- **Please give details of your answer. A direct answer without explanation is not counted.**
- **Your answers must be in English.**
- **Please carefully read problem statements.**
- **During the exam you are not allowed to borrow others' class notes.**
- **Try to work on easier questions first.**

Problem 1 (10 pts)

If we would like to prove

$$f(n) \neq O(g(n)),$$

we need to show the opposite statement of the definition of

$$f(n) = O(g(n)).$$

What is this opposite statement?

Answer

$$\forall c > 0, N \in \mathbb{N}, \exists n \geq N$$

such that

$$f(n) > cg(n).$$

Problem 2 (25 pts)

Consider the following two functions

$$f(n) = \begin{cases} n^3, & \text{if } n \text{ is odd} \\ n^2, & \text{if } n \text{ is even} \end{cases}$$

$$g(n) = \begin{cases} n^3, & \text{if } n \text{ is prime} \\ n^2, & \text{if } n \text{ is composite} \end{cases}$$

Which of the following statements are true?

(a) $f = O(n^2)$

(b) $f = O(n^3)$

(c) $g = O(n^2)$

(d) $g = O(n^3)$

(e) $f = O(g)$

(f) $g = O(f)$

(g) $n^2 = O(f)$

(h) $n^3 = O(f)$

(i) $n^2 = O(g)$

(j) $n^3 = O(g)$

You must prove the result of each sub-problem. If you think the statement is false, you should prove the definition that you wrote for problem 1.

Answer

Statements (b), (d), (f), (g), (i) are true.

(a) For any $c > 0, N \in \mathbb{N}$, we can let

$$n = \text{ an odd number larger than } c \text{ and } N.$$

Then

$$\begin{aligned} &f(n) = n^3 > cn^2 \\ \Rightarrow &f \neq O(n^2). \end{aligned}$$

(b) Let $c = 1$ and $N = 1$.

$$\begin{aligned} &cn^3 \geq n^2 \quad \forall n \geq N \\ \Rightarrow &cn^3 \geq f(n) \quad \forall n \geq N \\ \Rightarrow &f = O(n^3). \end{aligned}$$

(c) For any $c > 0, N \in \mathbb{N}$, we can let

$$n = \text{ a prime number larger than } c \text{ and } N.$$

Then

$$\begin{aligned} &g(n) = n^3 > cn^2 \\ \Rightarrow &g \neq O(n^2). \end{aligned}$$

(d) Let $c = 1$ and $N = 1$.

$$\begin{aligned} &cn^3 \geq n^2 \quad \forall n \geq N \\ \Rightarrow &cn^3 \geq g(n) \quad \forall n \geq N \\ \Rightarrow &g = O(n^3). \end{aligned}$$

(e) For any $c > 0, N \in \mathbb{N}$, we can let

$$n = \text{ an odd and composite number larger than } c \text{ and } N.$$

Then

$$\begin{aligned} &f(n) = n^3 > cg(n) = cn^2 \\ \Rightarrow &f \neq O(g). \end{aligned}$$

(f) Let $c = 1$ and $N = 3$.

Because all of the prime numbers except 2 are odd,

$$cf(n) = cn^3 \geq g(n) = n^3, \text{ when } n \text{ is prime and } n \neq 2$$

and

$$cf(n) = \begin{cases} cn^3 \\ cn^2 \end{cases} \geq g(n) = n^2, \text{ when } n \text{ is composite.}$$

Therefore,

$$\begin{aligned} &cf(n) \geq g(n) \quad \forall n \geq N \\ \Rightarrow &g = O(f). \end{aligned}$$

(g) Let $c = 1$ and $N = 1$.

$$\begin{aligned} &cn^3 \geq n^2 \quad \forall n \geq N \\ \Rightarrow &cf(n) \geq n^2 \quad \forall n \geq N \\ \Rightarrow &n^2 = O(f). \end{aligned}$$

(h) For any $c > 0, N \in \mathbb{N}$, we can let

$$n = \text{ an even number larger than } c \text{ and } N.$$

Then

$$\begin{aligned} &n^3 > cf(n) = cn^2 \\ \Rightarrow &n^3 \neq O(f). \end{aligned}$$

(i) Let $c = 1$ and $N = 1$.

