Computation Tree Logic (CTL)

Slides by Alessandro Artale
http://www.inf.unibz.it/~artale/

Some material (text, figures) displayed in these slides is courtesy of:
M. Benerecetti, A. Cimatti, M. Fisher, F. Giunchiglia, M. Pistore, M. Roveri, R.Sebastiani.

Summary

- Computation Tree Logic: Intuitions.
- CTL: Syntax and Semantics.
- CTL in Computer Science.
- CTL and Model Checking: Examples.
- CTL Vs. LTL.
- CTL*.

Computation Tree logic Vs. LTL

- LTL implicitly quantifies *universally* over paths.

  $\langle \mathcal{KM}, s\rangle \models \phi$ iff for every path π starting at s $\langle \mathcal{KM}, \pi\rangle \models \phi$

- Properties that assert the *existence* of a path cannot be expressed. In particular, properties which *mix* existential and universal path quantifiers cannot be expressed.
- The *Computation Tree Logic*, CTL, solves these problems!
  - CTL explicitly introduces *path quantifiers*!
  - CTL is the natural temporal logic interpreted over Branching Time Structures.

CTL at a glance

- CTL is evaluated over branching-time structures (Trees).
- CTL explicitly introduces *path quantifiers*:

  All Paths: **A**

  Exists a Path: **E**.
- Every temporal operator $-\Box(\mathbf{G}), \Diamond(\mathbf{F}), \bigcirc(\mathbf{X}), \mathcal{U}\,(\mathbf{U})-$ preceded by a path quantifier (**A** or **E**).
- **Universal modalities:** **AF, AG, AX, AU**
  The temporal formula is true in **all** the paths starting in the current state.
- **Existential modalities:** **EF, EG, EX, EU**
  The temporal formula is true in **some** path starting in the current state.

Summary

- Computation Tree Logic: Intuitions.
- CTL: Syntax and Semantics.
- CTL in Computer Science.
- CTL and Model Checking: Examples.
- CTL Vs. LTL.
- CTL*.

CTL: Syntax

Countable set Σ of *atomic propositions*: $p, q, \ldots$ the set FORM of formulas is:

$$\varphi, \psi \quad \rightarrow \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid$$

$$\mathbf{AX}\varphi \mid \mathbf{AG}\varphi \mid \mathbf{AF}\varphi \mid \varphi\mathbf{AU}\psi)$$

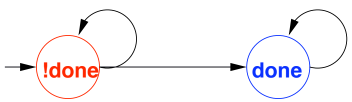
$$\mathbf{EX}\varphi \mid \mathbf{EG}\varphi \mid \mathbf{EF}\varphi \mid \varphi\mathbf{EU}\psi)$$

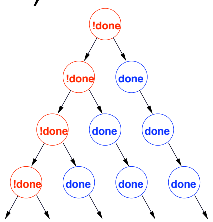
Intuition:

- E there Exists a path
- A in All paths
- F sometime in the Future
- G Globally in the future

CTL: Semantics

- We interpret our CTL temporal formulas over Kripke Models linearized as trees (e.g. $\mathbf{AF}\mathit{done}$).





- Universal modalities $(\mathbf{AF}, \mathbf{AG}, \mathbf{AX}, \mathbf{AU})$: the temporal formula is true in **all** the paths starting in the current state.
- Existential modalities $(\mathbf{EF}, \mathbf{EG}, \mathbf{EX}, \mathbf{EU})$: the temporal formula is true in **some** path starting in the current state.

CTL: Semantics (Cont.)

