On the Kahn Principle and Fair Networks

Stephen Brookes

Department of Computer Science
Carnegie Mellon University

MFPS

May 1998

KAHN NETWORKS

A model of deterministic systems. . .

- data as streams
  - V^∞ = finite and infinite sequences
  - ordered by prefix
- nodes as deterministic processes
  - processes communicate asynchronously on buffered channels
  - each process computes a *continuous* input-output function
  - $f : V_1^\infty \times \cdots \times V_k^\infty \to V^\infty$
- Kahn's principle
  - mutually recursive functions
  - network behavior is least fixed point

EXAMPLE

$$
\begin{array}{l}
\mathit{filter}(p,\ a,\ b) = \\
\quad \textbf{local}\ x\ \textbf{in} \\
\qquad \textbf{while true do} \\
\qquad\quad (a?x;\ \textbf{if}\ x\ \textbf{mod}\ p \neq 0\ \textbf{then}\ b!x);
\end{array}
$$

$$
\begin{array}{l}
\mathit{sift}(a,\ \mathit{out}) = \\
\quad \textbf{local}\ b,\ p\ \textbf{in} \\
\qquad \textbf{begin} \\
\qquad\quad a?p;\ \mathit{out}!p; \\
\qquad\quad \mathit{filter}(p,\ a,\ b) \parallel \mathit{sift}(b,\ \mathit{out}) \\
\qquad \textbf{end}
\end{array}
$$

$$\mathit{nats}(k,\ a) = a!k; \mathit{nats}(k+1,\ a)$$

$$
\begin{array}{l}
\mathit{primes}(\mathit{out}) = \\
\qquad \textbf{local}\ a\ \textbf{in}\ (\mathit{nats}(2, a) \parallel \mathit{sieve}(a, \mathit{out}))
\end{array}
$$

ADVANTAGES

- Language combines Algol and CSP
  - restricted subset
- Simple network calculus
  - cascade, feedback
  - juxtaposition
  - recursion
- Supports network analysis
  - safety: every output is prime
  - liveness: every prime will be output eventually
- Describes causality
  - $f(\epsilon) = \epsilon$ and $f(v) \neq \epsilon$ implies input causes output

OPERATIONAL JUSTIFICATION

Nodes are computing stations

- finite work in finite time
- compute output from input
- deterministic

Continuity matches intuition

(1) more input $\Rightarrow$ no less output

(2) finite output needs finite input

(3) infinite output appears as the limit of its finite prefixes

LIMITATIONS

- Deterministic
  - limited applicability
  - no shared input or output
  - visible channels unidirectional
  - node waits on at most one input
- Non-homogeneous
  - nodes are sequential
  - sequential composition is non-monotone
  - semantics of nodes given separately
  - prevents hierarchical analysis
- Doesn't easily generalize to non-deterministic case
  - Brock-Ackerman anomaly
  - problems with fairness

NON-DETERMINISM

- Sharing output channels

$$
\begin{aligned}
&\mathit{merge}(\mathit{left}, \mathit{right}, \mathit{out}) = \\
&\qquad \textbf{local } x, y \textbf{ in} \\
&\qquad\qquad \textbf{while true do } \mathit{left}?x; \mathit{out}!x \\
&\qquad\qquad \| \textbf{ while true do } \mathit{right}?y; \mathit{out}!y
\end{aligned}
$$

- Sharing input channels

$$
\begin{aligned}
&\mathit{split}(\mathit{in}, \mathit{left}, \mathit{right}) = \\
&\qquad \textbf{local } x \textbf{ in} \\
&\qquad\qquad \textbf{while true do } \mathit{in}?x; \mathit{left}!x \\
&\qquad\qquad \| \textbf{ while true do } \mathit{in}?x; \mathit{right}!x
\end{aligned}
$$

- Bi-directional channels

$$
\begin{aligned}
&\textbf{local } x, y \textbf{ in} \\
&\qquad \textbf{while true do } (a?x \| b?y;\ a!y \| b!x)
\end{aligned}
$$

GENERALIZING KAHN

Traditional aims:

- as simple as possible
- retain spirit of continuity
- least fixed point

Examples:

- stream relations, hiatons, scenarios
- I/O automata
- sets of continuous functions

Typical limitations:

- only continuous operations
- fairness absent or restricted
- operational justification

