

Introduction to Language Theory and Compilation Solutions

Session 4: Grammars revisited

Solutions

Ex. 1.

- The unproductive symbol removal algorithm stabilises with $V_i = \{S, B\}$ so we get:

$$G_1 = \langle \{S, B\}, \{a, b\}, \{S \rightarrow a, B \rightarrow b\}, S \rangle$$

- We notice B can't be accessed from S in this new grammar and can thus be removed. We end up with:

$$G' = \langle \{S\}, \{a\}, \{S \rightarrow a\}, S \rangle$$

2. STEP 1: Computational steps for V_i

i	V_i
0	\emptyset
1	$\{C, A\}$
2	$\{C, A, S\}$
3	$\{C, A, S\}$

STEP 3: We can now remove the inaccessible symbols

i	V_i
0	$\{S\}$
1	$\{S, A\}$
2	$\{S, A\}$

STEP 2: We get the following P'

$$\begin{array}{lcl} S & \rightarrow & A \\ A & \rightarrow & bS \\ & & b \\ C & \rightarrow & AS \\ & & b \end{array}$$

STEP 4: We finally obtain $G' = \langle V', P', T', S' \rangle$ where

- $V' = \{S, A\}$
- $P' = \{S \rightarrow A, A \rightarrow bS \mid b\}$
- $T' = \{b\}$
- $S' = S$

Ex. 2.

$$\begin{array}{lcl} E & \rightarrow & E+T \\ & & E-T \\ & & T \\ T & \rightarrow & T * F \\ & & T / F \\ & & F \\ F & \rightarrow & F \Rightarrow G \\ & & G \\ G & \rightarrow & ID \\ & & ID[E] \end{array}$$

Ex. 3.

$\langle \text{stmt} \rangle \rightarrow \text{if } \langle \text{expr} \rangle \text{ then } \langle \text{stmt-list} \rangle \langle \text{if-tail} \rangle$
 $\langle \text{if-tail} \rangle \rightarrow \text{end if}$
 $\langle \text{if-tail} \rangle \rightarrow \text{else } \langle \text{stmt-list} \rangle \text{ end if}$

Ex. 4.

$$\begin{aligned}
E &\rightarrow AB \\
A &\rightarrow T \\
B &\rightarrow +TB \\
&\quad \varepsilon \\
T &\rightarrow CD \\
C &\rightarrow P \\
D &\rightarrow *PD \\
&\quad \varepsilon \\
P &\rightarrow ID
\end{aligned}$$

Ex. 5. Useless symbols

1. **Remove unproductive symbols:** H and C are both unproductive as they are mutually recursive and cannot produce anything useful. Hence we can remove rules $H \rightarrow Ca$, $C \rightarrow Hb$ and $F \rightarrow aHD$.
2. **Remove inaccessible symbols:** by removing rule $F \rightarrow aHD$, D became inaccessible. We can thus remove $D \rightarrow ab$.

Left recursion and factoring

So far, we have:

$$\begin{aligned}
S &\rightarrow aE \mid bF \\
E &\rightarrow bE \mid \varepsilon \\
F &\rightarrow aF \mid aG \\
G &\rightarrow Gc \mid d
\end{aligned}$$

3. **Left recursion removal:** $G \rightarrow Gc$ is left recursive. We replace $G \rightarrow Gc \mid d$ with $G \rightarrow dG'$ and $G' \rightarrow cG' \mid \varepsilon$.
4. **Left factoring:** we replace $F \rightarrow aF \mid aG$ with $F \rightarrow aF'$ and $F' \rightarrow F \mid G$.

Check

We can now check whether our final grammar is LL(1) by building the corresponding action table and verifying that no conflicts arise.

			a	b	c	d	$\$$
(0)	$S' \rightarrow S\$$	S'	(0)	(0)	\times	\times	\times
(1,2)	$S \rightarrow aE \mid bF$	S	(1)	(2)	\times	\times	\times
(3,4)	$E \rightarrow bE \mid \varepsilon$	E	\times	(3)	\times	\times	(4)
(5)	$F \rightarrow aF'$	F	(5)	\times	\times	\times	\times
(6,7)	$F' \rightarrow F \mid G$	F'	(6)	\times	\times	(7)	\times
(8)	$G \rightarrow dG'$	G	\times	\times	\times	(8)	\times
(9,10)	$G' \rightarrow cG' \mid \varepsilon$	G'	\times	\times	(9)	\times	(10)