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

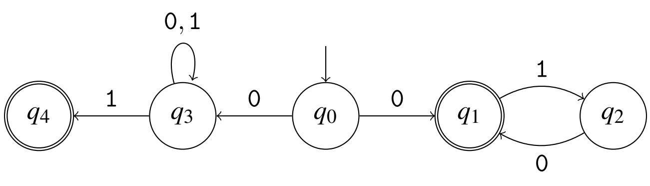
January, 18th, 2021

Instructions

- You can anwser in French or in English.
- If you answer on paper: 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.
- 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

Here is a finite automaton on the alphabet $\Sigma = \{0, 1\}$:



1. Is this automaton an ε-NFA? an NFA? a DFA? Justify your answers.
2. If the automaton is not a DFA, convert it into an equivalent DFA. Explain the technique you have used to perform this conversion and give the intermediate steps.
3. Convert this automaton into an equivalent regular expression, using the state elimination technique. Give the intermediate steps of the construction.

Question 2 — 3 points

In the lecture notes, we have seen several important closure properties of the classes of languages that we have studied. For example, we know that regular languages are closed under union (the union of two regular languages is always regular), but context-free languages are not closed under intersection (the intersection of two context-free languages is not always context-free).

There are, however, some important closure properties that we have not discussed. Here are two of them:

1. We claim that context-free languages are *closed under intersection with regular languages*. That is, for all context-free languages L_1, and all regular languages L_2: $L_1 \cap L_2$ is context-free. In order to prove this statement, you are asked to give an algorithm that receives a finite automaton A and a pushdown automaton P, and returns a pushdown automaton accepting the intersection of the languages of A and P. Hint: adapt the algorithm for computing the intersection of finite automata.
2. A *homomorphism* is a language transformation defined by a function $h : \Sigma \to \Sigma^*$ (assuming Σ is the alphabet of the language) that replaces each letter $a \in \Sigma$ by a finite word $h(a) \in \Sigma^*$. For example, if $\Sigma = \{a, b\}$; $h(a) = c$ and $h(b) = de$, then applying h to word aab yields $ccde$ (the two first a's have been replaced by c, and the last b has been replaced by de).

   Formally, we extend h to words as follows: if $w = w_1 w_2 \cdots w_n$, then $h(w) = h(w_1) \cdot h(w_2) \cdots h(w_n)$. By convention, $h(\varepsilon) = \varepsilon$. We can also extend h to languages, by applying it to all words in the language: $h(L) = \{h(w) \mid w \in L\}$. So, for example $h(\{ab, aab\}) = \{cde, ccde\}$.

   We claim that regular languages are *closed under such homomorphism*. That is, for all regular languages L: $h(L)$ is a regular language. You are asked to prove this statement. Hint: there are several ways to do so. For example, you can either follow the inductive definition of regular languages, or provide an algorithm that turns a finite automaton A into a finite automaton B s.t. $L(B) = h(L(A))$, and argue that the algorithm is correct.

Important: For this question, you are requested to give a formal answer. Your proofs should be expressed with the necessary formalism and your algorithms as well. Of course, you should also give some intuitive explanations along with your formal answers.

Question 3 — 3 points

Give the diagram of a *deterministic* pushdown automaton, on the alphabet $\Sigma = \{\texttt{0}, \texttt{1}, \texttt{a}\}$, that accepts the language $L = \left\{ (\texttt{01})^n \texttt{a} (\texttt{10})^n \mid n \geq \texttt{0} \right\}$ using the empty stack acceptance condition. Then, give the run of the automaton on the word `0101a1010` and explain why it is accepting.

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Question 4 — 3 points

Give the LR(0) canonical finite state machine (CFSM) of the following grammar (where the set of variables is $\{S', S, A, B, C\}$ and the set of terminals is $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$). **Important**: In your answer, make sure to clearly mark the items that are part of the kernel and those that are part of the closure, as we have done in the practicals and in the lectures. This is to ensure that you have actually built the answer by hand, and not used a tool to do it for you!

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

Is the grammar LR(0)? Is it SLR(1)? 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.