What will we do in CPS 140?

Questions

- Can you write a program to determine if a string is an integer?
  
  9998.89  8abab  789342

- Can you do this if your machine had no additional memory other than the program? (can’t store any values and look at them again)

- Can you write a program to determine if the following are arithmetic expressions?

  $$((34 + 7 \times (18/6)))$$

  $$((((((a + b) + c) \times d)(e + f))))$$

- Can you do this if your machine had no additional memory other than the program?

- Can you write a program to determine the value of the following expression?

  $$((34 + 7 \times (18/6)))$$

- Can you write a program to determine if a file is a valid C++ program?

- Can you write a program to determine if a C++ program given as input will ever halt?

Language Hierarchy
Power of Machines

<table>
<thead>
<tr>
<th>automata</th>
<th>Can do?</th>
<th>Can’t do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>finite automata (FA)</td>
<td>integers</td>
<td>arith expr</td>
</tr>
<tr>
<td>(no memory)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pushdown automata (PDA)</td>
<td>arith expr</td>
<td>compute expr</td>
</tr>
<tr>
<td>(only memory is stack)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turing machines (TM)</td>
<td>compute expr</td>
<td>decide if halts</td>
</tr>
<tr>
<td>(infinite memory)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application

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 Linz.

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) \( A \times B = \)
- \( |B| = \)
- \( |A \times B| = \)
- \( \emptyset \in B \cap C? \)
- (powerset) \( 2^B = \)

**Example** What are all the subsets of \( \{3,5\} \)?

How many subsets does a set \( S \) have?

| \( |S| \) | number of subsets |
|-------|-----------------|
| 0     | 1               |
| 1     | 2               |
| 2     | 4               |
| 3     | 8               |
| 4     | 16              |

How do you prove? Set \( S \) has \( 2^{|S|} \) subsets.
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

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

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

Definition of concatenation

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

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
  \( v^3 = vvv = vvvv = \)
- \( v^0 = \)
- \( w^R = \)
- \( |vv^Rw| = \)
- \( ab \circ \lambda = \)

Definition

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

Example

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

\( \Sigma^* = \)

\( \Sigma^+ = \)

Examples

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

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

\[ <\text{sentence}> \rightarrow <\text{subject}><\text{verb}><\text{d.o.}> \]

\[ <\text{subject}> \rightarrow <\text{noun}> \mid <\text{article}><\text{noun}> \]

\[ <\text{verb}> \rightarrow \text{hit} \mid \text{ran} \mid \text{ate} \]

\[ <\text{d.o.}> \rightarrow <\text{article}><\text{noun}> \mid <\text{noun}> \]

\[ <\text{noun}> \rightarrow \text{Fritz} \mid \text{ball} \]

\[ <\text{article}> \rightarrow \text{the} \mid \text{an} \mid \text{a} \]

**Examples** (derive a sentence)

Fritz hit the ball.

\[ <\text{sentence}> \rightarrow <\text{subject}><\text{verb}><\text{d.o.}> \]

\[ \rightarrow <\text{noun}><\text{verb}><\text{d.o.}> \]

\[ \rightarrow \text{Fritz}<\text{verb}><\text{d.o.}> \]

\[ \rightarrow \text{Fritz hit}<\text{d.o.}> \]

\[ \rightarrow \text{Fritz hit}<\text{article}><\text{noun}> \]

\[ \rightarrow \text{Fritz hit the}<\text{noun}> \]

\[ \rightarrow \text{Fritz hit the}<\text{ball}> \]

Can we also derive the sentences?

**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 \((S \in V)\)
- **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

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

Definition

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

Example

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

Example

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

\(G = \)

**Automata** Abstract model of a digital computer

![Diagram of Automata](image-url)