Quantitative logics

David N. Jansen

Objectives

overview over quantitative logics and (probabilistic) model checking

Objectives

- express requirements and create behaviour models
- know possibilities and limitations of model checking
- know a variety of quantitative logics
  and their model checking algorithms
- You could write (the mathematical logic part of)
  a simple model checker.

- ~~optimisation methods (BDDs, symbolic model checking)~~
- ~~solve linear equation systems~~

Practical matters

- website:
  https://lab.cs.ru.nl/algemeen/Quantitative_logics
- exercises: no obligations
- written exam

What do you know already?

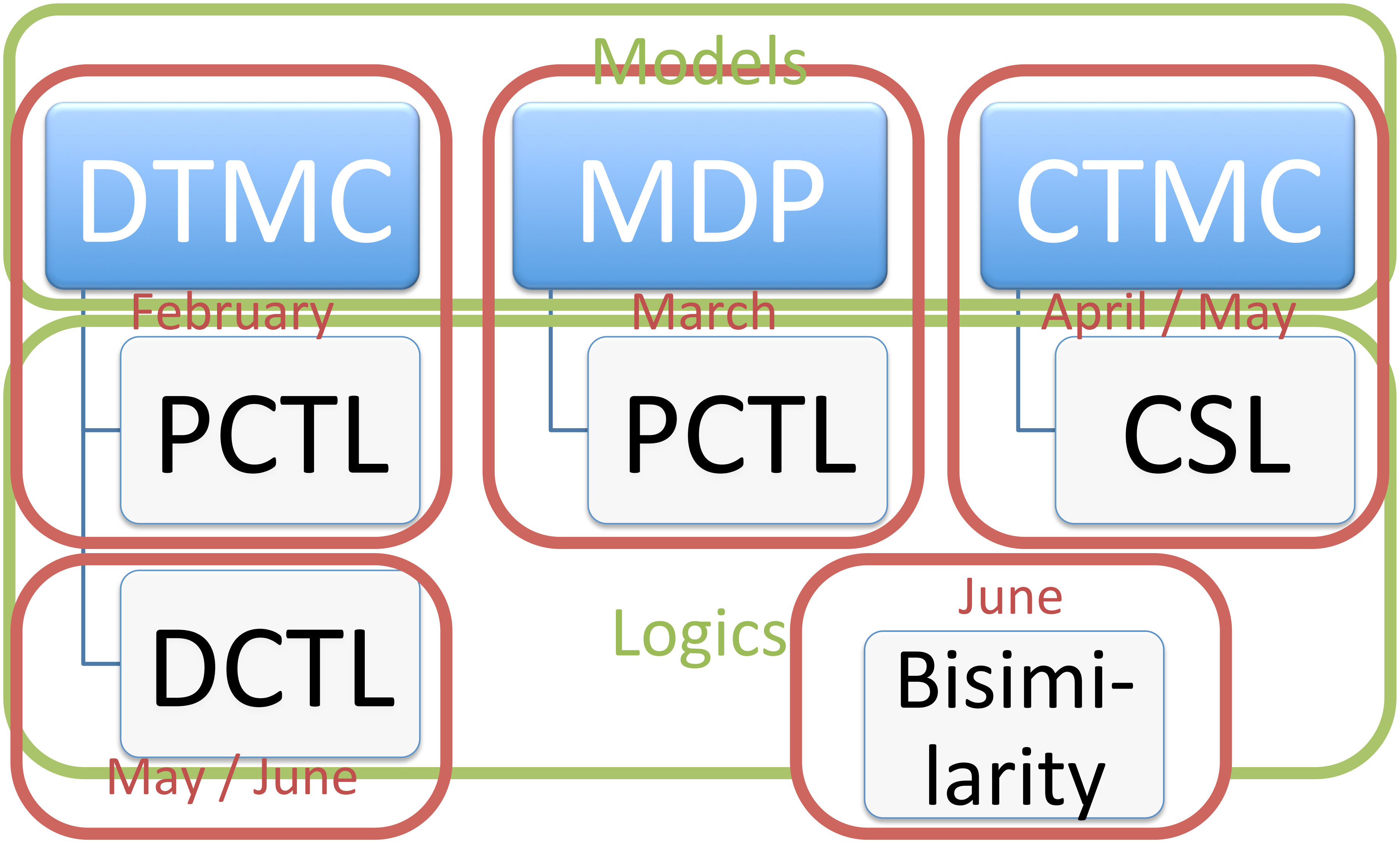
- model checking?
- probability theory?
- Markov chains?
- temporal logic?

