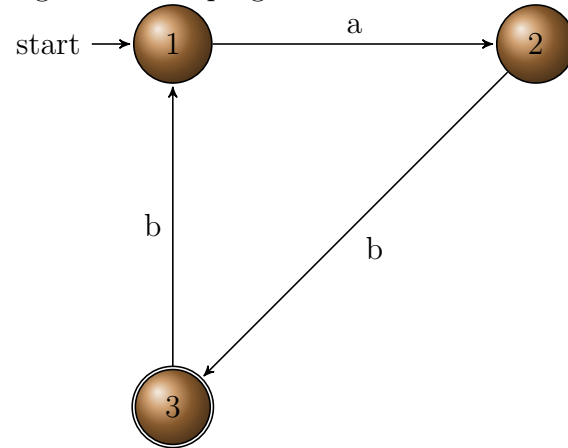


# CS375 Week 6

Jason N Mansfield

August 14, 2011

Figure 1: Pumping Lemma Contradiction



## 0.1

$$L = \{a^n b^{2n} \mid n \geq 1\} = \{abb, aabbbb, aaabbbbbbb, \dots\}$$

*Proof.* For any regular language  $L$ , there exists a number  $p$  such that for any string  $w$  in  $L$  of length at least  $p$  there are strings  $x,y,z$  such that

- $w = xyz$
- $|xy| \leq p$
- $|y| \geq 1$
- Then  $x = a^n, y = a^n, z = b^{p+1}$
- $xy^2z \notin L$
- This is a contradiction so the shown language is nonregular.

□

## 0.2

### 1. Palindromes

*Proof.* For any regular language  $L$ , there exists a number  $p$  such that for any string  $w$  in  $L$  of length at least  $p$  there are strings  $x,y,z$  such that

- (a) If  $w = xyz$ .
- (b) And if  $x = a, y = b, z = a$ .
- (c) or if  $x = b, y = a, z = b$ .
- (d) Then  $w = a^{90+1}ba^{90}$
- (e) and  $w = b^{90+1}ab^{90}$
- (f) Therefore Palindromes are nonregular

□

## 2. Equal

*Proof.* For any regular language  $L$ , there exists a number  $p$  such that for any string  $w$  in  $L$  of length at least  $p$  there are strings  $x,y,z$  such that

- (a)  $EQUAL = \{\Lambda \quad ab \quad ba \quad aabb \quad abab \quad abba \quad baab \quad baba \quad bbaa \quad aaabbb...\}$
- (b)  $\{a^n b^n\} = \mathbf{a^*b^*} \cap EQUAL$
- (c) If  $\mathbf{a^*b^*}$  is regular then so is the result of  $\mathbf{a^*b^*} \cap EQUAL$
- (d)  $w = xyz$
- (e)  $|xy| \leq p$
- (f)  $|y| \geq 1$
- (g) Then  $x = a^n, y = a^2, z = b^n$
- (h) So then  $xxyz$  would allow  $aaab$  which is not in this language.
- (i) Therefore this language is nonregular

□

## 0.3

Prove that the below generates the language defined by the regular expression:  $\mathbf{a^*bb}$

**Prod1**  $S \rightarrow \mathbf{aS} \mid \mathbf{bb}$

$$\begin{aligned}
 S &\Rightarrow aS \\
 &\Rightarrow aaS \\
 &\Rightarrow aaaS \\
 &\Rightarrow aaaaS \\
 &\Rightarrow aaaaaS \\
 &\Rightarrow aaaaaabb
 \end{aligned}
 \tag{1}$$

This derivation could continue infinitely until terminal bb is appended.

## 0.4

To generate aabbab using:  $a(a + b)^*$

$$\begin{aligned}
&\textbf{Prod1} \quad S \rightarrow \mathbf{aX} \\
&\textbf{Prod2} \quad X \rightarrow \mathbf{aX} \mid \mathbf{bX} \mid \Lambda \\
S &\implies aX \quad (\text{by Prod 1}) \\
&\implies aaX \quad (\text{by Prod 2}) \\
&\implies aabX \quad (\text{by Prod 2}) \\
&\implies aabbX \quad (\text{by Prod 2}) \\
&\implies aabbaX \quad (\text{by Prod 2}) \\
&\implies aabbabX \quad (\text{by Prod 2}) \\
&\implies aabbab\Lambda \quad (\text{by Prod 2})
\end{aligned} \tag{2}$$

