Probabilistic Model Checking

Lecture 9
Model Checking for CTMCs

Alessandro Abate

Department of Computer Science
University of Oxford
Overview

• CSL model checking
  – basic algorithm
  – untimed properties
  – time–bounded until
  – the S (steady–state) operator

• Rewards
  – reward structures for CTMCs
  – properties: extension of CSL
  – model checking
CSL: Continuous Stochastic Logic

- **CSL syntax:**

  - $\phi ::= \text{true} \mid a \mid \phi \land \phi \mid \neg \phi \mid P_{\sim p} [\psi] \mid S_{\sim p} [\phi]$ (state formulae)

  - $\psi ::= X \phi \mid \phi U^I \phi$ (path formulae)

  - where $a$ is an atomic proposition, $I$ an interval of $\mathbb{R}_{\geq 0}$, $p \in [0,1]$ and $\sim \in \{<,>,\leq,\geq\}$
**CSL model checking for CTMCs**

- **Algorithm for CSL model checking** [BHHK03]
  - inputs: CTMC $C=(S,s_{\text{init}},R,L)$, CSL formula $\phi$
  - output: $\text{Sat}(\phi) = \{ s \in S \mid s \models \phi \}$, the set of states satisfying $\phi$

- **Often, also consider quantitative results**
  - e.g. compute result of $P_{=?} \left[ F^{[0,t]} \right]$ minimum $\leq 100$

- **Basic algorithm similar to PCTL for DTMCs**
  - proceeds by induction on parse tree of $\phi$

- **For the non-probabilistic operators:**
  - $\text{Sat}(\text{true}) = S$
  - $\text{Sat}(a) = \{ s \in S \mid a \in L(s) \}$
  - $\text{Sat}(\neg \phi) = S \setminus \text{Sat}(\phi)$
  - $\text{Sat}(\phi_1 \land \phi_2) = \text{Sat}(\phi_1) \cap \text{Sat}(\phi_2)$
CSL model checking for CTMCs

- Main task: computing probabilities for $P_{\sim p} [\cdot]$ and $S_{\sim p} [\cdot]$

- $\phi ::= \text{true} \mid a \mid \phi \land \phi \mid \neg \phi$

- $P_{\sim p} [\ X \ \phi \ ] \mid P_{\sim p} [\ \phi \ U \ \phi \ ] \mid P_{\sim p} [\ \phi \ U^I \ \phi \ ] \mid S_{\sim p} [\ \phi \ ]$

- where $\phi_1 \ U \ \phi_2 \equiv \phi_1 \ U^{[0,\infty)} \ \phi_2$
Untimed properties

- Untimed properties can be verified on the embedded DTMC
  - properties of the form: $P_{\sim p} [ X \phi ]$ or $P_{\sim p} [ \phi_1 U \phi_2 ]$
  - use algorithms for checking PCTL against DTMCs

- Certain qualitative time–bounded until formulae can also be verified on the embedded DTMC
  - for any (non–empty) interval $I$

  $$ s \models P_{\sim 0} [ \phi_1 U I \phi_2 ] \text{ if and only if } s \models P_{\sim 0} [ \phi_1 U^{[0,\infty)} \phi_2 ] $$

  - can use precomputation algorithm Prob0
Model checking – Time–bounded until

- Compute $\text{Prob}(s, \phi_1 U^I \phi_2)$ for all states where $I$ is an arbitrary interval of the non–negative real numbers

- Lemmas:
  - $\text{Prob}(s, \phi_1 U^I \phi_2) = \text{Prob}(s, \phi_1 U^{\text{cl}(I)} \phi_2)$
    where $\text{cl}(I)$ denotes the closure of the interval $I$
  - $\text{Prob}(s, \phi_1 U^{[0, \infty)} \phi_2) = \text{Prob}^{\text{emb}(C)}(s, \phi_1 U \phi_2)$
    where $\text{emb}(C)$ is the embedded DTMC

- Therefore, 3 remaining cases to consider:
  - $I = [0, t]$ for some $t \in \mathbb{R}_{\geq 0}$, $I = [t, t']$ for some $t \leq t' \in \mathbb{R}_{\geq 0}$ and $I = [t, \infty)$ 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
  - (note similarity to bounded until for DTMCs)
- $\text{Prob}(s, \phi_1 \mathbin{U}^{[0,t]} \phi_2)$ equals
  - 1 if $s \in \text{Sat}(\phi_2)$,
  - 0 if $s \in \text{Sat}(\neg \phi_1 \land \neg \phi_2)$,
  - and otherwise equals

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

- One possibility: solve these integrals numerically
  - numerical integration, e.g. trapezoidal, Simpson, Romberg
  - expensive, possible issues with numerical stability
Time-bounded until: uniformisation

- Reduction to transient analysis...

