# Verification – Lecture 17
# CTL*

Bernd Finkbeiner – Sven Schewe
Rayna Dimitrova – Lars Kuhtz – Anne Proetzsch

Wintersemester 2007/2008

REVIEW

## LTL Fairness constraints

Let $\Phi$ and $\Psi$ be propositional logic formulas over *AP*.

1. An *unconditional LTL fairness constraint* is of the form:

$$ufair \;=\; \Box\Diamond\Psi$$

2. A *strong LTL fairness condition (compassion)* is of the form:

$$sfair \;=\; \Box\Diamond\Phi \;\longrightarrow\; \Box\Diamond\Psi$$

3. A *weak LTL fairness constraint (justice)* is of the form:

$$wfair \;=\; \Diamond\Box\Phi \;\longrightarrow\; \Box\Diamond\Psi$$

A LTL fairness assumption $fair$ is a conjunction of LTL fairness constraints.

# Fair satisfaction

For state $q$ in state graph *S* (over *AP*) without terminal states, let

$$\mathit{FairPaths}_{fair}(q) = \{\, \pi \in \mathit{Paths}(q) \mid \pi \models fair \,\}$$
$$\mathit{FairTraces}_{fair}(q) = \{\, \mathit{trace}(\pi) \mid \pi \in \mathit{FairPaths}_{fair}(q) \,\}$$

For LTL-formula $\varphi$, and fairness assumption $fair$:

$q \models_{fair} \varphi$ if and only if $\forall \pi \in \mathit{FairPaths}_{fair}(q).\, \pi \models \varphi$ and

$S \models_{fair} \varphi$ if and only if $\forall q_0 \in Q_0.\, q_0 \models_{fair} \varphi$

$\models_{fair}$ is the *fair satisfaction relation* for LTL; $\models$ the standard one for LTL





# Reducing $\models_{fair}$ to $\models$

For:

- state graph *S* without terminal states
- LTL formula $\varphi$, and
- LTL fairness assumption *fair*

it holds:

$S \models_{fair} \varphi$ if and only if $S \models (fair \rightarrow \varphi)$

verifying an LTL-formula under a fairness assumption can be done
using standard LTL model-checking algorithms

# Fairness constraints in CTL

- For LTL it holds: $S \models_{fair} \varphi$ if and only if $S \models (fair \rightarrow \varphi)$
- An analogous approach for CTL is not possible!
- Formulas of form $\forall(fair \rightarrow \varphi)$ and $\exists(fair \wedge \varphi)$ needed
- But: boolean combinations of path formulae are not allowed in CTL
- and: strong fairness constraints

$$\Box \Diamond b \rightarrow \Box \Diamond c \equiv \Diamond \Box \neg b \ \vee \ \Diamond \Box c$$

  cannot be expressed, since persistence properties are not in CTL
- Solution: change the semantics of CTL by ignoring unfair paths





# CTL fairness constraints

- A *strong CTL fairness constraint* is a formula of the form:

$$sfair = \bigwedge_{0<i\leqslant k} (\Box \Diamond \Phi_i \rightarrow \Box \Diamond \Psi_i)$$

  - where $\Phi_i$ and $\Psi_i$ (for $0 < i \leqslant k$) are CTL-formulas over *AP*
  - weak and unconditional CTL fairness constraints are defined analogously, e.g.

$$ufair = \bigwedge_{0<i\leqslant k} \Box \Diamond \Psi_i \quad \text{and} \quad wfair = \bigwedge_{0<i\leqslant k} (\Diamond \Box \Phi_i \rightarrow \Box \Diamond \Psi_i)$$

  - a CTL fairness assumption $fair$ is a conjunction of CTL fairness constraints.

$\Rightarrow$ a CTL fairness constraint is an LTL formula over CTL state formulas!

# Semantics of fair CTL

For CTL fairness assumption $fair$, relation $\models_{fair}$ is defined by:

$$
\begin{array}{lll}
s \models_{fair} a & \text{iff} & a \in \mathit{Label}(s) \\
s \models_{fair} \neg \Phi & \text{iff} & \neg (s \models_{fair} \Phi) \\
s \models_{fair} \Phi \vee \Psi & \text{iff} & (s \models_{fair} \Phi) \vee (s \models_{fair} \Psi) \\
s \models_{fair} \exists \varphi & \text{iff} & \pi \models_{fair} \varphi \text{ for } \textit{some fair} \text{ path } \pi \text{ that starts in } s \\
s \models_{fair} \forall \varphi & \text{iff} & \pi \models_{fair} \varphi \text{ for } \textit{all fair} \text{ paths } \pi \text{ that start in } s
\end{array}
$$

$$
\begin{array}{ll}
\pi \models_{fair} \bigcirc \Phi & \text{iff } \pi[1] \models_{fair} \Phi \\
\pi \models_{fair} \Phi \,\mathsf{U}\, \Psi & \text{iff } (\exists\, j \geqslant 0.\, \pi[j] \models_{fair} \Psi \ \wedge\ (\forall\, 0 \leqslant k < j.\, \pi[k] \models_{fair} \Phi))
\end{array}
$$

