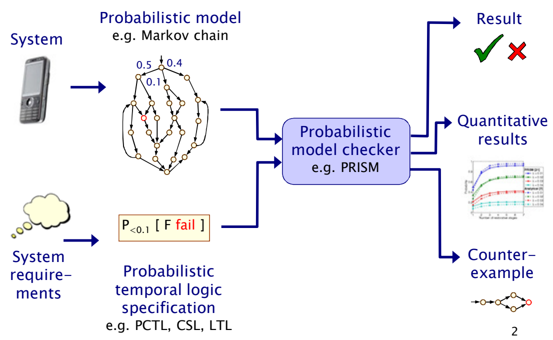
Probabilistic Model Checking (CTMCs)

Marta Kwiatkowska

Department of Computer Science

ROCKS Autumn School, October 2012

Probabilistic model checking



Overview

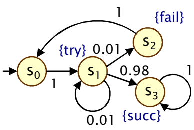
- Recap: Discrete-time Markov chains (DTMCs)
  - PCTL model checking
  - costs and rewards
- Continuous-time Markov chains (CTMCs)
  - exponential distribution and its properties
  - CTMCs: definition, examples, paths & probability spaces
  - CSL: A temporal logic for CTMCs
  - CSL model checking (uniformisation, steady-state)
  - costs & rewards

Recap

Discrete-time Markov chains

Discrete-time Markov chains

- Formally, a DTMC D is a tuple $(S, s_{init}, \mathbf{P}, L)$ where:
  - S is a finite 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
- Note: no deadlock states
  - i.e. every state has at least one outgoing transition
  - can add self loops to represent final/terminating states

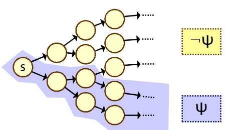


PCTL syntax

- PCTL syntax:
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p} [\, \psi \,]$ (state formulas)
    - ψ is true with probability ~p
  - $\psi ::= X\, \phi \quad \mid \quad \phi\, U^{\leq k}\, \phi \quad \mid \quad \phi\, U\, \phi$ (path formulas)
    - "next" "bounded until" "until"
  - where a is an atomic proposition, used to identify states of interest, $p \in [0,1]$ is a probability, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$
- A PCTL formula is always a state formula
  - path formulas only occur inside the P operator

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 ψ is true for an outgoing path satisfies ~p"
  - example: $s \vDash P_{<0.25} [\, X \text{ fail} \,] \Leftrightarrow$ "the probability of atomic proposition fail being true in the next state of outgoing paths from s is less than 0.25"
  - formally: $s \vDash P_{\sim p} [\psi] \Leftrightarrow \text{Prob}(s, \psi) \sim p$
  - where: $\text{Prob}(s, \psi) = \Pr_s \{\, \omega \in \text{Path}(s) \mid \omega \vDash \psi \,\}$
  - (sets of paths satisfying ψ are always measurable [Var85])



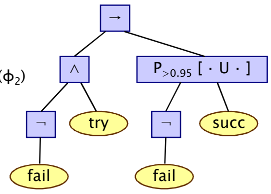
Prob(s, ψ) ~ p ?

PCTL model checking for DTMCs

- Algorithm for PCTL model checking [CY88,HJ94,CY95]
  - inputs: DTMC $D=(S, s_{init}, \mathbf{P}, L)$, PCTL formula ϕ
  - output: $\text{Sat}(\phi) = \{\, s \in S \mid s \vDash \phi \,\}$ = set of states satisfying ϕ
- What does it mean for a DTMC D to satisfy a formula ϕ?
  - sometimes, want to check that $s \vDash \phi\ \forall\ s \in S$, i.e. $\text{Sat}(\phi) = S$
  - sometimes, just want to know if $s_{init} \vDash \phi$, i.e. if $s_{init} \in \text{Sat}(\phi)$
- Sometimes, focus on quantitative results
  - e.g. compute result of P=? [F error]
  - e.g. compute result of P=? [$F^{\leq k}$ error] for $0 \leq k \leq 100$

PCTL model checking for DTMCs

- Basic algorithm proceeds by induction on parse tree of ϕ
  - example: $\phi = (\neg fail \wedge try) \rightarrow P_{>0.95}$ [$\neg$fail U succ]
- For the non-probabilistic operators:
  - Sat(true) = S
  - Sat(a) = { s ∈ S | a ∈ L(s) }
  - Sat($\neg\phi$) = S \ Sat(ϕ)
  - Sat($\phi_1 \wedge \phi_2$) = Sat(ϕ_1) ∩ Sat(ϕ_2)
- For the $P_{\sim p}$ [ψ] operator
  - need to compute the probabilities Prob(s, ψ) for all states s ∈ S
  - focus here on "until" case: $\psi = \phi_1$ U ϕ_2



9

PCTL until for DTMCs

- Computation of probabilities Prob(s, ϕ_1 U ϕ_2) for all s ∈ S
- First, identify all states where the probability is 1 or 0
  - S^{yes} = Sat($P_{\geq 1}$ [ϕ_1 U ϕ_2])
  - S^{no} = Sat($P_{\leq 0}$ [ϕ_1 U ϕ_2])
- Then solve linear equation system for remaining states
- We refer to the first phase as "precomputation"
  - two algorithms: Prob0 (for S^{no}) and Prob1 (for S^{yes})
  - algorithms work on underlying graph (probabilities irrelevant)
- Important for several reasons
  - reduces the set of states for which probabilities must be computed numerically (which is more expensive)
  - guarantees uniqueness of solutions
  - gives exact results for the states in S^{yes} and S^{no} (no round-off)
  - for $P_{\sim p}$[·] where p is 0 or 1, no further computation required

10

PCTL until – Linear equations

- Probabilities Prob(s, ϕ_1 U ϕ_2) can now be obtained as the unique solution of the following set of linear equations:

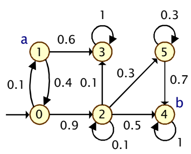
$$\text{Prob}(s, \phi_1 \text{ U } \phi_2) = \begin{cases} 1 & \text{if } s \in S^{yes} \\ 0 & \text{if } s \in S^{no} \\ \sum_{s' \in S} P(s,s') \cdot \text{Prob}(s', \phi_1 \text{ U } \phi_2) & \text{otherwise} \end{cases}$$

  - can be reduced to a system in $|S^?|$ unknowns instead of |S| where $S^? = S \setminus (S^{yes} \cup S^{no})$
- This can be solved with (a variety of) standard techniques
  - direct methods, e.g. Gaussian elimination
  - iterative methods, e.g. Jacobi, Gauss-Seidel, … (preferred in practice due to scalability)

11

PCTL until – Example

- Example: $P_{>0.8}$ [$\neg$a U b]

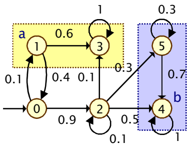


12

PCTL until – Example

- Example: $P_{>0.8}$ [¬a U b]

S^{no} =
Sat($P_{\leq 0}$ [¬a U b])

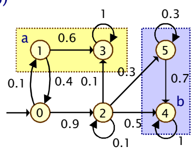


S^{yes} =
Sat($P_{\geq 1}$ [¬a U b])

PCTL until – Example

- Example: $P_{>0.8}$ [¬a U b]
- Let x_s = Prob(s, ¬a U b)
- Solve:

S^{no} =
Sat($P_{\leq 0}$ [¬a U b])



S^{yes} =
Sat($P_{\geq 1}$ [¬a U b])

