Homework 5

EE 290n - Advanced Topics in Systems Theory

Edward A. Lee

1. Suppose V is some set and $S = V^{**}$ is the set of finite and infinite sequences of elements of V. This exercise explores some of the properties of the CPO S^n with the pointwise prefix order, for some non-negative integer n.

   (a) Show that any two elements $a, b \in S^n$ that have an upper bound have a least upper bound.

   (b) Let $U \subset S^n$ be such that no two distinct elements of U are joinable. Prove that for all $s \in S^n$ there is at most one $u \in U$ such that $u \sqsubseteq s$.

   (c) Given $s \in S^n$, suppose that $Q(s) \subset S^n$ is a joinable set where for all $q \in Q(s)$, $q \sqsubseteq s$. Then show that there is an s' such that $s = (\bigvee Q(s)).s'$.

   Solution.

   (a) Let c be an upper bound of a and b. The $a \sqsubseteq c$ and $b \sqsubseteq c$. Under the pointwise prefix order, this implies that $\pi_i(a) \sqsubseteq \pi_i(c)$ and $\pi_i(b) \sqsubseteq \pi_i(c)$ for each $i \in \{1, \cdots, n\}$. Since $\pi_i(a)$ and $\pi_i(b)$ are ordinary sequences, if they are both prefixes of the same sequence $\pi_i(c)$, then it must be that either $\pi_i(a) \sqsubseteq \pi_i(b)$ or $\pi_i(b) \sqsubseteq \pi_i(a)$. We can construct a $d \in S^n$ where $\pi_i(d)$ is defined to be $\pi_i(b)$ if $\pi_i(a) \sqsubseteq \pi_i(b)$, and is defined to be $\pi_i(a)$ otherwise, for each $i \in \{1, \cdots, n\}$. Then clearly d is an upper bound of a and b, and moreover, $\pi_i(d) \sqsubseteq \pi_i(c)$ for each $i \in \{1, \cdots, n\}$, so d is a least upper bound under the pointwise prefix order.

   (b) Note first that the theorem is trivially true for $n = 0$. For $n > 0$, assume to the contrary that you have two distinct $u, u' \in U$ such that $u \sqsubseteq s$ and $u' \sqsubseteq s$ for some $s \in S^n$. Then s is an upper bound for $\{u, u'\}$. From part (a), $\{u, u'\}$ has a least upper bound, and hence u and u' are joinable, contradicting the assumption that no two distinct elements of U are joinable.

   (c) It is sufficient to show that $\bigvee Q(s) \sqsubseteq s$. Note first this is trivially true for $n = 0$, so we henceforth assume $n > 0$. Consider each dimension $i \in \{1, \cdots, n\}$. For each such i, there is a $q \in Q(s)$ such that $\pi_i(\bigvee Q(s)) = \pi_i(q)$. We know that $\pi_i(q) \sqsubseteq \pi_i(s)$, so we conclude that $\pi_i(\bigvee Q(s)) \sqsubseteq \pi_i(s)$ for each such i. Hence, $\bigvee Q(s) \sqsubseteq s$.

   $\square$

2. Consider the model shown in figure 1. Assume that data types are all $V = \{0, 1\}$. Assume f is a dataflow actor that implements an identity function and that Const is an actor that produces an infinite sequence $(0, 0, 0, \cdots)$. Obviously, the overall output of this model should be this same infinite sequence. The box labeled g indicates a composite actor. Find firing rules and firing function g for the composite actor to satisfy conditions 1 and 3 covered in class. Note that the composite actor has one input and two outputs.

   Solution. Let $U = \{(0), (1), \perp\}$ be the set of firing rules. Note that subsets $\{(0), \perp\}$ and $\{(1), \perp\}$ are joinable. Notice that the greatest lower bound of each of these sets is $\perp$, so the first part of rule 3 is satisfied. Let g be defined so that

$$g((0)) = ((0), \perp) \quad (1)$$
$$g((1)) = ((1), \perp) \quad (2)$$
$$g(\perp) = (\perp, (0)). \quad (3)$$

Note that this firing function yields, as desired, and infinite sequence $(0,0,0,\cdots)$. Note now that if $u = (0)$ and $u' = \perp$, then

$$g(u).g(u') = g(u').g(u).$$

The same is true if $u = (1)$ and $u' = \perp$, so the rest of rule 3 is satisfied. □

3. **Extra credit**. In theory, dataflow models with only boolean data types, switch, select, and logic functions are Turing complete. A simple function that should be implementable, but is not easy to implement using such primitives, is one that, given a sequence $(v_1, v_2, \cdots)$ produces a sequence where every block of five inputs is reversed, yielding

$$(v_5, v_4, v_3, v_2, v_1, v_{10}, v_9, \cdots).$$

I am looking for elegant dataflow models using the dynamic dataflow (DDF) director in Ptolemy II (under ExperimentalDirectors). An extension of this would use integer data types and given three sequences $v = (v_1, v_2, \cdots)$, $(n_1, n_2, \cdots)$, and $(m_1, m_2, \cdots)$ that would behave as follows: for every integer $i > 0$, it would consume n_i tokens from v and push them onto a stack, then pop m_i tokens from the stack (reversing their order) and produce them on the output. I am looking for an elegant dataflow model that performs this function. Note that I do not have a solution to this problem.

Solution. I got several solutions, all with nice ideas. The one I like the best is given by Xiaojun Liu. It can be found at:

```
http://embedded.eecs.berkeley.edu/concurrency/homework
     /Dataflow/XiaojunLiu_ExtraCredit.xml
```

□

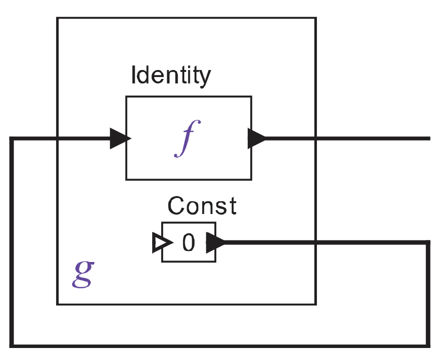

Figure 1: A model.