Q3 Read the handout of the first lecture and the handout about notation. Make sure you understand the concepts of strings and languages. In the context of the CFL-course, what is meant by the term *language*?

Strings are lists of characters; Languages are sets of strings

Q4 Give the definition for regular expressions — this is an inductive datatype. What is the meaning of a regular expression? (Hint: The meaning is de- fined recursively.)

Definition for regular expressions (inductive datatype):

Their inductive definition:

r	$::=$	$\mathbf{0}$	nothing
	$\mid$	$\mathbf{1}$	empty string / "" / []
	$\mid$	c	character
	$\mid$	$r_1 + r_2$	alternative / choice
	$\mid$	$r_1 \cdot r_2$	sequence
	$\mid$	r^*	star (zero or more)

Meaning of a regular expression: all the strings a regular expression can match

The Meaning of a Regex

...all the strings a regular expression can match.

$$
\begin{array}{rcl}
L(\mathbf{0}) & \stackrel{\text{def}}{=} & \{\} \\
L(\mathbf{1}) & \stackrel{\text{def}}{=} & \{[]\} \\
L(c) & \stackrel{\text{def}}{=} & \{[c]\} \\
L(r_1 + r_2) & \stackrel{\text{def}}{=} & L(r_1) \cup L(r_2) \\
L(r_1 \cdot r_2) & \stackrel{\text{def}}{=} & L(r_1) @ L(r_2) \\
L(r^*) & \stackrel{\text{def}}{=} &
\end{array}
$$

Q5 Assume the concatenation operation of two strings is written as $s_1 @ s_2$. Define the operation of *concatenating* two sets of strings. This operation is also written as _ @ _. According to this definition, what is *A* @ {} equal to? Is in general *A* @ *B* equal to *B* @ *A*?

A @ {} = {}. As {} represents the regular expression that cannot match any expression.

A @ B != B @ A. In the first expression, A concatenates with B, so the elements in A should start. In the second should be totally different.

Q6 Assume a set *A* contains 4 strings and a set *B* contains 7 strings. None of the strings is the empty string. How many strings are in *A* @ *B*?

≤ 4*7 = 28. As there may be duplicates.

Q7 How is the power of a language defined? (Hint: There are two rules, one for $\_^{0}$ and one for $\_^{n+1}$.)

Power of 0: $A^0 = \{[]\}$

Power of n+1: $A^{n+1} = A^n @ A$

Q8 Let A = {[a], [b], [c], [d]}. (1) How many strings are in A^4? (2) Consider also the case of A^4 where one of the strings in A is the empty string, for example A = {[a], [b], [c], []}.

(1) $4^4 = 256$
(2) $3^4 + 3^3 + 3^2 + 3^1 + 3^0 = 121$

Q9 (1) How many basic regular expressions are there to match only the string *abcd*? (2) How many if they cannot include 1 and 0? (3) How many if they are also not allowed to contain stars? (4) How many if they are also not allowed to contain _ + _?

(1) infinity

(2) infinity (eg: abcd 1*)

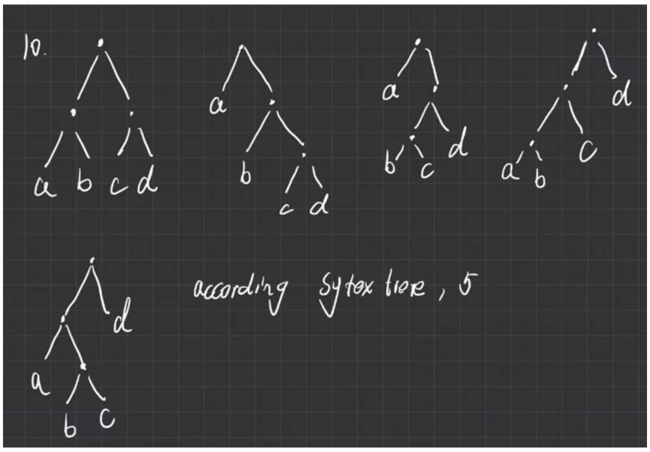
(3) infinity (eg: abcd + abcd...)

(4) 5

Q10 When are two regular expressions equivalent? Can you think of instances where two regular expressions match the same strings, but it is not so obvious that they do? For example *a* + *b* and *b* + *a* do not count...they obviously match the same strings, namely [*a*] and [*b*].

Two regular expressions are equivalent if they describe the same language.

Example: (a + b)* = (a + b)* + (a · (a + b)*)



Q11 What is meant by the notions *evil regular expressions* and by *catastrophic backtracking*?

Evil regular expressions are the regular expressions which may cause catastrophic backtracking in the regular expression engine.

Catastrophic backtracking is that the regular expression equation for a long time leads to the servers crash down.

Q12 Given the regular expression $(\boldsymbol{a}+\boldsymbol{b})^{\square} \cdot \boldsymbol{b} \cdot (\boldsymbol{a}+\boldsymbol{b})^{\square}$,which of the following regular expressions are equivalent

1) $(ab+bb)^{*} \cdot (a+b)^{*}$ False

2) $(a+b)^{*} \cdot (ba+bb+b) \cdot (a+b)^{*}$ True

3) $(a+b)^{*} \cdot (a+b) \cdot (a+b)^{*}$ False