DD2552 - Seminars on Theoretical Computer Science, Programming Languages and Formal Methods, Seminar 2

Karl Palmskog (palmskog@kth.se)

2021-09-02

Last Seminar and Today

Last seminar:

- syntax and informal semantics of CTL, computational tree logic
- intuition behind PCTL, the probabilistic extension of CTL

Today:

- towards a formal semantics of PCTL
- reasoning manually on DTMCs with PCTL

Reminder on Decidability and Truth

- problems can be expressed as questions having a yes/no answer
  - "is this list sorted?"
  - "does this graph have a k-coloring?"
- some questions can be answered by using an algorithm - we say they are **decidable**
- in this course, we will have many questions with "yes" answers that must be established "manually" without an obvious decision algorithm
- defining a logic often means defining what someone has to do to give a "yes" answer

Defining Logics in Computer Science

1. definition of logic's syntax
   - what is a formula?
   - sometimes includes lexing
2. definition of logic's semantics
   - when is a formula true?
   - may require use of mathematical structures
   - should be concise and canonical
3. metatheory of the logic
   - why do we care about formulas being true or false?
   - can we decide if formulas are true?
   - how efficiently can we decide?

Transition Systems Again

$\mathcal{M} = (S, \to, L)$ where

- S is a (finite) set of states
- $\to \subseteq S \times S$
- $L \colon S \mapsto 2^{\mathrm{AP}}$ (AP are atomic propositions)

Paths in a Transition System

A **path** for a transition system $(S, \rightarrow, L)$ is an infinite sequence

$$\pi = s_0, s_1, \ldots, s_n, \ldots$$

where $s_i \in S$ and $s_k \rightarrow s_{k+1}$.

We write the ith element of a path π as $\pi(i)$.

We write the prefix of a path ("trace") of length n as $\pi[0, n]$.

CTL Syntax Fragment

$$
\begin{aligned}
\phi &::= \top \mid a \mid \neg\phi \mid \phi \wedge \phi \\
\psi &::= \phi \, U^{\leq t} \, \phi \\
& t \in Z^{\geq 0}
\end{aligned}
$$

Defining CTL Semantics

- we define the relation $\mathcal{M}, s \models \phi$ recursively
  - "ϕ is true in the state s in $\mathcal{M}$"
- we then define the relation $\mathcal{M}, \pi \models \psi$
  - "ψ is true for the path π in $\mathcal{M}$"
- idea: we can unfold relations to manually prove by induction that a formula is true

More Operators in CTL

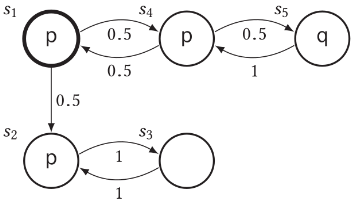
- how do we define $\phi \vee \phi'$?
- how do we define $\phi \rightarrow \phi'$?
- how do we define "weak until", $\phi \, W^{\leq t} \, \phi'$?

Discrete Time Markov Chains Again

$\mathcal{M} = (S, s_i, M, L)$ where

- S is a (finite) set of states
- $s_i \in S$ is the initial state
- $M\colon S \times S \mapsto [0,1]$ defines transition probabilities, where
  - for all $s \in S$, $\sum_{s' \in S} M(s, s') = 1$
- $L\colon S \mapsto 2^{\mathrm{AP}}$ (AP are atomic propositions)

Markov Chain Transition Probabilities



M	s_1	s_2	s_3	s_4	s_5
s_1	0	0.5	0	0.5	0
s_2	0	0	1	0	0
s_3	0	1	0	0	0
s_4	0.5	0	0	0	0.5
s_5	0	0	0	1	0

- probability of ending up in s_5 after two steps?
- probability of the sequence of steps s_1, s_2, s_3, s_2?

PCTL Syntax Fragment

$$\phi ::= \top \mid a \mid \neg\phi \mid \phi \wedge \phi \mid P_{\geq\theta}(\psi)$$
$$\psi ::= \phi\, U^{\leq t}\, \phi$$
$$t \in Z^{\geq 0}, \quad \theta \in [0, 1]$$

PCTL Semantics

- we follow CTL semantics for regular operators
- $P_{\geq\theta}(\psi)$ is defined using the measure operator μ_s

QuaTex Syntax

$$
\begin{aligned}
Q &::= D \ \ \underline{\texttt{eval}}\ \mathbf{E}[PExp]; \\
D &::= \ \text{set of } Defn \\
Defn &::= N(x_1, \ldots, x_m) = PExp; \\
SExp &::= c \mid f \mid F(SExp_1, \ldots, SExp_k) \mid x_i \\
PExp &::= SExp \mid \bigcirc N(SExp_1, \ldots, SExp_n) \\
&\quad \mid \underline{\texttt{if}}\ \ SExp\ \ \underline{\texttt{then}}\ \ PExp_1\ \ \underline{\texttt{else}}\ \ PExp_2\ \ \underline{\texttt{fi}}
\end{aligned}
$$

$$
\begin{aligned}
\texttt{UntilBounded}(\phi_1, \phi_2, t) = {} & \underline{\texttt{if}}\ t > time()\ \underline{\texttt{then}}\ 0\ \underline{\texttt{else}}\ \underline{\texttt{if}}\ \phi_2\ \underline{\texttt{then}}\ 1\ \underline{\texttt{else}} \\
& \underline{\texttt{if}}\ \phi_1\ \underline{\texttt{then}}\ \bigcirc\,(\texttt{UntilBounded}(\phi_1, \phi_2, t))\ \underline{\texttt{else}}\ 0\ \underline{\texttt{fi}}\ \underline{\texttt{fi}}\ \underline{\texttt{fi}};
\end{aligned}
$$

QuaTex Semantics

$$
\begin{aligned}
&(s)[\![c]\!]_D = c \\
&(s)[\![f]\!]_D = f(s) \\
&(s)[\![F(SExp_1, \ldots, SExp_k)]\!]_D = F((s)[\![SExp_1]\!]_D, \ldots, (s)[\![SExp_k]\!]_D) \\
&(s)[\![\mathbf{E}[PExp]]\!]_D = \mathbf{E}[(\pi)[\![PExp]\!]_D] \text{ for } \pi \in \mathit{Paths}(s) \\
&(\pi)[\![SExp]\!]_D = (\pi[0])[\![SExp]\!]_D \\
&(\pi)[\![\underline{\mathtt{if}}\ SExp\ \underline{\mathtt{then}}\ PExp_1\ \underline{\mathtt{else}}\ PExp_2\ \underline{\mathtt{fi}}]\!]_D = \\
&\qquad\qquad \text{if } (\pi[0])[\![SExp]\!]_D = \text{true then } (\pi)[\![PExp_1]\!]_D \text{ else } (\pi)[\![PExp_2]\!]_D \\
&(\pi)[\![\bigcirc N(SExp_1, \ldots, SExp_m)]\!]_D = \\
&\qquad\qquad (\pi^{(1)})[\![B[x_1 \mapsto (\pi[0])[\![SExp_1]\!]_D, \ldots, x_m \mapsto (\pi[0])[\![SExp_m]\!]_D]]\!]_D \\
&\qquad\qquad\qquad\qquad \text{where } N(x_1, \ldots, x_m) = B; \in D
\end{aligned}
$$