

This is a repository copy of *Probabilistic modelling and verification, and Animation in RoboChart*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/193112/>

Version: Published Version

Conference or Workshop Item:

Ye, Kangfeng (2022) Probabilistic modelling and verification, and Animation in RoboChart.
In: YorRobots and RoboStar Industry Exhibition, 11-12 Oct 2022, University of York.

Reuse

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial (CC BY-NC) licence. This licence allows you to remix, tweak, and build upon this work non-commercially, and any new works must also acknowledge the authors and be non-commercial. You don't have to license any derivative works on the same terms. More information and the full terms of the licence here:
<https://creativecommons.org/licenses/>

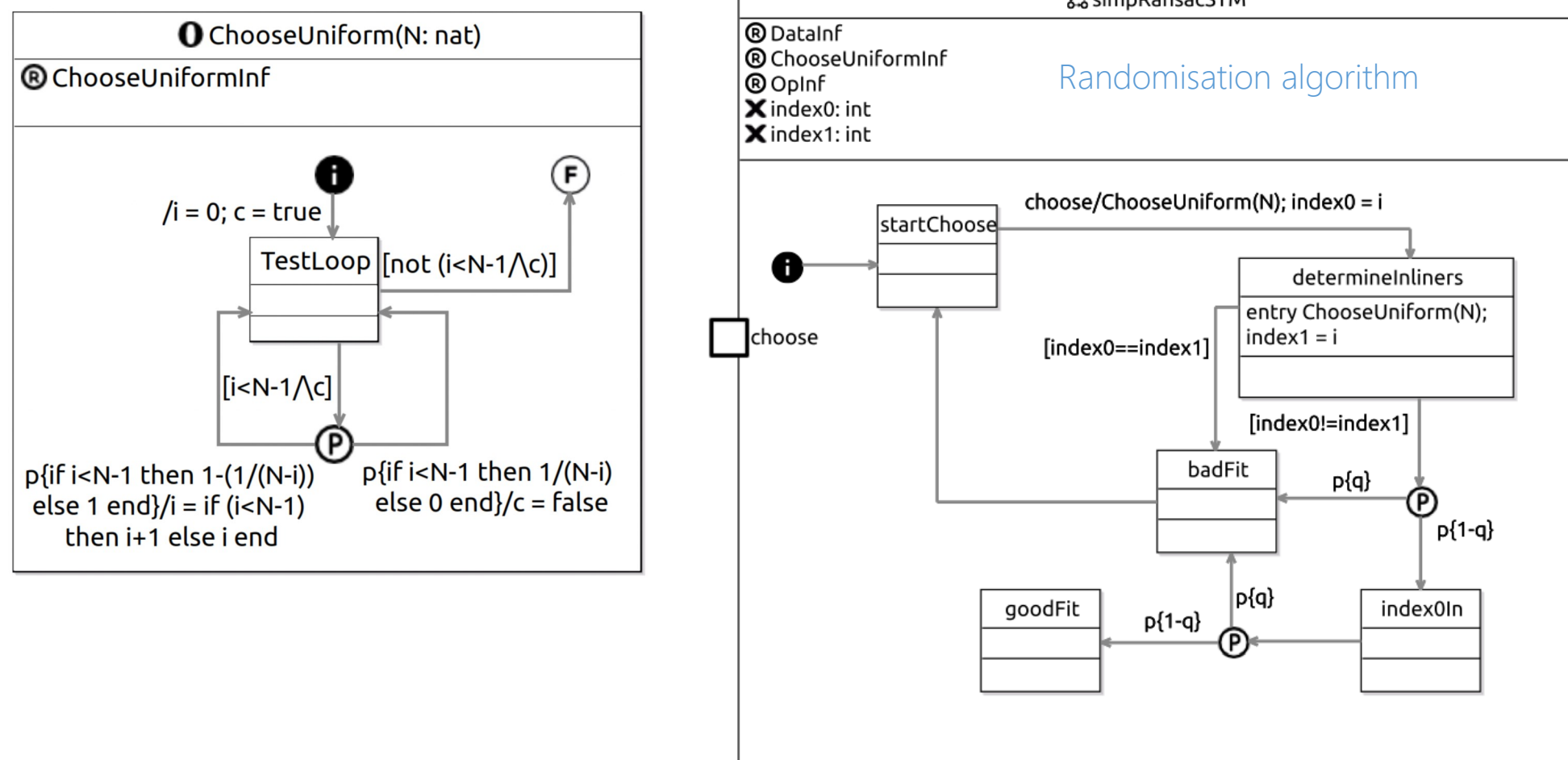
Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Probabilistic modelling and verification, and Animation in RoboChart