$\pi$ is a fair path iff $\pi \models fair$ for CTL fairness assumption $fair$





# Transition system semantics

- For CTL-state-formula $\Phi$, and fairness assumption *fair*, the *satisfaction set* $\mathit{Sat}_{fair}(\Phi)$ is defined by:

$$\mathit{Sat}_{fair}(\Phi) = \{ q \in Q \mid q \models_{fair} \Phi \}$$

- *S* satisfies CTL-formula $\Phi$ iff $\Phi$ holds in all its initial states:

$$S \models_{fair} \Phi \quad \text{if and only if} \quad \forall q_0 \in Q_0.\, q_0 \models_{fair} \Phi$$

  - this is equivalent to $Q_0 \subseteq \mathit{Sat}_{fair}(\Phi)$

# Fair CTL model-checking problem

For:

- finite state graph *S* without terminal states
- CTL formula $\Phi$ in ENF, and
- CTL fairness assumption *fair*

establish whether or not:

$$S \models_{fair} \Phi$$

use bottom-up procedure a la CTL to determine $Sat_{fair}(\Phi)$
using as much as possible standard CTL model-checking algorithms





# CTL fairness constraints

- A *strong CTL fairness constraint*: $sfair = \bigwedge_{0<i\leqslant k} (\Box \Diamond \Phi_i \rightarrow \Box \Diamond \Psi_i)$
  - where $\Phi_i$ and $\Psi_i$ (for $0 < i \leqslant k$) are CTL-formulas over *AP*
- Replace the CTL state-formulas in $sfair$ by fresh atomic propositions:

$$sfair := \bigwedge_{0<i\leqslant k} (\Box \Diamond a_i \rightarrow \Box \Diamond b_i)$$

  - where $a_i \in L(s)$ if and only if $s \in Sat(\Phi_i)$ (not $Sat_{fair}(\Phi_i)$!)
  - ... $b_i \in L(s)$ if and only if $s \in Sat(\Psi_i)$ (not $Sat_{fair}(\Psi_i)$!)
  - (for unconditional and weak fairness this goes similarly)
- Note: $\pi \models fair$ iff $\pi[j..] \models fair$ for some $j \geqslant 0$ iff $\pi[j..] \models fair$ for all $j \geqslant 0$

# Results for $\models_{fair}$ (1)

$s \models_{fair} \exists\bigcirc a$ if and only if $\exists s' \in$ *Successors*$(s)$ with $s' \models a$ and *FairPaths*$(s') \neq \varnothing$

$s \models_{fair} \exists(a \,\mathsf{U}\, a')$ if and only if there exists a finite path fragment

$$s_0\, s_1\, s_2 \ldots s_{n-1} s_n \in \textit{Paths}_{\textit{fin}}(s) \quad \text{with } n \geqslant 0$$

such that $s_i \models a$ for $0 \leqslant i < n$, $s_n \models a'$, and *FairPaths*$(s_n) \neq \varnothing$





# Results for $\models_{fair}$ (2)

$s \models_{fair} \exists\bigcirc a$ if and only if $\exists s' \in$ *Successors*$(s)$ with $s' \models a$ and $\underbrace{\textit{FairPaths}(s') \neq \varnothing}_{s' \models_{fair} \exists\Box \text{true}}$

$s \models_{fair} \exists(a \,\mathsf{U}\, a')$ if and only if there exists a finite path fragment

$$s_0\, s_1\, s_2 \ldots s_{n-1} s_n \in \textit{Paths}_{\textit{fin}}(s) \quad \text{with } n \geqslant 0$$

such that $s_i \models a$ for $0 \leqslant i < n$, $s_n \models a'$, and $\underbrace{\textit{FairPaths}(s_n) \neq \varnothing}_{s_n \models_{fair} \exists\Box \text{true}}$

