Probabilistic Hyperproperties with Rewards

NASA Formal Methods 2022

Oyendrila Dobe [#], *Lukas Wilke**, *Erika Ábrahám**, *Ezio Bartocci* **, *Borzoo Bonakdarpour* [#]

[#]Michigan State University (USA), *RWTH Aachen (Germany), **TU-Wien (Austria)

May 27, 2022

MICHIGAN STATE UNIVERSITY

RWTH AACHEN UNIVERSITY

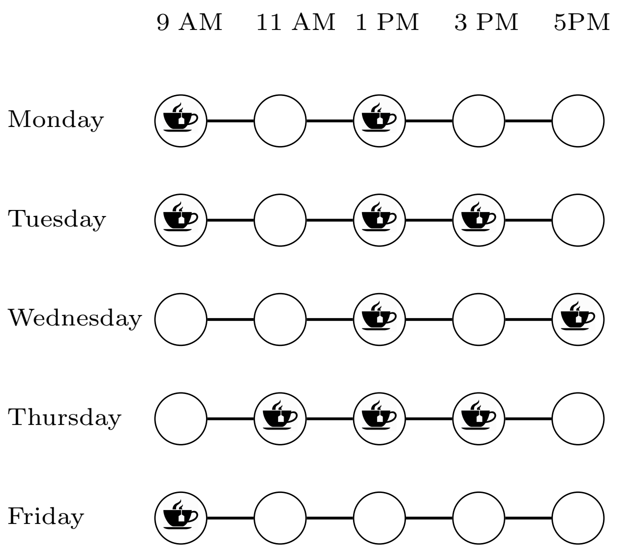
TU WIEN TECHNISCHE UNIVERSITÄT WIEN Vienna | Austria

Trace Property vs Hyperproperty[1]

Trace Property vs Hyperproperty[1]

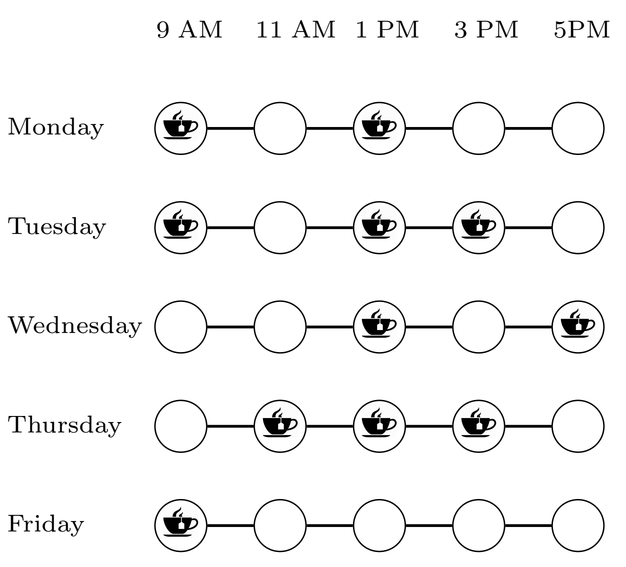
1. ‘Hyperproperties’, Clarkson and Schneider, 2010.

Trace Property vs Hyperproperty[1]



1. 'Hyperproperties', Clarkson and Schneider, 2010.

Trace Property vs Hyperproperty[1]

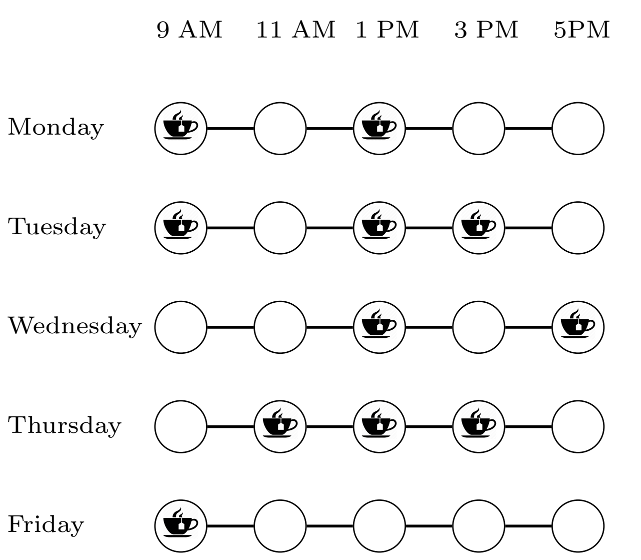


Trace property:

- I drink tea everyday - ◇ ☕ ✓

1. 'Hyperproperties', Clarkson and Schneider, 2010.

Trace Property vs Hyperproperty[1]

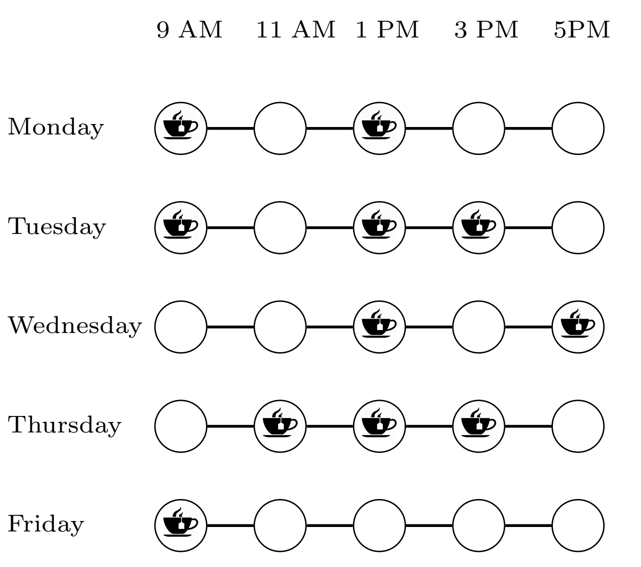


Trace property:

- I drink tea everyday - $\Diamond$ ☕ ✓
- I drink tea at the same time everyday - ✗

1. 'Hyperproperties', Clarkson and Schneider, 2010.

Trace Property vs Hyperproperty[1]



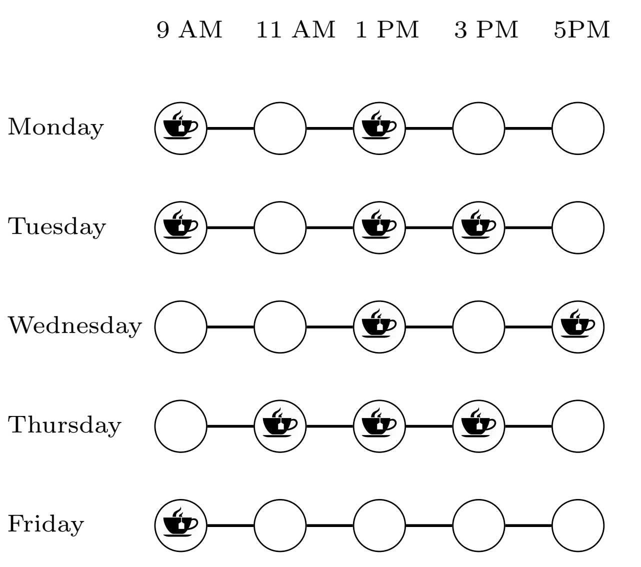
Trace property:

- I drink tea everyday - $\Diamond$ ☕ ✓
- I drink tea at the same time everyday - ✗

Hyperproperty:

1. 'Hyperproperties', Clarkson and Schneider, 2010.

Trace Property vs Hyperproperty[1]



Trace property:

- I drink tea everyday - $\Diamond$ ☕ ✓
- I drink tea at the same time everyday - ✗

Hyperproperty:

I drink tea at the same time everyday

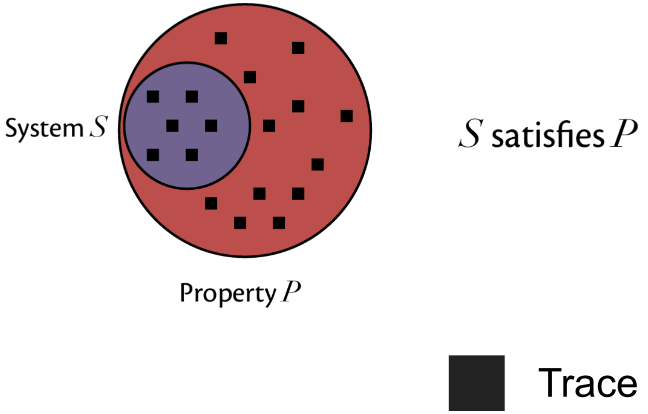
$\forall \pi . \forall \pi' . \Box$ (☕ $\leftrightarrow$ ☕) ✓

1. 'Hyperproperties', Clarkson and Schneider, 2010.

Trace Property vs Hyperproperty (contd.)

Trace Property vs Hyperproperty (contd.)

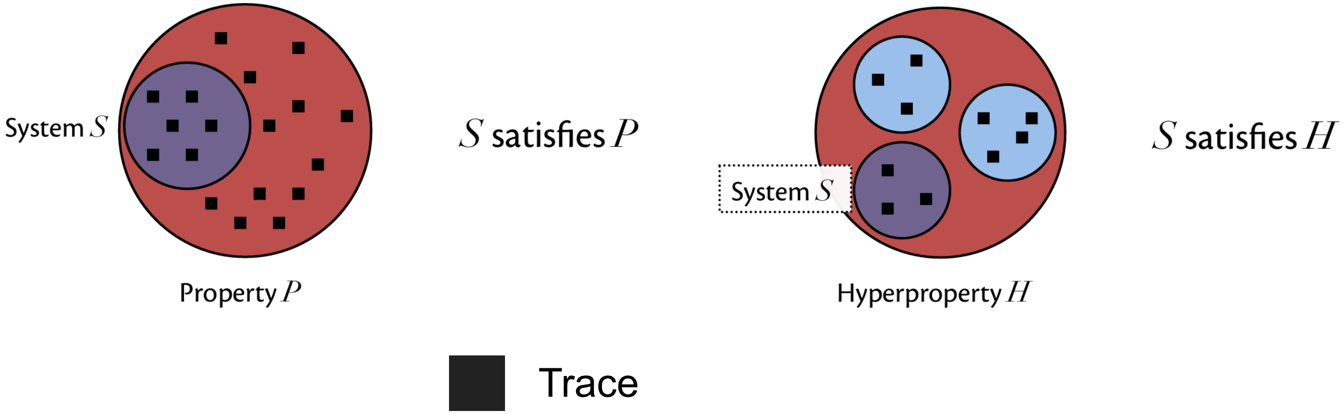
Fig. 1: Satisfaction of trace and hyper properties*



*Michael Clarkson's lecture notes on hyperproperties

Trace Property vs Hyperproperty (contd.)

Fig. 1: Satisfaction of trace and hyper properties*



*Michael Clarkson's lecture notes on hyperproperties

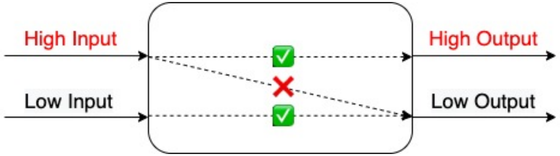
Hyperproperties[1,2] in Action

Hyperproperties[1,2] in Action

1. 'Hyperproperties', Clarkson and Schneider, 2010.
2. 'Temporal Logics for Hyperproperties', Clarkson, et al., POST 2014

Hyperproperties[1,2] in Action

Non-interference[3]:

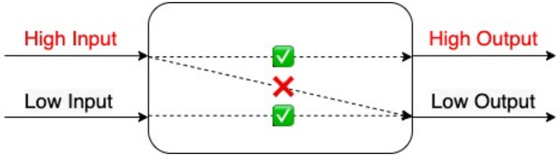


Should observe same low values for different high inputs

1. 'Hyperproperties', Clarkson and Schneider, 2010.
2. 'Temporal Logics for Hyperproperties', Clarkson, et al., POST 2014
3. 'Security policies and security models', Goguen and Meseguer.
4. 'Hyperproperties for Robotics', Wang, Nalluri, Pajic, ICRA 2020.

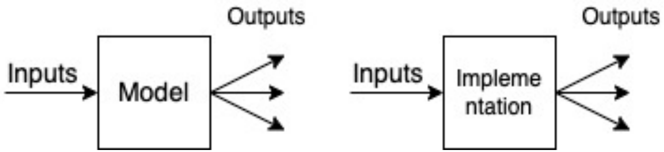
Hyperproperties[1,2] in Action

Non-interference[3]*:*



Should observe same low values for different high inputs

Conformance:

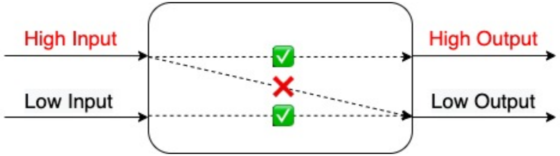


Should yield similar output after optimization / refactoring

1. 'Hyperproperties', Clarkson and Schneider, 2010.
2. 'Temporal Logics for Hyperproperties', Clarkson, et al., POST 2014
3. 'Security policies and security models', Goguen and Meseguer.
4. 'Hyperproperties for Robotics', Wang, Nalluri, Pajic, ICRA 2020.