## 0.5

To generate bbabaaa using:  $(a + b)^*a(a + b)^*a(a + b)^*$

$$\begin{aligned}
&\textbf{Prod1} \quad S \rightarrow \mathbf{XaXaX} \\
&\textbf{Prod2} \quad X \rightarrow \mathbf{aX} \mid \mathbf{bX} \mid \Lambda \\
S &\implies XaXaX \quad (\text{by Prod 1}) \\
&\implies XaXaaX \quad (\text{by Prod 2}) \\
&\implies bXaXaaX \quad (\text{by Prod 2}) \\
&\implies bXaXaa \quad (\text{by Prod 2}) \\
&\implies bXabXaa \quad (\text{by Prod 2}) \\
&\implies bXabaXaa \quad (\text{by Prod 2}) \\
&\implies bbXabaXaa \quad (\text{by Prod 2}) \\
&\implies bbabaXaa \quad (\text{by Prod 2}) \\
&\implies bbabaaa \quad (\text{by Prod 2})
\end{aligned} \tag{3}$$

## 0.6

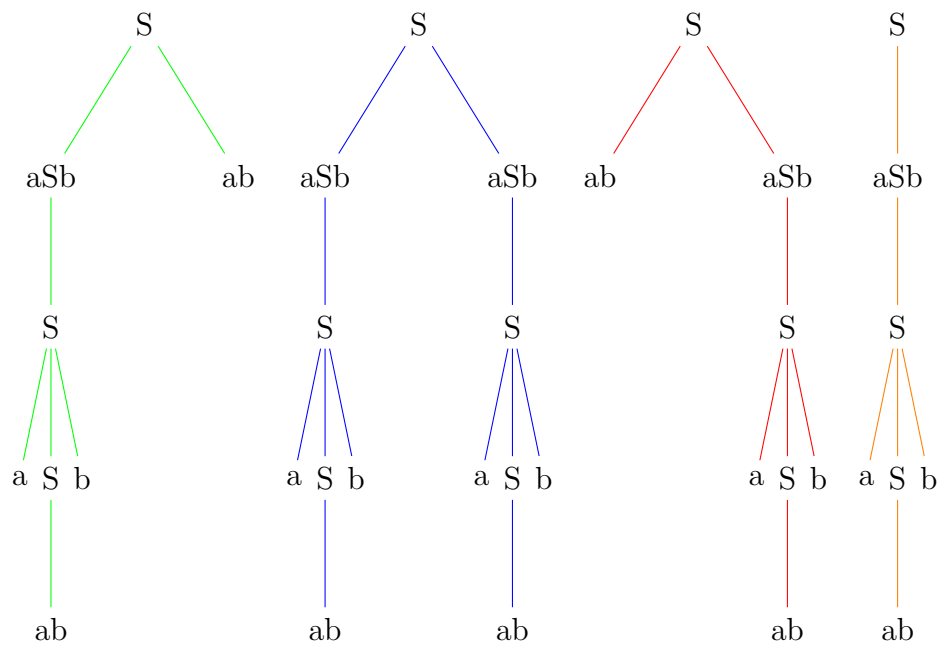
$$\begin{aligned}
&\textbf{Prod1} \quad S \rightarrow \mathbf{Xa} \\
&\textbf{Prod2} \quad X \rightarrow \mathbf{bbX} \mid \mathbf{bbS} \mid \mathbf{bb} \mid \Lambda
\end{aligned} \tag{4}$$

## 0.7

### 0.7.1

This CFG cannot match aaaa, abaa, or bbaa. The only string of length 4 is aabb:

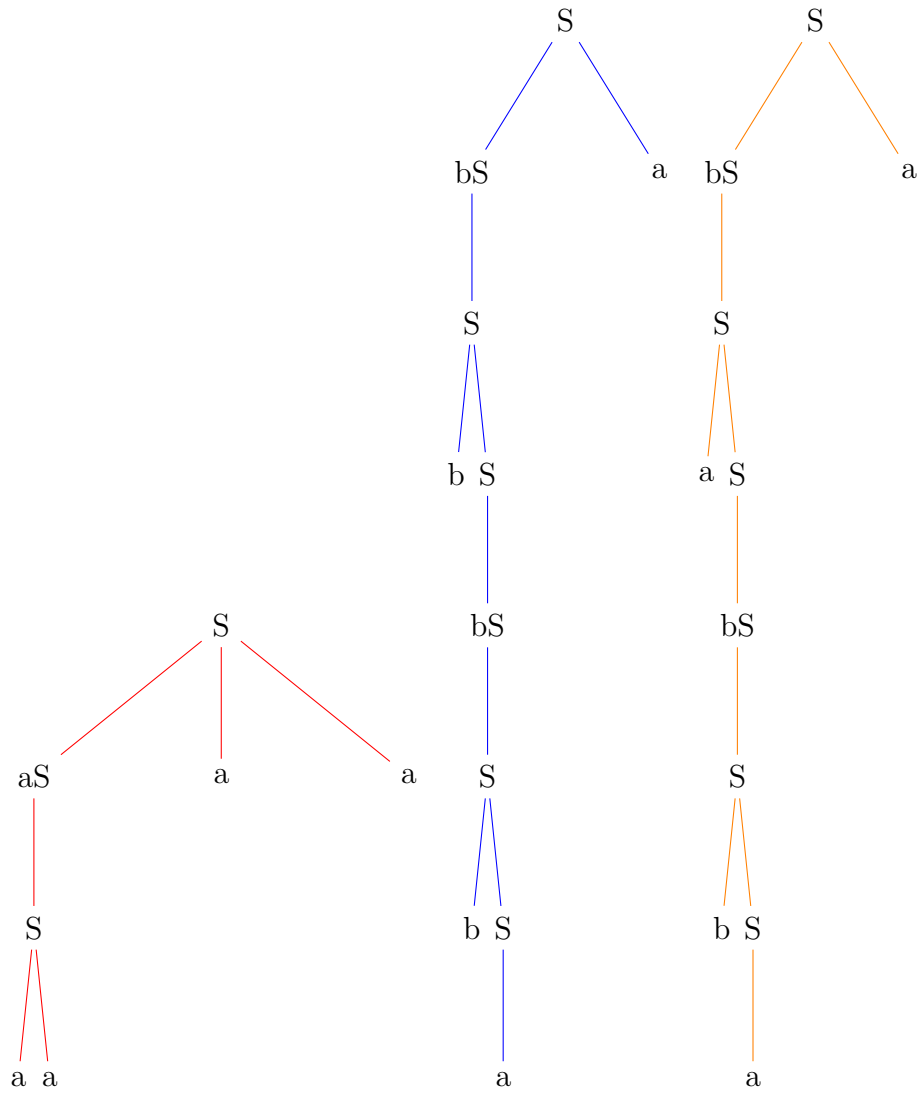
$$\text{Prod1} \quad S \rightarrow aSb \mid ab \quad (5)$$



## 0.7.2

For strings aaaa, abaa, or bbaa. This CFG can create the string aaaa, bbaa:

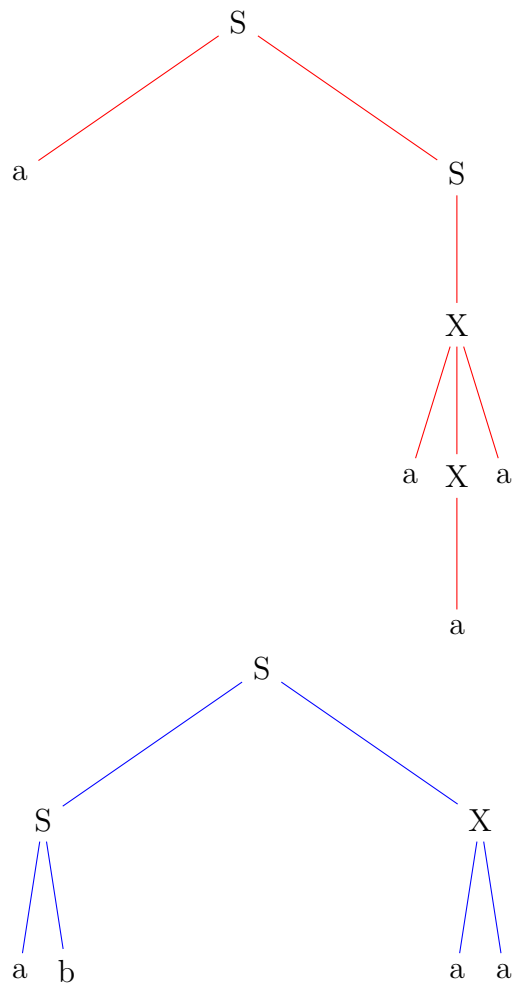
$$\text{Prod1} \quad S \rightarrow aS \mid bS \mid a \quad (6)$$