Today’s programme

- General overview

- Temporal logic
- Model checking basics
- Probability theory basics

Global overview

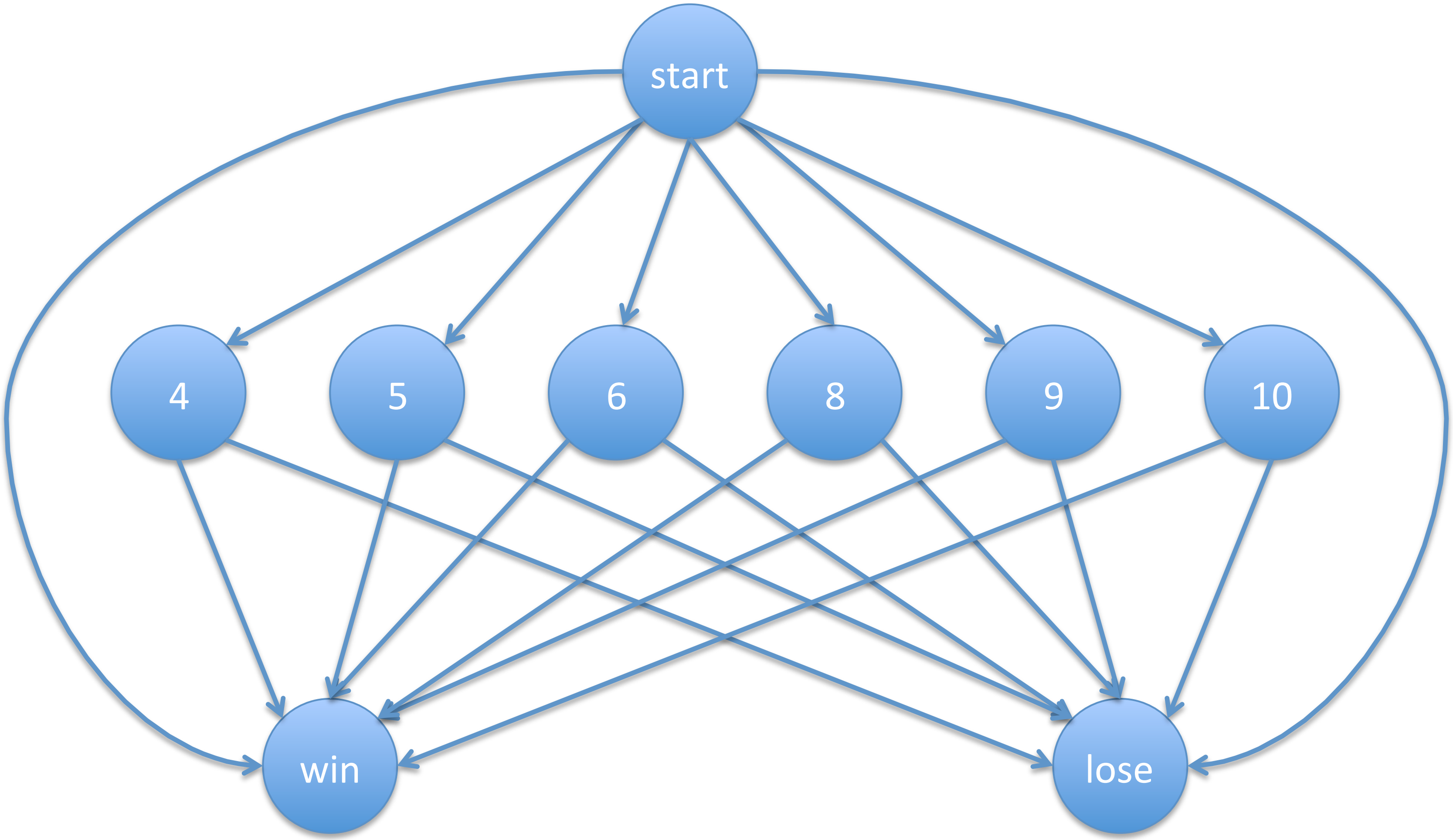


Models

- extensions of transition systems
  - discrete state space
  - transitions describe possible behaviours
- probabilistic choice between transitions
- CTMC: stochastic timing
  (probability distribution over how long a state lasts)

Example model: Craps game

- throw two dice and sum them
  - 7, 11: win immediately
  - 2, 3, 12: lose immediately
  - other result: this result is “point”
- throw the dice until the result is:
  - 7: lose
  - point: win
- can be modelled as a DTMC


start
4
5
6
8
9
10
win
lose

Temporal logic

Principles

- want to describe properties of behaviours
  - behaviour := sequence of computation steps
- extend propositional logic
  - use propositions to describe one state
- modality operators:
  - at some time $\Diamond$ F(uture)
  - always $\Box$ G(enerally)

General modal logics

- modality operators
  allow general interpretation:
  - in some possible world: $\Diamond$
  - in all possible worlds: $\Box$
- different varieties
  - temporal — possible worlds = future
  - knowledge — possible worlds = consistent with knowledge
  - deontic — possible worlds = consistent with moral obligations

