Branching-Time Logic

CTL

A : along All paths

E : along at least one path (there exists)

Computation Tree Logic, or CTL for short, is a *branching-time* logic, meaning that its model of time is a tree-like structure in which the future is not determined; there are different paths in the future, any one of which might be the 'actual' path that is realised.

SYNTAX

Definition 3.12 We define CTL formulas inductively via a Backus Naur form as done for LTL:

$$\phi ::= \perp \mid \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid \mathrm{AX}\,\phi \mid \mathrm{EX}\,\phi \mid \mathrm{AF}\,\phi \mid \mathrm{EF}\,\phi \mid \mathrm{AG}\,\phi \mid \mathrm{EG}\,\phi \mid \mathrm{A}[\phi\ \mathrm{U}\ \phi] \mid \mathrm{E}[\phi\ \mathrm{U}\ \phi]$$

where p ranges over a set of atomic formulas.

symbol pairs are indivisible!

CTL does not subsume LTL

Convention 3.13 We assume similar binding priorities for the CTL connectives to what we did for propositional and predicate logic. The unary connectives (consisting of $\neg$ and the temporal connectives AG, EG, AF, EF, AX and EX) bind most tightly. Next in the order come $\wedge$ and $\vee$; and after that come $\rightarrow$, AU and EU.

AU and EU mix infix & prefix notation

$$\mathrm{A}[\mathrm{AX}\,\neg p\ \mathrm{U}\ \mathrm{E}[\mathrm{EX}\,(p \wedge q)\ \mathrm{U}\ \neg p]]$$

