

CS 3133: Homework 2

DID NOT FINISH

MISREAD DUE DATE

Adam Camilli

September 11, 2017

1. **3.2** (page 97) Let G be the grammar

$$S \rightarrow ASB \mid \lambda$$

$$A \rightarrow aAb \mid \lambda$$

$$B \rightarrow bBa \mid ba$$

.

- (a) Give a leftmost derivation of $aabbba$.

$$\begin{aligned} S &\Rightarrow ASB \\ &\Rightarrow aAbSB \\ &\Rightarrow aaAbbSB \\ &\Rightarrow aabbSB \\ &\Rightarrow aabbB \\ &\Rightarrow aabbba \end{aligned}$$

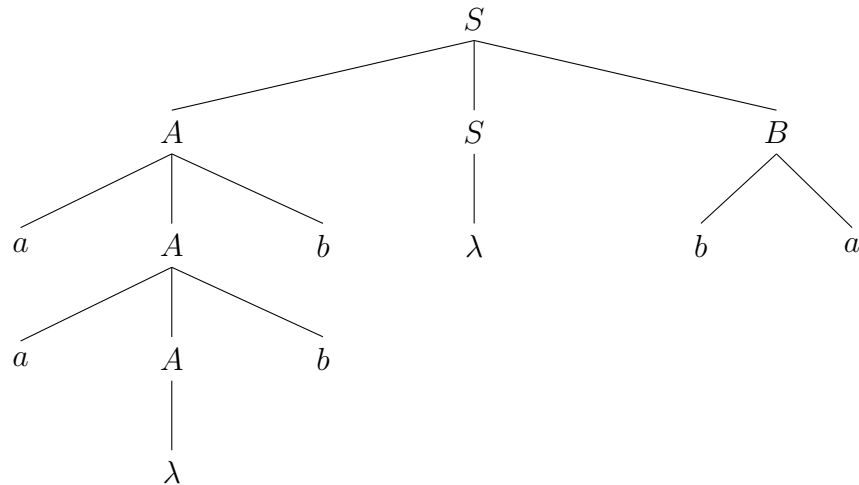
$$\begin{aligned} S &\rightarrow ASB \\ A &\rightarrow aAb \\ A &\rightarrow aAb \\ A &\rightarrow \lambda \\ S &\rightarrow \lambda \\ B &\rightarrow ba \end{aligned}$$

(b) Give a rightmost derivation of $abaabbbabbaa$.

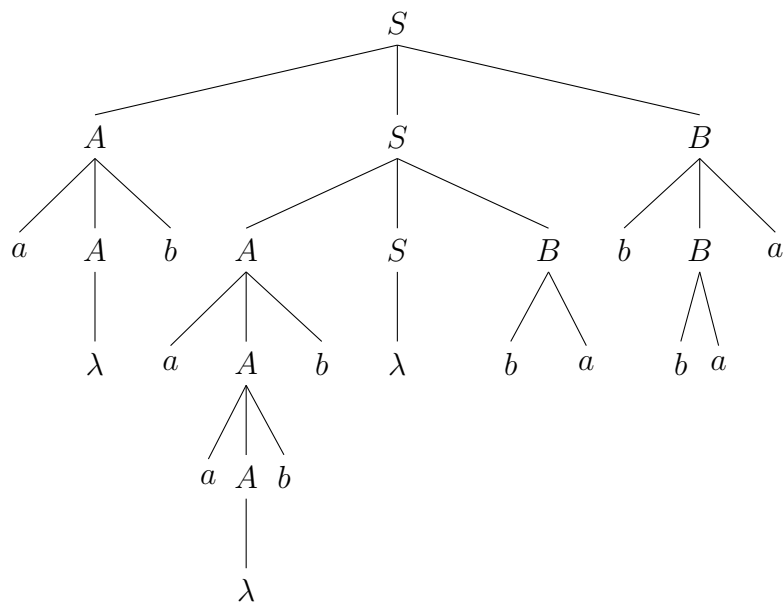
$S \Rightarrow ASB$	$S \rightarrow ASB$
$\Rightarrow ASbBa$	$B \rightarrow bBa$
$\Rightarrow ASbbaa$	$B \rightarrow ba$
$\Rightarrow AASBbbaa$	$S \rightarrow ASB$
$\Rightarrow AASbabbbaa$	$B \rightarrow ba$
$\Rightarrow AAbabbbaa$	$S \rightarrow \lambda$
$\Rightarrow AaAbbabbaa$	$A \rightarrow aAb$
$\Rightarrow AaaAbbbabbaa$	$A \rightarrow aAb$
$\Rightarrow Aaabbbabbaa$	$A \rightarrow \lambda$
$\Rightarrow aAaabbbabbaa$	$A \rightarrow aAb$
$\Rightarrow abaabbbabbaa$	$A \rightarrow \lambda$

(c) Build a derivation tree for the derivations in parts (a) and (b).

