Kahn's Principle and the Semantics of Discrete Event Systems

Xiaojun Liu

EECS 290N Report
December 17, 2004

University of California at Berkeley
Berkeley, CA 94720, USA

liuxj@eecs.berkeley.edu

Abstract

Kahn process networks (KPN) is an elegant model of computation for deterministic concurrent processes. The denotational semantics of KPN is based on the theory of partially order sets. This report explores the application of this approach to the semantics of discrete event systems.

1 Introduction

In Kahn process networks [3], computation is performed by a set of independent processes. Processes communicate streams of data asynchronously through unbounded FIFO channels. In the denotational semantics of KPN, a prefix ordering is defined on tuples of streams, which then form a CPO. If processes are continuous functions from their input tuple of streams to output, then Kahn's principle applies: any network of processes is a continuous function from its input to output.

The derivation of Kahn's principle is valid as long as the signals between processes are a CPO and the processes are continuous. In section 2, we use the tagged signal model [5] to give a very general definition of signals and a prefix order. The set of signals with the prefix order is a CPO. An extended Kahn's principle follows naturally.

Only limited timing information is present in a KPN — the ordering within each stream of data. Yates and Gao [6] extended Kahn's principle to networks of real-time processes. The processes communicate timed streams. A timed stream is a sequence of time-stamped tokens. The time stamps in a timed stream are strictly increasing and have a minimum time interval. A real-time process is a function from its input streams to output streams, and is required to introduce a minimum delay. Some common processes, such as timed merge, are not monotonic (hence not continuous), so Kahn's principle cannot be applied directly. A novel definition of the network functional – for real-time networks that may contain nonmonotonic processes – is presented and shown to be continuous. From this, Kahn's principle is extended to real-time networks. We present an alternative approach in section 3, applying the general signal definition to discrete event (DE) signals. Compared to [6], ours is a more direct extension of the principle, and is more general in some respects. For example, processes are not required to have a minimum delay.

Our work is part of the Ptolemy project [1]. One focus of the project is to study the simulation and programming models of various models of computation (MoC). In section 4 we turn to the operational semantics of real-time processes and DE systems. Our formulation of DE semantics leads to an alternative development of the Chandy-Misra approach [2] to distributed DE simulation, and suggests a way to handle multirate signals and delta delays in DE systems.

An alternative approach to the semantics of real-time processes is presented in [4]. It uses the theory of metric spaces to study the semantics of real-time networks. The connection and contrast between the two approaches will be presented in a future report.

2 Tagged signal model and prefix order

A partially ordered set of *tags* T capture the timing or ordering information in signals. For example, a tag may be the time stamp of an event, or the sequence number of a token in a stream of data. Let $\mathbb{L}(T)$ be the set of lower sets (also called down sets) of T. $\mathbb{L}(T)$, treated as a poset with the subset order, is a complete lattice.

Definition 1 (Signal) *Given a tag set T and a value set V, a* signal *is a function $s : L \to V$ where $L \in \mathbb{L}(T)$. Let $\mathbb{S}(T,V)$ denote the set of all signals with domain in $\mathbb{L}(T)$ and codomain V.*

Sometimes a value set V is augmented with a special element ε denoting the absence of value, $V_\varepsilon = V \cup \{\varepsilon\}$, or $\perp$ denoting unknown or undefined, $V_\perp = V \cup \{\perp\}$. Following are some examples of tag sets and signals:

- Any set A can be treated as a poset with the discrete order. $\mathbb{S}(A,V)$ is the set of all partial functions from A to V.
- T is $\mathbb{R}_+$, the set of nonnegative real numbers with the usual order. $\mathbb{S}(\mathbb{R}_+, \mathbb{Z}_\varepsilon)$ is the set of DE signals with integer values.

Definition 2 (Prefix order) *For any poset T of tags and set V of values, the prefix order relation $\leq_p$ on $\mathbb{S}(T,V)$ is, given $s_1 : L_1 \to V$ and $s_2 : L_2 \to V$,*

$$s_1 \leq_p s_2 \iff L_1 \subseteq L_2 \text{ and } \forall t \in L_1, s_1(t) = s_2(t)$$

Claim 1 *For any poset T of tags and set V of values, $(\mathbb{S}(T,V), \leq_p)$ is*

- *a poset.*
- *a CPO.*
- *a complete lower semilattice.*

Proof.

