Admission to the PhD School in Computer Science, University of Pisa, 2011

1. Solve two of the following exercises

a) A transition system over a set $A = \{a, a', \ldots\}$ of actions is a structure (Q, T) where $Q = \{q, q', \ldots\}$ is the set of states and $T \subseteq Q \times A \times Q$ is the transition relation. A transition system is basically an automaton without initial state and acceptance states. We now introduce a simple modal logic to express properties of transition systems. The syntax of the logic is given as follows:

$$P ::= tt \mid ff \mid P \wedge P \mid P \vee P \mid \langle a\rangle P \mid [a]P$$

We define when a state q satisfies the formula P (written $q \vDash P$) or it does not (written $q \nvDash P$).

$q \vDash tt$ $\qquad$ $q \nvDash ff$

$q \vDash P1 \wedge P2 \Longleftrightarrow q \vDash P1$ and $q \vDash P2$ $\qquad$ $q \vDash P1 \vee P2 \Longleftrightarrow q \vDash P1$ or $q \vDash P2$

$q \vDash \langle a\rangle P \Longleftrightarrow \exists\, q' \in Q$, such that $(q, a, q') \in T$ and $q' \vDash P$

$q \vDash [a]P \Longleftrightarrow \forall\, q' \in Q$, such that $(q, a, q') \in T$ and $q' \vDash P$

States q e q' are logical equivalent (and we write $q \simeq q'$) if and only if they satisfy the same logical formulae

i) Prove that $\simeq$ is an equivalence relation.

ii) Let $M1 = (Q1, q, F1, T1)$ and $M2 = (Q2, q', F2, T2)$ be finite state automata over the same alphabet. Discuss when $q \simeq q'$ implies that M1 and M2 recognize the same language (language equivalence). By providing an example, show that language equivalence does not imply logical equivalent.

b) Describe the divide-and-conquer technique for the design of efficient algorithms. Provide a specific example of a problem for which this technique allows us to design a time-optimal algorithm.

c) Consider a process level computation, consisting of a Master process and of identical processes $Worker_0$, …, $Worker_{n-1}$, able to execute an integer function F. The execution time of F has a high variance, depending on the input parameter value.

The Master process receives a, possibly unlimited, sequence of integer values, and schedules them over the set of workers. The scheduling strategy must achieve a good load balance of workers in spite of their variable execution times.

Describe the computation using a message-passing formalism. The candidate is free to chose such formalism, provided that a short definition/characterization is given.

No assumption about the run-time support of the process level has be used for expressing the solution.

d) Using the relational data model, provide a database schema representing an undirected graph $G = (V, E)$ where V is a set of nodes (vertices) and E is a set of edges between pairs of nodes, and specify the following queries over the above schema, using either the relational algebra or SQL syntax:

1) *degree*(x, k): a query returning a table *degree* that associates each node x with the number k of edges that connect to x;

2) *degree-distribution*(k, n): a query returning a table that associates each distinct degree value $k>0$ with the number n of nodes of degree k;

3) *common-neighbors*(x, y, m): a query returning a table that associates each pair of nodes x, y with the number m of the common neighbors of x and y (i.e., the nodes one hop away from both x and y); in formula

$$m = |\Gamma(x) \cap \Gamma(y)|$$

where, for each node x $\Gamma(x) = \{z \in V : (x,z) \in E\}$ is the set of neighbors of x.

4) *adamic-adar*(x, y, w) : a query returning a table that associates each pair of nodes x, y with the value w obtained by summation, for each common neighbor z of x and y, of the inverse logarithm of the degree of z; in formula, for each pair of nodes ⍰,⍰, the value ⍰ of the Adamic-Adar measure is defined as:

$$w = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|}$$

2. Identify and discuss a research problem in one of the following research areas:

a) data bases

b) computational complexity

c) programming languages

d) parallel and distributed systems