UNIVERSITÉ LIBRE DE BRUXELLES

# INFO-F-403 – Introduction to language theory and compiling
# First session examination

19 January 2024

## Instructions

- This is a closed book test. You are not allowed to use any kind of reference.
- You can anwser in French or in English.
- Write your first and last names on each sheet that you hand in.
- Write clearly: you can use a pencil or a ballpen or even a quill as long as your answers are readable!
- Always provide full and rigourous justifications along with your answers.
- This test is worth 12 points out of 20. The weight of each questions is given as a reference.
- The test lasts three hours.
- In your answers (diagrams representing automata, grammars,...), you can always use the conventions adopted in the course, without recalling them explicitely. If you deviate from these conventions, be sure to make it clear.

## Question 1 — 3 points

1. Define formally the syntax of DFA (deterministic finite automata). Do not forget to give the condition on the transition relation that makes the automaton deterministic.
2. Then, *draw* a DFA on the alphabet $\Sigma = \{0,1\}$ that accepts the set of all words where the number of 1's is a multiple of 5.
3. Give the formal syntax of your automaton (following the definition you have given in point 1). Show that your automaton respects the condition you have given for determinism.
4. Generalise your construction by explaining how one can build an automaton on the alphabet $\Sigma = \{0,1\}$ that accepts all words where the number of 1's is equal to $k \mod \ell$, for arbitrary values of $k$ and $\ell$ with $k < \ell$. That is, we want a general construction where, given $k$ and $\ell$, the automaton checks that dividing the number of 1's by $\ell$ gives a remainder of $k$.

   For example, for $k = 1$ and $\ell = 3$:
   - the automaton accepts 110101, because there are four 1's and $4 \mod 3 = 1$;
   - the automaton rejects 000011, because there are two 1's and $2 \mod 3 = 2 \neq 1$.

ULB UNIVERSITÉ LIBRE DE BRUXELLES

## Question 2 — 4 points

We consider the following language:

$$L = \{w_1 w_2 \cdots w_{2n} \mid n \geq 0 \text{ and for all } i \geq 0 \text{ there is } k_i \text{ s.t. } w_i = \mathtt{a}^{k_i}\mathtt{b}^{k_i}\}$$

That is, $L$ contains all the words $w_i$ that are a sequence of any *even* number of subwords of the form $\mathtt{a}^{k_i}\mathtt{b}^{k_i}$. Thus, each subword $w_i$ contains the same number $k_i$ of a's and b's, but the $k_i$'s need not be equal along the word.

For example, $\varepsilon$ and aabbaaabbb are in the language; but abbab and aabbababab are not (there are more b's than a's in the first subword of abbab; and the number of subword is odd in the second).

1. Give an automaton (either a finite automaton, or a pushdown automaton) that accepts $L$. Say whether your automaton is a DFA, NFA, $\varepsilon$-NFA or PDA; and, if it is a PDA, say whether it accepts by empty stack or accepting state.
2. Explain, intuitively, why your automaton accepts $L$.
3. Justify your choice of type of automaton. If you chose a PDA could you accept the language with an finite automaton? If you chose a finite automaton, could you have used a PDA? The expected justification is informal, you are not requested to give a proof.
4. Give, formally, an accepting run of your automaton on aabbab. Then, show, by using the formal notations of runs, that aab is not accepted.

## Question 3 — 2 points

1. Give the formal syntax of context free grammars (CFG)
2. Give the formal definitions of the functions First[1] and Follow[1]
3. Give the formal condition for such a grammar to be *strong* LL(1), and explain it intuitively.
4. Given two examples of CFG: one which is strong LL(1) and one which is not. Both grammars should generate an infinite language. Justify your answers.

## Question 4 — 3 points

We consider the following grammar, with $\{S', S, A, B, C\}$ and terminals $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \$\}$:

| | | | |
|---|---|---|---|
| (1) | $S'$ | $\rightarrow$ | $S\$$ |
| (2) | $S$ | $\rightarrow$ | a$B$b |
| (3) | | $\rightarrow$ | a$A$ |
| (4) | | $\rightarrow$ | c |
| (5) | $B$ | $\rightarrow$ | d$B$ |
| (6) | | $\rightarrow$ | a |
| (7) | $A$ | $\rightarrow$ | $AC$ |
| (8) | | $\rightarrow$ | b |
| (9) | $C$ | $\rightarrow$ | c |

1. Give its canonical finite state machine (CFSM).
2. Is it LR(0)?
3. Is it SLR(1)?

For the two last subquestions, justify your answers by giving action tables without conflicts when your answer is positive, or by pointing out the conflict(s) when your answer is negative.