

First name	Family name	Matr no.

1

Question 1

Multiple choice

(8 * 1 = 8 points)

T(rue) or F(false)? Please check!

(a) Lexical analysis:

- ☐T ☐F A state of a finite automaton is called *productive* if the automaton, starting in that state, accepts every input word.
- ☐T ☐F The worst-case time complexity of running the backtracking automaton is quadratic in the length of the input word.

(b) Syntactic analysis:

- ☐T ☐F Every regular language can be generated by an $LL(1)$ grammar.
- ☐T ☐F If a context-free grammar has a reduce/reduce conflict, then there exists a pair of productions such that the right-hand side of the first production is a suffix of the right-hand side of the second production.

(c) Semantic analysis:

- ☐T ☐F Every L-attributed grammar is strongly noncircular.
(**Reminder:** In an L-attributed grammar, attribute dependencies on the right-hand sides of productions are only allowed to run *from left to right*.)
- ☐T ☐F There are strongly noncircular attribute grammars that are circular.

(d) Code generation:

- ☐T ☐F The following runtime stack is a possible result of a computation in the abstract machine:

15 : 4 : 9 : -3 : 5 : 5 : 6 : 3 : 2 : 1 : 5 : 4 : 12 : 3 : 4 : 4 : 3 : 25 : 0 : 0 : 0 : 0 : 12

(**Reminder:** the topmost entry is to the left, and each activation block is of the form $sl : dl : ra : loc_1 : \dots : loc_k$.)

- ☐T ☐F If “jumping code” is generated for sequentially evaluating Boolean expressions, the machine instructions AND, OR, and NOT are no longer required.

First name	Family name	Matr no.

2

Question 2 Lexical Analysis (3 + 3 = 6 points)

The *rational numbers* are given by $\mathbb{Q} := \{m/n \mid m \in \mathbb{Z}, n \in \mathbb{N}_{>0}\}$. Here the denominator n can be omitted if $n = 1$. For example, the words $-1/2$, $0/5$, -0 , and $+42$ represent elements of $\mathbb{Q}$, but *not* 007 and $01/02$ (leading zeros) or $3/-4$ (signed denominator).

- (a) Give a regular expression describing $\mathbb{Q}$.
- (b) Give a nondeterministic finite automaton that accepts $\mathbb{Q}$.

First name	Family name	Matr no.

3

Question 3 Top-down syntax analysis (4 + 3 = 7 points)

(a) Transform the following grammar G into an equivalent $LL(1)$ grammar G' :

$$\begin{aligned} S &\rightarrow S A \mid A \\ A &\rightarrow a \mid (B) \\ B &\rightarrow A B \mid A \end{aligned}$$

(b) Prove the $LL(1)$ property of G' .

First name	Family name	Matr no.

4

Question 4 Bottom-up syntax analysis (5 + 2 = 7 points)

Let G be the following context-free grammar:

$$\begin{array}{lll}
 S & \rightarrow & E \quad (0) \\
 E & \rightarrow & E + F \mid F \quad (1, 2) \\
 F & \rightarrow & (E) \mid a \quad (3, 4)
 \end{array}$$

- (a) Compute all $LR(0)$ set of G .
- (b) Show that $G \in SLR(1)$ but $G \notin LR(0)$.

First name	Family name	Matr no.

5

Question 5

Semantic analysis

(8 points)

As is known, the language $L := \{a^n b^n c^n \mid n \in \mathbb{N}\}$ is not context-free. Give an attribute grammar which generates the (context-free) language $\{a, b, c\}^*$, and which indicates whether a derived word is an element of L (using a Boolean-valued attribute of the start symbol). To this aim, specify

- the context-free rules,
- the association of attributes to symbols,
- the domains of the attributes, and
- the attribute equations for each production.

First name	Family name	Matr no.

6

Question 6 Code generation (4 points)

Let the current state of the abstract machine AM for procedures be given by

$$(ca, d, p) := (7, -3, 9 : 4 : 26 : 3 : 7 : 4 : 3 : 36 : 5 : 10 : 4 : 40 : 1 : 2 : \dots).$$

We assume that the corresponding code contains the following instructions:

```

      ⋮
7:  LOAD (1, 2)      % (dif, off)
8:  ADD
9:  RET
      ⋮
26: CALL (38, 1, 3)  % (ca, dif, loc)

```

Determine the respective machine state after the execution of next four instructions.

(**Reminder:** the topmost entry of the runtime stack p is to the left, and each activation block is of the form $sl : dl : ra : loc_1 : \dots : loc_k$.)