BLG 311E Formal Languages and Automata

Recitation-1

Büşra Bayram
Büşranur Bülbül
Erdi Sarıtaş

Misc: Proving

Question

Prove $\sqrt{3}$ is irrational using proof-by-contradiction.

Solution

- Assume $\sqrt{3}$ is rational → then, we can write $\sqrt{3} = a/b \rightarrow a = \sqrt{3}b$
    - *a, b* are whole numbers, *b* not zero
- Let's take square of them → $a^2 = 3b^2$
- We can see a^2 is divisible by 3→ then, *a* should be too.
- Let's replace a with *3k* where $a = 3k$
- This makes, $a^2 = (3k)^2 = 9k^2 = 3b^2 \rightarrow b^2 = 3k^2$
- This mean b^2 is also divisible by 3. This makes *b* divisible by 3 also.
- This is a contradiction since *a/b* is assumed to simplified to lowest terms.
- However, if both *a* and *b* divisible by 3, they can be simplified more.
- As our initial assumption result in a contradiction ,$\sqrt{3}$ cannot be rational.

From number theory:
If a prime number d **divides** p^2, then d must **divide** p also.

Question

Prove the below relation using proof-by-induction.

$$\sum_{k=1}^{n} k \cdot 2^k = (n-1) \cdot 2^{n+1} + 2$$

Solution

Base Case → n=1

$$\sum_{k=1}^{n} k \cdot 2^k = (n-1) \cdot 2^{n+1} + 2$$

$$\sum_{k=1}^{1} k \cdot 2^k = 1 \cdot 2^1 = 2$$

$$(1-1) \cdot 2^{1+1} + 2 = 0 \cdot 2^2 + 2 = 2$$

Both sides result in 2 → base case holds

Solution (cont.)

$$\sum_{k=1}^{n} k \cdot 2^k = (n-1) \cdot 2^{n+1} + 2$$

Inductive Steps → assume n holds, prove n+1 also holds

$$\sum_{k=1}^{n+1} k \cdot 2^k = (n) \cdot 2^{n+2} + 2$$

$$\sum_{k=1}^{n+1} k \cdot 2^k = \sum_{k=1}^{n} k \cdot 2^k + (n+1) \cdot 2^{n+1} = \left((n-1) \cdot 2^{n+1} + 2\right) + (n+1) \cdot 2^{n+1}$$

$$(n-1) \cdot 2^{n+1} + (n+1) \cdot 2^{n+1} + 2$$

$$((n-1) + (n+1)) \cdot 2^{n+1} + 2$$

$$2n \cdot 2^{n+1} + 2$$

$$n \cdot 2^{n+2} + 2$$

Formal Languages

Question

Transform the given Moore machine into the Mealy model.

	0	1	Output
S_0	S_4	S_2	1
S_1	S_4	S_2	1
S_2	S_5	S_0	0
S_3	S_7	S_6	0
S_4	S_1	S_4	0
S_5	S_0	S_4	0
S_6	S_3	S_2	1
S_7	S_1	S_5	0

Mealy

	0	**1**
S_0		
S_1		
S_2		
S_3		
S_4		
S_5		
S_6		
S_7		

Solution

Moore

	0	1	Output
S_0	S_4	S_2	1
S_1	S_4	S_2	1
S_2	S_5	S_0	0
S_3	S_7	S_6	0
S_4	S_1	S_4	0
S_5	S_0	S_4	0
S_6	S_3	S_2	1
S_7	S_1	S_5	0

Mealy

	0	1
S_0	$S_4/0$	$S_2/0$
S_1	$S_4/0$	$S_2/0$
S_2	$S_5/0$	$S_0/1$
S_3	$S_7/0$	$S_6/1$
S_4	$S_1/1$	$S_4/0$
S_5	$S_0/1$	$S_4/0$
S_6	$S_3/0$	$S_2/0$
S_7	$S_1/1$	$S_5/0$

Question

Word production rules defined over the input alphabet {a, t} for an imaginary language complies with the following syllable rules:

- Empty words are not allowed.
- Wovels cannot be repeated in a word without a consonant in between.
- Any consonant can be repeated at most two times
- Words cannot start or end with a consonant repetition
- A single consonant cannot form a word

Provide a Moore machine definition that follows the given rules, where the letter Y is output for the words in the language and N is output for the words that does not belong to the language.

a) Provide the formal definition of the Moore machine by enlisting the elements of S, I, O, δ, ω.
b) Define the machine by providing a table of inputs, outputs, states and transitions
c) Define the machine by drawing the state-transition diagram of it

Solution

a) Provide the formal definition of the Moore machine by enlisting the elements of *S,I,O,δ,ω*

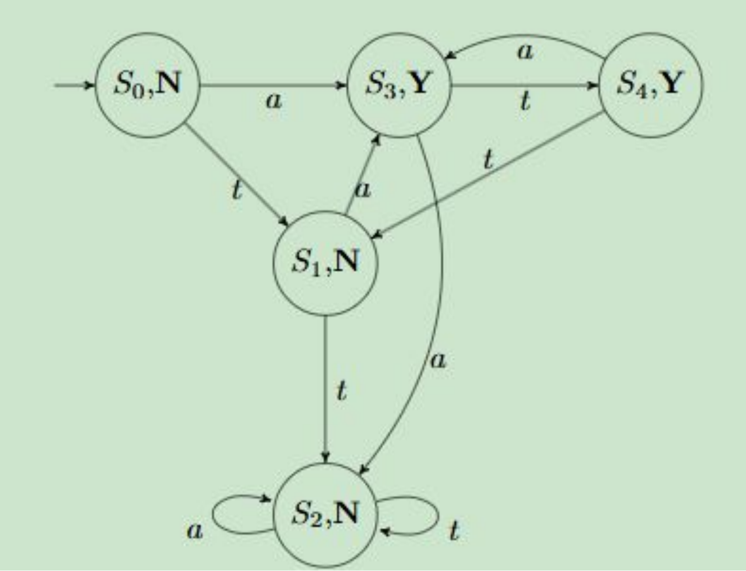
- S: {S0,S1,S2,S3,S4 }
- I : {a,t}
- O : {**Y, N**}
- δ: $\{(S_0,a,S_3),(S_1,a,S_3),(S_2,a,S_2),(S_3,a,S_2),(S_4,a,S_3),(S_0,t,S_1),(S_1,t,S_2),(S_2,t,S_2),(S_3,t,S_4),(S_4,t,S_1)\}$
- *ω:* $\{(S_0,N),(S_1,N),(S_2,N),(S_3,Y),(S_4,Y),\}$

Solution (cont.)

b)Define the machine by providing a table of inputs, outputs, states and transitions

	a	t	ω
S_0	S_3	S_1	**N**
S_1	S_3	S_2	**N**
S_2	S_2	S_2	**N**
S_3	S_2	S_4	**Y**
S_4	S_3	S_1	**Y**

c)Define the machine by drawing the state-transition diagram of it



Alphabets
and Languages

Question

Let Σ be any alphabet and let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$

a) Let $\Lambda \notin L_1$ Explain why $L_1\Sigma^* \neq \Sigma^*$

b) Let $\Lambda \in L_1$ and $\Lambda \in L_2$. Show using axioms and theorems of languages $(L_1\Sigma^*L_2)^* = \Sigma^*$

Solution

Let Σ be any alphabet and let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$

a) Let $\Lambda \notin L_1$ Explain why $L_1\Sigma^* \neq \Sigma^*$

- $L_1\Sigma^* = \{wv \mid w \in L_1, v \in \Sigma^*\}$, meaning every string starts with a prefix from L_1
- Σ^* contains all possible strings, including Λ (the empty string)
- If $\Lambda \notin L_1$, then no string in $L_1\Sigma^*$ can be purely from Σ^* without a prefix from L_1
- Thus, $\Lambda \notin L_1$, meaning $L_1\Sigma^* \neq \Sigma^*$.

b) Let $\Lambda \in L_1$ and $\Lambda \in L_2$. Show using axioms and theorems of languages $(L_1\Sigma^*L_2)^* = \Sigma^*$

