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

C++ program

lexical analysis

tokens

syntax analysis

parse tree

code generation

assembly language program

Set Theory - Read Chapter 1

A Set is a collection of elements.

\[ A = \{1,4,6,8\}, \quad B = \{2,4,8\}, \quad C = \{3,6,9,12,\ldots\}, \quad D = \{4,8,12,16,\ldots\} \]

- (union) \( A \cup B = \)
- (intersection) \( A \cap B = \)
- \( C \cap D = \)
- (member of) \( 42 \in C? \)
- (subset) \( B \subset C? \)
- \( B \cap A \subseteq D? \)
- (product) \( A \times B = \)
- \( |B| = \)
- \( \emptyset \in B \cap C? \)
- (powerset) \( 2^B = \)

Example

Prove: Set \( S \) has \( 2^{|S|} \) subsets.

| \(|S|\) | number of subsets |
|-------|------------------|
| 0     |                  |
| 1     |                  |
| 2     |                  |
| 3     |                  |
| 4     |                  |

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

Definition: An infinite set is *countable* if its elements have 1-1 correspondence with the positive integers.

Examples:

- S = \{ positive odd integers \}
- S = \{ real numbers \}
- S = \{(i,j) | i,j>0, are integers\}

Theorem Let S be an infinite countable set. Its powerset $2^S$ is not countable.

Proof - Diagonalization

- S is countable, so it’s elements can be enumerated.
  $S = \{s_1, s_2, s_3, s_4, s_5, s_6 \ldots \}$
  An element $t \in 2^S$ can be represented by a sequence of 0’s and 1’s such that the $i$th position in $t$ is 1 if $s_i$ is in $t$, 0 if $s_i$ is not in $t$. 

3
Example, \( \{s_2, s_3, s_5\} \) represented by

Example, set containing every other element from \( S \), starting with \( s_1 \) is \( \{s_1, s_3, s_5, s_7, \ldots \} \) represented by

Suppose \( 2^S \) countable. Then we can enumerate all its elements: \( t_1, t_2, \ldots \)

<table>
<thead>
<tr>
<th></th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>( s_4 )</th>
<th>( s_5 )</th>
<th>( s_6 )</th>
<th>( s_7 )</th>
<th>( \ldots )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( t_5 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( t_6 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( t_7 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \ldots )</td>
</tr>
</tbody>
</table>

| \( \ldots \) |   |   |   |   |   |   |   |   |

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: \( a, b, c, d, \ldots \)
- string names: \( u, v, w, \ldots \)
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
  \[ v^3 = vvv = v \circ v \circ v = \]
- \( v^0 = \)
- \( w^R = \)
- \( |v^Rw| = \)
- \( ab \circ \epsilon = \)

Definition

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

Example

\( \Sigma = \{a, b\} \)

\( \Sigma^* = \)

\( \Sigma^+ = \)

Examples

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

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

Definition

\( L^0 = \{\epsilon\} \)

\( L^2 = L \circ L \)
\[ L^3 = L \circ L \circ L \]
\[ L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \ldots \]
\[ L^+ = L^1 \cup L^2 \cup L^3 \ldots \]

**Example** Is \( L \) a countable set?

\( S = \{ w \in \Sigma^+ \}, \Sigma = \{a, b\} \)

**Regular Expressions**

Method to represent strings in a language

+ union (or)
\( \circ \) concatenation (AND) (can omit)
\( \ast \) star-closure (repeat 0 or more times)

**Example:**

\( (a + b)^* \circ a \circ (a + b)^* \)

**Example:**

\( (aa)^* \)

**Definition** Given \( \Sigma \),

1. \( \emptyset, e, a \in \Sigma \) are R.E.
2. If \( r \) and \( s \) are R.E. then
   - \( r+s \) is R.E.
   - \( rs \) is R.E.
   - \( r^* \) is R.E.
3. \( r \) is a R.E. iff it can be derived from (1) with a finite number of applications of (2).

**Definition:** \( L(r) = \) language denoted by R.E. \( r \).

1. \( \emptyset, \{ \varepsilon \}, \{a\} \) are L denoted by a R.E.
2. if \( r \) and \( s \) are R.E. then
   - (a) \( L(r+s) = L(r) \cup L(s) \)
   - (b) \( L(rs) = L(r) \circ L(s) \)
   - (c) \( L((r)^*) = (L(r)^*) \)

**Precedence Rules**

- * highest
  - \( \circ \)
  - +
Example:

\[ ab^* + c = \]

Examples:

1. \( \Sigma = \{a, b\}, \{w \in \Sigma^* \mid w \text{ has an odd number of } a\text{'s followed by an even number of } b\text{'s}\}. \)

2. \( \Sigma = \{a, b\}, \{w \in \Sigma^* \mid w \text{ has no more than 3 } a\text{'s and must end in } ab\}. \)

3. Regular expression for positive and negative integers

Grammars

```
grammar for english

<sentence> -> <subject><verb><d.o.>

<subject> -> <noun> | <article><noun>

<verb> -> hit | ran | ate

<d.o.> -> <article><noun>

<noun> -> Fritz | ball

<article> -> the | an | a
```

Examples

Fritz hit the ball.

```
<sentence> -> <subject><verb><d.o.>
  -> <noun><verb><d.o.>
  -> Fritz <verb><d.o.>
  -> Fritz hit <d.o.>
  -> Fritz hit <article><noun>
  -> Fritz hit the <noun>
  -> 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→y “means” replace x by y
  x∈(V∪T)+, y∈(V∪T)*
  where V, T, and P are finite sets.

Definition

w ⇒ z w derives z
w ⇒* z derives in 0 or more steps
w ⇓ z derives in 1 or more steps

Definition

G=(V,T,P,S)
L(G)={w∈T* | S ⇒ w}

Example

G={S, {a,b}, S, P}
P={S→ aaS, S→ b}
L(G)=

Example

L(G) = {a^nab^n | n > 0}
G =

Automata

Abstract model of a digital computer