Invariants

Definition

The class of LTL formulas that capture *invariants* is defined by $\Box g$ where

$$g ::= a \mid g \wedge g \mid \neg g.$$

Invariants

Definition

The class of LTL formulas that capture *invariants* is defined by $\Box g$ where

$$g ::= a \mid g \wedge g \mid \neg g.$$

Example

$\Box\neg$red

Safety properties

Safety properties are characterized by "nothing bad ever happens." For example, "a red light is immediately preceded by amber" is a safety property.

Safety properties

Safety properties are characterized by “nothing bad ever happens.” For example, “a red light is immediately preceded by amber” is a safety property.

Question

How can we express this property in LTL?

Safety properties

Safety properties are characterized by “nothing bad ever happens.” For example, “a red light is immediately preceded by amber” is a safety property.

Question

How can we express this property in LTL?

Answer

$\neg \mathsf{red} \wedge \Box(\bigcirc \mathsf{red} \Rightarrow \mathsf{amber})$

Liveness properties

Liveness properties are characterized by “something good eventually happens.” For example, “the light is infinitely often red” is a liveness property.

Liveness properties

Liveness properties are characterized by “something good eventually happens.” For example, “the light is infinitely often red” is a liveness property.

Question

How can we express this property in LTL?

Liveness properties

Liveness properties are characterized by "something good eventually happens." For example, "the light is infinitely often red" is a liveness property.

Question

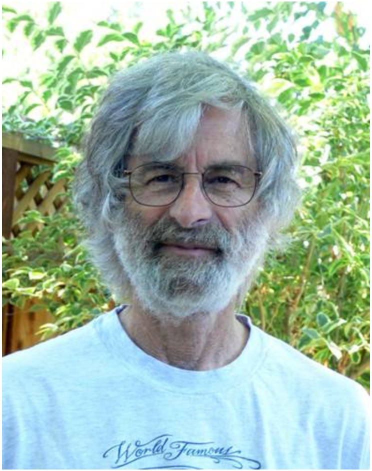
How can we express this property in LTL?

Answer

$\Box\Diamond\text{red}$

Leslie Lamport

- Won the Turing award in 2013.
- Won the Dijkstra prize three times (2000, 2005, 2014).
- Elected Fellow of the ACM in 2014.



Source: Leslie Lamport

Expressiveness of LTL

Question

Are there properties we cannot express in LTL?

Expressiveness of LTL

Question

Are there properties we cannot express in LTL?

Answer

Yes, for example, “Always a state satisfying *a* can be reached.”

Expressiveness of LTL

Theorem

There does not exists an LTL formula f with $TS \models f$ iff

$$\forall s \in I : \forall p \in Paths(s) : \forall m \geq 0 : \exists q \in Paths(p[m]) : \exists n \geq 0 : q[n] \models a$$

How to modify the logic?

$$\forall s \in I : \forall p \in Paths(s) : \forall m \geq 0 : \exists q \in Paths(p[m]) : \underbrace{\exists n \geq 0 : q[n] \models a}_{\Diamond a}$$

How to modify the logic?

$$\forall s \in I : \forall p \in Paths(s) : \forall m \geq 0 : \overbrace{\exists q \in Paths(p[m]) : \underbrace{\exists n \geq 0 : q[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

How to modify the logic?

$$\forall s \in I : \forall p \in \mathit{Paths}(s) : \underbrace{\forall m \geq 0 : \overbrace{\exists q \in \mathit{Paths}(p[m]) : \underbrace{\exists n \geq 0 : q[n] \models a}_{\Diamond a}}^{\exists \Diamond a}}_{\Box \exists \Diamond a}$$

How to modify the logic?

$$\forall s \in I : \overbrace{\forall p \in \mathit{Paths}(s) : \underbrace{\forall m \geq 0 : \overbrace{\exists q \in \mathit{Paths}(p[m]) : \underbrace{\exists n \geq 0 : q[n] \models a}_{\Diamond a}}^{\exists \Diamond a}}_{\Box \exists \Diamond a}}^{\forall \Box \exists \Diamond a}$$

How to modify the logic?

$$\overbrace{\exists p \in Paths(s) : \underbrace{\exists n \geq 0 : p[n] \models a}_{p \models \Diamond a}}^{? \models \exists \Diamond a}$$

Recall that $p \models \Diamond a$ expresses that path p satisfies formula $\Diamond a$.

