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

May 30th 2023

Overview

1. Background and Theory
2. PRISM Usage and Limitations
3. Case Study
4. Future Work
5. Conclusion

Background and theory

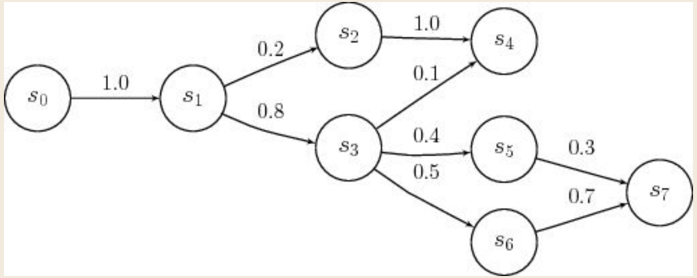
Probabilistic models

- Discrete Time Markov Chain
- Markov Decision Processes

Discrete Time Markov Chain

States: $\mathcal{S} = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$

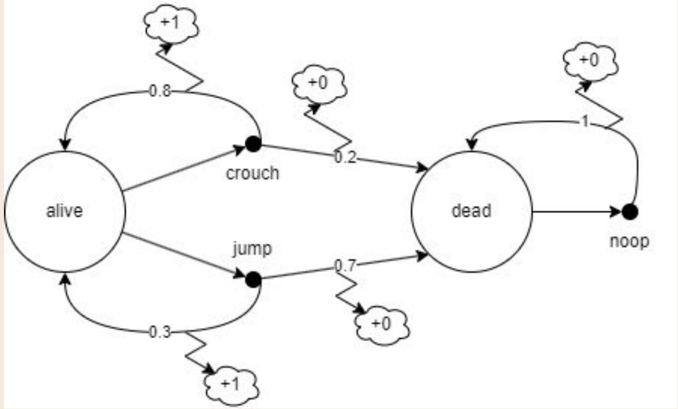
Example transition: $s_0 \rightarrow s_1$, probability of 1, thus $p(s_1|s_0) = 1$



Markov Decision Process

Model

- **States:**
  - alive (init. state)
  - dead
- **Actions:**
  - crouch
  - jump

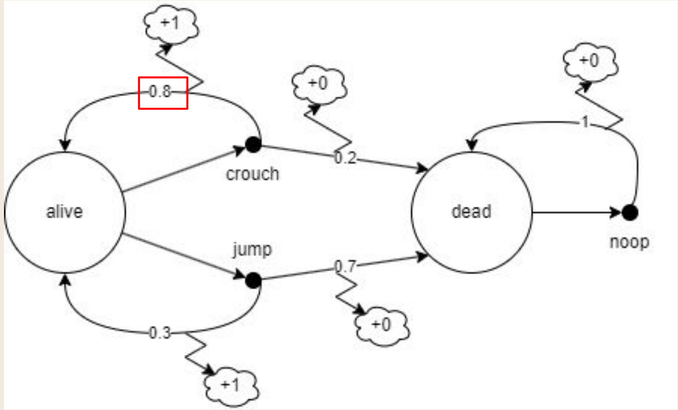


Non-deterministic, choose crouch or jump from alive ?

Markov Decision Process

Model

- **States:**
  - alive (init. state)
  - dead
- **Actions:**
  - crouch
  - jump



Non-deterministic, choose crouch or jump from alive ?

- **Env. dynamics:**

  p(s', r | s, a)

  Ex: $P[S_{t+1} = \text{alive}, R_{t+1} = 1 | S_t = \text{alive}, A_t = \text{crouch}] = 0.8$

Markov Decision Process

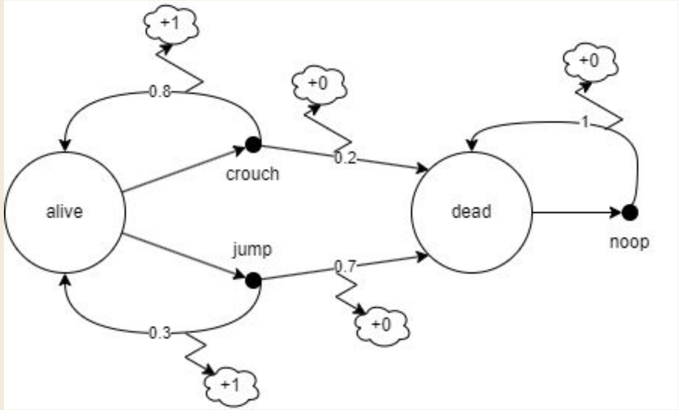
Policy & objective function

- **Policy/Strategy:**

$$\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$$

- **Expected cumulative reward:**

$$E(\pi) = \mathbb{E}_\pi \left[\sum_{i=1}^{\infty} R_i \middle| S_0 = s_0 \right]$$



