

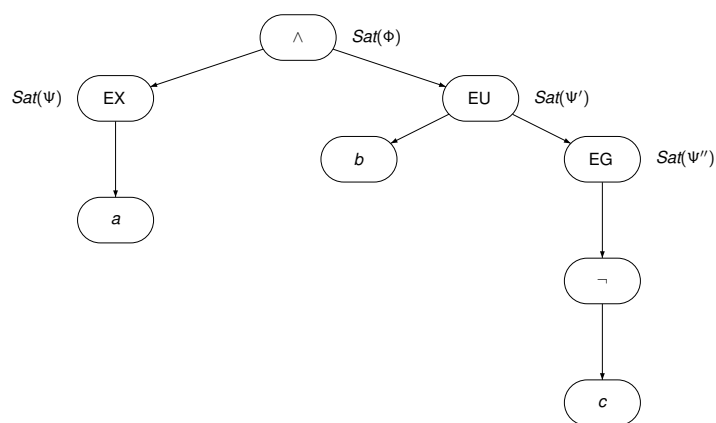
Model checking CTL

- ▶ How to check whether state graph TS satisfies CTL formula $\hat{\Phi}$?
 - ▶ convert the formula $\hat{\Phi}$ into the equivalent Φ in ENF
 - ▶ compute recursively the set $Sat(\Phi) = \{q \in S \mid q \models \Phi\}$
 - ▶ $TS \models \Phi$ if and only if each initial state of TS belongs to $Sat(\Phi)$
- ▶ Recursive **bottom-up** computation of $Sat(\Phi)$:
 - ▶ consider the parse-tree of Φ
 - ▶ start to compute $Sat(a_i)$, for all leaves in the tree
 - ▶ then go one level up in the tree and determine $Sat(\cdot)$ for these nodes

$$\text{e.g.,: } Sat(\underbrace{\Psi_1 \wedge \Psi_2}_{\text{node at level } i}) = Sat(\underbrace{\Psi_1}_{\text{node at level } i-1}) \cap Sat(\underbrace{\Psi_2}_{\text{node at level } i-1})$$

- ▶ then go one level up and determine $Sat(\cdot)$ of these nodes
- ▶ and so on..... until the root is treated, i.e., $Sat(\Phi)$ is computed