CTL: additional modalities

- modality operators:
  - at some time ◇ F(uture)
  - always □ G(enerally)

  - in some future E
  - in all futures A

Examples: mutual exclusion

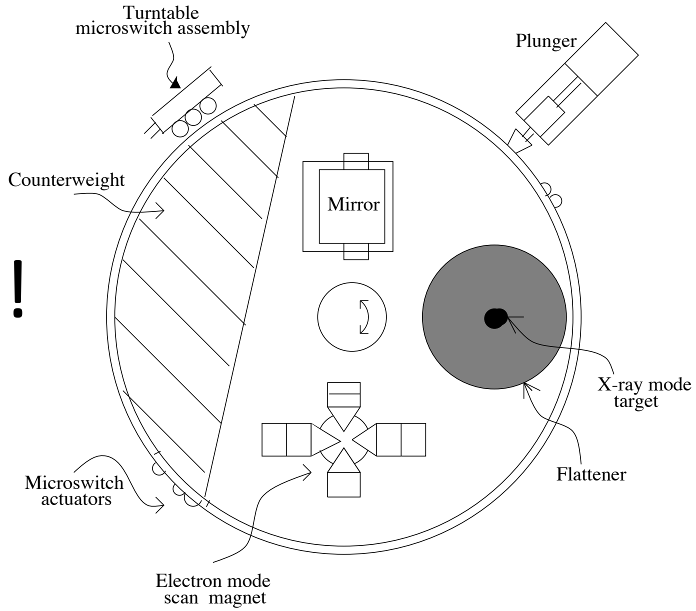
- Propositions:
  - $crit_i$ component i is in critical section
  - try_i component i wants to enter critical section
- It will never happen that both components are in their critical sections.
- In every state, a component may eventually enter its critical section.
- Whenever a component tries to enter its critical section, it will do so eventually.

Model checking basics

Why model checking?

- Therac-25: a medical irradiation device with deadly software
- generates electron beam
- can be converted to X-rays
  - about 1% efficiency
- wrong position → overdose!