- $L_1\Sigma^*L_2$ consists of words in the form ***uvx***, where $\boldsymbol{u \in L_1}$**,** $\boldsymbol{v \in \Sigma^*}$**,** and $\boldsymbol{x \in L_2}$**.**
- Since $\Lambda \in L_1$ and $\Lambda \in L_2$, we can choose $u = \Lambda$ and $x = \Lambda$, meaning **every string** $\boldsymbol{v \in \Sigma^*}$ **is included** in $L_1\Sigma^*L_2$.
- Taking the Kleene star (*) ensures all repetitions remain in Σ^*, so $(L_1\Sigma^*L_2)^* = \Sigma^*$.

Question

Let A and B be languages defined over Σ and $\Lambda \notin B$,

a) Propose a solution to the equation $A \cup XB = X$.
b) Show that your solution is correct.
c) Let $A=\{a, b\}$ and $B=\{aa, ab, ba, bb\}$. Write the solution set and give 5 example words from it.

Solution

a) Solution is $X=AB^*$

b) When we replace X in the equation:

$A \cup (AB^*)B = A \cup AB^+ = A(\{\Lambda\} \cup B^+) = AB^*$

Solution is correct.

c) $AB^* = \{a, b\}\{aa, ab, ba, bb\}^*$

Examples: {a}, {aaa}, {baa}, {bbbaa}, {aabbabb}

Question

Consider the following languages A, B and C defined over the alphabet $\Sigma = \{a,b\}$.

- $A=abb^{+}ba$
- $B=a(bb)^{+}ba$
- $C=a(bbb)^{*}a$

Answer each of the following questions considering the definition above.

a) Give an example string that is accepted by the all three languages.

b) Give an example string that is accepted by only A.

c) Give an example string that is accepted by only A and B.

d) Give an example string that is accepted by only A and C.

e) Give an example string that is accepted by only C.

f) Indicate if there is a subset/superset relation between any pair of the three languages.

Solution

- $A=abb^{+}ba \rightarrow ab^{n}a\ ,\ n \geq 3$
- $B=a(bb)^{+}ba \rightarrow ab^{2n+1}\ a\ ,\ n > 0$
- $C=a(bbb)^{*}a \rightarrow ab^{3n}a\ ,\ n \geq 0$

Answer each of the following questions considering the definition above.

a) Give an example string that is accepted by the all three languages : *abbba*
b) Give an example string that is accepted by only *A* : *abbbba*
c) Give an example string that is accepted by only *A* and *B*: *abbbbba*
d) Give an example string that is accepted by only *A* and *C*: *abbbbbba*
e) Give an example string that is accepted by only *C*: *aa*
f) Indicate if there is a subset/superset relation between any pair of the three languages.

$B \subset A$ as $(ab^{2n+1}a,\ n>0) \subset (ab^{n}a,\ n \geq 3)$

C has no subset/superset relation with the others as it is the only language that accepts and it is more restrictive than A and B for b's.

Question

Show that following expressions hold. If they do not hold give a counterexample.

A) $A^+A^+=A^+$

B) $(A^*B^*)^* = (B^*A^*)^*$

C) $(AB)^* = (BA)^*$

Solution

A) $\mathbf{A^+A^+=A^+}$

- **Does not hold.**
- **Let :** A={1}:
- A^+ ={1,11,111,1111,...,1^n,...}
- A^+A^+ = {11,111,1111,...,1^n,...}

B) (A*B*)* = (B*A*)*

- **Holds.**
- **By using Theorem 13 on page 55 of the slides:**
- $(A^*B^*)^* = (A \cup B)^* \rightarrow (B^*A^*)^* = (B \cup A)^* \rightarrow (B \cup A)^* = (A \cup B)^*$

C) (AB)* = (BA)*

- **Does not hold.**
- A = {0} and B = {1}
- $(AB)^* = \{\Lambda, 01, 0101, 010101, \ldots, (01)\ n\ , \ldots\}$ $(BA)^* = \{\Lambda, 10, 1010, 101010, \ldots, (10)\ n\ , \ldots\}$

