Formal Language Theory

Michael J. Watts

http://mike.watts.net.nz

Lecture Outline

- Languages
- Grammar
- Regular Sets
- Regular Expressions
- Parsing

Languages

- Alphabet
  - Words
  - Letters
  - Indivisible
- Strings
  - Sequences of alphabet elements
- Sentence
  - Well-formed string in a language
  - Grammatically correct

Languages

- Languages
  - Possible strings
- Natural vs machine
- Defined by a grammar
- There is a hierarchy of languages

Languages

- Chomsky hierarchy
  - Unrestricted grammars
  - Context-sensitive grammars
  - Context-free grammars
  - Regular grammars
- Most programming languages are context-free grammars

Grammar

- Formal definition of a language
- Fit within Chomsky's hierarchy
- Describes syntax
- Keywords
  - For computer languages
- Specified as production rules
  - A -> B

Regular Set

- Generated from the elements of the alphabet
- Uses three operations
  - Union
  - Concatenation
  - Kleene star
- A regular set can be described with a regular expression

Regular Expressions

- Specify sets of strings
- Regular expressions are used extensively in computing
  - Searching databases
  - File systems
  - String parsing
  - Structuring tokens in programming languages

Parsing

- Breaking a string into parts
  - Tokens
- Applies to
  - Searching
  - Compilation
- Uses finite state machines
  - Matches tokens

Parsing

- Groups tokens into grammatical groups
  - Hierarchical
- Checks that expressions match grammar
- Application of regular expressions
- Parsers can be generated from grammars
  - YACC/Bison

Summary

- Languages have specific components
  - Common across natural and machine languages
- Grammars define languages
  - Structure / content
- Regular expressions define sets of strings
  - Regular sets
- Parsing is breaking strings up into token