UNIVERSITY OF TWENTE.

Formal Methods & Tools.

A linear process-algebraic format for probabilistic systems with data

Mark Timmer

June 25, 2010

ACSD 2010

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

Probabilistic Model Checking

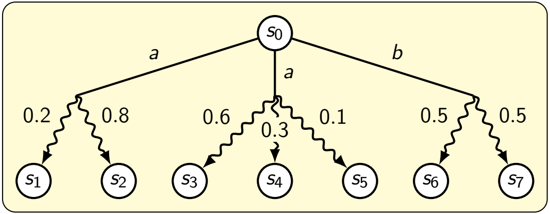
Probabilistic model checking:

- Verifying quantitative properties,
- Using a probabilistic model

Probabilistic Model Checking

Probabilistic model checking:

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

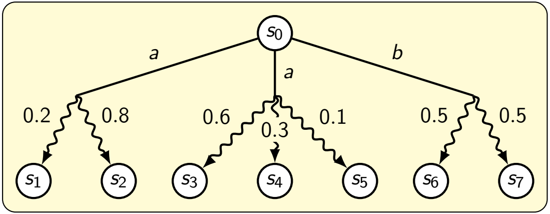


- Non-deterministically choose one of the three transitions
- Probabilistically choose the next state

Probabilistic Model Checking

Probabilistic model checking:

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

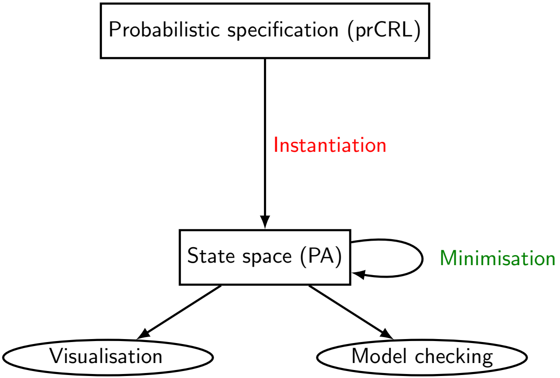


- Non-deterministically choose one of the three transitions
- Probabilistically choose the next state

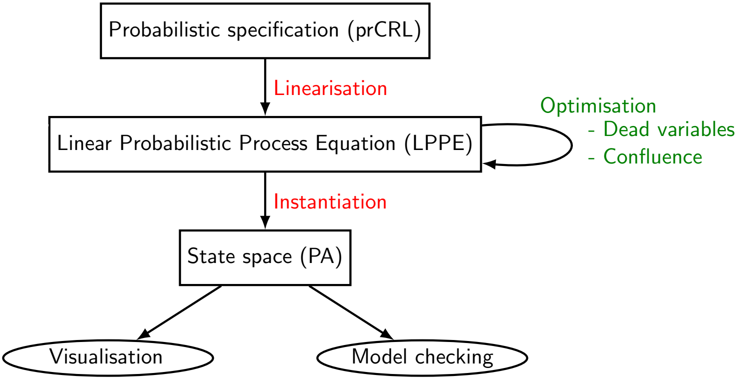
Limitations of previous approaches:

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

Overview of our approach



Overview of our approach



Strong probabilistic bisimulation

Equivalent PAs: strong probabilistic bisimilar PAs

Strong probabilistic bisimulation

Equivalent PAs: strong 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: strong 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

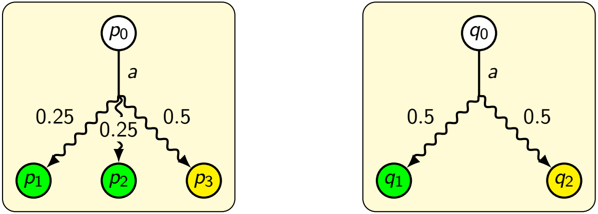
Equivalent PAs: strong probabilistic bisimilar PAs

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

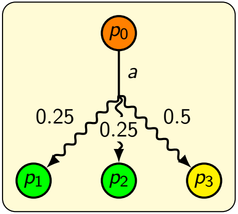
Equivalent PAs: strong probabilistic bisimilar PAs

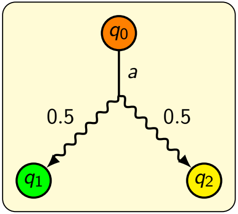
Strong bisimulation

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 $\mu \equiv_R \mu'$





Contents

Contents

A process algebra with data and probability: prCRL

Specification language prCRL:

- Based on μCRL (so data), with additional probabilistic choice
- 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

Specification language prCRL:

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

The grammar of prCRL process terms

Process terms in prCRL are obtained by the following grammar:

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

Process equations and processes

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

An example specification

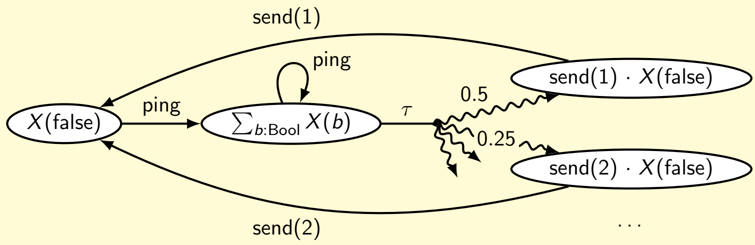
Sending an arbitrary natural number

$$
\begin{aligned}
X(\text{active} : \text{Bool}) = \\
& \text{not}(\text{active}) \Rightarrow \text{ping} \cdot \sum_{b:\text{Bool}} X(b) \\
+ \ \text{active} \quad & \Rightarrow \tau \sum\!\!\!\bullet_{n:\mathbb{N}^{>0}} \frac{1}{2^n} : \left(\text{send}(n) \cdot X(\text{false})\right)
\end{aligned}
$$

An example specification

Sending an arbitrary natural number

$$X(\text{active} : \text{Bool}) =$$
$$\text{not}(\text{active}) \Rightarrow \text{ping} \cdot \sum_{b:\text{Bool}} X(b)$$
$$+\ \text{active} \Rightarrow \tau \sum_{n:\mathbb{N}^{>0}} \frac{1}{2^n} : \Big(\text{send}(n) \cdot X(\text{false})\Big)$$



Compositionality using extended prCRL

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

Compositionality using extended prCRL

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

$$X(n : \{1,2\}) = \text{write}_X(n) \cdot X(n) + \text{choose} \sum_{n' : \{1,2\}}^{\bullet} \frac{1}{2} : X(n')$$

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

Compositionality using extended prCRL

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

$$X(n : \{1,2\}) = \text{write}_X(n) \cdot X(n) + \text{choose} \sum_{n' : \{1,2\}}^{\bullet} \frac{1}{2} : X(n')$$

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

$$Z = (X(1) \mid\mid Y(2))$$

Compositionality using extended prCRL

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

$$X(n : \{1,2\}) = \text{write}_X(n) \cdot X(n) + \text{choose} \sum_{n' : \{1,2\}}^{\bullet} \frac{1}{2} : X(n')$$

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

$$Z = (X(1) \,||\, Y(2))$$

$$\gamma(\text{choose}, \text{choose}') = \text{chooseTogether}$$

Compositionality using extended prCRL

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

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

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

$$Z = \partial_{\{\text{choose},\text{choose}'\}}(X(1) \,||\, Y(2))$$

$$\gamma(\text{choose}, \text{choose}') = \text{chooseTogether}$$

Compositionality using extended prCRL

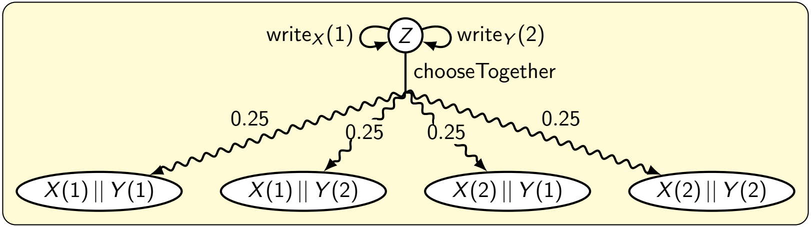
For compositionality we introduce extended prCRL. It extends prCRL by parallel composition, encapsulation, hiding and renaming.

