# Lecture 4
# Probabilistic temporal logics

Alessandro Abate

UNIVERSITY OF OXFORD

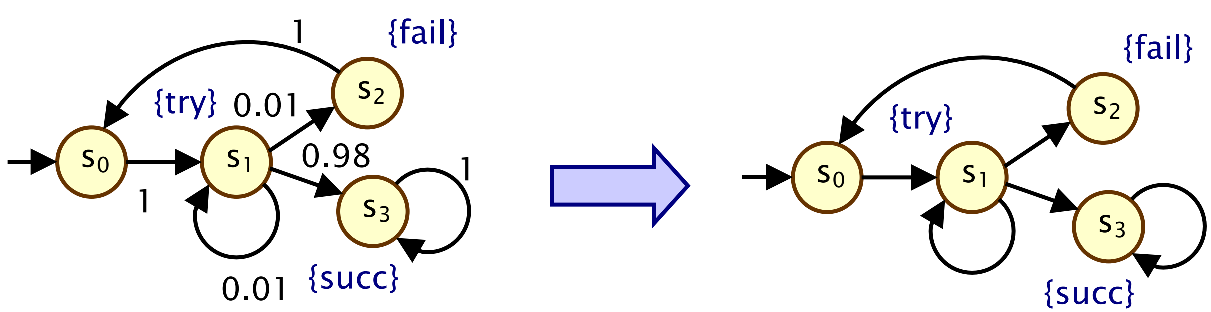
Department of Computer Science
University of Oxford

# Overview

- Temporal logics
- Non-probabilistic temporal logic
  - CTL
- Probabilistic temporal logic
  - PCTL = CTL + probabilities
- Qualitative vs. quantitative
- Linear-time properties
  - LTL, PCTL*

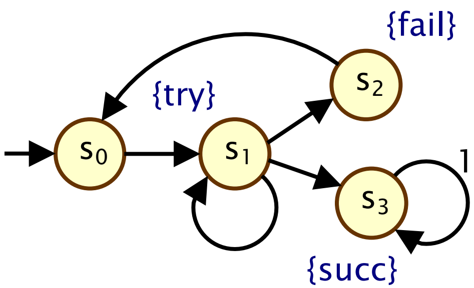
# Temporal logic

- Temporal logic
  - formal language for specifying and reasoning about how the behaviour of a system changes over time
  - extends propositional logic with modal/temporal operators
  - one important use: representation of properties of reactive system, to be verified by a model checker
- Logics used in this course are probabilistic extensions of temporal logics devised for non-probabilistic systems
  - So we revert briefly to (labelled) state-transition diagrams



# State-transition systems

- Labelled state-transition system (LTS) (or Kripke structure)
  - is a tuple $(S, s_{init}, \rightarrow, L)$ where:
  - $S$ is a set of states ("state space")
  - $s_{init} \in S$ is the initial state
  - $\rightarrow \subseteq S \times S$ is the transition relation
  - $L : S \rightarrow 2^{AP}$ is function labelling states with atomic propositions (taken from a set AP)




- DTMC $(S, s_{init}, \mathbf{P}, L)$ has underlying LTS $(S, s_{init}, \rightarrow, L)$
  - where $\rightarrow = \{ (s,s') \text{ s.t. } \mathbf{P}(s,s') > 0 \}$

# Paths – some notation

- Path $\omega = s_0 s_1 s_2 \ldots$ such that $(s_i, s_{i+1}) \in \rightarrow$ for $i \geq 0$
  - we write $s_i \rightarrow s_{i+1}$ as shorthand for $(s_i, s_{i+1}) \in \rightarrow$
- $\omega(i)$ is the (i+1)th state of $\omega$, i.e. $s_i$
- $\omega[\ldots i]$ denotes the (finite) prefix ending in the (i+1)th state
  - i.e. $\omega[\ldots i] = s_0 s_{1 \ldots} s_i$
- $\omega[i \ldots]$ denotes the suffix starting from the (i+1)th state
  - i.e. $\omega[i \ldots] = s_i s_{i+1} s_{i+2} \ldots$
- As for DTMCs, Path(s) = set of all infinite paths from s

