Compositional Performance Analysis using Probabilistic I/O Automata

Eugene W. Stark
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400 USA
stark@cs.sunysb.edu
http://www.cs.sunysb.edu/~stark

Support: AFOSR, NSF

Collaborators:

Scott Smolka, Xinxin Liu, Giridhar Pemmasani, Sue-hwey Wu

Target: Reactive Systems

Maintain ongoing *interaction* with their environment:

- Receive *stimuli* from environment
- Produce *responses* to environment
- *Control* rather than *compute*-intensive
- Often must satisfy *real-time* constraints
- Often must work in *probabilistic* environments

Typical Examples: Embedded systems, process control software, communications protocols.

Research Objectives

Create a powerful, integrated design environment for reactive systems:

- Provide *automated tools*, based on *formal methods*, for use throughout the design process.
- Enable (quantitative) *reliability* and *performance analysis* to be carried out on a design.
- Exploit *compositional structure* of system descriptions to combat *state-space explosion*.

Underlying Model: CTMCs

A *continuous-time Markov chain* is a transition system, augmented with:

- *probabilities* (on transitions)
  govern choice among possible execution paths
- *rates* (on states)
  determine exponentially distributed “sojourn times”

CTMCs have found many application areas in science and engineering.

It is possible to “solve” a CTMC to obtain quantitative predictions of system performance parameters.

State-Space Explosion

Traditional CTMC-based analysis techniques suffer from *state-space explosion*:

- Systems usually specified as collections of component subsystems.
- Number of CTMC states is exponential in number of components.
- Memory consumption limits size of systems that can be analyzed.
- Current limits are about 1M-10M states.

Avoiding State-Space Explosion

Techniques for avoiding state-space explosion in CTMC-based analysis are active current research topics:

- *State-space reduction* during CTMC generation. (equivalence based: probabilistic bisimulation/ "lumpability")
- Identifying and exploiting *product form properties*.
- *Tensor (Kronecker) algebra* to avoid "multiplying out" system description.
- Component-at-a-time treatment to avoid full CTMC generation.

Related Work: SPAs

- PEPA (Hillston: Edinburgh)
- TIPP (Hermanns, Rettelbach, Herzog: Erlangen)
- MPA (Buchholz), EMPA (Bernardo & Gorrieri: Bologna)
- SPADES (D'Argenio, Katoen, Brinksma: Twente)

Process-algebraic, compositional system specifications.

CTMC is extracted from a specification, then "solved" numerically.

Exploiting compositionality to avoid state-space explosion is of current interest.

Overview of Results

- PIOA: a compositional, CTMC-based model for reactive systems.
- Compositional theory of PIOA "behaviors".
  [Wu, Smolka & Stark, CONCUR'94, TCS 1996]
- Compositional performance analysis algorithms.
  [Stark & Smolka, LICS 1998]
- Implementation in Standard ML using *exact* arithmetic.
  [Pemmasani & Stark, PAPM 1999]
- Some examples with $10^7 - 10^8$ states.

Probabilistic I/O Automata

$$A = (E, Q, \iota, \mu, \rho)$$

where:

- $E = E^{\text{in}} + E^{\text{out}} + E^{\text{int}}$ is the *action signature*
- Q is a set of *states*
- ι is the *starting distribution*
- $\mu : Q \times E \times Q \to [0, 1]$ is the *transition probability function*
- $\rho : Q \to [0, \infty)$ is the *rate function*

such that...

- *Stochastic Conditions*:

  - *Input enabled*:

$$\forall q \in Q \ \ \forall e \in E^{\mathsf{in}} \quad \sum_{r \in Q} \mu(q,e,r) \ = \ 1$$

  - *Locally controlled*:
    $\forall q \in Q$

$$\rho(q) > 0 \ \ \Rightarrow \ \ \sum_{e \in E^{\mathsf{loc}}} \ \sum_{r \in Q} \mu(q,e,r) \ = \ 1$$

$$\rho(q) = 0 \ \ \Rightarrow \ \ \sum_{e \in E^{\mathsf{loc}}} \ \sum_{r \in Q} \mu(q,e,r) \ = \ 0$$

    where $E^{\mathsf{loc}} = E^{\mathsf{out}} + E^{\mathsf{int}}$.

