PUMPING LEMMA AND TEST 2 REVIEW

COMP 4200 – Formal Language

NON CFL

- Remark: CFL Pumping Lemma (PL) mainly used to show certain languages are ***not*** CFL.
- Example: Prove that $B = \{ a^n b^n c^n \mid n \geq 0 \}$ is non-CFL.
- Proof:
  - Suppose B is CFL, it implies that B has pumping length $p \geq 1$.
  - Consider string $s = a^p b^p c^p \in B$, so $|s| = 3p \geq p$.
  - PL: can split s into 5 pieces $s = uvxyz = a^p b^p c^p$ satisfying
    - $uv^i xy^i z \in B$ for all $i \geq 0$ → 1
    - $|vy| > 0$ → 2
    - $|vxy| \leq p$ → 3
  - For contradiction, show cannot split $s = uvxyz$ satisfying 1–3.
    - Show every possible split satisfying Condition 2 violates Condition 1.

- Recall s = uvxyz = $\underbrace{aa\ldots\ldots a}_{p}\ \underbrace{bb\ldots\ldots b}_{p}\ \underbrace{cc\ldots\ldots c}_{p}$
- Possibilities for split s = uvxyz satisfying Condition 2: $|vy| > 0$

  (i) Strings v and y are uniform [e.g., v = a ⋯ a and y = b ⋯ b].

  - Then uv^2xy^2z won't have same number of a's, b's and c's because $|vy| > 0$.
  - Hence, $uv^2xy^2z \notin B$.

  (ii) Strings v and y are not both uniform [e.g., v = a ⋯ ab ⋯ b and y = b ⋯ b].

  - Then $uv^2xy^2z \notin L(a^*b^*c^*)$: symbols not grouped together.
  - Hence, $uv^2xy^2z \notin B$.
- Thus, every split satisfying Condition 2 has $uv^2xy^2z \notin B$, so Condition 1 violated.
- **Contradiction,** so $B = \{ a^nb^nc^n \mid n \geq 0 \}$ is not a CFL.

Prove $C = \{ a^i b^j c^k \mid 0 \le i \le j \le k \}$ is not CFL

- Suppose C is CFL, so PL implies C has pumping length p.
- Take string $s = \underbrace{aa\ldots\ldots a}_{p}\,\underbrace{bb\ldots\ldots b}_{p}\,\underbrace{cc\ldots\ldots c}_{p} \in C$, so $|s| = 3p \ge p$.
- PL: can split $s = a^p b^p c^p$ into 5 pieces $s = uvxyz$ satisfying
  1. $uv^i xy^i z \in C$ for every $i \ge 0$,
  2. $|vy| > 0$,
  3. $|vxy| \le p$.
- Condition 3 implies vxy can't contain 3 different types of symbols.
- Two possibilities for v, x, y satisfying $|vy| > 0$ and $|vxy| \le p$:
  - I. If $vxy \in L(a^*b^*)$, then z has all the c's
    - I. string uv^2xy^2z has too few c's because z not pumped
    - II. Hence, $uv^2xy^2z \notin C$
  - II. If $vxy \in L(b^*c^*)$, then u has all the a's
    - I. string $uv^0xy^0z = uxz$ has too many a's
    - II. Hence, $uv^0xy^0z \notin C$
- Every split $s = uvxyz$ satisfying 2–3 violates 1, so C isn't CFL.

Prove $D = \{ ww \mid w \in \{0, 1\}^* \}$ is not CFL

- Suppose D is CFL, so PL implies D has pumping length p. •
- Take $s = \underbrace{00\ldots0}_{p}\ \underbrace{11\ldots1}_{p}\ \underbrace{00\ldots0}_{p}\ \underbrace{11\ldots1}_{p} \in D$, so $|s| = 4p \geq p$.
- PL: can split s into 5 pieces s = uvxyz satisfying
  1. $uv^ixy^iz \in D$ for every $i \geq 0$,
  2. $|vy| > 0$,
  3. $|vxy| \leq p$.

I. If vxy is entirely left of middle of $0^p\ 1^p\ 0^p\ 1^p$,
- then second half of uv^2xy^2z starts with a 1
- so can't write uv^2xy^2z as ww because first half starts with 0.

II. Similar reasoning: if vxy is entirely right of middle of $0^p\ 1^p\ 0^p\ 1^p$,
- then $uv^2xy^2z \notin D$

III. If vxy straddles middle of $0^p\ 1^p\ 0^p\ 1^p$,
- then $uv^0xy^0z = uxz = 0^p\ 1^j\ 0^k\ 1^p \notin D$ (because j or k < p)

- Every split s = uvxyz satisfying 2–3 violates 1, so D isn't CFL.