Let Σ be a set of atomic propositions. We interpret our CTL temporal formulas over Kripke Models:

$$\mathcal{KM} = \langle S, I, R, \Sigma, L \rangle$$

The semantics of a temporal formula is provided by the *satisfaction* relation:

$$\models : (\mathcal{KM} \times S \times \text{FORM}) \rightarrow \{\mathbf{true}, \mathbf{false}\}$$

CTL Semantics: The Propositional Aspect

We start by defining when an atomic proposition is true at a state/time "s_i"

$$\mathcal{KM}, s_i \models p \quad \textbf{iff} \quad p \in L(s_i) \qquad (\text{for } p \in \Sigma)$$

The semantics for the classical operators is as expected:

$$\mathcal{KM}, s_i \models \neg\varphi \quad \textbf{iff} \quad \mathcal{KM}, s_i \not\models \varphi$$

$$\mathcal{KM}, s_i \models \varphi \wedge \psi \quad \textbf{iff} \quad \mathcal{KM}, s_i \models \varphi \text{ and } \mathcal{KM}, s_i \models \psi$$

$$\mathcal{KM}, s_i \models \varphi \vee \psi \quad \textbf{iff} \quad \mathcal{KM}, s_i \models \varphi \text{ or } \mathcal{KM}, s_i \models \psi$$

$$\mathcal{KM}, s_i \models \varphi \Rightarrow \psi \quad \textbf{iff} \quad \text{if } \mathcal{KM}, s_i \models \varphi \text{ then } \mathcal{KM}, s_i \models \psi$$

$$\mathcal{KM}, s_i \models \top$$

$$\mathcal{KM}, s_i \not\models \bot$$

CTL Semantics: The Temporal Aspect

Temporal operators have the following semantics where $\pi = (s_i, s_{i+1}, \ldots)$ is a generic path outgoing from state s_i in $\mathcal{KM}$.

$$\mathcal{KM}, s_i \models \mathbf{AX}\varphi \quad \text{iff} \quad \forall \pi = (s_i, s_{i+1}, \ldots) \quad \mathcal{KM}, s_{i+1} \models \varphi$$

$$\mathcal{KM}, s_i \models \mathbf{EX}\varphi \quad \text{iff} \quad \exists \pi = (s_i, s_{i+1}, \ldots) \quad \mathcal{KM}, s_{i+1} \models \varphi$$

$$\mathcal{KM}, s_i \models \mathbf{AG}\varphi \quad \text{iff} \quad \forall \pi = (s_i, s_{i+1}, \ldots) \quad \forall j \geq i. \mathcal{KM}, s_j \models \varphi$$

$$\mathcal{KM}, s_i \models \mathbf{EG}\varphi \quad \text{iff} \quad \exists \pi = (s_i, s_{i+1}, \ldots) \quad \forall j \geq i. \mathcal{KM}, s_j \models \varphi$$

$$\mathcal{KM}, s_i \models \mathbf{AF}\varphi \quad \text{iff} \quad \forall \pi = (s_i, s_{i+1}, \ldots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \varphi$$

$$\mathcal{KM}, s_i \models \mathbf{EF}\varphi \quad \text{iff} \quad \exists \pi = (s_i, s_{i+1}, \ldots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \varphi$$

$$\mathcal{KM}, s_i \models (\varphi \mathbf{AU} \psi) \quad \text{iff} \quad \forall \pi = (s_i, s_{i+1}, \ldots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \psi \text{ and } \forall i \leq k < j : M, s_k \models \varphi$$

$$\mathcal{KM}, s_i \models \varphi \mathbf{EU} \psi) \quad \text{iff} \quad \exists \pi = (s_i, s_{i+1}, \ldots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \psi \text{ and } \forall i \leq k < j : \mathcal{KM}, s_k \models \varphi$$

