CSE 322

Exam Reviews
## Basic Concepts

<table>
<thead>
<tr>
<th>Formal Languages</th>
<th>Language/String Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Alphabet (Σ)</td>
<td>- “Regular” Operations:</td>
</tr>
<tr>
<td>- String (Σ*)</td>
<td>- Union (∪)</td>
</tr>
<tr>
<td>- Length (</td>
<td>x</td>
</tr>
<tr>
<td>- Empty String (ε)</td>
<td>- (Kleene) Star (*)</td>
</tr>
<tr>
<td>- Empty Language (Ø)</td>
<td>- Other:</td>
</tr>
<tr>
<td></td>
<td>- Intersection</td>
</tr>
<tr>
<td></td>
<td>- Complement</td>
</tr>
<tr>
<td></td>
<td>- Reversal</td>
</tr>
<tr>
<td></td>
<td>- ...</td>
</tr>
</tbody>
</table>
## Finite Defns of Infinite Languages

- **English, mathematical**
- **DFAs**
  - States
  - Start states
  - Accept states
  - Transitions ($\delta$ function)
  - $M$ accepts $w \in \Sigma^*$
  - $M$ recognizes $L \subseteq \Sigma^*$
- **Nondeterminism**
- **NFAs**
  - Transitions ($\delta$ relation)
    - Missing out-edges
    - $\varepsilon$-moves
    - Multiple out-edges
  - $N$ accepts $w \in \Sigma^*$
  - $N$ recognizes $L \subseteq \Sigma^*$
- **Regular Expressions**
  - $\emptyset$, $a \in \Sigma$, $\cup$, $\cdot$, $\ast$, ( )
- **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 $\rightarrow$ Reg Expr
  (Kleene/Floyd/Warshall: $R_{ij} R_{jj}^* R_{jk}$)
- Reg Expr $\rightarrow$ NFA
  (join NFAs w/ $\varepsilon$-moves)
- NFA $\rightarrow$ 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

• Key idea: once M is in some state q, it doesn’t remember how it got there.
  E.g. “hybrids”:
  if xy ∈ L(M) and
  x, x’ both go to q, then
  x’y ∈ L(M) too.
  
  E.g. “loops”:
  if xyz ∈ L(M) and
  x, xy both go to q, then
  xy^i z ∈ L(M) for all i ≥ 0.

• Cor: Pumping Lemma

• Important examples:
  L_1 = \{ a^n b^n | n > 0 \}
  L_2 = \{ w | \#_a(w) = \#_b(w) \}
  L_3 = \{ ww | w \in \Sigma^* \}
  L_4 = \{ ww^R | w \in \Sigma^* \}
  L_5 = \{ balanced \ parens \}

• Also: closure under ∩, complementation sometimes useful:
  – L_1 = L_2 ∩ a*b*

• PS: don’t say “Irregular”
Applications

• “globbing”
  – lpr *.txt

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

• Compilers:
  – Id ::= letter ( letter|digit )*  
  – Int ::= digit digit*  
  – Float ::=  
    d d* . d* ( ε | E d d* )  
  – (but not, e.g. expressions with nested, balanced parens, or variable names matched to declarations)

• Finite state models of circuits, control systems, network protocols, API’s, etc., etc.
Context-Free Grammars

- Terminals, Variables/Non-Terminals
- Start Symbol $S$
- Rules $\rightarrow$
- Derivations $\Rightarrow$, $\Rightarrow^*$
- Left/right-most derivations
- Derivation trees(parse trees)
- Ambiguity, Inherent ambiguity

- A key feature: recursion/nesting/matching, e.g.

  $$S \rightarrow (S)S \mid \varepsilon$$
Pushdown Automata

- States, Start state, Final states, stack
- Terminals ($\Sigma$), Stack alphabet ($\Gamma$)
- Configurations, Moves, |--, |--*, push/pop
Main Results

- **Closure**: union, dot, *, (Reversal)
  - every regular language is CFL
- **Non-Closure**: Intersection, complementation
- **Equivalence of CFG & PDA**
  - $\mathsf{CFG} \subseteq \mathsf{PDA}$:
    - top-down(match/expand), bottom-up (shift/reduce)
  - $\mathsf{PDA} \subseteq \mathsf{CFG}$: $A_{pq}$
- **Pumping Lemma & non-CFL’s**
- **Deterministic PDA $\neq$ Nondeterministic PDA**
Important Examples

• Some Context-Free Languages:
  – \{ a^n b^n \mid n > 0 \}
  – \{ w \mid \#_a(w) = \#_b(w) \}
  – \{ w w^R \mid w \in \{a,b\}^* \}
  – balanced parentheses
  – "C", Java, etc.

• Some Non-Context-Free Languages:
  – \{ a^n b^n c^n \mid n > 0 \}
  – \{ w \mid \#_a(w) = \#_b(w) = \#_c(w) \}
  – \{ w w \mid w \in \{a,b\}^* \}
  – "C", Java, etc.

Curiously, their complements are CFL’s
Applications

• Programming languages and compilers
• Parsing other complex input languages
  – html, sql, …
• Natural language processing/
  Computational linguistics
  – Requires handling ambiguous grammars
• Computational biology (RNA)
The big picture

Ability to specify and reason about abstract formal models of computational systems is an important life skill. Practice it.