UNIVERSITY OF TWENTE.

Formal Methods & Tools.

State Space Reduction of Linear Processes using Control Flow Reconstruction

Mark Timmer

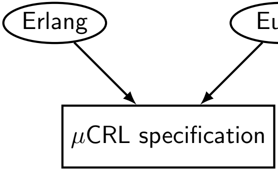
October 14, 2009

ATVA 2009

Joint work with Jaco van de Pol

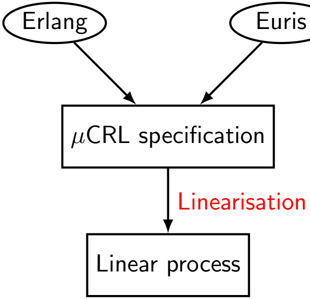
Contents

The μCRL toolset



System specification consisting of parallel processes

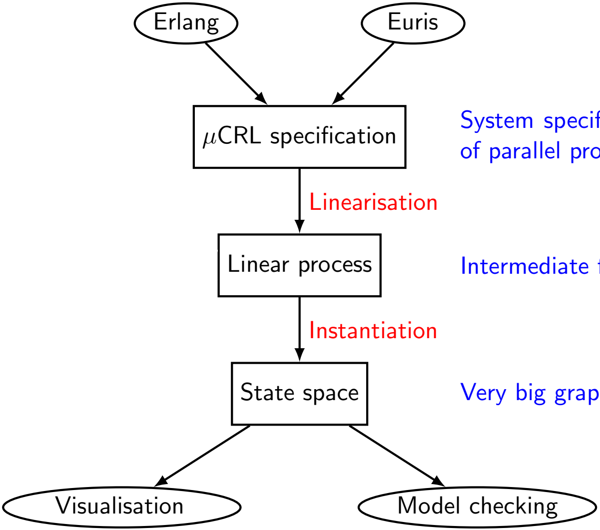
The μCRL toolset



System specification consisting of parallel processes

Intermediate format

The μCRL toolset

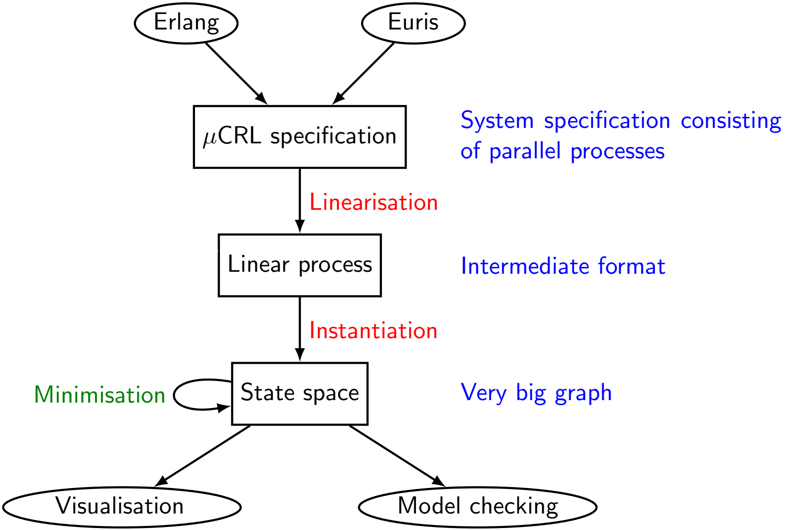


System specification consisting of parallel processes

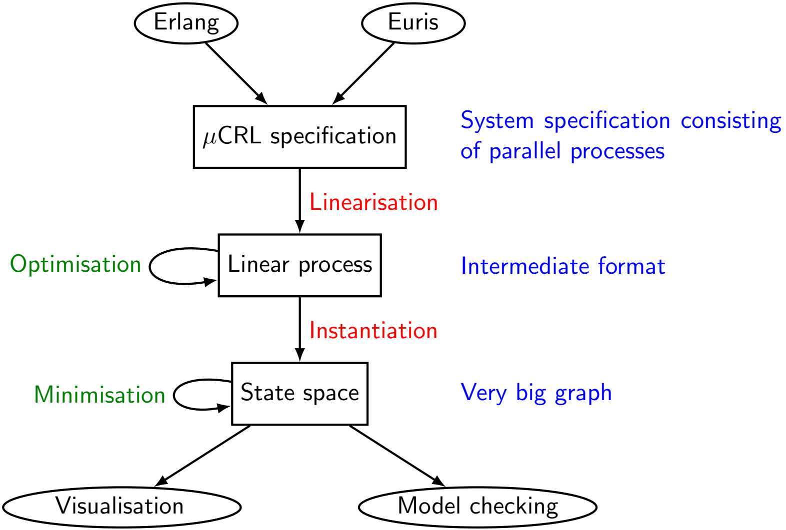
Intermediate format

Very big graph

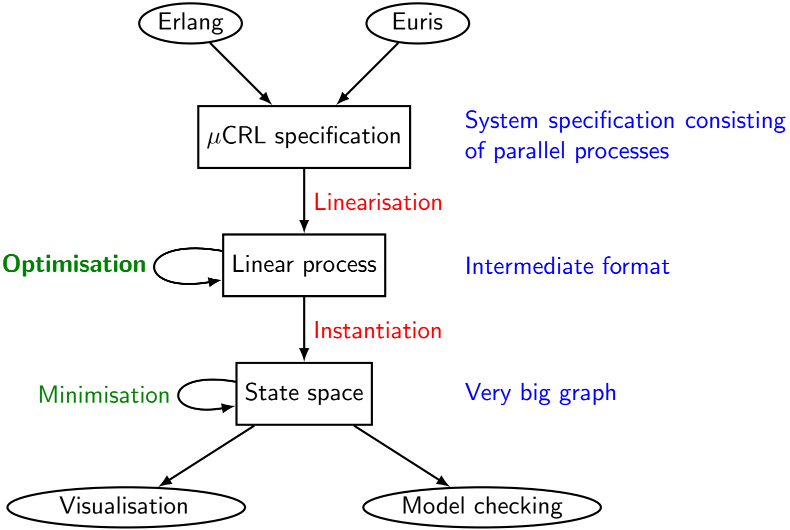
The μCRL toolset



The μCRL toolset



The μCRL toolset



The linear process equation

The basic structure of an LPE

$$\begin{aligned} X(\boldsymbol{d} : \boldsymbol{D}) = & \sum_{\boldsymbol{e_1} : \boldsymbol{E_1}} c_1(\boldsymbol{d}, \boldsymbol{e_1}) \Rightarrow a_1(\boldsymbol{d}, \boldsymbol{e_1}) \cdot X(g_1(\boldsymbol{d}, \boldsymbol{e_1})) \\ + & \ \dots \\ + & \sum_{\boldsymbol{e_n} : \boldsymbol{E_n}} c_n(\boldsymbol{d}, \boldsymbol{e_n}) \Rightarrow a_n(\boldsymbol{d}, \boldsymbol{e_n}) \cdot X(g_n(\boldsymbol{d}, \boldsymbol{e_n})) \end{aligned}$$

- d: a vector of global state variables
- e_i: a vector of local variables for summand i
- c_i: the enabling condition for summand i
- a_i: the (parameterised) action for summand i (possibly τ)
- g_i: the next-state function for summand i

The linear process equation

The basic structure of an LPE

$$
\begin{aligned}
X(\boldsymbol{d} : \boldsymbol{D}) = & \sum_{\boldsymbol{e_1} : \boldsymbol{E_1}} c_1(\boldsymbol{d}, \boldsymbol{e_1}) \Rightarrow a_1(\boldsymbol{d}, \boldsymbol{e_1}) \cdot X(g_1(\boldsymbol{d}, \boldsymbol{e_1})) \\
& + \quad \ldots \\
& + \sum_{\boldsymbol{e_n} : \boldsymbol{E_n}} c_n(\boldsymbol{d}, \boldsymbol{e_n}) \Rightarrow a_n(\boldsymbol{d}, \boldsymbol{e_n}) \cdot X(g_n(\boldsymbol{d}, \boldsymbol{e_n}))
\end{aligned}
$$

- d: a vector of global state variables
- e_i: a vector of local variables for summand i
- c_i: the enabling condition for summand i
- a_i: the (parameterised) action for summand i (possibly τ)
- g_i: the next-state function for summand i

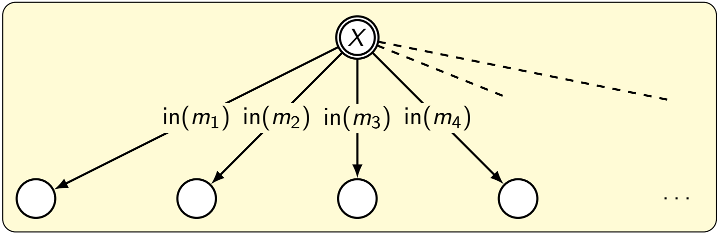
$$
\boldsymbol{d} \xrightarrow{a(\boldsymbol{p})} \boldsymbol{d}' \Leftrightarrow \exists i \,.\, \exists \boldsymbol{e_i} \,.\, c_i(\boldsymbol{d}, \boldsymbol{e_i}) = \texttt{true} \land a_i(\boldsymbol{d}, \boldsymbol{e_i}) = a(\boldsymbol{p}) \land g_i(\boldsymbol{d}, \boldsymbol{e_i}) = \boldsymbol{d}'
$$

An example

$$X = \sum_{m:\{m_1,\ldots,m_{10}\}} \mathsf{in}(m) \cdot (\tau \cdot \mathsf{loss} \cdot X + \tau \cdot \mathsf{out}(m) \cdot X)$$

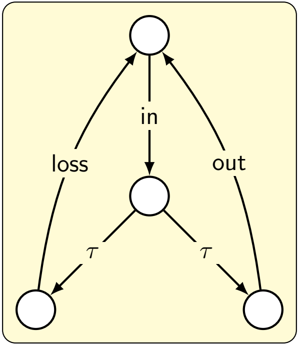
An example

$$X = \sum_{m:\{m_1,\ldots,m_{10}\}} \mathsf{in}(m) \cdot (\tau \cdot \mathsf{loss} \cdot X + \tau \cdot \mathsf{out}(m) \cdot X)$$



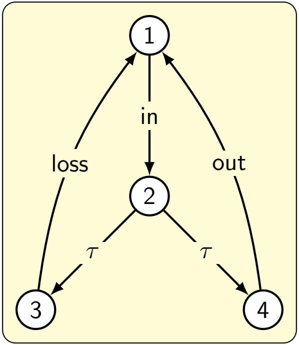
An example

$$X = \sum_{m:\{m_1,\ldots,m_{10}\}} \mathsf{in}(m) \cdot (\tau \cdot \mathsf{loss} \cdot X + \tau \cdot \mathsf{out}(m) \cdot X)$$



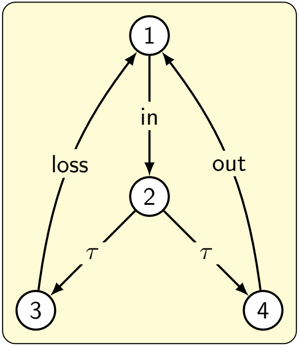
An example

$$X = \sum_{m:\{m_1,\ldots,m_{10}\}} \mathsf{in}(m) \cdot (\tau \cdot \mathsf{loss} \cdot X + \tau \cdot \mathsf{out}(m) \cdot X)$$



An example

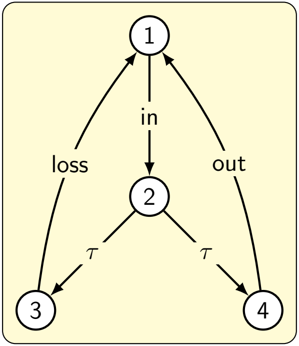
$$X = \sum_{m:\{m_1,\ldots,m_{10}\}} \mathsf{in}(m) \cdot (\tau \cdot \mathsf{loss} \cdot X + \tau \cdot \mathsf{out}(m) \cdot X)$$



$$\begin{aligned} X(pc\colon \{1,2,3,4\}, x : \{m_1, \ldots, m_{10}\}) = \\ \textstyle\sum_{m:\{m_1,\ldots,m_{10}\}} \quad & pc = 1 \Rightarrow \mathsf{in}(m) \cdot X(2,m) \\ + \quad & pc = 2 \Rightarrow \tau \cdot X(3,x) \\ + \quad & pc = 2 \Rightarrow \tau \cdot X(4,x) \\ + \quad & pc = 3 \Rightarrow \mathsf{loss} \cdot X(1,x) \\ + \quad & pc = 4 \Rightarrow \mathsf{out}(x) \cdot X(1,x) \end{aligned}$$

An example

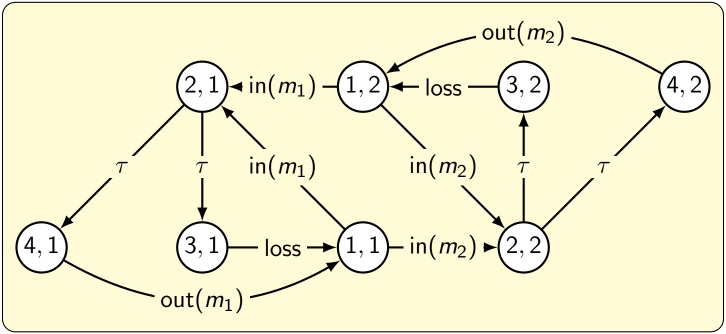
$$X = \sum_{m:\{m_1,\ldots,m_{10}\}} \mathsf{in}(m) \cdot (\tau \cdot \mathsf{loss} \cdot X + \tau \cdot \mathsf{out}(m) \cdot X)$$



