$\boxed{\text{Lecture 7}}$
11-02-2025

CTL

$\Phi ::= \text{true} \mid \underset{\in}{\overset{\text{Act}}{a}} \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \exists \varphi \mid \forall \varphi$ (state formulas)

$\varphi ::= \bigcirc \Phi \mid \Phi_1 \, U \, \Phi_2$ (path formulas)

CTL$^+$

$\Phi ::= \text{true} \mid \underset{\in}{\overset{\text{Act}}{a}} \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \exists \varphi \mid \forall \varphi$ (state)

$\varphi ::= \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \bigcirc \Phi \mid \Phi_1 \, U \, \Phi_2$ (path)

$\forall \varphi := \neg \exists \neg \varphi$ could be used to drop clause $\forall \varphi$ from the state formulas

CTL*

$\Phi ::= \text{true} \mid \underset{\in}{\overset{\text{Act}}{a}} \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \exists \varphi$ (state forms)

$\varphi ::= \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi_1 \, U \, \varphi_2$ (path forms)

$\forall \varphi := \neg \exists \neg \varphi$

LTL

$\varphi ::= \text{true} \mid \underset{\text{Act}}{\underset{\in}{a}} \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi_1 \, U \, \varphi_2$

CTL ENF (Existential Normal Form)

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

CTL PNF (Positive Normal Form)

$\Phi ::= \text{true} \mid \text{false} \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \exists \varphi \mid \forall \varphi$ (state forms)

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

Thm. For every CTL-formula there exists an equivalent CTL-ENF formula.

Proof.

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

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

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

$\forall \Box \Phi \equiv \neg \exists \Diamond \neg \Phi \equiv \neg \exists (\text{true} \, U \, \Phi)$

Thm. For every CTL-formula there exists an equivalent CTL-PNF formula.

Proof.

$\neg \text{true} \equiv \text{false}$

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

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

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

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

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

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

Aspect	Linear time	Branching time
"behaviour" in a state s	path-based: trace(s)	state based computation tree of s
temporal logic	LTL: path formula φ $s \models \varphi \Leftrightarrow$ $\Leftrightarrow \forall \pi \in Paths(s): \pi \models \varphi$	CTL: state formulae existential path quantification universal path quantification
Complexity of model checking problems	PSPACE-complete $O(\lvert TS \rvert \cdot \exp(\lvert \varphi \rvert))$	PTIME $O(\lvert TS \rvert \cdot \lvert \Phi \rvert)$
adequate subsumption and equivalence relations	trace inclusion and trace equivalence (can be checked in PSPACE-complete)	bisimulation subsumption bisimulation equivalence (can be checked in polynomial time)
fairness	no special techniques needed	special techniques needed

CTL-formulas Φ and Ψ over AP are equivalent (denoted $\Phi \equiv \Psi$) if $Sat(\Phi) = Sat(\Psi)$ for all transition systems TS over AP.

Normal Forms

Existential Normal Form (ENF)

$$\Phi ::= true \mid \underset{\in AP}{a} \mid \Phi \wedge \Phi \mid \neg \Phi \mid \exists \bigcirc \Phi \mid \exists (\Phi \, U \, \Phi) \mid \exists \Box \Phi$$

Thm. For every CTL-formula there is an equivalent CTL-formula in ENF.

Positive Normal Form

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

$$\varphi ::= \bigcirc \Phi \mid \Phi_1 \, U \, \Phi_2 \mid \Phi_1 \, \underset{\text{weak until}}{W} \, \Phi_2$$

Thm. For each CTL-formula there is an equivalent CTL-formula in PNF.

Weak Until:

intuitively not yet a definition

$$\pi \models \Phi W \Psi \Leftrightarrow \pi \models \Phi U \Psi \text{ or } \pi \models \Box(\Phi \wedge \neg \Psi)$$

$$\Leftrightarrow \pi \models \Phi U \Psi \text{ or } \pi \models \Box \Phi$$

Can be obtained by defining:

$$\exists(\Phi W \Psi) := \neg \forall((\Phi \wedge \neg \Psi) \, U \, (\neg \Phi \wedge \neg \Psi))$$

$$\forall(\Phi W \Psi) := \neg \exists((\Phi \wedge \neg \Psi) \, U \, (\neg \Phi \wedge \neg \Psi))$$

$\boxed{CTL^+}$ Extending CTL with Boolean Connectives

path-formulas

Syntax $\Phi ::= true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$ (same as for CTL; $a \in AP$)

$\varphi ::= \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\Phi \mid \Phi_1 \mathsf{U} \Phi_2$ (CTL^+ path formulas)

Examples

$\underbrace{\exists(a \mathsf{W} b)}_{\text{CTL-formula (after using the definition of W)}} \equiv \underbrace{\exists((a \mathsf{U} b) \vee \Box a)}_{CTL^+\text{-formula but not a CTL-formula}}$, $\underbrace{\exists\neg\bigcirc\Phi}_{CTL^+\text{ form}} \equiv \underbrace{\exists\bigcirc\neg\Phi}_{\text{CTL-form}}$

$\underbrace{\exists(\Diamond a \wedge \Diamond b)}_{CTL^+\text{-formula}} \equiv \underbrace{\exists\Diamond(a \wedge \exists\Diamond b) \vee \exists\Diamond(b \wedge \exists\Diamond a)}_{\text{CTL-formula}}$

CTL^+-formula: $\exists\neg(\Phi_1 \mathsf{U} \Phi_2)$ $\equiv \underbrace{\exists((\Phi_1 \wedge \neg\Phi_2) \mathsf{U} (\neg\Phi_1 \wedge \neg\Phi_2)) \vee \exists\Box\neg\Phi_2}_{\text{CTL-formula}}$

Thm. Every CTL^+-formula is equivalent to a CTL-formula.

