Models and Languages for Computational Systems Biology

Lecture 9: Continuous Time

Ian Stark

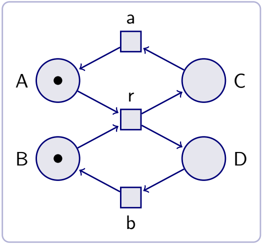
School of Informatics
The University of Edinburgh

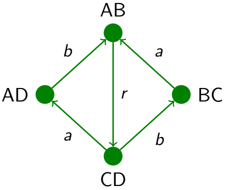
Monday 8 February 2010
Semester 2 Week 5

http://www.inf.ed.ac.uk/teaching/courses/mlcsb

Outline

How many states?



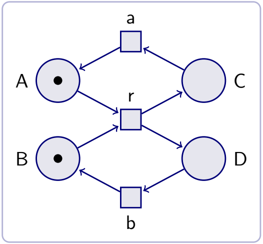


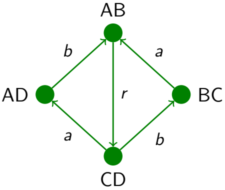
Identify at which states the CTL formula $\mathbf{AF}(\mathbf{EX}(\mathrm{AD}) \wedge \mathbf{EX}(\mathrm{BC}))$ holds.

Model-checking allows us to evaluate high-level temporal queries about behaviour of structured processes.

However, this becomes more work as the number of possible states increases.

How many states?





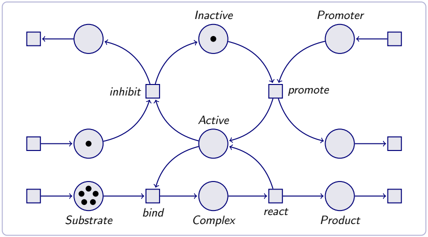
Identify at which states the CTL formula $\mathbf{AF}(\mathbf{EX}(\mathrm{AD}) \wedge \mathbf{EX}(\mathrm{BC}))$ holds.

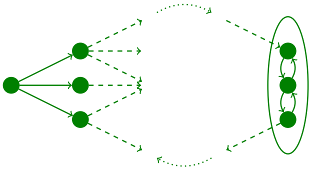
Model-checking allows us to evaluate high-level temporal queries about behaviour of structured processes.

However, this becomes more work as the number of possible states increases.

- 10, 100 states — compute directly by hand or eye.

How many states?





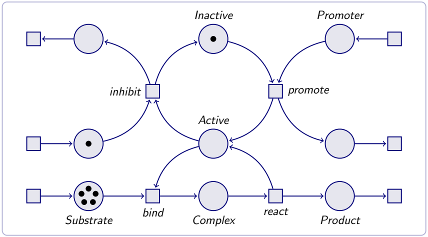
Identify at which states the CTL- formula $\langle \mathit{inhibit} \rangle \mathrm{E}(\neg(\mathit{Promoter})\ \mathbf{U}\ \mathit{Product})$ holds

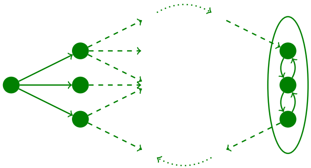
Model-checking allows us to evaluate high-level temporal queries about behaviour of structured processes.

However, this becomes more work as the number of possible states increases.

- 10, 100 states — compute directly by hand or eye.

How many states?





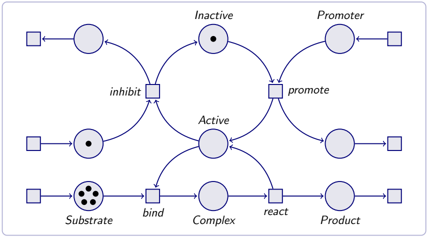
Identify at which states the CTL- formula $\langle \mathit{inhibit} \rangle \mathrm{E}(\neg(\mathit{Promoter})\ \mathbf{U}\ \mathit{Product})$ holds

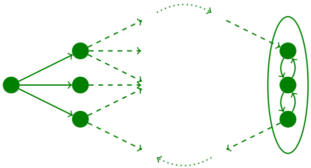
Model-checking allows us to evaluate high-level temporal queries about behaviour of structured processes.

However, this becomes more work as the number of possible states increases.

- 10, 100 states — compute directly by hand or eye.
- 1000, 10,000 states — direct, but by machine.

How many states?





Identify at which states the CTL- formula $\langle \mathit{inhibit} \rangle \mathrm{E}(\neg(\mathit{Promoter})\ \mathbf{U}\ \mathit{Product})$ holds

Model-checking allows us to evaluate high-level temporal queries about behaviour of structured processes.

However, this becomes more work as the number of possible states increases.

- 10, 100 states — compute directly by hand or eye.
- 1000, 10,000 states — direct, but by machine.
- 10^6, 10^{12}, ... — explicit state exploration infeasible.

