Theoretical Computer Science Exam, June 27th, 2017

The exam consists of **4 exercises**. Available time: 1 hr 20 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 …….. MATRICOLA ……

Exercise 1 (10 pts)

Consider the language $L = \{ x \in \{a, b\}^* \mid |x| \text{ is odd and the first, middle and last character of } x \text{ are equal} \}$

1. Write a context-free grammar that generates L
2. Design a push down automaton that accepts L
3. Argue that L is not regular

Exercise 2 (8 pts)

A procedure *sumArr*(a, b, *dim*, p) has 3 input parameters a, *dim*, p, and one output parameter b. Parameters a and b are arrays of integers having *dim* elements, and it can be assumed that *dim*≥3, while parameter p can be assumed to be equal to either 0 or 1.

If p=0 then after the execution of *sumArr* the first two elements of b are equal to the elements of a in the same positions, and every element from the third position on

- is equal to the sum of the two preceding elements if the second preceding element is greater or equal to the immediately preceding element;
- is equal to the product of the two preceding elements if the second preceding element is less than the immediately preceding element.

If p=1 then after the execution of *sumArr* every element of b is equal to the sum all elements of a having position less than or equal to it.

For instance, if a=[7 3 5 6 6 9 5], *dim*=7, p=0 then b=[7 3 10 15 30 12 54], while

if a=[7 3 5 6 6 9 5], *dim*=7, p=1 then b=[7 10 15 21 27 36 41].

Specify the procedure *sumArr* by means of a pre-condition and a post-conditions expressed in first-order logic with the usual relational and arithmetic operators typical of common programming languages (NB: no use of the summation operator like "$\sum_{\cdots}^{\cdots}$..." is allowed).

Exercise 3 (8 pts)

Let M be a generic Turing machine, considered as a language acceptor, so that $L(M)$ denotes the language accepted by M. Answer the following questions, providing concise justifications.

1. Is the set $S_1 = \{ M \mid |L(M)| = 13 \}$ decidable?
2. Is the set $S_2 = \{ M \mid |L(M)| \geq 13 \}$ decidable? Is it semidecidable?
3. Is the set $S_3 = \{ M \mid |L(M)| \leq 13 \}$ decidable? Is it semidecidable?
4. Is the set $S_4 = \{ \langle M, x \rangle \mid x \text{ is accepted by } M \text{ in } \leq 13 \text{ steps} \}$ decidable? Is it semidecidable?

Exercise 4 (8 pts)

Consider the language $L=\{a^n b^m c^{n \times m} \mid n, m>0 \}$

1. design, preferably in full detail, or at least sketch in a precise and unambiguous way, a Turing machine that accepts L, and evaluate its time and space complexity;
2. design, preferably in full detail, or at least sketch in a precise and unambiguous way, a RAM machine that accepts L, and evaluate its time and space complexity both with reference to the uniform and the logarithmic cost criterion;
3. determine which of the two above computing models (TM and RAM) is more efficient, considering also the logarithmic cost criterion. Provide justifications for your answers.

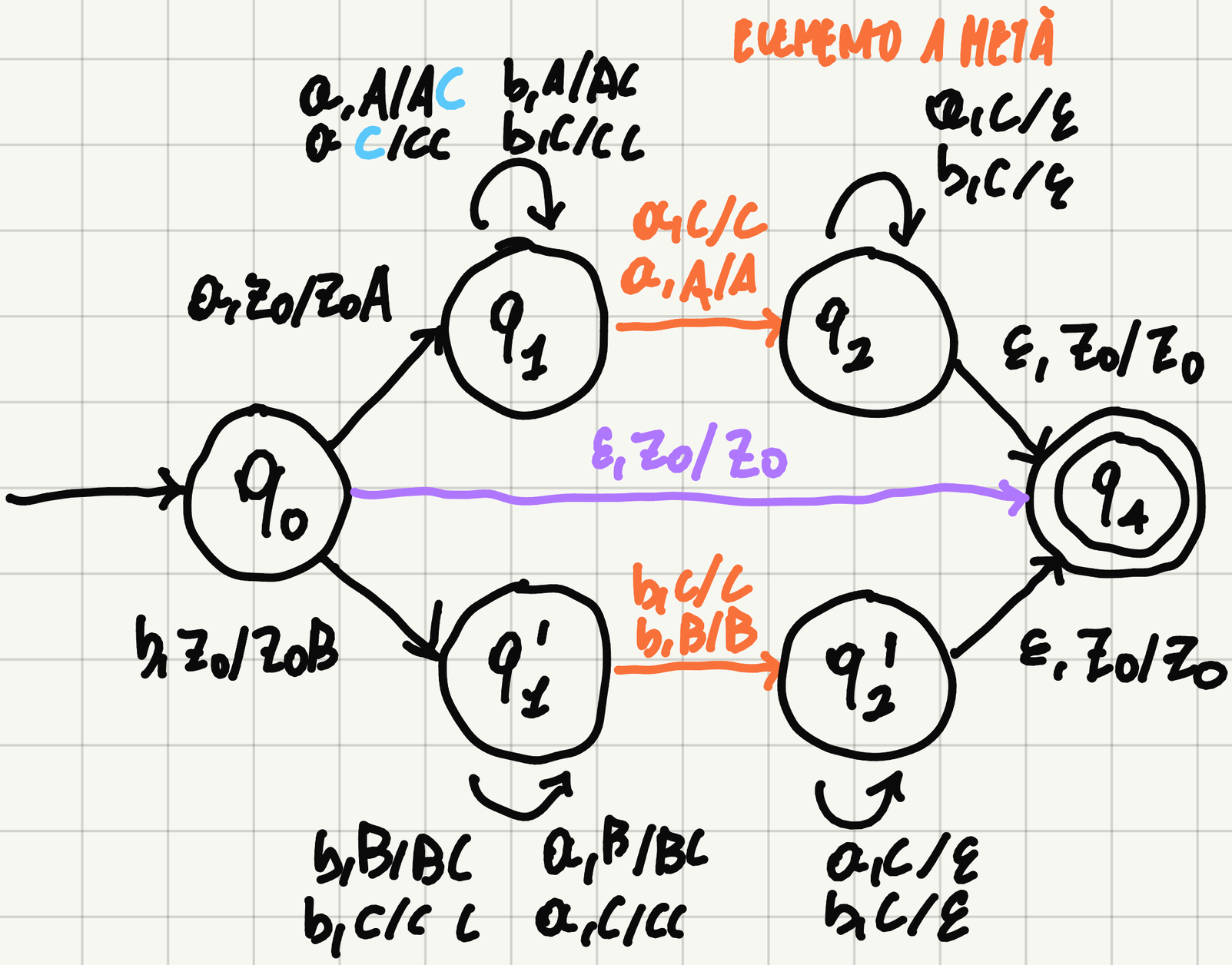
1

a) $S \to aS_1a \mid bS_2b \mid a \mid b$! NO È POSSIBILE IXI DISPARI

$S_1 \to aS_2a \mid bS_1a \mid aS_2b \mid bS_1b \mid a$

$S_2 \to aS_2a \mid bS_2a \mid aS_2b \mid bS_2b \mid b$

b) LE CONDIZIONI NON NOTE CI OBBLIGANO AD USARE UN AUTOMA NON DETERMINISTICO



C : ELEMENTO GENERICO

CASO STRINGA VUOTA

c) SE L FOSSE REGOLARE, UN AUTOMA A STATI FINITI SAREBBE SUFFICIENTE. TUTTAVIA, QUESTO NON SAREBBE IN GRADO NÉ DI TENERE MEMORIA DEL PRIMO VALORE PER POTERLO POI RIUSARE "IN MEZZO" E, SOPRATTUTTO, COME ULTIMO

2 INPUT:

- a ARRAY DI DIMENSIONE dim
- p PARAMETRO BOOLEANO

OUTPUT:

- b ARRAY

[PRE-CONDIZIONI] $dim \geq 3 \wedge (p=0 \vee p=1)$

[POST-CONDIZIONI] $[p=0 \rightarrow (b[0]=a[0] \wedge b[1]=a[1] \wedge$

$\wedge (\forall i (2 \leq i < dim \rightarrow ((a[i-2] \geq a[i-1] \rightarrow$

$\rightarrow b[i]=a[i-2]+a[i-1]) \wedge (a[i-2] < a[i-1]$

$\rightarrow b[i]=b[i-2] \cdot b[i-1])))))$

$(p=1 \rightarrow (b[0]=a[0] \wedge (\forall i (1 \leq i < dim \rightarrow$

$\rightarrow (b[i]=a[i] \cdot b[i-1])))))$

3

S_1 L'INSIEME È NON DECIDIBILE PER RICE THEOREM. INFATTI, SONO CONSIDERATE SOLO LE MACCHINE DI TURING CHE ACCETTANO LINGUAGGI LA CUI LUNGHEZZA DELLA STRINGA È PARI A 13.

QUESTO NON È NE L'INSIEME VUOTO E NE L'INSIEME DI TUTTE LE FUNZIONI COMPUTABILI

S_2 L'INSIEME È NON DECIDIBILE PER RICE THEOREM. INFATTI, SONO CONSIDERATE SOLO LE MACCHINE DI TURING CHE ACCETTANO LINGUAGGI LA CUI LUNGHEZZA DELLA STRINGA È ≥ 13.

QUESTO NON È NE L'INSIEME VUOTO E NE L'INSIEME DI TUTTE LE FUNZIONI COMPUTABILI. È PERÒ SEMIDECIDIBILE: APPLICANDO LA SIMULAZIONE DOVETAILING, CON UN NUMERO FINITO DI STEP AVREMO LA SITUAZIONE $|L(M)| \geq 13$ COME DESIDERATA

S_3 L'INSIEME È NON DECIDIBILE PER RICE THEOREM. INFATTI, SONO CONSIDERATE SOLO LE MACCHINE DI TURING CHE ACCETTANO LINGUAGGI LA CUI LUNGHEZZA DELLA STRINGA È < 13. NON È NEPPURE SEMIDECIDIBILE: INFATTI, STUDIANDO IL COMPLEMENTARE $\neg S_4$: $\{M \mid |L(M)| > 13\}$, ESSO È SEMIDECIDIBILE PER CONSIDERAZIONI

ANALOGHE AD $S_2 \rightarrow \neg S_2$ SEMIDECIDIBILE $\rightarrow S_2$ NO

S_4 L'INSIEME È DECIDIBILE. INFATTI, CI SI CHIEDE SE UNA x PUÒ ESSERE ACCETTATA DA UNA MACCHINA DI TURING IN NON PIÙ DI 13 STEP. SI TRATTA DI UNA DOMANDA "CHIUSA" CON RISPOSTA VERA (SÌ/NO).