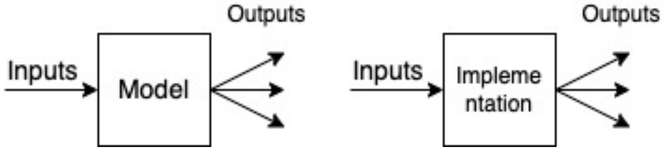
Hyperproperties[1,2] in Action

Non-interference[3]:



Should observe same low values for different high inputs

Conformance:



Should yield similar output after optimization / refactoring

Side-channel attacks:

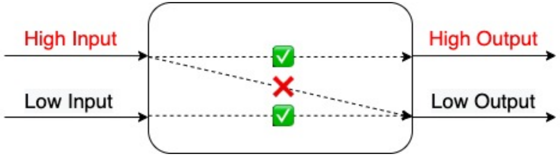
```
void mexp(){
   // b is secret
  c=0;
  if (b(i) = 1){
    // changes to c
  }
...
}
```

Should observe same execution times for different secrets

1. 'Hyperproperties', Clarkson and Schneider, 2010.
2. 'Temporal Logics for Hyperproperties', Clarkson, et al., POST 2014
3. 'Security policies and security models', Goguen and Meseguer.
4. 'Hyperproperties for Robotics', Wang, Nalluri, Pajic, ICRA 2020.

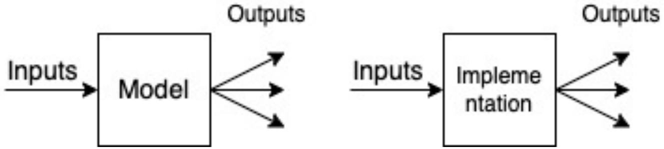
Hyperproperties[1,2] in Action

Non-interference[3]:



Should observe same low values for different high inputs

Conformance:



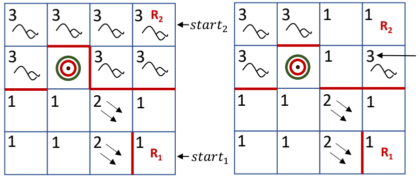
Should yield similar output after optimization / refactoring

Side-channel attacks:

```
void mexp(){
   // b is secret
  c=0;
  if (b(i) = 1){
    // changes to c
  }
  ...
}
```

Should observe same execution times for different secrets

Robotics path planning[4]:



Finding paths: shortest, robustness, opaqueness

1. 'Hyperproperties', Clarkson and Schneider, 2010.
2. 'Temporal Logics for Hyperproperties', Clarkson, et al., POST 2014
3. 'Security policies and security models', Goguen and Meseguer.
4. 'Hyperproperties for Robotics', Wang, Nalluri, Pajic, ICRA 2020.

Probabilistic Hyperproperties for DTMCs[1]

Probabilistic Hyperproperties for DTMCs[1]

1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol

1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol

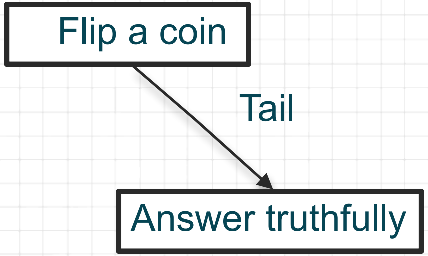
$\boxed{\text{Flip a coin}}$

1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol

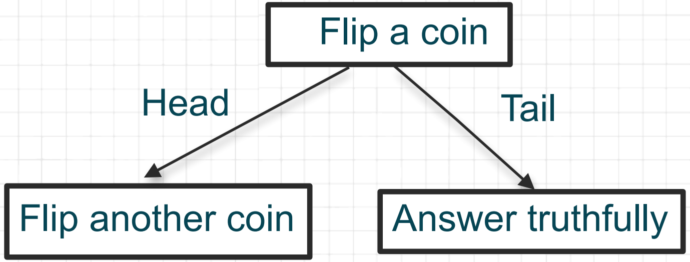


1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol

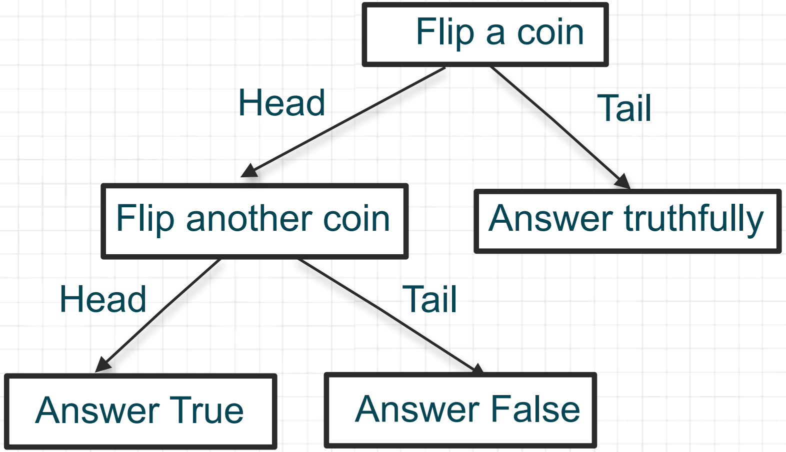


1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol

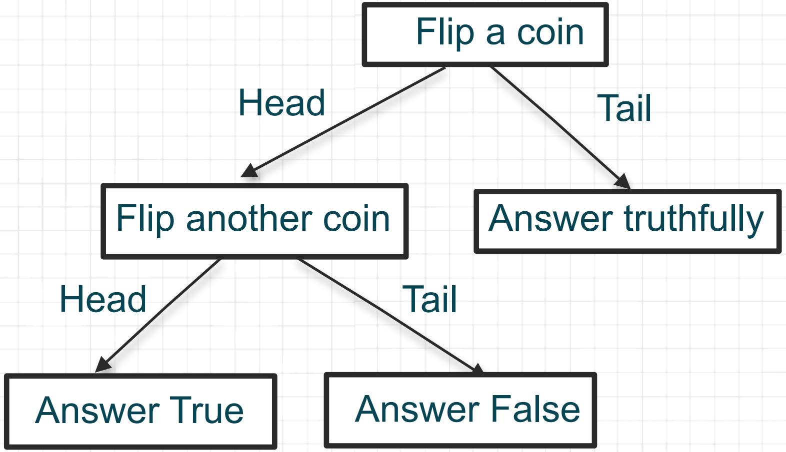


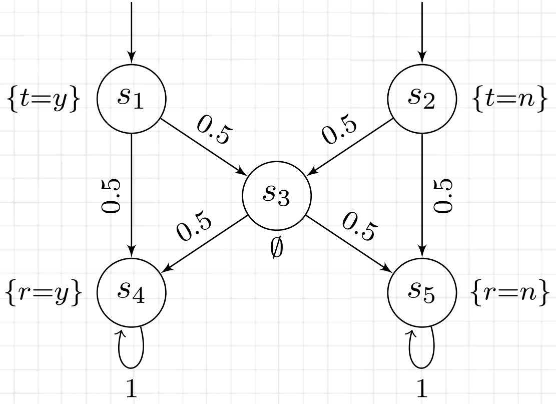
1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol



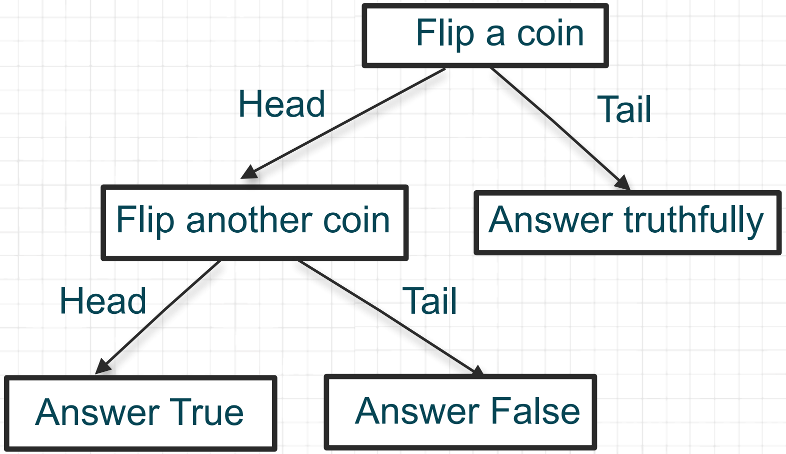


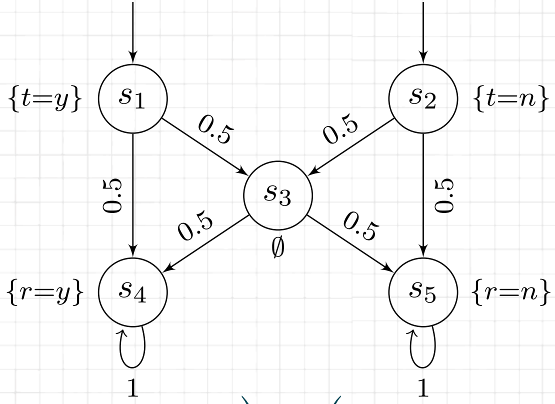
1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol





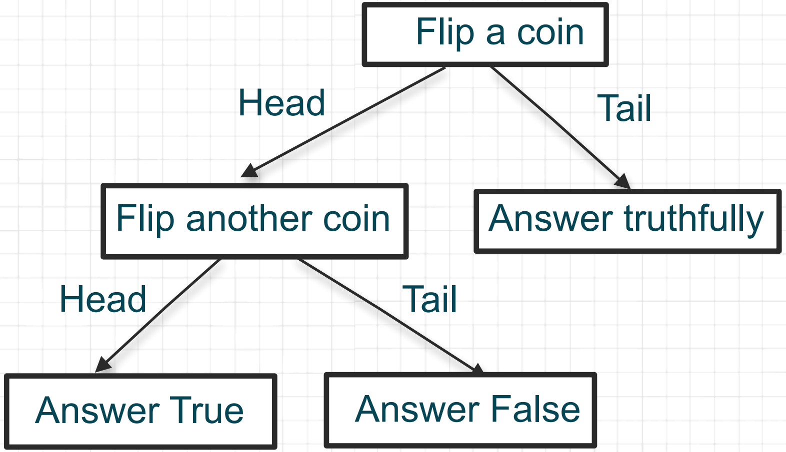
$$\psi = \forall s\,.\,\forall s'\,.\,\Big((\text{t=y})_s \wedge (\text{t=n})_{s'}\Big) \Rightarrow \Big(\mathbb{P}\,\Diamond(\text{r=y})_s \sim_\epsilon \mathbb{P}\Diamond(\text{r=y})_{s'}\Big)$$

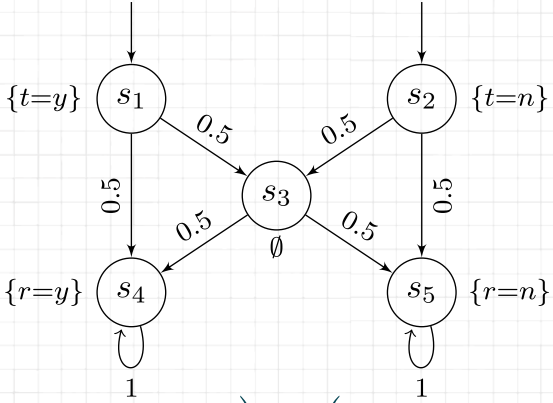
1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol





$$\psi = \forall s\,.\,\forall s'\,.\,\Big(\text{(t=y)}_s \wedge \text{(t=n)}_{s'}\Big) \Rightarrow \Big(\mathbb{P}\,\Diamond\text{(r=y)}_s \sim_\epsilon \mathbb{P}\Diamond\text{(r=y)}_{s'}\Big)$$

