

Advanced Formal Tools

PRISM: Probabilistic Model Checking

Created and actively maintained by:

- [Dave Parker](#) (University of Oxford)
- [Gethin Norman](#) (University of Glasgow)
- [Marta Kwiatkowska](#) (University of Oxford)

Tansen RAHMAN

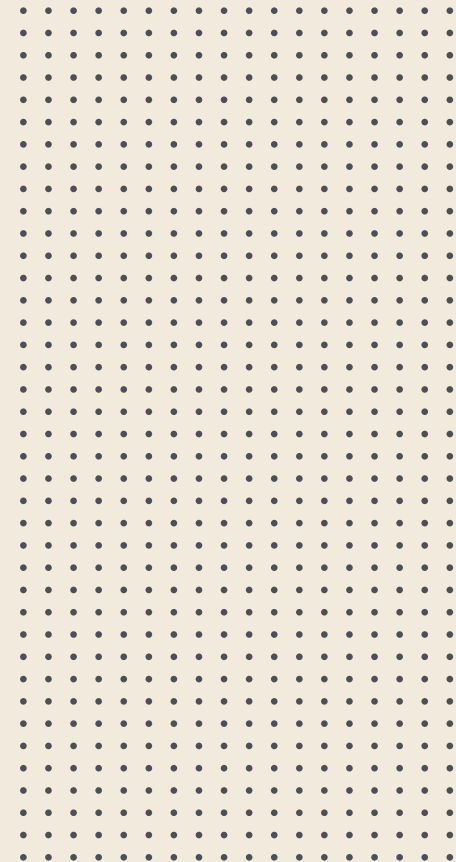
Tansen.Rahman@etu.unige.ch

Stéphane Nguyen

Stephane.Nguyen@etu.unige.ch

University of Geneva

March 28th 2023





Overview



1

Current Status

- Understand basic usage (modeling+analysis)
- Found limitations and have thought about solutions
- Found an interesting case study to base our own from

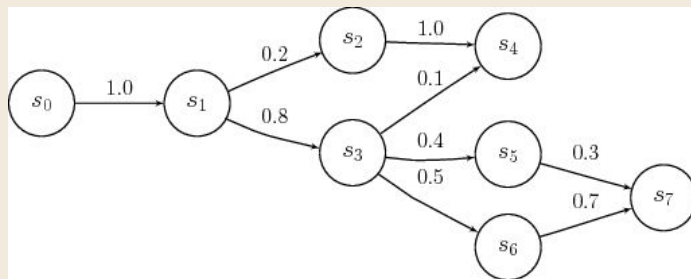
2

Goals

- Settle on a case after looking at our ideas
- Find a methodology to get results
- Implement and analyze them
- Write a report

Reminder

- DTMC:



$$p(s'|s)$$



Waiting for fate...

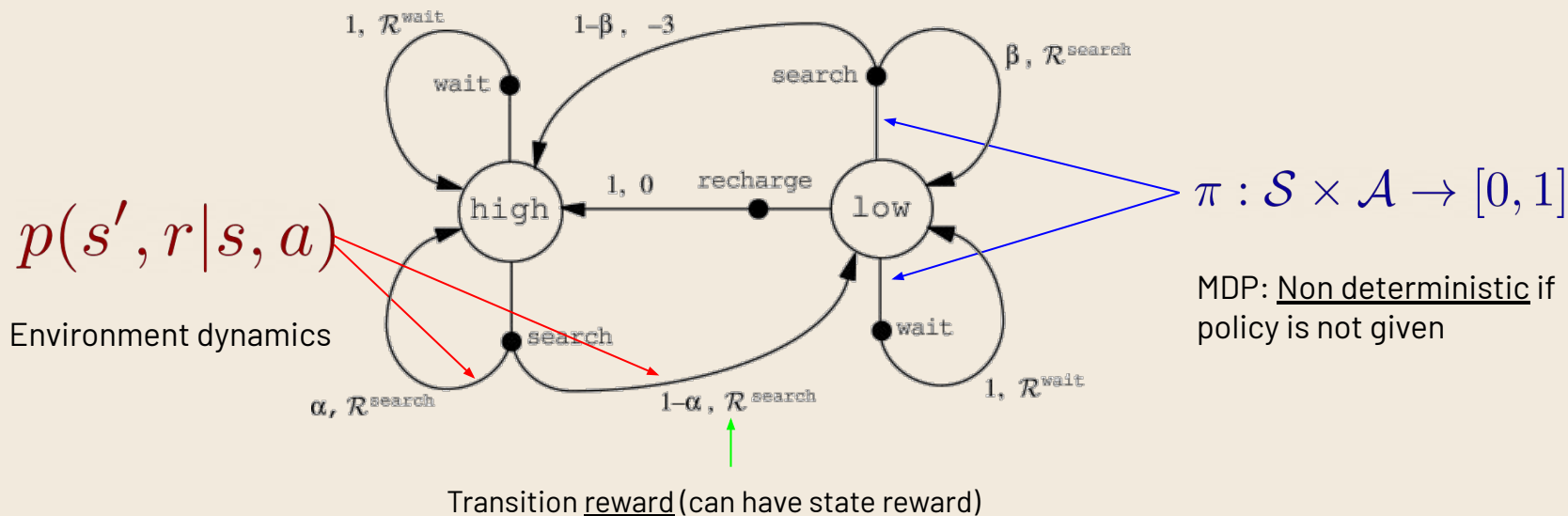
- Extension with **rewards** (see MRP)
- Properties**
 - How good is it to be in a given state ?
 - Proba. to reach a given state ?
 - etc.