Example



$$\Phi = \underbrace{EX a}_{\Psi} \wedge \underbrace{E(bU EG \neg c)}_{\Psi'}$$

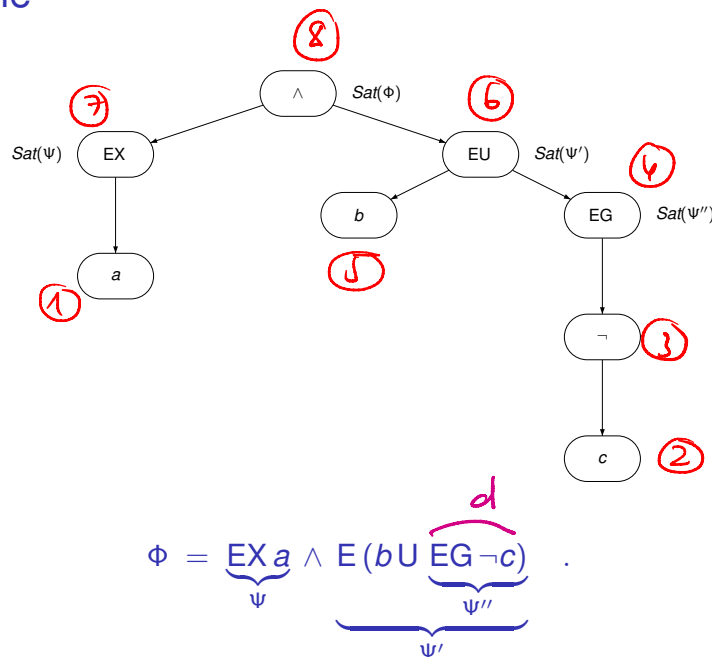
Model checking CTL

- ▶ How to check whether state graph TS satisfies CTL formula $\hat{\Phi}$?
 - ▶ convert the formula $\hat{\Phi}$ into the equivalent Φ in ENF
 - ▶ compute recursively the set $Sat(\Phi) = \{q \in S \mid q \models \Phi\}$
 - ▶ $TS \models \Phi$ if and only if each initial state of TS belongs to $Sat(\Phi)$
- ▶ Recursive **bottom-up** computation of $Sat(\Phi)$:
 - ▶ consider the parse-tree of Φ
 - ▶ start to compute $Sat(a_i)$, for all leaves in the tree
 - ▶ then go one level up in the tree and determine $Sat(\cdot)$ for these nodes

$$\text{e.g.,: } Sat(\underbrace{\Psi_1 \wedge \Psi_2}_{\text{node at level } i}) = Sat(\underbrace{\Psi_1}_{\text{node at level } i-1}) \cap Sat(\underbrace{\Psi_2}_{\text{node at level } i-1})$$

- ▶ then go one level up and determine $Sat(\cdot)$ of these nodes
- ▶ and so on..... until the root is treated, i.e., $Sat(\Phi)$ is computed

Example



Basic algorithm

Require: finite transition system TS with states S and initial states I ,
and CTL formula ϕ (both over AP)

Ensure: $TS \models \phi$

```
{compute the sets  $Sat(\phi) = \{q \in S \mid q \models \phi\}$ }
for all  $i \leq |\phi|$  do
  for all  $\psi \in Sub(\phi)$  with  $|\psi| = i$  do
    compute  $Sat(\psi)$  from  $Sat(\psi')$  {for maximal proper
       $\psi' \in Sub(\psi)$ }
  end for
end for
return  $I \subseteq Sat(\phi)$ 
```

Characterization of $Sat(1)$

For all CTL formulas ϕ, ψ over AP it holds:

$$Sat(\text{true}) = S$$

$$Sat(a) = \{q \in S \mid a \in L(q)\}, \text{ for any } a \in AP$$

$$Sat(\phi \wedge \psi) = Sat(\phi) \cap Sat(\psi)$$

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

$$Sat(EX\phi) = \{q \in S \mid Post(q) \cap Sat(\phi) \neq \emptyset\}$$

for a given finite transition system with states S

$$Sat(EF\phi) = \{q \in S \mid Post^*(q) \cap Sat(\phi) \neq \emptyset\}$$

$\left\{ \begin{array}{l} \text{true} \\ a \in AP \\ \wedge, \neg \\ EX \end{array} \right.$

$V \rightarrow \cup$

Characterization of $Sat(2)$

$$\varphi \cup \psi \equiv \varphi \vee \neg (\varphi \wedge (\varphi \cup \psi))$$

$Sat(E(\Phi \cup \Psi))$ is the **smallest** subset T of S , such that:

- (1) $Sat(\Psi) \subseteq T$ and
 (2) $(q \in Sat(\Phi) \text{ and } Post(q) \cap T \neq \emptyset) \Rightarrow q \in T$

- ▶ We show that for any T that satisfies (1) and (2), we have $Sat(E(\Phi \cup \Psi)) \subseteq T$.
- ▶ Let $s \in Sat(E(\Phi \cup \Psi))$. $E(\Phi \cup \Psi) \equiv \Psi \vee (\exists X E(\Phi \cup \Psi) \wedge \Phi)$
- ▶ If $s \in Sat(\Psi)$, then, by (1), $s \in T$.
- ▶ Otherwise, there exists a path $\pi = s_0 s_1 s_2 \dots$ starting in $s = s_0$ such that $\pi \models \Phi \cup \Psi$.
- ▶ Let $n > 0$ such that $s_n \models \Psi$ and $s_i \models \Phi$ for all $0 \leq i < n$.
- ▶ $s_n \in T$ by (1), because $s_n \in Sat(\Psi)$.
- ▶ $s_{n-1} \in T$ by (2), because $s_n \in Post(s_{n-1}) \cap T$ and $s_{n-1} \in Sat(\Phi)$.
- ▶ ...
- ▶ $s = s_0 \in T$ by (2), because $s_1 \in Post(s_0) \cap T$ and $s_0 \in Sat(\Phi)$.

