Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
P. Schneider, C. Zheng

November 11, 2016

Theoretical Computer Science - Bridging Course
Winter Term 2016
Exercise Sheet 4

Hand in (electronically or hard copy) before your weekly meeting but not later than 23:59, Wednesday, November 23, 2016

Exercise 1: Regular expressions (3 points)

If you use a calculator, floating point numbers are commonly illustrated in the following form:

-1.23456789E-123

Show that the language of floating point numbers in calculator notation is regular, by giving a regular expression for it.

Exercise 2: Context-Free Grammars (3+3+0 points)

Show that the following languages over the alphabet $\Sigma = \{a, b\}$ are context-free by giving a corresponding context-free grammar (CFG).

(a) $L_1 := \{w \in \Sigma^* \mid \text{the length of } w \text{ is odd}\}$.

(b) $L_2 := \{a^m b^n a^n \mid m, n \geq 1\}$

(c) $L_3 := \Sigma^* \setminus \{a^n b^n \mid n \geq 0\}$ (voluntary, no points!).

Exercise 3: Chomsky Normal Form (5 points)

Convert the following CFG into an equivalent CFG in Chomsky Normal Form (CNF). Write down the grammar you obtain after each step of the conversion algorithm.

$$\begin{aligned} A &\rightarrow BAB \mid B \mid \epsilon \\ B &\rightarrow 00 \mid \epsilon \end{aligned}$$

Exercise 4: Cocke-Younger-Kasami Algorithm (3+3 points)

Consider this CFG in CNF $G := (V := \{S, A, B, C, X, Y, Z\}, \Sigma := \{a, b, c\}, R, S)$ with R given by:

$$\begin{aligned} S &\rightarrow XY \\ A &\rightarrow a \\ B &\rightarrow b \\ C &\rightarrow c \\ X &\rightarrow AB \mid AZ \\ Y &\rightarrow c \mid CY \\ Z &\rightarrow XB \end{aligned}$$

For $w := x_1 \cdots x_n$ with $x_1, \ldots, x_n \in \Sigma$ we define $T_{i,j}$ with $1 \leq i \leq n, 1 \leq j \leq n - i + 1$ as the set of those variables of V from which we can derive $x_i \cdots x_{i+j-1}$ (a substring of w which starts at the i-th symbol of w and has length j). I.e. $T_{i,j} := \{X \in V \mid X \to^* x_i \ldots x_{i+j-1}\}$.

It holds that $w \in L(G)$ if and only if $S \in T_{1,n}$. The Cocke-Younger-Kasami (CYK) algorithm computes the sets $T_{i,j}$ (and stores them in a table $T[1..n, 1..n]$) for a given CFG in CNF and input string $w := x_1 \cdots x_n$. Its running time is polynomial in n ($\mathcal{O}(n^3)$ to be more specific).

Use the CYK-algorithm (manually) to test whether or not the following strings are in $L(G)$:

$$w_1 := aabbcc \text{ and } w_2 := aaabbc.$$

Hint: Draw the table $T[1..n, 1..n]$ and note down your intermediate results. The entries $T[i, j]$ with $j > n - i + 1$ are not required for this algorithm (you can mark them accordingly).

Algorithm 1 CYK-algorithm

Input: String $w = x_1 \cdots x_n$, CFG $G := (V, \Sigma, R, S)$ in CNF.

for $i := 1$ to n **do**
    $T[i, 1] := \{X \in V \mid X \to x_i\}$
end for
for $j := 2$ to n **do**
    for $i := 1$ to $n - (j - 1)$ **do**
        $T[i, j] := \emptyset$
        for $k := 1$ to $j - 1$ **do**
            $T[i, j] := T[i, j] \cup \{X \in V \mid X \to YZ \in R \text{ and } Y \in T[i, k] \text{ and } Z \in T[i + k, j - k]\}$
        end for
    end for
end for