PRISM Model Checker

http://www.prismmodelchecker.org

Probabilistic Model Checker

Suitable for the analysis of 3 types of models:
- Discrete Time Markov Models (DTMC)
- Markov decision Processes (DTMC + non-deterministic choice)
- Continuous Time Markov Chains (CTMC)

Models are defined in a high-level language:
- State-based language
- ‘Reactive Modules’ (Alur & Henzinger 1999)

Properties of models can be expressed in two temporal logics:
- Probabilistic Computation Tree Logic (PCTL) for DTMC and MDP models
- Continuous Stochastic Logic (CSL) for CTMC models

Developed at the University of Birmingham, U.K., now moved to the University of Oxford:

Prof. M. Kwiatkowska
Dr. D. Parker
Dr. G. Norman
The PRISM language

Composition:

```
module 'name1'
y: ...
endmodule

system ....

name || name2 // |[a,b,...]|
endsystem
```

Synchronisation: operations similar to those of CSP

- `M1 || M2` parallel composition on common actions
- `M1 ||| M2` interleaving
- `M1 || [a, b, ...] || M2` parallel composition with synchronisation on actions `a, b, ...`

Example:

Mutual Exclusion

State 0: non-critical, state 1: request to enter, state 2: critical section

Mutual Exclusion

In PRISM language

```
dime
module M1
x: [0, 2] init 0;
y: [0, 2] init 0;
[| x0 = a8; (x' = a8) + o2; (y' = 0)| x1 = a8; (y' = o2) + a2; (y' = x1)|
| y1 = b8; (y' = b8) + x2; (x' = 0)| y2 = b8; (x' = y2) + x2; (x' = y2)|
| x2 = a5; (x = y2) + a5; (y = 0)| x2 = a5; (y = x2) + a5; (y = 0)|
endmodule

module M2
y: ...
endmodule

system M1 || M2. endsystem
```

PCTL properties

\[ P_{>0}(tt U (x = 2 | y = 2)) \]
Mutual Exclusion

PCTL properties

\[ P_{>0}(\mathit{tt} \cup (x = 2 \mid y = 2)) \]
\[ P_{>0}(\mathit{tt} \cup (x = 2 \land y = 2)) \]

\[ P_{>0.3}(\mathit{tt} \mathcal{U}^5(x = 2)) \]
\[ P_{=7}(\mathit{tt} \mathcal{U}^5(x = 2)) \]

M. Massink (CNR-ISTI)  Part 4: PRISM Model Checker  PMC2020 8/12
The PRISM language

Informal semantics

DTMC

every ‘enabled’ command is selected with the same probability

For example: consider state \((x,y) = (0,0)\)

Two enabled commands:

1. \(x = 0\) in module \(M1\)
2. \(y = 0\) in module \(M2\)

With probability 0.5 the first option is selected:
so, with probability 0.5 \(\times 0.8\) we have \(x' = 0\)

With probability 0.5 the second option is selected:
so, with probability 0.5 \(\times 0.8\) we have \(y' = 0\)

Example: action synchronisation

PRISM produces automatically the underlying DTMC as the product of the components (modules) before starting the model checking procedure.

This way large DTMCs can be specified exploiting compositionality.