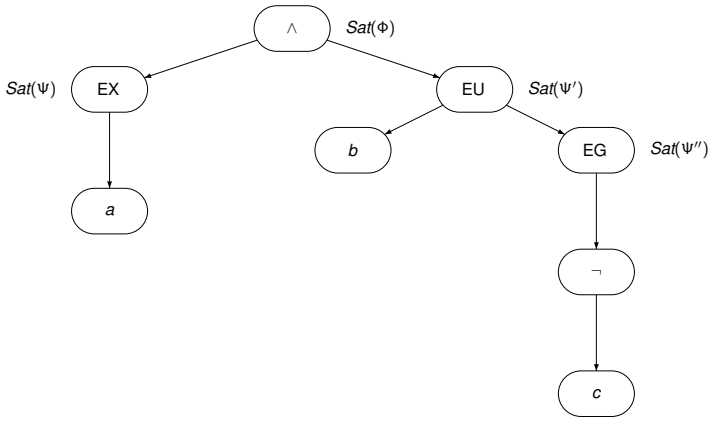
Model checking CTL

- How to check whether state graph TS satisfies CTL formula $\widehat{\Phi}$?
  - convert the formula $\widehat{\Phi}$ 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

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

  - 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



$$\Phi = \underbrace{\mathrm{EX}\, a}_{\Psi} \wedge \underbrace{\mathrm{E}\,(b\,\mathrm{U}\,\underbrace{\mathrm{EG}\,\neg c}_{\Psi''})}_{\Psi'} \quad .$$

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:

$$
\begin{aligned}
Sat(\text{true}) &= S \\
Sat(a) &= \{ q \in S \mid a \in L(q) \}, \text{ for any } a \in AP \\
Sat(\Phi \wedge \Psi) &= Sat(\Phi) \cap Sat(\Psi) \\
Sat(\neg\Phi) &= S \setminus Sat(\Phi) \\
Sat(\text{EX}\,\Phi) &= \{ q \in S \mid Post(q) \cap Sat(\Phi) \neq \emptyset \}
\end{aligned}
$$

for a given finite transition system with states S

Characterization of *Sat* (2)

$Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$ is the **smallest** subset T of S, such that:
(1) $Sat(\Psi) \subseteq T$ and
(2) $(q \in Sat(\Phi)$ and $Post(q) \cap T \neq \emptyset) \;\Rightarrow\; q \in T$

- We show that for any T that satisfies (1) and (2), we have $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi)) \subseteq T$.
- Let $s \in Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$.
- If $s \in Sat(\Psi)$, then, by (1), $s \in T$.
- Otherwise, there exists a path $\pi = s_0 s_1 s_2 \ldots$ starting in $s = s_0$ such that $\pi \models \Phi \,\mathrm{U}\, \Psi$.
- 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 $s_{n-1} \in Sat(\Phi)$.
- $\ldots$
- $s = s_0 \in T$ by (2), because $s_1 \in Post(s_0) \cap T$ and $s_0 \in Sat(\Phi)$.

Characterization of *Sat* (3)

$Sat(\mathrm{EG}\,\Phi)$ 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 $T \subseteq Sat(\mathrm{EG}\,\Phi)$.
- Let $s \in T$. We construct a path $\pi = s_0 s_1 s_2 \ldots$ 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 $s_2 \in Post(s_1) \cap T$.
  - $\ldots$.
- By (3), we have $s_i \in T \subseteq Sat(\Phi)$. Hence, $s \in Sat(\mathrm{EG}\,\Phi)$.

Computing $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$ (1)

$Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$ is the smallest set $T \subseteq Q$ such that:
(1) $Sat(\Psi) \subseteq T$ and (2) $(q \in Sat(\Phi)$ and $Post(q) \cap T \neq \emptyset) \;\Rightarrow\; q \in T$

- This suggests to compute $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$ iteratively:

  $T_0 = Sat(\Psi)$ 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:

$$T_0 \subseteq T_1 \subseteq \ldots \subseteq T_j \subseteq T_{j+1} \subseteq \ldots \subseteq Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$$

Computing $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$ (2)

- TS is finite, so for some $j \geq 0$ we have:
  $T_j = T_{j+1} = T_{j+2} = \ldots$
- 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.,
    $(q \in Sat(\Phi)$ 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 $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$ is the **smallest** set satisfying (1) and (2), $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi)) \subseteq T_j$ and thus $Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi)) = T_j$.
- Hence:
  $T_0 \subsetneq T_1 \subsetneq T_2 \subsetneq \ldots \subsetneq T_j = T_{j+1} = \ldots = Sat(\mathrm{E}(\Phi \,\mathrm{U}\, \Psi))$

Computing $Sat(\mathrm{E}(\Phi\,\mathrm{U}\,\Psi))$ (3)

Require: finite transition system with states S CTL-formula $\mathrm{E}(\Phi\,\mathrm{U}\,\Psi)$
Ensure: $Sat(\mathrm{E}(\Phi\,\mathrm{U}\,\Psi)) = \{\, q \in S \mid q \models \mathrm{E}(\Phi\,\mathrm{U}\,\Psi) \,\}$