- Make all $\phi_2$ states absorbing
  - from such a state $\phi_1 \cup [0,x] \phi_2$
    holds with probability 1

- Make all $\neg \phi_1 \land \neg \phi_2$ states absorbing
  - from such a state $\phi_1 \cup [0,x] \phi_2$
    holds with probability 0

- Formally: Construct CTMC $C[\phi_2][\neg \phi_1 \land \neg \phi_2]$
  - where for CTMC $C=(S, s_{\text{init}}, R, L)$, let $C[\theta]=(S, s_{\text{init}}, R[\theta], L)$, where
    $\theta$ state formula, $R[\theta](s,s')=R(s,s')$ if $s \notin \text{Sat}(\theta)$ and 0 otherwise
Time-bounded until: uniformisation

- Problem then reduces to calculating transient probabilities of the CTMC $C[\phi_2][\neg \phi_1 \land \neg \phi_2]$: 

\[ \text{Prob}(s, \phi_1 U^{[0,t]} \phi_2) = \sum_{s' \in \text{Sat}(\phi_2)} \prod_{s,t}^{C[\phi_2][\neg \phi_1 \land \neg \phi_2]} (s') \]

transient probability: starting in state $s$, the probability of being in state $s'$ at time $t$
Time–bounded until: uniformisation

• Can now adapt uniformisation to computing the vector of probabilities \(\text{Prob}(\phi_1 \cup^{[0,t]} \phi_2)\)
  
  – recall \(\Pi_t\) – matrix of transient probabilities: \(\Pi_t(s, s') = \prod_{s,t}(s')\)
  
  – can be computed via uniformisation: \(\Pi_t = \sum_{i=0}^{\infty} \gamma_{q,t,i} \cdot \left( P^{\text{unif}(C)} \right)^i\)

• Combining with: \(\text{Prob}(s, \phi_1 \cup^{[0,t]} \phi_2) = \sum_{s' \in \text{Sat}(\phi_2)} \prod_{s,t}^{C[\phi_2][\neg \phi_1 \land \neg \phi_2]}(s')\)

\[
\text{Prob}(\phi_1 \cup^{[0,t]} \phi_2) = \Pi_t^{C[\phi_2][\neg \phi_1 \land \neg \phi_2]} \cdot \phi_2
\]

\[
= \left( \sum_{i=0}^{\infty} \gamma_{q,t,i} \cdot \left( P^{\text{unif}(C[\phi_2][\neg \phi_1 \land \neg \phi_2])} \right)^i \right) \cdot \phi_2
\]

\[
= \sum_{i=0}^{\infty} \left( \gamma_{q,t,i} \cdot \left( P^{\text{unif}(C[\phi_2][\neg \phi_1 \land \neg \phi_2])} \right)^i \cdot \phi_2 \right)
\]

– (note analogy: for transient analysis, we post–multiplied from initial vector, now we pre–multiply with Sat–indicator vector)
Time-bounded until: uniformisation

• Have shown that we can calculate the probabilities as:

\[
\text{Prob}(\phi_1 \cup [0,t] \phi_2) = \sum_{i=0}^{\infty} \gamma_{q,t,i} \cdot \left( P_{\text{unif}(C[\phi_2][\neg \phi_1 \land \neg \phi_2])}^i \right) \cdot \phi_2
\]

• Infinite summation can be truncated using techniques by Fox and Glynn [FG88]

• Can compute iteratively to avoid matrix powers:

\[
\left( P_{\text{unif}(C)} \right)^0 \cdot \phi_2 = \phi_2
\]

\[
\left( P_{\text{unif}(C)} \right)^{i+1} \cdot \phi_2 = P_{\text{unif}(C)} \cdot \left( \left( P_{\text{unif}(C)} \right)^i \cdot \phi_2 \right)
\]

– (note slight imprecision in Greek gamma var, 1st equation)
Time-bounded until – Example

- $P_{>0.65} [ F^{[0,7.5]} \text{ full } ] \equiv P_{>0.65} [ \text{ true U}^{[0,7.5]} \text{ full } ]$
  - “probability of the queue becoming full within 7.5 time units”
- State $s_3$ satisfies full and no states satisfy $\neg \text{true}$
  - in $C[\text{full}][\neg \text{true } \land \neg \text{ full}]$ only state $s_3$ made absorbing

\[
\begin{bmatrix}
\frac{2}{3} & \frac{1}{3} & 0 & 0 \\
\frac{2}{3} & 0 & \frac{1}{3} & 0 \\
0 & \frac{2}{3} & 0 & \frac{1}{3} \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

matrix of $\text{unif}(C[\text{full}][\neg \text{true } \land \neg \text{full}])$
with uniformisation rate $\max_{s \in S} E(s) = 4.5$ ($= 3 + 3/2$)

