Probabilistic Model Checking with PRISM

Past, Present & Future

Marta Kwiatkowska
University of Oxford

Gethin Norman
University of Glasgow

Dave Parker
University of Birmingham

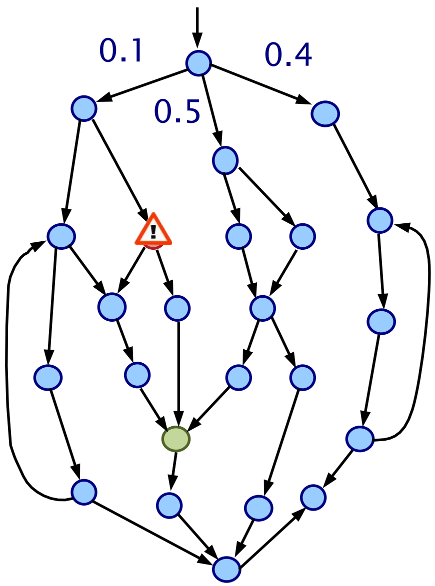
HVC 2016, Haifa, November 2016

Outline

- Probabilistic model checking and PRISM
- Themes and trends
- Advances and applications
- Current research topics
- Challenges & future directions

Probabilistic model checking

- Construction and analysis of probabilistic models
  - probability: failures, uncertainty, noise, randomisation, …
  - time: delays, time-outs, failure rates, …
  - costs: energy, resources, …



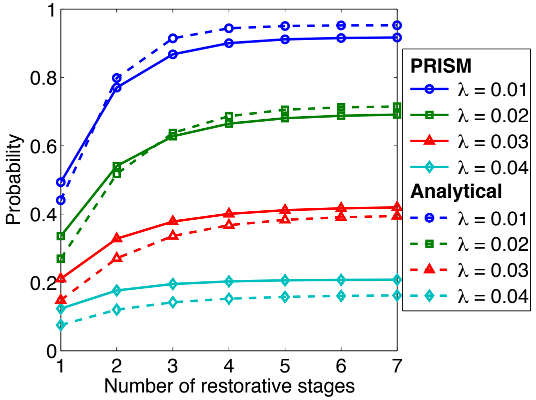
- Quantitative correctness properties expressed in temporal logic, e.g.:
  - *trigger* $\rightarrow P_{\geq 0.999} [\ F^{\leq 20}$ *deploy* $]$
  - "the probability of the airbag deploying within 20 milliseconds of being triggered is at least 0.999"
  - reliability, timeliness, performance, efficiency, …

PRISM

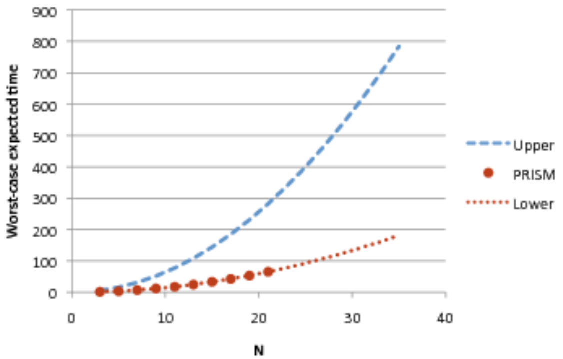
- A (brief) history
  - late 80s, early 90s: first underlying theory developed
  - 2001: first official public release of PRISM
  - 2011: version 4.0 - probabilistic real time systems
  - 2013: PRISM-games – stochastic multi-player games
- PRISM today
  - used in 100+ institutions; 50,000+ downloads
  - broadly applicable; many diverse use cases
  - many non-expert (and non-CS) users
  - 300 external papers (no involvement from PRISM team)
  - flaws found in real-systems; industrial usage

What can we do with PRISM?

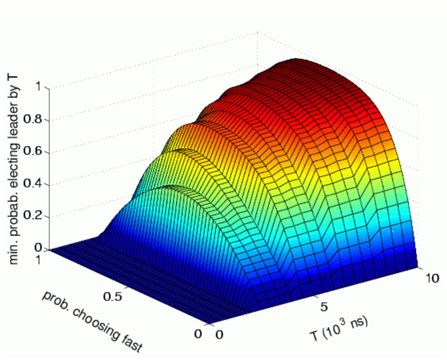
- Identify flaws in existing analyses
  - e.g. reliability of NAND multiplexing
- Investigate conjectures/models
  - e.g. Herman's self-stabilisation
  - e.g. FireWire root contention
  - e.g. cell signalling pathways (FGF)



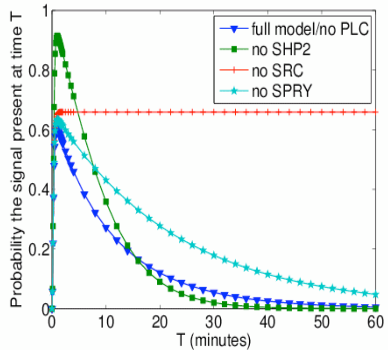
NAND



Herman



FireWire



FGF

Themes and trends

- Themes in the development of PRISM
  - theory-to-practice (and practice-to-theory)
  - wide collaboration (theory, algorithms, case studies)
  - open source software (and data)
  - overlaps/interacts with many other disciplines
- Trends
  - improvement in scalability to larger models
  - increasingly expressive/powerful classes of model
  - from verification problems to control problems
  - ever widening range of application domains

Trends

Models

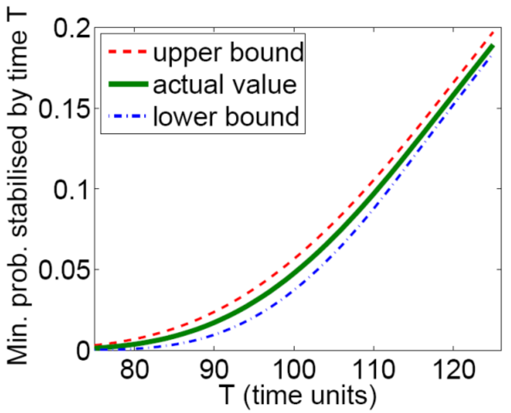
- discrete-time Markov chains
- probabilistic automata
- continuous-time Markov chains
- Markov decision processes
- probabilistic timed automata
- stochastic multi-player games
- …

