Overview

Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

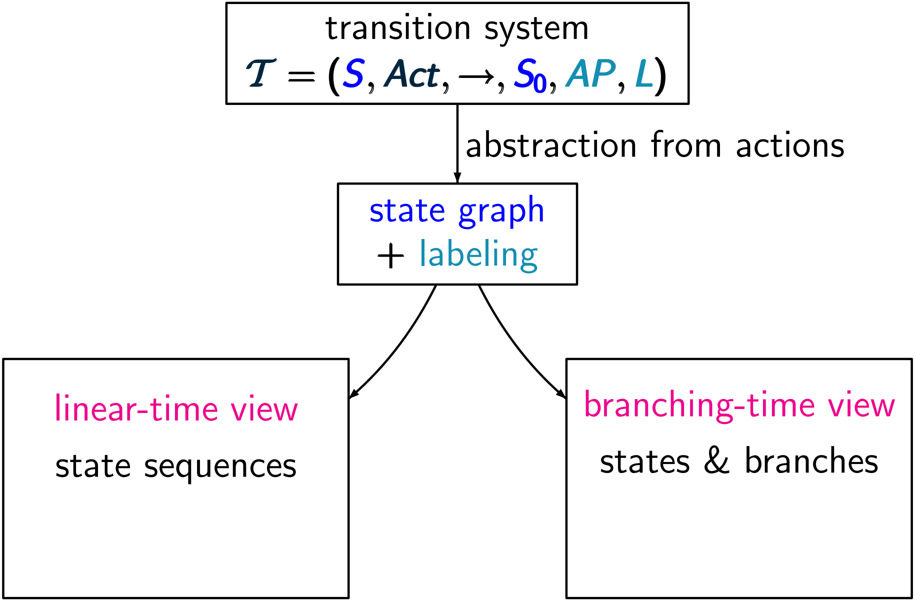
Linear Temporal Logic (LTL)

Computation Tree Logic

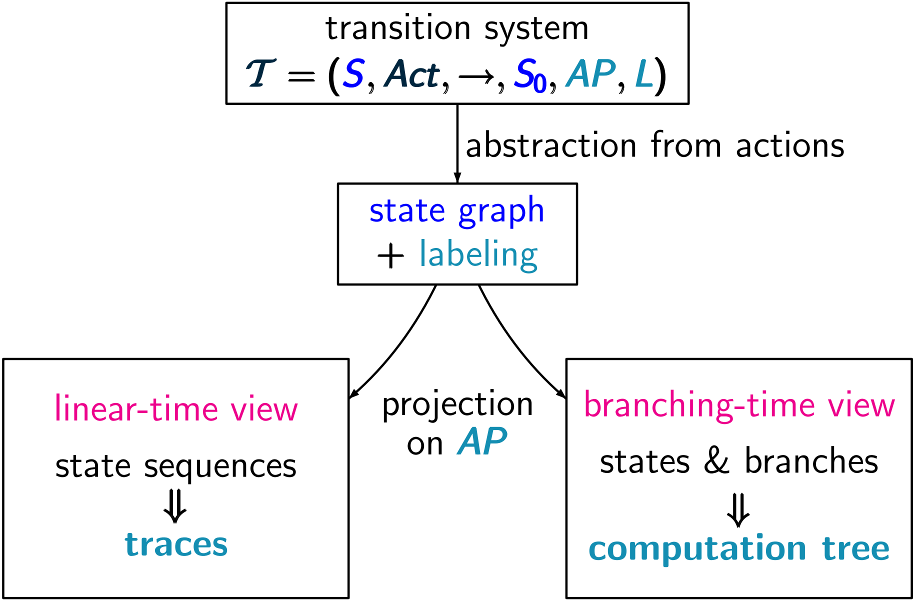
Equivalences and Abstraction

Linear vs branching time

Linear vs branching time



Linear vs branching time



Computation tree

Computation tree

The computation tree of a transition system $\mathcal{T} = (S, Act, \rightarrow, s_0, AP, L)$ arises by:

- unfolding into a tree
- abstraction from the actions
- projection of the states s to their labels $L(s) \subseteq AP$

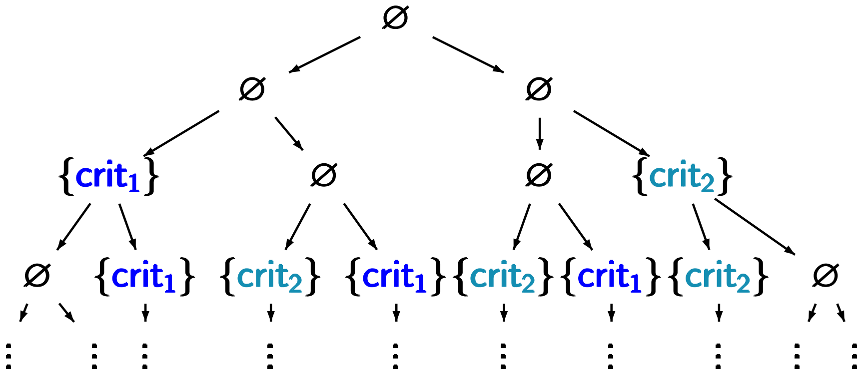
Computation tree

The computation tree of state s_0 in a transition system $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ arises by:

- unfolding $\mathcal{T}_{s_0} = (S, Act, \rightarrow, s_0, AP, L)$ into a tree
- abstraction from the actions
- projection of the states s to their labels $L(s) \subseteq AP$

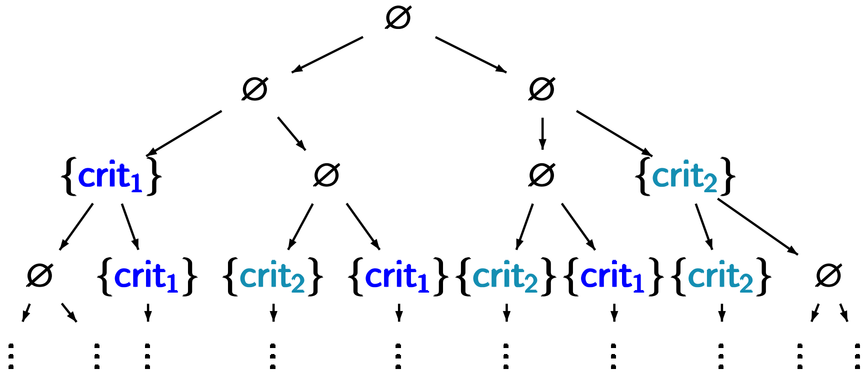
Example: computation tree

mutual exclusion with semaphore and $AP = \{crit_1, crit_2\}$:



Example: computation tree

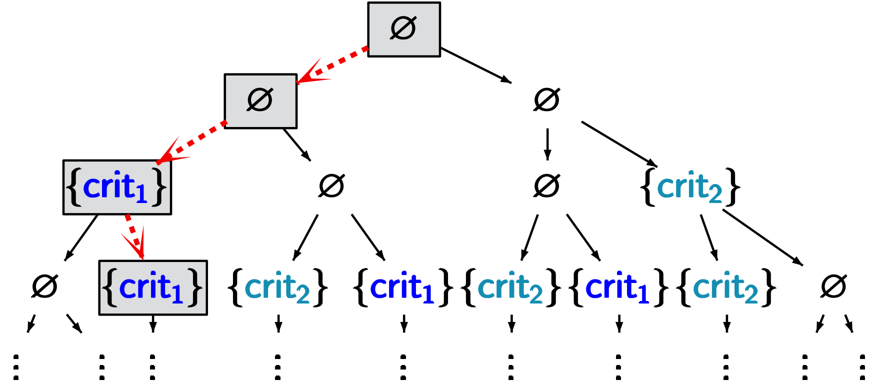
mutual exclusion with semaphore and $AP = \{crit_1, crit_2\}$:



path $\langle nc_1, nc_2 \rangle$ $\langle wait_1, nc_2 \rangle$ $\langle crit_1, nc_2 \rangle$ $\langle crit_1, wait_2 \rangle$ ...

Example: computation tree

mutual exclusion with semaphore and $AP = \{crit_1, crit_2\}$:



path	$\langle nc_1, nc_2 \rangle$	$\langle wait_1, nc_2 \rangle$	$\langle crit_1, nc_2 \rangle$	$\langle crit_1, wait_2 \rangle$	...
↓	↓	↓	↓	↓	
trace	$\varnothing$	$\varnothing$	$\{crit_1\}$	$\{crit_1\}$	...

Linear vs. branching time

	linear time	branching time
behavior	path based	state based

Linear vs. branching time

	linear time	branching time
behavior	path based traces	state based computation tree

Linear vs. branching time

	linear time	branching time
behavior	path based traces	state based computation tree
temporal logic	**LTL** path formulas	**CTL** state formulas

Linear vs. branching time

	linear time	branching time
behavior	path based traces	state based computation tree
temporal logic	**LTL** path formulas	**CTL** state formulas
model checking	PSPACE-complete $\mathcal{O}\big(\mathit{size}(\mathcal{T}) \cdot \exp(\lvert\varphi\rvert)\big)$	PTIME $\mathcal{O}\big(\mathit{size}(\mathcal{T}) \cdot \lvert\Phi\rvert\big)$

Linear vs. branching time

	linear time	branching time
behavior	path based traces	state based computation tree
temporal logic	**LTL** path formulas	**CTL** state formulas
model checking	PSPACE-complete $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \exp(\lvert\varphi\rvert))$	PTIME $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \lvert\Phi\rvert)$
impl. relation	trace inclusion trace equivalence PSPACE-complete	simulation bisimulation PTIME

Linear vs. branching time

	linear time	branching time
behavior	path based traces	state based computation tree
temporal logic	**LTL** path formulas	**CTL** state formulas
model checking	PSPACE-complete $\mathcal{O}\big(\mathit{size}(\mathcal{T}) \cdot \exp(\lvert\varphi\rvert)\big)$	PTIME $\mathcal{O}\big(\mathit{size}(\mathcal{T}) \cdot \lvert\Phi\rvert\big)$
impl. relation	trace inclusion trace equivalence PSPACE-complete	simulation bisimulation PTIME
fairness	can be encoded	requires special treatment

Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

Linear Temporal Logic (LTL)

Computation Tree Logic

- syntax and semantics of CTL ⟵
- expressiveness of CTL and LTL
- CTL model checking
- fairness, counterexamples/witnesses
- CTL^+ and CTL^*

Equivalences and Abstraction

Computation Tree Logic (CTL)

Computation Tree Logic (CTL)

CTL (state) formulas:

$$\Phi ::= \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2$$

CTL (state) formulas:

$$\Phi \quad ::= \quad \mathit{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \exists\varphi \;\big|\; \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \;\big|\; \Phi_1 \,\mathsf{U}\, \Phi_2$$

eventually:

$$\exists\Diamond\Phi \quad \stackrel{\text{def}}{=} \quad \exists(\mathit{true}\,\mathsf{U}\,\Phi)$$

CTL (state) formulas:

$$\Phi \quad ::= \quad \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2$$

eventually:

$$\exists\Diamond\Phi \quad \stackrel{\text{def}}{=} \quad \exists(\mathit{true}\,\mathsf{U}\,\Phi)$$

$$\forall\Diamond\Phi \quad \stackrel{\text{def}}{=} \quad ?$$

CTL (state) formulas:

$$\Phi \quad ::= \quad \mathit{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \exists\varphi \;\Big|\; \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \;\Big|\; \Phi_1 \,\mathsf{U}\, \Phi_2$$

eventually:

$$\exists\Diamond\Phi \quad \overset{\text{def}}{=} \quad \exists(\mathit{true}\,\mathsf{U}\,\Phi)$$

$$\forall\Diamond\Phi \quad \overset{\text{def}}{=} \quad \forall(\mathit{true}\,\mathsf{U}\,\Phi)$$

Computation Tree Logic (CTL)

CTL (state) formulas:

$$\Phi \ ::= \ \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \ ::= \ \bigcirc\Phi \mid \Phi_1 \mathsf{U} \Phi_2$$

eventually:

$$\exists\Diamond\Phi \ \stackrel{\text{def}}{=} \ \exists(\textit{true} \, \mathsf{U} \, \Phi)$$

$$\forall\Diamond\Phi \ \stackrel{\text{def}}{=} \ \forall(\textit{true} \, \mathsf{U} \, \Phi)$$

always:

$$\exists\Box\Phi \ \stackrel{\text{def}}{=} \ ?$$

Computation Tree Logic (CTL)

CTL (state) formulas:

$$\Phi \quad ::= \quad \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2$$

eventually:

$$\exists\Diamond\Phi \overset{\text{def}}{=} \exists(\mathit{true}\,\mathsf{U}\,\Phi)$$

$$\forall\Diamond\Phi \overset{\text{def}}{=} \forall(\mathit{true}\,\mathsf{U}\,\Phi)$$

always:

$$\exists\Box\Phi \overset{\text{def}}{=} ?$$

note: $\exists\neg\Diamond\neg\Phi$ is <u>no</u> **CTL** formula

Computation Tree Logic (CTL)