# Core model-checking algorithm

(* states are assumed to be labeled with $a_i$ and $b_i$ *)

compute *Sat*$_{fair}(\exists\Box \text{ true}) = \{ q \in Q \mid$ *FairPaths*$(q) \neq \varnothing \}$
**forall** $q \in$ *Sat*$_{fair}(\exists\Box \text{ true})$ **do** $L(q) := L(q) \cup \{ a_{fair} \}$ **od**

(* compute *Sat*$_{fair}(\Phi)$ *)

**for all** $0 < i \leqslant |\Phi|$ **do**
  **for all** $\Psi \in$ *Sub*$(\Phi)$ with $|\Psi| = i$ **do**
    **switch**($\Psi$):

| | | |
|---|---|---|
| true | : | *Sat*$_{fair}(\Psi) := Q$; |
| $a$ | : | *Sat*$_{fair}(\Psi) := \{ q \in Q \mid a \in L(s) \}$; |
| $a \wedge a'$ | : | *Sat*$_{fair}(\Psi) := \{ q \in Q \mid a, a' \in L(s) \}$; |
| $\neg a$ | : | *Sat*$_{fair}(\Psi) := \{ q \in Q \mid a \notin L(s) \}$; |
| $\exists\bigcirc a$ | : | *Sat*$_{fair}(\Psi) :=$ *Sat*$(\exists\bigcirc(a \wedge a_{fair}))$; |
| $\exists(a \,\mathsf{U}\, a')$ | : | *Sat*$_{fair}(\Psi) :=$ *Sat*$(\exists(a \,\mathsf{U}\, (a' \wedge a_{fair})))$; |
| $\exists\Box\, a$ | : | compute *Sat*$_{fair}(\exists\Box\, a)$ |

    **end switch**
    replace all occurrences of $\Psi$ (in $\Phi$) by the fresh atomic proposition $a_\Psi$
    **forall** $q \in$ *Sat*$_{fair}(\Psi)$ **do** $L(q) := L(q) \cup \{ a_\Psi \}$ **od**
  **od**
**od**
**return** $Q_0 \subseteq$ *Sat*$_{fair}(\Phi)$





# Characterization of *Sat*$_{fair}(\exists\Box\, a)$

$$q \models_{sfair} \exists\Box\, a \quad \text{where} \quad sfair = \bigwedge_{0<i\leqslant k} (\Box\Diamond a_i \rightarrow \Box\Diamond b_i)$$

iff there exists a finite path fragment $q_0 \ldots q_n$ and a cycle $q'_0 \ldots q'_r$ with:

1. $q_0 = q$ and $q_n = q'_0 = q'_r$
2. $q_i \models a$, for any $0 \leqslant i \leqslant n$, and $q'_j \models a$, for any $0 \leqslant j \leqslant r$, and
3. *Sat*$(a_i) \cap \{ q'_1, \ldots, q'_r \} = \varnothing$ or *Sat*$(b_i) \cap \{ q'_1, \ldots, q'_r \} \neq \varnothing$ for $0 < i \leqslant k$

## Computing $Sat_{fair}(\exists\Box\, a)$

- Consider only state $q$ if $q \models a$, otherwise *eliminate* $q$
  - change *S* into *S*$[a] = (Q', Q'_0, E', L')$ with $Q' =$ *Sat*$(a)$,
  - $E' = E \cap (Q' \times Q')$, $Q'_0 = Q_0 \cap Q'$, and $L'(q) = L(q)$ for $q \in Q'$

  $\Rightarrow$ each infinite path fragment in *S*$[a]$ satisfies $\Box\, a$

- $q \models_{fair} \exists\Box\, a$ iff there is a non-trivial SCC $D$ in *S*$[a]$ reachable from $q$:

  $D \cap$ *Sat*$(a_i) = \varnothing$ or $D \cap$ *Sat*$(b_i) \neq \varnothing$ for $0 < i \leqslant k$ (*)

- *Sat*$_{sfair}(\exists\Box\, a) = \{ q \in S \mid$ *Reach*$_{S[a]}(s) \cap T \neq \varnothing \}$
  - $T$ is the union of all non-trivial SCCs $C$ that contain $D$ satisfying (*)

how to compute the set $T$ of SCCs?