- ≥5 patients deceased



System verification

- solves **some** problems
  with software correctness

- verification:
  check whether system meets specification

- silent assumption: specification is correct
  + model behaves as system

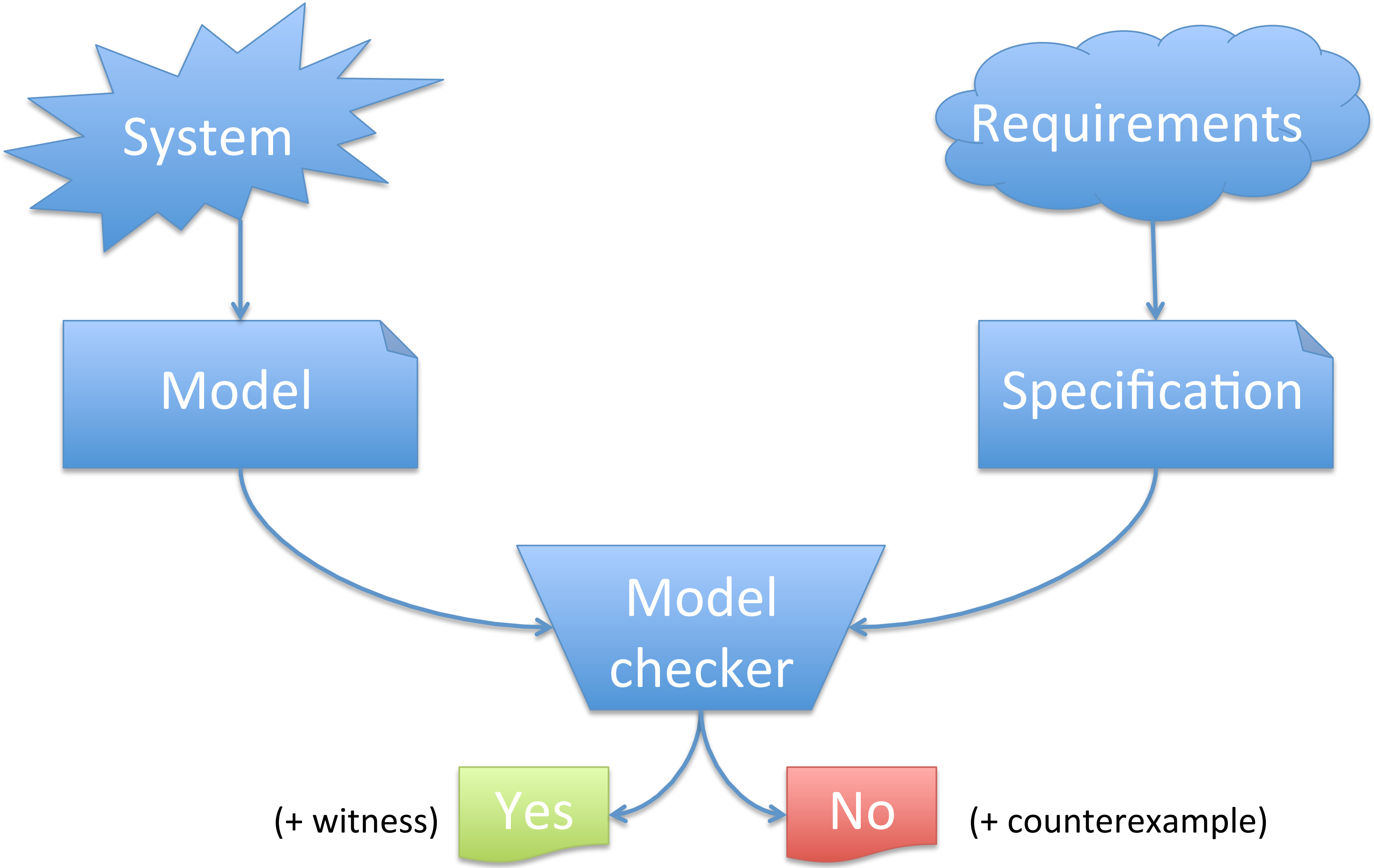
Formal system verification

- use mathematics
  to model and analyse ICT systems
- two main methods:
  - deductive proof
    - system model is a mathematical theory
    - often computer-assisted
  - model checking
    - system model is a finite automaton (or similar)
    - in principle fully automatic

Model checking: idea

- create a model of the behaviour
- specify desired behaviours
- look for a formal proof (automatically)
  - (proof rules should be simple enough that provability is decidable)

Model checking: idea



Property specification language: temporal logic

- typical properties:
  - Can the system reach a deadlock?
  - Can two processes be in the critical section?
  - Is the output correct upon termination?
- standard property:
  - Can the system reach an undesired state?
  - Will the system reach a desired state?

Mutual exclusion

- Requirements in CTL

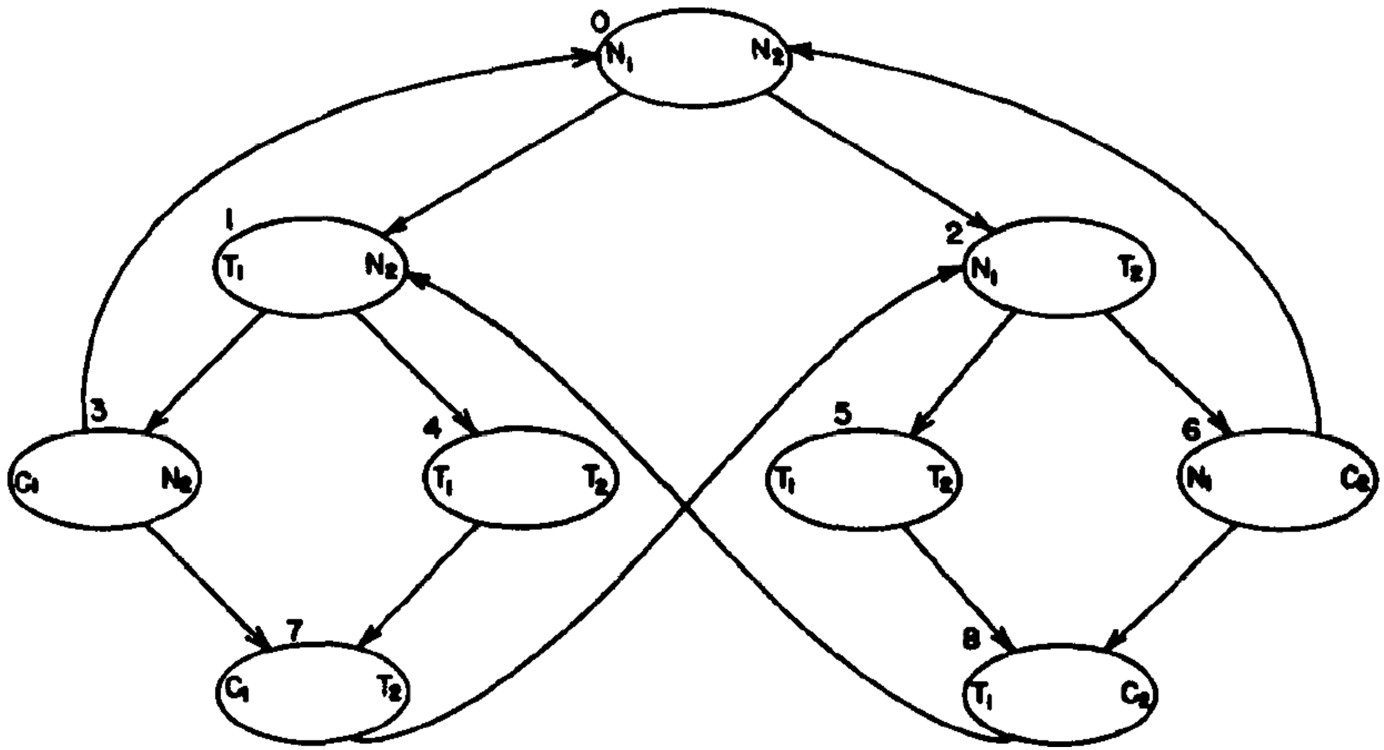
  „If process 1 tries to execute its critical section (T_1), it will eventually enter it (C_1).“

