Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters... Motivation an Intuition

How to communicate Syntax of CT Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving ou language

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

October 23, 2015

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation and Intuition

Intuition
How to
communicate
Syntax of CT
Semantics of
CTL

Some examples of what we can say

More about semantics

Equivalences

Improving our language

- 1 In previous chapters...
- 2 Motivation and Intuition
- 3 How to communicate
 - Syntax of CTL
 - Semantics of CTL
- 4 Some examples of what we can say
- 5 More about semantics
 - Equivalences
- 6 Improving our language

Previously on Temporal Logic Week... Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Motivation and Intuition

How to communicate

Syntax of CTL

Semantics of

Semantics of CTL Some examples of what we can say

More about semantics

Improving our language A brief introduction to Propositional Logic, its syntax and its semantics

Previously on Temporal Logic Week... Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Motivation and Intuition

How to communicate Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving our

- A brief introduction to Propositional Logic, its syntax and its semantics
- Formal models of time

Previously on Temporal Logic Week... Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

How to communicate Syntax of CTL Semantics of CTL

Some examples of what we can say

More about

semantics Equivalences Improving our

- A brief introduction to Propositional Logic, its syntax and its semantics
- Formal models of time
 - Frames and Flows of time

Previously on Temporal Logic Week... Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

what we can sav

■ A brief introduction to Propositional Logic, its syntax and its semantics

- Formal models of time
 - Frames and Flows of time
- Temporal Logic extends the Propositional Logic
 - \blacksquare The connectives H and G

Previously on Temporal Logic Week... Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Motivation and Intuition How to communicate Syntax of CTL Semantics of

Some examples of what we can say More about semantics

Improving our language

- A brief introduction to Propositional Logic, its syntax and its semantics
- Formal models of time
 - Frames and Flows of time
- Temporal Logic extends the Propositional Logic
 - \blacksquare The connectives H and G
- Some practical applications

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Intuition
How to
communicate
Syntax of CTL
Semantics of
CTL

Some examples o what we can say

More about semantics Equivalences

Improving our language

■ "What good is Temporal Logic?"

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

How to communicate Syntax of CTL Semantics of

Semantics of CTL Some examples of what we can say

More about semantics Equivalences

- "What good is Temporal Logic?"
 - Answer: "Temporal Logic is a good method for specifying and reasoning about a concurrent program".

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

How to communicate Syntax of CTL Semantics of

What we can say
More about
semantics
Equivalences

- "What good is Temporal Logic?"
 - Answer: "Temporal Logic is a good method for specifying and reasoning about a concurrent program".
- Some mathematical definitions about states of a program and (concurrent) programs

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

How to communicate Syntax of CTL Semantics of

what we can say
More about
semantics
Equivalences

■ "What good is Temporal Logic?"

- Answer: "Temporal Logic is a good method for specifying and reasoning about a concurrent program".
- Some mathematical definitions about states of a program and (concurrent) programs
- The Peterson's algorithm

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

How to communicate Syntax of CTL Semantics of

what we can say More about semantics Equivalences

- "What good is Temporal Logic?"
 - Answer: "Temporal Logic is a good method for specifying and reasoning about a concurrent program".
- Some mathematical definitions about states of a program and (concurrent) programs
- The Peterson's algorithm
- How the model checking works?

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

How to communicate Syntax of CTL Semantics of

what we can say More about semantics Equivalences

- "What good is Temporal Logic?"
 - Answer: "Temporal Logic is a good method for specifying and reasoning about a concurrent program".
- Some mathematical definitions about states of a program and (concurrent) programs
- The Peterson's algorithm
- How the model checking works?
- Some strengths and weaknesses of model checking

Previously on Temporal Logic Week... Linear Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Motivation and Intuition

How to communicate

Syntax of CTL Semantics of

Some examples of what we can say

More about semantics

Improving our language

■ Syntax and Semantics of LTL

Previously on Temporal Logic Week... Linear Temporal Logic

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Intuition
How to communicate
Syntax of CTI
Semantics of CTI