Question

$? \models \exists \Diamond a$.

How to modify the logic?

$$\overbrace{\exists p \in \mathit{Paths}(s) : \underbrace{\exists n \geq 0 : p[n] \models a}_{p \models \Diamond a}}^{? \models \exists \Diamond a}$$

Recall that $p \models \Diamond a$ expresses that path p satisfies formula $\Diamond a$.

Question

$? \models \exists \Diamond a$.

Answer

There exists a path p starting in state s such that $p \models \Diamond a$, hence, $s \models \exists \Diamond a$.

How to modify the logic?

$$\overbrace{\exists p \in Paths(s) : \underbrace{\exists n \geq 0 : p[n] \models a}_{p \models \Diamond a}}^{? \models \exists \Diamond a}$$

Recall that $p \models \Diamond a$ expresses that path p satisfies formula $\Diamond a$.

Question

$? \models \exists \Diamond a$.

Answer

There exists a path p starting in state s such that $p \models \Diamond a$, hence, $s \models \exists \Diamond a$.

Consequence

We should distinguish between *path formulas* and *state formulas*.

Computational Tree Logic

EECS 4315

www.eecs.yorku.ca/course/4315/

Syntax

The *state formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists g \mid \forall g$$

The *path formulas* are defined by

$$g ::= \bigcirc f \mid f \mathrel{\mathsf{U}} f$$

Syntax

The *state formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists g \mid \forall g$$

The *path formulas* are defined by

$$g ::= \bigcirc f \mid f \, \mathsf{U} \, f$$

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall (f \, \mathsf{U} \, f)$$

Computation tree logic

Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In, Dexter Kozen, editor, *Proceedings of Workshop on Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Yorktown Heights, NY, USA, May 1981. Springer-Verlag.

Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In, Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *Proceedings of the 5th International Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Torino, Italy, April 1982. Springer-Verlag.

Syntactic sugar

$$
\begin{array}{rcl}
\exists\Diamond f & = & \exists(\mathsf{true}\ \mathsf{U}\ f) \\
\forall\Diamond f & = & \forall(\mathsf{true}\ \mathsf{U}\ f) \\
\exists\Box f & = & \neg\forall(\mathsf{true}\ \mathsf{U}\ \neg f) \\
\forall\Box f & = & \neg\exists(\mathsf{true}\ \mathsf{U}\ \neg f)
\end{array}
$$

Example

Question

How to express “Each red light is preceded by an amber light” in CTL?

Example

Question

How to express "Each red light is preceded by an amber light" in CTL?

Answer

$\neg \text{red} \wedge \forall \Box (\text{amber} \vee \forall \bigcirc \neg \text{red})$

Example

Question

How to express "The light is infinitely often green" in CTL?

Example

Question

How to express “The light is infinitely often green” in CTL?

Answer

$\forall\Box\forall\Diamond$green

Semantics of CTL

$$\begin{array}{rcl} s \models a & \text{iff} & a \in \ell(s) \\ s \models f_1 \wedge f_2 & \text{iff} & s \models f_1 \wedge s \models f_2 \\ s \models \neg f & \text{iff} & s \not\models f \\ s \models \exists g & \text{iff} & \exists p \in \mathit{Paths}(s) : p \models g \\ s \models \forall g & \text{iff} & \forall p \in \mathit{Paths}(s) : p \models g \end{array}$$

and

$$\begin{array}{rcl} p \models \bigcirc f & \text{iff} & p[1] \models f \\ p \models f_1 \mathrm{~U~} f_2 & \text{iff} & \exists i \geq 0 : p[i] \models f_2 \wedge \forall 0 \leq j < i : p[j] \models f_1 \end{array}$$

Semantics of CTL

Question

Recall that

$$\exists\Diamond f = \exists(\text{true}\ \mathsf{U}\ f).$$

How is

$$s \models \exists\Diamond f$$

defined?