Incomparable expressiveness of LTL and CTL

Lemma. Φ a CTL-formula, φ a LTL-formula that results by eliminating all path quantifiers from Φ.
Then: $\underbrace{\Phi \equiv \varphi}_{\text{Forall TS: } TS \models \Phi \Leftrightarrow TS \models \varphi}$ or there is no LTL-formula that is equivalent to Φ.

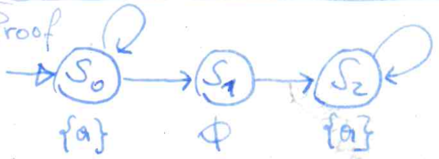
Examples:

$\forall\bigcirc a \equiv \bigcirc a$, $\forall(a \mathsf{U} b) \equiv a \mathsf{U} b$, $\forall\Diamond a \equiv \Diamond a$, $\forall\Box a \equiv \Box a$

$\forall\Box\forall\Diamond a \equiv \Box\Diamond a$

Proposition. $\forall\Diamond\forall\Box a \not\equiv \Diamond\Box a$.

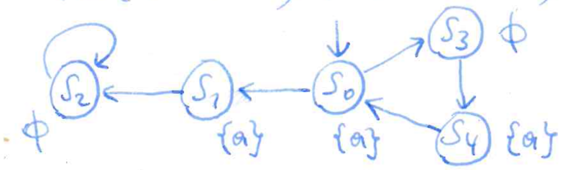
Proof



$s_0 \models_{LTL} \Diamond\Box a$ because $\pi = s_0 \ldots$ — $\pi = s_0^\omega \models \Diamond\Box a$; $\pi = s_0^* s_1 s_2^\omega \models \Diamond\Box a$

$s_0 \not\models \forall\Diamond\forall\Box a$ because $s_0^\omega \not\models \Diamond\forall\Box a$, $s_0 \not\models \forall\Box a$

$\forall\Diamond(a \wedge \forall\bigcirc a) \not\equiv \Diamond(a \wedge \bigcirc a)$



$s_0 s_1 s_2^\omega \models_{LTL} \Diamond(a \wedge \bigcirc a)$ since $s_0 s_1 s_2^\omega \models a \wedge \bigcirc a$

$s_0 s_1 s_2^\omega \not\models_{CTL} \Diamond(a \wedge \forall\bigcirc a)$ since $s_0 \not\models_{CTL} \forall\bigcirc a$, $s_1 \not\models_{CTL} \forall\bigcirc a$, $s_2 \not\models \forall\bigcirc a$

Hence $s_0 \not\models_{CTL} \forall\Diamond(a \wedge \forall\bigcirc a)$, yet $s_0 \models_{LTL} \Diamond(a \wedge \bigcirc a)$

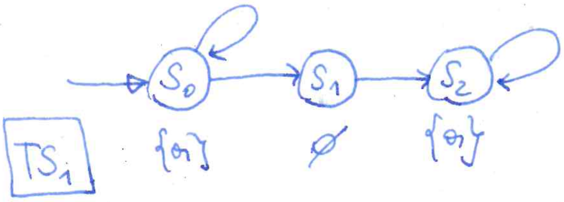
since $(s_0 s_3 s_4)^* s_0 s_1 s_2^\omega \models \Diamond(a \wedge \bigcirc a)$
$(s_0 s_3 s_4)^\omega \models \Diamond(a \wedge \bigcirc a)$.

Thm. Incomparable Expressiveness LTL/CTL

(a) There are LTL-formulas for which no equivalent CTL-formulas exist
e.g. $\Diamond\Box a$ and $\Diamond(a \wedge \bigcirc a)$.

(b) There are CTL-formulas for which no equivalent LTL-formulas exist
E.g. $\forall\Diamond\forall\Box a$ and $\forall\Diamond(a \wedge \forall\bigcirc a)$ and $\forall\Box\exists\Diamond a$

Proposition. (i) $\forall\Diamond\forall\Box a \not\equiv \Diamond\Box a$ (eventually always) $\Rightarrow$ there is no LTL-formula that is equivalent with $\forall\Diamond\forall\Box a$



$s_0 \models_{LTL} \Diamond\Box a$ because $\pi = s_0 \ldots$

- $\pi = s_0^\omega \models \Diamond\Box a$; $s_0^\omega \models a$; $s_0^\omega \models \Box a$
- $\pi = s_0^* s_1 s_2^\omega \models \Diamond\Box a$; $s_2^\omega \models a$; $s_2^\omega \models \Box a$