Some examples of what we can say More about semantics

Equivalences Improving our

- Syntax and Semantics of LTL
- \blacksquare ω -languages, Kripke structures, paths and traces
- Buchi automata and LTL model checking

Motivation

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous

Motivation and

How to communicate Syntax of CTI

Semantics of CTL Some examples of

What we can say
More about
semantics

Equivalences

Improving our

■ Needing of expressing uncertainty;

Motivation

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous

Motivation and

Intuition a

How to communicate Syntax of CT Semantics of CTL

Some examples of what we can sav

More about semantics

Improving our language

- Needing of expressing uncertainty;
- Different paths of the future;

Intuition

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition

Semantics of CTL

what we can sav

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.

Syntax Definition

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Syntax of CTL Semantics of CTL

what we can sav

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters... Motivation an

Intuition

How to commun

Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics

Improving ou language 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:

$$\phi ::= \bot \mid \top \mid p \mid \neg \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \to \phi \mid AX\phi \mid EX\phi \mid$$
$$AF\phi \mid EF\phi \mid AG\phi \mid EG\phi \mid A[\phi U\phi] \mid E[\phi U\phi]$$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation an

How to

Syntax of CTL

Semantics of CTL

Some examples o what we can say

semantics Equivalences

Improving ou language 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:

$$\phi ::= \bot \mid \top \mid p \mid \neg \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \to \phi \mid AX\phi \mid EX\phi \mid$$
$$AF\phi \mid EF\phi \mid AG\phi \mid EG\phi \mid A[\phi U\phi] \mid E[\phi U\phi]$$

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Syntax of CTL

Semantics of CTL

what we can sav

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and

Syntax of CTL Semantics of CTL

what we can sav

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

The path-specific operators can be read as:

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and

Syntax of CTL

Semantics of CTL what we can sav

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

The path-specific operators can be read as:

■ A: is the universal quantifier over paths. Read as: "in all possible paths";

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

How to

Syntax of CTL Semantics of CTL

CTL
Some examples o

More about semantics Equivalences

Improving our

The propositional operators: \neg , \lor , \land , \rightarrow have the same meaning of in the propositional logic.

The path-specific operators can be read as:

- A: is the universal quantifier over paths. Read as: "in all possible paths";
- *E*:is the existential quantifier over paths. Read as: "exists a path in which";

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

How to communicate

Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving our language The propositional operators: \neg , \lor , \land , \rightarrow have the same meaning of in the propositional logic.

The path-specific operators can be read as:

- A: is the universal quantifier over paths. Read as: "in all possible paths";
- *E*:is the existential quantifier over paths. Read as: "exists a path in which";

The temporal operators, as in LTL, can be read as:

■ X: "in the next state";

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

communicate
Syntax of CTL

Syntax of CTL Semantics of CTL

More about semantics

Equivalences

The propositional operators: \neg , \lor , \land , \rightarrow have the same meaning of in the propositional logic.

The path-specific operators can be read as:

- A: is the universal quantifier over paths. Read as: "in all possible paths";
- *E*:is the existential quantifier over paths. Read as: "exists a path in which";

The temporal operators, as in LTL, can be read as:

- X: "in the next state";
- \blacksquare F "There is some state in the future (eventually)";

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Syntax of CTL

Semantics of CTL

what we can sav

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

The path-specific operators can be read as:

- A: is the universal quantifier over paths. Read as: "in all possible paths";
- E:is the existential quantifier over paths. Read as: "exists a path in which";

The temporal operators, as in LTL, can be read as:

- X: "in the next state";
- F "There is some state in the future (eventually)";
- G "Globally (in all future states)";

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

Syntax of CTL

Semantics of CTL

More about semantics

Equivalences

The propositional operators: \neg , \lor , \land , \rightarrow have the same meaning of in the propositional logic.

