Concurrent Models of Computation for Embedded Software

Edward A. Lee

Professor, UC Berkeley

EECS 290n – Advanced Topics in Systems Theory

Fall, 2004

Copyright © 2004, Edward A. Lee, All rights reserved

Lecture 12: Tags and Discrete Signals

Tags, Time Stamps, and Events

The DE Tag system

- $T = R \times N$, real and natural numbers.
- Lexicographic order using natural ordering of R and N.

This is a totally ordered set.

- *Event:* a pair $e = (t, v) \in T \times V$ where V is a set of *values* and $t = (\tau, n)$ is a tag.
- *Time stamp:* of an event e is $\tau = \pi_1(\pi_1(e))$ (projection)
- *Index*: of an event e is $n = \pi_2(\pi_1(e))$ allowing distinct events with the same time stamp.

Note that events in a signal are totally ordered.

Signals

Signal: a set s of events with distinct tags.

Equivalently: a signal s is a partial function

$$s : T \to V$$

Tag Sets

A signal: $s = \{ e_1, e_2, \ldots \} = \{ (t_1, v_1), (t_2, v_2), \ldots \}$

Its tags: $\hat{\pi}_1(s) = \{ t_1, t_2, \ldots \}$

A *system:* $S = \{ s_1, s_2, \ldots \}$ is a set of signals.

Its tags: $\hat{\hat{\pi}}_1(S) = \pi_1(s_1) \cup \pi_1(s_2) \cup \ldots$

Discrete Signals

A signal s is *discrete* if there is an *order embedding* from its tag set $\pi_1(s)$ to the integers (under their usual order).

A system S (a set of signals) is *discrete* if there is an *order embedding* from its tag set $\pi_1(s)$ to the integers (under their usual order).

Terminology: Order Embedding

Given two posets A and B, an *order embedding* is a function $f : A \to B$ such that for all $a, a' \in A$,

$$a \le a' \Leftrightarrow f(a) \le f(a')$$

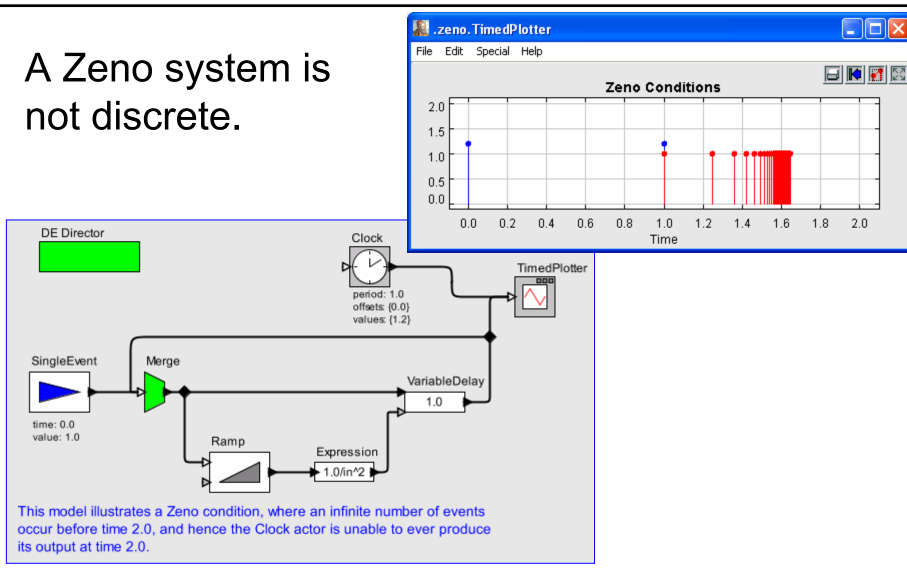
Exercise: Show that if A and B are two posets, and $f : A \to B$ is an order embedding, then f is *one-to-one.*

Examples

1. Suppose we have a signal s whose tag set is
   $$\{(\tau, 0) \mid \tau \in R\}$$
   (this is a *continuous-time* signal). This signal is not discrete.