# CTL

- CTL – Computation Tree Logic
- Syntax split into state and path formulae
  - specify properties of states/paths, respectively
  - a CTL formula is a state formula

- State formulae:
  - $\phi$ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | A $\psi$ | E $\psi$
  - where a $\in$ AP and $\psi$ is a path formula

- Path formulae
  - $\psi$ ::= X $\phi$ | F $\phi$ | G $\phi$ | $\phi$ U $\phi$
  - where $\phi$ is a state formula

Some of these operators (e.g. A, F, G) are derivable from others

X = "next"
F = "finally"
G = "globally"
U = "until"

# CTL semantics

- Semantics of state formulae:
  - $s \models \phi$ denotes "s satisfies $\phi$" or "$\phi$ is true in s"

- For a state s of an LTS $(S, s_{init}, \rightarrow, L)$:
  - $s \models true$ always
  - $s \models a \Leftrightarrow a \in L(s)$
  - $s \models \phi_1 \wedge \phi_2 \Leftrightarrow s \models \phi_1$ and $s \models \phi_2$
  - $s \models \neg\phi \Leftrightarrow s \not\models \phi$
  - $s \models A\,\psi \Leftrightarrow \omega \models \psi$ for all $\omega \in Path(s)$
  - $s \models E\,\psi \Leftrightarrow \omega \models \psi$ for some $\omega \in Path(s)$

# CTL semantics

- Semantics of path formulae:
  - $\omega \models \psi$ denotes "$\omega$ satisfies $\psi$" or "$\psi$ is true along $\omega$"

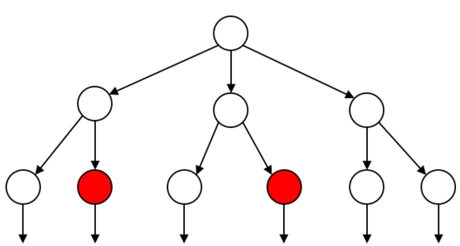
- For a path $\omega$ of an LTS $(S,s_{init},\rightarrow,L)$:

  - $\omega \models X\ \phi \quad \Leftrightarrow \quad \omega(1) \models \phi$
  - $\omega \models F\ \phi \quad \Leftrightarrow \quad \exists k \geq 0$ s.t. $\omega(k) \models \phi$
  - $\omega \models G\ \phi \quad \Leftrightarrow \quad \forall i \geq 0\ \omega(i) \models \phi$
  - $\omega \models \phi_1\ U\ \phi_2 \quad \Leftrightarrow \quad \exists k \geq 0$ s.t. $\omega(k) \models \phi_2$ and $\forall i<k\ \omega(i) \models \phi_1$

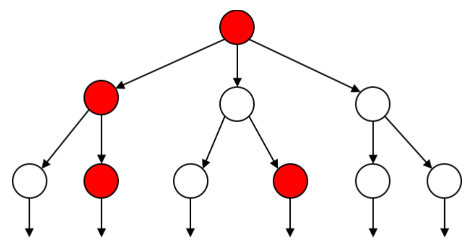
  - ( incidentally, $F\ \phi = \mathit{true}\ U\ \phi$ )

# CTL semantics

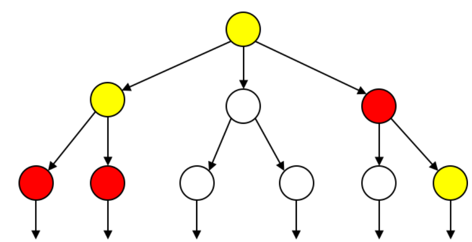
- Intuitive semantics:
  - of quantifiers (A/E) and temporal operators (F/G/U)