Markov Decision Process

Policy & objective function

- Policy/Strategy:

$$\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

- Expected cumulative reward:

$$E(\pi) = \mathbb{E}_\pi \left[\sum_{i=1}^{\infty} R_i \middle| S_0 = s_0 \right]$$

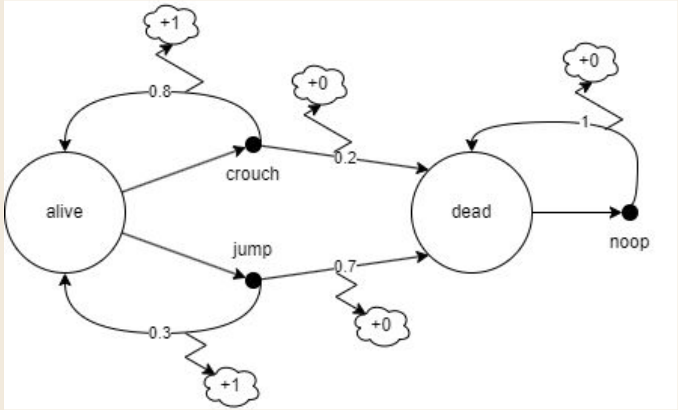
- Best policy:

$$\pi^* = \text{argmax}_\pi E(\pi)$$

- Max. expected cumulative reward:

$$E(\pi^*)$$

Rmax=? [F "end"] in PRISM with "end" : s=0 (dead)



Markov Decision Process

Policy & objective function

- Policy/Strategy:

$$\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$$

- Expected cumulative reward:

$$E(\pi) = \mathbb{E}_\pi \left[\sum_{i=1}^{\infty} R_i \middle| S_0 = s_0 \right]$$

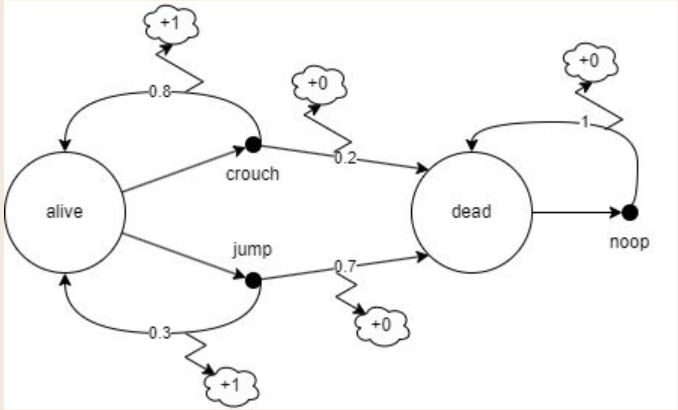
- Best policy:

$$\pi^* = \operatorname{argmax}_\pi E(\pi)$$

- Max. expected cumulative reward:

$$E(\pi^*)$$

Rmax=? [F "end"] in PRISM with "end" : s=0 (dead)



What about worst policy ?
Min. expected cumulative reward ?

Markov Decision Process

Reward structures

- Expected cumulative reward:

$$E(\pi) = \mathbb{E}_\pi \left[\sum_{i=1}^{\infty} R_i \middle| S_0 = s_0 \right]$$

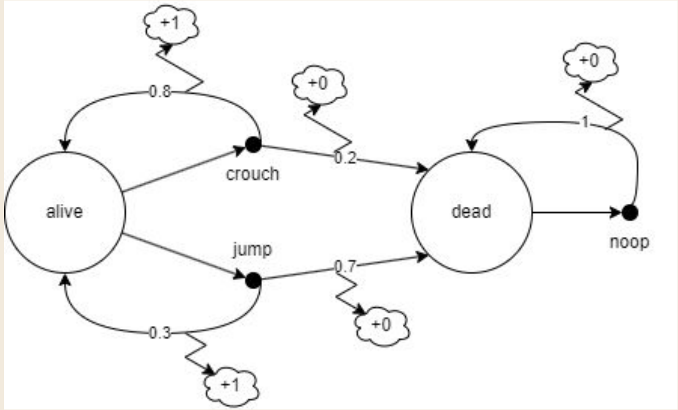
- Best policy:

$$\pi^* = \mathrm{argmax}_\pi E(\pi)$$

- Max. expected cumulative reward:

$$E(\pi^*)$$

- **Different reward distributions: reward structures**
  - Enable verification of different props.
  - Implicit verification of many MDPs.
  - Ex: # of steps



