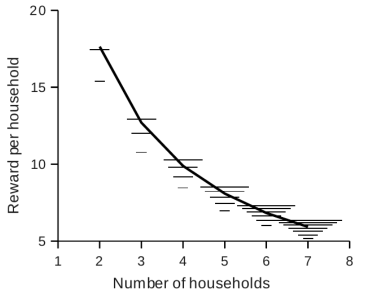
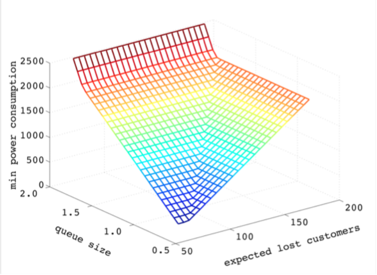
# PRISM-games 3.0

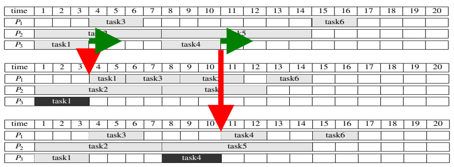
## Stochastic Game Verification

with

## Concurrency, Equilibria and Time



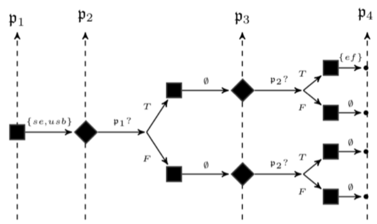




| time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ |  |  |  | task3 |  |  |  |  |  |  |  |  |  |  | task6 |  |  |  |  |  |
| $P_2$ | task2 |  |  |  |  |  |  | task5 |  |  |  |  |  |  |  |  |  |  |  |  |
| $P_3$ | task1 |  |  |  |  |  |  | task4 |  |  |  |  |  |  |  |  |  |  |  |  |

| time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ |  |  |  | task1 |  | task3 |  |  | task4 |  |  |  | task6 |  |  |  |  |  |  |  |
| $P_2$ | task2 |  |  |  |  |  |  | task5 |  |  |  |  |  |  |  |  |  |  |  |  |
| $P_3$ | task1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ |  |  |  | task3 |  |  |  |  |  |  | task4 |  |  |  | task6 |  |  |  |  |  |
| $P_2$ | task2 |  |  |  |  |  |  | task5 |  |  |  |  |  |  |  |  |  |  |  |  |
| $P_3$ | task1 |  |  |  |  |  |  | task4 |  |  |  |  |  |  |  |  |  |  |  |  |



# PRISM-games 3.0

## Stochastic Game Verification

with

## Concurrency, Equilibria and Time

Marta Kwiatkowska, Gethin Norman, Dave Parker, Gabriel Santos

University
of
Oxford

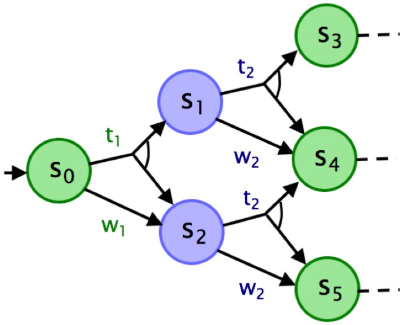
University
of
Glasgow

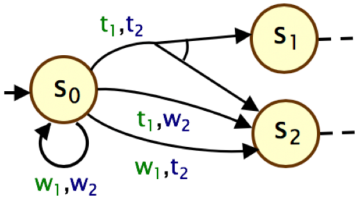
University
of
Birmingham

ERC
Advanced Grant
FUN2MODEL

# Stochastic Games

- Stochastic games:
  - nondeterminism: adversarial, controllers, …
  - probability: randomisation, failures, noise, …
  - players: competing/collaborating components
  - strategies: rational decisions made by players
  - costs (resources) & rewards (incentives)





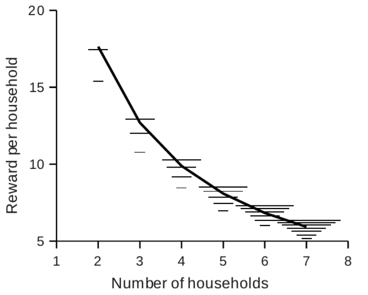
- Applications:
  - computer security, network/communication protocols, algorithms for distributed consensus, energy management, autonomous robotics, auctions, …

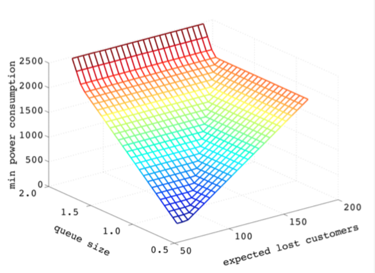
- Challenge:
  - how to design these systems correctly?
  - complex interactions between features