$x_4 = x_5 = 1$
$x_1 = x_3 = 0$
$x_0 = 0.1x_1+0.9x_2 = 0.8$
$x_2 = 0.1x_2+0.1x_3+0.3x_5+0.5x_4 = 8/9$
Prob(¬a U b) = $\underline{x}$ = [0.8, 0, 8/9, 0, 1, 1]

Sat($P_{>0.8}$ [¬a U b]) = { s_2,s_4,s_5 }

Costs and rewards

- We augment DTMCs with rewards (or, conversely, costs)
  - real-valued quantities assigned to states and/or transitions
  - these can have a wide range of possible interpretations
- Some examples:
  - elapsed time, power consumption, size of message queue, number of messages successfully delivered, net profit, …
- Costs? or rewards?
  - mathematically, no distinction between rewards and costs
  - when interpreted, we assume that it is desirable to minimise costs and to maximise rewards
  - we will consistently use the terminology “rewards” regardless

Reward-based properties

- Properties of DTMCs augmented with rewards
  - allow a wide range of quantitative measures of the system
  - basic notion: expected value of rewards
  - formal property specifications will be in an extension of PCTL
- More precisely, we use two distinct classes of property…
- Instantaneous properties
  - the expected value of the reward at some time point
- Cumulative properties
  - the expected cumulated reward over some period

DTMC reward structures

- For a DTMC $(S, s_{init}, \mathbf{P}, L)$, a reward structure is a pair $(\underline{\rho}, \iota)$
  - $\underline{\rho} : S \to \mathbb{R}_{\geq 0}$ is the state reward function (vector)
  - $\iota : S \times S \to \mathbb{R}_{\geq 0}$ is the transition reward function (matrix)
- Example (for use with instantaneous properties)
  - "size of message queue": $\underline{\rho}$ maps each state to the number of jobs in the queue in that state, ι is not used
- Examples (for use with cumulative properties)
  - "time-steps": $\underline{\rho}$ returns 1 for all states and ι is zero (equivalently, $\underline{\rho}$ is zero and ι returns 1 for all transitions)
  - "number of messages lost": $\underline{\rho}$ is zero and ι maps transitions corresponding to a message loss to 1
  - "power consumption": $\underline{\rho}$ is defined as the per-time-step energy consumption in each state and ι as the energy cost of each transition

17

PCTL and rewards

- Extend PCTL to incorporate reward-based properties
  - add an R operator, which is similar to the existing P operator

expected reward is ~r

- ϕ ::= … | $P_{\sim p}$ [ψ] | $R_{\sim r}$ [$I^{=k}$] | $R_{\sim r}$ [$C^{\leq k}$] | $R_{\sim r}$ [F ϕ]

"instantaneous" "cumulative" "reachability"

- where $r \in \mathbb{R}_{\geq 0}$, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$
- $R_{\sim r}$ [·] means "the expected value of · satisfies ~r"

18

Types of reward formulas

- Instantaneous: $R_{\sim r}$ [$I^{=k}$]
  - "the expected value of the state reward at time-step k is ~r"
  - e.g. "the expected queue size after exactly 90 seconds"
- Cumulative: $R_{\sim r}$ [$C^{\leq k}$]
  - "the expected reward cumulated up to time-step k is ~r"
  - e.g. "the expected power consumption over one hour"
- Reachability: $R_{\sim r}$ [F ϕ]
  - "the expected reward cumulated before reaching a state satisfying ϕ is ~r"
  - e.g. "the expected time for the algorithm to terminate"

19

Reward formula semantics

- Formal semantics of the three reward operators
  - based on random variables over (infinite) paths
- Recall:
  - $s \models P_{\sim p}[\psi] \Leftrightarrow Pr_s\{\omega \in Path(s) \mid \omega \models \psi\} \sim p$
- For a state s in the DTMC:
  - $s \models R_{\sim r}[I^{=k}] \Leftrightarrow Exp(s, X_{I=k}) \sim r$
  - $s \models R_{\sim r}[C^{\leq k}] \Leftrightarrow Exp(s, X_{C\leq k}) \sim r$
  - $s \models R_{\sim r}[F\,\Phi] \Leftrightarrow Exp(s, X_{F\Phi}) \sim r$

where: Exp(s, X) denotes the expectation of the random variable $X : Path(s) \to \mathbb{R}_{\geq 0}$ with respect to the probability measure Pr_s

20

Reward formula semantics

- Definition of random variables:
  - for an infinite path $\omega = s_0 s_1 s_2 \ldots$

$$X_{I=k}(\omega) = \underline{\rho}(s_k)$$

$$X_{C\leq k}(\omega) = \begin{cases} 0 & \text{if } k = 0 \\ \sum_{i=0}^{k-1} \underline{\rho}(s_i) + \iota(s_i, s_{i+1}) & \text{otherwise} \end{cases}$$

$$X_{F\phi}(\omega) = \begin{cases} 0 & \text{if } s_0 \in Sat(\phi) \\ \infty & \text{if } s_i \notin Sat(\phi) \text{ for all } i \geq 0 \\ \sum_{i=0}^{k_\phi - 1} \underline{\rho}(s_i) + \iota(s_i, s_{i+1}) & \text{otherwise} \end{cases}$$

  - where $k_\phi = \min\{ j \mid s_j \models \phi \}$

Model checking reward properties

- Instantaneous: $R_{\sim r}$ [$I^{=k}$]
- Cumulative: $R_{\sim r}$ [$C^{\leq k}$]
  - variant of the method for computing bounded until probabilities
  - solution of recursive equations
- Reachability: $R_{\sim r}$ [F ϕ]
  - similar to computing until probabilities
  - precomputation phase (identify infinite reward states)
  - then reduces to solving a system of linear equation
- For more details, see e.g. [KNP07a]

Part 2

Continuous-time Markov chains

Time in DTMCs

- Time in a DTMC (or MDP) proceeds in discrete steps
- Two possible interpretations:
  - accurate model of (discrete) time units
    - e.g. clock ticks in model of an embedded device
  - time-abstract
    - no information assumed about the time transitions take
- Continuous-time Markov chains (CTMCs)
  - dense model of time
  - transitions can occur at any (real-valued) time instant
  - modelled using exponential distributions
  - suits modelling of: performance/reliability (e.g. of computer networks, manufacturing systems, queueing networks), biological pathways, chemical reactions, …

Overview

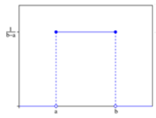
- Recap: Discrete-time Markov chains (DTMCs)
  - PCTL model checking
  - costs and rewards
- Part 2: Continuous-time Markov chains (CTMCs)
  - exponential distribution and its properties
  - CTMCs: definition, examples, paths & probability spaces
  - CSL: A temporal logic for CTMCs
  - CSL model checking (uniformisation, steady-state)
  - costs & rewards

Continuous probability distributions

- Defined by:
  - cumulative distribution function

    $F(t) = Pr(X \le t) = \int_{-\infty}^{t} f(x)\,dx$

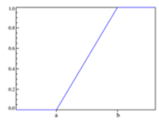
  - where f is the probability density function
  - Pr(X=t) = 0 for all t



- Example: uniform distribution: U(a,b)

$$f(t) = \begin{cases} \frac{1}{b-a} & \text{if } a \le t \le b \\ 0 & \text{otherwise} \end{cases}$$

$$F(t) = \begin{cases} 0 & \text{if } t < a \\ \frac{t-a}{b-a} & \text{if } a \le t < b \\ 1 & \text{if } t \ge b \end{cases}$$