"Pure" Infix : $\phi_1 \,\mathrm{AU}\, \phi_2$

"Pure" Prefix : $\mathrm{AU}\ \phi_1\ \phi_2$

$\mathrm{A}[\phi_1 \mathrm{U} \phi_2]$

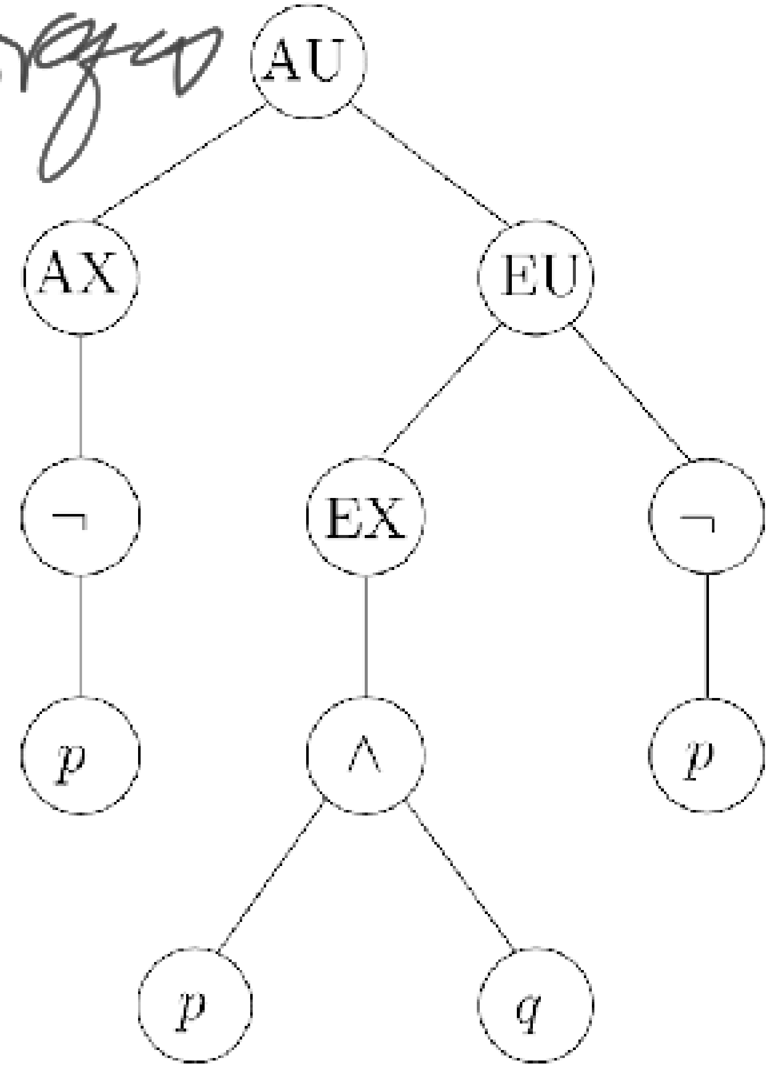


Figure 3.18. The parse ~~tree of~~ a CTL formula without infix notation.

Definition 3.14 A subformula of a CTL formula ϕ is any formula ψ whose parse tree is a subtree of ϕ's parse tree.

SEMANTICS OF CTL

Informally

CTL formulas are interpreted over transition systems (Definition 3.4). Let $\mathcal{M} = (S, \rightarrow, L)$ be such a model, $s \in S$ and ϕ a CTL formula. The definition of whether $\mathcal{M}, s \models \phi$ holds is recursive on the structure of ϕ, and can be roughly understood as follows:

- If ϕ is atomic, satisfaction is determined by L.
- If the top-level connective of ϕ (i.e., the connective occurring top-most in the parse tree of ϕ) is a boolean connective ($\wedge$, $\vee$, $\neg$, $\top$ etc.) then the satisfaction question is answered by the usual truth-table definition and further recursion down ϕ.
- If the top level connective is an operator beginning A, then satisfaction holds if all paths from s satisfy the 'LTL formula' resulting from removing the A symbol.
- Similarly, if the top level connective begins with E, then satisfaction holds if some path from s satisfy the 'LTL formula' resulting from removing the E.

In the last two cases, the result of removing A or E is not strictly an LTL formula, for it may contain further As or Es below. However, these will be dealt with by the recursion.

Formal definitions

Definition 3.15 Let $\mathcal{M} = (S, \rightarrow, L)$ be a model for CTL, s in S, ϕ a CTL formula. The relation $\mathcal{M}, s \vDash \phi$ is defined by structural induction on ϕ:

1. $\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \nvDash \bot$
2. $\mathcal{M}, s \vDash p$ iff $p \in L(s)$
3. $\mathcal{M}, s \vDash \neg\phi$ iff $\mathcal{M}, s \nvDash \phi$
4. $\mathcal{M}, s \vDash \phi_1 \wedge \phi_2$ iff $\mathcal{M}, s \vDash \phi_1$ and $\mathcal{M}, s \vDash \phi_2$
5. $\mathcal{M}, s \vDash \phi_1 \vee \phi_2$ iff $\mathcal{M}, s \vDash \phi_1$ or $\mathcal{M}, s \vDash \phi_2$
6. $\mathcal{M}, s \vDash \phi_1 \rightarrow \phi_2$ iff $\mathcal{M}, s \nvDash \phi_1$ or $\mathcal{M}, s \vDash \phi_2$.
7. $\mathcal{M}, s \vDash \mathrm{AX}\,\phi$ iff for all s_1 such that $s \rightarrow s_1$ we have $\mathcal{M}, s_1 \vDash \phi$. Thus, AX says: 'in every next state.'
8. $\mathcal{M}, s \vDash \mathrm{EX}\,\phi$ iff for some s_1 such that $s \rightarrow s_1$ we have $\mathcal{M}, s_1 \vDash \phi$. Thus, EX says: 'in some next state.' E is dual to A – in exactly the same way that $\exists$ is dual to $\forall$ in predicate logic.
9. $\mathcal{M}, s \vDash \mathrm{AG}\,\phi$ holds iff for all paths $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$, where s_1 equals s, and all s_i along the path, we have $\mathcal{M}, s_i \vDash \phi$. Mnemonically: for All computation paths beginning in s the property ϕ holds Globally. Note that 'along the path' includes the path's initial state s.
10. $\mathcal{M}, s \vDash \mathrm{EG}\,\phi$ holds iff there is a path $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$, where s_1 equals s, and for all s_i along the path, we have $\mathcal{M}, s_i \vDash \phi$. Mnemonically: there Exists a path beginning in s such that ϕ holds Globally along the path.