Prism Usage and Limitations

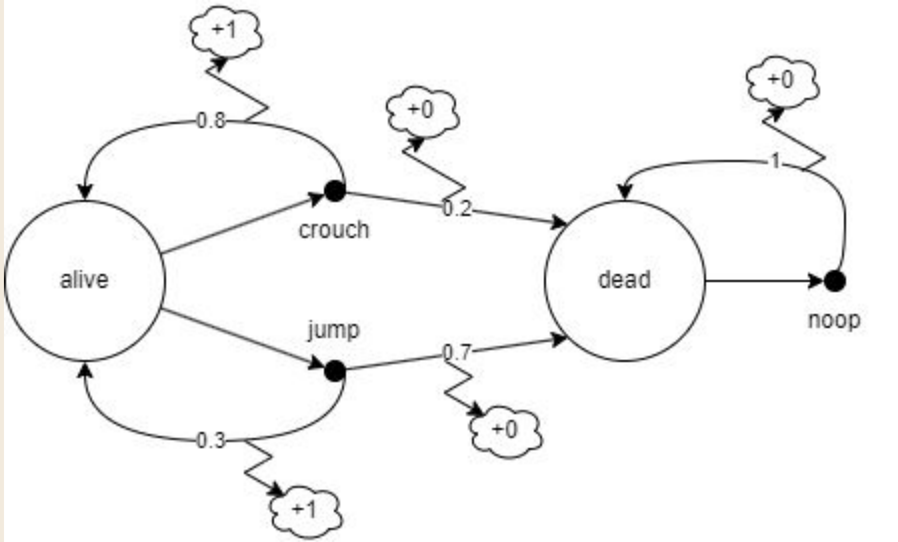
Usage

- Coding
- Analysis

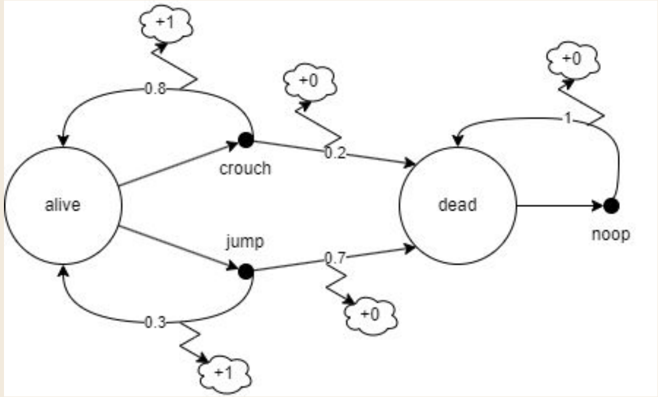
Limitations

PRISM Representation

```
mdp


module mdp_example
s : [0..1] init 1; // Alive initially

// non-deterministic choice from s =1
// jump vs crouch
[jump] (s=1)  → 0.3: (s'=1) + 0.7: (s'=0);
[crouch] (s=1) → 0.8: (s'=1) + 0.2: (s'0);
[noop] (s=0) → (s'=0); // absorbing state

endmodule

rewards
    (s=1) : 1;  // reward for staying alive
    (s=0) : 0;  // died
endrewards

rewards "steps"
    true : 1;
endrewards
```



PRISM Model Analysis



Simulator
- Sample/simulate paths from probabilities
- Manually choose transitions

Properties
- PRISM’s logics subsumes PCTL and others:
  - Extension with rewards
  - Extension with quantitative prop.

- Example of CTL vs PCTL for a given strategy:

CTL	EG s=1	Exist path where always alive
PCTL	P>0[G<=10 s=1]	Exist a path where alive for at least 10 steps
PRISM	P=?[G<=10 s=1]	PRISM can get the value (quant. prop.)