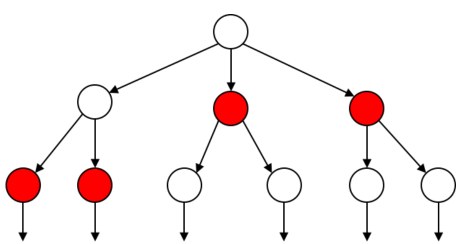
EF red



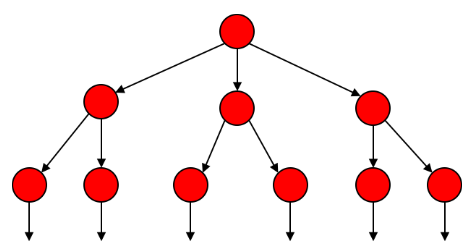
EG red



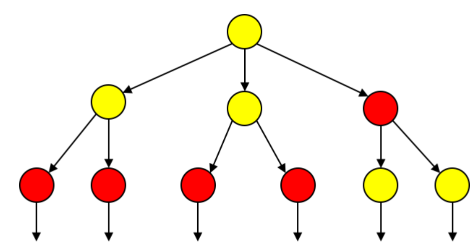
E [ yellow U red ]



AF red



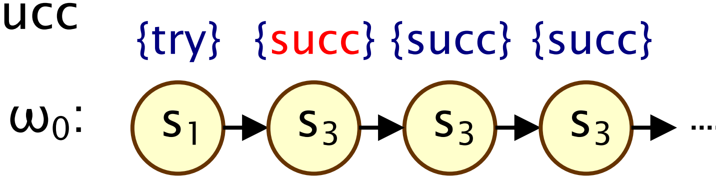
AG red



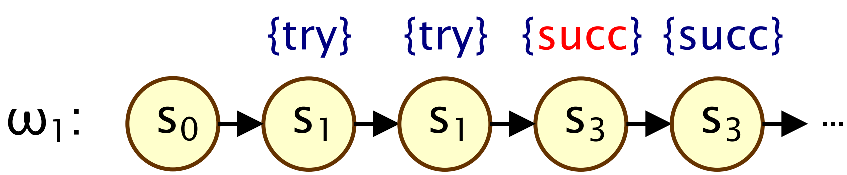
A [ yellow U red ]

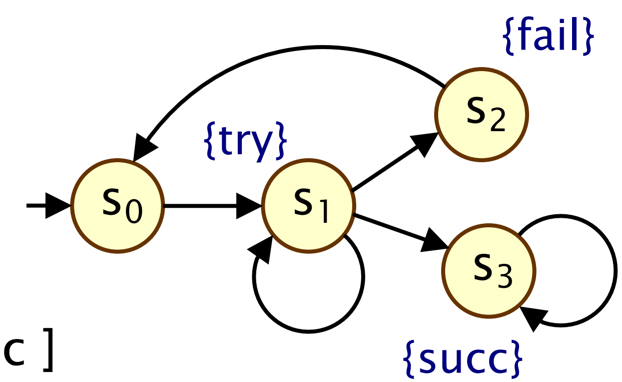
# CTL examples

- Some examples of satisfying paths:
  - $\omega_0 \vDash$ X succ




  - $\omega_1 \vDash \neg$fail U succ




- Example CTL formulae:
  - $s_1 \vDash$ try $\wedge$ $\neg$fail
  - $s_1 \vDash$ E [ X succ ] and $s_3 \vDash$ A [ X succ ]
  - $s_0 \vDash$ E [$\neg$fail U succ] but $s_0 \nvDash$ A [$\neg$fail U succ]

# CTL examples

- $AG\ (\neg(crit_1 \wedge crit_2))$
  - mutual exclusion
- $AG\ EF\ initial$
  - for every computation, it is always possible to return to the initial state
- $AG\ (request \rightarrow AF\ response)$
  - every request will eventually be granted
- $AG\ AF\ crit_1 \wedge AG\ AF\ crit_2$
  - for both critical sections, each process has access to each infinitely often

# CTL equivalences

- Basic propositional logic equivalences:
  - false $\equiv$ $\neg$true (false)
  - $\phi_1 \vee \phi_2 \equiv \neg(\neg\phi_1 \wedge \neg\phi_2)$ (disjunction)
  - $\phi_1 \rightarrow \phi_2 \equiv \neg\phi_1 \vee \phi_2$ (implication)

