School of Computing and Information Systems
COMP30026 Models of Computation Tutorial Week 1

5–7 August 2020

Introduction to tutes

Welcome to the first Models of Computation tutorial for 2020. We hope you will enjoy the tutorials, and also that you will contribute to making them enjoyable for your class mates. We have the luxury of having 2-hour tutes and we want to make the most of them, even if we cannot have face-to-face collaboration. Your tutor will explain how they would like to run the tutes. We are keen to preserve as much of the interaction and collaboration as we can, from the on-campus tutes that we are used to. Your willingness to engage online is critical, of course, and in particular we expect you to be active in tutes and in smaller Zoom break-out rooms.

Plan for the first week

This week the lecture will introduce some themes that come up later in the subject. Then we will plunge into a non-trivial Haskell program. The aim is to show that Haskell is useful for certain tasks, in particular tasks that involve symbolic computation. At this point many language details will be a mystery to you, and that's fine. I hope to convince you that, with Haskell, it is possible to build useful tools with little effort. In the next weeks, your job is to use the available Grok modules to learn basic Haskell. Grok is a good platform for that, certainly better than lectures. You should have been given Grok access by now. (If you are enrolling late, it may take us a day or two to get you access.)

The exercises

1. Introduce yourself to your classmates and tell them a bit about yourself. Your tutor may place you in Zoom break-out rooms before or after this exercise. You may also want to discuss any logistic/technological problems that you have (or think you may have) and share experiences from semester 1—how do we make tutorials work for all of us?

2. Visit Grok and work through Haskell Module 1, "Functions and Recursion". If you are fast, you may even get started on Module 2 on lists. Support the other students in your group. (We don't plan to use tutorial time on Grok from Week 2 onwards; this is just to make sure you get started.)

3. Consider an n by m grid. We want to find out how many different paths there are that one can travel along, starting from the bottom left and ending in the top right cell, given that *one can only move up or right*. Write a Haskell function

   ```
   paths :: Integer -> Integer -> Integer
   ```

   so that `paths m n` returns the number of such paths. (We use `Integer`s as the numbers can get quite large.)

4. In Lecture 1 we play a word rewriting game. Here is an example of a similar kind of game, the so-called Post's Correspondence Problem. We have a set of "dominoes" such as

$$\left\{\left[\frac{\texttt{b}}{\texttt{ca}}\right], \left[\frac{\texttt{a}}{\texttt{ab}}\right], \left[\frac{\texttt{ca}}{\texttt{a}}\right], \left[\frac{\texttt{abc}}{\texttt{c}}\right]\right\}$$

The dominoes use a small alphabet, say the letters `a`, `b`, and `c`. Each domino has a string written in the upper half, and one in the lower half. The given set of dominoes is always finite, but there is an unbounded supply of each domino, that is, we can use it many times. The goal is to find a sequence of dominoes which, when laid side by side, spell out identical non-empty strings across the top and the bottom (or alternatively, determine that no solution is possible). For example, the set given above has a solution, namely

$$\left[\frac{\texttt{a}}{\texttt{ab}}\right]\left[\frac{\texttt{b}}{\texttt{ca}}\right]\left[\frac{\texttt{ca}}{\texttt{a}}\right]\left[\frac{\texttt{a}}{\texttt{ab}}\right]\left[\frac{\texttt{abc}}{\texttt{c}}\right]$$

   (a) Can you find a solution to $\left\{\left[\frac{\texttt{baa}}{\texttt{abaaa}}\right], \left[\frac{\texttt{aaa}}{\texttt{aa}}\right]\right\}$?

   (b) How about $\left\{\left[\frac{\texttt{a}}{\texttt{cb}}\right], \left[\frac{\texttt{bc}}{\texttt{ba}}\right], \left[\frac{\texttt{c}}{\texttt{aa}}\right], \left[\frac{\texttt{abc}}{\texttt{c}}\right]\right\}$?

   (c) How about $\left\{\left[\frac{\texttt{ab}}{\texttt{aba}}\right], \left[\frac{\texttt{bba}}{\texttt{aa}}\right], \left[\frac{\texttt{aba}}{\texttt{bab}}\right]\right\}$?

5. What is the general algorithm that solves Post's Correspondence Problem?