CTL Semantics: Intuitions

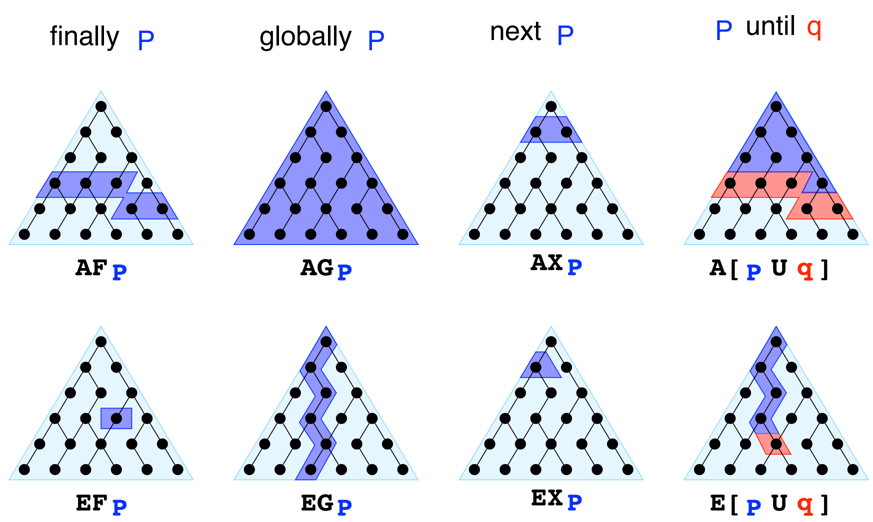
CTL is given by the standard boolean logic enhanced with temporal operators.

- "Necessarily Next". $\mathbf{AX}\varphi$ is true in s_t iff φ is true in every successor state s_{t+1}
- "Possibly Next". $\mathbf{EX}\varphi$ is true in s_t iff φ is true in one successor state s_{t+1}
- "Necessarily in the future" (or "Inevitably"). $\mathbf{AF}\varphi$ is true in s_t iff φ is inevitably true in **some** $s_{t'}$ with $t' \geq t$
- "Possibly in the future" (or "Possibly"). $\mathbf{EF}\varphi$ is true in s_t iff φ may be true in **some** $s_{t'}$ with $t' \geq t$

CTL Semantics: Intuitions (Cont.)

- "Globally" (or "always"). $\mathbf{AG}\varphi$ is true in s_t iff φ is true in **all** $s_{t'}$ with $t' \geq t$
- "Possibly henceforth". $\mathbf{EG}\varphi$ is true in s_t iff φ is possibly true henceforth
- "Necessarily Until". $(\varphi \mathbf{AU} \psi)$ is true in s_t iff necessarily φ holds until ψ holds.
- "Possibly Until". $(\varphi \mathbf{EU} \psi)$ is true in s_t iff possibly φ holds until ψ holds.

CTL Semantics: Intuitions (Cont.)



A Complete Set of CTL Operators

All CTL operators can be expressed via: $\mathbf{EX}, \mathbf{EG}, \mathbf{EU}$

- $\mathbf{AX}\varphi \equiv \neg\mathbf{EX}\neg\varphi$
- $\mathbf{AF}\varphi \equiv \neg\mathbf{EG}\neg\varphi$
- $\mathbf{EF}\varphi \equiv (\top\mathbf{EU}\varphi)$
- $\mathbf{AG}\varphi \equiv \neg\mathbf{EF}\neg\varphi \equiv \neg(\top\mathbf{EU}\neg\varphi)$
- $(\varphi\mathbf{AU}\psi) \equiv \neg\mathbf{EG}\neg\psi \wedge \neg(\neg\psi\mathbf{EU}(\neg\varphi \wedge \neg\psi))$

Summary

- Computation Tree Logic: Intuitions.
- CTL: Syntax and Semantics.
- CTL in Computer Science.
- CTL and Model Checking: Examples.
- CTL Vs. LTL.
- CTL*.

Safety Properties

Safety:

"something bad will not happen"

Typical examples:

$\mathbf{AG}\neg(reactor\_temp > 1000)$

$\mathbf{AG}\neg(one\_way \wedge \mathbf{AX}other\_way)$

$\mathbf{AG}\neg((x = 0) \wedge \mathbf{AXAXAX}(y = z/x))$

and so on.....

Usually: $\mathbf{AG}\neg$....

Liveness Properties

Liveness:

"something good will happen"

Typical examples:

$\mathbf{AF} rich$

$\mathbf{AF}(x > 5)$

$\mathbf{AG}(start \Rightarrow \mathbf{AF} terminate)$

and so on.....

Usually: $\mathbf{AF}\ldots$

Fairness Properties

Often only really useful when scheduling processes, responding to messages, etc.