- Path quantifiers (proof via semantics):
  - $A\ \psi \equiv \neg E(\neg\psi)$
  - $E\ \psi \equiv \neg A(\neg\psi)$

- Temporal operators for paths:
  - $F\ \phi \equiv \text{true}\ U\ \phi$
  - $G\ \phi \equiv \neg F(\neg\phi)$

Hence, e.g.:

$AG\ \phi \equiv \neg EF(\neg\ \phi)$

# CTL – Alternative notation

- Some commonly used notation (cf. [BK08] book)

- Temporal operators:
  - $F\,\phi \equiv \Diamond\,\phi$ ("diamond")
  - $G\,\phi \equiv \Box\,\phi$ ("box")
  - $X\,\phi \equiv \bigcirc\,\phi$
- Path quantifiers:
  - $A\,\psi \equiv \forall\,\psi$
  - $E\,\psi \equiv \exists\,\psi$

- Bracketing: none/round/square
  - $AF\,\psi$
  - $A\,(\,\psi_1\ U\ \psi_2\,)$
  - $A\,[\,\psi_1\ U\ \psi_2\,]$

# PCTL

- Temporal logic for describing properties of DTMCs
  - PCTL = Probabilistic Computation Tree Logic [HJ94]
  - essentially the same as the logic pCTL of [ASB+95]

- Extension of (non-probabilistic) temporal logic CTL
  - key addition is probabilistic operator P
  - quantitative extension of CTL's A and E operators

- Example
  - send $\rightarrow P_{\geq 0.95}$ [ $F^{\leq 10}$ deliver ]
  - "if a message is sent, then the probability of it being delivered within 10 steps is at least 0.95"

# PCTL syntax

- PCTL syntax:

$\psi$ 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, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$

- A PCTL formula is always a state formula (same as CTL)
  - path formulae only occur inside the P operator

# PCTL semantics for DTMCs

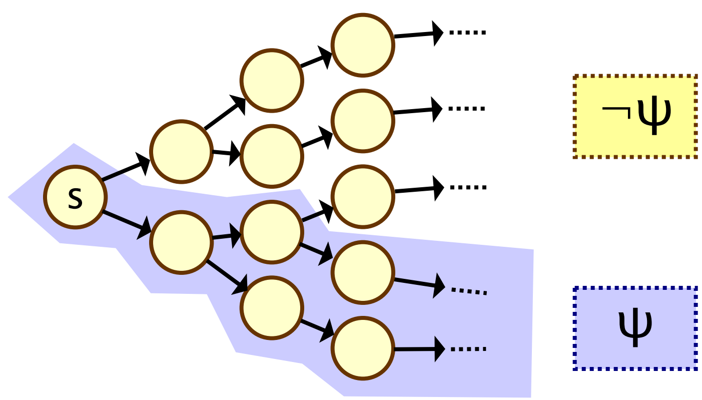
- Semantics for non-probabilistic operators same as for CTL:
  - $s \models \phi$ denotes "s satisfies $\phi$" or "$\phi$ is true in s"
  - $\omega \models \psi$ denotes "$\omega$ satisfies $\psi$" or "$\psi$ is true along $\omega$"

- For a state s of a DTMC $(S,s_{init},\mathbf{P},L)$:
  - $s \models \text{true}$ always
  - $s \models a \Leftrightarrow a \in L(s)$
  - $s \models \phi_1 \wedge \phi_2 \Leftrightarrow s \models \phi_1$ and $s \models \phi_2$
  - $s \models \neg\phi \Leftrightarrow s \not\models \phi$
- For a path $\omega$ of a DTMC $(S,s_{init},\mathbf{P},L)$:
  - $\omega \models X\,\phi \Leftrightarrow \omega(1) \models \phi$
  - $\omega \models \phi_1\, U^{\leq k}\, \phi_2 \Leftrightarrow \exists i \leq k$ such that $\omega(i) \models \phi_2$ and $\forall j<i,\ \omega(j) \models \phi_1$
  - $\omega \models \phi_1\, U\, \phi_2 \Leftrightarrow \exists k \geq 0$ s.t. $\omega(k) \models \phi_2$ and $\forall i<k\ \omega(i) \models \phi_1$

