CS150 Homework 4

Solution keys, fall, 2026

Problem 1. (10 points)

Construct a CFG for the set of all ternary strings of the form $0^i1^j2^k$, where $k = i + j$.

$$S \to 0S2 \mid X$$
$$X \to 1X2 \mid \epsilon$$

Problem 2. (Exercise 5.2.1, 10 points)

For the grammar and each of the strings in Exercise 5.1.2, give parse tree.

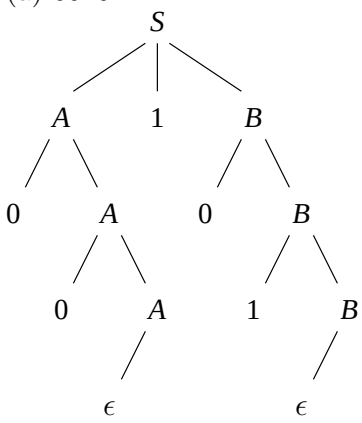
> **Exercise 5.1.2:** The following grammar generates the language of regular expression $0^*1(0+1)^*$:
>
> $$S \to A1B$$
> $$A \to 0A \mid \epsilon$$
> $$B \to 0B \mid 1B \mid \epsilon$$

(a) 00101.

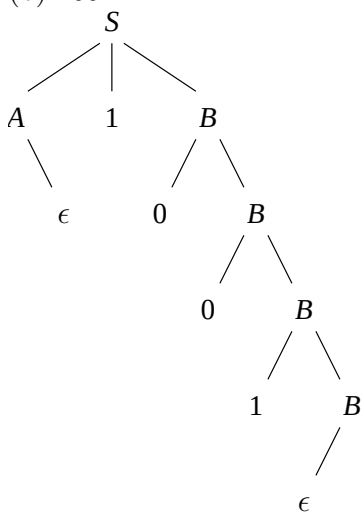
(b) 1001.

(c) 00011.

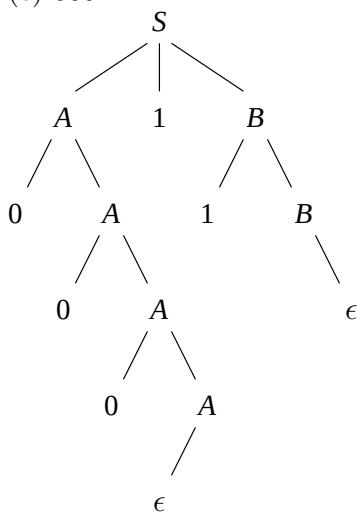
(a) 00101



(b) 1001



(c) 00011



Problem 3. (Exercise 5.4.7, 20 points)

The following grammar generate *prefix* expressions with operands x and y and binary operators $+$, $-$, and $*$:

$$E \to +EE \mid *EE \mid -EE \mid x \mid y$$

(a) Find leftmost and rightmost derivations, and a derivations tree for the string `+*-xyxy`.

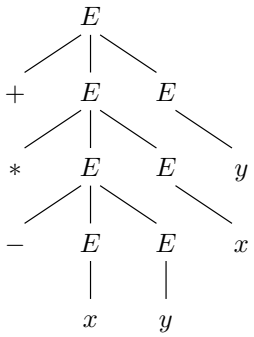
(b) Prove that this grammar is unambiguous. (*Hint*: show that the leftmost derivation is unique for any given input string.

(a) Leftmost derivation:

$$\begin{aligned} E &\underset{lm}{\Rightarrow} +EE \\ &\underset{lm}{\Rightarrow} +*EEE \\ &\underset{lm}{\Rightarrow} +*-EEEE \\ &\underset{lm}{\Rightarrow} +*-xEEE \\ &\underset{lm}{\Rightarrow} +*-xyEE \\ &\underset{lm}{\Rightarrow} +*-xyxE \\ &\underset{lm}{\Rightarrow} +*-xyxy \end{aligned}$$

Rightmost derivation:

$$\begin{aligned} E &\underset{rm}{\Rightarrow} +EE \\ &\underset{rm}{\Rightarrow} +Ey \\ &\underset{rm}{\Rightarrow} +*EEy \\ &\underset{rm}{\Rightarrow} +*Exy \\ &\underset{rm}{\Rightarrow} +*-EExy \\ &\underset{rm}{\Rightarrow} +*-Eyxy \\ &\underset{rm}{\Rightarrow} +*-xyxy \end{aligned}$$



(b) **Proof.** In this grammar, the application of each rule generates a string starting with a unique terminal symbols ($+$, $*$, $-$, x, or y). For any string w that belongs to the CFL, when we consider a leftmost variable E in the leftmost derivation of the string, there is only one rule that can be used to continue the derivation. This rule is uniquely determined by the next symbol in w to be derived. So there is only one leftmost derivation for w and hence the unambiguity of the grammar. □

Problem 4. (Exercise 6.1.1 (b)(c), 10 points)

Suppose the PDA $P = (\{q, p\}, \{0, 1\}, \{Z_0, X\}, \delta, q, Z_0, \{p\})$ has the following transition function:

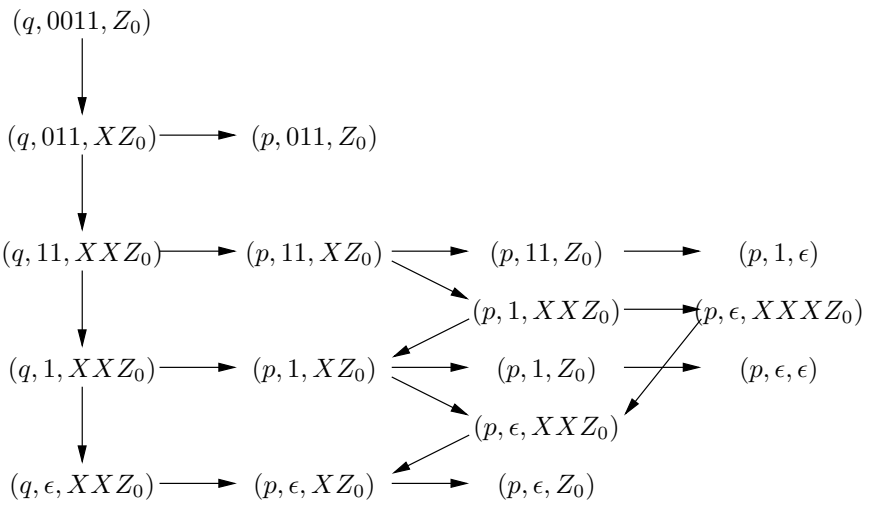
1. $\delta(q, 0, Z_0) = \{(q, XZ_0)\}$.
2. $\delta(q, 0, X) = \{(q, XX)\}$.
3. $\delta(q, 1, X) = \{(q, X)\}$.
4. $\delta(q, \epsilon, X) = \{(p, \epsilon)\}$.
5. $\delta(p, \epsilon, X) = \{(p, \epsilon)\}$.
6. $\delta(p, 1, X) = \{(p, XX)\}$.
7. $\delta(p, 1, Z_0) = \{(p, \epsilon)\}$.

Starting from the initial ID (q, w, Z_0), show all the reachable ID's when the input w is:

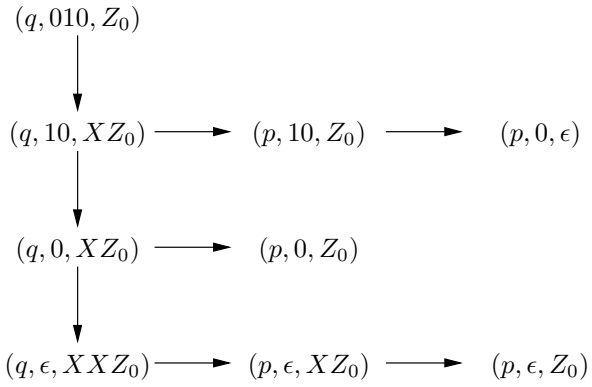
(b) 0011.

(c) 010.

(b) 0011:



(c) 010:


$(q, 010, Z_0)$
$(q, 10, XZ_0)$
$(p, 10, Z_0)$
$(p, 0, \epsilon)$
$(q, 0, XZ_0)$
$(p, 0, Z_0)$
(q, ϵ, XXZ_0)
(p, ϵ, XZ_0)
(p, ϵ, Z_0)

Problem 5. (Exercise 6.2.1 (b)(c), 20 points)

Design a PDA to accept each of the following languages. You may accept either by final state or by empty stack, whichever is more convenient.

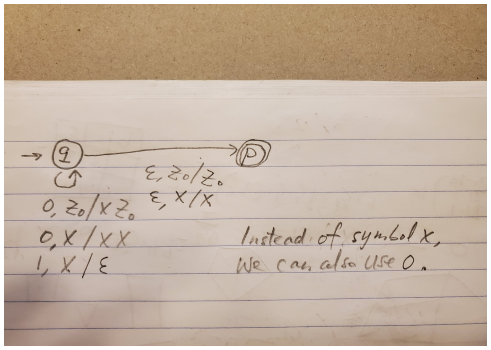
(b) The set of all strings of 0's and 1's such that no prefix has more 1's than 0's.

(c) The set of all strings of 0's and 1's with an equal number of 0's and 1's.

(b) PDA $P = (\{q, p\}, \{0, 1\}, \{Z_0, X\}, \delta, q, Z_0, \{p\})$:

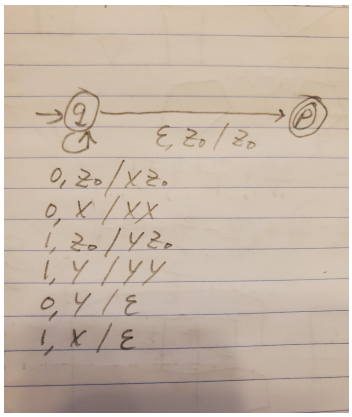
$$\delta(q, 0, Z_0) = \{(q, XZ_0)\}$$
$$\delta(q, 0, X) = \{(q, XX)\}$$
$$\delta(q, 1, X) = \{(q, \epsilon)\}$$
$$\delta(q, \varepsilon, X) = \{(p, X)\}$$
$$\delta(q, \epsilon, Z_0) = \{(p, Z_0)\}$$

To show an empty-stack PDA, replace (p,X) and (p,Z_0) by (q,\eps) in the last two transitions.



(c) PDA $P = (\{q, p\}, \{0, 1\}, \{Z_0, X, Y\}, \delta, q, Z_0, \{p\})$:

$$\delta(q, 0, Z_0) = \{(q, XZ_0)\}$$
$$\delta(q, 0, X) = \{(q, XX)\}$$
$$\delta(q, 0, Y) = \{(q, \epsilon)\}$$
$$\delta(q, 1, Z_0) = \{(q, YZ_0)\}$$
$$\delta(q, 1, Y) = \{(q, YY)\}$$
$$\delta(q, 1, X) = \{(q, \epsilon)\}$$
$$\delta(q, \epsilon, Z_0) = \{(p, Z_0)\}$$



Note that you are supposed to show the diagrams for each PDA although our solution keys list a sequence of transitions for each PDA.