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}, 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| =
- ∅ ∈ 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></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></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 \( 1, 2, \ldots, n \)

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:**

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

**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 \( t \).
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>(t_1)</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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 \(w = a_1a_2\ldots a_n\) and \(v = b_1b_2\ldots b_m\)

Then \(wv \text{ OR } vw = \)

See book for formal definitions of other operations.

**String Operations**

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

- size of string
  \(|w| + |v| = \)
- concatenation
  \(v^3 = vvv = vvovv = \)
- \(v^0 = \)
- \(w^R = \)
- \(|vv^Rw| = \)
- \(ab \circ \epsilon = \)

**Definition**

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

**Example**

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

\(\Sigma^* = \)

\(\Sigma^+ = \)

**Examples**

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

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

**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

\[ + \quad \text{union (or)} \]
\[ \circ \quad \text{concatenation (AND) (can omit)} \]
\[ * \quad \text{star-closure (repeat 0 or more times)} \]

**Example:**

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

**Example:**

\[ (aa)^* \]

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

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

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

3. Regular expression for positive and negative integers

Grammars

Grammar for English

\[
\begin{align*}
\langle \text{sentence} \rangle & \rightarrow \langle \text{subject} \rangle \langle \text{verb} \rangle \langle \text{d.o.} \rangle \\
\langle \text{subject} \rangle & \rightarrow \langle \text{noun} \rangle \mid \langle \text{article} \rangle \langle \text{noun} \rangle \\
\langle \text{verb} \rangle & \rightarrow \text{hit} \mid \text{ran} \mid \text{ate} \\
\langle \text{d.o.} \rangle & \rightarrow \langle \text{article} \rangle \langle \text{noun} \rangle \\
\langle \text{noun} \rangle & \rightarrow \text{Fritz} \mid \text{ball} \\
\langle \text{article} \rangle & \rightarrow \text{the} \mid \text{an} \mid \text{a}
\end{align*}
\]

Examples

Fritz hit the ball.

\[
\begin{align*}
\langle \text{sentence} \rangle & \rightarrow \langle \text{subject} \rangle \langle \text{verb} \rangle \langle \text{d.o.} \rangle \\
& \rightarrow \langle \text{noun} \rangle \langle \text{verb} \rangle \langle \text{d.o.} \rangle \\
& \rightarrow \text{Fritz} \langle \text{verb} \rangle \langle \text{d.o.} \rangle \\
& \rightarrow \text{Fritz hit} \langle \text{d.o.} \rangle \\
& \rightarrow \text{Fritz hit} \langle \text{article} \rangle \langle \text{noun} \rangle \\
& \rightarrow \text{Fritz hit the} \langle \text{noun} \rangle \\
& \rightarrow \text{Fritz hit the ball}
\end{align*}
\]

The ball hit Fritz.

The ball ate the ball

Syntactically correct?

Semantically correct?
Grammar

\[ G = (V, \Sigma, R, S) \] where

- \( V \) - variables (or nonterminals)
- \( \Sigma \) - terminals (or alphabet)
- \( R \) - rules (or productions)
- \( S \) - start variable (\( S \in V \))

\( x \rightarrow y \) "means" replace \( x \) by \( y \)

\( x \in (V \cup \Sigma)^+, y \in (V \cup \Sigma)^* \)

where \( V, \Sigma, \) and \( R \) are finite sets.

Definition

- \( w \Rightarrow z \) \( w \) derives \( z \)
- \( w \Rightarrow^* z \) derives in \( 0 \) or more steps
- \( w \Rightarrow^+ z \) derives in \( 1 \) or more steps

Definition

\[ G = (V, \Sigma, R, S) \]

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

Example

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

\[ R = \{ S \rightarrow aS, S \rightarrow b \} \]

\[ L(G) = \]

Example

\[ L(G) = \{ a^n c b^n | n > 0 \} \]

\[ G = \]

Automata

Abstract model of a digital computer

![Diagram of an automata](image-url)