Characterization of $Sat(3)$

$$EG \phi \equiv \phi \wedge \exists X EG \phi$$

$Sat(EG \phi)$ is the **largest** subset T of S , such that:

- (3) $T \subseteq Sat(\phi)$ and
 (4) $q \in T$ implies $Post(q) \cap T \neq \emptyset$

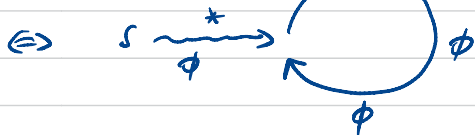
- ▶ We show that for any T that satisfies (3) and (4), we have $T \subseteq Sat(EG \phi)$.
- ▶ Let $s \in T$. We construct a path $\pi = s_0 s_1 s_2 \dots$ as follows:
 - ▶ $s_0 = s$
 - ▶ Since $s_0 \in T$, we find, by (4), a state $s_1 \in Post(s_0) \cap T$.
 - ▶ Since $s_1 \in T$, we find, by (4), a state $s_2 \in Post(s_1) \cap T$.
 - ▶ ...
- ▶ By (3), we have $s_i \in T \subseteq Sat(\phi)$. Hence, $s \in Sat(EG \phi)$.

Algorithme alternatif pour EG ϕ :
(! le ST n'a pas d'état puits)

$S \models EG \phi \iff \exists$ chemin infini

$s = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$ tg.

Si $\models \phi$



Strongly connected components (SCC)

$U \subseteq V$ SCC si $\forall v_1, v_2 \in U$:

$v_1 \xrightarrow{*} v_2$

(\exists cycle qui contient tout U)

On peut calculer les SCC en temps linéaire
(algorithme de Tarjan / Kosaraju)

- 1) On restreint le ST à $\text{Sat}(\phi) \rightarrow ST'$
- 2) On calcule les SCC de ST' .
- 3) $s \in \text{Sat}(EG \phi) \iff$

$s \xrightarrow{*} C$, avec C SCC, $|C| \geq 2$
(SCC non triviale)

\downarrow
chemin dans
ST

Computing $Sat(E(\Phi \cup \Psi))$ (1)

$Sat(E(\Phi \cup \Psi))$ is the smallest set $T \subseteq Q$ such that:

(1) $Sat(\Psi) \subseteq T$ and (2) $(q \in Sat(\Phi) \text{ and } Post(q) \cap T \neq \emptyset) \Rightarrow q \in T$

- ▶ This suggests to compute $Sat(E(\Phi \cup \Psi))$ iteratively:

$$T_0 = Sat(\Psi) \text{ and } T_{i+1} = T_i \cup \{q \in Sat(\Phi) \mid Post(q) \cap T_i \neq \emptyset\}$$

- ▶ T_i = states that can reach a Ψ -state in at most i steps via a Φ -path
- ▶ By induction on j it follows:

$$T_0 \subseteq T_1 \subseteq \dots \subseteq T_j \subseteq T_{j+1} \subseteq \dots \subseteq Sat(E(\Phi \cup \Psi))$$

Computing $Sat(E(\Phi \cup \Psi))$ (2)

