CSE 105
THEORY OF COMPUTATION

Winter 2022

https://cseweb.ucsd.edu/classes/wi22/cse105-a/
Today's learning goals

- Apply the Pumping Lemma in proofs of nonregularity
- Identify some nonregular sets
- Define context-free grammars
- Test if a specific string can be generated by a given context-free grammar
Pumping Lemma

If $A$ is a regular language, then there is a number $p$ (the pumping length) where, if $s$ is any string in $A$ of length at least $p$, then $s$ may be divided into three pieces, $s = x y z$ such that

- $|y| > 0$, and
- for each $i \geq 0$, $xy^i z \in A$,
- $|xy| \leq p$. 

Sipser p. 78 Theorem 1.70

# states in DFA recognizing A

Transition labels along loop
Proof strategy

To prove that a language $L$ is not regular

• Assume towards a contradiction that it is.
• Use Pumping Lemma to give $p$, a pumping length for $L$.
• Show that $p$ actually isn't a pumping length for $L$.
• Conclude that $L$ is not regular.
Another example

Claim: The set \( \{a^m b^n a^m | m,n \geq 0 \} \) is not regular.

Proof: …Consider the string \( s = \ldots \).

You must pick \( s \) carefully: we want \(|s| \geq p\) and \( s \) in \( L \).

Now we will demonstrate that "s cannot be pumped", thereby contradicting the assumption that \( p \) is a pumping length.

Which choices of \( s \) cannot be used to complete the proof?

- A. \( s = a^p b^p \)
- B. \( s = ab^p a \)
- C. \( s = a^p b^p a^p \)
- D. \( s = a^p ba^p \)
- E. None of the above (all of these choices work).
Another example

Claim: The set \{a^nb^ma^n | m,n \geq 0\} is not regular.

Proof: …Consider the string \(s = \ldots\)

You must pick \(s\) carefully: we want \(|s| \geq p\) and \(s\) in \(L\). Now we will prove a contradiction with the statement "\(s\) can be pumped"

Consider an arbitrary choice of \(x,y,z\) such that \(s = xyz, \ |y| > 0, \ |xy| \leq p\). This means that… What properties are guaranteed about \(x,y,z\)?

Consider \(i = \ldots\) In this case, \(xy^iz = \ldots\), which is not in \(L\), a contradiction with the Pumping Lemma applying to \(L\) and so \(L\) is not regular.
L = \{a^n b^m a^n : n, m \geq 0\}

Prove that L is not regular. Assume towards a contradiction that L is regular. By the pumping lemma, exists pumping length p for L.

Choose s = a^p b^p a^p. we have: 

\[ s \in L, |s| \geq p. \]

Let x, y, z be arbitrary such: s = xyz, |y| \geq 1, |xy| \leq p.

Need to show: exists i, xy^iz \notin L.

Since s starts with p a's, and |xy| \leq p, must have:

\[ x = a^k, y = a^i, z = a^{p-k-1} b^p a^p \quad l \geq 0, k \geq 1 \]

Choose i = 2. Then: xy^2z = a^{p+k} b^p a^p \notin L.

So L cannot be regular.

xy^2z = a^{p+2l+1} b a^p

\[ s = a^{p+2l} b a^{p+2l} \]

\[ x = a^k y = b^2 \quad \text{or} \quad x = a^k b a^2 y = a^m \]
Regular sets: not the end of the story

- Many **nice / simple / important** sets are not regular
- Limitation of the finite-state automaton model
  - Can't "count"
  - Can only remember finitely far into the past
  - Can't backtrack
  - Must make decisions in "real-time"
- We know computers are more powerful than this model…

Which conditions should we relax?
The next model of computation

• Idea: allow some memory of unbounded size
• How?
  • Generalization of regular expressions
  • Generalization for DFA

Context-free grammars
Pushdown Automata
Birds' eye view

- All languages over Σ
- Context-free languages over Σ
- Regular languages over Σ
- Finite languages over Σ
Context-free grammar

\((V, \Sigma, R, S)\)

**Variables:** finite set of *(usually upper case)* variables \(V\)

**Terminals:** finite set of *alphabet* symbols \(\Sigma\) \(V \cap \Sigma = \emptyset\)

**Rules/Productions:** finite set of allowed transformations \(R\)

\[ A \rightarrow u \quad \text{where} \quad A \in V, u \in (V \cup \Sigma)^* \]

**Start variable:** origination of each derivation \(S\)
Context-free language

The language generated by a CFG \((V, \Sigma, R, S)\) is

\[
\{ w \in \Sigma^* \mid \text{Starting with the Start variable and applying one or more rules, can derive } w \text{ on RHS} \}
\]

If \(G = (V, \Sigma, R, S)\) the language generated by \(G\) is denoted \(L(G)\).

Notation:

\[
S \Rightarrow^* w
\]

Terminology: sequence of rule applications is *derivation*.
An example?

Consider the CFG

\[ \{S\}, \{0\}, R, S \]

where R is the following set of rules

\[
\begin{align*}
S & \rightarrow 0S \\
S & \rightarrow \varepsilon
\end{align*}
\]

Is this a well-formed definition?

