

Problems Solved:

31	32	33	34	35
----	----	----	----	----

Name:**Matrikel-Nr.:**

Problem 31. Let Σ be an alphabet and A be a set ($A \subseteq \Sigma^*$). Let also A be semi-decidable, but not decidable. Prove that the complement of A is not decidable.

Problem 32. Let M_0, M_1, M_2, \dots be a list of all Turing machines with alphabet $\Sigma = \{0, 1\}$, such that the function $i \mapsto \langle M_i \rangle$ is computable. Let $w_i = 01^i0$ for all natural numbers i . Let $L = \{w_i \mid i \in \mathbb{N} \text{ and } M_i \text{ accepts } w_i\}$ and $\bar{L} = \Sigma^* \setminus L$.

- (a) Is L recursively enumerable?
- (b) Is \bar{L} recursively enumerable?
- (c) Is L recursive?
- (d) Is \bar{L} recursive?

Justify your answers.

Problem 33. Let L be a finite language over an alphabet $\{0, 1\}$. Is the following problem

For a Turing machine M it holds $L(M) \supseteq L$.

semi-decidable? Is it also decidable?

Problem 34. Which of the following problems are decidable? In each problem below, the input of the problem is the code $\langle M \rangle$ of a Turing machine M with input alphabet $\{0, 1\}$.

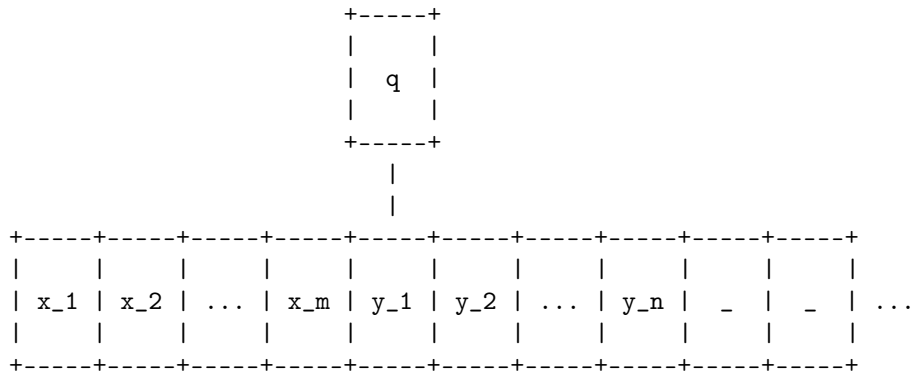
1. Is $L(M)$ empty?
2. Is $L(M)$ finite?
3. Is $L(M)$ regular?
4. Is $L(M) \subseteq \{0, 1\}^*$?
5. Is $L(M)$ not recursively enumerable?
6. Does M have an even number of states?

Problem 35. (a) Given any Turing machine M , construct a grammar G with the following property:

$$L(G) \neq \emptyset \iff M \text{ halts on the empty input } \epsilon. \quad (1)$$

The construction is supposed to be computable.

Hint: Encode reachable configurations



of the Turing machine as the sententials forms

$$\#x_1x_2 \dots x_mqy_1y_2 \dots y_n\#$$

of G . Simulate transitions of the Turing machine by productions of the grammar.

- (b) Is it decidable if a grammar G satisfies $L(G) \neq \emptyset$? (An instance of this decision problem is a grammar coded as a bit string.) Justify your answer.
- (c) Is it decidable if two grammars G_1 and G_2 describe the same language? (An instance of this decision problem is a bit string that encodes a pair (G_1, G_2) of grammars.) Justify your answer.