Introduction to Language Theory and Compilation
Solutions
Session 5: Grammars revisited

Solutions

Ex. 1.

1.
   - The unproductive symbol removal algorithm stabilises with $V_i = \{S, B\}$ so we get:
   $$G_1 = \langle \{S,B\}, \{a,b\}, \{S \to a, B \to b\}, S \rangle$$
   - We notice B can't be accessed from S in this new grammar and can thus be removed. We end up with:
   $$G' = \langle \{S\}, \{a\}, \{S \to a\}, s \rangle$$

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 2: We get the following P'

$$\begin{array}{rcl} S & \to & A \\ A & \to & bS \\ & & b \\ C & \to & AS \\ & & b \end{array}$$

STEP 3: We can now remove the inaccessible symbols

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

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

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

Ex. 2.

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

Ex. 3.

<stmt> → **if** <expr> **then** <stmt-list> <if-tail>
<if-tail> → **end if**
<if-tail> → **else** <stmt-list> **end if**

Ex. 4.

$$\begin{array}{lcl} E & \to & AB \\ A & \to & T \\ B & \to & +TB \\ & & \varepsilon \\ T & \to & CD \\ C & \to & P \\ D & \to & *PD \\ & & \varepsilon \\ P & \to & ID \end{array}$$

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 $H \to Ca$, $C \to Hb$ and $F \to aHD$.
2. **Remove inaccessible symbols**: by removing rule $F \to aHD$, D became inaccessible. We can thus remove $D \to ab$.

Left recursion and factoring

So far, we have:

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

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

Check

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

$$\begin{array}{rlcl} (0) & S' & \to & S\$ \\ (1,2) & S & \to & aE \mid bF \\ (3,4) & E & \to & bE \mid \varepsilon \\ (5) & F & \to & aF' \\ (6,7) & F' & \to & F \mid G \\ (8) & G & \to & dG' \\ (9,10) & G' & \to & cG' \mid \varepsilon \end{array}$$

	a	b	c	d	$\$$
S'	(0)	(0)	×	×	×
S	(1)	(2)	×	×	×
E	×	(3)	×	×	(4)
F	(5)	×	×	×	×
F'	(6)	×	×	(7)	×
G	×	×	×	(8)	×
G'	×	×	(9)	×	(10)