$TS_1 \models_{LTL} \Diamond\Box a$

$TS_1 \not\models_{CTL} \forall\Diamond\forall\Box a$

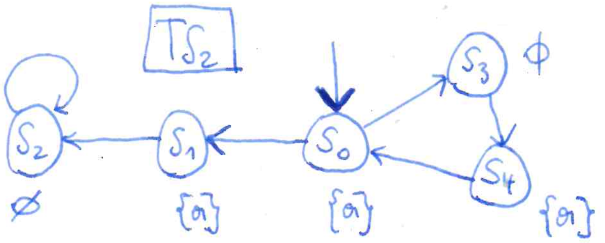
$s_0 \not\models_{CTL} \forall\Diamond\forall\Box a$

because $s_0^\omega \not\models \Diamond\forall\Box a$

$s_0 \not\models \forall\Box a$ since $s_0^* s_1 s_2^\omega \not\models \Box a$

$s_1 s_2^\omega \not\models a$

(ii) $\forall\Diamond(a \wedge \forall\bigcirc a) \not\equiv \Diamond(a \wedge \bigcirc a) \Rightarrow$ there is no LTL-formula that is equivalent with $\forall\Diamond(a \wedge \forall\bigcirc a)$



$s_0 \models_{LTL} \Diamond(a \wedge \bigcirc a)$ because $\pi = s_0 \ldots$

- $(s_0 s_3 s_4)^* s_1 s_2^\omega \models \Diamond(a \wedge \bigcirc a)$; $s_1 s_2^\omega \models a \wedge \bigcirc a$
- $(s_0 s_3 s_4)^\omega \models \Diamond(a \wedge \bigcirc a)$; $s_4 (s_0 s_3 s_4)^\omega \models a \wedge \bigcirc a$

$TS_2 \models_{LTL} \Diamond(a \wedge \bigcirc a)$

$s_0 \not\models_{CTL} \forall\Diamond(a \wedge \forall\bigcirc a)$

because $s_0 s_1 s_2^\omega \not\models \Diamond(a \wedge \forall\bigcirc a)$ $\quad TS_2 \not\models_{CTL} \forall\Diamond(a \wedge \forall\bigcirc a)$

$\Uparrow$

$s_0 \not\models \forall\bigcirc a$

$s_1 \not\models \forall\bigcirc a$

$s_2 \not\models \forall\bigcirc a$

Lemma. Φ a CTL-formula, φ the LTL formula that arises by deleting all path quantifiers from Φ.

Then: $\varphi \underset{LTL\ \ CTL}{\equiv} \Phi$ or there is no LTL-formula that is equivalent to Φ.

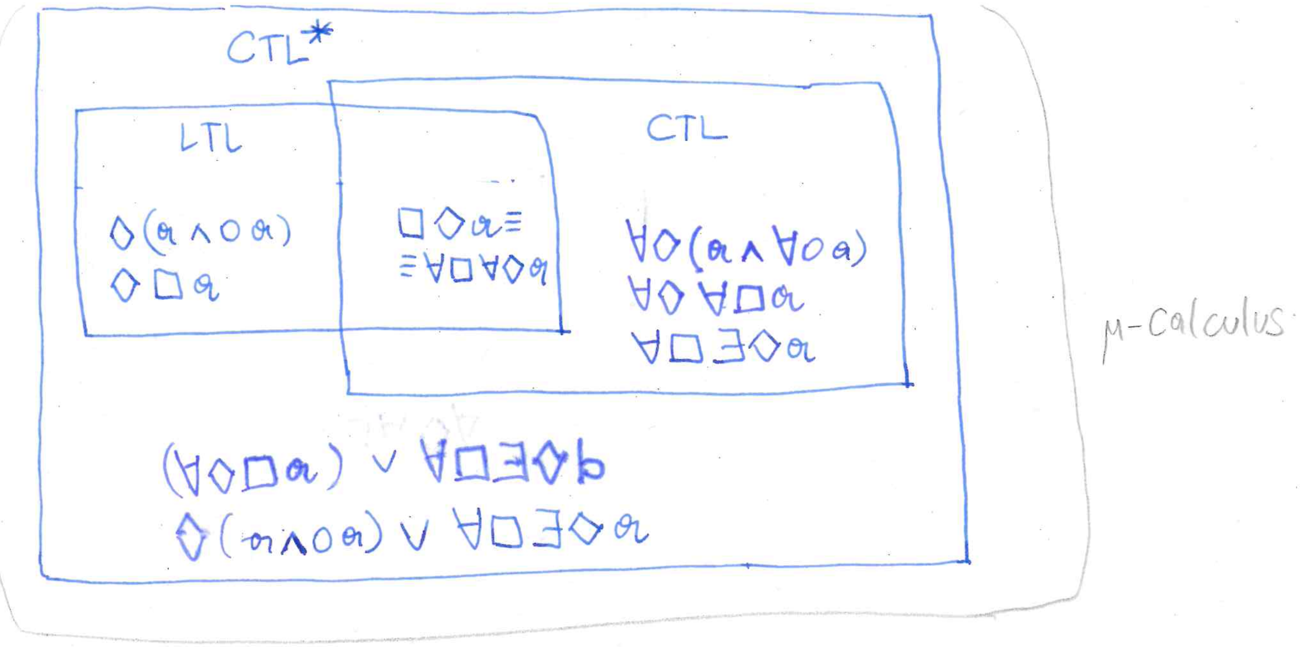
Examples: $\forall\bigcirc a \equiv \bigcirc a$, $\forall(a \mathcal{U} b) \equiv a \mathcal{U} b$, $\forall\Diamond a \equiv \Diamond a$, $\forall\Box a \equiv \Box a$

$\forall\Box\forall\Diamond a \equiv \Box\Diamond a$.