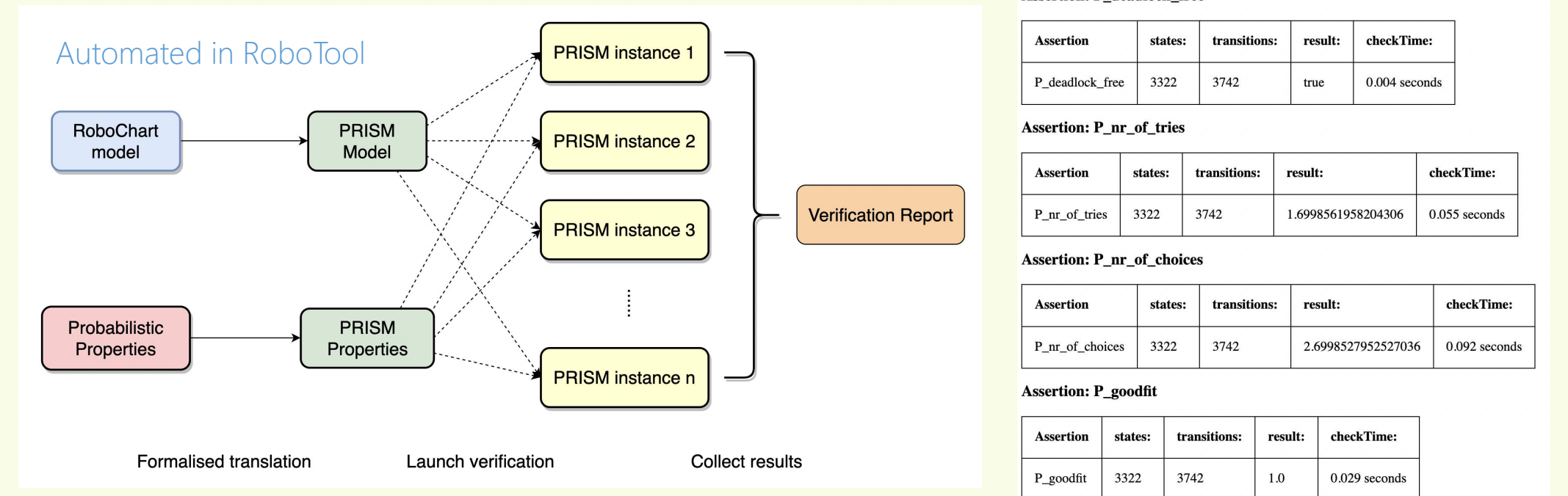
Probabilistic Modelling

- Probabilistic choice is made at **probabilistic junctions**.
- Each outgoing transition must have a probability value (inside `p{}`) between **0.0** and **1.0**.
- Probability values of all outgoing transitions from a probabilistic junction sum to **1.0**.



Probabilistic Model Checking

- RoboChart's probabilistic semantics given in MDP and automated generation of semantics for PRISM in RoboTool
- Formalised translation from RoboChart to PRISM
- Run multiple instances of PRISM: one for each property
- Easily extended to other probabilistic model checkers like Storm and MODEST.



Probabilistic Property Language

- Based on the PRISM's property language (**PCTL***) for DTMC and MDP
- Allow to specify properties using variables, expressions, states, events, functions, operations, etc. from RoboChart models.
- Properties are specified in a particular constant configuration, function definitions, or uncertain environment.

```
constants C1:           A constant configuration
ransacMOD::ransacRP::N set to 6,
and ransacMOD::ransacRP::p set to 1/3
prob property P_deadlock_free:
not Exists [Finally deadlock]
with constants C1
```