CTL (state) formulas:

$$\Phi \quad ::= \quad \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2$$

eventually:

$$\exists\Diamond\Phi \overset{\text{def}}{=} \exists(\mathit{true} \, \mathsf{U} \, \Phi)$$

$$\forall\Diamond\Phi \overset{\text{def}}{=} \forall(\mathit{true} \, \mathsf{U} \, \Phi)$$

always:

$$\exists\Box\Phi \overset{\text{def}}{=} \neg\forall\Diamond\neg\Phi$$

note: $\exists\neg\Diamond\neg\Phi$ is <u>no</u> **CTL** formula

CTL (state) formulas:

$$\Phi \quad ::= \quad \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2$$

eventually:

$$\exists\Diamond\Phi \stackrel{\text{def}}{=} \exists(\textit{true} \, \mathsf{U} \, \Phi)$$

$$\forall\Diamond\Phi \stackrel{\text{def}}{=} \forall(\textit{true} \, \mathsf{U} \, \Phi)$$

always:

$$\exists\Box\Phi \stackrel{\text{def}}{=} \neg\forall\Diamond\neg\Phi$$

$$\forall\Box\Phi \stackrel{\text{def}}{=} \neg\exists\Diamond\neg\Phi$$

note: $\exists\neg\Diamond\neg\Phi$ is <u>no</u> **CTL** formula

Syntax of CTL

CTL (state) formulas:

$$\Phi \quad ::= \quad \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2$$

$\bigcirc \quad \widehat{=} \quad$ next

$\mathsf{U} \quad \widehat{=} \quad$ until

Syntax of CTL

CTL (state) formulas:

$$\Phi \quad ::= \quad \textit{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \exists\varphi \;\big|\; \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \;\big|\; \Phi_1 \mathsf{U} \Phi_2 \;\big|\; \Diamond\Phi \;\big|\; \Box\Phi$$

$\bigcirc$	$\widehat{=}$	next	$\Diamond$	$\widehat{=}$	eventually
U	$\widehat{=}$	until	$\Box$	$\widehat{=}$	always

Examples for CTL formulas

CTL (state) formulas:

$$\Phi \ ::= \ \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \ ::= \ \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

mutual exclusion (safety) $\forall\Box(\neg \mathit{crit}_1 \vee \neg \mathit{crit}_2)$

CTL (state) formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \ ::= \ \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

mutual exclusion (safety) $\forall\Box(\neg crit_1 \vee \neg crit_2)$

"every request will be answered eventually"

$$\forall\Box(\, request \rightarrow \forall\Diamond response \,)$$

Examples for CTL formulas

CTL (state) formulas:

$$\Phi \ ::= \ \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \ ::= \ \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

mutual exclusion (safety) $\forall\Box(\neg crit_1 \vee \neg crit_2)$

"every request will be answered eventually"

$\forall\Box(\, request \rightarrow \forall\Diamond response\,)$

traffic lights $\forall\Box(\, yellow \rightarrow \forall\bigcirc red\,)$

Examples for CTL formulas

CTL (state) formulas:

$$\Phi \quad ::= \quad \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

mutual exclusion (safety) $\forall\Box(\neg crit_1 \vee \neg crit_2)$

"every request will be answered eventually"

$\forall\Box(\mathit{request} \rightarrow \forall\Diamond\mathit{response})$

traffic lights $\forall\Box(\mathit{yellow} \rightarrow \forall\bigcirc\mathit{red})$

reset possiblity $\forall\Box\exists\Diamond\mathit{start}$

CTL (state) formulas:

$$\Phi \ ::= \ \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

CTL path formulas:

$$\varphi \ ::= \ \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

mutual exclusion (safety) $\forall\Box(\neg crit_1 \vee \neg crit_2)$

"every request will be answered eventually"

$\forall\Box(\,request \rightarrow \forall\Diamond response\,)$

traffic lights $\forall\Box(\,yellow \rightarrow \forall\bigcirc red\,)$

reset possiblity $\forall\Box\exists\Diamond start$

unconditional process fairness $\forall\Box\forall\Diamond crit_1 \wedge \forall\Box\forall\Diamond crit_2$

Example: 15-puzzle

6	8	2	12
4	1	13	5
	9	10	14
7	11	15	3

$\overset{?}{\rightsquigarrow}$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Example: 15-puzzle

6	8	2	12
4	1	13	5
	9	10	14
7	11	15	3

$\overset{?}{\rightsquigarrow}$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

transition system has $\mathbf{16!} \approx \mathbf{2 \cdot 10^{13}}$ states

Example: 15-puzzle

6	8	2	12
4	1	13	5
	9	10	14
7	11	15	3

$\overset{?}{\rightsquigarrow}$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

transition system has $\mathbf{16!} \approx \mathbf{2 \cdot 10^{13}}$ states

$\uparrow$

states:	game configurations
transitions:	legal moves

Example: 15-puzzle

6	8	2	12
4	1	13	5
	9	10	14
7	11	15	3

$\overset{?}{\rightsquigarrow}$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

- transition system has $16! \approx 2 \cdot 10^{13}$ states
- representation as parallel system:

  $\mathit{left} \parallel \mathit{up} \parallel \mathit{down} \parallel \mathit{right}$

  with shared variables $\mathit{field}[i]$ for $i = 1, \ldots, 16$

Example: 15-puzzle

6	8	2	12
4	1	13	5
	9	10	14
7	11	15	3

$\overset{?}{\rightsquigarrow}$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

- transition system has **16!** $\approx 2 \cdot 10^{13}$ states
- representation as parallel system:

  $$\mathit{left} \parallel \mathit{up} \parallel \mathit{down} \parallel \mathit{right}$$

  with shared variables $\mathit{field}[i]$ for $i = 1, \ldots, 16$

CTL specification:

$$\exists\Diamond \bigwedge_{1 \leq i \leq 15} \text{“}\mathit{piece}\ i \text{ on } \mathit{field}[i]\text{”}$$

Example: 15-puzzle

6	8	2	12
4	1	13	5
	9	10	14
7	11	15	3

$\overset{?}{\rightsquigarrow}$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

- transition system has $16! \approx 2 \cdot 10^{13}$ states
- representation as parallel system:

  $$\textit{left} \,\|\, \textit{up} \,\|\, \textit{down} \,\|\, \textit{right}$$

  with shared variables $\textit{field}[i]$ for $i = 1, \ldots, 16$

CTL specification: seeking for a **witness** for

$$\exists\Diamond \bigwedge_{1 \leq i \leq 15} \text{"}\textit{piece } i \text{ on } \textit{field}[i]\text{"}$$

Semantics of CTL

Semantics of CTL

define a satisfaction relation $\models$ for CTL formulas over AP and a given TS $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$

Semantics of CTL

define a satisfaction relation $\models$ for CTL formulas over AP and a given TS $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ without terminal states *define* a satisfaction relation $\models$ for CTL formulas over AP and a given TS $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ without terminal states

- interpretation of state formulas over the states
- interpretation of path formulas over the paths (infinite path fragments)

Recall: semantics of LTL

for infinite path fragment $\pi = s_0\, s_1\, s_2\, \ldots$:

$\pi \models \mathit{true}$		
$\pi \models a$	iff	$s_0 \models a$,i.e., $a \in L(s_0)$
$\pi \models \varphi_1 \wedge \varphi_2$	iff	$\pi \models \varphi_1$ and $\pi \models \varphi_2$
$\pi \models \neg\varphi$	iff	$\pi \not\models \varphi$
$\pi \models \bigcirc\varphi$	iff	$\mathit{suffix}(\pi, 1) = s_1\, s_2\, s_3\, \ldots \models \varphi$
$\pi \models \varphi_1 \mathsf{U}\, \varphi_2$	iff	there exists $j \geq 0$ such that

$\mathit{suffix}(\pi, j) = s_j\ s_{j+1}\ s_{j+2}\ \ldots \models \varphi_2$ and

$\mathit{suffix}(\pi, k) = s_k\, s_{k+1}\, s_{k+2}\, \ldots \models \varphi_1$ for $0 \leq k < j$

Satisfaction relation for path formulas

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2\, \ldots$ be an infinite path fragment.

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2\, \ldots$ be an infinite path fragment.

$\pi \models \bigcirc \Phi$ iff $s_1 \models \Phi$

Let $\pi = s_0\, s_1\, s_2\, \ldots$ be an infinite path fragment.

$\pi \models \bigcirc \Phi$	iff	$s_1 \models \Phi$
$\pi \models \Phi_1 \,\mathsf{U}\, \Phi_2$	iff	there exists $j \geq 0$ such that
		$s_j \models \Phi_2$
		$s_k \models \Phi_1$ for $0 \leq k < j$

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2\, \ldots$ be an infinite path fragment.

$\pi \models \bigcirc \Phi$	iff	$s_1 \models \Phi$
$\pi \models \Phi_1 \,\mathsf{U}\, \Phi_2$	iff	there exists $j \geq 0$ such that
		$s_j \models \Phi_2$
		$s_k \models \Phi_1$ for $0 \leq k < j$

semantics of derived operators:

$\pi \models \Diamond \Phi$ iff there exists $j \geq 0$ with $s_j \models \Phi$

Satisfaction relation for path formulas

Let $\pi = s_0\, s_1\, s_2\, \ldots$ be an infinite path fragment.

$$\pi \models \bigcirc \Phi \quad \text{iff} \quad s_1 \models \Phi$$

$$\pi \models \Phi_1 \,\mathsf{U}\, \Phi_2 \quad \text{iff} \quad \text{there exists } j \geq 0 \text{ such that } \begin{array}{l} s_j \models \Phi_2 \\ s_k \models \Phi_1 \text{ for } 0 \leq k < j \end{array}$$

semantics of derived operators:

$$\pi \models \Diamond \Phi \quad \text{iff} \quad \text{there exists } j \geq 0 \text{ with } \quad s_j \models \Phi$$

$$\pi \models \Box \Phi \quad \text{iff} \quad \text{for all } j \geq 0 \text{ we have: } \quad s_j \models \Phi$$

Satisfaction relation for state formulas

Satisfaction relation for state formulas

$s \models \mathit{true}$

Satisfaction relation for state formulas

$s \models \mathit{true}$

$s \models a \quad$ iff $\quad a \in L(s)$

Satisfaction relation for state formulas

$s \models \mathit{true}$

$s \models a \quad$ iff $\quad a \in L(s)$

$s \models \Phi_1 \wedge \Phi_2 \quad$ iff $\quad s \models \Phi_1$ and $s \models \Phi_2$

Satisfaction relation for state formulas

$s \models \mathit{true}$		
$s \models a$	iff	$a \in L(s)$
$s \models \Phi_1 \wedge \Phi_2$	iff	$s \models \Phi_1$ and $s \models \Phi_2$
$s \models \neg\Phi$	iff	$s \not\models \Phi$

$s \models \mathit{true}$

$s \models a$ iff $a \in L(s)$

$s \models \Phi_1 \wedge \Phi_2$ iff $s \models \Phi_1$ and $s \models \Phi_2$

$s \models \neg\Phi$ iff $s \not\models \Phi$

$s \models \exists\varphi$ iff there is a path $\pi \in \mathit{Paths}(s)$ s.t. $\pi \models \varphi$

Satisfaction relation for state formulas

$s \models \mathit{true}$		
$s \models a$	iff	$a \in L(s)$
$s \models \Phi_1 \wedge \Phi_2$	iff	$s \models \Phi_1$ and $s \models \Phi_2$
$s \models \neg\Phi$	iff	$s \not\models \Phi$
$s \models \exists\varphi$	iff	there is a path $\pi \in \mathit{Paths}(s)$ s.t. $\pi \models \varphi$
$s \models \forall\varphi$	iff	for each path $\pi \in \mathit{Paths}(s)$: $\pi \models \varphi$

Satisfaction relation for state formulas

$s \models true$

$s \models a$ iff $a \in L(s)$

$s \models \Phi_1 \wedge \Phi_2$ iff $s \models \Phi_1$ and $s \models \Phi_2$

$s \models \neg\Phi$ iff $s \not\models \Phi$

$s \models \exists\varphi$ iff there is a path $\pi \in Paths(s)$ s.t. $\pi \models \varphi$

$s \models \forall\varphi$ iff for each path $\pi \in Paths(s)$: $\pi \models \varphi$

satisfaction set for state formula Φ:

$$Sat(\Phi) \stackrel{\text{def}}{=} \{s \in S : s \models \Phi\}$$

Interpretation of CTL formulas over a TS

Interpretation of CTL formulas over a TS

satisfaction of state formulas over a TS $\mathcal{T}$:

$$\mathcal{T} \models \Phi \quad \text{iff} \quad S_0 \subseteq Sat(\Phi)$$

where S_0 is the set of initial states

recall: $Sat(\Phi) = \{ s \in S : s \models \Phi \}$

satisfaction of state formulas over a TS $\mathcal{T}$:

$$\mathcal{T} \models \Phi \quad \text{iff} \quad S_0 \subseteq Sat(\Phi)$$
$$\text{iff} \quad s_0 \models \Phi \text{ for all initial states } s_0 \text{ of } \mathcal{T}$$

where S_0 is the set of initial states

recall: $Sat(\Phi) = \{s \in S : s \models \Phi\}$

Semantics of the next operator

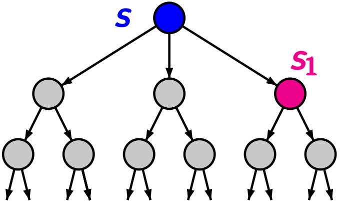
Semantics of the next operator

$s \models \exists \bigcirc \Phi$ iff there exists $\pi = s\, s_1\, s_2 \ldots \in \mathit{Paths}(s)$
s.t. $\pi \models \bigcirc \Phi$

Semantics of the next operator

$s \models \exists \bigcirc \Phi$ iff there exists $\pi = s\, s_1\, s_2 \ldots \in Paths(s)$
s.t. $\pi \models \bigcirc \Phi$, i.e., $s_1 \models \Phi$

$\exists \bigcirc \Phi$



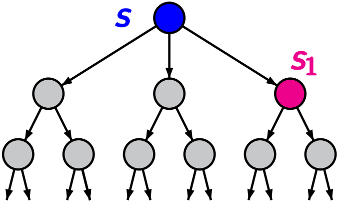
$Post(s) \cap Sat(\Phi) \neq \varnothing$

