Lecture 5:

- More on pumping
- Regular expressions
- Regular expressions = regular languages

Reading:
Sipser Ch 1.3

Mark Bun
February 5, 2020
More on Pumping
Pumping Lemma (Formal)

Let $L$ be a regular language.

Then there exists a “pumping length” $p$ such that

For every $w \in L$ where $|w| \geq p$,

$w$ can be split into three parts $w = xyz$ where:

1. $|y| > 0$
2. $|xy| \leq p$
3. $xy^iz \in L$ for all $i \geq 0$
General Strategy for proving $L$ is not regular

Proof by contradiction: assume $L$ is regular. Then there is a pumping length $p$.

1. Find $w \in L$ with $|w| \geq p$
2. Show that $w$ cannot be pumped
3. Conclude $L$ must not have been regular
Claim: \( L = \{0^i 1^j | i > j \geq 0\} \) is not regular

Proof: Assume \( L \) is regular with pumping length \( p \)

1. Find \( w \in L \) with \( |w| \geq p \)

2. Show that \( w \) cannot be pumped

Formally

If \( w = xyz \) with \( |xy| \leq p \), then...

\[ y = 0^k, k > 0 \]
\[ x = 0^m, m \geq 0 \]
\[ z = 0^{p+1-(m+k)}, 1 \leq p \]

\[ yyyz = \underbrace{000000} \ldots \underbrace{111111} \]

\[ xyyyz = \underbrace{000000}_y \underbrace{000000}_y \underbrace{000000}_y \]

\[ xyyyz \notin L \]
Reusing a Proof

Pumping a language can be lots of work...
Let’s try to reuse that work!

How might we show that

\[ \text{BALANCED} = \{ w \mid w \text{ has an equal # of 0s and 1s} \} \]

is not regular?

\[ \{0^n1^n \mid n \geq 0\} = \text{BALANCED} \cap \{ w \mid \text{all 0s in } w \text{ appear before all 1s}\} \]

Not regular

Assume for contradiction \( \text{BALANCED} \) is regular \( \Rightarrow \) RHS regular (closed under \( \cap \)) \( \Rightarrow \) \( \text{BALANCED} \) not regular
Using Closure Properties

If $A$ is not regular, we can show a related language $B$ is not regular

$$B \cap C = A$$

(any of $\circ, \cup, \cap$) or, for one language, $\{\neg, R, *\}$

By contradiction: If $B$ is regular, then $B \cap C (= A)$ is regular. But $A$ is not regular so neither is $B$!
Example \( \cup_{\text{uneq}} C = \sum^* \quad C = A \cup B \)

\[ A = C \setminus B, \quad B = C \setminus A \]

Prove \( B = \{0^i1^j | i \neq j\} \) is not regular using nonregular language \( A = \{0^n1^n | n \geq 0\} \) and regular language \( C = \{w | \text{all 0s in } w \text{ appear before all 1s}\} \)

\( B = A \setminus AC \)

\( A = \overline{B} \cap C \)

\( B \) regular \( \implies \) \( A \) regular \( \implies \) non-regular
Regular Expressions
Regular Expressions

• A different way of describing regular languages
• A regular expression expresses a (possibly complex) language by combining simple languages using the regular operations

“Simple” languages: $\emptyset, \{\varepsilon\}, \{a\}$ for some $a \in \Sigma$

Regular operations:

**Union:** $A \cup B$

**Concatenation:** $A \circ B = \{ab \mid a \in A, b \in B\}$

**Star:** $A^* = \{a_1a_2...a_n \mid n \geq 0 \text{ and } a_i \in A\}$

\[= \varepsilon \cup A \cup AA \cup AAA \cup AAAAA \cup \ldots\]
Regular Expressions – Syntax

A regular expression $R$ is defined recursively using the following rules:

1. $\varepsilon$, $\emptyset$, and $a$ are regular expressions for every $a \in \Sigma$

2. If $R_1$ and $R_2$ are regular expressions, then so are $(R_1 \cup R_2)$, $(R_1 \circ R_2)$, and $(R_1^*)$

Examples: (over $\Sigma = \{a, b, c\}$)

$(a \circ b)$   $(((a \circ (b^*)) \circ c) \cup (((a^*) \circ b))^*)$  $(\emptyset^*)$
Regular Expressions – Semantics

$L(R)$ = the language a regular expression describes

\[ L(a^*) = (L(a))^* = (\varepsilon a^3)^* \]

1. \( L(\emptyset) = \emptyset \)
2. \( L(\varepsilon) = \{\varepsilon\} \)
3. \( L(a) = \{a\} \) for every \( a \in \Sigma \)
4. \( L((R_1 \cup R_2)) = L(R_1) \cup L(R_2) \)
5. \( L((R_1 \circ R_2)) = L(R_1) \circ L(R_2) \)
6. \( L((R_1^*)) = (L(R_1))^* \)

Example: \( L(((a^*) \circ (b^*))) = \{a^i b^j \mid i, j \geq 0, i + j \leq 3\} \)
Simplifying Notation

• Omit \( \circ \) symbol: \((ab) = (a \circ b)\)

\[ R_1R_2 = R_1 \circ R_2 \]

• Omit many parentheses, since union and concatenation are associative:

\[ (a \cup b \cup c) = (a \cup (b \cup c)) = ((a \cup b) \cup c) \]

• Order of operations: Evaluate star, then concatenation, then union

\[ ab^* \cup c = (a(b^*)) \cup c \]

\[ ((a^*) \circ (b^*)) = a^* b^* \]
Examples

Let $\Sigma = \{0, 1\}$

1. $\{w \mid w \text{ contains exactly one } 1\}$
   \[0^* 1 0^*\]

2. $\{w \mid w \text{ has length at least 3 and its third symbol is } 0\}$
   \[(011) (\overline{011})^* 0 (\overline{011})^*\]

3. $\{w \mid \text{every odd position of } w \text{ is } 1\}$
   \[(011)^* (\in \cup_0 011)^* \quad 0 \quad 0\]
Syntactic Sugar

- For alphabet $\Sigma$, the regex $\Sigma$ represents $L(\Sigma) = \Sigma$
- For regex $R$, the regex $R^+ = RR^*$

$R^* = R^+ \cup \varepsilon$

Not captured by regular expressions: Backreferences

\[1 \backslash 2\]
Equivalence of Regular Expressions, NFAs, and DFAs
Regular Expressions Describe Regular Languages

**Theorem**: A language $A$ is regular if and only if it is described by a regular expression

**Theorem 1**: Every regular expression has an equivalent NFA

**Theorem 2**: Every NFA has an equivalent regular expression
Regular expression -> NFA

Theorem 1: Every regex has an equivalent NFA
Proof: Induction on size of a regex

Base cases:

\[ R = \emptyset \]

\[ R = \varepsilon \]

\[ R = a \]
Regular expression -> NFA

Theorem 1: Every regex has an equivalent NFA
Proof: Induction on size of a regex

Inductive step:

\[ R = (R_1 \cup R_2) \]

\[ R = (R_1 R_2) \]

\[ R = (R_1^*) \]
Example

Convert $(1(0 \cup 1))^*$ to an NFA
Example

Simplified

= 1

= 0, 1