The path-specific operators can be read as:

- A: is the universal quantifier over paths. Read as: "in all possible paths";
- *E*:is the existential quantifier over paths. Read as: "exists a path in which";

The temporal operators, as in LTL, can be read as:

- X: "in the next state";
- \blacksquare F "There is some state in the future (eventually)";
- *G* "Globally (in all future states)";
- $\blacksquare \varphi U\psi$: φ is true at least until ψ becomes true;

Syntax Notes

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

How to

Syntax of CTL Semantics of CTL

CTL Some examples of what we can sav

More about semantics

Improving or

■ Notice that, in CTL, the combination of path specific operators and temporal operators are atomic, e.g., AF is an atomic operator that can be read as "In all paths in the future there is some state where...";

Syntax Notes

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation and Intuition

communicate
Syntax of CTL

Semantics of CTL

what we can say More about semantics Equivalences

Improving our

- Notice that, in CTL, the combination of path specific operators and temporal operators are atomic, e.g., AF is an atomic operator that can be read as "In all paths in the future there is some state where...";
- Notice as well that the binary operators $A[\varphi U\psi]$ and $E[\varphi U\psi]$ can be represented as AU and EU, respectively;

Syntax Notes

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and Intuition

Syntax of CTL Semantics of

Semantics of CTL

What we can say
More about
semantics
Equivalences

Equivalences Improving our language

- Notice that, in CTL, the combination of path specific operators and temporal operators are atomic, e.g., *AF* is an atomic operator that can be read as "In all paths in the future there is some state where...";
- Notice as well that the binary operators $A[\varphi U\psi]$ and $E[\varphi U\psi]$ can be represented as AU and EU, respectively;
- We assume that, similarly to the \neg operator, the "new" unary operators (AX, EX, AF, EF, AG, and EG) have the first precedence. Next comes the \land and \lor operators. And at last the \rightarrow , AU and EU;

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

How to communicate Syntax of CTI

Syntax of CTL Semantics of CTL

Some examples o what we can say

More about semantics Equivalences

Improving ou language

- Examples of well-formed formulas:
 - $\blacksquare \ \textit{AG}(\textit{p} \lor \textit{EFq})$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Syntax of CTL Semantics of CTL

what we can sav

- Examples of well-formed formulas:
 - \blacksquare $AG(p \lor EFq)$
 - \blacksquare $AX(q \rightarrow E[(p \lor q)Ur])$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation and

Motivation ar Intuition

communicat

Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving ou language

- Examples of well-formed formulas:
 - $AG(p \lor EFq)$
 - $AX(q \to E[(p \lor q)Ur])$
 - $EFEGp \rightarrow AFr$ Note that this is binded as $(EFEGp) \rightarrow AFr$, not as $EFEG(p \rightarrow AFr)$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and

Intuition

communicat

Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving ou language

- Examples of well-formed formulas:
 - \blacksquare $AG(p \lor EFq)$
 - $\blacksquare AX(q \to E[(p \lor q)Ur])$
 - $EFEGp \rightarrow AFr$ Note that this is binded as $(EFEGp) \rightarrow AFr$, not as $EFEG(p \rightarrow AFr)$
- Example of formulas that are not well-formed:
 - $\blacksquare A \neg G \neg p$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation an

Motivation ar Intuition

How to

Syntax of CTL Semantics of CTL

Some examples o what we can say

More about semantics Equivalences

Improving ou language

- Examples of well-formed formulas:
 - $\blacksquare AG(p \lor EFq)$
 - $\blacksquare AX(q \to E[(p \lor q)Ur])$
 - $EFEGp \rightarrow AFr$ Note that this is binded as $(EFEGp) \rightarrow AFr$, not as $EFEG(p \rightarrow AFr)$
- Example of formulas that are not well-formed:
 - $\blacksquare A \neg G \neg p$
 - F[pUs]

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...