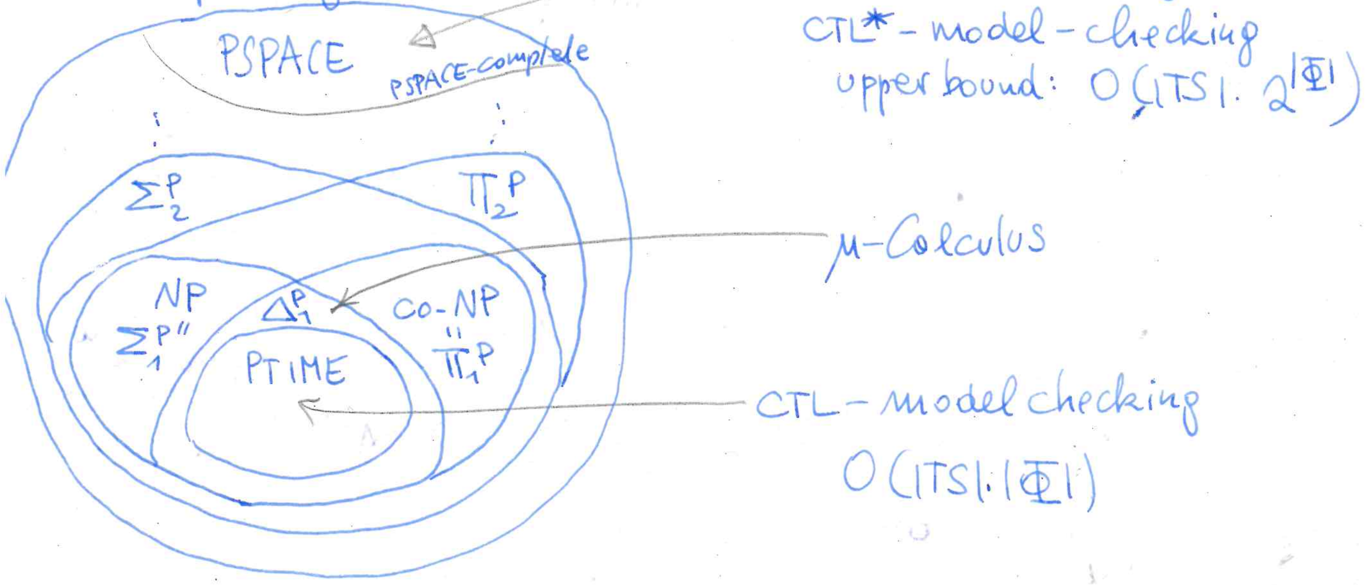
Theorem. Incomparable Expressiveness LTL / CTL

(a) There are LTL formulas for which no equivalent CTL-formulas exist. E.g. $\Diamond\Box a$, and $\Diamond(a \wedge \bigcirc a)$.

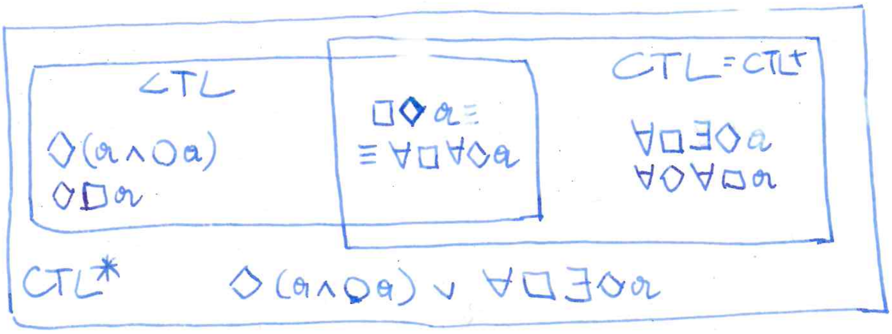
(b) There are CTL-formulas for which no equivalent LTL-formulas exist. E.g. $\forall\Diamond\forall\Box a$, $\forall\Diamond(a \wedge \forall\bigcirc a)$, $\forall\Box\exists\Diamond a$


CTL*
LTL
$\Diamond(a \wedge \bigcirc a)$
$\Diamond\Box a$
$\Box\Diamond a \equiv \forall\Box\forall\Diamond a$
CTL
$\forall\Diamond(a \wedge \forall\bigcirc a)$
$\forall\Diamond\forall\Box a$
$\forall\Box\exists\Diamond a$
$(\forall\Diamond\Box a) \vee \forall\Box\exists\Diamond a$
$\Diamond(a \wedge \bigcirc a) \vee \forall\Box\exists\Diamond a$
μ-Calculus

Complexity.


PSPACE
PSPACE-complete
Σ_2^P
Π_2^P
NP
Σ_1^P
Δ_1^P
co-NP
Π_1^P
PTIME
LTL-model checking
CTL*-model-checking
upper bound: $O(|TS| \cdot 2^{|\Phi|})$
μ-Calculus
CTL-model checking
$O(|TS| \cdot |\Phi|)$

Relationship between LTL, CTL, and CTL*



Thm. For the CTL*-formula $\Diamond(a \wedge \bigcirc a) \vee \forall\Box\exists\Diamond a$ there does not exist any equivalent LTL or CTL-formula.

