Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

October 19, 2015

In previous chapters...

Temporal Logic

Motivation and Intuition

Motivation

Intuition

How to communicate

Syntax of CTL

Semanthics of CTL

In previous chapters...

- Temporal Logic

Motivation and Intuition

- Motivation
- Intuition

How to communicate

- Syntax of CTL
- Semanthics of CTL

Previously on Temporal Logic Week...

Temporal Logic

Motivation

Needing of uncertainty;

Motivation

Needing of uncertainty;

Different paths of the future;

Intuition

In Computation Tree Logic (CTL) the model of time is a tree-like structure. This way, we cannot use Linear Temporal Logic (LTL) to express the existence of a certain path of time in which some event occurs.

In previous chapters...

Temporal Logic

Motivation and Intuition

Motivation

Intuition

How to communicate

Syntax of CTL

Semanthics of CTL

Syntax

Definition

The syntax of CTL consists on the syntax of temporal logic plus some path operators. The class of formulas can be defined in Backus-Naur form. If ϕ is a formula:

Syntax

Definition

The syntax of CTL consists on the syntax of temporal logic plus some path operators. The class of formulas can be defined in Backus-Naur form. If ϕ is a formula:

$$\phi ::= \bot \mid \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \mathtt{AX}\phi \mid \mathtt{EX}\phi \mid$$

$$\mathtt{AF}\phi \mid \mathtt{EF}\phi \mid \mathtt{AG}\phi \mid \mathtt{EG}\phi \mid \mathtt{A}[\phi \mathtt{U} \phi] \mid \mathtt{E}[\phi \mathtt{U} \phi]$$

Syntax
Definition

The syntax of CTL consists on the syntax of temporal logic plus some path operators. The class of formulas can be defined in Backus-Naur form. If ϕ is a formula:

$$\phi ::= \bot \mid \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \mathtt{AX}\phi \mid \mathtt{EX}\phi \mid$$

$$\mathtt{AF}\phi \mid \mathtt{EF}\phi \mid \mathtt{AG}\phi \mid \mathtt{EG}\phi \mid \mathtt{A}[\phi \mathtt{U} \phi] \mid \mathtt{E}[\phi \mathtt{U} \phi]$$

With p as a literal (atomic formula), $\mathtt{AX}$, $\mathtt{EX}$, $\mathtt{AF}$, $\mathtt{EF}$, $\mathtt{AG}$ e $\mathtt{EG}$ unary operators.

Syntax

Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

Syntax

Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

The temporal operators can be read (if φ is a formula) as follows:

Syntax
Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

The temporal operators can be read (if φ is a formula) as follows:

- $\mathtt{A}\phi$: φ is true in all possible paths;

Syntax
Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.
The temporal operators can be read (if φ is a formula) as follows:

- $\mathtt{A}\phi$: φ is true in all possible paths;
- $\mathtt{E}\phi$: φ exists a path in which ϕ is true;

Syntax

Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

The temporal operators can be read (if φ is a formula) as follows:

- $\mathtt{A}\phi$: φ is true in all possible paths;
- $\mathtt{E}\phi$: φ exists a path in which ϕ is true;

The path-specific operators can be read, considering φ and ψ formulas, as:

Syntax

Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

The temporal operators can be read (if φ is a formula) as follows:

- $\mathtt{A}\phi$: φ is true in all possible paths;
- $\mathtt{E}\phi$: φ exists a path in which ϕ is true;

The path-specific operators can be read, considering φ and ψ formulas, as:

- $\mathtt{X}\phi$: φ is true until next state;

Syntax

Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

The temporal operators can be read (if φ is a formula) as follows:

- $\mathtt{A}\phi$: φ is true in all possible paths;
- $\mathtt{E}\phi$: φ exists a path in which ϕ is true;

The path-specific operators can be read, considering φ and ψ formulas, as:

- $\mathtt{X}\phi$: φ is true until next state;
- $\mathtt{F}\varphi$: There is some state in the future where φ is true;

Syntax

Intuition

The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.

