Local Model-Checking of an Alternation-Free Value-Based Modal Mu-Calculus

Radu Mateescu

INRIA Rhône-Alpes / VASY	*CWI / SEN2*
Montbonnot Saint Martin	*Amsterdam*
France	*The Netherlands*

2nd VMCAI — September 19, 1998
Pisa, Italy

Introduction

Motivation:

verification of data-based temporal properties over finite-state systems

"*after a message* m *has been sent, the same message* m *will be eventually received*"

Approach:

- value-based extension of the modal μ-calculus
- local model-checking algorithm

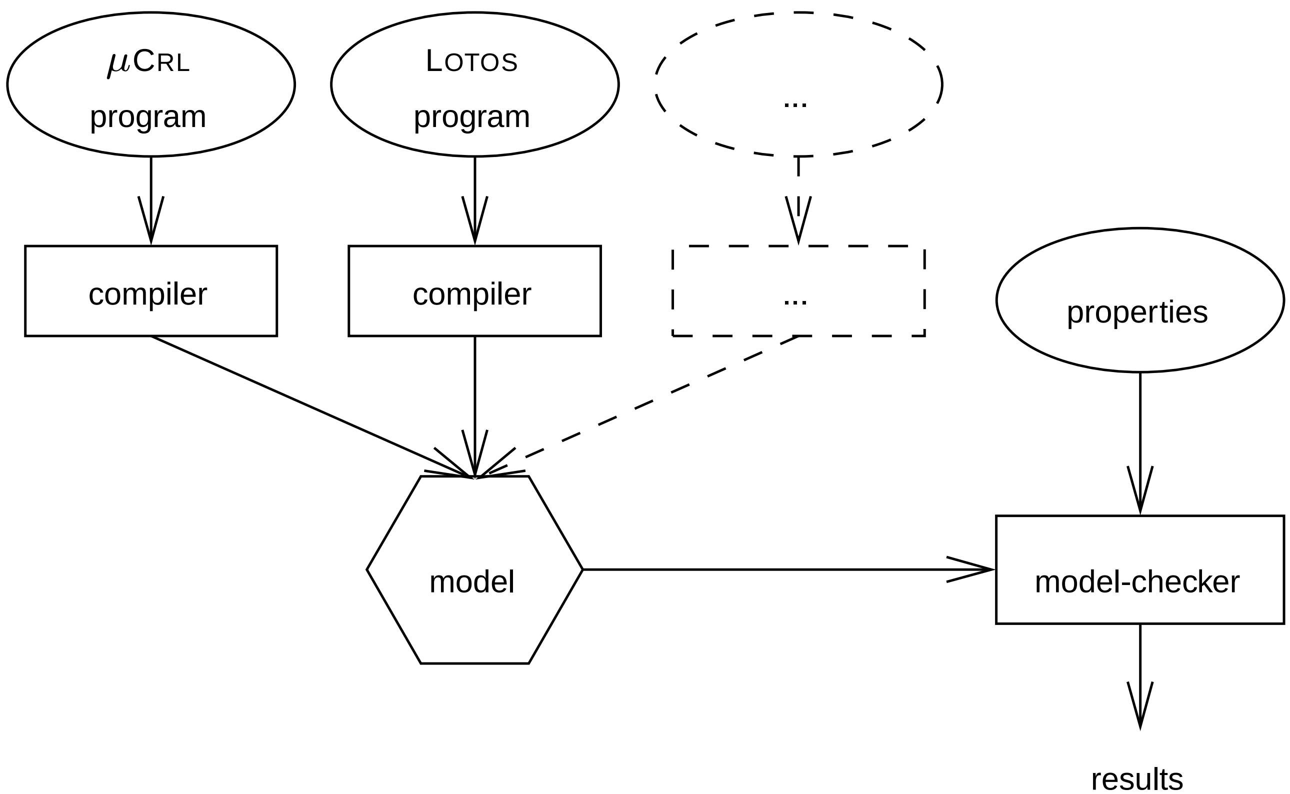
Related work:

value-based:
- modal logic [Hennessy-Liu-93]
- temporal logic [Groote-vanVlijmen-94]
- μ-calculus [Rathke-Hennessy-96]

