# DA53: Compilation and Language Theory
# Mid-Term Exam A2022

Duration: 1h30. No document allowed.
English recommended, French accepted.

November 18, 2022

---

## Part 1: Base Questions (6 points)

### Question 1.1:

What is a syntax tree in the context of a compiler?

### Question 1.2:

What is a parse tree in the context of a compiler?

### Question 1.3:

List the different stages or steps that are followed by a compiler for generating a binary executable code from a source program.

---

## Part 2: Lexical Analysis (10 points)

Let the language that is composed of:

1. the number constants (integer or decimal number, without the exponential part), e.g., "1.456".
2. the two comparison operators "=", "<>" that tests equility and innequality of two operands that are at the left or right of the operator.
3. the "print v" statement that print out its parameter "v". "v" is a value, not a variable since there is no variable in this language.
4. the "if-then" statement: `if (CONDITION) then STATEMENT`, where `CONDITION` is composed of numeric constants and comparison operator and `STATEMENT` may be another "if-then" statement or a "print v".

**Example:**

```
if 4 < 5 then
  if 7 <> 4 then
    print 1
else
  print 3
```

### Question 2.1:

What is the alphabet of this language?

### Question 2.2:

Write the table that is matching the regular expressions, the lexemes, the tokens and the attributes of the tokens in four columns.

| Token name | Example(s) of lexemes | Regular expression | Attribute if any |
|---|---|---|---|
| | | | |

### Question 2.3:

Draw the Nondeterministic Finite Automata (NFA) that is recognizing the language tokens. In a NFA, the nodes represents the current state (or step) in the recognition of a token. The edges in the NFA have a label with one or more characters that represent the character(s) to be recognized from a given state of the NFA. The label of a NFA edge may be $\epsilon$ for representing a transition with "nothing".

---

## Part 3: Syntax Analysis (4 points)

In this part, you must use the same language as in Part 2.

### Question 3.1:

Write the grammar rules (using the Backus Naur Form) for the language that is described in Part 2. You grammar will be analyzed with LL(0) approach. So that, you should avoid left-recursions and remove ambiguities from the rules.
**Reminder:** A BNF grammar is composed of production rules that have a head on the left, and a sequence of terminals (tokens) and nonterminals on the right.

```
NONTERMINAL-1 ::= TERMINAL-1 NONTERMINAL-2 TERMINAL-2 NONTERMINAL-3
```