

Problem of the DAY

Create a regular context-free grammar that generates $L = \{w \in \{a,b\}^* : \text{the number of } a\text{'s in } w \text{ is not divisible by } 3\}$

Hint: start by designing a DFA first then get the grammar from the DFA.

Recall: A regular context-free grammar can have at most one non-terminal on the RHS of each rule, if there is a non-terminal, it is the last symbol on the RHS of the rule.

No tutorials this week (Oct. 11/12).
Assignment #3: Due Fri. Oct. 21.

For Oct. 18/19: Some practice questions for
Tutorial #5 have been posted.

Old midterms are available on the CSC 320 web
page.

I'm away at a conference Oct. 19-21: No office
hours. Ask assignment questions early.

Our midterm is **Wed. Oct. 26** in class.

Midterm tutorial: Mon. Oct. 24, ECS 116.

Create a NDFA which accepts the language generated by this context-free grammar. Start symbol: S

$$S \rightarrow aa S$$

$$M \rightarrow E$$

$$S \rightarrow \varepsilon$$

$$E \rightarrow aa$$

$$S \rightarrow M$$

$$M \rightarrow bbb$$

$$M \rightarrow ab M$$

$$M \rightarrow b S$$

Side note: I am using a more general definition of a NDFFA here which permits transitions on strings. That is,

Δ is a relation which is a finite subset of

K	$\times$	Σ^*	$\times$	K
current		string		next
state				state

Theorem: If L has a regular context-free grammar, then L is a regular language.

Proof: A NDFA can be constructed that accepts the language that the regular context-free grammar generates.

Note: this is in the text as Exercise 3.1.10.

Given the regular context-free grammar $G=(V, \Sigma, R, S)$ construct a NDFA

$M = (K, \Sigma, \Delta, s, F)$ where

$K = (V - \Sigma) \cup \{f\}$, $s = S$, $F = \{f\}$

For each rule $T \rightarrow u R$ with $u \in \Sigma^*$, $R \in V - \Sigma$,
add a transition (T, u, R) to Δ .

For each rule $T \rightarrow u$ with u in Σ^* ,
add a transition (T, u, f) to Δ .

$$L = \{ a^n b^p : n \leq p \leq 3n, n, p \geq 0 \}$$

Start symbol S .

$$S \rightarrow a S b$$

$$S \rightarrow \varepsilon$$

$$S \rightarrow a S bb$$

$$S \rightarrow a S bbb$$

This works because any integer p can be expressed as:

$$p = r + 2(n - r) \quad \text{when } n \leq p \leq 2n, \text{ and}$$

$$p = 2r + 3(n - r) \quad \text{when } 2n \leq p \leq 3n.$$

Parse Trees:

Definition of parse trees.

Parse trees are created when programs are compiled to aid in checking syntax and determining semantics of a program.

Leftmost derivations.

Example of how parse trees are used to prove the pumping theorem for context-free languages.

$L = \{w \text{ in } \{a, b\}^* : w \text{ has the same number of } a\text{'s and } b\text{'s}\}$

Start symbol S

$S \rightarrow e$

$A \rightarrow a$

$B \rightarrow b$

$S \rightarrow a B$

$A \rightarrow a S$

$B \rightarrow b S$

$S \rightarrow b A$

$A \rightarrow b A A$

$B \rightarrow a B B$

S - generates strings with $\#a\text{'s} = \#b\text{'s}$

A - means need one "a"

B - means need one "b"

Three derivations for aabbab:

1. $\underline{S} \Rightarrow \underline{aB} \Rightarrow a\underline{aBB} \Rightarrow aab\underline{S}B \Rightarrow aab\underline{bAB} \Rightarrow aabb\underline{aB} \Rightarrow aabbab$
2. $\underline{S} \Rightarrow \underline{aB} \Rightarrow a\underline{aBB} \Rightarrow aab\underline{B} \Rightarrow aab\underline{bS} \Rightarrow aabb\underline{aB} \Rightarrow aabbab$
3. $\underline{S} \Rightarrow \underline{aB} \Rightarrow a\underline{aBB} \Rightarrow aaB\underline{bS} \Rightarrow aaBb\underline{aB} \Rightarrow aaBba\underline{b} \Rightarrow aabbab$

Leftmost derivation: leftmost non-terminal replaced at each step.

Which ones are leftmost?

Two of the derivations are essentially the same- they correspond to the same leftmost derivation. The other one is different.

A context-free grammar is **ambiguous** if there is a string in the language which has two or more distinct parse trees (or equivalently, two or more distinct leftmost derivations).

Parse Trees- $G = (V, \Sigma, R, S)$

Nodes of parse trees are each labelled with one symbol from $V \cup \{\epsilon\}$.

Node at top- **root**, labelled with start symbol S .

Leaves- nodes at the bottom.

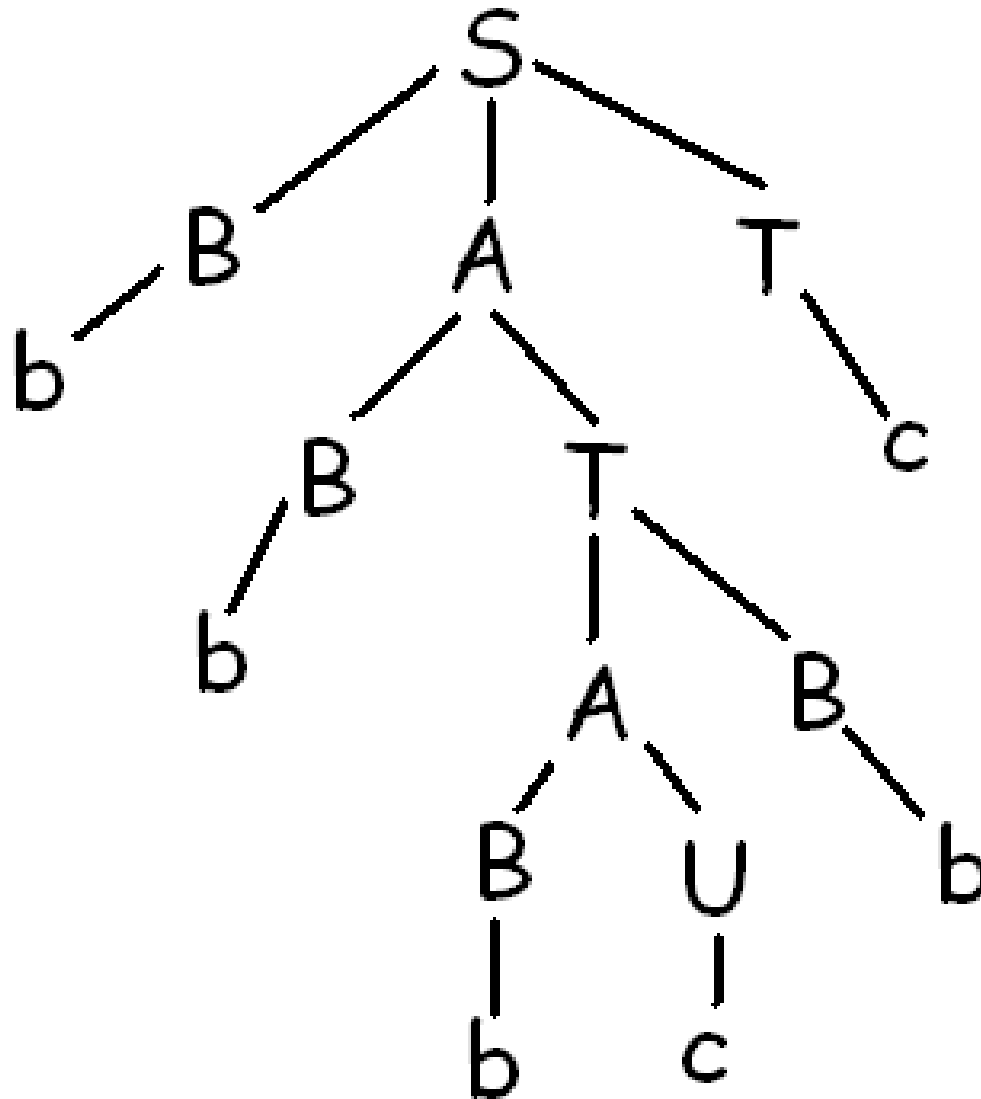
Yield- concatenate together the labels on the leaves going from left to right.