### 0.7.3

For strings aaaa, abaa, or bbaa. This CFG can create the string aaaa, abaa:

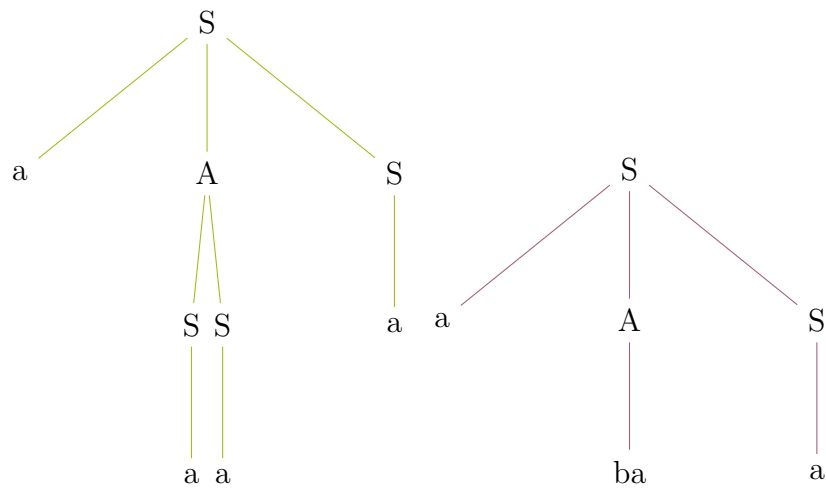
$$\begin{array}{ll} \text{Prod1} & S \rightarrow aS \mid aSb \mid X \\ \text{Prod2} & X \rightarrow aXa \mid a \end{array} \quad (7)$$



### 0.7.4

For strings aaaa, abaa, or bbaa. This CFG can create the string aaaa, abaa:

$$\begin{array}{ll} \text{Prod1} & S \rightarrow aAS \mid a \\ \text{Prod2} & A \rightarrow SbA \mid SS \mid ba \end{array} \quad (8)$$

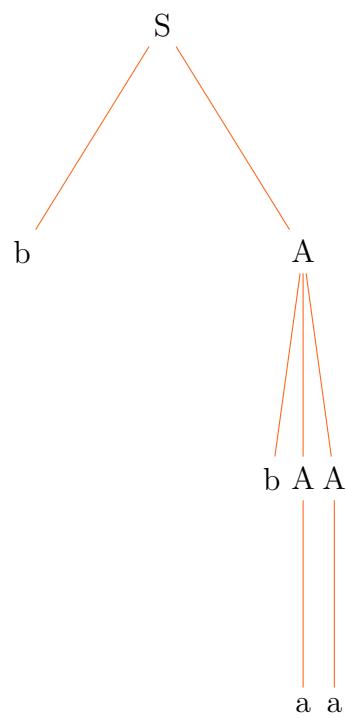




### 0.7.5

For strings aaaa, abaa, or bbaa. This CFG can create the string bbaa:

$$\begin{array}{ll} \text{Prod1} & S \rightarrow aB \mid bA \\ \text{Prod2} & A \rightarrow a \mid aS \mid bAA \\ \text{Prod3} & B \rightarrow b \mid bS \mid aBB \end{array} \quad (9)$$

