
CSE 322

Exam Reviews

Basic Concepts

- Formal Languages

- Alphabet (Σ)
- String (Σ^*)
- Length ($|x|$)
- Empty String (ε)
- Empty Language (\emptyset)

- Language/String Operations

- “Regular” Operations:
 - Union (\cup)
 - Concatenation (\bullet)
 - (Kleene) Star ($*$)
- Other:
 - Intersection
 - Complement
 - Reversal
 - Shuffle
 - ...

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
 - Multiple out-edges
 - ϵ -moves
 - N accepts $w \in \Sigma^*$
 - N recognizes $L \subseteq \Sigma^*$
- Regular Expressions
 - $\emptyset, \epsilon, a \in \Sigma, \cup, \cdot, *, ()$
- GNFA

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: $R_{ik} R_{kk}^* R_{kj}$)

Reg Expr \rightarrow NFA
(join NFAs w/ ϵ -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, ...)

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.

Non-Regular Languages

- Key idea: once M is in some state q, it doesn't remember how it got there.

E.g. “hybrids”:

if $xy \in L(M)$ and x, x' both go to q, then $x'y \in L(M)$ too.

E.g. “loops”:

if $xyz \in L(M)$ and x, xy both go to q, then $xy^iz \in L(M)$ for all $i \geq 0$.

- Cor: Pumping Lemma
- Important examples:

$$L_1 = \{ a^n b^n \mid n > 0 \}$$

$$L_2 = \{ w \mid \#_a(w) = \#_b(w) \}$$

$$L_3 = \{ ww \mid w \in \Sigma^* \}$$

$$L_4 = \{ ww^R \mid w \in \Sigma^* \}$$

$$L_5 = \{ \text{balanced parens} \}$$

- Also: closure under \cap , complementation sometimes useful:

$$- L_1 = L_2 \cap a^*b^*$$

- PS: don't say “Irregular”

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 (Σ), Stack alphabet (Γ)
- Configurations, Moves, \vdash , \vdash^* , push/pop

Main Results

- Every regular language is a CFL
- Closure: union, dot, *, (Reversal; \cap w/ Reg)
- Non-Closure: Intersection, complementation
- Equivalence of CFG & PDA
 - $\text{CFG} \subseteq \text{PDA}$:
top-down(match/expand), bottom-up (shift/reduce)
 - $\text{PDA} \subseteq \text{CFG}$: A_{pq}
- Pumping Lemma & non-CFL's
- Deterministic PDA \neq Nondeterministic PDA

Important Examples

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