Leftmost derivation: always replace the leftmost non-terminal at each step.

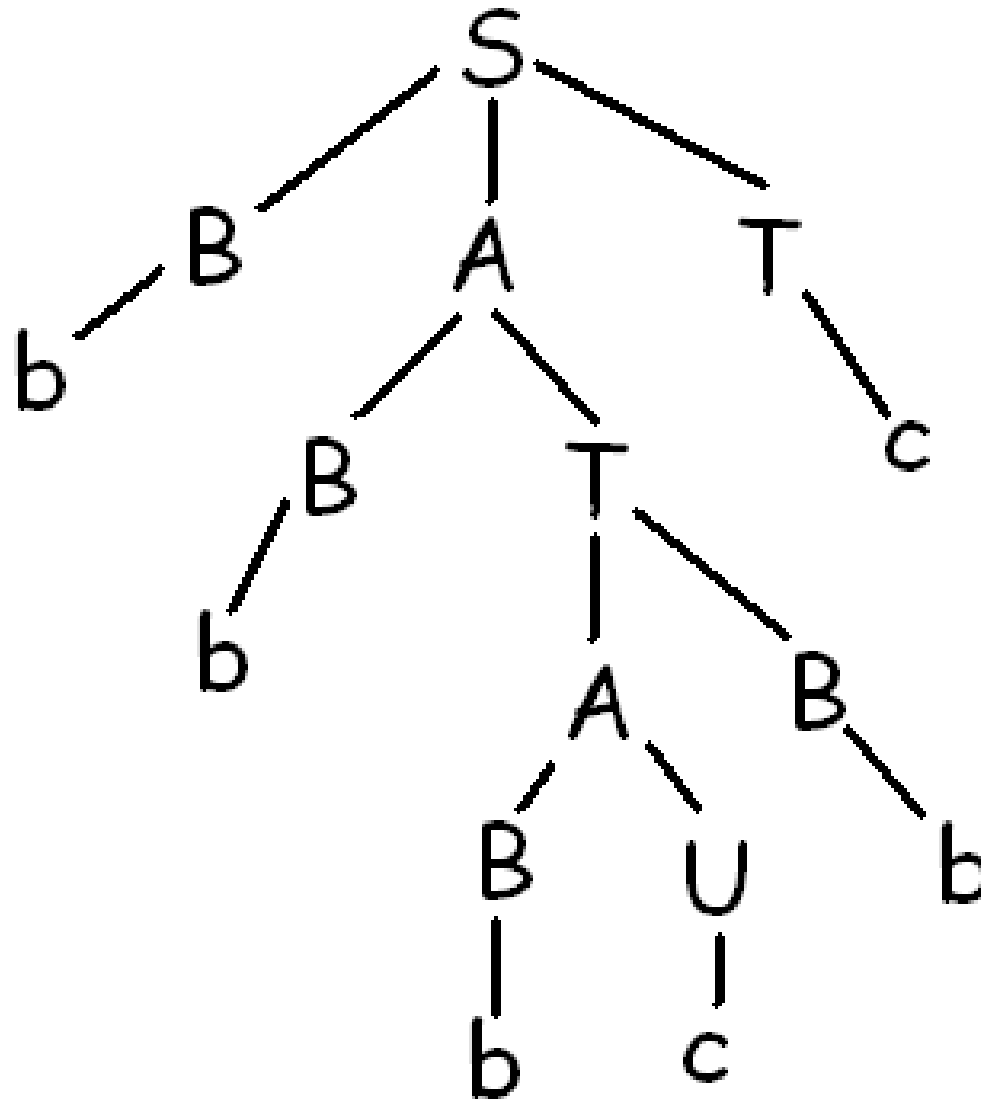
Theorem: Every derivation in a grammar G corresponds to a unique leftmost derivation.

Proof:

Construct the parse tree and determine the leftmost derivation from it.

