items and nonterminals:

### LL(1)-Lookahead Table

We set  $M[B, w] = i$  with  $B \rightarrow \gamma^i$  if  $w \in \text{First}_1(\gamma) \odot_1 \text{Follow}_1(B)$

... for example:  $S \rightarrow \epsilon^0 \mid aSb^1$

$$\text{First}_1(S) = \{\epsilon, a\} \quad \text{Follow}_1(S) = \{b, \epsilon\}$$

104/282

## Item Pushdown Automaton as LL(1)-Parser

Is  $G$  an  $LL(1)$ -grammar, we can index a lookahead-table with items and nonterminals:

### LL(1)-Lookahead Table

We set  $M[B, w] = i$  with  $B \rightarrow \gamma^i$  if  $w \in \text{First}_1(\gamma) \odot_1 \text{Follow}_1(B)$

... for example:  $S \rightarrow \epsilon^0 \mid aSb^1$

$$\text{First}_1(S) = \{\epsilon, a\} \quad \text{Follow}_1(S) = \{b, \epsilon\}$$

$S$ -rule 0:  ~~$\text{First}_1(\epsilon) \odot_1 \text{Follow}_1(S) = \{b, \epsilon\}$~~   
 $S$ -rule 1:  $\text{First}_1(aSb) \odot_1 \text{Follow}_1(S) = \{a\}$

104/282

## Item Pushdown Automaton as LL(1)-Parser

Is  $G$  an  $LL(1)$ -grammar, we can index a lookahead-table with items and nonterminals:

### LL(1)-Lookahead Table

We set  $M[B, w] = i$  with  $B \rightarrow \gamma^i$  if  $w \in \text{First}_1(\gamma) \odot_1 \text{Follow}_1(B)$

... for example:

$S \rightarrow \epsilon^0 \mid aSb^1$

$$\text{First}_1(S) = \{\epsilon, a\} \quad \text{Follow}_1(S) = \{b, \epsilon\}$$

$S$ -rule 0:  $\text{First}_1(\epsilon) \odot_1 \text{Follow}_1(S) = \{b, \epsilon\}$   
 $S$ -rule 1:  $\text{First}_1(aSb) \odot_1 \text{Follow}_1(S) = \{a\}$

	$\epsilon$	$a$	$b$
$S$	0	1	0

104/282

## Item Pushdown Automaton as LL(1)-Parser

For example:  $S \rightarrow \epsilon^0 \mid aSb^1$

The transitions of the according Item Pushdown Automaton:

0	$[S' \rightarrow \bullet S]$	$\epsilon$	$[S' \rightarrow \bullet S] [S \rightarrow \bullet]$
1	$[S' \rightarrow \bullet S]$	$\epsilon$	$[S' \rightarrow \bullet S] [S \rightarrow \bullet aSb]$
2	$[S \rightarrow \bullet aSb]$	$a$	$[S \rightarrow a \bullet Sb]$
3	$[S \rightarrow a \bullet Sb]$	$\epsilon$	$[S \rightarrow a \bullet Sb] [S \rightarrow \bullet]$
4	$[S \rightarrow a \bullet Sb]$	$\epsilon$	$[S \rightarrow a \bullet Sb] [S \rightarrow \bullet aSb]$
5	$[S \rightarrow a \bullet Sb] [S \rightarrow \bullet]$	$\epsilon$	$[S \rightarrow aS \bullet b]$
6	$[S \rightarrow a \bullet Sb] [S \rightarrow aSb \bullet]$	$\epsilon$	$[S \rightarrow aS \bullet b]$
7	$[S \rightarrow aS \bullet b]$	$b$	$[S \rightarrow aSb \bullet]$
8	$[S' \rightarrow \bullet S] [S \rightarrow \bullet]$	$\epsilon$	$[S' \rightarrow S \bullet]$
9	$[S' \rightarrow \bullet S] [S \rightarrow aSb \bullet]$	$\epsilon$	$[S' \rightarrow S \bullet]$

Lookahead table:

	$\epsilon$	$a$	$b$
$S$	0	1	0

105/282

## Topdown-Parsing

### Discussion

- A practical implementation of an  $LL(1)$ -parser via **recursive Descent** is a straight-forward idea
- However, **only a subset** of the deterministic contextfree languages can be parsed this way.

106/282

## Topdown-Parsing

### Discussion

- A practical implementation of an  $LL(1)$ -parser via **recursive Descent** is a straight-forward idea
- However, **only a subset** of the deterministic contextfree languages can be parsed this way.
- **Solution:** Going from  $LL(1)$  to  $LL(k)$
- The size of the occurring sets is rapidly increasing with larger  $k$
- Unfortunately, even  $LL(k)$  parsers are not sufficient to accept all deterministic contextfree languages.
- In practical systems, this often motivates the implementation of  $k = 1$  only ...

106/282

## Syntactic Analysis

## Chapter 4: Bottom-up Analysis

107/282