Assignment 3

**Exercise 3.2**

**Key:**

1. ?
   1. Means 0 or 1 occurrence
2. \*
   1. Means 0 or many occurrence
3. +
   1. Means 1 or many occurrence
4. {n,m} – Ask the Prof. if we can use this
   1. This is called a quantifier
   2. This means **at least n and at most m occurrences** (range quantifier).
   3. Example: [0-9]{1,3} matches **1 to 3 digits**.

Answer:

0 | [1-9] | [0-9]?[0-9]? ('[0-9] [0-9] [0-9])\*  
or   
 0 | [1-9]{0,2} ([0-9]{3})\*

**Exercise 3.2**

**Rules:**

1. Regular expressions

L a = {w ∈ {0,1}\* | |w| is odd}  
  
(0|1)(00|11|01|10)\*  
(0|1)(())

**--------------------------------------------------------------------------------------------  
  
Syntax of Regular Expression**

1. 1 and 0 are elementary regular expressions, see 3
2. From this the expression 1.0 can be formed according to 4
3. As ε is a regular expression, according to 2, we can build the (1.0 | ε)
4. (7) allows to bracket expressions, so you get (1⋅0 | ε)
5. according to (6) we get (1⋅0 | ε)\* as a regular expression.

**Notation Convention 5.3 - Precedence of the operators**

To make clear which structure is meant by a regular expression, there are rules of precedence for the operators, similar to arithmetic:

\* binds stronger than ⋅   
⋅ binds stronger than |  
  
VERY IMORTANT  
**Precedence does not dictate the position in the abstract syntax tree**. Instead, it tells us **how to group** the parts of the expression

From Script – Lesson 6 (Sara)

1. 01 can be also known as 0.1, which means multiply, dont confuse with \*, as that means 0 or many occurrences   
    .  
    / \  
    0 1
2. 01\*  
    .  
    / \  
    0 \*  
    /   
    1
3. (01)\*  
     
    \*  
    /   
    .  
    / \  
    0 1
4. 01| ε  
     
    |  
    / \  
    . ε  
    / \   
    0 1
5. (01)\*(01| ε)  
     
    .  
    / \  
    \* |  
    / / \   
    . . ε  
    / \ / \   
    0 1 0 1

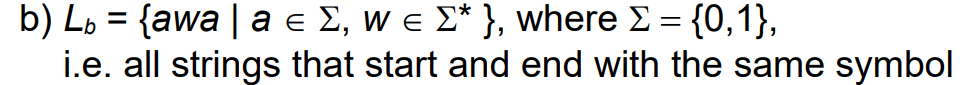
A screenshot of a computer

Description automatically generated

A black screen with white text

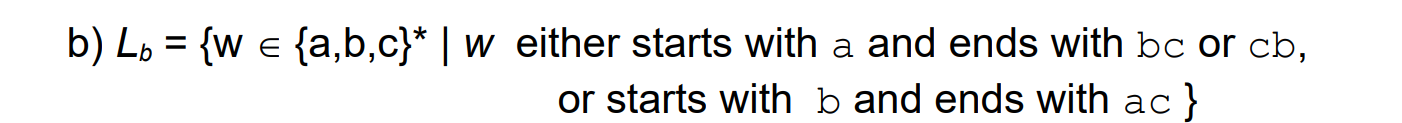
Description automatically generated

**Exercise 3.3**

1. **La = {w** ∈ **{0,1}\* | |w| is odd}**(0|1)(00|01|10|11)\*  
   (0|1)((0|1)(0|1))\*
2. ****0(0|1)\*0 |1(1|0)\*1
3. ****
4. (0|e)\*(1|e) | (1|e)\* (0|e)  
   (0|e)(10)\*(1|e)

Exercise 3.5

1. A close-up of a sign

   Description automatically generated  
   (0|1)\*(01)(0|1)\*
2.   
   a(a|b|c)\*(bc |cb) | b(a|b|c)\*ac
3.   
   (a|c)\*(b(a|c)\*b(a|c)\*)\*
4. a((a|bb)a)|aaa  
   a((a|bb)a)|a\*  
   a((a|b\*)a)|a\*  
     
   Prof approach:
   1. a(aa|bba)|aaa  
      aaa|abba|aaa  
      aaa|abba  
      a(a|bb)a

3.6

a) a((a | bb)a) | aaa

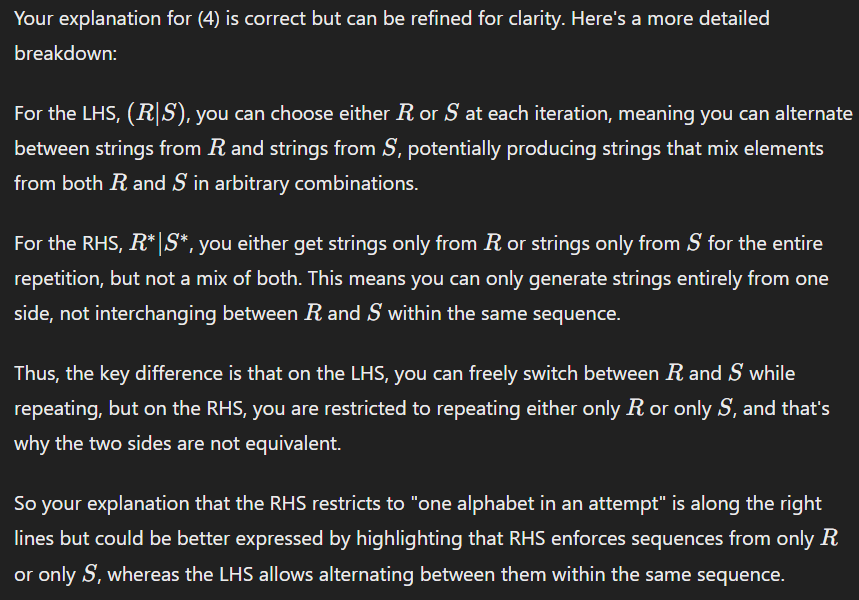
aaa|abba|aaa  
aaa|abba  
a(a|bb)a

b)  
aa | (ab | aa)\*  
(aa | ab)\*  
(a(a|b))\*

c)  
a(bd|cd) ( (ac|ab)d )\*)\*  
a((c|b)d) ( (a(c|b)d )\*)\*  
(a(b|c)d\*)\*  
(a(b|c)d)\*

Exercise 3.7

(4) The thing is the reason this is wrong because, with the RHS expression, we can see we can only have a possibility of one alphabet in an attempt, where as at the LHS, we can have as many times first expression R or S



A white letter on a black background

Description automatically generated

My question is that in that case, there is a case where this statement can be true and in one case it can be false,

**It will be true, when R\*on LHS is repeated once and for LHS twice**  
aa=aa  
**It will be false, when R\* on LHS empty and also R\* is empty on the RHS**a=””  
  
So in terms of saying if all cases are true in the question as its stated, then this should be considered as false

A screenshot of a computer

Description automatically generated

1. S -> AB -> aAbbB -> aaAbbbbB -> aacbbbbB ->aacbbbcB -> aacbbbc
2. S -> AB -> aAbbB -> acbbB -> acbb
3. S -> AB -> aAbbB -> acbbB -> No possible to derivate a from B