It is straightforward to verify that $\leq_p$ is a partial order.

Let S be a directed subset of $(\mathbb{S}(T,V), \leq_p)$. For any two signals $s_1 : L_1 \to V$ and $s_2 : L_2 \to V$, there exists $s_3 : L_3 \to V$ in S such that

$$s_1 \leq_p s_3 \text{ and } s_2 \leq_p s_3$$

For all $t \in L_1 \cap L_2$, $s_1(t) = s_3(t) = s_2(t)$. Define $s' \in \mathbb{S}(T,V)$ where

$$\text{dom}(s') = \bigcup_{s \in S} \text{dom}(s)$$

$$s'(t) = s(t), s \in S \text{ and } t \in \text{dom}(s)$$

s' is the LUB of S. $(\mathbb{S}(T,V), \leq_p)$ is a CPO.

To prove that $(\mathbb{S}(T,V), \leq_p)$ is a complete lower semilattice, we need to show that any subset $S' \subseteq \mathbb{S}(T,V)$ has a GLB. Let $E \subseteq \bigcap_{s \in S'} \text{dom}(s)$ be

$$E = \{t \in T \mid \forall s_1, s_2 \in S', s_1(t) = s_2(t)\}$$

Define $\hat{s} \in \mathbb{S}(T,V)$ where dom$(\hat{s})$ is the largest lower set of T contained in E, and

$$\hat{s}(t) = s(t), s \in S' \text{ and } t \in \text{dom}(s)$$

$\hat{s}$ is the GLB of S'.

Claim 2 (Generalized Kahn's principle) *If all processes in a network are continuous functions from input signals to output signals, then the network computes a continuous function from its input to output. The function is the least fixed point of a continuous functional determined by the network structure and the functions computed by the processes.*

The proof is a straightforward extension of that for the original Kahn's principle [3].

3 Discrete event semantics

3.1 DE signals

The tag set of discrete event signals is $\mathbb{R}_+$, a totally ordered set.

Lemma 1 *If the tag set T is totally ordered, then any directed subset of $\mathbb{S}(T,V)$ is a chain.*

When specifying a DE signal s, we use the convention that for all $t \in \text{dom}(s)$, $s(t)$ is absent ($= \varepsilon$) if not defined explicitly. Some examples of DE signals are:

- clock : $\mathbb{R}_+ \to \mathbb{Z}_\varepsilon$
$$\text{clock}(n) = 1, \forall n \in \mathbb{N} \tag{1}$$
- zeno : $\mathbb{R}_+ \to \mathbb{Z}_\varepsilon$
$$\text{zeno}(t) = 1, \forall t \in \mathbb{N} \cup \{1 - 2^{-n} \mid n \in \mathbb{N}\} \tag{2}$$
- reverse-zeno : $\mathbb{R}_+ \to \mathbb{Z}_\varepsilon$
$$\text{reverse-zeno}(t) = 1, \forall t \in \mathbb{N} \cup \{2^{-n} \mid n \in \mathbb{N}\} \tag{3}$$

3.2 DE processes

Most common DE processes are continuous with respect to the prefix order in definition 2. For example,

- add
Given the addition function $+ : V \times V \to V$, define $+_\varepsilon : V_\varepsilon \times V_\varepsilon \to V_\varepsilon$ as

$+_\varepsilon$	v_1	ε
v_2	$v_1 + v_2$	v_2
ε	v_1	ε

Define $s = \text{add}(s_1, s_2)$ as

$$\text{dom}(s) = \text{dom}(s_1) \cap \text{dom}(s_2)$$

$$s(t) = s_1(t) +_\varepsilon s_2(t), \forall t \in \text{dom}(s)$$

$$
\begin{aligned}
\text{dom}(s) &= \begin{cases} L_1 \cap L_2 \cup \{q \mid q \geq p \;\text{ and }\; \forall t \in [p,q], s_1(t) \in V\} & \text{if } p \notin L_2 \\ L_1 \cap L_2 \cup \{q \mid q \geq p \;\text{ and }\; \forall t \in (p,q], s_1(t) \in V\} & \text{if } p \in L_2 \end{cases} \\
s(t) &= \begin{cases} s_1(t) & \text{if } t \in \text{dom}(s) \;\text{ and }\; s_1(t) \in V \\ s_2(t) & \text{if } t \in \text{dom}(s) \;\text{ and }\; s_1(t) = \varepsilon \end{cases}
\end{aligned} \tag{4}
$$