- ▶ TS is finite, so for some $j \geq 0$ we have:
 $T_j = T_{j+1} = T_{j+2} = \dots$
- ▶ Therefore: $T_j = T_j \cup \{q \in Sat(\Phi) \mid Post(q) \cap T_j \neq \emptyset\}$
- ▶ Hence: $\{q \in Sat(\Phi) \mid Post(q) \cap T_j \neq \emptyset\} \subseteq T_j$
 - ▶ hence, T_j satisfies (2), i.e.,
 $(q \in Sat(\Phi) \text{ and } Post(q) \cap T_j \neq \emptyset) \Rightarrow q \in T_j$
 - ▶ further, $Sat(\Psi) = T_0 \subseteq T_j$ so, T_j satisfies (1), i.e.
 $Sat(\Psi) \subseteq T_j$
- ▶ As $Sat(E(\Phi \cup \Psi))$ is the **smallest** set satisfying (1) and (2),
 $Sat(E(\Phi \cup \Psi)) \subseteq T_j$ and thus $Sat(E(\Phi \cup \Psi)) = T_j$.
- ▶ Hence:
 $T_0 \subsetneq T_1 \subsetneq T_2 \subsetneq \dots \subsetneq T_j = T_{j+1} = \dots = Sat(E(\Phi \cup \Psi))$

① Calcul de $E(a \cup b)$:

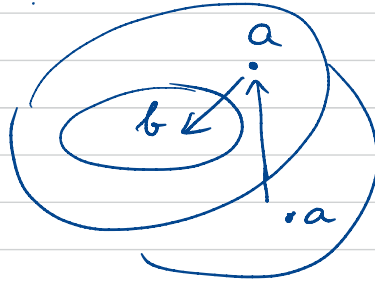
$$Y := \text{sat}(b)$$

Répéter

$$X := Y$$

$$Y := X \cup \{q : q \in \text{sat}(a) \wedge \text{post}(q) \in X\}$$

tant que $(Y \neq X)$ $O(|S|+1 \rightarrow 1)$

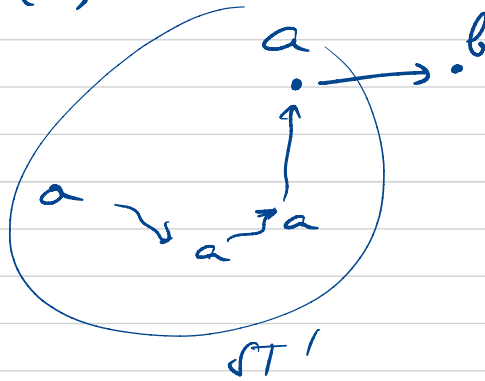


Alternatif :

$$z = \text{sat}(a) \cap \text{Pre}(\text{sat}(b))$$

$$ST' = ST \upharpoonright \text{sat}(a)$$

$$\text{Pre}_{ST'}^*(z)$$



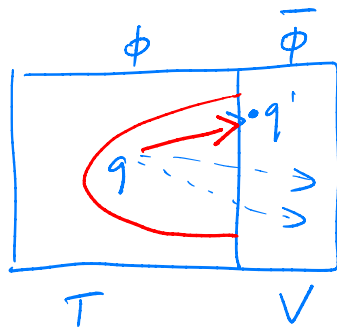
Computing $Sat(E(\Phi \cup \Psi))$ (3)

Require: finite transition system with states S CTL-formula $E(\Phi \cup \Psi)$

Ensure: $Sat(E(\Phi \cup \Psi)) = \{q \in S \mid q \models E(\Phi \cup \Psi)\}$

```

V := Sat(Ψ); {V administers states q with q ⊨ E(Φ ∪ Ψ)}
T := V; {T contains the already visited states q with q ⊨ E(Φ ∪ Ψ)}
while V ≠ ∅ do
  let q' ∈ V;
  V := V \ {q'};
  for all q ∈ Pre(q') do
    if q ∈ Sat(Φ) \ T then V := V ∪ {q}; T := T ∪ {q}; endif
  end for
end while
return T
    
```



$q \in T$ $q' \in V$
 $q \rightarrow q' \not\models \Phi$
 $\models \Phi$

Computing $Sat(EG\Phi)$

$V := S \setminus Sat(\Phi)$; {V contains any not visited q' with $q' \not\models EG\Phi$ }

$T := Sat(\Phi)$; {T contains any q for which $q \models EG\Phi$ has not yet been disproven}

for all $q \in Sat(\Phi)$ **do** $c[q] := |Post(q)|$; **od** {initialize array c }

while $V \neq \emptyset$ **do**

{loop invariant: $c[q] = |Post(q) \cap (T \cup V)|$ }

let $q' \in V$; { $q' \not\models \Phi$ }

$V := V \setminus \{q'\}$; { q' has been considered}

for all $q \in Pre(q')$ **do**

if $q \in T$ **then**

$c[q] := c[q] - 1$; {update counter $c[q]$ for predecessor q of q' }

if $c[q] = 0$ **then**

$T := T \setminus \{q\}$; $V := V \cup \{q\}$; { q does not have any successor in T }

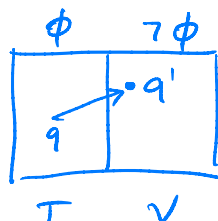
end if

end if

end for

end while

return T



$O(|S| + 1 \rightarrow 1)$

Alternative algorithm for $Sat(EG \phi)$

1. Consider only state q if $q \models \phi$, otherwise **eliminate** q
 - ▶ change states to $S' = Sat(\phi)$,
2. Determine all **non-trivial strongly connected components** in $TS[\phi]$
 - ▶ non-trivial SCC = maximal, connected subgraph with at least one edge
 - ⇒ any state in such SCC satisfies $EG \phi$
3. $q \models EG \phi$ is equivalent to “some **SCC is reachable** from q ”
 - ▶ this search can be done in a backward manner

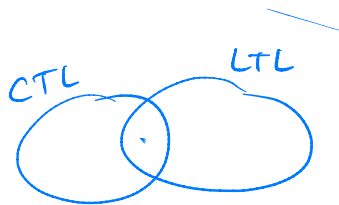
Time complexity

For transition system TS with N states and M edges,
and CTL formula ϕ , the CTL model-checking problem $TS \models \phi$
can be determined in time $\mathcal{O}(|\phi| \cdot (N + M))$

$$LTL \ \psi \quad \mathcal{O}((N + M) \cdot 2^{\mathcal{O}(|\psi|)})$$
$$|A_{\neg\psi}| \approx 2^{\mathcal{O}(|\psi|)}$$

Model-checking LTL versus CTL

- ▶ Let TS be a transition system with N states and M edges
- ▶ Model-checking LTL-formula ϕ has time-complexity $\mathcal{O}((N+M) \cdot 2^{|\phi|})$
 - ▶ linear in the state space of the system model
 - ▶ exponential in the length of the formula
- ▶ Model-checking CTL-formula ϕ has time-complexity $\mathcal{O}((N+M) \cdot |\phi|)$
 - ▶ linear in the state space of the system model and the formula
- ▶ Is model-checking CTL more efficient?



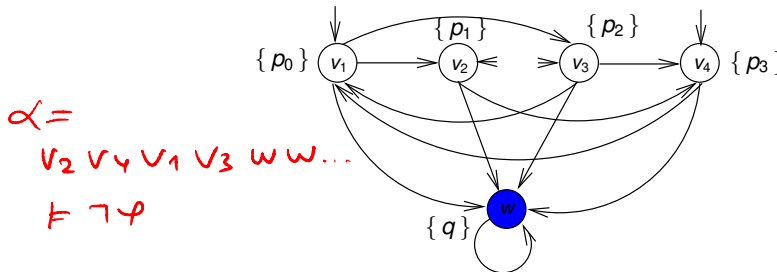
CTL, LTL
pas comparable

- il ex. ϕ de CTL tq. $\not\equiv \psi$ LTL, $\phi \equiv \psi$
- il ex. ψ de LTL tq. $\not\equiv \phi$ CTL, $\phi \equiv \psi$

Hamiltonian path problem

1 fois
→ NP-C

⇒ LTL-formulae can be exponentially shorter than their CTL-equivalent



- ▶ Existence of Hamiltonian path in LTL:
 $\gamma\psi = \bigwedge_i (\neg p_i \wedge G(p_i \rightarrow \neg G\neg p_i))$
- ▶ In CTL, all possible (= 4!) routes need to be encoded
 - $p_0 \wedge EX(p_1 \wedge EX(p_2 \wedge EX p_3) \dots) \vee$
 - $p_0 \wedge EX(p_1 \wedge EX(p_3 \wedge EX p_2) \dots) \vee$
 - \vdots

