Chris Fenton
CNC - Formal Languages
Homework 4

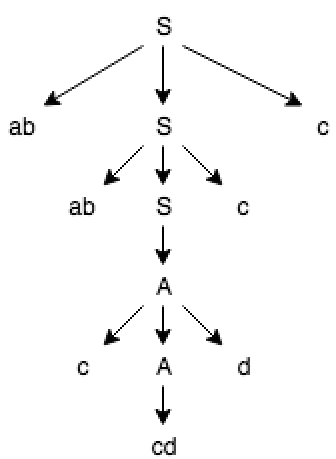
1. Let G be the grammar

S -> abSc | A
A -> cAd | cd

a. Give a derivation of *ababccddcc*.

Derivation		**Rule Applied**
S =>	abSc	S -> abSc
=>	ababScc	S -> abSc
=>	ababAcc	S -> A
=>	ababcAdcc	A -> cAd
=>	ababccddcc	A -> cd

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



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

$L(G) = \{ (ab)^n c^m d^m c^n \mid n \geq 0, m > 0 \}$

2. Let G be the grammar

S -> ASB | λ
A -> aAb | λ
B -> bBa | ba

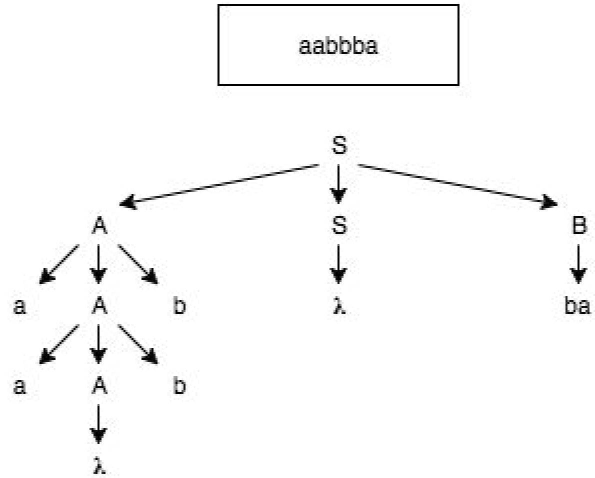
a. Give a leftmost derivation of *aabbba*.

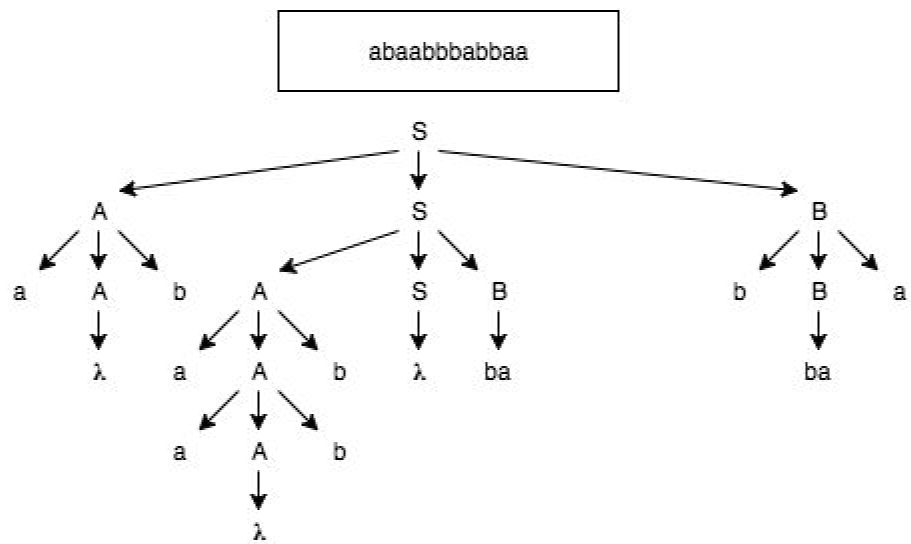
Derivation		Rule Applied
S =>	ASB	S -> ASB
=>	aAbSB	A -> aAb
=>	aaAbbSB	A -> aAb
=>	aabbSB	A -> λ
=>	aabbB	S -> λ
=>	aabbba	B -> ba

b. Give a rightmost derivation of *abaabbbabbaa*.

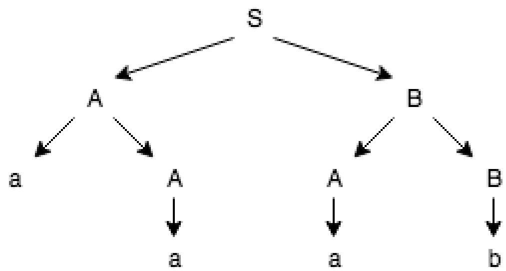
Derivation		Rule Applied
S =>	ASB	S -> ASB
=>	ASbBa	B -> bBa
=>	ASbbaa	B -> ba
=>	AASBbbaa	S -> ASB
=>	AASbabbaa	B -> ba
=>	AAbabbaa	S -> λ
=>	AaAbbabbaa	A -> aAb
=>	AaaAbbbabbaa	A -> aAb
=>	Aaabbbabbaa	A -> λ
=>	aAbaabbbabbaa	A -> aAb
=>	abaabbbabbaa	A -> λ

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





4. Let DT be the derivation tree



a. Give a leftmost derivation that generates the tree DT.

Derivation		Rule Applied
S =>	AB	S -> AB
=>	aAB	A -> aA
=>	aaB	A -> a
=>	aaAB	B -> AB
=>	aaaB	A -> a
=>	aaab	B -> b

b. Give a rightmost derivation that generates the tree DT.

Derivation		Rule Applied
S =>	AB	S -> AB
=>	AAB	B -> AB
=>	AAb	B -> b
=>	Aab	A -> a
=>	aAab	A -> aA
=>	aaab	A -> a

12. Construct a grammar over { *a*, *b* } whose language contains precisely the strings with the same number of *a*'s and *b*'s.

S -> A | λ
A -> aAb | bAa | λ

15. Give a regular grammar that generates the described language: The set of strings over {*a*, *b*, *c*} in which all the *a*'s precede the *b*'s, which in turn precede the *c*'s. It is possible that there are no *a*'s, *b*'s, or *c*'s.

S -> A | λ
A -> aB | λ
B -> bC | λ
C -> c | λ

21. Give a regular grammar that generates the described language: The set of strings over {*a*, *b*} that do not contain the substring *aba*.

I've got the regular expression b*(a*bbb*)*a*b*, but I can't figure out where to go from there to convert this to the corresponding regular grammar (even though I know it has one!).

25. Give a regular grammar that generates the described language: The set of strings over {*a*, *b*} with an even number of *a*'s or an odd number of *b*'s.