PIOAs: Interpretation

- *Sojourn time* in a state q is exponentially distributed, with rate $\rho(q)$.
- A *locally controlled* (non-input) transition is selected probabilistically upon completion of the sojourn in state q.
- *Input actions* can *preempt* the sojourn in state q.
  The new state is selected probabilistically.
- Choice between input or non-input is determined by a *race* between automaton and environment.
- *Closed* PIOAs (no input actions) determine continuous-time Markov chains (CTMCs).

Executions and Rated Traces

Executions:

$$\sigma : q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n$$

such that $\mu(q_k, a_{k+1}, q_{k+1}) > 0$ for $0 \leq k < n$.

Executions give full information (except for precise timings) about a system trajectory.

Rated Traces:

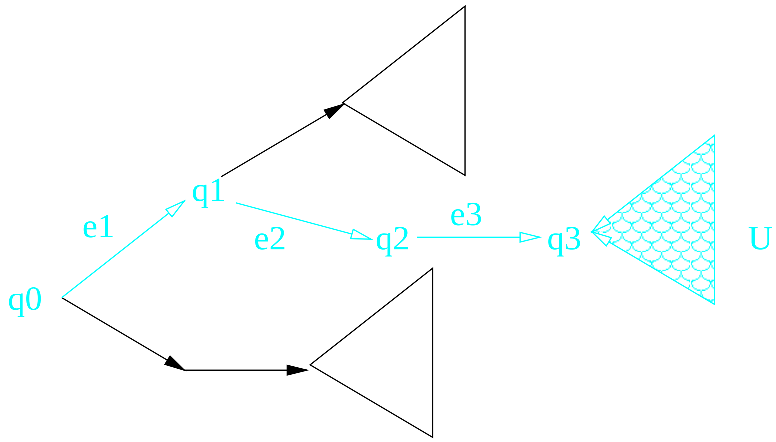
$$\alpha : r_0 \xrightarrow{a_1} r_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} r_n$$

where $r_k \in [0, \infty)$ for $0 \leq k \leq n$.

Rated traces give *externally observable* information about system trajectories.

Probability Measure on Executions

Closed PIOAs (those with no input actions) determine a probability measure on the set of executions.



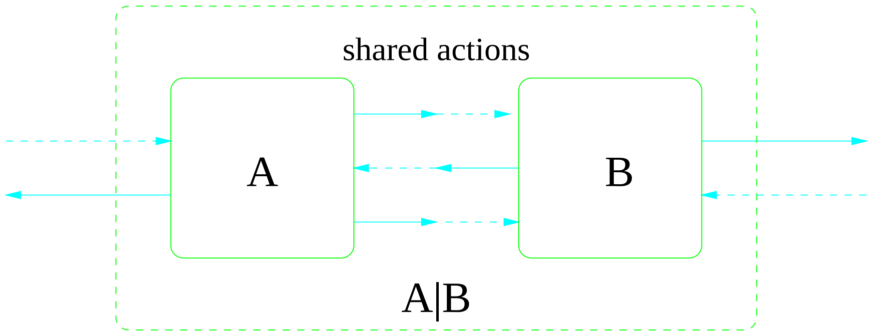
$$\Pr(U) = \prod_{k=0}^{n-1} \mu(q_k, e_{k+1}, q_{k+1})$$

Operations on PIOAs

PIOAs admit two important algebraic operations:

1. **Composition**:
   - Combines *compatible* PIOAs A and B into a *composite* PIOA $A|B$.
   - Components execute *concurrently* and *asynchronously*, but synchronize on shared actions.
   - *Race criterion* (using rates) governs control of next action.
2. **Restriction** (or Hiding):
   - Makes some output actions internal.

Composition of PIOAs: Idea



PIOAs A and B with "compatible" action signatures can be composed to form $A|B$.

- State set is cartesian product of components' sets.
- Synchronization by shared actions (output drives input).
- Rates, probabilities reflect the "race criterion".

Race Criterion

In a composite PIOA $A|B$, component A and B "race" for control of the next action.

