Gruppe	Hemmecke (10:15)	Hemmecke (11:00)	Popov (11:00)
Name		Matrikel	SKZ

Klausur 1

Berechenbarkeit und Komplexität

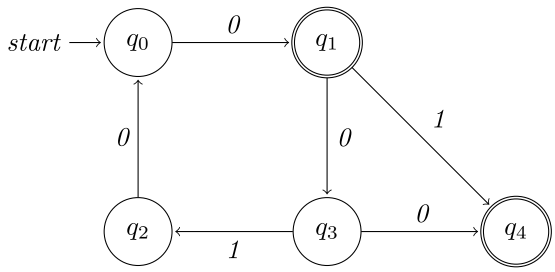
22. November 2013

Part 1 *NFSM2013*

Let N be the nondeterministic finite state machine

$$(\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \nu, \{q_0\}, \{q_1, q_4\}),$$

whose transition function ν is given below.



1		no	*Is $001000100101 \in L(N)$?*

A word $w \in L(N)$ with $|w| > 1$ never ends with 101.

2		no	*Is $L(N)$ finite?*
3		no	*Is $L(N) = L(r)$ for the regular expression $r = (0010)^*(01 + 0)$?*

Arden's lemma yields $r = (0010)^*0(00 + 1 + \varepsilon)$?

4	yes		*Let $L = \left\{ 01^n w \,\middle|\, n \in \mathbb{N}, w \in \{0,1\}^* \setminus L(N) \right\}$. Is L a regular language?*
5		no	*Is there a deterministic finite state machine M with less than 4 states such that $L(M) = L(N)$?*

The only words of $L(N)$ with less than 4 letters are 0, 01, and 000. All other words start with a finite repetition of the 4 letter word 0010. Obviously, a DFSM that is able to allow only 4-letter-repetitions (and not shorter repetitions) must have at least 4 states.

6	yes		*Is there a deterministic Turing machine $T = (Q, \Gamma, \sqcup, \{0,1\}, \delta, q, F)$ with $L(T) = L(N)^*$?*

We have $L(N)^* = L(r)$ for $r = ((0010)^*0(00 + 1 + \varepsilon))^*$. Hence, $L(N)^*$ is regular, and thus also recursively enumerable.

7		no	*Does for each Turing machine H (that halts on every input) exists a nondeterministic finite state machine F such that $L(F) = L(N) \cap L(H)$?*

For example, let H be such that $L(H) = \left\{ (0010)^{n^2}0(00 + 1 + \varepsilon) \,\middle|\, n \in \mathbb{N} \right\}$. Then $L(N) \cap L(H) = L(H)$. However, $L(H)$ is not regular.

Part 2 *Pumping2013*

Let

$$L_1 = \left\{ a^{(m^2)} b^n \,\middle|\, m, n \in \mathbb{N} \right\},$$

$$L_2 = \{ a^n b^n \mid n \in \mathbb{N}, n < 1000 \}.$$

8		no	*Is there a regular expression r such that $L(r) = L_1$?*
9	yes		*Is there a deterministic finite state machine M such that $L(M) = \overline{L_2} := \{a, b\}^* \setminus L_2$?*

L_2 is regular, i.e., its complement $\overline{L_2}$ is also regular.

10	yes		*Is there an enumerator Turing machine G such that $Gen(G) = L_1$?*
11	yes		*Is there a Turing machine M such that $L(M) = L_1 \cup L_2$?*
12	yes		*Is there an deterministic finite state machine D such that $L(D) = L_1 \cap L_2$?*

The language $L_1 \cap L_2$ is finite and thus regular.

Part 3 *WhileLoop2013*

Let T and H be Turing machines with the property that H halts on every input. Furtermore assume that T and H compute functions $t, h : \mathbb{N} \to \mathbb{N}$, respectively. (We assume that a natural number n is encoded on the tape as a string of n letters 0.)

13	yes		*Is there a WHILE-program that computes t?*

Every Turing machine can be simulated by a while program.

14		no	*Is there a LOOP-programm that computes h?*

The Ackermann function ack is a total function that is not primitive recursive. Hence, if H is the Turing machine that computes $h(n) = \mathrm{ack(n, n)}$, then we can assume that H holds on every input. However, since h is not primitive recursive, there cannot be a corresponding LOOP-program.

