Synthesis of Safe and Optimal Strategies

for Cyber-Physical Systems

Kim G Larsen

w Peter Gjøl Jensen

Aalborg University, DENMARK

AALBORG UNIVERSITET

erc

VILLUM FONDEN

UPPAAL 4.0

Stratego

Collaborators

- Alexandre David
- Marius Mikucionis
- Peter Jensen
- Jakob Taankvist
- Manfred Jaeger
- Axel Legay
- Sean Sedwards
- Pranav Ashok
- Jan Kretinsky
- Adrien Le Coënt
- Maximilian Weininger
- Florian Lorber
- Giorgio Bacci
- Giovanni Bacci

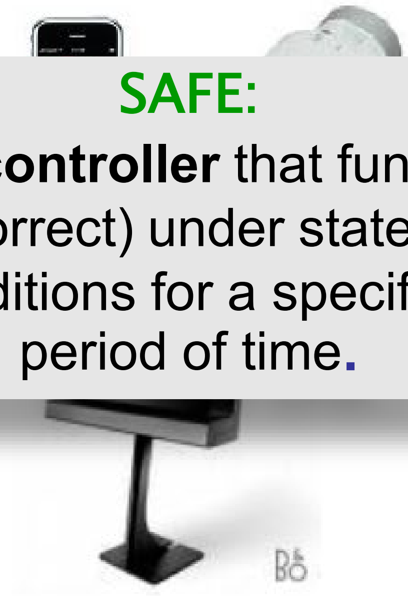
[ATVA19] Teaching Stratego to play Ball

[QEST19] SOS: Synthesis for Hybrid MDP

[ISOLA20] Approximating Euclidain by Imprecise MDPs

[ADHS21] Learning Safe and Optimal Control m Strategies for Storm Water Detention Ponds

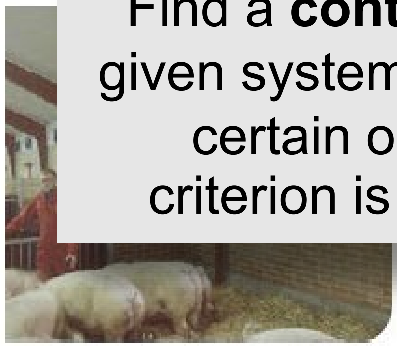
[ICAPS22] Distributed Fleet Management in Noisy Environments via Model-Predictive Control



SAFE:
Find **controller** that function (correct) under stated conditions for a specified period of time.

Discrete

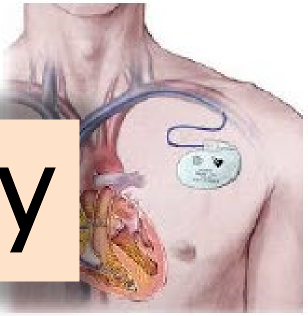
Real Time



OPTIMAL:
Find a **controller** for a given system such that a certain optimality criterion is achieved.

Resources

Stochasticity



COMPACT:
Find a representation of the **controller** that may fit a given embedded platform

Hybrid

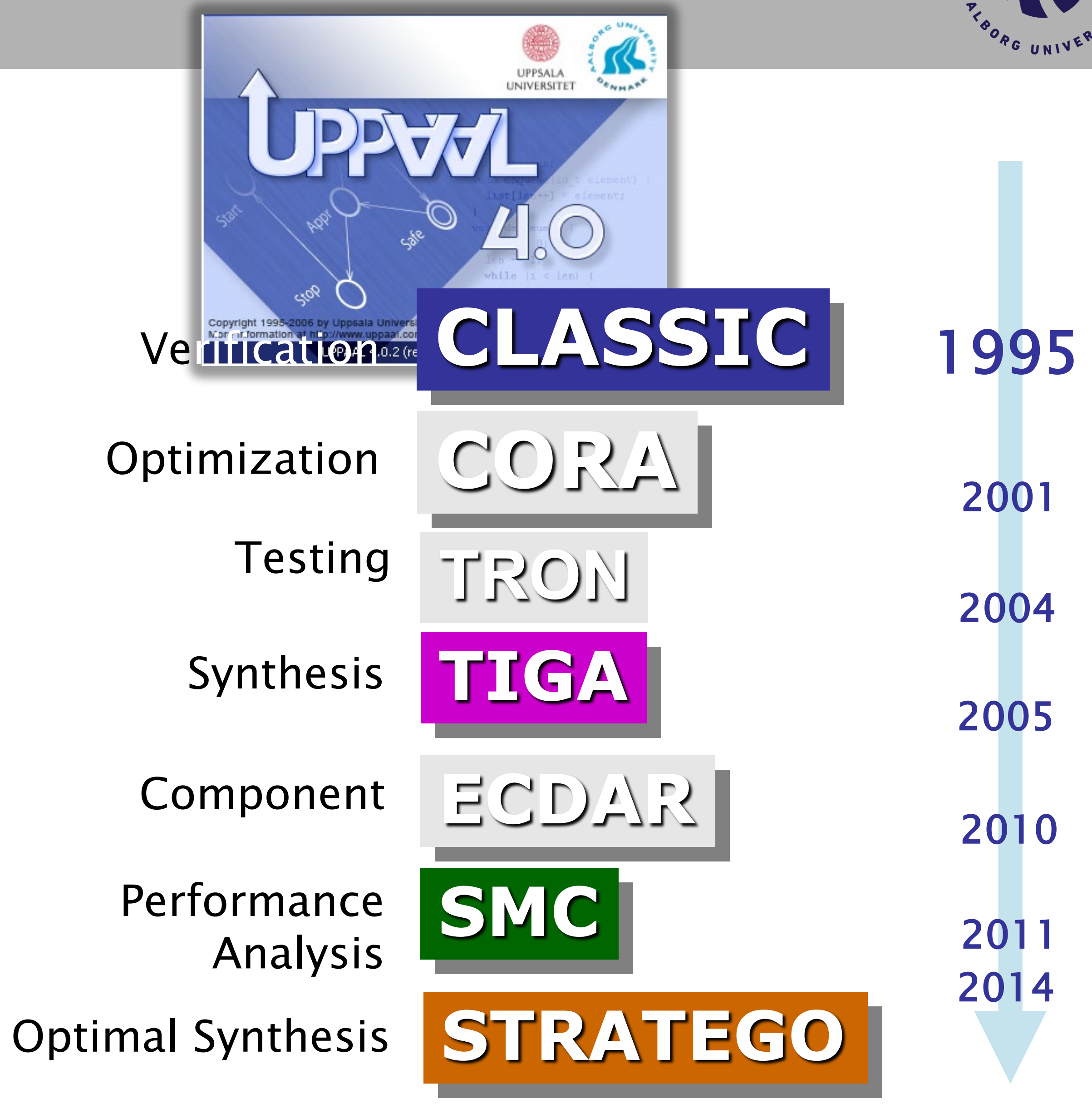
Networked ES

Big Data

Cyber-Physical Systems

Overview

- UPPAAL STRATEGO
  - Dagstuhl 19432
    Smart Farming Challenge
- Better Learning
- Approximation Methods
- Compact Strategies
- Applications
  - WORDLE
- Future Challenges



SMART FARMING
&
UPPAAL STRATEGO

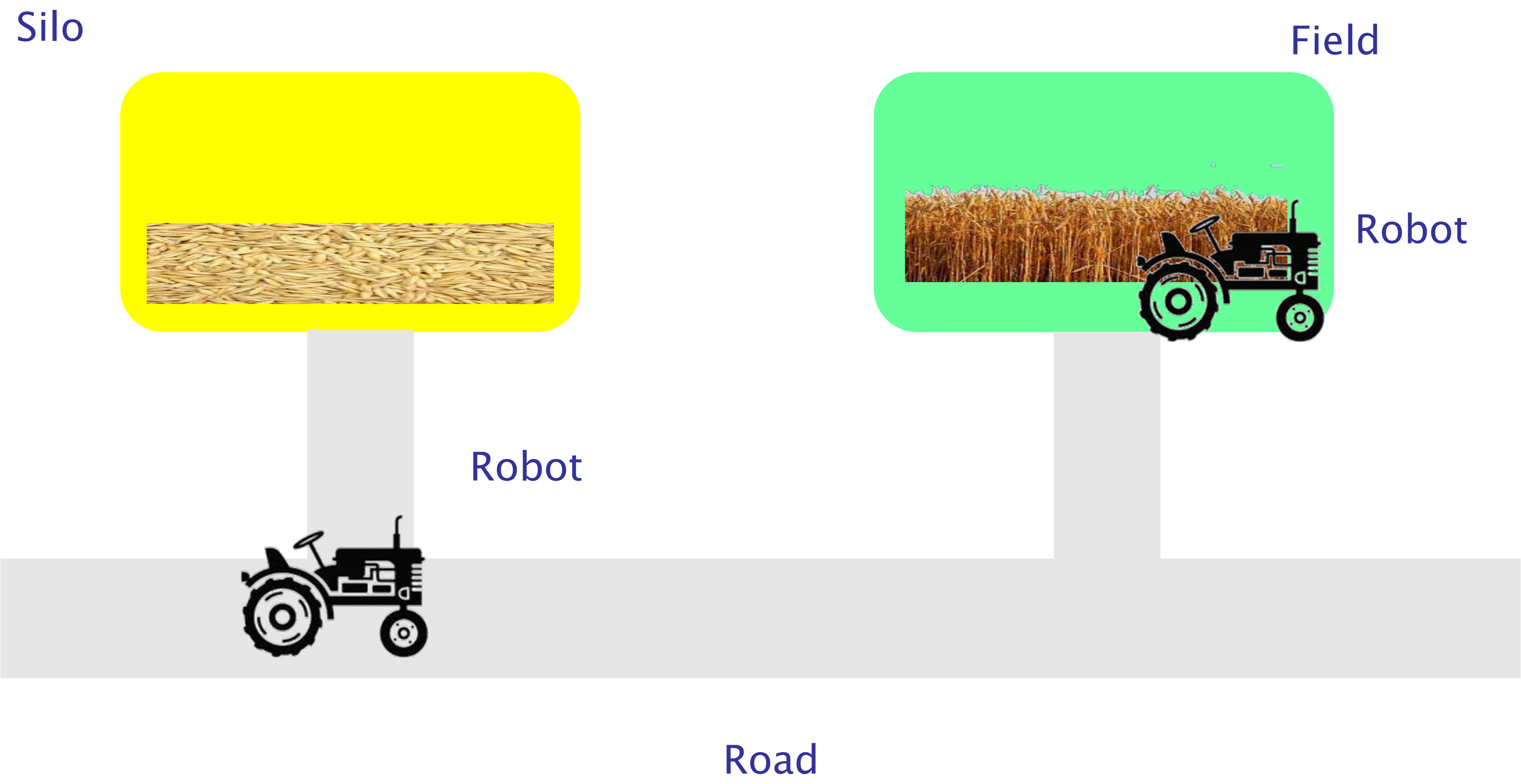
AALBORG UNIVERSITET

erc

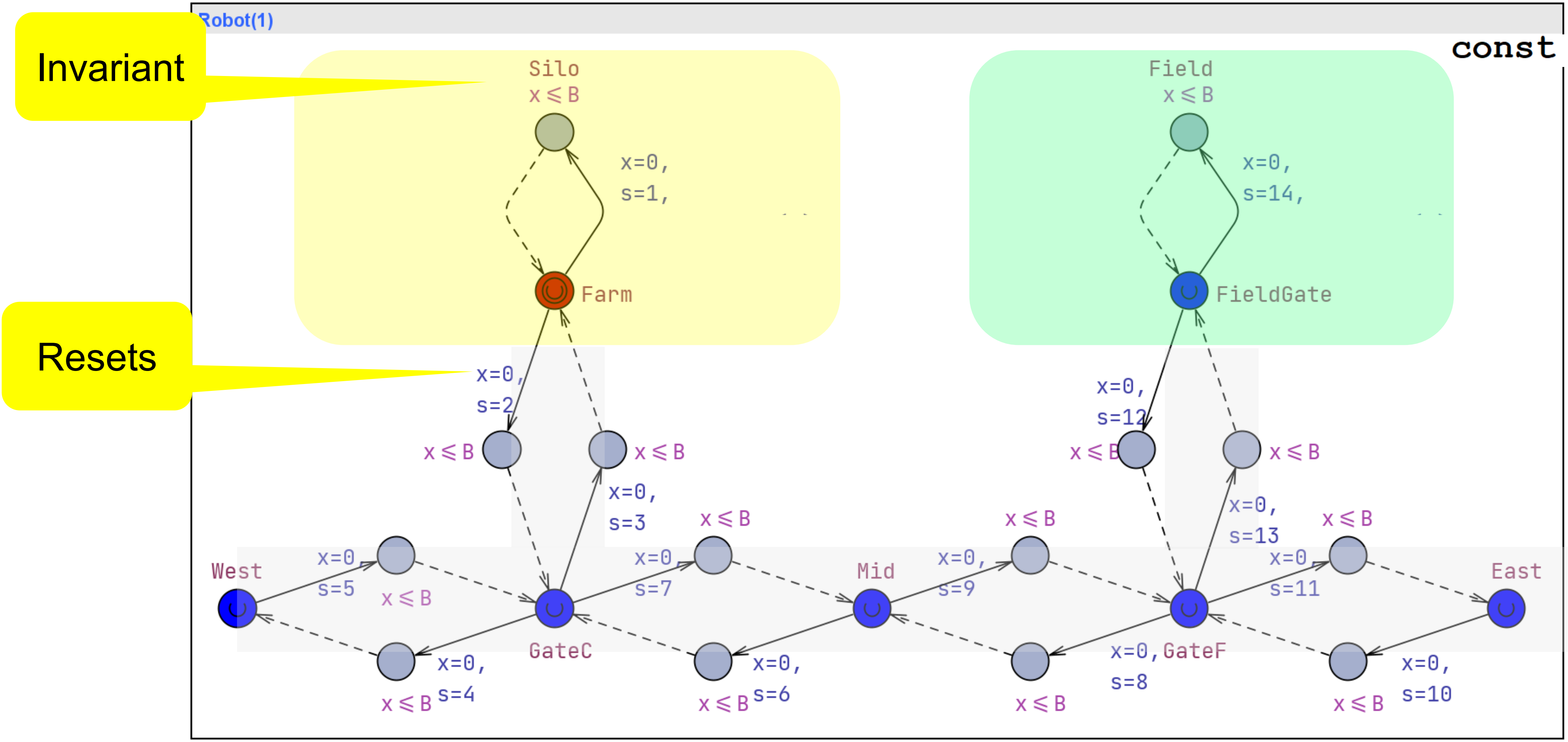
VILLUM FONDEN

TACAS15: David, Jensen, Larsen, Mikucionis, Taankvist: Uppaal Stratego.

Smart Farming / Dagstuhl 19432 Oct19



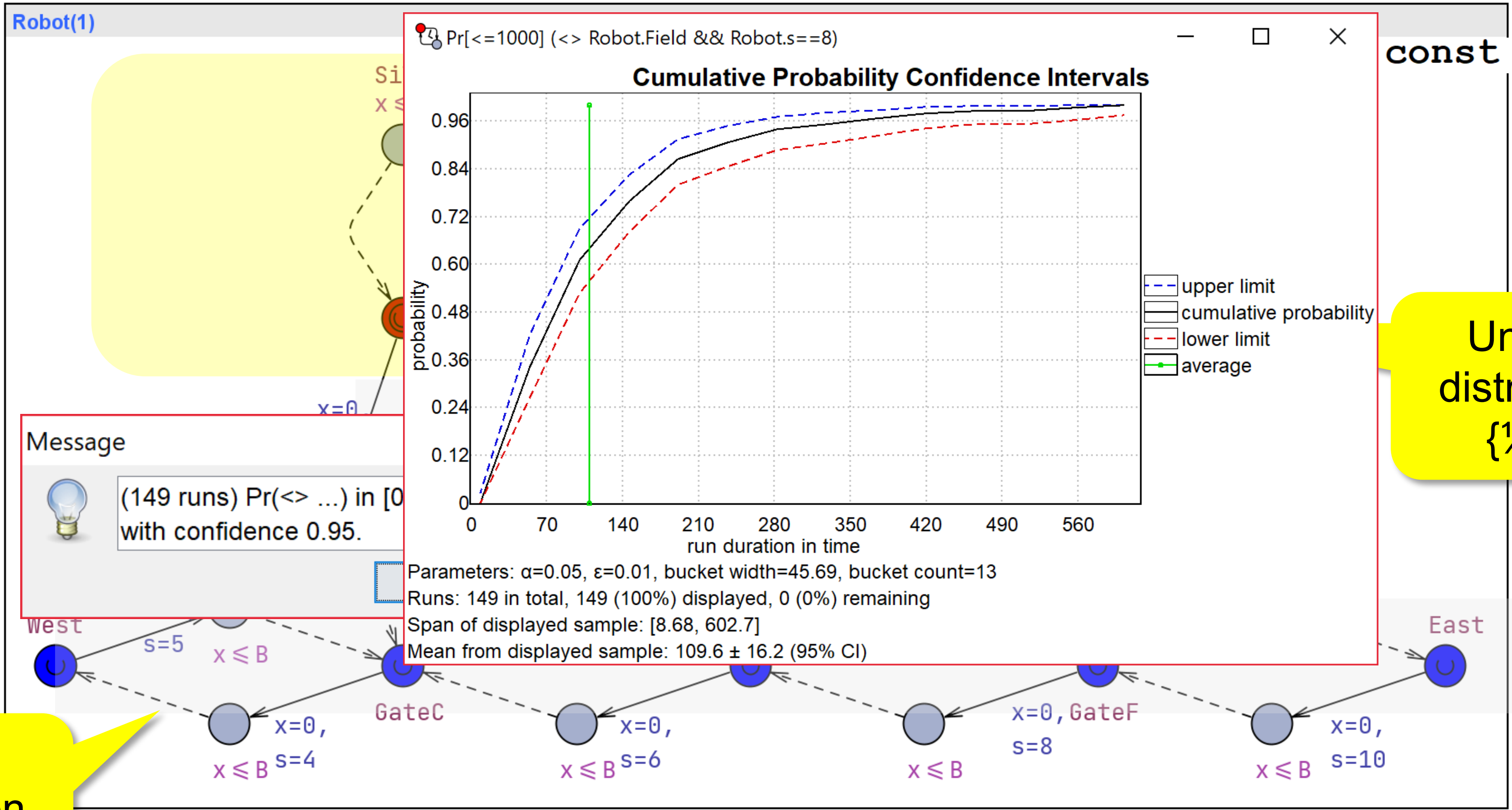
Smart Farming – Timed Automata




```
const int B=5;
clock x;
int s=1;
```