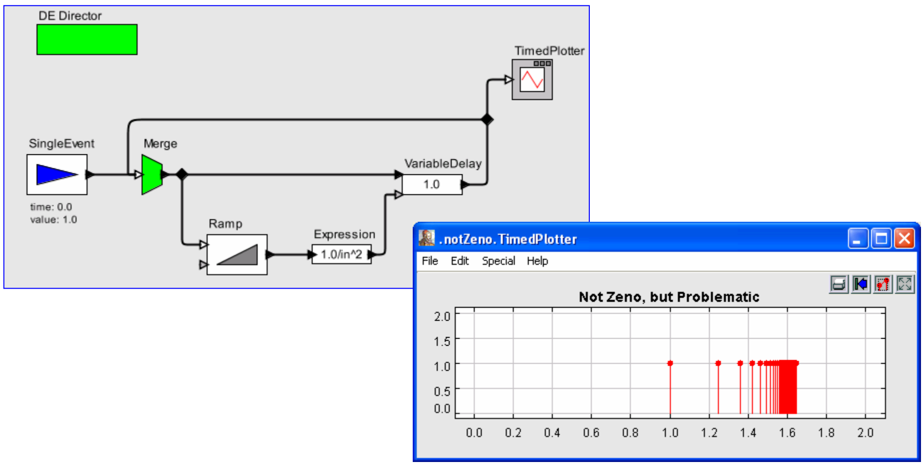
2. Suppose we have a signal s whose tag set is
   $$\{(\tau, 0) \mid \tau \in Rationals\}$$
   This signal is also not discrete.

A Zeno system is not discrete.



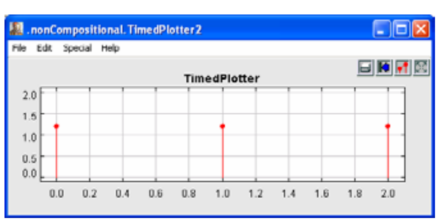
The tag set here includes $\{0, 1, 2, \ldots\}$ and $\{1, 1.25, 1.36, 1.42, \ldots\}$.
Exercise: Prove that this system is not discrete.

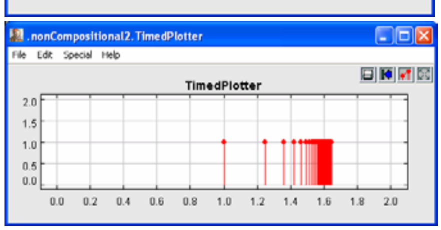
Is the following system discrete?



Discreteness is Not a Compositional Property

Given two discrete signals s, s' it is not necessarily true that $S = \{ s, s' \}$ is a discrete system.

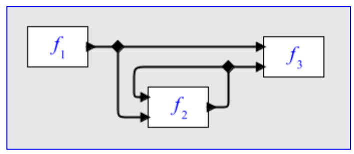




Putting these two signals in the same model creates a Zeno condition.

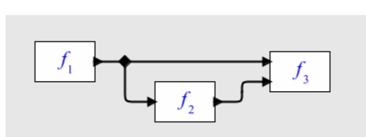
Question 1:

Can we find necessary and/or sufficient conditions to avoid Zeno systems?



Question 2:

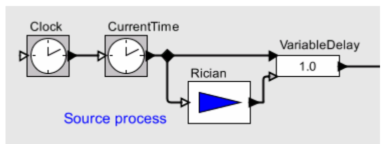
In the following model, if f_2 has no delay, should f_3 see two simultaneous input events with the same tag? Should it react to them at once, or separately?



In Verilog, it is nondeterministic. In VHDL, it sees a sequence of two distinct events separated by “delta time” and reacts twice, once to each input. In the Ptolemy II DE domain, it sees the events together and reacts once.