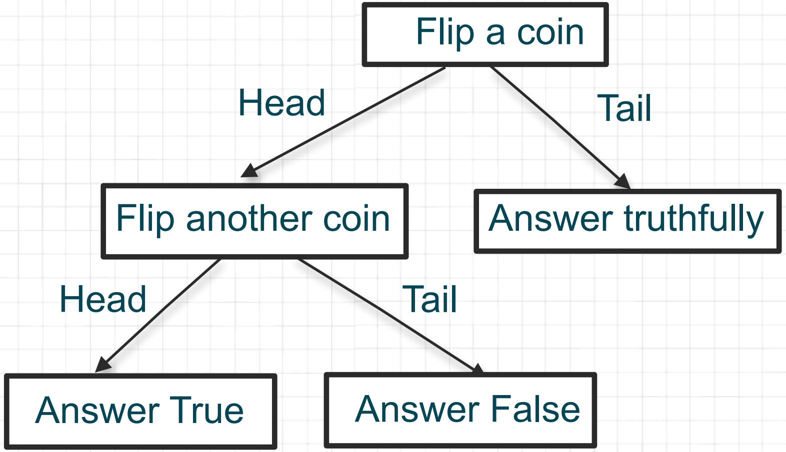
Quantification over initial states

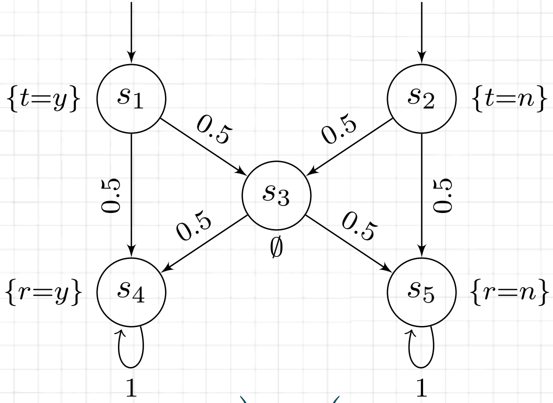
1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol





$$\psi = \forall s\,.\,\forall s'\,.\,\Big((\mathrm{t{=}y})_s \wedge (\mathrm{t{=}n})_{s'} \Big) \Rightarrow \Big(\mathbb{P}\,\Diamond(\mathrm{r{=}y})_s \sim_\epsilon \mathbb{P}\,\Diamond(\mathrm{r{=}y})_{s'} \Big)$$

Quantification over initial states

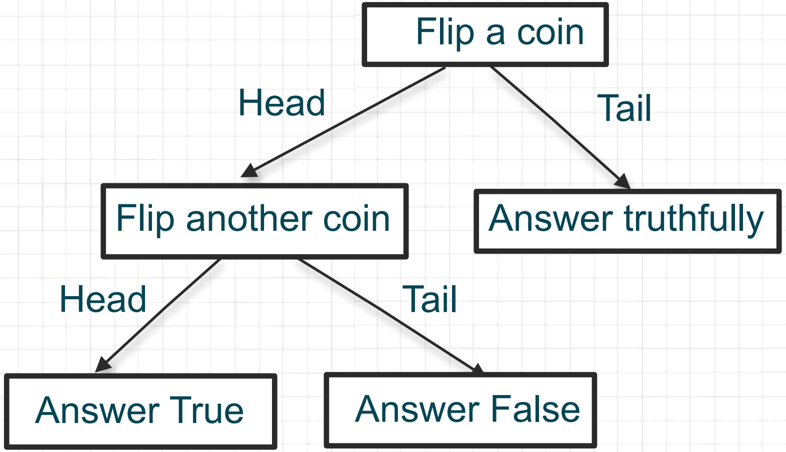
Index propositions

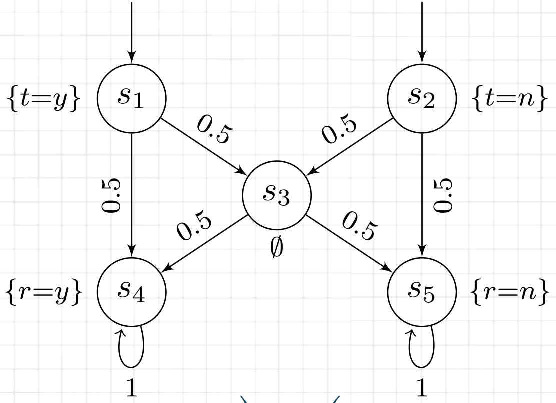
1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol





$$\psi = \underbrace{\forall s\,.\,\forall s'}_{\text{Quantification over initial states}}.\Big((t{=}y)_s \wedge (t{=}n)_{s'}\Big) \Rightarrow \Big(\underbrace{\mathbb{P}\,\Diamond(r{=}y)_s \sim_\epsilon \mathbb{P}\,\Diamond(r{=}y)_{s'}}_{\text{Temporal inside probability}}\Big)$$

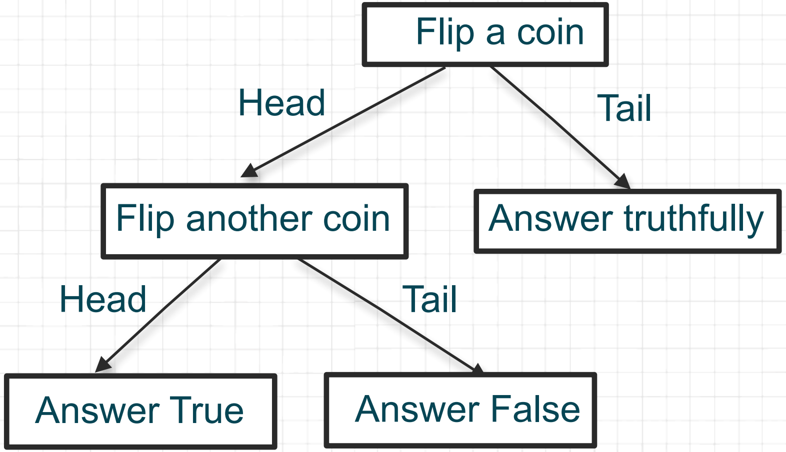
Index propositions

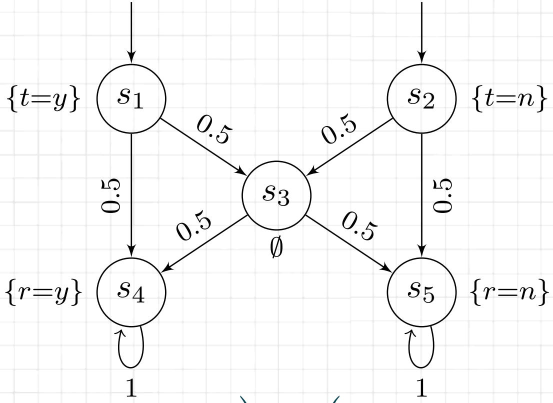
1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties for DTMCs[1]

- Motivation: Uncertainty and randomization.
- Probabilistic relation between traces.

Fig 2: A Differential Privacy protocol





$$\psi = \forall s . \forall s' . \left((t{=}y)_s \wedge (t{=}n)_{s'}\right) \Rightarrow \left(\mathbb{P}\,\Diamond(r{=}y)_s \sim_\epsilon \mathbb{P}\,\Diamond(r{=}y)_{s'}\right)$$

Quantification over initial states

Index propositions

Temporal inside probability

Cannot handle non-determinism!

1. 'HyperPCTL', Ábrahám and Bonakdarpour, QEST 2018.
2. 'The algorithmic foundations of differential privacy', Dwork and Roth.

Probabilistic Hyperproperties with Non-determinism[1,2]

Probabilistic Hyperproperties with Non-determinism[1,2]

1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.

1. ‘Probabilistic Hyperproperties of Markov Decision Processes’, Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. ‘Probabilistic Hyperproperties with Nondeterminism’, Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.

1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

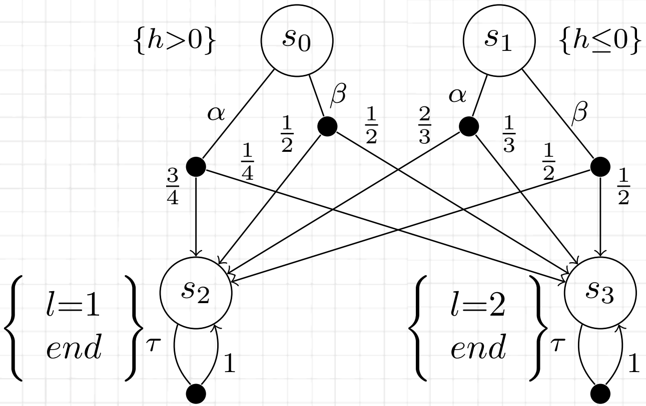
Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.

1. ‘Probabilistic Hyperproperties of Markov Decision Processes’, Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. ‘Probabilistic Hyperproperties with Nondeterminism’, Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

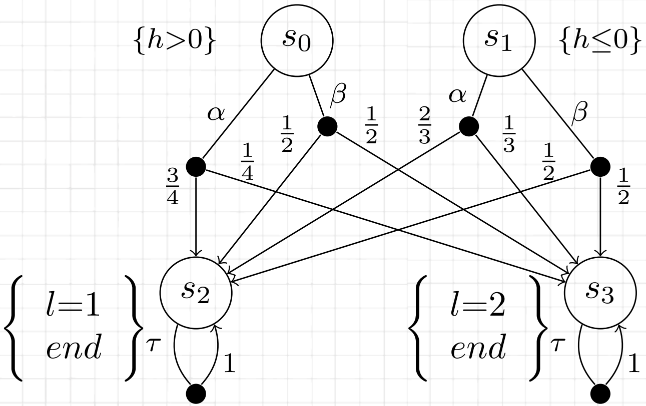
- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.



1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.

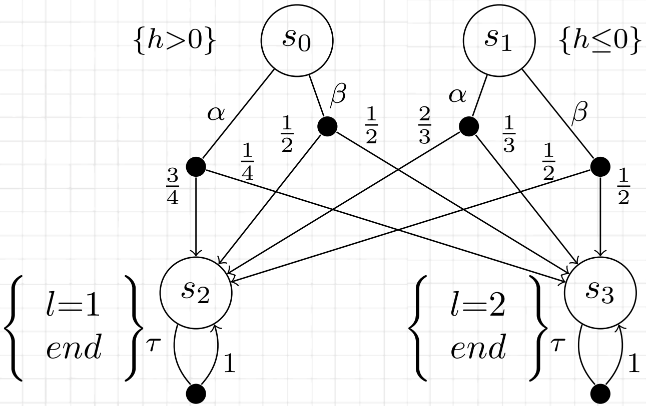


$$\exists sh\,.\,\forall s(sh)\,.\,\forall s'(sh)\,.\,\Big((h>0)_s \wedge (h\leq 0)_{s'}\Big) \Rightarrow \Big(\mathbb{P}\,\big(\Diamond(l=1)_s\big) = \mathbb{P}\,\big(\Diamond(l=1)_{s'}\big)\Big)$$

1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.



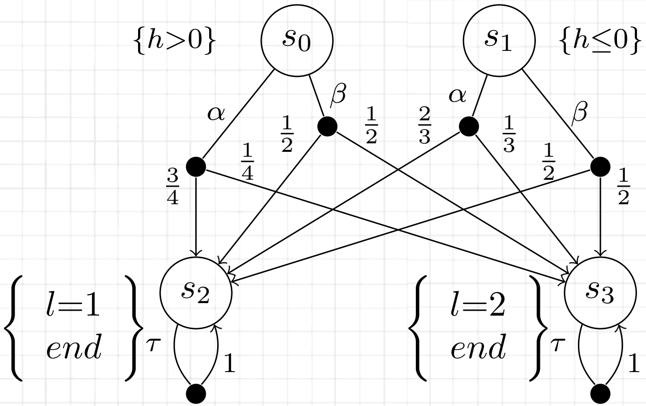
Scheduler quantifier

$$\exists sh \,.\, \forall s(sh) \,.\, \forall s'(sh) \,.\, \Big((h > 0)_s \wedge (h \leq 0)_{s'} \Big) \Rightarrow \Big(\mathbb{P}\left(\Diamond (l = 1)_s\right) = \mathbb{P}\left(\Diamond (l = 1)_{s'}\right) \Big)$$

1. ‘Probabilistic Hyperproperties of Markov Decision Processes’, Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. ‘Probabilistic Hyperproperties with Nondeterminism’, Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.



Scheduler quantifier

$$\exists sh . \forall s(sh) . \forall s'(sh) . \Big((h > 0)_s \wedge (h \leq 0)_{s'} \Big) \Rightarrow \Big(\mathbb{P}\left(\Diamond(l = 1)_s\right) = \mathbb{P}\left(\Diamond(l = 1)_{s'}\right) \Big)$$

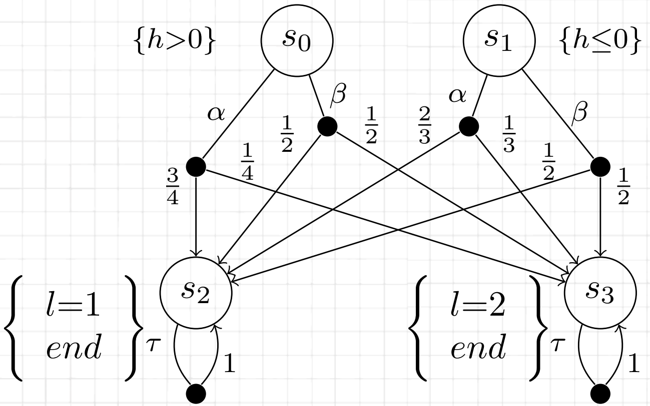
State quantifier scheduler

1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.