E<> Robot(1).Field && Robot(1).s==14

Smart Farming – Stochastic Timed Automata



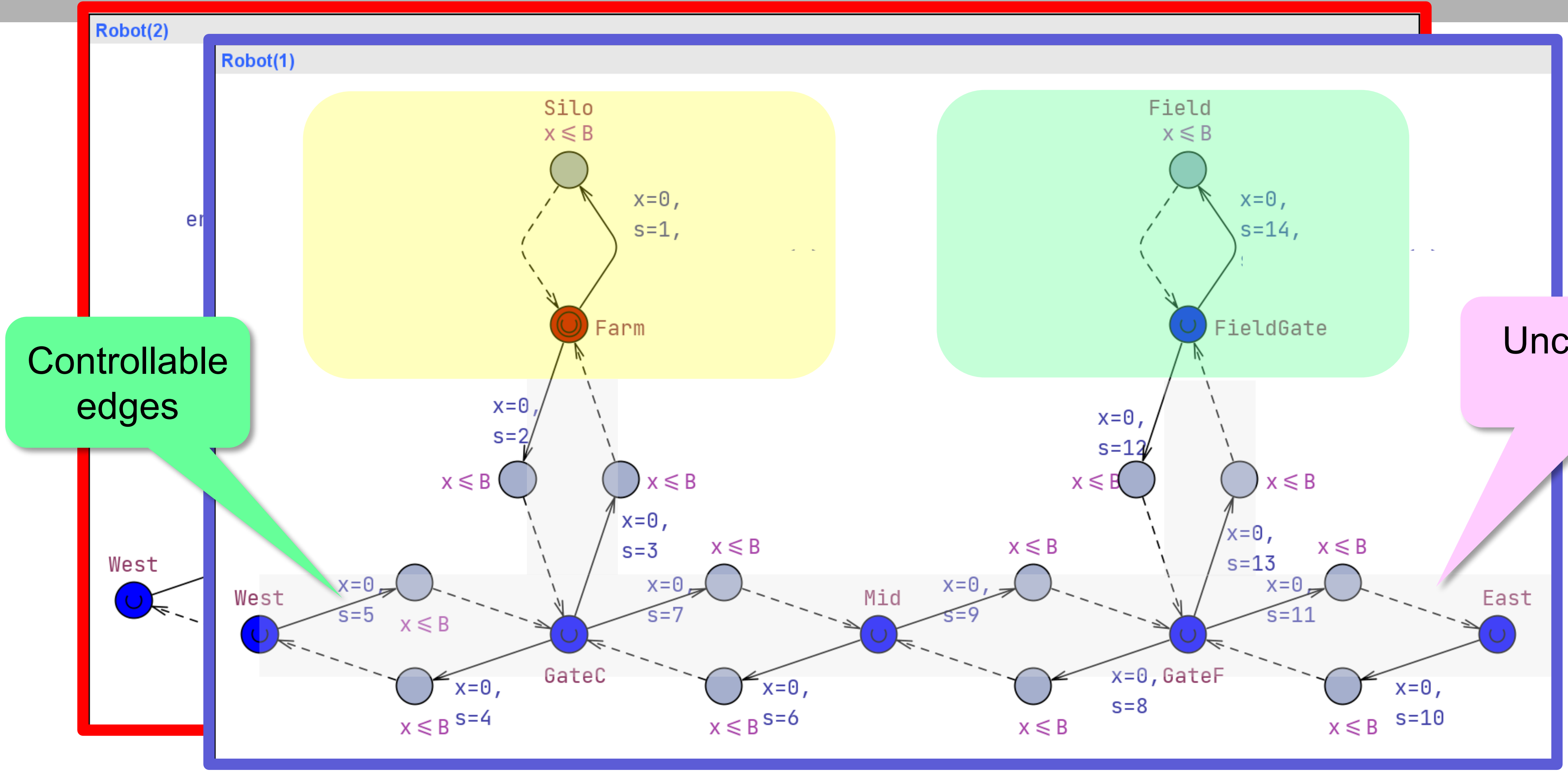

```
const int B=5;
clock x;
int s=1;
```

Uniform distribution {½,½}

Uniform distribution [0,B]

Pr[<=1000] (<> Robot(1).Field && Robot(1).s==14)

Smart Farming –Timed Game




```
int B=5;
clock x;
int s=1;
```

strategy segsafe = control: A[] ! (Robot(1).s>1 && Robot(1).s<14 && Robot(1).s==Robot(2).s) segsafe - Notepad

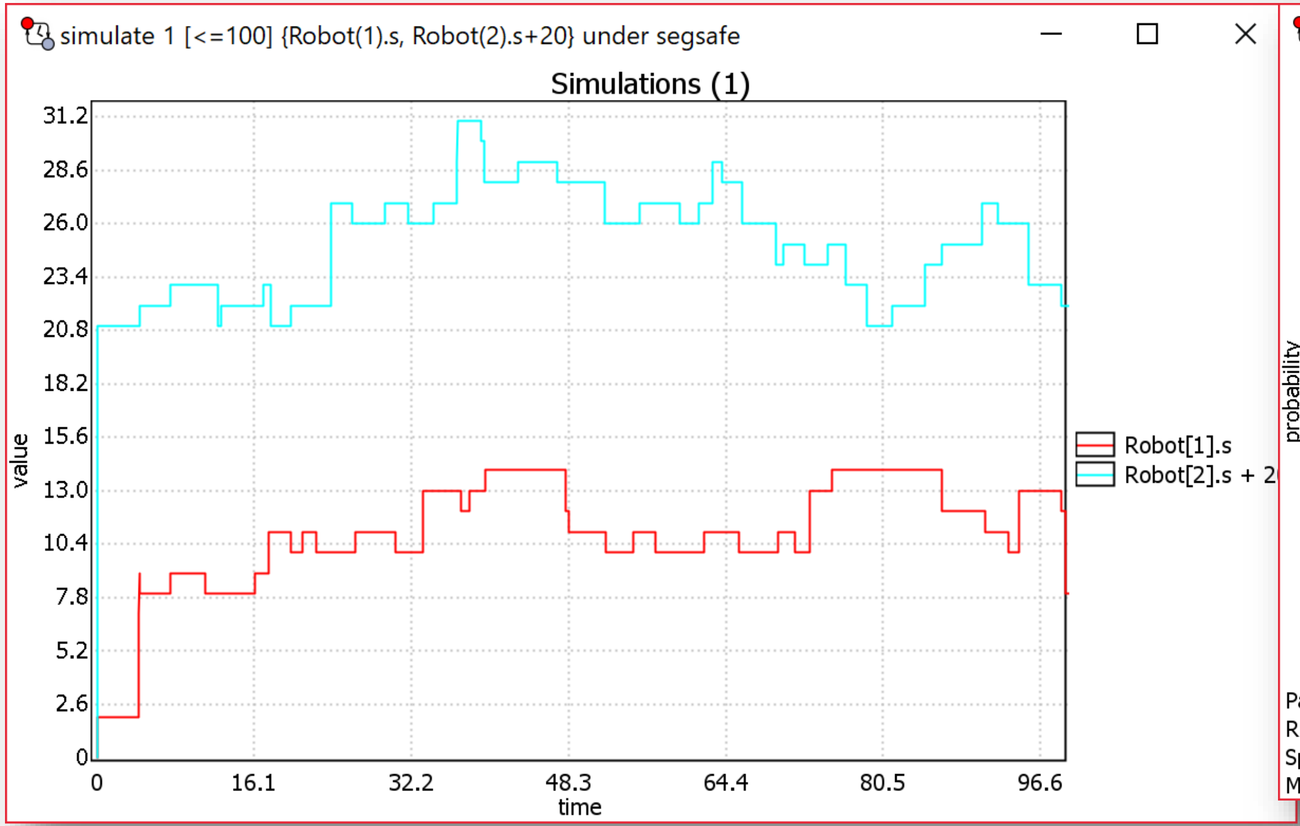
File Edit Format View Help

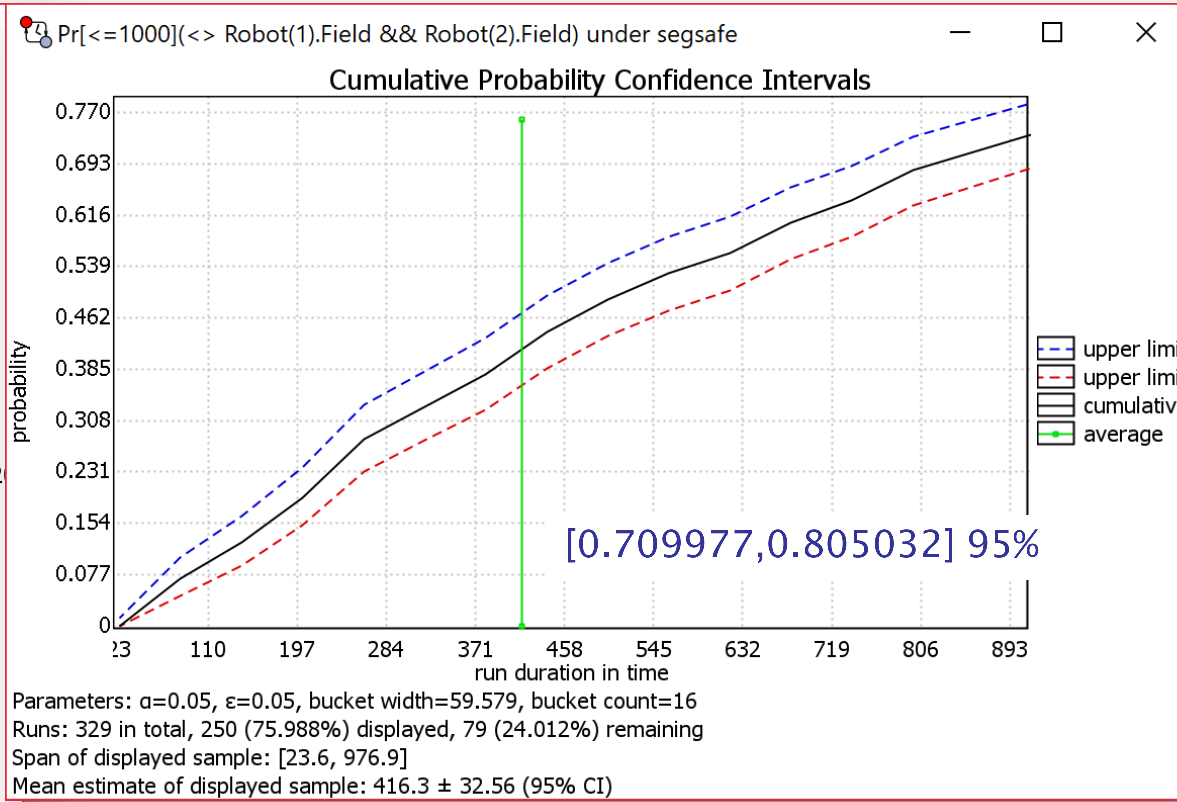
```
State: ( Robot(1)._id7 Robot(2)._id17 Rain.Wet Load._id55 Field._id49 Store._id50 ) w=5 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=7 Robot(2).s=10
While you are in        (Robot(1).x<=5 && Robot(2).x<=5), wait.

State: ( Robot(1)._id7 Robot(2).GateC Rain.Wet Load._id55 Field._id49 Store._id50 ) w=5 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=7 Robot(2).s=6
When you are in (Robot(1).x<=5), take transition Robot(2).GateC->Robot(2)._id2 { 1, tau, x := 0, s := 3 }
When you are in (Robot(1).x<=5), take transition Robot(2).GateC->Robot(2)._id5 { 1, tau, x := 0, s := 4 }

State: ( Robot(1)._id16 Robot(2).FieldGate Rain.Wet Load._id55 Field._id49 Store._id50 ) w=5 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=11 Robot(2).s=14
When you are in (Robot(1).x<=5), take transition Robot(2).FieldGate->Robot(2)._id13 { 1, tau, x := 0, s := 12 }
When you are in (Robot(1).x<=5), take transition Robot(2).FieldGate->Robot(2).Field { 1, tau, x := 0, s := 14, start_harvest(rid) }

State: ( Robot(1)._id14 Robot(2).Mid Rain.Dry Load._id55 Field._id49 Store._id50 ) w=1 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=13 Robot(2).s=7
When you are in (Robot(1).x<=5), take transition Robot(2).Mid->Robot(2)._id10 { 1, tau, x := 0, s := 9 }
When you are in (Robot(1).x<=5), take transition Robot(2).Mid->Robot(2)._id8 { 1, tau, x := 0, s := 6 }

State: ( Robot(1)._id11 Robot(2)._id7 Rain.Wet Load._id55 Field._id49 Store._id50 ) w=5 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=8 Robot(2).s=7
While you are in        (Robot(1).x<=5 && Robot(2).x<=5), wait.

State: ( Robot(1)._id13 Robot(2)._id4 Rain.Wet Load._id55 Field._id49 Store._id50 ) w=5 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=12 Robot(2).s=5
While you are in        (Robot(1).x<=5 && Robot(2).x<=5), wait.

State: ( Robot(1).GateF Robot(2)._id7 Rain.Dry Load._id55 Field._id49 Store._id50 ) w=1 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=12 Robot(2).s=7
When you are in (Robot(2).x<=5), take transition Robot(1).GateF->Robot(1)._id11 { 1, tau, x := 0, s := 8 }
When you are in (Robot(2).x<=5), take transition Robot(1).GateF->Robot(1)._id16 { 1, tau, x := 0, s := 11 }
When you are in (Robot(2).x<=5), take transition Robot(1).GateF->Robot(1)._id14 { 1, tau, x := 0, s := 13 }

State: ( Robot(1)._id1 Robot(2)._id13 Rain.Dry Load._id55 Field._id49 Store._id50 ) w=1 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=2 Robot(2).s=12
While you are in        (Robot(1).x<=5 && Robot(2).x<=5), wait.

State: ( Robot(1)._id14 Robot(2)._id10 Rain.Dry Load._id55 Field._id49 Store._id50 ) w=1 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=13 Robot(2).s=9
While you are in        (Robot(1).x<=5 && Robot(2).x<=5), wait.

State: ( Robot(1).GateC Robot(2)._id7 Rain.Dry Load._id55 Field._id49 Store._id50 ) w=1 harvesting[1]=0 harvesting[2]=0 storing[1]=0 storing[2]=0 Robot(1).s=6 Robot(2).s=7
When you are in (Robot(2).x<=5), take transition Robot(1).GateC->Robot(1)._id2 { 1, tau, x := 0, s := 3 }
When you are in (Robot(2).x<=5), take transition Robot(1).GateC->Robot(1)._id5 { 1, tau, x := 0, s := 4 }

State: ( Robot(1).Silo Robot(2)._id7 Rain.Wet Load._id55 Field._id49 Store._id50 ) w=5 harvesting[1]=0 harvesting[2]=0 storing[1]=1 storing[2]=0 Robot(1).s=1 Robot(2).s=7
While you are in        (Robot(1).x<=5 && Robot(2).x<=5), wait.
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

strategy segsafe = control: A[] ! (Robot(1).s>1 && Robot(1).s<14 && Robot(1).s==Robot(2).s)

Smart Farming – Timed Games





simulate 1 [<=100] {Robot(1).s, Robot(2).s+20} under segsafe

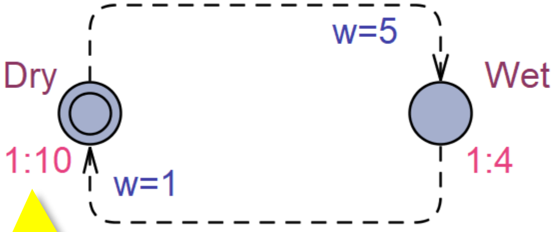
E<> Robot(1).Field && Robot(2).Field under segsafe

E<> Robot(1).s>1 && Robot(1).s<14 && Robot(1).s==Robot(2).s under segsafe

Pr[<=1000] (<> Robot(1).Field && Robot(2).Field==14) under segsafe

Smart Farming – Stochastic & Hybrid Stuff

Rain



Rates of exponential distributions

Field

```
f'==(((F-f)/(0.5*F))*w)*scale-
(harvesting[1]*f*(1-(ld[1]/capacity)) -
 harvesting[2]*f*(
```

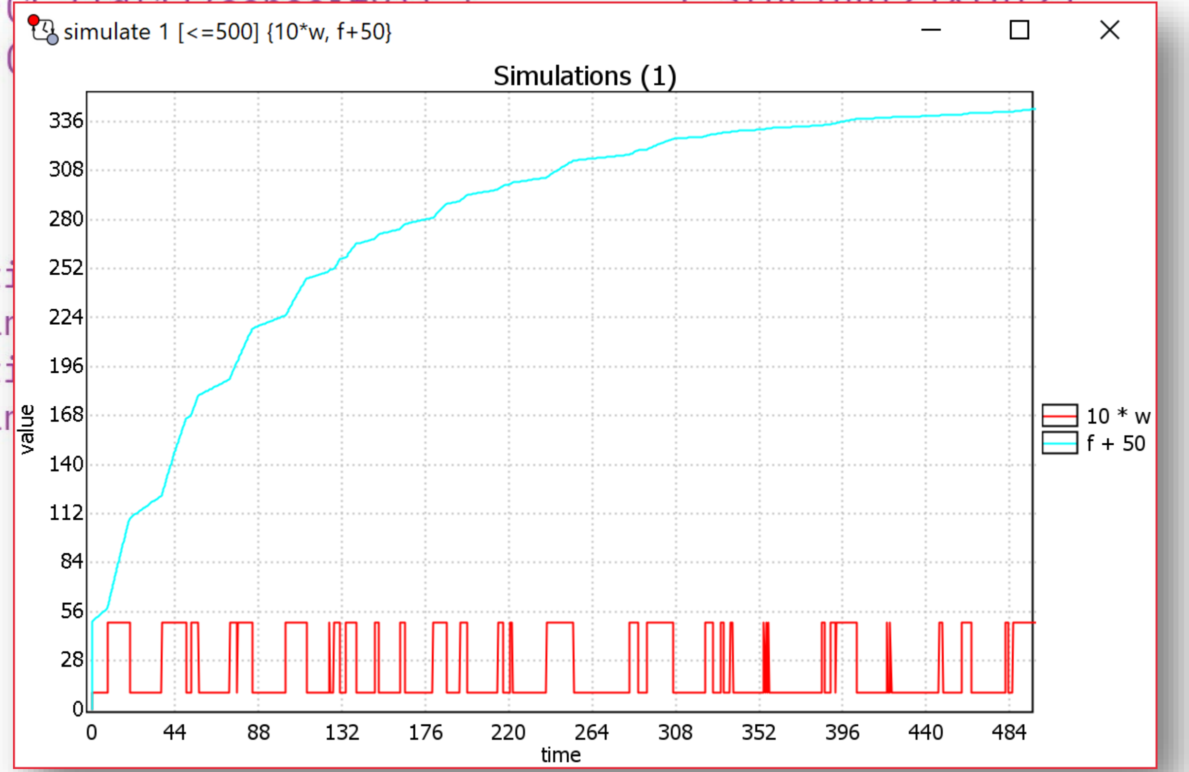
Load

