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
  – \( V^\infty \) = finite and infinite sequences
  – ordered by prefix

• nodes as deterministic processes
  – processes communicate asynchronously on buffered channels
  – each process computes a \textit{continuous} input-output function
  – \( f : V_1^\infty \times \cdots \times V_k^\infty \rightarrow V^\infty \)

• Kahn’s principle
  – mutually recursive functions
  – network behavior is least fixed point
EXAMPLE

\[
\text{filter}(p, a, b) = \\
\quad \text{local } x \text{ in} \\
\quad \quad \text{while true do} \\
\quad \quad \quad (a?x; \text{ if } x \mod p \neq 0 \text{ then } b!x); \\
\]

\[
\text{sift}(a, \text{ out}) = \\
\quad \text{local } b, p \text{ in} \\
\quad \quad \text{begin} \\
\quad \quad \quad a?p; \text{ out}!p; \\
\quad \quad \quad \text{filter}(p, a, b) \parallel \text{sift}(b, \text{ out}) \\
\quad \quad \text{end} \\
\]

\[
\text{nats}(k, a) = a!k; \text{nats}(k + 1, a) \\
\]

\[
\text{primes}(\text{out}) = \\
\quad \text{local } a \text{ in } (\text{nats}(2, a) \parallel \text{sieve}(a, \text{ out})) \\
\]

3
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

\[
merge(left, right, out) = \\
local x, y in \\
\text{while true do left?}x; out!x \\
\| \text{while true do right?}y; out!y
\]

• Sharing input channels

\[
split(in, left, right) = \\
local x in \\
\text{while true do in?}x; left!x \\
\| \text{while true do in?}x; right!x
\]

• Bi-directional channels

\[
local x, y in \\
\text{while true do } (a?x\|b?y; a!y\|b!x)
\]
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)

\[ P_1 = \text{local } x, y \text{ in } (i?x; o!x; i?y; o!y) \]
\[ P_2 = \text{local } x, y \text{ in } (i?x; i?y; o!x; o!y) \]
\[ str[P_1] \neq str[P_2] \]

\[ S[-] = \text{local } l', r', i \text{ in } \]
\[ D(l, l') \parallel D(r, r') \parallel M \parallel [-] \]

\[ D(l, l') = \text{local } z \text{ in } (l?z; l'!z; l'!z) \]

\[ M = \text{merge}(l', r', i) \]

\[ str[S[P_1]] = str[S[P_2]] \]
BROCK-ACKERMAN (2)

\[ T[-] = \text{local } h, r, o \text{ in} \]
\[ [-] || spray || times5 \]

\[ spray = \text{local } z \text{ in} \]
\[ \text{while true do } (o?z; b!z; h!z) \]

\[ times5 = \text{local } z \text{ in} \]
\[ \text{while true do } (h?z; r!(5 \times z)) \]

\[ \text{str}[S[P_1]] = \text{str}[S[P_2]] \]
\[ \text{str}[T[S[P_1]]] \neq \text{str}[T[S[P_2]]] \]

Stream relations are not compositional
for non-deterministic networks
IS CONTINUITY FAIR?

\[ B = \text{local } x \text{ in} \]
\[ \quad \text{while true do } (a?x; b!x) \]

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

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

STREAM BEHAVIORS

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

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:

\[
sink(a, b) = \text{local } x \text{ in } a?x
\]
\[
source(a, b) = b!0
\]

Their stream functions:

\[
str[sink] = \{ (\rho, \epsilon) \mid \rho \in V^\infty \}
\]
\[
str[source] = \{ (\rho, 0) \mid \rho \in V^\infty \}
\]

Not compositional:

\[
str[sink] = str[skip]
\]
\[
str[sink; source](\epsilon) = \epsilon
\]
\[
str[skip; source](\epsilon) = 0
\]
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

communication = state change
interference = action by environment
unrequited input = busy wait
CATEGORY of WORLDS

• Objects: countable sets

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

• Morphisms: expansions

\[ h = (f, Q) : W \rightarrow W' \]

where \( f : W' \rightarrow W \)

\[ Q \subseteq W' \times W' \]

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

\[ [\text{proc}] W = \mathcal{P}^\dagger((W \times W)^\infty) \]
\[ [\text{chan}[^\tau]] W = (V_\tau \rightarrow (W \rightarrow W)) \times (W \rightarrow (W \times V_\tau)\text{option}) \]