Part (a):



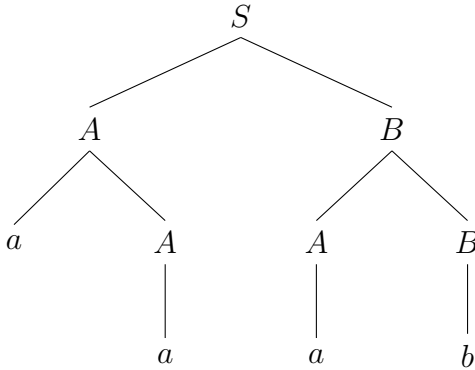
Part (b):



(d) Use set notation to define $L(G)$.

$$L(G) = \left\{ \left((ab)^x (ba)^y \right)^z \mid x \geq 0, y > 0, z > 0 \right\}$$

2. **3.4** (page 98) Let DT be the derivation tree



- (a) Give a leftmost derivation that generates the tree DT.
 - (b) Give a rightmost derivation that generates the tree DT.
 - (c) How many different derivations are there that generate DT?
3. **2.39d** (page 61) Use the regular expressions in Table 2.1 to establish the following identity: $(a \cup b)^* = (a^* \cup ba^*)^*$:

To get desired result, we can use identity 12. from Table 2.1, defined as

$$\begin{aligned}
 (a \cup b)^* &= (a^* \cup b)^* \\
 &= a^*(a \cup b)^* \\
 &= (a \cup ba^*)^* \\
 &= (a^*b^*)^* \\
 &= a^*(ba^*)^* \\
 &= (a^*b)^*a^*
 \end{aligned}$$

Take the intermediate result $(a \cup ba^*)^*$, and, using $u = a$ and $v = ba^*$, reapply the first parts of identity 12. to obtain desired identity:

$$\begin{aligned}
 (u \cup v)^* &= (u^* \cup v)^* && \text{(Step 1 of Identity 12.)} \\
 &= (a^* \cup ba^*)^* && \text{(Using } u = a \text{ and } v = ba^*)
 \end{aligned}$$

4. Let Σ be an alphabet, and u, v, w regular expressions over Σ . Are the following regular expression identities true?

(a) $u \cup (vw) = (u \cup v)(u \cup w)$

No. The lefthand expression contains v and w only in the combined form (vw) , while the righthand expression allows singular v and w . Here is one of many counterexamples that can therefore be formed:

Let alphabet $\Sigma = \{a, b, c\}$, and regular expressions $u = a, v = b, w = c$:

$$a \cup (bc) \not\supset ac$$

$$(a \cup b)(a \cup c) \supset ac$$

Therefore, $u \cup (vw) \neq (u \cup v)(u \cup w)$.

(b) $u^*(v \cup w) = u^*v \cup u^*w$

Yes. This is merely an application of the distributive regular expression identity

$$u \cup (v \cup w) = uv \cup uw$$

with $u = u^*$, which, since u, v , and w can represent any regular expression, must be valid.

5. **1.44** (page 39) Give a recursive definition of the set of ancestors of a node x in a tree.

Basis: The parent node p of node x is an ancestor of x : $p \in A(x)$

Recursive Step: An ancestor of p is an ancestor of x : $A(p) \in A(x)$

Closure: A node a is an ancestor of x if it is obtainable from x with finitely many applications of the recursive step.

6. **3.1** (page 97) Let G be the grammar

$$S \rightarrow abSc \mid A$$

$$A \rightarrow cAd \mid cd.$$

(a) Give a derivation of $ababccddcc$.

Defining the productions as

$$1. S \rightarrow abSc$$

$$2. S \rightarrow A$$

$$3. A \rightarrow cAd$$

$$4. A \rightarrow cd$$

we can derive it like so:

$$S \Rightarrow abSc \quad (1.)$$

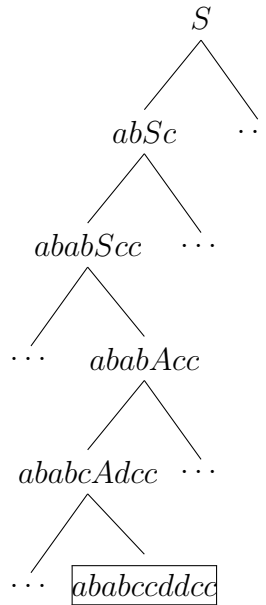
$$\Rightarrow ababScc \quad (1.)$$

$$\Rightarrow ababAcc \quad (2.)$$

$$\Rightarrow ababcAdcc \quad (3.)$$

$$\Rightarrow ababccddcc \quad (4.)$$

(b) Build the derivation tree for the derivation in part (a).



(c) Use set notation to define $L(G)$.

$$L(G) = \left\{ \{ab\}^n \{c\}^m cd \{d\}^m \{c\}^n \mid n \geq 1, m \geq 0 \right\}$$

Note that n and m , respectively, are essentially the number of times that recursive productions $S \rightarrow abSc$ and $A \rightarrow cAd$ are applied.