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 6th 2023

Overview

1 Current Status

- Understand basic usage (modeling+analysis)
- Learning coding syntax
- Thinking about limitations

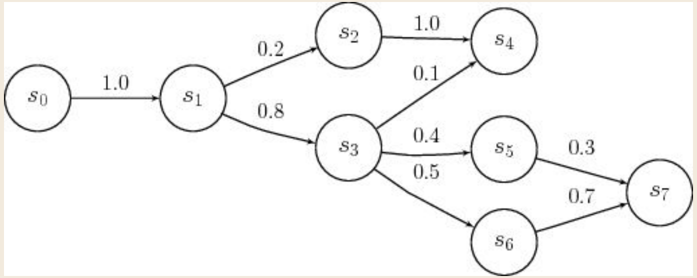
2 Goals

- Understand other probability models
- Learn advanced coding strategies (if any)
- Look into case studies
- Implement our own case study

Basic Usage

- **Probabilistic models in PRISM:**
  - discrete-time Markov chains (DTMCs)
  - continuous-time Markov chains (CTMCs)
  - Markov decision processes (MDPs)
  - probabilistic automata (PAs)
  - probabilistic timed automata (PTAs)
  - partially observable Markov decision processes (POMDPs)
  - partially observable probabilistic timed automata (POPTAs)

- Focus on DTMC to learn basic usage of PRISM



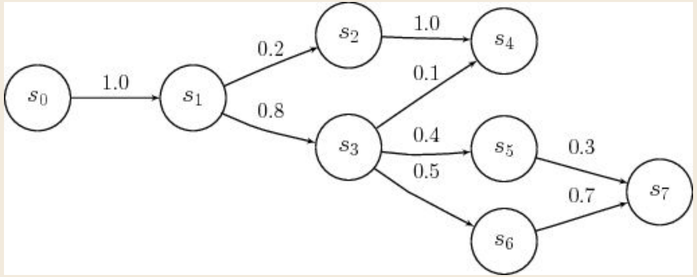
Model Analysis

Simulator

- Sample/simulate paths
- Manually choose transitions
- Reset path
- Choose number of steps

Properties

- Probability **P** s.t. property is true
- Is **P** ≥ value ?
- Expected cumulative reward
- **P** in terms of **x** variable



Options

- Solutions methods (Value Iteration, Policy Iteration, Linear Programming, etc.)
- Change stopping condition

Intro to PCTL for DTMCs

Probabilistic **C**omputation **T**ree **L**ogic = Extension of **CTL** with:

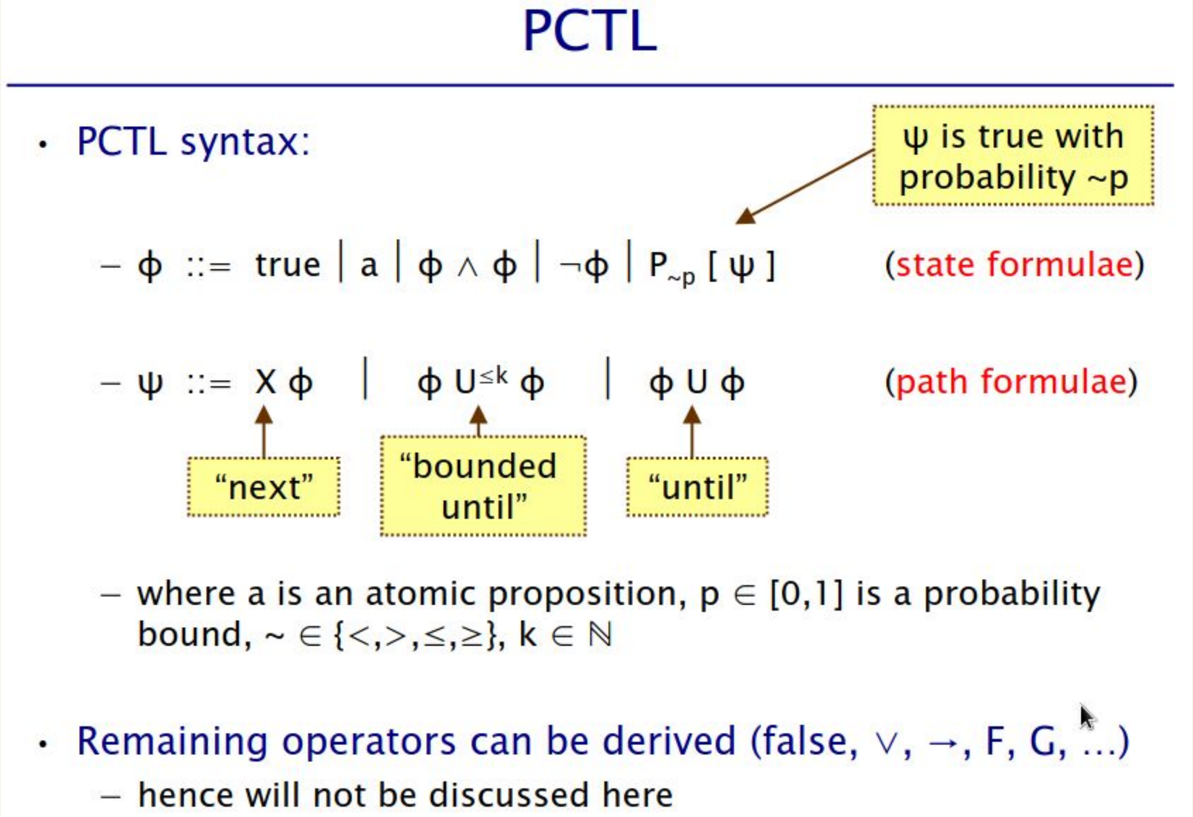
- Prop. holding for a portion of the computations ? e.g 50 %
  - Not just A (all) or E (exists) path(s)/run(s)/execution(s)
  - P properties in PRISM:

  P bound [pathprop]

- Prop. holding cont. during time interval ? (Discrete Time MC)
  - E.g : **P ≥ value [p1 U≤2 p2]**

"After a request for a service, there is at least a 98% probability that the service will be carried out within 2 seconds" - Wikipedia

PCTL Properties Syntax



PCTL

- PCTL syntax:

ψ is true with probability ~p

  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p} [\, \psi \,]$ (state formulae)
  - $\psi ::= X\,\phi \mid \phi\, U^{\leq k}\, \phi \mid \phi\, U\, \phi$ (path formulae)

“next” “bounded until” “until”

  - where a is an atomic proposition, $p \in [0,1]$ is a probability bound, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$

- Remaining operators can be derived (false, ∨, →, F, G, …)
  - hence will not be discussed here

Taken from: https://www.prismmodelchecker.org/lectures/pmc/05-dtmc%20model%20checking.pdf

PRISM Coding syntax

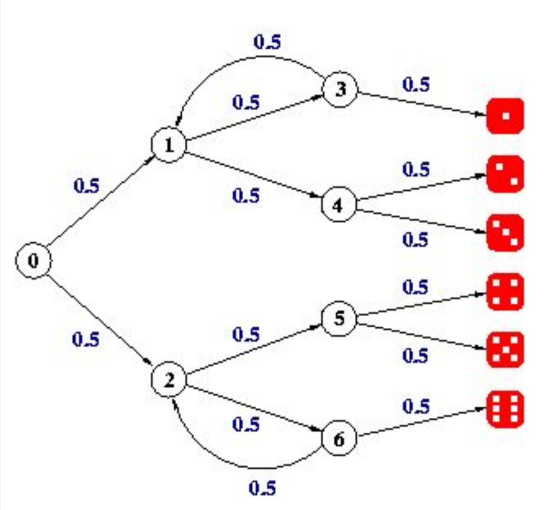
```
dtmc

module die

        // local state
        s : [0..7] init 0;
        // value of the die
        d : [0..6] init 0;

        [] s=0 -> 0.5 : (s'=1) + 0.5 : (s'=2);
        [] s=1 -> 0.5 : (s'=3) + 0.5 : (s'=4);
        [] s=2 -> 0.5 : (s'=5) + 0.5 : (s'=6);
        [] s=3 -> 0.5 : (s'=1) + 0.5 : (s'=7) & (d'=1);
        [] s=4 -> 0.5 : (s'=7) & (d'=2) + 0.5 : (s'=7) & (d'=3);
        [] s=5 -> 0.5 : (s'=7) & (d'=4) + 0.5 : (s'=7) & (d'=5);
        [] s=6 -> 0.5 : (s'=2) + 0.5 : (s'=7) & (d'=6);
        [] s=7 -> (s'=7);


    endmodule
```