```
ld[1]'==harvesti
          - storin
ld[2]'==harvesti
          - storin
```

Store

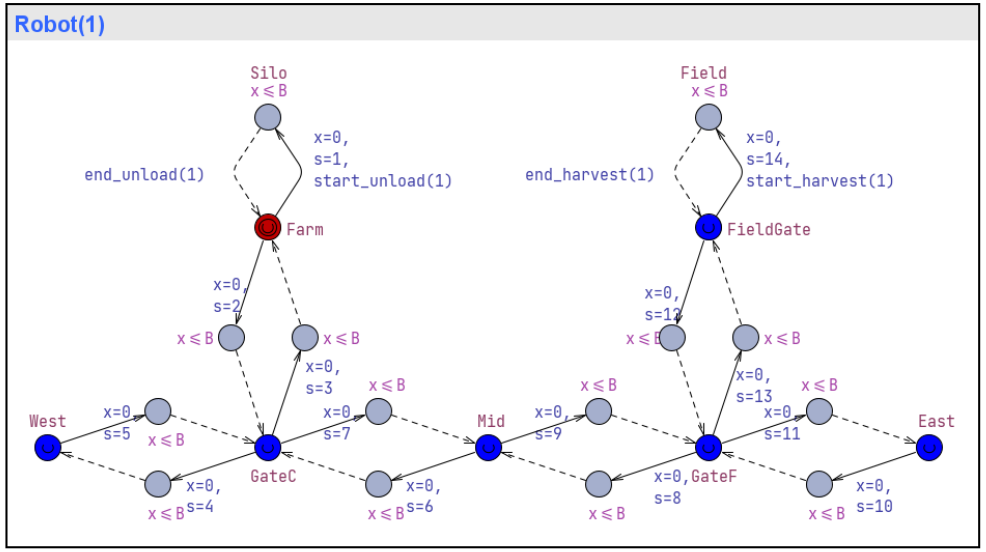
```
c'==storing[1]*ld[1]
    + storing[2]*ld[2]
```

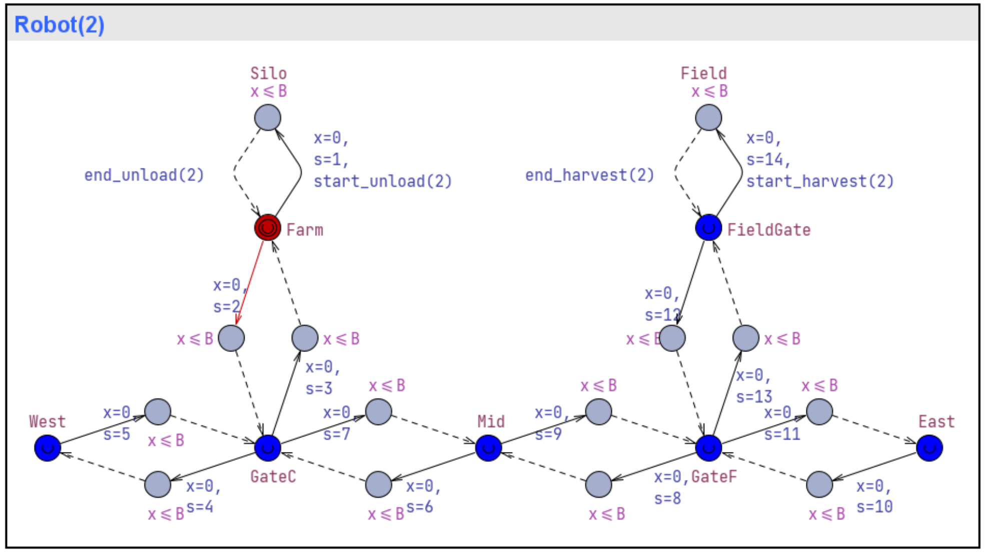
ODEs

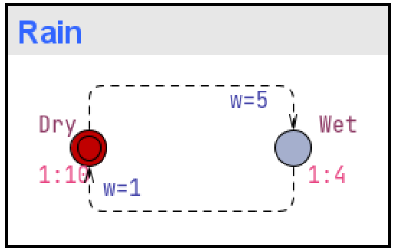


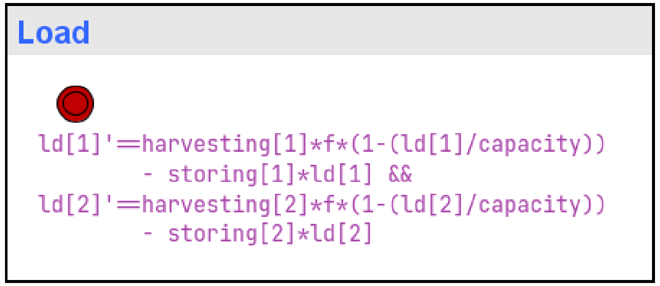

```
const double capacity = 40.0;
hybrid clock ld[id_t];
hybrid clock f, c;
```

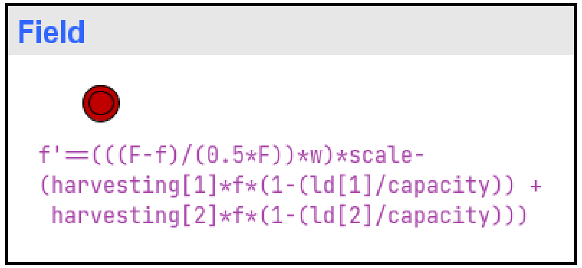
Smart Farming – Complete Model

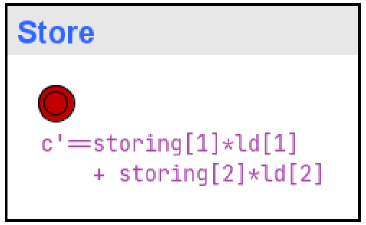








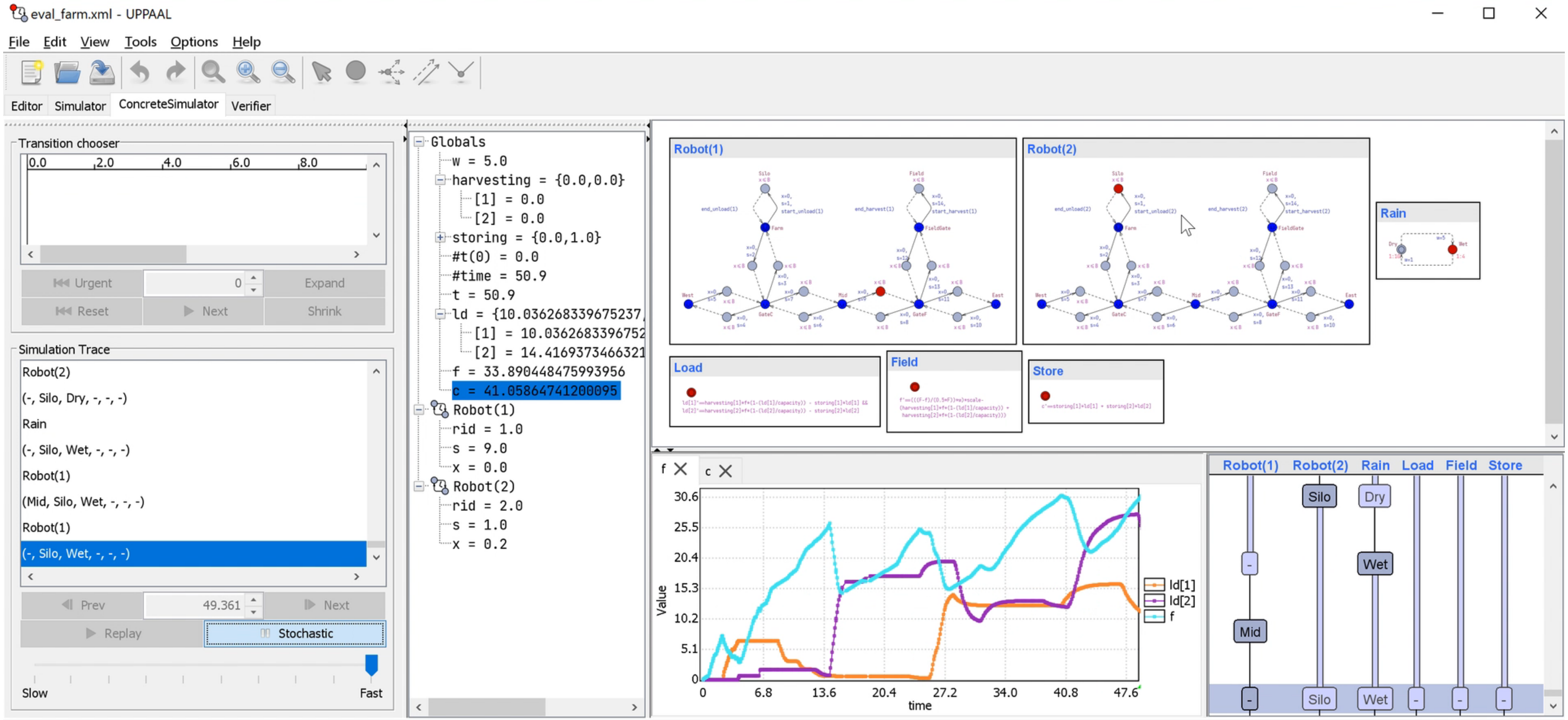




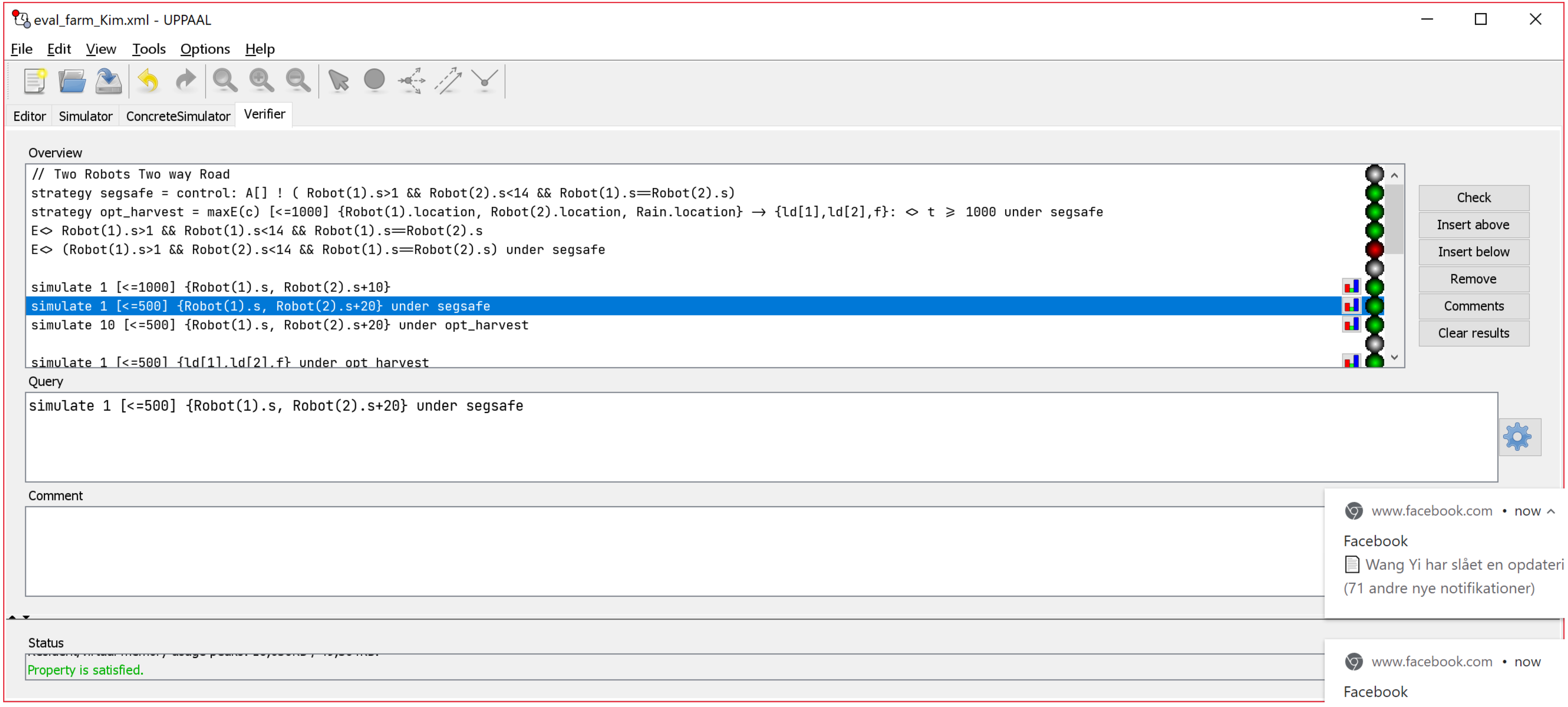

```
typedef int[1,2] id_t;
const int B=5;
int w=1;
clock t;


const double capacity = 40.0;
hybrid clock ld[id_t];
hybrid clock f, c;
```

```
bool harvesting[id_t] = {false, false};
bool storing[id_t] = {false,false};