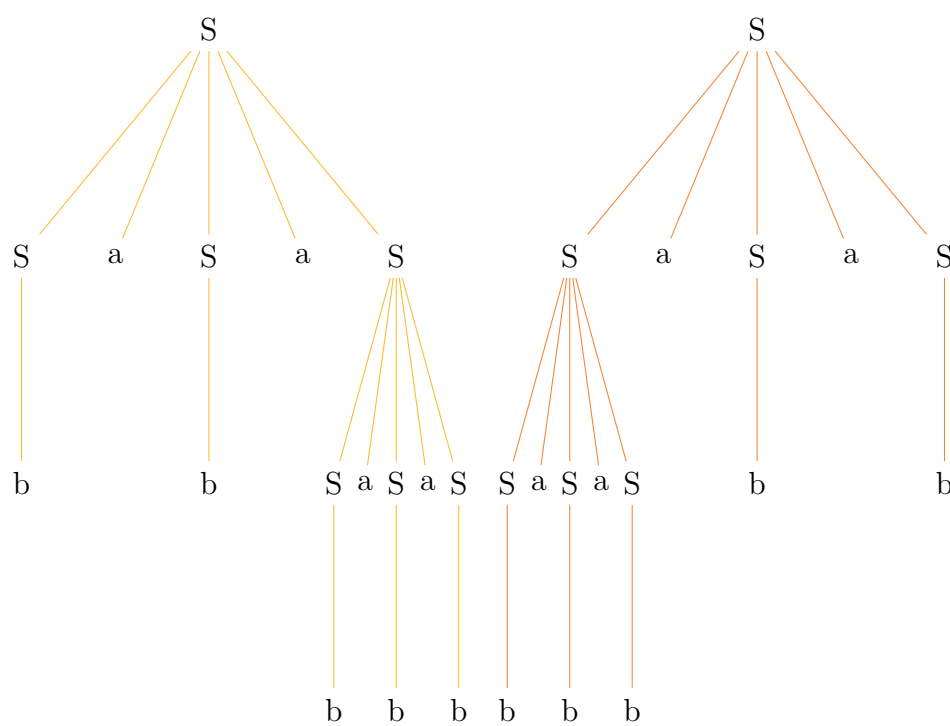


## 0.8

### 0.8.1

Ambiguous CFG with the string: babababab

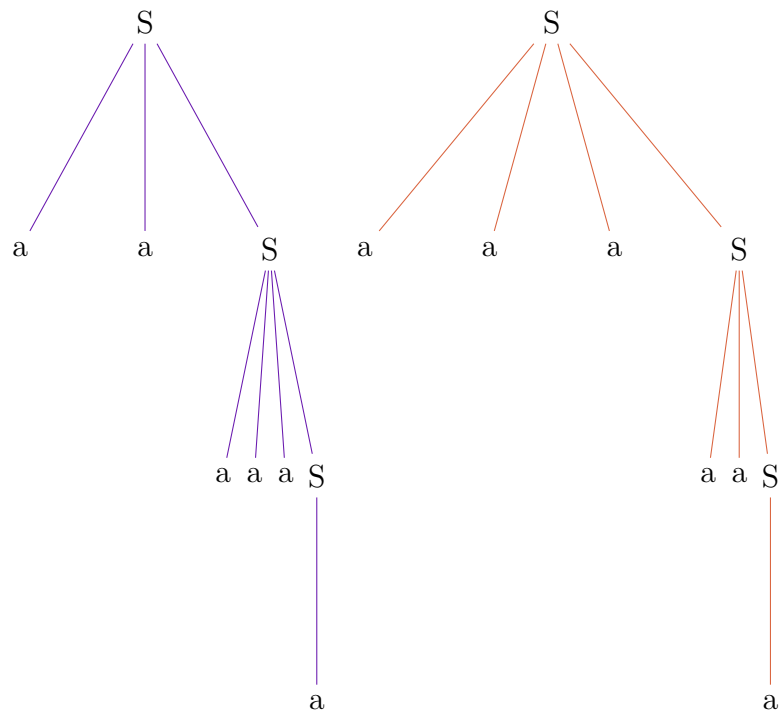
$$\text{Prod1} \quad \text{S} \rightarrow \text{SaSaS} \mid \text{b} \quad (10)$$



## 0.8.2

Ambiguous CFG with the string: aaaaaa

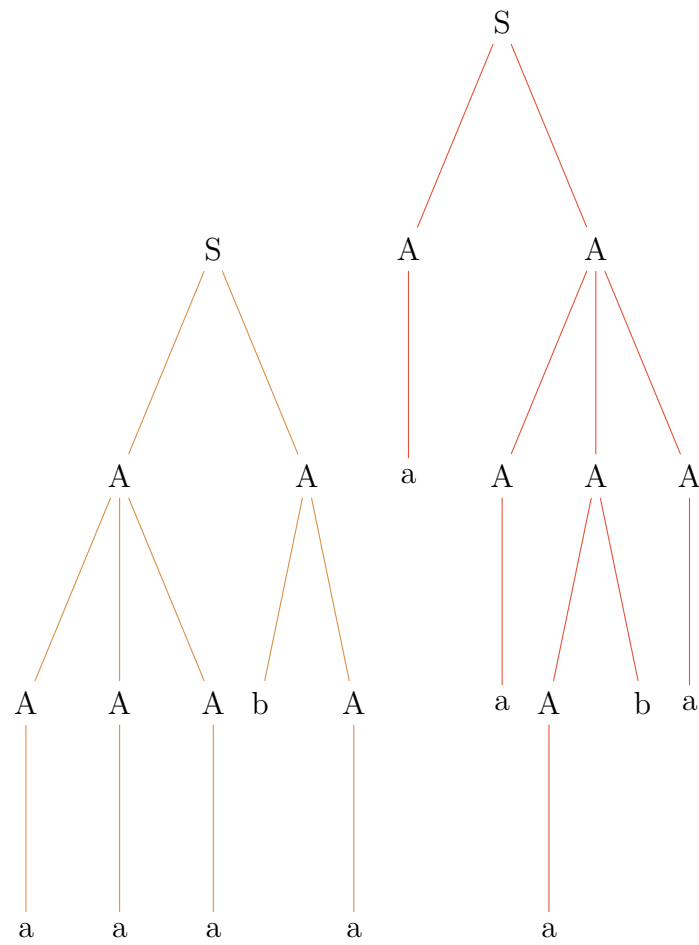
$$\text{Prod1} \quad S \rightarrow aaS \mid aaaS \mid a \quad (11)$$