Motivation ar Intuition

How to

Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving ou language

- Examples of well-formed formulas:
 - $AG(p \lor EFq)$
 - $\blacksquare \ AX(q \to E[(p \lor q)Ur])$
 - $EFEGp \rightarrow AFr$ Note that this is binded as $(EFEGp) \rightarrow AFr$, not as $EFEG(p \rightarrow AFr)$
- Example of formulas that are not well-formed:
 - $\blacksquare A \neg G \neg p$
 - \blacksquare F[pUs]
 - $A[pUs \land qUs]$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and Intuition
How to communicate
Syntax of CTL
Semantics of CTL

Some examples of what we can say More about semantics Equivalences

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation and Intuition How to

How to communicate Syntax of CT Semantics of CTL

what we can say
More about
semantics
Equivalences
Improving our

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and Intuition
How to communicate
Syntax of CTI
Semantics of

Some examples of what we can say More about semantics Equivalences

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

Definition (2)

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and Intuition
How to communicate
Syntax of CTI
Semantics of

Some examples of what we can say More about semantics Equivalences Improving our

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

Definition (2)

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

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTL Semantics of

Some examples of what we can say More about semantics Equivalences

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 $s \in S$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTL Semantics of CTL

Some examples of what we can say More about semantics Equivalences

- $\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \not\vDash \bot$ for all $s \in S$
- \mathcal{M} , $s \models p$ iff $p \in L(S)$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Semantics of

what we can sav

- $\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \not\vDash \bot$ for all $s \in S$
- $\blacksquare \mathcal{M}, s \models p \text{ iff } p \in L(S)$
- $\blacksquare \mathcal{M}, s \vDash \neg \varphi \text{ iff } \mathcal{M}, s \not\vDash \varphi$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTL Semantics of

Some examples of what we can say More about semantics

Equivalences

Equivalences Improving our

- \blacksquare $\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \not\vDash \bot$ for all $s \in S$
- \mathcal{M} , $s \models p$ iff $p \in L(S)$
- $\blacksquare \mathcal{M}, s \vDash \neg \varphi \text{ iff } \mathcal{M}, s \not\vDash \varphi$
- $\mathcal{M}, s \vDash \varphi_1 \land \varphi_2$ iff $\mathcal{M}, s \vDash \varphi_1$ AND $\mathcal{M}, s \vDash \varphi_2$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTI Semantics of

Some examples of what we can say More about semantics Equivalences

Equivalences Improving our

- \blacksquare $\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \not\vDash \bot$ for all $s \in S$
- \mathcal{M} , $s \models p$ iff $p \in L(S)$
- $\blacksquare \mathcal{M}, s \vDash \neg \varphi \text{ iff } \mathcal{M}, s \nvDash \varphi$
- $\mathcal{M}, s \vDash \varphi_1 \land \varphi_2$ iff $\mathcal{M}, s \vDash \varphi_1$ AND $\mathcal{M}, s \vDash \varphi_2$
- \mathcal{M} , $s \vDash \varphi_1 \lor \varphi_2$ iff \mathcal{M} , $s \vDash \varphi_1$ OR \mathcal{M} , $s \vDash \varphi_2$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTI Semantics of

Some examples of what we can say More about semantics
Equivalences

Improving our language

- \blacksquare $\mathcal{M}, s \vDash \top$ and $\mathcal{M}, s \not\vDash \bot$ for all $s \in S$
- \mathcal{M} , $s \models p$ iff $p \in L(S)$
- $\blacksquare \mathcal{M}, s \vDash \neg \varphi \text{ iff } \mathcal{M}, s \not\vDash \varphi$
- $\mathcal{M}, s \vDash \varphi_1 \land \varphi_2$ iff $\mathcal{M}, s \vDash \varphi_1$ AND $\mathcal{M}, s \vDash \varphi_2$
- \mathcal{M} , $s \vDash \varphi_1 \lor \varphi_2$ iff \mathcal{M} , $s \vDash \varphi_1$ OR \mathcal{M} , $s \vDash \varphi_2$
- $\blacksquare \mathcal{M}, s \vDash \varphi_1 \rightarrow \varphi_2 \text{ iff } \mathcal{M}, s \not\vDash \varphi_1 \text{ OR } \mathcal{M}, s \vDash \varphi_2$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation and Intuition
How to communicate
Syntax of CTI
Semantics of

Some examples of what we can say More about semantics

Equivalences Improving our 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:

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

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation and Intuition
How to communicate
Syntax of CTL
Semantics of

