What is PRISM?

- **PRISM**: Probabilistic (symbolic) model checker

- **PRISM today**
  - 12 types of probabilistic models, many probabilistic temporal logics
  - uses: logic, automata, Markov models, optimisation, SMT, simulation, game theory, artificial intelligence, learning...
  - >400 case studies across a broad range of application domains

- **PRISM development**: driven by challenges, applications, users
PRISM: A brief history

- **1998:** tool development begins (Birmingham)
  - symbolic probabilistic model checking
- **2001:** first official public release of PRISM
- **2004 & 2006:** PRISM 2.0 & 3.0
  - new GUI, logics, cost/reward models, simulator engines
- **2011:** PRISM 4.0
  - probabilistic real-time systems, PRISM benchmark suite
  - multi-objective model checking & assume-guarantee
- **2013:** first release of PRISM-games
- **2016 & 2020:** PRISM-games 2.0 & 3.0
  - multi-objective, concurrent stochastic games, equilibria
- **2020-2024:** policy/strategy synthesis, POMDPs, uncertain MDPs, ...
PRISM: A brief history

• 1998: tool development begins (Birmingham)
  • symbolic probabilistic model checking [TACAS’00,’02]

• 2001: first official public release of PRISM

• 2004 & 2006: PRISM 2.0 & 3.0 [TACAS’06]
  • new GUI, logics, cost/reward models, simulator engines [TACAS’04]

• 2011: PRISM 4.0
  • probabilistic real-time systems, PRISM benchmark suite
  • multi-objective model checking & assume-guarantee [TACAS’07,’10,’11]

• 2013: first release of PRISM-games [TACAS’12,’13]

• 2016 & 2020: PRISM-games 2.0 & 3.0
  • multi-objective, concurrent stochastic games, equilibria [TACAS’15,’16,’22]

• 2020-2024: policy/strategy synthesis, POMDPs, uncertain MDPs, ...
Early applications of PRISM

- Randomised distributed algorithms/protocols
  - modelled as MDPs/probabilistic automata
  - key motivating example for probabilistic verification

- Performance modelling & biochemical reactions
  - modelled as Markov chains

- FireWire protocol
- Bluetooth device discovery
- Herman’s self-stabilisation
- FGF cell signalling pathway experiments
- Reliable multiplexing in nanotechnology

Multiple flaws, bugs, anomalies found...
Enabling technologies

• Challenge-driven tool development
  - Symbolic model checking
    • [TACAS'00] [TACAS'02] [STTT'04] [CAV'06] ...
  - Real-time probabilistic verification
    • [TCS'02] [FMSD'06] [Info&Comp'07] [FORMATS'09] ...
  - Game-based abstraction refinement
    • [QEST’06] [VMCAI’09] [FMSD’10] [QEST’11] ...
  - Multi-objective & compositional verification
    • [TACAS’10] [QEST’10] [FASE’11] [Info&Comp’13] ...
  - Multi-agent model checking (stochastic games)
    • [TACAS’12] [Inf-&Comp’17] [FMSD’21] [TACAS’22] ...

\[
M_1 \models \langle A \rangle \geq q \quad \langle A \rangle \geq q \quad M_2 \models \langle G \rangle \geq p
\]
\[
M_1 \parallel M_2 \models \langle G \rangle \geq p
\]
PRISM models

• Increasing variety (and complexity) of probabilistic models supported

  • discrete-time Markov chains
  • probabilistic automata + concurrency
  • continuous-time Markov chains + exponential delays
  • Markov decision processes (MDPs) + policies / control
  • probabilistic timed automata + real-time clocks
  • partially observable MDPs + observability
  • stochastic multi-player games + multi-agent & strategies
  • concurrent stochastic games + concurrency & equilibria
  • interval Markov chains & MDPs + epistemic uncertainty
• Increasing variety (and complexity) of applications tackled
  - discrete-time Markov chains
  - probabilistic automata
  - continuous-time Markov chains
  - Markov decision processes
  - probabilistic timed automata
  - partially observable MDPs
  - stochastic multi-player games
  - concurrent stochastic games
  - interval Markov chains & MDPs

  ▪ Long-running autonomous mobile robots [IJRR’19]
    - via multi-objective MDPs

  ▪ Real-time task scheduling with faulty processors [FMSD’13]
    - probabilistic timed automata

  ▪ Human trust models for automated driving [TCPS’22]
    - via multi-objective POMDPs
PRISM applications

