Theoretical Computer Science Exam, January 16, 2023

The exam consists of **4 exercises**. 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)

Consider the language

$$L = \{ a^{2n} b^{2m} c^k \mid n \geq 0 \wedge m \geq 1 \wedge k \geq 1 \}$$

For instance, strings *aaaabbc*, *bbbbcc* $\in L$.

a) For the strings of language L, consider the translation $\tau_1(a^{2n} b^{2m} c^k) = a^n b^{3m} c$ and design an automaton of the least powerful type that computes such translation.

b) For the strings of language L, consider the translation $\tau_2(a^{2n} b^{2m} c^k) = b^{3m} a^n c^{\lfloor k/2 \rfloor}$; for instance, $\tau_2(aaaabbccccc) = bbbaacc$. Design the automaton of the least powerful type that computes such translation.

Exercise 2 (8 pts)

Consider a subprogram, called *checkOccurr*(*in*, *nIn*, *out*), having three parameters: an array of characters *in*, its size *nIn*, a Boolean *out*. The input parameters are: *in* and *nIn*, whereas *out* is an output parameter.

It is assumed that the number of elements of the array *in* is greater than 4, and that at most two of the elements of array *in* are equal to the character ***a***.

After execution of *checkOccurr* the output parameter *out* must be true if and only if

- the array *in* includes two distinct elements equal to character *a*, and
- the segment of the array *in* included within the two elements equal to the character *a* either
  - includes at least one element equal to character *b*, or
  - is composed entirely of characters ***c*** or is composed entirely of characters ***d***

1. Write pre- and post-conditions specifying the above described requirements for subprogram *checkOccurr*. Please assume that the array indices range from 1 to *nIn*.
2. For each of the following input/output data say if the procedure *checkOccurr* satisfies the specification if it computes the given value of the *out* parameter when it receives the given input parameter values; please provide suitable justifications for your answers.
   a) *nIn* = 6, *in* = [*b, a, c, b, c, a*], *out* = true
   b) *nIn* = 6, *in* = [*b, a, c, c, a, a*], *out* = false
   c) *nIn* = 6, *in* = [*b, a, b, b, a, b*], *out* = true
   d) *nIn* = 6, *in* = [*b, a, c, d, a, b*], *out* = true

Exercise 3 (8 pts)

Consider the set of prime numbers sorted in increasing order, so that one can refer to the first, the second, … the i-th prime number.

Please answer the following questions; provide suitable concise justifications for your answers.

1. Is the set S1 = $\{ <x, y> \mid y$ is the x-th prime number $\}$ decidable? Is it semidecidable?
2. Is the set S2 = $\{ z \mid \forall x \, \forall y \, (f_z(x, y) = 1$ if y is the x-th prime number; $f_z(x, y) = 0$ otherwise $) \}$ decidable?
3. Is the set S3 = $\{ z \mid \exists x \, \exists y \, (f_z(x, y) = 1$ if y is the x-th prime number ; $f_z(x, y) = 0$ otherwise $) \}$ decidable? Is it semidecidable?
4. Is the set S4 = $\{ <x, y> \mid \exists z \, (f_z(x, y) = 1$ if y is the x-th prime number; $f_z(x, y) = 0$ otherwise $) \}$ decidable? Is it semidecidable?

Exercise 4 (8 pts)

Consider the following C-like program, and assume that it reads from the input a ***positive*** value for variable n.

```
int n, i, s, k;
read(n);
i = 1;   s = 1;
while (i < n) {
        i = i + 1;
        k = i * i;
        s = s + k;
}
```

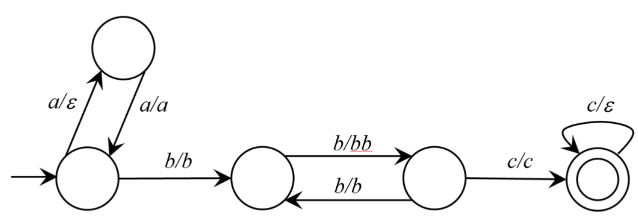
Assuming that the algorithm, after a standard translation, is executed on a RAM machine, evaluate its time and space complexity classes as a function of the input value n, using both the uniform and the logarithmic cost criteria.

Solutions

Exercise 1

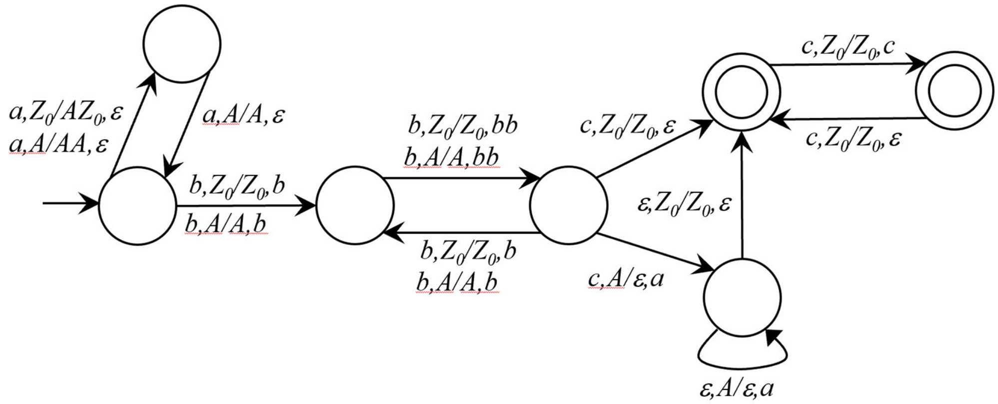
a) For the strings of language L, consider the translation $\tau_1(a^{2n} b^{2m} c^k) = a^n b^{3m} c$ and design the automaton of the least powerful type that computes such translation.