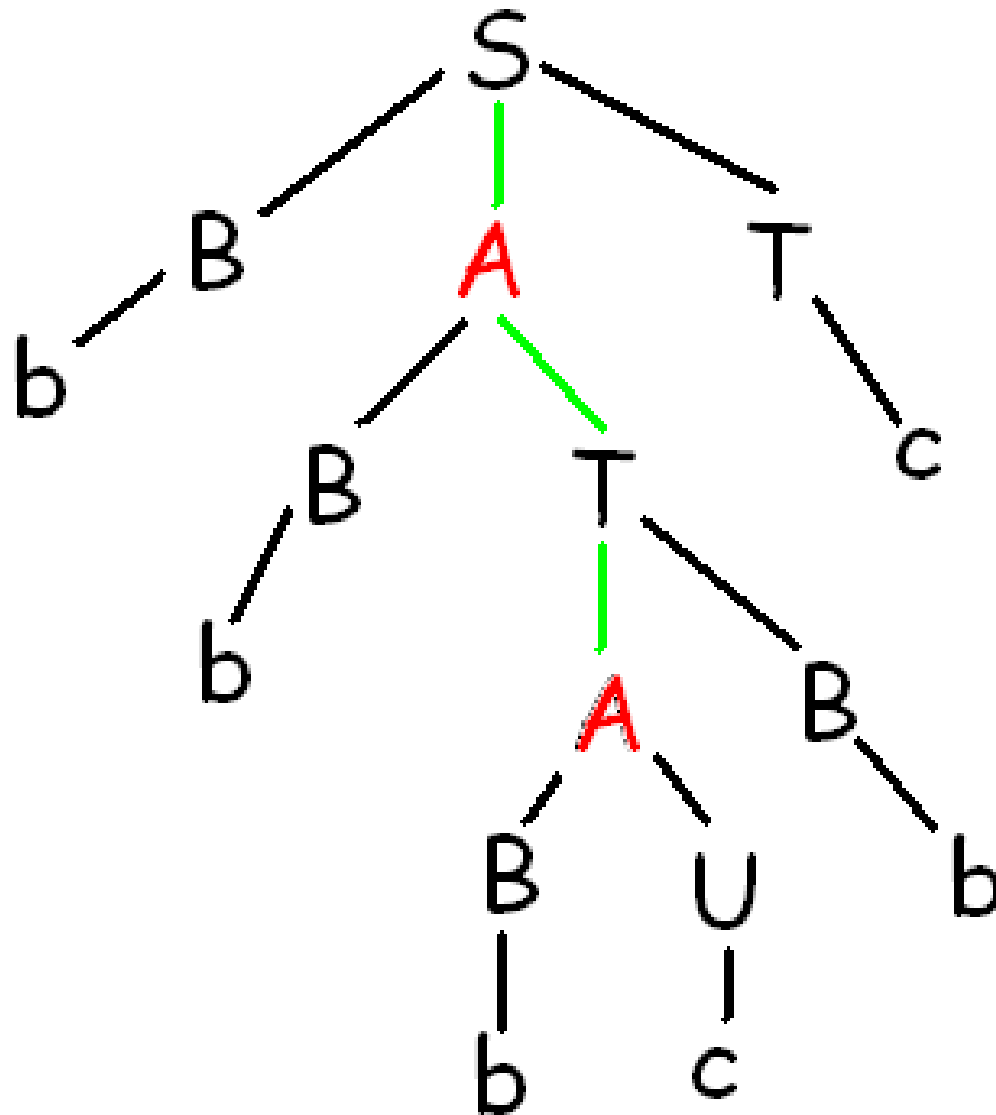


Find the
leftmost
derivation
for this
parse tree.



To pump:

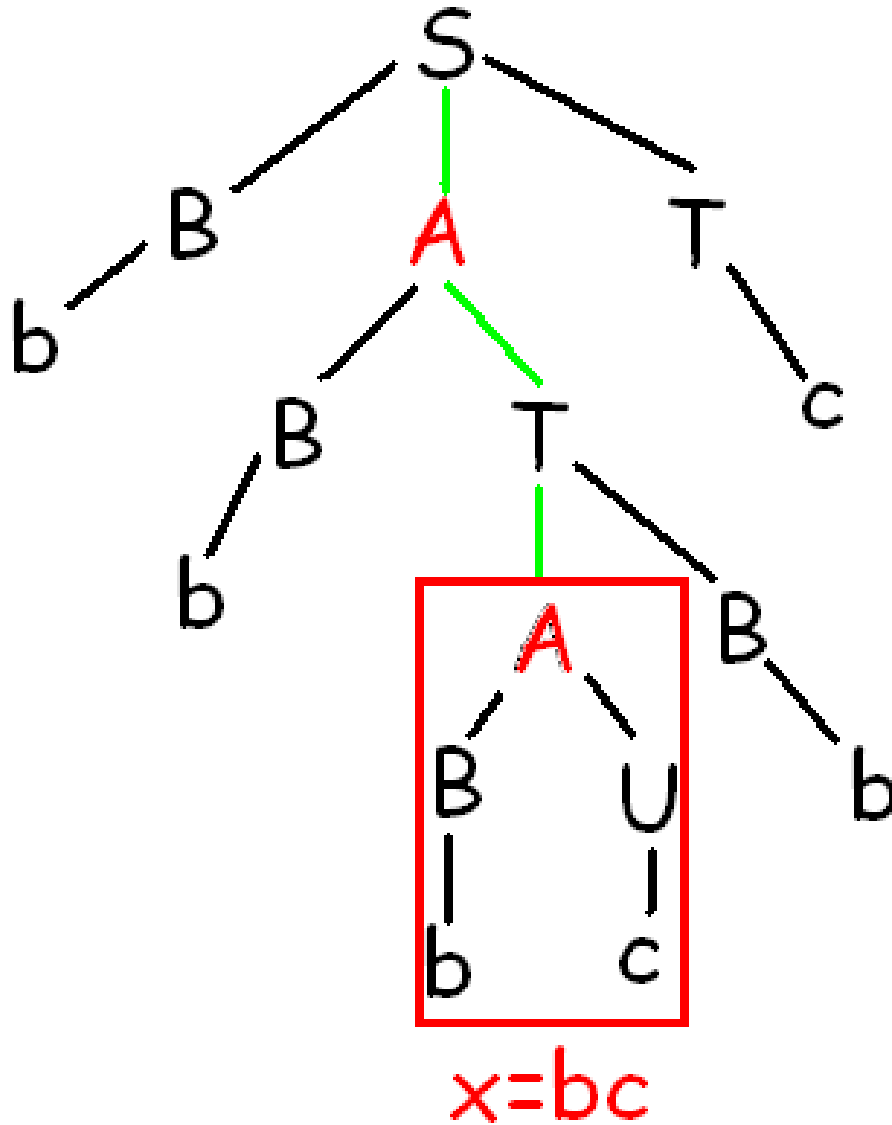
Look for a path from root with a repeated non-terminal.



Path:

