

A.A. 2013-2014
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. La SECONDA PARTE del progetto sarà illustrata dal docente nel secondo semestre.

Una opportuna ulteriore 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 in due modi diversi la probabilità di guasto dei componenti di un sistema e confronti i risultati ottenuti.

DOMINIO APPLICATIVO

Modello. Descrive la struttura di controllo del sistema considerato (ad es. un sistema software). È un diagramma di attività UML che può contenere esclusivamente:

- Nodo iniziale
- Nodo finale
- Decisione (branch) senza guardie
- Merge
- Fork
- Join
- Flussi
- Azioni

Per essere valido il modello deve contenere almeno i nodi iniziale e finale, un'azione, il flusso in essa entrante e quello da essa uscente. Un modello valido richiede inoltre che tutti i merge e join siano utilizzati esplicitamente.

Caso di test. Configurazione singola appartenente al dominio dei valori di ingresso del sistema.

Insieme dei casi di test (o test suite). Insieme di tutti i casi di test considerati in una sessione di lavoro.

Cammino di esecuzione globale (relativo a un caso di test). È il cammino di esecuzione (consistente col modello) che si attua in corrispondenza del caso di test. Tale cammino inizia necessariamente dal nodo iniziale e termina solitamente nel nodo finale, ma può verificarsi il caso che l'esecuzione si arresti prima di arrivare al nodo finale a causa di guasti che interrompono l'esecuzione stessa.

Dal momento che il modello può contenere strutture di controllo parallele, il cammino di esecuzione globale nel caso generale non è una sequenza di esecuzione di azioni bensì una sequenza dove ogni elemento può essere un'azione o un blocco di esecuzione corrispondente a una struttura parallela (che prevede l'esecuzione di sequenze di azioni concorrenti). Il cammino di esecuzione globale è quindi la sequenza di tutte le azioni e/o blocchi paralleli che il sistema esegue in corrispondenza del caso di test.

Blocco parallelo (relativo a un caso di test). Insieme di tutti i tratti di cammini di esecuzione concorrenti inerenti a una struttura di controllo parallela del modello (cioè a una porzione del modello compresa fra un fork e il join relativo), eseguiti in corrispondenza del caso di test. Qualora il modello contenga più strutture fork-join

annidate, il cammino di esecuzione globale conterrà i corrispondenti blocchi paralleli annidati.

Cammino di esecuzione parziale (relativo a un caso di test). È un sottocammino del cammino di esecuzione globale relativo al test. Esso compendia il tratto di esecuzione compreso fra il nodo iniziale e un flusso designato.

Insieme (di azioni) di un cammino. È l'insieme di azioni contenute in un cammino (globale o parziale), incluse quelle relative a ogni blocco parallelo interamente contenuto nel cammino considerato.

Prova (relativa a un caso di test). Una prova consiste nel dare in ingresso al sistema il caso di test considerato e rilevare (in uno o più punti dell'esecuzione) se il comportamento del sistema sia corretto o meno. Una rilevazione può avvenire in corrispondenza del flusso entrante nel nodo finale del modello e/o di un qualsiasi flusso di uscita di un'azione. Ciascuna rilevazione assume il valore convenzionale OK, se si riscontra un comportamento corretto, il valore convenzionale KO altrimenti.

Una prova viene rappresentata come una coppia (t, S) dove t è il caso di test eseguito ed S , detto *insieme di copertura*, è un insieme di coppie (*insieme del cammino, valore della rilevazione*), una coppia per ogni rilevazione effettuata, dove l'*insieme del cammino* è

quello relativo al cammino di esecuzione (parziale o globale) che ha portato sino al flusso in corrispondenza del quale è stata riscontrato il *valore della rilevazione*.

Prove omogenee. Due prove $(t1, S1)$ e $(t2, S2)$, dove $t1$ e $t2$ appartengono al medesimo test suite, si dicono omogenee se $S1 = S2$. La relazione di omogeneità fra prove porta all'individuazione di *classi di equivalenza* delle prove effettuate in corrispondenza del test suite considerato, dove il numero di prove che cadono in ciascuna classe è detto *cardinalità* della classe stessa.

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.

Diagnosi (minimale senza mascheramento) relativa a una prova. Data una prova (t, S) , una singola diagnosi d (minimale senza mascheramento) a essa relativa è un hitting set minimale della collezione di tutti gli insiemi

$$\underline{C}_i = C_i \setminus (\cup C_{ok}),$$

uno per ogni elemento $(C_i, KO) \in S$, dove $\cup C_{ok}$ è l'unione di tutti i C_j tali che $(C_j, OK) \in S$. In generale, esistono più hitting set minimali di tale collezione, dunque esistono più diagnosi corrispondenti alla stessa prova.

Si noti che ogni prova appartenente alla stessa classe di equivalenza di prove omogenee porta all'individuazione delle stesse diagnosi.

