Lecture 2
Discrete-time Markov Chains

Dr. Dave Parker

UNIVERSITY OF OXFORD

Department of Computer Science
University of Oxford

Probabilistic Model Checking

- Formal verification and analysis of systems that exhibit probabilistic behaviour
  - e.g. randomised algorithms/protocols
  - e.g. systems with failures/unreliability
- Based on the construction and analysis of precise mathematical models
- This lecture: discrete-time Markov chains

Overview

- Probability basics
- Discrete-time Markov chains (DTMCs)
  - definition, properties, examples
- Formalising path-based properties of DTMCs
  - probability space over infinite paths
- Probabilistic reachability
  - definition, computation
- Sources/further reading: Section 10.1 of [BK08]

Probability basics

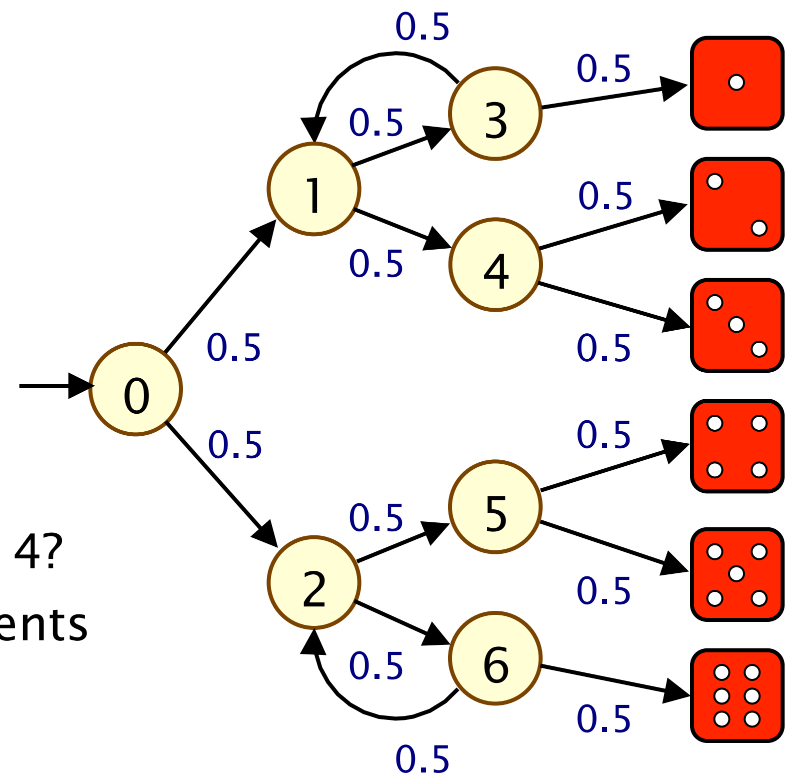
- First, need an experiment
  - The sample space Ω is the set of possible outcomes
  - An event is a subset of Ω, can form events $A \cap B$, $A \cup B$, $\Omega \setminus A$

- Examples:
  - toss a coin: $\Omega = \{H,T\}$, events: "H", "T"
  - toss two coins: $\Omega = \{(H,H),(H,T),(T,H),(T,T)\}$, event: "at least one H"
  - toss a coin ∞-often: Ω is set of infinite sequences of H/T event: "H in the first 3 throws"

- Probability is:
  - Pr("H") = Pr("T") = 1/2, Pr("at least one H") = 3/4
  - Pr("H in the first 3 throws") = 1/2 + 1/4 + 1/8 = 7/8

Probability example

- Modelling a 6-sided die using a fair coin
  - algorithm due to Knuth/Yao:
  - start at 0, toss a coin
  - upper branch when H
  - lower branch when T
  - repeat until value chosen

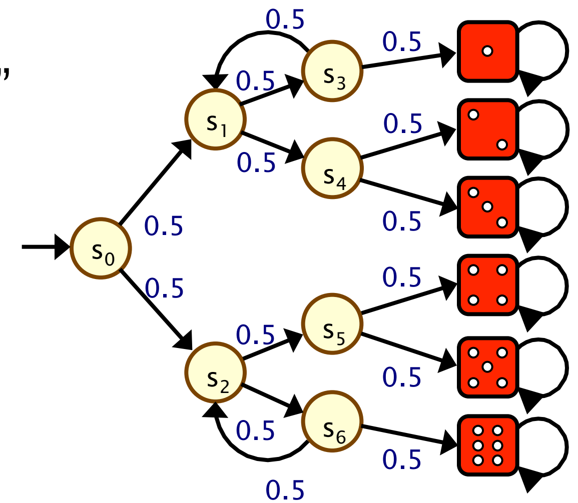


- Is this algorithm correct?
  - e.g. probability of obtaining a 4?
  - Obtain as disjoint union of events
  - THH, TTTHH, TTTTTHH, …
  - Pr("eventually 4")

    $= (1/2)^3 + (1/2)^5 + (1/2)^7 + \ldots = 1/6$

Example…

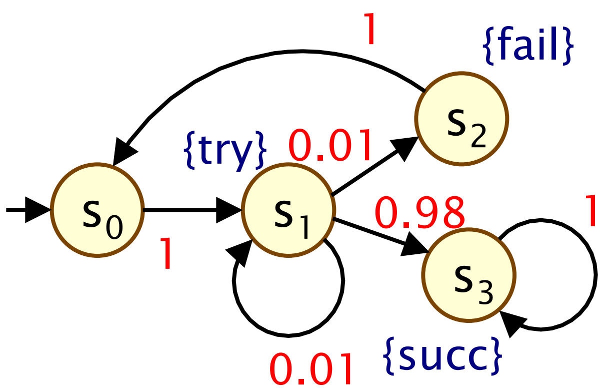
- Other properties?
  - "what is the probability of termination?"
- e.g. efficiency?
  - "what is the probability of needing more than 4 coin tosses?"
  - "on average, how many coin tosses are needed?"