  $T_1 \rightarrow A\Diamond C_1$

- Subformulas:

  T_1 $\quad$ C_1 $\quad$ $\neg T_1$ $\quad$ $A\Diamond C_1$ $\quad$ $\neg T_1 \vee A\Diamond C_1$

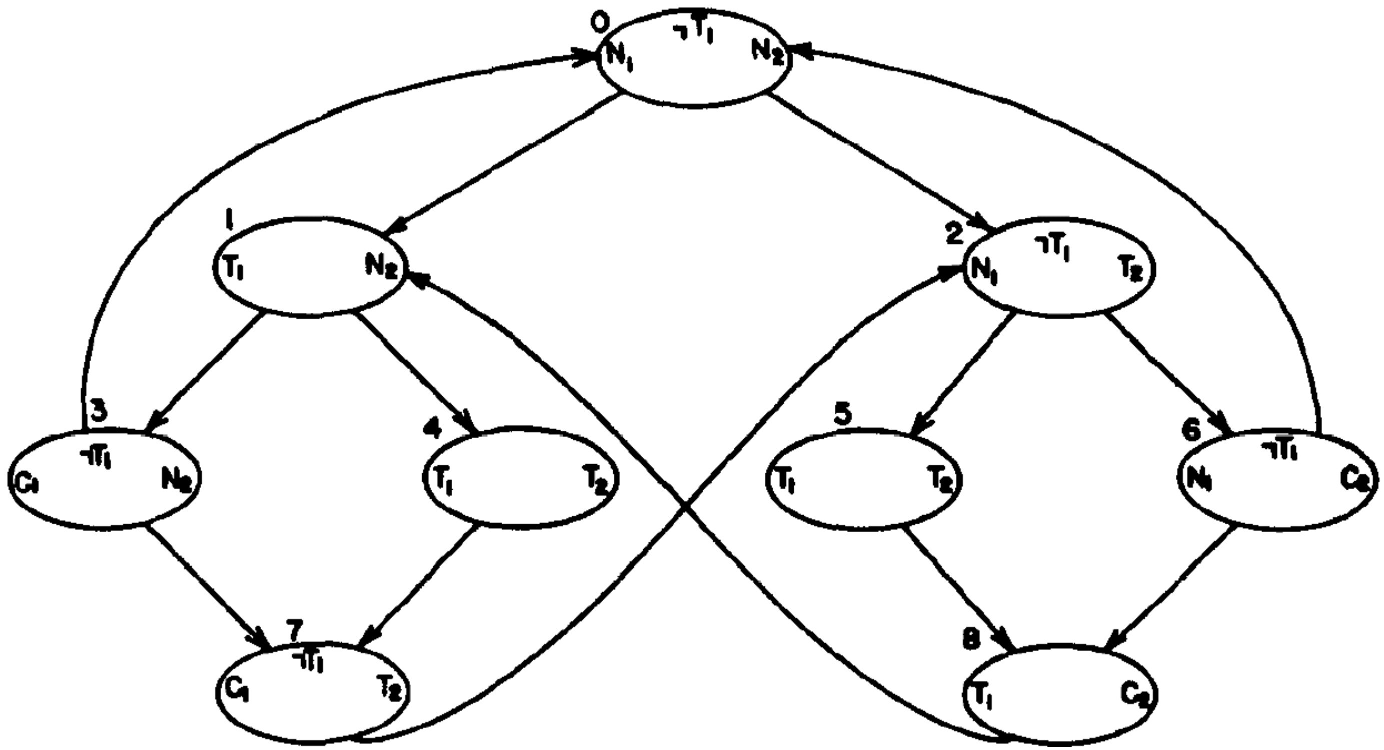
Model checking



$$\neg T_1 \vee A\Diamond C_1$$

© 1986 ACM Trans. Programming Languages and Systems

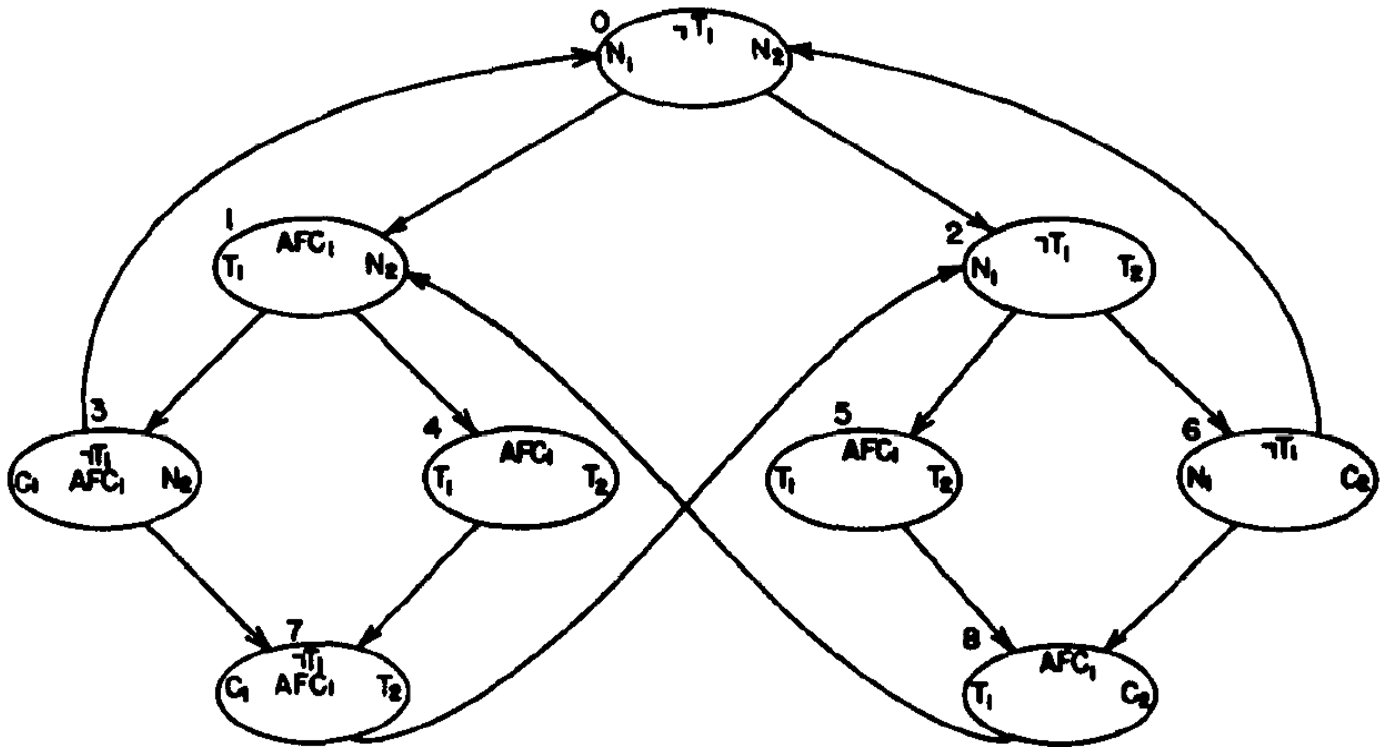
Model checking



$$\neg T_1 \vee A\Diamond C_1$$

© 1986 ACM Trans. Programming Languages and Systems

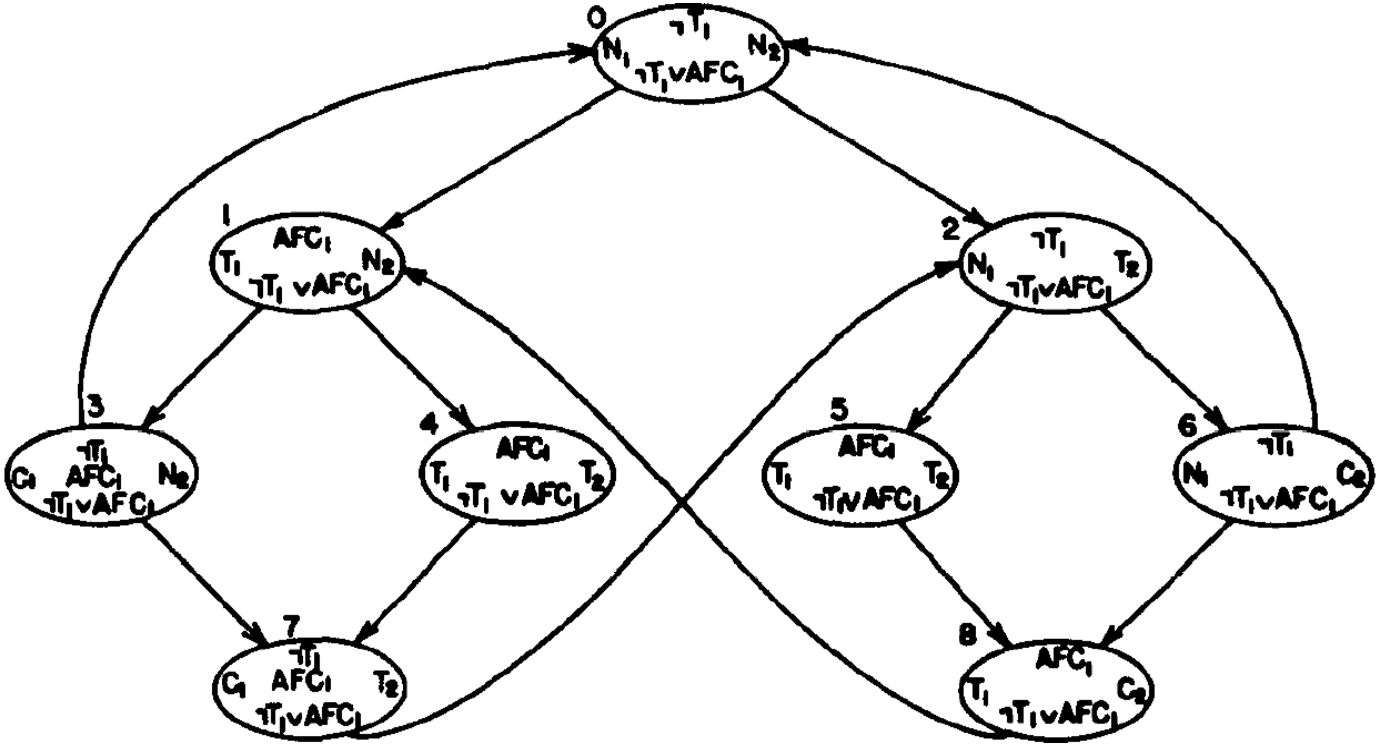
Model checking



© 1986 ACM Trans. Programming Languages and Systems

$\neg T_1 \vee A\Diamond C_1$

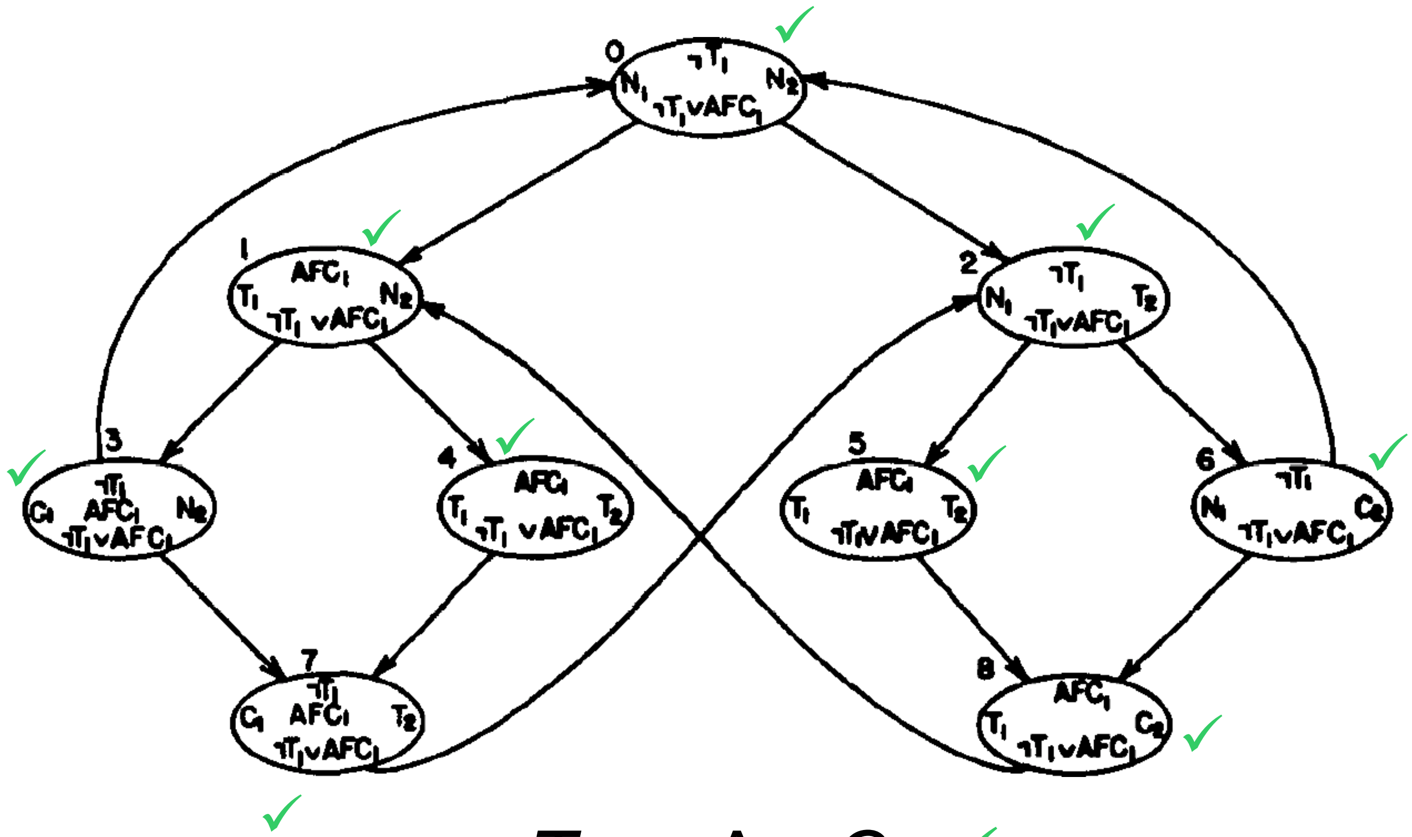
Model checking



$\neg T_1 \vee A\Diamond C_1$

© 1986 ACM Trans. Programming Languages and Systems

Model checking



$T_1 \rightarrow A\diamond C_1$ ✓

© 1986 ACM Trans. Programming Languages and Systems

Recapitulation

- System verification makes sure the system satisfies its specification.