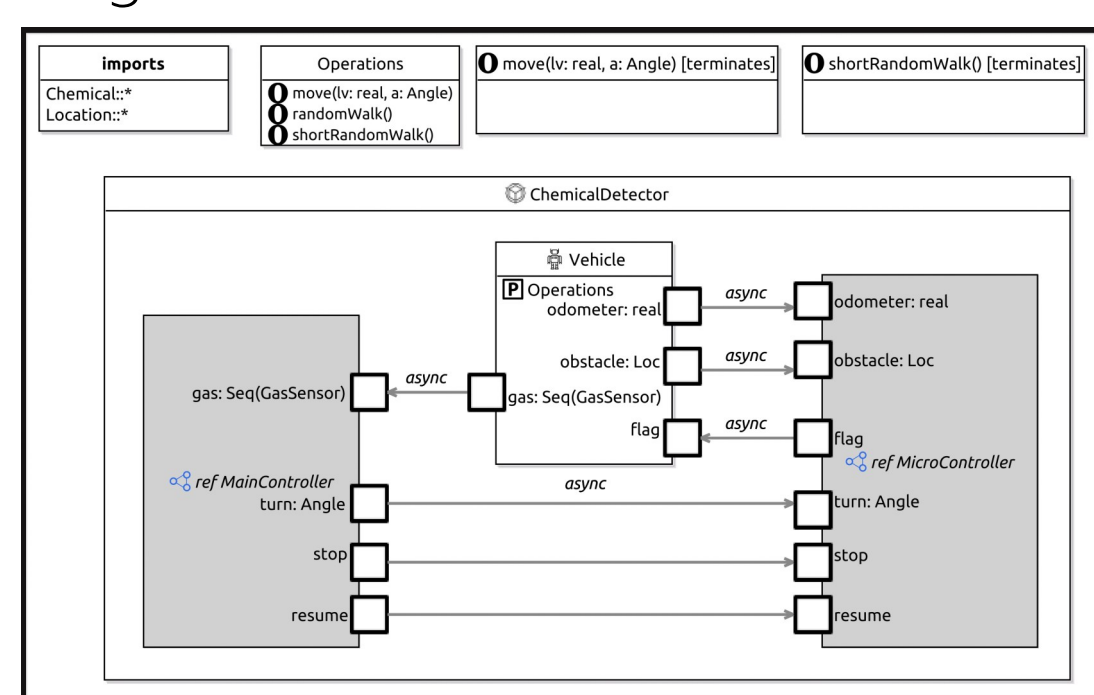
```
constants C1:           Simulation with multiple constant configurations
MAX from set {10,000 to 80,000 by step 1000},
M set to 30, and N set to 30
prob property P_prob_full_coverage_bound:
Prob>=0.8 of [Finally $visits == M * N]
using sim with CI at alpha=0.01, n=1000, and pathlen
=1000000
with constant C1
```

```
prob property P_goodfit:           Quantitative property
Prob=? of [Finally ransacMOD::ransacCTRL::stm_ref0
is in ransacMOD::ransacCTRL::stm_ref0::goodFit]
```

```
prob property P_1:           LTL
Forall [Globally (Finally (fd==2) and (Next (fd==0)))]
```

Formally Verified Animation for RoboChart

- Operational semantics of RoboChart given in interaction trees (mechanised in Isabelle/HOL)
- Generated Haskell code for animation (on terminal now)
- Able to animate a state machine, an operation, a controller, or a whole model