Semantics of CTL

Question

Recall that

$$\exists\Diamond f = \exists(\mathsf{true}\,\mathsf{U}\,f).$$

How is

$$s \models \exists\Diamond f$$

defined?

Answer

$\exists p \in Paths(s) : \exists i \geq 0 : p[i] \models f$

Semantics of CTL

Question

Recall that

$$\forall\Diamond f = \forall(\text{true}\,\mathsf{U}\,f).$$

How is

$$s \models \forall\Diamond f$$

defined?

Semantics of CTL

Question

Recall that

$$\forall\Diamond f = \forall(\text{true}\ \mathsf{U}\ f).$$

How is

$$s \models \forall\Diamond f$$

defined?

Answer

$\forall p \in Paths(s) : \exists i \geq 0 : p[i] \models f$

Semantics of CTL

Question

Recall that

$$\exists\Box f = \neg\forall(\mathrm{true}\,\mathsf{U}\,\neg f).$$

How is

$$s \models \exists\Box f$$

defined?

Semantics of CTL

Question

Recall that

$$\exists\Box f = \neg\forall(\text{true}\,\mathsf{U}\,\neg f).$$

How is

$$s \models \exists\Box f$$

defined?

Answer

$\exists p \in Paths(s) : \forall i \geq 0 : p[i] \models f$

Semantics of CTL

Question

Recall that

$$\forall\Box f = \neg\exists(\mathrm{true}\ \mathrm{U}\ \neg f).$$

How is

$$s \models \exists\Box f$$

defined?

Semantics of CTL

Question

Recall that

$$\forall\Box f = \neg\exists(\text{true} \, \mathsf{U} \, \neg f).$$

How is

$$s \models \exists\Box f$$

defined?

Answer

$\forall p \in Paths(s) : \forall i \geq 0 : p[i] \models f$

Semantics of CTL

$$TS \models f \text{ iff } \forall s \in I : s \models f.$$

Expressiveness of LTL and CTL

Theorem

The property

$$\forall s \in I : \forall p \in Paths(s) : \forall m \geq 0 : \exists q \in Paths(p[m]) : \exists n \geq 0 : q[n] \models a$$

cannot be captured by LTL, but is captured by the CTL formula $\forall \Box \exists \Diamond a$.

Expressiveness of LTL and CTL

Theorem

The property

$$\forall s \in I : \forall p \in \mathit{Paths}(s) : \exists i \geq 0 : \forall j \geq i : p[j..] \models a$$

cannot be captured by CTL, but is captured by the LTL formula $\Diamond\Box a$.

Model checking CTL

Definition

The *satisfaction set* $Sat(f)$ is defined by

$$Sat(f) = \{\, s \in S \mid s \models f \,\}.$$

Model checking CTL

Definition

The *satisfaction set* $Sat(f)$ is defined by

$$Sat(f) = \{ s \in S \mid s \models f \}.$$

Basic idea

Compute $Sat(f)$ by recursion on the structure of f.

$TS \models f$ iff $I \subseteq Sat(f)$.

Alternative view

Label each state with the subformulas of f that it satisfies.

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists(f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall(f \, \mathsf{U} \, f)$$

Question

What is $Sat(a)$?

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists(f \text{ U } f) \mid \forall \bigcirc f \mid \forall(f \text{ U } f)$$

Question

What is $Sat(a)$?

Answer

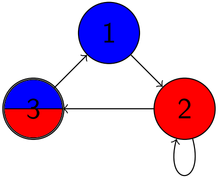
$Sat(a) = \{ s \in S \mid a \in \ell(s) \}$

Alternative view

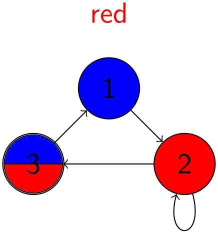
Label each state s satisfying $a \in \ell(s)$ with a.