Outline

- Background
- Value-based μ-calculus
- Applications
- Local model-checking
- Conclusion

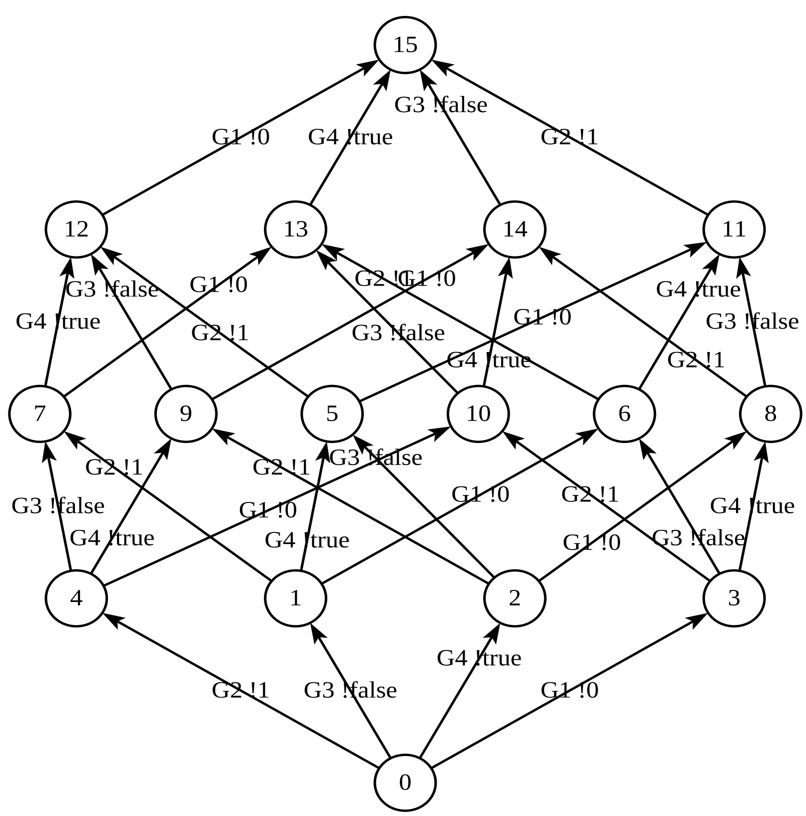
Verification by model-checking



Model

LTS (*Labelled Transition System*) $M = \langle S, A, T, s_0 \rangle$:

- S: set of *states*
- A: set of *actions* ($c\, v_1 \ldots v_n \in A$)
- $T \subseteq S \times A \times S$: *transition relation* ($s_1 \xrightarrow{a} s_2 \in T$)
- $s_0 \in S$: *initial state*



Syntax of the logic

Expressions:

$$
\begin{array}{lcl}
e & ::= & x \\
 & | & f(\vec{e})
\end{array}
$$

Action formulas:

$$
\begin{array}{lcl}
\alpha & ::= & c\ \vec{x}:\vec{t} \ \mid\ c\ \vec{e} \\
 & | & \neg\alpha \ \mid\ \alpha_1 \wedge \alpha_2 \ \mid\ \alpha_1 \vee \alpha_2
\end{array}
$$

State formulas:

$$
\begin{array}{lcl}
\varphi & ::= & tt \ \mid\ ff \ \mid\ e \rightarrow \varphi_1 \, [] \, \varphi_2 \\
 & | & \neg\varphi \ \mid\ \varphi_1 \wedge \varphi_2 \ \mid\ \varphi_1 \vee \varphi_2 \\
 & | & \langle\alpha\rangle\varphi \ \mid\ [\alpha]\varphi \\
 & | & Y(\vec{e}) \ \mid\ \mu Y(\vec{x}:\vec{t}:=\vec{e}).\varphi \ \mid\ \nu Y(\vec{x}:\vec{t}:=\vec{e}).\varphi
\end{array}
$$