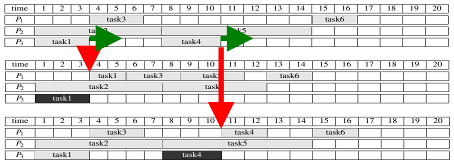
# PRISM-games

- PRISM-games
  - modelling and analysis of stochastic games
  - automated verification or synthesis of strategies with quantitative guarantees

- Example specification in rPATL
  - $\langle\langle \text{robot}_1 \rangle\rangle\ \text{P}_{\geq 0.95}\ [\ \text{F}^{\leq 10}\ \text{goal}_1\ ]$
  - “robot 1 has a strategy to ensure that,
    with probability at least 0.95, it reaches its goal in 10 steps,
    regardless of the strategies of other robots”



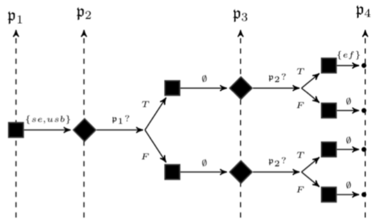




| time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | | | | task3 | | | | | | | | | | | task6 | | | | | |
| $P_2$ | task2 | | | | | | | task5 | | | | | | | | | | | | |
| $P_3$ | task1 | | | | | | | task4 | | | | | | | | | | | | |

| time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | | | | task1 | | task3 | | | task | | | | task6 | | | | | | | |
| $P_2$ | task2 | | | | | | | task | | | | | | | | | | | | |
| $P_3$ | task1 | | | | | | | | | | | | | | | | | | | |

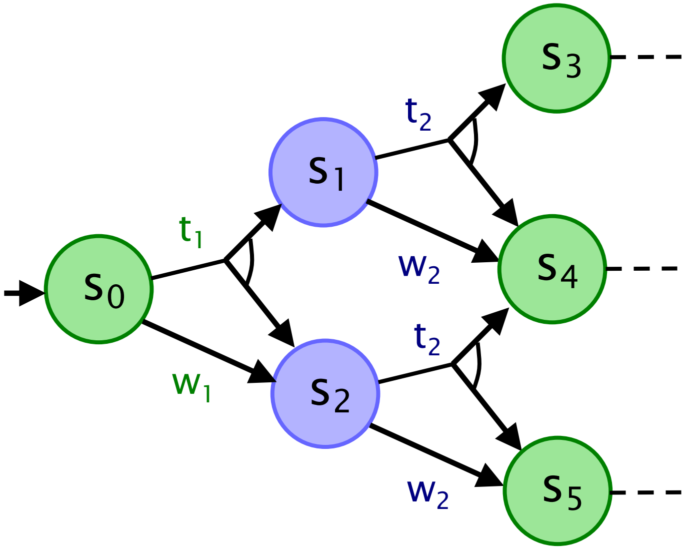
| time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | | | | task3 | | | | | | | task4 | | | | task6 | | | | | |
| $P_2$ | task2 | | | | | | | task5 | | | | | | | | | | | | |
| $P_3$ | task1 | | | | | | | task4 | | | | | | | | | | | | |

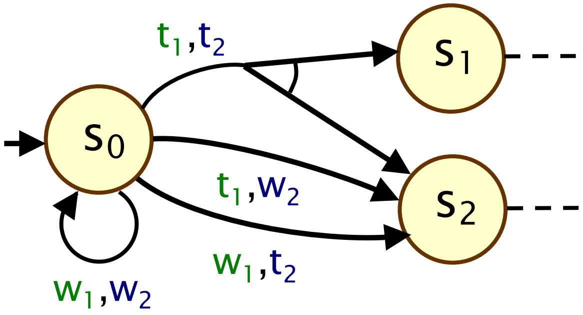


# Concurrent stochastic games

new

Turn-based stochastic games (TSGs) → Concurrent stochastic games (CSGs)