Fairness:

"something is successful/allocated infinitely often"

Typical example:

$\mathbf{AG}(\mathbf{AF} enabled)$

Usually: $\mathbf{AGAF}\ldots$

Summary

- Computation Tree Logic: Intuitions.
- CTL: Syntax and Semantics.
- CTL in Computer Science.
- CTL and Model Checking: Examples.
- CTL Vs. LTL.
- CTL*.

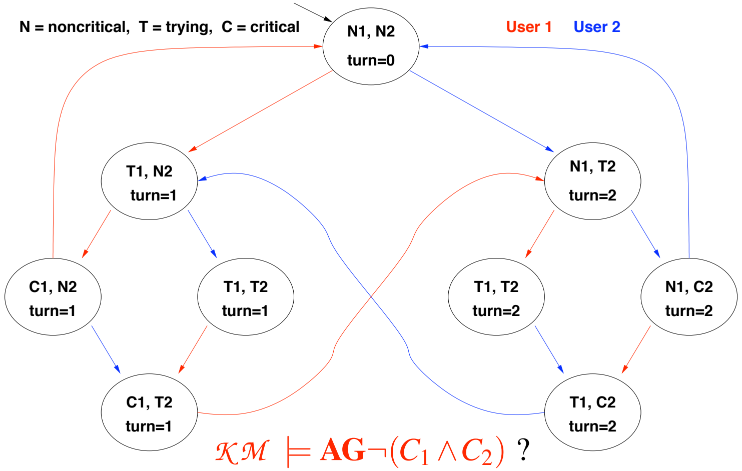
The CTL Model Checking Problem

The CTL Model Checking Problem is formulated as:

$$\mathcal{KM} \models \phi$$

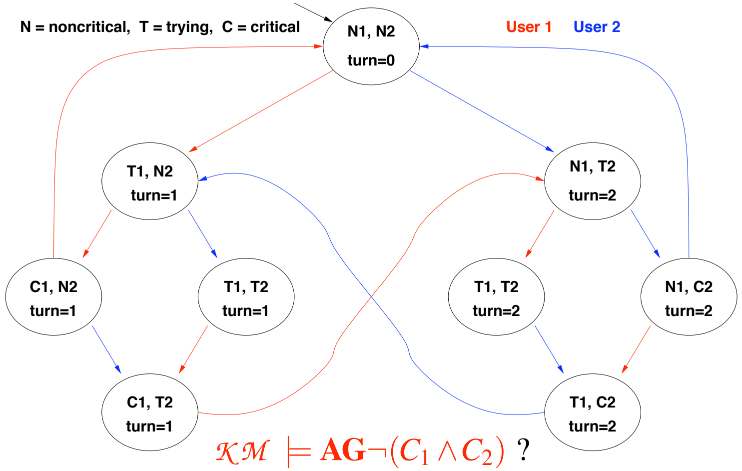
Check if $\mathcal{KM}, s_0 \models \phi$, for **every initial state**, s_0, of the Kripke structure $\mathcal{KM}$.

Example 1: Mutual Exclusion (Safety)



$\mathcal{KM} \models \mathbf{AG}\neg(C_1 \wedge C_2)$?

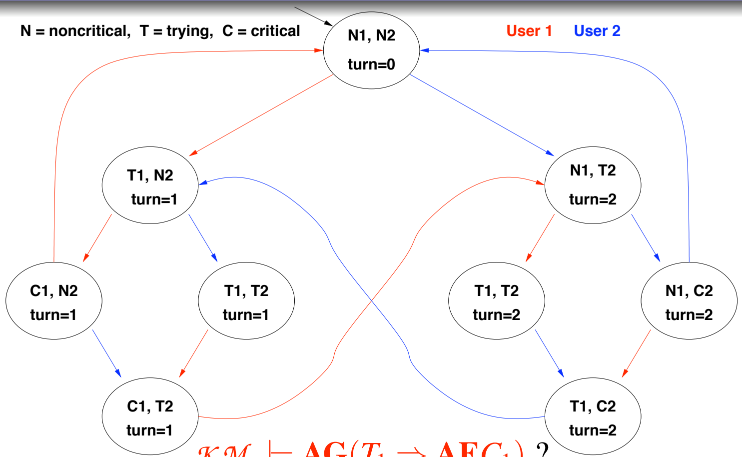
Example 1: Mutual Exclusion (Safety)



$\mathcal{KM} \models \mathbf{AG}\neg(C_1 \wedge C_2)$?