A. No: there's more than one rule
B. No: the same LHS gets sent to two different strings.
C. No: one of the string in the RHS has both variables and literals
D. Yes.
E. I don't know.
Context-free language

For CFG $G = (V, \Sigma, R, S)$, $L(G) = \{ w \in \Sigma^* |$ Starting with the Start variable and applying one or more rules, can derive $w$ on RHS$\}$.

What is the language of the CFG $\{(S), \{0\}, R, S\}$ with $R = \{S \rightarrow 0S, S \rightarrow \varepsilon\}$?

A. $\{0\}$  
B. $\{0, 0S\}$  
C. $\{0, 00, 000, \ldots\}$  
D. $\{\varepsilon, 0, 00, 000, \ldots\}$  
E. I don't know.
Context-free language

What is the language of the CFG \( \{S\}, \{0,1\}, R, S \) with

\[ R = \text{the set of rules} \]

\[
S \rightarrow 0S \\
S \rightarrow 1S \\
S \rightarrow \varepsilon
\]

A. \( L(0^*1^*) \)  
B. \( L(0^* U 1^*) \)  
C. \( L( 0 U 1 )^* \)  
D. \( L((0*1^*)^*) \)  
E. I don't know.
Designing a CFG

Building a CFG to describe the language

\{ abba \}

\[ V = \{ S, T, V, W \} \]
\[ \Sigma = \{ a, b \} \]
\[ R = \{ S \rightarrow aT \} \]

\[ S \rightarrow aT \rightarrow abV \rightarrow a bbW \rightarrow abba \]

Can CFGs describe simple sets?
Designing a CFG

Building a CFG to describe the language {abba}

V =
Σ =
R =
S =

What's the set of terminals of this CFG?
A. {a,b}
B. V U S U Σ
C. {S, a, b}
D. {a,b, ε}
E. I don't know.

Can CFGs describe simple sets?
Designing a CFG

Building a CFG to describe the language

\{ \text{abba} \}

\(( \{ S, T, V, W \}, \{ a,b \}, \{ S \rightarrow aT \, , \, T \rightarrow bV \, , \, V \rightarrow bW \, , \, W \rightarrow a \} ), \, S )\)

OR

\(( \{ S \}, \{ a,b \}, \{ S \rightarrow \text{abba} \}, \, S \)\)
Is every regular language a CFL?

- **Approach 1**: start with an arbitrary DFA $M$, build a CFG that generates $L(M)$.

- **Approach 2**: build CFGs for $\{a\}$, $\{\varepsilon\}$, $\{\}$; then show that the class of CFL is closed under the regular operations (union, concatenation, Kleene star).
Approach 1

Claim: Given any DFA M, there is a CFG whose language is \( L(M) \).

Construction:
Trace computation using variables to denote state
Given \( M = (Q, \Sigma, \delta, q_0, F) \) a DFA, define the CFG
\[ V = \{ S_i | q_i \text{ is in } Q \} \]
\[ \Sigma \]
\[ R = \{ S_i \rightarrow aS_j | \delta(q_i, a) = q_j \} \cup \{ S_i \rightarrow \epsilon | q_i \text{ is in } F \} \]
\[ S = S_0 \]

Then prove correctness…
Approach 2

If $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ are CFGs and $G_1$ describes $L_1$, $G_2$ describes $L_2$, how can we combine the grammars so we describe $L_1 \cup L_2$?

A. $G = (V_1 \cup V_2, \Sigma, R_1 \cup R_2, S_1 \cup S_2)$
B. $G = (V_1 \times V_2, \Sigma, R_1 \times R_2, (S_1, S_2))$
C. We might not always be able to: the class of CFG describable languages might not be closed under union.
D. I don't know.
Approach 2

If $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ are CFGs and $G_1$ describes $L_1$, $G_2$ describes $L_2$, how can we combine the grammars so we describe $L_1 \cup L_2$?
Designing a CFG

Building a CFG to describe the language

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

We know this set is not regular!

\[ R = \quad S \rightarrow aSb \mid \epsilon \quad (\{S, b, a, \epsilon, R, S\}) \quad \text{CFG} \]

\[ S \rightarrow \epsilon \]
\[ S \rightarrow aSb \rightarrow aab \rightarrow a \]
\[ S \rightarrow aSb \rightarrow aSbb \rightarrow aabbb \rightarrow aaabbb \]
Designing a CFG

Building a CFG to describe the language

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

One approach:
- what is shortest string in the language?
- how do we go from shorter strings to longer ones?
Designing a CFG

Building a CFG to describe the language

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

V = \{ S \}
Σ = \{ a,b \}
R =

Which rules would complete this CFG?

A. S \rightarrow \epsilon \mid ab
B. S \rightarrow \epsilon \mid aS \mid Sb
C. S \rightarrow \epsilon \mid aSb
D. We need another variable other than S.
E. I don't know.
Designing a CFG

Building a CFG to describe the language

\[ \{ 0^n1^m2^n \mid n,m \geq 0 \} \]

*Hint: work from the outside in.*

Also not a regular set
Designing a CFG

Building a CFG to describe the language

\[ \{ 0^n1^m2^n \mid n,m \geq 0 \} \]

*Hint: work from the outside in.*

\[ V = \{ S, T \} \]
\[ \Sigma = \{ 0, 1, 2 \} \]
\[ R = \{ S \rightarrow 0S \mid T \mid \epsilon, \quad T \rightarrow 1T \mid \epsilon \} \]

S

Also not a regular set