A finite state transducer suffices.



b) For the strings of language L, consider the translation $\tau_2(a^{2n} b^{2m} c^k) = b^{3m} a^n c^{\lfloor k/2 \rfloor}$; for instance, $\tau_2(aaaabbccc) = bbbaac$. Design the automaton of the least powerful type that computes such translation.

A pushdown transducer is necessary, because the b's must be output before the a's.



Exercise 2

1. Write pre- and post-conditions specifying the above described requirements for subprogram *checkOccurr.* Please assume that the array indices range from 1 to *nIn*.

pre-condition:

$nIn > 4 \;\wedge \neg\, \exists i\, \exists j\, \exists k\, (\; i \neq j \;\wedge\; i \neq k \;\wedge\; j \neq k \;\wedge\;\; in\,[i] = in\,[j] = in\,[k] = a\;)$

post-condition:

$$
\begin{aligned}
out \leftrightarrow\;\; \exists i\, \exists j\, \big(\; & 1 \leq i < j \leq nIn \;\;\wedge in\,[i] = in\,[j] = \; a \;\wedge \\
& (\;\; \exists p\,(\, i < p < j \wedge\; in\,[p] = \; b \;) \;\;\vee \\
& \quad \forall h\,(\, i < h < j \;\rightarrow\; in\,[h] = \; c \;\;) \;\;\vee \\
& \quad \forall k\,(\, i < k < j \;\rightarrow\; in\,[k] = \; d \;\;) \\
&) \\
\big)
\end{aligned}
$$

2. for each of the following input/output data say if the procedure *checkOccurr* satisfies the specification if it computes the given value of the *out* parameter when it receives the given input parameter values; please provide suitable justifications for your answers.

a) $nIn = 6$, $in = [b, a, c, b, c, a]$, $out =$ true

the specification is satisfied, because the precondition is true, and the postcondition is true

b) $nIn = 6$, $in = [b, a, c, c, a, a]$, $out =$ false

the specification is satisfied, because the precondition is false, and the postcondition is false

c) $nIn = 6$, $in = [b, a, b, b, a, b]$, $out =$ true

the specification is satisfied, because the precondition is true, and the postcondition is true

d) $nIn = 6$, $in = [b, a, c, d, a, b]$, $out =$ true

the specification is not satisfied, because the precondition is true, and the postcondition is false

Exercise 3

1. Is the set S1 = { $<x, y>$ | y is the x-th prime number } decidable? Is it semidecidable?

S1 is decidable (and hence semidecidable); it is possible to write a procedure that, given any pair of numbers $<x, y>$, determines is y is the x-th prime number.

2. Is the set S2 = { z | $\forall x\ \forall y$ ($f_z(x, y) = 1$ if y is the x-th prime number; $f_z(x, y) = 0$ otherwise) } decidable?

S2 is the set of indices (Goedel numbers) of the Turing machines that compute a function like the one mentioned in the answer concerning set S1. S2 satisfies the hypotheses of the Rice theorem, therefore it is not decidable.

3. Is the set S3 = { z | $\exists x\ \exists y$ ($f_z(x, y) = 1$ if y is the x-th prime number ; $f_z(x, y) = 0$ otherwise) } decidable? Is it semidecidable?

S3 is the set of indices (Goedel numbers) of the Turing machines that, for at least given one pair of numbers $<x, y>$, determines if y is the x-th prime number. S3 satisfies the hypotheses of the Rice theorem, therefore it is not decidable. S3 is semidecidable, by means of dovetail simulation (given a value z simulate the execution of f_z repeatedly for all possible arguments and for an increasing number of steps, etc. …).

4. Is the set S4 = { $<x, y>$ | $\exists z$ ($f_z(x, y) = 1$ if y is the x-th prime number; $f_z(x, y) = 0$ otherwise) } decidable? Is it semidecidable?

S4 is the set of all pairs of integers, because the computable function f_z mentioned in its definition exists: it is the one mentioned in the answer concerning set S1. Therefore S4 is decidable.

Exercise 4

Let us assume that the variables of this program are stored in the RAM memory as follows

n $\Rightarrow$ M[1]
i $\Rightarrow$ M[2]
s $\Rightarrow$ M[3]
k $\Rightarrow$ M[4]

The program computes the sum of the first n integer square numbers. This sum is notoriously $\Theta(n^3)$.

With the constant cost criterion the time complexity class is $\Theta(n)$, because the loop is executed n times; the space complexity class is $\Theta(1)$, because a constant amount of memory is used.

With the logarithmic cost criterion, the space complexity class is $\Theta(\log n^3) = \Theta(\log n)$ because a value $\Theta(n^3)$ must be stored and no greater value is ever computed nor stored.

Concerning the time complexity class, the dominating factor in the complexity estimation derives from the execution of the code fragment **`s = s + k`**. This will be translated into the following sequence of RAM instructions.

```
LOAD 3
ADD 4
STORE 3
```

At the i-th iteration of the loop these RAM instructions compute in M[0] and store in M[3] a value that is $\Theta(i^3)$

The overall cost is therefore $T(n) \approx \sum_{i=1}^{n}(\log i^3)$, which is $\Theta(n \log n)$.