Problem Set for Tutorial 6 - TDDD14/TDDD85

1 Context-Free Grammars

Exercise 1. Consider the CFG $G = (\{E, T, F\}, \{a, b, c, +, -, \cdot, /, (,)\}, P, E)$, where P comprises the productions

$$\begin{split} E &\rightarrow T \mid E + T \mid E - T \\ T &\rightarrow F \mid T \cdot F \mid T/F \\ F &\rightarrow a \mid b \mid c \mid (E) \end{split}$$

Find the derivation trees for the following strings.

1. $a \cdot b + c$ 2. $a + a - b \cdot (a/b + b/c)$

Exercise 2. Find CFGs which generate the following languages.

- 1. All strings in $\{0,1\}^*$ for which every 0 is followed by 1 immediately to the right.
- 2. All strings in $\{0,1\}^*$ which are palindromes.
- 3. $\{0^n 1^n \mid n \ge 0\}$
- 4. All string in $\{a, b\}^*$ containing at least one *a* and one *b*, such that the number of *a*'s preceding the first *b* is the same as the number of *b*'s following the last *a*.

Exercise 3. Consider the CFG $G = (\{S, A, B\}, \{a, b\}, P, S)$, where P comprises the productions

 $\begin{array}{l} S \rightarrow aB \mid bA \\ A \rightarrow a \mid aS \mid bAA \\ B \rightarrow b \mid bS \mid aBB \end{array}$

Show that G is ambiguous.

Exercise 4. Let G be a CFG consisting of the following productions (S is the start symbol):

 $\begin{array}{l} S \rightarrow AB \\ A \rightarrow SA \mid BB \mid bB \\ B \rightarrow b \mid aA \mid \epsilon \end{array}$

Find an equivalent CFG with a single ϵ -production $S \to \epsilon$, and without unit productions.

Exercise 5. Find equivalent Chomsky normal-form CFGs for the two CFGs below (S is the start symbol in both cases).

- 1. $S \to \neg S \mid (S \supset S) \mid p \mid q$

Solutions

Solution to Exercise 1. Our task is to find the derivation trees for the strings $a \cdot b + c$ and $a + a - b \cdot (a/b + b/c)$. We present a detailed solution for the first case and leave the second as an exercise.

A derivation for the string $a \cdot b + c$ is:

 $E \Rightarrow E + T \Rightarrow T + T \Rightarrow T \cdot F + F \Rightarrow F \cdot F + F \Rightarrow a \cdot F + F \Rightarrow a \cdot b + F \Rightarrow a \cdot b + c.$

The corresponding derivation tree is:



Solution to Exercise 2. S is the start symbol in all grammars below.

1.
$$S \Rightarrow 1S \mid 01S \mid \epsilon$$

2. $S \Rightarrow \epsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$

3.
$$S \Rightarrow 0S1 \mid e$$

4. $S \rightarrow aSb \mid ab \mid bAa$ $A \rightarrow \epsilon \mid aA \mid bA$

Justification: Any string over $\{a, b\}$ can be generated from A. Productions $S \to ab \mid bAa$ generate the first b and the last a (and the number of a's preceding the first b is the same as the number of b's following the last a, it is 1 for the first production and 0 for the second). Production $S \to aSb$ adds one a preceding the first b and one b following the last a.

Solution to Exercise 3. The string *aabbab* has two distinct left derivations: $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabbAB \Rightarrow aabbaB \Rightarrow aabbab$ $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabB \Rightarrow aabbS \Rightarrow aabbaB \Rightarrow aabbab$

Solution to Exercise 4. We follow the method from the lecture. First we add productions

to P in order to obtain the smallest $P_1 \supseteq P$ such that

(a) if $A \to \alpha B\beta$ and $B \to \epsilon$ are in P_1 then $A \to \alpha\beta$ is in P_1 .

Any nonempty terminal string derived from S in G can be derived in (N, Σ, P_1, S) without using any ϵ -production. So we can remove the ϵ -productions from P_1 , obtaining P'_1 .

Now we add productions to P'_1 in order to obtain the smallest $P_2 \supseteq P'_1$ such that

(b) if
$$A \to B$$
 and $B \to \gamma$ are in P_2 then $A \to \gamma$ is in P_2 .

Any terminal string derived from S in (N, Σ, P'_1, S) can be derived in (N, Σ, P_2, S) without using any unit production. Thus we can remove the unit productions from P_2 , obtaining P'_2 .

 $G' = (N, \Sigma, P'_2, S)$ is the result, $L(G') = L(G) - \{\epsilon\}$. (Notice that in [Kozen] the rules (a) and (b) are applied together. Doing this separately, as above, is also correct.)