## Unconditional fairness

$$ufair \equiv \bigwedge_{0<i\leqslant k} \Box\Diamond\, b_i$$

Let $T$ be the set union of all non-trivial SCCs $C$ of *S*$[a]$ satisfying

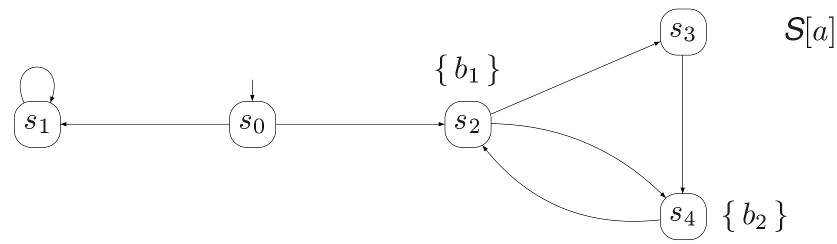
$$C \cap \textit{Sat}(b_i) \neq \varnothing \quad \text{for all } 0 < i \leqslant k$$

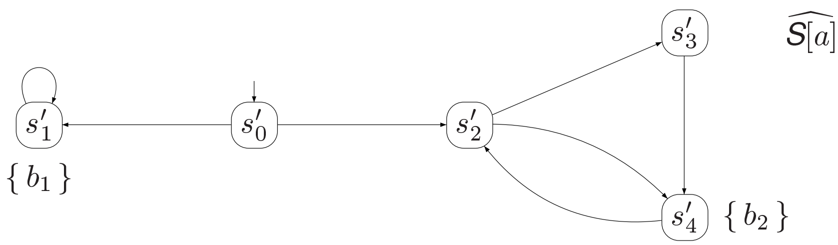
It now follows:

$$q \models_{ufair} \exists\Box\, a \quad \text{if and only if} \quad \textit{Reach}_{S[a]}(q) \cap T \neq \varnothing$$

$\Rightarrow$ $T$ can be determined by a simple graph analysis (DFS)

# Example




S$[a] \models_{ufair} \exists\Box\, a$ but $\widehat{S[a]} \not\models_{ufair} \exists\Box\, a$ with $ufair = \Box\Diamond\, b_1 \;\wedge\; \Box\Diamond\, b_2$



# Strong fairness

- $sfair \;=\; \Box\Diamond a_1 \rightarrow \Box\Diamond b_1$, i.e., $k{=}1$
- $q \models_{sfair} \exists\Box a$ iff $C$ is a non-trivial SCC in S$[a]$ reachable from $q$ with:

  (1) $C \,\cap$ *Sat*$(b_1) \neq \varnothing$, or

  (2) $D \,\cap$ *Sat*$(a_1) = \varnothing$, for some non-trivial SCC $D$ in $C$
- $D$ is a non-trivial SCC in the graph that is obtained from $C[\neg a_1]$
- For $T$ the union of non-trivial SCCs in satisfying (1) and (2):

$$q \models_{sfair} \exists\Box a \quad \text{if and only if} \quad \textit{Reach}_{S[a]}(q) \cap T \neq \varnothing$$

for several strong fairness constraints ($k > 1$), this is applied recursively
$T$ is determined by standard graph analysis (DFS)

# Time complexity

For state graph *S* with $N$ states and $M$ edges,
CTL formula $\Phi$, and CTL fairness constraint $fair$ with $k$ conjuncts,
the CTL model-checking problem $S \models_{fair} \Phi$
can be determined in time $\mathcal{O}(|\Phi| \cdot (N + M) \cdot k)$



# Syntax of CTL$^*$

CTL$^*$ *state-formulas* are formed according to:

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

where $a \in AP$ and $\varphi$ is a path-formula

CTL$^*$ *path-formulas* are formed according to the grammar:

$$\varphi ::= \Phi \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \neg\varphi \;\big|\; \bigcirc\varphi \;\big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

where $\Phi$ is a state-formula, and $\varphi$, $\varphi_1$ and $\varphi_2$ are path-formulas

in CTL$^*$: $\forall\varphi \;=\; \neg\exists\neg\varphi$.