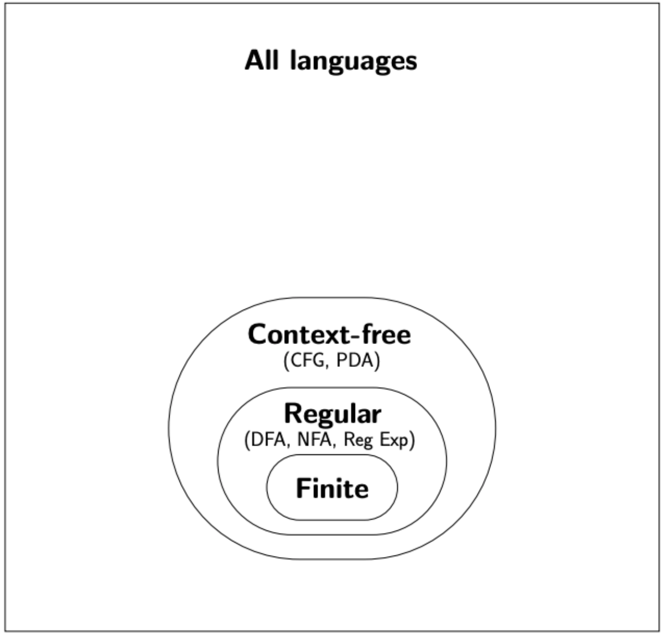
REMARKS ON CFL PUMPING LEMMA

Often more difficult to apply CFL pumping lemma (Theorem 2.34)

than pumping lemma for regular languages (Theorem 1.70).

- Carefully choose string s in language to get contradiction.
  - Not all strings s will give contradiction.
- CFL pumping lemma: "... can split s into 5 pieces s = uvxyz satisfying all of Conditions 1–3."
- To get contradiction, must show cannot split s into 5 pieces s = uvxyz satisfying all of Conditions 1–3.
  - Need to show every possible split s = uvxyz violates at least one of Conditions 1–3.

HIERARCHY OF LANGUAGES (SO FAR)



Examples

$\{ 0^n 1^n 2^n \mid n \geq 0 \}$

$\{ 0^n 1^n \mid n \geq 0 \}$

$(0 \cup 1)^*$

$\{ 110, 01 \}$

TOPICS

- Pumping lemma for regular languages
- CFG
- CNF, GNF
- Derivation of string
- Parse tree
- Ambiguity
- Equivalence of CFG and PDA

BLUE PRINT

Parts	Points
Part 1 – Short answers	
Part 2 – CFG to PDA	
Part 3 – Chomsky Normal Form	
Part 4 – Parse tree, Simplify	
Part 5 – PDA design	
Total	

SHORT ANSWERS

Give CFG for L = {w ∈ {0, 1}* | the length of w is odd and the middle symbol is 0}.

Give CFG for L = {w ∈ {0, 1}* | w = w^R and |w| is even}.

Find L(G) for S→aSa | bSb | a | b

Which of the followings cannot be designed by a PDA?

a. $a^n b^n c^i$, where n, i > 0
b. $a^n b^n c^n$, where n > 0
c. $a^n c^i b^n$, where n, i > 0
d. $c^i a^n b^n$, where n, i > 0

The pumping lemma for regular expression is used to prove that

a. Certain sets are regular
b. Certain sets are not regular
c. Certain regular grammar produce RE
d. Certain regular grammar does not produce RE

