Monday: Context-free grammars

These definitions are on pages 101-102.

Term	Typical symbol or Notation	Meaning
Context-free grammar (CFG)	G	$G = (V, \Sigma, R, S)$
The set of **variables**	V	Finite set of symbols that represent phases in production pattern
The set of **terminals**	Σ	Alphabet of symbols of strings generated by CFG $V \cap \Sigma = \emptyset$
The set of **rules**	R	Each rule is $A \to u$ with $A \in V$ and $u \in (V \cup \Sigma)^*$
The **start** variable	S	Usually on left-hand-side of first/ topmost rule
Derivation	$S \Rightarrow \cdots \Rightarrow w$	Sequence of substitutions in a CFG (also written $S \Rightarrow^* w$). At each step, we can apply one rule to one occurrence of a variable in the current string by substituting that occurrence of the variable with the right-hand-side of the rule. The derivation must end when the current string has only terminals (no variables) because then there are no instances of variables to apply a rule to.
Language **generated** by the context-free grammar G	$L(G)$	The set of strings for which there is a derivation in G. Symbolically: $\{w \in \Sigma^* \mid S \Rightarrow^* w\}$ i.e. $\{w \in \Sigma^* \mid$ there is derivation in G that ends in $w\}$
Context-free language		A language that is the language generated by some context-free grammar

Var → RHS

Examples of context-free grammars, derivations in those grammars, and the languages generated by those grammars

$G_1 = (\{S\}, \{0\}, R, S)$ with rules

set of variables; start variables; set of terminals

LHS RHS

$R = \{$ (1) $S \to 0S$, (2) $S \to 0$ $\}$.

In $L(G_1)$ ...

$S \Rightarrow 0S \Rightarrow 00S \Rightarrow 000$ so $000 \in L(G_1)$

$S \Rightarrow 0$ so $0 \in L(G_1)$

Not in $L(G_1)$ ... ε b/c all rules introduce (at least one) 0. $\varepsilon \notin L(G_1)$.

$R = \{S \to 0S,\ S \to 1S,\ S \to \varepsilon\}$

$G_2 = (\{S\}, \{0,1\}, R, S)$

$S \to 0S \mid 1S \mid \varepsilon$ (1) (2) (3)

notation abbreviating multiple rules with same variable on LHS.

In $L(G_2)$ ...

$S \overset{1}{\Rightarrow} 0S \overset{2}{\Rightarrow} 01S \overset{3}{\Rightarrow} 01\varepsilon = 01$

start variable

Not in $L(G_2)$ ... There is no such string!

G_2 lets us build up each string over $\{0,1\}$ by introducing characters from left to right. $L(G_2) = \{0,1\}^*$.

BUILD AT ANY PT IN THE STRING

$(\{S,T\}, \{0,1\}, R, S)$ with rules

2 variables!

$S \to T1T1T1T$ (1)

$T \to 0T \mid 1T \mid \varepsilon$ (2) (3) (4)

plug in some string over $\{0,1\}$

In $L(G_3)$ ...

$S \Rightarrow T1T1T1T \overset{4}{\Rightarrow} 1T1T1T \Rightarrow 1T1T1 \Rightarrow 11T1 \Rightarrow 111$

$S \Rightarrow T1T1T1T \Rightarrow T11T1T \Rightarrow T111T \Rightarrow T111 \Rightarrow 111$

$S \Rightarrow T1T1T1T \overset{2}{\Rightarrow} 0T1T1T1T \overset{4}{\Rightarrow} 01T1T1T \Rightarrow \cdots \Rightarrow 0111$

Not in $L(G_3)$ ...

Regular

$L(G_3) = \{w \in \{0,1\}^* \mid w \text{ has at least three 1s}\}$.

Note: rules can have any string over $V \cup \Sigma$ on RHS

both have variables on RHS

$\tilde{G} = (\{S,T\}, \{0,1\}, \{S \to 0S,\ T \to 1T\}, S)$

$S \Rightarrow 0S \cdots ?$ $S \Rightarrow 0S \Rightarrow 00S \cdots ?$ $S \Rightarrow 0S \Rightarrow 00S \cdots ?$

$L(\tilde{G}) = \emptyset$.

What about a grammar that generates a nonregular language?

$G_4 = (\{A, B\}, \{0, 1\}, R, A)$ with rules

$$A \to 0A0 \mid 0A1 \mid 1A0 \mid 1A1 \mid 1$$

(1) (2) (3) (4) (5)

In $L(G_4)$ ... 1, 010, 011

$A \Rightarrow 1$

$A \overset{1}{\Rightarrow} 0A0 \overset{4}{\Rightarrow} 01A10 \overset{2}{\Rightarrow} 010A110 \overset{5}{\Rightarrow} 0101110$

Not in $L(G_4)$ ... ε

$$L(G_4) = \{w \in \{0,1\}^* \mid w \text{ is odd length and middle character is } 1\}$$

Claim: $L(G_4)$ is nonregular.

(extra practice: prove this with pumping lemma)

Design a CFG to generate the language $\{a^n b^n \mid n \geq 0\}$

$G_5 = (\{S\}, \{a,b\}, \{S \to \varepsilon \mid aSb\}, S)$

V Σ R S

Sample derivation:

$S \Rightarrow \varepsilon$ (1)

$S \Rightarrow aSb \Rightarrow ab$ (2, 1)

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$ (2, 2, 1)

"grow out string from center"

Wednesday: Context-free languages

Warmup: Design a CFG to generate the language $\{a^i b^j \mid j \geq i \geq 0\}$

Sample derivation:

Design a PDA to recognize the language $\{a^i b^j \mid j \geq i \geq 0\}$

Theorem 2.20: A language is generated by some context-free grammar if and only if it is recognized by some push-down automaton.

Definition: a language is called **context-free** if it is the language generated by a context-free grammar. The class of all context-free language over a given alphabet Σ is called **CFL**.

Consequences:

- Quick proof that every regular language is context free
- To prove closure of the class of context-free languages under a given operation, we can choose either of two modes of proof (via CFGs or PDAs) depending on which is easier
- To fully specify a PDA we could give its 6-tuple formal definition or we could give its input alphabet, stack alphabet, and state diagram. An informal description of a PDA is a step-by-step description of how its computations would process input strings; the reader should be able to reconstruct the state diagram or formal definition precisely from such a descripton. The informal description of a PDA can refer to some common modules or subroutines that are computable by PDAs:
  - PDAs can "test for emptiness of stack" without providing details. *How?* We can always push a special end-of-stack symbol, \$, at the start, before processing any input, and then use this symbol as a flag.
  - PDAs can "test for end of input" without providing details. *How?* We can transform a PDA to one where accepting states are only those reachable when there are no more input symbols.

Suppose L_1 and L_2 are context-free languages over Σ. **Goal**: $L_1 \cup L_2$ is also context-free.

Approach 1: with PDAs

Let $M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_2, F_2)$ be PDAs with $L(M_1) = L_1$ and $L(M_2) = L_2$.

Define $M =$

Approach 2: with CFGs

Let $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ be CFGs with $L(G_1) = L_1$ and $L(G_2) = L_2$.

Define $G =$

Suppose L_1 and L_2 are context-free languages over Σ. **Goal**: $L_1 \circ L_2$ is also context-free.

Approach 1: with PDAs

Let $M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_2, F_2)$ be PDAs with $L(M_1) = L_1$ and $L(M_2) = L_2$.

Define $M =$

Approach 2: with CFGs

Let $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ be CFGs with $L(G_1) = L_1$ and $L(G_2) = L_2$.

Define $G =$

Summary

Over a fixed alphabet Σ, a language L is **regular**

iff it is described by some regular expression
iff it is recognized by some DFA
iff it is recognized by some NFA

Over a fixed alphabet Σ, a language L is **context-free**

iff it is generated by some CFG
iff it is recognized by some PDA

Fact: Every regular language is a context-free language.

Fact: There are context-free languages that are not nonregular.

Fact: There are countably many regular languages.

Fact: There are countably inifnitely many context-free languages.

Consequence: Most languages are **not** context-free!

Examples of non-context-free languages

$\{a^n b^n c^n \mid 0 \leq n, n \in \mathbb{Z}\}$
$\{a^i b^j c^k \mid 0 \leq i \leq j \leq k, i \in \mathbb{Z}, j \in \mathbb{Z}, k \in \mathbb{Z}\}$
$\{ww \mid w \in \{0,1\}^*\}$

(Sipser Ex 2.36, Ex 2.37, 2.38)

There is a Pumping Lemma for CFL that can be used to prove a specific language is non-context-free: If A is a context-free language, there there is a number p where, if s is any string in A of length at least p, then s may be divided into five pieces $s = uvxyz$ where (1) for each $i \geq 0$, $uv^i xy^i z \in A$, (2) $|uv| > 0$, (3) $|vxy| \leq p$. *We will not go into the details of the proof or application of Pumping Lemma for CFLs this quarter.*

Friday: Review

Week 5 at a glance

Textbook reading: Chapter 2

For Monday: Introduction to Section 2.1 (page 102)

For Wednesday: Figure 3.1 (Pages 165-167)

For Friday: Test 1 is Friday Feb 9 in discussion section 4pm-4:50pm WLH 2001. The test covers material in Weeks 1 through 4 and Monday of Week 5. To study for the exam, we recommend reviewing class notes (e.g. annotations linked on the class website, podcast, supplementary video from the class website), reviewing homework (and its posted sample solutions), and in particular *working examples* (extra examples in lecture notes, textbook examples listed in hw, review quizzes – PDFs now available on the class website, discussion examples) and getting feedback (office hours and Piazza).

Make sure you can:

- Classify the computational complexity of a set of strings by determining whether it is regular
  - Determine whether a language is recognizable by a (D or N) FA and/or a PDA
- Use context-free grammars and relate them to languages and pushdown automata
  - Identify the components of a formal definition of a context-free grammar (CFG)
  - Use context-free grammars and relate them to languages and pushdown automata.
  - Derive strings in the language of a given CFG
  - Determine the language of a given CFG
  - Design a CFG generating a given language

TODO:

Review quizzes based on class material each day.

Test this Friday in Discussion section.

Homework assignment 3 due next Thursday.

