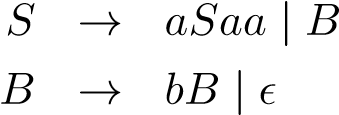
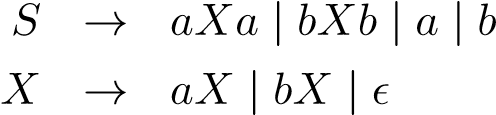
**Practice 2**

**Solutions**

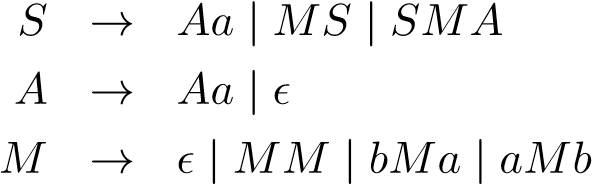
1. Give a context-free grammar (CFG) for each of the following languages over the alphabet Σ = {*a,b*}:
   1. All strings in the language *L* : {*anbma*2*n*|*n,m* ≥ 0}



* 1. All nonempty strings that start and end with the same symbol.



* 1. All strings with more a’s than b’s.



**2.** Consider the grammar below that generates roman numerals, with terminals {**c***,***l***,***x***,***v***,***i**}. *c* = 100*,l* = 50*,x* = 10*,v* = 5*,i* = 1. Notice that we use lowercase characters here to represent the numerals, to distinguish them from the non-terminals.

|  |  |  |
| --- | --- | --- |
| *S* | → | **x***TU* | **l***X* | *X* |
| *T* | → | **c** | **l** |
| *X* | → | **x***X* | *U* |
| *U* | → | **i***Y* | **v***I* | *I* |
| *Y* | → | **x** | **v** |



1. Draw a parse tree for 47: “xlvii”.

S

x

T

U

l

v

I

i

I

i

I

ε

1. Is this grammar ambiguous?

**No**

3. Let *L* be the language {*w* ∈ {*a,b*}∗|*w* contains exactly one more *b* than *a*}.

* 1. Give a context-free grammar that generates *L*.

**Solution:**

|  |  |  |
| --- | --- | --- |
| *S* | → | *ε*|*aB*|*bA* |
| *A* | → | *aS*|*bAA* |
| *B* | → | *bS*|*aBB* |

we know that the variable *B* generates the strings with exactly one more *b* than *a*, so if we declare *B* to be the start symbol in the above grammar, we have one solution to the problem. To make this problem more interesting, I will give a different solution:

* + 1. → *TbT*
    2. → *aTb*|*bTa*|*TT*|*ε*

You were not asked to explain how your grammar works, but here is an explanation for the above grammar. We know that the variable *T* generates the strings with the same number of *a*’s and *b*’s. Now suppose we have a string *w* with exactly one more *b* than *a*. Then, we have to show that *w* matches the rule *S* → *TbT*. We think of a counter running along *w* where *b* counts as +1 and *a* counts as −1. The count at the end of *w* is +1, so we divide *w* up into *ucv* where *c* is the symbol read when the count first reaches +1. The count at the end of *u* is either 0 or 2, but if the count is 2 at the end of *u*, then the count must have been 1 somewhere in the middle of *u*, contradicting how we picked *c*, so the count is 0 at the end of *u* and *c* must be a *b*. Since *u* brings the counter from 0 to 0, *u* must have the same number of *a*’s as *b*’s, and since *v* brings the counter from 1 to 1, *v* also has the same number of *a*’s as *b*’s. Thus, *w* = *ubv* matches the rule *S* → *TbT*.

* 1. Give a leftmost derivation and a parse tree in your grammar for thestring *abbabab*.

**Solution:** A leftmost derivation in the second grammar is: *S* ⇒ *TbT* ⇒ *aTbbT* ⇒ *abbT* ⇒ *abbaTb* ⇒ *abbabTab* ⇒ *abbabab*