Semantics of the next operator

$s \models \exists\bigcirc\Phi$ iff there exists $\pi = s\, s_1\, s_2\, \ldots \in Paths(s)$ s.t. $\pi \models \bigcirc\Phi$, i.e., $s_1 \models \Phi$

$s \models \forall\bigcirc\Phi$ iff for all $\pi = s\, s_1\, s_2\, \ldots \in Paths(s)$: $\pi \models \bigcirc\Phi$

$\exists\bigcirc\Phi$



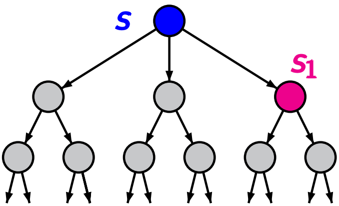
$Post(s) \cap Sat(\Phi) \neq \varnothing$

Semantics of the next operator

$s \models \exists \bigcirc \Phi$ iff there exists $\pi = s\, s_1\, s_2 \ldots \in Paths(s)$ s.t. $\pi \models \bigcirc \Phi$, i.e., $s_1 \models \Phi$

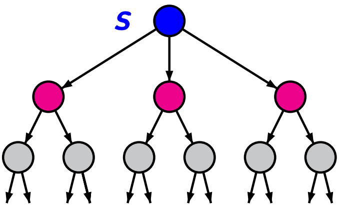
$s \models \forall \bigcirc \Phi$ iff for all $\pi = s\, s_1\, s_2 \ldots \in Paths(s)$: $\pi \models \bigcirc \Phi$, i.e., $s_1 \models \Phi$

$\exists \bigcirc \Phi$



$Post(s) \cap Sat(\Phi) \neq \varnothing$

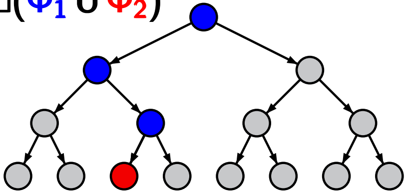
$\forall \bigcirc \Phi$



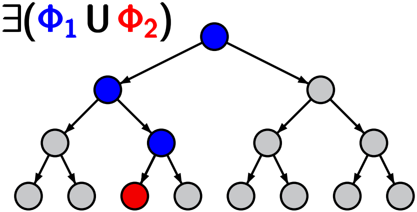
$Post(s) \subseteq Sat(\Phi)$

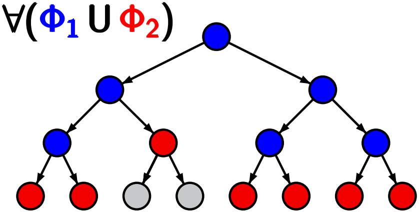
Semantics of until

$\exists(\Phi_1 \, \mathsf{U} \, \Phi_2)$



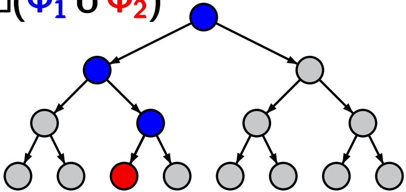
Semantics of until



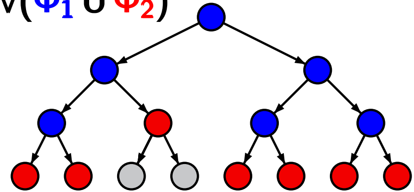


Semantics of until and eventually

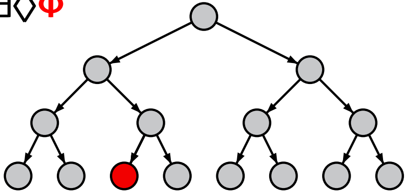
$\exists(\Phi_1 \, \mathsf{U} \, \Phi_2)$



$\forall(\Phi_1 \, \mathsf{U} \, \Phi_2)$

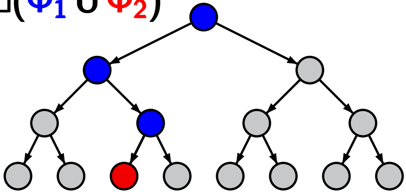


$\exists\Diamond\Phi$

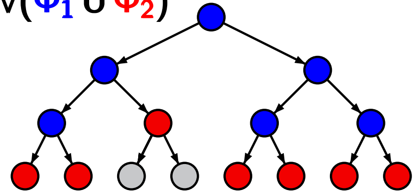


Semantics of until and eventually

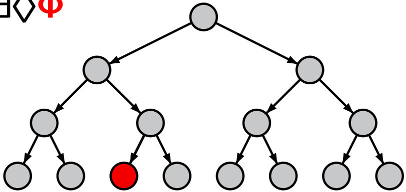
$\exists(\Phi_1 \, \mathsf{U} \, \Phi_2)$



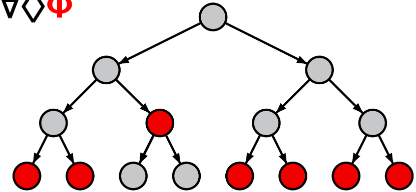
$\forall(\Phi_1 \, \mathsf{U} \, \Phi_2)$



$\exists\Diamond\Phi$

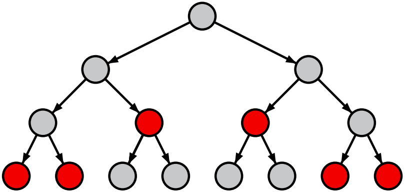


$\forall\Diamond\Phi$

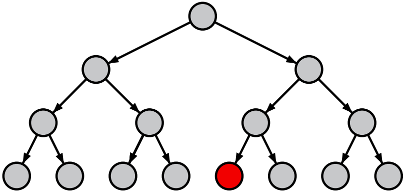


Semantics of eventually and always

$\forall\Diamond\Phi$

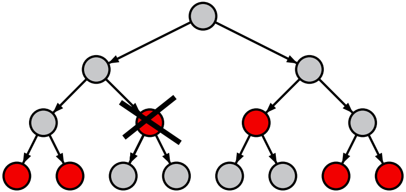


$\exists\Diamond\Phi$

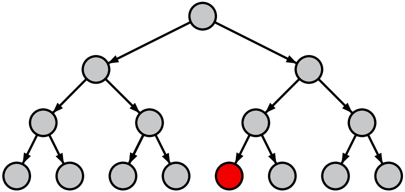


Semantics of eventually and always

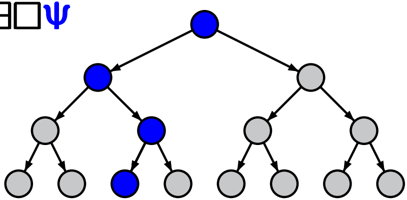
$\neg\forall\Diamond\Phi$



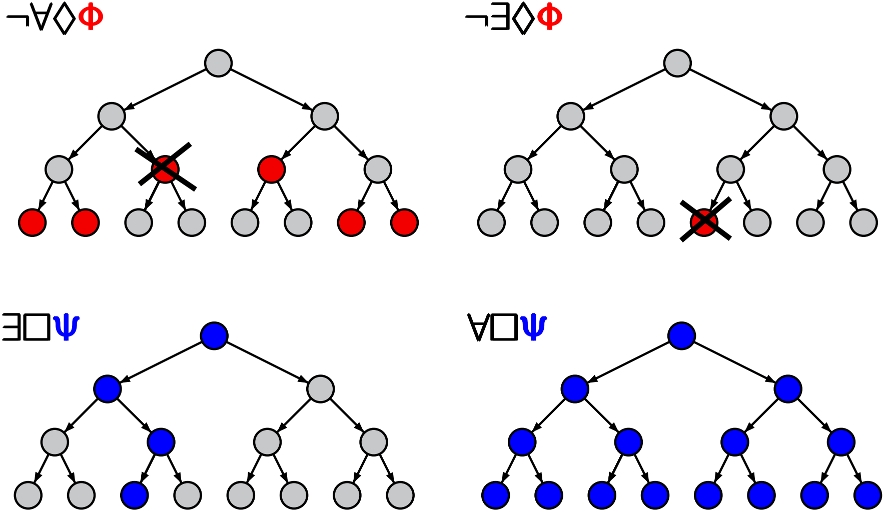
$\exists\Diamond\Phi$



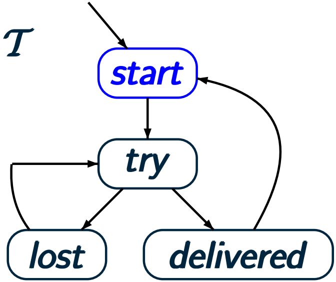
$\exists\Box\Psi$



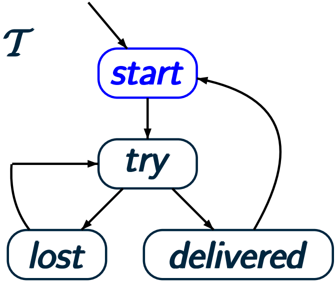
Semantics of eventually and always



Example for CTL semantics



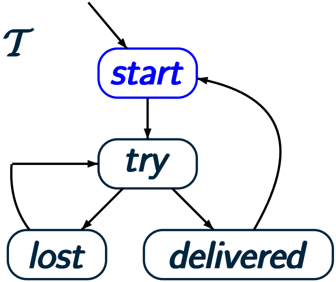
Example for CTL semantics



CTL formula

$\Phi = \forall\Box\forall\Diamond \textit{start}$

Example for CTL semantics

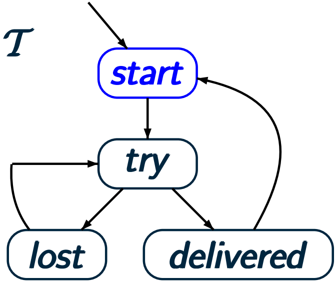


CTL formula

$\Phi = \forall\Box\ \boxed{\forall\Diamond start}$

$Sat(\forall\Diamond start) = \ ?$

Example for CTL semantics

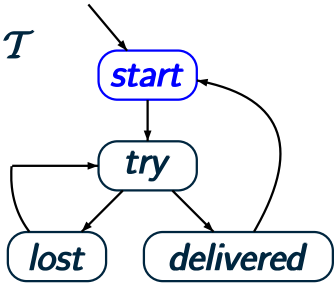


CTL formula

$\Phi = \forall\Box\ \boxed{\forall\Diamond start}$

$Sat(\forall\Diamond start) = \{start, delivered\}$

Example for CTL semantics

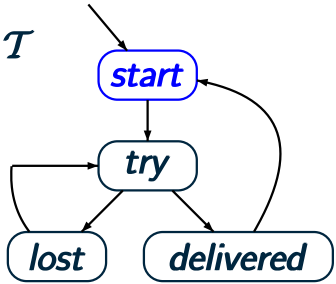


CTL formula

$\Phi = \forall\Box\forall\Diamond start \quad \hat{=} \quad \forall\Box(start \vee delivered)$

$Sat(\forall\Diamond start) = \{start, delivered\}$

Example for CTL semantics



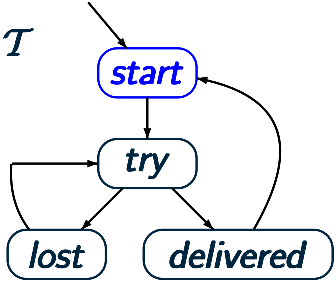
CTL formula

$$\Phi = \forall\Box\forall\Diamond start \;\hat{=}\; \forall\Box(start \vee delivered)$$

$$Sat(\forall\Diamond start) = \{start, delivered\}$$

$$Sat(\Phi) = \varnothing$$

Example for CTL semantics



$\mathcal{T} \not\models \forall\Box\forall\Diamond start$

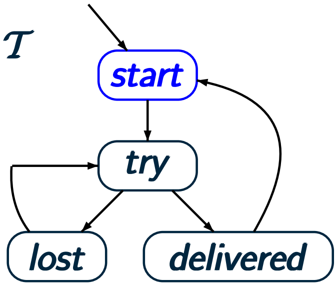
CTL formula

$\Phi = \forall\Box\forall\Diamond start \quad \widehat{=} \quad \forall\Box(start \vee delivered)$

$$Sat(\forall\Diamond start) = \{start, delivered\}$$

$$Sat(\Phi) = \varnothing$$

Example for CTL semantics



$\mathcal{T} \not\models \forall\Box\forall\Diamond \mathit{start}$

"infinitely often *start*"

CTL formula

$\Phi = \forall\Box\forall\Diamond \mathit{start} \quad \widehat{=} \quad \forall\Box(\mathit{start} \vee \mathit{delivered})$

$$\mathit{Sat}(\forall\Diamond \mathit{start}) = \{\mathit{start}, \mathit{delivered}\}$$

$$\mathit{Sat}(\Phi) = \varnothing$$

Specifying "infinitely often" in CTL

If s is a state in a TS and $a \in AP$ then:

$$s \models_{\text{CTL}} \forall\Box\forall\Diamond a$$

iff for all paths $\pi = s_0\, s_1\, s_2 \ldots \in Paths(s)$:

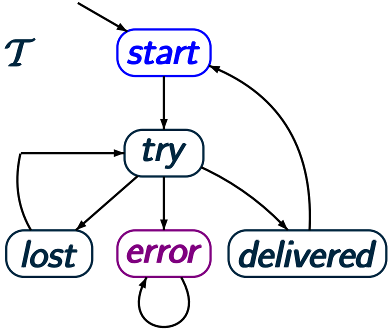
$$\overset{\infty}{\exists}\, i \geq 0. \quad \text{s.t.} \quad s_i \models a$$

Specifying "infinitely often" in CTL

If s is a state in a TS and $a \in AP$ then:

	$s \models_{\text{CTL}} \forall\Box\forall\Diamond a$
iff	for all paths $\pi = s_0\, s_1\, s_2 \ldots \in Paths(s)$:
	$\overset{\infty}{\exists}\, i \geq 0.$ s.t. $s_i \models a$
iff	$s \models_{\text{LTL}} \Box\Diamond a$

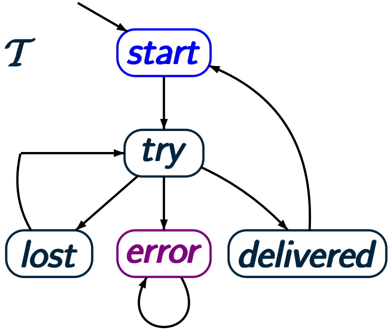
Example: CTL semantics



$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$ **?**

$\Phi_1 = \exists\Diamond\forall\Box\neg start$

Example: CTL semantics

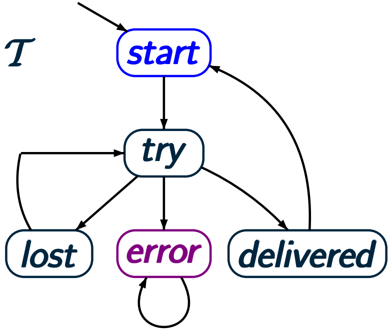


$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$ **?**

$\Phi_1 = \exists\Diamond \boxed{\forall\Box\neg start}$

$Sat(\forall\Box\neg start) = \{error\}$

Example: CTL semantics



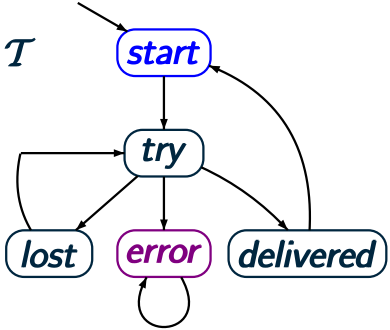
$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$?

$$\Phi_1 = \exists\Diamond \boxed{\forall\Box\neg start} \rightsquigarrow \exists\Diamond \boxed{error}$$

$$Sat(\forall\Box\neg start) = \{error\}$$

$$Sat(\exists\Diamond\forall\Box\neg start) = \text{?}$$

Example: CTL semantics



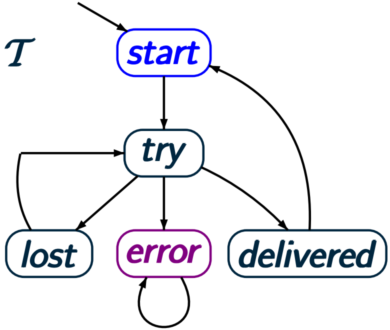
$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$ ✓

$\Phi_1 = \exists\Diamond \boxed{\forall\Box\neg start} \rightsquigarrow \exists\Diamond \boxed{error}$

$Sat(\forall\Box\neg start) = \{error\}$

$Sat(\exists\Diamond\forall\Box\neg start) = Sat(\exists\Diamond error) =$ "all states"

Example: CTL semantics

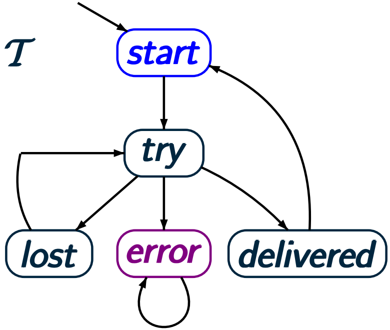


$\Phi_2 = \forall \bigcirc \exists \bigcirc \forall \Box \neg start$

$\mathcal{T} \models \exists \Diamond \forall \Box \neg start$

$\mathcal{T} \models \forall \bigcirc \exists \bigcirc \forall \Box \neg start$ **?**

Example: CTL semantics

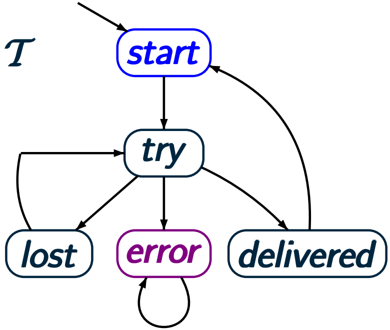


$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$ **?**

$\Phi_2 = \forall\bigcirc\exists\bigcirc \boxed{\forall\Box\neg start}$

$$Sat(\forall\Box\neg start) = \{error\}$$



$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$?

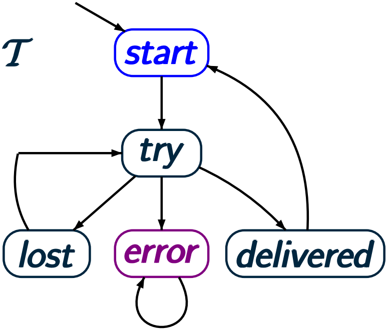
$\Phi_2 = \forall\bigcirc \; \boxed{\exists\bigcirc\forall\Box\neg start} \quad \rightsquigarrow \quad \forall\bigcirc \; \boxed{\exists\bigcirc error}$

$$Sat(\forall\Box\neg start) = \{error\}$$

$$Sat(\exists\bigcirc\forall\Box\neg start) = \{error, try\}$$

Example: CTL semantics

$\mathcal{T}$



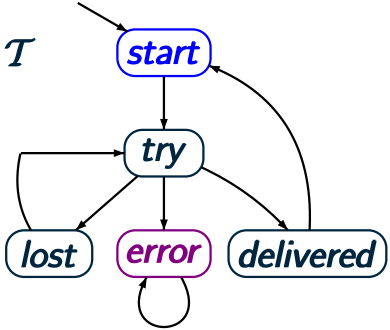
$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$?

$\Phi_2 = \forall\bigcirc\exists\bigcirc\forall\Box\neg start \quad \rightsquigarrow \quad \forall\bigcirc(error \vee try)$

$$Sat(\forall\Box\neg start) = \{error\}$$
$$Sat(\exists\bigcirc\forall\Box\neg start) = \{error, try\}$$
$$Sat(\forall\bigcirc\exists\bigcirc\forall\Box\neg start) = ?$$

Example: CTL semantics



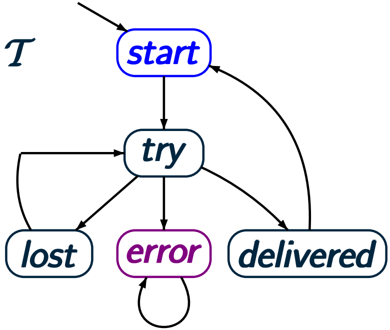
$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$ $\surd$

$\Phi_2 = \forall\bigcirc\exists\bigcirc\forall\Box\neg start \rightsquigarrow \forall\bigcirc(error \vee try)$

$$Sat(\forall\Box\neg start) = \{error\}$$
$$Sat(\exists\bigcirc\forall\Box\neg start) = \{error, try\}$$
$$Sat(\forall\bigcirc\exists\bigcirc\forall\Box\neg start) = \{error, lost, start\}$$

Example: CTL semantics



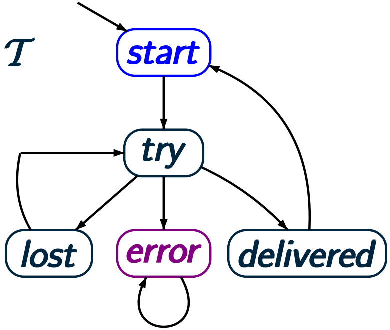
$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$

$\mathcal{T} \models \exists\bigcirc\forall\bigcirc\forall\Box\neg start$ **?**

$\Phi_3 = \exists\bigcirc\forall\bigcirc\forall\Box\neg start$

Example: CTL semantics



$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

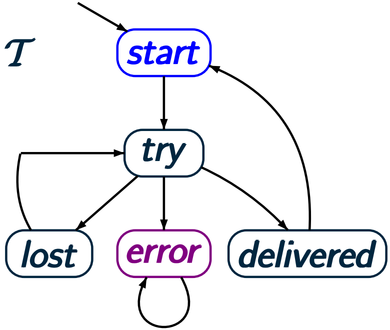
$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$

$\mathcal{T} \models \exists\bigcirc\forall\bigcirc\forall\Box\neg start$ **?**

$\Phi_3 = \exists\bigcirc\forall\bigcirc \boxed{\forall\Box\neg start}$

$Sat(\forall\Box\neg start) = \{error\}$

Example: CTL semantics



$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$

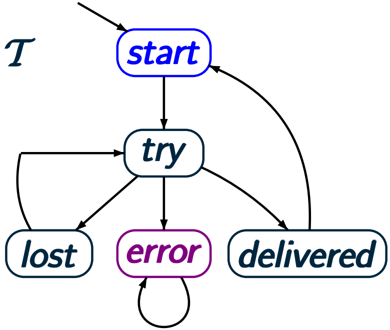
$\mathcal{T} \models \exists\bigcirc\forall\bigcirc\forall\Box\neg start$ **?**

$\Phi_3 = \exists\bigcirc \boxed{\forall\bigcirc\forall\Box\neg start} \rightsquigarrow \exists\bigcirc \boxed{\forall\bigcirc error}$

$$Sat(\forall\Box\neg start) = \{error\}$$

$$Sat(\forall\bigcirc\forall\Box\neg start) = \{error\}$$

Example: CTL semantics



$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$

$\mathcal{T} \models \exists\bigcirc\forall\bigcirc\forall\Box\neg start$?

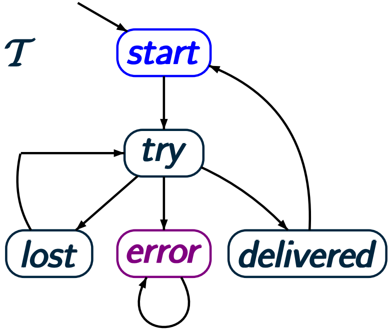
$\Phi_3 = \exists\bigcirc\forall\bigcirc\forall\Box\neg start \rightsquigarrow \boxed{\exists\bigcirc error}$

$$Sat(\forall\Box\neg start) = \{error\}$$

$$Sat(\forall\bigcirc\forall\Box\neg start) = \{error\}$$

$$Sat(\exists\bigcirc\forall\bigcirc\forall\Box\neg start) = \ ?$$

Example: CTL semantics



$\mathcal{T} \models \exists\Diamond\forall\Box\neg start$

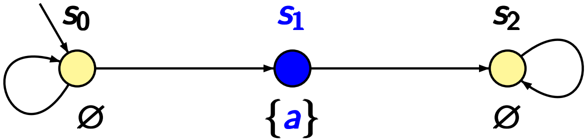
$\mathcal{T} \models \forall\bigcirc\exists\bigcirc\forall\Box\neg start$

$\mathcal{T} \not\models \exists\bigcirc\forall\bigcirc\forall\Box\neg start$

$\Phi_3 = \exists\bigcirc\forall\bigcirc\forall\Box\neg start \rightsquigarrow \boxed{\exists\bigcirc error}$

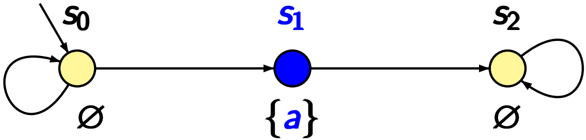
$$Sat(\forall\Box\neg start) = \{error\}$$
$$Sat(\forall\bigcirc\forall\Box\neg start) = \{error\}$$
$$Sat(\exists\bigcirc\forall\bigcirc\forall\Box\neg start) = \{error, try\}$$

Example: CTL semantics



does $\mathcal{T} \models \exists \bigcirc \forall \Box \neg a$ hold ?

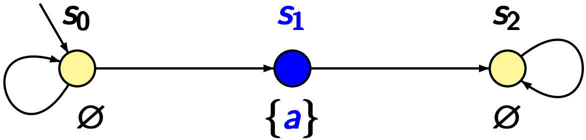
Example: CTL semantics



does $\mathcal{T} \models \exists \bigcirc \forall \Box \neg a$ hold ?

answer: no

Example: CTL semantics

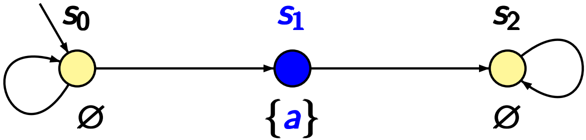


does $\mathcal{T} \models \exists\bigcirc\forall\Box\neg a$ hold ?

answer: no

$$Sat(\forall\Box\neg a) \;=\; \{s_2\}$$

Example: CTL semantics

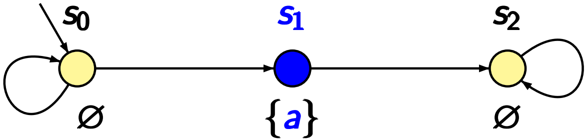


does $\mathcal{T} \models \exists\bigcirc\forall\Box\neg a$ hold ?

answer: no

$$Sat(\forall\Box\neg a) = \{s_2\}$$
$$Sat(\exists\bigcirc\forall\Box\neg a) = \{s_2, s_1\}$$

Example: CTL semantics

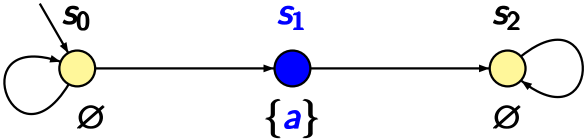


does $\mathcal{T} \models \exists\bigcirc\forall\Box\neg a$ hold ?

answer: no

does $\mathcal{T} \models \forall\Box\exists\bigcirc\neg a$ hold ?