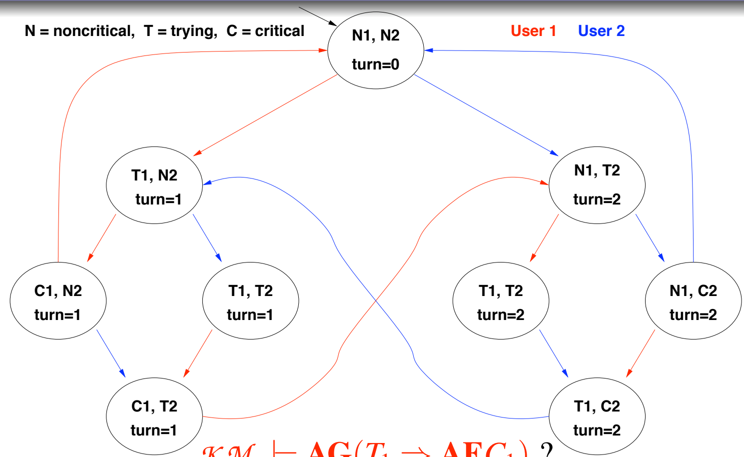
YES: There is no reachable state in which $(C_1 \wedge C_2)$ holds!
(Same as the $\Box\neg(C_1 \wedge C_2)$ in LTL.)

Example 2: Liveness



$\mathcal{KM} \models \mathbf{AG}(T_1 \Rightarrow \mathbf{AF}C_1)$?

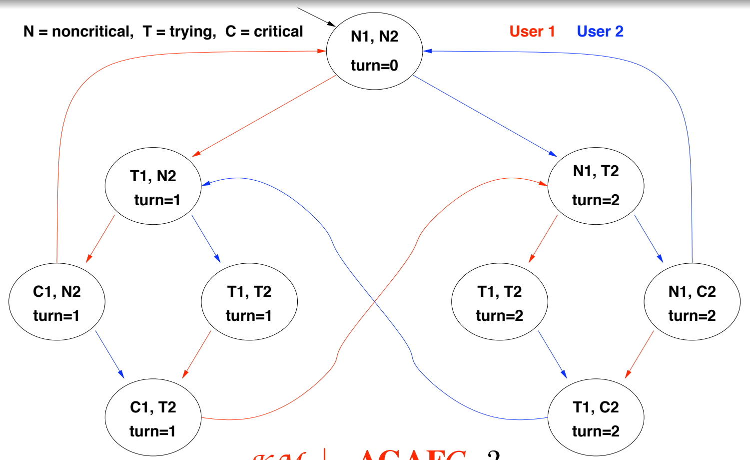
Example 2: Liveness



$\mathcal{KM} \models \mathbf{AG}(T_1 \Rightarrow \mathbf{AF}C_1)$?

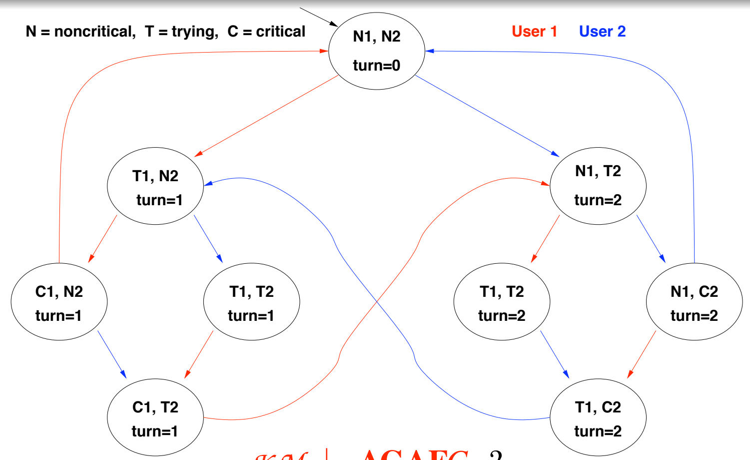
YES: every path starting from each state where T_1 holds passes through a state where C_1 holds.
(Same as $\Box(T_1 \Rightarrow \Diamond C_1)$ in LTL)

