Computation Tree Logic
Outline

• Motivation
• CTL structures
• Syntax of CTL
• Semantics of CTL
• Some examples
View Time as Branching

- Processes make *choices* as they execute
- Outcomes of choices cause different *futures*
- Branching-time logics allow quantification over the possible futures
View Time as Branching

Two types of operators

- Path operators:
  * $U$ (until)
  * $F$ (sometime)
  * $G$ (always)

- State operators:
  * $A$ (on all paths)
  * $E$ (on some path)
CTL Structures: FSM

• Assume a set of primitive propositions, \( P \)

• A finite state model \( M \) is a triple \( M = (S, R, L) \) where
  – \( S \) is a finite set of states
  – \( R \subseteq S \times S \) is a (total) transition relation
  – \( L : S \rightarrow 2^P \) labels each state with a set of propositions

• A path, \( \rho \), is an infinite sequence of states
  \[ \rho : s_0, s_1, s_2, \ldots \]
  such that \((s_i, s_{i+1}) \in R\), for \( i \geq 0 \)
CTL Structures: Computation Trees

• A finite state model $M = (S, R, L)$ and a state $s_0 \in S$ define an infinite *computation tree* $T$ where:
  – the root of $T$ is labeled $s_0$ and
  – $T$ contains an edge from a node labeled $s$ to a node labeled $t$ iff $(s, t) \in R$

• A CTL formula is evaluated on a computation tree, i.e., at a state in a FSM
Example: FSM

- $P$: \{x=0, x=1, y=0, y=1\}
- $S$: \{s_0, s_1, s_2\}
- $R$:
  \{(s_0, s_1), (s_1, s_0),
  (s_1, s_2), (s_2, s_0)\}
- $L$:
  $L(s_0) = \{x=0, y=0\}$
  $L(s_1) = \{x=1, y=0\}$
  $L(s_2) = \{x=1, y=1\}$
Example: Some paths

$s_0$, $s_1$, $s_0$, $s_1$, $s_0$, $s_1$, $s_0$, ...

$s_0$, $s_1$, $s_1$, $s_0$, $s_2$, $s_0$, $s_1$, $s_2$, ...

$s_0$, $s_1$, $s_0$, $s_1$, $s_2$, $s_0$, $s_1$, $s_2$, $s_0$, ...

$s_0$, $s_1$, $s_0$, $s_1$, $s_2$, $s_0$, $s_1$, $s_0$, $s_1$, ...

…
Example: Computation Trees
Syntax of CTL

The CTL formulas over $P$ are defined inductively:

If $p \in P$ and $f$ and $g$ are CTL formulas, then the following are CTL formulas:

(Propositions) $p$

(Boolean operators) $\neg f$ $f \land g$ $f \lor g$ ...

(Temporal operators) $AXf$ On all paths, next $f$

$EXf$ On some path, next $f$

$A[f \mathcal{U} g]$ On all paths, $f$ until $g$

$E[f \mathcal{U} g]$ On some path, $f$ until $g$
Semantics of CTL

• A CTL formula is evaluated at a state \( s_0 \in S \) in a finite state model \( M = (S, R, L) \).

• The models relation is defined inductively:

Proposition \( p \): \( (M, s_0) \vDash p \) if \( p \in L(s_0) \)

Boolean operators, as usual:

\[
(M, s_0) \vDash \neg f \quad \text{iff} \quad \neg ( (M, s_0) \vDash f ) \\
(M, s_0) \vDash f \land g \quad \text{iff} \quad ( (M, s_0) \vDash f ) \land (M, s_0) \vDash g \\
(M, s_0) \vDash f \lor g \quad \text{iff} \quad ( (M, s_0) \vDash f ) \lor (M, s_0) \vDash g \\
\ldots
\]
Semantics: CTL Temporal Operators

Always Next:

\((M, s_0) \models AXf\)

iff

\[ \forall t \ ( (s_0, t) \in R \Rightarrow (M, t) \models f ) \]


Sometime Next:

\((M, s_0) \models EXf\)

iff

\[ \exists t \ ( (s_0, t) \in R \land (M, t) \models f ) \]
Semantics: CTL Temporal Operators

Always Until:

\[(M, s_0) \models A[f \mathcal{U} g] \quad \text{iff} \]

for all paths \((s_0, s_1, s_2, \ldots)\),

\[\exists i \geq 0 \ [(M, s_i) \models g \land \forall j (0 \leq j < i \Rightarrow (M, s_j) \models f) ]\]
Semantics: CTL Temporal Operators

Always Until:

$$(M, s_0) \models E[f U g] \iff$$
for some path $(s_0, s_1, s_2, \ldots)$,
$$\exists i \geq 0 \ [(M, s_i) \models g \land \forall j \ (0 \leq j < i \ \Rightarrow \ (M, s_j) \models f)]$$
Semantics: CTL Temporal Operators

Inevitably: \( AF(f) \equiv A[True \ U \ f] \)
Semantics: CTL Temporal Operators

Potentially: $EF(f) \equiv E[True \ U f]$

$(M, s_0) \models EF(f)$ iff

for some path $(s_0, s_1, s_2, ...)$, $\exists i \geq 0 \ [ (M, s_i) \models f ]$
Semantics: CTL Temporal Operators

\[ EG(f) \equiv \neg AF(\neg f) \]

\((M, s_0) \models EG(f)\) iff

for some path \((s_0, s_1, s_2, \ldots)\), \(\forall i \geq 0\) \([(M, s_i) \models f]\)
Semantics: CTL Temporal Operators

Globally: $AG(f) \equiv \neg EF(\neg f)$

$(M, s_0) \models AG(f)$ iff for all path $(s_0, s_1, s_2, \ldots)$, $\forall i \geq 0 \ [(M, s_i) \models f]$
Example: Mutex Protocol

C – in critical section (CS);  N – not ready to enter CS;  T – trying to enter CS
Example: Mutex Protocol

\[(M, s_{init}) \models AG(\neg C1 \lor \neg C2) \]
Example: Mutex Protocol

\[(M, s_{init}) \models AF(C1) ??\]
Example: Mutex Protocol

\((M, s_{init}) \models AG(T1 \Rightarrow AF(C1))??\)
Summary

• CTL allows reasoning about possible futures of a state
• CTL formula is evaluated at a state in a FSM (or equivalently on an infinite computation tree)
• Combine a state op with a path op
  – State ops: $A, E$ – quantify over possible futures
  – Path ops: $X, F, G$ – quantify over states in a path