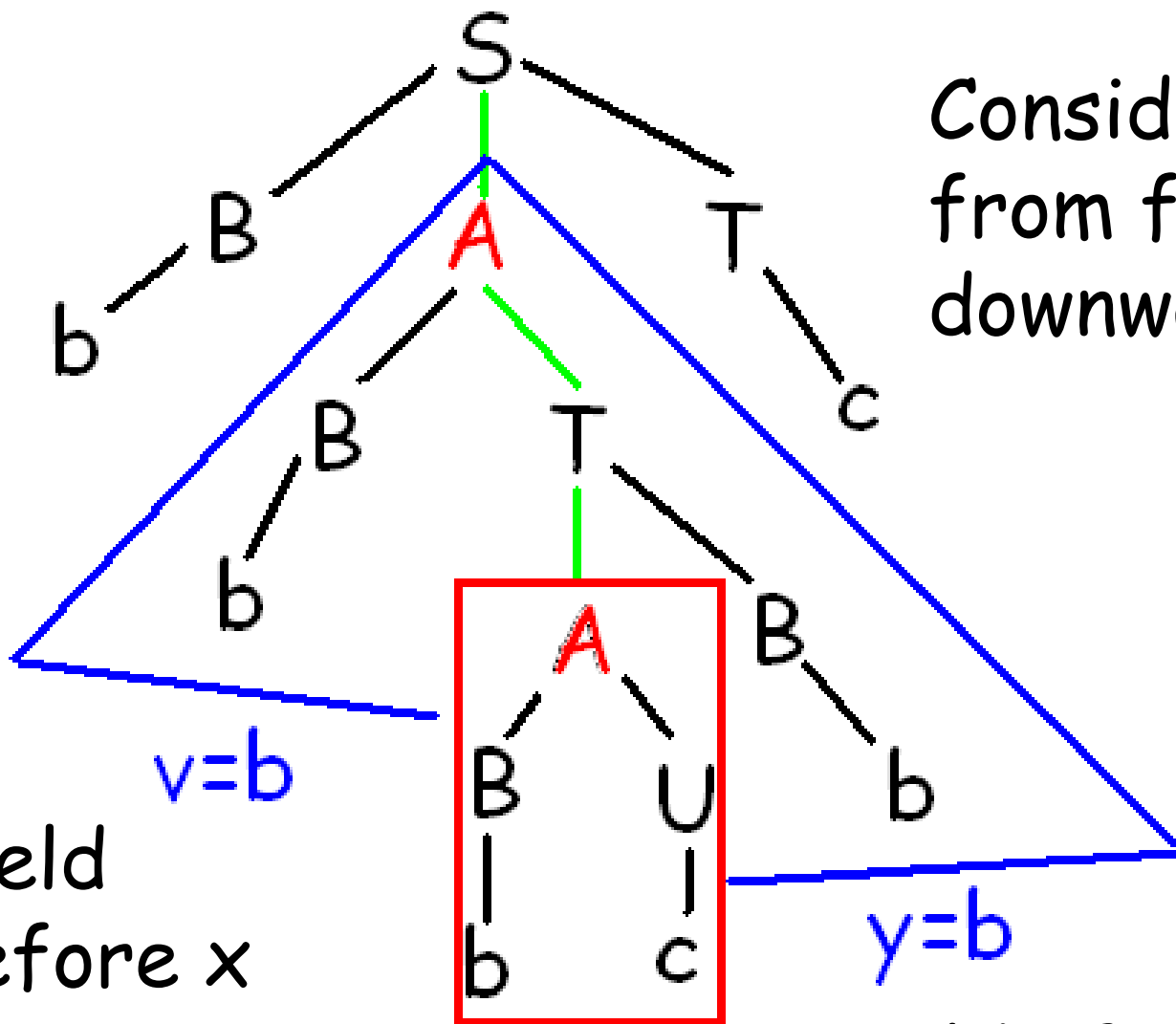
S - A - T - A

Non-terminal **A**
is repeated.



Yield from
second copy
downwards
gives x .

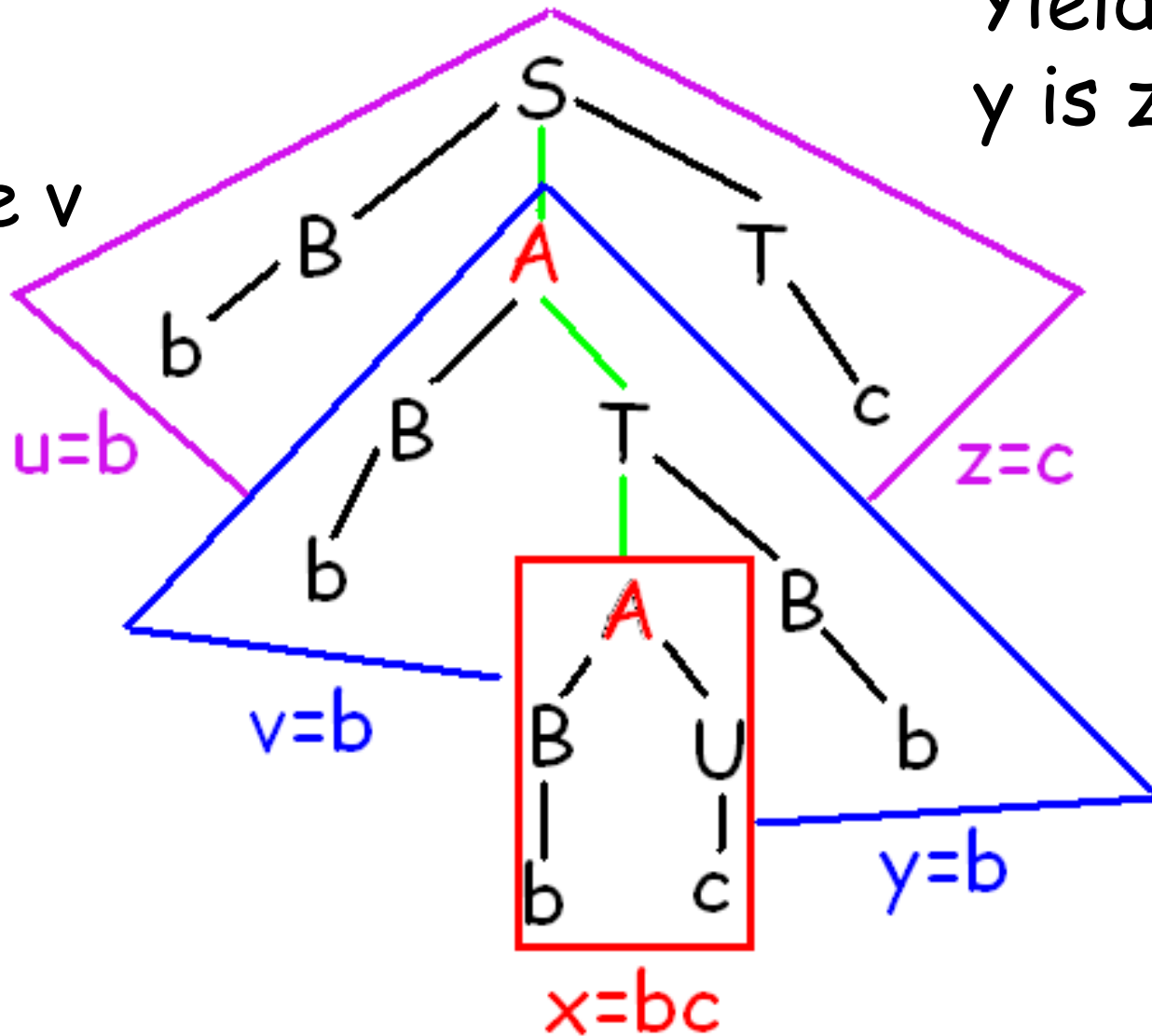
Consider tree
from first copy
downwards.