$s_3$ made absorbing
Time-bounded until – Example

- Computing the summation of matrix-vector multiplications

\[
\text{Prob}(\Phi_1 U^{[0,t]} \Phi_2) = \sum_{i=0}^{\infty} \left( Y_{q,t,i} \cdot \left( P^{\text{unif}(C[\Phi_2][\neg \Phi_1 \land \neg \Phi_2])} \right)^i \cdot \Phi_2 \right)
\]

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

- \( P_{>0.65}[ F^{[0,7.5]} \text{ full } ] \) satisfied in states \( s_1, s_2 \) and \( s_3 \)
Time-bound until $- P_{\sim p}[\phi_1 U^{[t,t']} \phi_2]$

- In this case the computation can be split into two parts:
  - 1. Probability of remaining in $\phi_1$ states until time $t$
    - can be computed as transient probabilities on the CTMC where states satisfying $\neg \phi_1$ have been made absorbing
  - 2. Probability of reaching a $\phi_2$ state, while remaining in states satisfying $\phi_1$, within the time interval $[0,t'−t]$
    - i.e. computing $\text{Prob}(\phi_1 U^{[0,t'-t]} \phi_2)$

$$\text{Prob}(s, \phi_1 U^{[t,t']} \phi_2) = \sum_{s' \in \text{Sat}(\phi_1)} \prod_{s,t}^{C[-\phi_1]} (s') \cdot \text{Prob}(s', \phi_1 U^{[0,t'-t]} \phi_2)$$

- sum over states satisfying $\phi_1$
- Probability of reaching state $s'$ at time $t$ and satisfying $\phi_1$ up until this point
- probability $\phi_1 U^{[0,t'-t]} \phi_2$ holds in $s'$
Time–bounded until \( P_{\sim p} [\phi_1 \cup [t,t'] \phi_2] \)

- Let \( \text{Prob}_{\phi_1}(s, \phi_1 U^{[0,t'-t]} \phi_2) = \text{Prob}(s, \phi_1 U^{[0,t'-t]} \phi_2) \) if \( s \in \text{Sat}(\phi_1) \), and 0 otherwise.
- From the previous slide we have:

\[
\text{Prob}(\phi_1 U^{[t,t']} \phi_2) = \prod_t^{C[-\phi_1]} \cdot \text{Prob}_{\phi_1}(\phi_1 U^{[0,t'-t]} \phi_2)
\]

\[
= \left( \sum_{i=0}^{\infty} Y_{q,t,i} \cdot \left( P_{\text{unif}(C[-\phi_1])} \right)^i \right) \cdot \text{Prob}_{\phi_1}(\phi_1 U^{[0,t'-t]} \phi_2)
\]

\[
= \sum_{i=0}^{\infty} \left( Y_{q,t,i} \cdot \left( P_{\text{unif}(C[-\phi_1])} \right)^i \cdot \text{Prob}_{\phi_1}(\phi_1 U^{[0,t'-t]} \phi_2) \right)
\]

- summation can be truncated using Fox and Glynn [FG88]
- can compute iteratively (only scalar and matrix–vector operations)
Time–bounded until – $P_{\sim p}[\phi_1 \cup^{[t,\infty)} \phi_2]$

- Letting $\text{Prob}_{\phi_1}(s, \phi_1 \cup^{[0,\infty)} \phi_2) = \text{Prob}(s, \phi_1 \cup^{[0,\infty)} \phi_2)$ if $s \in \text{Sat}(\phi_1)$ and 0 otherwise, we have:

$$\text{Prob}(\phi_1 \cup^{[t,\infty)} \phi_2) = \prod_t C_{[-\phi_1]} \cdot \text{Prob}_{\phi_1}^{\text{emb}(C)}(\phi_1 \cup \phi_2)$$

$$= \left( \sum_{i=0}^\infty Y_{q,t,i} \cdot \left( P_{\text{unif}(C[-\phi_1])} \right)^i \right) \cdot \text{Prob}_{\phi_1}^{\text{emb}(C)}(\phi_1 \cup \phi_2)$$

$$= \sum_{i=0}^\infty \left( Y_{q,t,i} \cdot \left( P_{\text{unif}(C[-\phi_1])} \right)^i \cdot \text{Prob}_{\phi_1}^{\text{emb}(C)}(\phi_1 \cup \phi_2) \right)$$