- delay_1
  Define $s' = \text{delay}_1(s)$ as

$$\text{dom}(s') = [0,1) \cup \{t+1 \mid t \in \text{dom}(s)\}$$

$$s'(t) = s(t-1), \forall t \in \text{dom}(s') \cap [1, \infty)$$

- merge
  Let $L_1 = \text{dom}(s_1)$, $L_2 = \text{dom}(s_2)$, and $p = \sup L_2$. Define $s = \text{merge}(s_1, s_2)$ as in equation 4.

3.3 Applying Kahn's principle

Consider a DE system model with 3 signals x, y, and z, satisfying the following equations:

$$
\begin{aligned}
x &= \text{reverse-zeno} \\
y &= \text{merge}(z, x) \\
z &= \text{delay}_1(y)
\end{aligned} \tag{5}
$$

The behavior of this model is the least fixed point solution of these equations:

$$
\begin{aligned}
y(t) &= 1, \quad \forall t \in \{k + 2^{-n} \mid k, n \in \mathbb{N}\} \\
z(t) &= 1, \quad \forall t \in \{k + 2^{-n} \mid k, n \in \mathbb{N}, k \geq 1\}
\end{aligned} \tag{6}
$$

If we change the last equation in 5 to

$$z = \text{lookahead}_1(y)$$

the system becomes non-causal. The behavior is still well-defined, with

$$\text{dom}(y) = \text{dom}(z) = \emptyset$$

3.4 Continuity and causality

It is important to note that continuity does not imply causality – the process that "looks ahead by 1" is continuous but not causal, nor does causality imply continuity – the process that produces an event after counting an infinite number of input events is causal but not continuous.

3.5 Discrete signals

$\mathbb{S}(\mathbb{R}_+, V_\varepsilon)$ contains many "weird" signals that do not appear in modeling physical systems, such as (taking $V = \mathbb{Z}$) the reverse-zeno signal.

Definition 3 (Discrete signal) *A signal $s \in \mathbb{S}(\mathbb{R}_+, V_\varepsilon)$ is* discrete *if $s^{-1}(V)$, the preimage of V, is order-isomorphic to a lower set of $\mathbb{N}$, and if $s^{-1}(V)$ is an infinite set,*

$$\text{dom}(s) = \bigcup_{t \in s^{-1}(V)} [0, t]$$

Let $\mathbb{D}(\mathbb{R}_+, V_\varepsilon)$ denote the set of discrete signals.

According to this definition, the signal $d : [0,1) \to \mathbb{Z}_\varepsilon$ with

$$d(t) = 1, \forall t \in \{1 - 2^{-n} \mid n \in \mathbb{N}\} \tag{7}$$

is discrete, but the signal $nd : [0,1] \to \mathbb{Z}_\varepsilon$ with

$$nd(t) = 1, \forall t \in \{1 - 2^{-n} \mid n \in \mathbb{N}\} \tag{8}$$

is not. As will become clear later in section 4.3, the discrete signals are the subset of DE signals that can be represented by timed streams.

Claim 3 *$(\mathbb{D}(\mathbb{R}_+, V_\varepsilon), \leq_p)$ is a poset, a CPO, and a complete lower semilattice.*

3.6 Multirate systems

Some DE processes may produce multiple output events with the same time stamp, such as *repeat* and a version of *merge* that does not discard any simultaneous input events. For such processes, we use the tag set $T_{DE} = \mathbb{R}_+ \times \mathbb{N}$ with the lexicographical order – a total order – to define DE signals. It is straightforward to generalize the previous definitions and claims. The generalization can handle the delta delay operations in VHDL as well.

