Instructions

- Classroom Problems C1.1–C1.3 will be discussed and solved at the tutorial session on Wed 15 Jan, 14–16, Room T4 (A238). No credit is given for these problems.
- Homework Problems H1.1–H1.3 you should solve on your own, and submit your solutions via the MyCourses interface by the deadline of Tue 21 Jan, 23:59, These problems will be individually graded on a scale of 0–2 points per problem.
- In preparing your solutions to the Homework Problems:
  1. Justify your solutions, be precise, and provide sufficient detail so that it is easy to follow your reasoning.
  2. Submit your solutions as an easily readable, single pdf file, which is either typeset or written in full sentences and clean handwriting.
  3. **[Code of Conduct]** You can discuss the problems with your colleagues and the course's teaching staff, but you must write the presentations of your solutions *independently* and *individually*, without any notes from such discussions.

Classroom Problems

C1.1 Prove the following statements:

(i) For any $r < s$, $n^r = o(n^s)$, i.e. $n^r = \mathcal{O}(n^s)$ but $n^s \neq \mathcal{O}(n^r)$.

(ii) For any $r, c > 1$, $n^r = o(c^n)$, i.e. $n^r = \mathcal{O}(c^n)$ but $c^n \neq \mathcal{O}(n^r)$.

(iii) For all $a, b > 1$, $\log_a n = \Theta(\log_b n)$.

C1.2 Prove that the family $\mathcal{L}$ of all languages over the alphabet $\Sigma = \{0, 1\}$ is uncountable ("nondenumerable"), i.e. that there cannot be any enumeration of such languages by integer indices $L_1, L_2, \ldots$ so that

$$\mathcal{L} = \{L_i \subseteq \Sigma^* \mid i = 1, 2, \ldots\}.$$

(*Hint:* Identify each binary string x with the integer whose binary representation is $1x$, i.e. $\epsilon \sim \#(1) = 1$, $0 \sim \#(10) = 2$, $1 \sim \#(11) = 3$, $00 \sim \#(100) = 4$, etc. Assume such an enumeration exists and consider the language $D = \{x \in \Sigma^* \mid x \notin L_x\}$.)

C1.3 Design (i.e. give the transition diagram for) a Turing machine M that computes the following function $f : \{1\}^* \longrightarrow \{1\}^*$:

$$f(x) = \begin{cases} 1^{n-1} & \text{if } x = 1^n \text{ and } n \text{ is odd} \\ \varepsilon & \text{otherwise} \end{cases}$$

where ε denotes the empty string. Thus, for instance, $f(111) = 11$ and $f(11) = \varepsilon$.

Homework Problems

H1.1 Prove the following statements:

(i) For any $r, s > 1$, $(\log_2 n)^s = o(n^r)$, i.e. $(\log_2 n)^s = \mathcal{O}(n^r)$ but $n^r \neq \mathcal{O}(\log_2 n)^s$.

(ii) Show that

$$\log_2 n! = \sum_{i=1}^{n} \log_2 i = \Theta(n \log_2 n).$$

(*Hint:* Construct an upper bound for the value of the sum using the $\log_2 n$ term, and a lower bound using the term $\log_2 \frac{n}{2}$.) (2 points)

H1.2 Design (i.e. give the transition diagram for) a Turing machine M that computes the following function $f : \{1\}^* \longrightarrow \{1\}^*$:

$$f(x) = \begin{cases} 1^{n-2} & \text{if } x = 1^n \text{ and } n \geq 2, \\ 1^{n+1} & \text{if } x = 1^n \text{ and } n < 2. \end{cases}$$

(For instance, $f(111) = 1$ and $f(1) = 11$).

Give the computation sequences of your machine, i.e. the lists of configurations the machine passes through until it halts, on inputs 11 and ε (the empty input string). (2 points)

H1.3 Consider languages over the binary alphabet $\Sigma = \{0, 1\}$.

(i) Prove that if a language $L \subseteq \Sigma^*$ is decided by some Turing machine, then so is its complement $\bar{L} = \Sigma^* \setminus L = \{x \in \Sigma^* \mid x \notin L\}$. Would the same argument prove the claim for accepting rather than deciding Turing machines? Why not?

(ii) Prove that if a language $L \subseteq \{0, 1\}^*$ is accepted by some Turing machine M_1, and also its complement $\bar{L}$ is accepted by some Turing machine M_2, then L is actually decided by some Turing machine M. (No need to present a detailed construction for the machine M. Just explain at a high level how one could combine an acceptor for L and an acceptor for $\bar{L}$ into a decider for L.) (2 points)