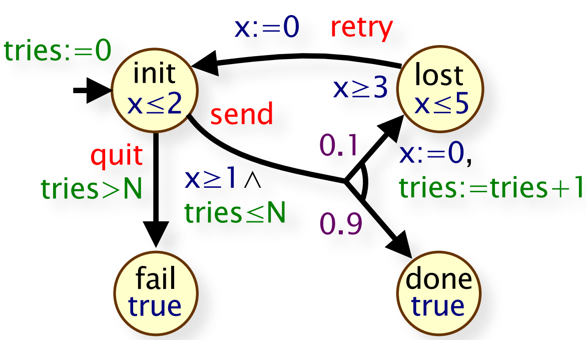
Application domains

- randomised distributed algorithms
- network/communication protocols
- computer security
- performance/reliability
- systems biology
- DNA computing
- robotics & autonomous vehicles
- wearable/implantable devices
- …

Enabling technologies

- Symbolic model checking
  - [TACAS'00] [TACAS'02] [STTT'04] [CAV'06] ...
- Real-time probabilistic verification
  - [TCS'02] [FMSD'06] [Info&Comp'07] [FORMATS'09] ...
- Quantitative abstraction refinement
  - [QEST'06] [VMCAI'09] [FMSD'10] [QEST'11] ...
- Compositional verification
  - [TACAS'10] [QEST'10] [FASE'11] [Info&Comp'13] ...
- And more...
  - statistical model checking, symmetry reduction, bisimulation minimisation, ...

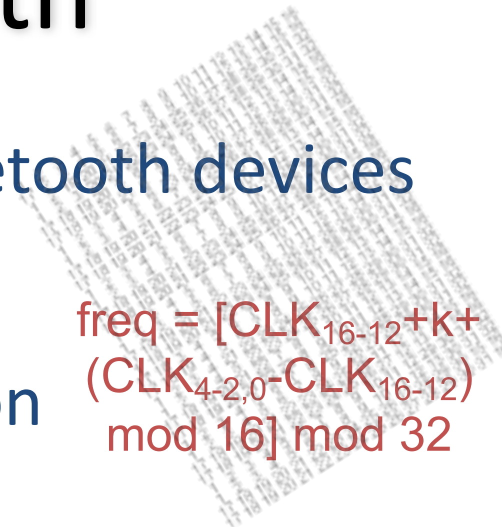




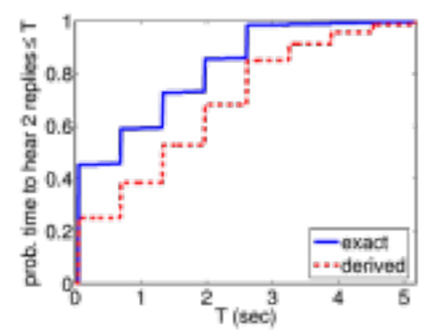
$$\frac{M_1 \models \langle A\rangle_{\geq q} \quad \langle A\rangle_{\geq q}\, M_2\, \langle G\rangle_{\geq p}}{M_1 \,||\, M_2 \models \langle G\rangle_{\geq p}}$$

Case study: Bluetooth

- Device discovery between a pair of Bluetooth devices
  - performance essential for this phase
- Detailed model from official specification
  - two asynchronous 28-bit clocks
  - pseudo-random hopping between 32 frequencies
  - random waiting scheme to avoid collisions
  - 32 Markov chains, over 3×10^{10} states each
  - 17,179,869,184 initial configurations



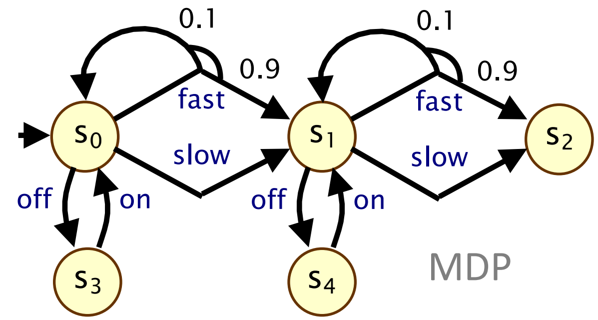
freq = $[CLK_{16-12}+k+(CLK_{4-2,0}-CLK_{16-12}) \bmod 16] \bmod 32$

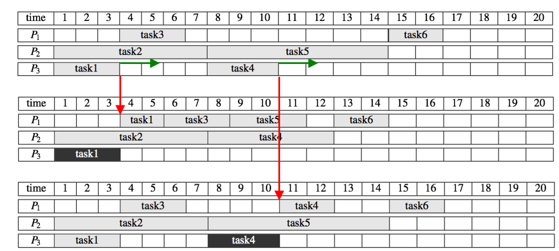


- Symbolic probabilistic model checking
  - "worst-case expected discovery time is at most 5.17s"

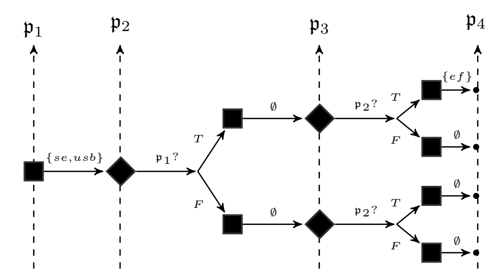
Strategy/controller synthesis

- Verification vs. control
  - verify that a system is “correct”, for any environment/adversary/… (counterexample yields flaw/attack/…)
  - synthesise a "correct-by-construction” controller from formal specification (witness yields strategy/controller)
- Applications
  - dynamic power management, robots/autonomous vehicle navigation , task/network scheduling, security, …