Put

Phrases as natural transformations

• When \( \pi \vdash P : \text{proc} \)

\[ \text{traces}[P] W : [\pi] W \rightarrow [\text{proc}] W \]

• When \( h : W \rightarrow W' \) and \( u' = [\pi] hu \)

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

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

\[ \alpha \beta \in t \land w \in W \Rightarrow \alpha \langle w, w \rangle \beta \in t \quad \text{stuttering} \]

\[ \alpha \langle w, w' \rangle \langle w', w'' \rangle \beta \in t \Rightarrow \alpha \langle w, w'' \rangle \beta \in t \quad \text{mumbling} \]

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
  $$\langle (v, \sigma), (v, \sigma0) \rangle$$

- 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

\[ \text{fairmerge}_A \in \mathcal{P}(A^\infty \times A^\infty \times A^\infty) \]

\[ \text{fairmerge}_A = \nu R. \text{both} \cdot R \cup \text{one} = \text{both}^* \cdot \text{one} \cup \text{both} \omega \]

where

\[ \text{both} = \{ (\alpha, \beta, \alpha \beta), (\alpha, \beta, \beta \alpha) \mid \alpha, \beta \in A^+ \} \]

\[ \text{one} = \{ (\alpha, \epsilon, \alpha), (\epsilon, \beta, \beta) \mid \alpha, \beta \in A^\infty \} \]

\[ \text{fairmerge} \text{ is natural} \]
CHOICE

An external choice

\[(a?x \rightarrow P_1)\square(b?x \rightarrow P_2)\]

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 \textit{either} \(a\) or \(b\) is empty
LOCAL CHANNELS

The traces of

\[ \text{local } h : \text{chan}[^\tau] \text{ in } P \]

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

- \textbf{local } h \textbf{ in } (h!e||h?x) = x:=e
- \textbf{local } h \textbf{ in } (h!0; P) = P \quad \text{if } h \text{ does not occur free in } P
- \textbf{local } h \textbf{ in } (h?x; P) = \textbf{while true do skip} \quad \text{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

\[
rel(T) = \{(\rho, \sigma) \mid \rho = \langle \rho_n \rangle, \sigma = \langle \sigma_n \rangle \& \\
\langle (\rho_0, \epsilon), (\delta_0, \sigma_0) \rangle \\
\langle (\delta_0 \rho_1, \epsilon), (\delta_1, \sigma_1) \rangle \\
\cdots \\
\langle (\delta_{n-1} \rho_n, \epsilon), (\delta_n, \sigma_n) \rangle \\
\cdots \in T\}
\]

EXAMPLES

\[
\begin{align*}
rel(traces[B]) &= \{(\rho, \rho) \mid \rho \in V^\infty\} \\
&= str[B] \\
rel(traces[primes]) &= str[primes] \\
rel(traces[ABP]) &= str[B]
\end{align*}
\]
ANOMALIES?

• Brock-Ackerman
  
  – $\text{traces}[P_1] \neq \text{traces}[P_2]$
  
  – $\text{traces}[S[P_1]] \neq \text{traces}[S[P_2]]$

• Sequential composition
  
  – $\text{traces}[\text{sink}] \neq \text{traces}[\text{skip}]$

• Buffers
  
  – $\text{traces}[B] \neq \text{traces}[B'] \neq \text{traces}[B_*]$
LAWS

• Symmetry

  \[\text{local } h_1 \text{ in }\]
  \[\text{local } h_2 \text{ in } P\]
  \[= \text{ local } h_2 \text{ in }\]
  \[\text{local } h_1 \text{ in } P\]

• Scope contraction

  \[\text{local } h \text{ in } (P_1 \parallel P_2) = \]
  \[(\text{local } h \text{ in } P_1) \parallel P_2\]

  when \( h \) not free in \( P_2 \)

• Functional laws

  \[(\lambda x. P)(Q) = P[Q/x]\]
  \[\text{rec } x. P = P[\text{rec } x. P/x]\]
FEEDBACK

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

JUXTAPOSITION

\[ \text{juxtapose}(N_1, N_2) = N_1 \| N_2 \]

CASCADE

\[ \text{cascade}(N_1, N_2) = \text{local } \bar{h} \text{ in } [\bar{h}/\bar{o}]N_1 \| [\bar{h}/\bar{i}]N_2 \]
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

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

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