6.045J/18.400J: Automata, Computability and Complexity Prof. Nancy Lynch

Recitation 3: Regular Expressions and Non-regular Languages

February 19, 2003 *Susan Hohenberger*

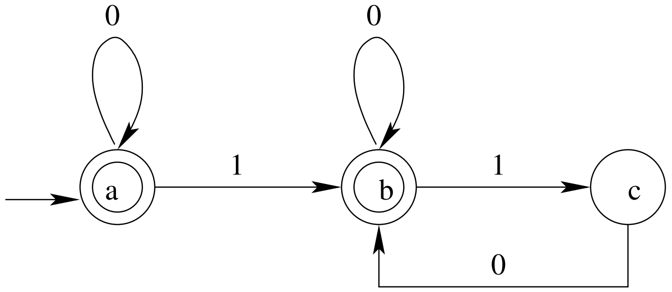
Problem 1: **Key terms.** Regular expression, generalized NFA, pigeon-hole principle, pumping lemma, pumping length, pumping up, pumping down.

Problem 2: **True or False?**

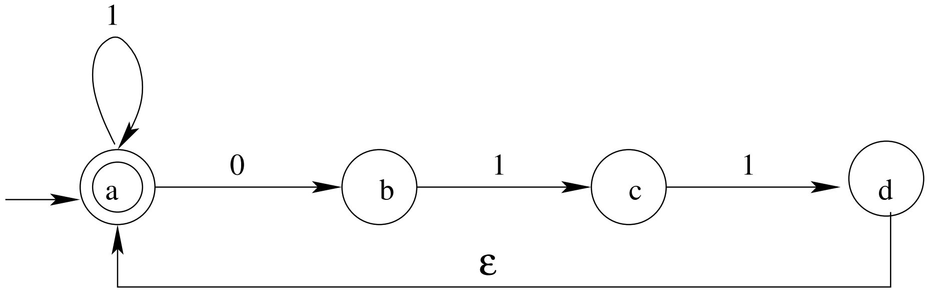
1. If L_1 and L_2 are regular, then $L_1 \cup L_2$ is regular.
2. If L_1 and L_2 are non-regular, then $L_1 \cap L_2$ is non-regular.
3. If L_1 is regular and L_2 is non-regular, then $L_1 \cup L_2$ is non-regular.
4. If L_1 is regular, L_2 is non-regular, and $L_1 \cap L_2$ is regular, than $L_1 \cup L_2$ is non-regular.
5. The following language is regular: The set of strings in $\{0, 1\}^*$ having the property that the number of 0's and the number of 1's differ by no more than 2.
6. The following language is regular: The set of strings in $\{0, 1\}^*$ havi'ng the property that in every prefix, the number of 0's and the number of 1's differ by no more than 2.

Problem 3: **Regular Expressions.** Write regular expressions for the following languages. The alphabet is $\{0, 1\}^*$.

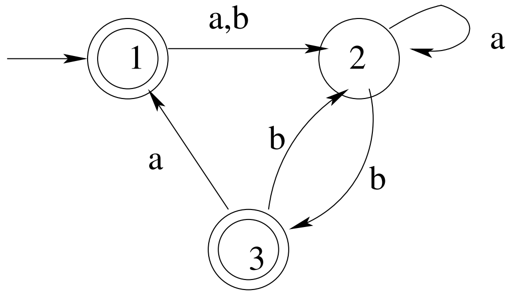
1. $A_1 = \{w|w$ contains at least two 0's$\}$.
2. $A_2 = \{w|w$ contains an even number of 0's$\}$.
3. $A_3 = \{w|w$ has no more than one occurence of the substring '00'$\}$.
4. $A_4 = \{w|w$ does not end with '01'$\}$.
5. $A_5 = \{w|w$ contains both '11' and '010' as substrings$\}$.
6. A_6:



7. A_7:



8. A_8: (from "fake" homework 2.5; here the alphabet is $\{a, b\}^*$.)



Problem 4: **Proving non-regularity: the Pumping Lemma.** Prove that the following languages are not regular.

1. $L_1 = \{0^n 1 0^{2n} | \; n \geq 0\}$.
2. $L_2 = \{0^i 1^j 0^k | \; k > i + j\}$.
3. $L_3 = \{x \in \{0, 1\}^* |$ no prefix of x has more 0s than 1s$\}$.
4. $L_4 = \{0^i 1^j | \; j$ is a multiple of $i\}$.
5. $L_5 = \{xx | \; x \in \{0, 1\}^*\}$.

Problem 5: **Unix "regular expressions".** Unix "regular expressions" have a slightly different syntax and set of operators than what we saw in class:

- Notational difference for union: $|$ instead of $\cup$
- $[abcde]$ is shorthand for $(a|b|c|d|e)$
- Additional unary operators and what they mean:
  $R?$: zero or one copies of R
  $R+$: at least one copy of R
  $R\{n\}$: exactly n copies of R
  $R\{n,\}$: at least n copies of R
  $R\{n, m\}$: between n and m copies of R

1. We would like to show that regexps with the above additional operators have the same expressive power as regular regexps. Show this by showing for each of the above, an equivalent regular expression using only regular operators. For example, $R?$ is equivalent to ($\epsilon \cup R$).

2. Unix "regular expressions" actually support also another operator, quoting from "man grep": "The backreference $\backslash n$, where n is a single digit, matches the substring previously matched by the nth parenthesized subexpression of the regular expression."

3. Show that with the backreference operator unix "regular expressions" are strictly more powerful then regular expressions, that is, they can represent non-regular languages. Do this by showing a unix "regular expression" using this operator for a non-regular language.