CSC410

AZADEH FARZAN

FALL 2020
Step 3: model checking against an LTL/CTL property
For LTL, you would need to know/learn about automata on infinite words ....
CTL Model Checking
**CTL Expansion Laws**

<table>
<thead>
<tr>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \exists (\Phi \cup \Psi) )</td>
</tr>
<tr>
<td>( \exists \Diamond \Phi )</td>
</tr>
<tr>
<td>( \exists \Box \Phi )</td>
</tr>
</tbody>
</table>

**Lemma.** Until is the **least** solution to the expansion law.

The following equation has many solutions:

\[
F \equiv \Psi \lor (\Phi \land \exists \circ F)
\]

Until is the smallest **set** that satisfies this equation.
Idea

\[(98737 \otimes 593) \oplus (\ominus 120)\]
The indicated formulae are fresh atomic propositions, i.e., \( \neg a \). The satisfaction sets for the leaves result directly from the labeling function. Consider the following state formula over the set \( AP \):

\[
\Phi = \exists \Diamond a \land \exists (b \lor \exists \Box \neg c)
\]

The satisfaction set for CTL formulae in ENF is of the following form:

\[
\text{Sat}(\Phi) = \exists \Diamond a \land \exists (b \lor \exists \Box \neg c)
\]

The satisfaction set for the leaves result directly from the labeling function. The above procedure thus only needs the satisfaction set for \( \Psi \) and \( \neg \Psi \). Note that \( \Psi \) and \( \exists \land \exists \circ \exists \Box \neg c \) can now be replaced by the atomic propositions in a similar way.

\( \text{Sat}(\Phi) \) : set of all states that satisfy \( \Phi \)
Let’s invent the algorithm together ....
Watch the white board lecture!
Recursive Rules for ENF

(a) \( \text{Sat}(\text{true}) = S \),

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

(c) \( \text{Sat}(\Phi \land \Psi) = \text{Sat}(\Phi) \cap \text{Sat}(\Psi) \),

(d) \( \text{Sat}(\neg \Phi) = S \setminus \text{Sat}(\Phi) \),

(e) \( \text{Sat}(\exists \bigcirc \Phi) = \{ s \in S \mid \text{Post}(s) \cap \text{Sat}(\Phi) \neq \emptyset \} \),

(f) \( \text{Sat}(\exists (\Phi \cup \Psi)) \) is the smallest subset \( T \) of \( S \), such that

(1) \( \text{Sat}(\Psi) \subseteq T \) and (2) \( s \in \text{Sat}(\Phi) \) and \( \text{Post}(s) \cap T \neq \emptyset \) implies \( s \in T \),

(g) \( \text{Sat}(\exists \square \Phi) \) is the largest subset \( T \) of \( S \), such that

(3) \( T \subseteq \text{Sat}(\Phi) \) and (4) \( s \in T \) implies \( \text{Post}(s) \cap T \neq \emptyset \).
Final words on Model Checking ...