

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 $A \rightarrow u$ with $A \in V$ and $u \in (V \cup \Sigma)^*$
The start variable	S	Usually on left-hand-side of first/ topmost rule
Derivation	$S \Rightarrow \dots \Rightarrow w$	Sequence of substitutions in a CFG (also written $S \Rightarrow^* w$). At each step, we can <u>apply</u> one rule to one <u>occurrence</u> of a variable in the current string by substituting that occurrence of the variable with the left ^{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. $\{w \in \Sigma^* \mid \text{there is derivation in } G \text{ that ends in } w\}$
Context-free language		A language that is the language generated by some context-free grammar

Var \rightarrow 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
set of terminals

start variables

LHS RHS
 $R = \{ \textcircled{1} S \rightarrow OS, \textcircled{2} S \rightarrow O \}$

In $L(G_1) \dots$

$S \xRightarrow{\textcircled{1}} OS \xRightarrow{\textcircled{1}} OOS \xRightarrow{\textcircled{2}} OOO \Rightarrow OOO \in L(G_1)$
 $S \xRightarrow{\textcircled{2}} O \Rightarrow O \in L(G_1)$

Not in $L(G_1) \dots$

ϵ b/c all rules introduce (at least one) 0. $\epsilon \notin L(G_1)$.

$$R = \{S \rightarrow OS, S \rightarrow 1S, S \rightarrow \epsilon\}$$

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

In $L(G_2) \dots$

$$S \xRightarrow{(1)} OS \xRightarrow{(2)} O1S \xRightarrow{(3)} O1\epsilon$$

01

start variable

$$S \rightarrow OS \mid 1S \mid \epsilon$$

(1) (2) (3)

notation abbreviating multiple roles with same variable on LHS.

Not in $L(G_2) \dots$ 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!

$$S \rightarrow T1T1T1T$$

$$T \rightarrow 0T \mid 1T \mid \epsilon$$

(2) (3) (4)

plugin some string over $\{0, 1\}$

In $L(G_3) \dots$

$$S \Rightarrow T1T1T1T \xRightarrow{(4)} 1T1T1T \Rightarrow 1T1T1 \Rightarrow 11T1 \Rightarrow 111$$

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

$$S \Rightarrow T1T1T1T \xRightarrow{(2)} 0T1T1T1T \xRightarrow{(4)} 01T1T1T \Rightarrow \dots \Rightarrow 0111$$

Not in $L(G_3) \dots$

Regular

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

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

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

$$S \Rightarrow OS \dots? \quad S \Rightarrow OS \Rightarrow OOS \dots? \quad S \Rightarrow OS \Rightarrow OOS \dots?$$

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

What about a grammar that generates a nonregular language?

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

$$A \rightarrow \overset{1}{0} \overset{2}{A} \overset{3}{0} \mid \overset{4}{0} \overset{5}{A} 1 \mid 1 \overset{6}{A} 0 \mid 1 \overset{7}{A} 1 \mid 1$$

In $L(G_4) \dots$

1

$$A \Rightarrow 1$$

010

011

$$A \xRightarrow{1} 0 \overset{4}{A} 0 \Rightarrow \underset{1}{0} \underset{1}{1} \underset{1}{A} \underset{1}{0} \xRightarrow{2} \underset{1}{0} \underset{1}{1} \underset{1}{0} \overset{5}{A} 1 1 \underset{1}{0} \Rightarrow 0101110$$

Not in $L(G_4) \dots$

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

$$G_5 = \left(\underbrace{\{S\}}_V, \underbrace{\{a, b\}}_\Sigma, \underbrace{\{S \rightarrow \varepsilon \mid a S b\}}_R, \underbrace{S}_S \right)$$

Sample derivation:

$$\begin{aligned} S &\xRightarrow{(1)} \varepsilon \\ S &\xRightarrow{(2)} a S b \xRightarrow{(1)} a b \\ S &\xRightarrow{(2)} a S b \xRightarrow{(2)} a a S b b \xRightarrow{(1)} a a b b \end{aligned}$$

"grow out string from center"

Wednesday: Context-free languages

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

$$G_1 = \left(\underbrace{\{S, T\}}_{\text{variables}}, \underbrace{\{a, b\}}_{\text{terminal}}, \underbrace{\{S \xrightarrow{(1)} Sb \mid T \xrightarrow{(2)} aTb \mid \varepsilon\}}_{\text{rules}}, \underbrace{S}_{\text{start var}} \right)$$

$$G_2 = \left(\{S\}, \{a, b\}, \{S \xrightarrow{(1)} Sb \mid aSb \xrightarrow{(2)} \varepsilon\}, S \right)$$

$a^i b^j$
at least
as many b's
as a's.