ESSENDO DECIDIBILE, È ANCHE SEMIDECIDIBILE

4

a OCCORRE USARE UNA MACCHINA DI TURING A DUE NASTRI, UNO PER LA LETTURA DELLE a E UNO PER LE b. TERMINATE, SI EFFETTUERÀ IL CALCOLO NECESSARIO PER STAMPARE IL GIUSTO NUMERO DI c.

b DETTA n LA LUNGHEZZA DELLA STRINGA, $T(n) = \Theta(n)$ E $S(n) = \Theta(n)$

UNA MACCHINA RAM UTILIZZA DUE CONTATORI PER MEMORIZZARE IL NUMERO DI OCCORRENZE DI a E b RISPETTIVAMENTE, UN TERZO CONTATORE CHE, CALCOLATO $n \times m$, VIENE USATO PER IL CONTEGGIO DELLE c. DETTA n LA LUNGHEZZA DELLA STRINGA:

COSTO UNIFORME

$T(n) = \Theta(n)$

$S(n) = \Theta(1)$ NUMERO FISSO DI CELLE DI MEMORIA RICHIESTE

COSTO LOGARITMICO

$T(n) = \tilde{\Theta}(n \log n)$ DETTO n IL NUMERO DI OCCORRENZE DI a_i, IL RISPETTIVO CONTATORE, INTERPRETANDO IN BINARIO IL NUMERO n, IMPIEGA UN TEMPO $\log n$ PER LA CONVERSIONE. ANALOGO PER b_i

$S(n) = \tilde{\Theta}(\log n)$ I NUMERI NEL CONTATORE VENGONO INTERPRETATI IN BINARIO, E QUINDI AGGIUNTO UN FATTORE $\log n$

c

CON UNA RAM CHE ADOPERA IL CRITERIO DI COSTO UNIFORME.

NON CI SONO DIFFERENZE PER QUANTO CONCERNE LA COMPLESSITÀ TEMPORALE, MENTRE ABBIAMO MIGLIORIE CON LA RAM RIGUARDO LA COMPLESSITÀ SPAZIALE

USANDO, INVECE, IL CRITERIO DI COSTO LOGARITMICO, ABBIAMO COMUNQUE MIGLIORIE RIGUARDO ALLA COMPLESSITÀ SPAZIALE MA PEGGIORAMENTI CON LA TEMPORALE

Theoretical Computer Science Exam, July 14th, 2017

The exam consists of **4 exercises**. Available time: 2 hr 00 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 …….. MATRICOLA ……

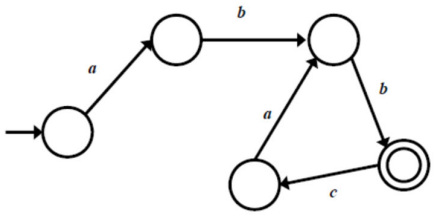
Exercise 1 (9 pts)

Given a language L over alphabet Σ, consider the following operation *pref*

$$pref(L) = \{ w \mid w \in \Sigma^* \text{ and } \exists x \in \Sigma^* \text{ such that } (wx \in L) \}$$

(notice that the result of *pref*(L), for any language L, is also a language)

Consider the regular language L_1 accepted by the following finite state automaton.



1. Describe by means of an English sentence the language *pref*(L) for a generic language L.
2. Write the first four strings of *pref*(L_1) in lexicographical order.
3. Is *pref*(L_1) a regular language? Please justify adequately you answer.
4. Is the family of regular languages closed under operation *pref* ? Please justify adequately you answer.

Exercise 2 (8 pts)

Write a monadic second-order logic formula over word structures that characterizes language L, of alphabet $\Sigma=\{a, b, c\}$ defined as follows $L=ab(bc)^+$.

Exercise 3 (8 pts)

Let us recall that a set S_1 is *reducible* (or, in other words, *it can be reduced*) to another set S_2 iff there exists a total computable function $t(x)$ such that, for every x, $x \in S_1$ if and only if $t(x) \in S_2$.

Now, in addition, we say that a set S_1 is *doubly reducible* to another set S_2 iff there exist two total computable functions $t_1(x)$ and $t_2(x)$ such that, for every x, $x \in S_1$ if and only if $t_1(x) \in S_2$ and $t_2(x) \notin S_2$.

Prove the following property, or disprove it if you think that it does not hold: if S_1 is reducible to S_2 and $S_2 \subset \mathbb{N}$ (i.e., S_2 is a proper subset of the universe set) then S_1 is doubly reducible to S_2.

Exercise 4 (8 pts)

Design, preferably in full detail, a k-tape Turing machine that accepts the language over alphabet $\Sigma=\{a, b, c\}$

$$L = \{ wcw \mid w \in \{a, b\}^+ \} \cap \{ wcw^R \mid w \in \{a, b\}^+ \}$$

with minimal time complexity, and determine as precisely as possible its time and space complexity.

1

a) IL LINGUAGGIO $pref(L)$ È RELATIVO AI PREFISSI, OSSIA AI PRIMI CARATTERI w, DI UNA STRINGA wx. x PUÒ ANCHE ESSERE LA STRINGA VUOTA

b)
1 - ε (STRINGA VUOTA)
2 - a
3 - ab
4 - abb

c) UN LINGUAGGIO SI DICE REGOLARE QUANDO PUÒ ESSERE RICONOSCIUTO DA UN AF (INDIPENDENTEMENTE DAL FATTO D' ESSERE DETERMINISTICI O NO). $Pref(L_1)$ È RICONOSCIBILE

d) DALL'AUTOMA PROPOSTO, CON "abb" IL PREFISSO

DATO UN GENERICO LINGUAGGIO REGOLARE, POSSIAMO DEFINIRE UN POSSIBILE STATO FINALE CON IN INPUT UNA CERTA STRINGA w E, COME STATI NON FINALI, EVENTUALI ALTRE STRINGHE x.

QUINDI, IN GENERALE, UN LINGUAGGIO REGOLARE È CHIUSO RISPETTO A $pref(L)$

2 $a(0) \wedge b(1) \wedge$

$\exists x ((b(x) \rightarrow c(x+1)) \wedge$

$\wedge((c(x) \wedge \neg last(x)) \rightarrow b(x+1)) \wedge$

$\wedge (last(x) \rightarrow c(x))$

3 S_1 RIDUCIBILE IN $S_2 \leftrightarrow \exists t(x) \mid \forall x, x \in S_1 \leftrightarrow t(x) \in S_2$

S_1 DOPPIAMENTE RIDUCIBILE IN $S_2 \leftrightarrow \exists t_1(x), t_2(x) \mid \forall x, x \in S_1 \leftrightarrow t_1(x) \in S_2$
$\wedge t_2(x) \in S_2$

S_1 RIDUCIBILE IN S_2, $S_2 \subset \mathbb{N} \rightarrow S_1$ DOPPIAMENTE RIDUCIBILE IN S_2

IL TEOREMA È FACILMENTE VERIFICABILE. INFATTI, SE S_1 È RIDUCIBILE IN S_2, $\exists t(x) \mid \forall x, x \in S_1 \leftrightarrow t(x) \in S_2$.

DALLA DEFINIZIONE DI "DOPPIAMENTE RIDUCIBILE", $t(x) = t_1(x)$ E.

DEFINENDO OPPORTUNAMENTE S_2 IN MODO CHE SI POSSANO DEFINIRE

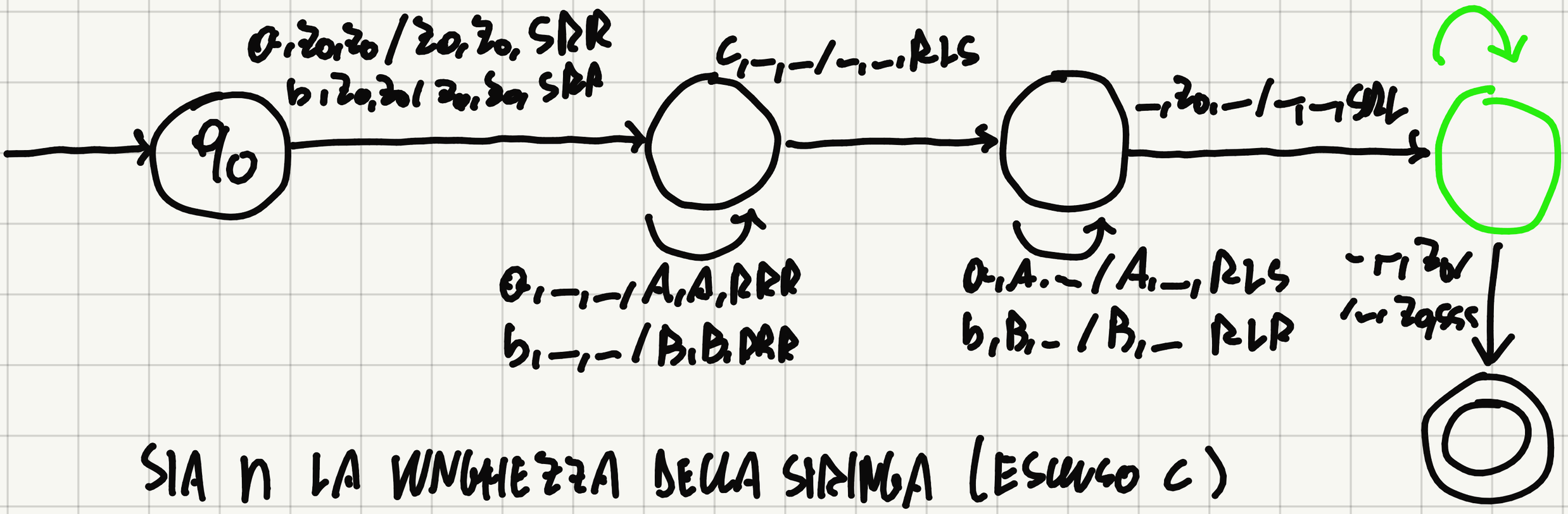
$t(x)$ COME IN PRECEDENZA E $t_2(x) \mid \forall x, x \in S_1 \rightarrow t_2(x) \notin S_2$

$\exists t(x), t_2(x) \mid \forall x, x \in S_1 \leftrightarrow t(x) \in S_2 \wedge t_2(x) \notin S_2$, CHE ALTRO

NON È CHE LA DOPPIA RIDUCIBILITÀ

4 OSSERVIAMO SUBITO CHE, ESSENDO UNA INTERSEZIONE, IL LINGUAGGIO SARÀ DEFINITO PER $w c w^R$ (PALINDROMI).

OCCORRE UNA MACCHINA DI TURING A 2 NASTRI, UNO PER LE a E UNO PER LE b. AL TERMINE, SI RIPERCORRE ALL'INDIETRO FINO A "c" PER VERIFICARE $w^R = w$



SIA n LA LUNGHEZZA DELLA STRINGA (ESCLUSO c)

$$T(n) = \underline{n+1} + \underline{\frac{1}{2}(n+1)} = \frac{3}{2}(n+1)$$

COMPLESSITÀ SPAZIALE: $S(n) = n-1$ (ESCLUSO c)

Theoretical Computer Science Exam, September 14th, 2017

The exam consists of **4 exercises**. Available time: 1 hr 45 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 …….. MATRICOLA ……

Exercise 1 (9 pts)

Design an automaton that accepts, and a grammar that generates, the languages defined below; both the automata and the grammar must be of the least powerful type.