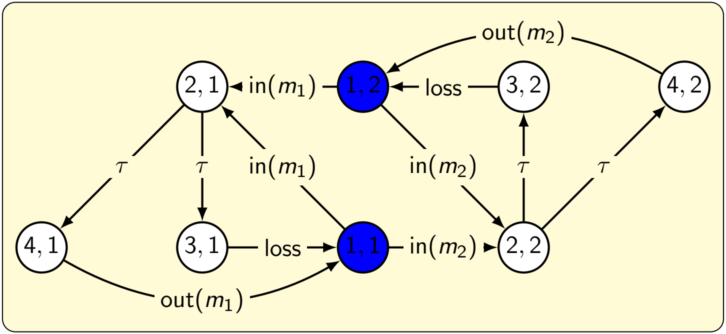
$$
\begin{aligned}
X(pc\colon \{1,2,3,4\}, x : \{m_1, \ldots, m_{10}\}) = \\
\textstyle\sum_{m:\{m_1,\ldots,m_{10}\}} \quad & pc = 1 \Rightarrow \mathsf{in}(m) \cdot X(2, m) \\
+ \quad & pc = 2 \Rightarrow \tau \cdot X(3, x) \\
+ \quad & pc = 2 \Rightarrow \tau \cdot X(4, x) \\
+ \quad & pc = 3 \Rightarrow \mathsf{loss} \cdot X(1, x) \\
+ \quad & pc = 4 \Rightarrow \mathsf{out}(x) \cdot X(1, x)
\end{aligned}
$$

Initial process: $X(1, m_1)$.

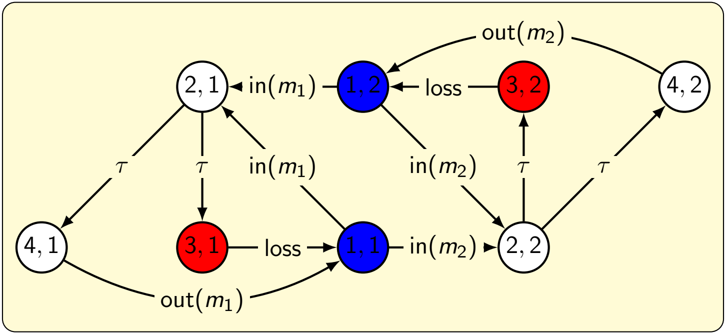
An example


out(m_2)
2, 1
in(m_1)
1, 2
loss
3, 2
4, 2
τ
τ
in(m_1)
in(m_2)
τ
τ
4, 1
3, 1
loss
1, 1
in(m_2)
2, 2
out(m_1)

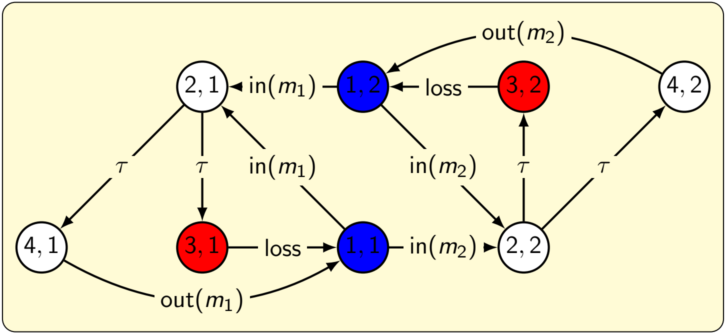
An example

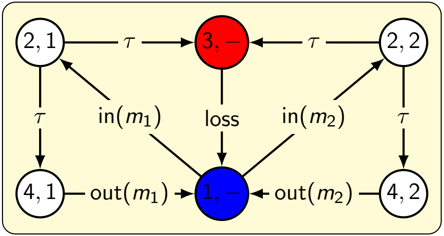

out(m_2)
2, 1
in(m_1)
1, 2
loss
3, 2
4, 2
τ
τ
in(m_1)
in(m_2)
τ
τ
4, 1
3, 1
loss
1, 1
in(m_2)
2, 2
out(m_1)

An example

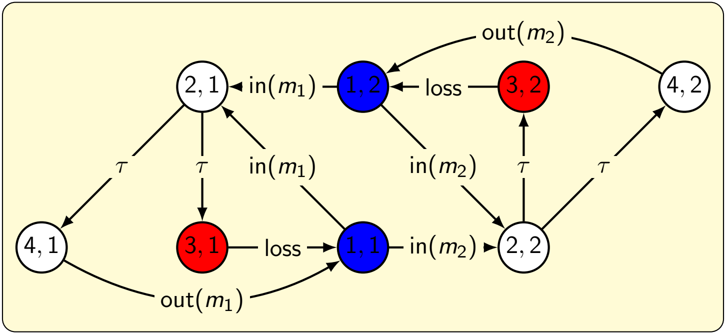

out(m_2)
2, 1
in(m_1)
1, 2
loss
3, 2
4, 2
τ
τ
in(m_1)
in(m_2)
τ
τ
4, 1
3, 1
loss
1, 1
in(m_2)
2, 2
out(m_1)

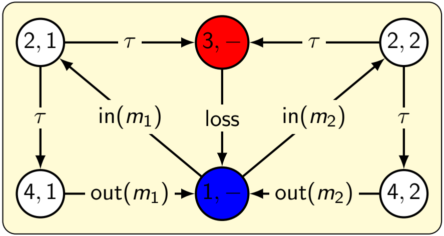
An example


out(m_2)
2, 1
in(m_1)
1, 2
loss
3, 2
4, 2
τ
τ
in(m_1)
in(m_2)
τ
τ
4, 1
3, 1
loss
1, 1
in(m_2)
2, 2
out(m_1)