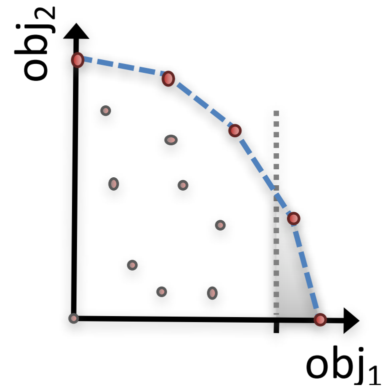
Task schedule [FMSD’13]



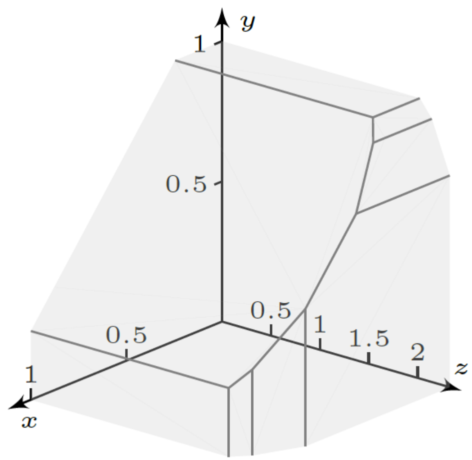
Attack-defence tree [CSF’16]

Multiple objectives

- Multi-objective controller synthesis [LMCS'08] [TACAS'11]
  - trade-offs between conflicting objectives



- Mix of optimisation and guarantees
  - e.g. "what strategy maximises probability of message transmission, whilst guaranteeing expected battery life-time is > 10 hrs?"
  - Pareto curve generation/approximation



- Extensions
  - permissive controller synthesis of multi-strategies for MDPs [LMCS'15]
  - multiple objectives for multi-player games (see later)

Robots & autonomous systems

- Navigation for mobile service robots
  - learnt probabilistic navigation maps
  - LTL task specifications + controller synthesis
  - ROS-based runtime planning implementation
  - multi-objective probabilistic guarantees on task completion/duration [IROS’14/IJCAI’15/CDC’16]

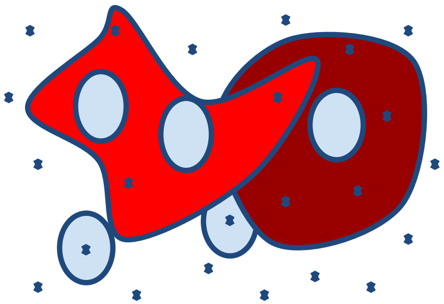
- Autonomous underwater vehicle navigation
  - incremental/parametric verification + controller synthesis
  - probabilistic programming + machine learning to generate realistic component/environment models at runtime

Parameter synthesis

- Synthesising models that are guaranteed to satisfy quantitative correctness properties is difficult
  - but we can synthesise controllers and parameters
- Parameter synthesis
  - given a parametric model and a property ϕ…
  - find the optimal parameter values, with respect to an objective function O, such that the property ϕ is satisfied, if such values exist
- Quantitative parameter synthesis
  - parameters: timing delays, rates
  - objectives: optimise probability, reward/volume

Quantitative parameter synthesis

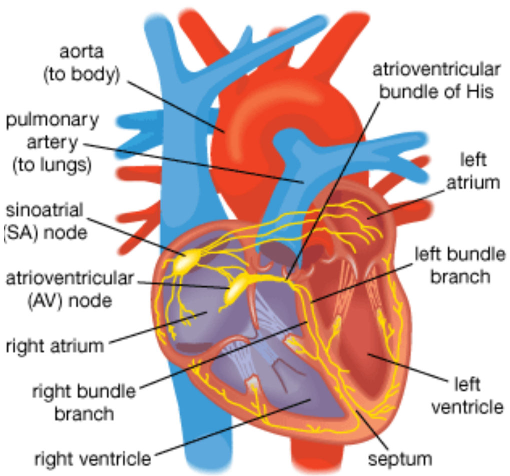
- Timed/hybrid automata
  - find optimal timing delays [EMSOFT2014] [HSB'15] [HSCC'16]
  - constraint solving, discretisation + sampling

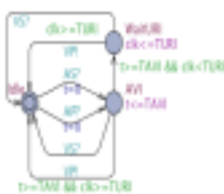


- Probabilistic timed automata
  - find delays to optimise probability [RP2014]
  - parametric symbolic abstraction-refinement
- Continuous-time Markov chains
  - find optimal rates [CMSB'14] [ActaInf'16], PRISM-PSY [TACAS'16]
  - constraint solving, uniformisation + sampling
- Focus: practical implementation, real-world usage