- Increasing variety (and complexity) of applications tackled
  - discrete-time Markov chains
  - probabilistic automata
  - continuous-time Markov chains
  - Markov decision processes
  - probabilistic timed automata
  - partially observable MDPs
  - stochastic multi-player games
  - concurrent stochastic games
  - interval Markov chains & MDPs

Verification + game theory

Computer security
- attack-defence scenarios
  - stochastic games strategies

Distributed energy protocols
- flaws fixed via incentives

[CSF’16]

[FMSD’13]
PRISM applications

Increasing variety (and complexity) of applications tackled

- discrete-time Markov chains
- probabilistic automata
- continuous-time Markov chains
- Markov decision processes
- probabilistic timed automata
- partially observable MDPs
- stochastic multi-player games
- concurrent stochastic games
- interval Markov chains & MDPs

Verification & epistemic uncertainty

- Robust AUV control [JAIR’23]
  - continuous-space + unknown noise → IMDP

- Robust anytime learning [NeurIPS’22]
  - MDP policies learnt from samples
  - IMDPs used for robust guarantees

- Deep reinforcement learning [FORMATS’22]
  - verified probabilistic policies from neural nets, via IMDPs
Who uses PRISM? (and how/why)

• PRISM bibliography\(^1\) lists >850 papers relating to PRISM
  - 375 “case studies”, >100 “extensions”, >250 “connections”

• PRISM applications & users
  - very wide (often non-expert) user base
  - broad applicability of PMC techniques/models
  - easy, self-contained install \(\text{Apple} \quad \text{Linux} \quad \text{Windows}\)
  - user interface: model editor, simulator, debugger, graph plotting, …
  - documentation, tutorials, examples

• General aims
  - stable, usable, flexible, coherent framework


• PRISM for teaching
  - common basis for the practical component of taught courses on (non-)probabilistic model checking
Diverse applications of PRISM

• Cloud computing
  ▪ live migration of VMs
  ▪ plan optimisation for performance guarantees

[Kikuchi/Matsumoto (Fujitsu), CLOUD’11] (Best paper)

• Football tactics
  ▪ team strategies learnt from data
  ▪ tactical efficiency analysed via probabilistic model checking

[Van Roy et al., JAIR’23, MIT-SSAC’24]

• Human-cell conversion
  ▪ for disease models, gene therapies
  ▪ design tool for optimisation and prediction, based on model checking

[Jung et al., Nature Communications’21]
Building on PRISM

- **Extending PRISM**
  - open-source codebase (GPL)
  - primarily implemented in Java
    - (some C code and various native libraries)
  - accessibility for student/external contributors
    - “explicit” engine is an easy entry point

- **Connections & tool chains**
  - via PRISM modelling language
    - e.g. PEPA, bigraphs, RoboChart, SBML
  - via explicit (textual) model files
  - programmatically via API
    - Java, Python, model generators
The PRISM language

- PRISM modelling language
  - simple textual language based on guarded commands
  - inspired by: SMV language, Reactive modules
  - de-facto standard for probabilistic verification

- Key ingredients
  - the basics: modules (variables + guarded commands), parallel composition, costs/rewards, parameters (constants)
  - also: clocks, observations, players, epistemic uncertainty, ...

- Some design decisions
  - consistent modelling language for many model types
  - (deliberately) simple/low-level, general-purpose language
The PRISM language

- **PRISM modelling language**
  - simple textual language based on guarded commands
  - inspired by: SMV language
  - de-facto standard for probabilistic verification

- **Key ingredients**
  - the basics: modules (variables + guarded commands), parallel composition, costs/rewards, parameters (constants)
  - also: clocks, observations, players, epistemic uncertainty, …

- **Some design decisions**
  - consistent modelling language for many model types
  - (deliberately) simple/low-level, general-purpose language

- **PRISM models as benchmarks**
  - **open data**: >15 years of supplementary materials pages on PRISM publications (~60)
  - **benchmark suite**: 36 scalable benchmarks & property queries classified by type

1 See also: Quantitative Verification Benchmark Set
PRISM: Where next?

• Coming soon (ish) to PRISM
  - modelling/property language extensions
  - more flexible compositional model specifications
  - improved API access (Java, Python, ...)
  - better tool interoperability

• Research advances in model checking functionality
  - epistemic uncertainty (e.g., intervals)
  - learning models/parameters from data,
  - neuro-symbolic models
  - stochastic games & equilibria
Summary

- **PRISM (& PRISM-games)**
  - approx. 25 years of continuous development
  - challenge-, application- and user-driven tool evolution
  - stable, usable, coherent framework for wide user base
  - many enhancements to come and challenges to tackle

PRISM contributors:

(and many more contributors to underlying theory and techniques)

prismmodelchecker.org