From individuals to sets

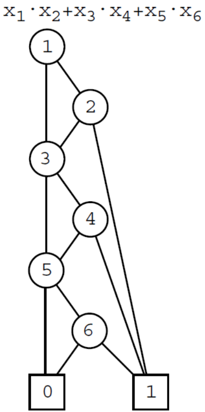
Working with individual states becomes infeasible. Instead, we can work with sets of states — which is anyway a better fit for branching-time assertions.

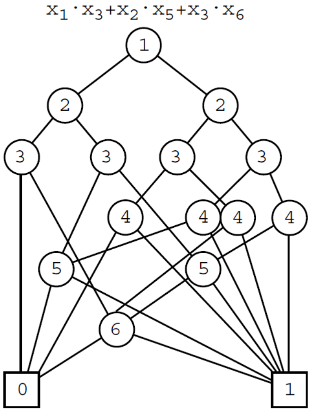
The key to success is choice of representation:

- Lists of states — no better.
- Bitmaps, trees — small improvement.
- Characteristic functions $\mathit{State} \to 2$ — very concise, limited manipulation (no comparison operation)
- Structured representation of characteristic functions: Binary Decision Diagrams — succinct, efficient operations, unique normal forms.

An efficient representation of sets can also express the transition relation $\to \subseteq \mathit{State} \times \mathit{State}$

Binary Decision Diagrams





Randal E. Bryant.
Graph-based algorithms for boolean function manipulation.
IEEE Transactions on Computers, C-35(8):677–691, 1986.

Think big

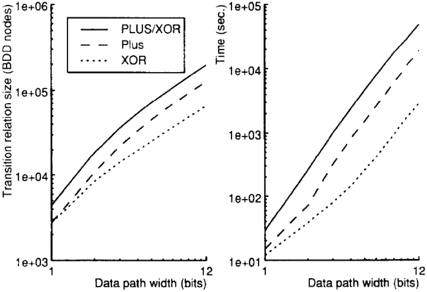


FIG. 4. Performance of BDD model checking algorithm on simple pipelines.

Figure 4 graphs the performance we obtained when checking formula 1 on a variety of pipelines of this type.

. . . The number of bits per register varied from 1 to 12. . . .

A pipeline with 12 bits has approximately 1.5×10^{29} reachable states.

Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill and L. J. Hwang
Symbolic model checking: 10^{20} states and beyond.
Information and Computation, 98(2):142–170, 1992.

Outline

Road map

- High-level model
  - Intuitive, human-readable; Modular, succinct, descriptive, ...
  - Precise, well-defined, compositional.
  - E.g. Petri Nets; later we shall also see process algebras in this rôle.
- Low-level semantic model
  - Describes system behaviour.
  - Can be mechanically generated from a high-level model.
  - E.g. labelled transition systems; later Markov chains and ODEs.
- Analysis
  - Pose questions about the model.
  - Questions and answers phrased on high-level model.
  - Mechanically evaluated on low-level semantics.
  - E.g. model checking temporal logic; but will also see simulation,

Let's do it again

So far we have seen this with qualitative, discrete models:

Petri nets $\longleftrightarrow$ transition systems $\longleftrightarrow$ LTL, CTL, CTL*, HML,. . .

We can now repeat this:

- Continuous time, rates and probabilities;
- Continuous space, amounts and concentrations;
- Qualitative vs. quantitative;
- Markov chains, ODEs, process algebras,. . .

Outline

Introducing time

All models so far have been qualitative: a behaviour is possible, or not.

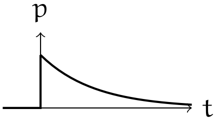
We introduce quantitative information about the relative probabilities of behaviours with stochastic actions: events that happen at a random time.

We qualify "random" by concentrating on events whose behaviour is stochastic, but time-invariant. This gives rise to a *negative exponential* probability distribution over time, with a single parameter, the *rate* at which the event happens.

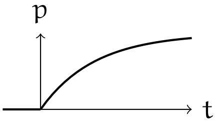
Negative exponential distribution

If random variable X has negative exponential distribution at rate r, then we have the following:

Probability density $\quad f(X,r) = \begin{cases} re^{-rt} & t \geqslant 0 \\ 0 & t < 0 \end{cases}$



Cumulative distribution $\quad F(X,r) = \begin{cases} 1 - e^{-rt} & t \geqslant 0 \\ 0 & t < 0 \end{cases}$



Mean $\quad E(X) = \dfrac{1}{r}$

Variance $\quad \mathrm{Var}(X) = \dfrac{1}{r^2}$

Properties of the negative exponential

- Memoryless:

$$\forall t, u > 0 \,.\, P(t < X < t + u \mid X > t) = P(0 < X < u)$$

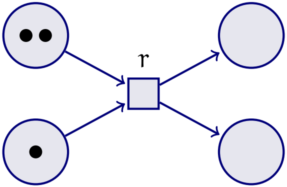
- Minimum of independent negative exponentials:

$$X \sim \mathrm{Exp}(r) \quad \& \quad Y \sim \mathrm{Exp}(s) \quad \Longrightarrow \quad \min(X, Y) \sim \mathrm{Exp}(r + s)$$

Suppose a repeating event happens at random, with λt occurrences expected within any time interval of length t (a *Poisson* process). Then the time until it next occurs, and the time between occurrences, are both random variables with negative exponential distribution, rate λ.