4 Operational semantics

4.1 Signal representation

We can represent a discrete signal by a sequence of time-stamped tokens. For example, a representation of the clock signal defined in equation 1 is $\{(n,1) \mid n \in \mathbb{N}\}$. Only including the "present" events in the sequence is not adequate. Consider these signals

$$
\begin{aligned}
u_0 : [0,1] \to \mathbb{Z}_\varepsilon \quad &\text{such that} \quad u_0(0) = 1 \\
u_1 : [0,1] \to \mathbb{Z}_\varepsilon \quad &\text{such that} \quad u_1(0) = u_1(1) = 1
\end{aligned} \tag{9}
$$

$u_0 \not\leq_p u_1$. If the signal representation only includes present events, then the representation of u_0 is $\{(0,1)\}$, a prefix of that of u_1 – $\{(0,1),(1,1)\}$.

4.2 Null events

We solve this problem by including "null events" in the signal representation. The representation of u_0 becomes $\{(0,1),(1,\varepsilon)\}$, making it clear that u_0 is not a prefix of u_1.

4.3 Timed streams

It is also necessary for the signal representation to distinguish between u_0 and

$$u_2 : [0,1) \to \mathbb{Z}_\varepsilon \text{ such that } u_2(0) = 1 \tag{10}$$

Definition 4 (Time stamp) *A* time stamp *is: $t \in \mathbb{R}_+$, or t^- for $t \in \mathbb{R}_+ \setminus \{0\}$, or ∞. Let TS denote the set of time stamps. Extend the usual order on $\mathbb{R}_+$ to TS by letting*

$$r < t^- < t, \forall r < t$$

Definition 5 (Timed stream) *A* timed stream *is an element of $(TS \times V_\varepsilon)^{**}$ (a finite or infinite sequence of tuples (ts, v) with $ts \in TS$ and $v \in V_\varepsilon$) such that the time stamps in the sequence are strictly increasing.*

We can represent the signal u_2 with the timed stream $\{(0,1),(1^-,\varepsilon)\}$. We can formulate a (many-to-one) mapping from timed streams to discrete signals, and show that the discrete signals are the subset of DE signals representable by timed streams.

We define an equivalence relation between timed streams as: two streams are equivalent if they represent the same discrete signal. We can later show that a DE process computes equivalent output streams from equivalent input streams.

4.4 Discrete event process networks

We extend KPN by replacing data streams with timed streams, and Kahn processes with DE processes. Ours is an alternative development of the Chandy-Misra approach.

4.5 Snapshots of DE processes

Given any causal DE process P and its input up to t, its input-output behavior at t is a (continuous) Kahn process. The simulation of a DE system at t can be treated as a Kahn process network.

5 Conclusion and future work

Using the tagged signal model, we propose a general definition of signals and a prefix order of signals. We show that the signals form a CPO, and present a generalized Kahn's principle. We apply the general principle to study the semantics of discrete event systems. We plan to implement a DE simulator using the proposed operational semantics and compare it with other strategies.

References

[1] C. Brooks, E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, and H. Zheng. Heterogeneous concurrent modeling and design in java. Technical Report Technical Memorandum UCB/ERL M04/27, University of California, July 29 2004.

[2] K. Chandy and J. Misra. Asynchronous distributed simulation via a sequence of parallel computations. *Communications of the ACM*, 24(11):198–206, April 1981.

[3] G. Kahn. The semantics of a simple language for parallel programming. In J. L. Rosenfeld, editor, *Information processing*, pages 471–475, Stockholm, Sweden, Aug 1974. North Holland, Amsterdam.

[4] E. A. Lee. Modeling concurrent real-time processes using discrete events. *Annals of Software Engineering*, 7:25–45, 1999.

[5] E. A. Lee and A. Sangiovanni-Vincentelli. A framework for comparing models of computation. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1217–1229, 1998.

[6] R. K. Yates and G. R. Gao. A kahn principle for networks of nonmonotonic real-time processes. In *Parallel Architectures and Languages Europe*, pages 209–227, 1993.