Some examples of what we can say More about semantics

- $\mathcal{M}, s \vDash AX\varphi$ iff for all s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \vDash \varphi$. Thus, AX says: "in every next state..."
- $\mathcal{M}, s \vDash EX\varphi$ iff exists s_1 that $s \to s_1$ and $M, s_1 \vDash \varphi$. Thus, EX says: "in some next state…"

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTI Semantics of

Some examples of what we can say More about semantics Equivalences Improving our

- $\mathcal{M}, s \vDash AX\varphi$ iff for all s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \vDash \varphi$. Thus, AX says: "in every next state..."
- $\mathcal{M}, s \vDash EX\varphi$ iff exists s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \vDash \varphi$. Thus, EX says: "in some next state…"
- \mathcal{M} , s, $\vDash AG\varphi$ iff for all paths $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, for all s_i , \mathcal{M} , $s_i \vDash \varphi$. Thus, AG says: "In all possible paths from now on in all next states..."

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTI Semantics of

Some examples of what we can say More about semantics Equivalences Improving our

- $\mathcal{M}, s \vDash AX\varphi$ iff for all s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \vDash \varphi$. Thus, AX says: "in every next state..."
- $\mathcal{M}, s \vDash EX\varphi$ iff exists s_1 that $s \to s_1$ and $\mathcal{M}, s_1 \vDash \varphi$. Thus, EX says: "in some next state…"
- $\mathcal{M}, s, \models AG\varphi$ iff for all paths $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow ...$ in which $s = s_1$, for all s_i , $\mathcal{M}, s_i \models \varphi$. Thus, AG says: "In all possible paths from now on in all next states..."
- \mathcal{M} , s, $\vDash AG\varphi$ iff exists some path $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, for all s_i , \mathcal{M} , $s_i \vDash \varphi$ Thus, EG says: "Exists a path from now on in all next states..."

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Chapters...
Motivation and Intuition
How to

How to communicate Syntax of CT Semantics of CTL

what we can say More about semantics Equivalences 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, $\vDash AF\varphi$ iff for all paths $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, exists s_i , \mathcal{M} , $s_i \vDash \varphi$. Thus, AF says: "In all possible paths from now on, in some next state..."

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTI Semantics of

Some examples of what we can say
More about

More about semantics Equivalences Improving our language

- \mathcal{M} , s, \vDash $AF\varphi$ iff for all paths $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, exists s_i , \mathcal{M} , $s_i \vDash \varphi$. Thus, AF says: "In all possible paths from now on, in some next state..."
- \mathcal{M} , s, $\vDash EF\varphi$ iff exists some path $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, that exists s_i , \mathcal{M} , $s_i \vDash \varphi$. Thus, EF says: "In some path from now on, in some next state..."

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation and Intuition
How to communicate
Syntax of CT
Semantics of

Some examples of what we can say

More about

More about semantics Equivalences Improving our language

- \mathcal{M} , s, $\vDash AF\varphi$ iff for all paths $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, exists s_i , \mathcal{M} , $s_i \vDash \varphi$. Thus, AF says: "In all possible paths from now on, in some next state..."
- \mathcal{M} , s, $\vDash EF\varphi$ iff exists some path $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, that exists s_i , \mathcal{M} , $s_i \vDash \varphi$. Thus, EF says: "In some path from now on, in some next state..."
- $\mathcal{M}, s, \models A[\varphi_1 U \varphi_2]$ iff for all paths $s_1 \to s_2 \to s_3 \to ...$ in which $s = s_1$, this path satisfies $\varphi_1 U \varphi_2$, i.e., exists s_i in the path such that $\mathcal{M}, s_i \models \varphi_2$ and, for all j < i, $\mathcal{M}, s_j \models \varphi_1$. Thus, AU says: "For all paths from now on, until some state..."

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

 $\varphi, \varphi_1, \varphi_2$ be well-formed formulas of CTL. And let p be an atom. The satisfaction of CTL formulas can be defined as Semantics of follows: what we can sav

> ■ \mathcal{M} , s, $\models E[\varphi_1 U \varphi_2]$ iff exists some path $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow ...$ in which $s = s_1$, this path satisfies $\varphi_1 U \varphi_2$. Thus, EU says: "In some path from now on, until some state..."

