322 Midterm Review

- **Formal Languages**
  - Alphabet ($\Sigma$)
  - String ($\Sigma^*$)
  - Length ($|x|$)
  - Empty String ($\varepsilon$)
  - Empty Language ($\emptyset$)

- **Language/String Operations**
  - “Regular” Operations:
    - Union ($\cup$)
    - Concatenation ($\cdot$)
    - (Kleene) Star ($^*$)
  - Other:
    - Intersection
    - Complement
    - Reversal
    - ...
Finite Defns of Infinite Languages

- English, mathematical
- DFAs
  - States
  - Start states
  - Accept states
  - Transitions (δ function)
  - M accepts \( w \in \Sigma^* \)
  - M recognizes \( L \subseteq \Sigma^* \)
- Nondeterminism
- NFAs
  - Transitions (δ relation)
    - Missing out-edges
    - \( \varepsilon \)-moves
    - Multiple out-edges
  - N accepts \( w \in \Sigma^* \)
  - N recognizes \( L \subseteq \Sigma^* \)
- Regular Expressions
  - \( \emptyset, a \in \Sigma, \cup, \cdot, *, ( ) \)
- GNFAs
Key Results, Constructions, Methods

• L is regular iff it is:
  – Recognized by a DFA
  – Recognized by a NFA
  – Recognized by a GNFA
  – Defined by a Regular Expr

Proofs:

  GNFA → Reg Expr
    (Kleene/Floyd/Warshall: $R_{ij} R_{jj}^* R_{jk}$)

  Reg Expr → NFA
    (join NFAs w/ $\varepsilon$-moves)

  NFA → DFA
    (subset construction)

• The class of regular languages is closed under:
  – Regular ops: union, concatenation, star
  – Also: intersection, complementation,
    (& reversal, prefix, no-prefix, …)

• NOT closed under $\subseteq$, $\supseteq$

• Also: Cross-product construction (union, …)
Non-Regular Languages

<table>
<thead>
<tr>
<th>Key idea: once M is in some state q, it doesn’t remember how it got there.</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.g. “hybrids”:</td>
</tr>
<tr>
<td>if $xy \in L(M)$ and</td>
</tr>
<tr>
<td>$x$, $x'$ both go to q, then</td>
</tr>
<tr>
<td>$x'y \in L(M)$ too.</td>
</tr>
<tr>
<td>E.g. “loops”:</td>
</tr>
<tr>
<td>if $xyz \in L(M)$ and</td>
</tr>
<tr>
<td>$x$, $xy$ both go to q, then</td>
</tr>
<tr>
<td>$xy^iz \in L(M)$ for all $i \geq 0$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cor: Pumping Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Important examples:</td>
</tr>
<tr>
<td>$L_1 = { a^n b^n</td>
</tr>
<tr>
<td>$L_2 = { w</td>
</tr>
<tr>
<td>$L_3 = { ww</td>
</tr>
<tr>
<td>$L_4 = { ww^R</td>
</tr>
<tr>
<td>$L_5 = { \text{balanced parens} }$</td>
</tr>
</tbody>
</table>

| Also: closure under $\cap$, complementation sometimes useful:       |
| $L_1 = L_2 \cap a^*b^*$                                             |

| PS: don’t say “Irregular”                                           |
Applications

• “globbing”
  – lpr *.txt

• pattern-match searching:
  – grep “Ruzzo.*terrific” *.txt

• Compilers:
  – $Id ::= \text{letter} ( \text{letter|digit} )^*$
  – $\text{Int ::= digit} \ \text{digit}^*$
  – $\text{Float ::= } d \ \text{d}^* . \ \text{d}^* ( \ \varepsilon | E \ d \ \text{d}^* )$