- Probabilistic model checking provides a framework for these kinds of properties…
  - modelling languages
  - property specification languages
  - model checking algorithms, techniques and tools

Discrete-time Markov chains

- State-transition systems augmented with probabilities
- States
  - set of states representing possible configurations of the system being modelled
- Transitions
  - transitions between states model evolution of system's state; occur in discrete time-steps
- Probabilities
  - probabilities of making transitions between states are given by discrete probability distributions

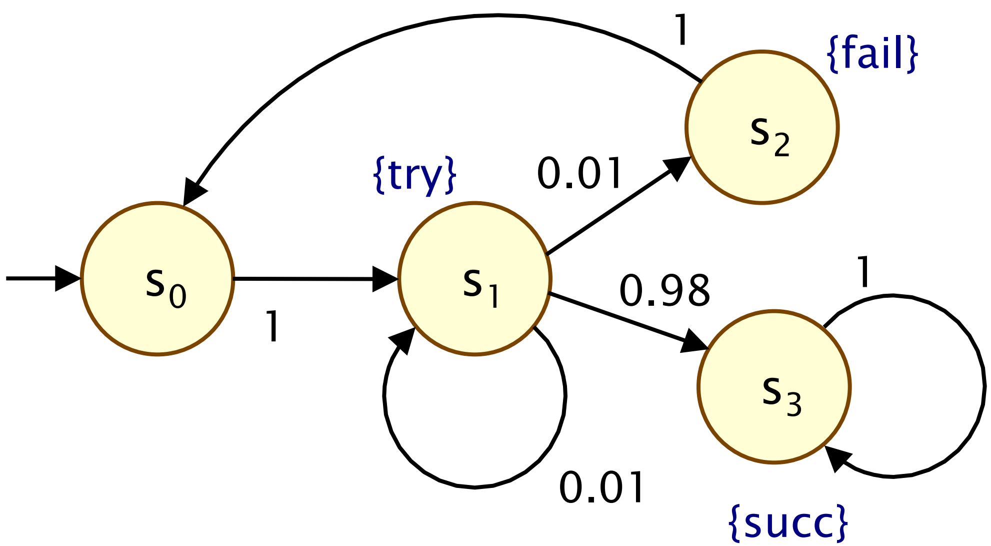


Markov property

- If the current state is known, then the future states of the system are independent of its past states
- i.e. the current state of the model contains all information that can influence the future evolution of the system
- also known as "memorylessness"

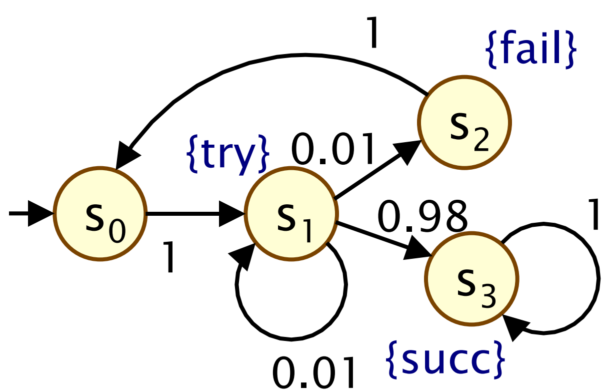
Simple DTMC example

- Modelling a very simple communication protocol
  - after one step, process starts trying to send a message
  - with probability 0.01, channel unready so wait a step
  - with probability 0.98, send message successfully and stop
  - with probability 0.01, message sending fails, restart



Discrete-time Markov chains

- Formally, a DTMC D is a tuple $(S,s_{init},\mathbf{P},L)$ where:
  - S is a set of states ("state space")
  - $s_{init} \in S$ is the initial state
  - $\mathbf{P} : S \times S \rightarrow [0,1]$ is the transition probability matrix where $\Sigma_{s' \in S} \mathbf{P}(s,s') = 1$ for all $s \in S$
  - $L : S \rightarrow 2^{AP}$ is function labelling states with atomic propositions (taken from a set AP)



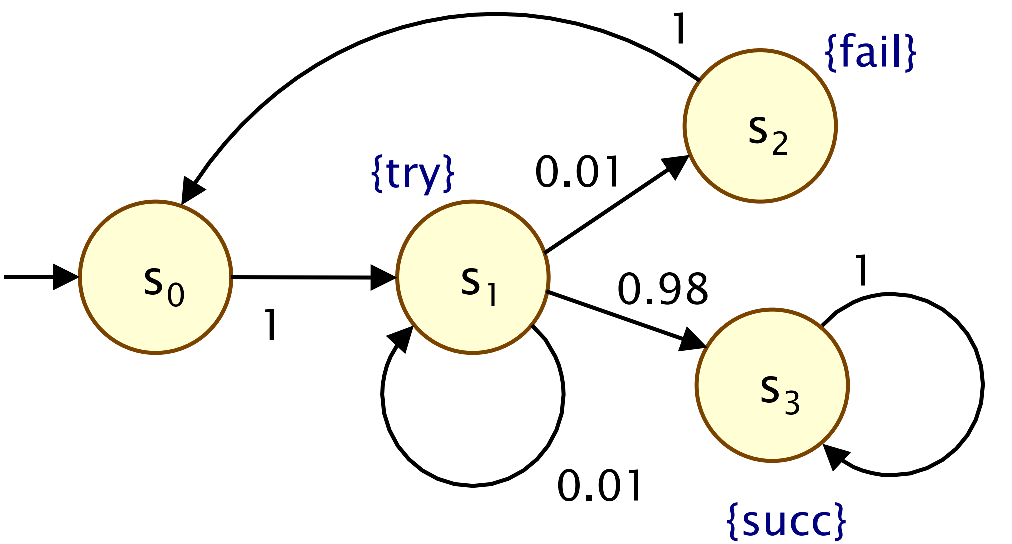
Simple DTMC example

$D = (S, s_{init}, P, L)$

$S = \{s_0, s_1, s_2, s_3\}$
$s_{init} = s_0$

$AP = \{try, fail, succ\}$
$L(s_0) = \emptyset,$
$L(s_1) = \{try\},$
$L(s_2) = \{fail\},$
$L(s_3) = \{succ\}$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Some more terminology

