CSC165, Winter 2013
Assignment 3

sample solution

1. **Prove or disprove:** $5n^3 - 3n^2 + 2n + 3$ is in $\mathcal{O}(2n^3 - n^2 + n + 1)$.

   Sample solution: The claim is true. It is fairly easy to see that both polynomials have non-negative values when n is a natural number, since the n^3 term dominates the negative n^2 term. What remains to be proved is:

$$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 5n^3 - 3n^2 + 2n + 3 \leq c(2n^3 - n^2 + n + 1)$$

   My strategy in the body of the proof is to over- or under-estimate each polynomial until I am comparing two monomials (one-term polynomials) for simplicity.
   Pick $c = 10$. Then $c \in \mathbb{R}^+$. # to introduce $\exists$
   Pick $B = 1$. Then $B \in \mathbb{N}$. # to introduce $\exists$
   Assume $n \in \mathbb{N}$ and $n \geq B$. # in order to introduce $\forall$ and $\Rightarrow$.
   Then

$$\begin{aligned}
5n^3 - 3n^2 + 2n + 3 &\leq 5n^3 + 2n + 3 \qquad \text{\# add } 5n^3 + 2n + 3 \text{ to both sides of } -3n^2 \leq 0 \\
&\leq 5n^3 + 2n^3 + 3n^3 \\
&\quad \text{\# multiply } 2n \times n^2 \text{ and } 3 \times n^3, n^2, n^3 \geq 1 \text{ since } n \geq B = 1. \\
&= 10n^3 \\
&= cn^3 \qquad \text{\# } c = 10 \\
&\leq c(n^3 + n^3 - n^2) \qquad \text{\#} n^3 - n^2 \geq 0, n \geq 1 \\
&= c(2n^3 - n^2) \\
&\leq c(2n^3 - n^2 + n + 1) \qquad \text{\# add } 2n^3 - n^2 \text{ to both sides of } 0 \leq n + 1
\end{aligned}$$

   Then $5n^3 - 3n^2 + 2n + 3 \leq c(2n^3 - n^2 + n + 1)$. # by transitivity
   Then $\forall n \in \mathbb{N}, n \geq B \Rightarrow 5n^3 - 3n^2 + 2n + 3 \leq c(2n^3 - n^2 + n + 1)$. # introduced $\forall$, $\Rightarrow$.
   Then $\exists c \in \mathbb{R}^+, B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 5n^3 - 3n^2 + 2n + 3 \leq c(2n^3 - n^2 + n + 1)$.
   # introduced $\exists$ twice
   Conclude $5n^3 - 3n^2 + 2n + 3 \in \mathcal{O}(2n^3 - n^2 + n + 1)$. # satisfies definition

2. **Prove or disprove:** $5n^3 - 3n^2 + 2n + 3$ is in $\Omega(2n^3 - n^2 + n + 1)$.

Sample solution The claim is true. Again, both polynomials have non-negative values when n is a natural number, so I need to prove:

$$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 5n^3 - 3n^2 + 2n + 3 \geq c(2n^3 - n^2 + n + 1)$$

Again, I try to reduce the number of terms I have to compare.

Pick $c = 1$. Then $c \in \mathbb{R}^+$. # in order to introduce $\exists$

Pick $B = 3$. Then $B \in \mathbb{N}$. # in order to introduce $\exists$.

Assume $n \in \mathbb{N}$ and $n \geq B$. # in order to introduce $\forall$ and $\Rightarrow$.

Then

$$\begin{aligned}
5n^3 - 3n^2 + 2n + 3 &\geq 5n^3 - 3n^2 \quad \text{\# add } 5n^3 - 3n^2 \text{ to both sides of } 2n + 3 \geq 0 \\
&\geq 4n^3 + n^3 - 3n^2 \quad \text{\# algebra} \\
&\geq 4n^3 \quad \text{\# add } 4n^3 \text{ to both sides of } n^3 - 3n^2 \geq 0, \text{ since } n \geq B \geq 3 \\
&= 4cn^3 = c(2n^3 + n^3 + n^3) \quad \text{\# since } c = 1 \\
&\geq c(2n^3 + n + 1) \quad \text{\# since } n^3 \geq n, n^3 \geq 1 \text{ when } n \geq B \geq 1 \\
&\geq c(2n^3 - n^2 + n + 1) \\
&\quad \text{\# add } 2n^3 + n + 1 \text{ to both sides of } 0 \geq -n^2, \text{ since } n \geq 0
\end{aligned}$$

Then $5n^3 - 3n^2 + 2n + 3 \geq c(2n^3 - n^2 + n + 1)$. # by transitivity

Then $\forall n \in \mathbb{N}, n \geq B \Rightarrow 5n^3 - 3n^2 + 2n + 3 \geq c(2n^3 - n^2 + n + 1)$. # introduced $\forall$ and $\Rightarrow$.

Then $\exists c \in \mathbb{R}^+, B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 5n^3 - 3n^2 + 2n + 3 \geq c(2n^3 - n^2 + n + 1)$.

introduced $\exists$ twice.

Conclude $5n^3 - 3n^2 + 2n + 3$ is in $\Omega(2n^3 - n^2 + n + 1)$. # satisfies the definition

3. **Prove or disprove:** $15 \ln n$ is in $\Omega(n/3)$. **Hint:** Consider using limit techniques from calculus, including l'Hôpital's rule as part of this proof. Please talk to your TA/instructor/Help Centre when needed.

