UNIVERSITY OF TWENTE.

Formal Methods & Tools.

A linear process algebraic format for probabilistic systems

Mark Timmer

January 19, 2010

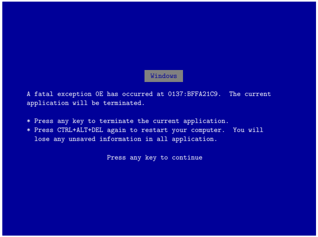
Joint work with Joost-Pieter Katoen, Jaco van de Pol, and Mariëlle Stoelinga

FMT Lunchmeeting

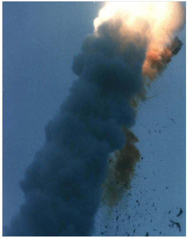
Contents

Introduction – Dependability

Dependability of computer systems is becoming more and more important.



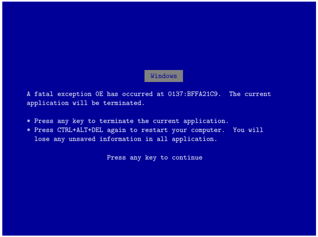
Windows blue screen



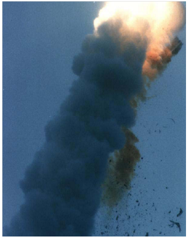
Ariane 5 crash

Introduction – Dependability

Dependability of computer systems is becoming more and more important.



Windows blue screen

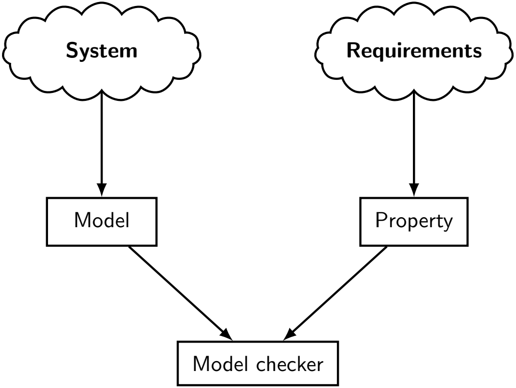


Ariane 5 crash

Our aim: use quantitative formal methods to improve system quality.

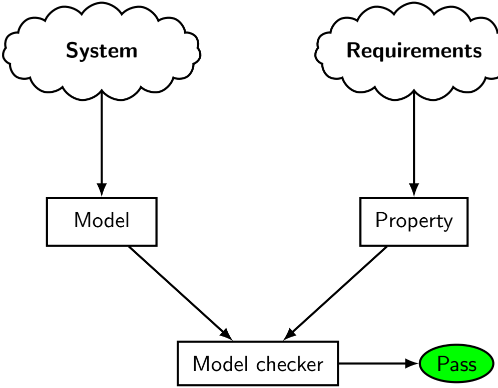
Introduction – Model Checking

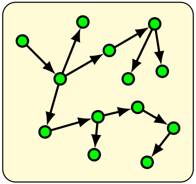
A popular solution is model checking; verifying properties of a system by constructing a model and ranging over its state space.



Introduction – Model Checking

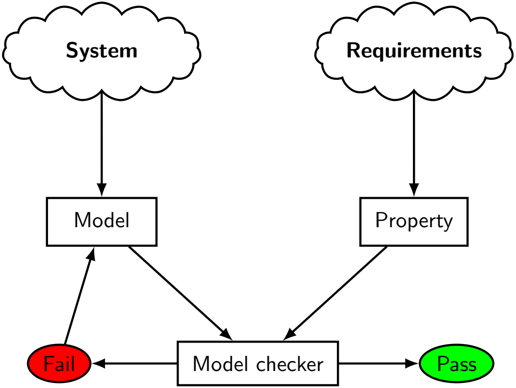
A popular solution is model checking; verifying properties of a system by constructing a model and ranging over its state space.

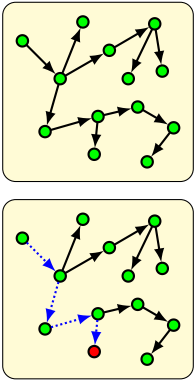




Introduction – Model Checking

A popular solution is model checking; verifying properties of a system by constructing a model and ranging over its state space.





Introduction – Probabilistic Model Checking

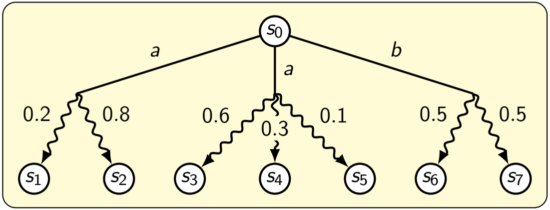
Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., a probabilistic automaton)

Introduction – Probabilistic Model Checking

Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., a probabilistic automaton)

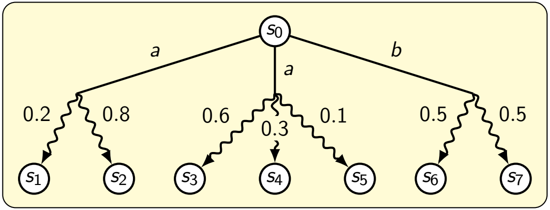


- Non-deterministically choose one of the three transitions
- Probabilistically choose the next state

Introduction – Probabilistic Model Checking

Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model (e.g., a probabilistic automaton)



- Non-deterministically choose one of the three transitions
- Probabilistically choose the next state

Applications:

- Dependability analysis
- Performance analysis

Introduction – Probabilistic Model Checking

Limitations of previous approaches:

- Susceptible to the state space explosion problem
- Restricted treatment of data

Introduction – Probabilistic Model Checking

Limitations of previous approaches:

- Susceptible to the state space explosion problem
- Restricted treatment of data

Our approach:

1. Define a probabilistic process algebra (prCRL), incorporating both data types and probabilistic choice
2. Define a linear format (the LPPE), enabling symbolic optimisations at the language level
3. Develop and implement a linearisation algorithm
4. Reduce state spaces before they are generated by manipulations of the linear format.

Strong probabilistic bisimulation

Equivalent PAs: strongly probabilistic bisimilar PAs

Strong probabilistic bisimulation

Equivalent PAs: strongly probabilistic bisimilar PAs

Strong bisimulation

An *equivalence relation* R is a strong bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} p'$ imply that $q \xrightarrow{a} q'$ such that $(p', q') \in R$.

Strong probabilistic bisimulation

Equivalent PAs: strongly probabilistic bisimilar PAs

Strong bisimulation

An *equivalence relation* R is a strong bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} p'$ imply that $q \xrightarrow{a} q'$ such that $(p', q') \in R$.