	CTL	LTL	CTL*
model checking	PTIME	PSPACE-complete	PSPACE-Complete
without fairness	$size(TS) \cdot \lvert\Phi\rvert$	$size(TS) \cdot exp(\lvert\Phi\rvert)$	$size(TS) \cdot exp(\lvert\Phi\rvert)$
with fairness	$size(TS) \cdot \lvert\Phi\rvert \cdot \lvert fair\rvert$	$size(TS) \cdot exp(\lvert\Phi\rvert) \cdot \lvert fair\rvert$	$size(TS) \cdot exp(\lvert\Phi\rvert) \cdot \lvert fair\rvert$
for fixed specifications	$O(size(TS))$	$O(size(TS))$	$O(size(TS))$
Satisfiability check	EXPTIME	PSPACE-complete	2EXPTIME
best known technique upper bound	$O(exp(\lvert\Phi\rvert))$	$exp(\lvert\Phi\rvert)$	$exp(exp(\lvert\Phi\rvert))$

CTL* (Emerson, Halpern, 1985/86)

Syntax

$$\Phi ::= \text{true} \mid \underset{\in AP}{a} \mid \Phi \wedge \Phi \mid \neg \Phi \mid \exists \varphi \qquad \text{(CTL*-formulas) state formulas}$$

$$\varphi ::= \Phi \mid \varphi \wedge \varphi \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi \,\mathsf{U}\, \varphi \qquad \text{path formulas}$$

defined: $\Diamond \varphi := \text{true} \,\mathsf{U}\, \varphi$ $\qquad \forall \varphi := \neg \exists \neg \varphi$

$\Box \varphi := \neg \Diamond \neg \varphi$

Example

$\forall \Box (\bigcirc \Diamond a \wedge \neg (b \,\mathsf{U}\, \Box c))$,

$\forall \bigcirc \Box \neg a \wedge \exists \Diamond \Box (a \vee \forall (b \,\mathsf{U}\, a))$

not CTL-formulas

Semantics

For $a \in AP$ and $TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ a transition system, and all $s \in S$:

$s \models a \quad :\Longleftrightarrow \quad a \in L(s)$

$s \models \neg \Phi \quad :\Longleftrightarrow \quad$ not $s \models \Phi$ (i.e. $s \not\models \Phi$)

$s \models \Phi \wedge \Psi \quad :\Longleftrightarrow \quad (s \models \Phi)$ and $(s \models \Psi)$

$s \models \exists \varphi \quad :\Longleftrightarrow \quad \pi \models \varphi$ for some $\pi \in Paths(s)$.

same as for CTL

For all paths π in S:

$\pi \models \Phi \quad :\Longleftrightarrow \quad \pi[0] \models \Phi$ $\qquad$ NEW for CTL*

$\pi \models \varphi_1 \wedge \varphi_2 \quad :\Longleftrightarrow \quad \pi \models \varphi_1$ and $\pi \models \varphi_2$

$\pi \models \neg \varphi \quad :\Longleftrightarrow \quad \pi \not\models \varphi$

same as for CTL+

$\pi \models \bigcirc \varphi \quad :\Longleftrightarrow \quad \pi_{\geq 1} \models \varphi$

$\pi \models \varphi_1 \,\mathsf{U}\, \varphi_2 \quad :\Longleftrightarrow \quad \exists j \geq 0. \,(\pi_{\geq j} \models \varphi_2 \wedge \forall 0 \leq k < j : \pi_{\geq k} \models \varphi_1)$

same as for CTL

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

$TS \models \Phi \quad :\Longleftrightarrow \quad \forall s_0 \in I : s_0 \models \Phi$.

Embedding of LTL in CTL*

Thm. $TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ a transition system without terminal states.

For every LTL-formula φ and for each $s \in S$:

$$\underbrace{s \models \varphi}_{\text{LTL-semantics}} \quad \Longleftrightarrow \quad \underbrace{s \models \forall \varphi}_{\text{CTL*-semantics}}$$

$$TS \models_{LTL} \varphi \quad \Longleftrightarrow \quad TS \models_{CTL^*} \forall \varphi$$

Exercises from Last time

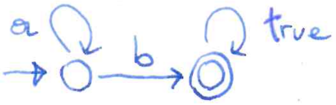
Ex 5.24 (d) Is $\Phi := \Box a \cup \Diamond b \to \Box(a \cup \Diamond b)$ valid / satisfiable?

Φ is satisfiable: $\underbrace{\phi\phi\phi\ldots}_{\phi^\omega \in (2^{AP})^\omega} \models \Phi$ because $\phi^\omega \not\models \Diamond b$
$\phi^\omega \not\models \Box a \cup \Diamond b$
hence $\phi^\omega \models \Box a \cup \Diamond b \to \ldots$

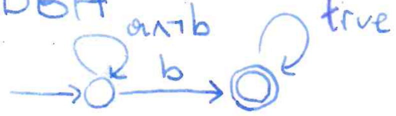
Φ is not valid: $\sigma := \underbrace{\{b\}\phi\phi\ldots}_{\{b\}\phi^\omega} \not\models \Phi$ because $\sigma = \{b\}\phi^\omega \models \Diamond b$
$\sigma = \{b\}\phi^\omega \models \Box a \cup \Diamond b$
$\sigma \not\models \Box(a \cup \Diamond b)$
since $\sigma_{\geq 1} \not\models \Diamond b$, $\sigma \not\models a$

Marokh's argumentation: $\Psi \cup \Diamond \Phi \equiv \Diamond \Phi$ for all Φ, Ψ!
Hence $(\Box a \cup \Diamond b \to \Box(a \cup \Diamond b)) \equiv \Diamond b \to \Box \Diamond b$ which is obviously not valid!