Si noti anche che, data una prova (t, S) , se per ogni coppia $(C_h, v_h) \in S$ è vero che $v_h = OK$, l'unica diagnosi (minimale globale senza mascheramento) relativa alla prova è quella vuota, cioè l'insieme D di tutte le diagnosi relative alla prova è il singoletto $\{ \emptyset \}$.

Probabilità di guasto di un'azione secondo il metodo 1 relativamente a una singola prova di una classe di equivalenza di prove omogenee. Sia D l'insieme di tutte le diagnosi (minimali senza mascheramento) relative a una prova (t, S) appartenente alla classe di equivalenza considerata, dove, come noto, una diagnosi d è un insieme di azioni.

Se l'azione A_k non appartiene ad alcun C_i tale che $(C_i, v_i) \in S$, allora la sua probabilità di guasto relativamente sia alla prova considerata sia alla classe di equivalenza è ignota.

Se l'azione A_k appartiene ad almeno un C_i tale che $(C_i, OK) \in S$, allora la sua probabilità di guasto relativamente alla prova considerata è nulla.

Altrimenti, indicato con n_k il numero di diagnosi $d \in D$ a cui A_k appartiene, la probabilità di guasto di A_k relativamente alla prova considerata è nulla se $n_k = 0$, mentre è

$$p(A_k) = \frac{\sum_{d | A_k \in d} \frac{1}{|d|}}{n_k} \text{ se } n_k \neq 0.$$

Se la probabilità di guasto di A_k , $p(A_k)$, relativa a una prova non è ignota, la probabilità di guasto di A_k relativa all'intera classe di equivalenza è il prodotto di $p(A_k)$ per la cardinalità della classe di equivalenza stessa ed è indicata come $\underline{p}(A_k)$.

Probabilità di guasto di un'azione secondo il metodo 1 relativamente a un test suite. Se la probabilità di guasto di un'azione A_k è ignota per tutte le classi di equivalenza delle prove appartenenti al test suite, allora la probabilità di guasto di A_k è ignota relativamente all'intero test suite.

Altrimenti la probabilità di guasto $P(A_k)$ di un'azione A_k relativamente al test suite è la media delle probabilità di guasto non ignote di A_k relative alle classi di equivalenza del test suite considerato (intesa come sommatoria delle probabilità di guasto non ignote di A_k divisa per la sommatoria delle cardinalità delle classi di equivalenza del test suite considerato secondo cui la probabilità di guasto di A_k non è ignota).

Coefficiente di Ochiai (relativo a due vettori binari). Affiancati idealmente due vettori V_p e V_q di uguale lunghezza, contenenti solo valori binari, si indica con n_{ab} il numero di valori pari ad a di V_p a cui corrisponde un valore pari a b in V_q . dove $a, b \in \{0,1\}$. Il coefficiente di Ochiai è il valore $\frac{n_{11}}{\sqrt{(n_{11}+n_{10})(n_{11}+n_{01})}}$.

Probabilità di guasto di un'azione secondo il metodo 2 relativamente a un test suite. Si immagini di creare una matrice M dove la generica k -ma colonna è relativa a un'azione A_k distinta del modello e ogni riga è relativa a un elemento $(C_j, v_j) \in S$ dove (t, S) è una prova relativa al test t che appartiene al test suite considerato. In ogni slot M_{jk} di tale riga si inserisce un 1 se l'azione A_k appartiene a C_j , 0 altrimenti.

Il numero di colonne di M è pari al numero di azioni distinte del modello (una colonna per ciascuna azione e viceversa).

Il numero di righe di M è pari al numero di coppie (C_j, v_j) , non necessariamente diverse l'una dall'altra, contenute negli insiemi di coperture di tutte le prove a cui i casi di test del test suite hanno dato luogo: a ciascuna riga corrisponde una coppia e viceversa.

Si immagini altresì di creare un vettore V affiancato (verticalmente) a M . Lo slot j -mo di V , adiacente alla riga di M relativa all'elemento $(C_j, v_j) \in S$, contiene 1 se $v_j = KO$, 0 altrimenti.

La probabilità di guasto di un'azione A_k è il coefficiente di Ochiai relativo a V e alla colonna di M corrispondente all'azione A_k .

Si noti che, applicando il metodo 2, la probabilità di guasto non è ignota per alcuna azione del modello. Tale probabilità vale 0 per tutte le azioni la cui probabilità è ignota secondo il metodo 1.

Elenco ordinato delle azioni per probabilità di guasto decrescente. Un volta applicato uno dei due metodi di calcolo delle probabilità di guasto delle azioni del modello, si può creare l'elenco per valore decrescente di tali probabilità, un valore di probabilità distinto per riga, dove a ogni probabilità è affiancato l'insieme delle azioni aventi tale probabilità di guasto.