void start_unload(id_t rid) {
    storing[rid] = true;
}

void end_unload(id_t rid) {
    storing[rid] = false;
}
```

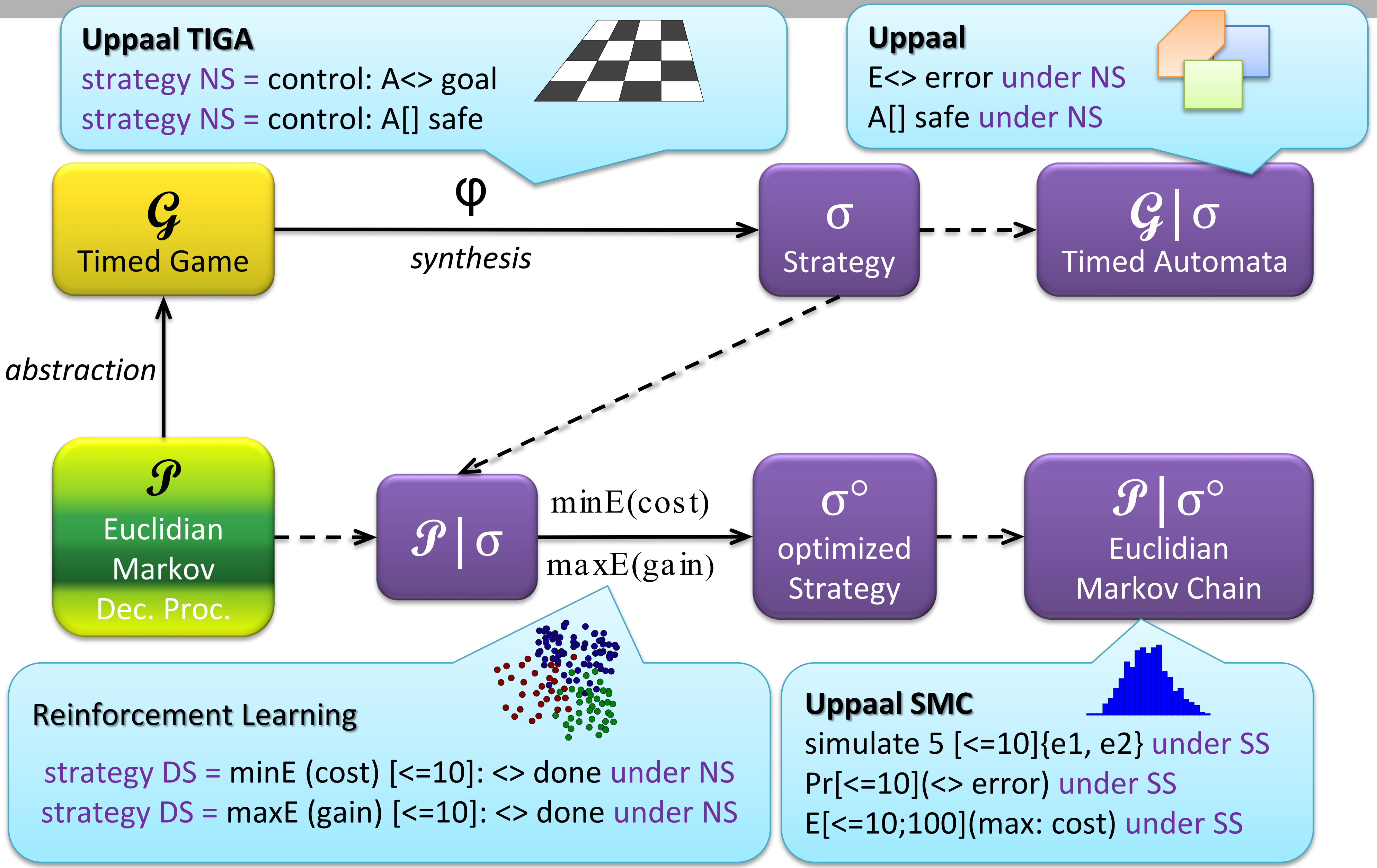
Farming Benchmark – in Stratego



Farming Benchmark – in Stratego



Workflow under UPPAAL Stratego



Smart Farr

Reward to be maximized

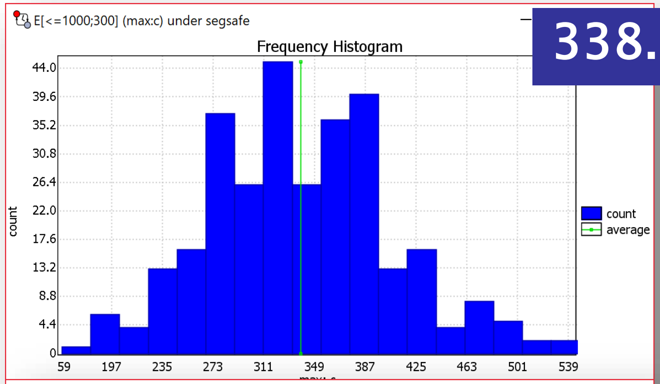
Goad

Partial Observability

Q1: strategy segsafe = ontrol: A[] ! (Rob t(1).s>1 && Robot(1).s<14 && Robot(1 s)

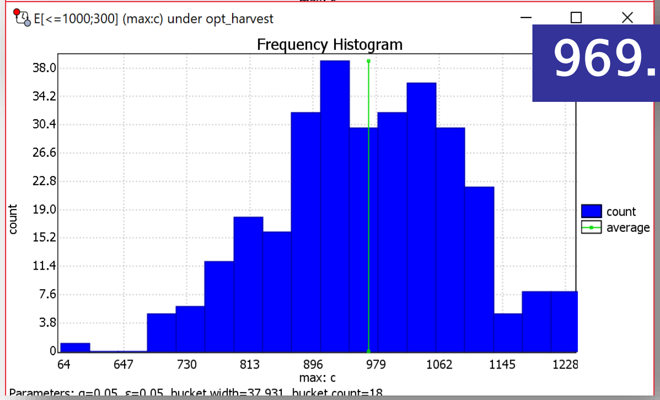
Q2: strategy opt_harvest =
maxE(c) [<=1000] {Robot(1).location, Robot(2).location, Rain.location} -> {ld1,ld2,t}:
<> t >= 1000 under segsafe

Q3: E[<=1000;300] (max:c) under segsafe



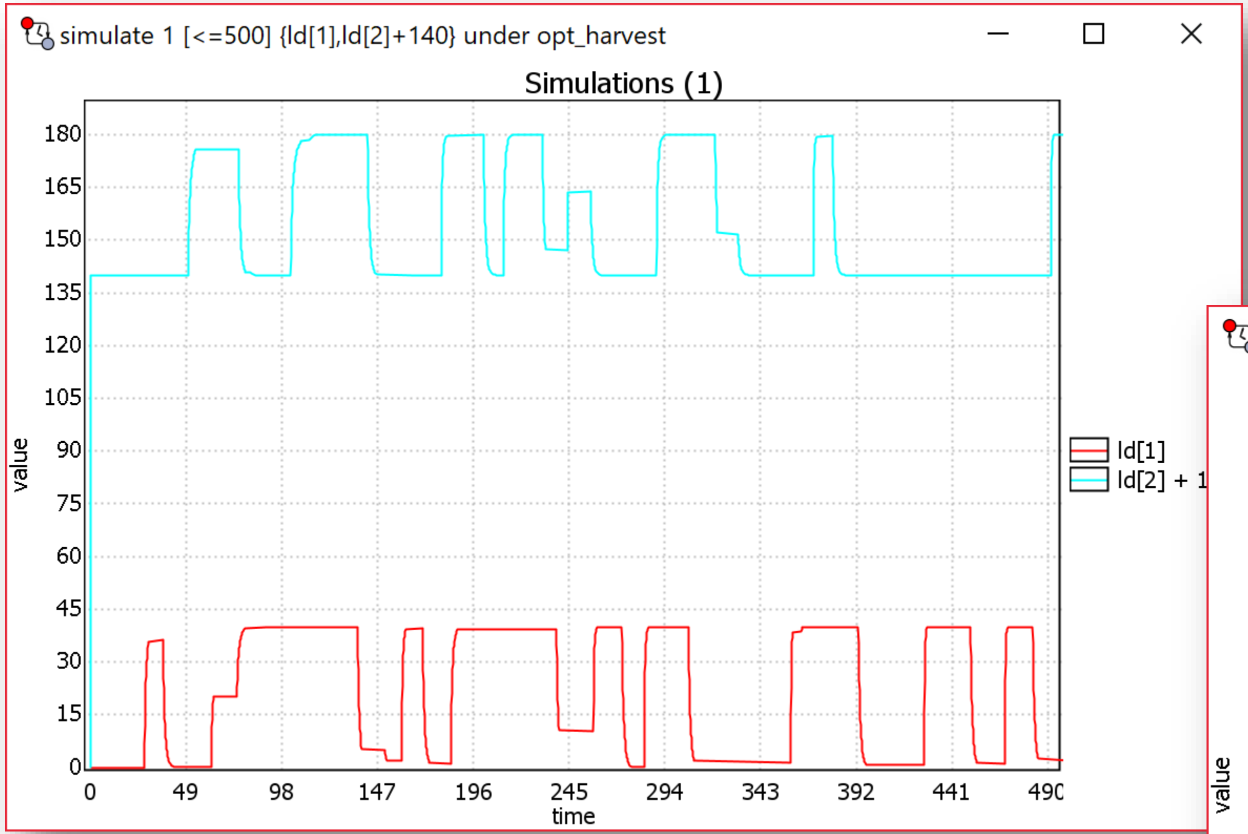
338.881 +/- 7.80819

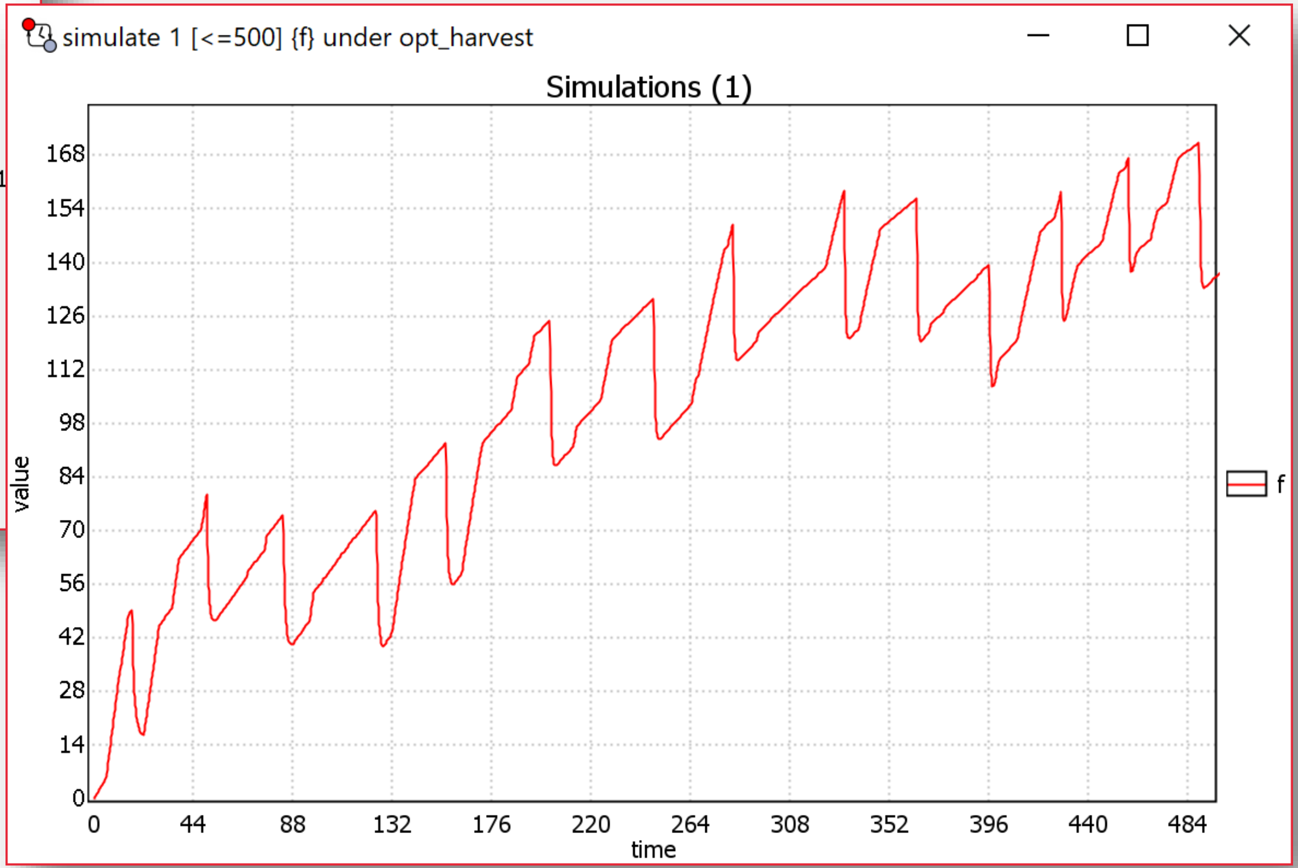
Q4: E[<=1000;300] (max:c) under opt_harvest



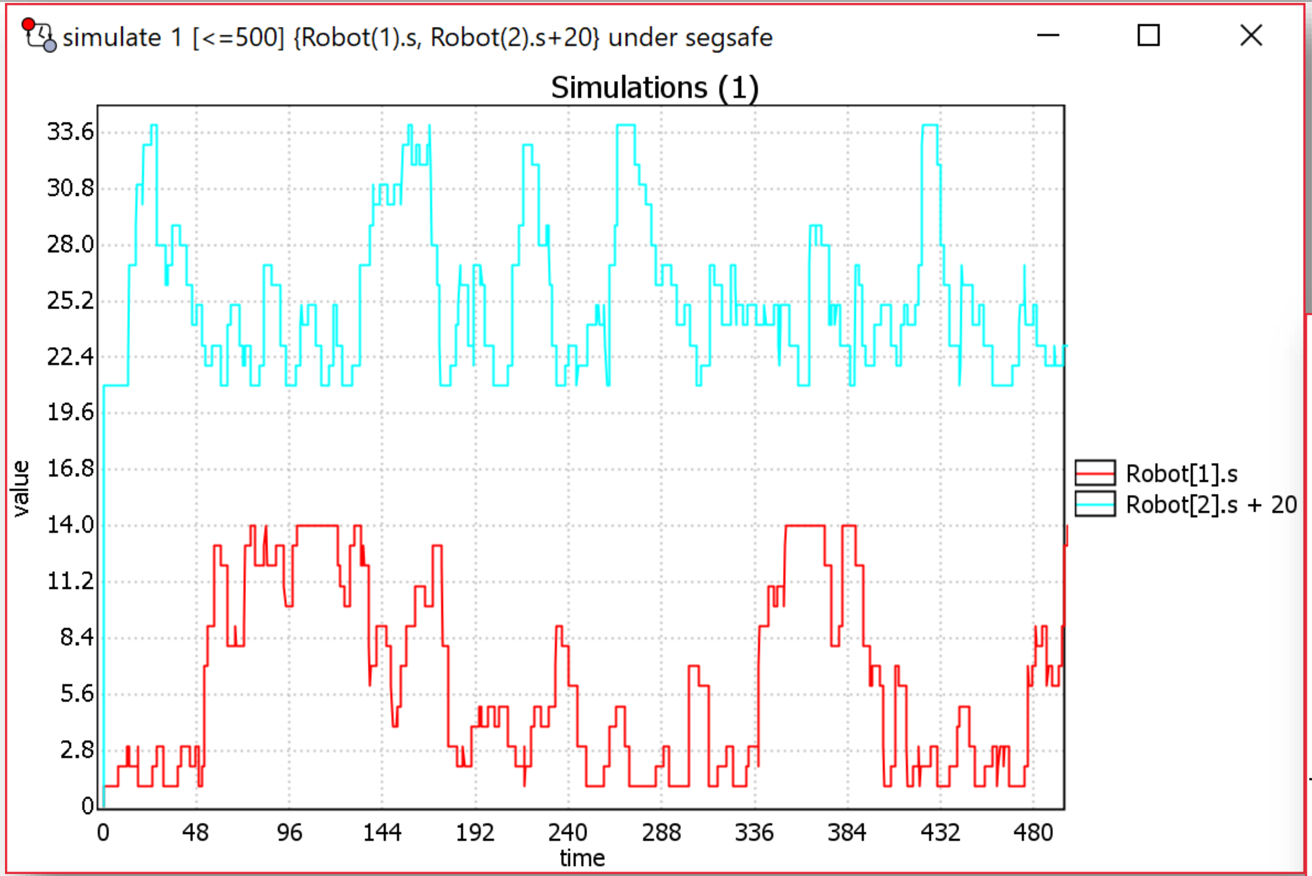
969.112 +/- 13.8988

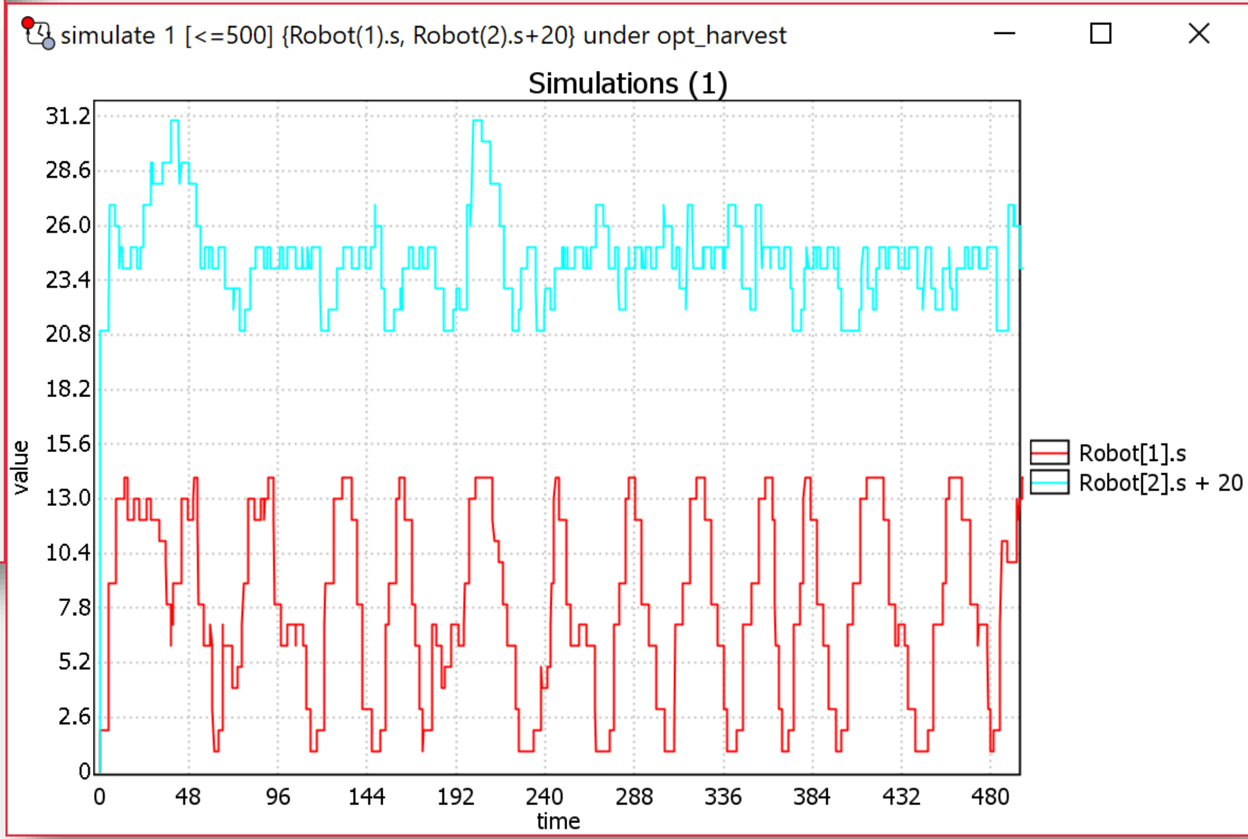
ld[1], ld[2] – f


simulate 1 [<=500] {ld[1],ld[2]+140} under opt_harvest
Simulations (1)
180
165
150
135
120
105
90
75
60
45
30
15
0
value
0
49
98
147
196
245
294
343
392
441
490
time
ld[1]
ld[2] + 1


simulate 1 [<=500] {f} under opt_harvest
Simulations (1)
168
154
140
126
112
98
84
70
56
42
28
14
0
value
0
44
88
132
176
220
264
308
352
396
440
484
time
f

Robots Movement





Reinforcement Learning

in UPPAAL Stratego

AALBORG UNIVERSITET

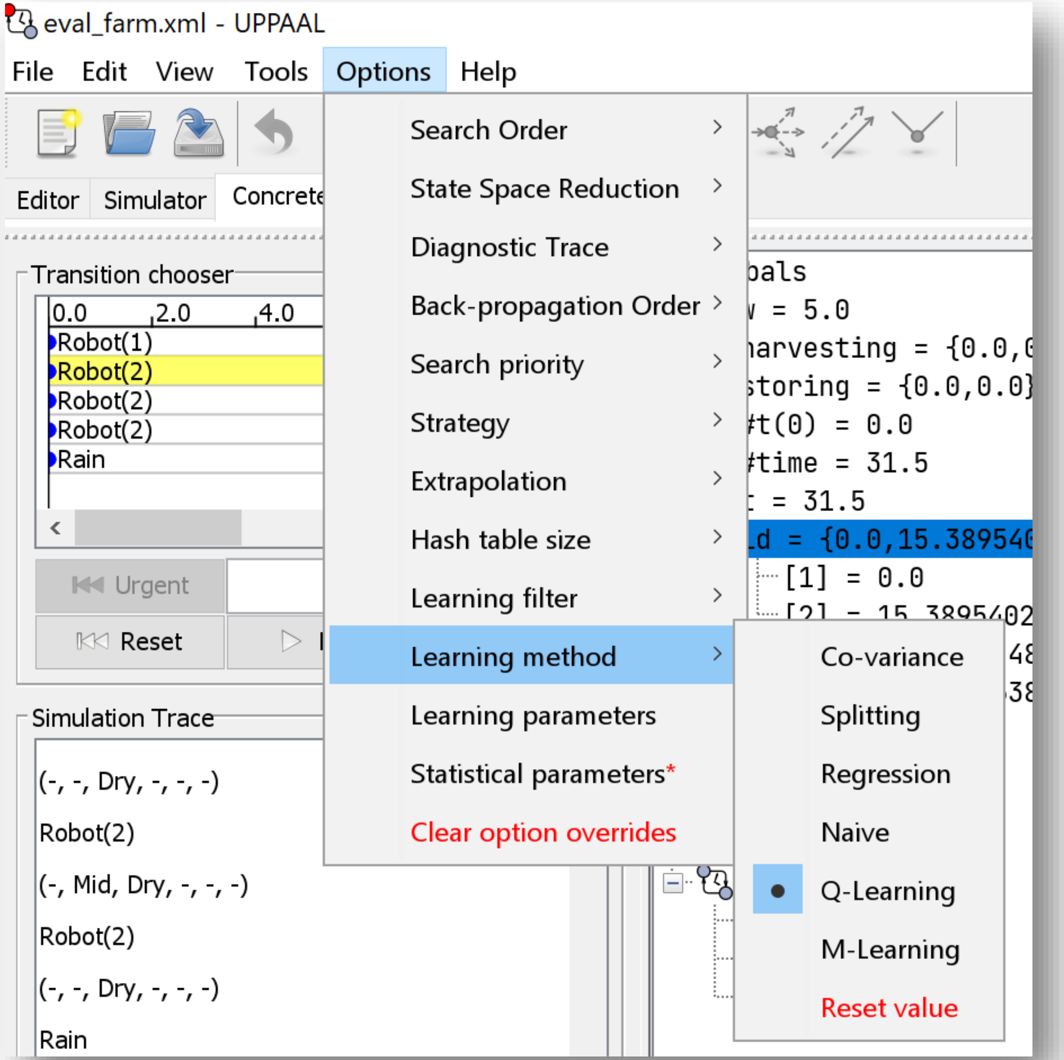
erc

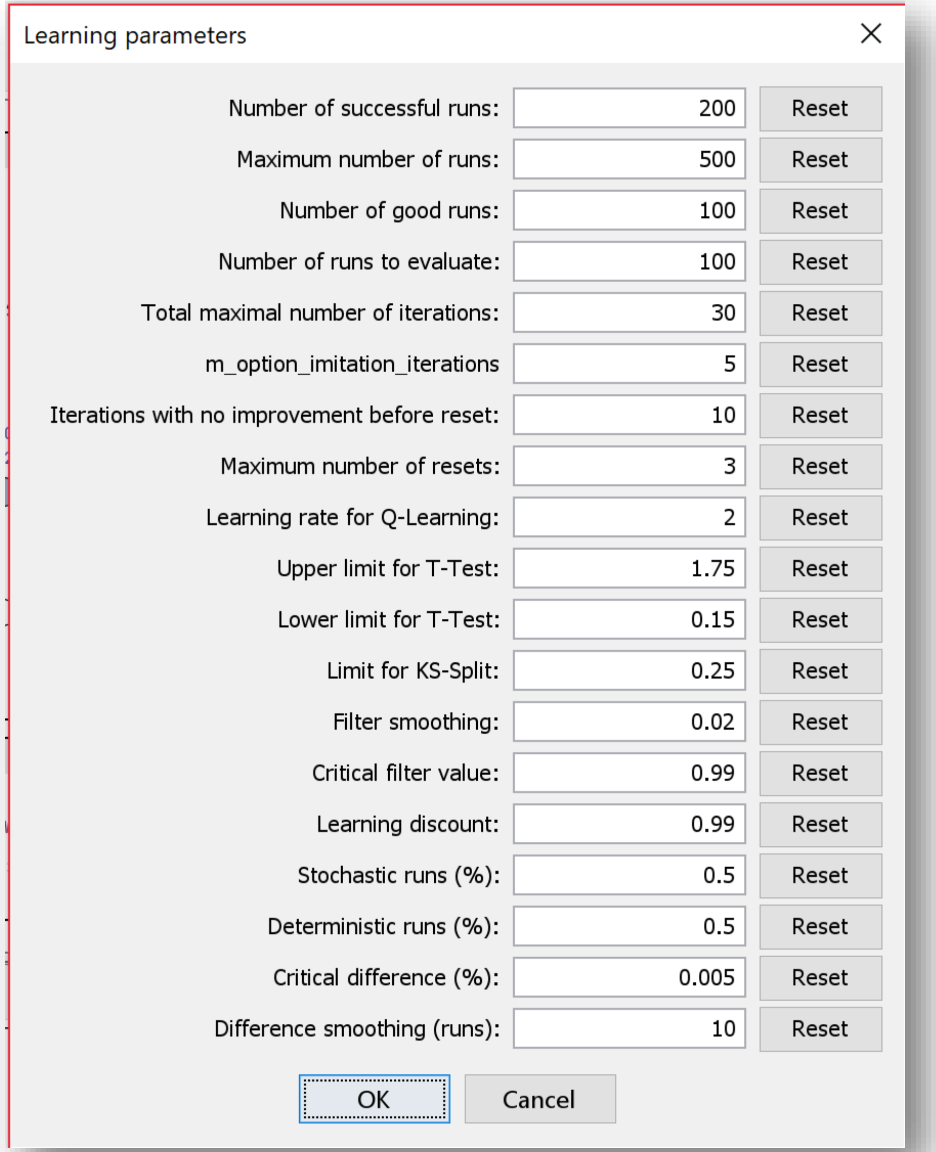
VILLUM FONDEN

UPPAAL 4.0

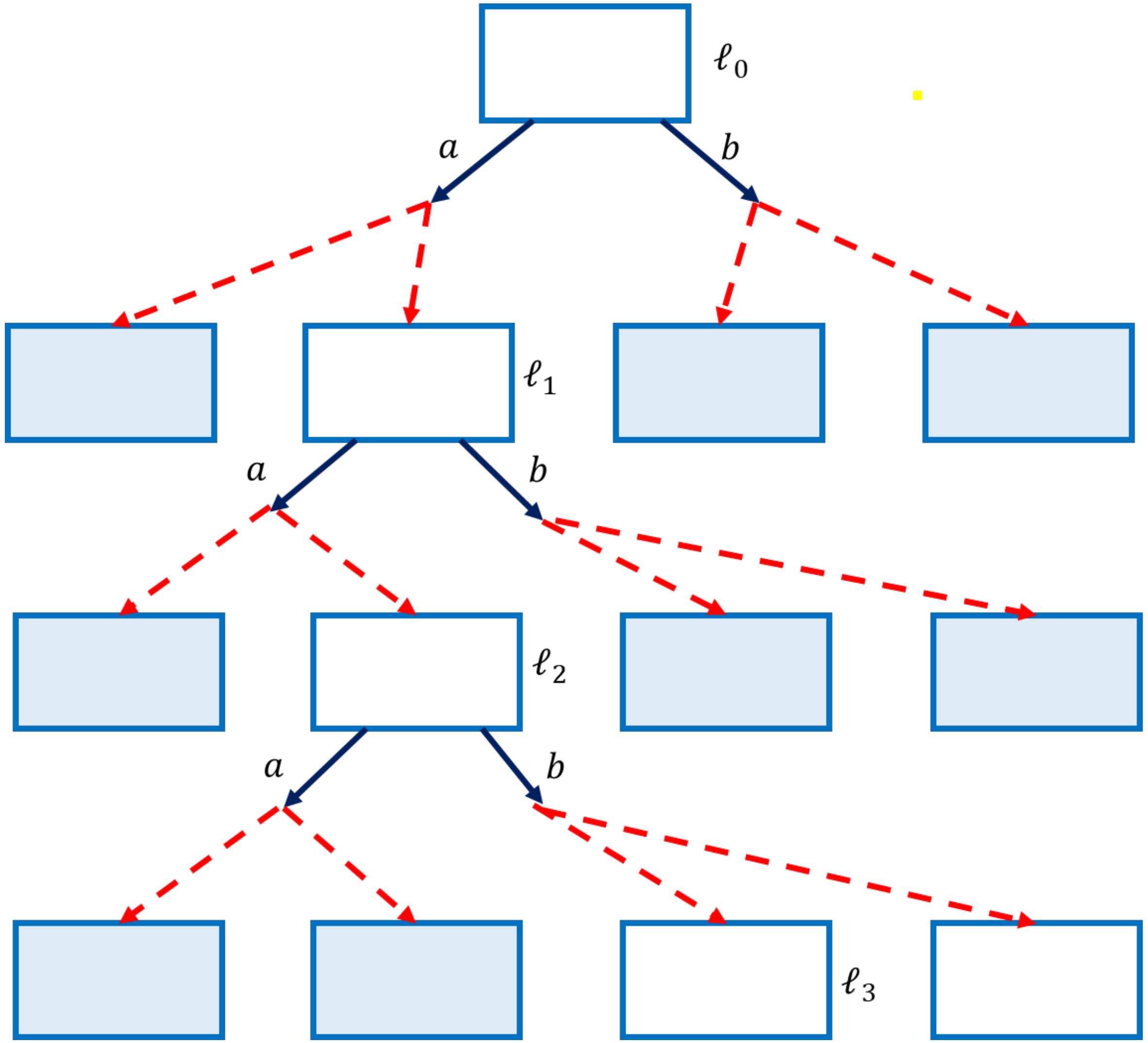
Stratego

Learning Methods & Parameters





Euclidean MDP



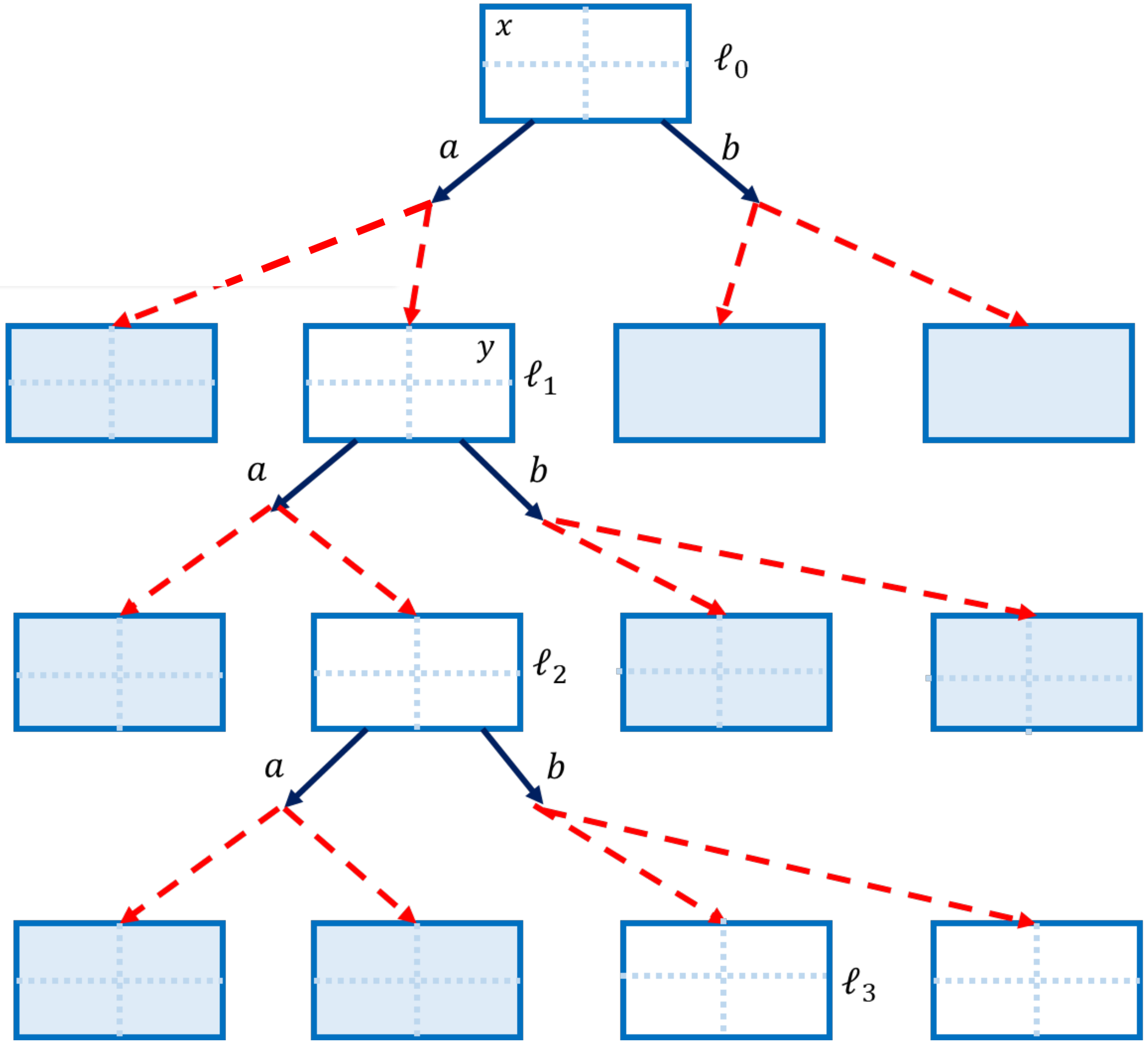
- States $\boldsymbol{S} \subseteq \mathbb{R}^{k}$
- $\boldsymbol{Act}$ is a finite set of actions
- Initial state $\boldsymbol{s_0} \in \boldsymbol{S}$
- Next state density function $\boldsymbol{T}: \boldsymbol{S} \times \boldsymbol{Act} \to (\boldsymbol{S} \to \mathbb{R}_{\geq 0})$
- Cost-function $\boldsymbol{C}: \boldsymbol{S} \times \boldsymbol{Act} \times \boldsymbol{S} \to \mathbb{R}$