2, 1
τ
3, –
τ
2, 2
τ
in(m_1)
loss
in(m_2)
τ
4, 1
out(m_1)
1, –
out(m_2)
4, 2

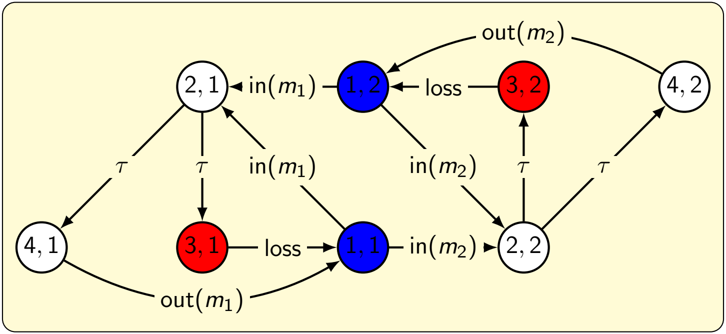
An example

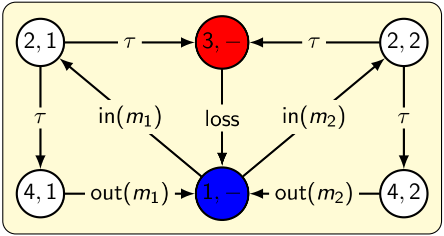




$$
\begin{aligned}
X(pc\colon \{1,2,3,4\}, x\colon \{m_1,\ldots,m_{10}\}) = \\
\textstyle\sum_m \; pc = 1 &\Rightarrow \text{in}(m) \cdot X(2, m) \\
+ \quad pc = 2 &\Rightarrow \tau \cdot X(3, x\ ) \\
+ \quad pc = 2 &\Rightarrow \tau \cdot X(4, x) \\
+ \quad pc = 3 &\Rightarrow \text{loss} \cdot X(1, x\ ) \\
+ \quad pc = 4 &\Rightarrow \text{out}(x) \cdot X(1, x\ )
\end{aligned}
$$

An example





$$
\begin{aligned}
X(pc\colon \{1,2,3,4\}, x\colon \{m_1,\ldots,m_{10}\}) = \\
\textstyle\sum_m \quad & pc = 1 \quad \Rightarrow \mathrm{in}(m)\cdot X(2,m) \\
+ \quad & pc = 2 \quad \Rightarrow \tau \cdot X(3, m_1) \\
+ \quad & pc = 2 \quad \Rightarrow \tau \cdot X(4, x) \\
+ \quad & pc = 3 \quad \Rightarrow \mathrm{loss}\cdot X(1, m_1) \\
+ \quad & pc = 4 \quad \Rightarrow \mathrm{out}(x)\cdot X(1, m_1)
\end{aligned}
$$

Control Flow Reconstruction

Goal: reductions on LPE format

Control Flow Reconstruction

Goal: reductions on LPE format

Problem: control flow is hidden in state parameters. Moreover, there are several control flows due to parallelism.

Control Flow Reconstruction

Goal: reductions on LPE format

Problem: control flow is hidden in state parameters. Moreover, there are several control flows due to parallelism.

Solution:

1. Detect control flow parameters
2. Identify clusters of summands
3. Assign data parameters to clusters

Control Flow Reconstruction

Goal: reductions on LPE format

Problem: control flow is hidden in state parameters. Moreover, there are several control flows due to parallelism.

Solution:

1. Detect control flow parameters
2. Identify clusters of summands
3. Assign data parameters to clusters

4. Deduce when data parameters are (globally) relevant
5. Transform the LPE

Control flow parameters

Observation: program counters (control flow parameters) are special.

Control flow parameters

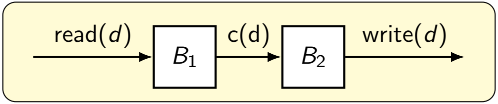
Observation: program counters (control flow parameters) are special.

$B_1 = \sum_{d\colon D} \mathit{read}(d) \cdot w(d) \cdot B_1$ $B_2 = \sum_{d\colon D} r(d) \cdot \mathit{write}(d) \cdot B_2$

Control flow parameters

Observation: program counters (control flow parameters) are special.

