CSE 105
THEORY OF COMPUTATION

Spring 2016

http://cseweb.ucsd.edu/classes/sp16/cse105-ab/
Today's learning goals

- Decide whether or not a string is described by a given regular expression
- Design a regular expression to describe a given language
- Convert between regular expressions and automata
- Give an example of a non-regular language
- Outline two strategies for proving that there are non-regular languages
Reminders

• **HW 3** due Friday

• **Exam 1** next week
  - Wednesday April 20 evening
  - 8pm-9:50pm
  - **SOLIS 107** (seating chart will be released shortly)
  - One (double-sided) handwritten 3" by 5" index card for notes

• **Review session for exam 1 on Monday April 18, Peterson 108, 8-10**
Proofs of correctness

- Is $w$ in $L(M)$?
- Need to consider sequence of states visited by machine
Proofs of correctness

For $M = (Q, \Sigma, \delta, q_0, F)$ a DFA, with $\delta : Q \times \Sigma \rightarrow Q$ its transition function, define $\delta^* : Q \times \Sigma^+ \rightarrow Q$ recursively as

$$\delta^*(q, a) = \delta(q, a) \quad \text{for } a \in \Sigma$$
$$\delta^*(q, aw) = \delta(\delta(q, a), w) \quad \text{for } a \in \Sigma, w \in \Sigma^+$$

"A nonempty string $w$ is accepted by $M$ iff $\delta^*(q_0, w) \in F$" 

A. True
B. False
C. I don't know.
Regular languages

To prove that a set of strings over the alphabet $\Sigma$ is regular,

- Build a **DFA** whose language is this set.
- Build an **NFA** whose language is this set.
- Use the **closure properties** of the class of regular languages to construct this set from others known to be regular.
  - Union  
  - Intersection  
  - Complementation  
  - Concatenation  
  - Flip bits  
  - Kleene star
Inductive application of closure

R is a regular expression over $\Sigma$ if

1. $R = a$, where $a \in \Sigma$
2. $R = \varepsilon$
3. $R = \emptyset$
4. $R = (R_1 \cup R_2)$, where $R_1, R_2$ are themselves regular expressions
5. $R = (R_1 \circ R_2)$, where $R_1, R_2$ are themselves regular expressions
6. $(R_1^*)$, where $R_1$ is a regular expression.

Watch out for overloaded symbols!
Regular expressions

Conventions:
- $\Sigma$ is shorthand for $\{0, 1\}$ if $\Sigma = \{0, 1\}$
- Parentheses may be omitted
- Precedence: star, then concatenation, then union
- $R^+$ is shorthand for $RR^*$, $R^k$ is shorthand for $R$ concatenated with itself $k$ times
- Circle indicated concatenation may be omitted

Which of the following is not a regular expression over $\{0, 1\}$?
A. $(\Sigma \Sigma \Sigma \Sigma)^*$  
B. $\Sigma \cap 1$  
C. $1^* \emptyset 0$  
D. $\varepsilon \varepsilon$  
E. I don't know
Syntax $\rightarrow$ Languages

The language described by a regular expression, L(R):

- $L\left( \frac{1}{2}0\frac{1}{2}0\frac{1}{2}0\frac{1}{2}\right) = \{0110, 01\overline{1}0111, \varepsilon\overline{3} = \{w | w_1 = 4k, k \in \mathbb{Z}_{\geq 0}\} \}
- $L\left( (3 \cdot 3) \right) = \{ \varepsilon \neq \phi, \varepsilon \}
- $L\left( 1^* \phi 0 \right) = \{ \varepsilon = \phi = L(\phi) \}
- $L\left( \overline{1} \right) = \{ \}$
L(R)

Which of the following strings is **not** in the language described by

\[
( ( (00)^* (11) ) \cup 01 )^* 
\]

A. 00
B. 01
C. 1101
D. ε
E. I don't know
Let \( L \) be the language over \( \{a, b\} \) described by the regular expression

\[
((a \cup \emptyset) b^*)^*
\]

Which of the following is not true about \( L \)?

A. Some strings in \( L \) have equal numbers of a's and b's
B. \( L \) contains the string aaaaaa
C. a's never follow b's in any string in \( L \)
D. \( L \) can also be represented by the regular expression \((ab^*)^*\)
E. More than one of the above.
Regular expressions in practice

- **Compilers**: first phase of compiling transforms Strings to Tokens *keywords, operators, identifiers, literals*
  - One regular expression for each token type

- **Other software tools**: grep, Perl, Python, Java, Ruby, …
"Regular = regular"

**Theorem:** A language is regular if and only if some regular expression describes it.

**Lemma 1.55:** If a language is described by a regular expression, then it is regular.

**Lemma 1.60:** If a language is regular, then it is described by some regular expression.
L(R) to NFA (to DFA)

- Idea: basic regular expressions are easy to implement as DFA, for inductive step of definition, use closure under regular operations.
- E.g.: build NFA recognizing the language described by $(00 \cup 11)^*$
DFA to regular expression

- Idea: use intermediate model **GNFA** whose labels are regular expressions

- E.g.: build regular expression describing language recognized by

\[ 0^* (0u1)^* \]
All roads lead to … regular sets?

Are there any languages over \( \{0,1\} \) that are not regular?

A. Yes: a language that is recognized by an NFA but not any DFA.

A. Yes: there is some infinite language of strings over \( \{0,1\} \) that is not described by any regular expression.

B. No: all languages over \( \{0,1\} \) are regular because that's what it means to be a language.

C. No: for each set of strings over \( \{0,1\} \), some DFA recognizes that set.

A. I don't know.
Reminders

- **HW 3** due Friday
- **Exam 1** next week
  - Wednesday April 20 evening
  - 8pm-9:50pm
  - **SOLIS 107** (seating chart will be released shortly)
  - One (double-sided) handwritten 3" by 5" index card for notes

- **Review session for exam 1 on Monday April 18, Peterson 108, 8-10**