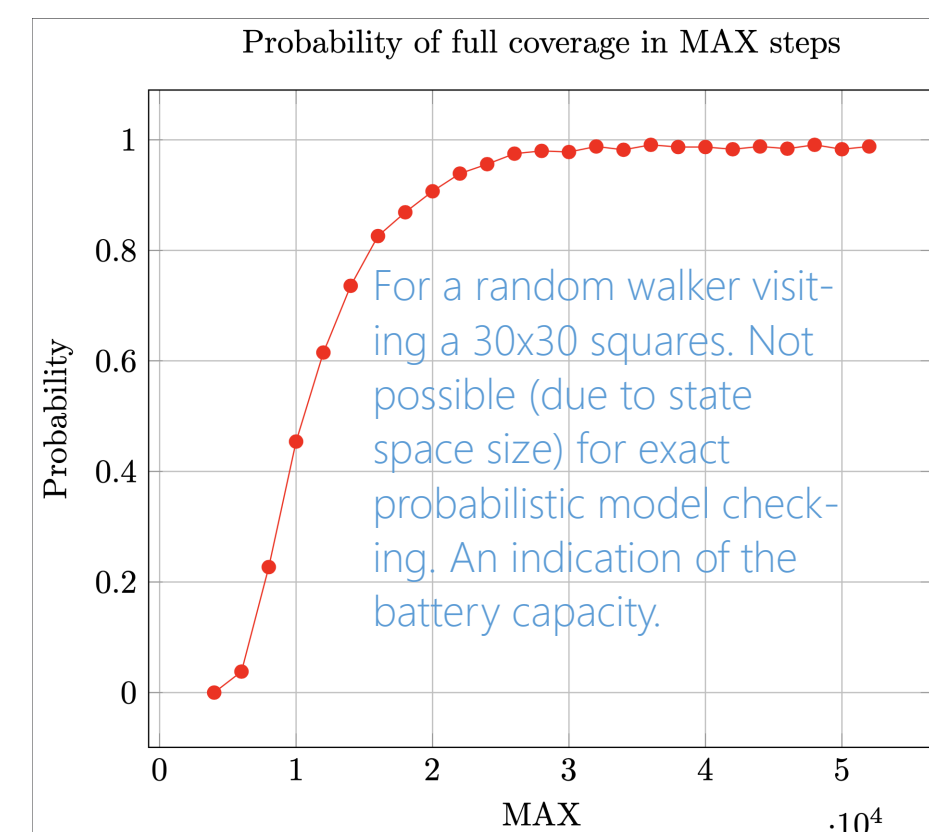
Animation of a chemical detector model in a scenario: detected an intensive gas.



```
Internal Activity...
Events: (1) RandomWalkCall (); (2) Gas (Din, []); (3) Gas (Din, [(0, 0)]); (4) Gas (Din, [(0, 1)]); (5) Gas (Din, [(1, 0)]); (6) Gas (Din, [(1, 1)]); (7) Gas (Din, [(0, 0), (0, 0)]); (8) Gas (Din, [(0, 0), (0, 1)]); (9) Gas (Din, [(0, 0), (1, 0)]); (10) Gas (Din, [(0, 0), (1, 1)]); (11) Gas (Din, [(0, 1), (0, 0)]); (12) Gas (Din, [(0, 1), (0, 1)]); (13) Gas (Din, [(0, 1), (1, 0)]); (14) Gas (Din, [(0, 1), (1, 1)]); (15) Gas (Din, [(1, 0), (0, 0)]); (16) Gas (Din, [(1, 0), (0, 1)]); (17) Gas (Din, [(1, 0), (1, 0)]); (18) Gas (Din, [(1, 0), (1, 1)]); (19) Gas (Din, [(1, 1), (0, 0)]); (20) Gas (Din, [(1, 1), (0, 1)]); (21) Gas (Din, [(1, 1), (1, 0)]); (22) Gas (Din, [(1, 1), (1, 1)]);
1
[Choose 1-22]: RandomWalkCall ()
Events: (1) Gas (Din, []); (2) Gas (Din, [(0, 0)]); (3) Gas (Din, [(0, 1)]); (4) Gas (Din, [(1, 0)]); (5) Gas (Din, [(1, 1)]); (6) Gas (Din, [(0, 0), (0, 0)]); (7) Gas (Din, [(0, 0), (0, 1)]); (8) Gas (Din, [(0, 0), (1, 0)]); (9) Gas (Din, [(0, 0), (1, 1)]); (10) Gas (Din, [(0, 1), (0, 0)]); (11) Gas (Din, [(0, 1), (0, 1)]); (12) Gas (Din, [(0, 1), (1, 0)]); (13) Gas (Din, [(0, 1), (1, 1)]); (14) Gas (Din, [(1, 0), (0, 0)]); (15) Gas (Din, [(1, 0), (0, 1)]); (16) Gas (Din, [(1, 0), (1, 0)]); (17) Gas (Din, [(1, 0), (1, 1)]); (18) Gas (Din, [(1, 1), (0, 0)]); (19) Gas (Din, [(1, 1), (0, 1)]); (20) Gas (Din, [(1, 1), (1, 0)]); (21) Gas (Din, [(1, 1), (1, 1)]);
9
[Choose 1-21]: Gas (Din, [(0, 0), (1, 1)])
Internal Activity...
Events: (1) MoveCall (0,Chemical_Angle_Front);
1
[Choose 1-1]: MoveCall (0,Chemical_Angle_Front)
Events: (1) Flag Dout;
1
[Choose 1-1]: Flag Dout
Internal Activity...
Terminated: ()
```