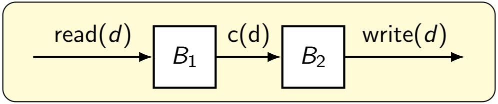
$B_1 = \sum_{d:D} read(d) \cdot w(d) \cdot B_1$ $B_2 = \sum_{d:D} r(d) \cdot write(d) \cdot B_2$



Control flow parameters

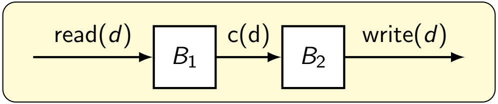
Observation: program counters (control flow parameters) are special.

$$B_1 = \sum_{d:\,D} read(d) \cdot w(d) \cdot B_1 \qquad B_2 = \sum_{d:\,D} r(d) \cdot write(d) \cdot B_2$$



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

Control flow parameters

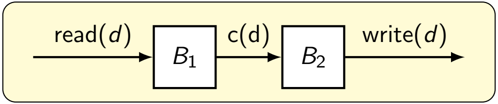


$$
\begin{array}{lll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & \\
\quad \sum_{d\colon D} & a = 1 & \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) \\
+ & b = 2 & \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) \\
+ & a = 2 \wedge b = 1 & \Rightarrow c(x) \cdot X(1, 2, x, x)
\end{array}
$$

In every summand, each control flow parameter

- is either left unchanged, or
- has a clear transition from a source value to a destination value

Control flow parameters

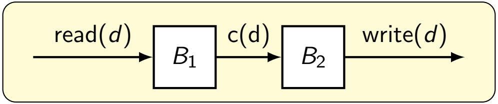


$$
\begin{aligned}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & \\
\textstyle\sum_{d\colon D} \quad a = 1 & \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) \\
+ \quad b = 2 & \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) \\
+ \quad a = 2 \wedge b = 1 & \Rightarrow c(x) \cdot X(1,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

In every summand, each control flow parameter

- is either left unchanged, or
- has a clear transition from a source value to a destination value

Control flow parameters

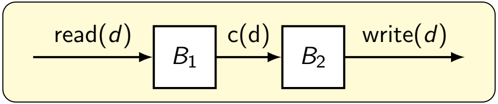


$$
\begin{aligned}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = \\
\textstyle\sum_{d\colon D} \quad & a = 1 && \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) \\
+ \quad & b = 2 && \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) \\
+ \quad & a = 2 \wedge b = 1 && \Rightarrow c(x) \cdot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

In every summand, each control flow parameter

- is either left unchanged, or
- has a clear transition from a source value to a destination value

Control flow parameters

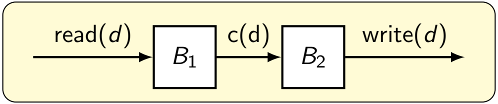


$$
\begin{aligned}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = \\
\textstyle\sum_{d\colon D} \quad & a = 1 && \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) \\
+ \quad & b = 2 && \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) \\
+ \quad & a = 2 \wedge b = 1 && \Rightarrow c(x) \cdot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

In every summand, each control flow parameter

- is either left unchanged, or
- has a clear transition from a source value to a destination value

Control flow parameters

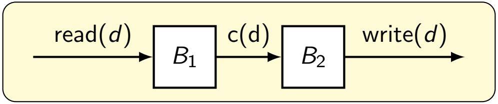


$$
\begin{aligned}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = &\\
\textstyle\sum_{d\colon D} \quad & a = 1 \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y)\\
+ \quad & b = 2 \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y)\\
+ \quad & a = 2 \wedge b = 1 \Rightarrow c(x) \cdot X(1, 2,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

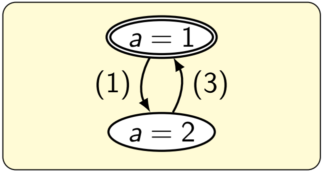
In every summand, each control flow parameter

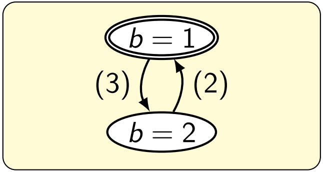
- is either left unchanged, or
- has a clear transition from a source value to a destination value

Control Flow Graph



$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\sum_{d\colon D} & a = 1 & \Rightarrow \mathrm{read}(d)\cdot X(2,b,d,y) & (1)\\
+ & b = 2 & \Rightarrow \mathrm{write}(y)\cdot X(a,1,x,y) & (2)\\
+ & a = 2 \wedge b = 1 & \Rightarrow c(x)\cdot X(1,2,x,x) & (3)
\end{array}
$$





The *belongs to* relation

Belongs to

A data parameter k belongs to a CFP j if the cluster of j contains all summands that

- either change k, or
- make use of k (in an action, condition or next-state)

The *belongs to* relation

Belongs to

A data parameter k belongs to a CFP j if the cluster of j contains all summands that

- either change k, or
- make use of k (in an action, condition or next-state)

$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\quad \sum_{d\colon D} & a = 1 & \Rightarrow \text{read}(d) \cdot X(2, b, \mathbf{d}, y) & (1) \\
+ & b = 2 & \Rightarrow \text{write}(\mathbf{y}) \cdot X(a, 1, x, y) & (2) \\
+ & a = 2 \wedge b = 1 & \Rightarrow c(\mathbf{x}) \cdot X(1, 2, x, \mathbf{x}) & (3)
\end{array}
$$