BROCK-ACKERMAN (1)

$$\begin{aligned} P_1 &= \textbf{local } x, y \textbf{ in } (i?x; o!x; i?y; o!y) \\ P_2 &= \textbf{local } x, y \textbf{ in } (i?x; i?y; o!x; o!y) \end{aligned}$$

$$str[\![P_1]\!] \neq str[\![P_2]\!]$$

$$\begin{aligned} S[-] &= \textbf{local } l', r', i \textbf{ in} \\ &\qquad D(l, l') \parallel D(r, r') \parallel M \parallel [-] \end{aligned}$$

$$D(l, l') = \textbf{local } z \textbf{ in } (l?z; l'!z; l'!z)$$

$$M = merge(l', r', i)$$

$$str[\![S[P_1]]\!] = str[\![S[P_2]]\!]$$

BROCK-ACKERMAN (2)

$$\begin{array}{rcl} T[-] & = & \textbf{local } h, r, o \textbf{ in} \\ & & \quad [-] \,\|\, \mathit{spray} \,\|\, \mathit{times5} \end{array}$$

$$\begin{array}{rcl} \mathit{spray} & = & \textbf{local } z \textbf{ in} \\ & & \quad \textbf{while true do } (o?z; b!z; h!z) \end{array}$$

$$\begin{array}{rcl} \mathit{times5} & = & \textbf{local } z \textbf{ in} \\ & & \quad \textbf{while true do } (h?z; r!(5 \times z)) \end{array}$$

$$\mathit{str}[\![S[P_1]]\!] = \mathit{str}[\![S[P_2]]\!]$$

$$\mathit{str}[\![T[S[P_1]]]\!] \neq \mathit{str}[\![T[S[P_2]]]\!]$$

Stream relations are not compositional
for non-deterministic networks

IS CONTINUITY FAIR?

$$B \;=\; \textbf{local } x \textbf{ in} \quad \textbf{while true do } (a?x; b!x)$$

$$B' \;=\; \textbf{local } x \textbf{ in} \quad \textbf{while true do } (\textbf{skip or } (a?x; b!x))$$

$$B_* \;=\; \textbf{local } x, n \textbf{ in} \quad n{:=}?; \textbf{for } i{:=}1 \textbf{ to } n \textbf{ do } (a?x; b!x)$$

STREAM BEHAVIORS

$$str[\![B]\!] = \{(\rho, \rho) \mid \rho \in V^{\infty}\}$$
$$str[\![B']\!] = \{(\rho, \sigma) \mid \sigma \leq \rho \;\&\; \rho, \sigma \in V^{\infty}\}$$
$$str[\![B_*]\!] = \{(\rho, \rho) \mid \rho \in V^{*}\} \quad \textit{not continuous}$$

No operational justification for imposing continuity

OPERATIONAL CONSIDERATIONS

- Rationale

(1) more input $\Rightarrow$ no less output

(2) finite output needs finite input

(3) infinite output occurs as limit of finite prefixes

- Non-deterministic case
  - (1), (2) hold, but not (3)
  - each finite prefix might come from a different computation
  - continuity rules out fairness
  - continuity confuses causality

Operational justification fails

A PROBLEM WITH SEQUENCING

Two deterministic processes:

$$\begin{array}{rcl} sink(a,b) & = & \textbf{local } x \textbf{ in } a?x \\ source(a,b) & = & b!0 \end{array}$$

Their stream functions:

$$\begin{array}{rcl} str[\![sink]\!] & = & \{(\rho,\epsilon) \mid \rho \in V^{\infty}\} \\ str[\![source]\!] & = & \{(\rho,0) \mid \rho \in V^{\infty}\} \end{array}$$

Not compositional:

$$\begin{array}{rcl} str[\![sink]\!] & = & str[\![\textbf{skip}]\!] \\ str[\![sink; source]\!](\epsilon) & = & \epsilon \\ str[\![\textbf{skip}; source]\!](\epsilon) & = & 0 \end{array}$$

ASSESSMENT

For non-deterministic networks:

- fairness is fundamental
  - abstracts from network details
- continuity is not operationally justifiable
- stream relations blur causality and cannot be composed
- stimulus-response behavior is important
  - need a more intensional model

but we'd still like to stay faithful to Kahn's Principle. . .

FAIR NETWORKS

- nodes are non-deterministic
  - asynchronous communication
- nodes *and* networks are processes
  - hierarchical network structure
- processes denote trace sets
  - stream relations extended in time
- fair parallel execution
  - a reasonable abstraction
- fixed point characterization
  - fair parallel composition
  - recursive process definitions
- operational justification
  - trace sets match operational semantics

ADVANTAGES

- compositional
  - no anomalies
- supports network analysis
  - safety and liveness properties
  - stimulus-response
- homogeneous
  - supports hierarchical analysis
- dynamic networks
  - recursion
  - nested parallelism
- fairness incorporated
  - vital for liveness
- can extract stream relation
  - agrees with Kahn interpretation in deterministic cases

TRACES

- A state is a tuple $w = (\bar{v}, \bar{\rho})$ giving the values of variables and contents of channels
- A trace is a sequence of state changes

  $$\langle w_0, w_0' \rangle \langle w_1, w_1' \rangle \ldots \langle w_n, w_n' \rangle \ldots$$

  recording a fair interaction
- A step $\langle w_i, w_i' \rangle$ models a finite sequence of atomic actions