15	yes		*Is t a recursive function?*
16	yes		*Is every primitive recursive function computable by a LOOP-program?*

Part 4 *Recursive2013*

Let $f : \mathbb{N} \to \mathbb{N}$ be a recursive function that is defined on $D = \mathrm{domain}(f) \subseteq \mathbb{N}$ and define another function $d : \mathbb{N} \to \mathbb{N}$ by

$$d(n) = \begin{cases} f(n) & \text{if } n \in D \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

17		no	*Can it be concluded that d is LOOP-computable?*

Not every total function is primitive recursive. The function $f(n) = d(n) = \mathrm{ack}(n, n)$ is a total function that is not primitive recursive. Only primitive recursive functions are LOOP-computable.

18	yes		*Does there exist an enumerator Turing machine M such that $\mathrm{Gen}(M) = \{\, 0^{f(n)} \mid n \in D \,\}$?*
19	yes		*Can it be concluded that the language $L = \{\, 0^n \mid n \in D \,\}$ is recursively enumerable?*

f is a recursive function iff f is Turing computable iff its graph is recursively enumerable. Thus one can construct an enumerator G that generates simply the first component of the graph of f, i.e. $\mathrm{Gen}(G) = D$. But that would mean D is recursively enumerable.

Part 5 *TM2013*

Let $M = (Q, \Gamma, \sqcup, \Sigma, \delta, q_0, F)$ *be a Turing machine with* $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $F = \{q_0\}$. *The transition function*

$$\delta : Q \times \Gamma \rightarrow_P Q \times \Gamma \times \{L, R\}$$

is given by the following table.

δ	0	1	$\sqcup$
q_0	$(q_1, 0, R)$	$(q_0, 0, R)$	–
q_1	$(q_0, 0, L)$	$(q_2, 0, R)$	$(q_0, 0, R)$
q_2	–	–	–

20 | | no

Is $q_0 110 \vdash 1 q_0 10 \vdash 11 q_0 0 \vdash 110 q_1 \sqcup \vdash 1100 q_0 \sqcup$ *a computation of* M*?*

M always writes a 0, so the computation rather looks like
$q_0 110 \vdash 0 q_0 10 \vdash 00 q_0 0 \vdash 000 q_1 \sqcup \vdash 0000 q_0 \sqcup$.

21 | | no

Is $001 \in L(M)$*?*

The machine M does not terminate. It rather loops between state q_0 and state q_1 and moves its head just between the two initial 0's.

22 | yes |

Is $L(M)$ *a recursively enumerable language?*

23 | yes |

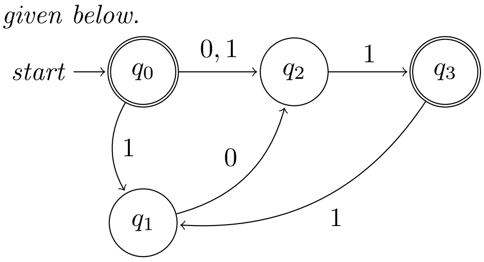
Is there a Turing machine H *that halts on every input with* $L(H) = L(M)$.

If the machine M hits 00, it jumps into a loop and the corresponding word will never be accepted. That's the only case where it will not terminate. So, one can simply change the transition function δ of M in one single place, namely we let $\delta(q_1, 0) = (q_2, 0, R)$ and take this modified Turing machine as H. H always moves its head to the right and must thus eventually (since the input is finite) hit a $\sqcup$. From there it will be at most one step to termination.

Part 6 *Open2013*

((2 points))

Let $N = (Q, \Sigma, \delta, q_0, F)$ *be a nondeterministic finite state machine with* $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $S = \{q_0\}$, $F = \{q_0, q_3\}$, *and transition function* δ *as given below.*



1. *Let* X_i *denote the regular expression for the language accepted by* N *when starting in state* q_i.

   Write down an equation system for $X_0, \ldots, X_3$.

2. *Give a regular expression* r *such that* $L(r) = L(N)$ *(you may apply Arden's Lemma to the result of 1).*

$$
\begin{aligned}
X_0 &= 1X_1 + (0+1)X_2 + \varepsilon \\
X_1 &= 0X_2 \\
X_2 &= 1X_3 \\
X_3 &= 1X_1 + \varepsilon \\
r &= ((0+1)+10)(110)^*1 + \varepsilon
\end{aligned}
$$