The *belongs to* relation

Belongs to

A data parameter k belongs to a CFP j if the cluster of j contains all summands that

- either change k, or
- make use of k (in an action, condition or next-state)

$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\quad \sum_{d\,:\,D} & a = 1 & \Rightarrow \text{read}(d) \cdot X(2, b, \mathbf{d}, y) & (1) \\
+ & b = 2 & \Rightarrow \text{write}(\mathbf{y}) \cdot X(a, 1, x, y) & (2) \\
+ & a = 2 \wedge b = 1 & \Rightarrow c(\mathbf{x}) \cdot X(1, 2, x, \mathbf{x}) & (3)
\end{array}
$$

So, x belongs to a and y belongs to b. Thus, relevance of x can be decided by looking only at the control flow of a.

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state s

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \Longleftrightarrow$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\quad \sum_{d\colon D} & a = 1 & \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) & (1) \\
+ & b = 2 & \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) & (2) \\
+ & a = 2 \wedge b = 1 & \Rightarrow \tau \cdot X(1, 2, x, x) & (3)
\end{array}
$$

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

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

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

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

So: $R(y, b, 2)$

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\quad \sum_{d\colon D} & a = 1 & \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) & (1) \\
\quad + & b = 2 & \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) & (2) \\
\quad + & a = 2 \wedge b = 1 & \Rightarrow \tau \cdot X(1, 2, x, x) & (3)
\end{array}
$$

So: $R(y, b, 2)$

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

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

So: $R(y, b, 2)$ and $R(x, a, 2)$

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\quad \sum_{d\colon D} & a = 1 & \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) & (1) \\
\quad + & b = 2 & \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) & (2) \\
\quad + & a = 2 \wedge b = 1 & \Rightarrow \tau \cdot X(1, 2, x, x) & (3)
\end{array}
$$

So: $R(y, b, 2)$ and $R(x, a, 2)$ (and therefore $\neg R(y, b, 1)$ and $\neg R(x, a, 1)$)

Relevance

$R(k, j, s)$: parameter k is relevant when CFP j is in state $s \iff$

There is a summand that can be taken when $d_j = s$, that either

- directly uses k for its condition or action, or
- indirectly uses k to determine the value of a parameter that is relevant after taking the summand

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

So: $R(y, b, 2)$ and $R(x, a, 2)$ (and therefore $\neg R(y, b, 1)$ and $\neg R(x, a, 1)$)

If $\neg R(k, j, s)$, then k is irrelevant when j is in state s

Transformation

Based on data flow analysis, irrelevant parameters can be changed.

Transformation

Based on data flow analysis, irrelevant parameters can be changed.

▷ To never increase the state space, replace by their initial value.

Transformation

Based on data flow analysis, irrelevant parameters can be changed.
▷ To never increase the state space, replace by their initial value.

$$
\begin{array}{lll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & \\
\quad \sum_{d\colon D} \; a = 1 & \Rightarrow \mathrm{read}(d) \cdot X(2, b, d, y) & (1) \\
+ \quad b = 2 & \Rightarrow \mathrm{write}(y) \cdot X(a, 1, x, y) & (2) \\
+ \quad a = 2 \wedge b = 1 & \Rightarrow c(x) \cdot X(1, 2, x, x) & (3)
\end{array}
$$

We saw: $\neg R(x, a, 1)$ and $\neg R(y, b, 1)$.

Transformation

Based on data flow analysis, irrelevant parameters can be changed.
▷ To never increase the state space, replace by their initial value.

$$
\begin{array}{llll}
X(a\colon \{1,2\}, b\colon \{1,2\}, x\colon D, y\colon D) = & & & \\
\quad \sum_{d\colon D} & a = 1 & \Rightarrow \mathrm{read}(d)\cdot X(2,b,d,y) & (1) \\
+ & b = 2 & \Rightarrow \mathrm{write}(y)\cdot X(a,1,x,y) & (2) \\
+ & a = 2 \wedge b = 1 & \Rightarrow c(x)\cdot X(1,2,x,x) & (3)
\end{array}
$$

We saw: $\neg R(x, a, 1)$ and $\neg R(y, b, 1)$.

Transformation

Based on data flow analysis, irrelevant parameters can be changed.
▷ To never increase the state space, replace by their initial value.

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

We saw: $\neg R(x, a, 1)$ and $\neg R(y, b, 1)$.

Transformation

Based on data flow analysis, irrelevant parameters can be changed.
▷ To never increase the state space, replace by their initial value.

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

We saw: $\neg R(x, a, 1)$ and $\neg R(y, b, 1)$.
So, assuming initially $x = y = k$

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

Transformation

Based on data flow analysis, irrelevant parameters can be changed.
▷ To never increase the state space, replace by their initial value.