Equivalence of LTL and CTL formulas

CTL-formula Φ and LTL-formula φ (both over AP) are **equivalent**, denoted $\Phi \equiv \varphi$, if for any transition system TS (over AP):

$$TS \models \Phi \quad \text{if and only if} \quad TS \models \varphi$$

Examples (1)

CTL-formula $AGAFa$ and LTL-formula GFa are equivalent.

$$1) \quad S \models AG \ AFa \quad \Rightarrow \quad S \models LTL \ GFa$$

Si $S = S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow \dots$ alors

$$\forall i \geq 0 : S_i \models AFa \quad (\text{à cause de } AG)$$

$$\Rightarrow \forall \text{ chemin } \pi \text{ issu de } S_i : \pi \models F a$$

En particulier, pour $\pi = S_i \rightarrow S_{i+1} \rightarrow \dots$

$$\Rightarrow \exists j \geq i : \underline{S_j \models a}$$

On veut de voir que $S \models LTL \ GFa$

$$2) \quad S \models LTL \ GFa \quad \Rightarrow \quad S \models \underline{AGAFa}$$

On veut montrer que tout desc. s' de S ($S \xrightarrow{*} s'$) : $s' \models \underline{AFa}$

On doit montrer que pour tout chemin

$$s' = s'_0 \rightarrow s'_1 \rightarrow \dots$$

$$\exists i \text{ tq. } s'_i \models a$$

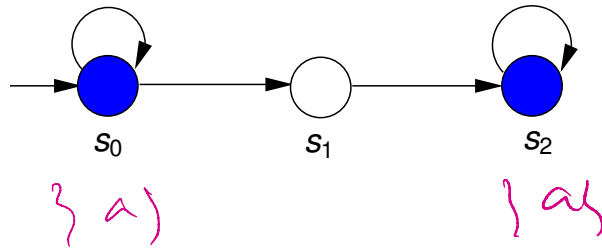
Mais on a : $S \xrightarrow{*} s' = s'_0 \rightarrow s'_1 \rightarrow \dots$

chemin qui sat. GFa (par hypothèse)

$$\Rightarrow \exists i \text{ tq. } s'_i \models a \quad (\text{parce que } a \text{ est satisfait } \infty\text{-souvent}).$$

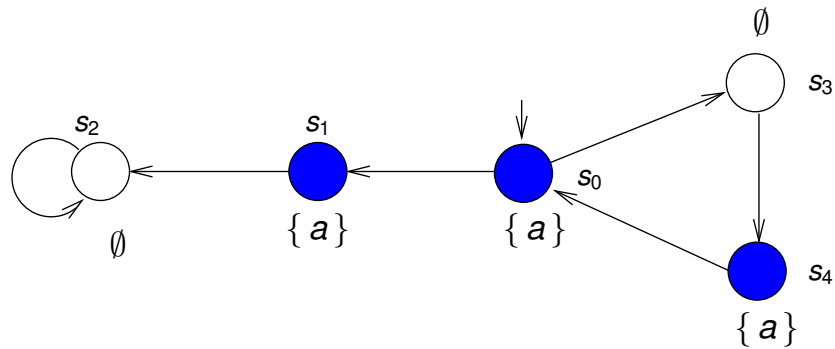
Examples (2)

$AFAG a$ is not equivalent to $FG a$



Examples (3)