Example: CTL semantics



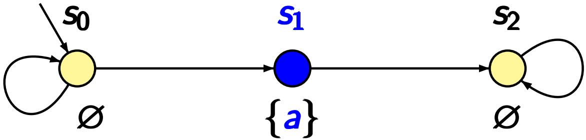
does $\mathcal{T} \models \exists\bigcirc\forall\Box\neg a$ hold ?

answer: no

does $\mathcal{T} \models \forall\Box\exists\bigcirc\neg a$ hold ?

answer: yes

Example: CTL semantics



does $\mathcal{T} \models \exists\bigcirc\forall\Box\neg a$ hold ?

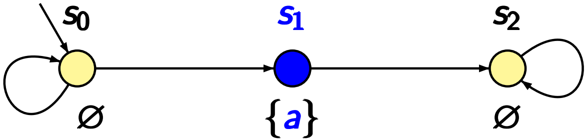
answer: no

does $\mathcal{T} \models \forall\Box\exists\bigcirc\neg a$ hold ?

answer: yes

$$Sat(\exists\bigcirc\neg a) = \{s_0, s_1, s_2\}$$

Example: CTL semantics



does $\mathcal{T} \models \exists\bigcirc\forall\Box\neg a$ hold ?

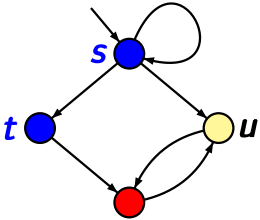
answer: no

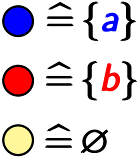
does $\mathcal{T} \models \forall\Box\exists\bigcirc\neg a$ hold ?

answer: yes

$$Sat(\exists\bigcirc\neg a) = \{s_0, s_1, s_2\}$$
$$Sat(\forall\Box\exists\bigcirc\neg a) = \{s_0, s_1, s_2\}$$

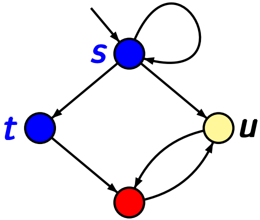
Example: CTL semantics

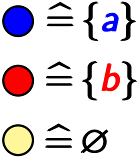




$\mathcal{T} \models \exists\Box\exists(a \,\mathsf{U}\, b)$?

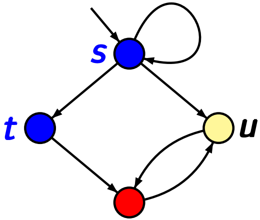
Example: CTL semantics

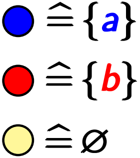




$\mathcal{T} \models \exists\Box\exists(a \,\mathsf{U}\, b)$ √ as $s \models \exists(a \,\mathsf{U}\, b)$

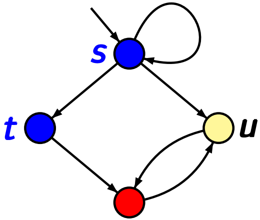
Example: CTL semantics

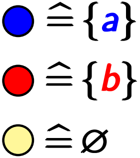




$\mathcal{T} \models \exists\Box\exists(a \,\mathsf{U}\, b)$ $\quad\checkmark\quad$ as $s\,s\,s\,\ldots \models \Box\exists(a \,\mathsf{U}\, b)$

Example: CTL semantics

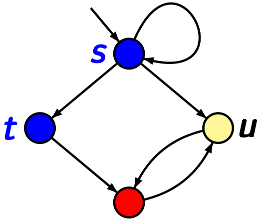


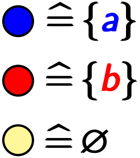


$\mathcal{T} \models \exists\Box\exists(a \mathsf{U} b)$ ✓ as $s\,s\,s\,\ldots \models \Box\exists(a \mathsf{U} b)$

$\mathcal{T} \models \exists((\exists\bigcirc a) \mathsf{U} b)$?

Example: CTL semantics

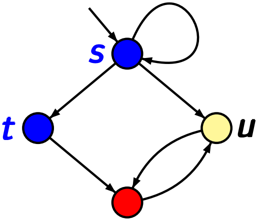


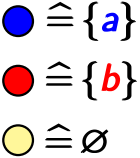


$\mathcal{T} \models \exists\Box\exists(a \,\mathsf{U}\, b)$ $\surd$ as $s\,s\,s\,\ldots \models \Box\exists(a \,\mathsf{U}\, b)$

$\mathcal{T} \not\models \exists((\exists\bigcirc a) \,\mathsf{U}\, b)$ as $t \not\models \exists\bigcirc a$, $u \not\models \exists\bigcirc a$

Example: CTL semantics



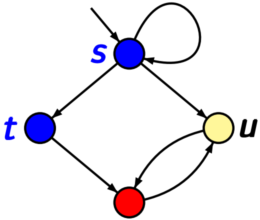


$\mathcal{T} \models \exists\Box\exists(a \,\mathsf{U}\, b)$ $\quad\surd\quad$ as $s\,s\,s\,\ldots \models \Box\exists(a \,\mathsf{U}\, b)$

$\mathcal{T} \not\models \exists((\exists\bigcirc a) \,\mathsf{U}\, b)$ $\quad$ as $t \not\models \exists\bigcirc a$, $u \not\models \exists\bigcirc a$

$\mathcal{T} \models \exists(a \,\mathsf{U}\, \forall(\neg a \,\mathsf{U}\, b))$ $\quad ?$

Example: CTL semantics

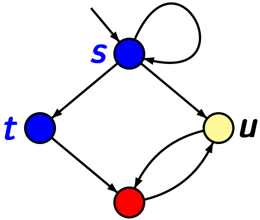


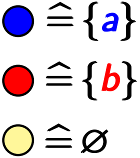
$\mathcal{T} \models \exists\Box\exists(a \,\mathsf{U}\, b)$ ✓ as $s\,s\,s \ldots \models \Box\exists(a \,\mathsf{U}\, b)$

$\mathcal{T} \not\models \exists((\exists\bigcirc a) \,\mathsf{U}\, b)$ as $t \not\models \exists\bigcirc a$, $u \not\models \exists\bigcirc a$

$\mathcal{T} \models \exists(a \,\mathsf{U}\, \forall(\neg a \,\mathsf{U}\, b))$ ✓

Example: CTL semantics

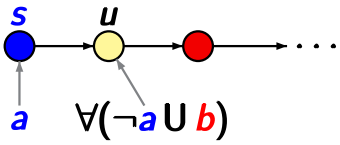




$\mathcal{T} \models \exists\Box\exists(a \mathsf{U} b)$ ✓ as $s\,s\,s \ldots \models \Box\exists(a \mathsf{U} b)$

$\mathcal{T} \not\models \exists((\exists\bigcirc a) \mathsf{U} b)$ as $t \not\models \exists\bigcirc a$, $u \not\models \exists\bigcirc a$

$\mathcal{T} \models \exists(a \mathsf{U} \forall(\neg a \mathsf{U} b))$ ✓



$\models a \mathsf{U} \forall(\neg a \mathsf{U} b)$

Correct or wrong?

Let $\mathcal{T}$ be a transition system and Φ a CTL formula.
Is the following statement correct **?**

$$\text{if} \quad \mathcal{T} \not\models \neg\Phi \quad \text{then} \quad \mathcal{T} \models \Phi$$

Let $\mathcal{T}$ be a transition system and Φ a CTL formula.
Is the following statement correct **?**

$$\text{if } \mathcal{T} \not\models \neg\Phi \text{ then } \mathcal{T} \models \Phi$$

answer: no

Correct or wrong?

Let $\mathcal{T}$ be a transition system and Φ a CTL formula.
Is the following statement correct **?**

$$\text{if } \mathcal{T} \not\models \neg\Phi \text{ then } \mathcal{T} \models \Phi$$

answer: no

$\mathcal{T} \models \Phi$ iff $s_0 \models \Phi$ for all initial states s_0

Correct or wrong?

Let $\mathcal{T}$ be a transition system and Φ a CTL formula.
Is the following statement correct **?**

if $\mathcal{T} \not\models \neg\Phi$ then $\mathcal{T} \models \Phi$

answer: no

$\mathcal{T} \models \Phi$	iff	$s_0 \models \Phi$ for all initial states s_0
$\mathcal{T} \not\models \neg\Phi$	iff	there exists an initial state s_0 with $s_0 \not\models \neg\Phi$

Let $\mathcal{T}$ be a transition system and Φ a CTL formula.
Is the following statement correct **?**

$$\text{if } \mathcal{T} \not\models \neg\Phi \text{ then } \mathcal{T} \models \Phi$$

answer: no

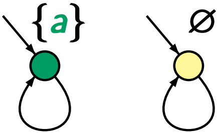
$\mathcal{T} \models \Phi$	iff	$s_0 \models \Phi$ for all initial states s_0
$\mathcal{T} \not\models \neg\Phi$	iff	there exists an initial state s_0 with $s_0 \not\models \neg\Phi$
	iff	there exists an initial state s_0 with $s_0 \models \Phi$

Let $\mathcal{T}$ be a transition system and Φ a CTL formula.
Is the following statement correct **?**

$$\text{if } \mathcal{T} \not\models \neg\Phi \text{ then } \mathcal{T} \models \Phi$$

answer: no

transition system $\mathcal{T}$ with 2 initial states:



$\mathcal{T} \not\models \exists\Box a$

$\mathcal{T} \not\models \neg\exists\Box a$

Hamilton path problem

Hamilton path problem

given: finite directed graph $G = (V, E)$

question: does G have a Hamilton path, i.e., a path that visits each node exactly once **?**

Hamilton path problem

given: finite directed graph $G = (V, E)$

question: does G have a Hamilton path, i.e., a path that visits each node exactly once **?**

goal: provide an encoding of the Hamilton path problem in **CTL**

Hamilton path problem

given: finite directed graph $G = (V, E)$

question: does G have a Hamilton path, i.e., a path that visits each node exactly once **?**

goal: provide an encoding of the Hamilton path problem in **CTL** by means of a transformation

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

Hamilton path problem

given: finite directed graph $G = (V, E)$

question: does G have a Hamilton path, i.e., a path that visits each node exactly once **?**

goal: provide an encoding of the Hamilton path problem in **CTL** by means of a transformation

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

s.t. G has a Hamilton path iff $\mathcal{T}_G \not\models \Phi$

CTL-encoding of the Hamilton path problem

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

s.t. G has a Hamilton path iff $\mathcal{T}_G \not\models \Phi$

CTL-encoding of the Hamilton path problem

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

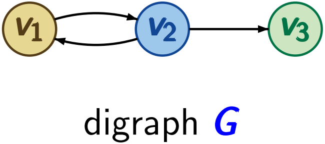
s.t. G has a Hamilton path iff $\mathcal{T}_G \not\models \Phi$

digraph G

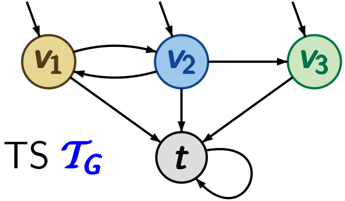
CTL-encoding of the Hamilton path problem

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

s.t. G has a Hamilton path iff $\mathcal{T}_G \not\models \Phi$



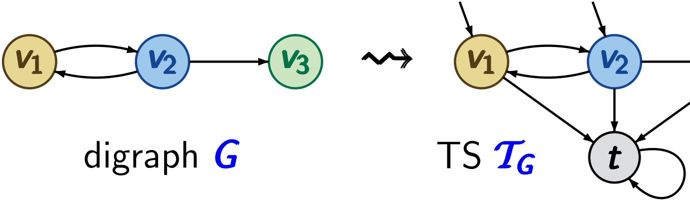
$\rightsquigarrow$



CTL-encoding of the Hamilton path problem

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

s.t. G has a Hamilton path iff $\mathcal{T}_G \not\models \Phi$



digraph G TS $\mathcal{T}_G$

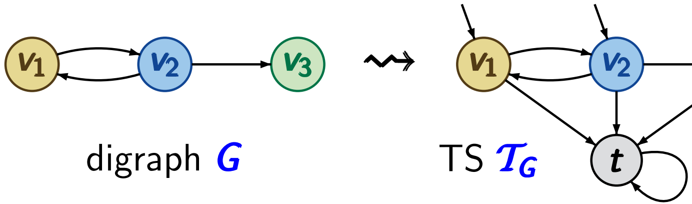
CTL formula Φ

$$(v_1 \wedge \exists\bigcirc(v_2 \wedge \exists\bigcirc v_3)) \vee (v_1 \wedge \exists\bigcirc(v_3 \wedge \exists\bigcirc v_2)) \vee$$
$$(v_2 \wedge \exists\bigcirc(v_1 \wedge \exists\bigcirc v_3)) \vee (v_2 \wedge \exists\bigcirc(v_3 \wedge \exists\bigcirc v_1)) \vee$$
$$(v_3 \wedge \exists\bigcirc(v_1 \wedge \exists\bigcirc v_2)) \vee (v_3 \wedge \exists\bigcirc(v_2 \wedge \exists\bigcirc v_1))$$

CTL-encoding of the Hamilton path problem

finite digraph G	$\rightsquigarrow$	finite TS $\mathcal{T}_G$ + CTL formula Φ

s.t. G has a Hamilton path iff $\mathcal{T}_G \not\models \Phi$



CTL formula Φ = negation of the formula

$$(v_1 \wedge \exists\bigcirc(v_2 \wedge \exists\bigcirc v_3)) \vee (v_1 \wedge \exists\bigcirc(v_3 \wedge \exists\bigcirc v_2)) \vee$$
$$(v_2 \wedge \exists\bigcirc(v_1 \wedge \exists\bigcirc v_3)) \vee (v_2 \wedge \exists\bigcirc(v_3 \wedge \exists\bigcirc v_1)) \vee$$
$$(v_3 \wedge \exists\bigcirc(v_1 \wedge \exists\bigcirc v_2)) \vee (v_3 \wedge \exists\bigcirc(v_2 \wedge \exists\bigcirc v_1))$$