Question

Given the language L = {ab, aa, baa}, which of the following string(s) are in L*? Explain your answers.

i) abaabaaabaa
ii) aaaabaaaa
iii) baaaaabaaaab
iv) baaaabaa

Solution

Given the language L = {ab, aa, baa}, which of the following string(s) are in L*? Explain your answers.

i) abaabaaabaa
ii) aaaabaaaa
iii) baaaaabaaaab
iv) baaaaabaa

Solution : i,ii and iv

Languages and Grammars - Chomsky Hierarchy of Grammars

Question

Consider the following grammar.

$$A \rightarrow AaA$$
$$A \rightarrow b$$

a) Find the language generated by the grammar.
b) What is the type of the grammar according to Chomsky hierarchy? Why?
c) Design another grammar with a more restrictive type that generates the same expression you found in (a). (e.g. if the given grammar is Type-1 design a Type-2 or Type-3 grammar.)
d) What is the type of the grammar you designed in (c)? Why?

Solution

a) (ba)*b or b(ab)*

b) It is type-2 since:

1. The length of the left side is always smaller than or equal to the length of the right side, and
2. The left side always consists of a single nonterminal.

It is not type-3 because there are multiple nonterminals on the right side (**AaA**).

c) New grammar can be:

A ::= bB | b
B ::= aA

d) It is type-3 since (1) length of the left side is always smaller than or equal to the length of the right side, (2) left side always consists of a single nonterminal, and (3) right side consists of a number of terminals, possibly followed by a single nonterminal.

Question

Given the language,

$L = \{ w \mid w \in \{0,1\}^* \text{ and } w \text{ does not contain } 011 \text{ as a substring} \}$,

a) Provide a Type-3 grammar for this language.

b) Write the regular expression for this language.

Solution

a.

$$\langle q_0 \rangle ::= 0\langle q_1 \rangle \mid 0 \mid 1\langle q_0 \rangle \mid 1$$

$$\langle q_1 \rangle ::= 0\langle q_1 \rangle \mid 0 \mid 1\langle q_2 \rangle \mid 1$$

$$\langle q_2 \rangle ::= 0\langle q_1 \rangle \mid 0$$

b.

$$L = 1^*(0 \vee 01)^*$$

Question

Produce the grammars for the following languages. State what type the grammar is in the Chomsky Hierarchy.

a) The language composed of strings containing an arbitrary number of sub-strings A or B followed by a single substring B.
b) The language composed by an even number of A's followed by an even number of B's.
c) The language over 0, 1 containing strings which have at least one 0 sur-rounded by 1's.

Solution

a) $(A|B)^*B$.

Grammar:

- $S \to AS \mid BS \mid B$

Chomsky Type: Regular (Type 3)

b) $\{ A^{2m}B^{2n} \mid m, n \geq 0 \}$

Grammar:

- $S \to AAS \mid BBY \mid AA \mid BB \mid \Lambda$
- $Y \to BBY$

Here, S generates an even number of A's and Y an even number of B's.

Chomsky Type: Regular (Type 3)

c) Strings over {0,1} having at least one occurrence of “101” (i.e., a 0 surrounded by 1’s).

Grammar:

- $S \rightarrow 0S \mid 1A$
- $A \rightarrow 1A \mid 0B$
- $B \rightarrow 0S \mid 1C \mid 1$
- $C \rightarrow 0C \mid 1C \mid 0 \mid 1$

Chomsky Type : Regular (Type 3)

Question

Determine the production rules for the grammars of the languages below and identify which Chomsky class they belong to.

a) $L(G) = \{a^n b^n \mid n \geq 1\}$

b) $L(G) = \{a^n b^{n+m} \mid n \geq 1, m \geq 1\}$

Solution

a) Grammar:

- $S \to a\,S\,b$
- $S \to ab$

Chomsky Type: Context-Free (Type 2)

Brief: The rule $S \to a\,S\,b$ adds one aa and one bb each time, while $S \to ab$ serves as the base case.

b) Grammar:

- $S \to T\,B$
- $T \to a\,T\,b \mid ab$
- $B \to b\,B \mid b$

Chomsky Type: Context-Free (Type 2)

Brief: T generates balanced $a^n b^n$ strings, and B adds at least one extra b.

