COMP50001: Algorithm Design & Analysis

Sheet 1 (Week 2)

Exercise 1.1

Given the following function concatenating two lists,

$$
\begin{array}{l}
(\mathbin{+\!\!+}) :: [\mathit{Int}] \rightarrow [\mathit{Int}] \rightarrow [\mathit{Int}] \\
[\,] \qquad \mathbin{+\!\!+}\ ys = ys \\
(x : xs) \mathbin{+\!\!+} ys = x : (xs \mathbin{+\!\!+} ys)
\end{array}
$$

with a recurrence relation $T(n, m)$, approximate the time it takes to compute $xs \mathbin{+\!\!+} ys$ for any list xs of length n and ys of length m.

Exercise 1.2

Consider an alternative strict time analysis function T', defined to be the same as T, except that T' is refined to have cost 1 instead 0 on variables, constants and primitive functions, i.e.

$$
\begin{aligned}
T'(x) &= 1 \\
T'(k) &= 1 \\
T'(f)\ x_1\ \cdots x_n &= 1
\end{aligned}
$$

Compute $T'(\mathit{length}\ xs)$ in terms of $T'(\mathit{length}\ (\mathit{tail}\ xs))$.

Exercise 1.3

Compute the strict running time $T(\mathit{length}\ \ (\mathit{insert}\ \ x\ \ xs))$ using the composition rule.

Exercise 1.4

Pattern matching can be added to the expression language e as follows:

$$e ::= \cdots \mid \textbf{case}\ e\ \textbf{of}\ [\,] \rightarrow e; (x : xs) \rightarrow e$$

Give an appropriate definition of $T(\textbf{case}\ e_1\ \textbf{of}\ [\,] \rightarrow e_2; (x : xs) \rightarrow e_3)$ for strict time analysis.

Exercise 1.5

(ADWH, p39, Exercise 2.3) Prove formally that $(n + 1)^2 \in \Theta(n^2)$ by exhibiting the necessary constants.

Exercise 1.6

(ADWH, p39, Exercise 2.5) Justify whether each of the following is true or false:

1. $2n^2 + 3n \in \Theta(n^2)$
2. $2n^2 + 3n \in O(n^3)$

3. $n \log n \in O(n\sqrt{n})$
4. $n + \sqrt{n} \in O(\sqrt{n} \log n)$
5. $2^{\log n} \in O(n)$

Exercise 1.7

Show formally that $o(g(n))$ is a proper subset of $O(g(n))$ for any function g using their definitions.

Exercise 1.8

Explain why there is no definition $\theta(g(n))$ that corresponds to $\Theta(g(n))$ even though there is $o(g(n))$ corresponding to $O(g(n))$ and $\omega(g(n))$ corresponding to $\Omega(g(n))$.

Exercise 1.1

Given the following function concatenating two lists,

$$
\begin{aligned}
&(+\!\!+) :: [Int] \to [Int] \to [Int] \\
&[\,] \quad\quad\ \ +\!\!+\ ys = ys \\
&(x : xs) +\!\!+\ ys = x : (xs +\!\!+\ ys)
\end{aligned}
$$

with a recurrence relation $T(n, m)$, approximate the time it takes to compute $xs +\!\!+\ ys$ for any list xs of length n and ys of length m.

$T(+\!\!+)\ xs\ ys = T(m, n)$ where m = length xs
n = length ys.

$T(+\!\!+)\ [\,]\ ys = 1$
$T(+\!\!+)\ (x:xs)\ ys = 1 + T(+\!\!+)\ xs\ ys$

$$
\begin{aligned}
T(0, n) &= 1 \\
T(m, n) &= 1 + T(m-1, n) \\
&= 1 + 1 + T(m-2, n) \\
&= \cdots \\
&= k + T(m-k, n) \quad (\text{for any } 0 \le k \le m) \\
&= \\
&= m + T(0, n) \\
&= m + 1
\end{aligned}
$$

Exercise 1.2

Consider an alternative strict time analysis function T', defined to be the same as T, except that T' is refined to have cost 1 instead 0 on variables, constants and primitive functions, i.e.

$$T'(x) = 1$$
$$T'(k) = 1$$
$$T'(f)\ x_1\ \cdots x_n = 1$$

Compute $T'(\mathit{length}\ \mathit{xs})$ in terms of $T'(\mathit{length}\ (\mathit{tail}\ \mathit{xs}))$.