INTUITION

$$\begin{aligned} \text{communication} &= \text{state change} \\ \text{interference} &= \text{action by environment} \\ \text{unrequited input} &= \text{busy wait} \end{aligned}$$

CATEGORY of WORLDS

- Objects: countable sets

  $W = (V_1 \times \cdots \times V_n) \times (H_1^* \times \cdots \times H_k^*)$

- Morphisms: expansions

$$
\begin{aligned}
&h = (f, Q) : W \rightarrow W' \\
&\text{where } f : W' \rightarrow W \\
&\qquad Q \subseteq W' \times W'
\end{aligned}
$$

INTUITION

- A world W is a set of states with the same "shape"
- A morphism $h : W \rightarrow W'$ is an "expansion"

FUNCTORIAL SEMANTICS

Types as functors

$$[\![\mathbf{proc}]\!]W = \mathcal{P}^{\dagger}((W \times W)^{\infty})$$

$$\begin{array}{rcll} [\![\mathbf{chan}[\tau]]\!]W & = & (V_\tau \to (W \to W)) & put \\ & \times & (W \to (W \times V_\tau)\,option) & get \end{array}$$

Phrases as natural transformations

- When $\pi \vdash P : \mathbf{proc}$

  $$traces[\![P]\!]W : [\![\pi]\!]W \to [\![\mathbf{proc}]\!]W$$

- When $h : W \to W'$ and $u' = [\![\pi]\!]hu$

  $$[\![\mathbf{proc}]\!]h(traces[\![P]\!]Wu) = traces[\![P]\!]W'u'$$

INTUITION

Naturality enforces
locality constraints

TRACE SEMANTICS

- A process denotes a total trace set
  - total relation, extended in time
  - complete recipe for interaction
- **Trace sets are closed**

$\alpha\beta \in t \;\&\; w \in W \;\Rightarrow \alpha\langle w, w\rangle\beta \in t$ stuttering

$\alpha\langle w, w'\rangle\langle w', w''\rangle\beta \in t \;\Rightarrow\; \alpha\langle w, w''\rangle\beta \in t$ mumbling

CAVEAT

Trace sets are not prefix-closed
and not closed under limit

A trace represents an entire computation

DOMAINS

Total trace sets form a domain

- ordered by reverse inclusion
- measures non-determinism
- not an information order

Traces form a domain

- ordered by prefix
- irrelevant and misleading

Powerdomains not needed

- cannot deal with fairness
- induce wrong ordering
- too complex

SEMANTIC DEFINITIONS

Assume $W = V \times V^*$

- **skip** has traces of form

$$\langle w_0, w_0 \rangle \ldots \langle w_k, w_k \rangle$$

- $h?x$ has traces of form
  $\langle (v, n\rho), (n, \rho) \rangle$
  $\langle (v_0, \epsilon), (v_0, \epsilon) \rangle \ldots \langle (v_k, \epsilon), (v_k, \epsilon) \rangle \ldots$

- $h!0$ has traces of form

$$\langle (v, \sigma), (v, \sigma 0) \rangle$$

- sequential composition

  concatenation

- parallel composition

  fair merge

PARALLEL COMPOSITION

$$h!1 \parallel \textbf{while true do } h!0 \;=\; (h!0)^* h!1 (h!0)^\omega$$

FAIRMERGE

$$\mathit{fairmerge}_A \in \mathcal{P}(A^\infty \times A^\infty \times A^\infty)$$

$$\begin{aligned} \mathit{fairmerge}_A &= \nu R.\, \mathit{both} \cdot R \cup \mathit{one} \\ &= \mathit{both}^* \cdot \mathit{one} \;\cup\; \mathit{both}^\omega \end{aligned}$$

where

$$\begin{aligned} \mathit{both} &= \{(\alpha, \beta, \alpha\beta), (\alpha, \beta, \beta\alpha) \mid \alpha, \beta \in A^+\} \\ \mathit{one} &= \{(\alpha, \epsilon, \alpha), (\epsilon, \beta, \beta) \mid \alpha, \beta \in A^\infty\} \end{aligned}$$

fairmerge is natural

CHOICE

An external choice

$$(a?x \rightarrow P_1) \Box (b?x \rightarrow P_2)$$

can

- input on a and behave like P_1
- input on b and behave like P_2
- busy-wait while a and b are empty

An internal choice

$$(a?x \rightarrow P_1) \sqcap (b?x \rightarrow P_2)$$

can busy-wait if *either* a or b is empty

LOCAL CHANNELS

The traces of

$$\textbf{local } h : \textbf{chan}[\tau] \textbf{ in } P$$

at world W are projected from the traces of P at $W \times V_\tau^*$ in which

- initially $h = \epsilon$
- h is never changed externally

EXAMPLES

- **local** h **in** $(h!e \| h?x) \;=\; x{:=}e$
- **local** h **in** $(h!0; P) \;=\; P$
  if h does not occur free in P
- **local** h **in** $(h?x; P) =$ **while true do skip**
  because of unrequited input

