CSC236 Intro. to the Theory of Computation
Lecture 12: RE→NFA→DFA→RE
¬ pumping → ¬ RL

Amir H. Chinaei, Fall 2016
Office Hours: W 2-4 BA4222
ahchinaei@cs.toronto.edu
http://www.cs.toronto.edu/~ahchinaei/

Course page:
http://www.csf.toronto.edu/~cs236h/fall/index.html
Section page:
http://www.csf.toronto.edu/~cs236h/fall/amir_lectures.html


**review**

- last lecture
  - FSA (nondeterministic and deterministic) = RE
    - NFA → DFA → RE → NFA

- this week:
  - more on RE→NFA
  - application of pumping lemma in proving a language is not regular

**NFA, DFA, regex**

- NFA → DFA → regex → NFA
  - BASE CASES
    - regex → NFA
      - ∅
      - ε
      - b

**NFA, DFA, regex**

- NFA → DFA → regex → NFA
  - RECURSIVE CASES
    - regex
      - NFA
      - $r_1 + r_2$

**NFA, DFA, regex**

- NFA → DFA → regex → NFA
  - RECURSIVE CASES
    - regex
      - NFA
      - $r_1 \cdot r_2$

**NFA, DFA, regex**

- NFA → DFA → regex → NFA
  - RECURSIVE CASES
    - regex
      - NFA
      - $r_1^*$
**Example 104**

\[ 0 (0 + 1) * 1 \]

\[ NFA \Rightarrow DFA \Rightarrow \text{regex} \Rightarrow NFA \]

\[ \text{Example 105} \]

\[ 0 (0 + 1) * 1 \text{ revisited (2nd algorithm)} \]

- make a transition from each symbol of alphabet to the next state
- make an e-transition from each brace to the next state
- make 3 e-transitions for each *
- make 2 e-transitions for each +

\[ NFA \Rightarrow DFA \Rightarrow \text{regex} \Rightarrow NFA \]

**NFA=DFA=regex**

- \[ NFA \Rightarrow DFA \Rightarrow \text{regex} \Rightarrow NFA \]
  - nicely done!
- \[ NFA \Rightarrow DFA \Rightarrow \text{regex} \Rightarrow NFA \]
  - 
- \[ NFA \Rightarrow DFA \Rightarrow \text{regex} \Rightarrow NFA \]
  - analogy:

**pumping lemma**

- If \( L \) is RL, then \( \exists p \geq 1 \) such that \( \forall \omega \in L, |\omega| \geq p, \omega = xyz \):
  - \( |xy| \leq p \)
  - \( |y| > 0 \)
  - \( \forall k \geq 0, xy^kz \in L \)

**application of pumping lemma is in proving non-regularity**

- assume the language is regular, apply the pumping lemma and run to a contradiction
- note:

**Example 105**

- Prove \( L = \{ \omega \in \Sigma^* | \omega = a^n b^n \ n \geq 0 \} \) is not regular.

**final notes**

- you have enhanced your analytical skills, in particular in systematic reasoning, proofs, program correctness, and simple computational models
- next?
  - CSC263: more algorithm analysis & data structures
  - CSC373: more algorithms complexities and paradigms
  - ...
  - CSC448: more formal languages and automata
- if I can be of any help, drop me a line or stop by BA4222.