- Model checking problem for MDPs is undecidable.



Scheduler quantifier

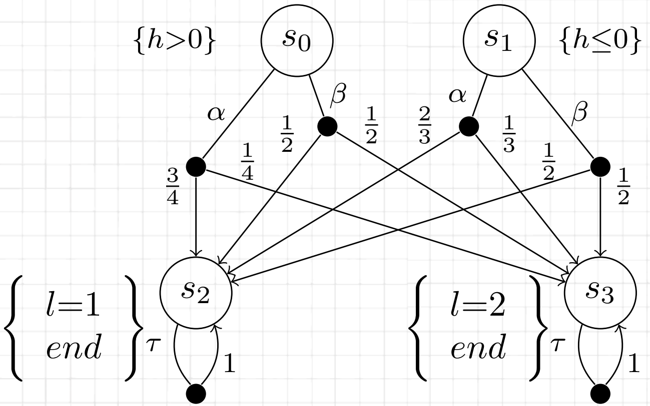
$$\exists sh\,.\,\forall s(sh)\,.\,\forall s'(sh)\,.\,\Big((h>0)_s \wedge (h\leq 0)_{s'}\Big) \Rightarrow \Big(\mathbb{P}\left(\Diamond(l=1)_s\right) = \mathbb{P}\left(\Diamond(l=1)_{s'}\right)\Big)$$

State quantifier scheduler

1. ‘Probabilistic Hyperproperties of Markov Decision Processes’, Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. ‘Probabilistic Hyperproperties with Nondeterminism’, Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.



- Model checking problem for MDPs is undecidable.
- In [2], restricted schedulers to memoryless and non-probabilistic.

Scheduler quantifier

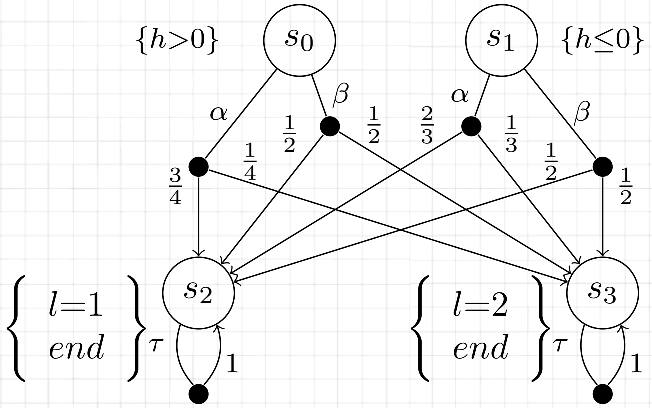
$$\exists sh\,.\,\forall s(sh)\,.\,\forall s'(sh)\,.\,\Big((h>0)_s \wedge (h\leq 0)_{s'}\Big) \Rightarrow \Big(\mathbb{P}\left(\Diamond(l=1)_s\right) = \mathbb{P}\left(\Diamond(l=1)_{s'}\right)\Big)$$

State quantifier scheduler

1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Probabilistic Hyperproperties with Non-determinism[1,2]

- Argues over combination of schedulers.
- [1] introduces PHL extending HyperCTL* and has path quantification.
- [2] extends HyperPCTL and has state quantification.



- Model checking problem for MDPs is undecidable.
- In [2], restricted schedulers to memoryless and non-probabilistic.

Does not support reward models!

Scheduler quantifier

$$\exists sh\,.\,\forall s(sh)\,.\,\forall s'(sh)\,.\,\Big((h>0)_s \wedge (h\leq 0)_{s'}\Big) \Rightarrow \Big(\mathbb{P}\left(\Diamond(l=1)_s\right) = \mathbb{P}\left(\Diamond(l=1)_{s'}\right)\Big)$$

State quantifier scheduler

1. 'Probabilistic Hyperproperties of Markov Decision Processes', Dimitrova, Finkbeiner, Torfah, ATVA 2020.
2. 'Probabilistic Hyperproperties with Nondeterminism', Ábrahám, Bartocci, Bonakdarpour, Dobe, ATVA 2020.

Applications

Side-Channel timing leaks

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c*2;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c+1;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Modeled as non-deterministic choice

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c+1;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Modeled as non-deterministic choice

Lack of code in else causes failure

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c+1;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Modeled as non-deterministic choice

Lack of code in else causes failure

$$\forall \hat{\sigma} \,.\, \forall \hat{s}(\hat{\sigma}) \,.\, \forall \hat{s}'(\hat{\sigma}) \,.\, \left(\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\ \mathsf{end}_{\hat{s}'})\right)$$

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c+1;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Modeled as non-deterministic choice

Lack of code in else causes failure

$$\forall \hat{\sigma} \,.\, \forall \hat{s}(\hat{\sigma}) \,.\, \forall \hat{s}'(\hat{\sigma}) \,.\, \left(\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\ \mathsf{end}_{\hat{s}'})\right)$$

Scheduler quantifier

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c-1;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Modeled as non-deterministic choice

$$\forall \hat{\sigma} \,.\, \underbrace{\forall \hat{s}(\hat{\sigma}) \,.\, \forall \hat{s}'(\hat{\sigma})}_{\text{For all different combination for 'b'}} \,.\, \Big(\mathbb{R}_{\hat{s}}(\Diamond\, \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\, \mathsf{end}_{\hat{s}'})\Big)$$

Scheduler quantifier

Lack of code in else causes failure

Side-Channel timing leaks

Figure: Snippet of Modular exponentiation in RSA.

```
void mexp(){
  c = 0; d = 1; i = k;
  while (i >= 0){
    i = i-1; c = c*2;
    d = (d*d) % n;
    if (b(i) = 1){
      c = c+1;
      d = (d*a) % n;
    }
  }
}
```

Lack of code in else causes failure

Modeled as non-deterministic choice

$$\forall \hat{\sigma} . \forall \hat{s}(\hat{\sigma}) . \forall \hat{s}'(\hat{\sigma}) . \left(\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\ \mathsf{end}_{\hat{s}'})\right)$$

Scheduler quantifier

For all different combination for ‘b’

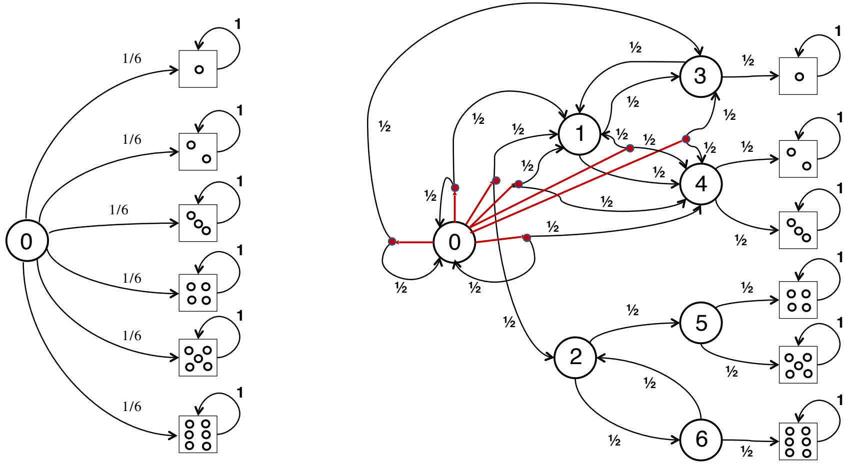
Ensure the code ends at the same time

Rewards helped us model time more efficiently

Probabilistic Conformance

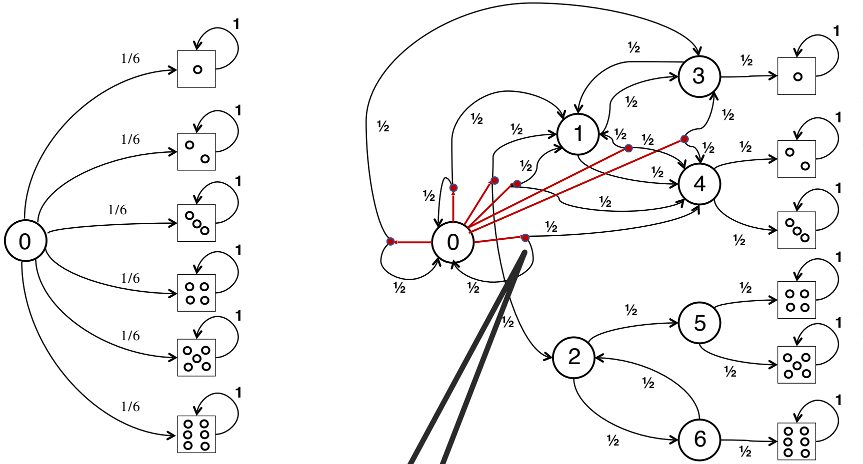
Probabilistic Conformance

Figure: (Left) Model of a fair 6-sided dice. (Right) Model of the Knuth-Yao algorithm to simulate the dice.



Probabilistic Conformance

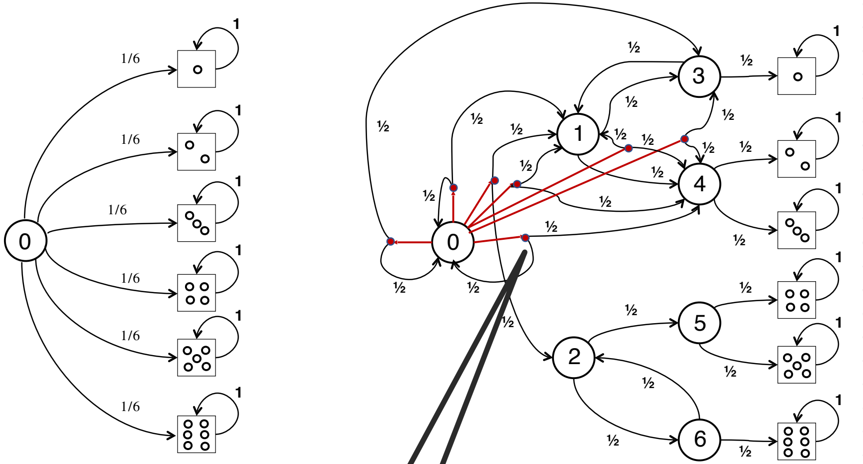
Figure: (Left) Model of a fair 6-sided dice. (Right) Model of the Knuth-Yao algorithm to simulate the dice.



Modeled as non-deterministic choices from each state

Probabilistic Conformance

Figure: (Left) Model of a fair 6-sided dice. (Right) Model of the Knuth-Yao algorithm to simulate the dice.



Modeled as non-deterministic choices from each state

$$\exists \hat{\sigma} . \forall \hat{s}(\hat{\sigma}) . \exists \hat{s}'(\hat{\sigma}) . \text{dieInit}_{\hat{s}} \rightarrow \left(\phi \wedge \mathbb{R}_{\hat{s}'}(F(\bigvee_{l=1}^{6} (die = l)_{\hat{s}'})) < 4 \right)$$

$$\phi = \text{coinInit}_{\hat{s}'} \wedge \bigwedge_{l=1}^{6} \left(\mathbb{P}(F(die = l)_{\hat{s}}) = \mathbb{P}(F(die = l)_{\hat{s}'}) \right)$$

Probabilistic Conformance

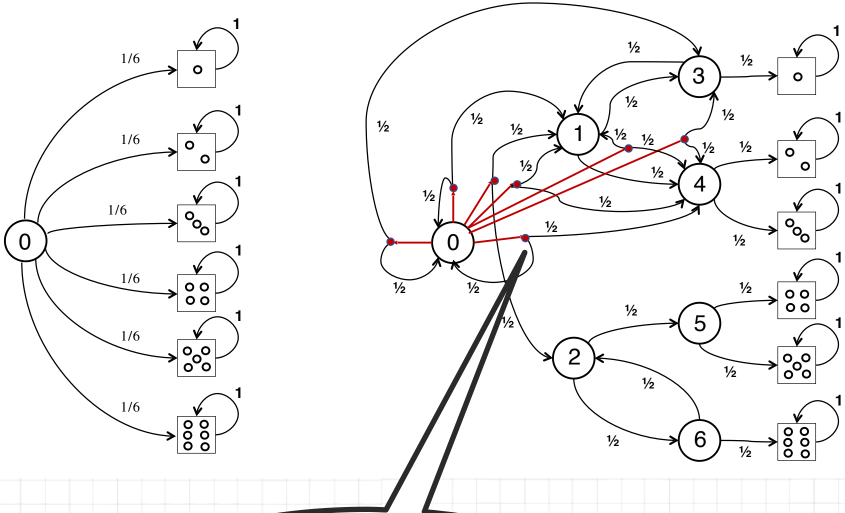


Figure: (Left) Model of a fair 6-sided dice. (Right) Model of the Knuth-Yao algorithm to simulate the dice.

