CPS 140 - Mathematical Foundations of CS  
Dr. Susan Rodger  
Section: Introduction (Ch. 1) (handout)

languages

 regular languages  
context-free languages  
recursively enumerable languages

grammars

 regular grammar  
CFG  
unrestricted grammar

automata

 finite automata  
pushdown automata  
Turing machine

Power of Machines

<table>
<thead>
<tr>
<th>automata</th>
<th>Can do?</th>
<th>Can’t do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>FA</td>
<td>integers</td>
<td>arith expr</td>
</tr>
<tr>
<td>PDA</td>
<td>arith expr</td>
<td>compute expr</td>
</tr>
<tr>
<td>TM</td>
<td>compute expr</td>
<td>decide if halts</td>
</tr>
</tbody>
</table>

Applications

Compiler

- Question: C++ program - is it valid?
- Question: language L, program P - is P valid?

Stages of a Compiler
Set Theory - Read Chapter 1 Linz.

A Set is a collection of elements.

A={1,4,6,8}, B={2,4,8}, C={3,6,9,12,...}, D={4,8,12,16,...}

- (union) A∪B=
- (intersection) A∩B=
- C∩D=
- (member of) 42 ∈ C?
- (subset) B⊂C?
- B∩A ⊆ D?
- (product) A×B=
- |B|=
- |A×B|=
- ∅ ∈ B∩C?
- (powerset) 2^B=

Example

Prove: Set S has 2^|S| subsets.

<table>
<thead>
<tr>
<th></th>
<th>number of subsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Technique: Proof by Induction
1. Basis: P(1)? Prove smallest instance is true.

2. Induction Hypothesis - I.H.
   Assume P(n) is true for 1,2,....,n

3. Induction Step - I.S.
   Show P(n+1) is true (using I.H.)

Proof of Example:

1. Basis:
2. I.H. Assume
3. I.S. Show

Ch. 1: 3 Major Concepts

- languages
- grammars
- automata

Languages

- $\Sigma$ - set of symbols, alphabet
- string - finite sequence of symbols
- language - set of strings defined over $\Sigma$

Examples

- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  $L = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, \ldots\}$
- $\Sigma = \{a, b, c\}$
  $L = \{ab, ac, cabb\}$
- $\Sigma = \{a, b\}$
  $L = \{a^n b^n \mid n > 0\}$

Notation
• symbols in alphabet: a, b, c, d, ...
• string names: u,v,w,...

Definition of concatenation
Let \( w = a_1 a_2 \ldots a_n \) and \( v = b_1 b_2 \ldots b_m \)
Then \( w \circ v \) OR \( wv = \)
See book for formal definitions of other operations.

String Operations
strings: \( w = abbc, v = ab, u = c \)

• size of string
  \( |w| + |v| = \)
• concatenation
  \( v^3 = vvv = vovov = \)
• \( v^0 = \)
• \( w^R = \)
• \( |vv^Rw| = \)
• \( ab \circ \lambda = \)

Definition
\( \Sigma^* = \) set of strings obtained by concatenating 0 or more symbols from \( \Sigma \)

Example
\( \Sigma = \{a, b\} \)
\( \Sigma^* = \)
\( \Sigma^+ = \)

Examples
\( \Sigma = \{a, b, c\}, \ L_1 = \{ab, bc, aba\}, \ L_2 = \{c, bc, bcc\} \)

• \( L_1 \cup L_2 = \)
• \( L_1 \cap L_2 = \)
• \( L_1 = \)
• \( \overline{L_1} = \)
• \( L_1 \cap L_2 = \)
• \( L_1 \circ L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\} = \)
Definition

$L^0 = \{\lambda\}$

$L^2 = L \circ L$

$L^3 = L \circ L \circ L$

$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \ldots$

$L^+ = L^1 \cup L^2 \cup L^3 \ldots$

Grammars

grammar for english

<sentence> → <subject><verb><d.o.>

<subject> → <noun>

<verb> → hit | ran | ate

<d.o.> → <article><noun>

<noun> → Fritz | ball

<noun> → ball

<article> → the | an | a

Examples

Fritz hit the ball.

The ball hit Fritz.

The ball ate the ball

Syntactically correct?

Semantically correct?

Grammar

G=(V,T,S,P) where

- V - variables (or nonterminals)
• **T** - terminals
• **S** - start variable (S∈V)
• **P** - productions (rules)
  
  \[ x \rightarrow y \text{ “means” replace } x \text{ by } y \]
  
  \[ x \in (V \cup T)^+, \ y \in (V \cup T)^* \]
  
  where V, T, and P are finite sets.

**Definition**

\[ w \Rightarrow z \quad \text{w derives z} \]

\[ w \Rightarrow^* z \quad \text{derives in 0 or more steps} \]

\[ w \Rightarrow^+ z \quad \text{derives in 1 or more steps} \]

**Definition**

\[ G = (V, T, P, S) \]

\[ L(G) = \{ w \in T^* \mid S \Rightarrow^* w \} \]

**Example**

\[ G = (\{S\}, \{a, b\}, S, P) \]

\[ P = \{ S \rightarrow aaS, S \rightarrow b \} \]

\[ L(G) = \]

**Example**

\[ L(G) = \{ a^nccb^n \mid n > 0 \} \]

\[ G = \]

**Automata**

Abstract model of a digital computer

---

![Automata Diagram](attachment:automata_diagram.png)