PRISM Limitations

DevOps

- No CTRL+F
- Missing standard graph tools, need to export data
- No for-loops

Modeling

- No negative rewards
- Can’t assign probability distribution over rewards
- Simulation uses a uniform strategy to resolve non-determinism

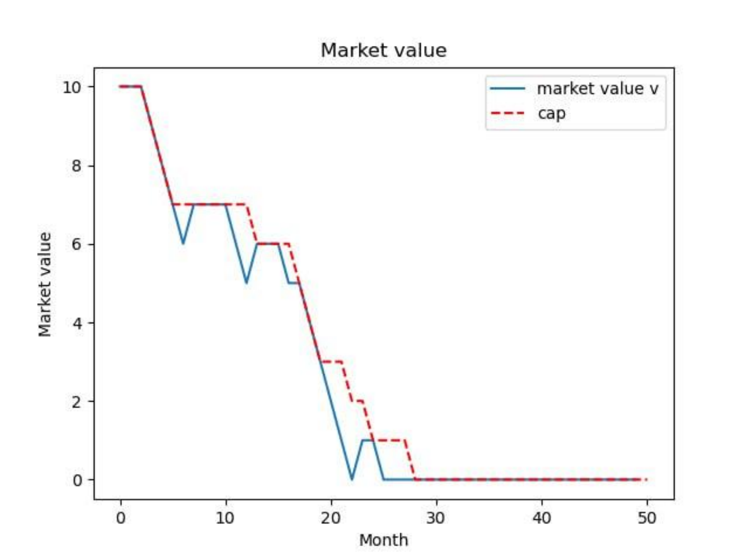
Case study: Market Bidding Investor