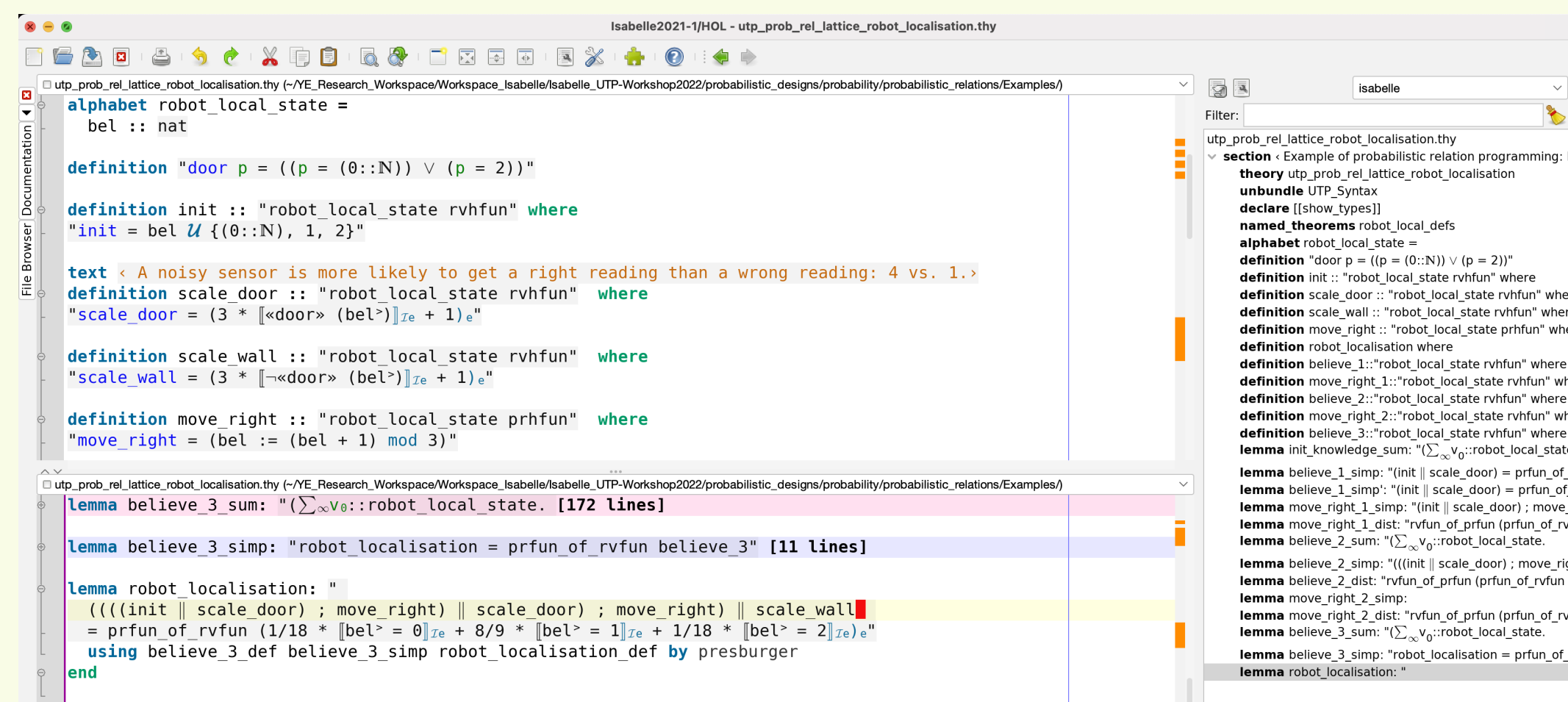
Statistic Model Checking

- Approximate results
- Analyse properties on a large number of (Monte Carlo) simulations
- Able to analyse big models
- Illustrations and debugging problems
- Design space exploration (DSE)
- Generate test cases that satisfy or



Theorem Proving

- Denotational semantics of probabilistic programs in UTP
- Both epistemic and aleatoric uncertainty
- Mechanised in Isabelle/UTP



- Able to reason about large models with an infinite state space

For any $N \geq 1$,

$$\left(\text{true} \vdash \left(\forall j \bullet j < (N - 1) \Rightarrow (\text{prob}'(\forall[j, \text{false}/i, c] = 1/N)) \wedge \right) \right) \sqsubseteq \text{ChooseUniform}(N)$$

- Bayesian belief model

Localisation: robot's belief in its current position changed after 3 sensor readings and two movements: very likely (nearly 90%) it is in front of wall now.