You're a **passive** entity with no power. **Want more control ?**

Single player

- **Finite state space, action space MDP:**

- DTMC + rewards + decisions



- **Analyze properties:**

- Maximum expected reward ?
- etc.

Two or more players

- **TSG: Turned-based Stochastic Game:**
MDP + players can only play in some states + one player per state.
- Extension of PRISM, PRISM-Games can analyze **stochastic multiplayer games**, including TSGs.



	Heads	Tails
Heads	+1, -1	-1, +1
Tails	-1, +1	+1, -1

Matching pennies

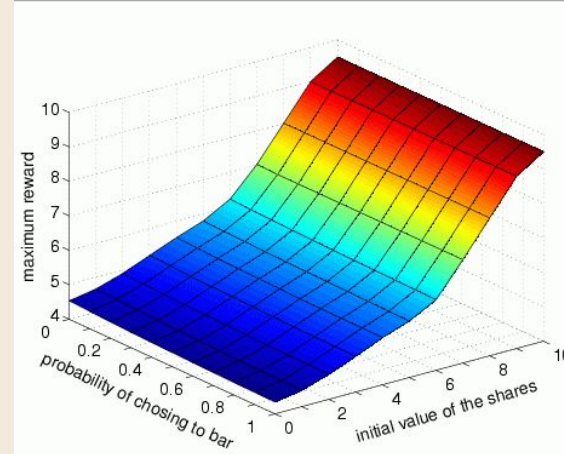
Limitations

- **Primitive language:**
Supports ranges: $i: [1...10]$;
But not lists nor for-loops
- **Limited features in UI:**
No support for 2D surface plots

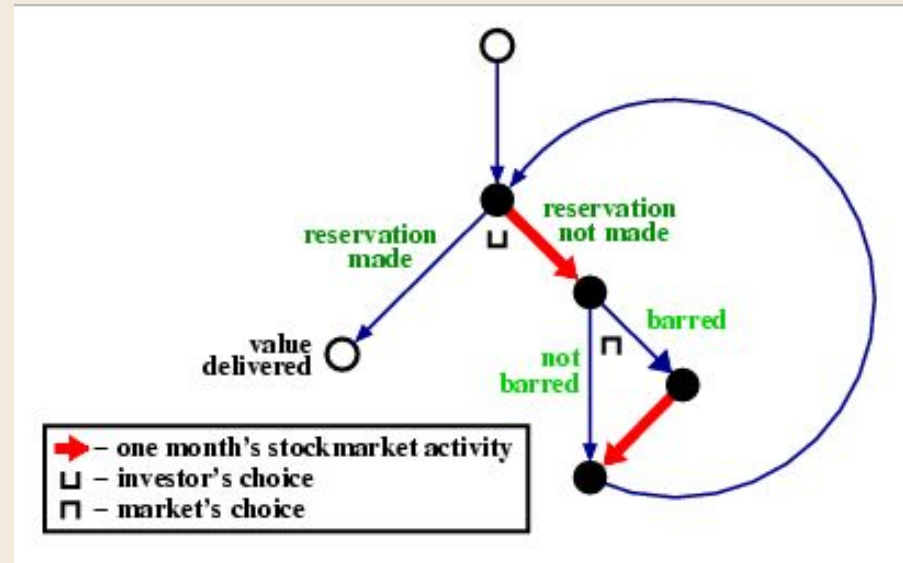
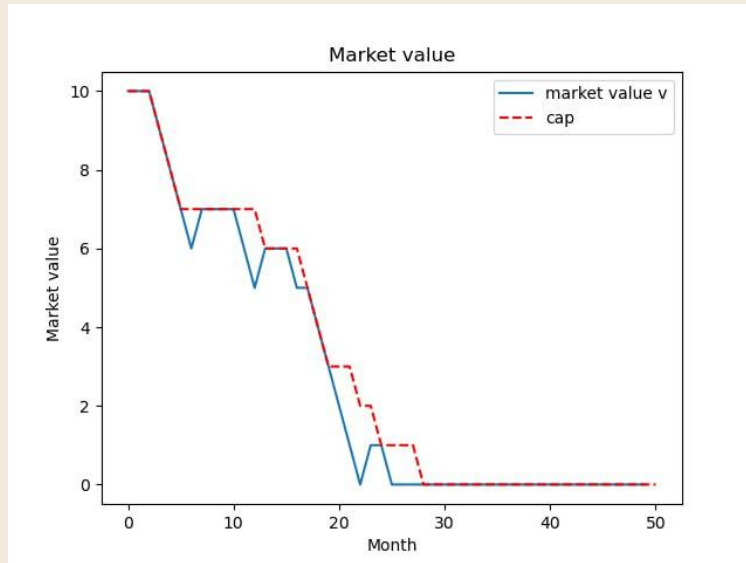
=> Just use Python!