Exponential distribution

- A continuous random variable X is exponential with parameter $\lambda > 0$ if the density function is given by:

$$f(t) = \begin{cases} \lambda \cdot e^{-\lambda t} & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases}$$

λ = "rate"

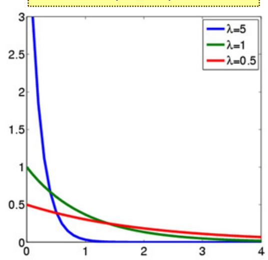
- Cumulative distribution function (for $t \ge 0$):

$$F(t) = Pr(X \le t) = \int_0^t \lambda \cdot e^{-\lambda x} dx = [-e^{-\lambda x}]_0^t = 1 - e^{-\lambda t}$$

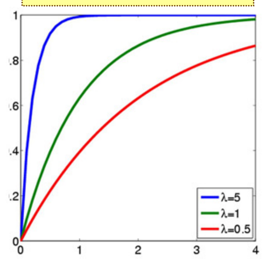
- Other properties:
  - negation: $Pr(X > t) = e^{-\lambda t}$
  - mean (expectation): $E[X] = \int_0^\infty x \cdot \lambda \cdot e^{-\lambda x} dx = \frac{1}{\lambda}$
  - variance: $Var(X) = 1/\lambda^2$

Exponential distribution – Examples

Probability density function



Cumulative distribution function



- The more λ increases, the faster the c.d.f. approaches 1

Exponential distribution

- Adequate for modelling many real-life phenomena
  - failures
    - e.g. time before machine component fails
  - inter-arrival times
    - e.g. time before next call arrives to a call centre
  - biological systems
    - e.g. times for reactions between proteins to occur
- Maximal entropy if just the mean is known
  - i.e. best approximation when only mean is known
- Can approximate general distributions arbitrarily closely
  - phase-type distributions

Exponential distribution – Properties

- Two useful properties of the exponential distribution:
- The exponential distribution is memoryless:
  - $Pr(X>t_1+t_2 \mid X>t_1) = Pr(X>t_2)$
  - it is the only memoryless continuous distribution
  - the discrete-time equivalent is the geometric distribution
- The minimum of two independent exponential distributions is an exponential distribution (parameter is sum)
  - $X_1 \sim Exponential(\lambda_1)$, $X_2 \sim Exponential(\lambda_2)$
  - $Y = min(X_1,X_2) \sim Exponential(\lambda_1+\lambda_2)$
  - generalises to minimum of n distributions

Overview

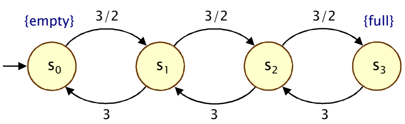
- Recap: Discrete-time Markov chains (DTMCs)
  - PCTL model checking
  - costs and rewards
- Part 2: Continuous-time Markov chains (CTMCs)
  - exponential distribution and its properties
  - CTMCs: definition, examples, paths & probability spaces
  - CSL: A temporal logic for CTMCs
  - CSL model checking (uniformisation, steady-state)
  - costs & rewards

Continuous-time Markov chains

- Continuous-time Markov chains (CTMCs)
  - labelled transition systems augmented with rates
  - continuous time delays, exponentially distributed
- Formally, a CTMC C is a tuple $(S,s_{init},\mathbf{R},L)$ where:
  - S is a finite set of states (“state space”)
  - $s_{init} \in S$ is the initial state
  - $\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ is the transition rate matrix
  - $L : S \to 2^{AP}$ is a labelling with atomic propositions
- Transition rate matrix assigns rates to each pair of states
  - used as a parameter to the exponential distribution
  - transition between s and s’ when $\mathbf{R}(s,s')>0$
  - probability triggered before t time units: $1 - e^{-\mathbf{R}(s,s')\cdot t}$

Simple CTMC example

- Modelling a queue of jobs
  - initially the queue is empty
  - jobs arrive with rate 3/2 (i.e. mean inter-arrival time is 2/3)
  - jobs are served with rate 3 (i.e. mean service time is 1/3)
  - maximum size of the queue is 3
  - state space: $S = \{s_i\}_{i=0..3}$ where s_i indicates i jobs in queue



33

Race conditions

- What happens when there exists multiple s' with **R**(s,s')>0?
  - race condition: first transition triggered determines next state
  - two questions:
  - 1. How long is spent in s before a transition occurs?
  - 2. Which transition is eventually taken?

- 1. Time spent in a state before a transition
  - minimum of exponential distributions
  - exponential with parameter given by summation:

$$E(s) = \sum_{s' \in S} \mathbf{R}(s,s')$$

  - probability of leaving a state s within [0,t] is $1-e^{-E(s)\cdot t}$
  - E(s) is the exit rate of state s
  - s is called absorbing if E(s)=0 (no outgoing transitions)

34

Race conditions…

- 2. Which transition is taken from state s?
  - the choice is independent of the time at which it occurs
  - e.g. if $X_1 \sim \text{Exponential}(\lambda_1)$, $X_2 \sim \text{Exponential}(\lambda_2)$
  - then the probability that $X_1<X_2$ is $\lambda_1/(\lambda_1+\lambda_2)$
  - more generally, the probability is given by…

- The embedded DTMC: $emb(C)=(S,s_{init},\mathbf{P}^{emb(C)},L)$
  - state space, initial state and labelling as the CTMC
  - for any $s,s' \in S$

$$\mathbf{P}^{emb(C)}(s,s') = \begin{cases} \mathbf{R}(s,s')/E(s) & \text{if } E(s) > 0 \\ 1 & \text{if } E(s) = 0 \text{ and } s = s' \\ 0 & \text{otherwise} \end{cases}$$

- Probability that next state from s is s' given by $\mathbf{P}^{emb(C)}(s,s')$

35

Two interpretations of a CTMC

- Consider a (non-absorbing) state $s \in S$ with multiple outgoing transitions, i.e. multiple $s' \in S$ with $\mathbf{R}(s,s')>0$

- 1. Race condition
  - each transition triggered after exponentially distributed delay
    - probability triggered before t time units: $1 - e^{-\mathbf{R}(s,s')\cdot t}$
  - first transition triggered determines the next state

- 2. Separate delay/transition
  - remain in s for delay exponentially distributed with rate E(s)
    - i.e. probability of taking an outgoing transition from s within [0,t] is given by $1-e^{-E(s)\cdot t}$
  - probability that next state is s' is given by $\mathbf{P}^{emb(C)}(s,s')$
    - i.e. $\mathbf{R}(s,s')/E(s) = \mathbf{R}(s,s') / \Sigma_{s' \in S} \mathbf{R}(s,s')$

36

Continuous-time Markov chains

- Infinitesimal generator matrix

$$\mathbf{Q}(s,s') = \begin{cases} \mathbf{R}(s,s') & s \neq s' \\ -\sum_{s \neq s'} \mathbf{R}(s,s') & \text{otherwise} \end{cases}$$

- Alternative definition: a CTMC is:
  - a family of random variables $\{ X(t) \mid t \in \mathbb{R}_{\geq 0} \}$
  - $X(t)$ are observations made at time instant t
  - i.e. $X(t)$ is the state of the system at time instant t
  - which satisfies…
- Memoryless (Markov property)

  $P[X(t_k)=s_k \mid X(t_{k-1})=s_{k-1}, \ldots, X(t_0)=s_0] = P[X(t_k)=s_k \mid X(t_{k-1})=s_{k-1}]$