- Rate of composite state (q_A, q_B) is sum:

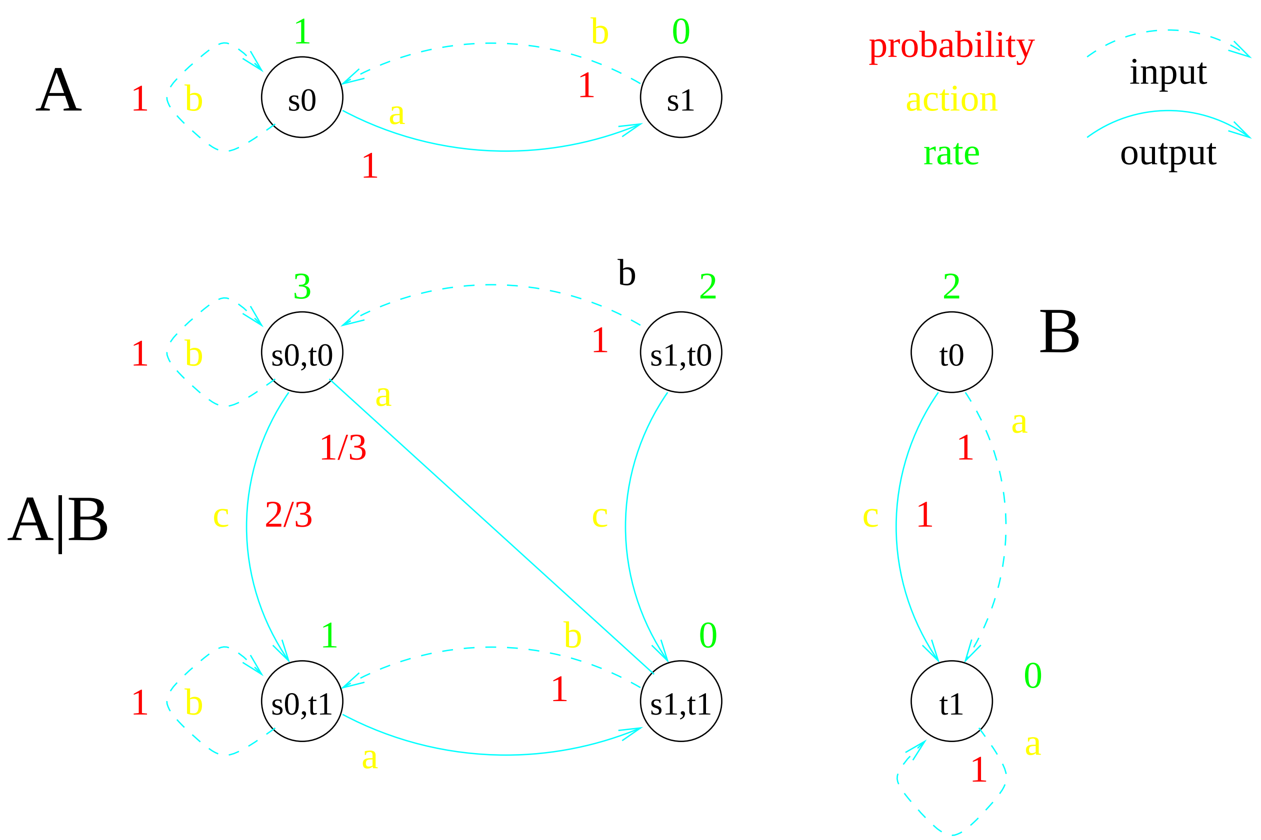
$$\rho(q_A) + \rho(q_B)$$

- Probabilities of non-input transitions weighted by ratios

$$\frac{\rho(q_A)}{\rho(q_A) + \rho(q_B)} \qquad \frac{\rho(q_B)}{\rho(q_A) + \rho(q_B)}$$

  giving the respective probabilities of A and B winning the race.

Composition of PIOAs: Example



Observables

An *observable* is a real-valued function Φ on rated traces:

$$\Phi : \text{Traces}(E) \to \mathbf{R}$$

Examples:

1. The *characteristic function* of a set $U \subseteq \text{Traces}(E)$:

$$\chi_U(\alpha) = \begin{cases} 1, & \text{if } \alpha \in U \\ 0, & \text{otherwise} \end{cases}$$

2. The *expected delay observable*:

$$T(r_0 \xrightarrow{a_1} r_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} r_n) = \sum_{k=0}^{n-1} \frac{1}{r_k}$$

Performance Measures via Observables

Let A be a *closed* PIOA with action signature E.

Let $\alpha\!\uparrow$ denote the set of "extensions" of a rated trace α. Suppose $U \subseteq \text{Traces}(E)$ is an antichain w.r.t. extension.