Example 3: Fairness



$\mathcal{KM} \models \mathbf{AGAF}C_1$?

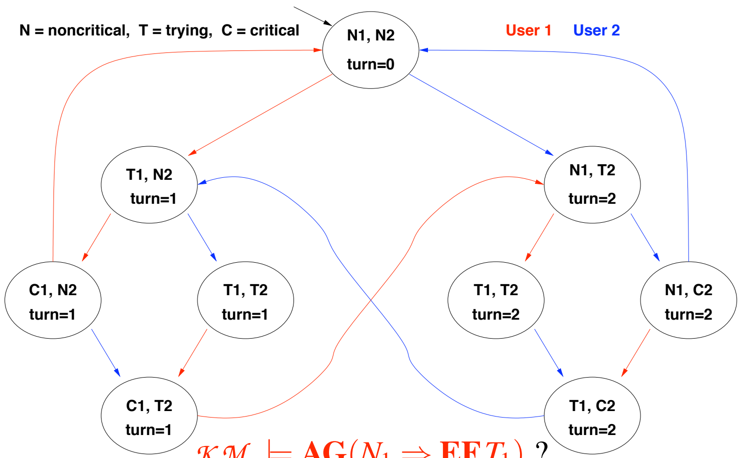
Example 3: Fairness



$\mathcal{KM} \models \mathbf{AGAF}C_1$?

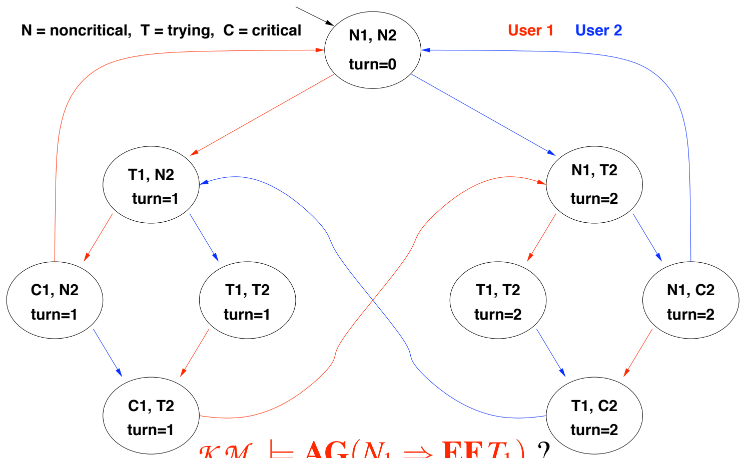
NO: e.g., in the initial state, there is the blue cyclic path in which C_1 never holds! (Same as $\Box\Diamond C_1$ in LTL)

Example 4: Non-Blocking



$\mathcal{KM} \models \mathbf{AG}(N_1 \Rightarrow \mathbf{EF}\,T_1)$?

Example 4: Non-Blocking



$\mathcal{KM} \models \mathbf{AG}(N_1 \Rightarrow \mathbf{EF}\,T_1)$?

YES: from each state where N_1 holds there is a path leading to a state where T_1 holds. (No corresponding LTL formulas)

Summary

- Computation Tree Logic: Intuitions.
- CTL: Syntax and Semantics.
- CTL in Computer Science.
- CTL and Model Checking: Examples.
- CTL Vs. LTL.
- CTL*.