Question

Consider the following grammar:

<S> ::= <A><B>
<A> ::= x<C>y
<C>y ::= y<C>y | y
x<C> ::= <C>
<B> ::= x<D>
<D> ::= <D>y | y
x<D> ::= x | <D>

a)State the **Chomsky type** of this **grammar**.

b)Provide the **most simple** Type-3 grammar for this language.

Solution

a) Chomsky Type:
This grammar is Type-0

b) The Most Simple Type-3 Grammar:

Regular expression is $(x \text{ or } \Lambda)y^{+}(x \text{ or } y)y^{*}$
An equivalent regular grammar for the given language is:

$S \rightarrow xA \mid yY$
$A \rightarrow yY$
$Y \rightarrow yY \mid xW \mid yW \mid x \mid y$
$W \rightarrow yW \mid y$

Question

Consider the following grammar,

$$S \rightarrow AB \mid BC$$
$$B \rightarrow AB \mid BC$$
$$AB \rightarrow ab$$
$$BC \rightarrow bc$$
$$A \rightarrow a$$
$$C \rightarrow c$$

a) State the type of the given grammar.

b) Heuristically state the regular expression that corresponds to the given grammar.

c) Provide an equivalent Type-3 grammar.

Solution

a) Type of the given grammar:

This grammar is **Type-1 .**

b) Regular Expression (heuristically):

The language generated consists of strings formed by combinations of ab and bc. The simplest strings can be constructed as ab or bc .We can enlarge the strings from left side by a's or from the right side by c's.This indicates that the regular expression is a^+bc^*or a^*bc^+.

c) Equivalent Type-3 Grammar:

Since Type-3 grammars are regular, we need right-linear production rules. An equivalent right-linear grammar is:

$S \rightarrow aA \mid aD \mid bC$

$A \rightarrow aA \mid bC \mid b$

$B \rightarrow bC \mid b$

$C \rightarrow cC \mid c$

$D \rightarrow aD \mid bC$

Regular Languages and Regular Expressions

Question

For the table below, fill out each cell with Y when the regular expression in each row can produce the word in the corresponding column; and N vice versa.

	abcabc	aaaccc	bcac	ccc
(a v b)*(b v c)*				
a^+(b v c)*				
(a v b)$^+$c*				
a*b* v b*c*				
(ab* v bc*)*				

Solution

	abcabc	aaaccc	bcac	ccc
(a v b)*(b v c)*	N	Y	N	Y
a^+(b v c)*	N	Y	N	N
(a v b)$^+$c*	N	Y	N	N
a*b* v b*c*	N	N	N	Y
(ab* v bc*)*	Y	N	N	N

Question

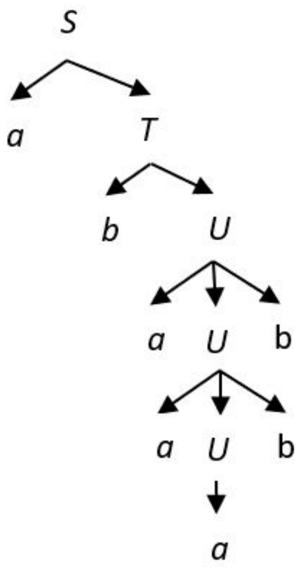
Consider the following grammar:

- $S \rightarrow aT \mid baS$
- $T \rightarrow bU \mid b$
- $U \rightarrow S \mid aUb \mid a$

1. Find which of the words aaab, aabbaab, and abaaabb can be derived
2. Provide an arithmetic rule between the number of a's (#a) and b's (#b) in the words derived

Solution

1. Only the abaaabb can be derived
2. Starting from the initial non-terminal and eliminating parts where #a = #b:
   - S → aT v baS → **(ba)***aT ⇒ aT
   - aT → a(bU v b) → abU v ab
   - S → ab$\mathbf{a}^{n}$(S v a)$\mathbf{b}^{n}$ v ab ⇒ abS v aba v ab
   - S → ab(S v a v λ) → **(ab)***(a v λ) ⇒ a v λ
   - *Thus #a = (#b +1) v #b*