$$\begin{aligned} &cn^3 \geq n^2 \quad \forall n \geq N \\ \Rightarrow &cg(n) \geq n^2 \quad \forall n \geq N \\ \Rightarrow &n^2 = O(g). \end{aligned}$$

(j) For any $c > 0, N \in \mathbb{N}$, we can let

$$n = \text{ a composite number larger than } c \text{ and } N.$$

Then

$$\begin{aligned} &n^3 > cg(n) = cn^2 \\ \Rightarrow &n^3 \neq O(g). \end{aligned}$$

Common Mistakes

1. You cannot have a statement like

$$\text{if } n \text{ is even, then } n^2 = O(n^3).$$

   Note that big-O is for a function of

$$N \to N$$

2. To prove that the statement is false, many write "we can find $n \geq N$ such that ... if n is composite". If you already "find" an n, then n may not be composite.

3. Some write

$$\text{no matter } f(n) = n^2 \text{ or } n^3, \text{ since } n^2 = O(n^2) \text{ and } n^2 = O(n^3), \text{ then } n^2 = O(f).$$

   We didn't really have such a statement. You need to prove it by the definition.

Problem 3 (20 pts)

Assume $g(n) \geq n$. Consider $O(2^{f(n)})$ and $2^{O(f(n))}$. Do we have

$$\begin{aligned} &g(n) = O(2^{f(n)}) \\ \Rightarrow\ &g(n) = 2^{O(f(n))} \end{aligned}$$

or

$$\begin{aligned} &g(n) = 2^{O(f(n))} \\ \Rightarrow\ &g(n) = O(2^{f(n)}) \end{aligned}$$

Answer

(a) We will show that

$$g(n) = O(2^{f(n)})$$
$$\Rightarrow g(n) = 2^{O(f(n))}$$

is true.

Proof:

Since $g(n) \geq n$ and $g(n) = O(2^{f(n)})$, $\exists c_1 > 0, N_1 \geq 1$, s.t.

$$n \leq g(n) \leq c_1 2^{f(n)}, \ \forall n \geq N_1, \tag{1}$$

To prove that $g(n) = 2^{O(f(n))}$, we must prove that

$$g(n) \leq c_1 2^{f(n)} \leq 2^{c_2 f(n)}, \ \forall n \geq N_2. \tag{2}$$

is true for some $c_2 > 0, N_2 \geq N_1$.

Which means we should prove that $\exists c_2 > 1, N_2 \geq N_1$ s.t. $\forall n \geq N_2$,

$$\log c_1 + f(n) \leq c_2 f(n)$$
$$\Rightarrow f(n) \geq \frac{\log c_1}{c_2 - 1}.$$

According to (1), we have

$$\log n \leq \log c_1 + f(n)$$
$$\Rightarrow f(n) \geq \log n - \log c_1$$

Therefore, we should prove that $\exists c_2 > 1, N_2 \geq N_1$, s.t.

$$\log n - \log c_1 \geq \frac{\log c_1}{c_2 - 1}$$
$$\Rightarrow \log n \geq \frac{c_2}{c_2 - 1} \log c_1$$

This is always true when n becomes large.

So given c_1, N_1, $\exists c_2 = 2, N_2 = \max\{N_1, 4c_1\}$, such that (2) is true.

Therefore, $g(n) = 2^{O(f(n))}$.

(b) We will show that

$$g(n) = 2^{O(f(n))}$$
$$\Rightarrow g(n) = O(2^{f(n)})$$

is not true.

Proof:

Let $g(n) = 2^{2n}$, $f(n) = n$. Since

$$g(n) \le 2^{2f(n)} \text{ for } n \ge 1,$$

we have

$$g(n) = 2^{O(f(n))}.$$

Assume $g(n) = O(2^{f(n)})$. There are $c > 0, N \ge 1$, such that $\forall n \ge N$,

$$2^{2n} \le c \cdot 2^n,$$
$$\Rightarrow 2^n \le c, \ \forall n \ge N.$$

However, we can't find a constant c to satisfy $2^n \le c, \ \forall n \ge N$, because the left side of the inequality goes to infinity when $n \to \infty$. Therefore, there is a contradiction.