# CTL$^*$ semantics

| | | |
|---|---|---|
| $s \models a$ | iff | $a \in L(s)$ |
| $s \models \neg \Phi$ | iff | not $s \models \Phi$ |
| $s \models \Phi \wedge \Psi$ | iff | $(s \models \Phi)$ and $(s \models \Psi)$ |
| $s \models \exists\varphi$ | iff | $\pi \models \varphi$ for some $\pi \in$ *Paths*$(s)$ |

| | | |
|---|---|---|
| $\pi \models \Phi$ | iff | $\pi[0] \models \Phi$ |
| $\pi \models \varphi_1 \wedge \varphi_2$ | iff | $\pi \models \varphi_1$ and $\pi \models \varphi_2$ |
| $\pi \models \neg\varphi$ | iff | $\pi \not\models \varphi$ |
| $\pi \models \bigcirc\Phi$ | iff | $\pi[1..] \models \Phi$ |
| $\pi \models \Phi \,\mathsf{U}\, \Psi$ | iff | $\exists\, j \geqslant 0.\ (\pi[j..] \models \Psi \ \wedge\ (\forall\, 0 \leqslant k < j.\, \pi[k..] \models \Phi))$ |



# Transition system semantics

- For CTL$^*$-state-formula $\Phi$, the *satisfaction set* *Sat*$(\Phi)$ is defined by:

$$Sat(\Phi) \;=\; \{\, q \in Q \mid q \models \Phi \,\}$$

- *S* satisfies CTL$^*$-formula $\Phi$ iff $\Phi$ holds in all its initial states:

$$S \models \Phi \quad \text{if and only if} \quad \forall q \in Q_0.\, q_0 \models \Phi$$

this is exactly as for CTL

## Embedding of LTL in CTL$^*$

For LTL formula $\varphi$ and *S* without terminal states (both over *AP*) and for each $q \in Q$:

$$\underbrace{q \models \varphi}_{\text{LTL semantics}} \quad \text{if and only if} \quad \underbrace{q \models \forall\varphi}_{\text{CTL}^* \text{ semantics}}$$

In particular:

$$S \models_{LTL} \varphi \quad \text{if and only if} \quad S \models_{CTL*} \forall\varphi$$



## CTL$^*$ is more expressive than LTL and CTL

For the CTL$^*$-formula over $AP = \{a, b\}$:

$$\Phi = (\forall\Diamond\Box a) \vee (\forall\Box\exists\Diamond b)$$

there does *not* exist any equivalent LTL- or CTL formula

# This logic is as expressive as CTL

$CTL^+$ *state-formulas* are formed according to:

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

where $a \in AP$ and $\varphi$ is a path-formula

$CTL^+$ *path-formulas* are formed according to the grammar:

$$\varphi ::= \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \bigcirc\Phi \;\Big|\; \Phi_1 \,\mathsf{U}\, \Phi_2$$

where $\Phi, \Phi_1, \Phi_2$ are state-formulas, and $\varphi$, $\varphi_1$ and $\varphi_2$ are path-formulas



# $CTL^+$ is as expressive as CTL

For example: $\underbrace{\exists(\Diamond a \wedge \Diamond b)}_{CTL^+ \text{ formula}} \equiv \underbrace{\exists\Diamond(a \wedge \exists\Diamond b) \;\wedge\; \exists\Diamond(b \wedge \exists\Diamond a)}_{\text{CTL formula}}$

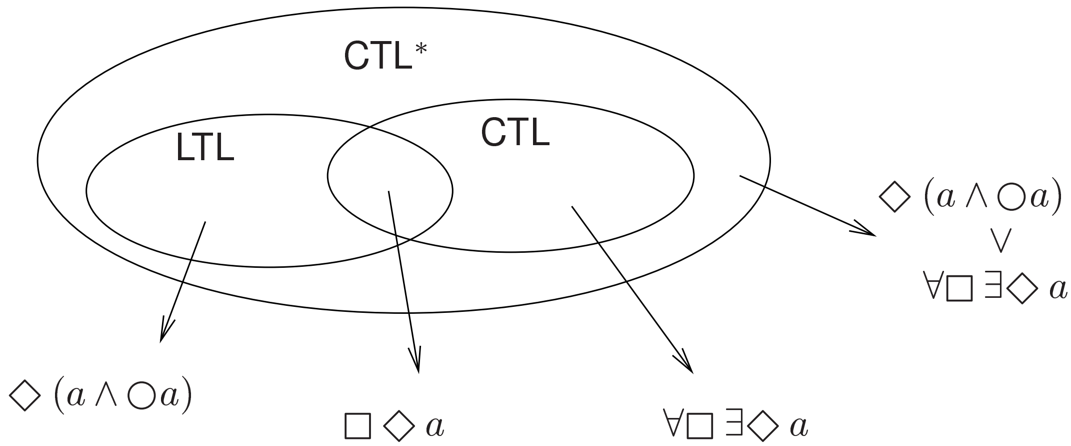
Some rules for transforming $CTL^+$ formulae into equivalent CTL ones:

$$\begin{aligned}
\exists\Big(\neg(\Phi_1 \,\mathsf{U}\, \Phi_2)\Big) &\equiv \exists\Big(\,(\Phi_1 \wedge \neg\Phi_2) \,\mathsf{U}\, (\neg\Phi_1 \wedge \neg\Phi_2)\,\Big) \;\vee\; \exists\Box\neg\Phi_2 \\
\exists\Big(\bigcirc\Phi_1 \wedge \bigcirc\Phi_2\Big) &\equiv \exists\bigcirc(\Phi_1 \wedge \Phi_2) \\
\exists\Big(\bigcirc\Phi \wedge (\Phi_1 \,\mathsf{U}\, \Phi_2)\Big) &\equiv \Big(\Phi_2 \wedge \exists\bigcirc\Phi\Big) \;\vee\; \Big(\Phi_1 \wedge \exists\bigcirc(\Phi \wedge \exists(\Phi_1 \,\mathsf{U}\, \Phi_2))\Big) \\
\exists\Big((\Phi_1 \,\mathsf{U}\, \Phi_2) \;\wedge\; (\Psi_1 \,\mathsf{U}\, \Psi_2)\Big) &\equiv \exists\Big((\Phi_1 \wedge \Psi_1) \,\mathsf{U}\, (\Phi_2 \wedge \exists(\Psi_1 \,\mathsf{U}\, \Psi_2))\Big) \;\vee \\
&\phantom{\equiv}\; \exists\Big((\Phi_1 \wedge \Psi_1) \,\mathsf{U}\, (\Psi_2 \wedge \exists(\Phi_1 \,\mathsf{U}\, \Phi_2))\Big) \\
&\vdots
\end{aligned}$$

adding boolean combinations of path formulae to CTL does not change its expressiveness

but $CTL^+$ formulae can be much shorter than shortest equivalent CTL formulae

## Relationship between LTL, CTL and CTL$^*$






## CTL$^*$ model checking

- Adopt the same bottom-up procedure as for (fair) CTL
- Replace each maximal proper sub-formula $\Psi$ by new proposition $a_\Psi$
  - $a_\Psi \in L(s)$ if and only if $s \in$ *Sat*$(\Psi)$
- Most interesting case: formulas of the form $\exists\varphi$
  - by replacing all maximal state sub-formulas in $\varphi$, an LTL-formula results!
- $q \models \exists\varphi$ iff $\underbrace{q \not\models \forall\neg\varphi}_{\text{CTL}^* \text{ semantics}}$ iff $\underbrace{q \not\models \neg\varphi}_{\text{LTL semantics}}$
  - $Sat_{CTL*}(\exists\varphi) = Q \setminus Sat_{LTL}(\neg\varphi)$

# CTL* model-checking algorithm

**for all** $i \leqslant |\,\Phi\,|$ **do**
  **for all** $\Psi \in$ *Sub*$(\Phi)$ with $|\,\Psi\,| = i$ **do**
    **switch**$(\Psi)$:

| | | |
|---|---|---|
| true | : | *Sat*$(\Psi) := Q$; |
| $a$ | : | *Sat*$(\Psi) := \{\, q \in Q \mid a \in L(q) \,\}$; |
| $a_1 \wedge a_2$ | : | *Sat*$(\Psi) :=$ *Sat*$(a_1) \cap$ *Sat*$(a_2)$; |
| $\neg a$ | : | *Sat*$(\Psi) := S \setminus$ *Sat*$(a)$; |
| $\exists\varphi$ | : | determine *Sat*$_{LTL}(\neg\varphi)$ by means of an LTL model-checker; |
| | : | *Sat*$(\Psi) := Q \setminus$ *Sat*$_{LTL}(\neg\varphi)$ |

    **end switch**
    *AP* := *AP* $\cup \{\, a_\Psi \,\}$;          (* introduce fresh atomic proposition *)
    replace $\Psi$ with $a_\Psi$
    **forall** $q \in$ *Sat*$(\Psi)$ **do** $L(q) := L(q) \cup \{\, a_\Psi \,\}$; **od**
  **od**
**od**
**return** $Q_0 \subseteq$ *Sat*$(\Phi)$



# Time complexity

For transition system *S* with $N$ states and $M$ transitions, CTL* formula $\Phi$, the CTL* model-checking problem *S* $\models \Phi$ can be determined in time $\mathcal{O}((N+M)\cdot 2^{|\Phi|})$.

the CTL* model-checking problem is PSPACE-complete