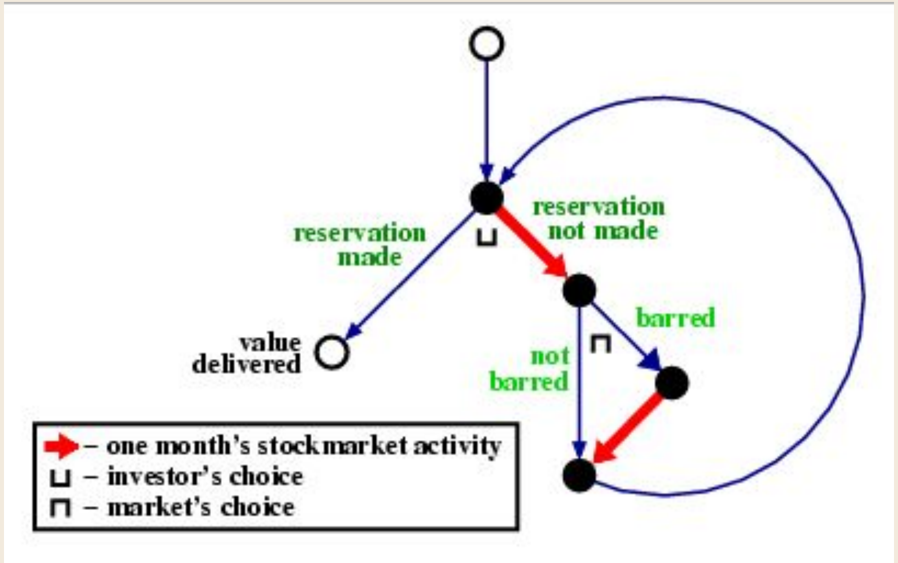
Description

- Motivations
- Visual representation
- State transition

Analysis and Results

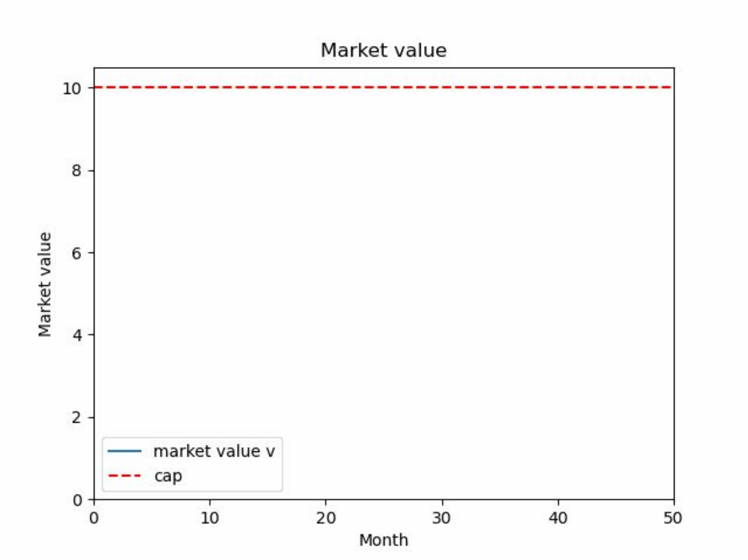
Original Case Study

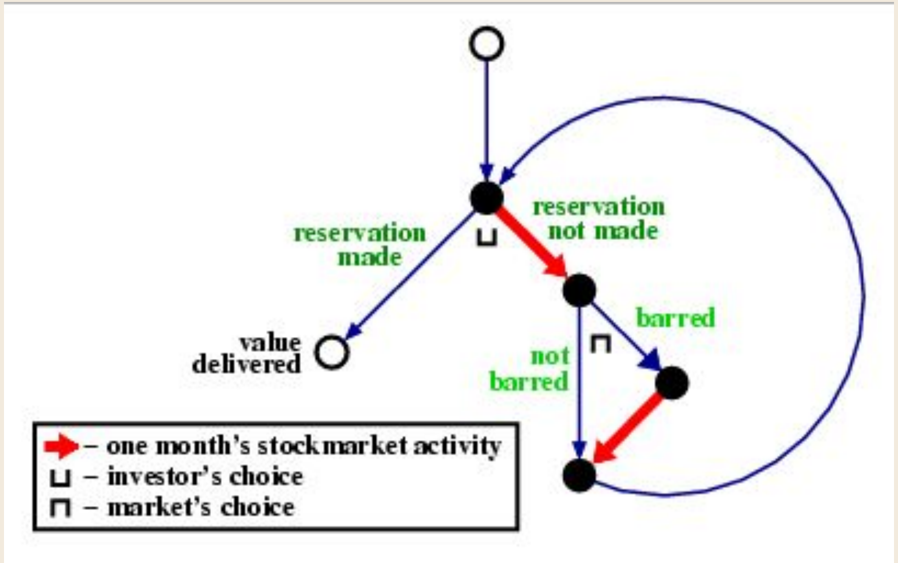




Source of second image: https://www.prismmodelchecker.org/casestudies/investor.php