### 0.8.3

Ambiguous CFG with the string: aaaba

$$\begin{array}{ll} \text{Prod1} & S \rightarrow AA \\ \text{Prod2} & A \rightarrow AAA \mid a \mid bA \mid Ab \end{array} \quad (12)$$



## 0.9

### 0.9.1

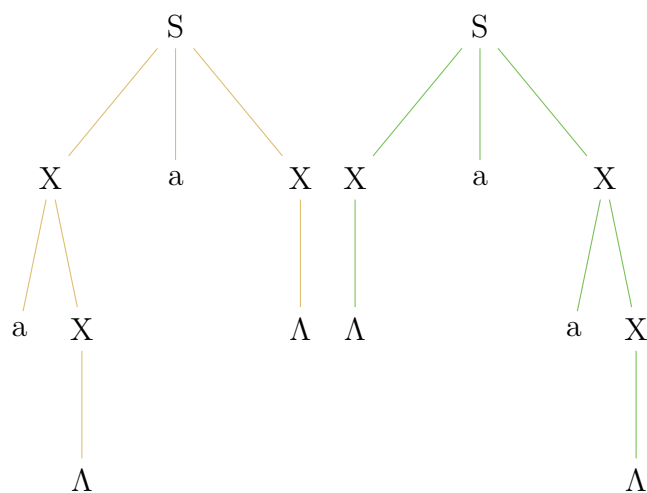
$$\begin{array}{ll} \text{Prod1} & S \rightarrow XaX \\ \text{Prod2} & X \rightarrow aX \mid bX \mid \Lambda \end{array} \quad (13)$$

Ambiguous

$$\begin{array}{l} S \Rightarrow XaX \\ X \Rightarrow aXaX \\ \quad \Rightarrow a\Lambda aX \\ \quad \Rightarrow aaX \\ \quad \Rightarrow aa\Lambda \\ \quad \Rightarrow aa \end{array} \quad (14)$$

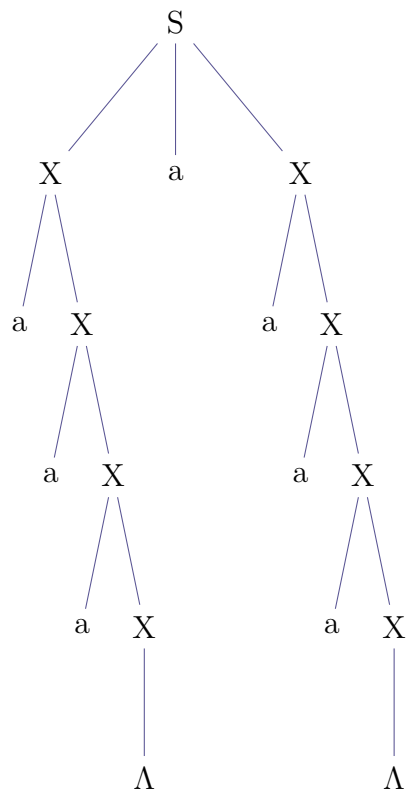
Second Example

$$\begin{array}{l} S \Rightarrow XaX \\ X \Rightarrow \Lambda aX \\ \quad \Rightarrow aaX \\ \quad \Rightarrow aa\Lambda \\ \quad \Rightarrow aa \end{array} \quad (15)$$