$U^{\leq k}$ not in CTL (but could easily be added)

# PCTL semantics for DTMCs

- Semantics of the probabilistic operator P
  - informal definition: $s \vDash P_{\sim p} [\ \psi\ ]$ means that "the probability, from state s, that $\psi$ holds on outgoing paths, satisfies $\sim p$"
  - example: $s \vDash P_{<0.25} [\ X\ \mathit{fail}\ ] \Leftrightarrow$ "the probability of atomic proposition *fail* being true in the next state of outgoing paths from s is less than 0.25"
  - formally: $s \vDash P_{\sim p} [\psi] \ \Leftrightarrow\ Prob(s, \psi) \sim p$
  - where: $Prob(s, \psi) = Pr_s \{\ \omega \in Path(s) \mid \omega \vDash \psi\ \}$




$Prob(s, \psi) \sim p$ ?

# PCTL equivalences for DTMCs

- Basic logical equivalences:
  - $\mathrm{false} \equiv \neg\mathrm{true}$ (false)
  - $\phi_1 \vee \phi_2 \equiv \neg(\neg\phi_1 \wedge \neg\phi_2)$ (disjunction)
  - $\phi_1 \rightarrow \phi_2 \equiv \neg\phi_1 \vee \phi_2$ (implication)
- Negation and probabilities
  - e.g. $\neg P_{>p} [\ \phi_1\ U\ \phi_2\ ] \equiv P_{\leq p} [\ \phi_1\ U\ \phi_2\ ]$

# Reachability and invariance

- Derived temporal operators, like CTL…

- Probabilistic reachability: $P_{\sim p}$ [ F $\phi$ ]
  - the probability of reaching a state satisfying $\phi$
  - $F\ \phi \equiv \text{true}\ U\ \phi$
  - "$\phi$ is eventually true"
  - bounded version: $F^{\leq k}\ \phi \equiv \text{true}\ U^{\leq k}\ \phi$

- Probabilistic invariance: $P_{\sim p}$ [ G $\phi$ ]
  - the probability of $\phi$ always remaining true
  - $G\ \phi \equiv \neg(F\ \neg\phi) \equiv \neg(\text{true}\ U\ \neg\phi)$
  - "$\phi$ is always true"
  - bounded version: $G^{\leq k}\ \phi \equiv \neg(F^{\leq k}\ \neg\phi)$

strictly speaking, G $\phi$ cannot be derived from the PCTL syntax in this way since there is no negation of PCTL path formulae

# Derivation of $P_{\sim p}$ [ G $\phi$ ]

- In fact, we can derive $P_{\sim p}$ [ G $\phi$ ] directly in PCTL…

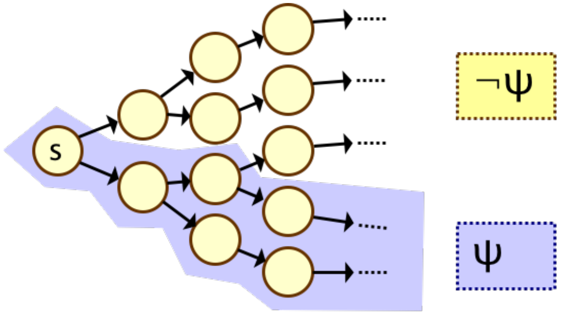
  – $s \vDash P_{>p} [\, G\ \phi \,] \;\Leftrightarrow\; \mathrm{Prob}(s, G\ \phi) > p$