11. $\mathcal{M}, s \vDash \mathrm{AF}\,\phi$ holds iff for all paths $s_1 \to s_2 \to \ldots$, where s_1 equals s, there is some s_i such that $\mathcal{M}, s_i \vDash \phi$. Mnemonically: for All computation paths beginning in s there will be some Future state where ϕ holds.
12. $\mathcal{M}, s \vDash \mathrm{EF}\,\phi$ holds iff there is a path $s_1 \to s_2 \to s_3 \to \ldots$, where s_1 equals s, and for some s_i along the path, we have $\mathcal{M}, s_i \vDash \phi$. Mnemonically: there Exists a computation path beginning in s such that ϕ holds in some Future state;
13. $\mathcal{M}, s \vDash \mathrm{A}[\phi_1 \mathrel{\mathrm{U}} \phi_2]$ holds iff for all paths $s_1 \to s_2 \to s_3 \to \ldots$, where s_1 equals s, that path satisfies $\phi_1 \mathrel{\mathrm{U}} \phi_2$, i.e., there is some s_i along the path, such that $\mathcal{M}, s_i \vDash \phi_2$, and, for each $j < i$, we have $\mathcal{M}, s_j \vDash \phi_1$. Mnemonically: All computation paths beginning in s satisfy that ϕ_1 Until ϕ_2 holds on it.
14. $\mathcal{M}, s \vDash \mathrm{E}[\phi_1 \mathrel{\mathrm{U}} \phi_2]$ holds iff there is a path $s_1 \to s_2 \to s_3 \to \ldots$, where s_1 equals s, and that path satisfies $\phi_1 \mathrel{\mathrm{U}} \phi_2$ as specified in 13. Mnemonically: there Exists a computation path beginning in s such that ϕ_1 Until ϕ_2 holds on it.