- P is a stochastic matrix, meaning it satisifes:
  - $P(s,s') \in [0,1]$ for all $s,s' \in S$ and $\Sigma_{s' \in S} P(s,s') = 1$ for all $s \in S$
- A sub-stochastic matrix satisfies:
  - $P(s,s') \in [0,1]$ for all $s,s' \in S$ and $\Sigma_{s' \in S} P(s,s') \leq 1$ for all $s \in S$
- An absorbing state is a state s for which:
  - $P(s,s) = 1$ and $P(s,s') = 0$ for all $s \neq s'$
  - the transition from s to itself is sometimes called a self-loop
- Note: Since we assume P is stochastic…
  - every state has at least one outgoing transition
  - i.e. no deadlocks (in model checking terminology)

DTMCs: An alternative definition

- Alternative definition… a DTMC is:
  - a family of random variables { X(k) | k=0,1,2,… }
  - where X(k) are observations at discrete time-steps
  - i.e. X(k) is the state of the system at time-step k
  - which satisfies…
- The Markov property ("memorylessness")
  - $Pr(X(k)=s_k \mid X(k-1)=s_{k-1}, \ldots , X(0)=s_0)$

    $= Pr(X(k)=s_k \mid X(k-1)=s_{k-1})$
  - for a given current state, future states are independent of past
- This allows us to adopt the "state-based" view presented so far (which is better suited to this context)

Other assumptions made here

- We consider time-homogenous DTMCs
  - transition probabilities are independent of time
  - $\mathbf{P}(s_{k-1},s_k) = Pr(X(k)=s_k \mid X(k-1)=s_{k-1})$
  - otherwise: time-inhomogenous

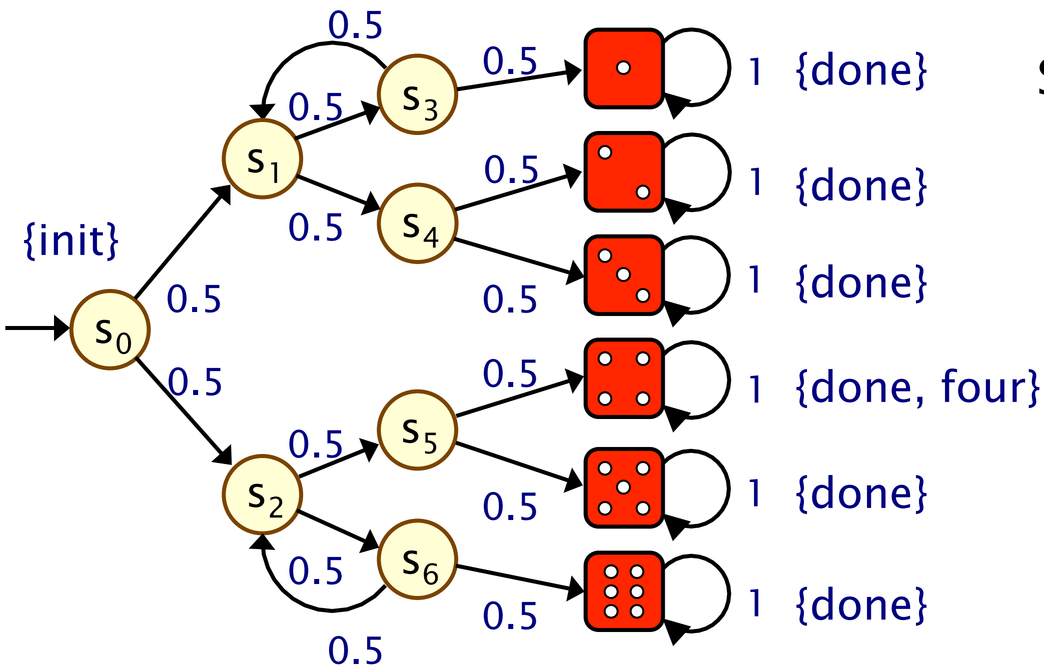
- We will (mostly) assume that the state space S is finite
  - in general, S can be any countable set

- Initial state $s_{init} \in S$ can be generalised…
  - to an initial probability distribution $s_{init} : S \rightarrow [0,1]$

- Transition probabilities are reals: $\mathbf{P}(s,s') \in [0,1]$
  - but for algorithmic purposes, are assumed to be rationals

DTMC example 2 – Coins and dice

- Recall Knuth/Yao's die algorithm from earlier:



$S = \{ s_0, s_1, \ldots, s_6, 1, 2, \ldots, 6 \}$

$s_{init} = s_0$

$P(s_0,s_1)=0.5$
$P(s_0,s_2)=0.5$
etc.

