COMP9020 Problem Set 8

Kai Engelhardt

September 29, 2016

1 State Machines

This series of exercises deals with the fast exponentiation algorithm described in section 6.3.1 of [LLM16].

Exercise 1 Translate the program pseudo code on page 140 of the textbook into a functioning python or C program. Run a few tests.

Exercise 2 Translate your program from the previous exercise into a transition diagram P.

Exercise 3 Define precise pre- and postconditions, ϕ and ψ, to capture the desired functionality.

Exercise 4 Find an inductive assertion network to prove $\{\phi\}\, P\, \{\psi\}$.

The next two exercise assume more than we covered in class, but the material is actually contained in the unused parts of slides 8a.

Exercise 5 Find a family of ranking functions to prove termination.

Exercise 6 Show the absence of deadlock.

2 Order of Growth

These are taken or derived from [AU95].

Exercise 7 Let $T(n)$ be defined[1] by the recurrence

$$T(n) = T(n-1) + g(n) \quad \text{, for } n > 1$$

Prove by induction on i that if $1 \leq i < n$, then

$$T(n) = T(n-i) + \sum_{j=0}^{i-1} g(n-j)$$

[1] "defined" should be in quotes because we don't state what $T(1)$ is.

Exercise 8 Suppose we have a recurrence of the form[2]

$$T(1) = a$$
$$T(n) = T(n/2) + g(n) \quad \text{, for } n > 1 \text{ a power of 2}$$

Give tight big-oh upper bounds on the solution if $g(n)$ is

1. n^2
2. $2n$
3. $g(n) = 10$
4. $n \log n$
5. 2^n

Exercise 9 Show the first part of the master theorem, that is, if

$$T(1) = a$$
$$T(n) = T(n-1) + n^k \quad \text{, for } n > 1$$

then $T(n)$ is $\mathcal{O}(n^{k+1})$. You may assume $k \geq 0$. Also, show that this is the tightest simple big-oh upper bound, that is, that $T(n)$ is not $\mathcal{O}(n^m)$ if $m < k+1$. *Hint:* Expand $T(n)$ in terms of $T(n-i)$, for $i = 1, 2, \ldots$, to get the upper bound. For the lower bound, show that $T(n)$ is at least cn^{k+1} for some particular $c > 0$.

Exercise 10 Solve the following recurrences, each of which has $T(1) = a$

1. $T(n) = 3T(n/2) + n^2$, for $n > 1$ a power of 2
2. $T(n) = 10T(n/3) + n^2$, for $n > 1$ a power of 3
3. $T(n) = 16T(n/4) + n^2$, for $n > 1$ a power of 4

Exercise 11 Solve the recurrence

$$T(1) = 1$$
$$T(n) = 3^n T(n/2) \quad \text{, for } n \geq 1 \text{ a power of 2}$$

References

[AU95] Alfred V. Aho and Jeffrey D. Ullman. *Foundations of Computer Science C Edition.* Computer Science Press, 1995. Out of print, available online at http://i.stanford.edu/~ullman/focs.html.

[LLM16] Eric Lehman, F. Thomson Leighton, and Albert R. Meyer. Mathematics for computer science. Available at https://courses.csail.mit.edu/6.042/spring16/mcs.pdf; check https://courses.csail.mit.edu/6.042 for newer versions, 2016.

[2]These are single term divide-and-conquer recurrences, not too common in practice (unfortunately).