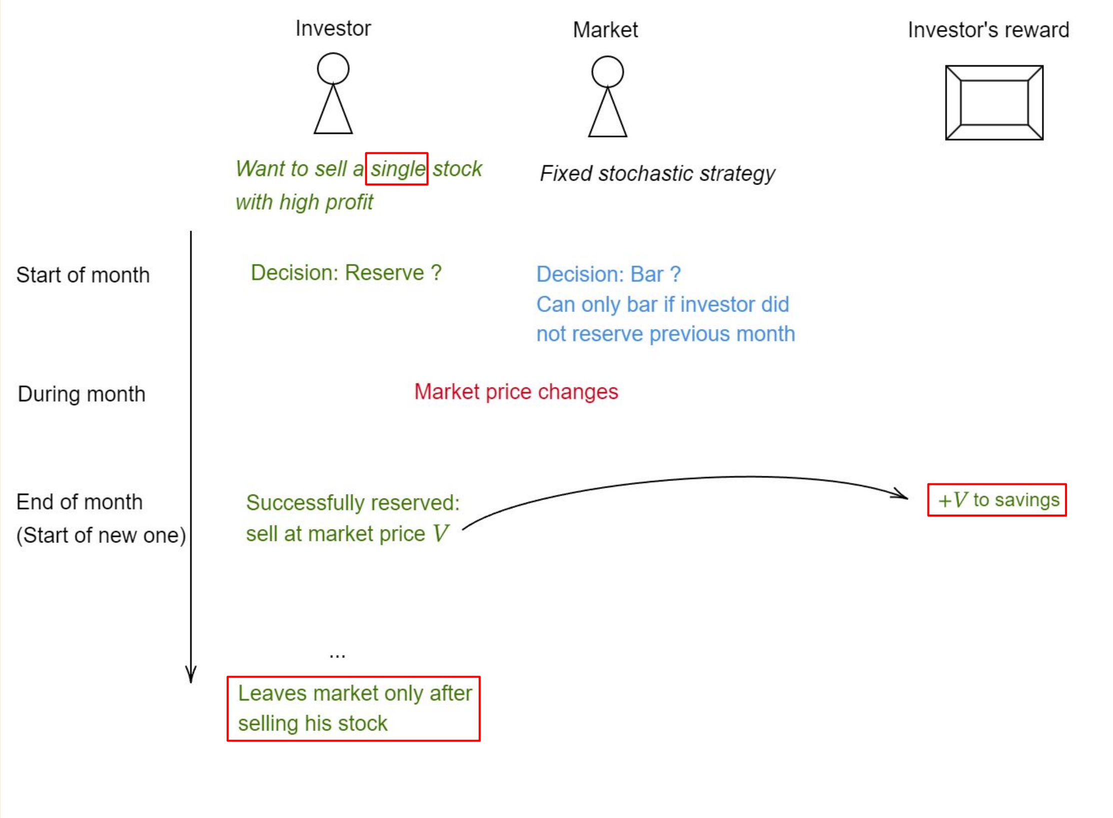
Original Case Study



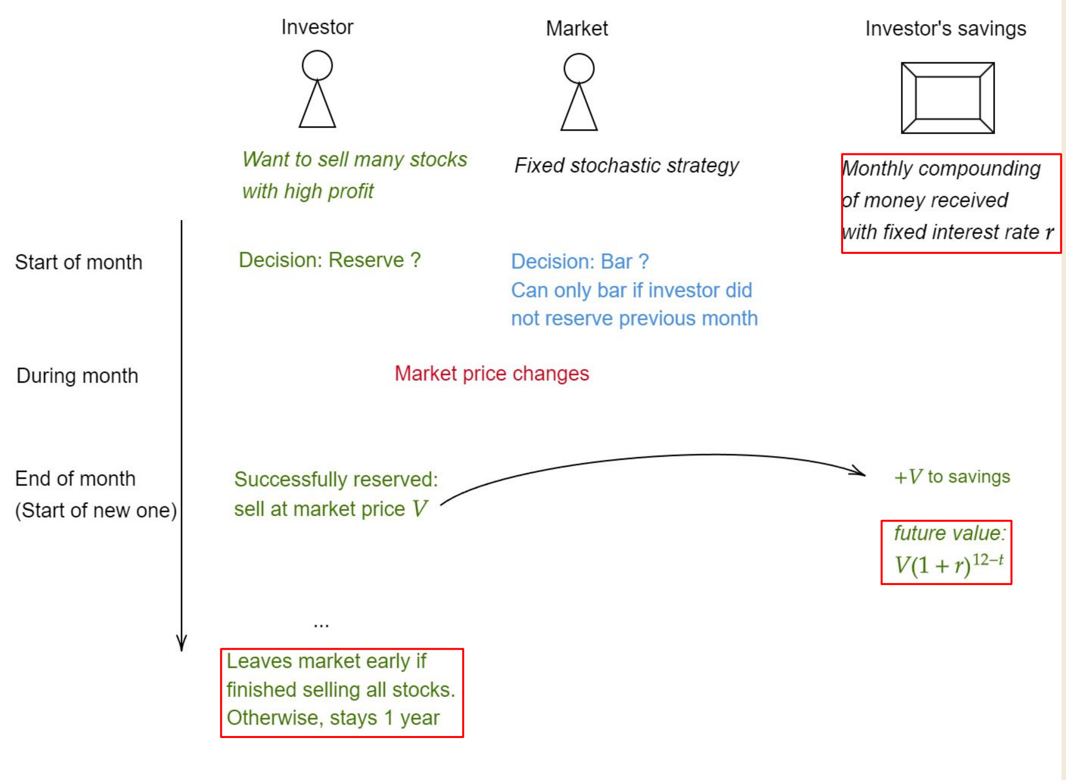


Source of second image: https://www.prismmodelchecker.org/casestudies/investor.php