Semantics of the logic (1)

Expressions:

$$[\![.]\!] : \mathit{Exp} \rightarrow \mathbf{DEnv} \rightarrow \mathbf{Val}$$

$$\begin{aligned} [\![x]\!] \varepsilon &= \varepsilon(x) \\ [\![f(\vec{e})]\!] \varepsilon &= f([\![\vec{e}]\!] \varepsilon) \end{aligned}$$

Action formulas:

$$[\![.]\!] : \mathit{AForm} \rightarrow \mathbf{DEnv} \rightarrow A \rightarrow \mathbf{Bool} \times \mathbf{DEnv}$$

$$\begin{aligned} [\![c\ \vec{x} : \vec{t}]\!] \varepsilon a &= \mathit{if}\ \exists \vec{v} : \vec{t}. a = c\ \vec{v}\ \mathit{then}\ (\mathbf{tt}, [\vec{v} / \vec{x}])\ \mathit{else}\ (\mathbf{ff}, [\,]) \\ [\![c\ \vec{e}]\!] \varepsilon a &= \mathit{if}\ a = c\ [\![\vec{e}]\!] \varepsilon\ \mathit{then}\ (\mathbf{tt}, [\,])\ \mathit{else}\ (\mathbf{ff}, [\,]) \\ [\![\neg \alpha]\!] \varepsilon a &= (\mathit{not}\ ([\![\alpha]\!] \varepsilon a)_1, [\,]) \\ [\![\alpha_1 \wedge \alpha_2]\!] \varepsilon a &= (([\![\alpha_1]\!] \varepsilon a)_1\ \mathit{and}\ ([\![\alpha_2]\!] \varepsilon a)_1, [\,]) \\ [\![\alpha_1 \vee \alpha_2]\!] \varepsilon a &= (([\![\alpha_1]\!] \varepsilon a)_1\ \mathit{or}\ ([\![\alpha_2]\!] \varepsilon a)_1, [\,]) \end{aligned}$$

Semantics of the logic (2)

State formulas:

$$[\![.]\!] : SForm \rightarrow \mathbf{PEnv} \rightarrow \mathbf{DEnv} \rightarrow 2^S$$