$$\Leftrightarrow \mathrm{Prob}(s, \neg(F\ \neg\phi)) > p$$

$$\Leftrightarrow 1 - \mathrm{Prob}(s, F\ \neg\phi) > p$$

$$\Leftrightarrow \mathrm{Prob}(s, F\ \neg\phi) < 1 - p$$

$$\Leftrightarrow s \vDash P_{<1-p} [\, F\ \neg\phi \,]$$



# Derivation of $P_{\sim p}$ [ G $\phi$ ]

- In fact, we can derive $P_{\sim p}$ [ G $\phi$ ] directly in PCTL…

  - $s \models P_{>p}\,[\,G\ \phi\,] \;\Leftrightarrow\; Prob(s, G\ \phi) > p$

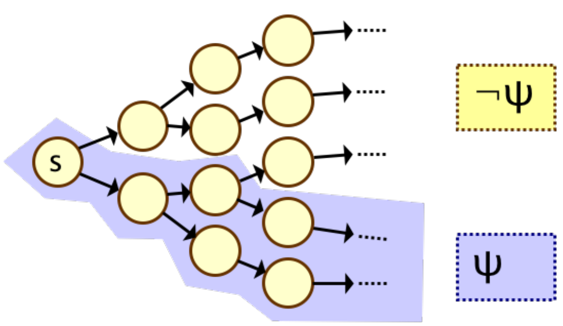
    $\Leftrightarrow\; Prob(s, \neg(F\ \neg\phi)) > p$

    $\Leftrightarrow\; 1 - Prob(s, F\ \neg\phi) > p$

    $\Leftrightarrow\; Prob(s, F\ \neg\phi) < 1 - p$

    $\Leftrightarrow\; s \models P_{<1-p}\,[\,F\ \neg\phi\,]$

- Other equivalences:
  - $P_{\geq p}\,[\,G\ \phi\,] \;\equiv\; P_{\leq 1-p}\,[\,F\ \neg\phi\,]$
  - $P_{<p}\,[\,G\ \phi\,] \;\equiv\; P_{>1-p}\,[\,F\ \neg\phi\,]$
  - $P_{>p}\,[\,G^{\leq k}\ \phi\,] \;\equiv\; P_{<1-p}\,[\,F^{\leq k}\ \neg\phi\,]$
  - etc.



# PCTL examples

- $P_{<0.4}$ [ $\neg fail_A$ U $fail_B$ ]
  - "the probability that component B fails before component A is less than 0.4"
- $\neg oper \rightarrow P_{\geq 1}$ [ F ( $P_{>0.99}$ [ $G^{\leq 100}$ oper ] ) ]
  - "if the system is not operational, it almost surely reaches a state from which it has a greater than 0.99 chance of staying operational for 100 time units"
- $P_{<0.05}$ [ F err/total>0.1 ]
  - "with probability at most 0.05, more than 10% of the NAND gate outputs are erroneous?"
- $P_{\geq 0.8}$ [ $F^{\leq k}$ reply_count=n ]
  - "the probability that the sender has received n acknowledgements within k clock-ticks is at least 0.8"

# PCTL and measurability

- Recall: probability space (Path(s), $\Sigma_{Path(s)}$, $Pr_s$)
  - $\Sigma_{Path(s)}$ contains cylinder sets $C(\omega)$ for all finite paths $\omega$ starting in s and is closed under complementation & countable union
- All the sets of paths expressed by PCTL are measurable
  - i.e. are elements of the $\sigma$-algebra $\Sigma_{Path(s)}$
  - see [Var85] (which has a stronger result)
- Next ($X\ \phi$)
  - cylinder sets constructed from paths of length one
- Bounded until ($\phi_1\ U^{\leq k}\ \phi_2$)
  - (finite number of) cylinder sets from paths of length at most k
- Until ($\phi_1\ U\ \phi_2$)
  - countable union of finite paths satisfying $\phi_1\ U^{\leq k}\ \phi_2$ for all $k \geq 0$

# Qualitative vs. quantitative properties

- P operator of PCTL can be seen as a quantitative analogue of the CTL operators A (for all) and E (there exists)

- Qualitative PCTL properties
  - $P_{\sim p}$ [ $\psi$ ] where p is either 0 or 1
- Quantitative PCTL properties
  - $P_{\sim p}$ [ $\psi$ ] where p is in the range (0,1)