Common Mistakes

1. In your proof (a), the property $g(n) \ge n$ should be used.
2. Many wrongly have
$$2^a \cdot 2^b = 2^{ab}.$$
3. Some write
$$f(n) \ge \log n - \log c$$
$$\Rightarrow f(n) \text{ is a strictly increasing function.}$$
This may not be true.

Problem 4 (20 pts)

Consider

$$f(n) = \log(1 + e^n)$$
$$g(n) = n$$
$$h(n) = n^2$$

Which of the following statements are true? For small-o, you can directly calculate the limit without getting into the definition of limit.

(a) $f(n) = O(g(n))$

(b) $f(n) = o(g(n))$

(c) $f(n) = O(h(n))$

(d) $f(n) = o(h(n))$

Answer

Statement (a),(c),(d) are true.

(a) $\because f(n) = \log(1 + e^n) < \log(2e^n) = \log(2) + n \leq 2n$, for $n \geq 1$

$\therefore \exists c = 2,\ N = 1,\ s.t.\ \forall n \geq N$

$$f(n) \leq cn$$

$\therefore f(n) = O(n) = O(g(n))$.

(b) According to L'Hospital Rule,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)} = \lim_{n\to\infty} \frac{\frac{e^n}{1+e^n}}{1} = \lim_{n\to\infty} (1 - \frac{1}{1+e^n}) = 1.$$

Therefore, $f(n) \neq o(g(n))$

(c) Since $n^2 \geq n$, from sub-problem (a), $\exists c = 2, N = 1$ such that

$$f(n) \leq cn \leq cn^2,\ \forall n \geq N.$$

Therefore, $f(n) = O(h(n))$

(d) According to L'Hospital Rule,

$$\lim_{n\to\infty} \frac{f(n)}{h(n)} = \lim_{n\to\infty} \frac{\frac{e^n}{1+e^n}}{2n} = 0.$$

Therefore, $f(n) = o(h(n))$

Common Mistakes

1. Some write
$$\lim_{n\to\infty} \frac{\log(1 + e^n)}{n} = \lim_{n\to\infty} \frac{\log(e^n)}{n},$$
but which basic properties of limit give you this?

Problem 5 (5 pts)

Is the following language Turing recognizable?

$$\bar{A}_{TM} = \{\langle M, w\rangle \mid \langle M, w\rangle \notin A_{TM}\},$$

where

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and accepts } w\}$$

Answer

It is not Turing recognizable.

We know that A_{TM} is Turing recognizable but undecidable. (Theorem 4.11)
Assume that $\bar{A}_{TM}$ is Turing recognizable.
Then A_{TM} is both co-Turing recognizable and Turing recognizable. By Theorem 4.22 in the textbook, A_{TM} is decidable.
However, we have proved that A_{TM} is not decidable, so there is a contradiction.

Common Mistakes

1. Some write

$$\text{undecidable} \Rightarrow \text{un-recognizable}.$$

   This is incorrect. A_{TM} is undecidable, but recognizable.

Problem 6 (20 pts)

Consider the following language

$$A = \{\langle R, S\rangle \mid R \text{ and } S \text{ are regular expressions and } L(R) \subseteq L(S)\}$$

Is it decidable?

Answer

Yes, it is decidable.

We first observe that

$$\begin{aligned} L(R) \subseteq L(S) &\text{ iff } \forall w \in L(R), w \in L(S) \\ &\text{ iff } \forall w \in L(R), w \notin \overline{L(S)} \\ &\text{ iff } L(R) \cap \overline{L(S)} = \emptyset \end{aligned}$$

We can construct a DFA C such that

$$L(C) = L(R) \cap \overline{L(S)}$$

with

1. The conversion of regular expressions to an equivalent NFA (procedures in Theorem 1.54).

2. Constructions for proving closure of regular languages under complementation and intersection.

3. Conversion from NFA to DFA.

We can then use Theorem 4.4 to test if $L(C) = L(R) \cap \overline{L(S)}$ is empty.

The following TM F decides A:
On input $\langle R, S \rangle$, where R and S are regular expressions.

1. Construct DFA C as described.

2. Run TM T from Theorem 4.4 on input $\langle C \rangle$.

3. If T accepts, *accept*. If T rejects, *reject*.