Theoretical Computer Science 2
Exercise Sheet 2

René Maseli — TU Braunschweig
Prof. Dr. Roland Meyer — Summer semester 2024

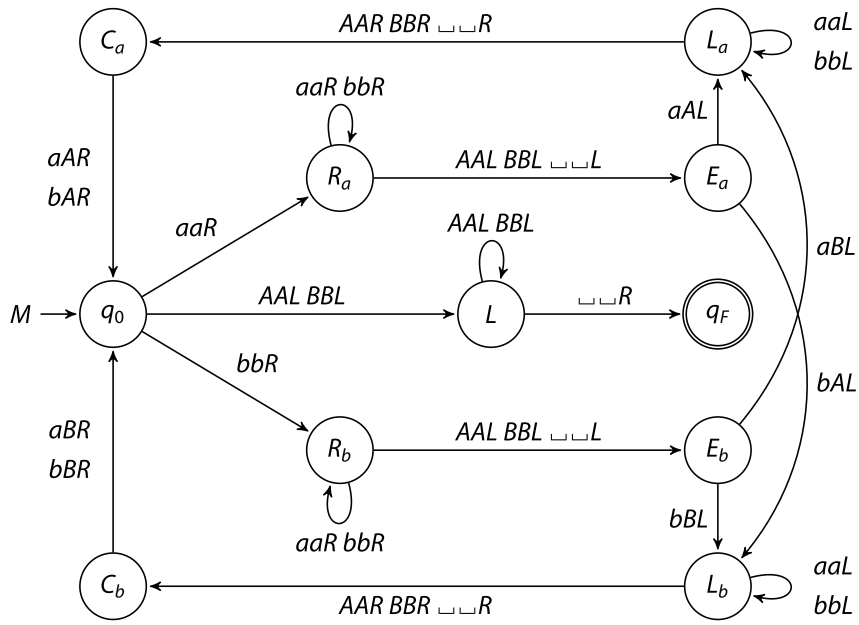
Release: 04/30/2024 — Due: 05/09/2023, 23:59

Hand in your solutions to the Vips directory of the Stud.IP course until thursday, 09.05.2023 11:59 pm. You should provide your solutions either directly as PDF file or as a readable scan or photo of your handwritten notes. Submit your results as a group of four. On the front page, state the **degree programme, name, surname and student id** of each member of your group.

Homework Exercise 1: TM-Analysis [4 points]
Betrachten Sie die Turing-Maschine $M = \langle Q, \{a, b\}, \{a, b, A, B, ␣\}, \delta, q_0, \{q_F\}\rangle$
wobei $Q = \{q_0, R_a, R_b, L_a, L_b, E_a, E_b, C_a, C_b, L, q_F\}$ und δ gegeben ist durch folgenden Graphen.

a) [1 point] Give the computation of this machine on input *aaab*, especially whenever it reaches q_0 or q_F.

b) [3 points] Determine the computed (partial) function, as well as an informal description of how this TM works. Briefly describe the task each state performs.



Homework Exercise 2: Composition of computable functions [4 points]
Let $f : A \not\to B$ and $g : B \not\to C$ partial computable functions.

a) [3 points] Show by constructing a suitable TM, that the composition $g \circ f : A \not\to C$ with $(g \circ f)(w) = g(f(w))$ is computable.

b) [1 point] For which inputs is this function undefined?

Remark: Just waiting for the right state in both TMs would not always do the trick. Why could this be? How could this cause problems in the resulting algorithm? You will need to beware the definition of computable functions from the lecture.

Homework Exercise 3: Operations on decidable languages [4 points]

Let $K, L \subseteq \Sigma^*$ be decidable languages. Prove that

a) [1 point] Prove that the union $K \cup L$,

b) [1 point] the intersection $K \cap L$,

c) [2 points] as well as the concatenation $K.L = \{k.l \in \Sigma^* \mid k \in K, l \in L\}$ are decidable.

Do this by providing deciders for each of the languages, given arbitrary deciders M_K for K and M_L for L, and explain their behavior. You do not need to give a formal construction as a tuple.