Strong probabilistic bisimulation

An *equivalence relation* R is a strong probabilistic bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} \mu$ imply that $q \xrightarrow{a} \mu'$ such that $\mu \equiv_R \mu'$

Strong probabilistic bisimulation

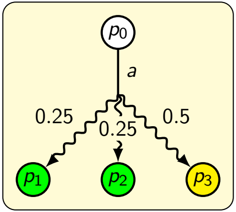
Equivalent PAs: strongly probabilistic bisimilar PAs

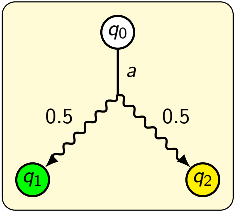
Strong bisimulation

An *equivalence relation* R is a strong bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} p'$ imply that $q \xrightarrow{a} q'$ such that $(p', q') \in R$.

Strong probabilistic bisimulation

An *equivalence relation* R is a strong probabilistic bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} \mu$ imply that $q \xrightarrow{a} \mu'$ such that $\mu \equiv_R \mu'$





Strong probabilistic bisimulation

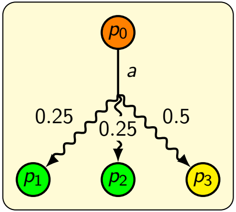
Equivalent PAs: strongly probabilistic bisimilar PAs

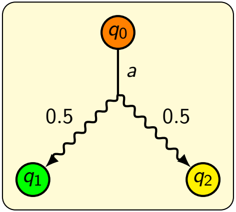
Strong bisimulation

An *equivalence relation* R is a strong bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} p'$ imply that $q \xrightarrow{a} q'$ such that $(p', q') \in R$.

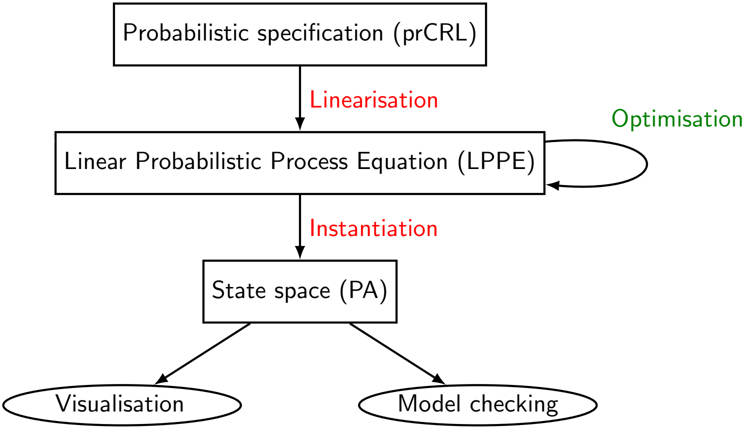
Strong probabilistic bisimulation

An *equivalence relation* R is a strong probabilistic bisimulation if $(p, q) \in R$ and $p \xrightarrow{a} \mu$ imply that $q \xrightarrow{a} \mu'$ such that $\mu \equiv_R \mu'$





Introduction – overview of our approach



A process algebra with data and probability: prCRL

Specification language prCRL:

- Based on μCRL (so data), with additional probabilistic choice
- Operational semantics defined in terms of probabilistic automata
- Minimal set of operators to facilitate formal manipulation
- Syntactic sugar easily definable

A process algebra with data and probability: prCRL

The grammar of prCRL process terms

Process terms in prCRL are obtained by the following grammar:

$$p ::= Y(\vec{t}) \mid c \Rightarrow p \mid p + p \mid \sum_{x:D} p \mid a(\vec{t}) \sum_{x:D}\!\!\bullet\, f : p$$

- c is a condition (boolean expression)
- a is an atomic action
- f is a real-valued expression yielding values in $[0, 1]$
- $\vec{t}$ is a vector of expressions

A process algebra with data and probability: prCRL

The grammar of prCRL process terms

Process terms in prCRL are obtained by the following grammar:

$$p ::= Y(\vec{t}) \mid c \Rightarrow p \mid p + p \mid \sum_{x:D} p \mid a(\vec{t}) \textstyle\sum\!\!\!\bullet_{x:D} f : p$$

- c is a condition (boolean expression)
- a is an atomic action
- f is a real-valued expression yielding values in $[0, 1]$
- $\vec{t}$ is a vector of expressions

Process equations and processes

A process equation is something of the form $X(\vec{g} : \vec{G}) = p$.

Some examples

Sending an arbitrary natural number

$$X = \tau \sum_{n:\mathbb{N}^{>0}} \frac{1}{2^n} : (\mathsf{send}(n) \cdot \sum_{j:\{*\}} 1.0 : X)$$

Some examples

Sending an arbitrary natural number

$$X = \tau \sum_{n:\mathbb{N}^{>0}} \tfrac{1}{2^n} : (\mathrm{send}(n) \cdot X)$$

Some examples

Sending an arbitrary natural number

$$X = \tau \sum_{n:\mathbb{N}^{>0}} \bullet \frac{1}{2^n} : (\mathrm{send}(n) \cdot X)$$

Sending ping messages until system crash

$$X = \mathrm{ping} \sum_{i:\{1,2\}} \bullet (i = 1 \ ? \ 0.1 \ : \ 0.9) : ((i = 1 \Rightarrow \mathrm{crash}) + (i \neq 1 \Rightarrow X))$$

Some examples

Sending an arbitrary natural number

$$X = \tau \sum_{n:\mathbb{N}^{>0}} \frac{1}{2^n} : (\mathsf{send}(n) \cdot X)$$

Sending ping messages until system crash

$$X = \mathsf{ping}(0.1 : \mathsf{crash} \oplus 0.9 : X)$$

Some examples

Sending an arbitrary natural number

$$X = \tau \sum_{n:\mathbb{N}^{>0}} \frac{1}{2^n} : (\mathsf{send}(n) \cdot X)$$

Sending ping messages until system crash

$$X = \mathsf{ping}(0.1 : \mathsf{crash} \oplus 0.9 : X)$$

Writing all Fibonacci numbers

$$X(p : \mathbb{N}, pp : \mathbb{N}) = \mathsf{write}(\mathsf{plus}(p, pp)) \cdot X(\mathsf{plus}(p, pp), p)$$

Operational semantics

$$\textsc{NChoice-L}\ \frac{p \xrightarrow{\alpha} \mu}{p + q \xrightarrow{\alpha} \mu}$$

$$\textsc{Implies}\ \frac{p \xrightarrow{\alpha} \mu}{c \Rightarrow p \xrightarrow{\alpha} \mu}\ \text{if } c \text{ holds}$$

Operational semantics

$$\textsc{NChoice-L}\ \frac{p \xrightarrow{\alpha} \mu}{p + q \xrightarrow{\alpha} \mu}$$

$$\textsc{Implies}\ \frac{p \xrightarrow{\alpha} \mu}{c \Rightarrow p \xrightarrow{\alpha} \mu} \text{ if } c \text{ holds}$$

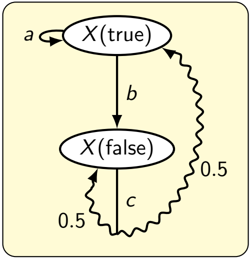
An arbitrary specification

$$\begin{aligned} X(x : \text{Bool}) = x \quad & \Rightarrow a \cdot X(x) + b \cdot X(\text{not}(x)) \\ + \text{not}(x) & \Rightarrow c(0.5 : X(\text{false}) \oplus 0.5 : X(\text{true})) \end{aligned}$$