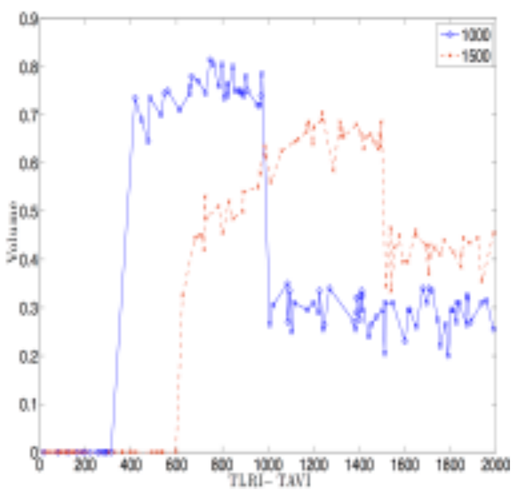
Applications

Pacemaker verification

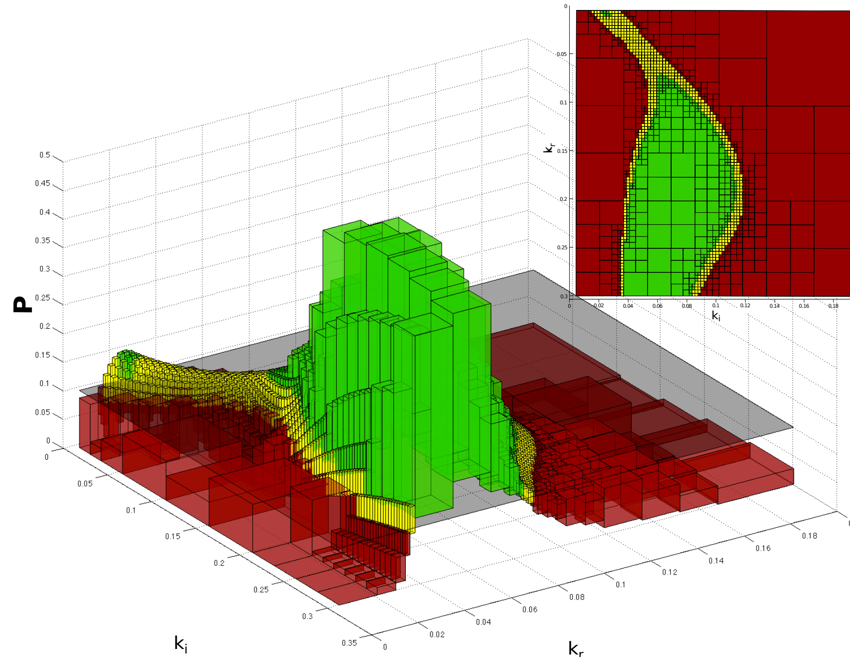




Maximal volume objective



Epidemic modelling



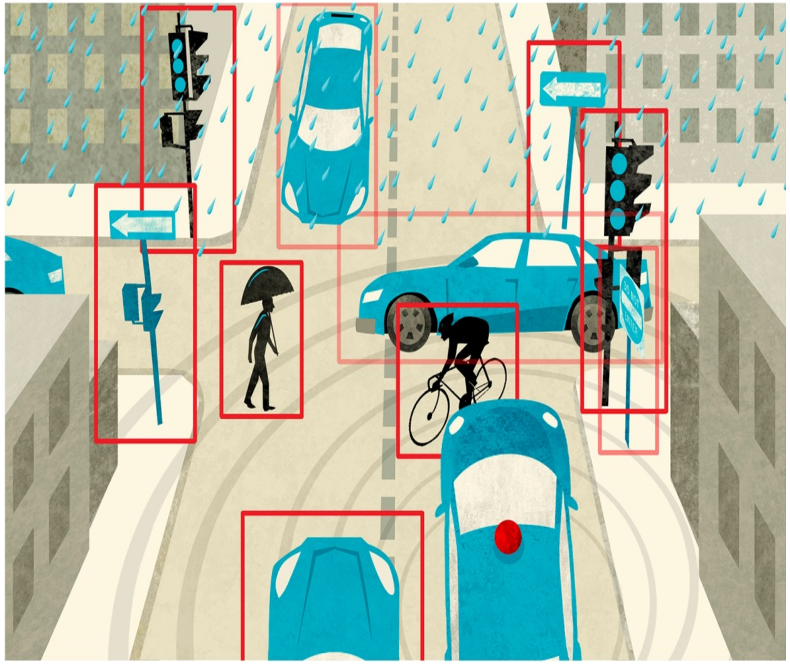
$$S + I \xrightarrow{k_i} I + I$$ Susceptible → Infected

$$I \xrightarrow{k_r} R$$ Infected → Recovered

ϕ: infection lasts at least 100 time units and ends within 120 time units

Mobile autonomy challenge

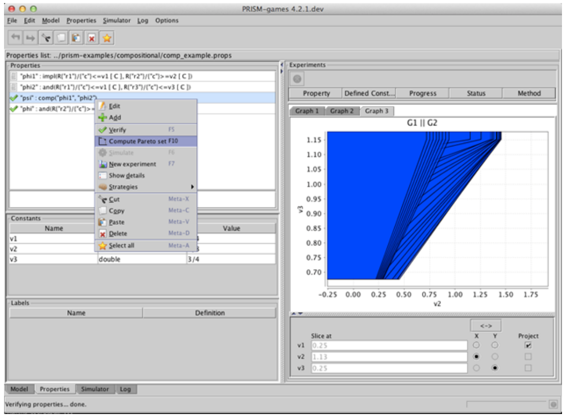
- Autonomous systems
  - interact with their environment, which is possibly adversarial
  - have goals/objectives, which may conflict
  - take decisions



- Model as stochastic games
  - well known from, e.g., decision making in economics
  - many application domains: security, energy grid, etc
- Tool PRISM-games, extension of PRISM [TACAS'16]

Stochastic multi-player games

- Probabilistic temporal logic with coalitions
  - probabilities, rewards (reachability, total, mean-payoffs/ratios, …) [FMSD’13] [ICALP’16] [ECC’16]

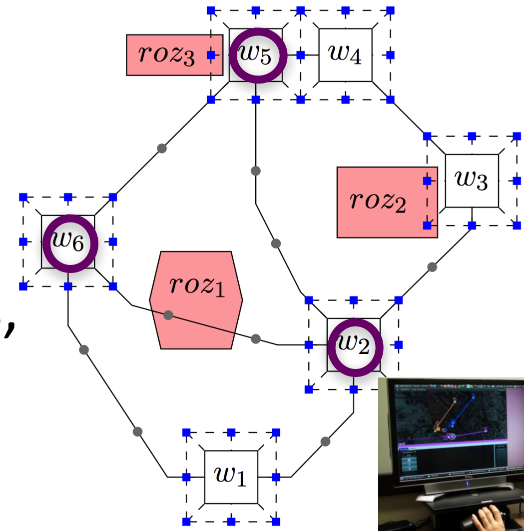


- Multi-objective strategy synthesis
  - Pareto set computation and optimal achievable trade-offs [MFCS’13] [QEST’13] [TACAS’15]