The temporal operators can be read (if φ is a formula) as follows:

- $\mathtt{A}\phi$: φ is true in all possible paths;
- $\mathtt{E}\phi$: φ exists a path in which ϕ is true;

The path-specific operators can be read, considering φ and ψ formulas, as:

- $\mathtt{X}\phi$: φ is true until next state;
- $\mathtt{F}\varphi$: There is some state in the future where φ is true;
- $\mathtt{G}\varphi$: Globally (in all future states) φ is true;

Syntax

Intuition

- The propositional operators: $\neg, \vee, \wedge, \rightarrow$ have the same meaning of in the propositional logic.
- The temporal operators can be read (if φ is a formula) as follows:
  - $\mathtt{A}\phi$: φ is true in all possible paths;
  - $\mathtt{E}\phi$: φ exists a path in which ϕ is true;
- The path-specific operators can be read, considering φ and ψ formulas, as:
  - $\mathtt{X}\phi$: φ is true until next state;
  - $\mathtt{F}\varphi$: There is some state in the future where φ is true;
  - $\mathtt{G}\varphi$: Globally (in all future states) φ is true;
  - $\varphi\mathtt{U}\psi$: φ is true at least until ψ becomes true;

Syntax
Notes

Notice that, in CTL, the combination of path specific operators and temporal operators are atomic, i.e., $\mathtt{AF}$ is a operator that can be read as "In all paths in the future there is some state where..."

Notice as well that the binary operators $\mathtt{A}[\varphi \mathtt{U} \psi]$ and $\mathtt{E}[\varphi \mathtt{U} \psi]$ can be represented as $\mathtt{AU}$

We assume that, similarly to the $\neg$ operator, the "new" unary operators ($\mathtt{AX}, \mathtt{EX}, \mathtt{AF}, \mathtt{EF}, \mathtt{AG}$, and $\mathtt{EG}$) have the first precedence. Next comes the $\wedge$ and $\vee$ operators. And at last the $\rightarrow$, $\mathtt{AU}$ and $\mathtt{EU}$

Examples

Examples of well-formed formulas:

$\mathtt{AG}(p \lor \mathtt{EF}q)$

Examples

Examples of well-formed formulas:

$$\mathtt{AG}(p \vee \mathtt{EF}q)$$

$$\mathtt{AX}(q \rightarrow \mathtt{E}[(p \vee q)\mathtt{U}r])$$

Examples

Examples of well-formed formulas:

$\mathtt{AG}(p \vee \mathtt{EF}q)$

$\mathtt{AX}(q \rightarrow \mathtt{E}[(p \vee q)\mathtt{U}r])$

$\mathtt{EFEG}p \rightarrow \mathtt{AF}r$ Note that this is binded as $(\mathtt{EFEG}p) \rightarrow \mathtt{AF}r$, not as $\mathtt{EFEG}(p \rightarrow \mathtt{AF}r)$

Examples

Examples of well-formed formulas:

$\mathtt{AG}(p \vee \mathtt{EF}q)$

$\mathtt{AX}(q \rightarrow \mathtt{E}[(p \vee q)\mathtt{U}r])$

$\mathtt{EFEG}p \rightarrow \mathtt{AF}r$ Note that this is binded as $(\mathtt{EFEG}p) \rightarrow \mathtt{AF}r$, not as $\mathtt{EFEG}(p \rightarrow \mathtt{AF}r)$

Example of formulas that are not well-formed:

$\mathtt{A}\neg\mathtt{G}\neg p$

Examples

Examples of well-formed formulas:

$\mathtt{AG}(p \vee \mathtt{EF}q)$

$\mathtt{AX}(q \rightarrow \mathtt{E}[(p \vee q)\mathtt{U}r])$

$\mathtt{EFEG}p \rightarrow \mathtt{AF}r$ Note that this is binded as $(\mathtt{EFEG}p) \rightarrow \mathtt{AF}r$, not as $\mathtt{EFEG}(p \rightarrow \mathtt{AF}r)$

Example of formulas that are not well-formed:

$\mathtt{A}\neg\mathtt{G}\neg p$

$F[p\mathtt{U}s]$

Examples

Examples of well-formed formulas:

$\mathtt{AG}(p \vee \mathtt{EF}q)$

$\mathtt{AX}(q \rightarrow \mathtt{E}[(p \vee q)\mathtt{U}r])$

$\mathtt{EFEG}p \rightarrow \mathtt{AF}r$ Note that this is binded as $(\mathtt{EFEG}p) \rightarrow \mathtt{AF}r$, not as $\mathtt{EFEG}(p \rightarrow \mathtt{AF}r)$

Example of formulas that are not well-formed:

$\mathtt{A}\neg\mathtt{G}\neg p$

$F[p\mathtt{U}s]$

$A[p\mathtt{U}s \wedge q\mathtt{U}s]$

Semantics

Definition of model

Different from usual logics, CTL formulas are interpreted by a transition system. Given an set of atoms:

Different from usual logics, CTL formulas are interpreted by a transition system. Given an set of atoms:

Definition (1)

A transition system $\mathcal{M}$ is a triple $\mathcal{M} = (S, \rightarrow, L)$ in which S is a set of states, $\rightarrow$ is a binary relation over S ($\rightarrow \subseteq S \times S$) and $L : S \rightarrow \mathcal{P}(Atoms)$ is a labelling function.