For the given grammar new productions are added as follows. In order to remove $\epsilon\text{-}$ productions:

production	with	production	gives	production
$B \to \epsilon$		$S \to AB$ $A \to BB$ $A \to bB$ $A \to B$		$S \to A$ $A \to B$ $A \to b$ $A \to \epsilon$
$A \rightarrow \epsilon$		$S \to AB$ $A \to SA$ $B \to aA$ $S \to A$		$S \rightarrow B$ $A \rightarrow S$ $B \rightarrow a$ $S \rightarrow \epsilon$
$S \to \epsilon$		$\begin{array}{c} A \to SA \\ A \to S \end{array}$		$\begin{array}{c} A \to A \\ A \to \epsilon \end{array}$
$B \to \epsilon$		$S \rightarrow B$		$S \to \epsilon$

The obtained set P'_1 of productions is:

$S \to AB$	A	B			
$A \to BB$	$\mid B$	bB	$\mid b \mid$	SA	S
$B \to aA$	a	b			

To get rid of unit productions:

production	with	production	gives	production
$S \to A$		$\begin{array}{l} A \rightarrow BB \\ A \rightarrow B \\ A \rightarrow bB \\ A \rightarrow b \\ A \rightarrow S \\ A \rightarrow S \end{array}$		$\begin{array}{l} S \rightarrow BB \\ S \rightarrow B \\ S \rightarrow bB \\ S \rightarrow b \\ S \rightarrow SA \\ S \rightarrow S \end{array}$
$S \rightarrow B$		$\begin{array}{c} B \rightarrow aA \\ B \rightarrow a \\ B \rightarrow b \end{array}$		$\begin{array}{l} S \rightarrow a A \\ S \rightarrow a \\ S \rightarrow b \end{array}$
$A \rightarrow B$		$\begin{array}{c} B \rightarrow aA \\ B \rightarrow a \\ B \rightarrow b \end{array}$		$\begin{array}{c} A \rightarrow a A \\ A \rightarrow a \\ A \rightarrow b \end{array}$
$A \rightarrow S$		$\begin{array}{l} S \to AB \\ S \to A \\ S \to B \end{array}$		$\begin{array}{c} A \to AB \\ A \to A \\ A \to B \end{array}$

The obtained set P'_2 of productions is:

 $\begin{array}{l} S \rightarrow AB \mid BB \mid bB \mid b \mid SA \mid aA \mid a \\ A \rightarrow AB \mid BB \mid bB \mid b \mid b \mid SA \mid aA \mid a \\ B \rightarrow aA \mid a \mid b \end{array}$

As we want to obtain a grammar equivalent to the initial one, the removed production $S \to \epsilon$ has to be added.

Solution to Exercise 5.

- 1. Introduce productions for each terminal symbol which does not occur on its own on the right hand side of some production, i.e.:
 - $\begin{array}{l} A \rightarrow \neg \\ B \rightarrow (\\ C \rightarrow \supset \\ D \rightarrow) \end{array}$

Then replace all such terminal symbols in the original grammar with the corresponding nonterminal from the productions above.

 $\begin{array}{l} S \rightarrow AS \mid BSCSD \mid p \mid q \\ A \rightarrow \neg \\ B \rightarrow (\\ C \rightarrow \supset \end{array}$

 $D \rightarrow)$

The only production above which is not in Chomsky normal-form is $S \rightarrow BSCSD$. We can systematically rewrite this production into a set of productions in Chomsky normal-form as follows:

 $S \rightarrow BSCSD$ is replaced by $S \rightarrow BE$ and $E \rightarrow SCSD$ $E \rightarrow SCSD$ is replaced by $E \rightarrow SF$ and $F \rightarrow CSD$ $F \rightarrow CSD$ is replaced by $R \rightarrow CG$ and $G \rightarrow SD$

Thus an equivalent Chomsky normal-form grammar is obtained:

$$\begin{split} S &\to AS \mid BE \mid p \mid q \\ E &\to SF \\ F &\to CG \\ G &\to SD \\ A &\to \neg \\ B &\to (\\ C &\to \supset \\ D &\to) \end{split}$$

2. First eliminate all unit productions. This yields

$$\begin{split} S &\to ABA \mid aA \mid a \mid bB \mid b \\ A &\to aA \mid a \mid bB \mid b \\ B &\to bB \mid b \end{split}$$

We then proceed as in the previous exercise: productions for terminal symbols are introduced where necessary and productions with right hand sides comprising three or more nonterminals are systematically rewritten into a set of productions in Chomsky normal-form. This results in the following grammar:

$$\begin{split} S &\rightarrow AE \mid CA \mid DB \mid a \mid b \\ A &\rightarrow CA \mid DB \mid a \mid b \\ B &\rightarrow DB \mid b \\ E &\rightarrow BA \\ C &\rightarrow a \\ D &\rightarrow b \end{split}$$

2 Advanced and Exam Like Exercises

Exercise 6. Consider the language P consisting of all properly balanced parentheses. That is, strings over (and) where each left parenthesis (has a matching right parenthesis). For example, the strings ((()())) and ()() are in P but the string)(() is not. Prove that P is context-free by providing a context-free grammar for P. Your grammar should be unambiguous (motivate why).

Solutions

Solution to Exercise 6. We prove that the language is context-free by constructing a context-free grammar which generates the language. A context-free grammar G for the language P is then given by:

$$S \to SS \mid (S) \mid \varepsilon$$

Here, we have made the assumption that $\varepsilon \in P$ since it, technically, fullfils the balanced parenthesis property. However, the grammar is *not* unambiguous since ε can be derived via, for example, $S \Rightarrow SS \Rightarrow \varepsilon S \Rightarrow \varepsilon \varepsilon$ or $S \Rightarrow \varepsilon$. Since the exercise specified that the grammar should be unambiguous we need to fix this, for example via the following grammar.

$$S \to (S)S \mid \varepsilon$$

This grammar is unambiguous. The rule $S \to (S)S$ ensures that every left parenthesis has a corresponding right parenthesis, and the derivation of the empty string is unique since the empty string can only be derived directly from S without further production.