CS 181u Applied Logic

Lecture 13

LTL Model Checking
The Big Picture

Reactive System Code satisfies \( \models \) Requirements

Transition System satisfies \( \models \) Temporal Logic Formula \( \phi \)

Model Checking
The goal of LTL Model Checking: given a transition system $\mathcal{M}$ and an LTL property $\phi$,

1. determine if $\mathcal{M} \models \phi$, and
2. if $\mathcal{M} \not\models \phi$, then give a counterexample execution path from $\mathcal{M}$. 
LTL Model Checking Algorithm

\[ \mathcal{M} \models \phi \iff \forall \pi \left[ \pi \models \phi \right] \]

LTL Model Checking

LTL Model Checking Algorithm Overview

1. Construct a Büchi automaton for \( \neg \phi, A_{\neg \phi} \).
2. Construct a Büchi automaton for \( \mathcal{M}, A_{\mathcal{M}} \).
3. Compute the automaton product \( A_{\neg \phi} \times A_{\mathcal{M}} \).
4. Check if \( A_{\neg \phi} \times A_{\mathcal{M}} \) has an accepting path.
   (a) No accepting path \( \Rightarrow \mathcal{M} \models \phi \).
   (b) An accepting path corresponds to a counterexample execution.
Büchi Automata

A Büchi Automaton, \( \mathcal{A} \), is a tuple
\[ \mathcal{A} = (\Sigma, S, \rightarrow, I, F) \]
where
- \( \Sigma \) is an alphabet of transition symbols,
- \( S \) is a set of states,
- \( \rightarrow \) is a transition relation,
- \( I \) is a set of initial states, and
- \( F \) is a set of accepting (a.k.a. Final) states.

A Büchi Automaton, \( \mathcal{A} \), accepts languages of infinite words. That is, for a Büchi automaton, \( \mathcal{A} \), \( \mathcal{L}(\mathcal{A}) \subseteq \Sigma^\omega \).

Acceptance condition: a Büchi automaton, \( \mathcal{A} \), accepts an infinite word \( u \) if there exists an execution path of \( \mathcal{A} \) when run on \( u \) that visits the set of states of \( F \) infinitely often.
LTL and Büchi Automata

Property: Any LTL formula for atomic propositions $AP$ has a Büchi automaton with alphabet $Σ = \mathcal{P}(AP)$.

Example 1: $φ = G p, \ AP = \{p\}$

Example 2: $φ = G p, \ AP = \{p, q\}$

Example 3: $φ = GF p, \ AP = \{p\}$

Example 4: $φ = GF p, \ AP = \{p, q\}$
A non-deterministic Büchi automaton allows multiple outgoing transitions with the same label. Acceptance condition is the same as before.

Example 3': $\phi = GF\ p$, $AP = \{p\}$

Non-determinism: for example, from state 0, this automaton can either stay at state 0 or go to state 1 on transition labeled $\{p\}$.

For any sequence of sets of propositions that always has a $p$ in the future, there is a corresponding execution path in $A_{GFp}$ that visits state 1 infinitely often.
1. LTL to BA conversion

There is an algorithm that constructs a Büchi automaton (BA) from any LTL formula. However, we will not give it in this lecture.

There is an online tool for converting LTL to BA:

http://www.lsv.fr/~gastin/ltl2ba/index.php
Given a transition system $\mathcal{M} = (S, \rightarrow, I)$ and labelling function $L : S \rightarrow AP$, we can construct a Büchi automaton $A_{\mathcal{M}} = (\Sigma, S', \Rightarrow, I', F)$ where

- $\Sigma = \mathcal{P}(AP)$
- $S' = S \cup \{i\}$ (new initial state $i$)
- $\rightarrow$ is a transition relation,
- $I' = \{i\}$ (state $i$ is the only initial state),
- $F = S'$ (all states are accepting states), and
- the new transition relation, $\Rightarrow$ is as defined in the following slides.
2. Transition System to Büchi Automata

Convert the transition relation, $\rightarrow$, in system $\mathcal{M}$ to the transition relation, $\Rightarrow$, in the Büchi automaton, $\mathcal{A}_\mathcal{M}$.

Informally: add a new init state, move the labels from any state to the incoming transitions for that state. We will illustrate two simple rules to do this:

<table>
<thead>
<tr>
<th></th>
<th>In $\mathcal{M}$</th>
<th>In $\mathcal{A}_\mathcal{M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-initial states</td>
<td>$S_j \xrightarrow{l} S_k$</td>
<td>$S_j \xrightarrow{l} S_k$</td>
</tr>
<tr>
<td>initial states</td>
<td>$\xrightarrow{l} S_k$</td>
<td>$i \xrightarrow{l} S_k$</td>
</tr>
</tbody>
</table>
3. Büchi Automata Product

Given Büchi automata
\[ \mathcal{A}_1 = (\Sigma, S_1, \rightarrow_1, I_1, F_1) \]
\[ \mathcal{A}_2 = (\Sigma, S_2, \rightarrow_2, I_2, F_2) \]

\[ \mathcal{A}_1 \times \mathcal{A}_2 = (\Sigma, S, \rightarrow, I, F) \text{ where} \]

- \( S = S_1 \times S_2 \)
- \( I = I_1 \times I_2 \)
- \( F = \{(f_1, f_2) : f_1 \in F_1 \land f_2 \in F_2\} \),
- \( \rightarrow \) is defined for two states and a label \( l \) when \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \) agree on \( l \). To illustrate:

\[ S_1 j \xrightarrow{l} S_1 k \quad \text{in } \mathcal{A}_1 \]
\[ S_2 r \xrightarrow{l} S_2 t \quad \text{in } \mathcal{A}_2 \]
\[ (S_1 j, S_2 r) \xrightarrow{l} (S_1 k, S_2 t) \quad \text{in } \mathcal{A}_1 \times \mathcal{A}_2 \]

---

\( S_1 j \) and \( S_2 r \) have the same label.