Hitting Probability:

$$\sum_{\alpha} \chi_U(\alpha)\, \mathsf{Pr}_A(\alpha\!\uparrow)$$

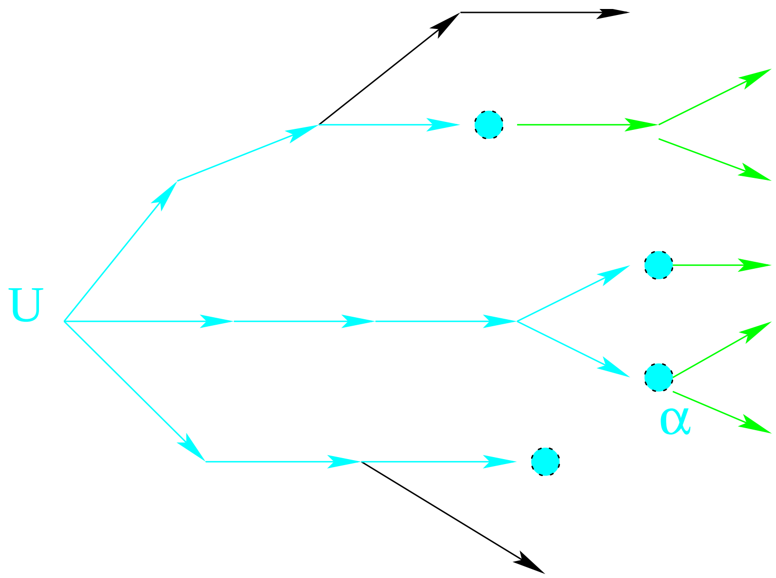
is the probability that A "hits" U.

Expected Time to Hitting: Suppose A almost surely hits U. Then

$$\sum_{\alpha} \chi_U(\alpha)\, T(\alpha)\, \mathsf{Pr}_A(\alpha\!\uparrow)$$

is the expected time for A to hit U.

Performance Measures: Idea



Hitting Probability:

$$\sum_{\alpha} \chi_U(\alpha)\ \mathsf{Pr}_A(\alpha\!\uparrow)$$

PIOA Behaviors

The *behavior* of a PIOA A is a *transformation of observables*:

$$\mathcal{B}_A^E : (\text{Traces}(E \cup E_A) \to \mathbf{R}) \to (\text{Traces}(E) \to \mathbf{R})$$

defined by:

$$\mathcal{B}_A^E[\Phi](\beta) \;\; = \sum_{\alpha \gg_A \beta} \Phi(\alpha)\, w_A(\alpha, \beta)$$

where

- $\alpha \gg_A \beta$ means "α *refines* β w.r.t. A."
- $w_A(\alpha, \beta)$ is a certain "weighting factor."

PIOA Behaviors: Intuition

$$\mathcal{B}_A^E[\Phi](\beta) \;=\; \sum_{\alpha \gg_A \beta} \Phi(\alpha)\, w_A(\alpha, \beta)$$

"Given constraint β, enumerate all A-refinements α of β, and form weighted sum of quantities $\Phi(\alpha)$."

1. β is a "constraint" imposed by the environment of A.
2. α "refines" β by (possibly) inserting actions of E_A and increasing rates in accordance with A.
3. $w_A(\alpha, \beta)$ accounts for transition probabilities of A, and also the "race" between A and its environment.

PIOA Behaviors: Closed Case

If A closed, then

$$\mathcal{B}^{\emptyset}_{A}[\Phi](\mathbf{0}) \;=\; \sum_{\alpha} \Phi(\alpha)\,\Pi(\alpha)\,\mathsf{Pr}_{A}(\alpha\!\uparrow)$$

where $\Pi(\alpha)$ is the product of all rates but the last in α, and $\mathbf{0}$ denotes the rated trace with no actions and rate zero.

Thus, *e.g.* the expected time for A to hit $U \subseteq \mathsf{Traces}(E_A)$ may be expressed as:

