Theory of Computer Science

G. Röger
Spring Term 2019

University of Basel
Computer Science

Exercise Sheet 6 — Solutions

Exercise 6.1 (Chomsky Normal Form; 2 Points)

Specify a grammer in Chomsky normal form that generates the same language as grammar $G = \langle \Sigma, V, P, S \rangle$ with $\Sigma = \{a, b, c\}$, $V = \{S, X, Y\}$ and the following rules P:

$$\begin{aligned} S &\to XY \\ X &\to c \\ X &\to cS \\ Y &\to abb \\ Y &\to aYb \\ Y &\to \epsilon \end{aligned}$$

Solution:

$G' = \langle \Sigma, V', P', S \rangle$ mit $V = \{S, X, Y, Z, A, B, C\}$ and P' consists of the following rules:

$$\begin{aligned} S &\to XY \mid c \mid CS & A &\to a \\ X &\to c \mid CS & B &\to b \\ Y &\to AZ \mid AB & C &\to c \\ Z &\to BB \mid YB \end{aligned}$$

Exercise 6.2 (Length of Derivations in Chomsky Normal Form; 2 Points)

Let G be a grammar in Chomsky normal form and $w \in \mathcal{L}(G)$ a non-empty word ($w \neq \varepsilon$), which is generated by G. Show that every derivation of w from the start variable of G consists of exactly $2|w| - 1$ steps.

Solution:

A grammar is in Chomsky Normal Form if each rule has one of the following three forms: (a) $A \to BC$, (b) $A \to a$ or (c) $S \to \varepsilon$, where A, B, C are variables, S is the start variable (and B and C are not the start variable) and a is a terminal symbol.

Let w be a word of the language with length $n > 0$ which is generated by the grammar. Since rules of type (c) are only utilized to derive the empty word, they are not relevant here. In order to introduce the n terminal symbols of w, any derivation of the word needs to apply rules of type (b) exactly n times. Each of these rule applications remove one nonterminal symbol which means that for each of these rule applications there must have been one nonterminal symbol before. These nonterminal symbols must have been introduced by applying rules of type (a). Each such application increases the number of nonterminal symbols by 1. Since the start variable is already such a nonterminal symbol, we thus need to apply *at least* $n-1$ times a rule of type (a). Nonterminal symbols are only eliminated by rules of type (b), which means that we can apply rules of type (a) *at most* $n-1$ times. Thus each derivation of w consists of exactly $n + n - 1 = 2n - 1$ derivation steps.

Exercise 6.3 (PDAs, 2 Points)

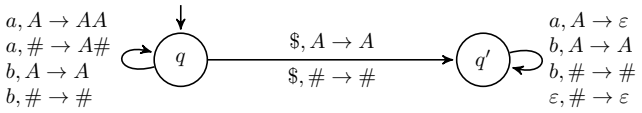
Specify a PDA that accepts that language

$$L = \{w_1 \$ w_2 \mid w_1, w_2 \in \{a, b\}^* \text{ and } w_1 \text{ and } w_2 \text{ contain the same number of } a\text{s}\}$$

over $\Sigma = \{a, b, \$\}$.

Solution:

$M = (\{q, q'\}, \{a, b, \$\}, \{A, \#\}, \delta, q, \#)$ with the following transition function δ:

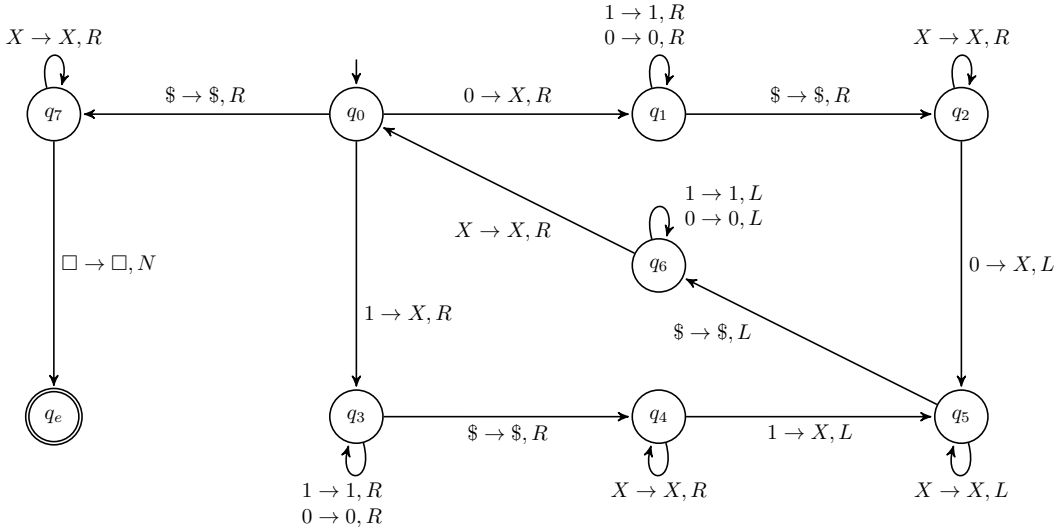


Exercise 6.4 (Nondeterministic Turing Machines; 4 Points)

Consider language $L = \{w\$w \mid w \in \{0,1\}^*\}$ over $\{0, 1, \$\}$. Specify the state/transition diagram of an NTM M with $\mathcal{L}(M) = L$. Also explain the behaviour of your TM in words.

Solution:

$M = \langle\{q_0, q_1, \ldots, q_7, q_e\}, \{0, 1, \$\}, \{0, 1, \$, X, \square\}, \delta, q_0, \square, \{q_e\}\rangle$ with the following transition function δ:



The TM always crosses out (with tape symbol X) and remembers the next symbol of the left part of the input and then tries to cross out the corresponding symbol on the right. If it is possible to cross out all symbols this way, it can terminate.

Whenever the machine is in state q_0, the R/W head is on the first input symbol that has not yet been processed. If it is a 0, it replaces it with an X and moves to the right, traversing all remaining symbols from the left part of the input (q_1), the separator \$, and all positions of the right part that have already been crossed out (q_2). If it ends up on a 0, it can cross it out (transition from q_4 to q_5) and move the head back over the crossed out symbols of the right part (q_5), the separator, and the non-processed symbols of the left part (q_6). The cycle $q_0, q_3, q_4, q_5, q_6, q_0$ works analogously, but crossing out symbol 1.

If all 0s and 1s of the left word have been crossed out, the TM reads a \$ in state q_0. It still needs to verify that also all symbols of the right word have been processed, which is done in the path via q_7 to q_e.