Linear Temporal Logic EECS 4315

www.cse.yorku.ca/course/4315/

www.cse.yorku.ca/course/4315/ EECS 4315

ъ

イロト 不得 とくほ とくほ とう

The weak until operator is defined by

 $\varphi \mathsf{W} \psi = (\varphi \mathsf{U} \psi) \lor \Box \phi$

æ

ヘロト 人間 とくほとくほとう

The weak until operator is defined by

$$\varphi \mathsf{W} \psi = (\varphi \mathsf{U} \psi) \vee \Box \phi$$

Question

 $\pi\models\varphi\,\mathsf{W}\,\psi\,\mathsf{iff?}$

æ

ヘロト 人間 とくほ とくほ とう

The weak until operator is defined by

$$\varphi \, \mathsf{W} \, \psi = (\varphi \, \mathsf{U} \, \psi) \vee \Box \phi$$

Question

 $\pi\models\varphi\,\mathsf{W}\,\psi\,\mathsf{iff?}$

Answer

$$(\exists i \ge \mathbf{0} : \pi[i..] \models \psi \text{ and } \forall \mathbf{0} \le j < i : \pi[j..] \models \varphi) \text{ or } \forall i \ge \mathbf{0} : \pi[i..] \models \varphi$$

æ

The set of LTL formulas in positive normal form is defined by the grammar

 $\varphi ::= \mathsf{true} \mid \mathsf{false} \mid a \mid \neg a \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \bigcirc \varphi \mid \varphi \lor \varphi \mid \varphi \lor \varphi \mid \varphi \lor \varphi$

イロト イポト イヨト イヨト

The set of LTL formulas in positive normal form is defined by the grammar

$$\varphi ::= \mathsf{true} \mid \mathsf{false} \mid a \mid \neg a \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \bigcirc \varphi \mid \varphi \lor \varphi \mid \varphi \lor \varphi \mid \varphi \lor \varphi$$

Theorem

For each LTL formula there exists an equivalent LTL formula in positive normal form.

Are there properties we cannot express in LTL?

э

・ロト ・ 同ト ・ ヨト ・ ヨト

Are there properties we cannot express in LTL?

Answer

Yes, for example, "Always a state satisfying a can be reached"

프 > - - 프 > - -

< D > < P >

Theorem

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

 $\forall \pi \in Paths(TS) : \forall m \ge 0 : \exists \pi' \in Paths(\pi[m]) : \exists n \ge 0 : \pi'[n] \models a.$

イロト イポト イヨト イヨト

$\forall \pi \in \textit{Paths}(\textit{TS}) : \forall m \ge 0 : \exists \pi' \in \textit{Paths}(\pi[m]) : \underbrace{\exists n \ge 0 : \pi'[n] \models a}_{\Diamond a}$

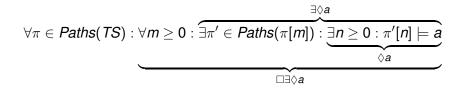
 $\equiv \rightarrow$

$$\forall \pi \in \textit{Paths}(\textit{TS}) : \forall m \ge 0 : \exists \pi' \in \textit{Paths}(\pi[m]) : \exists n \ge 0 : \pi'[n] \models a \\ \Diamond a \\ \Diamond a \\ \end{pmatrix}$$

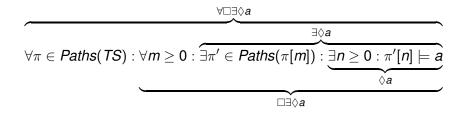
ww.cse.yorku.ca/course/4315/ EECS 4315

æ

<ロト < 回 > < 注 > < 注 > .



 $\equiv \rightarrow$



э

イロト イボト イヨト イヨト

How to Modify the Logic?

$$\overbrace{\exists \pi' \in \textit{Paths}(\pi[m]) : \underbrace{\exists n \ge 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

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



<ロト < 回 > < 回 > < 回 > .

How to Modify the Logic?

$$\overbrace{\exists \pi' \in \textit{Paths}(\pi[m]) : \underbrace{\exists n \ge 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

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



Answer

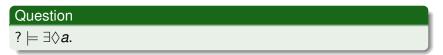
There exists a path π starting in state *s* such that $\pi \models \Diamond a$, hence, $s \models \exists \Diamond a$.

э

How to Modify the Logic?

$$\overbrace{\exists \pi' \in \textit{Paths}(\pi[m]) : \underbrace{\exists n \ge 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

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



Answer

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

Consequence

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

10/23

www.cse.yorku.ca/course/4315/ EECS 4315

The state formulas are defined by

$$\Phi ::= \mathsf{true} \mid a \mid \Phi \land \Phi \mid \neg \Phi \mid \exists \varphi \mid \forall \varphi$$

The path formulas are defined by

$$\varphi ::= \bigcirc \Phi \mid \Phi \cup \Phi$$

< ∃ →

Computation tree logic (CTL)

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.