- Goal states $\boldsymbol{G} \subseteq \boldsymbol{S}$

Strategy $\boldsymbol{\sigma}: \boldsymbol{S} \to (\boldsymbol{Act} \to [0,1])$

$\mathbb{E}_{\boldsymbol{\sigma}}(\boldsymbol{G}, \boldsymbol{s})$:
Expected cost of reaching $\boldsymbol{G}$ under $\boldsymbol{\sigma}$.

Find $\boldsymbol{\sigma}^*$ st. $\mathbb{E}_{\boldsymbol{\sigma}^*}(\boldsymbol{G}, \boldsymbol{s}) = \inf_{\boldsymbol{\sigma}} \mathbb{E}_{\boldsymbol{\sigma}}(\boldsymbol{G}, \boldsymbol{s})$

Partitioning (IMDP)



- Finite partition A (respecting G)
- Next state transition-function
  $T_A: A \times Act \times A \rightarrow [0,1] \times [0,1]$
- Cost-function
  $C_A: A \times Act \times A \rightarrow \mathbb{R} \times \mathbb{R}$

$$\mathbb{E}_A^{min}(G, x) \qquad \mathbb{E}_A^{max}(G, x)$$

Theorem

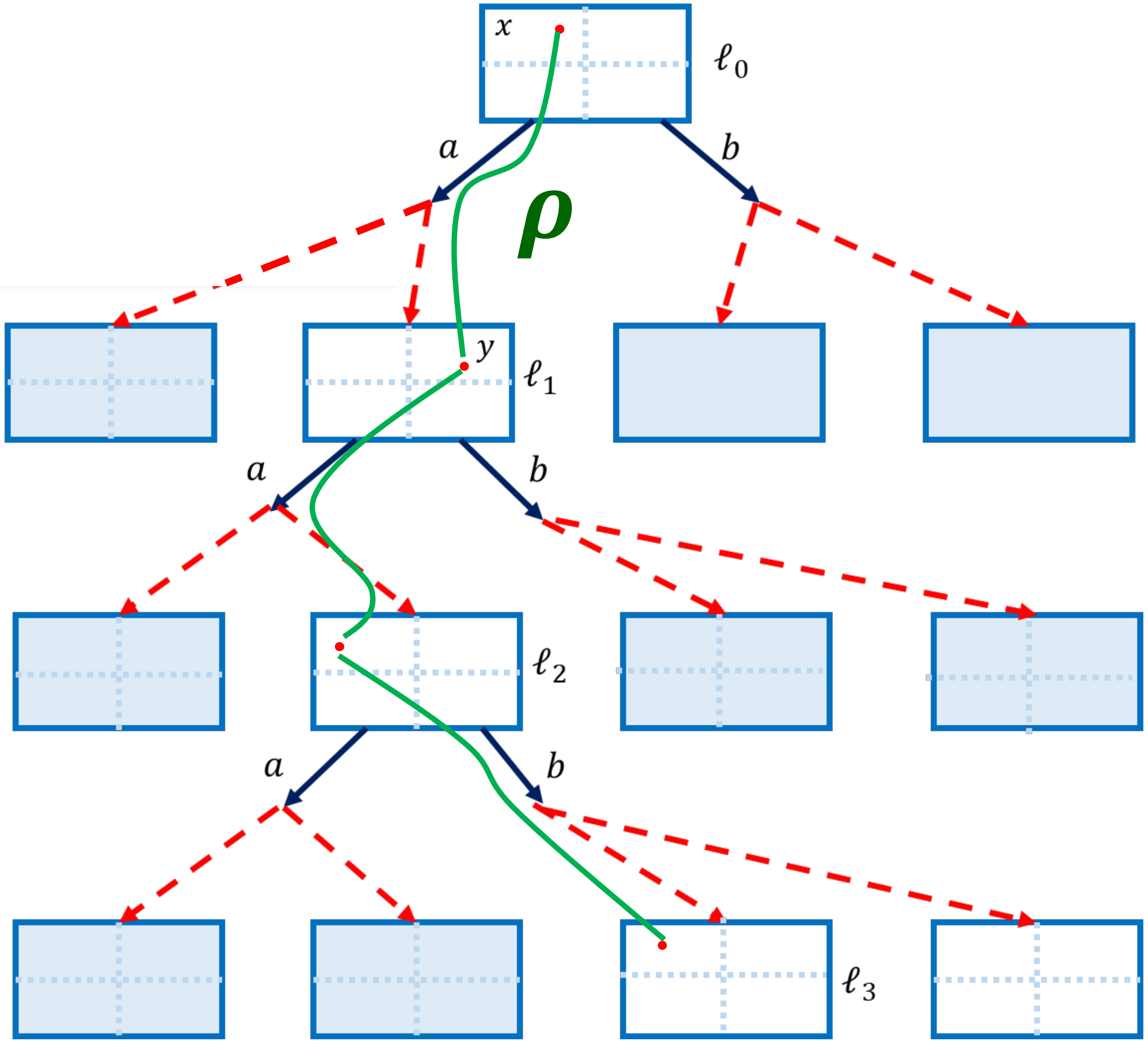
$$\mathbb{E}_A^{min}(G, [s]) \leq \inf_{\sigma} \mathbb{E}_\sigma(G, s) \leq \mathbb{E}_A^{max}(G, [s])$$

Theorem
(Bounded horizon, continuous cost & transition)

Let $A_0 \sqsubseteq A_1 \sqsubseteq \cdots \sqsubseteq A_i \sqsubseteq \cdots$ a be refining sequence "of arbitrary precision". Then

$$\inf_{i \to \infty} \mathbb{E}_{A_i}^{min}(G, [s]_{A_i}) = \inf_{\sigma} \mathbb{E}_\sigma(G, s) = \inf_{i \to \infty} \mathbb{E}_{A_i}^{max}(G, [s]_{A_i})$$

Q– & M–Learning



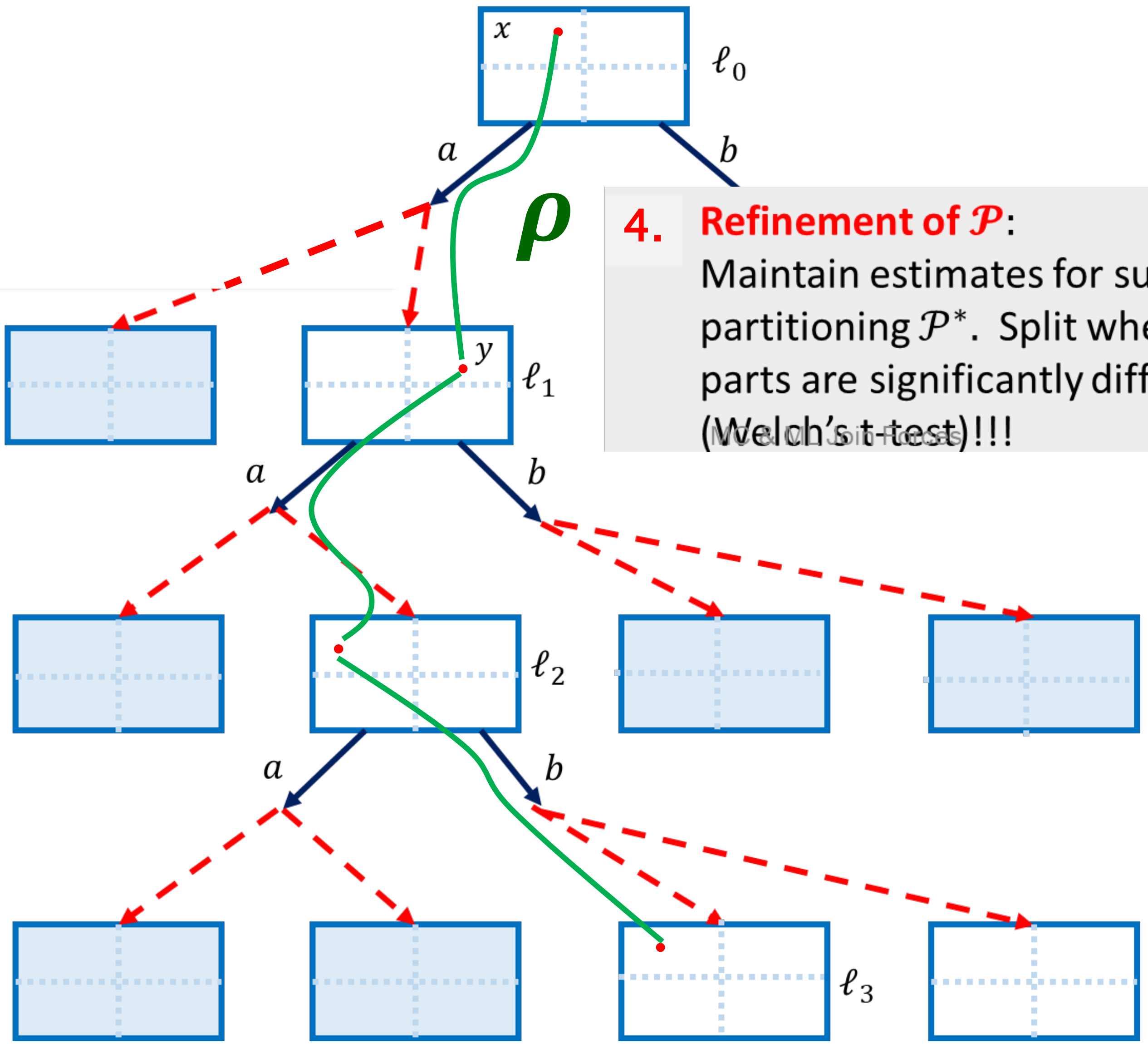
$Q_a^{\ell,x}, Q_b^{\ell,x} : \mathbb{R}_{\geq 0} \quad x \in \mathcal{P}$
$\#(\ell, x) : \mathbb{N}$
$\#\big((\ell, x), \alpha, \{\ell', y)\big) : \mathbb{N}$
$C\big((\ell, x), \alpha, (\ell', y)\big) : \mathbb{R}_{\geq 0}$
$x, y \in \mathcal{P}, \alpha \in \{a, b\}$

REPEATEDLY

1. Generate a random run ρ
   - using $Q_a^{\ell,x}$ and $Q_b^{\ell,x}$ as weights for randomly choosing between a and b in ℓ
   - Using model for stochastic choice.

Q– and M–Learning

$Q_a^{\ell,x}, Q_b^{\ell,x} : \mathbb{R}_{\geq 0} \quad x \in \mathcal{P}$
$\#(\ell, x) : \mathbb{N}$
$\#\big((\ell,x), \alpha, \{\ell', y)\big) : \mathbb{N}$
$C\big((\ell,x), \alpha, (\ell', y)\big) : \mathbb{R}_{\geq 0}$