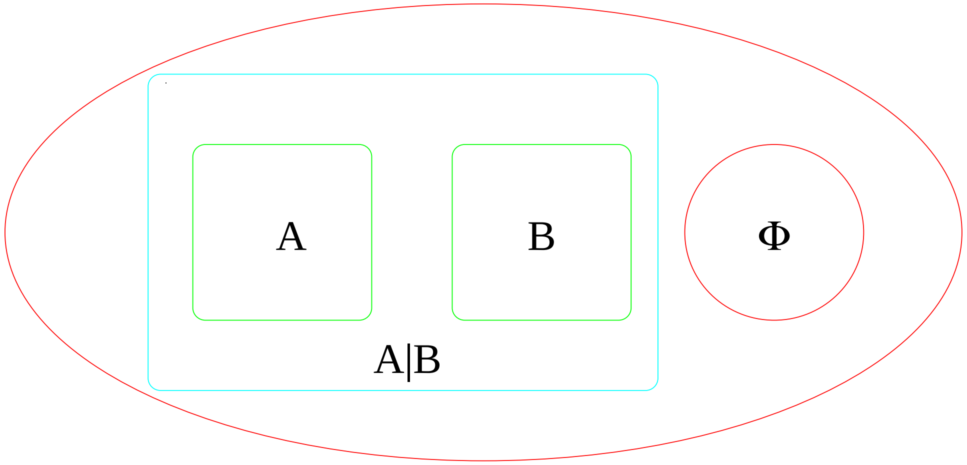
$$\mathcal{B}^{\emptyset}_{A}[\chi_U T \;/\; \Pi](\mathbf{0})$$

Similarly for other reward-based transient performance measures.

Compositionality of PIOA Behaviors

Theorem [*Wu, Stark, Smolka, Liu*] Suppose A and B are compatible PIOAs, and E is a set of actions. Then

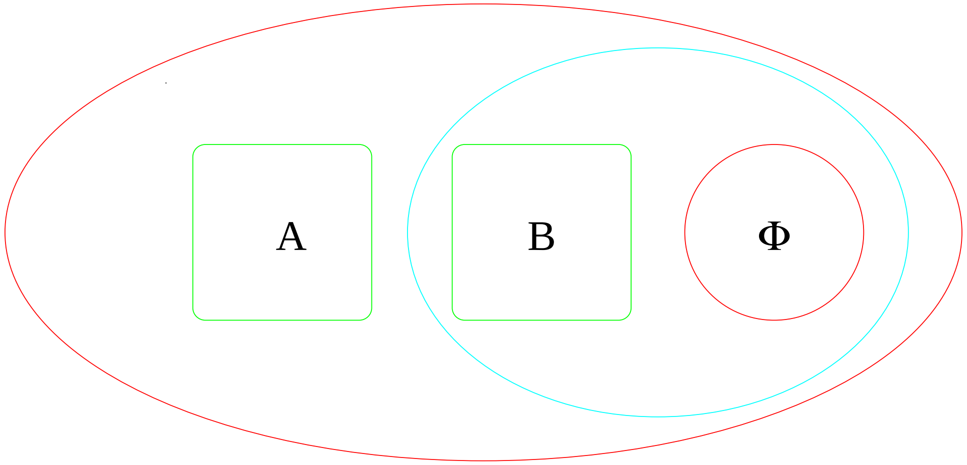
$$\mathcal{B}^{E}_{A|B}[\Phi] = \mathcal{B}^{E}_{A}[\mathcal{B}^{E\cup E_A}_{B}[\Phi]].$$



Compositionality of PIOA Behaviors

Theorem [*Wu, Stark, Smolka, Liu*] Suppose A and B are compatible PIOAs, and E is a set of actions. Then

$$\mathcal{B}^{E}_{A|B}[\Phi] = \mathcal{B}^{E}_{A}[\mathcal{B}^{E \cup E_A}_{B}[\Phi]].$$



Compositional Performance Analysis

Goal: Given closed PIOA $A = A_1|A_2|\ldots|A_n$ and observable Φ, compute $\mathcal{B}^{\emptyset}_{A}[\Phi](\mathbf{0})$.

Method: Let $E = E_1 \cup E_2 \cup \ldots \cup E_n$.

1. **Apply** successively, the behaviors of A_1, A_1, ..., A_n:
$$\Phi, \quad \mathcal{B}^{E}_{A_1}[\Phi], \quad \mathcal{B}^{E}_{A_2}[\mathcal{B}^{E}_{A_1}[\Phi]], \quad \ldots, \quad \mathcal{B}^{E}_{A}[\Phi].$$
2. **Restrict** $\mathcal{B}^{E}_{A}[\Phi]$ to $\mathcal{B}^{\emptyset}_{A}[\Phi]$.
3. **Evaluate** $\mathcal{B}^{\emptyset}_{A}[\Phi](\mathbf{0})$.

A suitable finite representation of observables is required to implement this method.