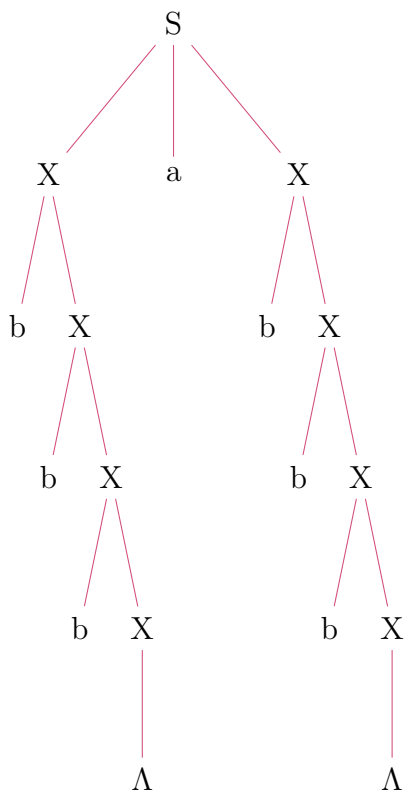
# Non-Ambiguous

$$\begin{aligned}
 S &\Rightarrow XaX \\
 X &\Rightarrow aXaX \\
 &\Rightarrow aaXaX \\
 &\Rightarrow aaaXaX \\
 &\Rightarrow aaaa\Lambda X \\
 &\Rightarrow aaaaaX \\
 &\Rightarrow aaaaaaX \\
 &\Rightarrow aaaaaaaX \\
 &\Rightarrow aaaaaaa\Lambda \\
 &\Rightarrow aaaaaaa
 \end{aligned}
 \tag{16}$$



Second Example

$$\begin{aligned}
 S &\Rightarrow XaX \\
 X &\Rightarrow bXaX \\
 &\Rightarrow bbXaX \\
 &\Rightarrow bbbXaX \\
 &\Rightarrow bbb\Lambda aX \\
 &\Rightarrow bbbbbX \\
 &\Rightarrow bbbbbbX \\
 &\Rightarrow bbbbbb\Lambda \\
 &\Rightarrow bbbbbb
 \end{aligned}
 \tag{17}$$



## 0.9.2

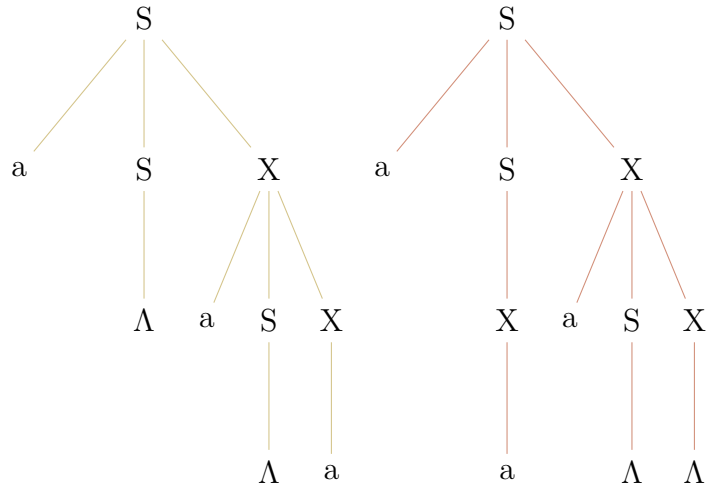
Ambiguous

$$\begin{array}{ll} \text{Prod1} & S \rightarrow aSX \mid \Lambda \\ \text{Prod2} & X \rightarrow aX \mid a \end{array} \quad (18)$$

$$\begin{array}{l} S \Rightarrow aSX \\ \Rightarrow a\Lambda X \\ \Rightarrow aX \\ \Rightarrow aaSX \\ \Rightarrow aa\Lambda X \\ \Rightarrow aaa \end{array} \quad (19)$$