Operational semantics

$$\textsc{NChoice-L}\ \frac{p \xrightarrow{\alpha} \mu}{p + q \xrightarrow{\alpha} \mu}$$

$$\textsc{Implies}\ \frac{p \xrightarrow{\alpha} \mu}{c \Rightarrow p \xrightarrow{\alpha} \mu} \text{ if } c \text{ holds}$$



An arbitrary specification

$$\begin{aligned} X(x : \mathrm{Bool}) = x \quad & \Rightarrow a \cdot X(x) + b \cdot X(\mathrm{not}(x)) \\ + \mathrm{not}(x) & \Rightarrow c(0.5 : X(\mathrm{false}) \oplus 0.5 : X(\mathrm{true})) \end{aligned}$$

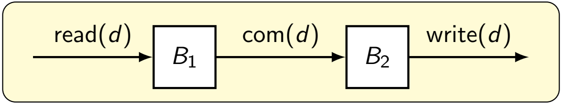
Linear process equations

In the non-probabilistic setting, LPEs are given by

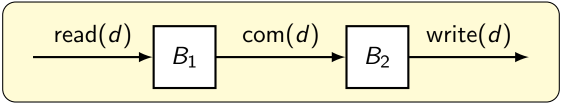
$$X(\vec{g} : \vec{G}) = \sum_{\vec{d_1}:\vec{D_1}} c_1 \Rightarrow a_1(b_1) \cdot X(\vec{n_1})$$
$$\dots$$
$$+ \sum_{\vec{d_k}:\vec{D_k}} c_k \Rightarrow a_k(b_k) \cdot X(\vec{n_k})$$

- $\vec{G}$ is a type for state vectors
- $\vec{D_i}$ a type for local variable vectors for summand i
- c_i is the enabling condition of summand i
- a_i is an atomic action, with action-parameter vector b_i
- $\vec{n_i}$ is the next-state vector of summand i.

Linear process equations – An example



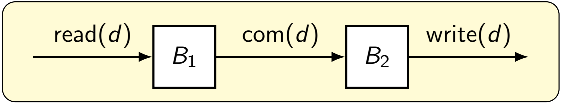
Linear process equations – An example



$$B_1 = \sum_{d:D} \text{read}(d) \cdot \text{com}(d) \cdot B_1$$

$$B_2 = \sum_{d:D} \overline{\text{com}}(d) \cdot \text{write}(d) \cdot B_2$$

Linear process equations – An example



$$B_1 = \sum_{d:D} \mathrm{read}(d) \cdot \mathrm{com}(d) \cdot B_1$$

$$B_2 = \sum_{d:D} \overline{\mathrm{com}}(d) \cdot \mathrm{write}(d) \cdot B_2$$

$$\begin{aligned}
X(a : \{1,2\}, b : \{1,2\}, x : D, y : D) = \\
\textstyle\sum_{d:D} \quad & a = 1 && \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) && (1)\\
+ \quad & a = 2 \wedge b = 1 && \Rightarrow \mathrm{com}(x) \cdot X(1, 2, x, x) && (2)\\
+ \quad & b = 2 && \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) && (3)
\end{aligned}$$

A linear format for prCRL: the LPPE

In the probabilistic setting, LPPEs are given by

$$X(\vec{g} : \vec{G}) = \sum_{\vec{d_1}:\vec{D_1}} c_1 \Rightarrow a_1(b_1) \sum\nolimits^{\bullet}_{\vec{e_1}:\vec{E_1}} f_1 : X(\vec{n_1})$$

$$\dots$$

$$+ \sum_{\vec{d_k}:\vec{D_k}} c_k \Rightarrow a_k(b_k) \sum\nolimits^{\bullet}_{\vec{e_k}:\vec{E_k}} f_k : X(\vec{n_k})$$

A linear format for prCRL: the LPPE

In the probabilistic setting, LPPEs are given by

$$
\begin{aligned}
X(\vec{g} : \vec{G}) = & \sum_{\vec{d_1}:\vec{D_1}} c_1 \Rightarrow a_1(b_1) \sum\!\!\!\!\bullet_{\vec{e_1}:\vec{E_1}} f_1 : X(\vec{n_1}) \\
& \ldots \\
& + \sum_{\vec{d_k}:\vec{D_k}} c_k \Rightarrow a_k(b_k) \sum\!\!\!\!\bullet_{\vec{e_k}:\vec{E_k}} f_k : X(\vec{n_k})
\end{aligned}
$$

Advantages of using LPPEs instead of prCRL specifications:

- Easy state space generation
- Straight-forward parallel composition
- **Symbolic optimisations enabled at the language level**

A linear format for prCRL: the LPPE

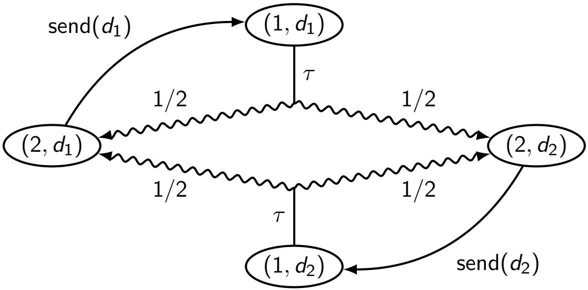
An example

$$X(\mathrm{pc} : \{1, 2\}, d : D) = \mathrm{pc} = 1 \Rightarrow \tau \sum_{e:D}^{\bullet} \frac{1}{|D|} : X(2, e)$$
$$+ \mathrm{pc} = 2 \Rightarrow \mathrm{send}(d) \cdot X(1, d))$$

A linear format for prCRL: the LPPE

An example

$$X(\text{pc} : \{1, 2\}, d : D) = \text{pc} = 1 \Rightarrow \tau \sum_{e:D}^{\bullet} \frac{1}{|D|} : X(2, e)$$
$$+ \text{pc} = 2 \Rightarrow \text{send}(d) \cdot X(1, d))$$

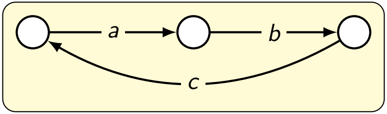


Linearisation

$$X = a \cdot b \cdot c \cdot X$$

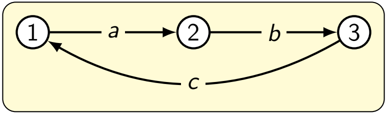
Linearisation

$$X = a \cdot b \cdot c \cdot X$$



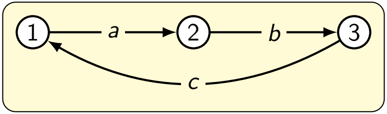
Linearisation

$$X = a \cdot b \cdot c \cdot X$$



Linearisation

$$X = a \cdot b \cdot c \cdot X$$



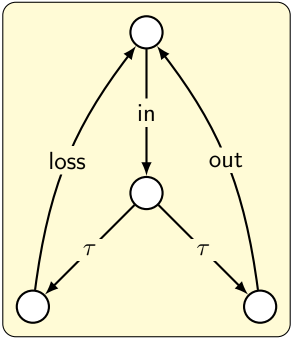
$$\begin{aligned} Y(pc\colon \{1,2,3\}) = & \\ & pc = 1 \Rightarrow a \cdot Y(2) \\ + & pc = 2 \Rightarrow b \cdot Y(3) \\ + & pc = 3 \Rightarrow c \cdot Y(1) \end{aligned}$$