Modeled as non-deterministic choices from each state

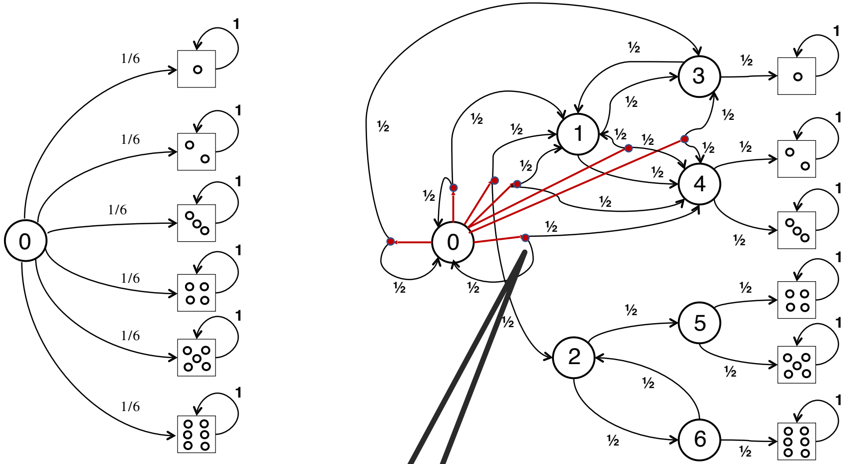
$$\exists \hat{\sigma} \,.\, \forall \hat{s}(\hat{\sigma}) \,.\, \exists \hat{s}'(\hat{\sigma}) \,.\, \text{dieInit}_{\hat{s}} \rightarrow \left(\phi \wedge \mathbb{R}_{\hat{s}'}(F(\bigvee_{l=1}^{6} (die = l)_{\hat{s}'})) < 4 \right)$$

$$\phi = \text{coinInit}_{\hat{s}'} \wedge \bigwedge_{l=1}^{6} \underbrace{\left(\mathbb{P}(F(die = l)_{\hat{s}}) = \mathbb{P}(F(die = l)_{\hat{s}'})\right)}_{}$$

Probability distribution of die faces should be the same

Probabilistic Conformance

Figure: (Left) Model of a fair 6-sided dice. (Right) Model of the Knuth-Yao algorithm to simulate the dice.



Modeled as non-deterministic choices from each state

Limiting solutions to within 4 tosses!

$$\exists \hat{\sigma} \,.\, \forall \hat{s}(\hat{\sigma}) \,.\, \exists \hat{s}'(\hat{\sigma}) \,.\, \mathsf{dieInit}_{\hat{s}} \rightarrow \left(\phi \wedge \overbrace{\mathbb{R}_{\hat{s}'}(F(\bigvee_{l=1}^{6} (die = l)_{\hat{s}'})) < 4}^{} \right)$$

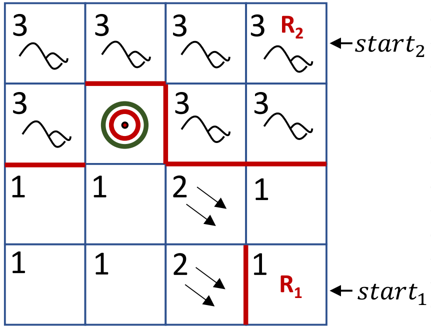
$$\phi = \mathsf{coinInit}_{\hat{s}'} \wedge \bigwedge_{l=1}^{6} \underbrace{\left(\mathbb{P}(F(die = l)_{\hat{s}}) = \mathbb{P}(F(die = l)_{\hat{s}'})\right)}_{}$$

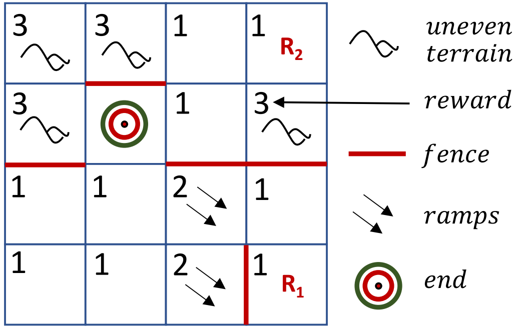
Probability distribution of die faces should be the same

Rewards helped us filter efficient solutions

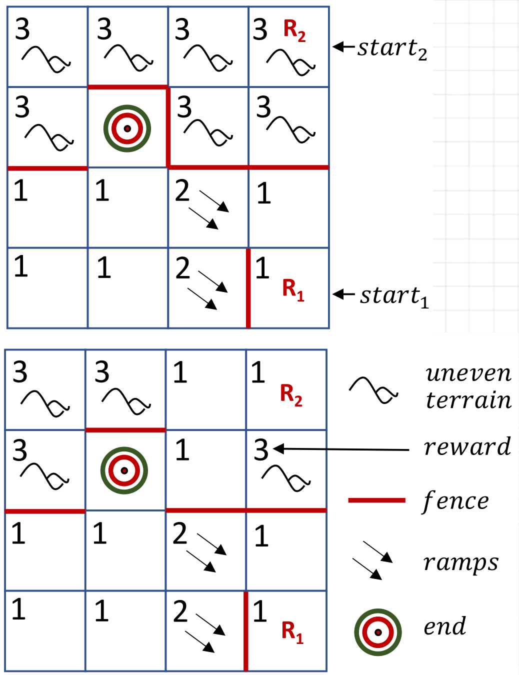
Multi-agent path planning

Multi-agent path planning





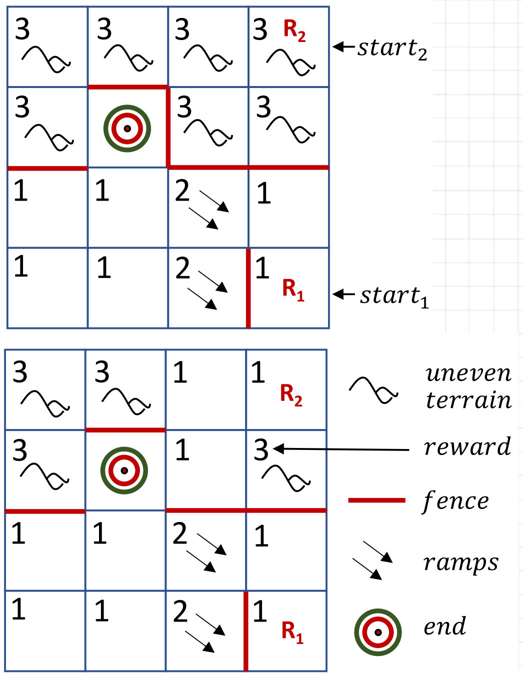
Multi-agent path planning



$$\varphi_{target} = \forall \hat{\sigma} . \forall \hat{s}(\hat{\sigma}) . \forall \hat{s}'(\hat{\sigma}) . \psi \rightarrow \left(\mathbb{R}_{\hat{s}}(\Diamond \mathsf{end}_{\hat{s}}) < \mathbb{R}_{\hat{s}'}(\Diamond \mathsf{end}_{\hat{s}'}) \right)$$

$$\psi = \left(\mathsf{start}_{1_{\hat{s}}} \wedge \mathsf{start}_{2_{\hat{s}'}} \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}}) = 1 \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}'}) = 1 \right)$$

Multi-agent path planning

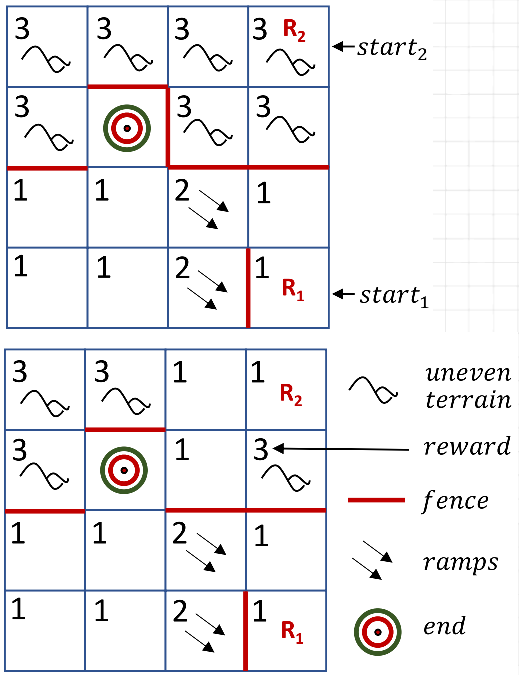


$$\varphi_{target} = \forall \hat{\sigma} . \forall \hat{s}(\hat{\sigma}) . \forall \hat{s}'(\hat{\sigma}) . \psi \rightarrow \left(\mathbb{R}_{\hat{s}}(\Diamond \mathsf{end}_{\hat{s}}) < \mathbb{R}_{\hat{s}'}(\Diamond \mathsf{end}_{\hat{s}'}) \right)$$

$$\psi = \left(\mathsf{start1}_{\hat{s}} \wedge \mathsf{start2}_{\hat{s}'} \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}}) = 1 \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}'}) = 1 \right)$$

Ensures both robots reach goal state

Multi-agent path planning



Robot R1 should spend less energy

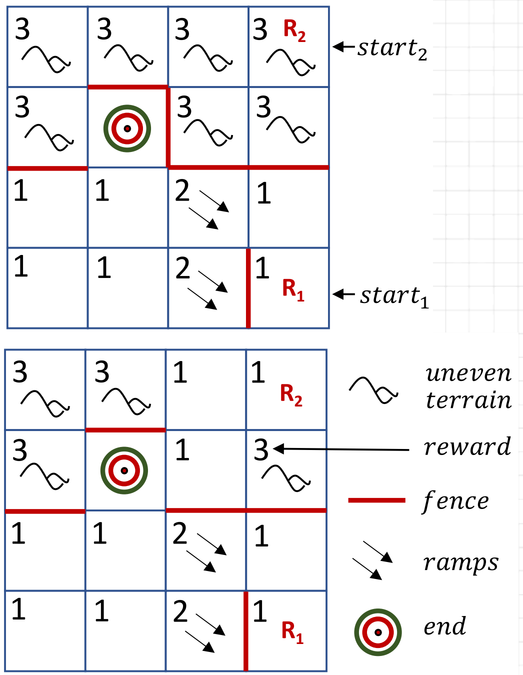
$$\varphi_{target} = \forall \hat{\sigma}\,.\,\forall \hat{s}(\hat{\sigma})\,.\,\forall \hat{s}'(\hat{\sigma})\,.\,\psi \rightarrow \left(\mathbb{R}_{\hat{s}}(\Diamond \mathsf{end}_{\hat{s}}) < \mathbb{R}_{\hat{s}'}(\Diamond \mathsf{end}_{\hat{s}'})\right)$$

$$\psi = \left(\mathsf{start1}_{\hat{s}} \wedge \mathsf{start2}_{\hat{s}'} \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}}) = 1 \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}'}) = 1\right)$$

Ensures both robots reach goal state

Multi-agent path planning

Figure: The maze on the top satisfies φ_{target}, while the bottom one violates it.



Robot R1 should spend less energy

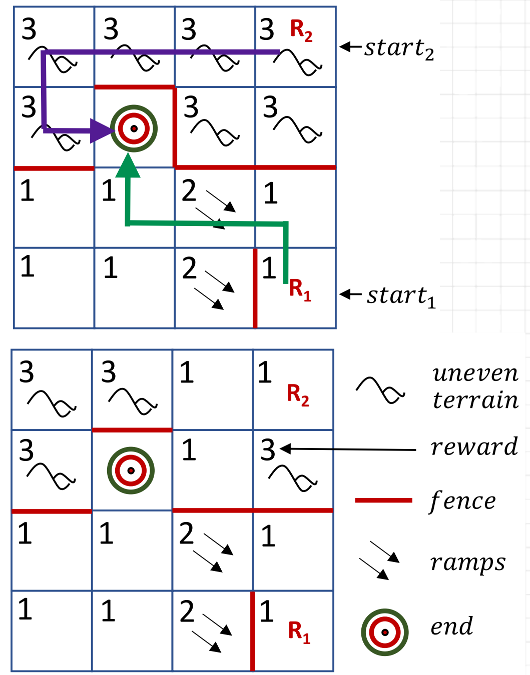
$$\varphi_{target} = \forall \hat{\sigma}\,.\,\forall \hat{s}(\hat{\sigma})\,.\,\forall \hat{s}'(\hat{\sigma})\,.\,\psi \rightarrow \Big(\mathbb{R}_{\hat{s}}(\Diamond \mathsf{end}_{\hat{s}}) < \mathbb{R}_{\hat{s}'}(\Diamond \mathsf{end}_{\hat{s}'})\Big)$$

$$\psi = \Big(\mathsf{start1}_{\hat{s}} \wedge \mathsf{start2}_{\hat{s}'} \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}}) = 1 \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}'}) = 1\Big)$$

Ensures both robots reach goal state

Multi-agent path planning

Figure: The maze on the top satisfies φ_{target}, while the bottom one violates it.