---
A Complete Example

Consider this transition system, $\mathcal{M}$, with $AP = \{p\}$

We want to check the property $\phi = F G \neg p$
A Complete Example

Consider this transition system, $\mathcal{M}$, with $AP = \{p\}$

We want to check the property $\phi = F G \neg p$

Step 1: Construct $A_{\neg \phi}$. $\neg \phi = \neg F G \neg p = G F p$

This is the same NBA we saw earlier in the slides (Example 3').
A Complete Example

Consider this transition system, $\mathcal{M}$, with $AP = \{p\}$

We want to check the property $\phi = F \; G \; \neg p$

Step 2: Construct $A_{\mathcal{M}}$.
A Complete Example

Consider this transition system, $\mathcal{M}$, with $AP = \{p\}$

We want to check the property $\phi = F G \neg p$

Step 2: Construct $A_{\mathcal{M}}$. 

\[ 
\begin{array}{c}
0 \\
\emptyset \\
\{p\} \\
\emptyset
\end{array}
\quad \rightarrow 
\begin{array}{c}
1 \\
\{p\} \\
\emptyset
\end{array}
\quad \rightarrow 
\begin{array}{c}
2
\end{array}
\quad \rightarrow 
\begin{array}{c}
i
\emptyset
\end{array}
\quad \rightarrow 
\begin{array}{c}
0 \\
\emptyset \\
\{p\} \\
\emptyset
\end{array}
\quad \rightarrow 
\begin{array}{c}
1 \\
\{p\} \\
\emptyset
\end{array}
\quad \rightarrow 
\begin{array}{c}
2
\end{array}
\quad \rightarrow 
\begin{array}{c}
i
\emptyset
\end{array}
\]
A Complete Example

Consider this transition system, $\mathcal{M}$, with $AP = \{p\}$

We want to check the property $\phi = F G \neg p$

Step 2: Construct $A_\mathcal{M}$. 
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$. 

\begin{center}
\begin{tikzpicture}[node distance=2cm,>=latex,thick]
  \node[initial,state] (0) {$0$};
  \node[state] (1) {$1$};
  \node[state] (2) {$2$};
  \node[state] (r) {};  
  \path[->]
  (0) edge [loop above] node {$\emptyset, \{p\}$} (0)
  (0) edge [loop below] node {$\emptyset$} (0)
  (0) edge [loop above right] node {$\emptyset, \{p\}$} (0)
  (0) edge [loop right] node {$\emptyset$} (0)
  (1) edge [loop above] node {$\emptyset, \{p\}$} (1)
  (1) edge [loop below] node {$\emptyset$} (1)
  (1) edge [loop above right] node {$\emptyset, \{p\}$} (1)
  (1) edge [loop right] node {$\emptyset$} (1)
  (2) edge [loop above right] node {$\emptyset, \{p\}$} (2)
  (2) edge [loop below] node {$\emptyset$} (2)
  (2) edge [loop right] node {$\emptyset$} (2)
  (2) edge [loop above] node {$\emptyset$} (2)
  (0) edge [right] node {$\{p\}$} (1)
  (1) edge [right] node {$\{p\}$} (2)
  (2) edge [left] node {$\{p\}$} (0)
\end{tikzpicture}
\end{center}
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.
A Complete Example

Step 3: Construct $A_{\mathcal{M}} \times A_{\neg \phi}$.
Step 3: Construct $A_M \times A_{\neg \phi}$.
A Complete Example

Step 3: Construct $A_M \times A_{\neg\phi}$. 
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.
A Complete Example

Step 3: Construct $A_M \times A_{\neg\phi}$.
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$. 
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$. 
A Complete Example

Step 3: Construct $A_M \times A_{-\phi}$. 
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$. 

![Diagram](image-url)
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$. 
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$. 
Step 3: Construct $A_M \times A_{\neg \phi}$. 

A Complete Example
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.
Step 3: Construct $A_M \times A_{\neg \phi}$. 
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.
Step 3: Construct $A_M \times A^{-\phi}$. 

A Complete Example
A Complete Example

**Step 3:** Construct $A_M \times A_{\neg \phi}$.

**Step 4:** Is there an accepting path (a path that visits an accept state infinitely often)?
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.

Step 4: Is there an accepting path (a path that visits an accept state infinitely often)?

Yes. $i0, 00, 10, (20, 11)^\omega$
A Complete Example

Step 3: Construct $A_M \times A_{\neg \phi}$.

Step 4: Is there an accepting path (a path that visits an accept state infinitely often)?

Yes. $i0, 00, 10, (20, 11)^\omega$

This corresponds to $0, (1, 2)^\omega$ in the original transition system $M$. Since we have found a counter example path, $M \models \neg \phi$. 