$M, s \models EF\,\phi$

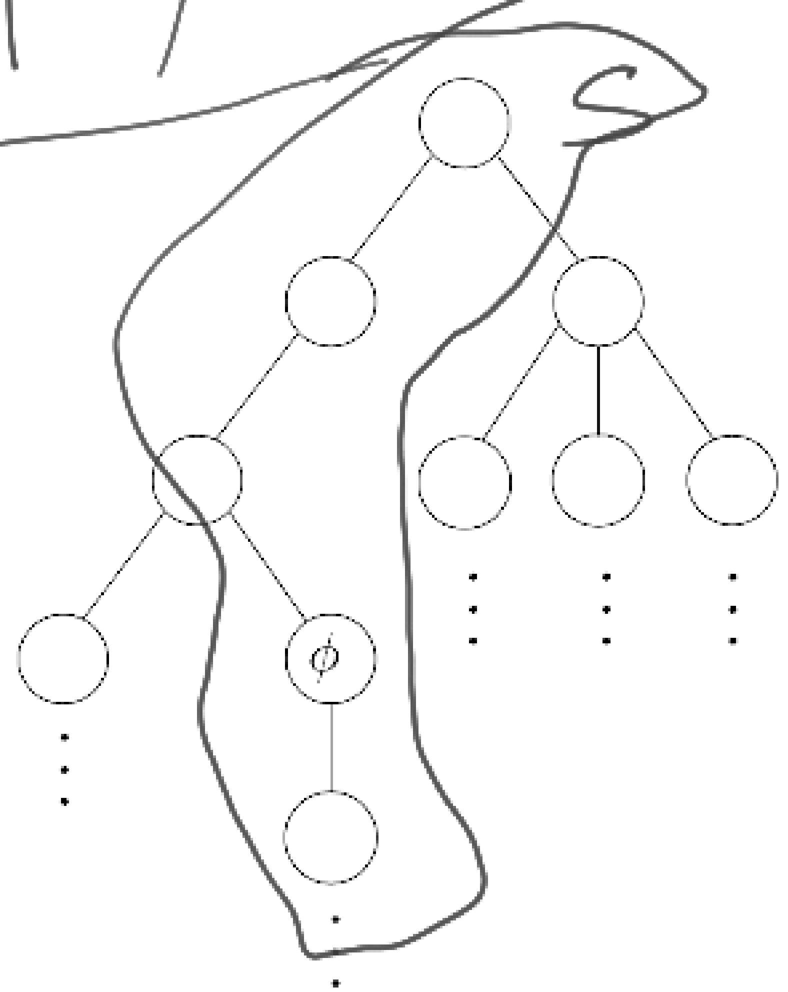
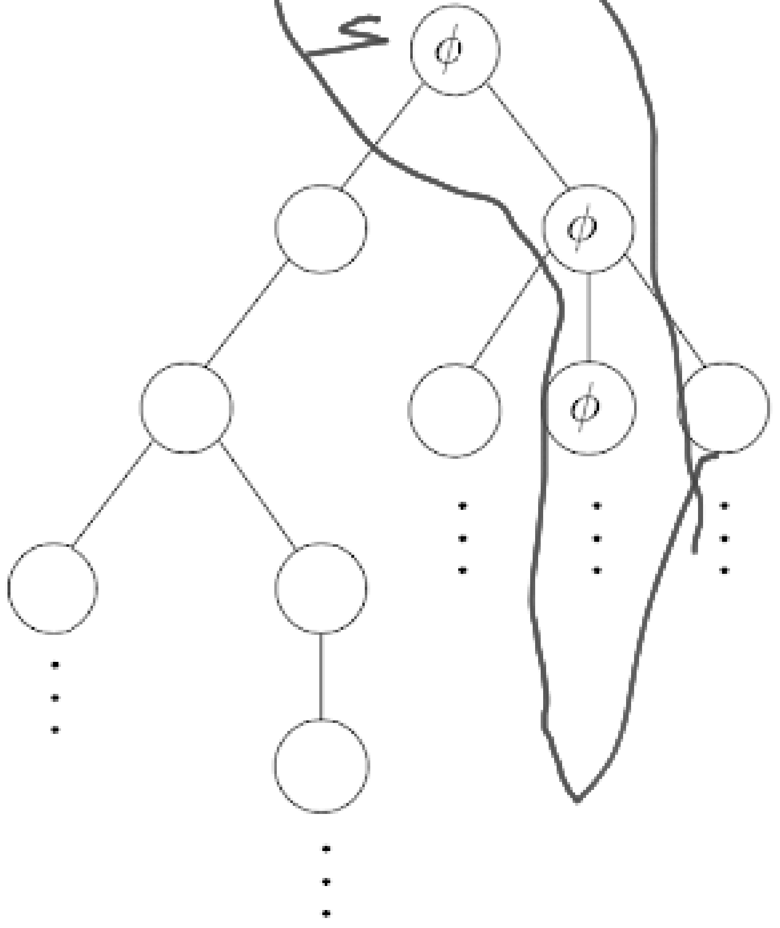
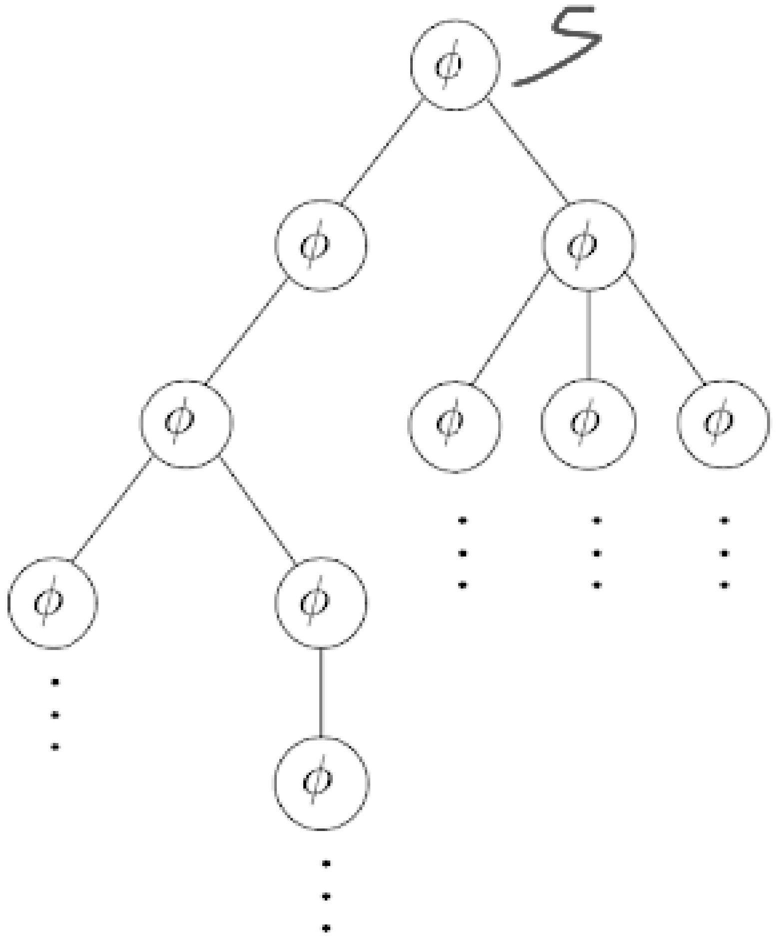