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

We saw: $\neg R(x,a,1)$ and $\neg R(y,b,1)$.
So, assuming initially $x = y = k$

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

For $|D| = 5$: state space reduction from 60 to 36 states.

Correctness and effectiveness

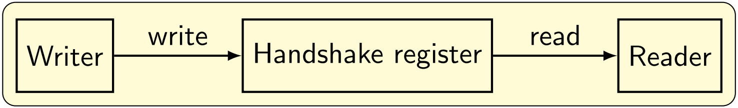
Theorem: correctness

The transformed LPE is strongly bisimilar to the original

Theorem: effectiveness

The number of reachable states of the transformed LPE is at most as large as the number of reachable states in the original

Case study: a handshake register



- Recentness

  Any value read was at some point during reading the last value written

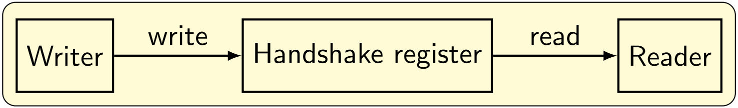
- Sequentiality

  The values of sequential reads occur in the same order as they were written

- Waitfree

  Completion of reads/writes in a bounded number of steps

Case study: a handshake register



- Recentness

  Any value read was at some point during reading the last value written
- Sequentiality

  The values of sequential reads occur in the same order as they were written
- Waitfree

  Completion of reads/writes in a bounded number of steps

Building blocks:

- 4x safe register (random read during writing)
- 4x atomic boolean register

Verifying the implementation

- Model the handshake register specification as a μCRL process
- Model the implementation as a μCRL process
- Generate their state spaces
- Minimise with respect to some equivalence
- Check for graph equivalence

Verifying the implementation

- Model the handshake register specification as a μCRL process
- Model the implementation as a μCRL process
- Generate their state spaces
- Minimise with respect to some equivalence
- Check for graph equivalence

Problem: state space explosion

Verifying the implementation

- Model the handshake register specification as a μCRL process
- Model the implementation as a μCRL process
- Generate their state spaces
- Minimise with respect to some equivalence
- Check for graph equivalence

Problem: state space explosion

Solution: Apply `stategraph`! (and compare to `parelm`)

Applying stategraph

	constelm | parelm | constelm		
	states	time (expl.)	time (symb.)
$\|D\| = 2$	540,736	0:23.0	0:04.5
$\|D\| = 3$	13,834,800	10:10.3	0:06.7
$\|D\| = 4$	142,081,536	–	0:09.0
$\|D\| = 5$	883,738,000	–	0:11.9
$\|D\| = 6$	3,991,840,704	–	0:15.4

	constelm | stategraph | constelm		
	states	time (expl.)	time (symb.)
$\|D\| = 2$	45,504	0:02.4	0:01.3
$\|D\| = 3$	290,736	0:12.7	0:01.4
$\|D\| = 4$	1,107,456	0:48.9	0:01.6
$\|D\| = 5$	3,162,000	2:20.3	0:01.8
$\|D\| = 6$	7,504,704	5:26.1	0:01.9

Other case studies

Other specifications `stategraph` was applied to:

- An Automatic In-flight Data Acquisition unit for a helicopter
- A cache coherence protocol for a distributed JVM
- The sliding window protocol
- An automatic translation from Erlang to μCRL of a distributed resource locker in Ericsson's AXD 301 switch

Other case studies

Other specifications `stategraph` was applied to:

- An Automatic In-flight Data Acquisition unit for a helicopter
- A cache coherence protocol for a distributed JVM
- The sliding window protocol
- An automatic translation from Erlang to μCRL of a distributed resource locker in Ericsson's AXD 301 switch

Results:

- Reductions in the number of states (up to 20 percent)
- Reductions in the number of parameters (up to 75 percent)
- Reductions in the number of summands (up to 25 percent)

Conclusions and Future Work

Conclusions:

- Novel method for reconstructing control flow
  - Even control flow hiding in state parameters is found
- Data flow analysis based on this control flow
  - Resetting variables that are no longer relevant (globally!)
  - Decreases in states, parameters and summands
  - Reductions obtained before generating the entire state space
- Precise proofs of correctness *and* decrease of state space
- Case studies show that impressive results are indeed obtained

Conclusions and Future Work

Conclusions:

- Novel method for reconstructing control flow
  - Even control flow hiding in state parameters is found
- Data flow analysis based on this control flow
  - Resetting variables that are no longer relevant (globally!)
  - Decreases in states, parameters and summands
  - Reductions obtained before generating the entire state space
- Precise proofs of correctness *and* decrease of state space
- Case studies show that impressive results are indeed obtained

Future work:

- Investigate additional applications for the reconstructed control flow
  - Invariant generation
  - Visualisation (already implemented)
  - Improve confluence checking
- Use more precise abstractions based on control flow
- Apply these techniques to a probabilistic linear format

Questions

?