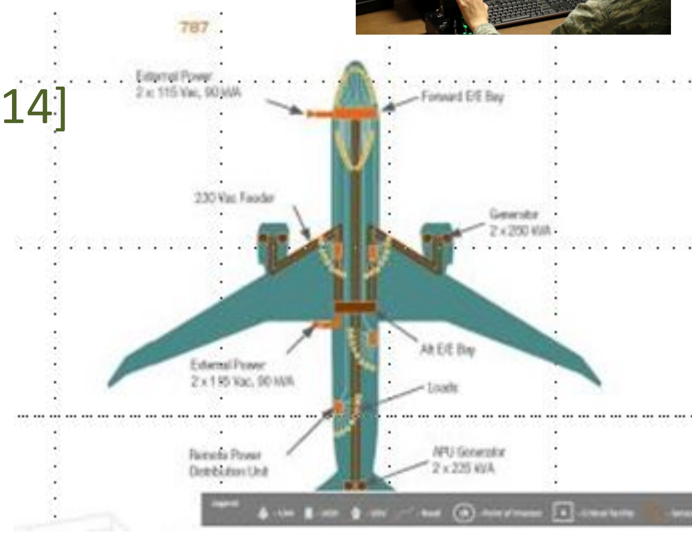
- Compositional strategy synthesis [CONCUR’14] [Inf & Comp’16]
  - assume-guarantee + multi-objective strategy synthesis
  - e.g. local strategies for $G_1 \models \phi_A$, $G_2 \models \phi_A \Rightarrow \phi_B$ compose to a global strategy for $G_1 || G_2 \models \phi_B$

Applications

- UAV path planning [ICCPS'15]
  - human operator + low-level piloting
  - quantitative mission objectives: minimise time/fuel, restricted zones, operator fatigue/workload
  - multi-objective MDPs, stoch. games

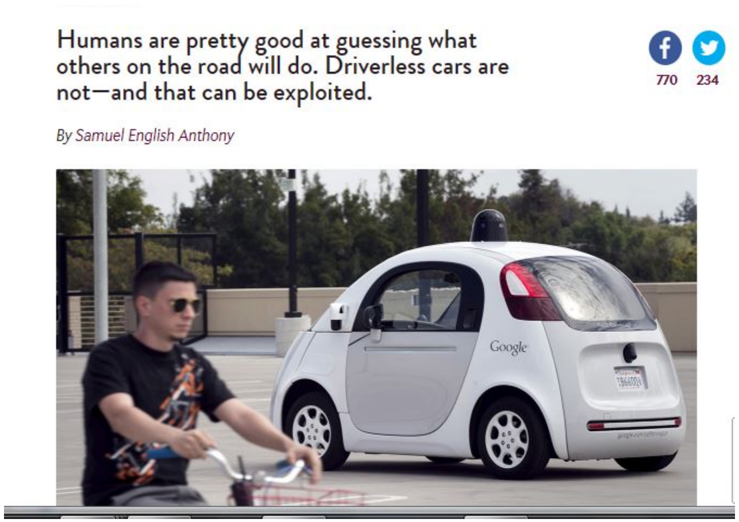


- Aircraft power distribution [CONCUR'14]
  - compositional strategy synthesis in stochastic games (PRISM-games)
  - specify control objectives in LTL using mean payoff



Are games sufficient?

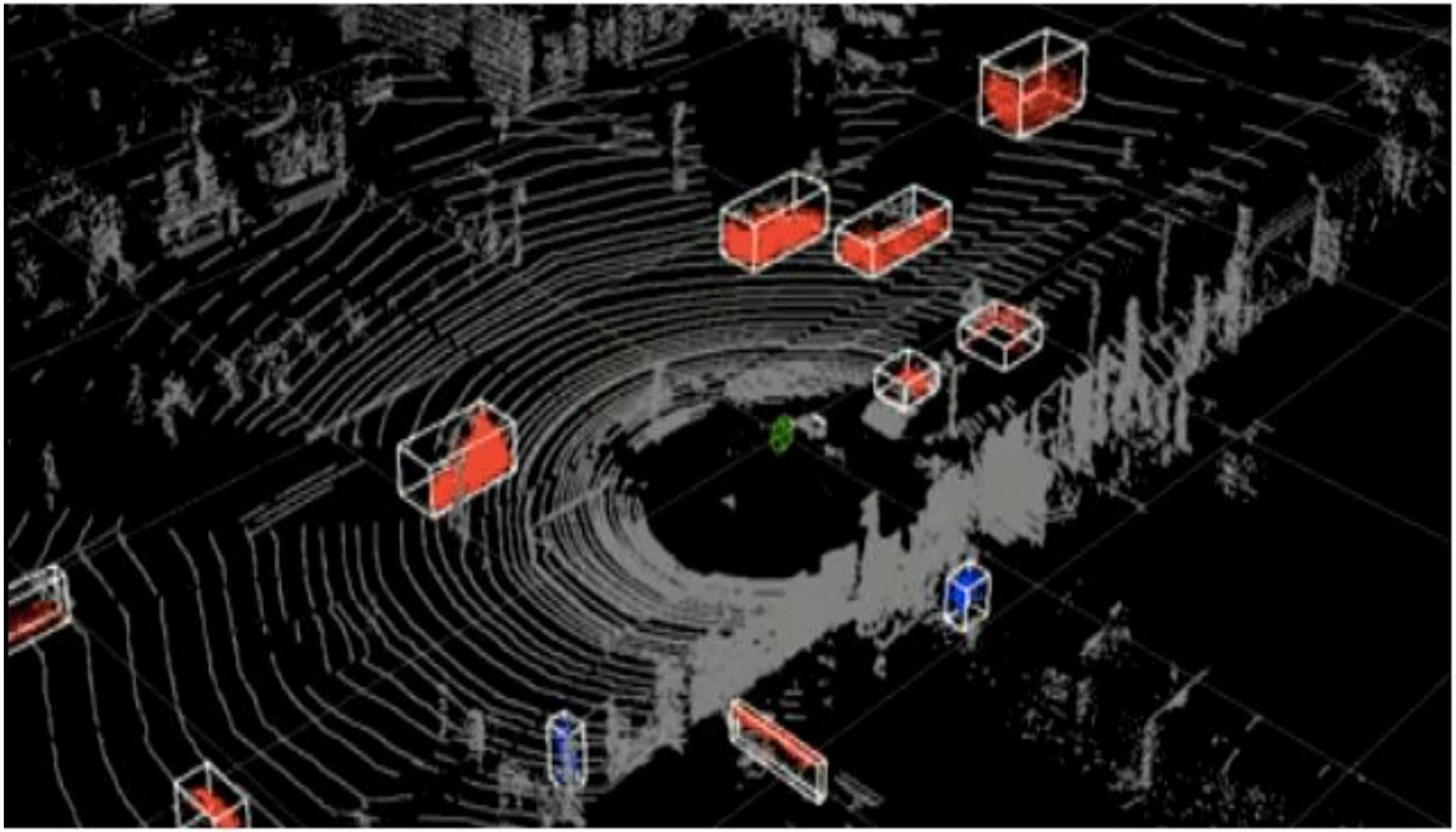
- Complex decisions!
  - goals
  - perception
  - situation awareness
  - context (social, regulatory)
- What about social subtleties?
- What to do in emergency?
  - moral decisions, handover to driver, obey traffic rules
- Need to make robots human-like…
  - need multi-modal communication, cognitive reasoning, trust, ethics, …