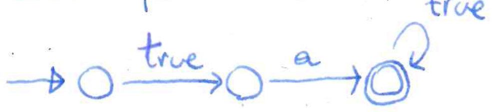
nondeterministic Büchi automaton NBA for $a \cup b$ with 2 states



deterministic Büchi-automaton with 2 states DBA

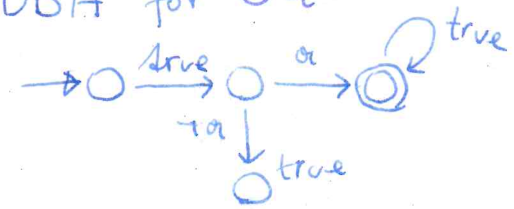


NBA for $\bigcirc a$

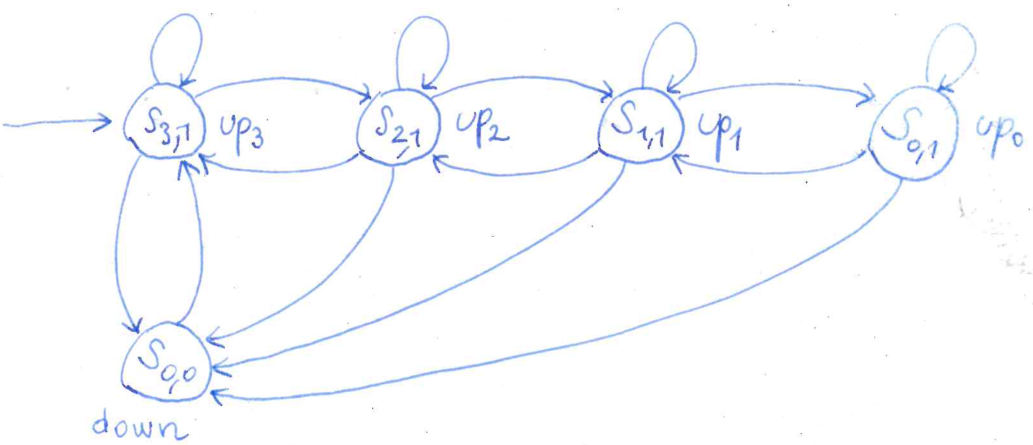


3 states

DBA for $\bigcirc a$



4 states



Possibly the system never goes down $\exists \Box \neg down$

Invariantly the system never goes down $\forall \Box \neg down$

It is always possible to start as new $\forall \Box \exists \Diamond up_3$

The system always eventually goes down and is operational until going down $\forall((up_3 \vee up_2) \cup down)$

CTL Model Checking

CTL Model Checking Problem
Input: a transition system TS, and a CTL formula φ
Question: does $TS \models \Phi$ hold?

TS is assumed to be finite, with no terminal states.

Recall: $Sat(\Phi) := \{ s \in S \mid s \models \Phi \}$ ... states of S in which Φ is satisfied.

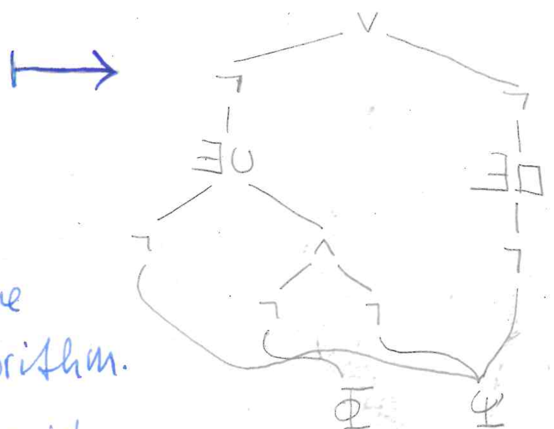
We will use CTL-formulas in ENF existential normal form

$\boxed{\text{CTL-ENF}}$ $\quad \Phi ::= true \mid \underset{\in Act}{a} \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \exists \bigcirc \Phi \mid \exists(\Phi_1 \mathsf{U} \Phi_2) \mid \exists \Box \Phi$

Recall:
every CTL-formula can be transformed into an equivalent CTL-ENF formula (although with an exponential overhead)

e.g. $\forall(\Phi \mathsf{U} \Psi) \longmapsto \neg \exists(\neg \Psi \mathsf{U} (\neg \Phi \wedge \neg \Psi)) \vee \neg \exists \Box \neg \Psi$
(3 occurrences of Ψ)

This overhead could be avoided by using dag-representations of formulas.



The overhead is relevant for the **determination** of the complexity of the CTL-model checking algorithm.