Intervallo di posizione di un'azione entro un elenco ordinato. Dato un elenco ordinato come sopra definito, per intervallo di posizione di un'azione A_k si intende l'intervallo di valori interi il cui estremo inferiore è la somma, incrementata di una unità, del numero di azioni che si trovano nelle righe superiori dell'elenco rispetto alla riga in cui si trova A_k e il cui estremo superiore è pari al valore dell'estremo inferiore incrementato del numero di azioni contenute nella stessa riga di A_k , A_k esclusa.

Distanza fra elenchi ordinati relativa a una singola azione. Dati due elenchi ordinati di azioni e un'azione che appartiene a entrambi, la distanza fra i due elenchi relativa a tale azione è nulla se i due intervalli di posizione dell'azione nei due elenchi si sovrappongono almeno per un valore. Se non è così, tale distanza è la differenza fra l'estremo inferiore dell'intervallo di valori interi più elevati e l'estremo superiore dell'altro intervallo.

Distanza totale fra elenchi ordinati di azioni. È la sommatoria delle distanze di tutte le azioni contenute in entrambi gli elenchi.

Si noti che se i due elenchi sono stati prodotti rispettivamente con il metodo 1 e con il metodo 2, le azioni condivise fra i due sono quelle presenti nell'elenco creato in base al metodo 1.

Distanza media per azione fra elenchi ordinati. È la distanza totale divisa per il numero di azioni condivise fra i due elenchi ordinati.

REQUISITI FUNZIONALI

L'utente fornisce in ingresso all'applicazione:

- a) il modello del sistema considerato,
- b) l'insieme di copertura di una prova per ogni classe di equivalenza del test suite considerato,
- c) la cardinalità di ogni classe di equivalenza del test suite considerato.

L'applicazione deve innanzi tutto controllare che il modello fornito sia valido (oppure forzare l'immissione di modelli validi), impedendo che siano fornite in ingresso le informazioni di cui ai precedenti punti b) e c) se il modello non lo fosse. Per ogni insieme di copertura fornito, l'applicazione deve controllare che ogni elemento dello stesso sia valido (oppure forzare l'introduzione solo di elementi validi). Un elemento (*insieme del cammino, valore della rilevazione*) è singolarmente valido se *valore della rilevazione* è OK o KO, se *insieme del cammino* è un sottoinsieme delle azioni del modello e contiene azioni che effettivamente possono appartenere allo stesso cammino di esecuzione (globale o parziale). Inoltre un insieme di copertura è valido se tutti gli insiemi del cammino dei suoi elementi possono effettivamente essere insiemi di azioni eseguite dato un unico test.

L'applicazione produce in uscita:

- per ogni prova di cui al precedente punto b), l'insieme delle diagnosi (minimali senza mascheramento) corrispondente, corredato dalla cardinalità della classe di equivalenza a cui la prova appartiene,
- i due elenchi delle azioni ordinati per probabilità decrescente prodotti rispettivamente in base al metodo 1 e al metodo 2,
- la distanza globale fra i due elenchi ordinati suddetti e la distanza media per azione,
- (opzionalmente) un report stampabile , ad esempio un semplice testo, contenente sia le informazioni in ingresso sia quelle in uscita.

Sessioni diverse possono prendere in considerazione modelli di sistemi diversi e/o prove diverse e/o cardinalità diverse.

Si noti che, sebbene le probabilità di guasto delle azioni secondo il metodo 2 siano state definite in termini di una matrice e un vettore, non è necessario utilizzare tali strutture dati al fine di effettuare il calcolo delle probabilità stesse.

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 e/o agli insiemi di copertura e/o alla cardinalità delle classi di equivalenza) può essere interattiva o batch (o, eventualmente, entrambe le forme possono essere supportate). Non è richiesta la creazione di una interfaccia grafica.

Il modello del sistema considerato, essendo un diagramma UML, 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 modello né si è richiesto il salvataggio in memoria di massa di modelli, insiemi di copertura e diagnosi relative a ogni insieme di copertura) 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 a 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 potrebbe essere creato e/o visualizzato graficamente.
- L'architettura esterna potrebbe diventare distribuita.
- Gli attuali due metodi di calcolo delle probabilità di guasto delle azioni potrebbero cambiare nel tempo e/o il loro numero potrebbe aumentare.
- Il modo di stimare quantitativamente la differenza fra i due elenchi prodotti in uscita potrebbe cambiare nel tempo o si potrebbero introdurre stime aggiuntive.

Richieste (relative alla PRIMA PARTE DEL PROGETTO)

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 di installazione e 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.

CONSEGNA DEL MATERIALE

Circa le modalità (obbligatorie) di consegna contestuale del materiale relativo a entrambe le parti del progetto, si rimanda al file 0_prologo_2013_14 e si ricorda che sia il codice, sia la documentazione da consegnare ai fini del superamento della prova orale devono essere suddivisi nelle due parti in cui il progetto è strutturato.