$F(a \wedge X a)$ is not equivalent to $AF(a \wedge AX a)$



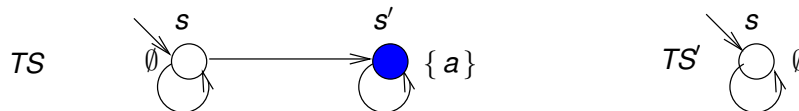
LTL and CTL are incomparable

- ▶ Some LTL-formulas cannot be expressed in CTL, e.g.,
 - ▶ $FG a$
 - ▶ $F(a \wedge X a)$
 - ▶ Some CTL-formulas cannot be expressed in LTL, e.g.,
 - ▶ $AF AG a$
 - ▶ $AF(a \wedge AX a)$
 - ▶ $AG EF a$
- ⇒ Cannot be expressed = there does not exist an **equivalent** formula

Example

The CTL-formula $AG EF a$ cannot be expressed in LTL

- ▶ Proof by contradiction: assume $\varphi \equiv AG EF a$; let:

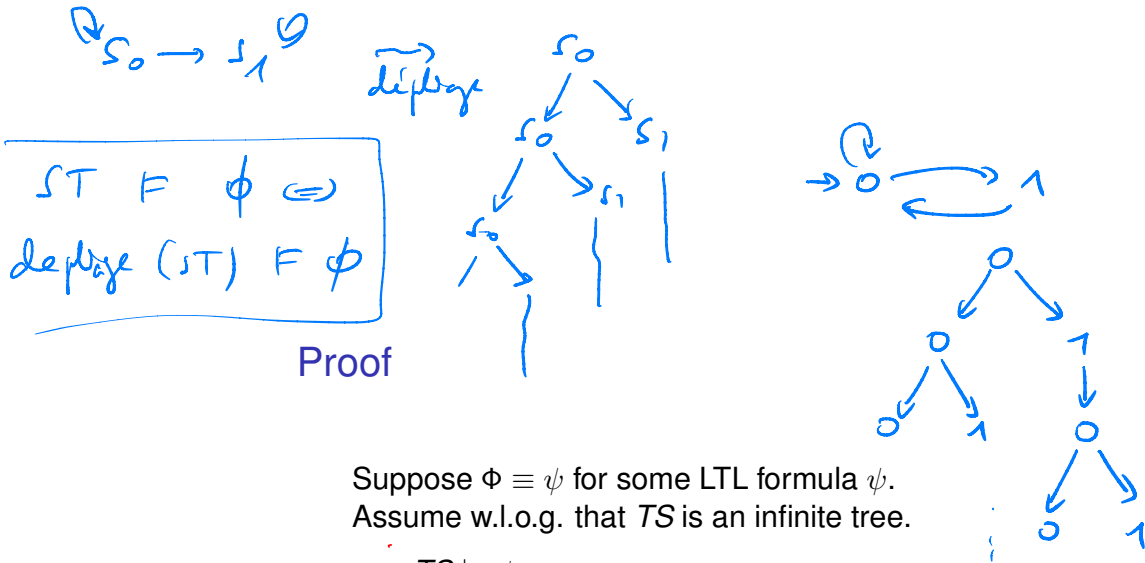


- ▶ $TS \models AG EF a$, and thus, by assumption, $TS \models \varphi$
- ▶ $Paths(TS') \subseteq Paths(TS)$, thus $TS' \models \varphi$
- ▶ **But** $TS' \not\models AG EF a$, because path $s^\omega \not\models G EF a$

Comparing LTL and CTL

Let Φ be a CTL-formula, and φ the LTL-formula obtained by eliminating all path quantifiers in Φ . Then: [Clarke & Draghicescu]

$\Phi \equiv \varphi$ or there does not exist any LTL-formula that is equivalent to Φ



Proof

Suppose $\Phi \equiv \psi$ for some LTL formula ψ .
Assume w.l.o.g. that TS is an infinite tree.

$TS \models \psi$

- LTL iff for all paths π in TS , π satisfies ψ
- iff for all paths π in TS , the transition system TS_π with the single path π satisfies ψ



AGa
EGa

- iff for all paths π in TS , TS_π satisfies Φ (because $\Phi \equiv \psi$)
- iff for all paths π in TS , TS_π satisfies φ (because there is only a single path)
- iff for all paths π in TS , π satisfies φ
- iff $TS \models \varphi$