1) L1 = { $w \in \{a, b\}^*$ | ($\#_a(w) + 2 \cdot \#_b(w)$) MOD 5 = 0 }
   where $\#_a(w)$ and $\#_b(w)$ denote the number of symbols a and b occurring in string w, and MOD denotes the integer division remainder operator. For instance, $abb \in$ L1, $\varepsilon \in$ L1, $aba \notin$ L1.
   In addition, write the computation of the automaton, and a derivation of the grammar, that respectively accept and generate the string *abb*.
2) L2 = { $a^n b^m c^k$ | $m, n, k \geq 0$ and $m = n + k$ }
   For instace, $a^2b^5c^3 \in$ L2, $\varepsilon \in$ L2, $a^2b^4c^3 \notin$ L2.

Exercise 2 (8 pts)

Preliminary part

1. Two natural numbers a and b are said to be *mutually prime* if their greatest common divisor is 1. Formalize this notion by means of a binary predicate

   mutuallyPrime(a, b)

   to be specified by means of a formula in mathematical logics that uses the customary predicates and operators of integer arithmetic on natural numbers.

2. Consider a subprogram

   computePrimes (v, n, xnp, x, y)

   where v and n are input parameters, while *xnp*, x and y are output parameters; v is an array of integer numbers having $n>1$ elements that are all distinct; parameter *xnp* is boolean and after execution it is true if and only if there exists a pair of elements in the array that are not mutually prime; if this is the case, the output parameters x and y are set equal to the positions of a pair of array elements ($v[x]$, $v[y]$) such that *mutuallyPrime*($v[x]$, $v[y]$) does not hold (otherwise x and y are undefined). For instance, if v=[3, 5, 7, 11, 8, 13] (hence n=6) then *xnp=false* and the value of x and y is undefined. As another instance, if v=[3, 5, 10, 9, 12, 8] (hence n=6) then *xnp=true*, and one may have x=2 and y=3 (we assume array indices starting from 1), corresponding to the pair of elements (5, 10) not being mutually prime.

   Specify subprogram *computePrimes* by means of pre- and post-conditions; to this end you may use the previously specified *mutuallyPrime* predicate.

3. (Optional) Modify the specification of the previous point so that parameter *xnp* is true if and only if there exists a pair of elements in the array that are not mutually prime, but in this case the output parameters x and y are set equal to the positions of a pair of array elements such that they are not mutually prime, and there does not exist any other different pair of array elements having the same property of not being mutually prime, and such that their product is greater than that of $v[x]$ and $v[y]$. For instance, if v=[3, 5, 10, 9, 12, 8] then *xnp=true*, x=3 and y=5 corresponding to the pair of elements (10, 12), because all other pairs of array elements that are not mutually prime are such that their product is less than 120=10·12 (for instance, another pair of elements that are not mutually prime is (12, 9) but their product is less than 120).

Exercise 3 (8 pts)

Preliminarily, prove that the union of two semidecidable sets, S_1 and S_2, is also semidecidable. Your argument should be as formal and precise as possible.

Prove the following property: the difference set, $D - S$, between any decidable set D and any semidecidable set S strictly included in D (i.e., such that $S \subset D$ and $S \neq D$) is *not* necessarily semidecidable.

Exercise 4 (8 pts)

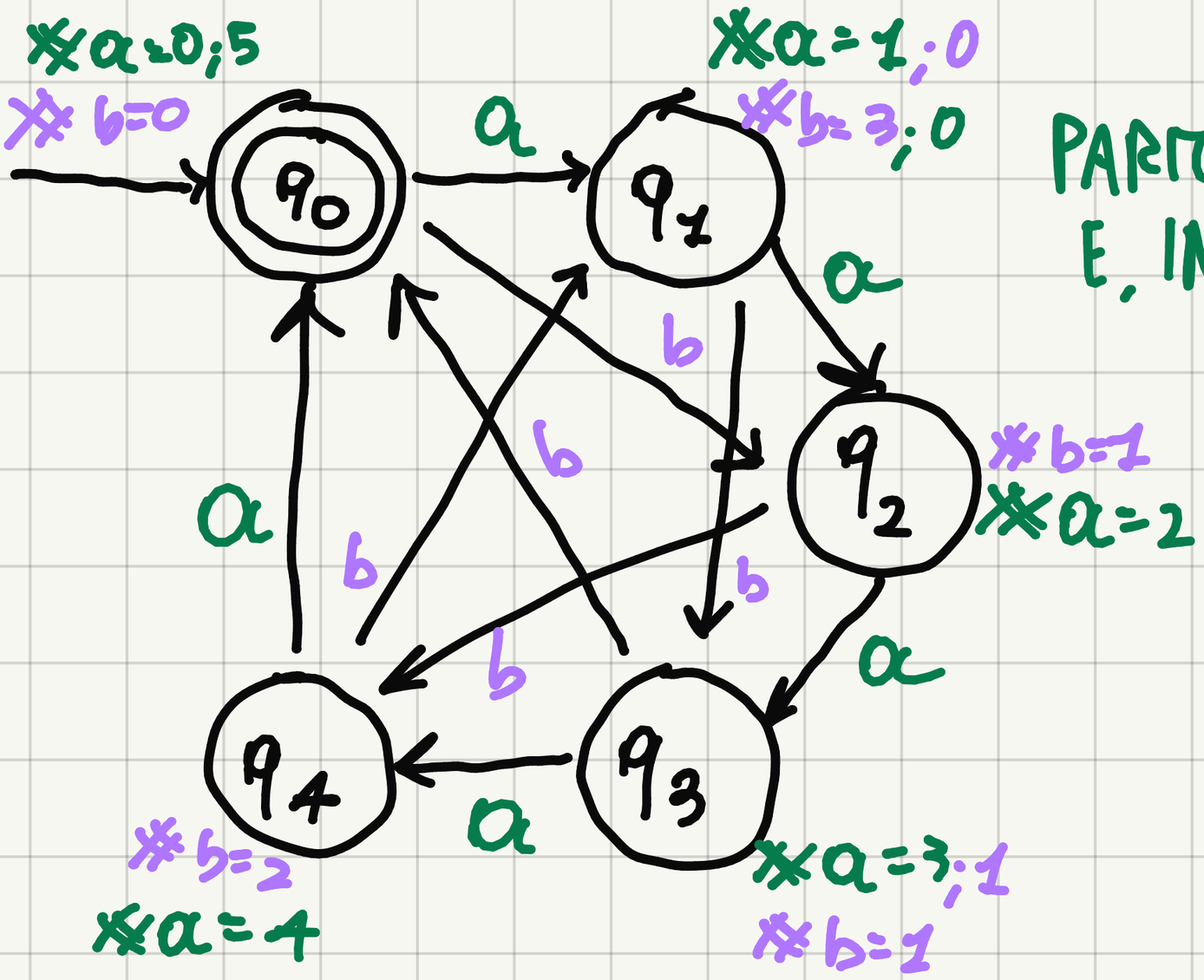
Consider language $L = \{ a^n b^m c^k \mid m, n, k \geq 1 \text{ and } k = 2 \cdot n \text{ and } m = n + k \}$.

Design a Turing machine (preferably having only one memory tape) that recognizes the language with complexity *class* $\Theta(n)$. For this machine, determine in a precise way the time complexity $T(n)$ as a *function* of the input string length.

Sketch a single tape Turing machine that recognizes the language with a minimal complexity class, providing an adequate justification for your reply.

