

Esame di ammissione al Dottorato di Ricerca in Informatica, anno 2011

1. Il candidato svolga due esercizi a scelta tra i quattro seguenti

- a) Assumiamo come dato un insieme $A = \{a, a', \dots\}$ di azioni. Un sistema di transizione su A è una struttura (Q, T) dove $Q = \{q, q', \dots\}$ è l'insieme degli stati e $T \subseteq Q \times A \times Q$ è la relazione di transizione. Un sistema di transizione deve essere pensato come un automa (non necessariamente a stati finiti) senza stato iniziale e senza stati di accettazione.

Introduciamo una semplice logica modale per descrivere proprietà di sistemi di transizione. La sintassi delle formule logiche P risulta:

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

Di seguito viene presentata la relazione di soddisfacibilità, ovvero quando uno stato q soddisfa una formula P ($q \models P$) o quando non la soddisfa ($q \not\models P$):

$$\begin{aligned} q \models \text{tt} & \quad q \not\models \text{ff} \\ q \models P_1 \wedge P_2 & \Leftrightarrow q \models P_1 \text{ e } q \models P_2 & q \models P_1 \vee P_2 & \Leftrightarrow q \models P_1 \text{ o } q \models P_2 \\ q \models \langle a \rangle P & \Leftrightarrow \exists q' \in Q, \text{ tale che } (q, a, q') \in T \text{ vale che } q' \models P \\ q \models [a]P & \Leftrightarrow \forall q' \in Q, \text{ tale che } (q, a, q') \in T \text{ vale che } q' \models P \end{aligned}$$

Due stati q e q' sono logicamente indistinguibili (e scriviamo $q \simeq q'$) se e solo se soddisfano le stesse formule logiche

i) Dimostrare che $\simeq$ è una relazione di equivalenza.

ii) Siano $M_1 = (Q_1, q, F_1, T_1)$ e $M_2 = (Q_2, q', F_2, T_2)$ due automi a stati finiti sullo stesso alfabeto. Dimostrare che se $q \simeq q'$ allora i due automi riconoscono lo stesso linguaggio. Tramite un esempio mostrare che la tradizionale equivalenza tra automi (due automi sono equivalenti se riconoscono lo stesso linguaggio) non implica l'equivalenza logica.

- b) Si descriva la tecnica di divide-et-impera per lo sviluppo di algoritmi efficienti, fornendo uno specifico esempio di problema per il quale tale tecnica consente di trovare la soluzione in tempo ottimale.

- c) Si consideri una computazione a livello di processi, consistente di un processo Master e dei processi identici $\text{Worker}_0, \dots, \text{Worker}_{n-1}$, capaci di eseguire una funzione F da interi a interi. Il tempo di esecuzione della funzione F è fortemente variabile a seconda del valore del parametro d'ingresso.

Il processo Master riceve una sequenza, eventualmente illimitata, di valori interi. Il suo compito è di schedare i valori della sequenza d'ingresso nei confronti dei worker. La strategia di scheduling

deve assicurare un buon bilanciamento del carico dei worker a fronte dei tempi di esecuzione variabili della funzione F .

Esprimere la computazione con un formalismo a scambio di messaggi a scelta del candidato. Di tale formalismo si dia prima una breve caratterizzazione.

La soluzione non deve far uso di ipotesi sul supporto a tempo di esecuzione del livello dei processi.

- d) Si definisca nel modello relazionale uno schema di base di dati che rappresenta un grafo (non diretto) G composto da un insieme di nodi V e di archi E , e si definiscano le seguenti interrogazioni, in riferimento allo schema introdotto, utilizzando l'algebra relazionale oppure la sintassi SQL:

1) $degree(x, k)$: l'interrogazione deve restituire una relazione $degree$ che associa ad ogni nodo x il numero k degli archi incidenti su x ;

2) $degree-distribution(k, n)$: l'interrogazione deve restituire una relazione che associa ad ogni valore distinto $k > 0$ del grado, il numero n dei nodi che possiedono quel grado;

3) $common-neighbors(x, y, m)$: l'interrogazione deve restituire una relazione che associa ad ogni coppia di nodi x, y il numero m dei nodi adiacenti comuni (raggiungibili sia da x che da y con un solo arco); in formula

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

dove, per ogni nodo x , $\Gamma(x) = \{z \in V \mid (x, z) \in E\}$ è l'insieme dei vicini di x .

4) $adamic-adar(x, y, w)$: l'interrogazione deve restituire una relazione che associa ad ogni coppia di nodi x, y il valore w ottenuto sommando, per ciascun nodo z adiacente sia ad x che ad y , l'inverso del logaritmo del grado di z ; in formula, dati due nodi x, y , il valore w della misura di Adamic-Adar è definito da:

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

2. Nell'ambito di una delle aree di ricerca sotto elencate, individuare un problema di ricerca e discuterlo:

- a) basi di dati
- b) complessità computazionale
- c) linguaggi di programmazione
- d) sistemi paralleli e distribuiti

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 $A = \{a, a', \dots\}$ of actions is a structure (Q, T) where $Q = \{q, q', \dots\}$ 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 $q \models P$) or it does not (written $q \not\models P$).

$$q \models tt \qquad q \not\models ff$$

$$q \models P1 \wedge P2 \Leftrightarrow q \models P1 \text{ and } q \models P2$$

$$q \models P1 \vee P2 \Leftrightarrow q \models P1 \text{ or } q \models P2$$

$$q \models \langle a \rangle P \Leftrightarrow \exists q' \in Q, \text{ such that } (q, a, q') \in T \text{ and } q' \models P$$

$$q \models [a]P \Leftrightarrow \forall q' \in Q, \text{ such that } (q, a, q') \in T \text{ and } q' \models P$$

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

i) Prove that $\approx$ 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. Prove that if $q \approx q'$ then $M1$ 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 $Worker_0, \dots, Worker_{n-1}$, 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 , $\Gamma(x) = \{z \in V \mid (x, z) \in E\}$ 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 x, y , the value w 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