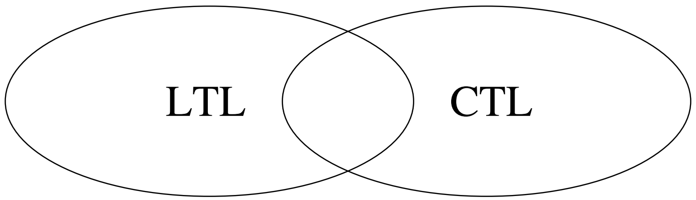
LTL Vs. CTL: Expressiveness

- Many CTL formulas cannot be expressed in LTL (e.g., those containing paths quantified existentially) E.g., $\mathbf{AG}(N_1 \Rightarrow \mathbf{EF}T_1)$
- Many LTL formulas cannot be expressed in CTL E.g., $\Box\Diamond T_1 \Rightarrow \Box\Diamond C_1$ (Strong Fairness in LTL) i.e, formulas that select a *range* of paths with a property ($\Diamond p \Rightarrow \Diamond q$ Vs. $\mathbf{AG}(p \Rightarrow \mathbf{AF}q)$)
- Some formluas can be expressed both in LTL and in CTL (typically LTL formulas with operators of nesting depth 1) E.g., $\Box\neg(C_1 \wedge C_2)$, $\Diamond C_1$, $\Box(T_1 \Rightarrow \Diamond C_1)$, $\Box\Diamond C_1$

LTL Vs. CTL: Expressiveness (Cont.)

CTL and LTL have incomparable expressive power.

The choice between LTL and CTL depends on the application and the personal preferences.



Summary

- Computation Tree Logic: Intuitions.
- CTL: Syntax and Semantics.
- CTL in Computer Science.
- CTL and Model Checking: Examples.
- CTL Vs. LTL.
- CTL*.

The Computation Tree Logic CTL*

- CTL* is a logic that combines the expressive power of LTL and CTL.
- Temporal operators can be applied without any constraints.
- $\mathbf{A}(\mathbf{X}\varphi \vee \mathbf{XX}\varphi)$.
  Along all paths, φ is true in the next state or the next two steps.
- $\mathbf{E}(\mathbf{GF}\varphi)$.
  There is a path along which φ is infinitely often true.

CTL*: Syntax

Countable set Σ of atomic propositions: $p, q, \ldots$ we distinguish between *States Formulas* (evaluated on states):

$$\varphi, \psi \rightarrow p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathbf{A}\alpha \mid \mathbf{E}\alpha$$

and *Path Formulas* (evaluated on paths):

$$\alpha, \beta \rightarrow \varphi \mid \neg\alpha \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \mathbf{X}\alpha \mid \mathbf{G}\alpha \mid \mathbf{F}\alpha \mid (\alpha \mathbf{U} \beta)$$

The set of CTL* formulas FORM is the set of state formulas.

CTL* Semantics: State Formulas

We start by defining when an atomic proposition is true at a state "s_0"

$$\mathcal{KM}, s_0 \models p \quad \textbf{iff} \quad p \in L(s_0) \qquad (\text{for } p \in \Sigma)$$

The semantics for *State Formulas* is the following where $\pi = (s_0, s_1, \ldots)$ is a generic path outgoing from state s_0:

$\mathcal{KM}, s_0 \models \neg\varphi$	**iff**	$\mathcal{KM}, s_0 \not\models \varphi$
$\mathcal{KM}, s_0 \models \varphi \wedge \psi$	**iff**	$\mathcal{KM}, s_0 \models \varphi$ and $\mathcal{KM}, s_0 \models \psi$
$\mathcal{KM}, s_0 \models \varphi \vee \psi$	**iff**	$\mathcal{KM}, s_0 \models \varphi$ or $\mathcal{KM}, s_0 \models \psi$
$\mathcal{KM}, s_0 \models \mathbf{E}\alpha$	**iff**	$\exists \pi = (s_0, s_1, \ldots)$ such that $\mathcal{KM}, \pi \models \alpha$
$\mathcal{KM}, s_0 \models \mathbf{A}\alpha$	**iff**	$\forall \pi = (s_0, s_1, \ldots)$ then $\mathcal{KM}, \pi \models \alpha$