Quantitative verification for trust?

- Social trust: fundamental for mobile autonomy [LK16b]
  - influenced by external factors, such as social norms
  - also internal: personality, motivation, preferences
  - subjective: would you trust an autonomous taxi to take your child to school?
- Formulate a temporal logic to express X's trust in Y for G, based on probabilistic belief [HK17]
- Admits a model checking procedure, which can:
  - be used in decision-making for robots
  - explain decisions, i.e. who is accountable for the action

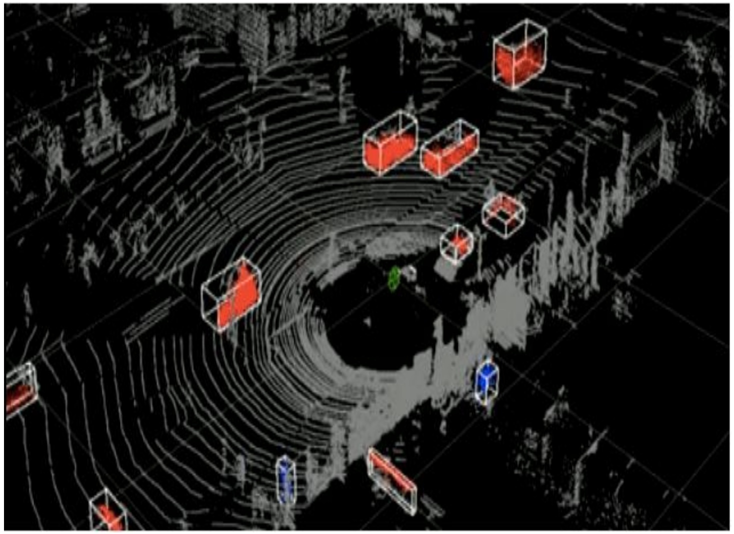
Perception software



Credits: Oxford Robotics Institute

Things that can go wrong…

- …in perception software
  - sensor failure
  - object detection failure



- Machine learning software
  - not clear how it works
  - does not offer guarantees

- Verification for machine learning?
  - some progress towards safety verification for neural networks

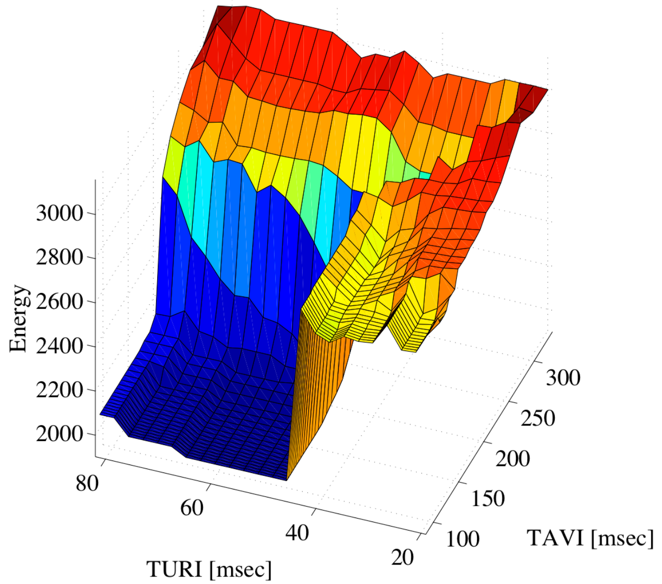
Personalisation challenge

- Device must adapt to physiology of human wearer
  - achieved through model parameterisation
  - parameter estimation, optimal parameter synthesis
- Multiple uses
  - automation of personalised medical intervention
  - device safety assurance, for testing
  - reproduce the unique characteristics for authentication
- Focus on ECG based devices
  - pacemaker models, heart models, synthetic ECGs
  - future work on anxiety monitoring and control

Pacemaker verification/optimisation

- Hybrid model-based framework
  - timed automata model for pacemaker
  - hybrid heart models in Simulink (non-linear ODEs)

- Properties
  - (basic safety) maintain 60-100 beats per minute
  - optimisation of energy usage & cardiac output [HSB'16] [HSCC'16]
  - in-silico analysis of rate-adaptive pacemakers [ICHI'14]
  - hardware in the loop [EMBC'15]