1

a) AUTOMA: È NECESSARIO UN AUTOMA A STATI FINITI



PARTO LAVORANDO CON LE a E, IN SEGUITO, REGOLO LE b

abb:

$q_0 \to a \to q_1 \to$

$b \to q_3 \to b \to q_5$

GRAMMATICA:

$S \to \varepsilon$

$S \to aA \mid bB$

$A \to aB \mid bC$

$B \to aC \mid bD$

$C \to aD \mid bS$

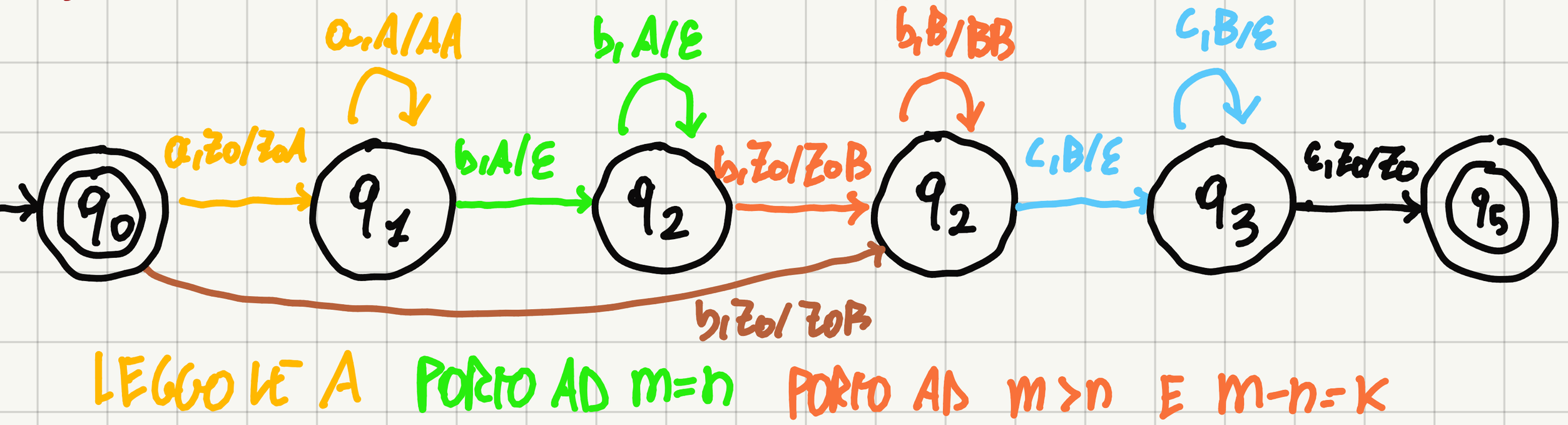
$D \to aS \mid bA$

SI NOTI CHE $S \to q_0$, $A \to q_1$, $B \to q_2$...

abb:

$S \to aA \to abC \to abbS \to abb$

b) AUTOMA: LE CONDIZIONI SUGLI ESPONENTI RICHIEDONO UN AUTOMA A PILA



LEGGO LE A PORTO AD m=n PORTO AD m>n E m-n=k

K MOSSE PER LEGGERE C CASO n=0

$a^n b^m c^k \quad m,n,k \geq 0$
$m = n+k$

GRAMMATICA:

$S \to A \cdot B$

$A \to aAb \mid \varepsilon$

CON A GESTISCO FINO m=n, CON B AGGIUNGO LE K

$B \to bBc \mid \varepsilon$

2

1 mutuallyPrime(a,b) $\leftrightarrow \neg\exists k,h,d\,(d>1 \wedge a=d\cdot k \wedge$

$\wedge\ b=k\cdot h)$

computePrimes(v, n, xnp, x, y)

INPUT:

2

- v ARRAY DI DIMENSIONE n
- n INTERO >1

OUTPUT:

- xnp BOOLEANO

PRE-CONDIZIONI] $(n>1) \wedge \forall i,j\,(1\le i<n \wedge 1\le j<n \wedge i\ne j \rightarrow v[i]\ne v[j])$

POST-CONDIZIONI] $(xnp=1) \leftrightarrow (1\le i<j<n \wedge \neg$mutuallyPrime$(v[x],v[y]))$

3

POSTCONDIZIONI ALTERNATIVE] $(xnp=T) \leftrightarrow (1\le i<j<n \wedge \neg$mutuallyPrime$(v[x],v[y])$

$\wedge \neg\exists w,z\,(1\le w<z<n \wedge ((w\ne i \wedge z\ne j) \vee$

$\vee (w\ne j \wedge z\ne i)) \wedge$

$\neg$mutuallyPrime$(v[w],v[z]) \wedge$

$v[w]\cdot v[z] > v[i]\cdot v[j])$

3 a

SIANO S_1 E S_2 DUE SET SEMIDECIDIBILI

$S_1 = \{x \mid \exists y, y \in \mathbb{N} \wedge x = g_{S_1}(y)\}$

$S_2 = \{x \mid \exists y, y \in \mathbb{N} \wedge x = g_{S_2}(y)\}$

CON $g_{S_1}(y)$ E $g_{S_2}(y)$ FUNZIONI TOTALI E COMPUTABILI.

DEFINIAMO $g_S(y) = g_{S_1}(y) \cup g_{S_2}(y)$ COME UNA FUNZIONE CHE "UNISCE" LE DUE. ESSA RISULTA TOTALE E COMPUTABILE, PER CUI

$S = S_1 \cup S_2 = \{x \mid \exists y, y \in \mathbb{N} \wedge (x = g_{S_1}(y) \vee x = g_{S_2}(y))\}$

→ S È SEMIDECIDIBILE

b

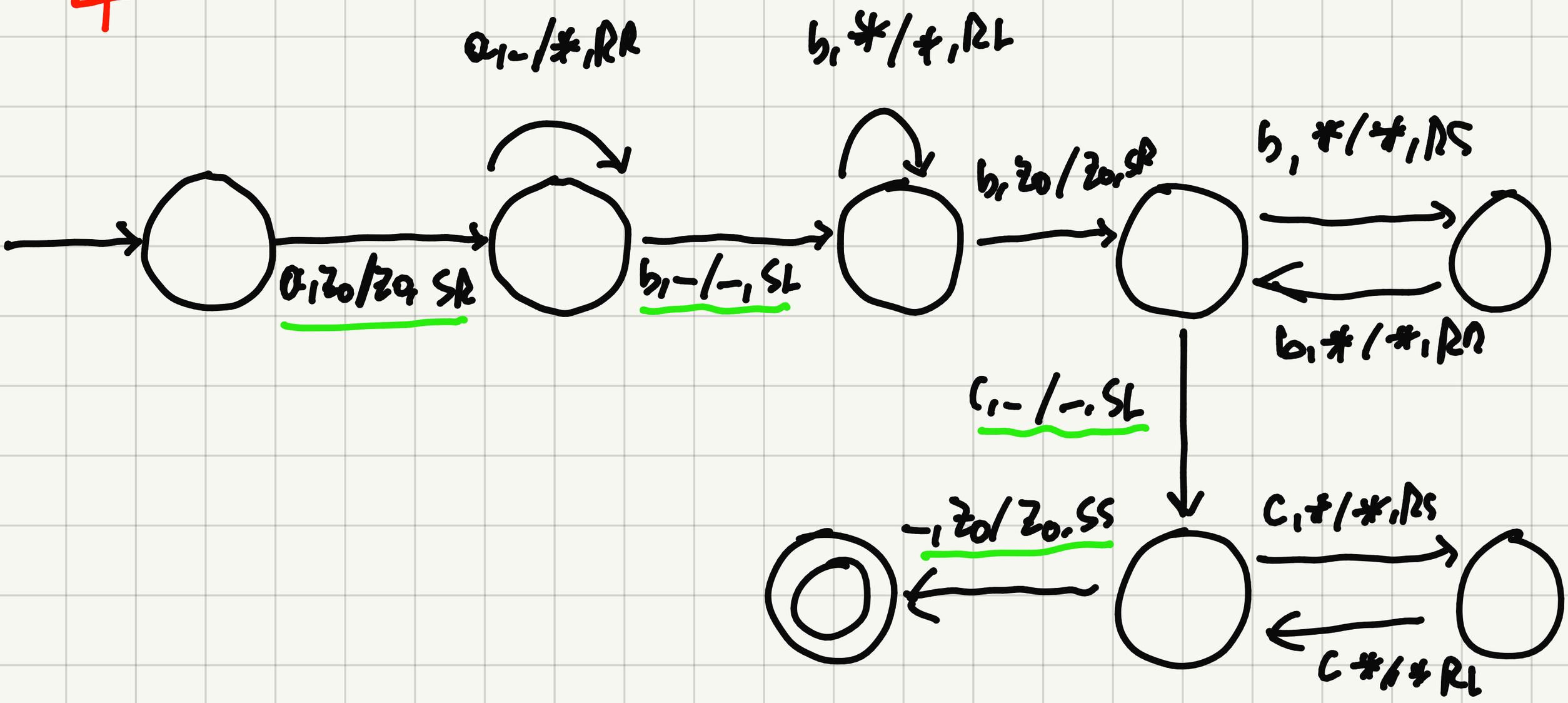
SIA D DECIDIBILE E SCD, $S \neq D$, SEMIDECIDIBILE. D-S?

DEFINIAMO $\bar{D} = U - D$. SE D È DECIDIBILE, LO È ANCHE $\bar{D}$

SUPPONENDO D-S SEMIDECIDIBILE, LO SAREBBE ANCHE $(D-S) \cup \bar{D}$ IN QUANTO UNIONE TRA DUE SEMIDECIDIBILI. QUESTO NON È ALTRO CHE $\bar{S}$.

VISTO CHE NON SAPPIAMO NULLA SULLA DECIDIBILITÀ, SOLAMENTE UNO TRA S E $\bar{S}$ (IN GENERALE) È SEMIDECIDIBILE, PER CUI NON POSSIAMO DARE RISULTATI DIRETTI PER D-S

4



$T(n) = n + 4$

UNA MT CON CLASSE DI COMPLESSITÀ MINIMA DIPENDE DA $\Theta(n^2)$,

E DÈ IL CASO IN CUI L'AUTOMA VA CONTINUAMENTE AVANTI ED INDIETRO PER

LA GESTIONE DI b E c

Theoretical Computer Science Exam, June 18th - July 2nd, 2018

The exam consists of **4 exercises**. Available time: 2 hr 00 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 …….. MATRICOLA ……

I have passed the midterm exam and I intend to skip the first part (Ex.1-2) (select one): **YES NO**

Exercise 1 (7.5 pts)

Design minimal power automata accepting the languages defined as follows (as usual it is intended that, for every symbol k, the function $\#k(x)$ denotes the number of occurrences of k in the string x):

1) $L_1 = \{ x \in \{a, b, c\}^* \mid x$ contains no more than one pair of consecutive a's $\}$ For instance, $cbcaab \in L_1$, $aba \in L_1$, $cbaabcaab \notin L_1$, $cbaaab \notin L_1$.

2) $L_2 = \{ x \in \{a, b, c, d\}^* \mid \#a(x) + \#b(x) = \#c(x) + \#d(x) \}$ NB: solutions with few states and alphabet symbols are preferred.

