University of Twente
Department of Computer Science,
Faculty EEMCS
November, 1, 2012, 8:45 – 12:15h.

Year 2012-2013
Period 1A
Interim examination

Computability and computational complexity
(192111700)

The use of course material during the examination is permitted. To hand in work produced with the help of others (by cribbing, by passing notes or other messages, by wireless consultation of persons or (internet) sources during the examination), is fraudulent.

Your answers to questions must be supported by a clear argument. To give an answer without explanation will have a negative influence on the number of points you will earn. This applies also to incomplete and unclear explanations.

You can earn 90 points in total. To determine your mark, the number of points earned is incremented by 10, and the total is divided by 10. The result is rounded to the nearest natural number.

There are 8 exercises, on 3 pages. The exercises are **not** ordered in increasing difficulty (nor conversely). Moreover, the number of points to be earned for an exercise is not necessarily a measure for the difficulty of that exercise.

Please put your name, your student number and the course name on every sheet of paper you hand in.

Veel succes, good luck, bonne chance!!

Exercise 1 (5 points).

Discuss the truth (or falsehood) of the statement:

The intersection of a recursive language and a recursively enumerable language is recursive.

Exercise 2 (15 points).

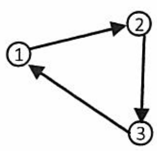
We use the following representation for directed graphs.

Vertices are numbered, numbers are sequences of 1's (unary number representation).

Arcs are represented by pairs of numbers, separated by a single 0 (so 1101111 is the arc from node 2 to node 4).

The graph is represented as the list of its arcs, separated by double 0's.

So 1011001101110011101 represents the graph



(a) Design a non-deterministic Turing machine which solves the Hamiltonian circuit problem in polynomial time (using the above representation). Do not forget to give an argument for the polynomial character of your solution.

(b) Would it make a difference for the tractability (polynomial character, non-deterministic or not) of possible solutions if we use a binary number notation in the graph representation (and hashes # as separators)?

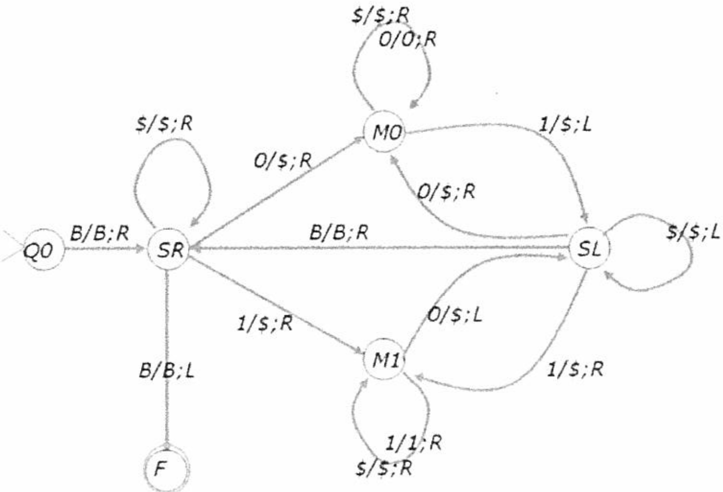
Exercise 3 (10 points).

Discuss the truth (or falsehood) of the statements:

(a) *It is decidable whether a given recursively enumerable language is in fact regular.*

(b) *It is decidable whether a given word w belongs to the intersection of two context-sensitive languages.*

Exercise 4 (12 points).



(a) The picture (copied from the lecture slides) shows a TM which accepts the language $L =_{def} \{w \in \{0,1\}^* \mid$ the number of 0's in w equals the number of 1's in $w\}$. Q0 is the start state, F is the final state.
Show that the time complexity of this solution is **not** linear

(b) Design a two-tape TM which gives a linear time solution of the problem.

Exercise 5 (24 points).

We consider binary TMs with input alphabet $\Sigma=\{0,1\}$, and tape alphabet $\{0,1,B\}$.

We use R(M) to denote the representation of the Turing machine M in a sequence of 0's and 1's; $R(M) \in \Sigma^*$.

We use L(M) to denote the language associated with the Turing machine M, so $L(M) =_{def} \{ w \in \Sigma^* \mid$ M with input w halts in a final (accepting) state).

For a Turing machine M we define the language L^M by
$L^M =_{def} \{1^k \mid$ M with blank tape halts within k steps$\}$.

(a) Given M, design a 2-tape Turing machine M* which accepts L^M, i.e. such that $L^M = L(M^*)$.

(b) Is L^M a recursive language?

$H_B =_{def} \{ R(M) \mid M$ halts with input λ (an empty tape) $\}$

(c) Show that H_B can be reduced to $\{R(M) \mid L^M$ is an infinite language$\}$

(d) Consider the language $\{R(M) \mid L(M)$ is a finite language $\}$. Is it recursively enumerable?

Exercise 6 (6 points).

Discuss the truth (or falsehood) of the statements:

The complement of a context-free language is context-sensitive.

Exercise 7 (12 points).

An off-line Turing machine is a 2-tape machine with a write protection on one of its tapes: the input tape. So with every transition of the machine, the contents of this input tape remain the same, while only the contents of the second tape can change.

Given is a problem L, which is decided by an off-line Turing machine M with a space complexity bound $s(n) = n$. All computations of M halt.

The machine M has the somewhat peculiar property that it never erases or changes the contents of non-blank tape squares. In some transitions it will write a new symbol, but only at a tape square which was blank until then. Once written, the symbol will stay at that square during the entire remaining computation; the square will not be erased or rewritten.

Show that the time complexity of M's computations has a polynomial upper bound.

Exercise 8 (6 points).

Discuss the truth (or falsehood) of the statement:

A language L such that $L \in NP$, but $L \notin NPC$ (the set of NP-complete problems), can exist only if $P \neq NP$.