Theoretical Computer Science Exam, February 8th, 2022

The exam consists of **<u>4 exercises</u>**. Available time: 1 hr 30 min.; you may write your answers in Italian, if you wish.

All exercises must be addressed for the written exam to be considered sufficient.

F.NAME …………………………….. L.NAME …………………………………… PERSON CODE …………………

Exercise 1 (8 pts)

1. Consider the following language

$$L_1 = \{ a^m b^n \mid m > 0 \ \wedge \ n > 0 \ \wedge \ n > m \ \wedge \ n - m \text{ is even } \}$$

For instance, strings *abbb*, *aabbbb, abbbbb* $\in L_1$, while ε, *abb*, *aaabbb* $\notin L_1$.

a) Design an automaton, of the least powerful type, that accepts language L_1.

b) Write a computation of the automaton accepting string *aabbbb*

c) Write a grammar, of the least general type, that generates language L_1.

d) For the grammar generating L_1 write a derivation of the string *aabbbb*

2. Next, consider the language

$$L_2 = \{ a^m b^n \mid m > 0 \ \wedge \ n > 0 \ \wedge \ m > n \ \wedge \ m - n \text{ is even } \}$$

For instance, strings *aaab*, *aaaabb, aaaaab* $\in L_2$, while ε, *aab*, *aaabbb* $\notin L_2$.

Consider the language $L_3 = L_1 \cap L_2$

a) Design an automaton, of the least powerful type, that accepts language L_3.

b) Write a grammar, of the least general type, that generates language L_3.

Exercise 2 (8 pts)

Consider the following languages L_1 and L_2 over the alphabet $\{a, b, c\}$, defined as follows.

L_1 includes the strings where there is at least one pair of a's that are separated only by letters b. For instance, *bcabbbabbcba* $\in L_1$, while *cabcabbcab* $\notin L_1$.

L_2 includes the strings where all pairs of a's are separated by an odd number of letters, and at least one of these is a c. For instance, *bacbbacac* $\in L_2$, while *babcbabac* $\notin L_2$.

a) Write a sentence, in a monadic logic over words, first-order or second-order only if necessary, that characterizes the language L_1.

b) Write a sentence, in a monadic logic over words, first-order or second-order only if necessary, that characterizes the language L_2.

Exercise 3 **(8 pts)**

Consider the following two sets S_1 and S_2.

$$S_1 = \{\, k \mid \forall x \; (f_k(x) = \perp \;\vee\; f_k(x) > x) \,\}$$

$$S_2 = \{\, k \mid \exists x \, \exists h \, (h > 0 \;\wedge\; f_k(x) = x - h) \,\}$$

Answer the following questions, providing precise, clear justifications.

a) Is set S_1 decidable? Is it semidecidable?

b) Is set S_2 decidable? Is it semidecidable?

c) Is set $S_1 \cap S_2$ decidable? Is it semidecidable?

Exercise 4 **(8 pts)**

Given a string x over a k-element alphabet $\Sigma = \{a_1, \dots a_k\}$ sketch (that is, describe informally but precisely) a multitape Turing machine that, taking x as input, determines which is the symbol of the alphabet that occurs most times in x. In case different symbols occur a maximal number of times, then the machine indicates arbitrarily one of them. For instance, in the case when $\Sigma = \{ a, b, c\}$ and $x = cbcbaabcb$ the machine indicates b as the symbol occurring most times; if x = $cbcbaabc$ the machine can indicate either b or c. You can choose the number of tapes of the Turing machine.

Analyze the time and space complexity of the Turing machine as a function of the length of the input string $n = |x|$.

Then sketch (that is, describe informally but precisely) a RAM machine that performs the same computation and analyze its time and space complexity, still as a function of the length of the input string $n = |x|$, using both the uniform and the logarithmic cost criterion.

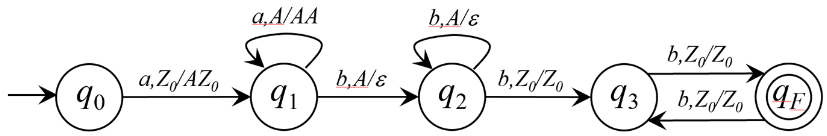
Please provide suitable justifications for the results of your analysis.

Solutions

Exercise 1

1.

- Design an automaton, of the least powerful type, that accepts language L_1.



- Write a computation of the automaton accepting string *aabbbb*

$<q_0, aabbbb, Z_0>$ |- $< q_1, abbbb, AZ_0>$ |- $< q_1, bbbbd, AAZ_0>$ |- $< q_2, bbb, AZ_0>$ |- $< q_2, bd, Z_0>$ |- $< q_3, b, Z_0>$ |- $< q_F, \varepsilon, Z_0>$

- Write a grammar, of the least general type, that generates language L_1.

