ULB UNIVERSITÉ LIBRE DE BRUXELLES

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

January, 20th, 2020

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.
- 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 — 2 points

Explain what is a *symbol table*. In particular, you answer should address the following points:

1. explain how the symbol table is used during the different phases of the compiling processes;
2. explain why we sometimes need to store a *tree* of symbol tables;
3. explain why we might want to store all the keywords of the language in the symbol table, and what are the benefits of doing so.

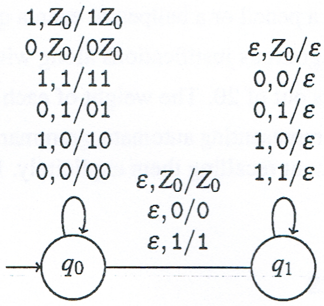
Use examples to support your answers whenever you find it appropriate!

Question 2 — 4 points

1. Give the algorithm that transforms an ε-NFA into an equivalent DFA (accepting the same language).
2. We want to build a DFA for the following language L: 'The set of all words on the alphabet $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$ in which, every time there is an a at position i, there is no b at position $i+2$ (if this position exists)'. For example, the words aaaa, aacabc and ababa are in L, while abcabb and aaaab are not. To obtain such a DFA, apply the following steps:
   (a) build an NFA that accepts the *complement* of L (it is much easier to build an NFA than a DFA to accept the complement of L, so try and exploit non-determinism);
   (b) turn this NFA into an equivalent DFA using the procedure described above; and
   (c) compute the complement of this DFA.

Question 3 — 4 points

1. Given a PDA $P = \langle Q,\Sigma,\Gamma,\delta,q_0,Z_0,F\rangle$, define the following notions: (i) *configuration* of a PDA; (ii) *configuration change* of a PDA; and (iii) the *accepted language by empty stack*, also denoted $N(P)$.
2. Then, consider the following PDA P on the alphabet $\Sigma = \{0,1\}$—**Pay attention!** although this PDA looks very similar to one we have seen at the course, it is actually different, so make sure you read all the details!



then,
   (a) give a description (not necessarily formal) of the empty stack $N(P)$ language accepted by this PDA;
   (b) for each of the three following words, determine whether the above PDA accepts it (by empty stack). When the PDA *does* accept the word, prove it using the definitions you have given above (i.e., give an accepting sequence of configurations for the word): (i) 0110, (ii) 001101, (iii) 01110.
   (c) say whether the language $N(P)$ of this PDA: (i) is context-sensitive; (ii) is context-free; (iii) is regular. Make sure you justify your answers as formally as possible!

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Question 4 — 2 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 $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$), as well as the action table of the LR(0) parser:

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