Exercise 2 (7.5 pts)

Specify, by means of suitable pre- and post-conditions, a subprogram **inflate**(a, n, b, m) where:

1. the parameter a is an input array of positive integers having $n > 1$ elements, sorted in either increasing or decreasing order, and
2. b is an output array parameter, of size m, that also stores positive integers. (Indices in the arrays a and b start from 0)
3. the number of elements of array b is a multiple of n, i.e., it is equal to $k \cdot n$ for some $k>1$, and b stores the same elements as a, in the same order, each one multiplied by k and repeated k times; for instance, if a = [5 7 9] then a possible value for b is [15 15 15 21 21 21 27 27 27] (in this case k = 3)

As a variation of the specification above, consider one in which the first two points are unchanged, and the third point is substituted by the following ones:

3. if a is sorted in increasing order then the first k elements of b are: $a[0] \cdot k$ followed by the next immediately successive k-1 integer values, in increasing order; the successive k elements of b are $a[1] \cdot k$ followed by the next immediately successive k-1 integer values, in increasing order, etc.; for instance, if a = [5 7 9] then a possible value for b is [15 16 17 21 22 23 27 28 29] (in this case k = 3)
4. similarly, if a is sorted in decreasing order then the first k elements of b are: $a[0] \cdot k$ followed by the next immediately preceding k-1 integer values, in decreasing order; the successive k elements of b are $a[1] \cdot k$ followed by the next immediately preceding k-1 integer values, in decreasing order, etc. for instance, if a = [9 7 5] then a possible value for b is [36 35 34 33 28 27 26 25 20 19 18 17] (in this case k = 4)

Exercise 3 (7.5 pts)

Consider a semidecidable, non-empty set S (so that it admits a generating function $g_S(x)$), and a decidable set D (so that its characteristic predicate $c_D(x)$ is computable), D being strictly included in S. Prove that the difference set S - D is semidecidable. To this end, please define function $g_{S-D}(x)$, the generating function of set S - D.

How does the answer change if set D is not necessarily included in S?

How does the answer change if set S may be empty?

(Remember that the difference set of two sets A and B is defined as $A - B = \{ x \mid x \in A \wedge x \notin B \}$)

Exercise 4 (7.5 pts)

Let us recall the definition of a ***finer*** relation: a relation R' is said to be finer than another relation R iff

$$\forall x, y \, ((x \, R' \, y) \rightarrow (x \, R \, y)).$$

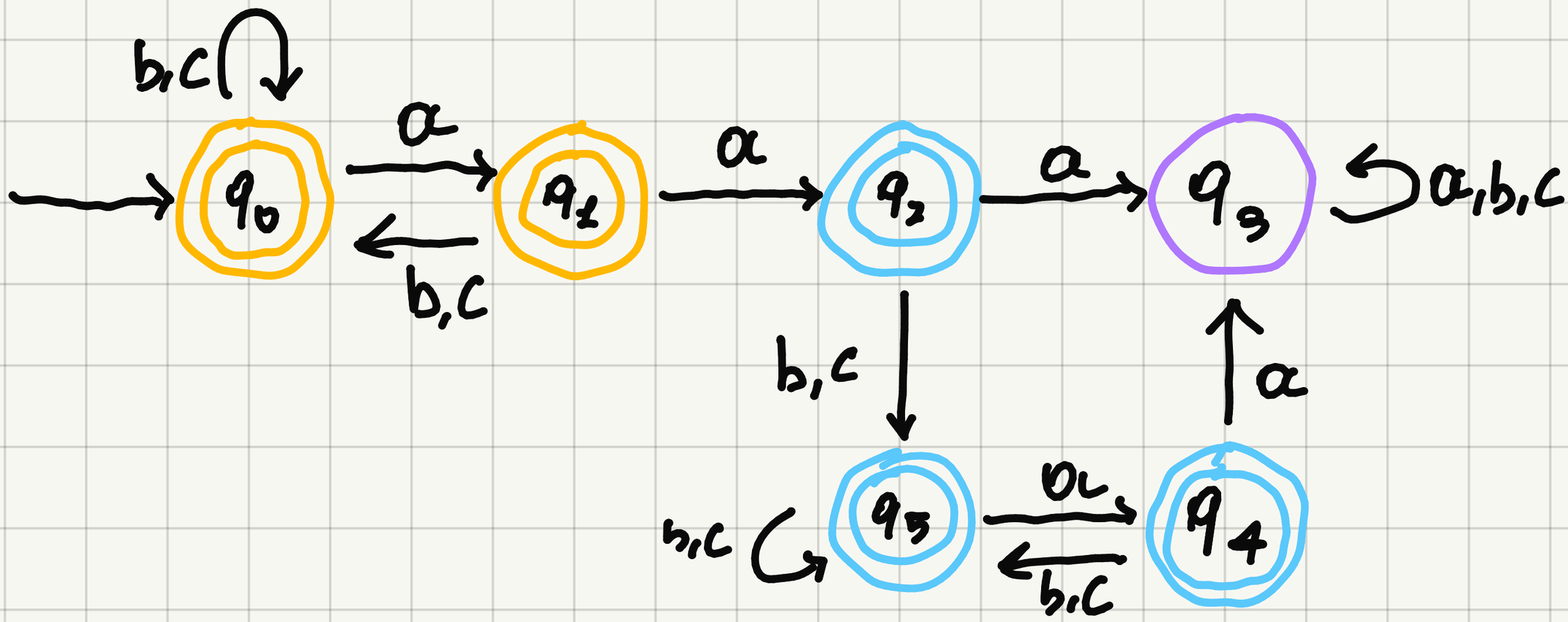
Then, consider two complexity functions $f(n)$ and $g(n)$. Function $g(n)$ is said to be an asymptotic upperbound for function $f(n)$, written in symbols as $f(n) \in \mathcal{O}(g(n))$, or equivalently ($f \, \mathcal{O} \, g$), iff there exist a positive integer k and a positive constant c such that

$$\forall n \, (\, n \geq k \rightarrow f(n) \leq \mathrm{c} \cdot g(n) \,)$$

1. Discuss, providing a suitable justification, possibly supported by simple examples, the following questions
   a. is relation Θ finer than $\mathcal{O}$?
   b. is relation $\mathcal{O}$ finer than Θ?
2. Relation $\mathcal{O}$ does not enjoy all the algebraic properties of relation Θ that are useful in the analysis of (time and space) complexity: specifically, which property of the Θ relation is not enjoyed by relation $\mathcal{O}$?

1

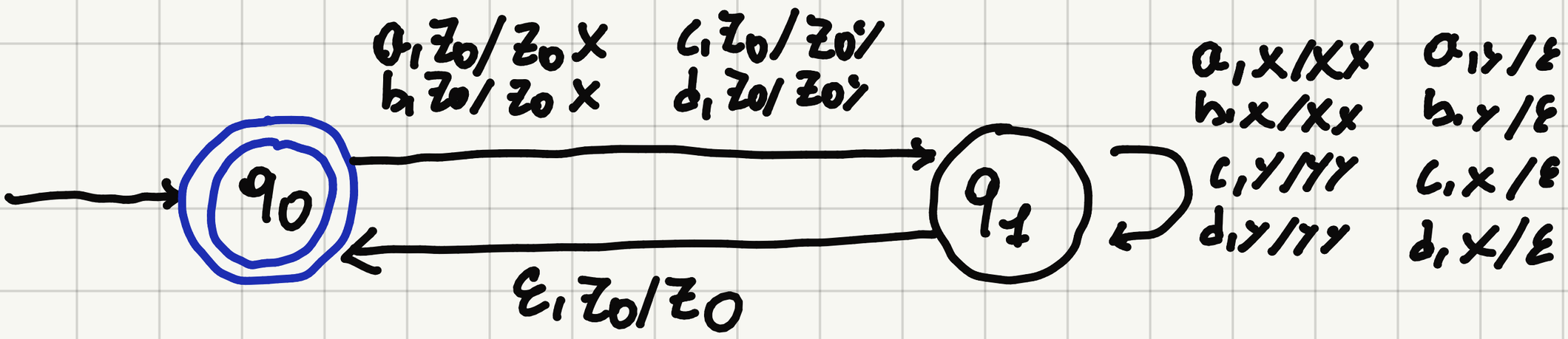
a) UN AUTOMA A STATI FINITI È SUFFICIENTE



NESSUNA COPPIA DI 0L UNA COPPIA DI 0L UNA TRIPLA/DUE COPPIE

b) LA CONDIZIONE SUL NUMERO DI OCCORRENZE POSSIAMO GESTIRLA

CON UN AUTOMA A PILA DETERMINISTICO



ALL'INIZIO, #a(x) = #b(x) = #c(x) = #d(x)

2 INPUT:

- a ARRAY DI DIMENSIONE n ORDINATO (CRESCENTE O DECRESCENTE)
- n > 1

OUTPUT:

- b ARRAY DI DIMENSIONE m
- m = k·n

PRECONDIZIONI $(n>1) \wedge \forall i\,(0 \leq i < n \rightarrow$

$\rightarrow (a[i] < a[i+1])) \vee \forall i\,(0 \leq i < n$

$\rightarrow (a[i] > a[i+1]))$

POSTCONDIZIONI $\exists k\,(m = k \cdot n \wedge \forall 0 \leq i < m \wedge \forall 0 \leq j < n$

$\rightarrow \forall (0 \leq i < k \cdot j \rightarrow b[kj+i] = a[j]$

VARIANTI

- $\exists k\,(k>1 \wedge m = k \cdot n \wedge \forall ij\,(0 \leq i < n \wedge 0 \leq j < k$

$\rightarrow (a[0] < a[1] \rightarrow b[ki+j] = k \cdot a[i]+j))$

- $\exists k\,(k>1 \wedge m = k \cdot n \wedge \forall ij\,(0 \leq i < n \wedge 0 \leq j < k$

$\rightarrow (a[0] > a[1] \rightarrow b[ki+j] = k \cdot a[i]-j))$

3 SE MIDECIDIBILE E NON VUOTO S

DECIDIBILE $D \subset S$

$g_{S-D}(x)$ $S-D$ È SICURAMENTE NON VUOTO. DEFINIAMO

UN $K \in g_{S-D}(x)$ $g_S(x) = \begin{cases} g_S(x) & C_D(g_S(x)) = 0 \\ K & \text{ALTRIMENTI} \end{cases}$ $\rightarrow$

$\rightarrow$ SEMIDECIDIBILE

VARIANTI

- SE $D \supseteq S$, ALLORA $S-D = \emptyset$, SEMIDECIDIBILE PER DEFINIZIONE
- SE S FOSSE VUOTO. ANCHE $\emptyset - D$ LO SAREBBE, E QUINDI SEMIDECIDIBILE PER DEFINIZIONE

4

a RICORDIAMO CHE $f \in \sim(g) \leftrightarrow \exists c \mid \lim_{n\to\infty} \frac{f(n)}{g(n)}$

E CHE, PER DEFINIZIONE. $f \in O(g) \leftrightarrow \forall n (n \geq K \rightarrow f(n) \leq g(n) \cdot c)$

OSSERVANDOLE. O STABILISCE LA SOLA ESISTENZA DI UN ESEMPIO

SUPERIORE, MENTRE $\sim$ GARANTISCE CHE LA CONDIZIONE RESISTE

ALL'INFINITO $\rightarrow$ $\sim$ È "FINER"

b

TRANSITIVITÀ $f \in O(g) \wedge g \in O(h) \rightarrow f \in O(h)$?