Robot R1 should spend less energy

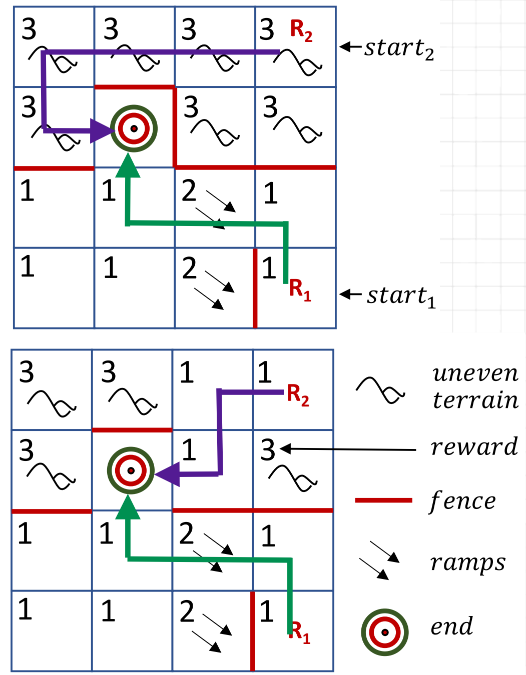
$$\varphi_{target} = \forall \hat{\sigma}\,.\,\forall \hat{s}(\hat{\sigma})\,.\,\forall \hat{s}'(\hat{\sigma})\,.\,\psi \rightarrow \left(\mathbb{R}_{\hat{s}}(\Diamond \mathsf{end}_{\hat{s}}) < \mathbb{R}_{\hat{s}'}(\Diamond \mathsf{end}_{\hat{s}'})\right)$$

$$\psi = \left(\mathsf{start1}_{\hat{s}} \wedge \mathsf{start2}_{\hat{s}'} \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}}) = 1 \wedge \mathbb{P}(\Diamond \mathsf{end}_{\hat{s}'}) = 1\right)$$

Ensures both robots reach goal state

Multi-agent path planning

Figure: The maze on the top satisfies φ_{target}, while the bottom one violates it.



$$\varphi_{target} = \forall \hat{\sigma} . \forall \hat{s}(\hat{\sigma}) . \forall \hat{s}'(\hat{\sigma}) . \psi \rightarrow \left(\mathbb{R}_{\hat{s}}(\Diamond \text{end}_{\hat{s}}) < \mathbb{R}_{\hat{s}'}(\Diamond \text{end}_{\hat{s}'}) \right)$$

$$\psi = \left(\text{start}_{1_{\hat{s}}} \wedge \text{start}_{2_{\hat{s}'}} \wedge \mathbb{P}(\Diamond \text{end}_{\hat{s}}) = 1 \wedge \mathbb{P}(\Diamond \text{end}_{\hat{s}'}) = 1 \right)$$

Robot R1 should spend less energy

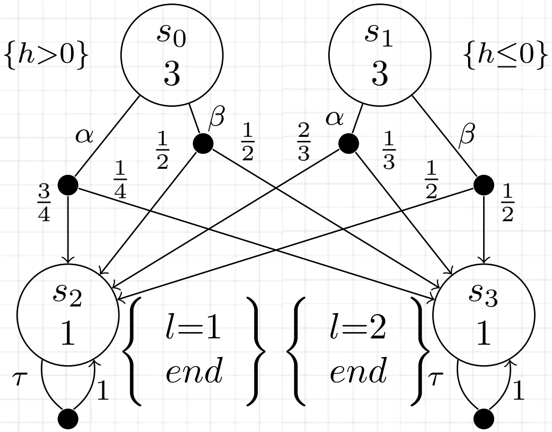
Ensures both robots reach goal state

Rewards helped us analyze cost of path planning

Methodology

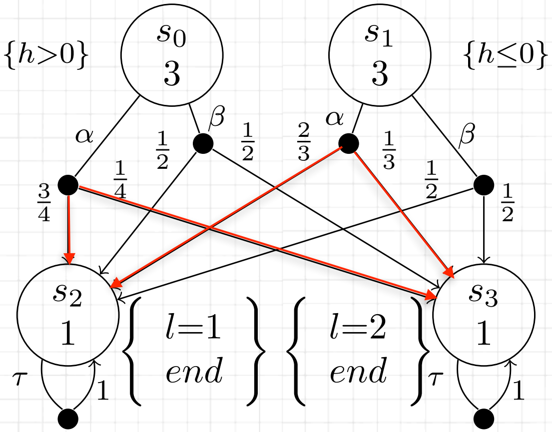
Probabilistic Hyperproperties with Rewards

Probabilistic Hyperproperties with Rewards



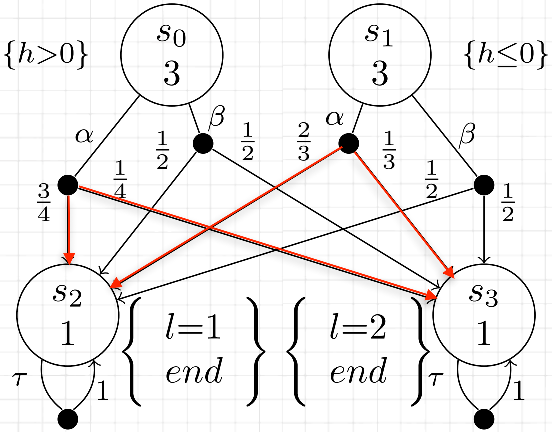
$$\exists \hat{\sigma}_1 \,.\, \exists \hat{\sigma}_2 \,.\, \forall \hat{s}(\hat{\sigma}_1) \,.\, \forall \hat{s}'(\hat{\sigma}_2) \,.\, \Big((h > 0)_{\hat{s}} \wedge (h \leq 0)_{\hat{s}'} \Big) \rightarrow$$
$$\Big(\mathbb{R}_{\hat{s}}(\Diamond \ \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond \ \mathsf{end}_{\hat{s}'}) \Big)$$

Probabilistic Hyperproperties with Rewards



$$\exists\hat{\sigma}_1 \,.\, \exists\hat{\sigma}_2 \,.\, \forall\hat{s}(\hat{\sigma}_1) \,.\, \forall\hat{s}'(\hat{\sigma}_2) \,.\, \Big((h > 0)_{\hat{s}} \land (h \leq 0)_{\hat{s}'}\Big) \rightarrow$$
$$\Big(\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\ \mathsf{end}_{\hat{s}'})\Big)$$

Probabilistic Hyperproperties with Rewards



$$\exists\hat{\sigma}_1 \,.\, \exists\hat{\sigma}_2 \,.\, \forall\hat{s}(\hat{\sigma}_1) \,.\, \forall\hat{s}'(\hat{\sigma}_2) \,.\, \Big((h > 0)_{\hat{s}} \wedge (h \leq 0)_{\hat{s}'}\Big) \rightarrow \Big(\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\ \mathsf{end}_{\hat{s}'})\Big)$$

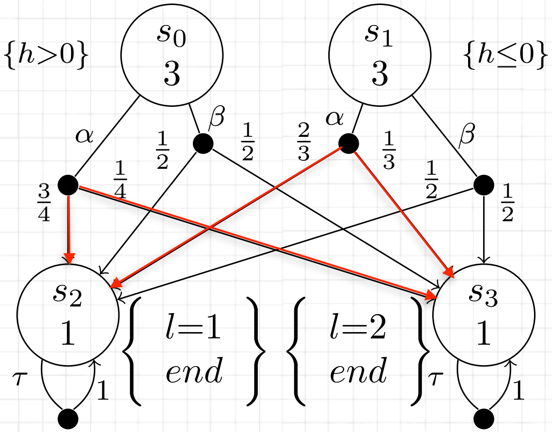
When $\hat{s} = s_0$

$$\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = 3 + \frac{3}{4} * 1 + \frac{1}{4} * 1 = 4$$

When $\hat{s}' = s_1$

$$\mathbb{R}_{\hat{s}}(\Diamond\ \mathsf{end}_{\hat{s}}) = 3 + \frac{2}{3} * 1 + \frac{1}{3} * 1 = 4$$

Probabilistic Hyperproperties with Rewards



- Different Cases:

$$\exists \hat{\sigma}_1 . \exists \hat{\sigma}_2 . \forall \hat{s}(\hat{\sigma}_1) . \forall \hat{s}'(\hat{\sigma}_2) . \Big((h > 0)_{\hat{s}} \wedge (h \leq 0)_{\hat{s}'} \Big) \rightarrow \Big(\mathbb{R}_{\hat{s}}(\Diamond \text{ end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond \text{ end}_{\hat{s}'}) \Big)$$

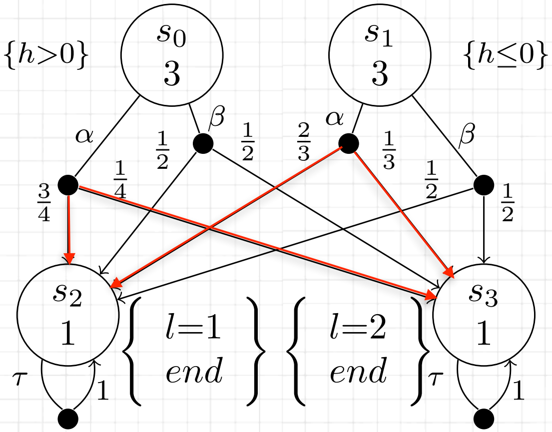
When $\hat{s} = s_0$

$$\mathbb{R}_{\hat{s}}(\Diamond \text{ end}_{\hat{s}}) = 3 + \frac{3}{4} * 1 + \frac{1}{4} * 1 = 4$$

When $\hat{s}' = s_1$

$$\mathbb{R}_{\hat{s}}(\Diamond \text{ end}_{\hat{s}}) = 3 + \frac{2}{3} * 1 + \frac{1}{3} * 1 = 4$$

Probabilistic Hyperproperties with Rewards



- Different Cases:
  - ✓ Compare rewards across computation trees.

$$\mathbb{R}_{\hat{s}}(F\ \text{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(F\ \text{end}_{\hat{s}'})$$

$$\exists\hat{\sigma}_1\,.\,\exists\hat{\sigma}_2\,.\,\forall\hat{s}(\hat{\sigma}_1)\,.\,\forall\hat{s}'(\hat{\sigma}_2)\,.\,\Big((h>0)_{\hat{s}} \wedge (h\leq 0)_{\hat{s}'}\Big) \rightarrow \Big(\mathbb{R}_{\hat{s}}(\Diamond\ \text{end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond\ \text{end}_{\hat{s}'})\Big)$$

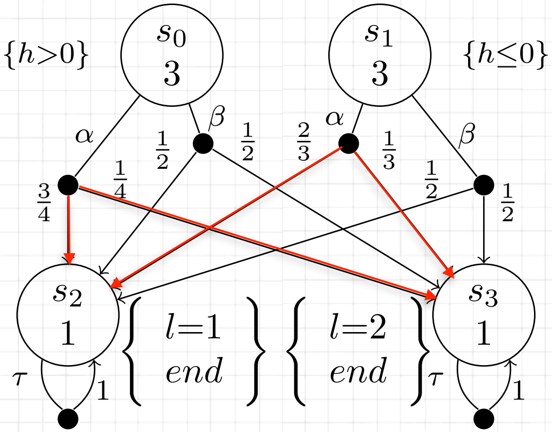
When $\hat{s} = s_0$

$$\mathbb{R}_{\hat{s}}(\Diamond\ \text{end}_{\hat{s}}) = 3 + \frac{3}{4} * 1 + \frac{1}{4} * 1 = 4$$

When $\hat{s}' = s_1$

$$\mathbb{R}_{\hat{s}}(\Diamond\ \text{end}_{\hat{s}}) = 3 + \frac{2}{3} * 1 + \frac{1}{3} * 1 = 4$$

Probabilistic Hyperproperties with Rewards



$$\exists\hat{\sigma}_1 . \exists\hat{\sigma}_2 . \forall\hat{s}(\hat{\sigma}_1) . \forall\hat{s}'(\hat{\sigma}_2) . \Big((h>0)_{\hat{s}} \wedge (h \leq 0)_{\hat{s}'}\Big) \rightarrow$$
$$\Big(\mathbb{R}_{\hat{s}}(\Diamond \text{ end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(\Diamond \text{ end}_{\hat{s}'})\Big)$$

- Different Cases:
  - ✓ Compare rewards across computation trees.

    $$\mathbb{R}_{\hat{s}}(F \text{ end}_{\hat{s}}) = \mathbb{R}_{\hat{s}'}(F \text{ end}_{\hat{s}'})$$

