CSE 322: Regular Expressions and Finite Automata

✦ Last Time: Definition of a Regular Expression
   ✓ R is a regular expression iff R is a string over \( \Sigma \cup \{ \varepsilon, \emptyset, (, ), \cup, * \} \) and R is:
   1. Some symbol \( a \in \Sigma \), or
   2. \( \varepsilon \), or
   3. \( \emptyset \), or
   4. (R1 \( \cup \) R2) where R1 and R2 are regular exps., or
   5. R1\( ^* \) where R1 is a regular expression.

✦ Precedence: Evaluate * first, then \( \cup \), then *
   ✓ E.g. 0 \( \cup \) 1\( ^* \) = 0 \( \cup \) (1\( ^* \) (1*)) = \{0\} \( \cup \) \{1, 11, 111, … \}

Examples

✦ What is R for each of the following languages?
  1. \( L(R) = \{ w \mid w \text{ contains exactly two 0's} \} \)
  2. \( L(R) = \{ w \mid w \text{ contains at least two 0's} \} \)
  3. \( L(R) = \{ w \mid w \text{ contains an even number of 0's} \} \)
  4. \( L(R) = \{ w \mid w \text{ does not contain 00} \} \)
  5. \( L(R) = \{ w \mid w \text{ is a valid identifier in C} \} \)

Regular Expressions and Finite Automata

✦ What is the relationship between regular expressions and DFAs/NFAs?
✦ Specifically:
  1. Given a reg. exp. R, can we create an NFA N such that \( L(R) = L(N) \)?
  2. Given an NFA N (or its equivalent DFA M), can we come up with a reg. exp. R such that \( L(M) = L(R) \)?

I think so... do you??
From Regular Expressions to NFAs

✦ Problem: Given any regular expression \( R \), how do we construct an NFA \( N \) such that \( L(N) = L(R) \)?

✦ Soln.: Use the multi-part definition of regular expressions!!

---

From NFAs/DFAs to Regular Expressions

✦ Problem: Given any NFA \( N \), how do we construct a regular expression \( R \) such that \( L(N) = L(R) \)?

✦ Solution: First, convert NFA \( N \) to an equivalent DFA \( M \) to keep things simple. Then:

   ➤ Idea: Collapse 2 or more edges in \( M \) labeled with single symbols to a new edge labeled with an equivalent regular expression

   ➤ This results in a “generalized” NFA (GNFA)

   ➤ Our goal: Get a GNFA with 2 states (start and accept) connected by a single edge labeled with the required regular expression \( R \)

---

Next Class: How to pump up them strings…

✦ Next time: Beyond the regular world…

   ➤ Pumping Lemma for showing non-regularity of languages

✦ Things to do over the weekend:

   ➤ Finish reading Chapter 1
   ➤ Start (and finish?) homework #3
   ➤ Watch a Kevin Bacon movie (optional)
   ➤ Have a great weekend!