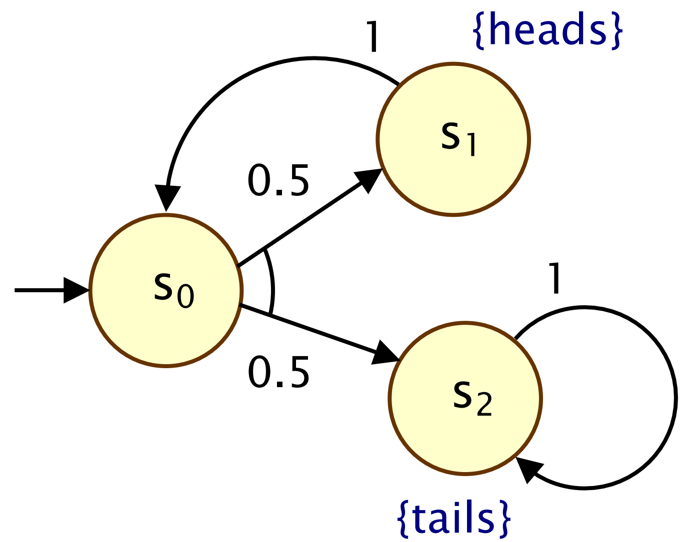
- $P_{>0}$ [ F $\phi$ ] is identical to EF $\phi$
  - there exists a *finite* path to a $\phi$-state
- $P_{\geq 1}$ [ F $\phi$ ] is (similar to but) weaker than AF $\phi$
  - a $\phi$-state is reached "almost surely"
  - see next slide…

# Example: Qualitative/quantitative

- Toss a coin repeatedly until "tails" is thrown

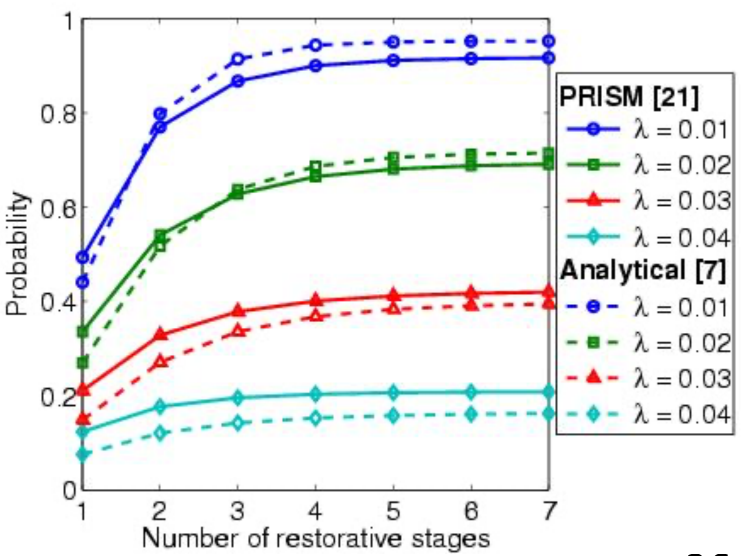
- Is "tails" always eventually thrown?
  - CTL: AF "tails"
  - Result: false
  - Counterexample: $s_0 s_1 s_0 s_1 s_0 s_1 \ldots$

- Does the probability of eventually throwing "tails" equal one?
  - PCTL: $P_{\geq 1}$ [ F "tails" ]
  - Result: true
  - Infinite path $s_0 s_1 s_0 s_1 s_0 s_1 \ldots$ has zero probability



# Quantitative properties

- Consider a PCTL formula $P_{\sim p}$ [ ψ ]
  - if the probability is unknown, how to choose the bound p?
- When the outermost operator of a PTCL formula is P
  - PRISM allows formulae of the form $P_{=?}$ [ ψ ]
  - "what is the probability that path formula ψ is true?"
- Model checking is no harder, it computes the values anyway
- Useful to spot patterns, trends
- Example
  - $P_{=?}$ [ F err/total>0.1 ]
  - "what is the probability that 10% of the NAND gate outputs are erroneous?"