$$
\begin{aligned}
& T'(\text{length } xs) \\
= {} & T'(\text{length})\ xs + T'(xs) \\
= {} & 1 + T'(\text{if null } xs \text{ then } 0 \text{ else } 1 + \text{length } (\text{tail } xs)) + 1 \\
= {} & 2 + T'(\text{null } xs) + \text{if null } xs \text{ then} \\
& \qquad T'(0) \text{ else } T'(1 + \text{length } (\text{tail } xs)) \\
= {} & 2 + T'(\text{null})\ xs + T'(xs) + \text{if null } xs \text{ then } T'(0) \\
& \qquad \text{else } T'(1 + \text{length } (\text{tail } xs)) \\
= {} & 4 + \text{if null } xs \text{ then } T'(0) \text{ else} \\
& \qquad T'(1 + \text{length } (\text{tail } xs)) \\
= {} & \ldots \\
= {} & 4 + \text{if null } xs \text{ then } 1 \\
& \qquad \text{else } 4 + T'(\text{length})\ (\text{tail } xs)
\end{aligned}
$$

Exercise 1.3

Compute the strict running time $T(\mathit{length}\ (\mathit{insert}\ x\ xs))$ using the composition rule.

$$T(\mathit{length}\ (\mathit{insert}\ x\ xs))$$
$$=$$
$$T(\mathit{length})\ (\mathit{insert}\ x\ xs) + T(\mathit{insert})\ x\ xs$$

Exercise 1.4

Pattern matching can be added to the expression language e as follows:

$$e ::= \cdots \mid \textbf{case}\ e\ \textbf{of}\ [\,] \to e; (x : xs) \to e$$

Give an appropriate definition of $T(\textbf{case}\ e_1\ \textbf{of}\ [\,] \to e_2; (x : xs) \to e_3)$ for strict time analysis.

$$T\left(\begin{array}{l}\text{case } e \text{ of } [\,] \to e_1 \\ \qquad\quad (x:xs) \to e_2\end{array}\right) =$$

$$\begin{array}{l}T(e) + \text{case } e \text{ of} \\ \qquad [\,] \to T(e_1) \\ \qquad (x:xs) \to T(e_2)\end{array}$$

Exercise 1.5

(ADWH, p39, Exercise 2.3) Prove formally that $(n+1)^2 \in \Theta(n^2)$ by exhibiting the necessary constants.

$$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$$

$$O(g(n)) = \{ f \mid \exists \delta > 0. \exists n_0 > 0. \forall n > n_0. f(n) \leq \delta \cdot g(n) \}$$
$$\Omega(g(n)) = \{ f \mid \exists \delta > 0. \exists n_0 > 0. \forall n \geq n_0. f(n) \geq \delta \cdot g(n) \}$$

$$f(n) = (n+1)^2$$
$$g(n) = n^2$$

$$f(n) \leq \delta \cdot g(n)$$

$$(n+1)^2 \leq \delta n^2$$
$$\Leftrightarrow$$
$$n^2 + 2n + 1 \leq \delta n^2$$
$$\Leftrightarrow$$
$$0 \leq (\delta - 1) n^2 - 2n - 1$$
$$\Leftrightarrow \quad \{ \text{assume } \delta = 4 \}$$
$$0 \leq (3n+1)(n-1)$$

Exercise 1.6

(ADWH, p39, Exercise 2.5) Justify whether each of the following is true or false:

1. $2n^2 + 3n \in \Theta(n^2)$ ✓
2. $2n^2 + 3n \in O(n^3)$ ✓

3. $n \log n \in O(n\sqrt{n})$
4. $n + \sqrt{n} \in O(\sqrt{n} \log n)$
5. $2^{\log n} \in O(n)$

Exercise 1.7

Show formally that $o(g(n))$ is a proper subset of $O(g(n))$ for any function g using their definitions.

$\forall \delta > 0 . \exists n_0 > 0 . \forall n > n_0 . f(n) < \delta g(n)$ $\quad o(g(n))$

pick $\delta = 1$

$\Downarrow$

$\exists \delta > 0 . \exists n_0 > 0 . \forall n > n_0 . f(n) \leq \delta g(n)$ $\quad O(g(n))$

Pick $\delta = n_0 = 1$ in def. $O(g(n))$

$\forall n > n_0 . g(n) \leq g(n)$ $\quad g \in O(g(n))$

but $g \notin o(g(n))$

$$\neg(\forall \delta > 0 . \exists n_0 > 0 . \forall n > n_0 . f(n) < \delta g(n))$$

$$\Leftrightarrow$$

$$\exists \delta > 0 . \forall n_0 > 0 . \exists n > n_0 . f(n) \geq \delta g(n)$$

Choose $\delta = 1$

Choose $n = n_0 + 1$

$$g(n) \geq g(n)$$

Therefore $g(n) \notin o(g(n))$

Exercise 1.8

Explain why there is no definition $\theta(g(n))$ that corresponds to $\Theta(g(n))$ even though there is $o(g(n))$ corresponding to $O(g(n))$ and $\omega(g(n))$ corresponding to $\Omega(g(n))$.

$$o(g(n)) = O(g(n)) \setminus \Theta(g(n))$$

$$\omega(g(n)) = \Omega(g(n)) \setminus \Theta(g(n))$$

$$\theta(g(n)) = \Theta(g(n)) \setminus \Theta(g(n))$$

$$= \emptyset$$