Take an arbitrary model \mathcal{M} . Let s, s_1, s_2, s_3 be states in S. Let

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation and Intuition
How to communicate
Syntax of CTL
Semantics of CTL

Some examples of what we can say

More about semantics
Equivalences

Equivalences Improving our ■ "It's possible to get to a state where something has started but it's not ready": $EF(started \land \neg ready)$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTL Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

■ "It's possible to get to a state where something has started but it's not ready": $EF(started \land \neg ready)$

■ "A certain process is enabled infinitely often on every computation path": AG(AFenabled)

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Some examples of what we can sav

- "It's possible to get to a state where something has started but it's not ready": $EF(started \land \neg ready)$
- "A certain process is enabled infinitely often on every computation path": AG(AFenabled)
- "An upwards travelling lift at the second floor does not change its direction when it has passengers wishing to go to the fifth floor":

 $AG(floor2 \land directionup \land button5 \rightarrow A[directionup Ufloor5])$

Equivalences

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...

Motivation and

How to communicate

Syntax of CT Semantics of CTL

Some examples of what we can say

semantics Equivalences

Improving out

Definition

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.

Equivalences

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and

Semantics of CTL what we can sav

Equivalences

Definition

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.

Notation: we denote the semantic equivalence of φ and ψ by $\varphi \equiv \psi$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation and

Motivation and Intuition

How to communicate Syntax of CT Semantics of CTL

CTL Some examples of what we can say

More about semantics Equivalences

Improving our

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and

Semantics of CTL

what we can sav

Equivalences

- $\blacksquare \neg AF\varphi \equiv EG\neg \varphi$
- $\blacksquare \neg \mathit{EF}\varphi \equiv \mathit{AG}\neg \varphi$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...

Motivation and

How to communicate Syntax of CT Semantics of CTL

Some examples of what we can say

More about semantics
Equivalences

Improving our language

$$\blacksquare \neg AF\varphi \equiv EG\neg \varphi$$

$$\blacksquare \neg \textit{EF}\varphi \equiv \textit{AG}\neg \varphi$$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...

Motivation an

How to communicate

Syntax of CT Semantics of CTL

Some examples of what we can say

semantics Equivalences

Improving our

$$\quad \blacksquare \ \neg \textit{AF}\varphi \equiv \textit{EG}\neg \varphi$$

$$\blacksquare \neg \mathit{EF}\varphi \equiv \mathit{AG}\neg\varphi$$

$$\blacksquare \neg AX\varphi \equiv EX\neg \varphi$$

$$AF\varphi \equiv A[\top U\varphi]$$

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Semantics of CTL

what we can sav

Equivalences

$$\blacksquare \neg AF\varphi \equiv EG\neg \varphi$$

$$\blacksquare \neg \mathit{EF}\varphi \equiv \mathit{AG}\neg\varphi$$

$$AF\varphi \equiv A[\top U\varphi]$$

$$\blacksquare \ \textit{EF}\varphi \equiv \textit{E}[\top \textit{U}\varphi]$$

Minimum set of CTL connectives

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CTL Semantics of CTL

Some examples of what we can say

semantics Equivalences

Improving ou language Because of the equivalences shown and the ones in propositional logic, we can have some minimum sets of conectives for the CTL syntax. One of them is defined in Backus-Naur formalism below:

$$\phi ::= \top \mid p \mid \neg \phi \mid \phi \rightarrow \phi \mid AX\phi \mid A[\phi U\phi] \mid E[\phi U\phi]$$

That's all we need?

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters... Motivation an Intuition

How to communicate Syntax of CT Semantics of CTL

Some examples of what we can say

More about semantics
Equivalences

Improving our language Even if CTL allow explicit quantification over paths, it cannot allow some expressions to be formed. For example, we cannot say, as in LTL: "All paths in which have p on them, also have q on them".

This expression can be translated in LTL as follows:

$$Fp \rightarrow Fq$$

That's all we need?