Sample solution: The claim is false. There is no issue about both functions having non-negative values on $\mathbb{N}$ (except $\ln 0$ is undefined), so I must prove the negation of the contition of $\Omega(n/3)$:

$$\neg\left(\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 15 \ln n \geq c(n/3)\right) \Leftrightarrow \forall c \in \mathbb{R}^+, \forall B \in \mathbb{N}, \exists n \in \mathbb{N}, n \geq B \wedge 15 \ln n < c(n/3)$$

I'll use the limit techniques from calculus.

Assume $c \in \mathbb{R}^+$ and assume $B \in \mathbb{N}$.

Then

$$\lim_{n\to\infty} \frac{15 \ln n}{n/3} = \lim_{n\to\infty} \frac{45}{n} = 0 \quad \text{\# L'Hôpital's rule and } \lim_{n\to\infty} 1/n = 0.$$

Then $\forall c' \in \mathbb{R}^+, \exists n' \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n' \Rightarrow |15 \ln n/(n/3)| < c'$ # Definition of limit

Then $\exists n'' \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n'' \Rightarrow |15 \ln n/(n/3)| < c$. # since $c \in \mathbb{R}^+$, by previous line.

Pick $n = \max(n'', B, 1)$. Then $n \in \mathbb{N}$ and $n \geq B$. # by choice of n.

Then $15 \ln n/(n/3) \leq |15 \ln n/(n/3)| < c$. # by choice of n

Then $15 \ln n < c(n/3)$. # multiply both sides by $n/3 > 0$, since $n \geq 1$.

Then $\exists n \in \mathbb{N}, n \geq B \wedge 15 \ln n > c(n/3)$. # introduced $\exists$

Then $\forall c \in \mathbb{R}^+, \forall B \in \mathbb{N}, \exists n \in \mathbb{N}, n \geq B \wedge 15 \ln n < c(n/3)$. # introduced $\forall$ twice.

Conclude $15 \ln n$ is not in $\Omega(n/3)$. # violates the definition

4. **Prove or disprove:** 3^n is in $\mathcal{O}(2^n)$. **Hint:** Consider using the limit techniques of calculus and notice that

$$\lim_{n\to\infty} \frac{3^n}{2^n} = \lim_{n\to\infty} \left(\frac{3}{2}\right)^n$$

Sample solution: The claim is false. Both 3^n and 2^n are positive for natural numbers n, so the issue hinges on proving the negation of $3^n \in \mathcal{O}(2^n)$:

$$\begin{aligned} & \neg(\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 3^n \leq c2^n) \\ \Leftrightarrow \quad & \forall c \in \mathbb{R}^+, \forall B \in \mathbb{N}, \exists n \in \mathbb{N}, n \geq B \wedge 3^n > c2^n \end{aligned}$$

Assume $c \in \mathbb{R}^+$ and $B \in \mathbb{N}$. # in order to introduce $\forall$

Then $\lim_{n\to\infty} 3^n/2^n = \lim_{n\to\infty}(3/2)^n = \infty$.
$\lim_{n\to\infty} x^n = \infty$ if $x > 1$.
Then $\forall \varepsilon \in \mathbb{R}^+, \exists n_\varepsilon \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n_\varepsilon \Rightarrow 3^n/2^n > \varepsilon$.
by definition of $\lim_{n\to\infty} 3^n/2^n = \infty$.
Then $\exists n_c \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n_c \Rightarrow 3^n/2^n > c$. # By previous line, since $c \in \mathbb{R}^+$.
Pick $n = B + n_c$. Then $n \in \mathbb{N}$ and $n \geq B$. # by choice of n.
Then $3^n/2^n > c$. # by choice of n.
Then $3^n > c2^n$. # multiply both sides by positive 2^n.
Then $\exists n \in \mathbb{N}, n \geq B \wedge 3^n > c2^n$. # introduced $\exists$.

Conclude $\forall c \in \mathbb{R}^+, \forall B \in \mathbb{N}, \exists n \in \mathbb{N}, n \geq B \wedge 3^n > c2^n$. # introduced $\forall$ twice.

Then $3^n \notin \mathcal{O}(2^n)$. # violates definition

5. Prove that the function `true_that` below is not computable:

```
def true_that(f, I, n) :
   """
   Return true when the if statement on line n of function f
   executes on input I, and false otherwise.
   """
```

Emulate the technique from the course notes to reduce `halt` to `true_that`

Sample solution: I use a proof by contradiction. The key idea is to use the putative `true_that` to check whether a function reaches a particular line when the way is blocked by a call to `g(i)` — the function that may-or-may-not halt. I use parameter name `g` in my definition of `halt` to avoid confusion with the parameter in `true_that`

Assume, for the sake of contradiction, that `true_that` is computable.

Then, assuming the body of the definition of `true_that` is filled in, the following python code is executable:

```
def true_that(f, I, n) :
   """ Return True iff the statement on line n of function f
       executes on input I.
   """
   # implementation omitted...

def halt(g,i) :
   def P(x) : # ignore parameter x
      g(i) # execution passes this line iff g halts
      if True : return "whoohoo!"

   return true_that(P, 7, 2)
```

Then the `if` statement on line 2 of function `P` executes iff `g(i)` halts.
Then `true_that(P, 7, 2)` returns `True` if `g(i)` halts, `False` otherwise.
Then `halt(g,i)` returns `True` if `g(i)` halts, `False` otherwise. This is the specification of `halt(g,i)` which we proved in class to be non-computable.
Contradiction!

The assumption that `true_that` is computable lead to a contradiction. Therefore the assumption is false, and `true_that` is non-computable.

I assume that the call to `g(i)` is wrapped in an appropriate `try/catch` clause, so that execution passes to the following line unless `g(i)` has an infinite loop. I don't actually show the `try/catch` clause because it clutters things up.