# EE 144/244: Fundamental Algorithms for System Modeling, Analysis, and Optimization Fall 2014

## Temporal logic – CTL

Stavros Tripakis
University of California, Berkeley



## Linear-Time vs. Branching-Time Properties

So far we have been talking about properties of **linear** behaviors (sequences, traces).

But some properties are not linear, e.g.:

*"it is possible to recover from any fault"*

or

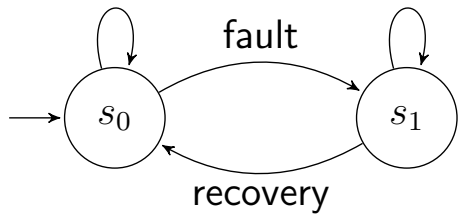
*"there exists a way to get back to the initial state from any reachable state"*

# Linear-Time vs. Branching-Time Properties

*"it is possible to recover from any fault"*

Based on *one* (linear) behavior alone,[1] we cannot conclude whether our system satisfies the property.

E.g., the following system satisfies the property, although it contains a behavior that stays forever in state $s_1$:




[1] if we had *all* linear behaviors of a system, we could in principle reconstruct its branching behavior as well



# Linear-Time vs. Branching-Time Temporal Logics

**Linear-time**: the "solutions" (models) of a temporal logic formula are infinite **sequences** (traces).

**Branching-time**: the "solutions" (models) of a temporal logic formula are infinite **trees**.

- Hence the name "Computation Tree Logic" for CTL.

# Branching-Time Temporal Logic: CTL

We will simplify and define the semantics of CTL directly on states of a transition system (Kripke structure).



# CTL (Computation Tree Logic) – Syntax

CTL formulas are defined by the following grammar:

$$
\begin{aligned}
\phi \quad ::= \quad & p \mid q \mid \ldots \\
& \mid \phi_1 \wedge \phi_2 \mid \neg\phi_1 \\
& \mid \mathsf{EG}\phi_1 \mid \mathsf{AG}\phi_1 \\
& \mid \mathsf{EF}\phi_1 \mid \mathsf{AF}\phi_1 \\
& \mid \mathsf{EX}\phi_1 \mid \mathsf{AX}\phi_1 \\
& \mid \mathsf{E}(\phi_1 \,\mathsf{U}\, \phi_2) \mid \mathsf{A}(\phi_1 \,\mathsf{U}\, \phi_2)
\end{aligned}
$$

E ("there exists a path") and A ("for all paths") are called *path quantifiers*.

# CTL (Computation Tree Logic) – Syntax

Examples of CTL formulas:

$$\mathsf{AG}p$$

$$\mathsf{EF}q$$

$$\mathsf{AGEF}(p \Rightarrow q)$$

Alternative notation: $\forall\Box p$, $\exists\Diamond q$, $\forall(p\,\mathsf{U}\,q)$, etc.



# CTL – Semantics: Intuition

Let $s$ be a state of the Kripke structure.
Then $s$ satisfies the CTL formula $\mathsf{EG}\phi$, written

$$s \models \mathsf{EG}\phi$$

iff **there exists** a trace $\sigma$ starting from $s$ and satisfying $\mathsf{G}\phi$, i.e.:

$$\exists \text{ trace } \sigma \text{ starting from } s : \sigma \models \mathsf{G}\phi$$

Note:

- The 2nd $\models$ is the LTL satisfaction relation.
- The 1st $\models$ refers to the CTL satisfaction relation.
- To be more pedantic:

$$s \models_{CTL} \mathsf{EG}\phi \qquad \text{iff} \qquad \exists \text{ trace } \sigma \text{ starting from } s \text{ s.t. } \sigma \models_{LTL} \mathsf{G}\phi$$

# CTL – Semantics: Intuition

$$s \models \mathsf{AG}\phi$$

iff **every** trace $\sigma$ starting from $s$ satisfies $\mathsf{G}\phi$:

$$\forall \text{ traces } \sigma \text{ starting from } s : \sigma \models \mathsf{G}\phi$$



# Examples

Let's construct transition systems (Kripke structures) satisfying or violating the following CTL formulas:

$$\mathsf{AG}p$$

$$\mathsf{AF}p$$

$$\mathsf{EG}p$$

$$\mathsf{EF}p$$

# Facts about CTL

**Quiz**: do we need EF$\phi$? Can we express it in terms of other CTL modalities?



# CTL – Formal Semantics

Let $(P, S, S_0, L, R)$ be a Kripke structure and let $s \in S$.

A *trace starting from* $s$ is an infinite sequence $\sigma = \sigma_0, \sigma_1, \cdots$, such that there is an infinite path $s = s_0, s_1, \cdots$ starting from $s$, and $\sigma_i = L(s_i)$ for all $i$.

Satisfaction relation for CTL:

| | | |
|---|---|---|
| $s \models p$ | iff | $p \in L(s)$ |
| $s \models \phi_1 \wedge \phi_2$ | iff | $s \models \phi_1$ and $s \models \phi_2$ |
| $s \models \neg\phi$ | iff | $s \not\models \phi$ |
| $s \models \mathsf{EG}\phi$ | iff | $\exists$trace $\sigma$ starting from $s : \sigma \models_{LTL} \mathsf{G}\phi$ |
| $s \models \mathsf{AG}\phi$ | iff | $\forall$traces $\sigma$ starting from $s : \sigma \models_{LTL} \mathsf{G}\phi$ |
| $s \models \mathsf{EX}\phi$ | iff | $\exists$trace $\sigma$ starting from $s : \sigma \models_{LTL} \mathsf{X}\phi$ |
| $s \models \mathsf{E}(\phi_1 \,\mathsf{U}\, \phi_2)$ | iff | $\exists$trace $\sigma$ starting from $s : \sigma \models_{LTL} \phi_1 \,\mathsf{U}\, \phi_2$ |

...

# CTL – Examples

How to express these properties in CTL?

*"$p$ holds at all reachable states"* AG$p$

*"there exists a way to get back to the initial state from any reachable state"* AG EF *init*

*"$p$ is inevitable"* AF $p$

*"$p$ is possible"* EF $p$

How would you express the last two in LTL?



# Bibliography


Baier, C. and Katoen, J.-P. (2008).
*Principles of Model Checking*.
MIT Press.

Clarke, E., Grumberg, O., and Peled, D. (2000).
*Model Checking*.
MIT Press.

Emerson, E. A. (1990).
Handbook of theoretical computer science (vol. b).
chapter Temporal and modal logic, pages 995–1072. MIT Press.

Huth, M. and Ryan, M. (2004).
*Logic in Computer Science: Modelling and Reasoning about Systems*.
Cambridge University Press.

Manna, Z. and Pnueli, A. (1991).
*The Temporal Logic of Reactive and Concurrent Systems: Specification*.
Springer-Verlag, New York.

Manna, Z. and Pnueli, A. (1995).
*Temporal Verification of Reactive Systems: Safety*.
Springer-Verlag, New York.

Pnueli, A. (1981).
A temporal logic of concurrent programs.
*Theoretical Computer Science*, 13:45–60.