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

chapters...
Motivation and Intuition
How to communicate
Syntax of CTL
Semantics of CTL

Some examples of what we can say More about semantics

semantics
Equivalences
Improving our
language

We can try expressing it as $AFp \rightarrow AFq$ but it does not have the same meaning. This one statement means "If all paths have a p along them, then all paths have a q along then"

That's all we need?

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation and Intuition
How to communicate
Syntax of CTI
Semantics of

More about semantics

semantics Equivalences Improving our language We can try expressing it as $AFp \to AFq$ but it does not have the same meaning. This one statement means "If all paths have a p along them, then all paths have a q along then" We can try to translate it as $AG(p \to AFq)$ which is closer, but not exactly the same. This one means "for all paths, in all states on the future, if they hold p then, all paths will eventually hold q"

Presenting CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate

Syntax of CT Semantics of CTL

what we can say

More about
semantics

Improving our language

For this, we can extend the CTL by dropping the constraint that every temporal operator (X, U, F, G) has to be associated with an unique path quantifier (A, E).

Presenting CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to communicate Syntax of CT

Syntax of CTI Semantics of CTL Some examples

what we can say
More about
semantics
Equivalences

Improving our language

For this, we can extend the CTL by dropping the constraint that every temporal operator (X, U, F, G) has to be associated with an unique path quantifier (A, E).

This allows us to generate some statements:

Presenting CTL* Statements only possible with CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Chapters...
Motivation and Intuition
How to communicate
Syntax of CTL
Semantics of CTL

Some examples of what we can say

More about semantics Equivalences

Improving our language

■ "In all possible paths, q is true until r is true or p is true until r is true": $A[qUr \lor pUr]$

Presenting CTL* Statements only possible with CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Motivation and Intuition How to

How to communicate Syntax of CTL Semantics of CTL

what we can sa More about semantics

Improving our language

- "In all possible paths, q is true until r is true or p is true until r is true": $A[qUr \lor pUr]$
- "There is a path in which p eventually occurring will occur in all states": E[GFp]

Presenting CTL* Statements only possible with CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters...
Motivation a

How to communicate Syntax of CTL Semantics of

More about semantics

Improving our language

- "In all possible paths, q is true until r is true or p is true until r is true": $A[qUr \lor pUr]$
- "There is a path in which p eventually occurring will occur in all states": E[GFp]
- "In all paths, p will occur in the next state or in the next of the next": $A[Xp \lor XXp]$

Presenting CTL* CTL* syntax

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

Chapters...
Motivation and Intuition
How to communicate

communicate Syntax of CTI Semantics of CTL

More about semantics

Improving our language

The syntax of CTL* can be defined with the BNF bellow:

$$\phi ::= \bot \mid \top \mid p \mid \neg \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \rightarrow \phi \mid A[\alpha] \mid E[\alpha] \mid$$

$$\alpha ::= \phi | \ \neg \alpha \ | \ \alpha \wedge \alpha \ | \ \alpha \vee \alpha \ | \ \alpha \rightarrow \alpha \ | \ \alpha \textit{U}\alpha \ | \ \textit{G}\alpha \ | \ \textit{F}\alpha \ | \ \textit{X}\alpha |$$

With the same meanings of each operator.

Presenting CTL* LTL \subset CTL* and CTL \subset CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation ar Intuition How to

How to communicate Syntax of CT Semantics of CTL

Some examples o what we can say More about semantics

semantics Equivalences Improving our language Although we don't define path operators to LTL we can assume that it consider in all paths. Therefore, we can say that a formula ϕ in LTL is a formula $A[\phi]$ in CTL*;

Presenting CTL* LTL \subset CTL* and CTL \subset CTL*

Computation Tree Logic

Luis Tertulino & Ronaldo Silveira

In previous chapters... Motivation ar Intuition How to

How to communicate Syntax of CT Semantics of CTL

Some examples o what we can say More about semantics

semantics Equivalences Improving our language Although we don't define path operators to LTL we can assume that it consider in all paths. Therefore, we can say that a formula ϕ in LTL is a formula $A[\phi]$ in CTL*; For CTL, it is trivial;