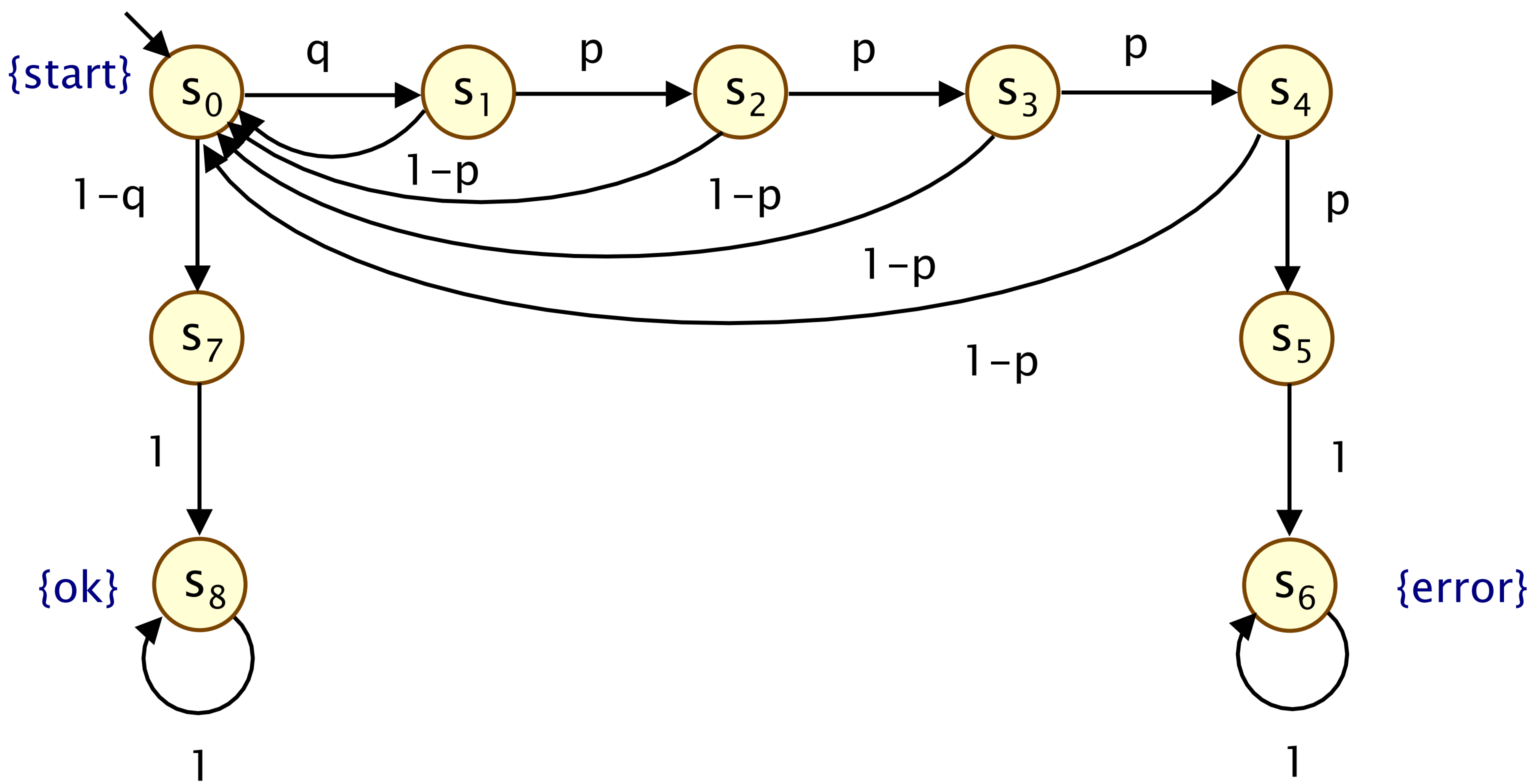
$L(s_0) = \{init\}$
etc.

DTMC example 3 – Zeroconf

- Zeroconf = "Zero configuration networking"
  - self-configuration for local, ad-hoc networks
  - automatic configuration of unique IP for new devices
  - simple; no DHCP, DNS, …
- Basic idea:
  - 65,024 available IP addresses (IANA-specified range)
  - new node picks address U at random
  - broadcasts "probe" messages: "Who is using U?"
  - a node already using U replies to the probe
  - in this case, protocol is restarted
  - messages may not get sent (transmission fails, host busy, …)
  - so: nodes send multiple (n) probes, waiting after each one

DTMC for Zeroconf

- $n=4$ probes, m existing nodes in network
- probability of message loss: p
- probability that new address is in use: $q = m/65024$

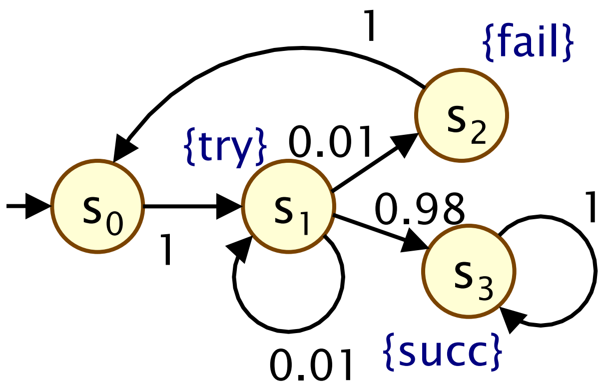


Properties of DTMCs

- Path-based properties
  - what is the probability of observing a particular behaviour (or class of behaviours)?
  - e.g. "what is the probability of throwing a 4?"
- Transient properties
  - probability of being in state s after t steps?
- Steady-state
  - long-run probability of being in each state
- Expectations
  - e.g. "what is the average number of coin tosses required?"

DTMCs and paths

- A path in a DTMC represents an execution (i.e. one possible behaviour) of the system being modelled
- Formally:
  - infinite sequence of states $s_0 s_1 s_2 s_3 \ldots$ such that $P(s_i, s_{i+1}) > 0 \;\; \forall i \geq 0$
  - infinite unfolding of DTMC

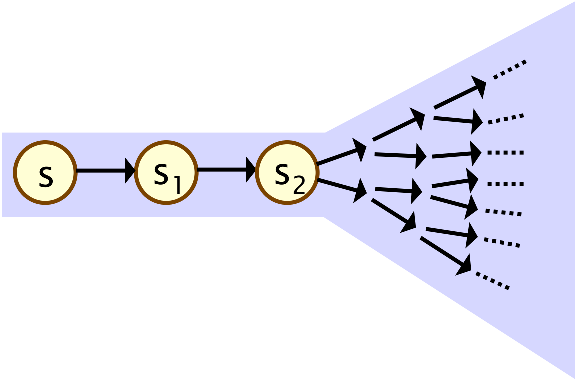


- Examples:
  - never succeeds: $(s_0 s_1 s_2)^\omega$
  - tries, waits, fails, retries, succeeds: $s_0 s_1 s_1 s_2 s_0 s_1 (s_3)^\omega$
- Notation:
  - Path(s) = set of all infinite paths starting in state s
  - also sometimes use finite (length) paths
  - $Path_{fin}(s)$ = set of all finite paths starting in state s