Limitations (In-progress)

Known/Documented

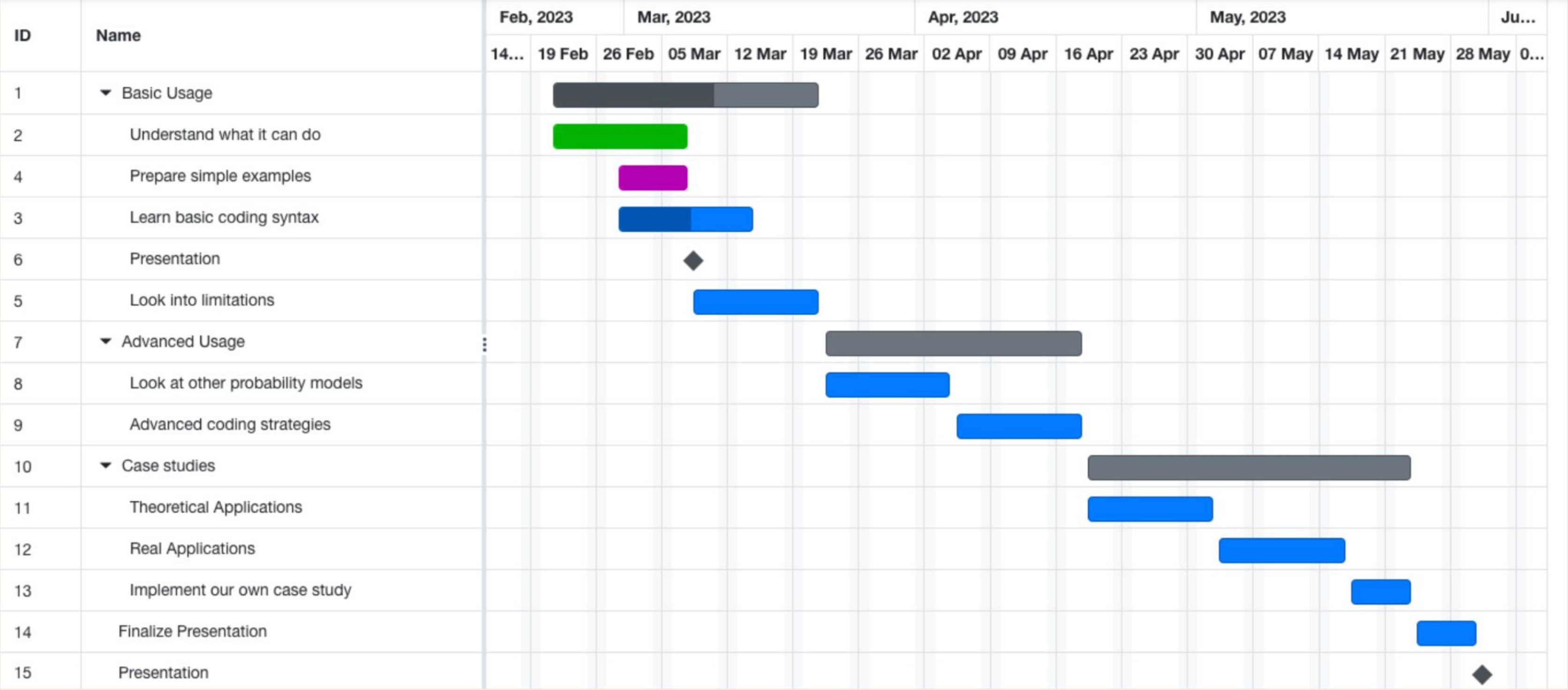
- Variable ordering affects performance

Possible improvements (might already exist)

- Avoid creating states one-by-one
- Import model representation

Demo

Project Progression



ID	Name
1	▾ Basic Usage
2	Understand what it can do
4	Prepare simple examples
3	Learn basic coding syntax
6	Presentation
5	Look into limitations
7	▾ Advanced Usage
8	Look at other probability models
9	Advanced coding strategies
10	▾ Case studies
11	Theoretical Applications
12	Real Applications
13	Implement our own case study
14	Finalize Presentation
15	Presentation

Conclusion