  - ✓ Compute rewards in one tree until we reach a state in another.

    $$\mathbb{R}_{\hat{s}}\ (\text{good}_{\hat{s}}\ U \text{ end}_{\hat{s}'}) < \ 4$$

When $\hat{s} = s_0$

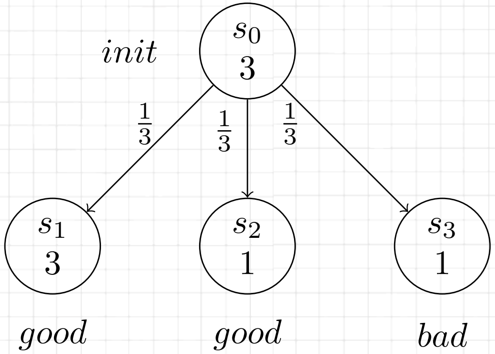
$$\mathbb{R}_{\hat{s}}(\Diamond \text{ end}_{\hat{s}}) = 3 + \frac{3}{4} * 1 + \frac{1}{4} * 1 = 4$$

When $\hat{s}' = s_1$

$$\mathbb{R}_{\hat{s}}(\Diamond \text{ end}_{\hat{s}}) = 3 + \frac{2}{3} * 1 + \frac{1}{3} * 1 = 4$$

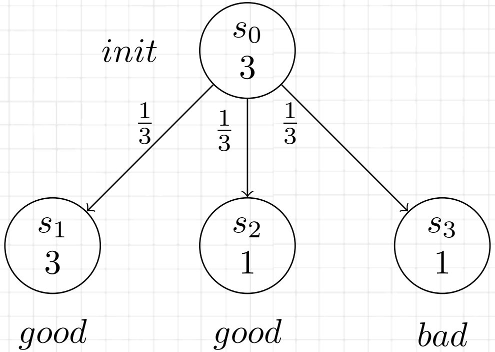
Undefinedness of Rewards

Undefinedness of Rewards



$\mathbb{R}(\bigcirc$ good$) = 3 + \frac{1}{3} * 3 + \frac{1}{3} * 1 + ? =$ undefined

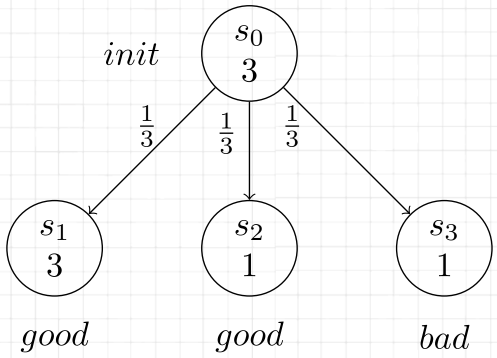
Undefinedness of Rewards



$$\mathbb{R}(\bigcirc\ \text{good}) = 3 + \frac{1}{3} * 3 + \frac{1}{3} * 1 + ? = \text{undefined}$$

Rewards is undefined if reachability probability is <1

Undefinedness of Rewards

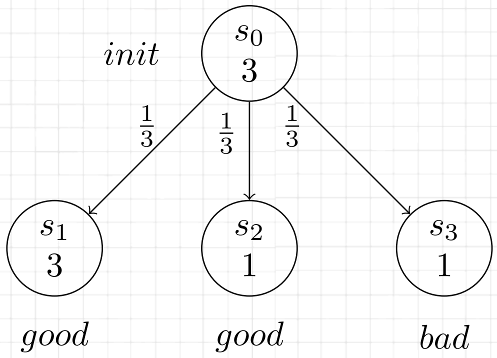


$$\mathbb{R}(\bigcirc\ \text{good}) = 3 + \frac{1}{3} * 3 + \frac{1}{3} * 1 + ? = \text{undefined}$$

Rewards is undefined if reachability probability is <1

- Challenge: Propagation of this undefinedness

Undefinedness of Rewards



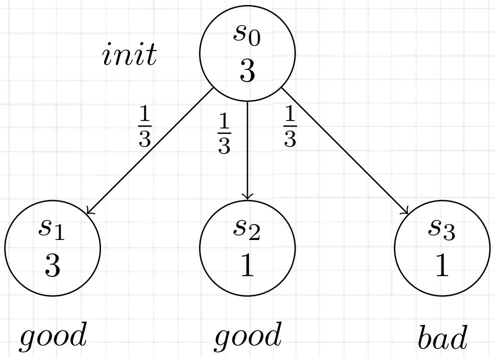
$$\mathbb{R}(\bigcirc\ \text{good}) = 3 + \frac{1}{3} * 3 + \frac{1}{3} * 1 + ? = \text{undefined}$$

Rewards is undefined if reachability probability is <1

- Challenge: Propagation of this undefinedness
- Easy Solution:

  Any component of a property undefined → overall result is undefined.

Undefinedness of Rewards



$\mathbb{R}(\bigcirc$ good$) = 3 + \frac{1}{3} * 3 + \frac{1}{3} * 1 + ? =$ undefined

Rewards is undefined if reachability probability is <1

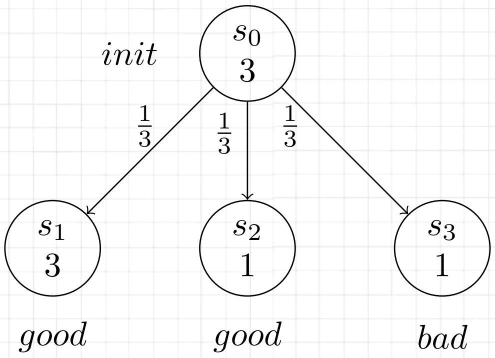
- Challenge: Propagation of this undefinedness
- Easy Solution:

  Any component of a property undefined → overall result is undefined.

- Our guiding concept:

  Overall definedness can be concluded from partial definedness of a property.

Undefinedness of Rewards



$$\mathbb{R}(\bigcirc\ \text{good}) = 3 + \frac{1}{3} * 3 + \frac{1}{3} * 1 + ? = \text{undefined}$$

$\boxed{\text{Rewards is undefined if reachability probability is <1}}$

- Challenge: Propagation of this undefinedness
- Easy Solution:

  Any component of a property undefined → overall result is undefined.

- Our guiding concept:

  Overall definedness can be concluded from partial definedness of a property.

false $\bigwedge$ undefined = false

undefined U false = false

Overview of Algorithm

Overview of Algorithm

Input MDP satisfies the HyperPCTL formula

iff

SMT encoding is satisfied

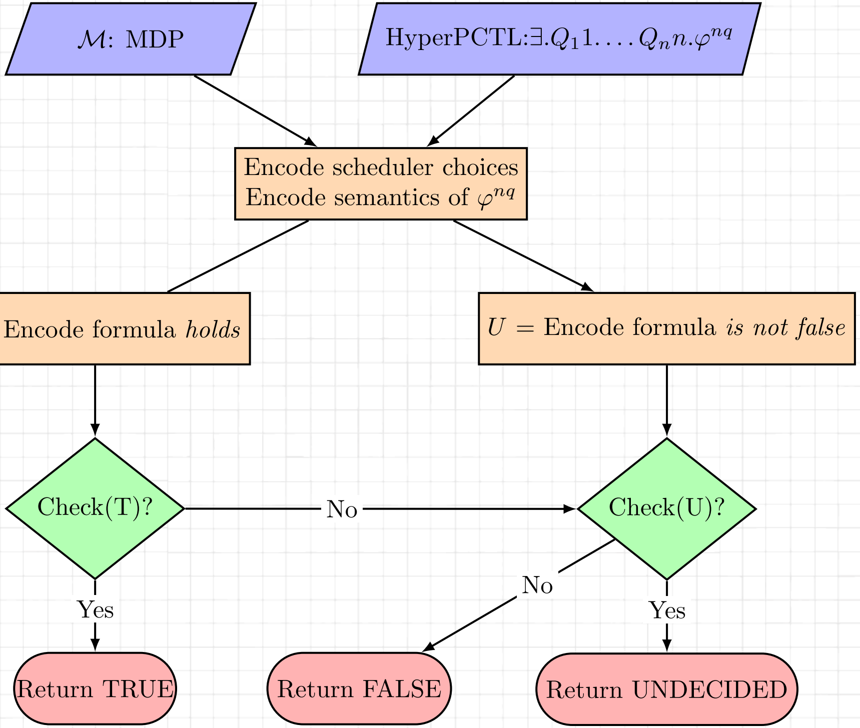
Overview of Algorithm

Input MDP satisfies the HyperPCTL formula

iff

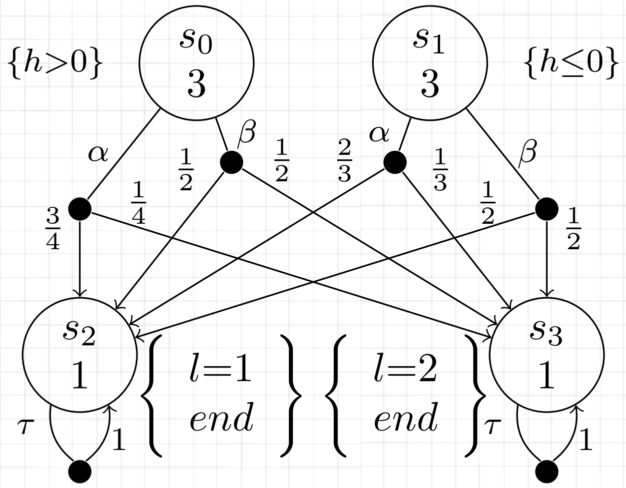
SMT encoding is satisfied

Algorithm for Existential Scheduler



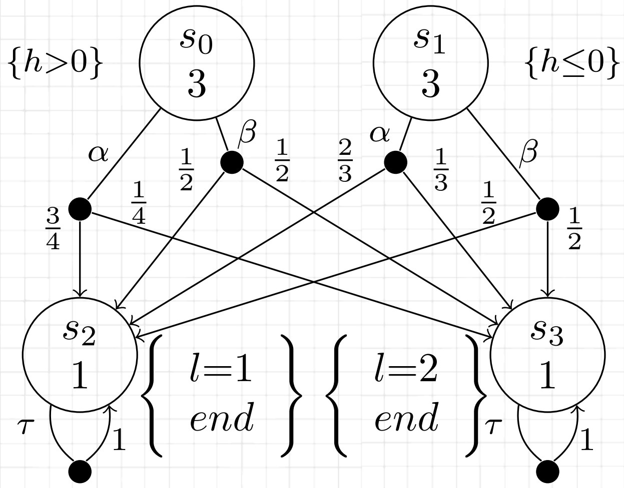
Example of Encoding

Example of Encoding



$$\varphi = \mathbb{R}_{\hat{s}}(\bigcirc\ \mathsf{end}_{\hat{s}})$$

Example of Encoding



$\varphi = \mathbb{R}_{\hat{s}}(\bigcirc\ \text{end}_{\hat{s}})$

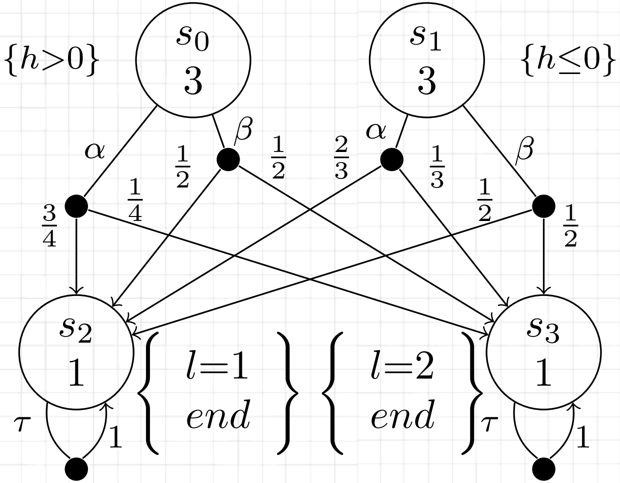
Encoding for φ **=** ***Encoding for*** $\mathbb{P}(\bigcirc\ \textit{end}_{\hat{s}}) \bigwedge$

$\left(\textit{val}_{s_0,\mathbb{P}(\bigcirc\ \textit{end}_{\hat{s}})} \neq 1 \vee \neg \textit{def}_{s_0,\mathbb{P}(\bigcirc\ \textit{end}_{\hat{s}})}\right) \leftrightarrow \neg \textit{def}_{s_0,\varphi} \bigwedge \ldots$

$\left[\textit{def}_{s_0,\varphi} \wedge \textit{act}_{s_0} = \alpha\right] \rightarrow \left[\textit{val}_{s_0,\varphi} = 3 + (\frac{3}{4} * 1 + \frac{1}{4} * 1)\right] \bigwedge$

