Day 2 - 1/19/2024

Review

Formal Languages

A formal language is a set of strings. An alphabet is the set of characters that may appear in the strings.

A language over $\{0, 1\}$:

$$\{\varepsilon, 01, 0101, 010101, ...\}$$

Describing

Two ways to describe a language:

1. Generator: Can produce all strings in a language
2. Recognizer: Can recognize all strings in a language

Grammars

A grammar is a list of one or more productions (a string substitution rule).

Example:

$$\begin{aligned} &1. \quad S \to xSy \\ &2. \quad S \to a \\ &3. \quad S \to \varepsilon \end{aligned}$$

Usage

1. Start with start symbol (S)
2. Apply production ($\to$)
3. Go until no more *nonterminals* (uppercase letters)

Example: derive $xxxyyy$

$$S \to \overbrace{xSy}^{1} \to \overbrace{xxSyy}^{1} \to \overbrace{xxxSyyy}^{1} \to \overbrace{xxxyyy}^{3}$$

Example: What language does this generate?

$$\{\varepsilon, a, \text{xy}, \text{xxyy}, \ldots, \text{xay}, \text{xxayy}, \ldots\}$$

Example: Based on Grammar D, write a derivation for xy.

Grammar D

1. $S \to AB$
2. $A \to x$
3. $B \to y$

$$S \to \overbrace{AB}^{1} \to \overbrace{xB}^{2} \to \overbrace{xy}^{3}$$

Exercises

7. Write the grammar that generates the language:
   $\{ab, abb, abbb, \ldots\}$

   1. $S \to abX$
   2. $X \to Xb$
   3. $X \to \varepsilon$

8. How can you rewrite the grammar from 7. to also generate the string "a"?

   Remove the b from the first production.

The Chomsky Hierarchy

Includes 4 nested categories of languages, types 3, 2, 1, and 0.

Regular Language

A regular language is one that can generated with a grammar that has one of:

1. $A \to \varepsilon$
2. $A \to b$
3. $A \to bC$

Context-Free Language

A context-free language is one that can be generated from a grammar where each left-hand production consists of a single nonterminal.

$$n. \quad A \to [\text{anything}]$$

Context-Sensitive Language

We don't care about context-sensitive languages.

Computably Enumerable Language

A computably enumerable language is one that can be described by a grammar.

The recognizer is a **Turing machine**.

Regular Languages

Regular languages have two important applications:

1. In most PLs, the set of all lexems of a particular kind forms a regular language.
2. He skipped the rest of the slide.

Grammar

As stated above:

1. $A \to \varepsilon$
2. $A \to b$
3. $A \to bC$

Here's an example:

1. $S \to \varepsilon$
2. $S \to t$
3. SxB
4. ByS

What language does this grammar generate?

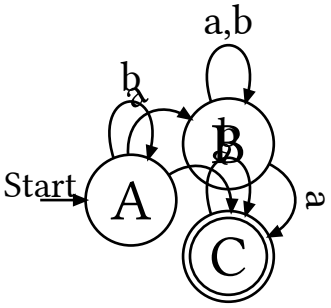
$$\{\varepsilon, t, xyt, ...\}$$

More examples occurred.

Finite Automata

A deterministic finite automata (DFA) is a kind of recognizer for regular languages.

A DFA consists of a bunch of states and a translation between those states.



Incredible, I know.

Languages that are recognized by DFAs are regular languages.

Exercise: I won't draw the diagram but what language is recognized by it?

$$\{y, yy, yyy, ..., xx, xxxx, xxxxxx, ..., xyx, xyxy, xxyxx, xxyxxy, ...\}$$

0 is an even number