$\frac{a^i b^i}{T} b^{j-i}$

$$L(G_1) = L(G_2) = L.$$

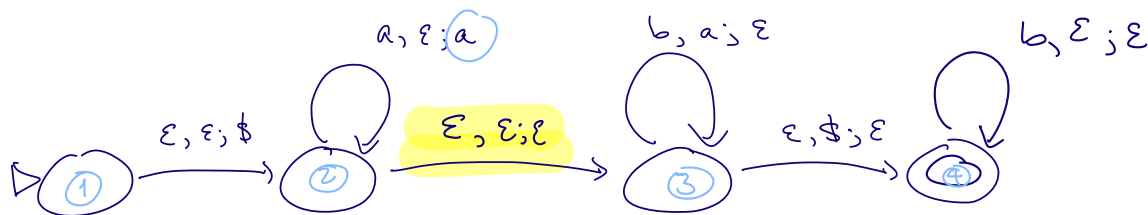
Sample derivation:

in G_1 $S \xrightarrow{(1)} Sb \xrightarrow{(1)} Sbb \xrightarrow{(1)} Sbbb \xrightarrow{(2)} Tbbb \xrightarrow{(4)} bbbb$

in G_1 $S \xrightarrow{(1)} Sb \xrightarrow{(1)} Sbb \xrightarrow{(1)} Sbbb \xrightarrow{(2)} Tbbb \xrightarrow{(3)} aTbbb \xrightarrow{(4)} abbbb$

in G_2 $S \xrightarrow{(1)} Sb \xrightarrow{(2)} aSb \xrightarrow{(2)} aaSb \xrightarrow{(1)} aaSbbb \xrightarrow{(3)} aaabbbb$

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



read a's,
push them on
stack.

read b's,
popping one
a from stack for each b.

special symbol to help detect empty stack.

on bbbb start at ① move to ② move to ③ move to ④
read b's in ④, accept

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:

DFA
NFA
Regexp.

PDA
CFG.

(hw3).

- 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 description. 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 $M = (\underbrace{Q_1 \cup Q_2 \cup \{q_{\text{new}}\}}_Q, \Sigma, \underbrace{\Gamma_1 \cup \Gamma_2}_\Gamma, \delta, q_{\text{new}}, F_1 \cup F_2)$

Assume

$$Q_1 \cap Q_2 = \emptyset$$

$$q_{\text{new}} \notin Q_1 \cup Q_2$$

$$\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon$$

where

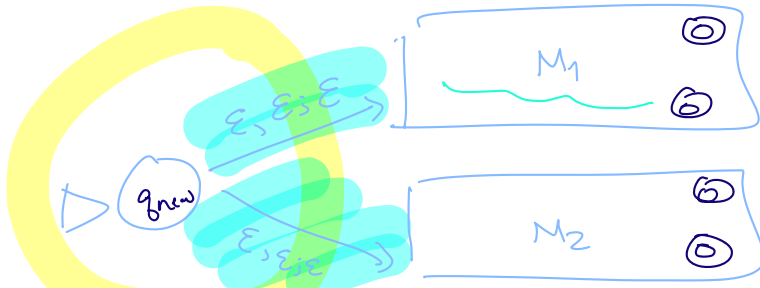
$$\delta((q, a, b)) =$$

$$\delta_1(q, a, b) \text{ if } q \in Q_1, a \in \Sigma_\epsilon, b \in \Gamma_{1, \epsilon}$$

$$\delta_2(q, a, b) \text{ if } q \in Q_2, a \in \Sigma_\epsilon, b \in \Gamma_{2, \epsilon}$$

$$\{(q_1, \epsilon), (q_2, \epsilon)\} \text{ if } q = q_{\text{new}}, a = \epsilon, b = \epsilon$$

otherwise



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 $G = (V_1 \cup V_2 \cup \{S_{\text{new}}\}, \Sigma, R_1 \cup R_2 \cup \{S_{\text{new}} \rightarrow S_1 \mid S_2\}, S_{\text{new}})$