A different approach to avoid the complexity increase due to this transformation, is to extend the CTL-model checking algorithm to deal also with formulas $\forall \bigcirc \Phi$, $\forall(\Phi \mathsf{U} \Psi)$, and $\forall \Box \Phi$.

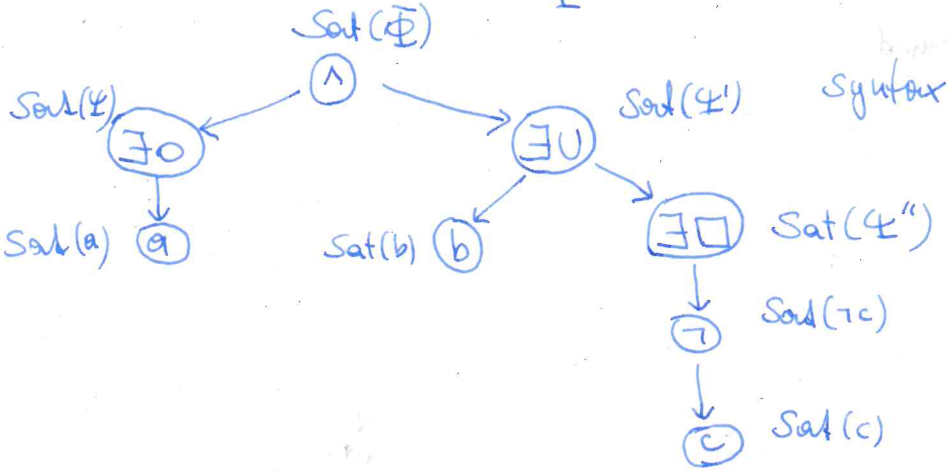
Basic idea of the model-checking algorithm for CTL:

(i) Compute $Sat(\Phi)$ by induction on subformulas of Φ

(ii) Then $TS \models \Phi \iff \underbrace{I}_{\text{initial states of } \Phi} \subseteq Sat(\Phi)$

Example: $AP = \{a, b, c\}$

$$\Phi = \underbrace{\exists \bigcirc a}_{\Psi} \wedge \underbrace{\exists (b \; U \; \underbrace{\exists \Box \neg c}_{\Psi''})}_{\Psi'}$$



syntax tree

we calculate the satisfaction sets $Sat(\tilde{\Psi})$ for all subformulas $\tilde{\Psi}$ of Φ by induction over the syntax tree

Characterization of Sat(·) for CTL formulae in ENF

Theorem. Let $TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ be a transition system. For all CTL-formulae Φ, Ψ over AP:

(a) $Sat(true) = S$,

(b) $Sat(a) = \{s \in S \mid a \in L(s)\}$ for all $a \in Act$.

(c) $Sat(\Phi \wedge \Psi) = Sat(\Phi) \cap Sat(\Psi)$.

(d) $Sat(\neg\Phi) = S \setminus Sat(\Phi)$

(e) $Sat(\exists \bigcirc \Phi) = \{s \in S / Post(s) \cap Sat(\Phi) \neq \emptyset\}$

(f) $Sat(\exists \Phi \cup \Psi)$ = the <u>smallest</u> subset $T \subseteq S$ such that (i) $Sat(\Psi) \subseteq T$, and (ii) $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset \Rightarrow s \in T$

(g) $Sat(\exists \square \Phi)$ = the largest subset $T \subseteq S$ such that (i) $T \subseteq Sat(\Phi)$, and (ii) $s \in T \Rightarrow Post(s) \cap T \neq \emptyset$.

Theorem. Time Complexity of CTL-model checking

For transition system TS with N states and K transitions, and CTL-formula Φ, the CTL-model checking problem $TS \models \Phi$ can be determined in time $O((N+K) \cdot |\Phi|)$.

Derived characterizations for CTL-formulas of the forms $\forall \bigcirc \Phi$, $\forall \Phi \cup \Psi$, $\forall \square \Phi$:

(h) $Sat(\forall \bigcirc \Phi) = \{s \in S / Post(s) \subseteq Sat(\Phi)\}$

(i) $Sat(\forall \Phi \cup \Psi)$ is the smallest set $T \subseteq S$ such that

$$Sat(\Psi) \cup \{s \in Sat(\Phi) / Post(s) \subseteq T\} \subseteq T$$

(j) $Sat(\forall \square \Phi)$ is the largest set $T \subseteq S$ such that

$$T \subseteq \{s \in Sat(\Phi) \mid Post(s) \subseteq T\}.$$

Alternative Formulation of Sat($\exists \Phi U \Psi$) and Sat($\exists \Box \Psi$)

$$\exists \Phi U \Psi \equiv \Psi \vee (\Phi \wedge \exists \bigcirc (\exists \Phi U \Psi)).$$

Thus $\exists \Phi U \Psi$ is a fixed point of:

$$F \equiv \Psi \vee (\Phi \wedge \exists \bigcirc (F)). \qquad (*)$$

But also $\exists (\Phi W \Psi)$ is a solution, but it is larger in the sense that $Sat(\exists \Box (\Phi W \Psi)) \supseteq Sat(\exists \Box (\Phi U \Psi))$.