Linearisation

$$X = \sum_{d:D} \mathrm{in}(d) \cdot (\tau \cdot \mathrm{loss} \cdot X + \tau \cdot \mathrm{out}(d) \cdot X)$$

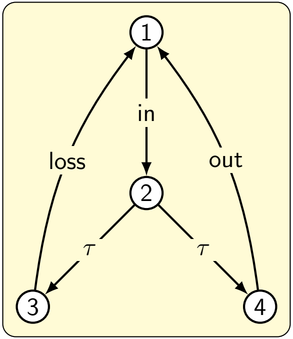
Linearisation

$$X = \sum_{d:D} \mathrm{in}(d) \cdot (\tau \cdot \mathrm{loss} \cdot X + \tau \cdot \mathrm{out}(d) \cdot X)$$



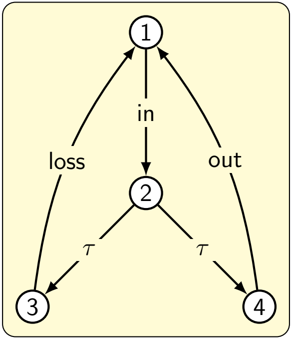
Linearisation

$$X = \sum_{d:D} \mathrm{in}(d) \cdot (\tau \cdot \mathrm{loss} \cdot X + \tau \cdot \mathrm{out}(d) \cdot X)$$



Linearisation

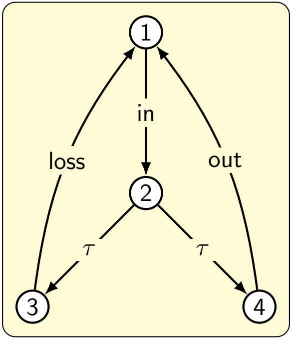
$$X = \sum_{d:D} \text{in}(d) \cdot (\tau \cdot \text{loss} \cdot X + \tau \cdot \text{out}(d) \cdot X)$$



$$\begin{aligned} Y(pc\colon \{1,2,3,4\}, x\colon D) = & \\ \textstyle\sum_{d:D} \quad & pc = 1 \Rightarrow \text{in}(d) \cdot Y(2,d) \\ + \quad & pc = 2 \Rightarrow \tau \cdot Y(3,x) \\ + \quad & pc = 2 \Rightarrow \tau \cdot Y(4,x) \\ + \quad & pc = 3 \Rightarrow \text{loss} \cdot Y(1,x) \\ + \quad & pc = 4 \Rightarrow \text{out}(x) \cdot Y(1,x) \end{aligned}$$

Linearisation

$$X = \sum_{d:D} \text{in}(d) \cdot (\tau \cdot \text{loss} \cdot X + \tau \cdot \text{out}(d) \cdot X)$$



$$\begin{aligned} & Y(pc\colon \{1, 2,3, 4\}, x\colon D) = \\ & \quad \textstyle\sum_{d:D} \quad pc = 1 \Rightarrow \text{in}(d) \cdot Y(2, d) \\ & \quad + \qquad pc = 2 \Rightarrow \tau \cdot Y(3, x) \\ & \quad + \qquad pc = 2 \Rightarrow \tau \cdot Y(4, x) \\ & \quad + \qquad pc = 3 \Rightarrow \text{loss} \cdot Y(1, x) \\ & \quad + \qquad pc = 4 \Rightarrow \text{out}(x) \cdot Y(1, x) \end{aligned}$$

Initial process: $Y(1, d_1)$.

Linearisation

$$X(d : D) = \sum_{e:D} a(d+e) \sum\!\!\!\!\!\!\bullet_{f:D} \ \tfrac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D}\!\bullet\; \tfrac{1}{|D|} : \left(c(e)\cdot c(f)\cdot X(5) + c(e+f)\cdot X(5)\right)$$

1 $X_1(d:D, e:D, f:D) =$
$\sum_{e:D} a(d+e) \sum_{f:D}\!\bullet\; \frac{1}{|D|} : \; (c(e)\cdot c(f)\cdot X(5) + c(e+f)\cdot X(5))$

Linearisation

$$X(d : D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} : \Big(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \Big)$$

1 $X_1(d : D, e : D, f : D) =$
$\sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} : (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

Linearisation

$$X(d : D) = \sum_{e:D} a(d + e) \sum_{f:D}\!\!\bullet \ \tfrac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e + f) \cdot X(5)\right)$$

1 $X_1(d : D, e : D, f : D) =$
$\quad \sum_{e:D} a(d + e) \sum\!\!\bullet_{f:D} \frac{1}{|D|} : (c(e) \cdot c(f) \cdot X(5) + c(e + f) \cdot X(5))$

---

2 $X_1(d : D, e : D, f : D) = \sum_{e:D} a(d + e) \sum\!\!\bullet_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$

Linearisation

$$X(d : D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$$

1 $X_1(d : D, e : D, f : D) =$
$\sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

---

2 $X_1(d : D, e : D, f : D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d : D, e : D, f : D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

Linearisation

$$X(d : D) = \sum_{e:D} a(d + e) \sum_{f:D} \bullet \frac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e + f) \cdot X(5) \right)$$

1 $X_1(d : D, e : D, f : D) =$
$\sum_{e:D} a(d + e) \sum_{f:D} \bullet \frac{1}{|D|} : (c(e) \cdot c(f) \cdot X(5) + c(e + f) \cdot X(5))$

---

2 $X_1(d : D, e : D, f : D) = \sum_{e:D} a(d + e) \sum_{f:D} \bullet \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d : D, e : D, f : D) = c(e) \cdot c(f) \cdot X(5) + c(e + f) \cdot X(5)$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$$

1 $X_1(d:D, e:D, f:D) =$
$\sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$

---

2 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

---

3 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$

Linearisation

$$X(d : D) = \sum_{e:D} a(d+e) \sum_{f:D} \tfrac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$$

1 $X_1(d : D, e : D, f : D) =$
$\sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \ (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

---

2 $X_1(d : D, e : D, f : D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d : D, e : D, f : D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

---

3 $X_1(d : D, e : D, f : D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d : D, e : D, f : D) = c(e) \cdot X_3(d, e, f) + c(e+f) \cdot X_1(5, e, f)$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \Big(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \Big)$$

1 $X_1(d:D, e:D, f:D) =$
$\quad \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