ヘロト 人間 トイヨト イヨト

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

www.cse.yorku.ca/course/4315/ EECS 4315

æ

How to express "Each red light is preceded by a green light" in CTL?

æ

ヘロト 人間 とくほ とくほとう

How to express "Each red light is preceded by a green light" in CTL?

Answer

 $\neg \text{red} \land \forall \Box (\text{green} \lor \forall \bigcirc \neg \text{red})$

æ

(E) < E) </p>

 $\langle \Box \rangle \langle \Box \rangle$

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

Đ.

ヘロト 人間 とくほ とくほとう

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

Answer

∀⊟∀⊘green

æ

→ E → + E → -

Semantics of CTL

$$\begin{array}{c} s \models \mathsf{true} \\ s \models a \quad \mathsf{iff} \quad a \in \ell(s) \\ s \models \Phi \land \Psi \quad \mathsf{iff} \quad s \models \Phi \text{ and } s \models \Psi \\ s \models \neg \Phi \quad \mathsf{iff} \quad \mathsf{not}(s \models \Phi) \\ s \models \exists \varphi \quad \mathsf{iff} \quad \exists \pi \in \mathsf{Paths}(s) : \pi \models \varphi \\ s \models \forall \varphi \quad \mathsf{iff} \quad \forall \pi \in \mathsf{Paths}(s) : \pi \models \varphi \end{array}$$

and

$$\begin{array}{ccc} \pi \models \bigcirc \Phi & \text{iff} & \pi[1] \models \Phi \\ \pi \models \Phi \cup \Psi & \text{iff} & \exists i \ge 0 : \pi[i] \models \Psi \text{ and } \forall 0 \le j < i : \pi[j] \models \Phi \end{array}$$

æ

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

The *satisfaction set* $Sat(\Phi)$ is defined by

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

æ

▶ < ∃ >

Question		
Recall that		
	$\exists \Diamond \Phi = \exists (true \ U \ \Phi).$	
How is		
	$oldsymbol{s}\models\exists\Diamond\Phi$	
defined?		

Ξ.

イロト イロト イヨト イヨト

Question		
Recall that		
	$\exists \Diamond \Phi = \exists (true \ U \ \Phi).$	
How is		
	$oldsymbol{s}\models\exists\Diamond\Phi$	
defined?		

Answer

$$\exists \pi \in \textit{Paths}(s) : \exists i \geq 0 : \pi[i] \models \Phi.$$

æ

イロト イロト イヨト イヨト

Question		
Recall that		
	$\forall \Diamond \Phi = \forall (true \ U \ \Phi)$	
How is		
	$oldsymbol{s}\models orall \Diamond \Phi$	
defined?		

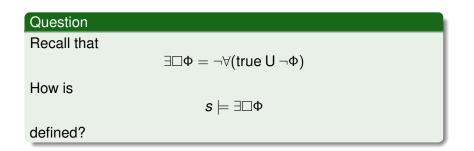
æ

Question		
Recall that		
	$\forall \Diamond \Phi = \forall (true \ U \ \Phi)$	
How is		
	$oldsymbol{s}\models orall \Diamond \Phi$	
defined?		

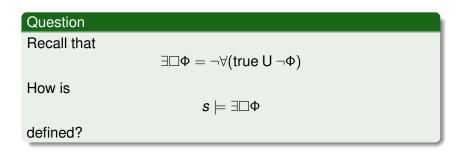
Answer

$$\forall \pi \in \textit{Paths}(s) : \exists i \geq 0 : \pi[i] \models \Phi.$$

æ



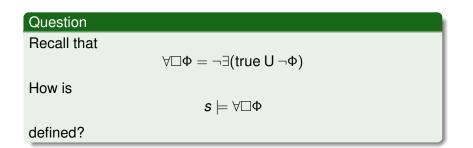
æ



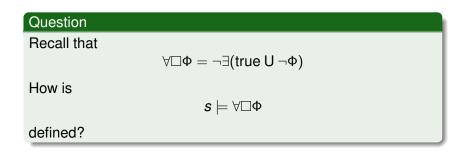
Answer

$$\exists \pi \in Paths(s) : \forall i \geq 0 : \pi[i] \models \Phi.$$

イロト イロト イヨト イヨト 三日



æ



Answer

$$\forall \pi \in Paths(s) : \forall i \geq 0 : \pi[i] \models \Phi.$$

イロト イロト イヨト イヨト 三日

Theorem

The property

 $\forall \pi \in Paths(TS) : \forall m \ge 0 : \exists \pi' \in Paths(\pi[m]) : \exists n \ge 0 : \pi'[n] \models a$

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

프 에 제 프 에 다

Theorem

The property

$$\forall \pi \in \textit{Paths}(\textit{TS}) : \exists i \ge 0 : \forall j \ge i : \pi[j..] \models a$$

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

э

イロト イポト イヨト イヨト