Equivalence of CTL formulas

Equivalence of CTL formulas

$\Phi_1 \equiv \Phi_2$ iff for all transition systems $\mathcal{T}$:

$$\mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

$\Phi_1 \equiv \Phi_2$ iff for all transition systems $\mathcal{T}$:

$$\mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

quantification over all transition systems $\mathcal{T}$

- without terminal states
- over *AP* if Φ_1 and Φ_2 are CTL formulas over *AP*

Equivalence of CTL formulas

$\Phi_1 \equiv \Phi_2$ iff for all transition systems $\mathcal{T}$:

$$\mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

iff for all transition systems $\mathcal{T}$:

$$Sat(\Phi_1) = Sat(\Phi_2)$$

quantification over all transition systems $\mathcal{T}$

- without terminal states
- over AP if Φ_1 and Φ_2 are CTL formulas over AP

Equivalence of CTL formulas

$\Phi_1 \equiv \Phi_2$ iff for all transition systems $\mathcal{T}$:

$$\mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

iff for all transition systems $\mathcal{T}$:

$$Sat(\Phi_1) = Sat(\Phi_2)$$

Examples:

$$\neg\neg\Phi \equiv \Phi$$

$$\neg(\Phi \wedge \Psi) \equiv \neg\Phi \vee \neg\Psi$$

Equivalence of CTL formulas

$\Phi_1 \equiv \Phi_2$ iff for all transition systems $\mathcal{T}$:

$$\mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

iff for all transition systems $\mathcal{T}$:

$$Sat(\Phi_1) = Sat(\Phi_2)$$

Examples:

$$\neg\neg\Phi \equiv \Phi$$

$$\neg(\Phi \wedge \Psi) \equiv \neg\Phi \vee \neg\Psi$$

$$\vdots$$

$$\neg\forall\bigcirc\Phi \equiv \exists\bigcirc\neg\Phi$$

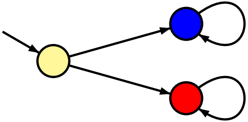
Correct or wrong?

$$\exists\Diamond(a \wedge b) \equiv \exists\Diamond a \wedge \exists\Diamond b$$

Correct or wrong?

$$\exists\Diamond(a \wedge b) \equiv \exists\Diamond a \wedge \exists\Diamond b$$

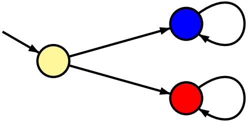
wrong, e.g,



Correct or wrong?

$$\exists\Diamond(a \wedge b) \;\equiv\; \exists\Diamond a \wedge \exists\Diamond b$$

wrong, e.g,

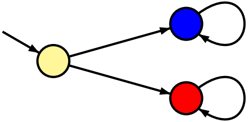


$$\forall\Diamond(a \wedge b) \;\equiv\; \forall\Diamond a \wedge \forall\Diamond b$$

Correct or wrong?

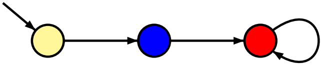
$\exists\Diamond(a \wedge b) \equiv \exists\Diamond a \wedge \exists\Diamond b$

wrong, e.g,



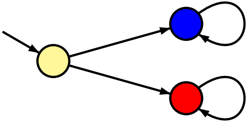
$\forall\Diamond(a \wedge b) \equiv \forall\Diamond a \wedge \forall\Diamond b$

wrong, e.g.,



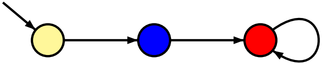
$$\exists\Diamond(a \wedge b) \equiv \exists\Diamond a \wedge \exists\Diamond b$$

wrong, e.g,



$$\forall\Diamond(a \wedge b) \equiv \forall\Diamond a \wedge \forall\Diamond b$$

wrong, e.g.,



but:

$$\forall\Box(\Phi_1 \wedge \Phi_2) \equiv \forall\Box\Phi_1 \wedge \forall\Box\Phi_2$$
$$\exists\Diamond(\Phi_1 \vee \Phi_2) \equiv \exists\Diamond\Phi_1 \vee \exists\Diamond\Phi_2$$

Correct or wrong?

$$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$$

Correct or wrong?

$$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$$

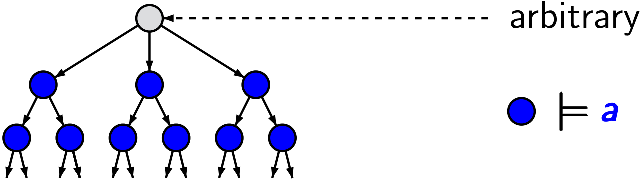
correct.

Correct or wrong?

$\forall\bigcirc\forall\Box a \;\equiv\; \forall\Box\forall\bigcirc a$

correct.

both formulas require computation trees of the form:



Correct or wrong?

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

Correct or wrong?

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

wrong,

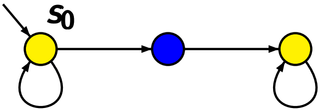
Correct or wrong?

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

wrong, e.g.,

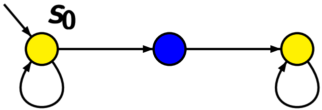


$\forall\bigcirc\forall\Box a \;\equiv\; \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \;\equiv\; \exists\Box\exists\bigcirc a$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

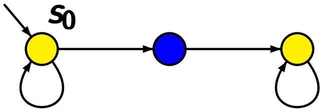
Correct or wrong?

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

note: $Sat(\exists\Box a) = \varnothing$

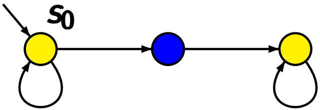
Correct or wrong?

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

$s_0 \models \exists\Box\exists\bigcirc a$

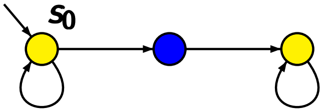
Correct or wrong?

$\forall\bigcirc\forall\Box a \;\equiv\; \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \;\equiv\; \exists\Box\exists\bigcirc a$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

$s_0 \models \exists\Box\exists\bigcirc a$

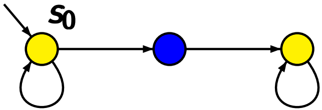
$s_0 \models \exists\bigcirc a$

$$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$$

correct.

$$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

$s_0 \models \exists\Box\exists\bigcirc a$

$s_0 \models \exists\bigcirc a$

$\Longrightarrow \quad s_0\, s_0\, s_0 \ldots \models \Box\exists\bigcirc a$

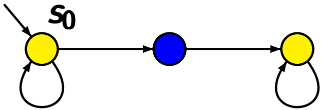
Correct or wrong?

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

$s_0 \models \exists\Box\exists\bigcirc a$

$$s_0 \models \exists\bigcirc a$$

$$\Longrightarrow \quad s_0\, s_0\, s_0 \ldots \models \Box\exists\bigcirc a$$

$$\Longrightarrow \quad s_0 \models \exists\Box\exists\bigcirc a$$

Weak until W

Weak until W

in **LTL**: $\varphi \, \mathsf{W} \, \psi \stackrel{\text{def}}{=} (\varphi \, \mathsf{U} \, \psi) \vee \Box\varphi$

in **CTL**: **?**

Weak until W

in **LTL**: $\varphi \, \mathsf{W} \, \psi \overset{\text{def}}{=} (\varphi \, \mathsf{U} \, \psi) \lor \Box\varphi$

duality of **U** and **W**:

$$\neg(\varphi \, \mathsf{U} \, \psi) \equiv (\varphi \land \neg\psi) \, \mathsf{W} \, (\neg\varphi \land \neg\psi)$$

$$\neg(\varphi \, \mathsf{W} \, \psi) \equiv (\varphi \land \neg\psi) \, \mathsf{U} \, (\neg\varphi \land \neg\psi)$$

in **CTL**: **?**

Weak until W

in **LTL**: $\varphi \, \mathbf{W} \, \psi \stackrel{\text{def}}{=} (\varphi \, \mathbf{U} \, \psi) \vee \Box\varphi$

duality of **U** and **W**:

$$\neg(\varphi \, \mathbf{U} \, \psi) \equiv (\varphi \wedge \neg\psi) \, \mathbf{W} \, (\neg\varphi \wedge \neg\psi)$$

$$\neg(\varphi \, \mathbf{W} \, \psi) \equiv (\varphi \wedge \neg\psi) \, \mathbf{U} \, (\neg\varphi \wedge \neg\psi)$$

definition of **W** in **CTL** on the basis of duality rules:

$$\exists(\Phi \, \mathbf{W} \, \Psi) \stackrel{\text{def}}{=} \neg\forall((\Phi \wedge \neg\Psi) \, \mathbf{U} \, (\neg\Phi \wedge \neg\Psi))$$

Weak until W

in **LTL**: $\varphi \, \mathsf{W} \, \psi \overset{\text{def}}{=} (\varphi \, \mathsf{U} \, \psi) \vee \Box\varphi$

duality of **U** and **W**:

$$\neg(\varphi \, \mathsf{U} \, \psi) \equiv (\varphi \wedge \neg\psi) \, \mathsf{W} \, (\neg\varphi \wedge \neg\psi)$$

$$\neg(\varphi \, \mathsf{W} \, \psi) \equiv (\varphi \wedge \neg\psi) \, \mathsf{U} \, (\neg\varphi \wedge \neg\psi)$$

definition of **W** in **CTL** on the basis of duality rules:

$$\exists(\Phi \, \mathsf{W} \, \Psi) \overset{\text{def}}{=} \neg\forall((\Phi \wedge \neg\Psi) \, \mathsf{U} \, (\neg\Phi \wedge \neg\Psi))$$

$$\forall(\Phi \, \mathsf{W} \, \Psi) \overset{\text{def}}{=} \neg\exists((\Phi \wedge \neg\Psi) \, \mathsf{U} \, (\neg\Phi \wedge \neg\Psi))$$

Weak until W in CTL

definition of **W** in **CTL** on the basis of duality rules:

$$\exists(\Phi \, \mathsf{W} \, \Psi) \stackrel{\text{def}}{=} \neg\forall\big((\Phi \wedge \neg\Psi) \, \mathsf{U} \, (\neg\Phi \wedge \neg\Psi) \big)$$

$$\forall(\Phi \, \mathsf{W} \, \Psi) \stackrel{\text{def}}{=} \neg\exists\big((\Phi \wedge \neg\Psi) \, \mathsf{U} \, (\neg\Phi \wedge \neg\Psi) \big)$$

Weak until W in CTL

definition of **W** in **CTL** on the basis of duality rules:

$$\exists(\Phi \, \mathsf{W} \, \Psi) \overset{\text{def}}{=} \neg\forall\big((\Phi \wedge \neg\Psi) \, \mathsf{U} \, (\neg\Phi \wedge \neg\Psi)\big)$$

$$\forall(\Phi \, \mathsf{W} \, \Psi) \overset{\text{def}}{=} \neg\exists\big((\Phi \wedge \neg\Psi) \, \mathsf{U} \, (\neg\Phi \wedge \neg\Psi)\big)$$

note that:

$$\exists(\Phi \, \mathsf{W} \, \Psi) \equiv \exists(\Phi \, \mathsf{U} \, \Psi) \vee \exists\Box\Phi$$

Weak until W in CTL

definition of **W** in **CTL** on the basis of duality rules:

$$\exists(\Phi \, W \, \Psi) \overset{\text{def}}{=} \neg\forall((\Phi \wedge \neg\Psi) \, U \, (\neg\Phi \wedge \neg\Psi))$$

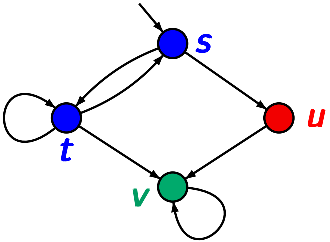
$$\forall(\Phi \, W \, \Psi) \overset{\text{def}}{=} \neg\exists((\Phi \wedge \neg\Psi) \, U \, (\neg\Phi \wedge \neg\Psi))$$

note that:

$$\exists(\Phi \, W \, \Psi) \equiv \exists(\Phi \, U \, \Psi) \vee \exists\Box\Phi$$

$$\forall(\Phi \, W \, \Psi) \not\equiv \forall(\Phi \, U \, \Psi) \vee \forall\Box\Phi$$

Weak until W in CTL



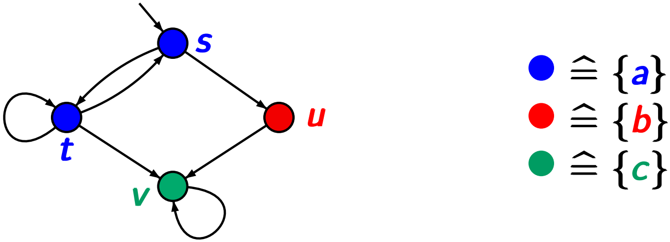
$\bullet \widehat{=} \{a\}$

$\bullet \widehat{=} \{b\}$

$\bullet \widehat{=} \{c\}$

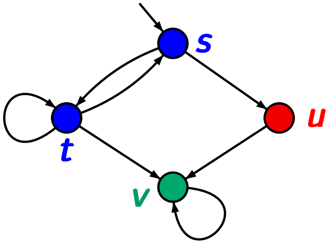
$\mathcal{T} \models \forall\Diamond\exists(a \text{ W } c)$ **?**

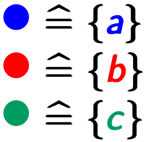
Weak until W in CTL



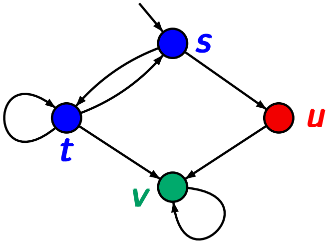
$\mathcal{T} \models \forall\Diamond\exists(a \mathsf{W} c)$ $\surd$ as $s \models \exists(a \mathsf{W} c)$

Weak until W in CTL





$\mathcal{T} \models \forall\Diamond\exists(a \; \mathsf{W} \; c)$ $\surd$ as $s\, s_1\, s_2 \ldots \models \Diamond\exists(a \; \mathsf{W} \; c)$

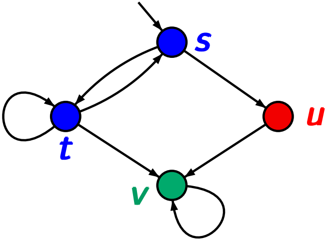


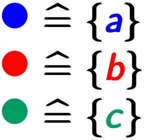
● $\widehat{=}$ $\{a\}$
● $\widehat{=}$ $\{b\}$
● $\widehat{=}$ $\{c\}$

$\mathcal{T} \models \forall\Diamond\exists(a \mathsf{W} c)$ √ as $s \models \exists(a \mathsf{W} c)$

$\mathcal{T} \models \exists(a \mathsf{W} \exists\Box b)$?