# Limitations of PCTL

- PCTL, although useful in practice, has limited expressivity
  - essentially: probability of reaching states in X, passing only through states in Y (possibly, within k time steps)

- Alternative logics can be used, for example:
  - LTL [Pnu77], the non-probabilistic linear-time temporal logic
  - PCTL* [ASB+95,BdA95] which subsumes both PCTL and LTL

- To introduce these logics, we return briefly again to non-probabilistic logics and models…

# Branching vs. Linear time

- In CTL, temporal operators (on paths) always appear inside A or E
  - in LTL, temporal operators can be combined
- LTL but not CTL:
  - $F\ [\ req \wedge X\ ack\ ]$
  - "eventually a request occurs, followed immediately by an acknowledgement"
- CTL but not LTL:
  - AG EF initial
  - "for every computation, it is always possible to return to the initial state"

# LTL

- LTL syntax
  - path formulae only
  - $\psi ::= \text{true} \mid a \mid \psi \wedge \psi \mid \neg\psi \mid X\,\psi \mid \psi\,U\,\psi$
  - where $a \in AP$ is an atomic proposition

- LTL semantics (for a path $\omega$)
  - $\omega \vDash \text{true}$ always
  - $\omega \vDash a \quad \Leftrightarrow \quad a \in L(\omega(0))$
  - $\omega \vDash \psi_1 \wedge \psi_2 \quad \Leftrightarrow \quad \omega \vDash \psi_1$ and $\omega \vDash \psi_2$
  - $\omega \vDash \neg\psi \quad \Leftrightarrow \quad \omega \nvDash \psi$
  - $\omega \vDash X\,\psi \quad \Leftrightarrow \quad \omega[1\ldots] \vDash \psi$
  - $\omega \vDash \psi_1\,U\,\psi_2 \quad \Leftrightarrow \quad \exists k \geq 0$ s.t. $\omega[k\ldots] \vDash \psi_2$ and $\forall i<k\ \omega[i\ldots] \vDash \psi_1$

# LTL

- LTL semantics
  - implicit universal quantification over paths
  - i.e. for an LTS $M = (S, s_{init}, \rightarrow, L)$ and LTL formula $\psi$
  - $s \vDash \psi$ iff $\omega \vDash \psi$ for all paths $\omega \in Path(s)$
  - $M \vDash \psi$ iff $s_{init} \vDash \psi$
- e.g:
  - A F [ req $\wedge$ X ack ]
  - "it is always the case that, eventually, a request occurs, followed immediately by an acknowledgement"
- Derived operators like CTL, for example:
  - $F\ \psi \equiv true\ U\ \psi$
  - $G\ \psi \equiv \neg F(\neg \psi)$

# LTL + probabilities

- Same idea as PCTL: probabilities of sets of path formulae
  - for a state s of a DTMC and an LTL formula ψ:
  - Prob(s, ψ) = $Pr_s$ { ω ∈ Path(s) | ω ⊨ ψ }
  - all such path sets are measurable
- Examples from DTMC lectures
- Repeated reachability: "always eventually…"
  - Prob(s, GF send)
  - e.g. "what is the probability that the protocol successfully sends a message infinitely often?"
- Persistence properties: "eventually forever…"
  - Prob(s, FG stable)
  - e.g. "what is the probability of the leader election algorithm reaching, and staying in, a stable state?"

# PCTL*

- PCTL* subsumes both (probabilistic) LTL and PCTL

- State formulae:
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p} [\ \psi\ ]$
  - where $a \in AP$ and $\psi$ is a path formula

- Path formulae:
  - $\psi ::= \phi \mid \psi \wedge \psi \mid \neg\psi \mid X\ \psi \mid \psi\ U\ \psi$
  - where $\phi$ is a state formula

- A PCTL* formula is a state formula $\phi$
  - e.g. $P_{>0.1}$ [ GF $crit_1$ ] $\wedge$ $P_{>0.1}$ [ GF $crit_2$ ]

# Summing up…

- Temporal logics:
  - formal languages for specifying and reasoning about the behaviour of a system evolving over time

| | | |
|---|---|---|
| CTL | $\Phi$ | non-probabilistic (e.g. LTSs) |
| LTL | $\psi$ | |
| PCTL | $\Phi$ | probabilistic (e.g. DTMCs) |
| LTL + prob. | Prob(s, $\psi$) | |
| PCTL* | $\Phi$ | |