Probabilistic modelling and verification, and Animation in RoboChart

Kangfeng Ye, Jim Woodcock, and Simon Foster

ROBOSTAR

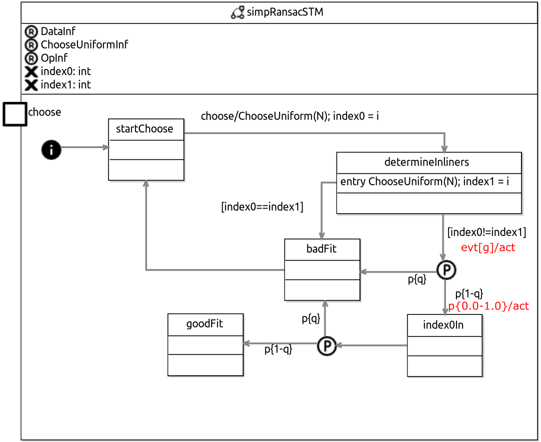
robostar.cs.york.ac.uk

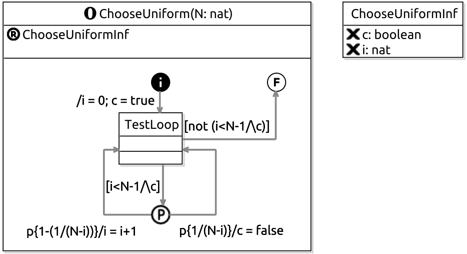
October, 2022

UNIVERSITY of York

Engineering and Physical Sciences Research Council

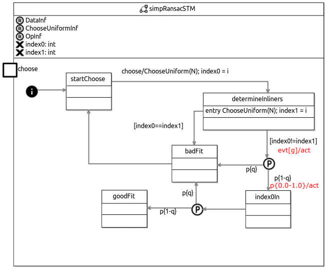
Probabilistic modelling: probabilistic junctions





ROBOSTAR

Probabilistic verification: probabilistic property language

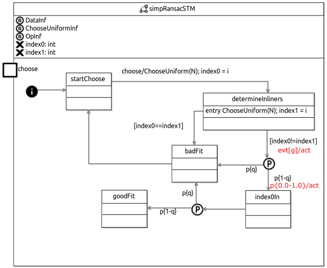



```
constants C1:
  ransacMOD::ransacRP::N set to 6,
  and ransacMOD::ransacRP::p set to 1/3
prob property P_deadlock_free:
  not Exists [ Finally deadlock]
  with constants C1

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

prob property P_nr_of_choices:
  Reward {nrchoices} =? of [
        Reachable ransacMOD::ransacCTRL::stm_ref0 is in
          ransacMOD::ransacCTRL::stm_ref0::goodFit]
```

Probabilistic verification: model checking with PRISM



Result report

Assertion: P_deadlock_free

Assertion	states:	transitions:	result:	checkTime:
P_deadlock_free	3322	3742	true	0.004 seconds

Assertion: P_nr_of_tries

Assertion	states:	transitions:	result:	checkTime:
P_nr_of_tries	3322	3742	1.6998561958204306	0.055 seconds

Assertion: P_nr_of_choices

Assertion	states:	transitions:	result:	checkTime:
P_nr_of_choices	3322	3742	2.6998527952527036	0.092 seconds

Assertion: P_goodfit

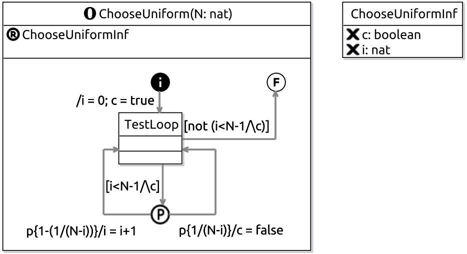
Assertion	states:	transitions:	result:	checkTime:
P_goodfit	3322	3742	1.0	0.029 seconds

Large finite model or infinite model

State space explosion

Previous example analyses $N = 6$. However, if

- $N = 100$: construction (4s) + checking (0.002s);
- $N = 10,000$: 8s + 0.004s;
- $N = 100,000$: 830s + 0.011s;
- $N = 1,000,000$: not finished after several hours;
- $N = 1, \ldots\ldots\ldots\ldots$: ?