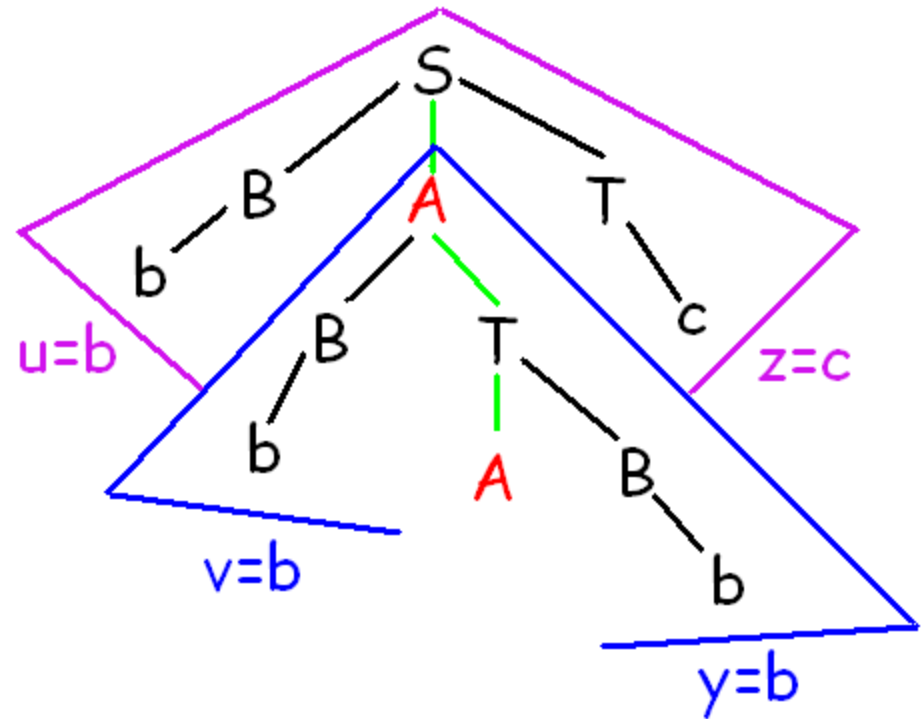
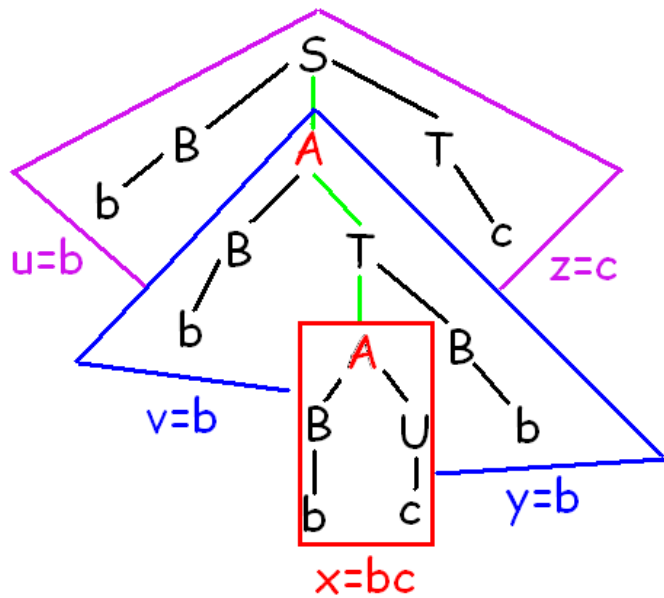
Yield
before x
is v .

Yield after
 x is y .

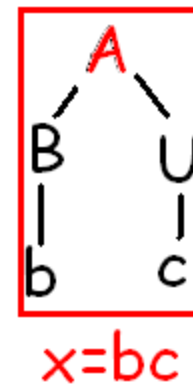
Yield of
whole
tree
before v
is u .

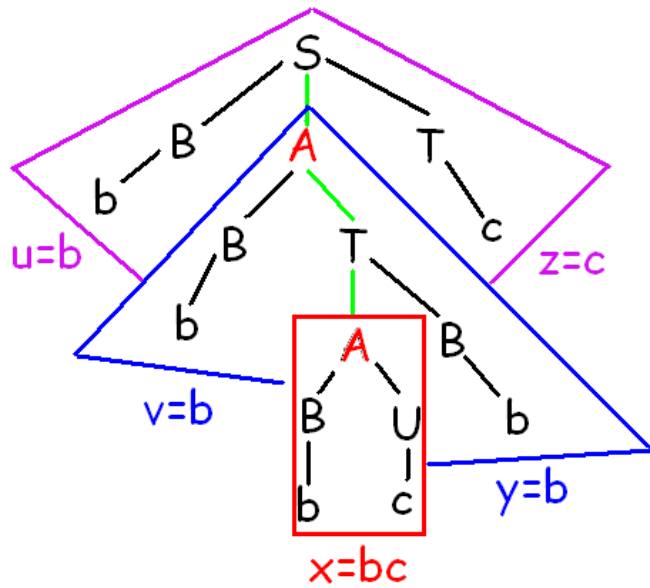


Yield after
 y is z .

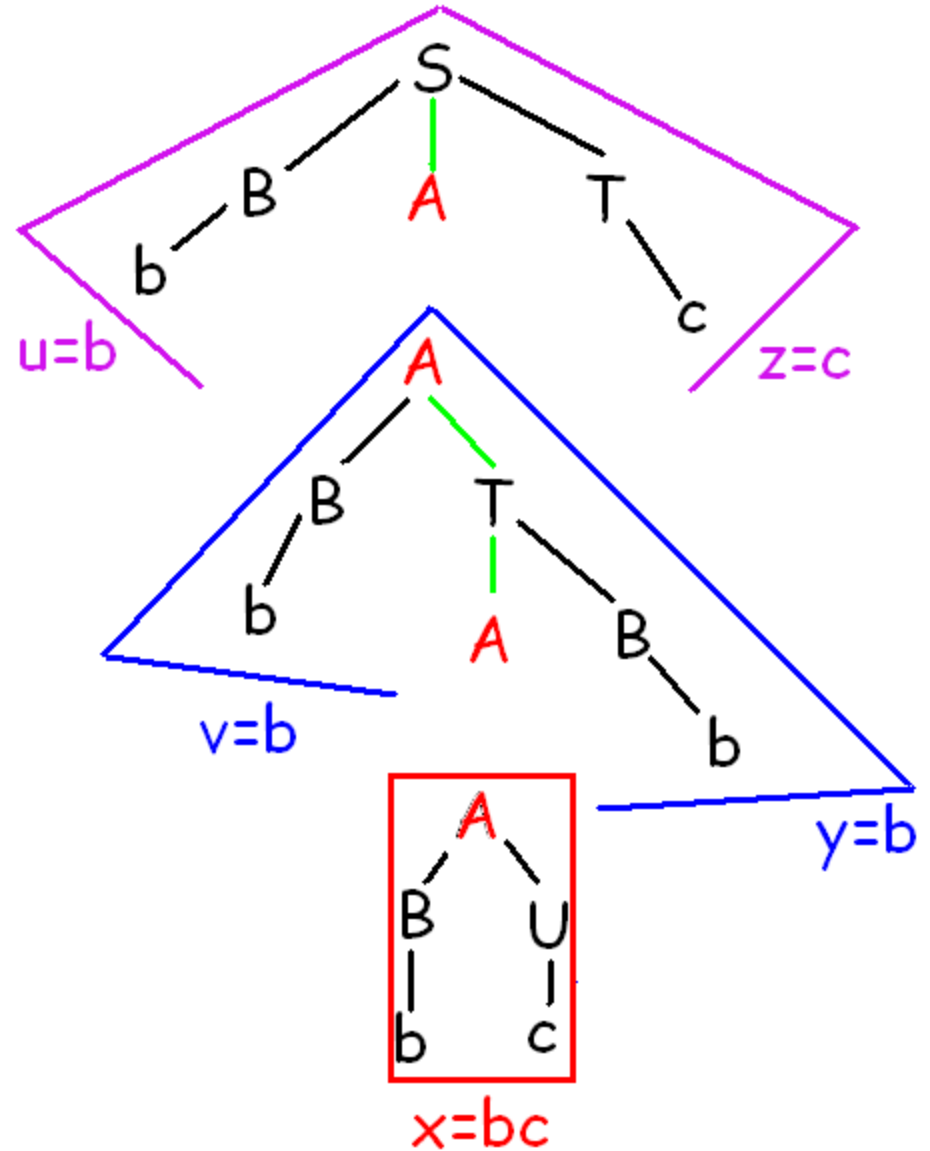


Cut off part of
tree from
second copy
downwards.