$f(n) \leq g(n) \cdot c \quad c \cdot g(n) \leq h(n) \cdot c^2 \rightarrow f(n) \leq h(n) \cdot c^2$ ✓

RIFLESSIVITÀ $f \in O(f)$? $f(n) \leq f(n) \cdot c$ ✓

SIMMETRIA $f(n) \leq g(n) \cdot c \wedge g(n) \leq f(n) \cdot c$

NO ESEMPIO: $n \leq c \cdot n^2$ ✓ MA NON VALE CHE $n^2 \leq c \cdot n$

Theoretical Computer Science Exam, July 10th, 2018

The exam consists of **4 exercises**. Available time: 1 hr 45 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 …….. MATRICOLA ……

I have passed the midterm exam and I intend to skip the first part (Ex.1-2) (select one): **YES NO**

Exercise 1 (9 pts)

Design minimal power automata that accept the following languages.

1) $\{ a^m b^n c^p d^q \mid m, n, p, q \geq 1 \text{ and } m + n = p + q \}$
2) $\{ a^m b^n c^p d^q \mid m, n, p, q \geq 1 \text{ and } m = p \text{ and } n = q \}$
3) $\{ a^m b^n c^p \mid m,n,p \geq 1$ and (m is odd if and only if n is odd) and (if p is even then n is even) $\}$

Exercise 2 (8 pts)

Write a Monadic First Order (or Monadic Second Order if necessary) logic formula that characterizes the language, over the alphabet $I = \{a, b\}$, whose strings are such that

1. all the a's occur at positions with an odd index value (remember that in the word structure the index of position starts with the value 0), and
2. they include an odd number of a's.

Exercise 3 (8 pts)

1. Consider the following function

   $$h(x) = \textbf{if } f_x(x) = x \textbf{ then } 1 \textbf{ else } \perp$$

   Is function h computable? Provide a clear justification for your answer.

2. (Optional) Consider the following set $S = \{ z \mid f_z(z) = z \}$. Is set S semidecidable? Justify your answer.

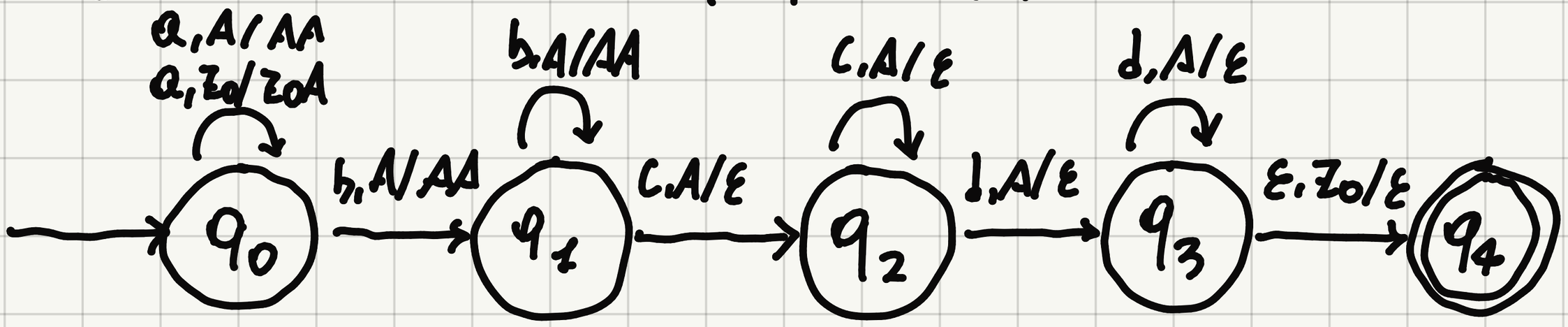
Exercise 4 (8 pts)

Consider the language, over the alphabet $I = \{a, b\}$, defined as follows $L = \{ aba^2b \ldots a^kb \mid k \geq 1 \}$. In other words, the strings of L can be enumerated as follows: $L = \{ ab, abaab, abaabaaab, abaabaaabaaaab, abaabaaabaaaabaaaaab, \ldots \}$.

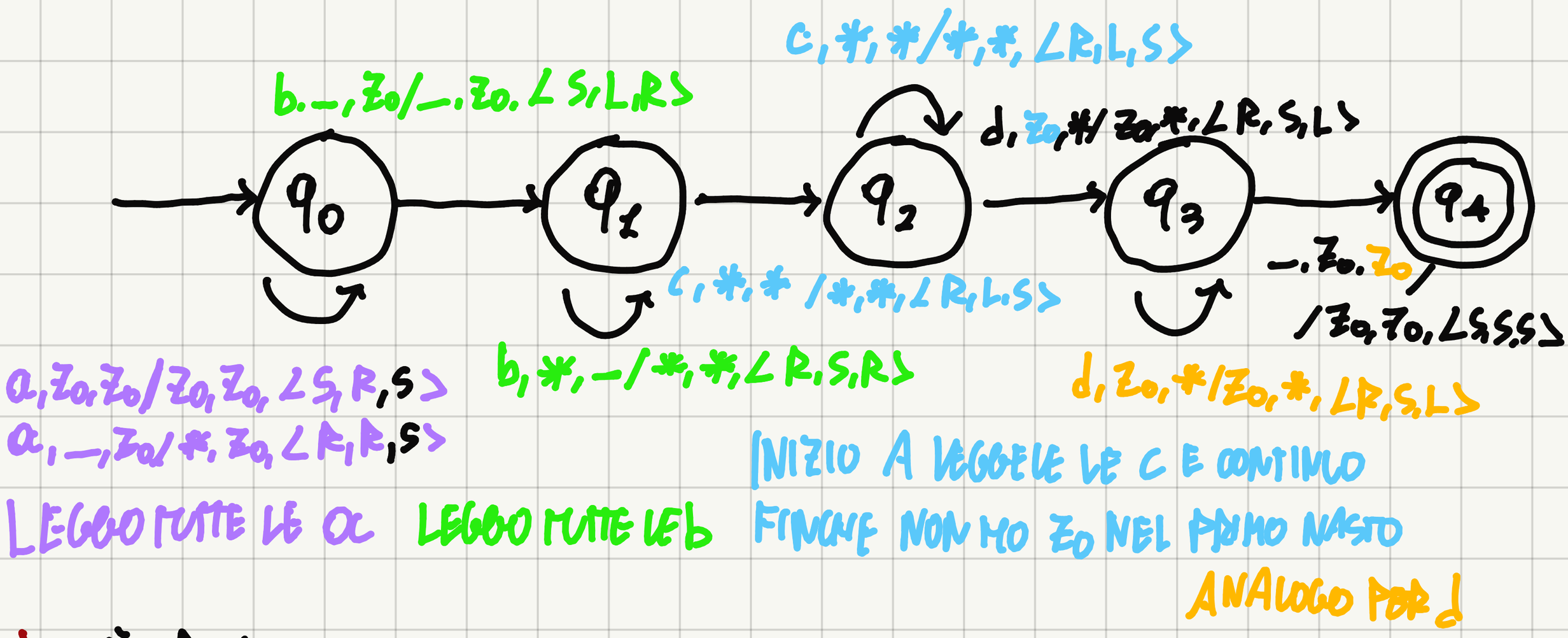
Describe, in a precise and clear way, a Turing machine and a RAM machine that accept language L, preferably with minimal time complexity. Determine which of the two machines is more efficient, in a separate way for what concerns time and for what concerns space; suitably motivate your answers.