Linear Representations

An n-*dimensional linear representation* is a tuple:

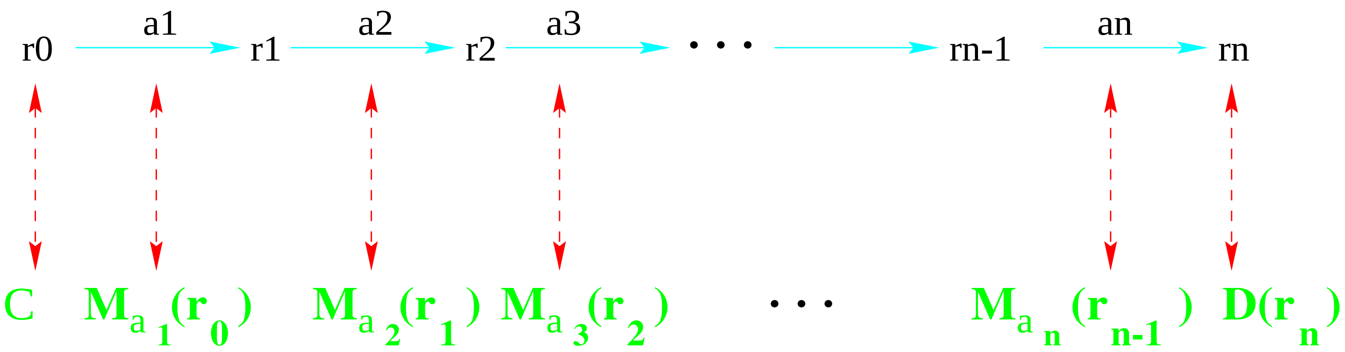
$$R = (C,\ \{M_a(x) : a \in E\},\ D(x))$$

where:

- The *start vector* C is an n-dimensional row vector with *rational numbers* as entries.
- The *transition matrices* $M_a(x)$ are $n \times n$ matrices whose entries are *rational functions* in the unknown (rate) x.
- The *output vector* $D(x)$ is an n-dimensional column vector whose entries are *rational functions* in x.

Linear Representations as Automata on Rated Traces

Linear representations can be viewed as "vector automata" that execute on rated traces.



The ith transition multiplies the current state vector by the matrix $M_{a_i}(r_{i-1})$.

(*cf.* linear representations of rational formal power series)

Representable Observables

An observable: $\Phi : \text{Traces}(E) \to \mathbf{R}$ is *representable* if there exists a linear representation

$$R = (C,\ \{M_a(x) : a \in E\},\ D(x))$$

such that for all rated traces

$$\alpha : r_0 \xrightarrow{a_1} r_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} r_k$$

the quantity $\Phi(\alpha)$ is given by:

$$\Phi(\alpha) \;=\; C \left(\prod_{i=1}^{k} M_{a_i}(r_{i-1}) \right) D(r_k).$$

(*Notation*: Φ_R is the observable represented by R.)

Operations on Representations

The following operations on representations are computable:

- **Evaluation** of a representation on a rated trace:

$$(R, \alpha) \qquad \Rightarrow \qquad \Phi_R(\alpha)$$

- **Application** of a PIOA behavior to a representation

$$(A, R) \;\; \Rightarrow \;\; R' \qquad\qquad \text{with } \Phi_{R'} = \mathcal{B}_A^E[\Phi_R]$$

- **Restriction** of a representation to a subset of actions:

$$R \;\; \Rightarrow \;\; R' \qquad\qquad \text{with } \Phi_{R'}(\alpha') = \sum_{\alpha \gg \alpha'} \Phi_R(\alpha)$$

- **Minimization** of the dimension of a representation:

Evaluation

Given a linear representation:

$$R = (C,\ \{M_a(x) : a \in E\},\ D(x))$$

and a rated trace (with rational rates):

$$\alpha : r_0 \xrightarrow{a_1} r_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} r_k$$

we may compute:

$$\Phi_R(\alpha) \ = \ C \left(\prod_{i=1}^{k} M_{a_i}(r_{i-1}) \right) D(r_k)$$

Application

Application forms a kind of product of a PIOA and a representation. Given n-dimensional representation

$$R = (C,\ \{M_a(x) : a \in E\},\ D(x))$$

and m-state PIOA

$$A = (E_A, Q, \iota, \mu, \rho)$$

with $E_A \subseteq E$, form mn-dimensional representation

$$R' = (C',\ \{M'_a(x) : a \in E\},\ D'(x))$$

such that $\Phi_{R'} = \mathcal{B}_A^E[\Phi_R]$.