---

2 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

---

3 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d, e, f) + c(e+f) \cdot X_1(5, e, f)$
$X_3(d:D, e:D, f:D) = c(f) \cdot X(5)$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \tfrac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$$

1 $X_1(d:D, e:D, f:D) =$
$\sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} : (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

---

2 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

---

3 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d, e, f) + c(e+f) \cdot X_1(5, e, f)$
$X_3(d:D, e:D, f:D) = c(f) \cdot X(5)$

---

4 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} \cdot X_2(d, e, f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d, e, f) + c(e+f) \cdot X_1(5, e, f)$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \Big(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \Big)$$

1 $X_1(d:D, e:D, f:D) =$
$\sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} : \ (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

---

2 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

---

3 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$
$X_3(d:D, e:D, f:D) = c(f) \cdot X(5)$

---

4 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D}^{\bullet} \tfrac{1}{|D|} : \Big(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \Big)$$

1 $X_1(d:D, e:D, f:D) =$
$\sum_{e:D} a(d+e) \sum^{\bullet}_{f:D} \frac{1}{|D|} : \ (c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5))$

---

2 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum^{\bullet}_{f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5)$

---

3 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum^{\bullet}_{f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$
$X_3(d:D, e:D, f:D) = c(f) \cdot X(5)$

---

4 $X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum^{\bullet}_{f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$
$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$
$X_3(d:D, e:D, f:D) = c(f) \cdot X_1(5,e,f)$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} : \left(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \right)$$

$$X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{f:D} \bullet \frac{1}{|D|} \cdot X_2(d,e,f)$$
$$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$$
$$X_3(d:D, e:D, f:D) = c(f) \cdot X_1(5,e,f)$$

Linearisation

$$X(d:D) = \sum_{e:D} a(d+e) \sum_{\bullet f:D} \frac{1}{|D|} : \Big(c(e) \cdot c(f) \cdot X(5) + c(e+f) \cdot X(5) \Big)$$

$$X_1(d:D, e:D, f:D) = \sum_{e:D} a(d+e) \sum_{\bullet f:D} \frac{1}{|D|} \cdot X_2(d,e,f)$$
$$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$$
$$X_3(d:D, e:D, f:D) = c(f) \cdot X_1(5,e,f)$$

$$X(\mathrm{pc}:\{1,2,3\}, d:D, e:D, f:D) =$$
$$\mathrm{pc}=1 \Rightarrow \sum_{e:D} a(d+e) \sum_{\bullet f:D} \frac{1}{|D|} \cdot X(2,d,e,f)$$
$$+\ \mathrm{pc}=2 \Rightarrow c(e) \cdot X(3,d,e,f)$$
$$+\ \mathrm{pc}=2 \Rightarrow c(e+f) \cdot X(1,5,e,f)$$
$$+\ \mathrm{pc}=3 \Rightarrow c(f) \cdot X(1,5,e,f)$$

Linearisation

In general, we always linearise in two steps:

1. Transform the specification to intermediate regular form (IRF) (every process is a summation of single-action terms)
2. Merge all processes into one big process by introducing a program counter

In the first step, global parameters are introduced to remember the values of bound variables.

Theorem

A specification S and the specification S' obtained by linearising S are strongly probabilistic bisimilar.

Extended prCRL

For compositionality we introduce extended prCRL. It extends prCRL by parallel composition, encapsulation, hiding and renaming.

Extended prCRL

For compositionality we introduce extended prCRL. It extends prCRL by parallel composition, encapsulation, hiding and renaming.

The grammar of extended prCRL process terms

Process terms in extended prCRL are obtained by:

$$q ::= p \mid q \,||\, q \mid \partial_E(q) \mid \tau_H(q) \mid \rho_R(q)$$

- $q_1 \,||\, q_2$: parallel composition with ACP-style communication
- $\partial_E(q)$: encapsulation of all actions in E
- $\tau_H(q)$: hiding of all actions in H
- $\rho_R(q)$: renaming of actions according to the function R

Operational semantics of parallel composition

$$\textsc{Par-L}\ \frac{p \xrightarrow{\alpha} \mu}{p \,||\, q \xrightarrow{\alpha} \mu'} \text{ where } \forall p' \,.\, \mu'(p' \,||\, q) = \mu(p')$$

Operational semantics of parallel composition

$$\text{PAR-L}\ \frac{p \xrightarrow{\alpha} \mu}{p \,||\, q \xrightarrow{\alpha} \mu'} \text{ where } \forall p' \,.\, \mu'(p' \,||\, q) = \mu(p')$$

$$X(d : D) = \text{out}(d) \sum_{d' : D} \frac{1}{|D|} : X(d') \qquad Y(n : \mathbb{N}) = \text{write}(n) \cdot Y(n+1)$$

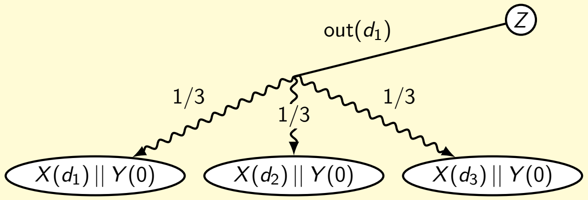
$$Z = X(d_1) \,||\, Y(0)$$

Operational semantics of parallel composition

$$\textsc{Par-L}\ \frac{p \xrightarrow{\alpha} \mu}{p \,||\, q \xrightarrow{\alpha} \mu'} \text{ where } \forall p' \,.\, \mu'(p' \,||\, q) = \mu(p')$$

$$X(d : D) = \mathrm{out}(d) \sum_{d' : D} \frac{1}{|D|} : X(d') \qquad Y(n : \mathbb{N}) = \mathrm{write}(n) \cdot Y(n+1)$$

$$Z = X(d_1) \,||\, Y(0)$$

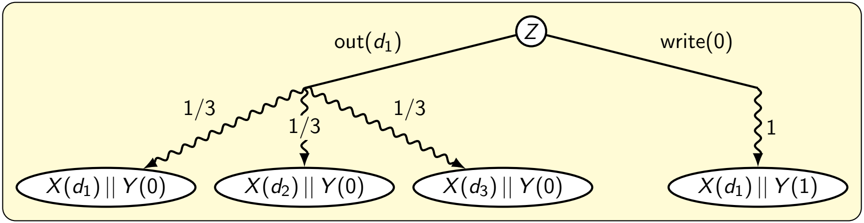


Operational semantics of parallel composition

$$\textsc{Par-L}\ \frac{p \xrightarrow{\alpha} \mu}{p \,||\, q \xrightarrow{\alpha} \mu'} \text{ where } \forall p' \,.\, \mu'(p' \,||\, q) = \mu(p')$$

$$X(d : D) = \mathsf{out}(d) \sum_{d' : D} \frac{1}{|D|} : X(d') \qquad Y(n : \mathbb{N}) = \mathsf{write}(n) \cdot Y(n+1)$$

$$Z = X(d_1) \,||\, Y(0)$$



Operational semantics of parallel composition

For communication we assume a partial function

$$\gamma \colon \mathsf{Act} \times \mathsf{Act} \to \mathsf{Act}$$

When $(a, b) \in \mathrm{dom}(\gamma)$, communication between a and b yields $\gamma(a, b)$.