1

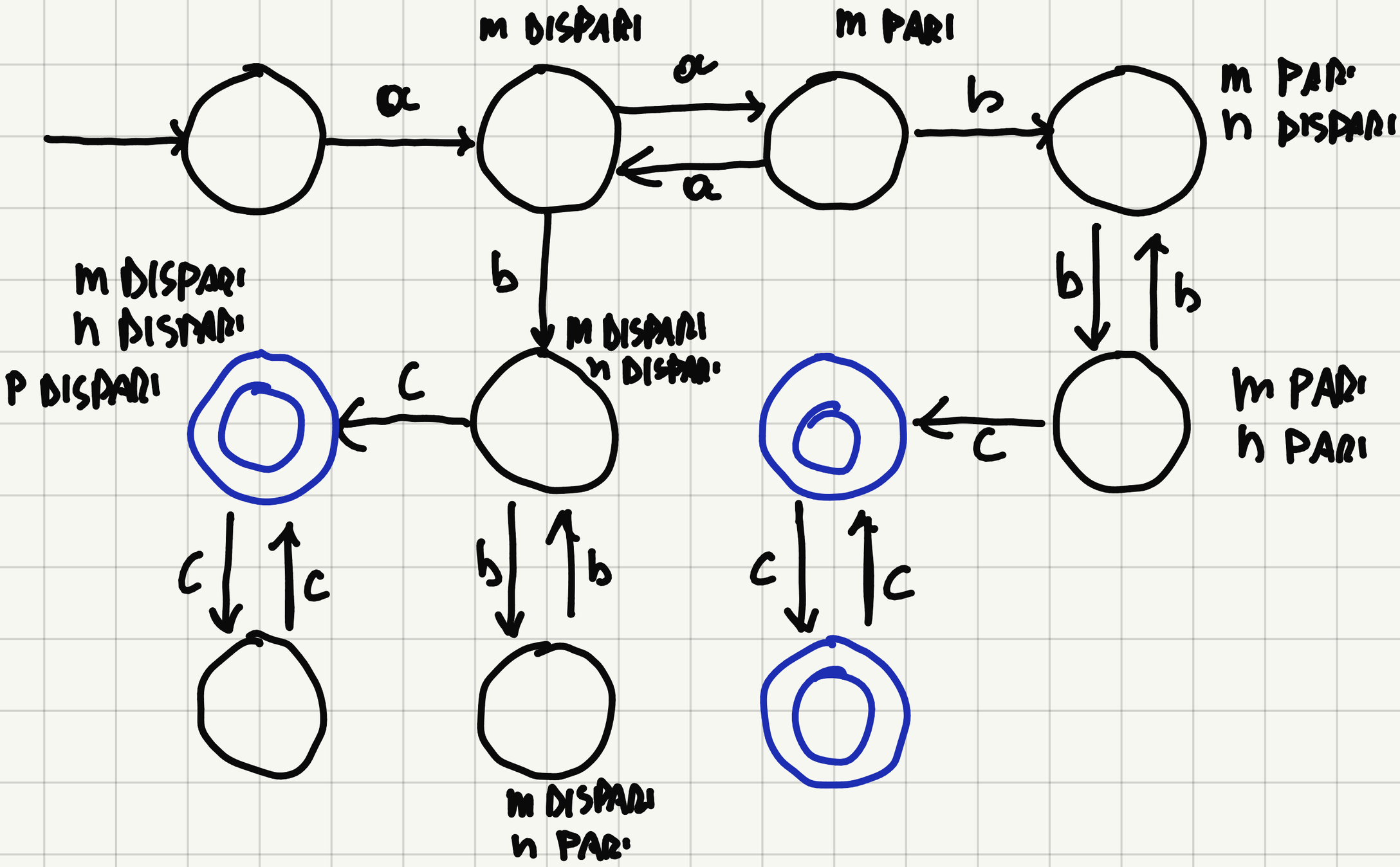
a) OCCORRE L'USO DI UN AUTOMA A PILA



b) PER TENERE MEMORIA DI m DISPARI E n DISPARI E P). OCCORRE UNA MACCHINA DI TURING A DUE NASTRI



c) È RICHIESTO UN AUTOMA A STATI FINITI NON DETERMINISTICO



2

a INTRODUCIAMO IL PREDICATO D(x) "POSIZIONE DI INDICE DISPARI"

$$\neg D(0) \wedge \exists x \big((\neg last(x) \wedge D(x) \rightarrow \neg D(x+1)) \wedge$$

$$\wedge (\neg last(x) \wedge \neg D(x) \rightarrow D(x+1)) \wedge$$

$$\wedge (a(x) \leftrightarrow D(x)) \wedge (last(x) \rightarrow D(x)) \big)$$

b INTRODUCIAMO IL PREDICATO N(x) "NUMERO DISPARI DI a"

$$\neg N(0) \wedge \exists x \big(([N(x) \wedge a(x)) \wedge \neg last(x) \rightarrow \neg N(x+1)] \wedge$$

$$(\neg N(x) \wedge a(x) \wedge \neg last(x) \rightarrow N(x+1)) \wedge$$

$$(last(x) \rightarrow N(x)) \big)$$

3 a

LA FUNZIONE È COMPUTABILE. INFATTI h(x) SIMULA IL COMPORTAMENTO DI $f_x(x)$, NOTA FUNZIONE COMPUTABILE. SE L'ESECUZIONE TERMINA E $f_x(x) = x$, h(x) RESTITUISCE 1; ALTRIMENTI ENTRA IN LOOP ↓

b S È IMMAGINE DELLA FUNZIONE TOTALE E COMPUTABILE DEFINITA NEL PUNTO a, E QUINDI È SEMIDECIDIBILE

4. È SUFFICIENTE UNA MACCHINA DI TURING A SINGOLO NASTRO PER LEGGERE STRINGHE DEL TIPO $a^i b$. LA TESTINA PUÒ ALTERNATIVAMENTE MUOVERSI TRA DESTRA E SINISTRA OGNI "NUOVA STRINGA". DETTA n LA LUNGHEZZA DELLA STRINGA, $T(n) = \Theta(n)$ E $S(n) = \Theta(\sqrt{n})$

(LA COMPLESSITÀ SPAZIALE DIPENDE DALL'ULTIMA "STRINGA")

UNA MACCHINA RAM PUÒ LEGGERE LA STRINGA PER INTERO ED UTILIZZARE UN CONTATORE PER LE OPPORTUNE VERIFICHE SU a. ALLORA.

DETTA n LA LUNGHEZZA DELLA STRINGA

COSTO UNIFORME

$T(n) = \Theta(n)$

$S(n) = \Theta(1)$ (NUMERO FISSO DI CELLE DI MEMORIA RICHIESTE)

COSTO LOGARITMICO

$T(n) = \Theta(n \log n)$ DETTO h IL NUMERO DI OCCORRENZE DI a, IL RISPETTIVO CONTATORE, INTERPRETANDO IN BINARIO IL NUMERO h, IMPIEGA UN TEMPO $\log h$ PER LA CONVERSIONE. ANALOGO PER b

$S(n) = \Theta(\log n)$ I NUMERI NEL CONTATORE VENGONO INTERPRETATI IN BINARIO. E QUINDI AGGIUNTO UN FATTORE $\log n$

IN CONCLUSIONE. LA RAM È SEMPRE SUPERIORE PER $S(n)$. MENTRE RISULTA MENO EFFICIENTE PER $T(n)$ CON IL CRITERIO DI COSTO LOGARITMICO.

Theoretical Computer Science Exam, September 13th, 2018

The exam consists of **4 exercises**. Available time: 1 hr 45 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 …….. MATRICOLA ……

I have passed the midterm exam and I intend to skip the first part (Ex.1-2) (select one): **YES NO**

Exercise 1 (8 pts)

Consider the following grammar G:

$$S \rightarrow A\ B$$
$$A \rightarrow a\ c\ C$$
$$B \rightarrow C\ c\ b$$
$$C \rightarrow c\ C \mid C\ c \mid c$$
$$C\ C \rightarrow D$$
$$D \rightarrow d\ D \mid d$$
$$A\ B \rightarrow c\ C\ c$$

a) Identify the (smallest) language family of $L(G)$; provide a precise and concise description of $L(G)$.

b) Design an automaton, of the least powerful type, that accepts $L(G)$.

Exercise 2 (8 pts)

Specify, by means of suitable pre- and post-conditions, a subprogram **gcd_lcm**(*a*, *b*, *c*, *out*) where *a*, *b*, *c* are input numeric parameters that may be assumed to have positive integer values such that $a < b < c$, and *out* is a Boolean output parameter. After execution of the subprogram, the output parameter *out* is true if and only if one of the following conditions holds.

- *a* is the greatest common divisor of *b* and c
- *c* is the least common multiple of *a* and *b*

Exercise 3 (8 pts)

Consider the following set

$$S = \{\ x \ \mid \ |\{ y \mid f_x(y) \neq \perp \}| \leq 1 \ \}$$

i.e., the set of TM indices *x* such that the computable function f_x is defined for no more than one value. Answer the following questions, providing suitable justifications for your reply.

1. Is set *S* decidable?
2. Is set *S* semidecidable?
3. Is set $\neg S$ (the complement of *S*) semidecidable?

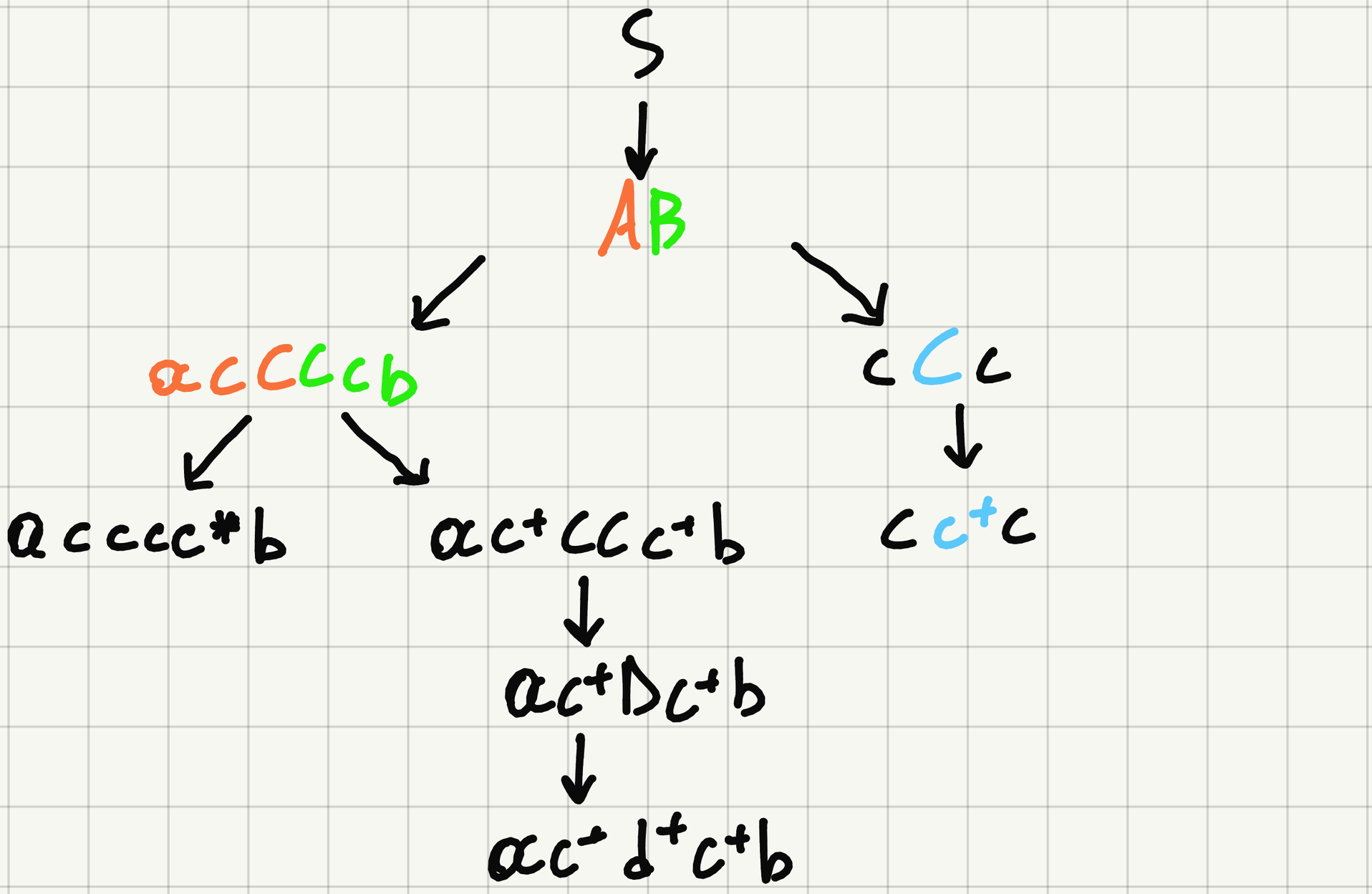
Exercise 4 (9 pts)

Consider the language, over the alphabet {0, 1} that includes strings of the type *xy*, where $|x| = 5$ and *x* encodes in binary the position of a character, inside string y, which must be equal to 1.