Weak until W in CTL

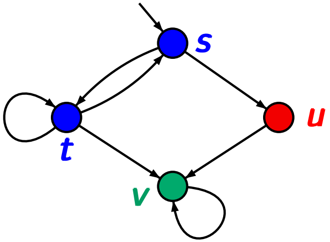




$\mathcal{T} \models \forall\Diamond\exists(a \;\mathsf{W}\; c)$ $\surd$ as $s \models \exists(a \;\mathsf{W}\; c)$

$\mathcal{T} \models \exists(a \;\mathsf{W}\; \exists\Box b)$ $\surd$ as $s \models \exists\Box a$

Weak until W in CTL



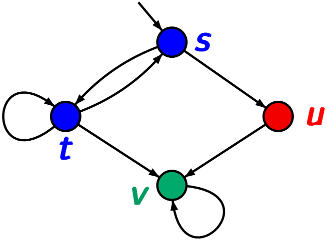
● $\widehat{=}$ $\{a\}$
● $\widehat{=}$ $\{b\}$
● $\widehat{=}$ $\{c\}$

$\mathcal{T} \models \forall\Diamond\exists(a \,\mathsf{W}\, c)$ √ as $s \models \exists(a \,\mathsf{W}\, c)$

$\mathcal{T} \models \exists(a \,\mathsf{W}\, \exists\Box b)$ √ as $s \models \exists\Box a$

$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \,\mathsf{W}\, (a \wedge b))$ **?**

Weak until W in CTL



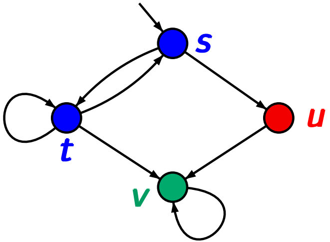
● $\widehat{=}$ $\{a\}$
● $\widehat{=}$ $\{b\}$
● $\widehat{=}$ $\{c\}$

$\mathcal{T} \models \forall\Diamond\exists(a \mathsf{W} c)$ ✓ as $s \models \exists(a \mathsf{W} c)$

$\mathcal{T} \models \exists(a \mathsf{W} \exists\Box b)$ ✓ as $s \models \exists\Box a$

$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \mathsf{W} (a \wedge b))$ ✓

Weak until W in CTL



● $\widehat{=}$ $\{a\}$
● $\widehat{=}$ $\{b\}$
● $\widehat{=}$ $\{c\}$

$\mathcal{T} \models \forall\Diamond\exists(a \mathsf{W} c)$ $\surd$ as $s \models \exists(a \mathsf{W} c)$

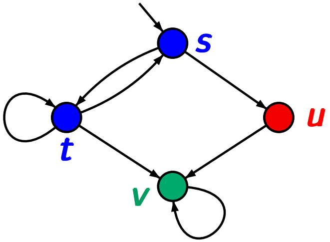
$\mathcal{T} \models \exists(a \mathsf{W} \exists\Box b)$ $\surd$ as $s \models \exists\Box a$

$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \mathsf{W} (a \wedge b))$ $\surd$

$\uparrow$

three types of paths: $(s\,t)^\omega$ or $(s\,t)^+ v^\omega$ or $(s\,t)^* s\,u\,v^\omega$

Weak until W in CTL



$\bullet \; \widehat{=} \; \{a\}$ (blue)
$\bullet \; \widehat{=} \; \{b\}$ (red)
$\bullet \; \widehat{=} \; \{c\}$ (green)

$\mathcal{T} \models \forall\Diamond\exists(a \,\mathsf{W}\, c)$ $\surd$ as $s \models \exists(a \,\mathsf{W}\, c)$

$\mathcal{T} \models \exists(a \,\mathsf{W}\, \exists\Box b)$ $\surd$ as $s \models \exists\Box a$

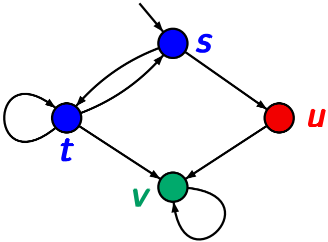
$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \,\mathsf{W}\, (a \wedge b))$ $\surd$

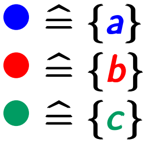
↑

three types of paths: $(s\,t)^\omega$ or $(s\,t)^+\,v^\omega$ or $(s\,t)^*\,s\,u\,v^\omega$

in all three cases: $\pi \models \Box\exists\bigcirc(b \vee c)$

Weak until W in CTL

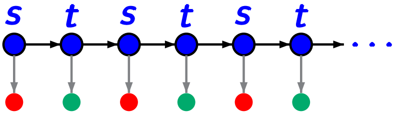




$\mathcal{T} \models \forall\Diamond\exists(a \ \mathsf{W} \ c)$ $\surd$ as $s \models \exists(a \ \mathsf{W} \ c)$

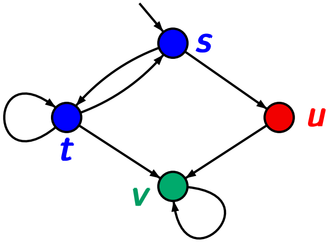
$\mathcal{T} \models \exists(a \ \mathsf{W} \ \exists\Box b)$ $\surd$ as $s \models \exists\Box a$

$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \ \mathsf{W} \ (a \wedge b))$ $\surd$



$\models \Box\exists\bigcirc(b \vee c)$

Weak until W in CTL

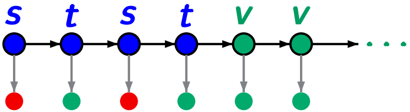


● $\widehat{=}$ $\{a\}$
● $\widehat{=}$ $\{b\}$
● $\widehat{=}$ $\{c\}$

$\mathcal{T} \models \forall\Diamond\exists(a \, \mathsf{W} \, c)$ √ as $s \models \exists(a \, \mathsf{W} \, c)$

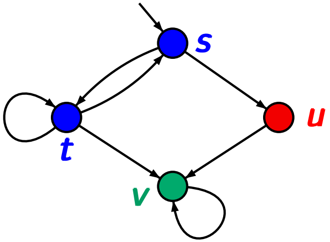
$\mathcal{T} \models \exists(a \, \mathsf{W} \, \exists\Box b)$ √ as $s \models \exists\Box a$

$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \, \mathsf{W} \, (a \wedge b))$ √



$\models \Box\exists\bigcirc(b \vee c)$

Weak until W in CTL

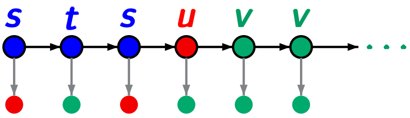


$\bullet \mathrel{\widehat{=}} \{a\}$ (blue)
$\bullet \mathrel{\widehat{=}} \{b\}$ (red)
$\bullet \mathrel{\widehat{=}} \{c\}$ (green)

$\mathcal{T} \models \forall\Diamond\exists(a \mathsf{W} c)$ √ as $s \models \exists(a \mathsf{W} c)$

$\mathcal{T} \models \exists(a \mathsf{W} \exists\Box b)$ √ as $s \models \exists\Box a$

$\mathcal{T} \models \forall((\exists\bigcirc(b \vee c)) \mathsf{W} (a \wedge b))$ √



$\models \Box\exists\bigcirc(b \vee c)$

Expansion laws

Expansion laws

$$\exists(\Phi \, \mathsf{U} \, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \, \mathsf{U} \, \Psi))$$

Expansion laws

$$\exists(\Phi \mathsf{U} \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \mathsf{U} \Psi))$$

$$\forall(\Phi \mathsf{U} \Psi) \equiv ?$$

Expansion laws

$$\exists(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \, \mathsf{U} \, \Psi))$$

$$\forall(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \, \mathsf{U} \, \Psi))$$

Expansion laws

$$\exists(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \exists \bigcirc \exists(\Phi \, \mathsf{U} \, \Psi))$$

$$\forall(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \forall \bigcirc \forall(\Phi \, \mathsf{U} \, \Psi))$$

$$\exists \Diamond \Psi \equiv \Psi \vee \exists \bigcirc \exists \Diamond \Psi$$

$$\forall \Diamond \Psi \equiv \Psi \vee \forall \bigcirc \forall \Diamond \Psi$$

Expansion laws

$$\exists(\Phi \,\mathsf{U}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \,\mathsf{U}\, \Psi))$$

$$\forall(\Phi \,\mathsf{U}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \,\mathsf{U}\, \Psi))$$

$$\exists\Diamond\Psi \;\equiv\; \Psi \vee \exists\bigcirc\exists\Diamond\Psi$$

$$\forall\Diamond\Psi \;\equiv\; \Psi \vee \forall\bigcirc\forall\Diamond\Psi$$

$$\exists(\Phi \,\mathsf{W}\, \Psi) \;\equiv$$

Expansion laws

$$\exists(\Phi \,\mathsf{U}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \,\mathsf{U}\, \Psi))$$

$$\forall(\Phi \,\mathsf{U}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \,\mathsf{U}\, \Psi))$$

$$\exists\Diamond\Psi \;\equiv\; \Psi \vee \exists\bigcirc\exists\Diamond\Psi$$

$$\forall\Diamond\Psi \;\equiv\; \Psi \vee \forall\bigcirc\forall\Diamond\Psi$$

$$\exists(\Phi \,\mathsf{W}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \,\mathsf{W}\, \Psi))$$

Expansion laws

$$\exists(\Phi \mathsf{U} \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \mathsf{U} \Psi))$$

$$\forall(\Phi \mathsf{U} \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \mathsf{U} \Psi))$$

$$\exists\Diamond\Psi \equiv \Psi \vee \exists\bigcirc\exists\Diamond\Psi$$

$$\forall\Diamond\Psi \equiv \Psi \vee \forall\bigcirc\forall\Diamond\Psi$$

$$\exists(\Phi \mathsf{W} \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \mathsf{W} \Psi))$$

$$\forall(\Phi \mathsf{W} \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \mathsf{W} \Psi))$$

Expansion laws

$$\exists(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \, \mathsf{U} \, \Psi))$$

$$\forall(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \, \mathsf{U} \, \Psi))$$

$$\exists\Diamond\Psi \equiv \Psi \vee \exists\bigcirc\exists\Diamond\Psi$$

$$\forall\Diamond\Psi \equiv \Psi \vee \forall\bigcirc\forall\Diamond\Psi$$

$$\exists(\Phi \, \mathsf{W} \, \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \, \mathsf{W} \, \Psi))$$

$$\forall(\Phi \, \mathsf{W} \, \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \, \mathsf{W} \, \Psi))$$

$$\exists\Box\Phi \equiv ?$$

Expansion laws

$$\exists(\Phi \,\mathsf{U}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \,\mathsf{U}\, \Psi))$$

$$\forall(\Phi \,\mathsf{U}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \,\mathsf{U}\, \Psi))$$

$$\exists\Diamond\Psi \;\equiv\; \Psi \vee \exists\bigcirc\exists\Diamond\Psi$$

$$\forall\Diamond\Psi \;\equiv\; \Psi \vee \forall\bigcirc\forall\Diamond\Psi$$

$$\exists(\Phi \,\mathsf{W}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \,\mathsf{W}\, \Psi))$$

$$\forall(\Phi \,\mathsf{W}\, \Psi) \;\equiv\; \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \,\mathsf{W}\, \Psi))$$

$$\exists\Box\Phi \;\equiv\; \Phi \wedge \exists\bigcirc\exists\Box\Phi$$

Expansion laws

$$\exists(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \, \mathsf{U} \, \Psi))$$

$$\forall(\Phi \, \mathsf{U} \, \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \, \mathsf{U} \, \Psi))$$

$$\exists\Diamond\Psi \equiv \Psi \vee \exists\bigcirc\exists\Diamond\Psi$$

$$\forall\Diamond\Psi \equiv \Psi \vee \forall\bigcirc\forall\Diamond\Psi$$

$$\exists(\Phi \, \mathsf{W} \, \Psi) \equiv \Psi \vee (\Phi \wedge \exists\bigcirc\exists(\Phi \, \mathsf{W} \, \Psi))$$

$$\forall(\Phi \, \mathsf{W} \, \Psi) \equiv \Psi \vee (\Phi \wedge \forall\bigcirc\forall(\Phi \, \mathsf{W} \, \Psi))$$

$$\exists\Box\Phi \equiv \Phi \wedge \exists\bigcirc\exists\Box\Phi$$

$$\forall\Box\Phi \equiv \Phi \wedge \forall\bigcirc\forall\Box\Phi$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \quad \equiv \quad \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \quad \equiv \quad \neg\exists\Box\neg\Phi$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

duality of **U** and **W**, e.g.:

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

duality of **U** and **W**, e.g.:

$$\forall(\Phi \,\mathbf{U}\, \Psi)$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

duality of **U** and **W**, e.g.:

$$\forall(\Phi \,\mathsf{U}\, \Psi) \equiv \neg\exists((\Phi \wedge \neg\Psi) \,\mathsf{W}\, (\neg\Phi \wedge \neg\Psi))$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

duality of **U** and **W**, e.g.:

$$\forall(\Phi \,\mathsf{U}\, \Psi) \equiv \neg\exists((\Phi \wedge \neg\Psi) \,\mathsf{W}\, (\neg\Phi \wedge \neg\Psi))$$

$$\equiv \neg\exists((\neg\Psi) \,\mathsf{W}\, (\neg\Phi \wedge \neg\Psi))$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

duality of **U** and **W**, e.g.:

$$\begin{aligned} \forall(\Phi \, \mathsf{U} \, \Psi) &\equiv \neg\exists((\Phi \wedge \neg\Psi) \, \mathsf{W} \, (\neg\Phi \wedge \neg\Psi)) \\ &\equiv \neg\exists((\neg\Psi) \, \mathsf{W} \, (\neg\Phi \wedge \neg\Psi)) \\ &\equiv \neg\exists((\neg\Psi) \, \mathsf{U} (\neg\Phi \wedge \neg\Psi)) \;\wedge\; \neg\exists\Box\neg\Psi \end{aligned}$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

duality of **U** and **W** yields

$$\forall(\Phi \,\mathsf{U}\, \Psi) \equiv \neg\exists((\neg\Psi)\,\mathsf{U}(\neg\Phi \wedge \neg\Psi)) \wedge \neg\exists\Box\neg\Psi$$

Duality laws

duality of $\Box$ and $\Diamond$:

$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$

$$\forall\Diamond\Phi \equiv \neg\exists\Box\neg\Phi$$

self-duality of $\bigcirc$:

$$\forall\bigcirc\Phi \equiv \neg\exists\bigcirc\neg\Phi$$

$$\exists\bigcirc\Phi \equiv \neg\forall\bigcirc\neg\Phi$$

derivation of $\forall\mathsf{U}$ from $\exists\mathsf{U}$ and $\exists\Box$:

$$\forall(\Phi\,\mathsf{U}\,\Psi) \equiv \neg\exists((\neg\Psi)\,\mathsf{U}(\neg\Phi\wedge\neg\Psi)) \wedge \neg\exists\Box\neg\Psi$$

$\forall\mathsf{U}$ and $\forall\bigcirc$ are expressible via $\exists\mathsf{U}$, $\exists\bigcirc$ and $\exists\Box$

Existential normalform for CTL

For each **CTL** formula Ψ there is an equivalent **CTL** formula Φ built by

- operators of propositional logic
- the modalities $\exists\bigcirc$, $\exists\mathbf{U}$ and $\exists\Box$.

For each **CTL** formula Ψ there is an equivalent **CTL** formula Φ built by

- operators of propositional logic
- the modalities $\exists\bigcirc$, $\exists \mathsf{U}$ and $\exists\Box$.

$$\Phi \;::=\; \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \mid \exists\Box\Phi$$

For each **CTL** formula Ψ there is an equivalent **CTL** formula Φ built by

- operators of propositional logic
- the modalities $\exists\bigcirc$, $\exists\mathsf{U}$ and $\exists\Box$.

$$\Phi \;::=\; \mathit{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \exists\bigcirc\Phi \;\big|\; \exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \;\big|\; \exists\Box\Phi$$

transformation $\Psi \rightsquigarrow \Phi$ relies on:

$$\forall\bigcirc\Psi \rightsquigarrow \neg\exists\bigcirc\neg\Psi$$

$$\forall(\Psi_1 \,\mathsf{U}\, \Psi_2) \rightsquigarrow \neg\exists(\neg\Psi_2 \,\mathsf{U}(\neg\Psi_1 \wedge \neg\Psi_2)) \wedge \neg\exists\Box\neg\Psi_2$$

Example: ∃-normal form for CTL formula

CTL formula $\rightsquigarrow$ CTL formula in ∃-normal form

$$\forall\bigcirc\Psi \rightsquigarrow \neg\exists\bigcirc\neg\Psi$$

$$\forall(\Psi_1 \,\mathsf{U}\, \Psi_2) \rightsquigarrow \neg\exists(\neg\Psi_2 \,\mathsf{U}(\neg\Psi_1 \wedge \neg\Psi_2)) \wedge \neg\exists\Box\neg\Psi_2$$

$$\forall((\forall\bigcirc a) \,\mathsf{U}\, \neg c)$$

Example: ∃-normal form for CTL formula

CTL formula $\rightsquigarrow$ CTL formula in ∃-normal form

$$\forall \bigcirc \Psi \rightsquigarrow \neg \exists \bigcirc \neg \Psi$$

$$\forall (\Psi_1 \mathsf{U} \Psi_2) \rightsquigarrow \neg \exists (\neg \Psi_2 \mathsf{U} (\neg \Psi_1 \wedge \neg \Psi_2)) \wedge \neg \exists \Box \neg \Psi_2$$

$$\forall ((\forall \bigcirc a) \mathsf{U} \neg c)$$

$$\equiv \neg \exists (\neg \neg c \mathsf{U} ((\neg \forall \bigcirc a) \wedge \neg \neg c)) \wedge \neg \exists \Box \neg \neg c$$

Example: ∃-normal form for CTL formula

CTL formula $\rightsquigarrow$ CTL formula in ∃-normal form

$$\forall\bigcirc\Psi \rightsquigarrow \neg\exists\bigcirc\neg\Psi$$

$$\forall(\Psi_1 \,\mathsf{U}\, \Psi_2) \rightsquigarrow \neg\exists(\neg\Psi_2 \,\mathsf{U}(\neg\Psi_1 \wedge \neg\Psi_2)) \wedge \neg\exists\Box\neg\Psi_2$$

$$\forall((\forall\bigcirc a) \,\mathsf{U}\, \neg c)$$

$$\equiv \quad \neg\exists(\neg\neg c \,\mathsf{U}((\neg\forall\bigcirc a) \wedge \neg\neg c)) \quad \wedge \quad \neg\exists\Box\neg\neg c$$

$$\equiv \quad \neg\exists(c \,\mathsf{U}((\neg\forall\bigcirc a) \wedge c)) \quad \wedge \quad \neg\exists\Box c$$

Example: ∃-normal form for CTL formula

CTL formula	$\rightsquigarrow$	CTL formula in ∃-normal form
$\forall\bigcirc\Psi$	$\rightsquigarrow$	$\neg\exists\bigcirc\neg\Psi$
$\forall(\Psi_1 \mathsf{U} \Psi_2)$	$\rightsquigarrow$	$\neg\exists(\neg\Psi_2 \mathsf{U}(\neg\Psi_1 \wedge \neg\Psi_2)) \wedge \neg\exists\Box\neg\Psi_2$

$$\forall((\forall\bigcirc a) \mathsf{U} \neg c)$$

$$\equiv \quad \neg\exists(\neg\neg c \,\mathsf{U}((\neg\forall\bigcirc a) \wedge \neg\neg c)) \quad \wedge \quad \neg\exists\Box\neg\neg c$$

$$\equiv \quad \neg\exists(c \,\mathsf{U}((\neg\forall\bigcirc a) \wedge c)) \quad \wedge \quad \neg\exists\Box c$$

$$\equiv \quad \neg\exists(c \,\mathsf{U}((\exists\bigcirc\neg a) \wedge c)) \quad \wedge \quad \neg\exists\Box c$$

Positive normal form

Positive normal form

- negation only on the level of literals

Positive normal form

- negation only on the level of literals
- uses for each operator its dual

Positive normal form

- negation only on the level of literals
- uses for each operator its dual
  * ***true*** and ***false***
  * conjunction and disjunction

Positive normal form for CTL

- negation only on the level of literals
- uses for each operator its dual
  * *true* and *false*
  * conjunction and disjunction
  * existential and universal path quantification
  * until and weak until

Positive normal form for CTL

- negation only on the level of literals
- uses for each operator (one of) its dual(s)
  - ***true*** and ***false***
  - conjunction and disjunction
  - existential and universal path quantification
  - until and weak until

    alternatively: until and release

Positive normal form for CTL

- negation only on the level of literals
- uses for each operator (one of) its dual(s)
  * ***true*** and ***false***
  * conjunction and disjunction
  * existential and universal path quantification
  * until and weak until

    alternatively: until and release

but no additional operator for $\bigcirc$ required syntax of **CTL** formulas in **PNF**:

state formulas:

$$\Phi \;::=\; \mathit{true} \;\big|\; \mathit{false} \;\big|\; a \;\big|\; \neg a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \Phi_1 \vee \Phi_2 \;\big|\; \exists\varphi \;\big|\; \forall\varphi$$

path formulas:

$$\varphi \;::=\; \bigcirc\Phi \;\big|\; \Phi_1 \,\mathsf{U}\, \Phi_2 \;\big|\; \Phi_1 \,\mathsf{W}\, \Phi_2$$

syntax of **CTL** formulas in **PNF**:

state formulas:

$$\Phi ::= true \mid false \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \exists\varphi \mid \forall\varphi$$

path formulas:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \, \mathsf{U} \, \Phi_2 \mid \Phi_1 \, \mathsf{W} \, \Phi_2$$

For each **CTL** formula Ψ there is an equivalent **CTL** formula Φ in **PNF**

CTL $\rightsquigarrow$ CTL-PNF

CTL formula $\rightsquigarrow$ CTL formula in PNF

CTL formula $\leadsto$ CTL formula in PNF

$\neg$*true* $\leadsto$ *false*

CTL formula $\rightsquigarrow$ CTL formula in PNF

$\neg \mathit{true} \rightsquigarrow \mathit{false}$

$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2$ de Morgan laws

CTL formula $\rightsquigarrow$ CTL formula in PNF

$\neg \mathit{true} \rightsquigarrow \mathit{false}$

$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2$ de Morgan laws

$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$

CTL formula $\rightsquigarrow$ CTL formula in PNF

$\neg \textit{true} \rightsquigarrow \textit{false}$

$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2$ de Morgan laws

$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$

$\neg\forall\bigcirc\Phi \rightsquigarrow \exists\bigcirc\neg\Phi$

CTL formula ⇝ CTL formula in PNF

$\neg \mathit{true} \rightsquigarrow \mathit{false}$

$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2$ de Morgan laws

$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$

$\neg\forall\bigcirc\Phi \rightsquigarrow \exists\bigcirc\neg\Phi$

$\neg\exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \forall((\Phi_1 \wedge \neg\Phi_2)\, \mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$

CTL formula ⇝ CTL formula in PNF

$\neg \mathit{true} \rightsquigarrow \mathit{false}$

$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2$ de Morgan laws

$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$

$\neg\forall\bigcirc\Phi \rightsquigarrow \exists\bigcirc\neg\Phi$

$\neg\exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \forall((\Phi_1 \wedge \neg\Phi_2) \,\mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$

$\neg\forall(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \exists((\Phi_1 \wedge \neg\Phi_2) \,\mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$

CTL $\rightsquigarrow$ CTL-PNF

CTL formula $\rightsquigarrow$ CTL formula in PNF

$\neg \mathit{true} \rightsquigarrow \mathit{false}$

$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2$ de Morgan laws

$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$

$\neg\forall\bigcirc\Phi \rightsquigarrow \exists\bigcirc\neg\Phi$

$\neg\exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \forall((\cancel{\Phi_1} \wedge \neg\Phi_2) \,\mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$

$\neg\forall(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \exists((\cancel{\Phi_1} \wedge \neg\Phi_2) \,\mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$

CTL formula $\rightsquigarrow$ CTL formula in PNF

$$\neg \mathit{true} \rightsquigarrow \mathit{false}$$

$$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2 \qquad \text{de Morgan laws}$$

$$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$$

$$\neg\forall\bigcirc\Phi \rightsquigarrow \exists\bigcirc\neg\Phi$$

$$\neg\exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \forall(\neg\Phi_2 \,\mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$$

$$\neg\forall(\Phi_1 \,\mathsf{U}\, \Phi_2) \rightsquigarrow \exists(\neg\Phi_2 \,\mathsf{W}\, (\neg\Phi_1 \wedge \neg\Phi_2))$$

CTL formula $\rightsquigarrow$ CTL formula in PNF

$$\neg \mathit{true} \rightsquigarrow \mathit{false}$$

$$\neg(\Phi_1 \wedge \Phi_2) \rightsquigarrow \neg\Phi_1 \vee \neg\Phi_2 \qquad \text{de Morgan laws}$$

$$\neg\exists\bigcirc\Phi \rightsquigarrow \forall\bigcirc\neg\Phi$$

$$\neg\forall\bigcirc\Phi \rightsquigarrow \exists\bigcirc\neg\Phi$$

$$\neg\exists(\Phi_1 \mathsf{U} \Phi_2) \rightsquigarrow \forall(\neg\Phi_2 \mathsf{W} (\neg\Phi_1 \wedge \neg\Phi_2))$$

$$\neg\forall(\Phi_1 \mathsf{U} \Phi_2) \rightsquigarrow \exists(\neg\Phi_2 \mathsf{W} (\neg\Phi_1 \wedge \neg\Phi_2))$$

... exponential blowup possible ...