Example

red



Example



$1 \mapsto \emptyset$

$2 \mapsto \{\text{red}\}$

$3 \mapsto \{\text{red}\}$

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists(f \mathsf{U} f) \mid \forall \bigcirc f \mid \forall(f \mathsf{U} f)$$

Question

What is $Sat(f \wedge g)$?

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall (f \, \mathsf{U} \, f)$$

Question

What is $Sat(f \wedge g)$?

Answer

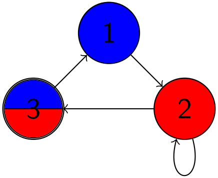
$Sat(f \wedge g) = Sat(f) \cap Sat(g)$

Alternative view

Label states, that are labelled with both f and g, also with $f \wedge g$.

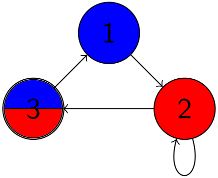
Example

red ∧ blue



Example

red $\wedge$ blue



$1 \mapsto \{\text{blue}\}$

$2 \mapsto \{\text{red}\}$

$3 \mapsto \{\text{red}, \text{blue}, \text{red} \wedge \text{blue}\}$

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists(f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall(f \, \mathsf{U} \, f)$$

Question

What is $Sat(\neg f)$?

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \; \mathsf{U} \; f) \mid \forall \bigcirc f \mid \forall (f \; \mathsf{U} \; f)$$

Question

What is $Sat(\neg f)$?

Answer

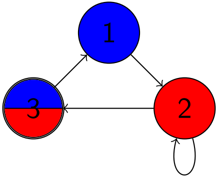
$Sat(\neg f) = S \setminus Sat(f)$

Alternative view

Label each state, that is not labelled with f, with $\neg f$.

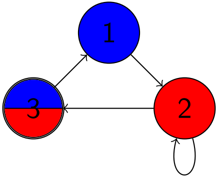
Example

$\neg(\text{red} \wedge \text{blue})$



Example

$$\neg(\text{red} \wedge \text{blue})$$



$$1 \mapsto \{\text{blue}, \neg(\text{red} \wedge \text{blue})\}$$
$$2 \mapsto \{\text{red}, \neg(\text{red} \wedge \text{blue})\}$$
$$3 \mapsto \{\text{red}, \text{blue}, \text{red} \wedge \text{blue}\}$$

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists(f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall(f \, \mathsf{U} \, f)$$

Question

What is $Sat(\exists \bigcirc f)$?

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \mathbin{\mathsf{U}} f) \mid \forall \bigcirc f \mid \forall (f \mathbin{\mathsf{U}} f)$$

Question

What is $Sat(\exists \bigcirc f)$?

Answer

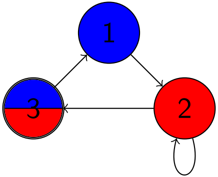
$Sat(\exists \bigcirc f) = \{\, s \in S \mid succ(s) \cap Sat(f) \neq \emptyset \,\}$ where $succ(s) = \{\, t \in S \mid s \to t \,\}$.

Alternative view

Labels those states, that have a direct successor labelled with f, with $\exists \bigcirc f$.

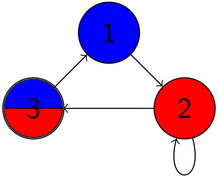
Example

$\exists \bigcirc$ red



Example

$\exists \bigcirc$ red



$1 \mapsto \{\exists \bigcirc \text{red}\}$

$2 \mapsto \{\text{red}, \exists \bigcirc \text{red}\}$

$3 \mapsto \{\text{red}\}$

Second progress report

Submit the second progress report before Saturday March 30. If you submit your progress report before the deadline and you have made good progress with your project since the first progress report, you will receive 5 towards the mark for your project (40 in total).

Course evaluation

The course evaluation for this course can now be completed at `https://courseevaluations.yorku.ca`

I would really appreciate it if you would take the time to complete the course evaluation. Your feedback allows me to improve the course for future students.

If at least 80 of the students in the course (that is, 12) complete the evaluation, I will bring cup cakes for the last lecture.