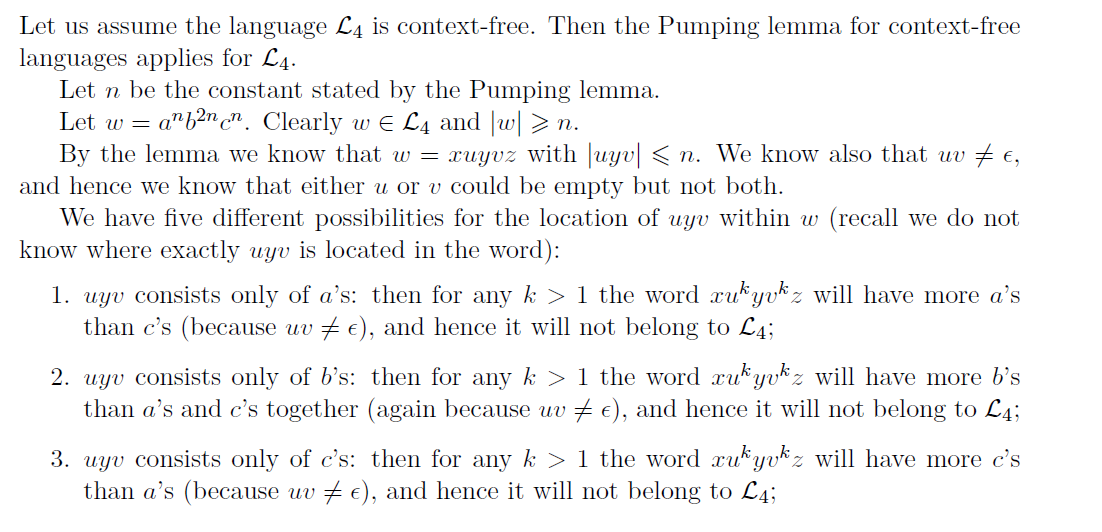
(c). Give an unambiguous grammar for the language *L* of the previous problem.

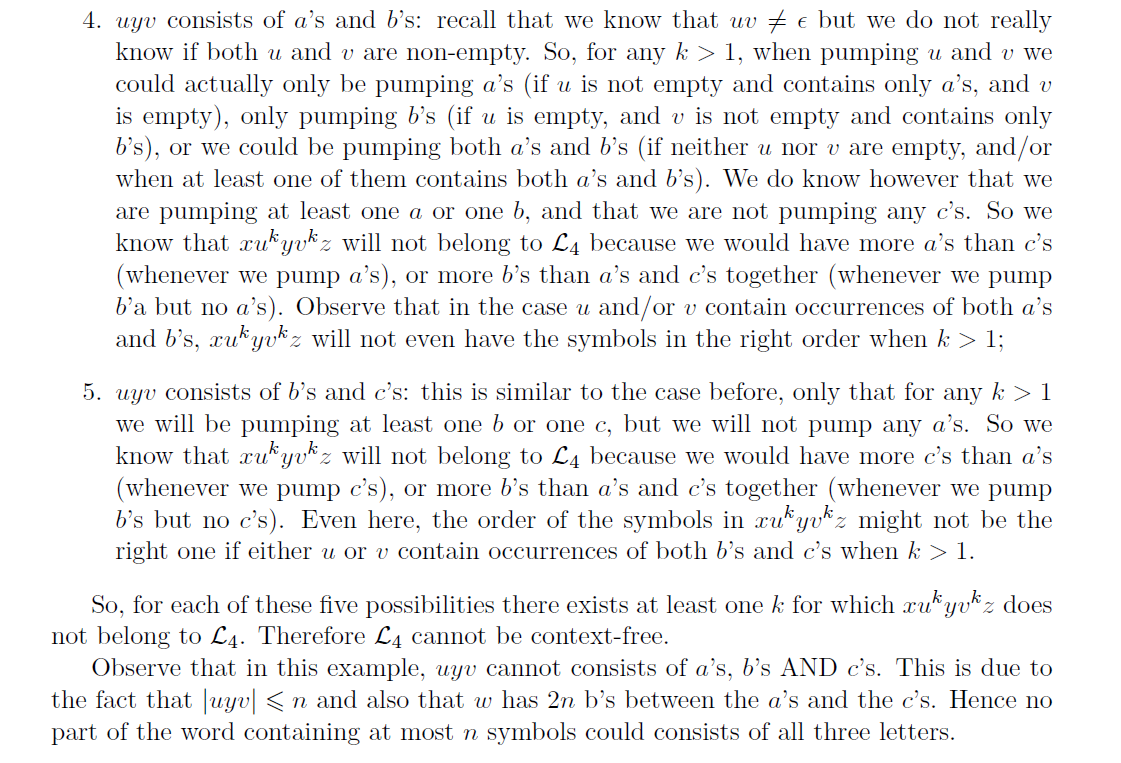
(This is a difficult problem, but give it a try. As a hint, you can use three variables other than the start symbol. One variable generates strings with the same number of *a*’s as *b*’s, the second variable generates strings with the same number of *a*’s as *b*’s that have the additional property that every prefix has at least as many *a*’s as *b*’s, and the third variable generates all strings with the same number of *a*’s as *b*’s that have the additional property that every prefix has at least as many *b*’s as *a*’s.)

**Solution:**

|  |  |  |
| --- | --- | --- |
| *S* | → | *WbT* |
| *T* | → | *ε*|*aWbT*|*bV aT* |
| *W* | → | *ε*|*aWbW* |
| *V* | → | *ε*|*bV aV* |

2. Prove L4 = {} is not a Context-free Language





3. Prove L5 = {s2s | s} is not a Context-free Language