1 10 point question

CFG TO PDA

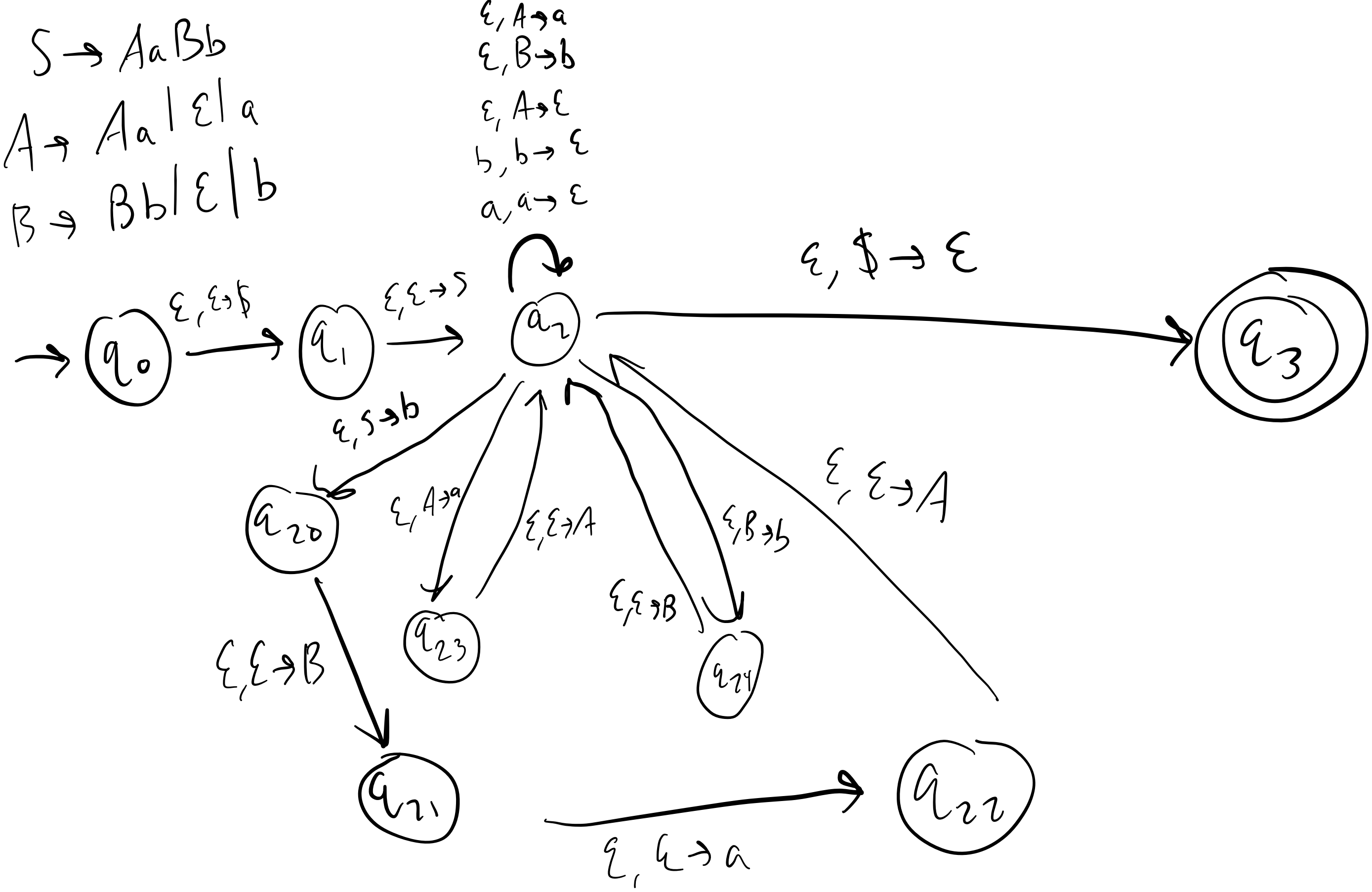
- Construct a PDA equivalent of the grammar given below: S → aAA A →aS| bS| a

$\varepsilon, \beta \to \varepsilon$

$S \to AaBb$

$A \to Aa \mid \varepsilon \mid a$

$B \to Bb \mid \varepsilon \mid b$



1 10 point question

CNF

Convert the following CFG into an equivalent CFG in Chomsky normal form,
S → BSB|B|ε
B → 00|ε

$S \to S X$

$S_0 \to S$

$S \to BSB | B | \varepsilon$

$B \to 00 | \varepsilon$

$S \to \varepsilon$

$S_0 \to S$

$S \to BSB | B | BB$

$B \to 00 | \varepsilon$

$B \to \varepsilon$

$S_0 \to S$

$S \to BSB | B | BB | BS | SB | \cancel{S}$

$B \to 00$

Unit Productions

$S_0 \to BSB | BB | BS | SB | 00$

$S \to BSB | BB | BS | SB | 00$

$B \to 00$

CNF Form

$S_0 \to XB | BB | BS | SB | YY$

$S \to XB | BB | BS | SB | YY$

$B \to YY$

$Y \to 0$

$X \to BS$

$S \to 0A0 \mid 1B1 \mid BB$
$A \to C$
$B \to S \mid \varepsilon$
$C \to S \mid \varepsilon$

$B \to \varepsilon$

$S \to 0A0 \mid 1B1 \mid BB \mid 1B \mid B1 \mid B \mid \varepsilon$
$A \to C$
$B \to S$
$C \to S \mid \varepsilon$

$C \to \varepsilon$

$S \to 0A0 \mid 1B1 \mid BB \mid 1B \mid B1 \mid B \mid \varepsilon$
$A \to C \mid \varepsilon$
$B \to S$
$C \to S$

$A \to \varepsilon$

$S \to 0A0 \mid 1B1 \mid BB \mid 1B \mid B1 \mid B \mid \varepsilon \mid 00$
$A \to C$
$B \to S$
$C \to S$

$S \to \varepsilon$

$S_0 \to S$
$S \to 0A0 \mid 1B1 \mid BB \mid 1B \mid B1 \mid B \mid 00$
$A \to C$
$B \to S$
$C \to S$