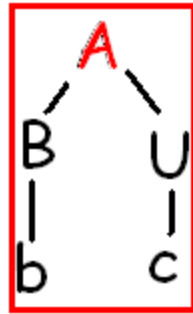
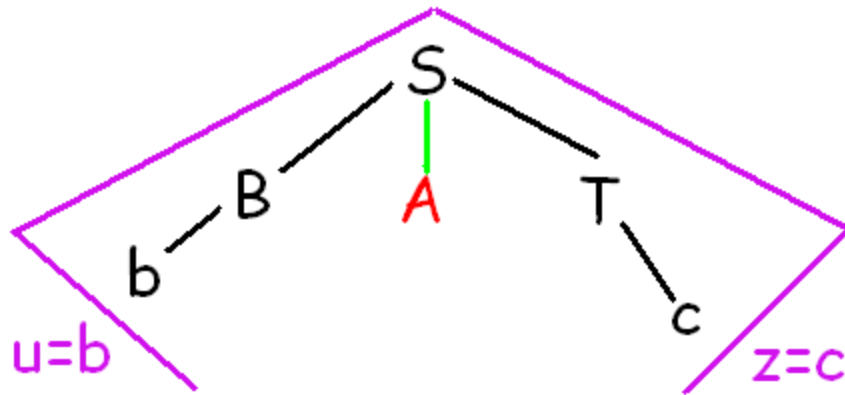




Cut off
part from
first copy
downwards.



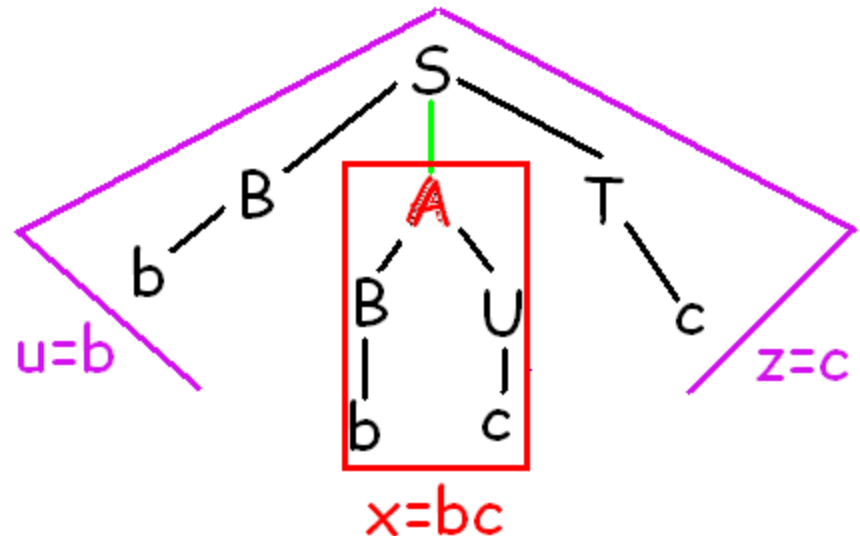
To pump zero times:



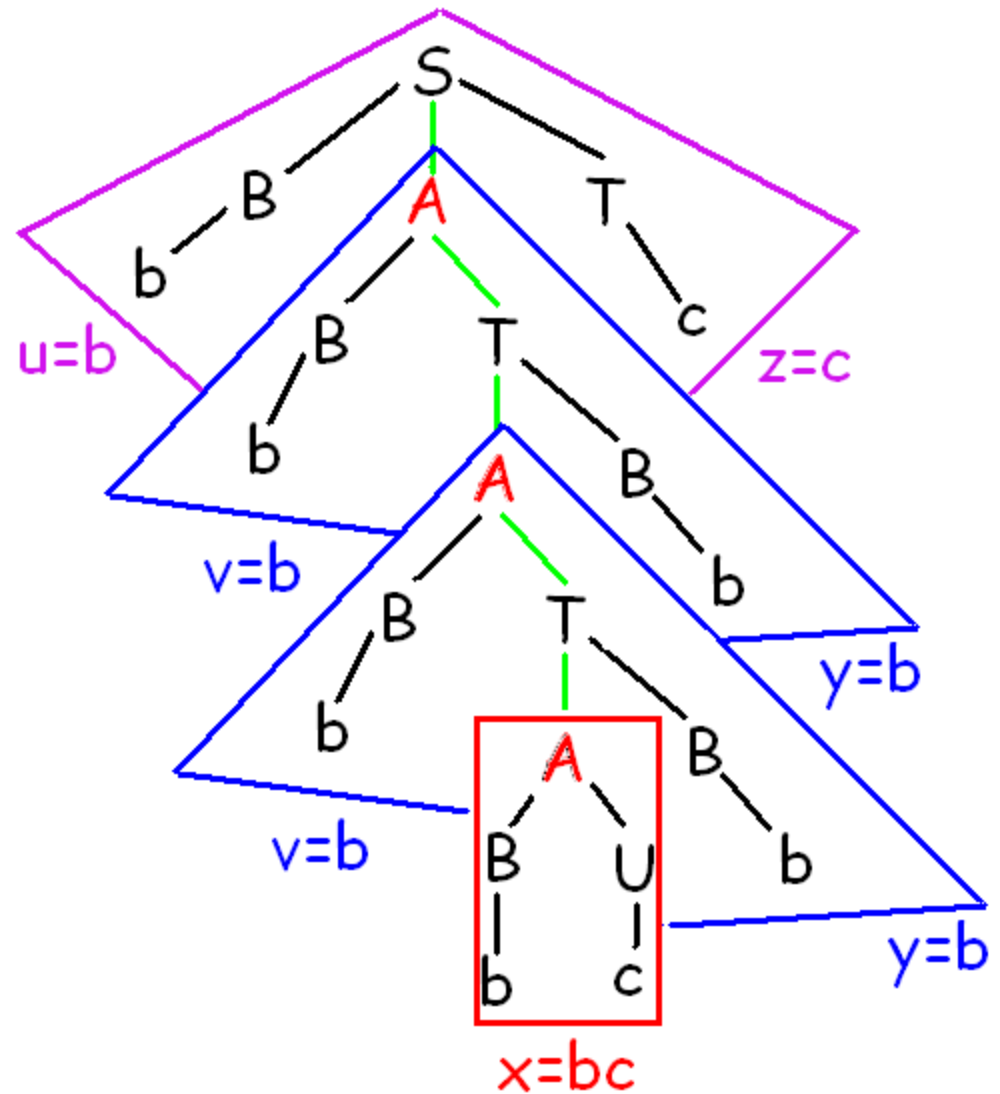
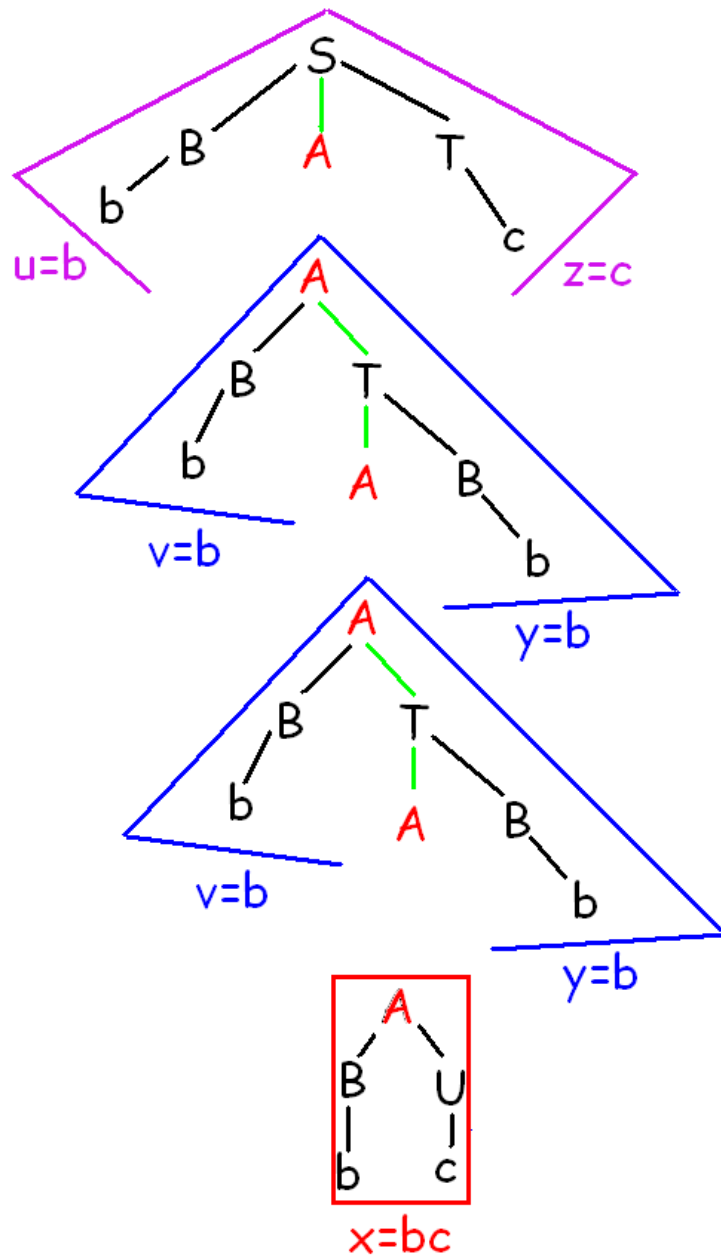
New yield:

$$u x z = b \text{ } b c \text{ } c$$

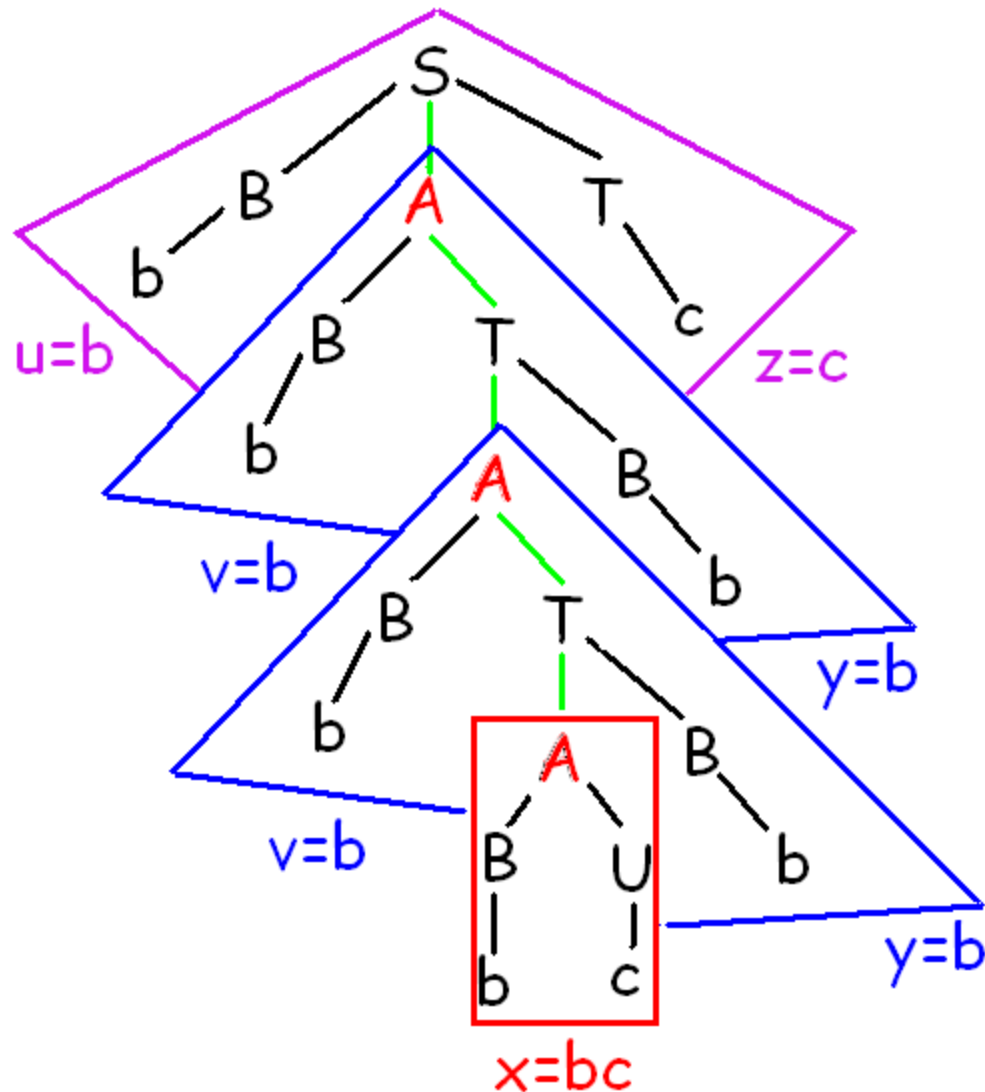
$$= u v^0 x y^0 z$$



To pump twice:



Pumping twice:



New yield is:

$u v v x y y z$

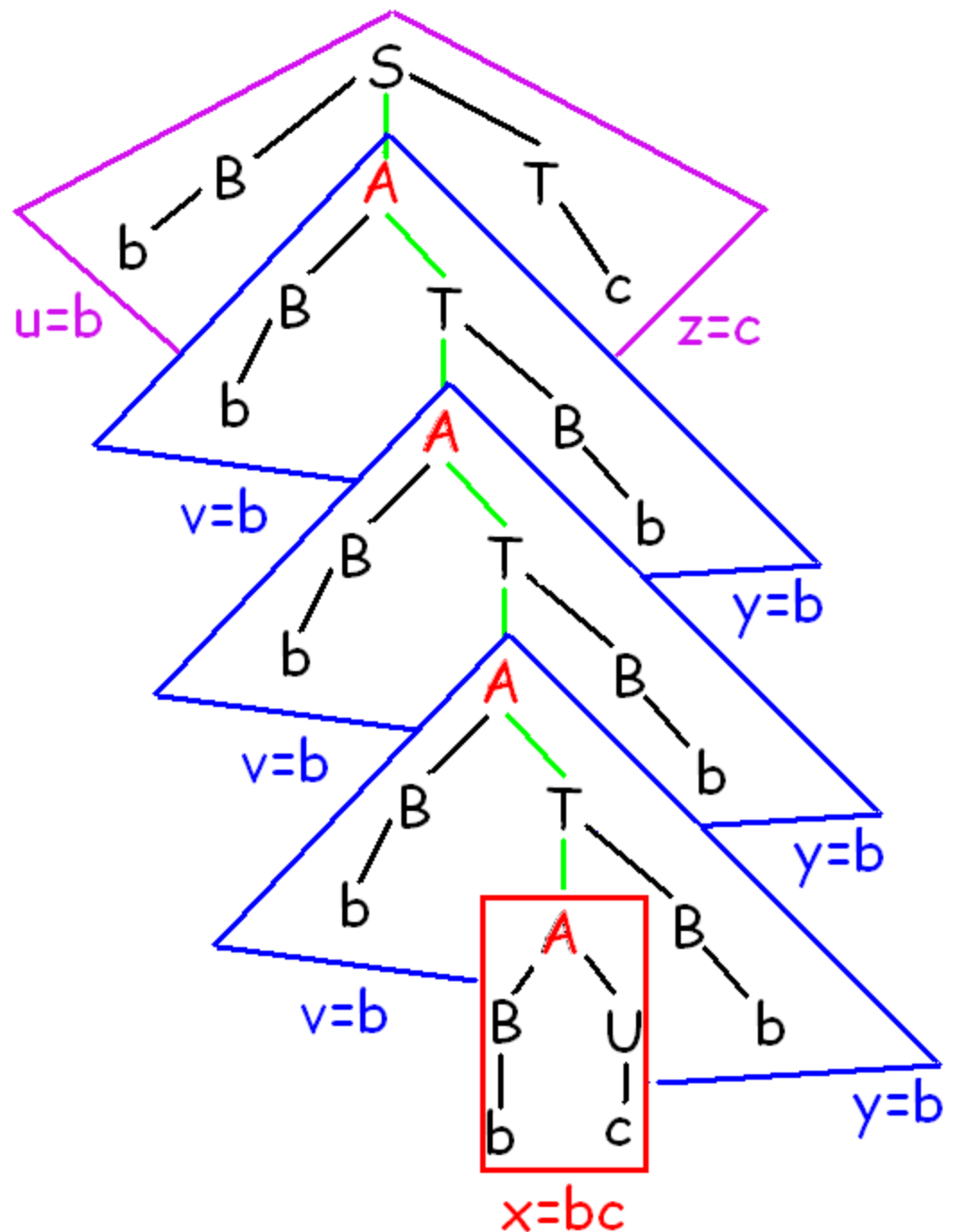
$= u v^2 x y^2 z$

To pump
three
times:

New yield is:

$u v v v x y y y z$

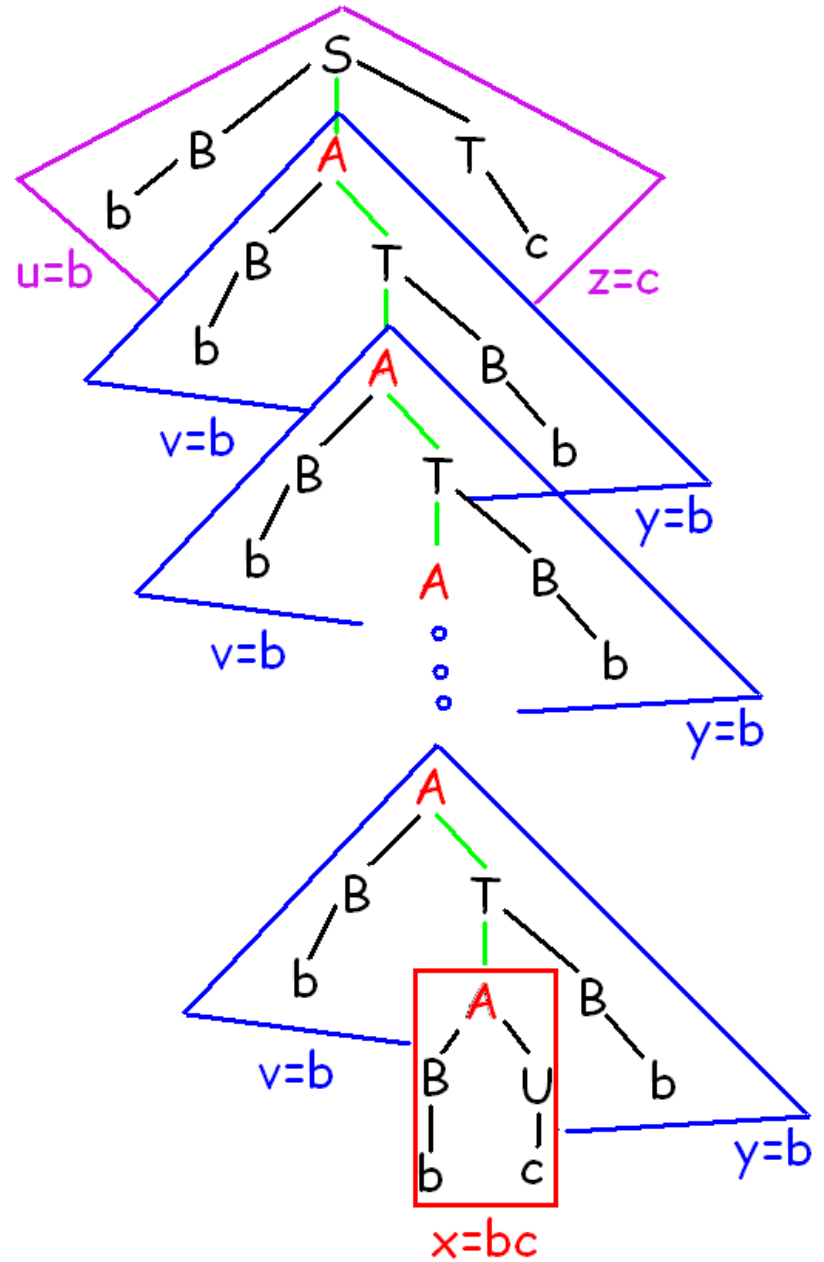
$= u v^3 x y^3 z$



To pump n times:

New yield is:

$$= u v^n x y^n z$$



The Pumping Theorem for Context-Free Languages:

Let G be a context-free grammar.

Then there exists some constant k which depends on G such that for any string w which is generated by G with $|w| \geq k$,

there exists u, v, x, y, z , such that

1. $w = u v x y z$,
2. $|v| + |y| \geq 1$, and
3. $u v^n x y^n z$ is in L for all $n \geq 0$.

Prove the following languages are context-free by designing context-free grammars which generate them:

$$L_1 = \{a^n b^n c^p : n, p \geq 0\}$$

$$L_2 = \{a^n b^p c^n : n, p \geq 0\}$$

$$L_3 = \{a^n b^m : n \neq m, n, m \geq 0\}$$

$$\text{Hint: } L_3 = \{a^n b^m : n < m\} \cup \{a^n b^m : n > m\}$$

$$L_4 = \{c u c v c : |u| = |v|, u, v \in \{a, b\}^*\}$$

What is $L_1 \cap L_2$?