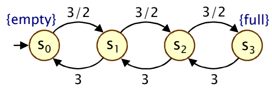
37

Simple CTMC example…

$C = (S, s_{init}, \mathbf{R}, L)$

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

$s_{init} = s_0$



AP = {empty, full}

$L(s_0)=\{\text{empty}\}$, $L(s_1)=L(s_2)=\varnothing$ and $L(s_3)=\{\text{full}\}$

$$\mathbf{R} = \begin{bmatrix} 0 & 3/2 & 0 & 0 \\ 3 & 0 & 3/2 & 0 \\ 0 & 3 & 0 & 3/2 \\ 0 & 0 & 3 & 0 \end{bmatrix} \quad \mathbf{P}^{emb(C)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2/3 & 0 & 1/3 & 0 \\ 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} -3/2 & 3/2 & 0 & 0 \\ 3 & -9/2 & 3/2 & 0 \\ 0 & 3 & -9/2 & 3/2 \\ 0 & 0 & 3 & -3 \end{bmatrix}$$

transition rate matrix | embedded DTMC | infinitesimal generator matrix

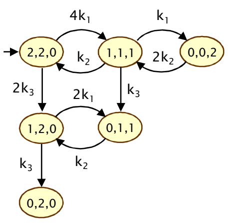
38

Another example

- Chemical reaction system: two species A and B
- Two reactions:

$$A + B \underset{k_2}{\overset{k_1}{\rightleftarrows}} AB \qquad A \xrightarrow{k_3}$$

  - reversible reaction under which species A and B bind to form AB (forwards rate = $|A| \cdot |B| \cdot k_1$, backwards rate = $|AB| \cdot k_2$)
  - degradation of A (rate $|A| \cdot k_3$)
  - $|X|$ denotes number of molecules of species X
- CTMC with state space
  - (|A|,|B|,|AB|)
  - initially (2,2,0)



40

Paths of a CTMC

- An infinite path ω is a sequence $s_0 t_0 s_1 t_1 s_2 t_2 \ldots$ such that
  - $\mathbf{R}(s_i,s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for all $i \in \mathbb{N}$
  - amount of time spent in the jth state: $time(\omega,j)=t_j$
  - state occupied at time t: $\omega@t=s_j$ where j smallest index such that $\Sigma_{i \leq j} t_j \geq t$
- A finite path is a sequence $s_0 t_0 s_1 t_1 s_2 t_2 \ldots t_{k-1} s_k$ such that
  - $\mathbf{R}(s_i,s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for all $i<k$
  - s_k is absorbing ($R(s,s') = 0$ for all $s' \in S$)
  - amount of time spent in the ith state only defined for $j \leq k$: $time(\omega,j)=t_j$ if $j<k$ and $time(\omega,j)=\infty$ if $j=k$
  - state occupied at time t: if $t \leq \Sigma_{i \leq k} t_j$ then $\omega@t$ as above otherwise $t > \Sigma_{i \leq k} t_j$ then $\omega@t=s_k$
- Path(s) denotes all infinite and finite paths starting in s

41

Recall: Probability spaces

- 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 ∅ 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 space (Ω, Σ, Pr)
  - Ω is the sample space
  - Σ is the set of events: σ-algebra on Ω
  - Pr : Σ → [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

42

Probability space

- Sample space: Path(s) (set of all paths from a state s)
- Events: sets of infinite paths
- Basic events: cylinders
  - cylinders = sets of paths with common finite prefix
  - include time intervals in cylinders
- Cylinder is a sequence $s_0,I_0,s_1,I_1,\ldots,I_{n-1},s_n$
  - $s_0,s_1,s_2,\ldots,s_n$ sequence of states where $\mathbf{R}(s_i,s_{i+1})>0$ for $i<n$
  - $I_0,I_1,I_2,\ldots,I_{n-1}$ sequence of of nonempty intervals of $\mathbb{R}_{\geq 0}$
- $Cyl(s_0,I_0,s_1,I_1,\ldots,I_{n-1},s_n)$ set of (infinite and finite paths):
  - $\omega(i)=s_i$ for all $i \leq n$ and $time(\omega,i) \in I_i$ for all $i < n$

43

Probability space

- Define measure over cylinders by induction
- $Pr_s(Cyl(s))=1$
- $Pr_s(Cyl(s,I,s_1,I_1,\ldots,I_{n-1},s_n,I',s'))$ equals:

$$Pr_s(Cyl(s,I,s_1,I_1,\ldots,I_{n-1},s_n)) \cdot \mathbf{P}^{emb(C)}(s_n,s') \cdot \left(e^{-E(s_n)\cdot \inf I'} - e^{-E(s_n)\cdot \sup I'}\right)$$

probability transition from s_n to s' (defined using embedded DTMC)

probability time spent in state s_n is within the interval I'

44

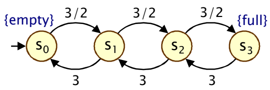
Probability space

- Probability space $(Path(s), \Sigma_{Path(s)}, Pr_s)$ [BHHK03]
- Sample space Ω = Path(s) (infinite and finite paths)
- Event set $\Sigma_{Path(s)}$
  - least σ-algebra on Path(s) containing all cylinders sets $Cyl(s_0,I_0,\ldots,I_{n-1},s_n)$ where:
    - $s_0,\ldots,s_n$ ranges over all state sequences with $\mathbf{R}(s_i,s_{i+1})>0$ for all i
    - $I_0,\ldots,I_{n-1}$ ranges over all sequences of non-empty intervals in $\mathbb{R}_{\geq 0}$ (where intervals are bounded by rationals)
- Probability measure Pr_s
  - Pr_s extends uniquely from probability defined over cylinders

45

Probability space – Example

- Probability of leaving the initial state s_0 and moving to state s_1 within the first 2 time units of operation?
- Cylinder $Cyl(s_0,(0,2],s_1)$



- $Pr_{s0}(Cyl(s_0,(0,2],s_1))$

$= Pr_{s0}(Cyl(s_0)) \cdot \mathbf{P}^{emb(C)}(s_0,s_1) \cdot (e^{-E(s0)\cdot 0} - e^{-E(s0)\cdot 2})$
$= 1 \cdot 1 \cdot (e^{-3/2\cdot 0} - e^{-3/2\cdot 2})$
$= 1 - e^{-3}$
≈ 0.95021

Transient and steady-state behaviour

- Transient behaviour
  - state of the model at a particular time instant
  - $\pi^C_{s,t}(s')$ is probability of, having started in state s, being in state s' at time t (in CTMC C)
  - $\pi^C_{s,t}(s') = Pr_s\{ \omega \in Path^C(s) \mid \omega@t=s' \}$
- Steady-state behaviour
  - state of the model in the long-run
  - $\pi^C_s(s')$ is probability of, having started in state s, being in state s' in the long run
  - $\pi^C_s(s') = \lim_{t\to\infty} \pi^C_{s,t}(s')$
  - intuitively: long-run percentage of time spent in each state

Overview

- Recap: Discrete-time Markov chains (DTMCs)
  - PCTL model checking
  - costs and rewards
- Part 2: Continuous-time Markov chains (CTMCs)
  - exponential distribution and its properties
  - CTMCs: definition, examples, paths & probability spaces
  - CSL: A temporal logic for CTMCs
  - CSL model checking (uniformisation, steady-state)
  - costs & rewards