$x, y \in \mathcal{P}, \alpha \in \{a, b\}$



4. Refinement of $\mathcal{P}$:
Maintain estimates for sub-partitioning $\mathcal{P}^*$. Split when sub-parts are significantly different (Welch's t-test)!!!

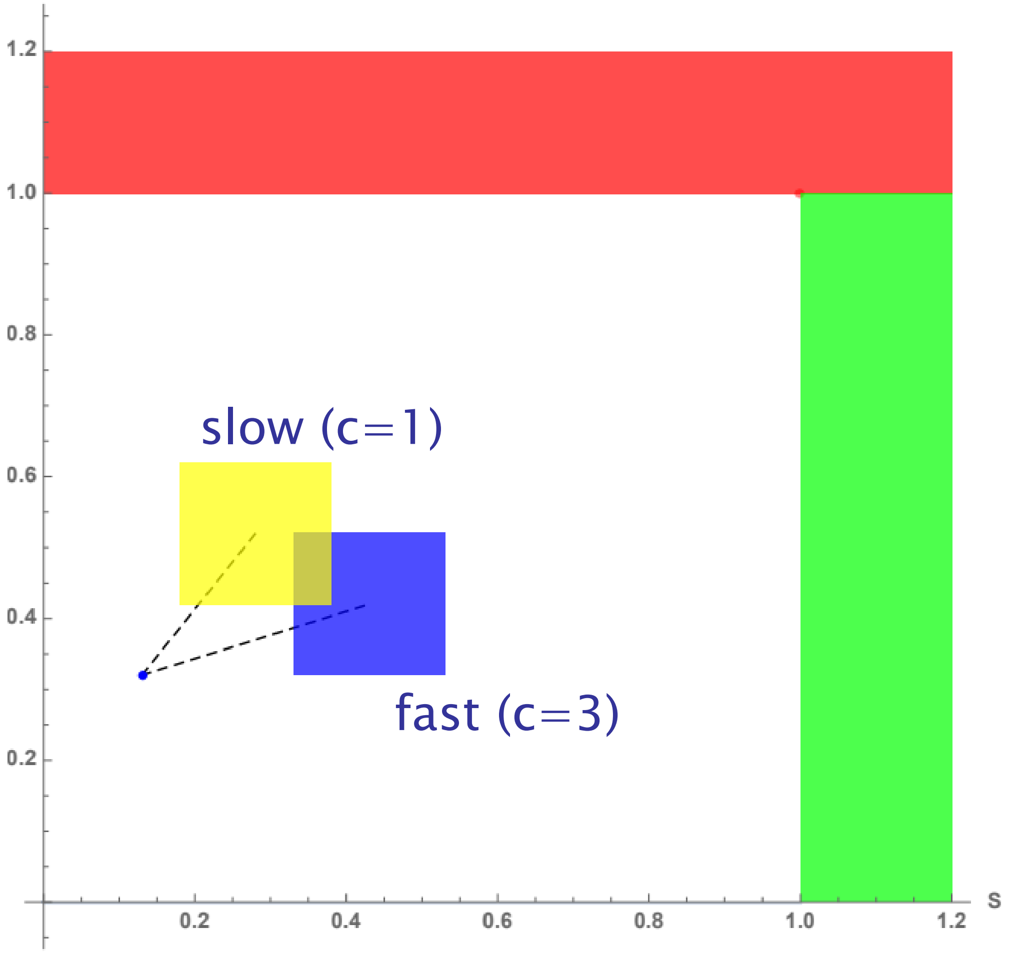
REPEATEDLY

1. Generate a random run ρ
   - using $Q_a^{\ell,x}$ and $Q_b^{\ell,x}$ as weights for randomly choosing between a and b in ℓ
   - Using model for stochastic choice.
2. Update
   $\#(\ell, x)$
   $\#\big((\ell,x), \alpha, (\ell', y)\big)$
   $C\big((\ell,x), \alpha, (\ell', y)\big)$
   along the run
3. Backwards update $Q_a^{\ell,x}$ and $Q_b^{\ell,x}$

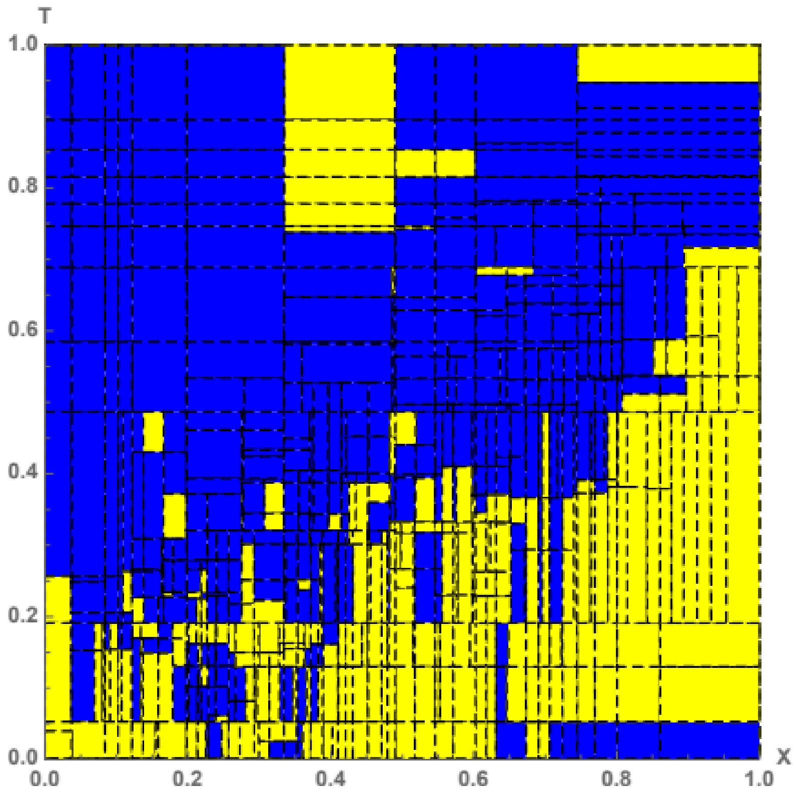
$$Q_a^{\ell_0,x} = \left(1 - \frac{1}{\#(\ell_0,x)}\right) \cdot Q_a^{\ell_0,x} + \frac{1}{\#(\ell_0,x)} \cdot \left(C(s_0, a, s_1) + \min_{\alpha} Q_\alpha^{\ell_1,y}\right)$$

Approximations

Semi–Random Walk

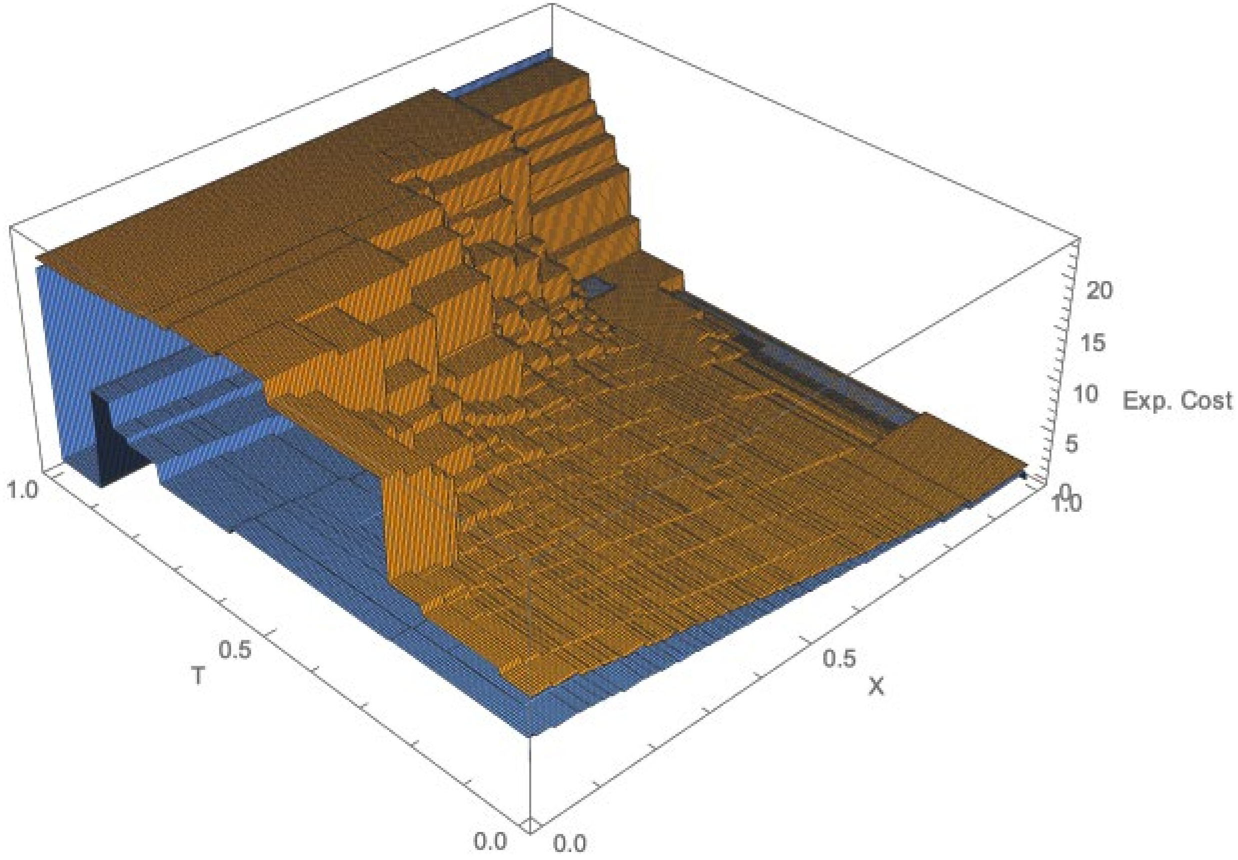


EMDP

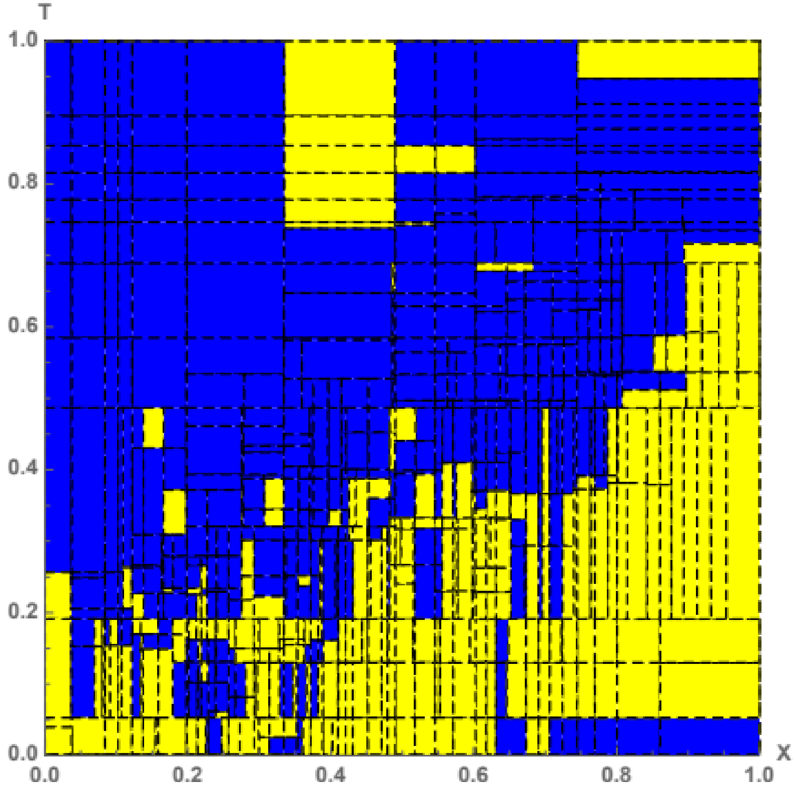


Strategy learned by STRATEGO

Semi-Random Walk

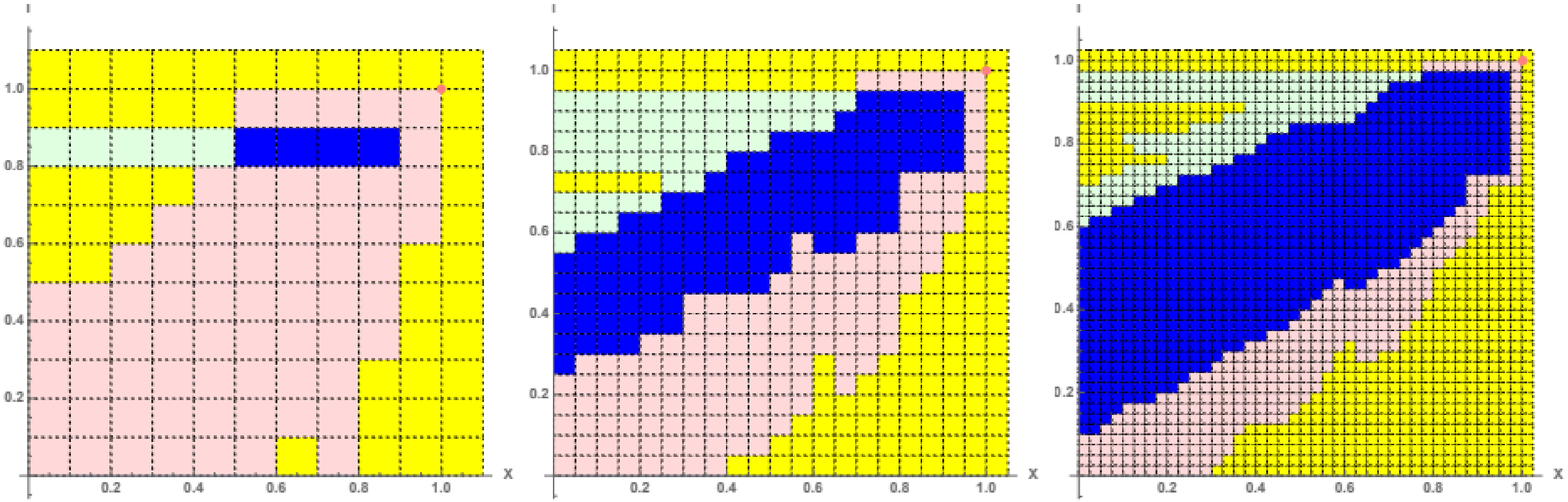


Lower/Upper Cost-bound



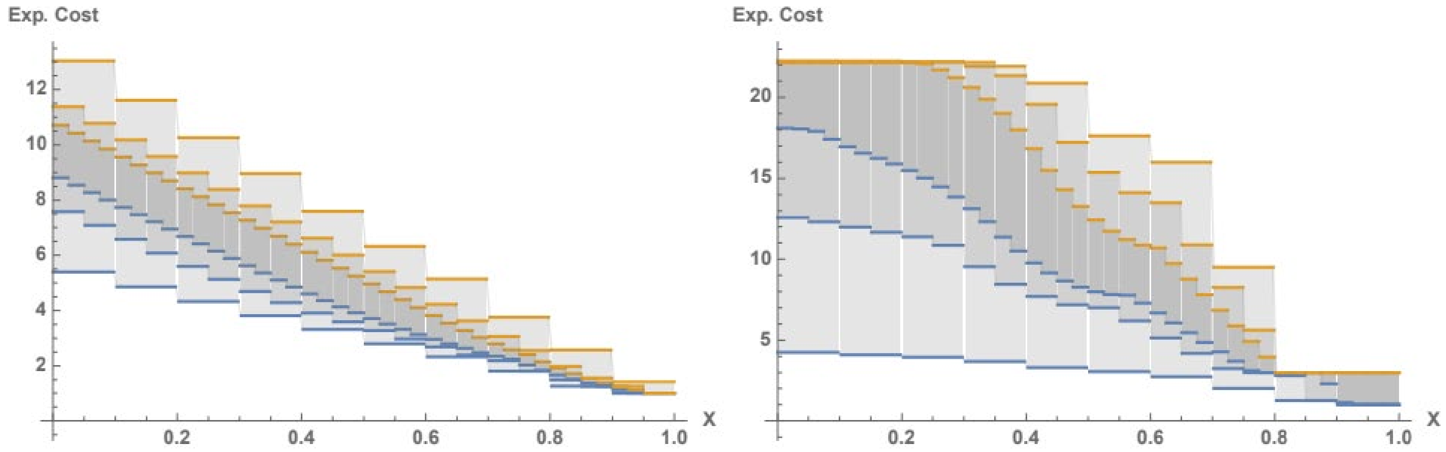
Strategy learned by STRATEGO

Strategies from Successive Partitioning

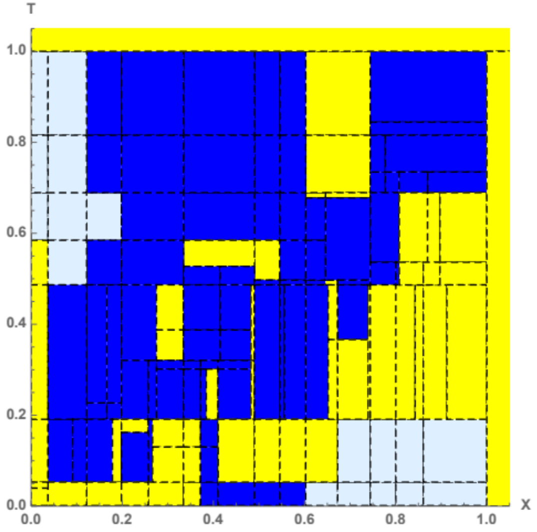


Obtained from lower and upper expected cost approximations

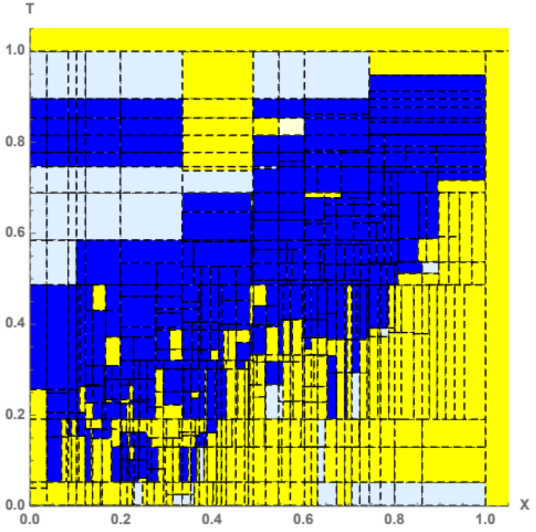
L/U expected costs
for t=0.0, t=0.7
and different Δ.



Strategies from Learning



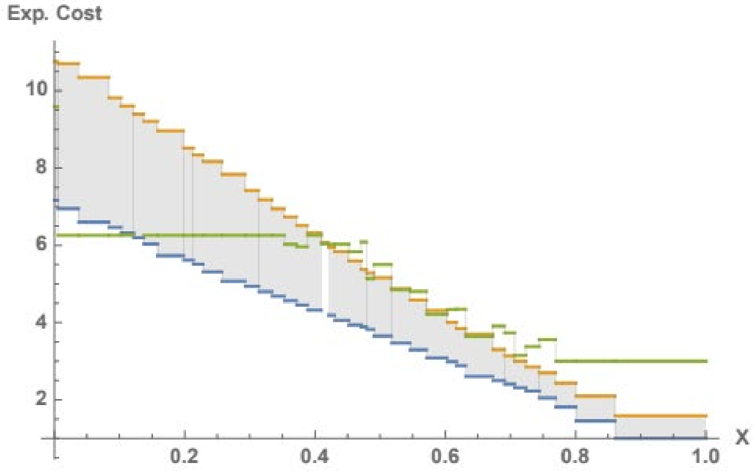
k=27

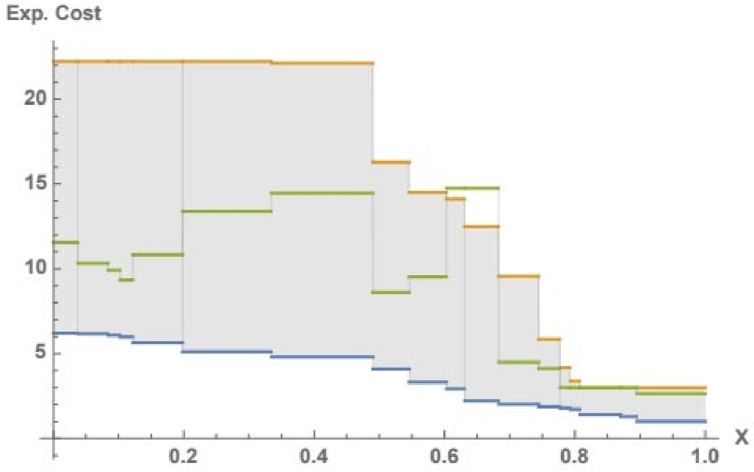


k=205

Expected cost functions for k=205
Along t=0.0, t=0.7.

Yellow: Upper expected cost
Blue: Lower expected cost
Green: Learned expected cost



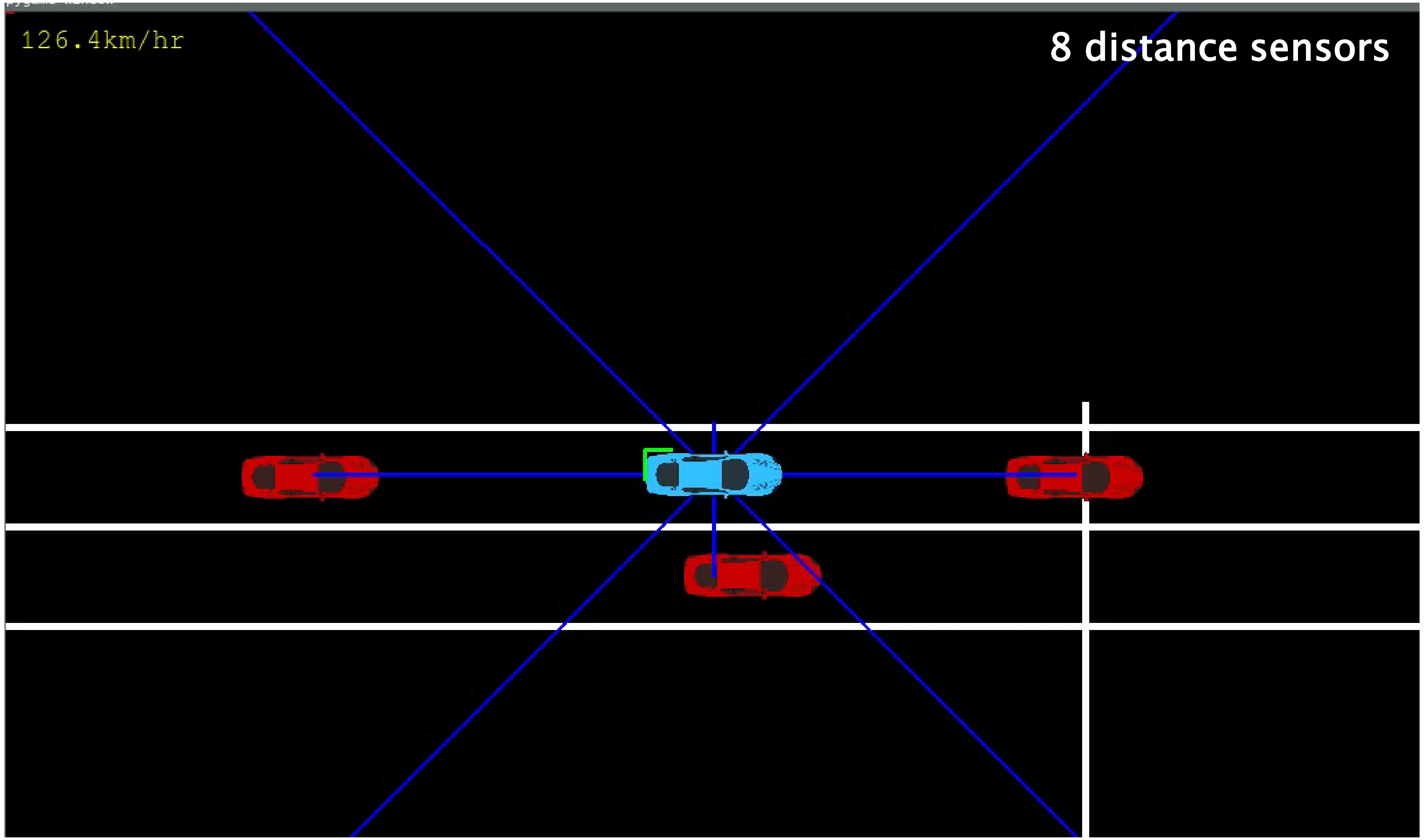


CHALLENGE
Prove convergence of Q-learning for EMDPs

Compact Strategies

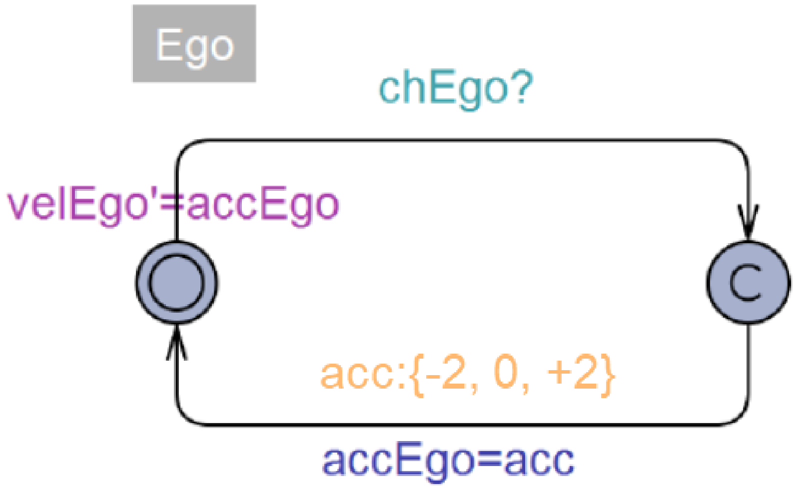
UPPAAL Adaptive Cruise Control

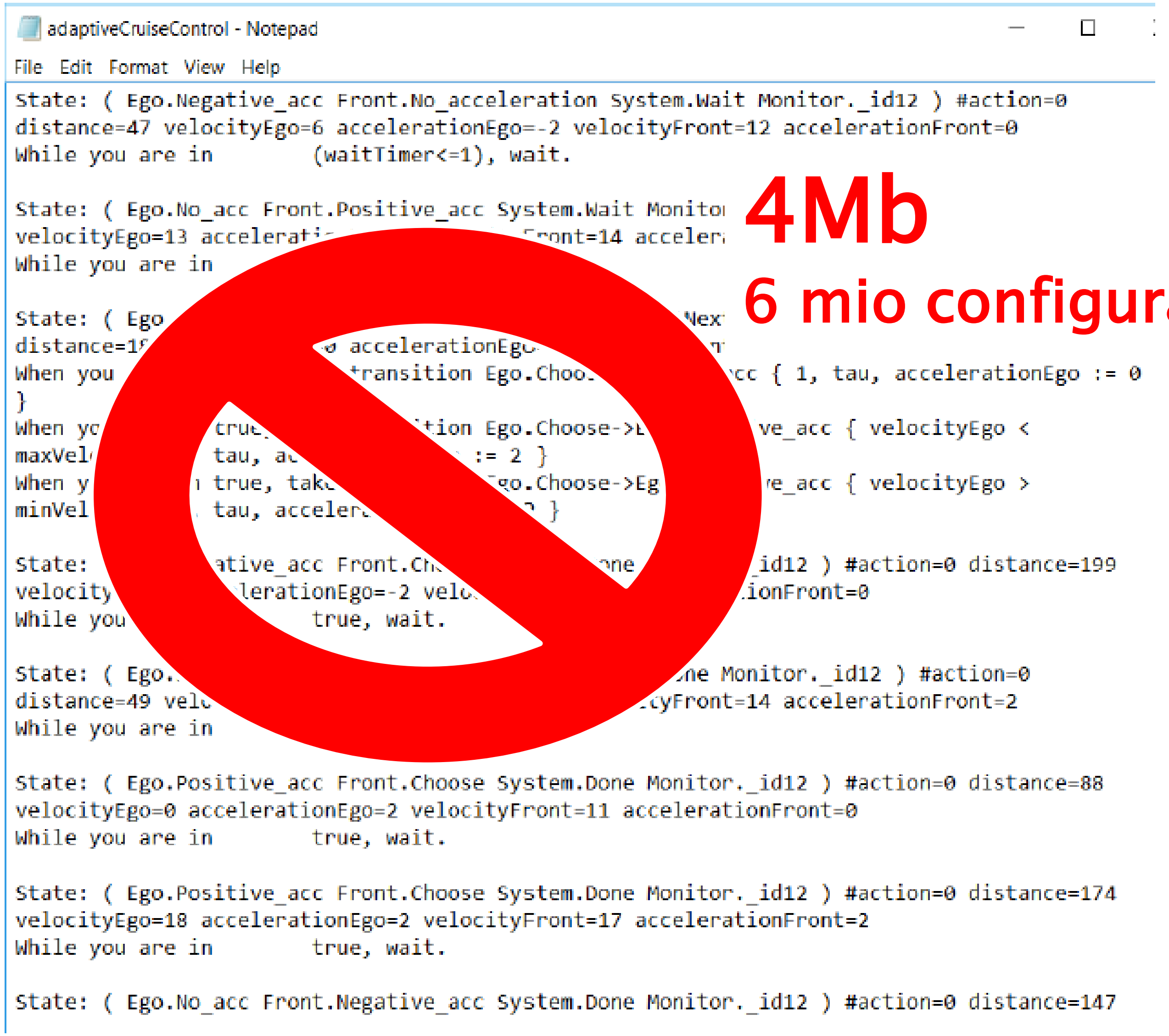
“Optimal” Strategy



Safe and “Optimal” Strategy

Strategy – Explicit

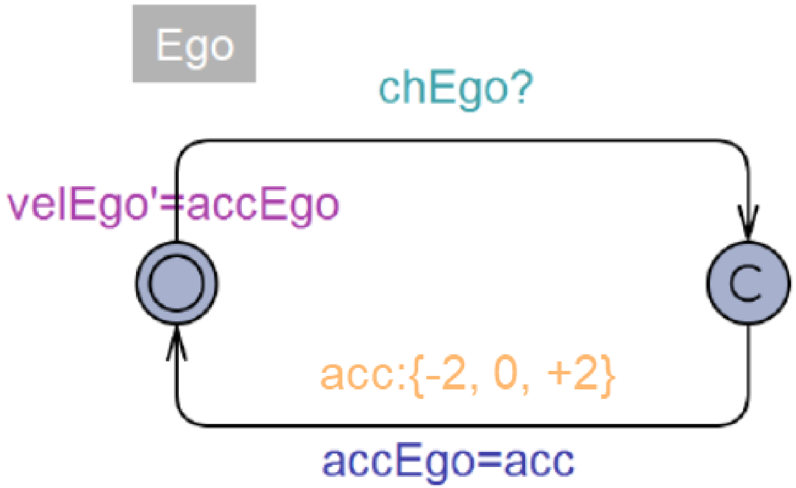




4Mb

6 mio configurations

Strategy – Decision Tree