CTL* Semantics: Path Formulas

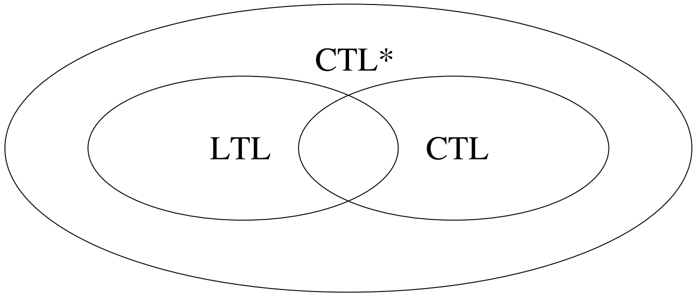
The semantics for *Path Formulas* is the following where $\pi = (s_0, s_1, \ldots)$ is a generic path outgoing from state s_0 and π^i denotes the suffix path $(s_i, s_{i+1}, \ldots)$:

$\mathcal{KM}, \pi \models \varphi$	**iff**	$\mathcal{KM}, s_0 \models \varphi$
$\mathcal{KM}, \pi \models \neg\alpha$	**iff**	$\mathcal{KM}, \pi \not\models \alpha$
$\mathcal{KM}, \pi \models \alpha \wedge \beta$	**iff**	$\mathcal{KM}, \pi \models \alpha$ and $\mathcal{KM}, \pi \models \beta$
$\mathcal{KM}, \pi \models \alpha \vee \beta$	**iff**	$\mathcal{KM}, \pi \models \alpha$ or $\mathcal{KM}, \pi \models \beta$
$\mathcal{KM}, \pi \models \mathbf{F}\alpha$	**iff**	$\exists i \geq 0$ such that $\mathcal{KM}, \pi^i \models \alpha$
$\mathcal{KM}, \pi \models \mathbf{G}\alpha$	**iff**	$\forall i \geq 0$ then $\mathcal{KM}, \pi^i \models \alpha$
$\mathcal{KM}, \pi \models \mathbf{X}\alpha$	**iff**	$\mathcal{KM}, \pi^1 \models \alpha$
$\mathcal{KM}, \pi \models \alpha \mathbf{U} \beta$	**iff**	$\exists i \geq 0$ such that $\mathcal{KM}, \pi^i \models \beta$ and $\forall j.(0 \leq j \leq i)$ then $\mathcal{KM}, \pi^j \models \alpha$

CTLs Vs LTL Vs CTL: Expressiveness

CTL* subsumes both CTL and LTL

- φ in CTL $\Longrightarrow$ φ in CTL* (e.g., $\mathbf{AG}(N_1 \Rightarrow \mathbf{EF}T_1)$)
- φ in LTL $\Longrightarrow$ $\mathbf{A}\varphi$ in CTL* (e.g., $\mathbf{A}(\mathbf{GF}T_1 \Rightarrow \mathbf{GF}C_1)$)
- LTL $\cup$ CTL $\subset$ CTL* (e.g., $\mathbf{E}(\mathbf{GF}p \Rightarrow \mathbf{GF}q)$)



CTL* Vs LTL Vs CTL: Complexity

The following Table shows the Computational Complexity of checking *Satisbiability*

Logic	Complexity
LTL	PSpace-Complete
CTL	ExpTime-Complete
CTL*	2ExpTime-Complete

CTL* Vs LTL Vs CTL: Complexity (Cont.)

The following Table shows the Computational Complexity of *Model Checking* (M.C.)

- Since M.C. has 2 inputs – the model, $\mathcal{M}$, and the formula, φ – we give two complexity measures.

Logic	Complexity w.r.t. $\lvert\varphi\rvert$	Complexity w.r.t. $\lvert\mathcal{M}\rvert$
LTL	PSpace-Complete	P (linear)
CTL	P-Complete	P (linear)
CTL*	PSpace-Complete	P (linear)