CSL

- Temporal logic for describing properties of CTMCs
  - CSL = Continuous Stochastic Logic [ASSB00,BHHK03]
  - extension of (non-probabilistic) temporal logic CTL
  - transient, steady-state and path-based properties
- Key additions:
  - probabilistic operator P (like PCTL)
  - steady state operator S
- Example: down $\rightarrow P_{>0.75}$ [¬fail $U^{\leq[1,2.5]}$ up]
  - when a shutdown occurs, the probability of a system recovery being completed between 1 and 2.5 hours without further failure is greater than 0.75
- Example: $S_{<0.1}$[insufficient_routers]
  - in the long run, the chance that an inadequate number of routers are operational is less than 0.1

CSL syntax

- CSL syntax:
  - ϕ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | $P_{\sim p}$ [ψ] | $S_{\sim p}$ [ϕ] (state formulae)
  - ψ ::= X ϕ | ϕ U^{I} ϕ (path formulae)
  - where a is an atomic proposition, I interval of $\mathbb{R}_{\geq 0}$, p $\in$ [0,1], and $\sim \in \{<,>,\leq,\geq\}$
  - unbounded until U is a special case: ϕ_1 U ϕ_2 $\equiv$ ϕ_1 $U^{[0,\infty)}$ ϕ_2
- Quantitative properties: $P_{=?}$ [ψ] and $S_{=?}$ [ϕ]
  - where P/S is the outermost operator

ψ is true with probability ~p

"next"

"time bounded until"

in the "long run" ϕ is true with probability ~p

CSL semantics for CTMCs

- CSL formulae interpreted over states of a CTMC
  - s $\models$ ϕ denotes ϕ is "true in state s" or "satisfied in state s"
- Semantics of state formulae:
  - for a state s of the CTMC (S,s_{init},**R**,L):

- s $\models$ a $\Leftrightarrow$ a $\in$ L(s)
- s $\models$ $\phi_1 \wedge \phi_2$ $\Leftrightarrow$ s $\models$ ϕ_1 and s $\models$ ϕ_2
- s $\models$ $\neg\phi$ $\Leftrightarrow$ s $\models$ ϕ is false
- s $\models$ $P_{\sim p}$ [ψ] $\Leftrightarrow$ Prob(s, ψ) ~ p
- s $\models$ $S_{\sim p}$ [ϕ] $\Leftrightarrow$ $\Sigma_{s' \models \phi}$ $\pi_s(s')$ ~ p

Probability of, starting in state s, satisfying the path formula ψ

Probability of, starting in state s, being in state s' in the long run

CSL semantics for CTMCs

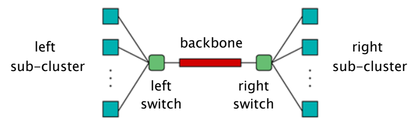
- Prob(s, ψ) is the probability, starting in state s, of satisfying the path formula ψ
  - Prob(s, ψ) = Pr_s {$\omega \in Path_s$ | $\omega \models \psi$ }
- Semantics of path formulae:
  - for a path ω of the CTMC:
  - $\omega \models X\,\phi$ $\Leftrightarrow$ ω(1) is defined and $\omega(1) \models \phi$
  - $\omega \models \phi_1\, U^{I}\, \phi_2$ $\Leftrightarrow$ $\exists t \in I$. ($\omega@t \models \phi_2$ $\wedge$ $\forall t'<t$. $\omega@t' \models \phi_1$)

if ω(0) is absorbing, ω(1) not defined

there exists a time instant in the interval I where ϕ_2 is true and ϕ_1 is true at all preceding time instants

CSL example – Workstation cluster

- Case study: Cluster of workstations [HHK00]
  - two sub-clusters (N workstations in each cluster)
  - star topology with a central switch
  - components can break down, single repair unit



  - minimum QoS: at least ¾ of the workstations operational and connected via switches
  - premium QoS: all workstations operational and connected via switches

CSL example – Workstation cluster

- $S_{=?}$ [minimum]
  - the probability in the long run of having minimum QoS
- $P_{=?}$ [$F^{[t,t]}$ minimum]
  - the (transient) probability at time instant t of minimum QoS
- $P_{<0.05}$ [$F^{[0,10]}$ ¬minimum]
  - the probability that the QoS drops below minimum within 10 hours is less than 0.05
- ¬minimum → $P_{<0.1}$ [$F^{[0,2]}$ ¬minimum]
  - when facing insufficient QoS, the chance of facing the same problem after 2 hours is less than 0.1

CSL example – Workstation cluster

- minimum → $P_{>0.8}$ [minimum $U^{[0,t]}$ premium]
  - the probability of going from minimum to premium QoS within t hours without violating minimum QoS is at least 0.8
- $P_{=?}$ [¬minimum $U^{[t,\infty)}$ minimum]
  - the chance it takes more than t time units to recover from insufficient QoS
- ¬r_switch_up → $P_{<0.1}$ [¬r_switch_up U ¬l_switch_up]
  - if the right switch has failed, the probability of the left switch failing before it is repaired is less than 0.1
- $P_{=?}$ [$F^{[2,\infty)}$ $S_{>0.9}$[minimum]]
  - the probability of it taking more than 2 hours to get to a state from which the long-run probability of minimum QoS is >0.9

Overview

- Recap: Discrete-time Markov chains (DTMCs)
  - PCTL model checking
  - costs and rewards
- Part 2: Continuous-time Markov chains (CTMCs)
  - exponential distribution and its properties
  - CTMCs: definition, examples, paths & probability spaces
  - CSL: A temporal logic for CTMCs
  - CSL model checking (uniformisation, steady-state)
  - costs & rewards

CSL model checking

- Model checking a CSL formula ϕ on a CTMC
  - basic algorithm proceeds by induction on parse tree of ϕ
  - non-probabilistic operators (true, a, ¬, ∧) identical to PCTL
- Main task: computing probabilities for $P_{\sim p}$ [·] and $S_{\sim p}$ [·]
- Untimed properties can be verified on the embedded DTMC
  - properties of the form: $P_{\sim p}$ [X ϕ] or $P_{\sim p}$ [ϕ_1 U ϕ_2]
  - use algorithms for checking PCTL against DTMCs
- Which leaves…
  - time-bounded until operator: $P_{\sim p}$ [ϕ U^I ϕ]
  - steady-state operator: $S_{\sim p}$ [ϕ]

Model checking – Time-bounded until

- Compute Prob(s, ϕ_1 U^I ϕ_2) for all states where I is an arbitrary interval of the non-negative real numbers
- Note:
  - Prob(s, ϕ_1 U^I ϕ_2) = Prob(s, ϕ_1 $U^{cl(I)}$ ϕ_2) where cl(I) denotes the closure of the interval I
  - Prob(s, ϕ_1 $U^{[0,\infty)}$ ϕ_2) = $Prob^{emb(C)}$(s, ϕ_1 U ϕ_2) where emb(C) is the embedded DTMC
- Therefore, 3 remaining cases to consider:
  - I = [0,t] for some $t \in \mathbb{R}_{\geq 0}$ (described in this lecture)
  - I = [t,t'] for some $t \leq t' \in \mathbb{R}_{\geq 0}$ or I = [t,∞) for some $t \in \mathbb{R}_{\geq 0}$
- Two methods: 1. Integral equations; 2. Uniformisation

Time-bounded until (integral equations)