Unit Productions

$S \to S_0$
$S \to 0A0 \mid 1B1 \mid BB \mid 11 \mid B \mid 00$
$A \to S$
$B \to S$

$S_0 \to 0A0 \mid 1B1 \mid BB \mid 11 \mid \cancel{S} \mid 00$
$S \to 0A0 \mid 1B1 \mid BB \mid 11 \mid \cancel{S} \mid 00$
$A \to 0A0 \mid 1B1 \mid BB \mid 11 \mid \cancel{S} \mid 00$
$B \to 0A0 \mid 1B1 \mid BB \mid 11 \mid \cancel{S} \mid 00$

SIMPLIFY THE GRAMMAR AND PARSE TREE

ε prod. | unit prod. | useless symbol

3 ways

Simplify the given grammar,

S → AB
A → a
B → C | b
C → D
D → E | bC
E → d | Ab

Useless - nonterminating | nonreachable

S → ASB | ε
A → aAS | a
B → SbS | A | bb

PARSE TREE

Prove that the following CFG is ambiguous. $S \rightarrow S + S \mid S * S \mid (S) \mid a$
Draw parse tree for the string "a + a * a".

1 10 point question

PDA DESIGN

Construct a PDA for the language L = {a^nCb^{2n}, where n ≥ 1}.

Logic:

Read a push a

Read C do nothing

Read odd b do nothing

Read even b pop a

$\delta(q_0, a, z_0) = (q_1, az_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, c, a) = (q_2, a)$

$\delta(q_2, b, a) = (q_3, a)$

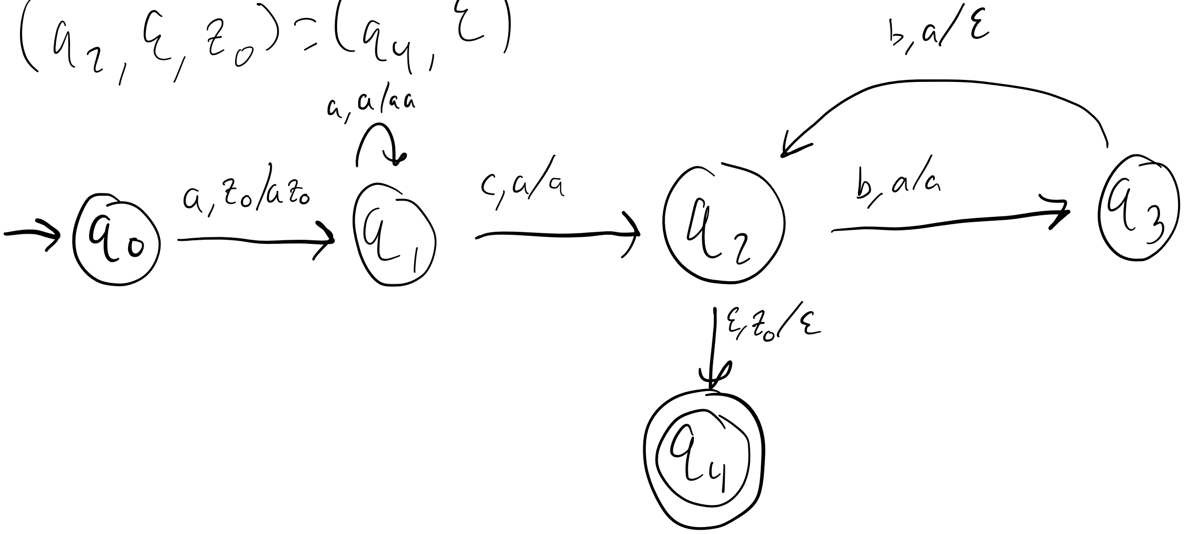
$\delta(q_3, b, a) = (q_2, \varepsilon)$

$\delta(q_2, \varepsilon, z_0) = (q_4, \varepsilon)$

Go to new state for new variable

$\delta(x, y, z) = (x, y)$

current state, reading, top of stack, new state, new stack



X=X

$L = \{ a^p b^q c^m \mid p + m = q \}$ $\quad m = q - p$

Logic: Read a push a
Read b pop a

$\delta(q_0, a, z_0) = (q_1, az_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, \varepsilon)$

$\delta(q_2, b, a) = (q_2, \varepsilon)$

$\delta(q_2, b, z_0) = (q_3, bz_0)$

$\delta(q_3, b, b) = (q_3, bb)$

$\delta(q_3, c, b) = (q_4, \varepsilon)$

$\delta(q_4, c, b) = (q_4, \varepsilon)$

$\delta(q_4, \varepsilon, z_0) = (q_5, \varepsilon)$