Problem Set 1

Jaan Tollander de Balsch

February 21, 2020

Papadimitriou (1994), chapters: 1, 2.1-2.5

H1.1

i)

We say that $g(n)$ dominates $f(n)$ if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \to 0.$$

Let prove $(\log_2 n)^s = O(n^r)$ for $s, r > 1$, that is n^r dominates $(\log_2 n)^s$.

$$\lim_{n\to\infty} \frac{(\log_2 n)^s}{n^r} = \left(\lim_{n\to\infty} \frac{\log_2 n}{n^{r/s}} \right)^s$$

Using L'hopital's rule

$$\left(\lim_{n\to\infty} \frac{1}{(r/s) n^{r/s} \log(2)} \right)^s \to 0$$

From the above limit, we see that $n^r \neq O((\log_2 n)^s)$, since n^r dominates $(\log_2 n)^s$.

ii)

Upper bound

$$\begin{aligned}\log_2 n! &= \sum_{i=1}^{n} \log_2 i \\ &\leq \sum_{i=1}^{n} \log_2 n \\ &= n \log_2 n.\end{aligned}$$

All the terms $\log_2 i$ are smaller or equal to $\log_2 n$.

Lower bound (assuming even n)

$$\begin{aligned}\log_2 n! &= \sum_{i=1}^{n} \log_2 i \\ &= \sum_{i=1}^{n/2} \log_2 i + \sum_{i=n/2}^{n} \log_2 i \\ &\geq (n/2) \log_2 (n/2) \\ &= (n/2)(\log_2 n - 1).\end{aligned}$$

Half of the terms $\log_2 i$ are larger of equal to $\log_2(n/2)$.

Therefore, the tight bound is

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

H1.2

We define our Turing machine as $M = (K, \Sigma, \delta, s)$ where the set of states is $K = \{s, q_1, q_2, q_3, q_4\}$, the set of symbols is $\Sigma = \{1, \sqcup, \triangleright\}$ and the following transition function δ:

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$	Explanation
s	1	$(q_1, 1, \rightarrow)$	Move head to next 1
s	$\sqcup$	$(h, 1, \rightarrow)$	Add 1 and halt
s	$\triangleright$	$(s, \triangleright, \rightarrow)$	
q_1	1	$(q_2, 1, \rightarrow)$	Move head to next one
q_1	$\sqcup$	$(h, 1, \rightarrow)$	Add 1 and halt
q_2	1	$(q_2, 1, \rightarrow)$	Move cursor to the end of the string of 1's
q_2	$\sqcup$	$(q_3, \sqcup, \leftarrow)$	Reached end of the string
q_3	1	$(q_2, \sqcup, \leftarrow)$	Remove first 1
q_4	1	$(h, \sqcup, \leftarrow)$	Remove second 1 and halt

Unreachable states are left out for simplicity.

The execution of the Turing machine for $f(\epsilon)$

$$s, \rhd \underline{\sqcup} \to h, \rhd 1\underline{\sqcup}$$

The execution of the Turing machine for $f(11)$

$$\begin{aligned}
&s, \rhd \underline{1}1\sqcup \to \\
&q_1, \rhd 1\underline{1}\sqcup \to \\
&q_2, \rhd 11\underline{\sqcup} \to \\
&q_3, \rhd 1\underline{1}\sqcup \to \\
&q_4, \rhd \underline{1} \sqcup \sqcup \to \\
&h, \underline{\rhd} \sqcup \sqcup\sqcup
\end{aligned}$$

H1.3

We define a binary alphabet $\Sigma = \{0, 1\}$.

i)

If language $L \subseteq \Sigma^*$ is decidable by a Turing machine M then

- If $x \in L$, $M(x) = \textit{yes}$ and
- If $x \notin L$, $M(x) = \textit{no}$

Then, the complement $\bar{L} = \Sigma^* \setminus L = \{x \in \Sigma^* \mid x \notin L\}$ is decidable by a Turing machine M'

- If $x \in \bar{L}$, $M'(x) = \textit{yes}$ and
- If $x \notin \bar{L}$, $M'(x) = \textit{no}$

By definition we have

- $x \in \bar{L}$ implies $x \notin L$ and $M(x) = \textit{no}$
- $x \notin \bar{L}$ implies $x \in L$ and $M(x) = \textit{yes}$

Turing machine M' decides *yes* when M decides *no* and vice versa.

If Turing machine M accepts L if for every string $x \in \Sigma - \{\sqcup\}^*$

- If $x \in L$ then $M(x) = \textit{yes}$
- But, if $x \notin L$, $M(x) = \nearrow$.

The complement language $\bar{L}$ is not accepted by a Turing machine M', because if $x \notin L$ which implies $x \in \bar{L}$ and $M(x) = \nearrow$. Therefore, Turing machine M may not halt on this input, which implies that Turing machine M' may also not halt.

ii)

If L is semidecidable it implies $x \in L$, then $M_1(x) = yes$.

If $\bar{L}$ is semidecidable it implies $x \in \bar{L}$, then $M_2(x) = yes$.

Since $x \in \bar{L}$ implies $x \notin L$, all inputs $x \in L$ and $x \notin L$ are decided by some Turing machine, therefore, L is decidable.

References

Papadimitriou, C.H., 1994. *Computational complexity.* Addison-Wesley.pp.I–XV, 1–523.