- Computing the probabilities reduces to determining the least solution of the following set of integral equations:
- Prob(s, ϕ_1 $U^{[0,t]}$ ϕ_2) equals
  - 1 if s ∈ Sat(ϕ_2),
  - 0 if s ∈ Sat($\neg\phi_1 \wedge \neg\phi_2$)
  - and otherwise equals

$$\int_0^t \sum_{s' \in S} \left(P^{emb(C)}(s,s') \cdot E(s) \cdot e^{-E(s) \cdot x} \right) Prob(s', \phi_1 \, U^{[0,t-x]} \, \phi_2) \, dx$$

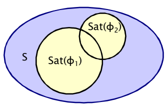
probability of moving from s to s' at time x

probability, in state s', of satisfying until before t–x time units elapse

- One possibility: solve these integrals numerically
  - e.g. trapezoidal, Simpson and Romberg integration
  - expensive, possible problems with numerical stability

Time-bounded until (uniformisation)

- Reduction to transient analysis...
  - on a modified CTMC C'



- Make all ϕ_2 states absorbing
  - in such a state ϕ_1 $U^{[0,x]}$ ϕ_2 holds with probability 1
- Make all $\neg\phi_1 \wedge \neg\phi_2$ states absorbing
  - in such a state ϕ_1 $U^{[0,x]}$ ϕ_2 holds with probability 0
- Formally: modified CTMC C' = C[ϕ_2][$\neg\phi_1 \wedge \neg\phi_2$]
  - where for CTMC C=(S,s_{init},**R**,L), let C[θ]=(S,s_{init},**R**[θ],L) where **R**[θ](s,s')=**R**(s,s') if s ∉ Sat(θ) and 0 otherwise

Time-bounded until (uniformisation)

- Problem then reduces to calculating transient probabilities in the modified CTMC C' :

$$Prob(s, \phi_1 \, U^{[0,t]} \, \phi_2) = \sum_{s' \in Sat(\phi_2)} \underline{\pi}^{C'}_{s,t}(s')$$

$\underline{\pi}_{s,t}{}^{C'}(s')$: transient probability in C': starting in state s, the probability of being in state s' at time t

- To compute for all states s:

$$\underline{Prob}(\phi_1 \, U^{[0,t]} \, \phi_2) = \Pi^{C'}_t \cdot \underline{\phi_2}$$

  - where $\underline{\phi_2}$ is a 0-1 vector characterising ϕ_2
  - and $\Pi^{C'}_t$ is the matrix of all transient probabilities in C'

Computing transient probabilities

- Π_t – matrix of transient probabilities
  - $\Pi_t(s,s')=\underline{\pi}_{s,t}(s')$
- Π_t solution of the differential equation: $\Pi_t' = \Pi_t \cdot \mathbf{Q}$
  - $\mathbf{Q}$ infinitesimal generator matrix
- Can be expressed as a matrix exponential and therefore evaluated as a power series

$$\Pi_t = e^{\mathbf{Q}\cdot t} = \sum_{i=0}^{\infty} (\mathbf{Q}\cdot t)^i / i!$$

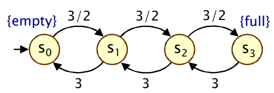
  - computation potentially unstable
  - probabilities instead computed using uniformisation

Uniformisation

- Uniformised DTMC unif(C) of CTMC C =(S,s_{init},**R**,L):
  - unif(C) = (S,s_{init},$\mathbf{P}^{unif(C)}$,L)
  - set of states, initial state and labelling the same as C
  - $\mathbf{P}^{unif(C)} = \mathbf{I} + \mathbf{Q}/q$
  - **I** is the $|S|\times|S|$ identity matrix
  - $q \geq \max \{ E(s) \mid s \in S \}$ is the uniformisation rate
- Each time step (epoch) of uniformised DTMC corresponds to one exponentially distributed delay with rate q
  - if E(s)=q transitions the same as embedded DTMC (residence time has the same distribution as one epoch)
  - if E(s)<q add self loop with probability 1–E(s)/q (residence time longer than 1/q so one epoch may not be 'long enough')

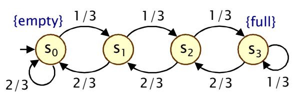
Uniformisation – Example

- CTMC C:



$$R = \begin{bmatrix} 0 & 3/2 & 0 & 0 \\ 3 & 0 & 3/2 & 0 \\ 0 & 3 & 0 & 3/2 \\ 0 & 0 & 3 & 0 \end{bmatrix}$$

- Uniformised DTMC unif(C)
  - let uniformisation rate $q = \max_s \{ E(s) \} = 4.5$
  - $\mathbf{P}^{unif(C)} = \mathbf{I} + \mathbf{Q}/q$



$$\mathbf{P}^{unif(C)} = \begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 2/3 & 0 & 1/3 & 0 \\ 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 2/3 & 1/3 \end{bmatrix}$$

Uniformisation

- Using the uniformised DTMC the transient probabilities can be expressed by:

$$\begin{aligned}\Pi_t &= e^{\mathbf{Q}\cdot t} = e^{q\cdot(\mathbf{P}^{unif(C)}-\mathbf{I})\cdot t} = e^{(q\cdot t)\cdot \mathbf{P}^{unif(C)}} \cdot e^{-q\cdot t} \\ &= e^{-q\cdot t} \cdot \left(\sum_{i=0}^{\infty} \frac{(q\cdot t)^i}{i!} \cdot \left(\mathbf{P}^{unif(C)}\right)^i \right) \\ &= \sum_{i=0}^{\infty} \left(e^{-q\cdot t} \cdot \frac{(q\cdot t)^i}{i!} \right) \cdot \left(\mathbf{P}^{unif(C)}\right)^i \\ &= \sum_{i=0}^{\infty} \gamma_{q\cdot t,i} \cdot \left(\mathbf{P}^{unif(C)}\right)^i\end{aligned}$$

ith Poisson probability with parameter q·t

$\mathbf{P}^{unif(C)}$ stochastic (all entries in [0,1] & rows sum to 1), therefore computations with **P** more numerically stable than **Q**

Uniformisation

$$\Pi_t = \sum_{i=0}^{\infty} \gamma_{q \cdot t,i} \cdot \left(\mathbf{P}^{unif(C)}\right)^i$$

- $(\mathbf{P}^{unif(C)})^i$ is probability of jumping between each pair of states in i steps
- $\gamma_{q \cdot t,i}$ is the ith Poisson probability with parameter $q \cdot t$
  - the probability of i steps occurring in time t, given each has delay exponentially distributed with rate q
- Can truncate the (infinite) summation using the techniques of Fox and Glynn [FG88], which allow efficient computation of the Poisson probabilities

Time-bounded until (uniformisation)

- Recall that for model checking, we require:

$$\underline{\text{Prob}}(\phi_1\ U^{[0,t]}\ \phi_2) = \Pi_t^{C'} \cdot \underline{\phi_2}$$

- So, using uniformisation:

$$\underline{\text{Prob}}(\phi_1\ U^{[0,t]}\ \phi_2) = \sum_{i=0}^{\infty} \left(\gamma_{q \cdot t,i} \cdot \left(\mathbf{P}^{unif(C')}\right)^i \cdot \underline{\phi_2} \right)$$

- This can be computed efficiently using matrix-vector multiplication (avoiding matrix powers):

$$\left(\mathbf{P}^{unif(C')}\right)^0 \cdot \underline{\phi_2} = \underline{\phi_2}$$
$$\left(\mathbf{P}^{unif(C')}\right)^{i+1} \cdot \underline{\phi_2} = \mathbf{P}^{unif(C')} \cdot \left(\left(\mathbf{P}^{unif(C')}\right)^i \cdot \underline{\phi_2} \right)$$