- Model checking is a method for system verification, (in principle) fully automatic.
- Model checking reads
  - a behavioural system model (transition system)
  - a property (temporal logic)

Probability theory

What are probabilities?

- general: a measure how likely an outcome is

- frequentist interpretation:
  the expected number an outcome appears
  if an experiment is repeated often
- bets interpretation:
  the proportion of money someone bets
  on a single outcome

Where do probabilities come from?

- mathematical concept
- to model random process
  (relation between cause and result is not completely known)
  - metaphysical randomness
    (cause does not completely determine the result)
  - I am not interested in/I do not know the cause
  - No scientist (currently) knows the exact cause
    (but given enough time & money, one could find out)
  - The cause lies outside the realm of science

How to define a probability space

- Example: throw a die
- Possible outcomes: Ω = {⚀,⚁,⚂,⚃,⚄,⚅}
- Probability weight: P(⚀) = $\frac{1}{6}$
  P(⚁) = $\frac{1}{6}$
  etc.

Notation

- $P(A)$ = probability that A happens
- $P(A,B)$ = probability that both A and B happen
- $P(A|B)$ = probability that A happens under the condition that B has happened

- $P(A|B) = P(A,B) \, / \, P(B)$

conditional probability

Throw more dice

- How probable is an even outcome?
  - $P(\text{even}) = P(⚁) + P(⚃) + P(⚅) = \frac{1}{2}$
- How probable is the outcome ⚅, given that we know the outcome is even?
  - $P(⚅|\text{even}) = P(⚅,\text{even}) / P(\text{even}) = \frac{1}{6} / \frac{1}{2} = \frac{1}{3}$
  - $P(\text{even}|⚅) = \underline{P}(\text{even}, ⚅) / \mathrm{P}(⚅) = 1$

Another example

- Some process (e. g. solving a quiz) takes between 2 and 5 minutes, each duration D having equal probability.
- What is the probability that it takes exactly $D = 3.1415$ minutes?
