Tutorial 9

Exercise 1 (compulsory)

Which of these definitions of the O-notation are correct?

1. $f = O(g)$ iff
   there exist positive integers c and n_0 such that for all $n \geq n_0$ we have that $f(n) \geq c \cdot g(n)$
2. $f = O(g)$ iff
   for all positive integers c and n_0 it is the case that for all $n \geq n_0$ we have that $f(n) \leq c \cdot g(n)$
3. $f = O(g)$ iff
   there exist positive integers c and n_0 such that for all $n \geq n_0$ we have that $f(n) \leq c \cdot g(n)$
4. $f = O(g)$ iff
   for all positive integers c and n_0 it is the case that there exists an $n \geq n_0$ such that $f(n) \leq c \cdot g(n)$

Solution:

1. incorrect
2. incorrect
3. correct
4. incorrect

Exercise 2 (compulsory)

Which of the following claims are true? Give precise arguments (proofs) for your answers.

1. $3n^2 + 2n + 7 = O(n^2)$
2. $n^2 = O(n \log n)$
3. $3^n = 2^{O(n)}$ (Hint: $3 = 2^{\log_2 3}$)
4. $n^2 = o(n^3)$
5. $n^2 = o(n^2)$
6. $n = o(2n)$

Solution:

1. True. Take $c = 12$ and $n_0 = 1$. Then clearly $3n^2 + 2n + 7 \leq 3n^2 + 2n^2 + 7n^2 \leq 12n^2$ and so $3n^2 + 2n + 7 \leq c \cdot n^2$ for all $n \geq n_0 = 1$.
2. False. By contradiction. Assume that there are constants c and n_0 such that $n^2 \leq c \cdot n \log n$ for all $n \geq n_0$. This would mean that $n \leq c \cdot \log n$ and hence $\frac{n}{\log n} \leq c$ for all $n \geq n_0$. However, this cannot be the case as $\frac{n}{\log n}$ goes to ∞ as n goes to ∞ and hence the expression $\frac{n}{\log n}$ will eventually overgrow any chosen constant c. Contradiction.
3. True. Note that $3 = 2^{\log_2 3}$. Hence $3^n = (2^{\log_2 3})^n = 2^{n \log_2 3} = 2^{O(n)}$.
4. True. Because $\lim_{n \to \infty} \frac{n^2}{n^3} = \lim_{n \to \infty} \frac{1}{n} = 0$.
5. False. Note that $\lim_{n \to \infty} \frac{n^2}{n^2} = 1 \neq 0$.
6. False. Note that $\lim_{n \to \infty} \frac{n}{2n} = \lim_{n \to \infty} \frac{1}{2} = \frac{1}{2} \neq 0$.

Exercise 3 (compulsory)

Assume a 5-tape Turing machine M running in time $O(n^3)$. What is the time complexity of the corresponding single-tape Turing machine simulating M?

Solution:
The time complexity is $O(n^6)$ because $(n^3)^2 = n^6$.

Exercise 4 (compulsory)

Which of the following statements about the class P are correct?

1. P is the class of all languages that are decidable by deterministic single-tape Turing machines running in polynomial time.
2. P is the class of all languages such that if $w \in P$ then there is a deterministic single-tape Turing machine which accepts the string w in polynomial time.
3. P is the class of all languages that are decidable by deterministic multi-tape Turing machines running in polynomial time.
4. A language L belongs to P iff there is a constant k and a decider M running in time $O(n^k)$ such that $L = L(M)$.
5. A language L belongs to P iff $L \in$ TIME(2^n).

Give 5 languages that are in the class P. Does A_{TM} belong to P?

Solution:

1. Correct.
2. Incorrect. The elements of P are languages, not strings! In fact the whole statement is just one big nonsense.
3. Correct. Note that any multi-tape TM running in polynomial can be simulated by a single-tape TM running also in polynomial time (it is only quadratically slower).
4. Correct.
5. Incorrect. The implication from left to right of course holds, but the one from right to left does not hold.

The following languages (for example) belong to P:

- $PATH$ (see the slides/book for the definition)
- $\emptyset$
- $\{a^k b^k \mid k \geq 0\}$
- $\{\langle G \rangle \mid G$ is a connected graph $\}$
- $\{\langle M \rangle \mid M$ is a TM which has more than 10 states $\}$

The language A_{TM} does not belong to P because A_{TM} is an undecidable language and P contains only decidable languages.

Exercise 5 (compulsory)

Define the language ALL_{DFA} and show that $ALL_{DFA} \in P$.

Solution:

$$ALL_{DFA} \stackrel{\text{def}}{=} \{\langle M \rangle \mid \ M \text{ is a DFA and } L(M) = \Sigma^* \ \}$$

Given an DFA M, our task is to find out whether M accepts all strings from Σ^* or not. Note that it is easy to check whether M rejects some string (we can simply search whether a non-accepting state is reachable from the initial state; this takes only polynomial time using e.g. depth first search). If a non-accepting state is reachable in M then surely there is a string not belonging $L(M)$, $L(M) \neq \Sigma^*$ and $\langle M \rangle \notin ALL_{DFA}$; if only accepting states are reachable in the DFA M then $L(M) = \Sigma^*$ and $\langle M \rangle \in ALL_{DFA}$. Hence we just described a polynomial time algorithm for deciding ALL_{TM} and so $ALL_{TM} \in P$.

Exercise 6 (optional but easy)

Show that if L is a regular language then $L \in \text{TIME}(n)$.