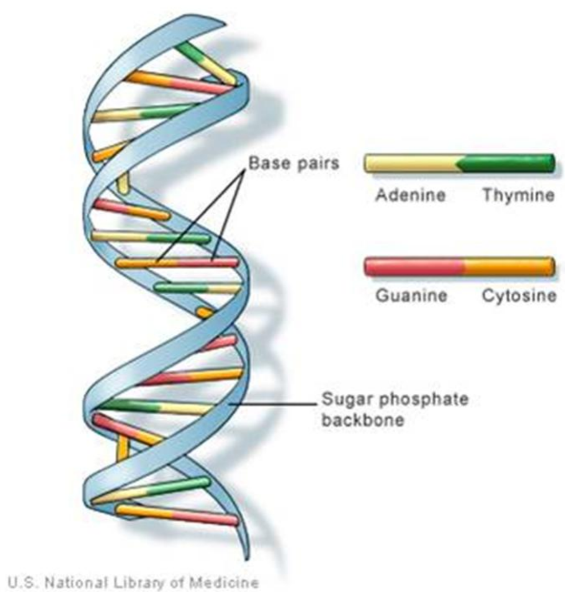


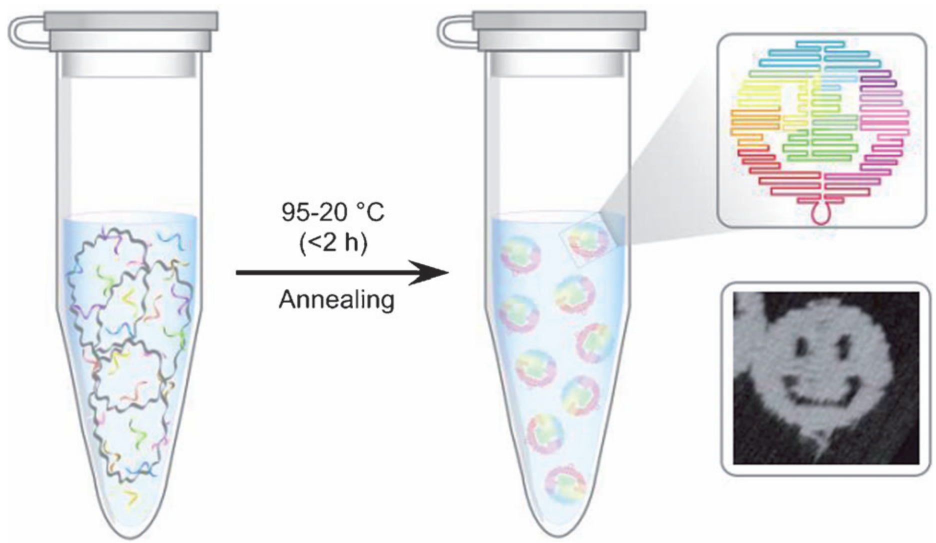
DNA computation challenge

- Moore's law, hence need to make devices smaller…
- DNA computation, directly at the molecular level
  - DNA logic circuit designs & programmable nanorobotics
  - asynchronous DNA circuit designs [DNA'16]
- Many applications envisaged
  - e.g. bio-sensing, point of care diagnostics, …
- Apply quantitative verification and synthesis to
  - find design flaws in DNA computing devices [JRSI'12]
  - analyse reliability and performance of molecular walkers
  - automatically synthesise reaction rates to guarantee a specified level of reliability

DNA nanostructures

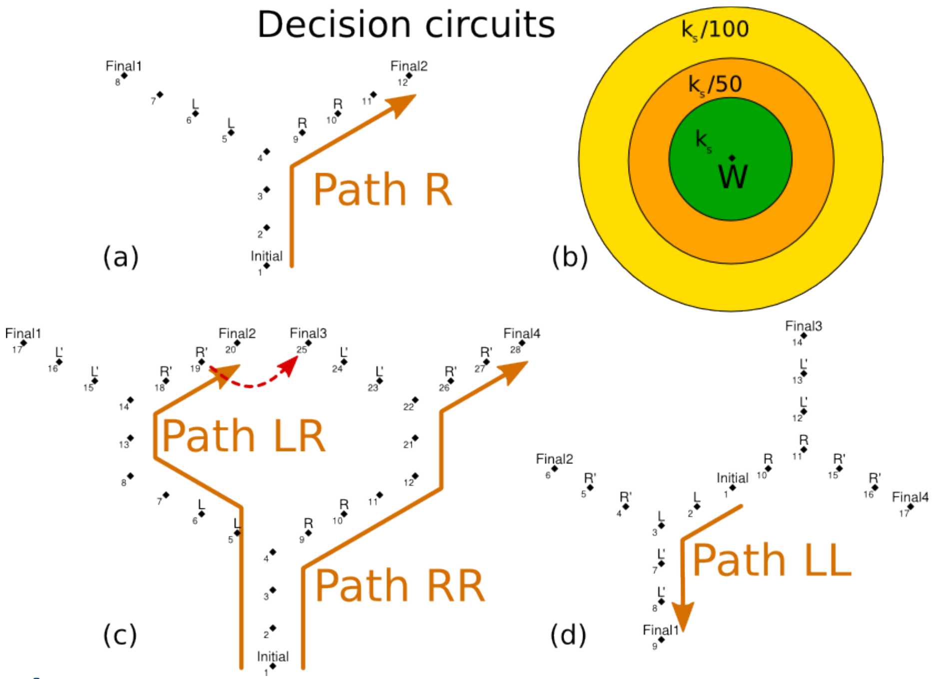
- DNA origami [Rothemund, *Nature* 2006]
  – DNA can self-assemble into structures – “molecular IKEA?”
  – programmable self-assembly (can form tiles, nanotubes, boxes that can open, etc.)