Figure 3.19. A system whose starting state satisfies EF ϕ.



$M, s \models EG\,\phi$

Figure 3.20. A system whose starting state satisfies $\mathrm{EG}\ \phi$.



$M, s \models AG\,\phi$

Figure 3.21. A system whose starting state satisfies $\mathrm{AG}\ \phi$.

EQUIVALENCES BET. CTL Formulas

Definition 3.16 Two CTL formulas ϕ and ψ are said to be semantically equivalent if any state in any model which satisfies one of them also satisfies the other; we denote this by $\phi \equiv \psi$.

A : universal quantifier over paths

E : existential "

G : Universal quantifier on states along a particular path

F : existential "

DE MORGAN'S RULES

$$\neg \mathrm{AF}\, \phi \equiv \mathrm{EG}\, \neg \phi$$

$$\neg \mathrm{EF}\, \phi \equiv \mathrm{AG}\, \neg \phi$$

$$\neg \mathrm{AX}\, \phi \equiv \mathrm{EX}\, \neg \phi.$$

$$\mathrm{AF}\,\phi \;\equiv\; \mathrm{A}[\top\ \mathrm{U}\ \phi] \qquad \mathrm{EF}\,\phi \;\equiv\; \mathrm{E}[\top\ \mathrm{U}\ \phi]$$

ADEQUATE SET OF CONNECTIVES IN CTL

Theorem 3.17 A set of temporal connectives in CTL is adequate if, and only if, it contains at least one of $\{\mathrm{AX}, \mathrm{EX}\}$, at least one of $\{\mathrm{EG}, \mathrm{AF}, \mathrm{AU}\}$ and EU.

EXPRESSIVE POWERS OF CTL & LTL

Things you can express in LTL but not in CTL

→ LTL formula ϕ: $Fp \rightarrow Fq$

$s \models \phi$

cannot write in CTL

What about $AFp \rightarrow AFq$?

Venn Diagram

(Expressivity of LTL & CTL)