Visual Representation

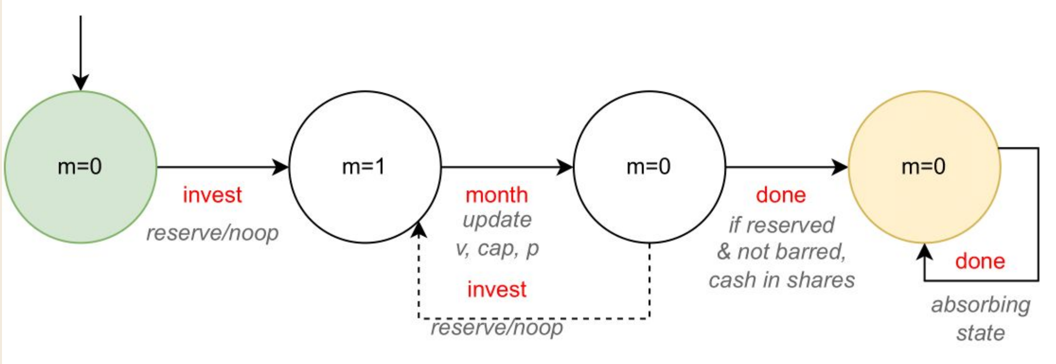


Extended Version

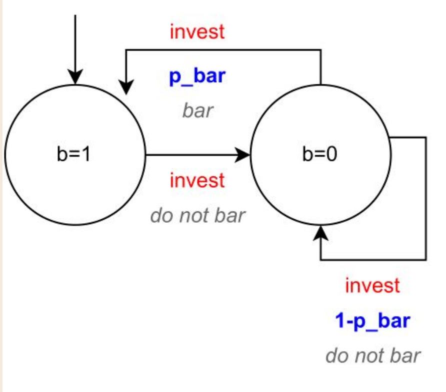


State transition Representation

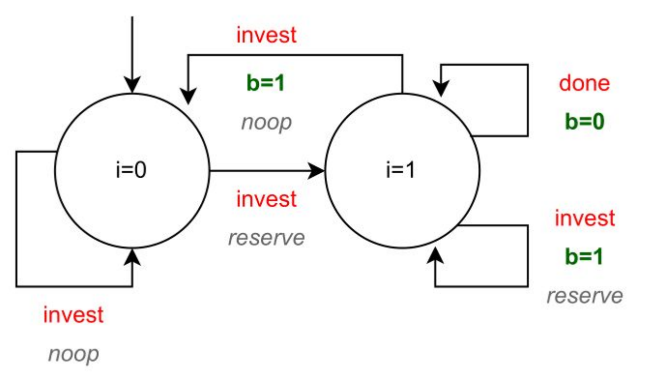
Month Module:



Market Module:

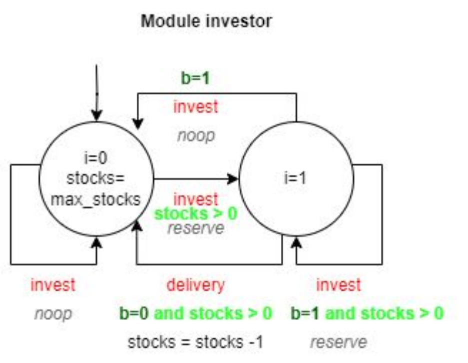


Investor Module:

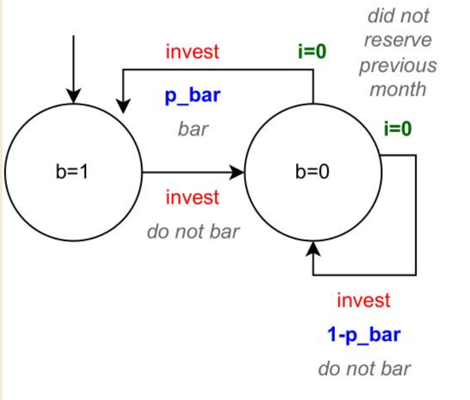


Representation with Changes

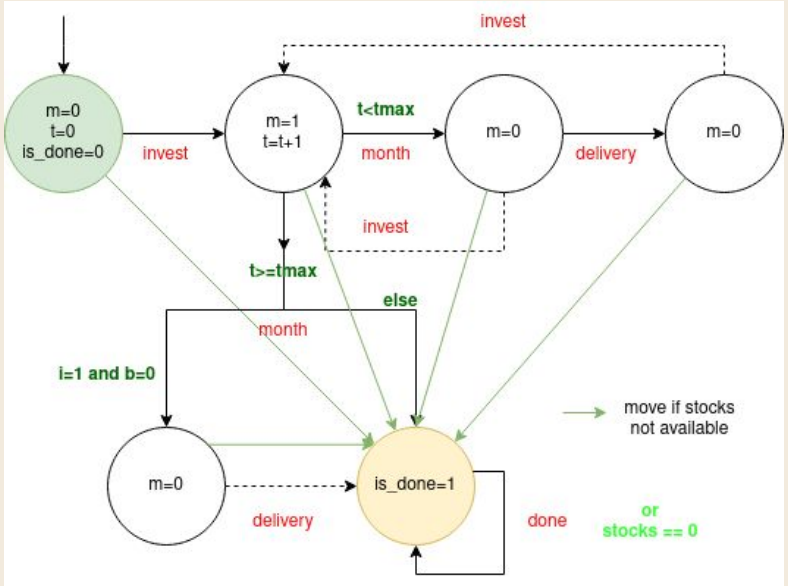
Investor Module:



Market Module:



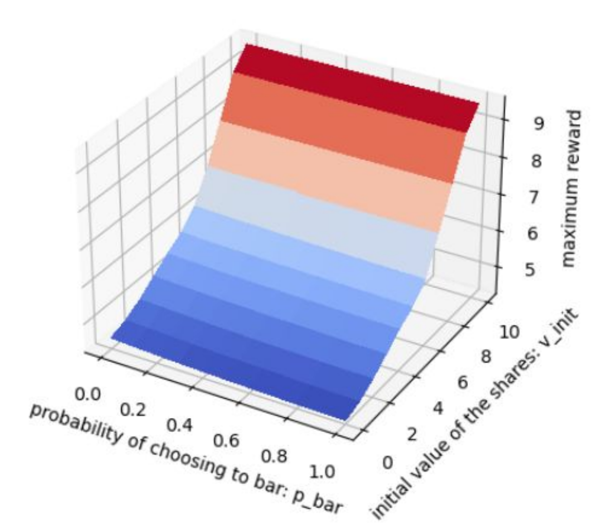
Month Module:

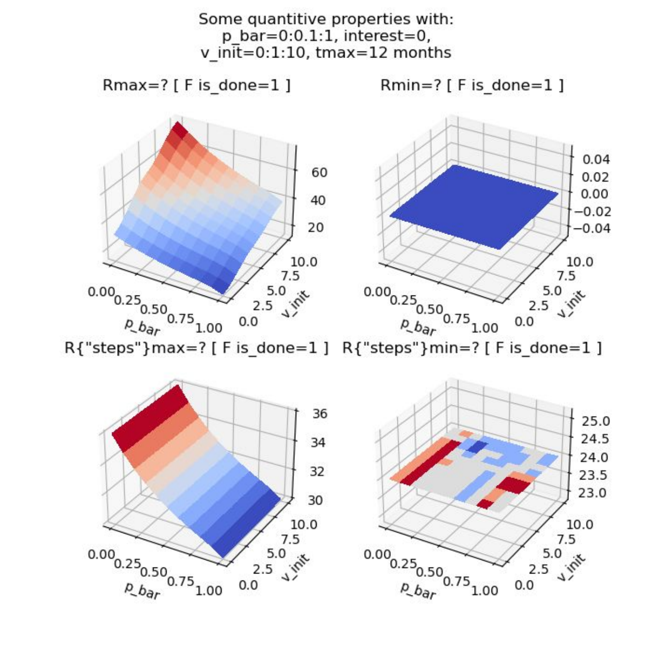


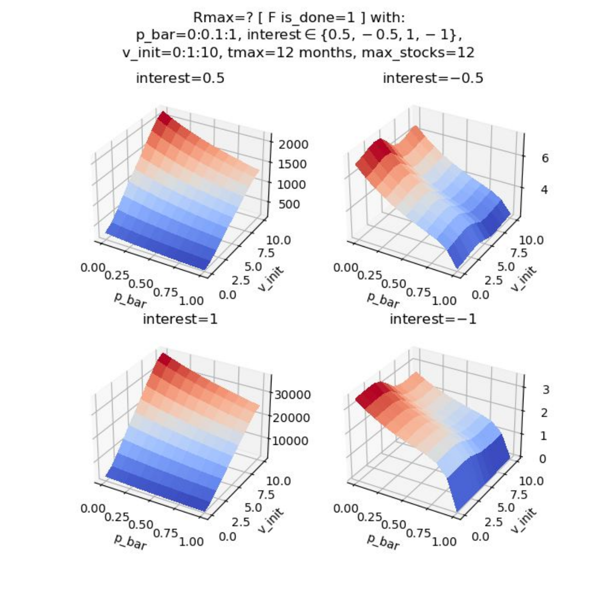
Results

Zero interest rate

Non-zero interest rate







Original case study's results

Our case study's results

Future Work

- Analyze effect of interest together with max_stocks on number of steps
- Analyze effect of max_stocks on expected reward

PRISM-Games extension

- Analyze the case where market has no predefined strategy (stochastic multiplayer game)
- Introduce another investor, they take turns buying/selling to each other
- Implement a stock Future (both actors settle on a price now, for later)

Conclusion