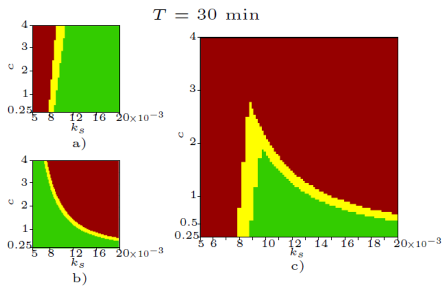


DNA walker circuits

- Computing with DNA walkers [NatComp'14]
  - branching tracks laid out on DNA origami tile
  - starts at 'initial', signals when reaches 'final'
  - can control 'left'/'right' decision
  - any Boolean function

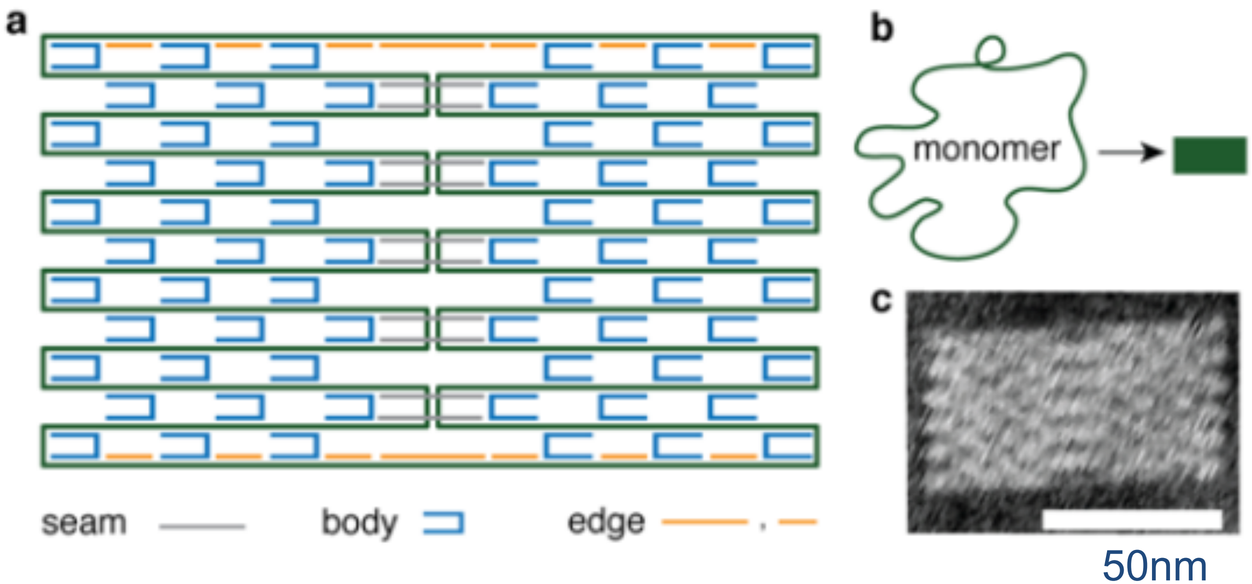


- Parameter synthesis of rates
  - for guaranteed reliability level [CMSB'14]



DNA origami tiles

- DNA origami tiles



- Aim: understand how to control the folding pathway
  - formulate an abstract Markov chain model
  - yields predictions; perform a range of experiments, consistent with predictions [Nature'15]

Conclusions

- Probabilistic model checking & PRISM
  - 15 years since first official tool release
  - significant advances in underlying theory & technologies
  - successfully deployed in many application domains

- Many research challenges and applications ahead
  - verification, synthesis, learning, trust, cognitive models, …
  - autonomous systems, DNA computing, personalised wearable/implantable devices, …

http://www.prismmodelchecker.org/

Acknowledgements

- Contributors (to PRISM & its underlying theory)
  - Aistis Simaitis, Alberto Puggelli, Alessandro Bruni, Alexandru Mereacre, Alistair John Strachan, Andrej Tokarčík, Andrew Hinton, Antonio Pacheco, Archit Taneja, Ashutosh Trivedi, Benoit Barbot, Bruno Lacerda, Carlos Bederian, Charles Harley, Chris Thachuk, Christel Baier, Christian Dehnert, Christian von Essen, Christopher Ziegler, Chunyan Mu, Clemens Wiltsche, Dave Parker, Ernst Moritz Hahn, Frits Dannenberg, Fuzhi Wang, Ganindu Prabhashana, Gethin Norman, Håkan Younes, Holger Hermanns, Hongyang Qu, Jan Křetínský, Jens Katelaan, Jeremy Sproston, Joachim Klein, Joachim Meyer-Kayser, Joost-Pieter Katoen, Kenneth Chan, Klaus Draeger, Kousha Etessami, Lovejeet Singh, Lu Feng, Luca de Alfaro, Marcin Copik, Marco Diciolla, Maria Svorenova, Mark Kattenbelt, Markus Siegle, Marta Kwiatkowska, Mateusz Ujma, Maximilian Probst, Mihalis Yannakakis, Mike Arthur, Milan Ceska, Moshe Vardi, Muhammad Omer Saeed, Nick Hawes, Nicola Paoletti, Nicolas Basset, Nicolas Del Piano, Nishan Kamaleson, Paolo Ballarini, Pedro D'Argenio, Qixia Yuan, Radu Calinescu, Rashid Mehmood, Roberto Segala, Sebastian Vermehren, Sergio Giro, Steffen Märcker, Stephen Gilmore, Taolue Chen, Tingting Han, Vincent Nimal, Vojtěch Forejt, Xueyi Zou, Yi Zhang, Zak Cohen, …

  (and many more collaborators on case studies & projects)

- Project funders

EPSRC · European Commission · erc · DARPA · BRITISH COUNCIL · dti · Microsoft Research · QinetiQ · OXFORD MARTIN SCHOOL · Google SUMMER of CODE 2014