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

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

$$Z = \partial_{\{\text{choose},\text{choose}'\}}(X(1) \mid\mid Y(2))$$

$$\gamma(\text{choose}, \text{choose}') = \text{chooseTogether}$$



Contents

A linear format for prCRL: the LPPE

LPPEs are a subset of prCRL specifications:

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

A linear format for prCRL: the LPPE

LPPEs are a subset of prCRL specifications:

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

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

LPPEs are a subset of prCRL specifications:

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

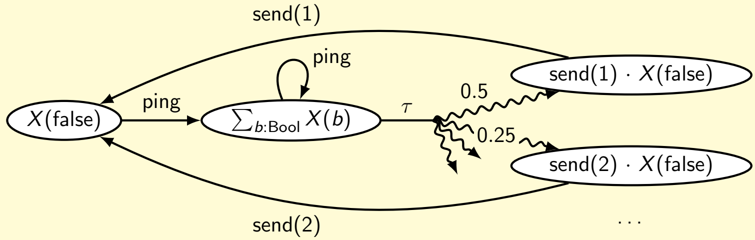
Advantages of using LPPEs instead of prCRL specifications:

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

Theorem

Every specification (without unguarded recursion) can be linearised to an LPPE, preserving strong probabilistic bisimulation.

Linear Probabilistic Process Equations – an example



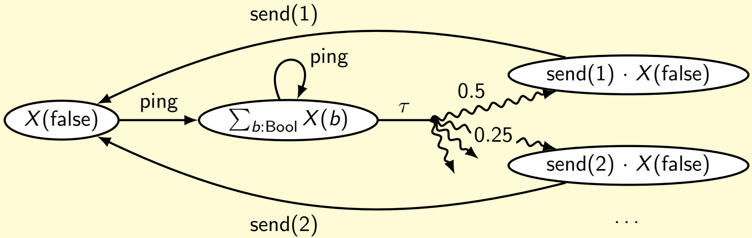
Specification in prCRL

$$X(\text{active} : \text{Bool}) =$$

$$\text{not}(\text{active}) \Rightarrow \text{ping} \cdot \sum_{b:\text{Bool}} X(b)$$

$$+ \text{ active} \Rightarrow \tau \sum\!\!\!\bullet_{n:\mathbb{N}^{>0}} \frac{1}{2^n} : \text{send}(n) \cdot X(\text{false})$$

Linear Probabilistic Process Equations – an example



Specification in prCRL

$$X(\text{active} : \text{Bool}) =$$

$$\text{not}(\text{active}) \Rightarrow \text{ping} \cdot \sum_{b:\text{Bool}} X(b)$$

$$+ \text{ active} \Rightarrow \tau \sum_{n:\mathbb{N}^{>0}} \bullet \; \frac{1}{2^n} : \text{send}(n) \cdot X(\text{false})$$

Specification in LPPE

$$X(pc : \{1..3\}, n : \mathbb{N}^{\geq 0}) =$$

$$+ \; pc = 1 \Rightarrow \text{ping} \cdot X(2, 1)$$

$$+ \; pc = 2 \Rightarrow \text{ping} \cdot X(2, 1)$$

$$+ \; pc = 2 \Rightarrow \tau \sum_{n:\mathbb{N}^{>0}} \bullet \; \frac{1}{2^n} : X(3, n)$$

$$+ \; pc = 3 \Rightarrow \text{send}(n) \cdot X(1, 1)$$

Contents

Linearisation: a simple example without data

Consider the following prCRL specification:

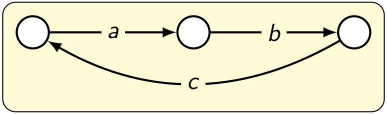
$$X = a \cdot b \cdot c \cdot X$$

Linearisation: a simple example without data

Consider the following prCRL specification:

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

The control flow of X is given by:

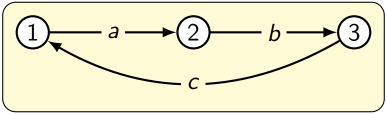


Linearisation: a simple example without data

Consider the following prCRL specification:

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

The control flow of X is given by:

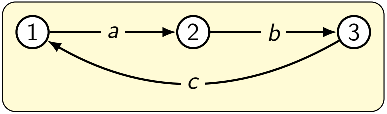


Linearisation: a simple example without data

Consider the following prCRL specification:

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

The control flow of X is given by:



The corresponding LPPE (initialised with pc = 1):

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

Linearisation: a more complicated example with data

Consider the following prCRL specification:

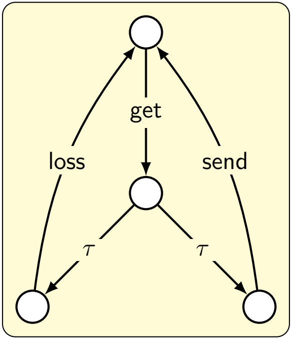
$$X = \sum_{d:D} \text{get}(d) \cdot (\tau \cdot \text{loss} \cdot X + \tau \cdot \text{send}(d) \cdot X)$$

Linearisation: a more complicated example with data

Consider the following prCRL specification:

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

Control flow:

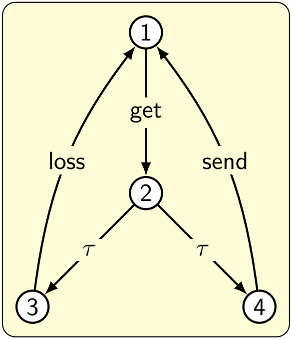


Linearisation: a more complicated example with data

Consider the following prCRL specification:

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

Control flow:

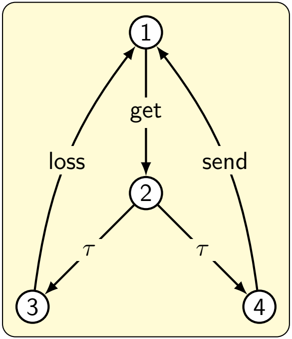


Linearisation: a more complicated example with data

Consider the following prCRL specification:

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

Control flow:



LPPE:

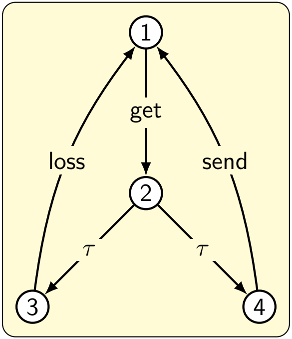
$$\begin{aligned}
Y(pc\colon \{1,2,3,4\}, x\colon D) = \\
\textstyle\sum_{d:D} \quad & pc = 1 \Rightarrow \mathrm{get}(d) \cdot Y(2, d) \\
+ \quad & pc = 2 \Rightarrow \tau \cdot Y(3, x) \\
+ \quad & pc = 2 \Rightarrow \tau \cdot Y(4, x) \\
+ \quad & pc = 3 \Rightarrow \mathrm{loss} \cdot Y(1, x) \\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

Linearisation: a more complicated example with data

Consider the following prCRL specification:

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

Control flow:



LPPE:

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

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

---

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: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

---

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

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

1 $X_1(d:D, e:D, f:D) =$

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

---

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

$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$

$X_3(d:D, e:D, f:D) = c(f) \cdot X_1(5,e,f)$

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

$X_2(d:D, e:D, f:D) = c(e) \cdot X_3(d,e,f) + c(e+f) \cdot X_1(5,e,f)$

$X_3(d:D, e:D, f:D) = c(f) \cdot X_1(5,e,f)$

Linearisation: a more algorithmic approach

Consider the following prCRL specification:

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

---

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

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

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.

Contents

Case study: a leader election protocol

- Implementation in Haskell:
  - Linearisation: from prCRL to LPPE
  - Parallel composition of LPPEs, hiding, renaming, encapsulation
  - Generation of the state space of an LPPE
  - Automatic constant elimination and summand simplification
- Manual dead variable reduction

Case study: a leader election protocol

- Implementation in Haskell:
  - Linearisation: from prCRL to LPPE
  - Parallel composition of LPPEs, hiding, renaming, encapsulation
  - Generation of the state space of an LPPE
  - Automatic constant elimination and summand simplification
- Manual dead variable reduction

Case study

Leader election protocol à la Itai-Rodeh

- Two processes throw a die
  - *The process with the highest number will be leader*
  - *In case of a tie: throw again*