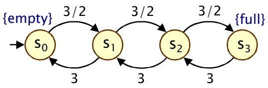
Time-bounded until – Example

- $P_{>0.65}$ [$F^{[0,7.5]}$ full] ≡ $P_{>0.65}$ [true $U^{[0,7.5]}$ full]
  - "probability of the queue becoming full within 7.5 time units"
- State s_3 satisfies full and no states satisfy ¬true
  - in C[full][¬true ∧¬ full] only state s_3 made absorbing

$$\begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 2/3 & 0 & 1/3 & 0 \\ 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

matrix of unif(C[full][¬true ∧¬full]) with uniformisation rate $\max_{s \in S} E(s)=4.5$

s_3 made absorbing



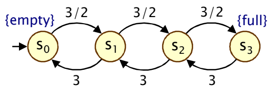
Time-bounded until – Example

- Computing the summation of matrix-vector multiplications

$$\underline{\text{Prob}}(\phi_1\ U^{[0,t]}\ \phi_2) = \sum_{i=0}^{\infty} \left(\gamma_{q \cdot t,i} \cdot \left(\mathbf{P}^{unif(C')}\right)^i \cdot \underline{\phi_2} \right)$$

  - yields $\underline{\text{Prob}}$($F^{[0,7.5]}$ full) ≈ [0.6482, 0.6823, 0.7811, 1]

- $P_{>0.65}$[$F^{[0,7.5]}$ full] satisfied in states s_1, s_2 and s_3



Model Checking – Steady-state

- A state s satisfies the formula $S_{\sim p}[\phi]$ if $\Sigma_{s' \models \phi}\ \underline{\pi}^C_s(s') \sim p$
  - $\underline{\pi}^C_s(s')$ is the probability, having started in state s, of being in state s' in the long run
  - thus model checking reduces to computing and then summing steady-state probabilities for the CTMC
- Steady-state probabilities: $\underline{\pi}^C_s(s') = \lim_{t\to\infty} \underline{\pi}^C_{s,t}(s')$
  - limit exists for all finite CTMCs
  - need to consider underlying graph structure of CTMC
  - i.e. its bottom strongly connected components (BSCCs)
  - irreducible CTMC (comprises one BSCC)
    - solution of one linear equation system
  - reducible CTMC (multiple BSCCs)
    - solve for each BSCC, combine results

70

Irreducible CTMCs

- For an irreducible CTMC:
  - the steady-state probabilities are independent of the starting state: denote the steady state probabilities by $\underline{\pi}^C(s')$
- These probabilities can be computed as
  - the unique solution of the linear equation system:

$$\underline{\pi}^C \cdot \mathbf{Q} = \underline{0} \quad \text{and} \quad \sum_{s\in S} \underline{\pi}^C(s) = 1$$

where **Q** is the infinitesimal generator matrix of C

- Solved by standard means:
  - direct methods, such as Gaussian elimination
  - iterative methods, such as Jacobi and Gauss-Seidel

71

Balance equations

$$\underline{\pi}^C \cdot \mathbf{Q} = \underline{0} \quad \text{and} \quad \sum_{s\in S} \underline{\pi}^C(s) = 1$$

balance the rate of leaving and entering a state

normalisation

For all s ∈ S:

$$\underline{\pi}^C(s) \cdot (-\Sigma_{s'\neq s} \mathbf{R}(s,s')) + \Sigma_{s'\neq s} \underline{\pi}^C(s') \cdot \mathbf{R}(s',s) = 0$$

$$\Leftrightarrow$$

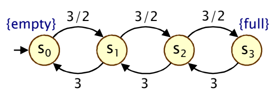
$$\underline{\pi}^C(s) \cdot \Sigma_{s'\neq s} \mathbf{R}(s,s') = \Sigma_{s'\neq s} \underline{\pi}^C(s') \cdot \mathbf{R}(s',s)$$

Equivalent to: $\underline{\pi}^C \cdot \mathbf{P} = \underline{\pi}^C$ where **P** is matrix for embedded DTMC

72

Steady-state – Example

- Model check $S_{<0.1}$[full] on CTMC:



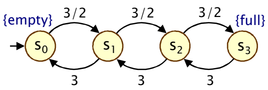
$$\mathbf{Q} = \begin{bmatrix} -3/2 & 3/2 & 0 & 0 \\ 3 & -9/2 & 3/2 & 0 \\ 0 & 3 & -9/2 & 3/2 \\ 0 & 0 & 3 & -3 \end{bmatrix}$$

- CTMC is irreducible (comprises a single BSCC)
  - steady state probabilities independent of starting state
- Solve: $\underline{\pi} \cdot \mathbf{Q} = 0$ and $\Sigma\ \underline{\pi}(s) = 1$

73

Steady-state – Example

- Model check $S_{<0.1}$[full] on CTMC:



- Solve:

$$\begin{aligned}
-3/2\cdot\underline{\pi}(s_0) + 3\cdot\underline{\pi}(s_1) &= 0\\
3/2\cdot\underline{\pi}(s_0) - 9/2\cdot\underline{\pi}(s_1) + 3\cdot\underline{\pi}(s_2) &= 0\\
3/2\cdot\underline{\pi}(s_1) - 9/2\cdot\underline{\pi}(s_2) + 3\cdot\underline{\pi}(s_3) &= 0\\
3/2\cdot\underline{\pi}(s_2) - 3\cdot\underline{\pi}(s_3) &= 0\\
\underline{\pi}(s_0) + \underline{\pi}(s_1) + \underline{\pi}(s_2) + \underline{\pi}(s_3) &= 1
\end{aligned}$$

  - solution: $\underline{\pi}$ = [8/15, 4/15, 2/15, 1/15]
  - $\Sigma_{s' \models Sat(full)}\ \underline{\pi}(s') = 1/15 < 0.1$
  - so all states satisfy $S_{<0.1}$[full]

Reducible CTMCs

- For a reducible CTMC:
  - the steady-state probabilities $\underline{\pi}^C(s')$ depend on start state s
- Find all BSCCs of CTMC, denoted bscc(C)
- Compute:
  - steady-state probabilities $\underline{\pi}^T$ of sub-CTMC for each BSCC T
  - probability $Prob^{emb(C)}$(s, F T) of reaching each T from s
- Then:

$$\underline{\pi}^C_s(s') = \begin{cases} Prob^{emb(C)}(s,\ F\ T)\cdot\underline{\pi}^T(s') & \text{if } s'\in T \text{ for some } T \in bscc(C)\\ 0 & \text{otherwise}\end{cases}$$

CSL model checking complexity

- For CSL model checking of a CTMC, complexity is:
  - linear in $|\Phi|$ and polynomial in $|S|$
  - linear in $q\cdot t_{max}$ (t_{max} is maximum finite bound in intervals)
- Unbounded until ($P_{\sim p}[\Phi_1\ U^{[0,\infty)}\ \Phi_2]$) and steady-state ($S_{\sim p}[\Phi]$)
  - require solution of linear equation system of size |S|
  - can be solved with Gaussian elimination: cubic in |S|
  - precomputation algorithms (max |S| steps)
- Time-bounded until ($P_{\sim p}[\Phi_1\ U^I\ \Phi_2]$)
  - at most two iterative sequences of matrix-vector products
  - operation is quadratic in the size of the matrix, i.e. |S|
  - total number of iterations bounded by Fox and Glynn
  - the bound is linear in the size of q·t (q uniformisation rate)

