

A.A. 2011-2012
ALLIEVI DEL III ANNO IN INGEGNERIA INFORMATICA

PRIMA PARTE DEL PROGETTO DA PRESENTARE
OBBLIGATORIAMENTE
COME PROVA (NON ESCLUSIVA) D'ESAME
DELL'INSEGNAMENTO
INGEGNERIA DEL SOFTWARE (9 CFU)

N.B. Una opportuna attività proposta dal docente, tesa a estendere o approfondire il progetto (prima e seconda parte) realizzato nell'ambito dell'insegnamento di Ingegneria del Software e svolta autonomamente dal singolo studente, con produzione di un elaborato finale individuale, può essere l'oggetto della PROVA FINALE (3 CFU) per il conseguimento della LAUREA IN INGEGNERIA INFORMATICA.

Si desidera realizzare un'applicazione che determini *diagnosi (minimali) con mascheramento comprensive degli ingressi* di un sistema (artefatto o naturale) di cui siano disponibili un *modello topologico* e un'osservazione.

DOMINIO APPLICATIVO

Modello topologico (di un sistema). È un grafo orientato aciclico i cui nodi sono di tre tipi, denominati *azioni*, *nodi d'ingresso* e *nodi di uscita*, e i cui archi sono denominati *flussi*. Un'azione ha almeno un flusso entrante e almeno un flusso uscente. Un nodo di ingresso non ha alcun flusso entrante e ha almeno un flusso uscente. Un nodo di uscita non ha alcun flusso uscente e ha un flusso entrante. Se i flussi in uscita a un'azione o a un nodo di ingresso sono più di uno, ciascuno di essi deve avere come destinazione un nodo distinto. Un flusso non può mai collegare un nodo d'ingresso con uno di uscita. Un'azione dotata di un flusso uscente diretto a un nodo d'uscita non ha altri flussi uscenti.

Osservazione (di un sistema). È un vettore di valori ternari (etichettabili rispettivamente come OK, KO e OKM), un valore per ogni nodo di uscita del sistema. Siano ok, ko e okm il numero di nodi di uscita che assumono rispettivamente il valore OK, KO e OKM.

Hitting set (di una collezione di insiemi). È un insieme che presenta un'intersezione non vuota rispetto a tutti gli insiemi della collezione data.

Hitting set minimale (di una collezione di insiemi). È un hitting set (della collezione di insiemi data) tale che nessun suo sottoinsieme è un hitting set.

Sottoinsieme interno (relativo a un nodo d'uscita OK). È l'insieme di tutti i nodi da cui dipende il nodo d'uscita che assume il valore OK considerato. Un nodo d'uscita dipende da un altro nodo se, nel grafo orientato (aciclico) che rappresenta il sistema, esiste almeno un cammino che, partendo da un nodo d'ingresso, comprende tale nodo e raggiunge il nodo di uscita considerato.

Conflitto strutturale (relativo a un nodo d'uscita KO). È l'insieme di tutti i nodi da cui dipende il nodo di uscita che assume il valore KO considerato.

Conflitto strutturale con mascheramento (relativo a un nodo d'uscita OKM). È l'insieme di tutti i nodi da cui dipende il nodo di uscita che assume il valore OKM considerato.

Diagnosi (minimale) con mascheramento. È un insieme di nodi (non d'uscita) del modello topologico considerato prodotto dal seguente procedimento:

a) Si calcolino tutti i seguenti (ko + okm) insiemi differenza

- $\underline{C_i} = C_i \setminus (\cup IS)$,
- $\underline{CM_j} = CM_j \setminus (\cup IS)$,

dove C_i è un *conflitto strutturale* (relativo a un nodo di uscita KO), CM_j è un *conflitto strutturale con mascheramento* (relativo a un nodo di uscita OKM), e $\cup IS$ è l'unione di tutti gli ok *sottoinsiemi interni* (relativi a tutti i nodi di uscita OK).

b) Per ciascuno nodo j appartenente all'insieme degli okm nodi di uscita OKM, si determini almeno una coppia di insiemi non vuoti

- $\underline{CM_j} \setminus \underline{C_i}$
- $\underline{CM_j} \cap \underline{C_i}$

c) Si generino in uscita gli *hitting set minimali* della collezione di insiemi contenente tutti gli insiemi prodotti al precedente punto b) e tutti gli insiemi $\underline{C_h}$ che non sono stati usati nel calcolo degli stessi. Ciascun di tali hitting set minimali è una diagnosi (minimale) con mascheramento.

REQUISITI FUNZIONALI

L'utente fornisce in ingresso all'applicazione:

- il modello topologico del sistema considerato,
- un'osservazione del sistema considerato.

L'applicazione produce in uscita un insieme di diagnosi (minimali) con mascheramento del sistema modellato, data l'osservazione dello stesso, dove tale insieme di diagnosi corrisponde al soddisfacimento della specifica operativa di cui alla pagina precedente, in particolare a una scelta operata al punto b).

Sessioni diverse possono prendere in considerazione modelli di sistemi diversi e/o osservazioni diverse e/o calcolare insiemi diversi di diagnosi minimali con mascheramento (corrispondenti a scelte diverse operate al punto b) della pagina precedente).