- summation can be truncated using Fox and Glynn [FG88]
- can compute iteratively (only scalar and matrix–vector operations)
Model Checking – $S_{\sim p}[\phi ]$

- A state $s$ satisfies the formula $S_{\sim p}[\phi ]$ if $\sum_{s'} \models \phi \pi^C_s(s') \sim p$
  - $\pi^C_s(s')$ is probability, having started in state $s$, of being in state $s'$ in the long run
- Thus reduces to computing and then summing steady-state probabilities for the CTMC (recall results earlier)

- If CTMC is irreducible:
  - solution of one system of linear equations
- If CTMC is reducible:
  - determine set of BSCCs for the CTMC
  - solve two systems of linear equations, for each BSCC $T$:
    1. one to obtain the vector $\text{ProbReach}^{\text{emb}(C)}(T)$
    2. the other to compute the steady state probabilities $\pi^T$ for $T$
$S_{\sim_p [ \Phi ]}$ – Example

- $S_{<0.1}$ [full]
- CTMC is irreducible (comprises a single BSCC)
  - steady state probabilities independent of starting state
  - can be computed by solving $\pi \cdot Q = 0$ and $\sum \pi(s) = 1$

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

Diagram:

- $S_0$ to $S_1$ with transition rate $3/2$
- $S_1$ to $S_2$ with transition rate $3/2$
- $S_2$ to $S_3$ with transition rate $3/2$
- $S_3$ to $S_0$ with transition rate $3$
- States:
  - $S_0$: {empty}
  - $S_1$
  - $S_2$
  - $S_3$: {full}
$S_{\sim_p}[\phi]$ – Example

$-3/2 \cdot \pi(s_0) + 3 \cdot \pi(s_1) = 0$
$3/2 \cdot \pi(s_0) - 9/2 \cdot \pi(s_1) + 3 \cdot \pi(s_2) = 0$
$3/2 \cdot \pi(s_1) - 9/2 \cdot \pi(s_2) + 3 \cdot \pi(s_3) = 0$
$3/2 \cdot \pi(s_2) - 3 \cdot \pi(s_3) = 0$

$\pi(s_0) + \pi(s_1) + \pi(s_2) + \pi(s_3) = 1$

- solution: $\pi = [ 8/15, 4/15, 2/15, 1/15 ]$
- $\sum_{s'} = \text{Sat(full)} \pi(s') = 1/15 < 0.1$
- so all states satisfy $S_{<0.1}[\text{full}]$
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: expected value of rewards
  – formal property specifications as an extension of CSL

• For a CTMC \((S, s_{\text{init}}, R, L)\), a reward structure is a pair \((\rho, \iota)\)
  – \(\rho : S \rightarrow \mathbb{R}_{\geq 0}\) is a vector of state rewards
  – \(\iota : S \times S \rightarrow \mathbb{R}_{\geq 0}\) is a matrix of transition rewards

• For cumulative reward-based properties of CTMCs
  – state rewards interpreted as rate at which reward is gained
  – if the CTMC remains in state \(s\) for \(t \in \mathbb{R}_{\geq 0}\) time units, a reward of \(t \cdot \rho(s)\) is acquired
Reward structures – Examples

• Example: “size of message queue”
  – $\rho(s_i)=i$ and $\iota(s_i,s_j)=0 \ \forall \ i,j$

• Example: “time for which queue is not full”
  – $\rho(s_i)=1$ for $i<3$, $\rho(s_3)=0$ and $\iota(s_i,s_j)=0 \ \forall \ i,j$
Reward structures – Examples

- Example: “number of requests served”

\[ \rho = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]
CSL and rewards

- **PRISM** extends CSL to incorporate reward–based properties
  - adds R operator, as in PCTL

\[
\phi ::= \ldots | R_{\sim r} [ l=t ] | R_{\sim r} [ C \leq t ] | R_{\sim r} [ F \phi ] | R_{\sim r} [ S ]
\]

- where \( r, t \in \mathbb{R}_{\geq 0} \), \( \sim \in \{<,>,\leq,\geq\} \)

- \( R_{\sim r} [ \cdot ] \) means “the expected value of \( \cdot \) satisfies \( \sim r \)”
Types of reward formulae

- **Instantaneous**: $R_{\sim r}[I^=t]$
  - the expected value of the reward at time instant $t$ is $\sim r$
  - “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 $\sim r$
  - “the expected requests served within the first 4.5 seconds of operation is less than 10”