- $1 = \sum_{x \in [2,5]} P(D = x) = \sum_{x \in [2,5]} p_0 = \infty \cdot p_0$
- and therefore $p_0 = 1 / \infty = 0$
- $P(D \leq 3) = \sum_{x \in [2,3]} p_0 = \sum_{x \in [2,3]} 0 = 0$

Oops?

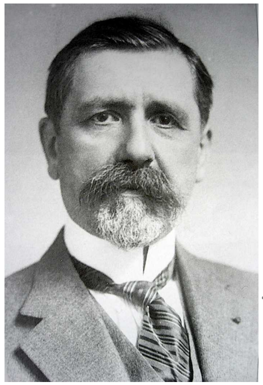
A better definition

- assign probabilities to subsets of Ω in a systematic way
- A σ-algebra $\mathcal{A}$ is a set of subsets:
  - $\Omega \in \mathcal{A}$
  - $A \in \mathcal{A} \rightarrow \Omega \backslash A \in \mathcal{A}$
  - $A_i \in \mathcal{A}$ for all $i = 1,2,\ldots \rightarrow \bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$
- Generally: It is sensible to assign a probability to each set in the σ-algebra.

Example: Borel-σ-algebra

- $\Omega = \mathbb{R}$
- $\mathcal{B}$ = the smallest σ-algebra
  that contains all intervals $[r,s)$, for $r,s \in \mathbb{R}$

- standard σ-algebra for the real numbers
- Émile Borel, French mathematician,
  1871–1956, wrote *Le Hasard*



neurotree.org