Si noti che,

- se l'osservazione non contempla almeno un valore KO e almeno uno OKM, non esiste alcuna diagnosi con mascheramento, ovvero l'insieme di tali diagnosi è vuoto (una condizione sufficiente affinché ciò avvenga è che il sistema sia dotato di un solo nodo d'uscita);
- se esiste un indice j tale che, al variare dell'indice i , per ogni coppia distinta (i,j) almeno uno degli insiemi_differenza $\underline{CM_j} \setminus \underline{C_i}$ e $\underline{CM_j} \cap \underline{C_i}$ è vuoto (una condizione sufficiente affinché ciò avvenga è che non esistano nodi condivisi fra conflitti strutturali e conflitti strutturali con mascheramento), l'insieme delle diagnosi (minimali) con mascheramento è anch'esso vuoto;
- la specifica operativa di diagnosi minimale con mascheramento è valida anche se il grafo che rappresenta il modello topologico è disconnesso. L'insieme di tutte le diagnosi minimali con mascheramento relative a un modello topologico il cui grafo è disconnesso (cioè è composto da più grafi connessi) è una sorta di “prodotto cartesiano” degli insiemi di tutte le diagnosi minimali con mascheramento relative a ciascuno grafo connesso componente se e solo se nessuno di tali insiemi è vuoto. Ad esempio, se il grafo del sistema è composto da due grafi connessi e $\{d1,d2,d3\}$ e $\{d4,d5\}$ sono gli insiemi di tutte le diagnosi (minimali) con mascheramento ad essi relative, l'insieme delle diagnosi (minimali) con mascheramento dell'intero sistema è $\{d1 \cup d4, d1 \cup d5, d2 \cup d4, d2 \cup d5, d3 \cup d4, d3 \cup d5\}$.

REQUISITI NON FUNZIONALI

Il linguaggio di programmazione da adottare è Java.

L'architettura esterna da realizzare per l'applicazione è stand alone.

Requisito non prescrittivo ma importante in sede di valutazione è l'impiego di precondizioni, postcondizioni e invarianti di classe entro il codice Java.

La modalità di introduzione delle informazioni d'ingresso (relative al modello topologico e/o all'osservazione) può essere interattiva o batch (o, eventualmente, entrambe le forme possono essere supportate). Non è richiesta la creazione di una interfaccia grafica.

Il modello topologico del sistema considerato è passibile di rappresentazione grafica. La visualizzazione di tale rappresentazione non è un requisito.

NOTA

I requisiti (funzionali e non) di cui sopra sono deliberatamente espressi a un alto livello di astrazione (ad esempio, non si sono imposti limiti alle dimensioni del sistema considerato né si è richiesto il salvataggio dei modelli topologici in memoria di massa) al fine di consentire agli ingegneri del software di fornire un'interpretazione personale, che comporta sempre l'aggiunta di ulteriori requisiti.

ESTENSIONI FUTURE

Si elencano di seguito alcuni possibili punti di estensione dell'applicazione, non perché i requisiti ad essi relativi siano soddisfatti ma perché anticipare i cambiamenti è un importante principio di progettazione. Altri possibili cambiamenti potranno essere previsti dagli ingegneri del software.

- Il sistema di interazione potrebbe divenire grafico (se non lo è già).
- Il modello topologico del sistema potrebbe essere creato e/o visualizzato graficamente.
- L'architettura esterna potrebbe diventare distribuita.
- L'applicazione potrebbe determinare non solo diagnosi (minimali) con mascheramento comprensive degli ingressi ma anche non comprensive degli ingressi, per calcolare le quali si utilizzano sottoinsiemi interni, conflitti strutturali e conflitti strutturali con mascheramento tutti privati dei nodi di ingresso. Si noti che, in tal caso, il grafo relativo al sistema potrebbe divenire disconnesso dopo l'ideale soppressione di tutti i nodi di ingresso (e dei flussi da essi uscenti).
- All'applicazione potrebbe essere richiesto di estrarre tutte le diagnosi, anziché solo un sottoinsieme delle stesse, corrispondente a una singola scelta operata al punto b) di pag. 4, oppure tutte e sole le diagnosi (minimali) con mascheramento che godono di particolari proprietà (ad esempio, con cardinalità massima stabilita).

Richieste

Agli studenti è richiesto di realizzare un'applicazione software che soddisfi i requisiti sopra esposti. Ogni gruppo (costituito al più da tre persone) dovrà:

- 1) indicare (e giustificare) in forma scritta il modello di processo adottato;
- 2) produrre la documentazione di progetto, comprendente
 - casi d'uso (comprensivi dell'espressione dei requisiti aggiuntivi), sia in forma testuale, sia in forma di diagramma UML,
 - diagramma UML delle classi,
 - diagrammi UML comportamentali (opzionali),e qualsiasi altra specifica ritenuta opportuna;
- 3) redigere un breve manuale d'uso;
- 4) presentare in formato sia cartaceo, sia elettronico quanto richiesto ai precedenti punti da 1 a 3;
- 5) consegnare codice sorgente + codice interpretabile + (preferibilmente) codice eseguibile;
- 6) preparare ed effettuare un'intera dimostrazione.