Assume

$$V_1 \cap V_2 = \emptyset$$

$$S_{\text{new}} \notin V_1 \cup V_2$$

$$S_{\text{new}} \Rightarrow S_1 \Rightarrow \text{_____}$$

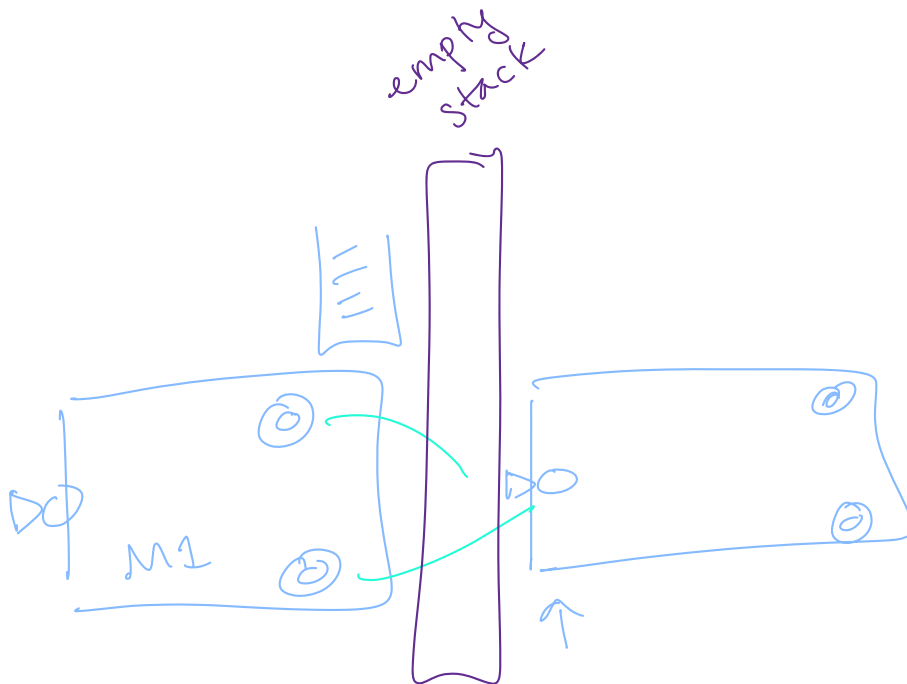
$$S_{\text{new}} \Rightarrow S_2 \Rightarrow \text{_____}$$

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 $M =$



HW3

Approach 2: with CFGs

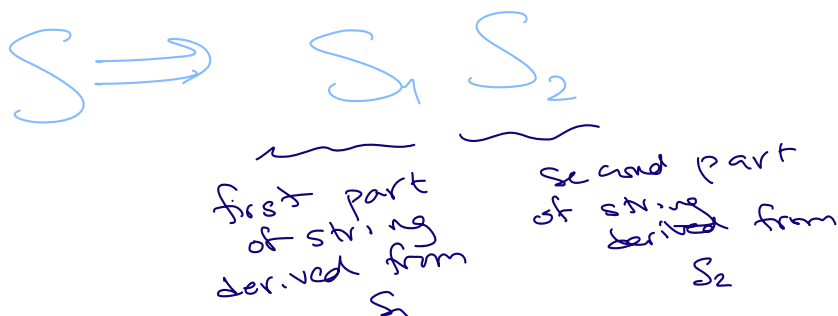
Let $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ be CFGs with $L(G_1) = L_1$ and $L(G_2) = L_2$.

Define $G = (V_1 \cup V_2 \cup \{S_{new}\}, \Sigma, R_1 \cup R_2 \cup \{S_{new} \rightarrow S_1 S_2\}, S_{new})$

ASSUME

$V_1 \cap V_2 = \emptyset$

$S_{new} \notin V_1 \cup V_2$



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 infinitely many context-free languages.

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

Examples of non-context-free languages

(Monday)

$$\{a^n b^n c^n \mid 0 \leq n, n \in \mathbb{Z}\}$$

$$\{a^i b^j c^k \mid 0 \leq i \leq j \leq k, i \in \mathbb{Z}, j \in \mathbb{Z}, k \in \mathbb{Z}\}$$

$$\{ww \mid w \in \{0, 1\}^*\}$$

(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.