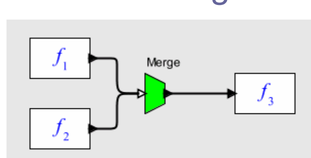
Example

In the following segment of a model, clearly we wish that the VariableDelay see the output of Rician when it processes an input from CurrentTime.



Question 3:

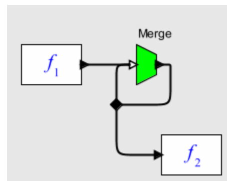
What if the two sources in the following model deliver an event with the same tag? Can the output signal have distinct events with the same tag?



Recall that we require that a signal be a partial function $s : T \to V$, where V is a set of possible event values (a data type), and T is a totally ordered set of *tags*.

Question 4:

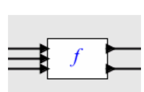
What does this mean?



The Merge presumably does not introduce delay, so what is the meaning of this model?

Mathematical Framework

Let the set of all signals be $A = [T \to V]$ where T is a totally ordered set and V is a set of values. Let an actor



be a function $f : A^n \to A^m$. What are the constraints on these functions such that:

1. Compositions of actors are determinate.
2. Feedback compositions have a meaning.
3. We can rule out Zeno behavior.

Can We Re-Use Prefix Orders?

Since tags are totally ordered, signals can be thought of as sequences. Can we just re-use PN semantics?

Signals as Sequences of Events

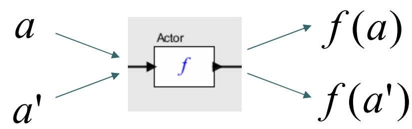
A discrete signal s is a set of events with distinct tags where there is an order embedding from the tags to the integers. Thus, a signal is equivalently a *sequence* s' of events, a partial function

$$s' : N \to T \times V$$

where the tags are ordered,

$$n < m \quad \Rightarrow \quad \pi_1(s'(n)) < \pi_1(s'(m))$$

Prefix Order on Signals



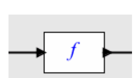
Consider using the prefix order on signals and requiring actors to be monotonic functions:

$$a \sqsubseteq a' \quad \Rightarrow \quad f(a) \sqsubseteq f(a')$$

Will this be an adequate basis for DE semantics?

First Problem: Ensuring that Tags are Distinct

Consider an actor:

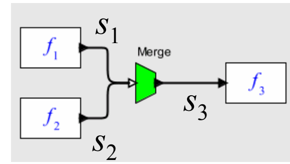


where, for each input event e it produces the output $((0, 0), 0)$, an event with tag $(0, 0)$. The output sequence does not have distinct tags. But the function is monotonic in the prefix order.

Simple solution: Do not allow actors to specify the index. The output sequence becomes:

$(((0, 0), 0), ((0, 1), 0), ((0, 2), 0), \ldots)$

Example: Merge Actor



The output cannot be defined to be simply the union of the input events, because the output may then have duplicate tags.

Define the Merge actor so that if the inputs have events with the same time stamp t:

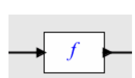
$$s_1 = \{\ldots ((t, 0), v_1), ((t, 1), v_2), \ldots)$$
$$s_2 = \{\ldots ((t, 0), q_1), ((t, 1), q_2), \ldots)$$

the output will interleave these as follows:

$$s_3 = \{\ldots ((t, 0), v_1), ((t, 1), q_1), ((t, 2), v_2), ((t, 3), q_2), \ldots)$$

Second Problem: Causality

Consider an actor:



where, for each input event e with time stamp τ it produces an output event with time stamp $\tau - 1$. This actor is monotonic in the prefix order, but could be used to build time travel machines.

Looks like a prefix order alone won't do the job…

Conclusion and Open Issues

- A *discrete system* is one where the there is an order embedding from the set of tags in the system to the integers.
- Monotonic functions on a prefix order does not appear to be sufficient for DE semantics.