Paths and probabilities

- To reason (quantitatively) about this system
  - need to define a probability space over paths

- Intuitively:
  - sample space: Path(s) = set of all infinite paths from a state s
  - events: sets of infinite paths from s
  - basic events: cylinder sets (or "cones")
  - cylinder set Cyl(ω), for a finite path ω = set of infinite paths with the common finite prefix ω
  - for example: Cyl(ss_1s_2)



Probability spaces

- Let Ω be an arbitrary non-empty set

- A σ-algebra (or σ-field) on Ω is a family Σ of subsets of Ω closed under complementation and countable union, i.e.:
  - if $A \in \Sigma$, the complement $\Omega \setminus A$ is in Σ
  - if $A_i \in \Sigma$ for $i \in \mathbb{N}$, the union $\cup_i A_i$ is in Σ
  - the empty set $\varnothing$ is in Σ

- Elements of Σ are called measurable sets or events

- Theorem: For any family F of subsets of Ω, there exists a unique smallest σ-algebra on Ω containing F

Probability spaces

- Probability space (Ω, Σ, Pr)
  - Ω is the sample space
  - Σ is the set of events: σ-algebra on Ω
  - $Pr : \Sigma \rightarrow [0,1]$ is the probability measure:
    $Pr(\Omega) = 1$ and $Pr(\cup_i A_i) = \Sigma_i Pr(A_i)$ for countable disjoint A_i

Probability space - Simple example

- Sample space Ω
  - $\Omega = \{1,2,3\}$

- Event set Σ
  - e.g. powerset of Ω
  - $\Sigma = \{ \emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\} \}$
  - (closed under complement/countable union, contains ∅)

- Probability measure Pr
  - e.g. $Pr(1) = Pr(2) = Pr(3) = 1/3$
  - $Pr(\{1,2\}) = 1/3+1/3 = 2/3$, etc.

Probability space – Simple example 2

- Sample space Ω
  - $\Omega = \mathbb{N} = \{ 0,1,2,3,4,\ldots \}$
- Event set Σ
  - e.g. $\Sigma = \{ \emptyset$, "odd", "even", $\mathbb{N} \}$
  - (closed under complement/countable union, contains $\emptyset$)
- Probability measure Pr
  - e.g. Pr("odd") = 0.5, Pr("even") = 0.5

Probability space over paths

- Sample space Ω = Path(s)

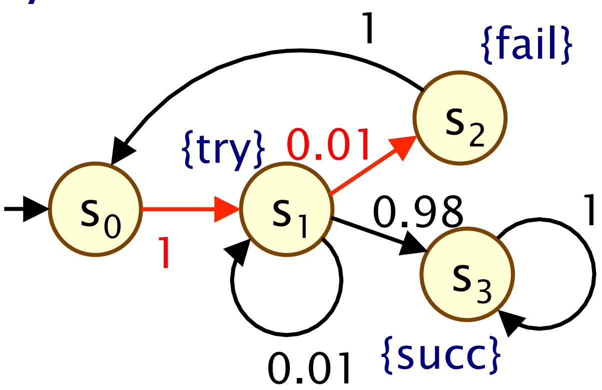
  set of infinite paths with initial state s
- Event set $\Sigma_{Path(s)}$
  - the cylinder set Cyl(ω) = { ω' ∈ Path(s) | ω is prefix of ω' }
  - $\Sigma_{Path(s)}$ is the least σ-algebra on Path(s) containing Cyl(ω) for all finite paths ω starting in s
- Probability measure Pr_s
  - define probability $\mathbf{P}_s(\omega)$ for finite path $\omega = ss_1 \ldots s_n$ as:
    - $\mathbf{P}_s(\omega) = 1$ if ω has length one (i.e. ω = s)
    - $\mathbf{P}_s(\omega) = \mathbf{P}(s,s_1) \cdot \ldots \cdot \mathbf{P}(s_{n-1},s_n)$ otherwise
    - define $Pr_s(Cyl(\omega)) = \mathbf{P}_s(\omega)$ for all finite paths ω
  - Pr_s extends uniquely to a probability measure $Pr_s:\Sigma_{Path(s)} \rightarrow [0,1]$

- See [KSK76] for further details

Paths and probabilities – Example

- Paths where sending fails immediately
  - $\omega = s_0 s_1 s_2$
  - $Cyl(\omega)$ = all paths starting $s_0 s_1 s_2 \ldots$
  - $\mathbf{P}_{s0}(\omega) = \mathbf{P}(s_0,s_1) \cdot \mathbf{P}(s_1,s_2)$

    $= 1 \cdot 0.01 = 0.01$
  - $Pr_{s0}(Cyl(\omega)) = \mathbf{P}_{s0}(\omega) = 0.01$



- Paths which are eventually successful and with no failures
  - $Cyl(s_0 s_1 s_3) \cup Cyl(s_0 s_1 s_1 s_3) \cup Cyl(s_0 s_1 s_1 s_1 s_3) \cup \ldots$
  - $Pr_{s0}(Cyl(s_0 s_1 s_3) \cup Cyl(s_0 s_1 s_1 s_3) \cup Cyl(s_0 s_1 s_1 s_1 s_3) \cup \ldots)$

    $= \mathbf{P}_{s0}(s_0 s_1 s_3) + \mathbf{P}_{s0}(s_0 s_1 s_1 s_3) + \mathbf{P}_{s0}(s_0 s_1 s_1 s_1 s_3) + \ldots$

    $= 1 \cdot 0.98 + 1 \cdot 0.01 \cdot 0.98 + 1 \cdot 0.01 \cdot 0.01 \cdot 0.98 + \ldots$

    $= 0.9898989898\ldots$

    $= 98/99$

Reachability

- Key property: probabilistic reachability
  - probability of a path reaching a state in some target set $T \subseteq S$
  - e.g. "probability of the algorithm terminating successfully?"
  - e.g. "probability that an error occurs during execution?"