- **Reachability**: $R_{\sim r}[F \phi]$
  - the expected reward cumulated before reaching $\phi$ is $\sim r$
  - “the expected requests served before the queue becomes full”

- **Steady-state**: $R_{\sim r}[S]$
  - the long-run average expected reward is $\sim r$
  - “expected long-run queue size is at least 1.2”
Reward properties in PRISM

- **Quantitative form:**
  - e.g. $R_{=\text{?}} [ C \leq t ]$
  - 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. $R_{\text{num\_req}=\text{?}} [ C \leq 4.5 ]$
  - “the expected number of requests served within the first 4.5 seconds of operation”
  - e.g. $R_{\text{pow}=\text{?}} [ C \leq 4.5 ]$
  - “the expected power consumption within the first 4.5 seconds of operation”
Reward formula semantics

- Formal semantics of the four reward operators:

  \[ s \vDash R_{\sim r} [ I=t ] \iff \text{Exp}(s, X_{I=t}) \sim r \]
  \[ s \vDash R_{\sim r} [ C\leq t ] \iff \text{Exp}(s, X_{C\leq t}) \sim r \]
  \[ s \vDash R_{\sim r} [ F \Phi ] \iff \text{Exp}(s, X_{F\Phi}) \sim r \]
  \[ s \vDash R_{\sim r} [ S ] \iff \lim_{t \to \infty} \left( \frac{1}{t} \cdot \text{Exp}(s, X_{C\leq t}) \right) \sim r \]

- Where recall:
  - \text{Exp}(s, X) denotes the expectation of the random variable \( X : \text{Path}(s) \to \mathbb{R}_{\geq 0} \) with respect to the probability measure \( \text{Pr}_s \)
Reward formula semantics

- Definition of random variables:
  
  - path $\omega = s_0t_0s_1t_1s_2 \ldots$
    
  - $X_{i=k}(\omega) = \rho(\omega @ t)$
  
  - $X_{C\leq t}(\omega) = \sum_{i=0}^{i_{t-1}} (t_i \cdot \rho(s_i) + \iota(s_i,s_{i+1}) + \left(t - \sum_{i=0}^{j_{t-1}} t_i\right) \cdot \rho(s_{j_t}))$
  
  - $X_{F\phi}(\omega) = \begin{cases} 
  0 & \text{if } s_0 \in \text{Sat}(\phi) \\
  \infty & \text{if } s_i \notin \text{Sat}(\phi) \text{ for all } i \geq 0 \\
  \sum_{i=0}^{k_{\phi}-1} t_i \cdot \rho(s_i) + \iota(s_i,s_{i+1}) & \text{otherwise} 
  \end{cases}$

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

- (note: typo in index of first formula: $I=k$ should be $I=t$)
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**: $R_{\sim r}[F \phi]$
  - 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 $S_{\sim r}[\phi]$
  - *graph based analysis* (compute BSCCs)
  - *solve systems of linear equations* (compute steady state probabilities of each BSCC)
CSL model checking complexity

- For model checking of a CTMC complexity:
  - linear in $|\Phi|$ and polynomial in $|S|$
  - linear in $q \cdot t_{\text{max}}$ ($t_{\text{max}}$ is maximum finite bound in intervals, $q$ is uniformisation rate)

- $P_{\sim p}[\Phi_1 \cup [0, \infty) \Phi_2], S_{\sim p}[\Phi], R_{\sim r}[F \Phi]$ and $R_{\sim r}[S]$
  - require solution of linear equation system of size $|S|$
  - can be solved with Gaussian elimination: cubic in $|S|$
  - precomputation algorithms (max $|S|$ steps)

- $P_{\sim p}[\Phi_1 \cup [l_1, \infty) \Phi_2], R_{\sim r}[C \leq t]$ and $R_{\sim r}[l=t]$
  - 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 $q \cdot t$
Summing up…

- **Model checking a CSL formula \( \phi \) on a CTMC**
  - recursive: bottom–up traversal of parse tree of \( \phi \)
- **Main work: computing probabilities for P and S operators**
  - untimed \( (X \, \Phi, \, \Phi_1 \, U \, \Phi_2) \): perform on embedded DTMC
  - time–bounded until: use uniformisation–based methods, rather than more expensive solution of integral equations
  - other forms of time–bounded until, i.e. \([t_1,t_2]\) and \([t,\infty)\), reduce to two sequential computations like for \([0,t]\)
  - S operator: summation of steady–state probabilities
- **Rewards – similar to DTMCs**
  - except for continuous–time accumulation of state rewards
  - extension of CSL with R operator
  - model checking of R comparable with that of P