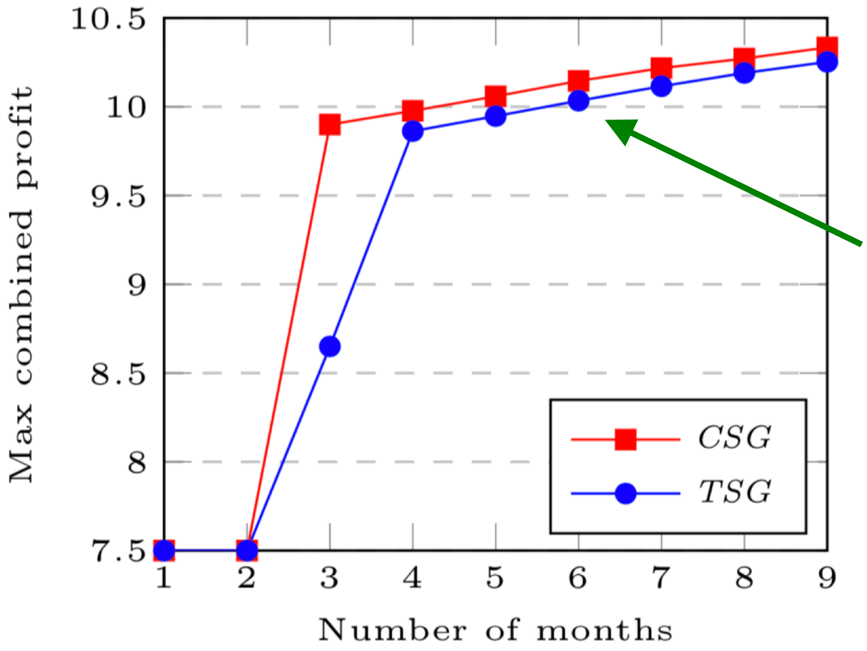




- Players make concurrent, independent decisions

# Concurrent stochastic games

Example: Maximising expected investor profit in a futures market



Too pessimistic: unrealistic strategy for adversary

# CSGs in PRISM-games 3.0

```
csg
player p1 user1 endplayer
player p2 user2 endplayer
// Users (senders)
module user1
        s1 : [0..1] init 0; // has player 1 sent?
        e1 : [0..emax] init emax; // energy level of player 1
        [w1] true -> (s1'=0); // wait
        [t1]  e1>0 -> (s1'=c’ ? 0 : 1) & (e1'=e1-1); // transmit
endmodule
module user2 = user1 [ s1=s2, e1=e2, w1=w2, t1=t2 ] endmodule
// Channel: used to compute joint probability distribution for transmission failure
module channel
        c : bool init false; // is there a collision?
        [t1,w2] true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 1 transmits
        [w1,t2] true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 2 transmits
        [t1,t2]  true -> q2 : (c'=false) + (1-q2) : (c'=true); // both users transmit
endmodule
```

# CSGs in PRISM-games 3.0

```
csg
player p1 user1 endplayer
player p2 user2 endplayer
// Users (senders)
module user1
        s1 : [0..1] init 0; // has player 1 sent?
        e1 : [0..emax] init emax; // energy level of player 1
        [w1] true -> (s1'=0); // wait
        [t1]  e1>0 -> (s1'=c’ ? 0 : 1) & (e1'=e1-1); // transmit
endmodule
module user2 = user1 [ s1=s2, e1=e2, w1=w2, t1=t2 ] endmodule
// Channel: used to compute joint probability distribution for transmission failure
module channel
        c : bool init false; // is there a collision?
        [t1,w2] true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 1 transmits
        [w1,t2] true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 2 transmits
        [t1,t2]  true -> q2 : (c'=false) + (1-q2) : (c'=true); // both users transmit
endmodule
```

Player = one or more modules

Variable updates can refer to other variables updated simultaneously

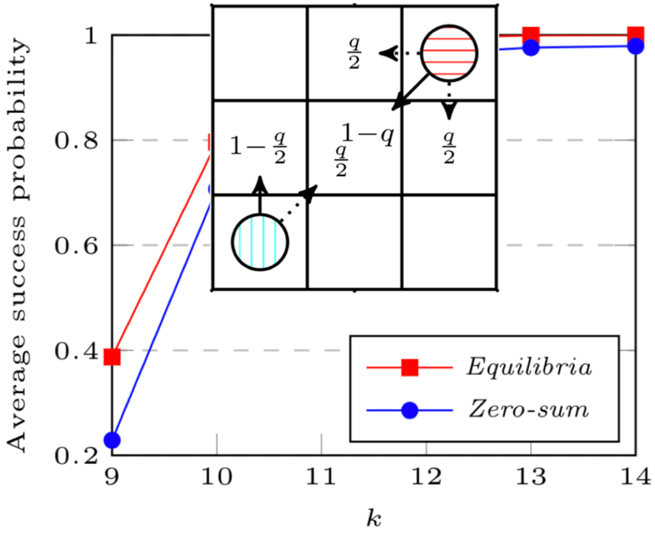
Action lists used to specify synchronisation

# Equilibria-based properties

new

Zero-sum properties → Equilibria-based properties

- Example: multi-robot coordination
  - $\langle\langle robot_1 \rangle\rangle_{max=?}$ P [ $F^{\leq k}$ $goal_1$ ]
  - maximise probability of robot 1 reaching its goal, regardless of robot 2
  - $\langle\langle robot_1 : robot_2 \rangle\rangle_{max=?}$ (P [ $F^{\leq k}$ $goal_1$ ]+P [$F^{\leq k}$ $goal_2$])
  - find strategies where robots 1 & 2 have no incentive to change actions and maximise joint goal probability



Social-welfare optimal Nash equilibrium

# PRISM-games 3.0

new

- Probabilistic timed games (turn-based)

new

- 10 new/expanded case studies
  - multi-robot coordination, network trust models, Aloha, intrusion detection, public good games, ...

- More information at:
  - prismmodelchecker.org/games/
  - documentation, examples, case studies, papers
  - downloads: 🍎 🐧 ⊞ + CAV'20 artefact VM

GitHub

- Open source (GPLv2)
  - github.com/prismmodelchecker/prism-games