- Dual of reachability: invariance
  - probability of remaining within some class of states
  - Pr("remain in set of states T") = 1 - Pr("reach set $S \backslash T$")
  - e.g. "probability that an error never occurs"

- We will also consider other variants of reachability
  - time-bounded, constrained ("until"), …

Reachability probabilities

- Formally: ProbReach(s, T) = Pr_s(Reach(s, T))
  - where Reach(s, T) = { $s_0 s_1 s_2 \ldots \in$ Path(s) | s_i in T for some i }

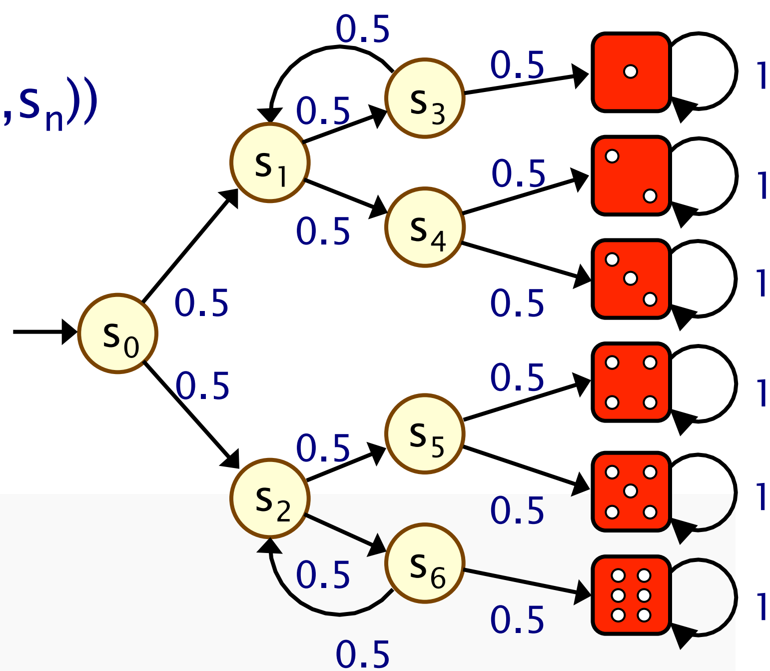
- Is Reach(s, T) measurable for any $T \subseteq S$? Yes…
  - Reach(s, T) is the union of all basic cylinders Cyl($s_0 s_1 \ldots s_n$) where $s_0 s_1 \ldots s_n$ in Reach_{fin}(s, T)
  - Reach_{fin}(s, T) contains all finite paths $s_0 s_1 \ldots s_n$ such that: $s_0 = s$, $s_0, \ldots, s_{n-1} \notin T$, $s_n \in T$
  - set of such finite paths $s_0 s_1 \ldots s_n$ is countable

- Probability
  - in fact, the above is a disjoint union
  - so probability obtained by simply summing…

Computing reachability probabilities

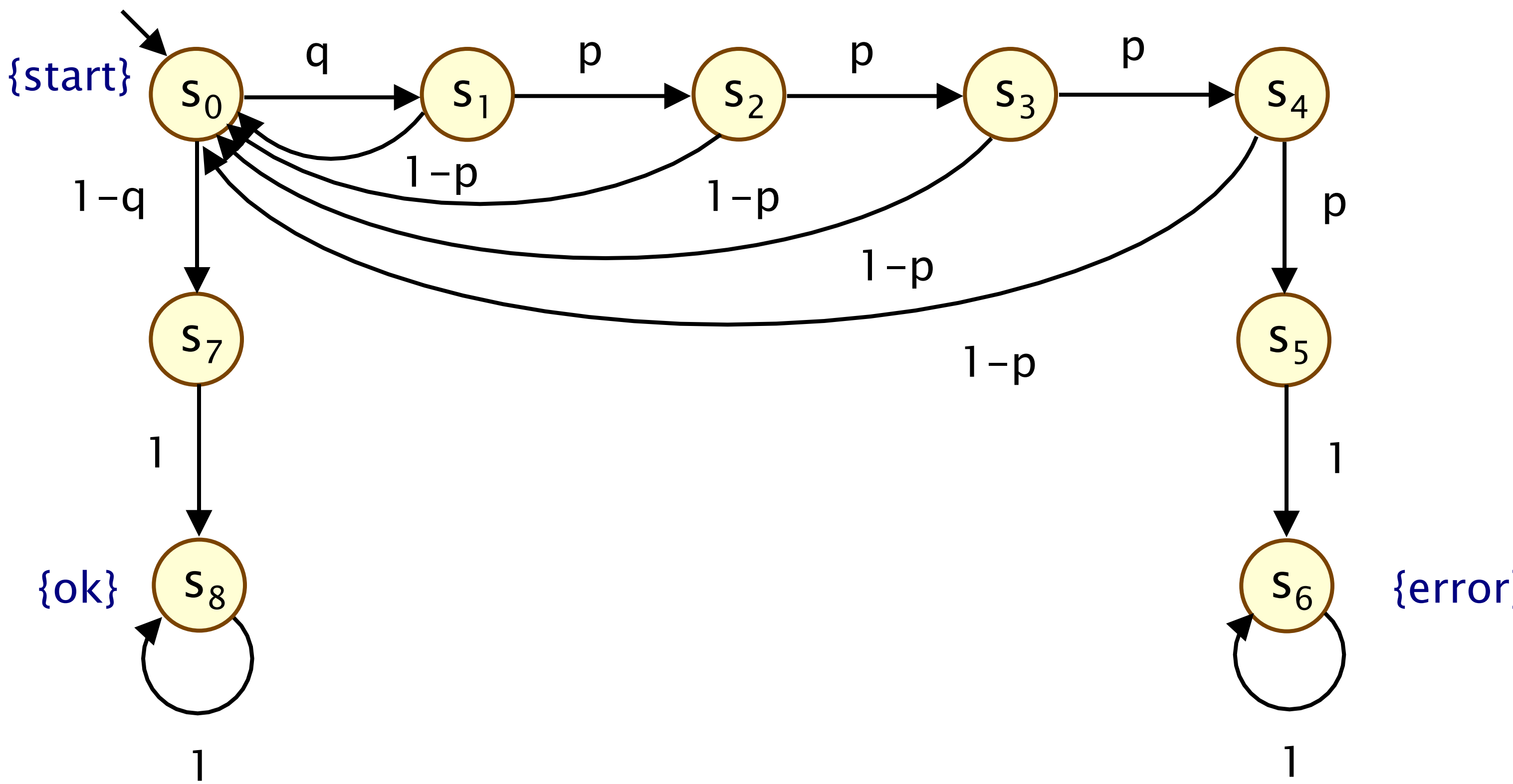
- Compute as (infinite) sum…
- $\Sigma_{s_0,\ldots,s_n \in \text{Reachfin}(s,\, T)}\ \Pr_{s0}(\text{Cyl}(s_0,\ldots,s_n))$

