Modeling Computation

Introduction to Formal Languages and Automata

Context Free Languages

Finite automata recognize regular languages.

An example of a language that cannot be recognized by finite automata

$$L = \{a^n b^n \mid n \geq 0\}$$

L is a <u>context free language</u>,

which is <u>generated</u> or <u>derived</u> by a

<u>grammar</u>.

Simple English Grammar

1. A sentence is a noun phrase followed by a verb phrase.
   - <sentence> → <noun phrase><verb phrase>
2. A noun phrase is a proper noun, or a determiner followed by adjective followed by a common noun.
   - <noun phrase> → <proper noun>
   - <noun phrase> → <determiner><adjective><common noun>
3. A verb phrase is a verb, or a verb followed by an adverb.
   - <verb phrase> → <verb> | <verb><adverb>

Generating a Sentence

<sentence> ⇒ <noun phrase><verb phrase>

⇒ <proper noun><verb phrase>

⇒ Denmark <verb phrase>

⇒ Denmark <verb><adverb>

⇒ Denmark exists <adverb>

⇒ Denmark exists poorly

Regular expressions are also a kind of language generator

Ex: ba^* generates $L = \{ba^n \mid n \geq 0\}$

As a set of rules:

$$S \to bA$$
$$A \to aA$$
$$A \to \varepsilon$$

Derivation of $baaa$:

$$\underbrace{S \Rightarrow bA \Rightarrow baA \Rightarrow baaA \Rightarrow baaaA \Rightarrow baaa}_{\text{derivation}}$$

Regular Grammars

A regular grammar is a grammar that describes a regular language, and has rules of the form:

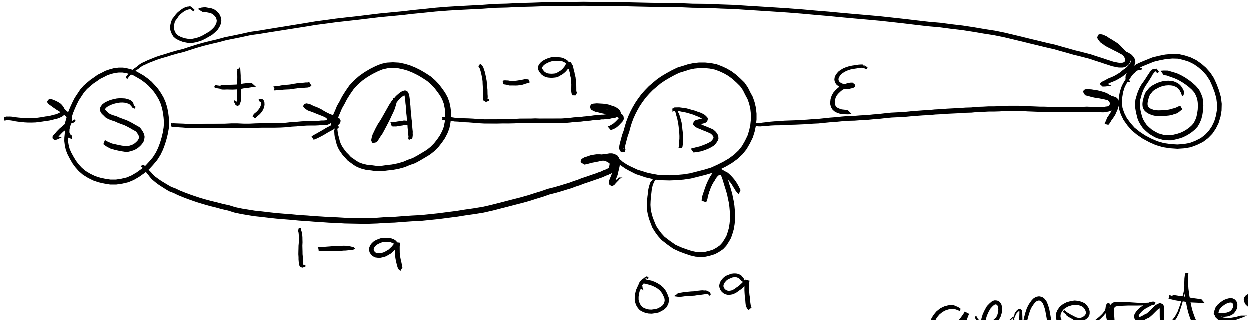
$A \to a$ or $A \to aB$ or $A \to \varepsilon$

One-to-one correspondence between regular grammars and NFAs

$$S \to 0C \mid +A \mid -A \mid 1B \mid 2B \mid \cdots \mid 9B$$

$$A \to 1B \mid 2B \mid \cdots \mid 9B$$

$$\mathrm{B} \to 0B \mid 1B \mid 2B \mid \cdots \mid 9B \mid \epsilon$$



generates integers

Context Free Grammar (CFG)

$G = (\Sigma, NT, R, S)$

Σ is a finite set of terminals

NT is a finite set of non-terminals

$R \subseteq NT \times (\Sigma \cup NT)^*$ is the set of rules

$S \in NT$ is the start symbol

The language $L(G)$ generated or derivable by G is the set of all strings derivable from the start symbol.

regular grammar

If G is a RG, then $L(G)$ is a regular language.

context-free grammar

If G is a CFG, then $L(G)$ is a context-free language.

CFGs have rules of the form

$$A \to \gamma, \qquad \gamma \in (\Sigma \cup NT)^*$$

Ex: $S \to aSb \mid \epsilon$

What language does this grammar generate?

$\Sigma = \{a, b\}$

$NT = \{S\}$

$R = \{S \to aSb, S \to \varepsilon\} = \{(S, aSb), (S, \varepsilon)\}$

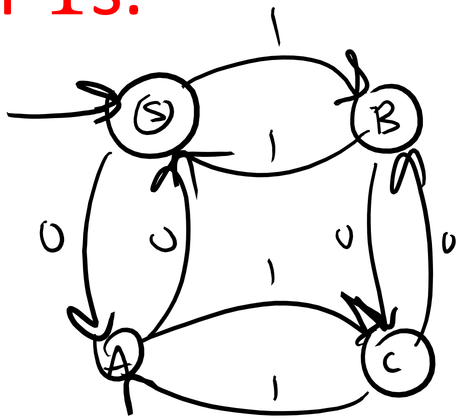
$S = S$

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$
$\Rightarrow aaabbb$

- - -

$L = \{a^n b^n \mid n \geq 0\}$

Construct a grammar that generates strings that have an even number of 0s and an even number of 1s.



$S \rightarrow 0A \mid 1B \mid \varepsilon$

$A \rightarrow 0S \mid 1C$

$B \rightarrow 1S \mid 0C$

$C \rightarrow 0B \mid 1A$

Construct a grammar that generates strings that have an equal number of 0s and 1s.