Similarly for space: the distance between mutations on a DNA strand has negative exponential distribution.

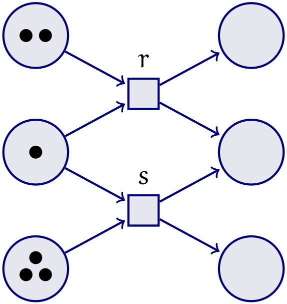
Stochastic Petri Nets



To introduce continuous time into a Petri nets, we label each transition with a rate.

When enabled, the transition fires at that rate (exponential distribution) until it is no longer enabled.

Stochastic Petri Nets

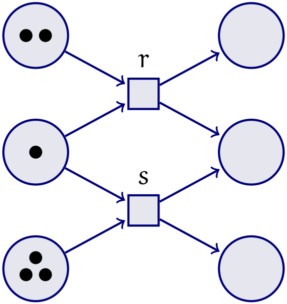


To introduce continuous time into a Petri nets, we label each transition with a rate.

When enabled, the transition fires at that rate (exponential distribution) until it is no longer enabled.

Where more than one transition is enabled, we have a race.

Stochastic Petri Nets



To introduce continuous time into a Petri nets, we label each transition with a rate.

When enabled, the transition fires at that rate (exponential distribution) until it is no longer enabled.

Where more than one transition is enabled, we have a race.

The combined rate of firing is $r + s$, with respective probabilities

$$\frac{r}{r+s} \qquad \text{and} \qquad \frac{s}{s+r} \;.$$

Next time

Qualitatively, a stochastic Petri net has exactly the same transitions and behaviours as the underlying Petri net.

Next time

Qualitatively, a stochastic Petri net has exactly the same transitions and behaviours as the underlying Petri net.

However, we can now distinguish between the probabilities of different runs, and their duration.

Next time

Qualitatively, a stochastic Petri net has exactly the same transitions and behaviours as the underlying Petri net.

However, we can now distinguish between the probabilities of different runs, and their duration.

We replace the derived transition system by a *Markov chain*, and in general a *continuous-time Markov chain* or *CTMC*.

Next time

Qualitatively, a stochastic Petri net has exactly the same transitions and behaviours as the underlying Petri net.

However, we can now distinguish between the probabilities of different runs, and their duration.

We replace the derived transition system by a *Markov chain*, and in general a *continuous-time Markov chain* or *CTMC*.

We can explore this behaviour through simulation, or by model-checking using a probabilistic temporal logic such as PCTL.

Outline

Reading

Homework

Read the following short articles.

- Lucas Laursen.
  Computational biology: Biological logic.
  Nature 462:408–410, 2009.

- Jasmin Fisher and Thomas A Henzinger.
  Executable cell biology.
  Nature Biotechnology 25:1239–1249, 2007.

- Nathalie Chabrier and Fran cois Fages.
  Symbolic model checking of biochemical networks.
  In *Computational Methods in Systems Biology, Proceedings of the First International Workshop, CMSB 2003.* Lecture Notes in Computer Science 2602, pages 149–162. Springer-Verlag, 2003

References (1/2)

Randal E. Bryant.
Graph-based algorithms for boolean function manipulation.
IEEE Transactions on Computers, C-35(8):677–691, 1986.
Reprinted in M. Yoeli, editor, *Formal Verification of Hardware Design*, IEEE Computer Society Press, 1990, pages 253–267. Electronic version with annotations available at http://www.cs.cmu.edu/~bryant/pubdir.

Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill and L. J. Hwang
Symbolic model checking: 10^{20} states and beyond.
Information and Computation, 98(2):142–170, 1992.

References (2/2)

- Falko Bause and Pieter S. Kritzinger.
  Stochastic Petri Nets.
  Vieweg Verlag, 2nd edition, 2002.
  Out of print, but electronic version available at
  http://www4.cs.uni-dortmund.de/~Bause/spnbook2.html

- M. Ajmone Marsan.
  Stochastic Petri nets: An elementary introduction.
  In *Advances in Petri Nets 1989*, Lecture Notes in Computer Science 424, pages 1–29. Springer-Verlag, 1990.