However: $\exists (\Phi U \Psi)$ is the <u>least</u> solution of $(*)$:

(f)' $Sat(\exists (\Phi U \Psi))$ is the smallest set $T \subseteq S$ such that

$$Sat(\Psi) \cup \{s \in Sat(\Phi) / Post(s) \cap T \neq \emptyset\} \subseteq T.$$

with μ-Calculus notation:

$$\exists (\Phi U \Psi) \simeq \underbrace{\mu F. (\Psi \vee (\Phi \wedge \exists \bigcirc \Psi))}_{\mu\text{-Calculus notation.}}$$

Also:

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

Hence $\exists \Box \Phi$ is a fixed point of

$$F \equiv \Phi \wedge \exists \bigcirc F.$$

Indeed it is the Largest fixed point w.r.t. "measure" $Sat(\cdot)$.

(g)' $Sat(\exists \Box \Phi)$ is the Largest set $T \subseteq S$ such that

$$T \subseteq \{s \in Sat(\Phi) / Post(s) \cap T \neq \emptyset\}.$$

with μ- Calculus notation:

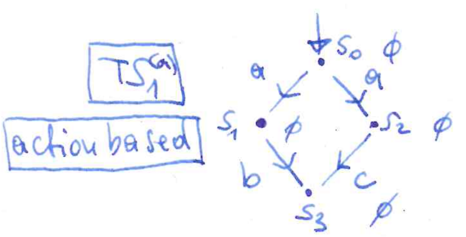
$$\exists \Box \Phi \simeq \underbrace{\nu F. (\Phi \wedge \exists \bigcirc \Phi)}_{\mu\text{- Calculus notation.}}$$

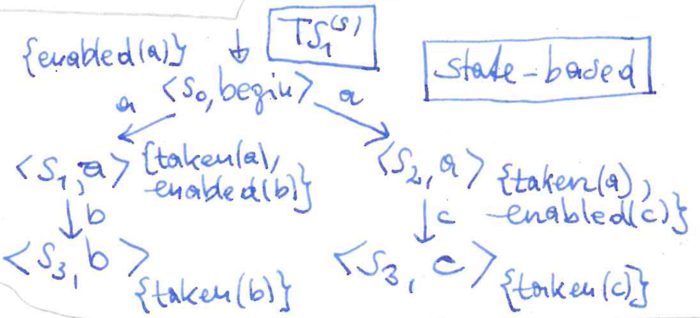
MOTIVATION for CTL (Shortcoming of LTL)

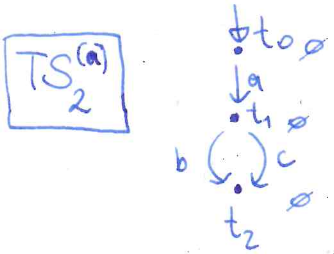
CTL permits finer distinctions between transition systems:

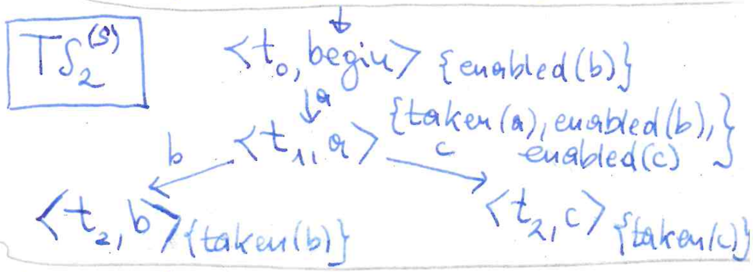
① LTL-satisfiability of formulas in a TS does not distinguish between trace-equivalent systems

② CTL-satisfiability in a TS does not distinguish between bisimilar systems.
(bisimilarity is a finer equivalence than trace equivalence).









$TS_1^{(s)}$ and $TS_2^{(s)}$ have the same traces!
$\Rightarrow$ they satisfy the same formulas (due to ①)

Hence: "After taken a-step, always a b-step is enabled" is not expressible correctly.

Attempt: $\bigcirc(\text{taken}(a) \rightarrow \text{enabled}(b)) =: \Phi$

Yet: $TS_1^{(s)} \not\models \Phi$, and $TS_2^{(s)} \not\models \Phi$, because: $s_0 s_2 s_3 \not\models \Phi$, $t_0 t_1 t_2 \not\models \Phi$

However CTL can make a distinction between TS_1 and TS_2.

$\Psi := \forall\bigcirc(\text{taken}(a) \longrightarrow \text{enabled}(b))$

"On all paths, after taking an a-step, a b-step is possible"

$TS_1^{(s)} \not\models \Psi$, but $TS_2^{(s)} \models \Psi$.

We note that TS_1 and TS_2 are not bisimilar. (Hence (2) is not applicable.)

① $\text{Traces}(TS_1) = \text{Traces}(TS_2) \overset{\text{if } TS_1, TS_2 \text{ image-finite}}{\Longleftarrow\!\!\Longrightarrow} \forall \text{LTL-formulas } \Phi\, [TS_1 \models \Phi \Leftrightarrow TS_2 \models \Phi]$

$\Uparrow$ if TS_1, TS_2 are image-finite $\Downarrow$ also if TS_1, TS_2 are image-finite

② $TS_1 \sim TS_2 \overset{\text{if } TS_1, TS_2 \text{ image-finite}}{\Longleftarrow\!\!\Longrightarrow} \forall \text{CTL-formulas } \Psi\, [TS_1 \models \Psi \Leftrightarrow TS_2 \models \Psi]$