  $= \Sigma_{s_0,\ldots,s_n \in \text{Reachfin}(s,\, T)}\ \mathbf{P}(s_0,\ldots,s_n)$
- Example:
  – ProbReach(s_0, {4})



Computing reachability probabilities

- ProbReach(s_0, {s_6}) : compute as infinite sum?
  - doesn't scale...



Computing reachability probabilities

- Alternative: derive a linear equation system
  - solve for all states simultaneously
  - i.e. compute vector <u>ProbReach</u>(T)

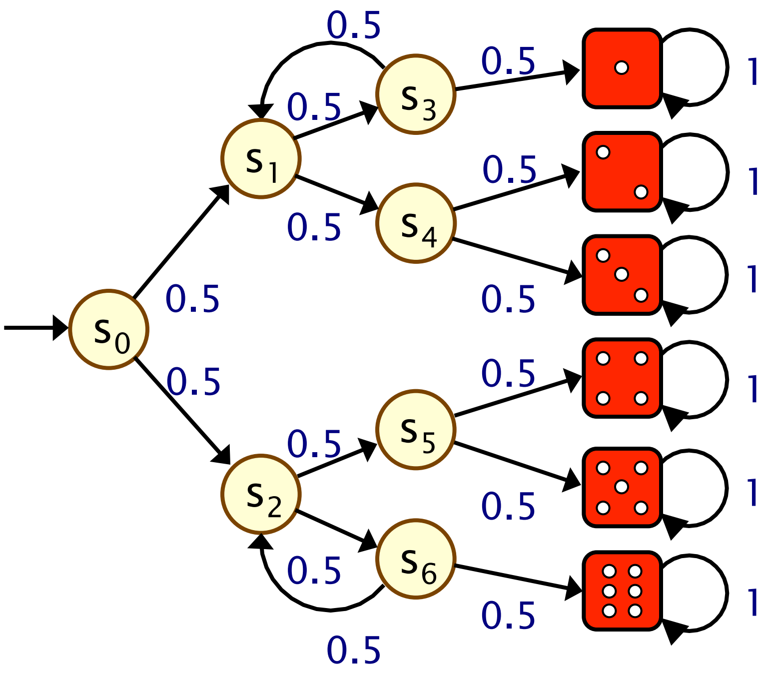
- Let x_s denote ProbReach(s, T)

- Solve:

$$x_s = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if T is not reachable from s} \\ \sum_{s' \in S} P(s,s') \cdot x_{s'} & \text{otherwise} \end{cases}$$

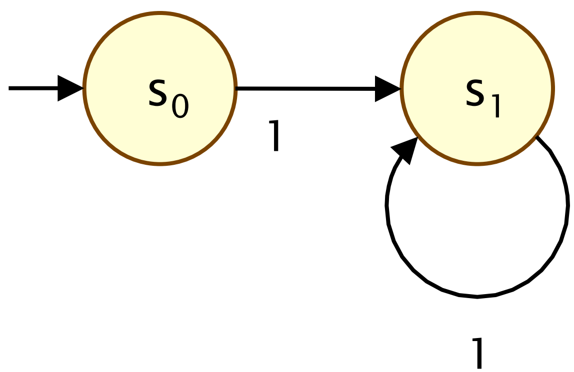
Example

- Compute ProbReach(s_0, {4})



Unique solutions

- Why the need to identify states that cannot reach T?
- Consider this simple DTMC:
  - compute probability of reaching $\{s_0\}$ from s_1



  - linear equation system: $x_{s_0} = 1,\ x_{s_1} = x_{s_1}$
  - multiple solutions: $(x_{s_0}, x_{s_1}) = (1,p)$ for any $p \in [0,1]$

Computing reachability probabilities

- Another alternative: least fixed point characterisation
- Consider functions of the form:
  - $F : [0,1]^S \rightarrow [0,1]^S$

vectors of probabilities for each state

- And define:
  - $\underline{y} \leq \underline{y}'$ iff $\underline{y}(s) \leq \underline{y}'(s)$ for all s
- $\underline{y}$ is a fixed point of F if $F(\underline{y}) = \underline{y}$
- A fixed point $\underline{x}$ of F is the least fixed point of F if $\underline{x} \leq \underline{y}$ for any other fixed point $\underline{y}$