- **Others:**
State-space explosion
Variable ordering
Simulation considers uniform strategy,
Calculations consider optimal strategy

```
[reply] receiver=2 & y1=0 -> 1/(maxr+1) : (receiver'=3) & (y1'=0)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*1)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*2)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*3)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*4)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*5)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*6)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*7)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*8)
+ 1/(maxr+1) : (receiver'=3) & (y1'=2*9)
```



Case study: Futures Investor





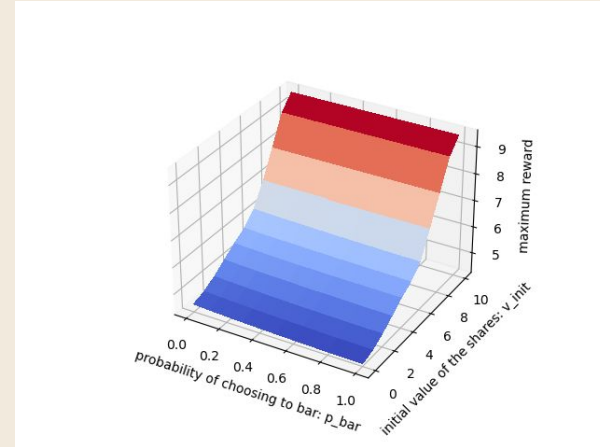
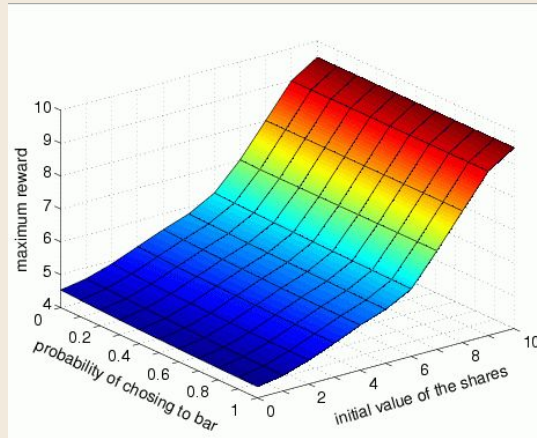
Possible Changes



- Change end state to a time limit, allow several investments
- Have a limit on the number of investments
- Add time reward (money now is worth more than money later)
- Introduce another investor, they take turn buying/selling to each other
- Implement an actual future (both actors settle on a price now, for later)

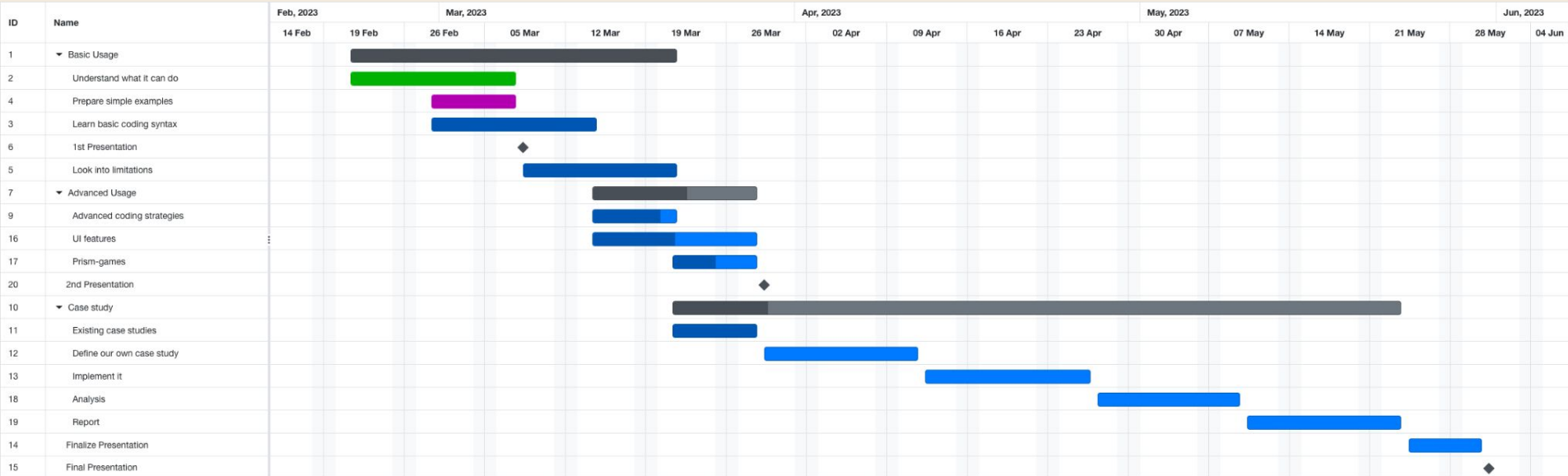
Methodology

- Reproducing original case study's results by modifying the PRISM model:



- Use as inspiration to analyze our own and to look for interesting properties

Project Progression





Conclusion