Context Free Languages and Parsing

Question

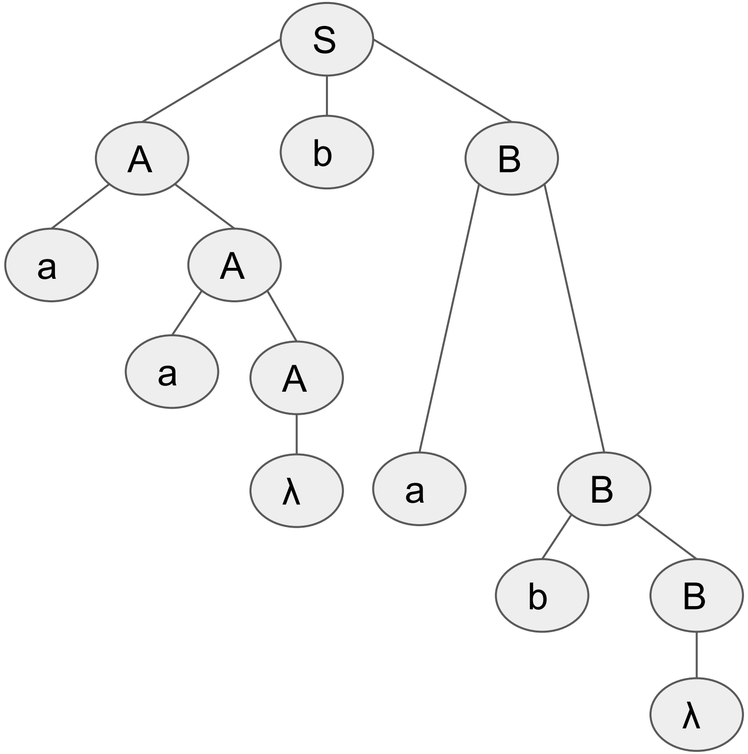
Consider the following grammar and find the language generated by the grammar:

- $S \rightarrow AbB$
- $A \rightarrow aA \mid \lambda$
- $B \rightarrow aB \mid bB \mid \lambda$

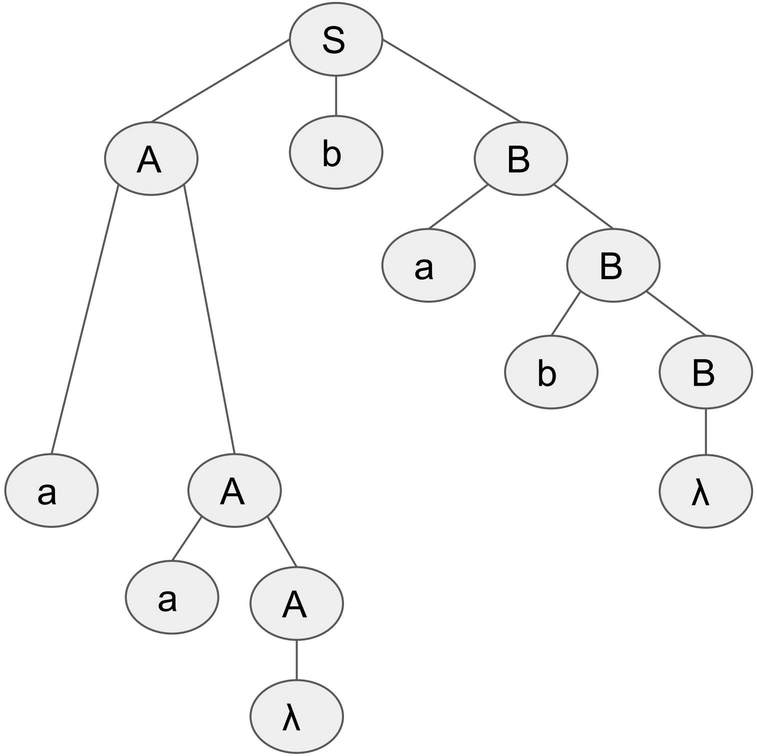
Derive the leftmost and rightmost parse trees of the following string: “aabab”