Operational semantics of parallel composition

For communication we assume a partial function

$$\gamma \colon \text{Act} \times \text{Act} \to \text{Act}$$

When $(a, b) \in \text{dom}(\gamma)$, communication between a and b yields $\gamma(a, b)$.

$$\frac{p \xrightarrow{a(\vec{t})} \mu \qquad q \xrightarrow{b(\vec{t})} \mu'}{p \,||\, q \xrightarrow{c(\vec{t})} \mu''} \ \text{if } \gamma(a, b) = c, \ \forall p', q' \,.\, \mu''(p' \,||\, q') = \mu(p') \cdot \mu'(q')$$

Operational semantics of parallel composition

$$X(n : \{2,3\}) = \text{write}(n) \cdot X(n) + c \sum_{n' : \{2,3\}} \frac{1}{2} : X(n')$$

$$Y(m : \{2,3\}) = \text{write}'(m^2) \cdot Y(m) + c' \sum_{m' : \{2,3\}} \frac{1}{2} : Y(m')$$

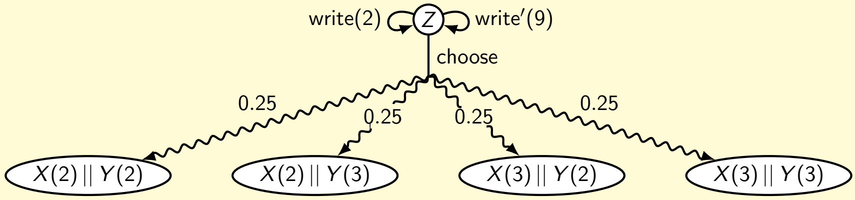
$$Z = \partial_{\{c,c'\}}(X(2) \,||\, Y(3)) \qquad \gamma(c, c') = \text{choose}$$

Operational semantics of parallel composition

$$X(n : \{2,3\}) = \text{write}(n) \cdot X(n) + c \sum_{n' : \{2,3\}} \frac{1}{2} : X(n')$$

$$Y(m : \{2,3\}) = \text{write}'(m^2) \cdot Y(m) + c' \sum_{m' : \{2,3\}} \frac{1}{2} : Y(m')$$

$$Z = \partial_{\{c,c'\}}(X(2) \,||\, Y(3)) \qquad \gamma(c, c') = \text{choose}$$



Linearisation of parallel composition

Linearisation of $X \,||\, Y$ is done compositionally: we first transform X and Y to LPPE, and then put them in parallel.

Linearisation of parallel composition

Linearisation of $X \mid\mid Y$ is done compositionally: we first transform X and Y to LPPE, and then put them in parallel.

$$X(g) = \sum_{d} c \Rightarrow a(b) \sum\nolimits^{\bullet}_{e} f : X(n)$$

$$Y(g') = \sum_{d'} c' \Rightarrow a'(b') \sum\nolimits^{\bullet}_{e'} f' : Y(n')$$

Linearisation of parallel composition

Linearisation of $X \,||\, Y$ is done compositionally: we first transform X and Y to LPPE, and then put them in parallel.

$$X(g) = \sum_{d} c \Rightarrow a(b) \sum\!\!\!\!\bullet_{e} \; f : X(n)$$

$$Y(g') = \sum_{d'} c' \Rightarrow a'(b') \sum\!\!\!\!\bullet_{e'} \; f' : Y(n')$$

$$\begin{aligned} Z(g, g') = &\sum_{d} c \Rightarrow a(b) \sum\!\!\!\!\bullet_{e} \; f : Z(n, g') \\ &+ \sum_{d'} c' \Rightarrow a'(b') \sum\!\!\!\!\bullet_{e'} \; f' : Z(g, n') \\ &+ \sum_{(d,d')} c \wedge c' \wedge b = b' \Rightarrow \gamma(a, a')(b) \sum\!\!\!\!\bullet_{(e,e')} \; f \cdot f' : Z(n, n') \end{aligned}$$

Linearisation of hiding, encapsulation and renaming

Linearisation of hiding, encapsulation and renaming is also done compositionally.

Linearisation of hiding, encapsulation and renaming

Linearisation of hiding, encapsulation and renaming is also done compositionally.

$$
\begin{aligned}
X(g : G) &= \sum_{d_1 : D_1} c_1 \Rightarrow a_1(b_1) \sum\!\!\!\!\bullet_{e_1 : E_1} f_1 : X(n_1) \\
&+ \sum_{d_2 : D_2} c_2 \Rightarrow a_2(b_2) \sum\!\!\!\!\bullet_{e_2 : E_2} f_2 : X(n_2) \\
&+ \sum_{d_3 : D_3} c_3 \Rightarrow a_3(b_3) \sum\!\!\!\!\bullet_{e_3 : E_3} f_3 : X(n_3)
\end{aligned}
$$

Linearisation of hiding, encapsulation and renaming

Linearisation of hiding, encapsulation and renaming is also done compositionally.

$$\begin{aligned}\tau_{\{a_2\}}(X(g : G)) = &\sum_{d_1:D_1} c_1 \Rightarrow a_1(b_1) \sum\!\!\!\bullet_{e_1:E_1} f_1 : X(n_1) \\ &+ \sum_{d_2:D_2} c_2 \Rightarrow a_2(b_2) \sum\!\!\!\bullet_{e_2:E_2} f_2 : X(n_2) \\ &+ \sum_{d_3:D_3} c_3 \Rightarrow a_3(b_3) \sum\!\!\!\bullet_{e_3:E_3} f_3 : X(n_3)\end{aligned}$$

Linearisation of hiding, encapsulation and renaming

Linearisation of hiding, encapsulation and renaming is also done compositionally.

$$\begin{aligned}\tau_{\{a_2\}}(X(g : G)) = & \sum_{d_1:D_1} c_1 \Rightarrow a_1(b_1) \sum_{e_1:E_1}\!\!\bullet\, f_1 : X(n_1) \\ + & \sum_{d_2:D_2} c_2 \Rightarrow \tau \sum_{e_2:E_2}\!\!\bullet\, f_2 : X(n_2) \\ + & \sum_{d_3:D_3} c_3 \Rightarrow a_3(b_3) \sum_{e_3:E_3}\!\!\bullet\, f_3 : X(n_3)\end{aligned}$$

Linearisation of hiding, encapsulation and renaming

Linearisation of hiding, encapsulation and renaming is also done compositionally.

$$\begin{aligned}\tau_{\{a_2\}}(X(g : G)) = & \sum_{d_1:D_1} c_1 \Rightarrow a_1(b_1) \sum_{e_1:E_1}\!\!\bullet\; f_1 : X(n_1) \\ + & \sum_{d_2:D_2} c_2 \Rightarrow \tau \sum_{e_2:E_2}\!\!\bullet\; f_2 : X(n_2) \\ + & \sum_{d_3:D_3} c_3 \Rightarrow a_3(b_3) \sum_{e_3:E_3}\!\!\bullet\; f_3 : X(n_3)\end{aligned}$$

Theorem

A specification S in extended prCRL and the specification S′ obtained by linearising S are strongly probabilistic bisimilar.

