15-312 Foundations of Programming Languages
Recitation 12: Concurrency

Daniel Spoonhower
spoons+@cs

November 19, 2003

1 Finding the Maximum with Futures

Say we're interested in finding the maximum value found in a tree (for the sake of example, we'll ignore negative numbers). The following MinML code will do the job (assuming a definition of `tree` as in Assignment 8).

```
fun max (t : tree) : int =>
  case t
    of Leaf => 0
     | Node(x, l, r) =>
          let lmax = max l in
            let rmax = max r in
              if x < lmax then if lmax < rmax then rmax else lmax fi
              else if x < rmax then rmax else x fi fi
            end
          end;
```

How long would it take to run this code (i.e. how many transitions would the E machine make)?

How could we parallelize this code? (Give a few examples.) How long would it take to run in parallel? In this case, we are not just interested in the total number of E machine steps, but in the number of clock ticks that elapse. How many processors would be used in each parallel version?

2 Sequential Processes and Simulation

More on Vending Machines

In class, we discussed a sequential process that described a vending machine; here's we will explore another example. Recall our definition of process expressions.

$$
\begin{array}{llll}
\textit{process identifiers} & A & & \\
\textit{names} & \alpha & & \\
\textit{process expressions} & P & ::= & A \mid \alpha.P + \ldots + \alpha.P
\end{array}
$$

Say a new vending machine was installed on campus, but that the machine had no labels: it just has a coin slot and two buttons, $\overline{\mathsf{red}}$ and $\overline{\mathsf{blue}}$. What could we do to determine how the machine works?

Let's try various sequences of pressing buttons and adding coins, then and see what happens. Given these sequences, we might construct a description of the machine that looks like this:

$$
\begin{array}{lcl}
A & \overset{\mathsf{def}}{=} & \mathsf{quarter}.B + \mathsf{dime}.C \\
B & \overset{\mathsf{def}}{=} & \overline{\mathsf{red}}.A + \overline{\mathsf{red}} + A \\
C & \overset{\mathsf{def}}{=} & \overline{\mathsf{blue}}.A + \mathsf{dime}.D + \mathsf{quarter}.B \\
D & \overset{\mathsf{def}}{=} & \mathsf{dime}.B + \mathsf{quarter}.B
\end{array}
$$

(What would the state diagram for this machine look like? Also, give a regular expression that gives valid sequences of input and output names for this machine.)

While this seems like a reasonable specification for a vending machine, what if the machine we received was not up to spec? What if we received a broken machine with the following states (instead of C and E above)?

$$
\begin{array}{lcl}
C' & \overset{\mathsf{def}}{=} & \overline{\mathsf{blue}}.A + \mathsf{dime}.D + \mathsf{quarter}.B + \mathsf{quarter}.E' \\
E' & \overset{\mathsf{def}}{=} & \cdot
\end{array}
$$

Could we necessary tell the difference between the broken machine and one that was functioning properly? How long would we have to wait before we realized that we were using a broken machine?

(Now turn the situation around: your friend comes up to you and says, "I've got this broken vending machine. Will you help me fix it?" You are skeptical, however: the machine seems fine to you. Could your (misguided) friend be trying to fool you? How could you tell if the machine really was broken?)