$V := Sat(\Psi)$; {V administers states q with $q \models \mathrm{E}(\Phi\,\mathrm{U}\,\Psi)$}
$T := V$; {T contains the already visited states q with $q \models \mathrm{E}(\Phi\,\mathrm{U}\,\Psi)$}
while $V \neq \emptyset$ **do**
  let $q' \in V$;
  $V := V \setminus \{\, q' \,\}$;
  for all $q \in Pre(q')$ **do**
    if $q \in Sat(\Phi) \setminus T$ **then** $V := V \cup \{\, q \,\}$; $T := T \cup \{\, q \,\}$; **endif**
  end for
end while
return T

Computing $Sat(\mathrm{EG}\,\Phi)$

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

$T := Sat(\Phi)$; {T contains any q for which $q \models \mathrm{EG}\,\Phi$ 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

Alternative algorithm for $Sat(\mathrm{EG}\,\Phi)$

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

   $\Rightarrow$ any state in such SCC satisfies $\mathrm{EG}\,\Phi$
3. $q \models \mathrm{EG}\,\Phi$ is equivalent to "some SCC is reachable from q"
   - this search can be done in a backward manner

Time complexity

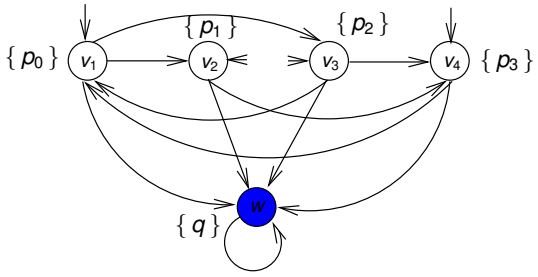
For transition system TS with N states and M edges,
and CTL formula Φ, the CTL model-checking problem $TS \models \Phi$
can be determined in time $\mathcal{O}(|\,\Phi\,| \cdot (N + M))$

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?

Hamiltonian path problem

$\Rightarrow$ LTL-formulae can be exponentially shorter than their CTL-equivalent



- Existence of Hamiltonian path in LTL:
  $\bigwedge_i \left(\Diamond p_i \ \wedge \ \Box(p_i \rightarrow \ \bigcirc\Box\neg p_i) \right)$
- In CTL, all possible (= 4!) routes need to be encoded

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):

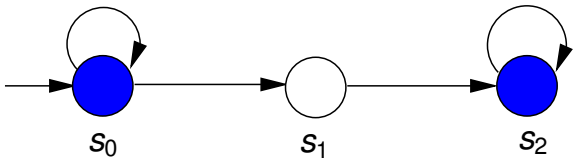
$$TS \models \Phi \quad \text{if and only if} \quad TS \models \varphi$$

Examples (1)

CTL-formula $\mathsf{A}\,\mathsf{G}\,\mathsf{A}\,\mathsf{F}\,a$ and LTL-formula $\mathsf{G}\,\mathsf{F}\,a$ are equivalent.

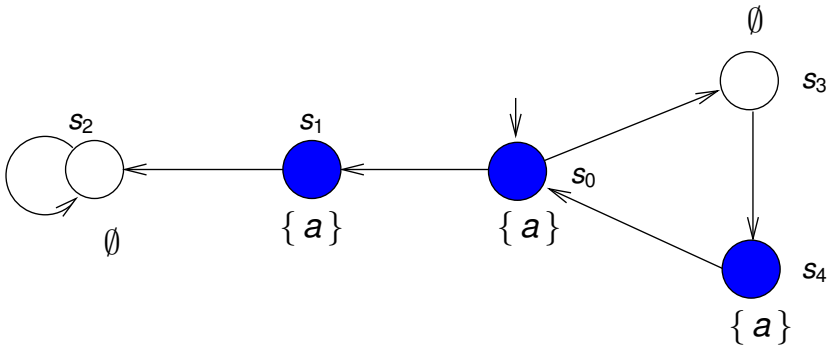
Examples (2)

AF AG a is not equivalent to F G a



Examples (3)

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



LTL and CTL are incomparable

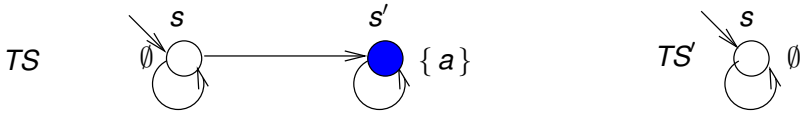
- Some LTL-formulas cannot be expressed in CTL, e.g.,
  - $\mathsf{F}\,\mathsf{G}\,a$
  - $\mathsf{F}\,(a \wedge \mathsf{X}\,a)$
- Some CTL-formulas cannot be expressed in LTL, e.g.,
  - $\mathsf{AF}\,\mathsf{AG}\,a$
  - $\mathsf{AF}\,(a \wedge \mathsf{AX}\,a)$
  - $\mathsf{AG}\,\mathsf{EF}\,a$

$\Rightarrow$ Cannot be expressed = there does not exist an equivalent formula

Example

The CTL-formula $\mathsf{AG}\,\mathsf{EF}\,a$ cannot be expressed in LTL

- Proof by contradiction: assume $\varphi \equiv \mathsf{AG}\,\mathsf{EF}\,a$; let:



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

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.

$TS \models \psi$

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

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 $TS \models \varphi$