  $S \rightarrow a\ S\ b \mid a\ B\ b$

  $B \rightarrow b\ b\ B \mid b\ b$

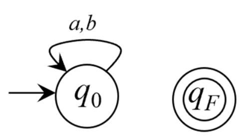
- For the grammar generating L_1 write a derivation of the string *aabbbb*

  $\mathrm{S} \Rightarrow a\ S\ b \Rightarrow a\ a\ B\ b\ b \Rightarrow a\ a\ b\ b\ b\ b$

2. $L_3 = L_1 \cap L_2$ is empty, because the two languages L_1 and L_2 are disjoint.

- Design an automaton, of the least powerful type, that accepts language L_3.

  The following simple finite state automaton accepts the empty language.



- Write a grammar that generates language L_3.

  The following simple regular grammar generates the empty language.

  $S \rightarrow a\ S \mid b\ S$

Exercise 2

L_1 includes the strings where there is at least one pair of a's that are separated only by letters b. For instance, $bcabbbabbcba \in L_1$, while $cabcabbcab \notin L_1$.

Write a sentence, in a monadic logic over words, first-order or second-order only if necessary, that characterizes the language L_1.

$$\exists x\, \exists y\, (\, x < y \wedge a(x) \wedge a(y) \wedge \;\; \forall z\, (\, x < z < y \;\rightarrow\; b(z)\,)\,)$$

L_2 includes the strings where all pairs of a's are separated by an odd number of letters, and at least one of these is a c. For instance, $bacbbacac \in L_2$, while $babcbabac \notin L_2$.

Write a sentence, in a monadic logic over words, first-order or second-order only if necessary, that characterizes the language L_2.

$$\begin{aligned}
&\forall x\, \forall y\, (\;\; x < y \wedge a(x) \wedge a(y) \;\;\rightarrow \\
&\qquad\qquad \exists X\;\; (\, X(x) \wedge X(y) \wedge \forall z\, (\, \neg\, last(z) \rightarrow (\, X(z) \leftrightarrow \;\neg\, X(z{+}1)\,)\;\;\;)\;\;)\;\wedge \\
&\qquad\qquad \exists v\, (\, x < v < y \;\wedge\; c(v) \;\;\;) \\
&\qquad)
\end{aligned}$$

Exercise 3

Consider the following two sets S_1 and S_2.

$$S_1 = \{ k \mid \forall x \ (f_k(x) = \perp \ \vee \ f_k(x) > x) \}$$

$$S_2 = \{ k \mid \exists x \, \exists h (\ h > 0 \ \wedge \ f_k(x) = x - h) \}$$

- Is set S_1 decidable? Is it semidecidable?

It is not decidable, as a consequence of the Rice theorem. It is not even semidecidable, because the complement set,

$$\neg S_1 = \{ k \mid \exists x \ (f_k(x) \neq \perp \ \wedge \ f_k(x) \leq x) \}$$

is semidecidable, by means of (dovetailing) simulation.

- Is set S_2 decidable? Is it semidecidable?

It is not decidable, as a consequence of the Rice theorem. It is semidecidable, by means of (dovetailing) simulation.

- Is set $S_1 \cap S_2$ decidable? Is it semidecidable?

It is decidable (hence also semidecidable), because $S_1 \cap S_2 = \emptyset$, as the two sets are disjoint.

Exercise 4

The Turing machine uses k tapes. While it reads the strings it counts the number of occurrence of every element of the alphabet, using one tape for each element, and storing the number in unary form. When the input string is completely read the memory heads are placed the the right of the written portion of the memory tape. Then the memory heads are moved leftwards simultaneously as long as possible. The last memory head that reaches the initial position of the tape identifies the alphabet symbol that occurs most times in the input string.

Both the time and the space complexity are $\Theta(n)$.

The RAM machine uses k counter variables to count the occurrences in x of the various alphabet elements while it reads the input string. When the input string is read the RAM performs $k - 1$ comparisons among the counters to identify the maximum value.

With the uniform cost criterion the time complexity is $\Theta(n)$ and the space complexity is constant, $\Theta(1)$. With the logarithmic cost criterion the time complexity is $\Theta(n \log n)$, because the increment operation of a value h requires time $\Theta(\log h)$; the space complexity is $\Theta(\log n)$ because a constant number k of counters of size $\Theta(\log n)$ is used.