$M'_a(x)$ is a m-dimensional matrix of n-dimensional blocks:

$$M'_a(x) = \left(\begin{array}{c|c|c} & j & \\ & & \\ \hline & & \\ i & p_{a,i,j} M_a(x + \rho(q_i)) & \\ & & \\ \hline & & \\ & & \end{array}\right)$$

where

$$p_{a,i,j} = \begin{cases} \mu(q_i, a, q_j)\rho(q_i), & \text{if } a \in E_A^{\mathsf{out}} + E_A^{\mathsf{int}} \\ \mu(q_i, a, q_j), & \text{otherwise} \end{cases}$$

C' and $D'(x)$ are also given in block form:

$$C' = \left(\iota(q_1)\, C \quad \iota(q_2)\, C \quad \ldots \quad \iota(q_m)\, C \right)$$

$$D'(x) = \begin{pmatrix} D(x + \rho(q_1)) \\ D(x + \rho(q_2)) \\ \ldots \\ D(x + \rho(q_m)) \end{pmatrix}$$

Restriction

Restriction removes actions from a representation, by a process of formal summation.

Given n-dimensional representation

$$R = (C,\ \{M_a(x) : a \in E\},\ D(x))$$

and subset F of E, restriction produces the n-dimensional representation:

$$R \backslash F = (C,\ \ M_F^*(x) D(x),\ \ \{M_F^*(x) M_a(x) : a \in E - F\}),$$

where

$$M_F(x) = \sum_{a \in F} M_a(x) \qquad\qquad M_F^*(x) = \sum_{k=0}^{\infty} (M_F(x))^k$$

Note: Given a rated trace

$$\alpha : r_0 \xrightarrow{a_1} r_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} r_k$$

where $a_i \in E - F$ for $1 \leq i \leq k$,

$$\begin{aligned} R \backslash F(\alpha) = \; & C \; M_F^*(r_0) \; M_{a_1}(r_0) \\ & \quad M_F^*(r_1) \; M_{a_2}(r_1) \\ & \quad\quad \ldots \; M_F^*(r_{k-1}) \; M_{a_k}(r_{k-1}) \\ & \quad\quad\quad M_F^*(r_k) \; D_{a_k}(r_k) \end{aligned}$$

so that

$$R \backslash F(\alpha) = \sum_{\alpha' \gg \alpha} R(\alpha')$$

Minimization

Minimization takes an n-dimensional representation

$$R = (C,\ \{M_a(x) : a \in E\},\ D(x))$$

and produces a representation

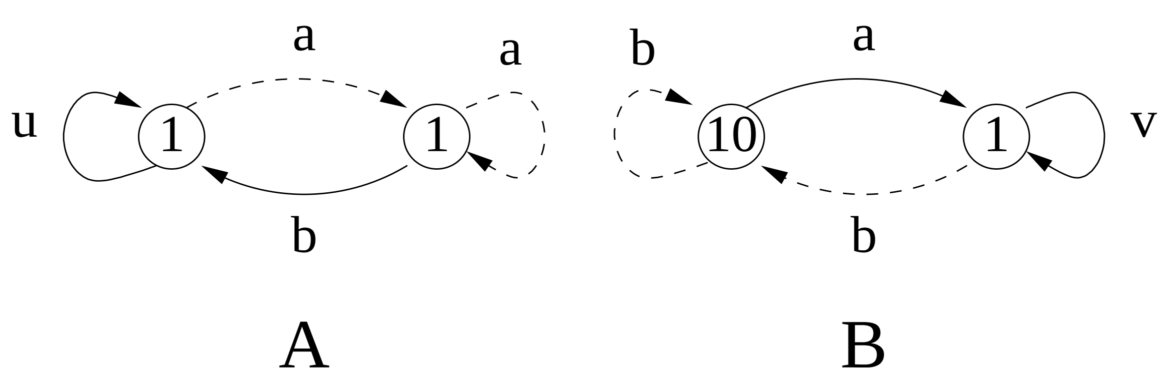
$$R' = (C',\ \{M'_a(x) : a \in E\},\ D'(x))$$

of minimum dimension such that $\Phi_{R'} = \Phi_R$.

Idea: Find and "kill" irrelevant subspaces of n-space:

- The largest *unreachable* subspace.
- The largest *unobservable* subspace.