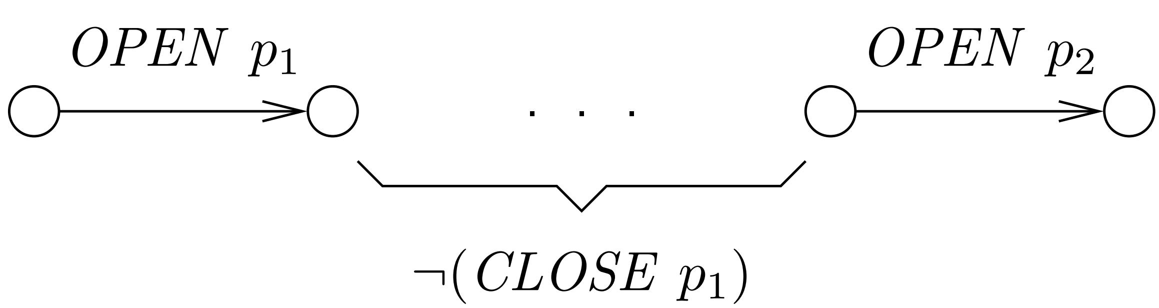
$$\begin{aligned}
[\![tt]\!]\rho\varepsilon &= S \\
[\![f\!f]\!]\rho\varepsilon &= \emptyset \\
[\![e \rightarrow \varphi_1 \,[]\, \varphi_2]\!]\rho\varepsilon &= \mathit{if}\ [\![e]\!]\varepsilon\ \mathit{then}\ [\![\varphi_1]\!]\rho\varepsilon\ \mathit{else}\ [\![\varphi_2]\!]\rho\varepsilon \\
[\![\neg\varphi]\!]\rho\varepsilon &= S \setminus [\![\varphi]\!]\rho\varepsilon \\
[\![\varphi_1 \wedge \varphi_2]\!]\rho\varepsilon &= [\![\varphi_1]\!]\rho\varepsilon \cap [\![\varphi_2]\!]\rho\varepsilon \\
[\![\varphi_1 \vee \varphi_2]\!]\rho\varepsilon &= [\![\varphi_1]\!]\rho\varepsilon \cup [\![\varphi_2]\!]\rho\varepsilon \\
[\![\langle\alpha\rangle\varphi]\!]\rho\varepsilon &= \{s \in S \mid \exists s' \in S, a \in A. s \xrightarrow{a} s' \wedge ([\![\alpha]\!]\varepsilon a)_1 \\
&\qquad \wedge s \in [\![\varphi]\!]\rho(\varepsilon \oslash ([\![\alpha]\!]\varepsilon a)_2)\} \\
[\![[\alpha]\varphi]\!]\rho\varepsilon &= \{s \in S \mid \forall s' \in S, a \in A. (s \xrightarrow{a} s' \wedge ([\![\alpha]\!]\varepsilon a)_1) \\
