Decision Properties of Regular Languages

General Discussion of “Properties” Membership, Emptiness, Etc.
Properties of Language Classes

• A *language class* is a set of languages.
• We have one example: the regular languages.
• We’ll see many more in this class.
• Language classes have two important kinds of properties:
  1. Decision properties.
  2. Closure properties.
Representation of Languages

• Representations can be formal or informal.

• **Example** (formal): represent a language by a RE or DFA defining it.

• **Example** (informal): a logical or prose statement about its strings:
  
  • \( \{0^n1^n \mid n \text{ is a nonnegative integer}\} \)
  
  • “The set of strings consisting of some number of 0’s followed by the same number of 1’s.”
Decision Properties

• A *decision property* for a class of languages is an algorithm that takes a formal description of a language (e.g., a DFA) and tells whether or not some property holds.

• **Example**: Is language $L$ empty?
Subtle Point: Representation Matters

• You might imagine that the language is described informally, so if my description is “the empty language” then yes, otherwise no.

• But the representation is a DFA (or a RE that you will convert to a DFA).

• Can you tell if \( L(A) = \emptyset \) for DFA A?
Closure Properties

- A closure property of a language class says that given languages in the class, an operator (e.g., union) produces another language in the same class.

- Example: the regular languages are obviously closed under union, concatenation, and (Kleene) closure.

- Use the RE representation of languages.
Why Closure Properties?

1. Helps construct representations.
2. Helps show (informally described) languages not to be in the class.
Example: Use of Closure Property

- We can easily prove $L_1 = \{0^n1^n \mid n \geq 0\}$ is not a regular language.
- $L_2 = \text{the set of strings with an equal number of 0's and 1's isn't either, but that fact is trickier to prove.}$
- Regular languages are closed under $\cap$.
- If $L_2$ were regular, then $L_2 \cap L(0^*1^*) = L_1$ would be, but it isn't.
The Membership Question

• Our first decision property is the question: “is string \( w \) in regular language \( L \)?”

• Assume \( L \) is represented by a DFA \( A \).

• Simulate the action of \( A \) on the sequence of input symbols forming \( w \).
Example: Testing Membership

0 1 0 1 1

Next symbol

Start 0

Current state

A 1 B 1 C 0,1
Example: Testing Membership

0 1 0 1 1

Next symbol

0

Current state

Start

0

0,1
Example: Testing Membership

\[0 \ 1 \ 0 \ 1 \ 1\]

Next symbol

Current state

Start

A \[\overset{0}{\rightarrow}\] B \[\overset{1}{\rightarrow}\] C

B \[\overset{1}{\rightarrow}\] C

C \[\overset{0,1}{\rightarrow}\]
Example: Testing Membership
Example: Testing Membership

```
0 1 0 1 1
```

Next symbol

Current state

Start

A

B

C

0

1

0,1
Example: Testing Membership

0 1 0 1 1

Next symbol

Current state
What if the Regular Language Is not Represented by a DFA?

- There is a circle of conversions from one form to another:
The Emptiness Problem: An Algorithm

- Given a regular language, does the language contain any string at all?
- Assume representation is DFA.
- Construct the transition graph.
- Compute the set of states reachable from the start state.
- If any final state is reachable, then yes, else no.