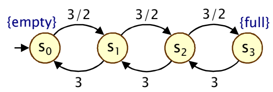
Overview

- Recap: Discrete-time Markov chains (DTMCs)
  - PCTL model checking
  - costs and rewards
- Part 2: Continuous-time Markov chains (CTMCs)
  - exponential distribution and its properties
  - CTMCs: definition, examples, paths & probability spaces
  - CSL: A temporal logic for CTMCs
  - CSL model checking (uniformisation, steady-state)
  - costs & rewards

Rewards (or costs)

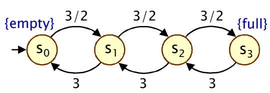
- Like DTMCs, we can augment CTMCs with rewards
  - real-valued quantities assigned to states and/or transitions
  - can be interpreted in two ways: instantaneous/cumulative
  - properties considered here: expected value of rewards
  - formal property specifications in an extension of CSL
- For a CTMC $(S,s_{init},\mathbf{R},L)$, a reward structure is a pair $(\underline{\rho},\iota)$
  - $\underline{\rho} : S \to \mathbb{R}_{\geq 0}$ is a vector of state rewards
  - $\iota : S \times S \to \mathbb{R}_{\geq 0}$ is a matrix of transition rewards
- For cumulative reward-based properties of CTMCs
  - state rewards interpreted as rate at which reward gained
  - if the CTMC remains in state s for $t \in \mathbb{R}_{>0}$ time units, a reward of $t \cdot \underline{\rho}(s)$ is acquired

Reward structures – Examples



- Example: "size of message queue" ← instantaneous
  - $\underline{\rho}(s_i)=i$ and $\iota(s_i,s_j)=0$ $\forall i,j$
- Example: "time for which queue is not full" ← cumulative
  - $\underline{\rho}(s_i)=1$ for $i<3$, $\underline{\rho}(s_3)=0$ and $\iota(s_i,s_j)=0$ $\forall i,j$

Reward structures – Examples



- Example: "number of requests served" ← cumulative

$$\rho = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \iota = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

CSL and rewards

- PRISM extends CSL to incorporate reward-based properties
  - adds R operator like the one added to PCTL

expected reward is ~r

  - ϕ ::= ... | $R_{\sim r}$ [$I^{=t}$] | $R_{\sim r}$ [$C^{\leq t}$] | $R_{\sim r}$ [F ϕ] | $R_{\sim r}$ [S]

"instantaneous" "cumulative" "reachability" "steady-state"

  - where $r,t \in \mathbb{R}_{\geq 0}$, $\sim \in \{<,>,\leq,\geq\}$
- $R_{\sim r}$ [·] means "the expected value of · satisfies ~r"

Types of reward formulae

- Instantaneous: $R_{\sim r}$ [$I^{=t}$]
  - the expected value of the reward at time-instant t is ~r
  - "the expected queue size after 6.7 seconds is at most 2"
- Cumulative: $R_{\sim r}$ [$C^{\leq t}$]
  - the expected reward cumulated up to time-instant t is ~r
  - "the expected requests served within the first 4.5 seconds of operation is less than 10"
- Reachability: $R_{\sim r}$ [F ϕ]
  - the expected reward cumulated before reaching ϕ is ~r
  - "the expected requests served before the queue becomes full"
- Steady-state $R_{\sim r}$ [S]
  - the long-run average expected reward is ~r
  - "expected long-run queue size is at least 1.2"

Reward properties in PRISM

- Quantitative form:
  - e.g. $R_{=?}$ [$C^{\leq t}$]
  - what is the expected reward cumulated up to time-instant t?
- Add labels to R operator to distinguish between multiple reward structures defined on the same CTMC
  - e.g. $R_{\{num\_req\}=?}$ [$C^{\leq 4.5}$]
  - "the expected number of requests served within the first 4.5 seconds of operation"
  - e.g. $R_{\{pow\}=?}$ [$C^{\leq 4.5}$]
  - "the expected power consumption within the first 4.5 seconds of operation"

Reward formula semantics

- Formal semantics of the four reward operators:
  - $s \models R_{\sim r}$ [$I^{=t}$] ⇔ $Exp(s, X_{I=t}) \sim r$
  - $s \models R_{\sim r}$ [$C^{\leq t}$] ⇔ $Exp(s, X_{C\leq t}) \sim r$
  - $s \models R_{\sim r}$ [F Φ] ⇔ $Exp(s, X_{F\Phi}) \sim r$
  - $s \models R_{\sim r}$ [S] ⇔ $\lim_{t\to\infty}(1/t \cdot Exp(s, X_{C\leq t})) \sim r$
- where:
  - Exp(s, X) denotes the expectation of the random variable $X : Path(s) \to \mathbb{R}_{\geq 0}$ with respect to the probability measure Pr_s

Reward formula semantics

- Definition of random variables:
  - path $\omega = s_0 t_0 s_1 t_1 s_2 \ldots$

$$X_{I=k}(\omega) = \underline{\rho}(\omega @ t)$$

(state of ω at time t)

$$X_{C\leq t}(\omega) = \sum_{i=0}^{j_t-1} \left(t_i \cdot \underline{\rho}(s_i) + \iota(s_i, s_{i+1}) \right) + \left(t - \sum_{i=0}^{j_t-1} t_i \right) \cdot \underline{\rho}(s_{j_t})$$

(time spent in state s_i; time spent in state s_{j_t} before t time units have elapsed)

$$X_{F\phi}(\omega) = \begin{cases} 0 & \text{if } s_0 \in Sat(\phi) \\ \infty & \text{if } s_i \notin Sat(\phi) \text{ for all } i \geq 0 \\ \sum_{i=0}^{k_\phi - 1} t_i \cdot \underline{\rho}(s_i) + \iota(s_i, s_{i+1}) & \text{otherwise} \end{cases}$$

  - where $j_t = \min\{ j \mid \Sigma_{i\leq j} t_i \geq t \}$ and $k_\phi = \min\{ i \mid s_i \models \phi \}$

Model checking reward formulae

- Instantaneous: $R_{\sim r}$ [$I^{=t}$]
  - reduces to transient analysis (state of the CTMC at time t)
  - use uniformisation
- Cumulative: $R_{\sim r}$ [$C^{\leq t}$]
  - extends approach for time-bounded until
  - based on uniformisation
- Reachability: $R_{\sim r}$ [F ϕ]
  - can be computed on the embedded DTMC
  - reduces to solving a system of linear equations
- Steady-state: $R_{\sim r}$ [S]
  - similar to steady state formulae $S_{\sim r}$ [ϕ]
  - graph based analysis (compute BSCCs)
  - solve systems of linear equations (compute steady state probabilities of each BSCC)

86

Summary (CTMCs)

- Exponential distribution
  - suitable for modelling failures, waiting times, reactions, …
  - nice mathematical properties
- Continuous-time Markov chains
  - transition delays modelled as exponential distributions
  - probability space over paths
- CSL: Continuous Stochastic Logic
  - extension of PCTL for properties of CTMCs
- CSL model checking
  - extension of PCTL model checking for DTMCs
  - uniformisation: efficient iterative method for transient prob.s
- Costs & rewards
  - capture a wide range of useful quantitative properties
  - properties specified in extension of CSL

87

Thanks for your attention

For an extended version of this material see

M. Kwiatkowska, G. Norman and D. Parker. Stochastic Model Checking. In *SFM'07*, volume 4486 of Lecture Notes in Computer Science (Tutorial Volume), pages 220-270, Springer, 2007.

See also

- http://www.prismmodelchecker.org