Case study: a leader election protocol

Implementation

- Haskell: one-to-one mapping of algorithms to code
- Linearisation: from prCRL to LPPE
- Parallel composition of LPPEs, hiding, renaming, encapsulation

Case study: a leader election protocol

Implementation

- Haskell: one-to-one mapping of algorithms to code
- Linearisation: from prCRL to LPPE
- Parallel composition of LPPEs, hiding, renaming, encapsulation

Case study: leader election protocol à la Itai-Rodeh

- Two processes throw a coin
  - *Both heads or both tails* $\rightarrow$ *throw again*
  - *One of them heads* $\rightarrow$ *this will be the leader*

Case study: a leader election protocol

Implementation

- Haskell: one-to-one mapping of algorithms to code
- Linearisation: from prCRL to LPPE
- Parallel composition of LPPEs, hiding, renaming, encapsulation

Case study: leader election protocol à la Itai-Rodeh

- Two processes throw a coin
  - *Both heads or both tails → throw again*
  - *One of them heads → this will be the leader*
- More precise:
  - *Passive thread: receive value of opponent*
  - *Active thread: flip, send, compare (or block)*

A prCRL model of the leader election protocol

$$P(id : \{1, 2\}, val : D, set : Bool) =$$

A prCRL model of the leader election protocol

$$P(id : \{1, 2\}, val : D, set : Bool) =$$
$$set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true))$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false)
\end{aligned}
$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id : Id) =
\end{aligned}
$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id : Id) = \\
&\quad flip(id) \sum_{d:D}\!\bullet\ \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e).
\end{aligned}
$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id : Id) = \\
&\quad flip(id) \sum_{d:D}\!\!\bullet\; \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e).
\end{aligned}
$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id : Id) = \\
&\quad flip(id) \sum_{d:D} \bullet \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e). \\
&\qquad (\ (d = e \Rightarrow A(id)) \\
&\qquad +(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id)) \\
&\qquad +(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id)))
\end{aligned}
$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id:\{1,2\}, val:D, set:Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id:Id) = \\
&\quad flip(id) \sum_{d:D} \bullet \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e). \\
&\qquad (\ (d = e \Rightarrow A(id)) \\
&\qquad +(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id)) \\
&\qquad +(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id))) \\
&C(id:Id) = \qquad\qquad P(id, heads, false) \,||\, A(id)
\end{aligned}
$$

A prCRL model of the leader election protocol

$$P(id : \{1,2\}, val : D, set : Bool) =$$
$$set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true))$$
$$+ \; set = true \Rightarrow getVal(val).P(id, val, false)$$
$$A(id : Id) =$$
$$flip(id) \sum_{d:D}\!\!\bullet \; \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e).$$
$$(\; (d = e \Rightarrow A(id))$$
$$+(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id))$$
$$+(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id)))$$
$$C(id : Id) = \qquad P(id, heads, false) \,||\, A(id)$$

$$\gamma(getVal, readVal) = checkVal$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id : Id) = \\
&\quad flip(id) \sum_{d:D}\!\!\bullet \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e). \\
&\qquad (\ (d = e \Rightarrow A(id)) \\
&\qquad +(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id)) \\
&\qquad +(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id))) \\
&C(id : Id) = \partial_{getVal, readVal}(P(id, heads, false) \,||\, A(id))
\end{aligned}
$$

$$\gamma(getVal, readVal) = checkVal$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false)) \\
&A(id : Id) = \\
&\quad flip(id) \sum_{d:D} \bullet \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e). \\
&\qquad (\ (d = e \Rightarrow A(id)) \\
&\qquad +(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id)) \\
&\qquad +(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id))) \\
&C(id : Id) = \partial_{getVal, readVal}(P(id, heads, false) \,||\, A(id)) \\
&S = \qquad C(1) \,||\, C(2)
\end{aligned}
$$

$$\gamma(getVal, readVal) = checkVal$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
&P(id : \{1,2\}, val : D, set : Bool) = \\
&\qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
&\quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
&A(id : Id) = \\
&\quad flip(id) \sum_{d:D}\!\!\bullet\, \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e). \\
&\qquad (\ (d = e \Rightarrow A(id)) \\
&\qquad +(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id)) \\
&\qquad +(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id))) \\
&C(id : Id) = \partial_{getVal, readVal}(P(id, heads, false) \,||\, A(id)) \\
&S = \qquad\qquad C(1) \,||\, C(2)
\end{aligned}
$$

$$\gamma(rec, send) = comm \qquad \gamma(getVal, readVal) = checkVal$$

A prCRL model of the leader election protocol

$$
\begin{aligned}
& P(id : \{1,2\}, val : D, set : Bool) = \\
& \qquad set = false \Rightarrow \sum_{d:D} rec(id, other(id), d) \cdot P(id, d, true)) \\
& \quad + set = true \Rightarrow getVal(val).P(id, val, false) \\
& A(id : Id) = \\
& \quad flip(id) \sum_{d:D} \bullet \tfrac{1}{2} : send(other(id), id, d) \cdot \sum_{e:D} readVal(e). \\
& \qquad (\ (d = e \Rightarrow A(id)) \\
& \qquad +(d = heads \wedge d \neq e \Rightarrow leader(id) \cdot A(id)) \\
& \qquad +(d = tails \wedge d \neq e \Rightarrow follower(id) \cdot A(id))) \\
& C(id : Id) = \partial_{getVal, readVal}(P(id, heads, false) \,||\, A(id)) \\
& S = \partial_{send, rec}(C(1) \,||\, C(2))
\end{aligned}
$$

$$\gamma(rec, send) = comm \qquad \gamma(getVal, readVal) = checkVal$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 18 parameters and 14 summands