RECURSION

Recursive process definitions

$$B = a?x; b!x; B$$

correspond to guarded functions on total trace sets,

$$F(t) = \{a?v; b!v; \alpha \mid v \in V \ \& \ \alpha \in t\}$$

with least solutions

$$B = \{a?v; b!v \mid v \in V\}^{\omega}$$

obtained by iteration

Generalizes to mutually recursive families

STREAM RELATIONS

For a trace set T over $V_i^* \times V_o^*$ let $rel(T) \subseteq V_i^\infty \times V_o^\infty$ be

$$
\begin{array}{l}
rel(T) = \{(\rho, \sigma) \mid \ \rho = \langle \rho_n \rangle, \ \sigma = \langle \sigma_n \rangle \ \& \\
\qquad \langle (\rho_0, \epsilon), (\delta_0, \sigma_0) \rangle \\
\qquad\quad \langle (\delta_0 \rho_1, \epsilon), (\delta_1, \sigma_1) \rangle \\
\qquad\qquad \ldots\ldots\ldots\ldots\ldots \\
\qquad\qquad\quad \langle (\delta_{n-1} \rho_n, \epsilon), (\delta_n, \sigma_n) \rangle \\
\qquad\qquad\qquad \ldots\ldots\ldots\ldots\ldots \in T \}
\end{array}
$$

EXAMPLES

$$
\begin{array}{rcl}
rel(traces[\![B]\!]) & = & \{(\rho, \rho) \mid \rho \in V^\infty\} \\
 & = & str[\![B]\!] \\
rel(traces[\![primes]\!]) & = & str[\![primes]\!] \\
rel(traces[\![ABP]\!]) & = & str[\![B]\!]
\end{array}
$$

ANOMALIES?

- Brock-Ackerman
  - $traces[\![P_1]\!] \neq traces[\![P_2]\!]$
  - $traces[\![S[P_1]]\!] \neq traces[\![S[P_2]]\!]$
- Sequential composition
  - $traces[\![sink]\!] \neq traces[\![\mathbf{skip}]\!]$
- Buffers
  - $traces[\![B]\!] \neq traces[\![B']\!] \neq traces[\![B_*]\!]$

LAWS

- Symmetry

$$\begin{aligned}&\textbf{local } h_1 \textbf{ in}\\ &\quad \textbf{local } h_2 \textbf{ in } P\\ &\qquad = \textbf{local } h_2 \textbf{ in}\\ &\qquad\quad \textbf{local } h_1 \textbf{ in } P\end{aligned}$$

- Scope contraction

$$\begin{aligned}&\textbf{local } h \textbf{ in } (P_1 \| P_2) \ =\\ &\quad (\textbf{local } h \textbf{ in } P_1) \| P_2\end{aligned}$$

when h not free in P_2

- Functional laws

$$(\lambda x.P)(Q) = P[Q/x]$$
$$\textbf{rec } x.P = P[\textbf{rec } x.P/x]$$

FEEDBACK

$$feedback(N, \bar{i}, \bar{o}) \;=\; \textbf{local } \bar{i} \textbf{ in } [\bar{i}/\bar{o}]N$$

JUXTAPOSITION

$$juxtapose(N_1, N_2) \;=\; N_1 \| N_2$$

CASCADE

$$\begin{aligned} &cascade(N_1, N_2) \;=\; \\ &\quad \textbf{local } \bar{h} \textbf{ in} \\ &\qquad [\bar{h}/\bar{o}]N_1 \parallel [\bar{h}/\bar{i}]N_2 \end{aligned}$$

CONCLUSION

Trace semantics

- can handle non-determinism
  - bi-directional channels
  - shared channels
  - fair parallelism
- generalizes stream functions
- is faithful to Kahn's spirit
- validates natural laws
- provides a unifying semantic model
  - shared-variable parallelism
  - non-deterministic networks
  - CSP
- is operationally justified

FURTHER WORK

- Applications
  - security protocols
  - deadlock analysis
- Methodology
  - unification of paradigms
  - exploiting fairness
- Concurrent objects
  - private state + methods
- Language design
  - Parallel Algol, Idealized CSP
- Full abstraction
- Connection with game semantics

ALTERNATING BIT

$$ABP = \textbf{local } send, trans, reply, ack \textbf{ in } Accept(0) \parallel Medium \parallel Replying(1)$$

- *Medium* is non-deterministic
  - may lose or replicate
  - cannot lose forever
  - cannot replicate forever
- *ABP* is deterministic
  - behaves like a buffer
- Fairness is crucial
  - guarantees liveness