&\qquad \Rightarrow s \in [\![\varphi]\!]\rho(\varepsilon \oslash ([\![\alpha]\!]\varepsilon a)_2)\} \\
[\![Y(\vec{e})]\!]\rho\varepsilon &= (\rho(Y))([\![\vec{e}]\!]\varepsilon) \\
[\![\mu Y(\vec{x}:\vec{t}:=\vec{e}).\varphi]\!]\rho\varepsilon &= (\sqcap \{F:\vec{t} \rightarrow 2^S \mid \Phi_{\rho\epsilon}(F) \sqsubseteq F\})([\![\vec{e}]\!]\varepsilon) \\
[\![\nu Y(\vec{x}:\vec{t}:=\vec{e}).\varphi]\!]\rho\varepsilon &= (\sqcup \{F:\vec{t} \rightarrow 2^S \mid F \sqsubseteq \Phi_{\rho\epsilon}(F)\})([\![\vec{e}]\!]\varepsilon)
\end{aligned}$$

where $\Phi_{\rho\epsilon}(F) = \lambda \vec{v}:\vec{t}.[\![\varphi]\!](\rho \oslash [F/Y])(\epsilon \oslash [\vec{v}/\vec{x}])$

Safety properties

Mutual exclusion:

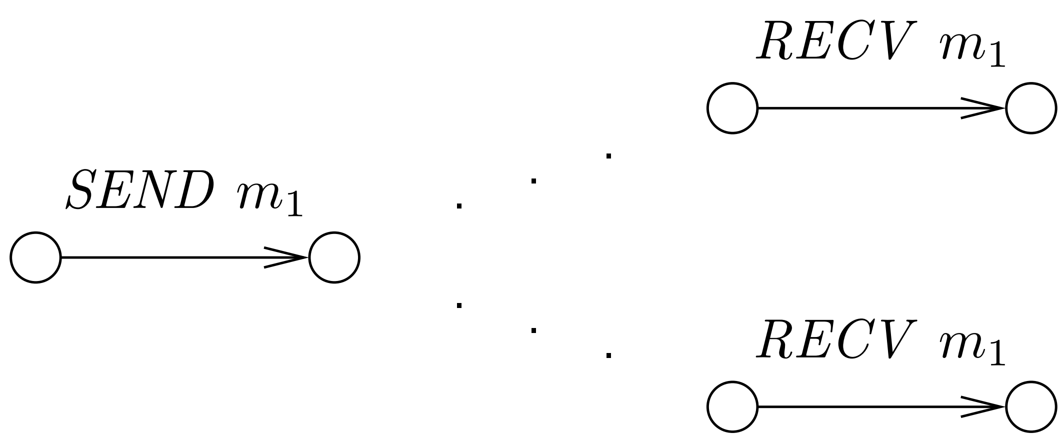
$$
\begin{aligned}
[OPEN\ p_1{:}Pid]\ \neg\ \mu Y(p{:}Pid{:=}p_1).(&\\
&\langle OPEN\ p_2{:}Pid\rangle(p_2{\neq}p)\ \vee\\
&\langle\neg(CLOSE\ p)\rangle Y(p)\\
)&
\end{aligned}
$$



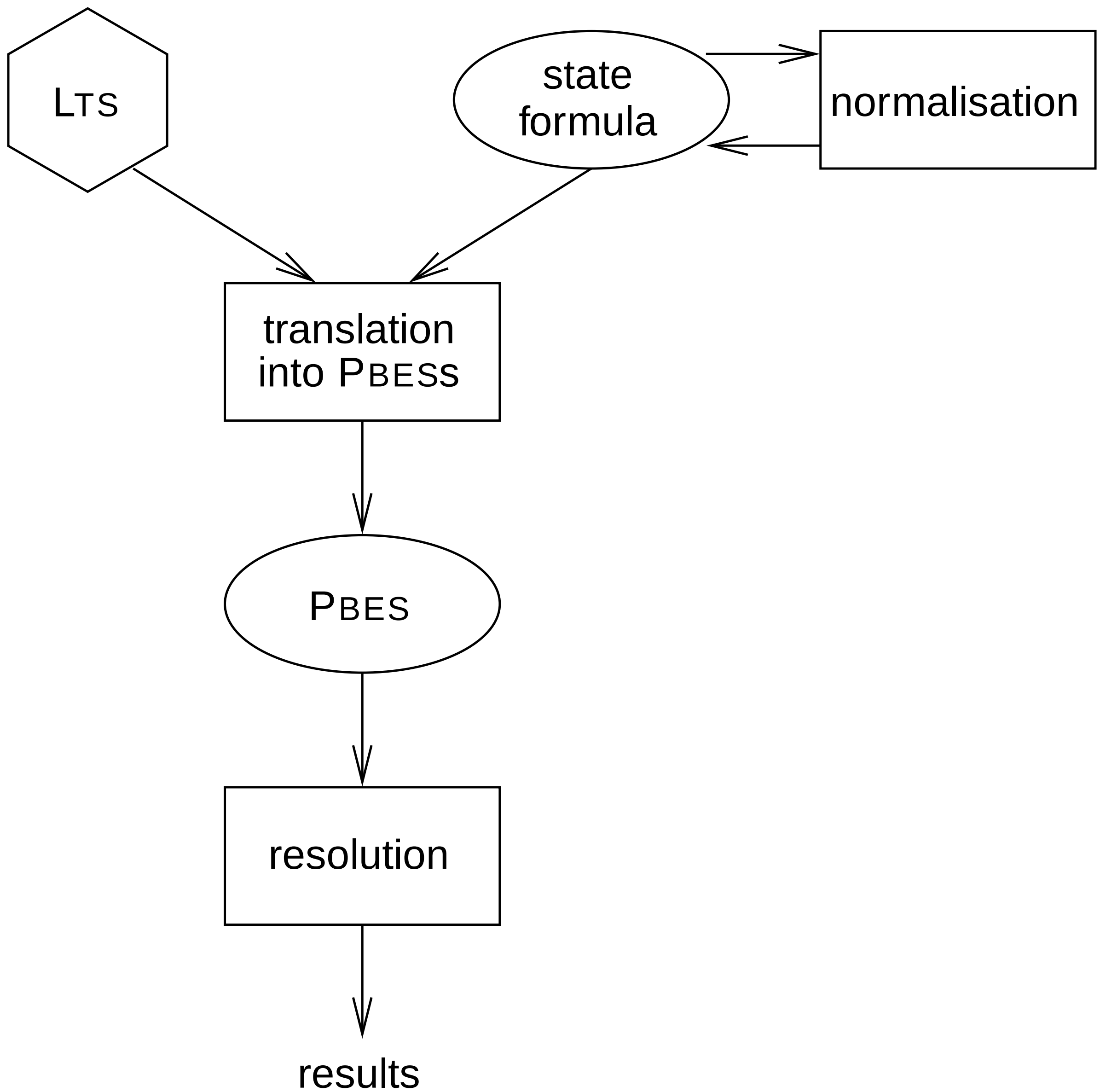
Liveness properties

Correct message transmission:

$$[SEND\ m_1{:}Msg]\ \ \mu Y(m{:}Msg{:=}m_1).(\\ \langle tt\rangle tt \wedge\\ [\neg(RECV\ m)]Y(m)\\ )$$



Local model-checking

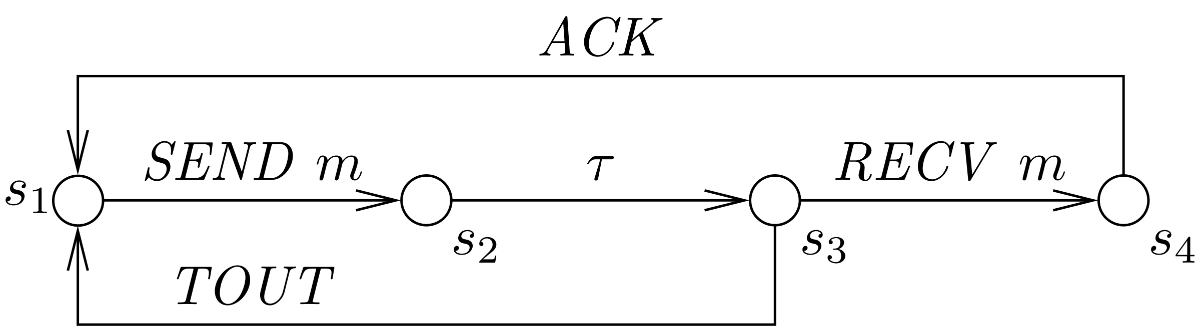


Example (1)

State formula:

$$
\begin{aligned}
&\nu Y_1.([SEND\ m_1{:}Msg]\\
&\quad \mu Y_2(m_2{:}Msg{:=}m_1).(\langle RECV\ m_2\rangle tt \vee\\
&\qquad\qquad\qquad \langle\neg(SEND\ m_2)\rangle Y_2(m_2))\\
&\quad \wedge\ [tt]Y_1)
\end{aligned}
$$

LTS model:

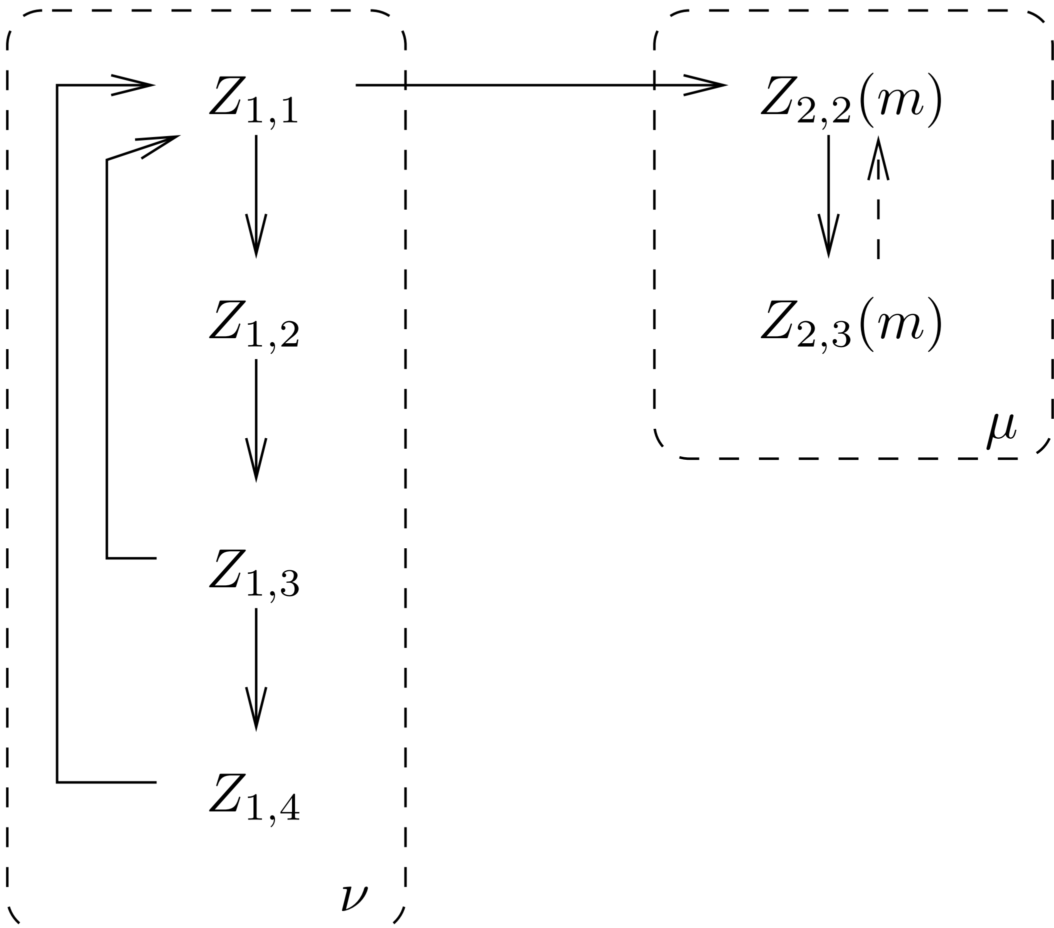


Translation into a PBES:

$$
\left\{
\begin{aligned}
Z_{1,1} &\stackrel{\nu}{=} Z_{2,2}(m) \wedge Z_{1,2}\\
Z_{1,2} &\stackrel{\nu}{=} Z_{1,3}\\
Z_{1,3} &\stackrel{\nu}{=} Z_{1,1} \wedge Z_{1,4}\\
Z_{1,4} &\stackrel{\nu}{=} Z_{1,1}
\end{aligned}
\right.
\quad
\left\{
\begin{aligned}
Z_{2,1}(m_2{:}Msg) &\stackrel{\mu}{=} m \neq m_2 \wedge Z_{2,2}(m_2)\\
Z_{2,2}(m_2{:}Msg) &\stackrel{\mu}{=} Z_{2,3}(m_2)\\
Z_{2,3}(m_2{:}Msg) &\stackrel{\mu}{=} m = m_2 \vee Z_{2,1}(m_2)\\
Z_{2,4}(m_2{:}Msg) &\stackrel{\mu}{=} Z_{2,1}(m_2)
\end{aligned}
\right.
$$

Example (2)

Resolution of the PBES:



Discussion

Complexity of the algorithm:

- linear in the size of the dependency graph between boolean instances $Z_{i,j}(\vec{v})$

In general:

- termination not guaranteed (possibly infinite dependency graph)

In practice:

- fixed points without parameters:

$$O(|\varphi| \cdot (|S| + |T|))$$

- fixed points with *restricted parameters* [Rathke-Hennessy-96]:

$$O\Big(|\varphi| \cdot (|S| + |T|) \cdot |A|^{arity(\varphi)}\Big)$$

Conclusion

Results:

- definition of a value-based mu-calculus
- local model-checking algorithm for the alternation-free fragment

Future work:

- implementation of the model-checking algorithm
- extension to the full mu-calculus
- application to abstract interpretation