Simple Example



(all probabilities are 1)

Compute: expected time for $A|B$ to perform first b action

Initial Representation:

$$C = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad M_a = M_t = M_u = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{x} & \frac{1}{x^2} \\ 0 & 0 & 0 & \frac{1}{x} \end{pmatrix}$$

$$M_b = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{x} & \frac{1}{x^2} & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

After applying A (dimension 8, minimizes to dimension 5):

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \qquad M_a = \begin{pmatrix} 0 & 0 & \frac{1}{x+1} & 0 & 0 \\ 0 & \frac{1}{x+1} & \frac{1}{(x+1)^2} & 0 & 0 \\ 0 & 0 & \frac{1}{x+1} & 0 & 0 \\ 0 & \frac{1}{x+1} & \frac{1}{(x+1)^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M_b = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{(x+1)^2} \\ 0 & 0 & 0 & 0 & \frac{1}{(x+1)^2} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M_u = \begin{pmatrix} \frac{1}{x+1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{1}{(x+1)^2} & 0 & 0 & \frac{1}{x+1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M_v = \begin{pmatrix} \frac{1}{x+1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{x+1} & \frac{1}{(x+1)^2} & 0 \\ 0 & 0 & \frac{1}{x+1} & 0 & 0 \\ \frac{1}{(x+1)^2} & 0 & 0 & \frac{1}{x+1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

After applying B (dimension 10, minimizes to dimension 5):

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$M_a = \begin{pmatrix} 0 & 0 & 0 & \frac{10}{x+11} & 0 \\ 0 & 0 & \frac{10}{x+11} & \frac{10}{(x+11)^2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad M_b = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{(x+2)^2} \\ 0 & 0 & 0 & 0 & \frac{1}{x+2} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M_u = \begin{pmatrix} \frac{1}{x+11} & 0 & 0 & 0 & 0 \\ \frac{1}{(x+11)^2} & \frac{1}{x+11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} M_v = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{x+2} & \frac{1}{(x+2)^2} & 0 \\ 0 & 0 & 0 & \frac{1}{x+2} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Restriction and evaluation to obtain final result:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{x+11} & 0 & 0 & \frac{10}{x+11} & 0 \\ \frac{1}{(x+11)^2} & \frac{1}{x+11} & \frac{10}{x+11} & \frac{10}{(x+11)^2} & 0 \\ 0 & 0 & \frac{1}{x+2} & \frac{1}{(x+2)^2} & \frac{1}{(x+2)^2} \\ 0 & 0 & 0 & \frac{1}{x+2} & \frac{1}{x+2} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}^* \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Result: 11/10.

Implementation: PIOATool

Currently about 12,000 lines of Standard ML.

- Exact rational arithmetic
- Polynomial and rational functions
- PIOAs
- Sparse vectors and matrices
- Representations and analysis algorithms

PIOATool: Functionality

- *Basic systems* are defined directly as PIOAs,
- *Composite systems* are constructed by composing collections of subsystems.
- *Performance measures* (transient) can be defined. E.g.:
  - Probability of event occurrence
  - Expected time until (or reward at) event occurrence
- *Compositional performance analysis* treats components one-at-a-time, minimizing at each step.
- Tool computes *exact results* using *rational arithmetic* and can perform *symbolic analysis*.

PIOATool: Performance

Jai-alai Example: Exact probability of first player winning an n-player, p-point match ($2n+1$ components).

Players (n)	**Points** (p)	**States**	**Time**
2	2	12,005	13 seconds
2	3	19,845	23 seconds
2	4	29,645	41 seconds
2	6	55,125	124 seconds
2	8	88,445	318 seconds
3	2	878,080	302 seconds
4	2	78,764,805	13850 seconds

(Timings on 333MHz UltraSparc.)

Mutual Exclusion Example

Expected time to get resource in an n-user system (n+1 components).

Users (n)	**States**	**Time**
2	175	3 seconds
3	1125	19 seconds
4	6875	112 seconds
5	40625	732 seconds
6	234375	5354 seconds

(Timings on Pentium II/450MHz.)

Future Work

Theory

- Steady-state calculations via symbolic analysis
- Relation to bisimulation-based reductions
- Phase-type distributions
- Tensor representations, BDDs

Implementation

- Floating point vs. exact rational arithmetic
- Language for specifying systems, performance measures
- Improve PIOATool capabilities, performance
- Integrate with Concurrency Workbench