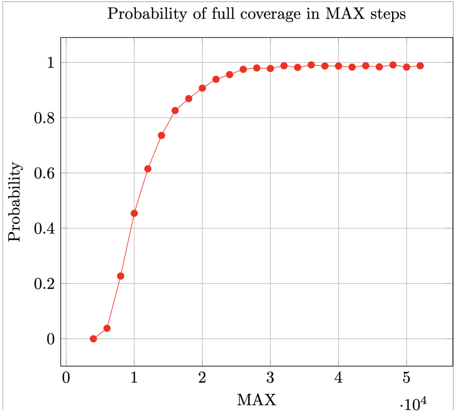
Large finite model or infinite model

Statistical model checking

- Approximate results (vs. exact)
- Monte Carlo simulations (executions)
- Analyse properties on simulations

Random walker

30 x 30 squares

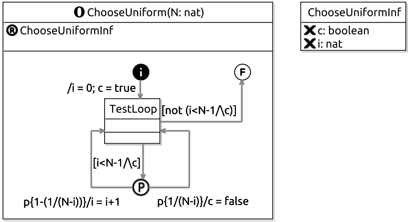


Large finite model or infinite model

Theorem Proving (UTP, Isabelle/UTP)

For any $N \geq 1$,

$$\left(\textbf{true} \vdash \left(\begin{array}{l} (\forall j \bullet j < (N-1) \Rightarrow (prob' \, (\mathbf{v}[j, false/i, c] = 1/N))) \wedge \\ prob' \, (\mathbf{v}[(N-1), true/i, c]) = 1/N \end{array} \right) \right) \sqsubseteq ChooseUniform(N)$$

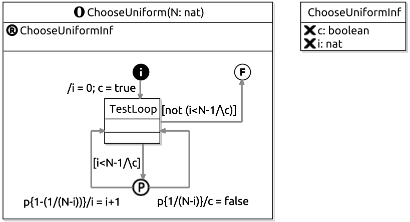


Large finite model or infinite model

Theorem Proving (UTP, Isabelle/UTP)

For any $N \geq 1$,

$$\left(\textbf{true} \vdash \left(\begin{array}{l}(\forall j \bullet j < (N-1) \Rightarrow (prob' \, (\mathbf{v}[j, false/i, c] = 1/N))) \land \\ prob' \, (\mathbf{v}[(N-1), true/i, c]) = 1/N\end{array}\right)\right) \sqsubseteq ChooseUniform(N)$$



Interpretation:

- If j is between 0 and $(N-2)$, $P(i = j) = 1/N$ and $c = false$
- If j is equal to $(N-1)$, $P(i = j) = 1/N$ and $c = true$

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

ROBOSTAR

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

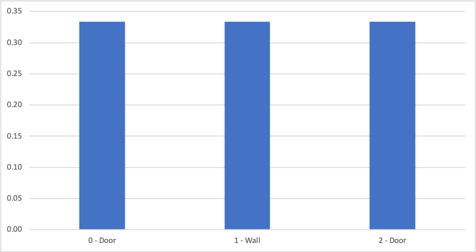
Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

init



Robot's belief

Large finite model or infinite model

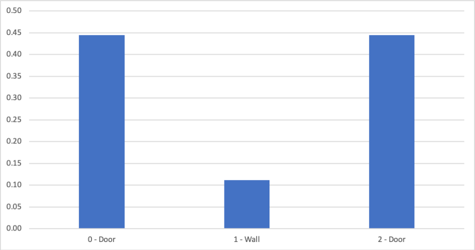
Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

init

init || sdoor



Robot's belief

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

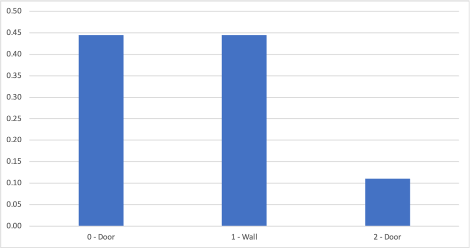
Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

init

init $\parallel$ sdoor

(init $\parallel$ sdoor) ; mright



Robot's belief

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

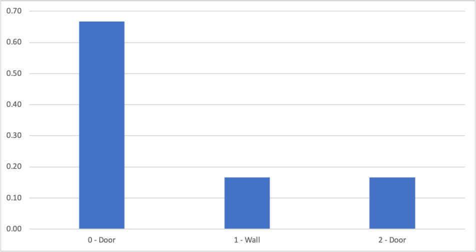
Imperfect door sensor: 4 times more likely to be right than wrong

init

$\mathsf{init} \parallel \mathsf{sdoor}$

$(\mathsf{init} \parallel \mathsf{sdoor})\ ;\ \mathsf{mright}$

$((\mathsf{init} \parallel \mathsf{sdoor})\ ;\ \mathsf{mright}) \parallel \mathsf{sdoor}$



