Lecture L&M 5

Chomsky Normal Form, CYK Parsing

5.1 Noncontracting grammars

For each of the following grammars, construct an equivalent grammar G_L with a nonrecursive start symbol that is "essentially noncontracting", i.e. only the (new) start symbol may have a λ-rule.

(Extra:) Give a regular expression that describes the language accepted by each grammar.

A. Grammar G:

$$\begin{aligned} S &\to a\,S \mid b\,S \mid B \\ B &\to b\,b \mid C \mid \lambda \\ C &\to c\,C \mid \lambda \end{aligned}$$

B. Grammar G:

$$\begin{aligned} S &\to A\,B\,C \mid a\,B\,C \\ A &\to a\,A \mid B\,C \\ B &\to b\,B \mid \lambda \\ C &\to c\,C \mid \lambda \end{aligned}$$

5.2 Chain rules

For each of the following grammars, construct an equivalent grammar G_C without chain rules. (Note that the grammars have no λ-rules, but you may need to make the start symbol nonrecursive.)

C. Grammar G:

$$\begin{aligned} S &\to A\,S \mid A \\ A &\to a\,A \mid b\,B \mid C \\ B &\to b\,B \mid b \\ C &\to c\,C \mid B \end{aligned}$$

D. *(extra)* Grammar G:

$$\begin{aligned} S &\to A\,B \mid C \\ A &\to a\,A \mid B \\ B &\to b\,B \mid C \\ C &\to c\,C \mid a \mid A \end{aligned}$$

5.3 Useless nonterminals

For each of the following grammars, construct an equivalent grammar G_T without nonterminating symbols, and then an equivalent grammar G_U without unreachable (and without nonterminating) symbols. Use the algorithms given in the lecture (and in the book).

E. Grammar G:

$$\begin{aligned} S &\to A\,A \mid C\,D \mid b\,B \\ A &\to a\,A \mid a \\ B &\to b\,B \mid b\,C \\ C &\to c\,B \\ D &\to d\,D \mid d \end{aligned}$$

F. *(extra)* Grammar G:

$$\begin{aligned} S &\to A\,C\,H \mid B\,B \\ A &\to a\,A \mid a\,F \\ B &\to C\,F\,H \mid b \\ C &\to a\,C \mid D\,H \\ D &\to a\,D \mid B\,D \mid C\,a \\ F &\to b\,B \mid b \\ H &\to d\,H \mid d \end{aligned}$$

5.4 Finalisation of CNF rules

For each of the following grammars, construct an equivalent one in Chomsky normal form (CNF). (Note that the grammars already have a nonrecursive start symbol, are essentially noncontracting, and contain neither chain rules nor useless symbols.)

G. Grammar G:

$$\begin{aligned} S &\to aA \mid ABa \\ A &\to AA \mid a \\ B &\to AbB \mid bb \end{aligned}$$

H. Grammar G:

$$\begin{aligned} S &\to aAbB \mid ABC \mid a \\ A &\to aA \mid a \\ B &\to bBcC \mid b \\ C &\to abc \end{aligned}$$

Note: The right-hand sides of the grammar in Exercise G have at most three symbols. The main difference in Exercise H is that you have to deal with right-hand sides of more than three symbols. During the tutorial, after doing G, we recommend that you start with H, but as soon as you feel that you have understood the principle, move on to the next exercise.

5.5 The CYK parsing algorithm

I. Given the context-free (CNF) grammar $G = (V, \Sigma, P, S)$ with $V = \{S, A, B, C, D, E\}$, $\Sigma = \{a, b, c\}$ and P consisting of the rules

$$\begin{aligned} S &\rightarrow CA \mid BD \\ A &\rightarrow a \\ B &\rightarrow b \mid c \\ C &\rightarrow AE \\ D &\rightarrow EB \mid b \\ E &\rightarrow BD \mid CA \end{aligned}$$

(a) Construct the X_{ij}-matrix for the words $acba$ and $baca$ according to the CYK-algorithm.

(b) Use the X_{ij}-matrix to determine whether $acba$ and $baca$ belong to $\mathcal{L}(G)$, and if yes, what the corresponding derivation tree is.

(c) *(extra)* Would you call the CYK-algorithm a "top-down" or a "bottom-up" parsing algorithm, or neither? Motivate your answer.

J. *(extra)* Given the context-free grammar $G = (V, \Sigma, P, S)$ with $V = \{S, A, B, C, D\}$, $\Sigma = \{a, b, c\}$ and P consisting of the rules

$$\begin{aligned} S &\rightarrow A\,C \mid D\,B \\ A &\rightarrow a \\ B &\rightarrow b \mid c \\ C &\rightarrow S\,A \\ D &\rightarrow B\,S \mid b \end{aligned}$$

(You can ignore the recursion in the start symbol S for this exercise.)

(a) Construct for the words $abca$ and $acab$ the X_{ij}-matrix.

(b) Use the X_{ij}-matrix to determine whether $abca$ and $acab$ belong to $\mathcal{L}(G)$, and if yes, what the corresponding derivation tree is.

K. *(extra)* Give the upper diagonal matrix produced by the CYK algorithm when run with the Chomsky normal form grammar G below and the input strings *abbb* and *aabbb*.

$$S \rightarrow aXb \mid ab$$