

Introduction to formal methods

September 8, 2008

Chapter 4

Propositional Branching Time Temporal Logic, CTL*

Branching time temporal logic, or Computation Tree Logic (CTL*), formalizes reasoning about different worlds that are related in a branching order, which reflects several time lines. This means, that models include branching but also include linear paths. Apart from the “along a path” operator U , also operators to reason about the existence of paths is introduced. All of these operators are expressible via adding one new operator. The level of reasoning remains that of propositions.

There are again some changes in notation, again for historical reasons.

4.1 Syntax of CTL*

Let PV be a set of propositional variables containing the symbol **true**. The set of state formulas and the set of path formulas of CTL* are defined inductively:

- S1. every element of PV is a state formula,
- S2. if φ and ψ are state formulas, then so are $\varphi \vee \psi$ and $\varphi \wedge \psi$,
- S3. if φ is a path formula, then $A\varphi$ and $E\varphi$ are state formulas,
- P1. any state formula φ is also a path formula,
- P2. if φ, ψ are path formulas, then so are $\varphi \wedge \psi$ and $\varphi \vee \psi$,
- P3. if φ, ψ are path formulas, then so are $X\varphi$, and $(\varphi U \psi)$.

The modal operator A has the intuitive meaning “for all paths”, E - “there is a path satisfying”, U denotes Until. The language of CTL* consists of the set of all state formulas.

The following abbreviations will be used:

- $F\varphi \stackrel{def}{=} U(\mathbf{true}, \varphi)$,
- $G\varphi \stackrel{def}{=} \neg F\neg\varphi$.

Sublogic of CTL*

CTL: The sublogic of CTL* in which the state modalities A, E and the path modalities X, U may only appear paired in the combinations AX, A(... U ...), and EX, E(... U ...).

4.2 Semantics of CTL*

Let $M = (W, R, V)$ be a *model*, where W is a non-empty set of worlds, $R \subseteq W \times W$ is a total binary relation on W (i.e., each world has at least one successor), and $V : W \times PV \rightarrow \{tt, ff\}$ is a valuation function which assigns to each world and each propositional variable the value *tt* or *ff*. A (*forward*) *path* starting at w_0 is an infinite sequence $\pi = (w_0, w_1, \dots)$ of worlds such that $(w_i, w_{i+1}) \in R$ for each $i \geq 0$. The suffix (w_i, w_{i+1}, \dots) of the path π is denoted by π^i . $M, w \models \phi$ denotes that the formula ϕ is *true* at the world w in the model M , whereas $M, \pi \models \phi$ denotes that the formula ϕ is *true* on the path π in the model M . M is omitted, if it is implicitly understood. For a world w and a path π , the relation \models is defined inductively as follows:

- S1. $w \models p$ iff $V(w, p) = tt$, for $p \in PV$,
- S2. $w \models \alpha \wedge \beta$ iff $w \models \alpha$ and $w \models \beta$,
 $w \models \neg\alpha$ iff not $w \models \alpha$,
- S3. $w \models E\alpha$ iff $\pi \models \alpha$ for some computation π starting at w ,
- P1. $\pi \models \alpha$ iff $w_0 \models \alpha$ for any state formula α ,
- P2. $\pi \models \alpha \wedge \beta$ iff $\pi \models \alpha$ and $\pi \models \beta$,
 $\pi \models \neg\alpha$ iff not $\pi \models \alpha$,

P3. $\pi \models X\alpha$ iff $\pi^1 \models \alpha$,

$\pi \models (\alpha U \beta)$ iff $(\exists i \geq 0) \pi^i \models \beta$ and $(\forall j : 0 \leq j < i) \pi^j \models \alpha$, .

A state formula α is *valid in the model* M (written $M \models \alpha$), if for every state w in M , $M, w \models \alpha$. A set of state formulas L is *valid in the model* M (written $M \models L$), if for every formula $\alpha \in L$, $M \models \alpha$. A state formula α is said to be *valid* (written $\models \alpha$), if for every model M , $M \models \alpha$. A state formula α is *satisfiable*, if for some model M and some state w in M , $M, w \models \alpha$. In this case M is said to be a model of α . A state formula α is said to be a *semantical consequence* of a set of state formulas L (written $L \models \alpha$), if $M \models L$ implies $M \models \alpha$, for every model M .

4.3 MUTEX

We can now formalize the third property that we formulated about MUTEX more properly. And also still the first and second one.

- Mutual Exclusion: $M \models AG\neg(c_1 \wedge c_2)$.
- Eventual Access: $M \models AG(t_1 \Rightarrow AFc_1)$.
- Non-blocking: $M \models AG(n_1 \Rightarrow EFt_1)$.