Solution

Left-Most



Right-Most



Question

Consider the following grammar and find the language generated by the grammar:

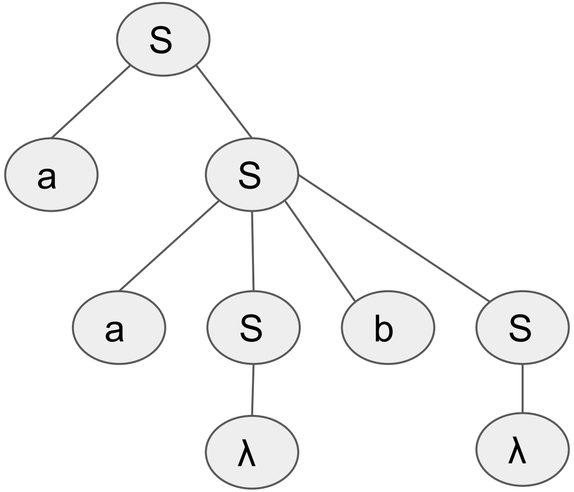
- $S \rightarrow aS \mid aSbS \mid \lambda$

A. Derive the leftmost and rightmost parse trees of the following string: “aab”
B. The given grammar is ambiguous. Find an unambiguous version.

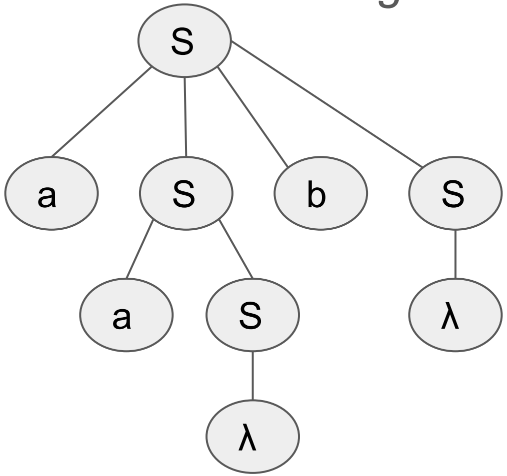
Solution

A. Derive parse trees

Left-Most



Right-Most



Solution (cont.)

B. Find an unambiguous grammar.

The idea is to introduce another nonterminal T that cannot generate an unbalanced a. That strategy corresponds to the usual rule in programming languages that an “else” is associated with the closest previous unmatched “then”. Here, we force a b to match the previous unmatched a. The grammar:

- $S \to aS \mid aTbS \mid \lambda$
- $T \to aTbT \mid \lambda$

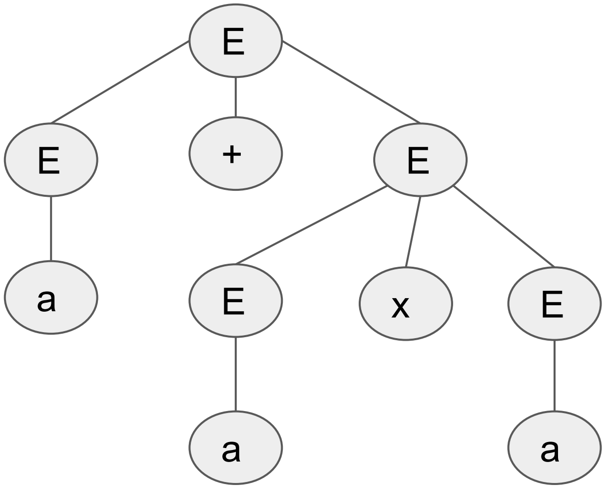
Question

Determine that whether the string “ a+a*a ” has ambiguity or not using the given CFG below;

- $E \rightarrow E+E \mid ExE \mid (E) \mid a$

Solution

Parse Tree 1



Different parse trees, same result! The CFG is ambiguous!

Parse Tree 2