```
Ego.Choose <= 0: 3 (1481817.0)
Ego.Choose > 0
|  velocityEgo <= -10: 0 (39705.0/18380.0)
|  velocityEgo > -10
|  |  distance <= 200
|  |  |  velocityEgo <= 18
|  |  |  |  velocityEgo <= 12
|  |  |  |  |  distance <= 184
|  |  |  |  |  |  velocityEgo <= 0: 2 (331464.0/20857
|  |  |  |  |  |  velocityEgo > 0
|  |  |  |  |  |  |  distance <= 122
|  |  |  |  |  |  |  |  distance <= 102: 2 (132918.0/8(
|  |  |  |  |  |  |  |  distance > 102
|  |  |  |  |  |  |  |  |  velocityEgo <= 2
|  |  |  |  |  |  |  |  |  |  velocityFront <= 12: 1 (625
|  |  |  |  |  |  |  |  |  |  velocityFront > 12
|  |  |  |  |  |  |  |  |  |  |  velocityFront <= 13: 2 (162.0)
|  |  |  |  |  |  |  |  |  |  |  velocityFront > 13
|  |  |  |  |  |  |  |  |  |  |  |  distance <= 110: 2 (870.0/363.0)
|  |  |  |  |  |  |  |  |  |  |  |  distance > 110
|  |  |  |  |  |  |  |  |  |  |  |  |  velocityFront <= 15
|  |  |  |  |  |  |  |  |  |  |  |  |  |  velocityFront <= 14: 1 (207.0/99.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  velocityFront > 14: 2 (63.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  velocityFront > 15
|  |  |  |  |  |  |  |  |  |  |  |  |  |  distance <= 116
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  velocityFront <= 17: 2 (126.0/54.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  velocityFront > 17: 1 (129.0/39.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  distance > 116: 1 (108.0)
|  |  |  |  |  |  |  |  |  velocityEgo > 2
```

Learning Algorithms for Decision Tree (ID3, D4.5, CART)

→ 65 lines

Jan Kretisnsky, Pranav Ashkot, TUM

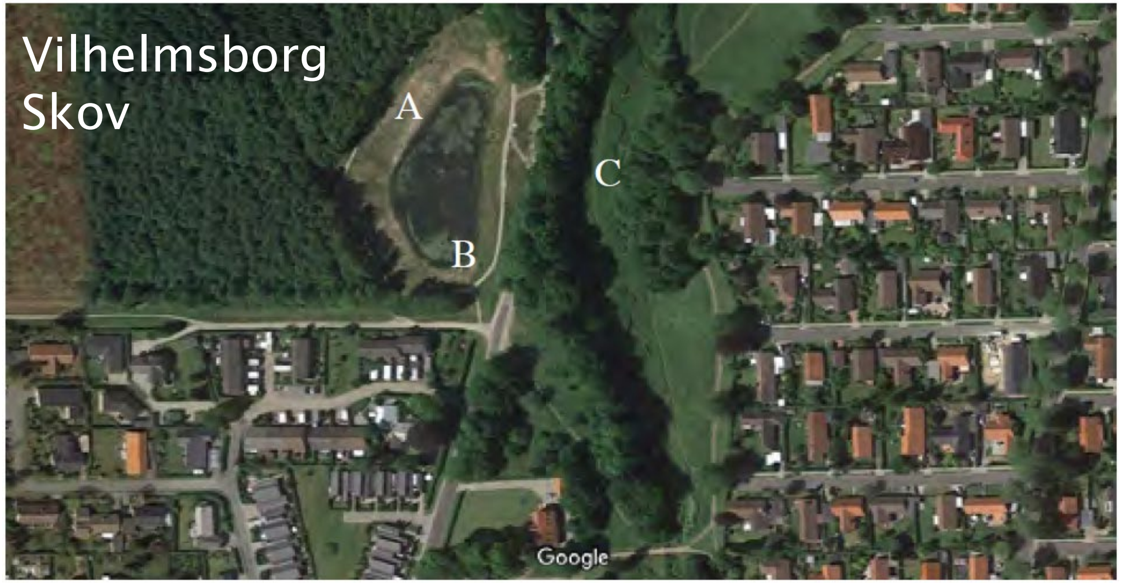
[QEST19] SOS: Synthesis for Hybrid MDP

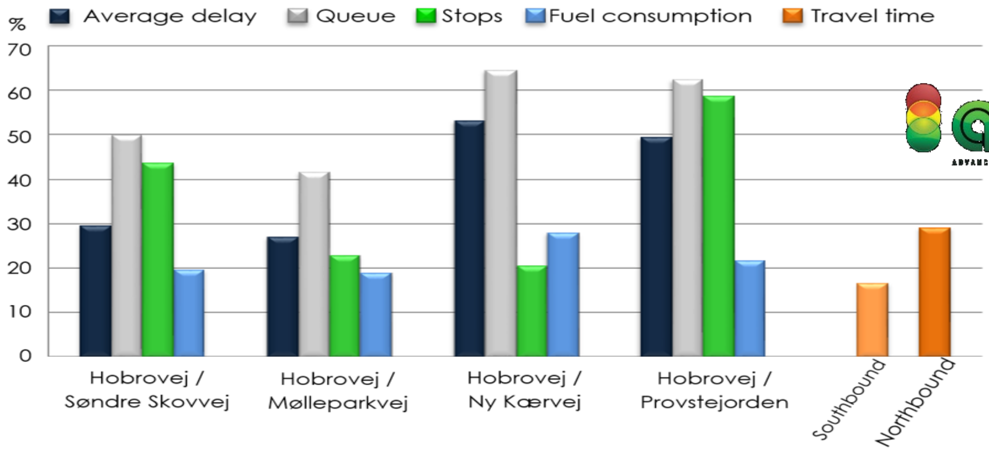
[TACAS21]
dtControl 2.0: Explainable Strategy Representation via Decision Tree Learning Steered by Experts.

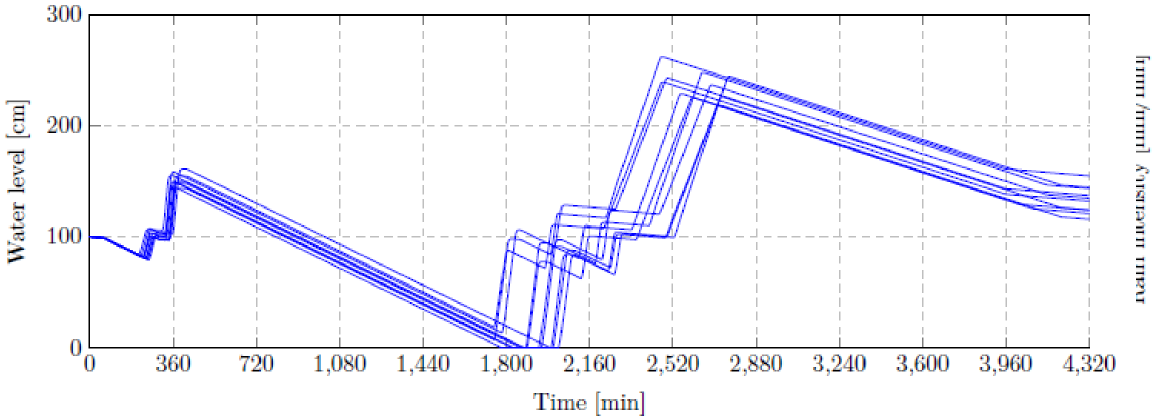
Safe

Applications

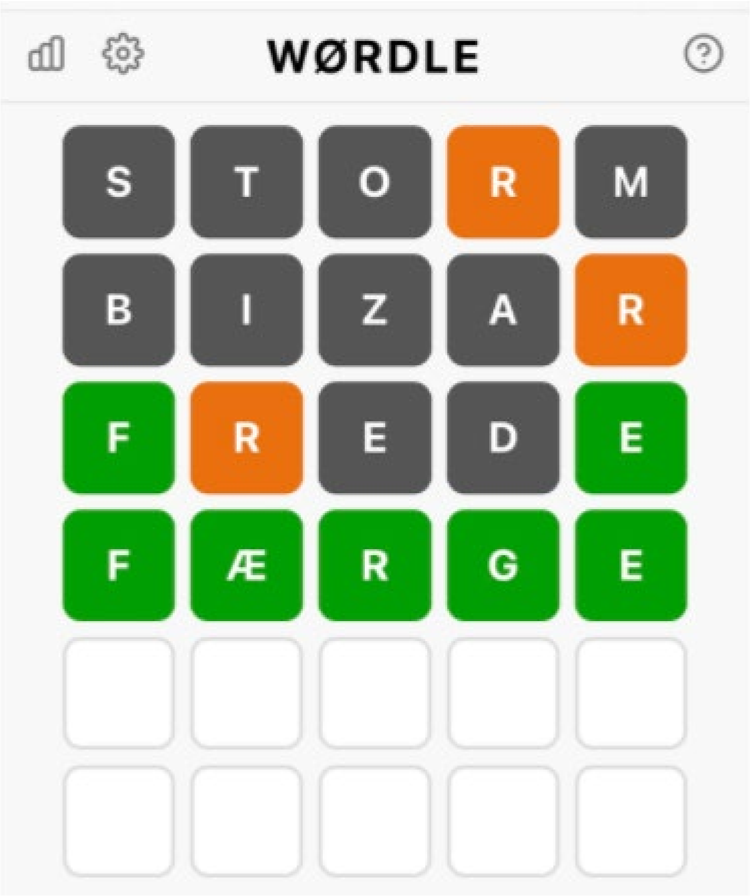
Intellingent Transport -- Smart Water







Wørdle



Sådan spiller du WØRDLE

Gæt dagens ord i 6 forsøg eller mindre.

EKSEMPLER

F L Æ S K

Bogstavet **Æ** indgår i ordet på den korrekte plads.

S O V E R

Bogstavet **S** indgår i ordet, men på den *forkerte* plads.

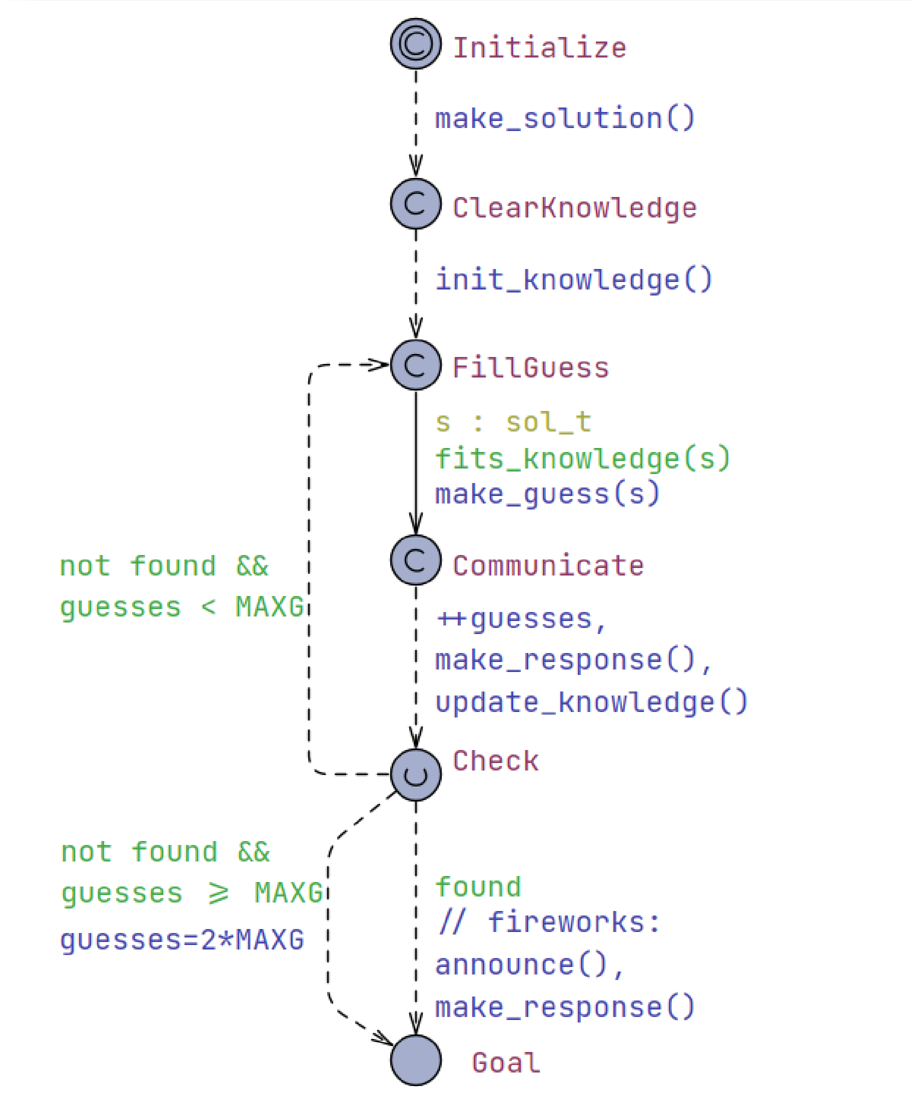
K O N T O

Bogstavet **T** indgår *ikke* ordet.

WØRDLE in UPPAAL Stratego

7771 words

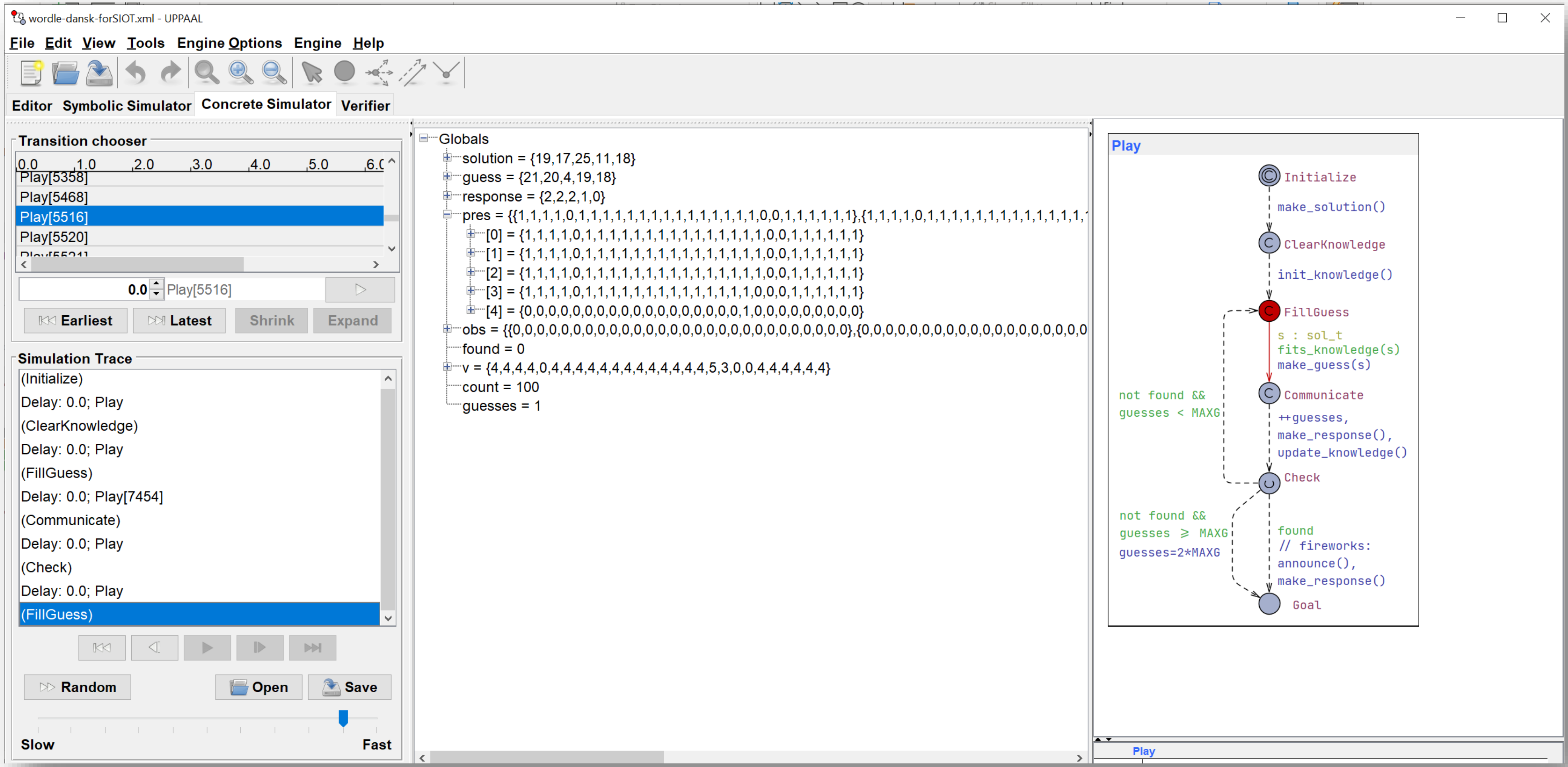
```
const col_t words[NWORDS][pos_t] = {
    {0,3,17,25,19},   //    0: ADRÆT
    {0,5,10,14,12},   //    1: AFKOM
    {0,5,18,11,27},   //    2: AFSLÅ
    {0,5,18,19,27},   //    3: AFSTÅ
    {0,10,17,24,11},  //    4: AKRYL
    {0,10,19,8,4},    //    5: AKTIE
    {0,11,0,17,12},   //    6: ALARM
    {0,11,1,20,4},    //    7: ALBUE
    {0,11,12,4,13},   //    8: ALMEN
    {0,12,15,20,11},  //    9: AMPUL
    {0,13,6,17,4},    //   10: ANGRE
    {0,13,10,4,11},   //   11: ANKEL
    {0,13,10,4,17},   //   12: ANKER
    {0,15,15,4,11},   //   13: APPEL
    {0,17,14,12,0},   //   14: AROMA
    {0,17,21,25,21},  //   15: ARVÆV
    {0,19,11,0,18},   //   16: ATLAS
    {1,4,5,17,8},     //   17: BEFRI
    {1,4,19,14,13},   //   18: BETON
    {1,8,9,14,1},     //   19: BIJOB
    {1,8,11,0,6},     //   20: BILAG
    {1,14,18,0,19},   //   21: BOSAT
    {2,8,3,4,17},     //   22: CIDER
    {3,8,11,11,4},    //   23: DILLE
    {3,8,17,10,4},    //   24: DIRKE
    {3,8,18,10,14},   //   25: DISKO
    {3,14,21,13,4},   //   26: DOVNE
    {3,17,8,5,19},    //   27: DRIFT
```



WØRDLE in UPPAAL Stratego

7771 words

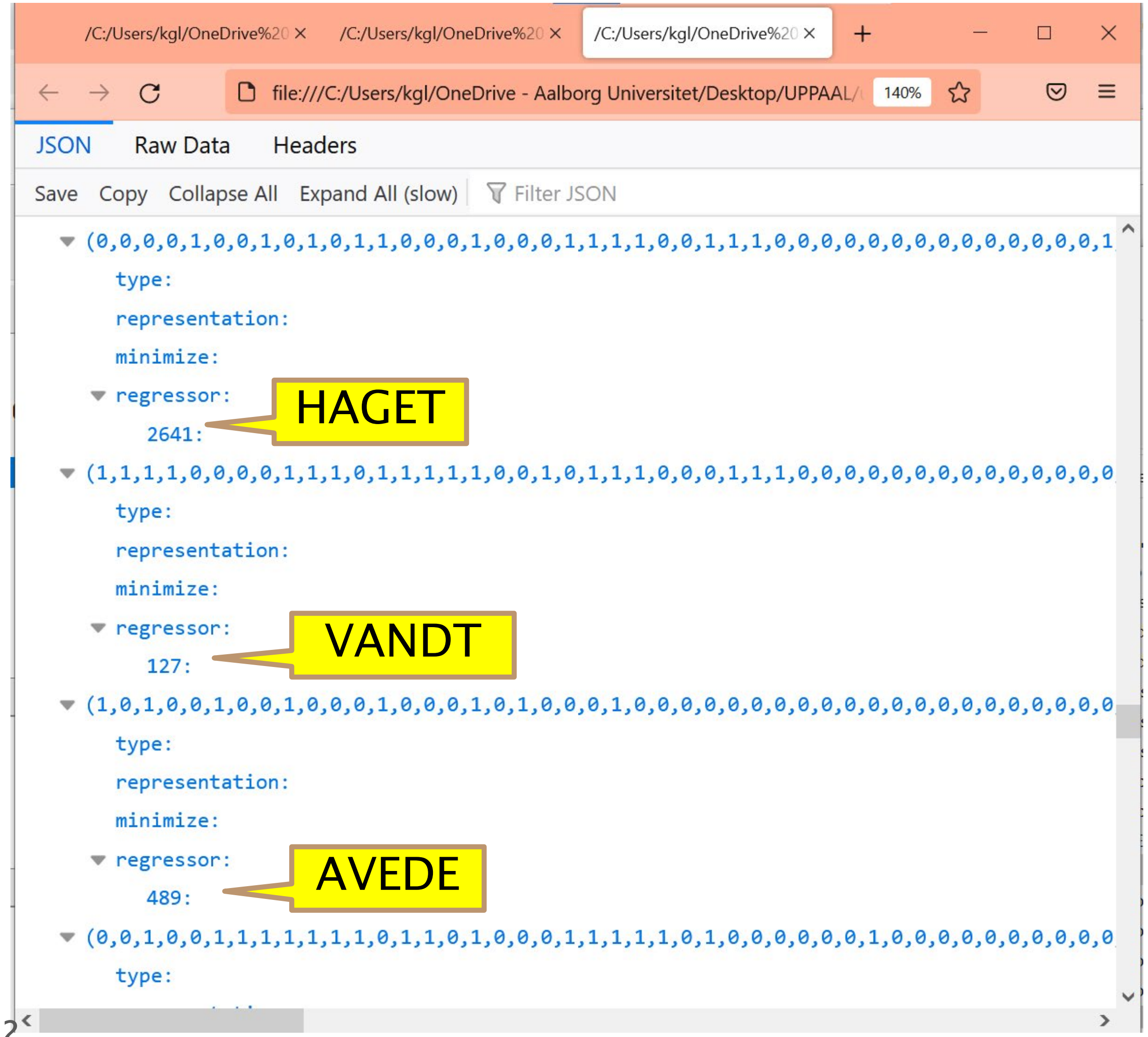
```
const col_t words[NWORDS][pos_t] = {
    {0,3,17,25,19},   //    0: ADRÆT
    {0,5,10,14,12},   //    1: AFKOM
    {0,5,18,11,27},   //    2: AFSLÅ
    {0,5,18,19,27},   //    3: AFSTÅ
    {0,10,17,24,11},  //    4: AKRYL
    {0,10,19,8,4},    //    5: AKTIE
    {0,11,0,17,12},   //    6: ALARM
    {0,11,1,20,4},    //    7: ALBUE
    {0,11,12,4,13},   //    8: ALMEN
    {0,12,15,20,11},  //    9: AMPUL
    {0,13,6,17,4},    //   10: ANGRE
    {0,13,10,4,11},   //   11: ANKEL
    {0,13,10,4,17},   //   12: ANKER
    {0,15,15,4,11},   //   13: APPEL
    {0,17,14,12,0},   //   14: AROMA
    {0,17,21,25,21},  //   15: ARVÆV
    {0,19,11,0,18},   //   16: ATLAS
    {1,4,5,17,8},     //   17: BEFRI
    {1,4,19,14,13},   //   18: BETON
    {1,8,9,14,1},     //   19: BIJOB
    {1,8,11,0,6},     //   20: BILAG
    {1,14,18,0,19},   //   21: BOSAT
    {2,8,3,4,17},     //   22: CIDER
    {3,8,11,11,4},    //   23: DILLE
    {3,8,17,10,4},    //   24: DIRKE
    {3,8,18,10,14},   //   25: DISKO
    {3,14,21,13,4},   //   26: DOVNE
    {3,17,8,5,19},    //   27: DRIFT
```



RESULTS

Random guessing legal word that fits knowledge
6.56 guesses

Stratego (Reinforcement Learning)
4.47 guesses