Second Example

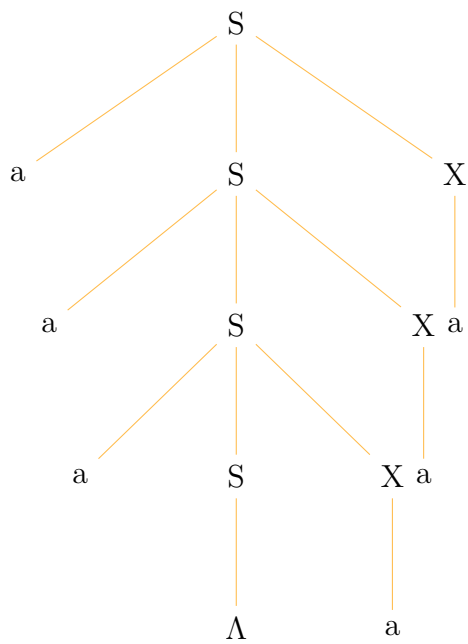
$$\begin{array}{l} S \Rightarrow aSX \\ \Rightarrow aXX \\ \Rightarrow aaX \\ \Rightarrow aaaSX \\ \Rightarrow aaa\Lambda X \\ \Rightarrow aaa\Lambda \end{array} \quad (20)$$





**Non-Ambiguous**

$$\begin{aligned}
 S &\Rightarrow aSX \\
 &\Rightarrow aaSXX \\
 &\Rightarrow aaaSXXa \\
 &\Rightarrow aaa\Lambda Xaa \\
 &\Rightarrow aaaaaa
 \end{aligned}
 \tag{21}$$



### 0.9.3

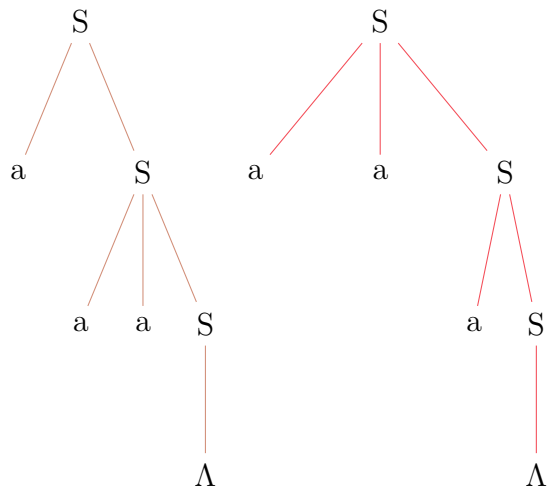
$$\text{Prod1} \quad S \rightarrow aS \mid bS \mid aaS \mid \Lambda \quad (22)$$

Ambiguous

$$\begin{aligned} S &\Rightarrow aS \\ &\Rightarrow aaS \\ &\Rightarrow aaaS \\ &\Rightarrow aaa\Lambda \\ &\Rightarrow aaa \end{aligned} \quad (23)$$

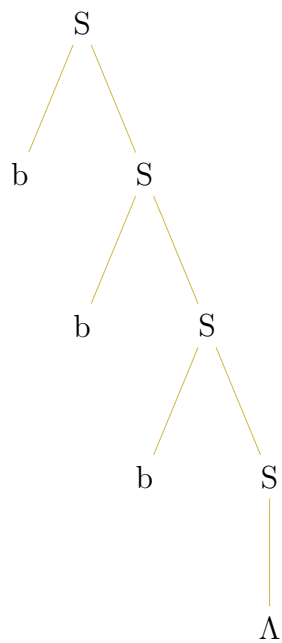
Second Example

$$\begin{aligned} S &\Rightarrow aaS \\ &\Rightarrow aaaS \\ &\Rightarrow aaa\Lambda \\ &\Rightarrow aaa \end{aligned} \quad (24)$$



## Non-Ambiguous

$$\begin{aligned} S &\Rightarrow bS \\ &\Rightarrow bbS \\ &\Rightarrow bbbS \\ &\Rightarrow bbb\Lambda \\ &\Rightarrow bbb \end{aligned} \tag{25}$$



## 0.10

### 0.10.1

$$\text{Prod1} \quad S \rightarrow aS \mid bS \mid a \quad (26)$$

### 0.10.2

$$\begin{array}{ll} \text{Prod1} & S \rightarrow aSb \mid bX \\ \text{Prod2} & X \rightarrow bX \mid b \end{array} \quad (27)$$

## 0.11

The CFG's generated by the Regular Languages over the alphabet  $\Sigma = \{ab\}$ :

### 0.11.1

The language defined by  $(aaa + b)^*$ :

### 0.11.2

The language defined by  $(a + b)^*(bbb + aaa)(a + b)^*$ :

### 0.11.3

All the strings that end in b and have an even number of b's in total:

### 0.11.4

The set of all strings of odd length:

### 0.11.5

All strings with exactly one a or exactly one b:

### 0.11.6

All strings with an odd number of a's or and even number of b's: