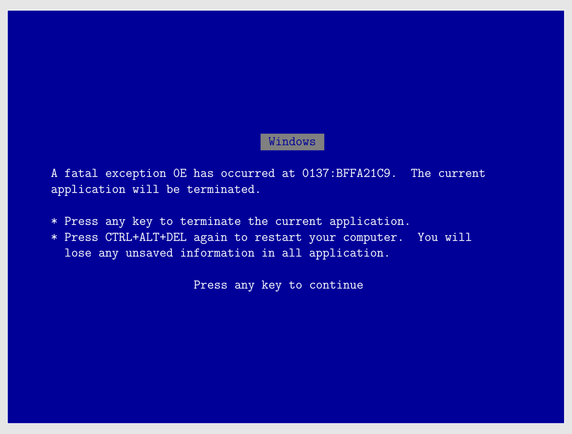
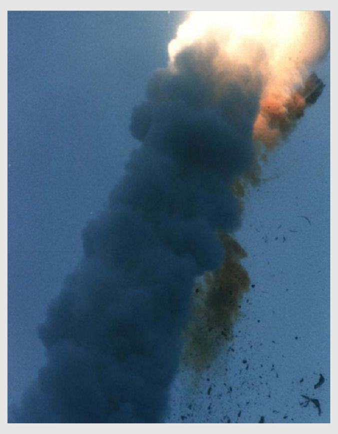
Probabilistic specifications with data types

Joost-Pieter Katoen, Jaco van de Pol, Mariëlle Stoelinga, Mark Timmer
Formal Methods and Tools – Department of Computer Science – University of Twente

1. Introduction

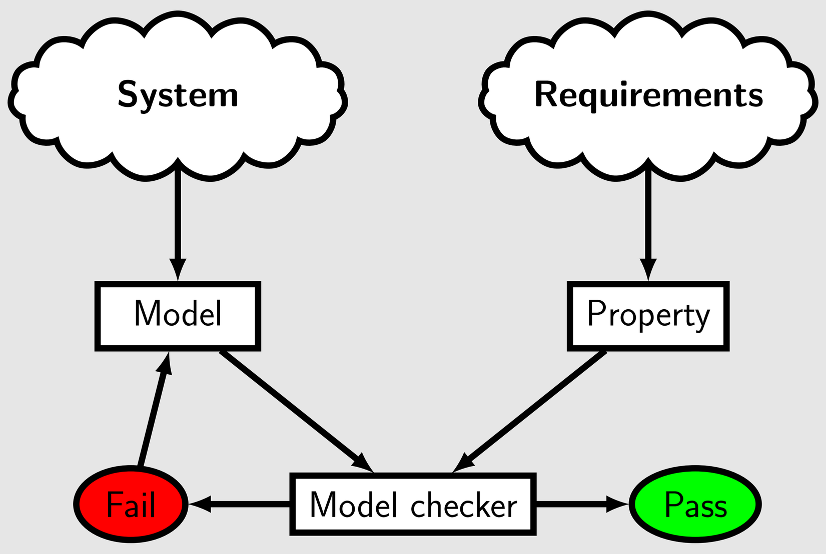
Dependability of computer systems is becoming more and more important.


Windows blue screen


Ariane 5 crash

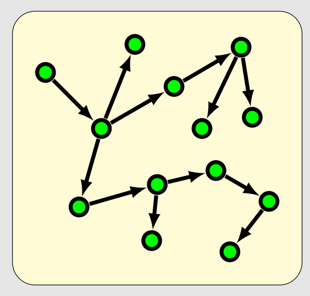
Our aim is to use formal methods to improve system quality.

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

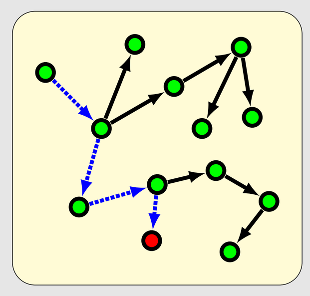


An overview of model checking

Pass:



Fail:



The output of model checking

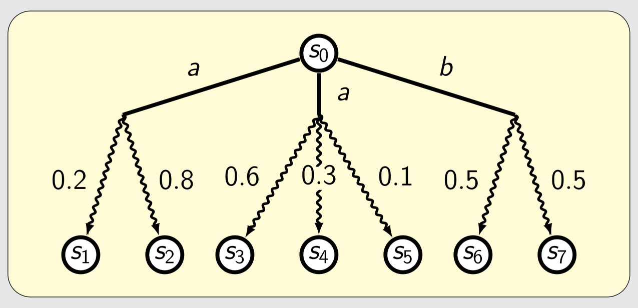
2. Probabilistic model checking

Probabilistic model checking:

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

Applications:

- Dependability analysis
- Performance analysis



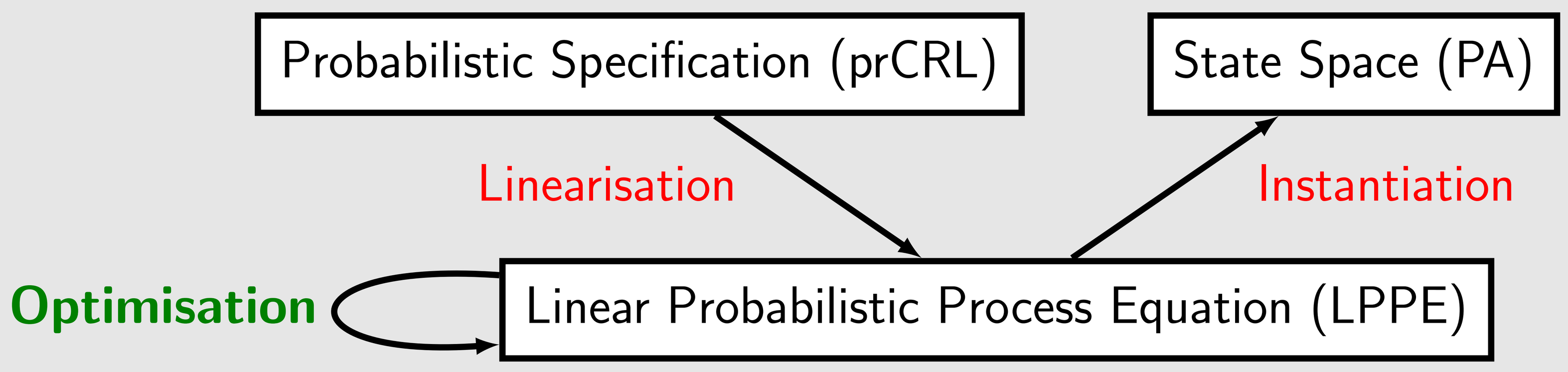
A probabilistic automaton (PA)

Limitations of previous approaches:

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

3. Our approach; an overview

Main idea: we introduce a process algebra prCRL, incorporating both data types and probabilistic choice. It has a linear format (the LPPE), enabling symbolic optimisations at the language level. Therefore, the state space can be reduced before it is generated.



4. The process algebra prCRL

We introduce the specification language prCRL, give by

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

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

- 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

Example: $X = \tau \sum\!\!\!\!\bullet_{n:\mathbb{N}} \frac{1}{2^n} : send(n) \cdot X$. This specification repeatedly chooses a natural number n with probability $\frac{1}{2^n}$, and then sends the number.

5. The linear format: LPPE

We define LPPEs (linear probabilistic process equations) as follows:

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

Advantages of LPPEs:

- The state space can be generated very easily
- Parallel composition can be applied in a straight-forward manner
- Symbolic optimisations are enabled at the language level

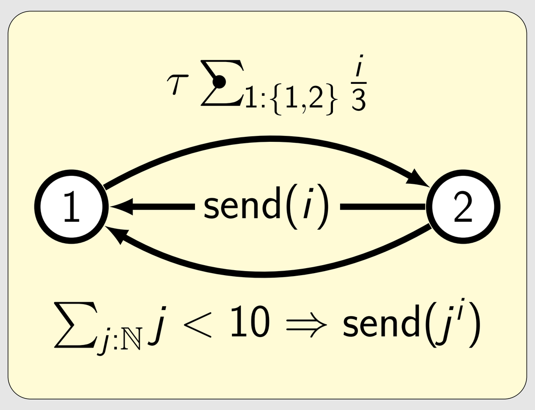
6. Linearisation

Given the following specification in prCRL:

$$X = \tau \sum\!\!\!\!\bullet_{i:\{1,2\}} \frac{i}{3} : \left(\text{send}(i) \cdot X + \sum_{j:\mathbb{N}} j < 10 \Rightarrow \text{send}(j^i) \cdot X \right)$$

The corresponding linear form is:

$$\begin{aligned} & X(\text{pc} : \{1,2,3\}, i : \{1,2\}) = \\ & \qquad \text{pc} = 1 \Rightarrow \tau \textstyle\sum\!\!\!\!\bullet_{i:\{1,2\}} \frac{i}{3} : X(2, i) \\ + & \qquad \text{pc} = 2 \Rightarrow \text{send}(i) \cdot X(1, i) \\ + & \sum_{j:\mathbb{N}} \text{pc} = 2 \wedge j < 10 \Rightarrow \text{send}(j^i) \cdot X(1, i) \end{aligned}$$



A graphical representation of X

For more complicated systems the ideas behind linearisation remain the same:

- Introduce a program counter to remember the location in the formula
- Introduce global parameters to remember bound variables

We developed an algorithm to transform any prCRL specification to an LPPE, proved it correct, and implemented it.

7. Results and Future Work

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 their corresponding LPPE, proved it correct, and implemented it.

Future work:

- Applying existing optimisation techniques, such as constant elimination, liveness analysis and confluence reduction, to LPPEs.

8. Acknowledgments

This research is supported by NWO under grant 612.063.817 (SYRUP) and grant Dn 63-257 (ROCKS), and by the European Union under FP7-ICT-2007-1 grant 214755 (QUASIMODO).

UNIVERSITY OF TWENTE.

Formal Methods & Tools

CTIT Centre for Telematics and Information Technology

NWO Netherlands Organisation for Scientific Research