```
strategy s = minE(guesses) [#<=MAXS] {

obs[0][0],obs[0][1],obs[0][2],obs[0][3],obs[0][4],obs[0][5],obs
[0][6],obs[0][7],obs[0][8],obs[0][9],obs[0][10],obs[0][11],obs[
0][12],obs[0][13],obs[0][14],obs[0][15],obs[0][16],obs[0][17],
obs[0][18],obs[0][19],obs[0][20],obs[0][21],obs[0][22],obs[0][
23],obs[0][24],obs[0][25],obs[0][26],obs[0][27],obs[0][27],

obs[1][0],obs[1][1],obs[1][2],obs[1][3],obs[1][4],obs[1][5],obs
[1][6],obs[1][7],obs[1][8],obs[1][9],obs[1][10],obs[1][11],obs[
1][12],obs[1][13],obs[1][14],obs[1][15],obs[1][16],obs[1][17],
obs[1][18],obs[1][19],obs[1][20],obs[1][21],obs[1][22],obs[1][
23],obs[1][24],obs[1][25],obs[1][26],obs[1][27],obs[1][27],

obs[2][0],obs[2][1],obs[2][2],obs[2][3],obs[2][4],obs[2][5],obs
[2][6],obs[2][7],obs[2][8],obs[2][9],obs[2][10],obs[2][11],obs[
2][12],obs[2][13],obs[2][14],obs[2][15],obs[2][16],obs[2][17],
obs[2][18],obs[2][19],obs[2][20],obs[2][21],obs[2][22],obs[2][
23],obs[2][24],obs[2][25],obs[2][26],obs[2][27],obs[2][27],

obs[3][0],obs[3][1],obs[3][2],obs[3][3],obs[3][4],obs[3][5],obs
[3][6],obs[3][7],obs[3][8],obs[3][9],obs[3][10],obs[3][11],obs[
3][12],obs[3][13],obs[3][14],obs[3][15],obs[3][16],obs[3][17],
obs[3][18],obs[3][19],obs[3][20],obs[3][21],obs[3][22],obs[3][
23],obs[3][24],obs[3][25],obs[3][26],obs[3][27],obs[3][27],

obs[4][0],obs[4][1],obs[4][2],obs[4][3],obs[4][4],obs[4][5],obs
[4][6],obs[4][7],obs[4][8],obs[4][9],obs[4][10],obs[4][11],obs[
4][12],obs[4][13],obs[4][14],obs[4][15],obs[4][16],obs[4][17],
obs[4][18],obs[4][19],obs[4][20],obs[4][21],obs[4][22],obs[4][
23],obs[4][24],obs[4][25],obs[4][26],obs[4][27],obs[4][27]
} -> {}: <> Play.Goal
```

More Synthesis & Ongoing Research

- Convergence of Q-learning for IMDP.
- Partial Observability
- Learning strategy profiles for composite systems
- Online/Offline
- ..

VILLUM FONDEN