Case study: a leader election protocol

- Implementation in Haskell:
  - Linearisation: from prCRL to LPPE
  - Parallel composition of LPPEs, hiding, renaming, encapsulation
  - Generation of the state space of an LPPE
  - Automatic constant elimination and summand simplification
- Manual dead variable reduction

Case study

Leader election protocol à la Itai-Rodeh

- Two processes throw a die
  - *The process with the highest number will be leader*
  - *In case of a tie: throw again*
- More precisely:
  - *Passive thread: receive value of opponent*
  - *Active thread: roll, send, compare (or block)*

A prCRL model of the leader election protocol

$$
\begin{aligned}
& P(id : \{one, two\}, val : Die, set : Bool) = \\
& \qquad set = false \Rightarrow \sum_{d:Die} communicate(id, other(id), d) \cdot P(id, d, true) \\
& \quad + set = true \Rightarrow checkValue(val) \cdot P(id, val, false) \\
& A(id : \{one, two\}) = \\
& \quad roll(id) \sum_{d:Die} \bullet \tfrac{1}{6} : \overline{communicate}(other(id), id, d) \cdot \sum_{e:Die} \overline{checkValue}(e) \cdot \\
& \qquad \big(\ (d = e \Rightarrow A(id)) \\
& \qquad + (d > e \Rightarrow leader(id) \cdot A(id)) \\
& \qquad + (e > d \Rightarrow follower(id) \cdot A(id))\big) \\
& C(id : \{one, two\}) = P(id, 1, false) \,||\, A(id) \\
& S = C(one) \,||\, C(two)
\end{aligned}
$$

Reductions on the leader election protocol model

In order to obtain reductions first linearise:

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

Reductions on the leader election protocol model

In order to obtain reductions first linearise:

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

Before reductions:

- 18 parameters
- 14 summands
- 3763 states
- 6158 transitions

Reductions on the leader election protocol model

In order to obtain reductions first linearise:

$$
\begin{aligned}
&\mathit{pc21} = 3 \wedge \quad \mathit{set11} \quad \Rightarrow \\
&\mathit{checkValue}(\mathit{val11}) \sum_{(k1,k2):\{*\}\times\{*\}} 1.0 : \\
&Z(\mathit{val11}, \mathit{false}, 4, \mathit{d21}, \mathit{val11}, \\
&\quad \mathit{val12}, \mathit{set12}, \mathit{pc22}, \mathit{d22}, \mathit{e22})
\end{aligned}
$$

Before reductions:

- 18 parameters
- 14 summands
- 3763 states
- 6158 transitions

After reductions:

- 10 parameters
- 12 summands

Reductions on the leader election protocol model

In order to obtain reductions first linearise:

$$pc21 = 3 \wedge \quad set11 \quad \Rightarrow$$
$$checkValue(val11) \sum_{(k1,k2):\{*\}\times\{*\}} 1.0 :$$
$$Z(\quad 1 \quad, false, \quad 4, \quad d21, val11,$$
$$val12, set12, \quad pc22, \quad d22, e22)$$

Before reductions:

- 18 parameters
- 14 summands
- 3763 states
- 6158 transitions

After reductions:

- 10 parameters
- 12 summands
- 1693 states (-55%)
- 2438 transitions (-60%)

Contents

Conclusions and Future Work

Conclusions / Results

- We developed the process algebra prCRL, incorporating both data and probability.
- We defined a normal form for prCRL, the LPPE; starting point for symbolic optimisations and easy state space generation.
- We provided a linearisation algorithm to transform prCRL specifications to LPPEs, proved it correct, implemented it, and used it to show significant reductions on a case study.

Conclusions and Future Work

Conclusions / Results

- We developed the process algebra prCRL, incorporating both data and probability.
- We defined a normal form for prCRL, the LPPE; starting point for symbolic optimisations and easy state space generation.
- We provided a linearisation algorithm to transform prCRL specifications to LPPEs, proved it correct, implemented it, and used it to show significant reductions on a case study.

Future work

- Develop additional reduction techniques, for instance confluence reduction (in progress).
- Generalise proof techniques such as cones and foci to the probabilistic case.

Questions

Questions?