Recap

Model

- Finite-state Machine
- Infinite runs

Specification

- Linear Time Logic (LTL)
- Computation Tree Logic (CTL)
- CTL*
Recap

LTL

- Unique successors
- Infinite runs (words)
- Operators:
  - Finally $F$
  - Globally $G$
  - Next $X$
  - Until $U$
LTL Model Checking Problem

- LTL model checking seeks to answer the question
  \[ M \models \phi \] hold?
- or, equivalently:
  \[ \forall \phi \in \text{Path}(M). \pi_0 \models \phi \] hold?
- The universal quantification is over the infinite set of paths, and each path is infinitely long
- How can we check infinitely many paths?
Language of Transition System

- For a Kripke structure $M = \langle S, S_0, R, L \rangle$
- Let’s consider the set of states $S$ as an alphabet $\Sigma$
- Each infinite path $\pi$ is then an infinite word
- The set of all paths of $M$ is the language $L(M)$ accepted by $M$

Example.

$$M$$

- $S_0$$  \rightarrow$$  S_1$
- $S_1$$ \rightarrow$$  S_2$
- $S_2$$ \rightarrow$$  S_0$

$$L(M)$$

- $\{S_0 S_1 S_2 S_2 S_2 \cdots$
- $S_0 S_1 S_0 S_1 S_2 S_2 S_2 \cdots$
- $S_0 S_1 S_0 S_1 S_0 S_1 S_2 S_2 S_2 \cdots$
- $\cdots,$
- $S_0 S_1 S_0 S_1 S_0 S_1 S_0 S_1 S_0 S_1 S_0 S_1 \cdots$ }
Automata over Finite Traces

- (Regular) Finite automaton with accepting states
- All finite traces (words) that take the automaton into the accepting state are “in its language”
- But behaviors (and traces) have infinite length
- So we need a new notion of acceptance
• $\omega$-automata: accepts (or reject) words of infinite length
• An $\omega$-word is an infinite sequence of letters
• The set of all $\omega$-words is denoted by $\Sigma^\omega$
• There are different kinds of $\omega$-automata
  - Büchi automata, Rabin automata, Street automata, parity automata, ...
• (non-deterministic) Büchi automata are commonly used for model checking
• They express the $\omega$-regular languages
Büchi Automata

- Same syntax as DFAs and NFAs, but different acceptance condition
- A run of a Büchi automaton on an $\omega$-word is an infinite sequence of states and transitions
- A run is accepting if it visits the set of final states infinitely often
- Final states renamed to accepting states

![Büchi Automaton Diagram]

Language of DFA (regular expression):

$$(a + b)^* (ab)^*$$

If interpret as a Büchi automaton over infinite words: ($\omega$-regular expression):

$$(a + b)^* (ab)^\omega$$
A (non-deterministic) Büchi automaton $\langle S, \Sigma, \rightarrow, S_0, A \rangle$ consists of:

- $S$ a finite set of states
- $\Sigma$ an alphabet
- $\rightarrow \subseteq S \times \Sigma \times S$ transition relation
- $S_0 \subseteq S$ set of initial states
- $A \subseteq S$ set of accepting states

An infinite word is accepted by a Büchi automaton iff there is a run of the automaton on which some accepting state is visited infinitely often.
Exercise

Construct Büchi automata accepting the following languages over \( \Sigma = \{a, b, c\} \)

- \( L_1 = \{ \alpha \in \Sigma^\omega \mid \alpha \text{ contains } ab \text{ exactly once} \} \)
- \( L_2 = \{ \alpha \in \Sigma^\omega \mid \alpha \text{ contains } ab \text{ at least once} \} \)
- \( L_3 = \{ \alpha \in \Sigma^\omega \mid \alpha \text{ contains } ab \text{ infinitely often} \} \)
- \( L_4 = \{ \alpha \in \Sigma^\omega \mid \alpha \text{ contains } ab \text{ only finitely often} \} \)
From LTL Properties to Büchi Automata

- Size of the property automaton can be exponential in the size of the LTL formula
  - Recall the complexity of LTL model checking
• Introduce a new initial state
• Move the labels on a state to the incoming edge(s)
• Make all states final
Büchi automata are closed under

- Union (similar to the case of finite automata)
- Intersection
- Complement
Büchi Automata: Language Emptiness Check

- Given a Büchi automaton, is the language accepted by the automaton empty?
- i.e., does it accept any string?
- A Büchi automaton accepts a string when the corresponding run visits an accepting state infinitely often

To check emptiness:

- Look for a cycle which contains an accepting state and is reachable from the initial state
- Find a strongly connected component that contains an accepting state, and is reachable from the initial state
- If no such cycle can be found the language accepted by the automaton is empty
We reformulate the LTL model checking problem to:

\[ L(M) \cap \overline{L(\phi)} = \emptyset \]

Now:

1. Observe that \( \overline{L(\phi)} = L(\neg \phi) \)
2. Let \( A_\phi \) be a Büchi automaton such that \( L(\phi) = L(A_\phi) \)
3. Compute the product of \( M \) and \( A \)

\[ L(M \times A) = L(M) \cap L(A) \]
4. So, to check \( M \models \phi \), instead check

\[ L(M \times A_{\neg \phi}) = \emptyset \]