Computational Tree Logic

1 Syntax of CTL

Definition: State formula

A state formula in CTL is defined by the following grammar

$$\Phi := true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid A\varphi \mid E\varphi$$

where a is an atomic proposition, and Φ_1, Φ_2 are state formula and φ is a path formula defined by

$$\varphi = X\Phi \mid G\Phi \mid F\Phi \mid \Phi_1 U \Phi_2 \mid \Phi_1 W \Phi 2$$

Here the symbols stand for

- A - for all
- E - There exists
- X - Next
- G - Globally (always)
- F - Finally (eventually)
- U - Until
- W - Weak until

Informally, the quantifiers should come in pairs, a quantifier over paths followed by a path specific one

2 Formal semantics

Given a set of atomic propositions AP and a transition system $TS = (S, \rightarrow, I, AP, L)$

Definition: Satisfaction relation

Defined recursively by

$$\begin{array}{ll} s \models true & \\ s \models a & \text{iff } a \in L(s) \\ s \models \Phi_1 \wedge \Phi_2 & \text{iff } s \models \Phi_1 \text{ and } s \models \Phi_2 \\ s \models \neg\Phi & \text{iff } s \not\models \Phi \\ s \models \mathbf{A}\varphi & \text{iff } \pi \models \varphi \text{ for all } \pi \in \text{Paths}(s) \\ s \models E\varphi & \text{iff } \pi \models \varphi \text{ for some } \pi \in \text{Paths}(s) \end{array}$$

For a state $s \in S$ and by

$$\begin{array}{ll} \pi \models X\Phi & \pi_1 \models \Phi \\ \pi \models \Phi_1 U \Phi_2 & \text{iff there is } i \geqslant 0 s.t. \pi_i \models \Phi_2 \text{ and } \pi_j \models \Phi_1 \text{ for all } 0 \leqslant j < i \end{array}$$

Where the path π is viewed as a sequence of states $\pi_0, \pi_1 ...$

3 CTL vs LTL

We can transform a state CTL formula into an LTL one by simply omitting the quantifiers over paths. The resulting formula is equivalent to the original one iff the property defined by the original, CTL, formula is expressible in the LTL.

4 CTL*

Definition: State formula in CTL*

A state formula in CTL* is defined by the following grammar

$$\Phi := true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid A\varphi \mid E\varphi$$

where a is an atomic proposition, and Φ_1, Φ_2 are the state formulae and φ is a path formula, defined by

$$\varphi = \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid X\Phi \mid G\Phi \mid F\Phi \mid \Phi_1 U \Phi_2 \mid \Phi_1 W \Phi 2$$

where a path satisfies a state formula iff the state state of the path satisfies it:

$$\pi \models \Phi \text{ iff } \pi_0 \models \Phi$$

5 Model Checking LTL

Lemma - Existential normal form for CTL
A formula in CTL may be generated by the following grammar

$$\Phi := \text{true} | a | \Phi_1 \wedge \Phi_2 | \neg\Phi | \mathbf{EX}\Phi | \mathbf{E}(\Phi_1 \mathbf{U} \Phi_2) | \mathbf{EG}\Phi$$

Definition

Given a TS $(S, \rightarrow, I, AP, L)$, for a CTL-formula Φ, denote by $Sat(\Phi)$ the subsets of states that satisfy Φ, i.e.
$Sat(\Phi) = \{s \in S | s \models \Phi\}$

Theorem 1 *$Sat(\Phi)$ satisfies the following*

- *$Sat(true)=S$*
- *$Sat(a) = \{s \in S | a \in L(s)\}$ for every $a \in AP$*
- *$Sat(\Phi_1 \wedge \Phi_2) = Sat(\Phi_1) \cap Sat(\Phi_2)$*
- *$Sat(\neg\Phi) = S \backslash Sat(\Phi)$*
- *$Sat(EX\Phi) = \{s \in S | succ(s) \cap Sat(\Phi) \neq \emptyset\}$*
- *$Sat(E(\Phi_1 U \Phi_2))$ is the smallest $T \subseteq S$ such that*
  - *$Sat(\Phi_2) \subseteq T$, and*
  - *$s \in Sat(\Phi_2)$ and $succ(s) \cap T \neq \emptyset \Rightarrow s \in T$*
- *$Sar(EG\Phi)$ is the biggest $T \subseteq S$ such that*
  - *$T \subseteq Sat(\Phi)$, and*
  - *$s \in T \Rightarrow succ(s) \cap T \neq \emptyset$*

where succ(s) is the direct successor os s, i.e. $succ(s) = \{t \in S | s \rightarrow t\}$