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 18 parameters and 14 summands

$$Z(pc11 : \{1..1\}, id11 : Ids, val11 : D, set11 : Bool, d11 : D, pc21 : \{1..4\}, id21 : Id, d21 : D, e21 : D, pc12 : \{1..1\}, id12 : Ids, val12 : D, set12 : Bool, d12 : D, pc22 : \{1..4\}, id22 : Id, d22 : D, e22 : D) =$$

$$\ldots$$

$$\sum_{e21:D} pc21 = 3 \land pc11 = 1 \land set11 \land val11 = e21 \Rightarrow checkVal(val11) \sum\!\!\!\bullet_{(k1,k2):\{*\}\times\{*\}} multiply(1.0, 1.0) : Z(1, id11, val11, false, tails, 4, id21, d21, e21, pc12, id12, val12, set12, d12, pc22, id22, d22, e22)$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 18 parameters and 14 summands

$$
\begin{aligned}
&Z(pc11 : \{1..1\}, id11 : Ids, val11 : D, set11 : Bool, d11 : D, pc21 : \{1..4\},\\
&\quad id21 : Id, d21 : D, e21 : D, pc12 : \{1..1\}, id12 : Ids, val12 : D,\\
&\quad set12 : Bool, d12 : D, pc22 : \{1..4\}, id22 : Id, d22 : D, e22 : D) =\\
&\ldots\\
&\sum_{e21:D} pc21 = 3 \wedge pc11 = 1 \wedge set11 \wedge val11 = e21 \Rightarrow\\
&\quad checkVal(val11) \sum_{(k1,k2):\{*\}\times\{*\}} multiply(1.0, 1.0) :\\
&\quad Z(1, id11, val11, false, tails, 4, id21, d21, e21,\\
&\qquad pc12, id12, val12, set12, d12, pc22, id22, d22, e22)
\end{aligned}
$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 18 parameters and 14 summands

$$Z(pc11 : \{1..1\}, id11 : Ids, val11 : D, set11 : Bool, d11 : D, pc21 : \{1..4\}, id21 : Id, d21 : D, e21 : D, pc12 : \{1..1\}, id12 : Ids, val12 : D, set12 : Bool, d12 : D, pc22 : \{1..4\}, id22 : Id, d22 : D, e22 : D) =$$

$$\ldots \sum_{e21:D} pc21 = 3 \wedge pc11 = 1 \wedge set11 \wedge val11 = e21 \Rightarrow checkVal(val11) \sum\!\!\!\!\bullet_{(k1,k2):\{*\}\times\{*\}} multiply(1.0, 1.0) : Z(1, id11, val11, false, tails, 4, id21, d21, e21, pc12, id12, val12, set12, d12, pc22, id22, d22, e22)$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 18 parameters and 14 summands

$$Z(pc11 : \{1..1\}, id11 : Ids, val11 : D, set11 : Bool, d11 : D, pc21 : \{1..4\}, id21 : Id, d21 : D, e21 : D, pc12 : \{1..1\}, id12 : Ids, val12 : D, set12 : Bool, d12 : D, pc22 : \{1..4\}, id22 : Id, d22 : D, e22 : D) =$$

$$\ldots$$

$$\sum_{e21:D} pc21 = 3 \land pc11 = 1 \land set11 \land val11 = e21 \Rightarrow checkVal(val11) \sum_{(k1,k2):\{*\}\times\{*\}}^{\bullet} multiply(1.0, 1.0) : Z(1, id11, val11, false, tails, 4, id21, d21, e21, pc12, id12, val12, set12, d12, pc22, id22, d22, e22)$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
→ LPPE with 18 parameters and 14 summands

$$Z(pc11 : \{1..1\}, id11 : Ids, val11 : D, set11 : Bool, d11 : D, pc21 : \{1..4\},$$
$$id21 : Id, d21 : D, e21 : D, pc12 : \{1..1\}, id12 : Ids, val12 : D,$$
$$set12 : Bool, d12 : D, pc22 : \{1..4\}, id22 : Id, d22 : D, e22 : D) =$$

$$\ldots$$

$$\sum_{e21:D} pc21 = 3 \wedge pc11 = 1 \wedge set11 \wedge val11 = e21 \Rightarrow$$
$$checkVal(val11) \sum_{(k1,k2):\{*\}\times\{*\}}^{\bullet} multiply(1.0, 1.0) :$$
$$Z(1, id11, val11, false, tails, 4, id21, d21, e21,$$
$$pc12, id12, val12, set12, d12, pc22, id22, d22, e22)$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 18 parameters and 14 summands

$$Z(pc11 : \{1..1\}, id11 : Ids, val11 : D, set11 : Bool, d11 : D, pc21 : \{1..4\},$$
$$id21 : Id, d21 : D, e21 : D, pc12 : \{1..1\}, id12 : Ids, val12 : D,$$
$$set12 : Bool, d12 : D, pc22 : \{1..4\}, id22 : Id, d22 : D, e22 : D) =$$

$\ldots$

$$\sum_{e21:D} pc21 = 3 \wedge pc11 = 1 \wedge set11 \wedge val11 = e21 \Rightarrow$$
$$checkVal(val11) \sum\!\!\!\!\bullet_{(k1,k2):\{*\}\times\{*\}} multiply(1.0, 1.0) :$$
$$Z(1, id11, val11, false, tails, 4, id21, d21, e21,$$
$$pc12, id12, val12, set12, d12, pc22, id22, d22, e22)$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 10 parameters and 12 summands

$$
\begin{aligned}
&Z(\mathit{val11} : D, \mathit{set11} : \mathit{Bool}, \mathit{pc21} : 1..4, \mathit{d21} : D, \mathit{e21} : D,\\
&\quad \mathit{val12} : D, \mathit{set12} : \mathit{Bool}, \mathit{pc22} : 1..4, \mathit{d22} : D, \mathit{e22} : D) =\\
&\quad \ldots\\
&\quad \mathit{pc21} = 3 \wedge \mathit{set11} \Rightarrow \mathit{checkVal}(\mathit{val11}) \sum_{k:\{*\}}\!\bullet\, 1.0 :\\
&\quad Z(\mathit{heads}, \mathit{false}, 4, \mathit{d21}, \mathit{val11}, \mathit{val12}, \mathit{set12}, \mathit{pc22}, \mathit{d22}, \mathit{e22})
\end{aligned}
$$

Reductions on the leader election protocol model

In order to obtain reductions: first linearise
$\rightarrow$ LPPE with 10 parameters and 12 summands

$$
\begin{aligned}
&Z(\mathit{val11} : D, \mathit{set11} : \mathit{Bool}, \mathit{pc21} : 1..4, \mathit{d21} : D, \mathit{e21} : D,\\
&\quad \mathit{val12} : D, \mathit{set12} : \mathit{Bool}, \mathit{pc22} : 1..4, \mathit{d22} : D, \mathit{e22} : D) =\\
&\quad \ldots\\
&\quad \mathit{pc21} = 3 \wedge \mathit{set11} \Rightarrow \mathit{checkVal}(\mathit{val11}) \sum_{k:\{*\}}\!\!\bullet\; 1.0 :\\
&\quad Z(\mathit{heads}, \mathit{false}, 4, \mathit{d21}, \mathit{val11}, \mathit{val12}, \mathit{set12}, \mathit{pc22}, \mathit{d22}, \mathit{e22})
\end{aligned}
$$

State space: from 127 to 93 states.

Conclusions and Future Work

Conclusions / Results

- We developed the process algebra prCRL, incorporating both data and probability.
- We defined a linear format for prCRL, the LPPE, providing the starting point for effective symbolic optimisations and easy state space generation.
- We provided a linearisation algorithm to transform prCRL specifications to LPPEs, proved it correct, and implemented it.

Future work

Applying existing optimisation techniques to LPPEs

- Automating the translation from LPPE to PA
- Branching bisimulation preserving reductions (e.g., confluence reduction)

Questions

Questions?