Robot's belief

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

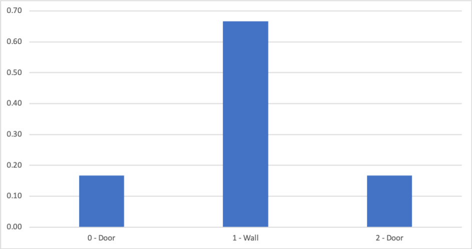
init

init || sdoor

(init || sdoor) ; mright

((init || sdoor) ; mright) || sdoor

(((init || sdoor) ; mright) || sdoor) ; mright



Robot's belief

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

init

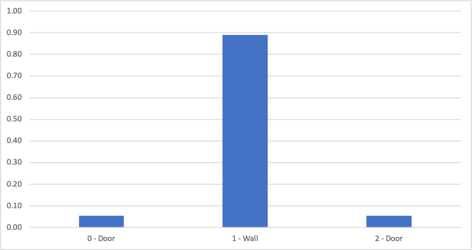
init || sdoor

(init || sdoor) ; mright

((init || sdoor) ; mright) || sdoor

(((init || sdoor) ; mright) || sdoor) ; mright

((((init || sdoor) ; mright) || sdoor) ; mright) || swall



Robot's belief

ROBOSTAR

Large finite model or infinite model

Theorem Proving (epistemic uncertainty)

Bayesian belief model: learn new facts

Imperfect door sensor: 4 times more likely to be right than wrong

init

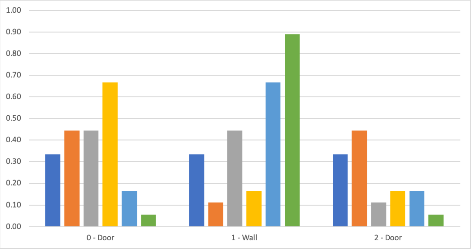
init || sdoor

(init || sdoor) ; mright

((init || sdoor) ; mright) || sdoor

(((init || sdoor) ; mright) || sdoor) ; mright

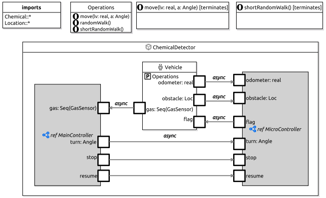
((((init || sdoor) ; mright) || sdoor) ; mright) || swall



Robot's belief

ROBOSTAR

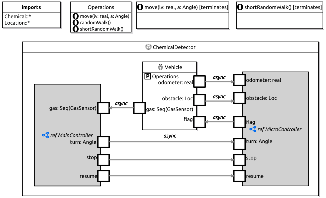
Animation of RoboChart




```
Starting ITree animation...
Events: (1) RandomWalkCall (); (2) Gas (Din, []); ...;
[Choose: 1-22]: 1
Events: (1) Gas []; (2) Gas [(0,0)]; (3) Gas [(0,1)]; ...;
  (9) Gas [(0,0),(1,1)]; ...; (21) Gas [(1,1),(1,1)];
[Choose: 1-21]: 9
Events: (1) MoveCall (0,Chemical_Angle_Front);
[Choose: 1-1]: 1
Events: (1) Flag Dout;
[Choose: 1-1]: 1
Terminated: ()
```

ROBOSTAR

Animation of RoboChart




```
[Choose: 1-22]: 1 RandomWalkCall ()
[Choose: 1-21]: 4 Gas (Din,[(1, 0)])
[Choose: 1-22]: 1 MoveCall (1,Chemical_Angle_Front)
[Choose: 1-24]: 2 Obstacle (Din,Location_Loc_right)
[Choose: 1-23]: 1 Odometer (Din,0)
[Choose: 1-22]: 1 MoveCall (1,Chemical_Angle_Left)
[Choose: 1-21]: 8 Gas (Din,[(0, 0),(1, 0)])
[Choose: 1-22]: 1 MoveCall (1,Chemical_Angle_Front)
[Choose: 1-24]: 1 Obstacle (Din, Location_Loc_left)
[Choose: 1-23]: 2 Odometer (Din,1)
[Choose: 1-23]: 1 Odometer (Din,0)
[Choose: 1-22]: 1 MoveCall (1,Chemical_Angle_Right)
[Choose: 1-21]: 4 Gas (Din,[(1, 0)])
[Choose: 1-22]: 1 MoveCall (1,Chemical_Angle_Front)
[Choose: 1-24]: 2 Obstacle (Din,Location_Loc_right)
[Choose: 1-23]: 1 Odometer (Din,0)
[Choose: 1-22]: 1 Stuck_timeout Din
[Choose: 1-22]: 1 ShortRandomWalkCall ()
```

Thank you!

https://robostar.cs.york.ac.uk/