Least fixed point

- ProbReach(T) is the least fixed point of the function F:

$$F(\underline{y})(s) = \begin{cases} 1 & \text{if } s \in T \\ \sum_{s' \in S} P(s,s') \cdot \underline{y}(s') & \text{otherwise.} \end{cases}$$

- This yields a simple iterative algorithm to approximate ProbReach(T):
  - $\underline{x}^{(0)} = \underline{0}$ (i.e. $\underline{x}^{(0)}(s) = 0$ for all s)
  - $\underline{x}^{(n+1)} = F(\underline{x}^{(n)})$
  - $\underline{x}^{(0)} \leq \underline{x}^{(1)} \leq \underline{x}^{(2)} \leq \underline{x}^{(3)} \leq \ldots$
  - ProbReach(T) $= \lim_{n\to\infty} \underline{x}^{(n)}$

in practice, terminate when for example:

$$\max_s \mid \underline{x}^{(n+1)}(s) - \underline{x}^{(n)}(s)) \mid < \varepsilon$$

for some user-defined tolerance value ε

Least fixed point

- Expressing ProbReach as a least fixed point…
  - corresponds to solving the linear equation system using the power method
    - other iterative methods exist (see later)
    - power method is guaranteed to converge
  - generalises non-probabilistic reachability
  - can be generalised to:
    - constrained reachability (see PCTL "until")
    - reachability for Markov decision processes
  - also yields bounded reachability probabilities…

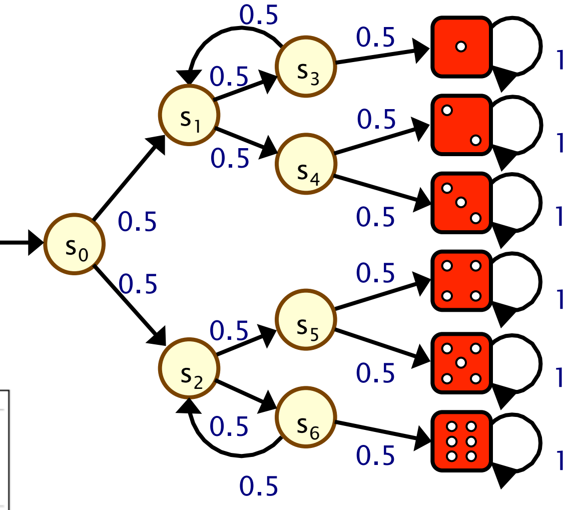
Bounded reachability probabilities

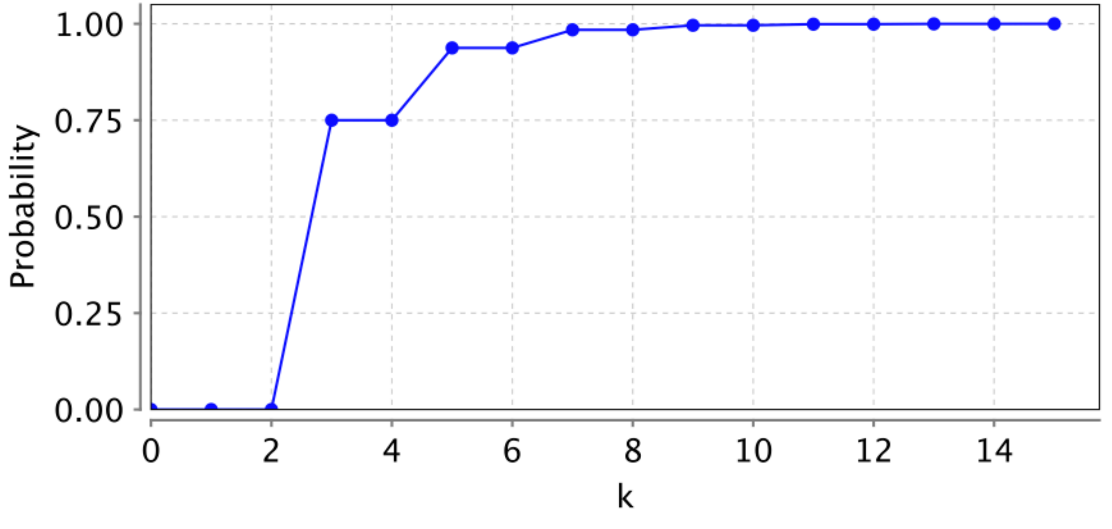
- Probability of reaching T from s within k steps
- Formally: $\text{ProbReach}^{\leq k}(s, T) = \text{Pr}_s(\text{Reach}^{\leq k}(s, T))$ where:
  - $\text{Reach}^{\leq k}(s, T) = \{ s_0 s_1 s_2 \ldots \in \text{Path}(s) \mid s_i \text{ in } T \text{ for some } i \leq k \}$
- $\underline{\text{ProbReach}}^{\leq k}(T) = \underline{x}^{(k+1)}$ from the previous fixed point
  - which gives us…

$$\text{ProbReach}^{\leq k}(s, T) = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } k = 0 \,\&\, s \notin T \\ \sum_{s' \in S} P(s,s') \cdot \text{ProbReach}^{\leq k-1}(s', T) & \text{if } k > 0 \,\&\, s \notin T \end{cases}$$

(Bounded) reachability

- $\text{ProbReach}(s_0, \{1,2,3,4,5,6\}) = 1$
- $\text{ProbReach}^{\leq k}(s_0, \{1,2,3,4,5,6\}) = \ldots$





Summing up…

- Discrete-time Markov chains (DTMCs)
  - state-transition systems augmented with probabilities
- Formalising path-based properties of DTMCs
  - probability space over infinite paths
- Probabilistic reachability
  - infinite sum
  - linear equation system
  - least fixed point characterisation
  - bounded reachability

Next lecture

- Thur 12pm

- Discrete-time Markov chains…
  - transient
  - steady-state
  - long-run behaviour