Homework Exercise 4: Totality [4 points]

With the lecture's Halting Problem in mind, consider the following, slightly-different problem.

Totality	
Given:	Turing machine M
Question:	Does M halt on **every** input $w \in \{0, 1\}^*$?

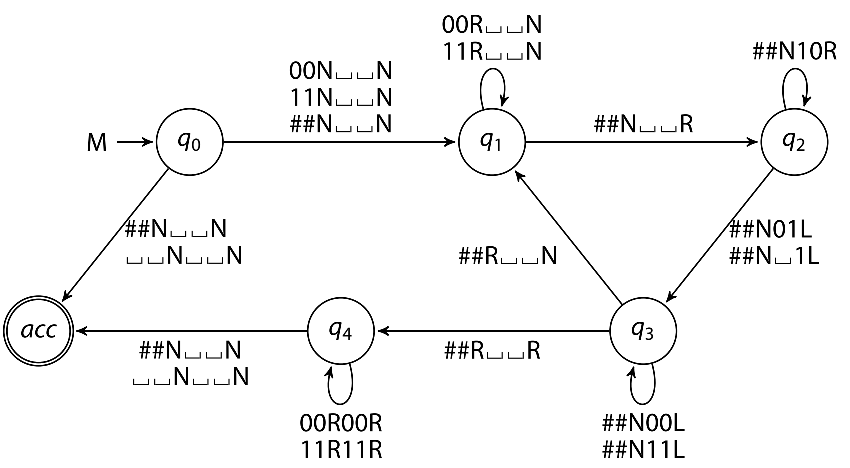
a) [2 points] Let $f : \mathbb{N} \rightarrow \{0, 1\}^*$ be a computable partial function. Show that $d : \mathbb{N} \rightarrow \{0, 1\}^*$ with $d(n) := 101M_{f(n)}(f(n))$ is also computable. Here, let $A(x)$ denote the output of a TM A on input x, if there is one, and otherwise be undefined.

b) [1 point] Given that Totality would be recursively-enumerable, show that a **total**, computable d as defined above exists.

c) [1 point] Now show by contradiction with the use of a diagonalization, that Totality is not recursively-enumerable.

Exercise 5:

Given the following non-deterministic Turing Machine M with two tapes. It is labeled by 6-tuples in $(\Gamma \times \Gamma \times \{L, N; R\})^2$. Describe the behavior of the machine M, especially the behavior of each of its control states. (Hint: It uses least-significant-bit-first encoding)



Exercise 6:

Show that the problem **Prime Number** is decidable.

Prime Number	
Given:	A number $n \in \mathbb{N}$
Question:	Is n a prime number?

Give an algorithm in pseudo code.

Exercise 7:

Konstruieren Sie zu einem beliebigen PDA $A = \langle Q, \Sigma, \Gamma, q_0, \#, \delta \rangle$ mit Akzeptanz beim leeren Stack, eine NTM M mit $\mathcal{L}(M) = \mathcal{L}(A)$. Erklären Sie, warum ihre Konstruktion korrekt ist.

Exercise 8:

Betrachten Sie das Problem **List Membership** und die dazugehörige Sprache über $\Sigma = \{0, 1, \#\}$. Konstruieren Sie formal einen Entscheider für $L_{\text{List Membership}}$. Sie dürfen auch mehrere Bänder benutzen.

List Membership	
Given:	Liste von Zahlen $n_1, n_2, \ldots, n_m \in \mathbb{N}$ und eine Zahl $k \in \mathbb{N}$
Question:	Taucht k in der Liste auf?

Exercise 9:

Zeigen Sie, dass das Problem **Uniqueness** entscheidbar ist. Nutzen Sie dazu eine Darstellung Ihrer Wahl. Dazu können Sie Ihren Entscheider für $L_{\text{List Membership}}$ als Subroutine benutzen (siehe vorherige Aufgabe).

Uniqueness	
Given:	Liste von Zahlen $n_1, n_2, \ldots, n_m \in \mathbb{N}$
Question:	Sind alle Zahlen in der Liste paarweise verschieden?