Describe, in a clear, complete and unambiguous way, a single tape Turing machine and a RAM machine that accept the above language; evaluate the space and time complexity using all the available cost criteria.

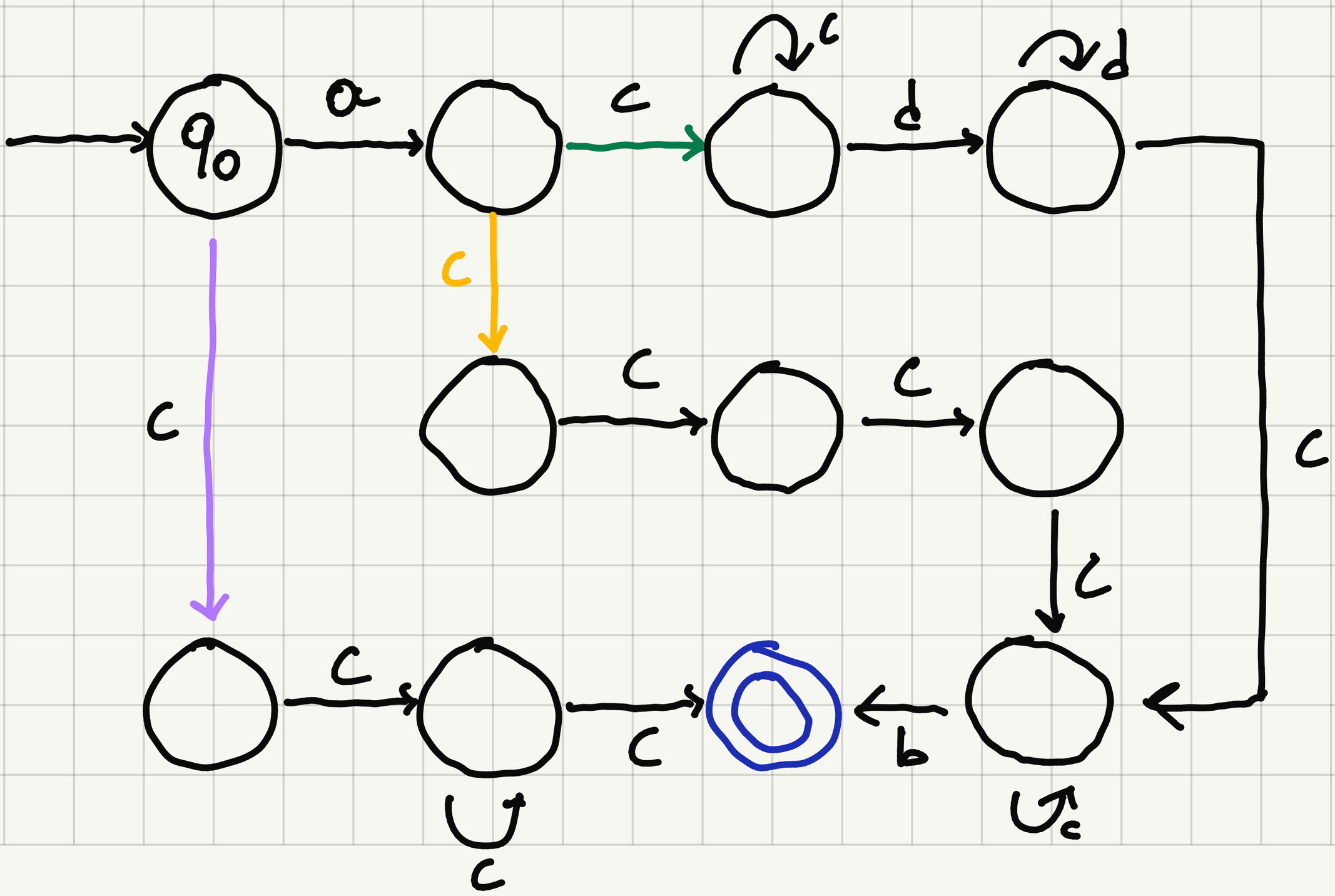
1

a) INIZIAMO CON UNA VISUALIZZAZIONE AD ALBERO



OTTENIAMO CHE $L = \{acccc^* b\} \cup \{ac^+d^+c^+b\} \cup \{cc^-c\}$

b) È SUFFICIENTE UN AUTOMA A STATI FINITI E, VISTO IL NON NOTO STATO INIZIALE, SI USERÀ UN NON DETERMINISTICO $L = L_1 \cdot L_2 \cdot L_3$



2

INPUT: • a, b, c INTERI POSITIVI

OUTPUT: • OUT BOOLEANO

PRECONDIZIONI] $c > b \wedge b > a \wedge a > 0 \wedge b > 0 \wedge c > 0$

POSTCONDIZIONI] $(OUT = T) \leftrightarrow$

$(\exists k > 0, k' > 0 (b = ka \wedge c = k'a \wedge \forall v > k', k (b = va \rightarrow v = k \wedge c = va \rightarrow v = k')))$

$\vee (\exists k, k' > 0 (c = ak \wedge c = bk' \wedge \forall v < k, k' (c = av \rightarrow v = k \wedge c = bv \rightarrow v = k')))$

3

a IL SET È NON DECIDIBILE PER RICE THEOREM. INFATTI SI SISTEMA FISSANDO X (INDICE DELLA M.T.) E SI RICERCANO FUNZIONI DEFINITE PER NON PIÙ DI UN VALORE. QUESTO NON È, CHIARAMENTE, NE L'INSIEME VUOTO NÈ UNIVERSO. PER CUI, S NON DECIDIBILE

b L'ANALISI DIRETTA DI S RISULTA COMPLESSA. VERIFICHIAMO $\neg S = \{x \mid |\{y \mid \beta_x(y) \neq \perp\}| > 1\}$. FISSATO X, È IMMEDIATO VERIFICARE CHE $\neg S$ È SEMIDECIDIBILE con DOVETAILING

→ S È NON SEMIDECIDIBILE

c $\neg S$ È SEMIDECIDIBILE

4

CONSIDERIAMO IL CASO PEGGIORE. SIA x=11111 E y LA CODIFICA DELL'ULTIMO CARATTERE, IN POSIZIONE 5. $11111_2 = 31_{10}$

⟶ IL CASO PESSIMO PREVEDE 36 MOSSE.

UNA MACCHINA DI TURING LEGGE I 5 CARATTERI DI x E SALVA LE POSIZIONI DEGLI 1. SUCCESSIVAMENTE, UNA DI QUESTE VERRÀ USATA PER SCRIVERE y.

DETTA n LA LUNGHEZZA DELLA STRINGA:

COSTO UNIFORME

$T(n) = \Theta(1)$ *

$S(n) = \Theta(n)$

COSTO LOGARITMICO

$T(n) = \Theta(1)$

$S(n) = \Theta(1)$

LE STRINGHE ACCETTABILI IN INPUT SONO GIÀ IN BINARIO

Theoretical Computer Science Exam, January 17th, 2019

The exam consists of **4 exercises**. Available time: 1 hr 40 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 …….. MATRICOLA ……

I have passed the midterm exam and I intend to skip the first part (Ex.1-2) (select one): **YES NO**

Exercise 1 (8 pts)

Consider the following (partial) function, defined on even-length strings and undefined for odd-length strings.

$src(x) = y\, z^R$ if and only if $x = yz$ and $|y| = |z|$

Thus, for instance, $src(abcd) = abdc$, $src(abbabb) = abbbba$, $src(\varepsilon) = \varepsilon$.

The *src* function is also generalized by applying it to languages, with the following, customary definition.

$src(L) = \{\ src(x)\ \mid x \in L\ \}$

In words, $src(L)$ is defined as the language that contains exactly the strings obtained by applying the *src* function to a string of L.

Next, consider the grammar G, with axiom S and alphabet $\{a, b\}$ defined by the following rules.

$S \rightarrow a\, P\, a \mid \varepsilon$

$P \rightarrow b\, S\, b$

1. Write the four shortest strings of $L(G)$, in order of increasing length.
2. Write a concise, formal and precise definition of $L(G)$.
3. Write the result of applying function *src* to the four shortest strings of $L(G)$ found in the previous question.
4. Write a concise, formal and precise definition of $src(L(G))$, and determine which is is least powerful type of automaton that accepts $src(L(G))$.
5. Write a context-free grammar that generates $src(L(G))$.
6. Determine if the family of regular languages is closed under the *src* function. Provide a justification for your answer.

Exercise 2 (8 pts)

Consider the two language expressions

$$L_1 = (\ (\ a \mid b\ )\ (\ c \mid d\ )\ )^+ \quad \text{and} \quad L_2 = (\ (\ a \mid b\ )\ (\ b \mid c\ )\ )^+$$

Notice that the two languages are not disjointed and none of the two is included in the other; in fact, the following string inclusions hold (where the operator '-' here denotes set difference):

$$x_1 = ad \in L_1 - L_2, \quad x_{12} = ac \in L_1 \cap L_2, \quad x_2 = ab \in L_2 - L_1.$$

1. Write a monadic logic sentence φ_1 (first-order, or second-order only if necessary) such that $L_1 = L(\varphi_1)$. Verify that $x_2 = ab \notin L(\varphi_1)$ and explain in logical terms the reason why $ab \not\models \varphi_1$.
2. Write a monadic logic sentence φ_2 (first-order, or second-order only if necessary) such that $L_2 = L(\varphi_2)$. Verify that $x_1 = ad \notin L(\varphi_2)$ and $y = abb \notin L(\varphi_2)$ and explain in logical terms the reason why $ad \not\models \varphi_2$ and $abb \not\models \varphi_2$.