$\left[\textit{def}_{s_0,\varphi} \wedge \textit{act}_{s_0} = \beta\right] \rightarrow \left[\textit{val}_{s_0,\varphi} = 3 + (\frac{1}{2} * 1 + \frac{1}{2} * 1)\right] \bigwedge \ldots$

Example of Encoding



$$\varphi = \mathbb{R}_{\hat{s}}(\bigcirc\ \text{end}_{\hat{s}})$$

Find probability of the property

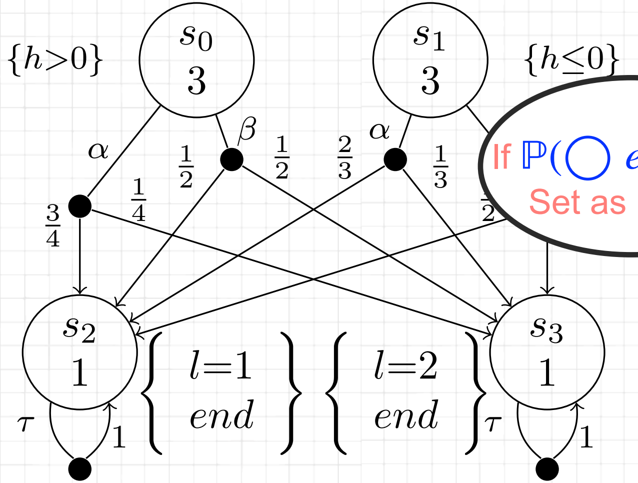
Encoding for φ **=** $\boxed{\textbf{\textit{Encoding for }} \mathbb{P}(\bigcirc\ \textit{end}_{\hat{s}})}\ \bigwedge$

$$\left(\textit{val}_{s_0,\mathbb{P}(\bigcirc\ \textit{end}_{\hat{s}})} \neq 1 \vee \neg \textit{def}_{s_0,\mathbb{P}(\bigcirc\ \textit{end}_{\hat{s}})}\right) \leftrightarrow \neg \textit{def}_{s_0,\varphi} \bigwedge \ldots$$

$$\left[\textit{def}_{s_0,\varphi} \wedge \textit{act}_{s_0} = \alpha\right] \rightarrow \left[\textit{val}_{s_0,\varphi} = 3 + \left(\frac{3}{4} * 1 + \frac{1}{4} * 1\right)\right] \bigwedge$$

$$\left[\textit{def}_{s_0,\varphi} \wedge \textit{act}_{s_0} = \beta\right] \rightarrow \left[\textit{val}_{s_0,\varphi} = 3 + \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right)\right] \bigwedge \ldots$$

Example of Encoding



$$\varphi = \mathbb{R}_{\hat{s}}(\bigcirc\ \text{end}_{\hat{s}})$$

Find probability of the property

If $\mathbb{P}(\bigcirc\ end_{\hat{s}})<1$
Set as undef

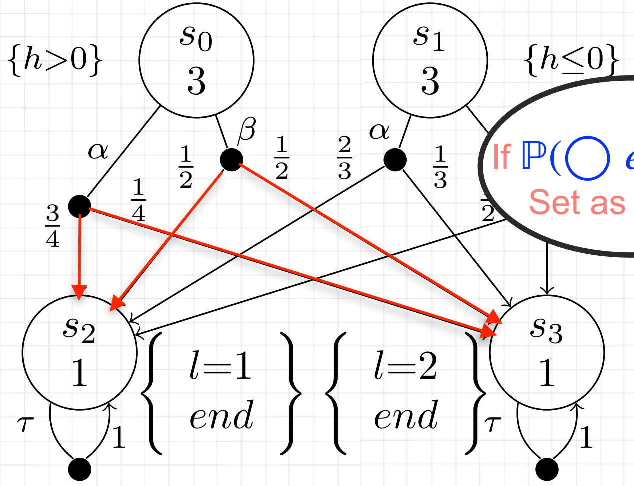
Encoding for φ = $\boxed{\textbf{\textit{Encoding for }} \mathbb{P}(\bigcirc\ \textbf{\textit{end}}_{\hat{s}})} \bigwedge$

$$\boxed{\textbf{\textit{val}}_{s_0,\mathbb{P}(\bigcirc\ \textbf{\textit{end}}_{\hat{s}})} \neq 1 \vee \neg \textbf{\textit{def}}_{s_0,\mathbb{P}(\bigcirc\ \textbf{\textit{end}}_{\hat{s}})}) \leftrightarrow \neg \textbf{\textit{def}}_{s_0,\varphi}} \bigwedge \ldots$$

$$[\textbf{\textit{def}}_{s_0,\varphi} \wedge \textbf{\textit{act}}_{s_0} = \alpha] \rightarrow [\textbf{\textit{val}}_{s_0,\varphi} = 3 + (\frac{3}{4} * 1 + \frac{1}{4} * 1)] \bigwedge$$

$$[\textbf{\textit{def}}_{s_0,\varphi} \wedge \textbf{\textit{act}}_{s_0} = \beta] \rightarrow [\textbf{\textit{val}}_{s_0,\varphi} = 3 + (\frac{1}{2} * 1 + \frac{1}{2} * 1)] \bigwedge \ldots$$

Example of Encoding



$\varphi = \mathbb{R}_{\hat{s}}(\bigcirc\ \text{end}_{\hat{s}})$

Find probability of the property

If $\mathbb{P}(\bigcirc\ end_{\hat{s}})$<1
Set as undef

Encoding for φ **=** $\boxed{\textbf{\textit{Encoding for }} \mathbb{P}(\bigcirc\ \textbf{\textit{end}}_{\hat{s}})}\ \bigwedge$

$\boxed{\textbf{val}_{s_0,\mathbb{P}(\bigcirc\ \textbf{end}_{\hat{s}})} \neq 1 \vee \neg\textbf{def}_{s_0,\mathbb{P}(\bigcirc\ \textbf{end}_{\hat{s}})}) \leftrightarrow \neg\textbf{def}_{s_0,\varphi}}\ \bigwedge \ldots$

$$\left[\textbf{def}_{s_0,\varphi} \wedge \textbf{act}_{s_0} = \alpha\right] \rightarrow \left[\textbf{val}_{s_0,\varphi} = 3 + (\frac{3}{4} * 1 + \frac{1}{4} * 1)\right] \bigwedge$$

$$\left[\textbf{def}_{s_0,\varphi} \wedge \textbf{act}_{s_0} = \beta\right] \rightarrow \left[\textbf{val}_{s_0,\varphi} = 3 + (\frac{1}{2} * 1 + \frac{1}{2} * 1)\right] \bigwedge \ldots$$

Define case for each action and its transitions

Evaluation

Evaluation

Case study		VR	Running time (s) Encoding	Solving	Total	#SMT variables	#states	#transitions
TA	1-bit key	×	0.11	0.01	0.12	344	8	10
	16-bit key	×	16.41	3.69	20.10	19244	68	100
	30-bit key	×	143.49	44.64	188.13	62868	124	184
	45-bit key	×	774.53	1304.98	2079.51	137448	184	274
PC	s=(0)	✓	5.03	2.03	7.06	7281	20	186
	s=(0,1,2)	✓	6.66	8.91	15.57	7281	20	494
	s=(0,. . .,4)	✓	8.82	35	43.82	7281	20	802
	s=(0,. . .,6)	✓	11.64	53.05	64.69	7281	20	1110
RO	3x3	✓	0.87	0.05	0.92	2179	18	66
	3x3	×	0.93	0.05	0.98	2179	18	66
	4x4	✓	3.55	0.28	3.83	6561	32	160
	4x4	×	3.43	0.25	3.68	6561	32	148
	5x5	✓	13.07	0.5	13.57	15651	50	250
	5x5	×	13.19	0.98	14.17	15651	50	250
	6x6	✓	44.52	1.04	45.56	32041	72	398
	6x6	×	44.65	7.48	52.13	32041	72	398
HS	$n = 3$	✓	0.1	0.01	0.11	489	8	28
	$n = 5$	✓	0.95	0.13	1.08	2369	32	244
IJ	$n = 3$	✓	0.08	0.01	0.09	169	7	21
	$n = 4$	✓	0.24	0.04	0.28	601	15	56
	$n = 5$	✓	0.89	0.33	1.22	2233	31	140
	$n = 6$	✓	3.93	19.39	23.32	8569	63	336

Experimental results: VR: Verification result. TA: Timing attack. PC: Probabilistic conformance. RO: Robotics example. HS: Herman's algorithm. IJ: Israeli-Jaflon's algorithm. ✓: the result is true. ×: the result is false.

Evaluation

Case study		VR	Running time (s) Encoding	Solving	Total	#SMT variables	#states	#transitions
TA	1-bit key	×	0.11	0.01	0.12	344	8	10
	16-bit key	×	16.41	3.69	20.10	19244	68	100
	30-bit key	×	143.49	44.64	188.13	62868	124	184
	45-bit key	×	774.53	1304.98	2079.51	137448	184	274
PC	s=(0)	✓	5.03	2.03	7.06	7281	20	186
	s=(0,1,2)	✓	6.66	8.91	15.57	7281	20	494
	s=(0,...,4)	✓	8.82	35	43.82	7281	20	802
	s=(0,...,6)	✓	11.64	53.05	64.69	7281	20	1110
RO	3x3	✓	0.87	0.05	0.92	2179	18	66
	3x3	×	0.93	0.05	0.98	2179	18	66
	4x4	✓	3.55	0.28	3.83	6561	32	160
	4x4	×	3.43	0.25	3.68	6561	32	148
	5x5	✓	13.07	0.5	13.57	15651	50	250
	5x5	×	13.19	0.98	14.17	15651	50	250
	6x6	✓	44.52	1.04	45.56	32041	72	398
	6x6	×	44.65	7.48	52.13	32041	72	398
HS	$n = 3$	✓	0.1	0.01	0.11	489	8	28
	$n = 5$	✓	0.95	0.13	1.08	2369	32	244
IJ	$n = 3$	✓	0.08	0.01	0.09	169	7	21
	$n = 4$	✓	0.24	0.04	0.28	601	15	56
	$n = 5$	✓	0.89	0.33	1.22	2233	31	140
	$n = 6$	✓	3.93	19.39	23.32	8569	63	336

Experimental results: VR: Verification result. TA: Timing attack. PC: Probabilistic conformance. RO: Robotics example. HS: Herman's algorithm. IJ: Israeli-Jaflon's algorithm. ✓: the result is true. ×: the result is false.

Evaluation

Case study		VR	Running time (s) Encoding	Solving	Total	#SMT variables	#states	#transitions
TA	1-bit key	×	0.11	0.01	0.12	344	8	10
	16-bit key	×	16.41	3.69	20.10	19244	68	100
	30-bit key	×	143.49	44.64	188.13	62868	124	184
	45-bit key	×	774.53	1304.98	2079.51	137448	184	274
PC	s=(0)	✓	5.03	2.03	7.06	7281	20	186
	s=(0,1,2)	✓	6.66	8.91	15.57	7281	20	494
	s=(0,...,4)	✓	8.82	35	43.82	7281	20	802
	s=(0,...,6)	✓	11.64	53.05	64.69	7281	20	1110
RO	3x3	✓	0.87	0.05	0.92	2179	18	66
	3x3	×	0.93	0.05	0.98	2179	18	66
	4x4	✓	3.55	0.28	3.83	6561	32	160
	4x4	×	3.43	0.25	3.68	6561	32	148
	5x5	✓	13.07	0.5	13.57	15651	50	250
	5x5	×	13.19	0.98	14.17	15651	50	250
	6x6	✓	44.52	1.04	45.56	32041	72	398
	6x6	×	44.65	7.48	52.13	32041	72	398
HS	$n = 3$	✓	0.1	0.01	0.11	489	8	28
	$n = 5$	✓	0.95	0.13	1.08	2369	32	244
IJ	$n = 3$	✓	0.08	0.01	0.09	169	7	21
	$n = 4$	✓	0.24	0.04	0.28	601	15	56
	$n = 5$	✓	0.89	0.33	1.22	2233	31	140
	$n = 6$	✓	3.93	19.39	23.32	8569	63	336

Experimental results: VR: Verification result. TA: Timing attack. PC: Probabilistic conformance. RO: Robotics example. HS: Herman's algorithm. IJ: Israeli-Jaflon's algorithm. ✓: the result is true. ×: the result is false.

Conclusion

Conclusion

Logic

Extended HyperPCTL to express reward operators

Conclusion

Logic

Extended HyperPCTL to express reward operators

Algorithm

Provided algorithms to evaluate state-based reward operators

Conclusion

Logic

Extended HyperPCTL to express reward operators

Algorithm

Provided algorithms to evaluate state-based reward operators

Implementation

Extended our tool HyperPROB[1] to accommodate restricted rewards.

1. 'HYPERPROB: A Model Checker for Probabilistic Hyperproperties', Dobe, Ábrahám, Bartocci, Bonakdarpour, FM 2021.