Semantics

Definition of model

Different from usual logics, CTL formulas are interpreted by a transition system. Given an set of atoms:

Definition (1)

A transition system $\mathcal{M}$ is a triple $\mathcal{M} = (S, \rightarrow, L)$ in which S is a set of states, $\rightarrow$ is a binary relation over S ($\rightarrow \subseteq S \times S$) and $L : S \rightarrow \mathcal{P}(Atoms)$ is a labelling function.

Definition (2)

A model is a duple $\mathcal{M}, s$ in which $\mathcal{M}$ is a transition system and s is a state.

Semantics

Definition of model

Different from usual logics, CTL formulas are interpreted by a transition system. Given an set of atoms:

Definition (1)

A transition system $\mathcal{M}$ is a triple $\mathcal{M} = (S, \rightarrow, L)$ in which S is a set of states, $\rightarrow$ is a binary relation over S ($\rightarrow \subseteq S \times S$) and $L : S \rightarrow \mathcal{P}(Atoms)$ is a labelling function.

Definition (2)

A model is a duple $\mathcal{M}, s$ in which $\mathcal{M}$ is a transition system and s is a state.

Notation: we will use $\mathcal{M}, s \vDash \varphi$ to say that the model $\mathcal{M}, s$ satisfies the formula φ

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

$\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \nvDash \bot$ for all $s \in S$

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

$\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \nvDash \bot$ for all $s \in S$

$\mathcal{M}, s \vDash p$ iff $p \in L(S)$

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

$\mathcal{M}, s \models \top$ and $\mathcal{M}, s \not\models \bot$ for all $s \in S$

$\mathcal{M}, s \models p$ iff $p \in L(S)$

$\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

$\mathcal{M}, s \models \top$ and $\mathcal{M}, s \not\models \bot$ for all $s \in S$

$\mathcal{M}, s \models p$ iff $p \in L(S)$

$\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$

$\mathcal{M}, s \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{M}, s \models \varphi_1$ AND $\mathcal{M}, s \models \varphi_2$

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

$\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \nvDash \bot$ for all $s \in S$

$\mathcal{M}, s \vDash p$ iff $p \in L(S)$

$\mathcal{M}, s \vDash \neg\varphi$ iff $\mathcal{M}, s \nvDash \varphi$

$\mathcal{M}, s \vDash \varphi_1 \wedge \varphi_2$ iff $\mathcal{M}, s \vDash \varphi_1$ AND $\mathcal{M}, s \vDash \varphi_2$

$\mathcal{M}, s \vDash \varphi_1 \vee \varphi_2$ iff $\mathcal{M}, s \vDash \varphi_1$ OR $\mathcal{M}, s \vDash \varphi_2$

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

$\mathcal{M}, s \models \top$ and $\mathcal{M}, s \not\models \bot$ for all $s \in S$

$\mathcal{M}, s \models p$ iff $p \in L(S)$

$\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$

$\mathcal{M}, s \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{M}, s \models \varphi_1$ AND $\mathcal{M}, s \models \varphi_2$

$\mathcal{M}, s \models \varphi_1 \vee \varphi_2$ iff $\mathcal{M}, s \models \varphi_1$ OR $\mathcal{M}, s \models \varphi_2$

$\mathcal{M}, s \models \varphi_1 \rightarrow \varphi_2$ iff $\mathcal{M}, s \not\models \varphi_1$ OR $\mathcal{M}, s \models \varphi_2$

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

> $\mathcal{M}, s \vDash \mathtt{AX}\varphi$ iff for all s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \vDash \varphi$. Thus, $\mathtt{AX}$ says: "in every next state..."

Semantics

Satisfaction

Take an arbitrary model $\mathcal{M}$. Let s, s_1, s_2, s_3 be states in S. Let $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as follows:

> $\mathcal{M}, s \models \mathtt{AX}\varphi$ iff for all s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \models \varphi$. Thus, `AX` says: "in every next state..."
>
> $\mathcal{M}, s \models \mathtt{EX}\varphi$ iff exists s_1 that $s \to s_1$ and $M, s_1 \models \varphi$. This, `EX` says: "in some next state..."
>
> $M, s, \models \mathtt{AG}\varphi$ iff for all paths $s_1 \to s_2 \to s_3 \to ...$