ESERCIZI

CCS

Esercizio A. Studiare una delle seguenti relazioni comportamentali

1. Simulation (testo, pag. 52)
2. Ready simulation (testo, pag. 52)
3. 2-nested simulation (testo pag. 70)
4. Branching bisimulation (testo, pag. 64)

In particolare, studiare:

- le proprietà di preordine o di equivalenza
- la relazione con la bisimulation equivalence
- le proprietà di congruenza

e, facoltativamente:

- una possibile caratterizzazione logica (nello stile del teorema di Hennessy-Milner)
- una possibile caratterizzazione come gioco

Esercizio B. Dimostrare che ogni processo del CCS finito termina in un numero finito di passo (con e senza somme infinite).

Esercizio C. Dimostrare che la codifica del CCS con value passing nel CCS puro è vista a lezione corretta e completa rispetto alla semantica operazionale.

Esercizio D. Dimostrare che la trace equivalence è una congruenza per il CCS.

Esercizio E. Siano T e T' due LTS e si assuma che esista un morfismo $f : T \rightarrow T'$ tra i due LTS visti come grafi con archi etichettati. Discutere la relazione esistente tra le proprietà di f e le proprietà di bisimulazione.

Esercizio F. Indichiamo con $a^i = a. \ldots a. 0$, ovvero una sequenza di i azioni a. Sia inoltre $A^\omega = a. A^\omega$. Dimostrare che i processi $\Sigma_i a^i$ e $A^\omega + \Sigma_i a^i$ soddisfano le stesse formule della logica HML, ma non sono bisimili. E se si considera la logica con ricorsione?

Esercizio G. Mostrare che la codifica di uno degli operatory Inv, Pos, Safe, Even, Until (strong o weak) nel mu-calcolo cattura la proprietà intesa, ad es. che
[[Even(ϕ)]] = { P | per ogni computazione completa $P = P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots$ esiste i tale che $P_i \models \phi$ }.

Esercizio H. Mostrare che la semantica della logica di Hennessy-Milner vista a lezione è ben definita, ovvero che quando si considerano minimi e massimi punti fissi, le funzioni corrispondenti sono monotone su di un reticolo. Discutere se e come potrebbe essere inclusa la negazione nella logica..

Esercizio I. Discutere la relazione esistente tra la congruenza osservazionale e la massima congruenza contenuta nella bisimilarità debole.

Esercizio L. Dimostrare che la semantica $[[P]]\eta$ di una formula HML è indipendente dal valore che η

associa a variabili non libere in P.

Esercizio M. Discutere una estensione del CCS con un operatore di composizione sequenziale tra processi $P; Q$. Dare una semantica operazionale e analizzare la possibilità di avere una codificare in CCS dell'operatore definito.

Esercizio N. Dimostrare che la chiusura contestuale della bisimilarità debole è la più grande congruenza che raffina la bisimilarità debole. Dimostrare che la suddetta relazione è la congruenza osservazionale.

Esercizio O. Dimostrare che la chiusura contestuale della bisimilarità debole è la più grande congruenza che raffina la bisimilarità debole. Dimostrare che la suddetta relazione è la congruenza osservazionale.

Esercizio P. Verificare che se F è il funzionale della bisimulazione, allora le relazioni $F^{(n)}(Proc\ x\ Proc)$ sono tutte equivalenze

Esercizio Q. Verificare se quando D è un reticolo completo numerabile e F : D -> D una funzione monotona allora il minimo punto fisso si ottiene come $sup_n F^{(n)}(0)$, dove 0 è il minimo del reticolo. Fonire una dimostrazione o un controesempio.

Esercizio R. Bisimulazione e proprietà di congruenza nel pi-calcolo.

MINI-PROGETTI

Modellare in CCS uno dei seguenti algoritmi o protocolli e provarne la correttezza (si veda il Cap. 7 del libro). Le verifiche possono basarsi su di un tool didattico (CWB, TAPAS, SLMC).

- Uno dei problemi degli algoritmi di mutua esclusione nei capitoli 5, 6 o 7 del libro The Little Book of Semaphores di Allen B. Downey.
- Alternating bit protocol (si veda il link su Wikipedia o su Answers oppure qui).
- Il protocollo Carrier-Sense Multiple Access with Collision Detection (si veda, CSMA/CD, che contiene anche un riferimento ad un lavoro strettamente collegato).
- Bounded retransmission protocol (si veda qui).
- Prisoner's game
- Dijkstra's concurrent programming control problem (mutual exclusion for 3 processes) - si veda questo articolo

Una alternativa e` studiare e sperimentare strumemti più sofisticati

- CADP (Construction and Analysis of Distributed Processes)
- mCRL2: A language and tool set to study communicating processes with data
- PAT: Process Analysis Toolkit

APPROFONDIMENTI

Aspetti teorici

- Caratterizzazione della bisimilarità e del model checking come gioco (si veda il lavoro di Colin Stirling e il libro a pag. 115).
- Algoritmi di decisione per la bisimulazione (Kannelakis e Smolka, Paige e Tarjan) - si veda qui.
- Bisimulazione, coinduzione: una visione storica (si veda l'articolo di Davide Sangiorgi).

- Timed CCS / Timed Automata / Timed bisimulation (libro)
- Un approccio, proposto da Leifer e Milner, per la derivazione automatica di bisimulazioni che sono congruenze [articolo]

Altri calcoli, con funzionalità diverse

- Spi-calcolo, un calcolo per l'analisi di protocolli crittografici [Articolo]
- Ambient-calcolo, un calcolo per descrivere sistemi con mobilità (computazione su dispositivi mobili e codice mobile) [Articolo] e la corrispondente logica spaziale [Articolo]
- CCS reversibile per systems biology [Articolo]
- Formalizzazione di web-services [Articolo]
- Calcoli fondazionali per il service oriented computing [articolo]
- Session types - tipi comportamentali per la concorrenza [presentazione]
- Autonomic computing [Articolo]

Linguaggi reali, modelli di comunicazione simili a quelli analizzati

- Il linguaggio Jolie per la programmazione service-oriented
- Il linguaggio ORC per l'orchestrazione di servizi web
- Il linguaggio Go di Google
- Il linguaggio Scala
- Il linguaggio Erlang
- Pict: un linguaggio di programmazione basato sul pi-calcolo.
- Dal pi-calcolo a Java e viceversa [Articolo 1 + Articolo 2]
- Modello ad attori [Articolo1][Articolo2][Articolo3]
- Language-based security su Android [Articolo]
- Memorie transazionali [Articolo1][Articolo2][Articolo3]