

Tesi di Laurea Magistrale in Ingegneria Informatica

Implementazione e sperimentazione di criteri di terminazione per processi di testing random

Anno Accademico 2015/2016

relatore

Ch.mo Prof. Porfirio Tramontana

correlatore

Ing. Nicola Amatucci

candidato

Andrea Messalino

matr. M63/000163

Cos'è il testing automatico?

- **È l'utilizzo di speciali strumenti per il collaudo del software senza il ricorso a test effettuati a mano (in genere molto tedious e dispendiosi in termini di tempo)**
- **Più efficiente rispetto a soluzioni manuali, ma serve un metodo affidabile per scegliere gli eventi da testare...**

Cos'è il testing casuale?

- **Generazione automatica di input aleatori ed indipendenti tra loro**
- **Facile da usare, ma si rischia di avere test simili o ridondanti**
- **Criteri di terminazione per auto-arrestare il processo**

Stato dell'arte

- **Tecniche di testing casuale:**
 - **Pure Random Techniques**
 - **Adaptive Random Techniques**
 - **Guided Random Techniques**
 - **Fuzz Techniques**
- **Valutazione delle prestazioni:**
 - **comparazioni sulle metriche (codice coperto, errori rilevati, ...)**
 - **modelli sia teorici sia empirici**
- **Criteri di terminazione:**
 - **limite massimo di tempo e/o di eventi**
 - **interruzione dopo un dato numero di sessioni senza miglioramenti**
 - **modelli statistici e/o di uso operativo**
 - **«effetto saturazione»**

- [illegible]

Criteri di terminazione

- **Criteri di terminazione adottati:**

- $CT1 = 1 \Leftrightarrow PCC(ST_i) = PUC(ST) \quad \forall ST_i \in ST$

- $CT2 = 1 \Leftrightarrow \exists ST_i \in ST : PCC(ST_i) = PUC(ST)$

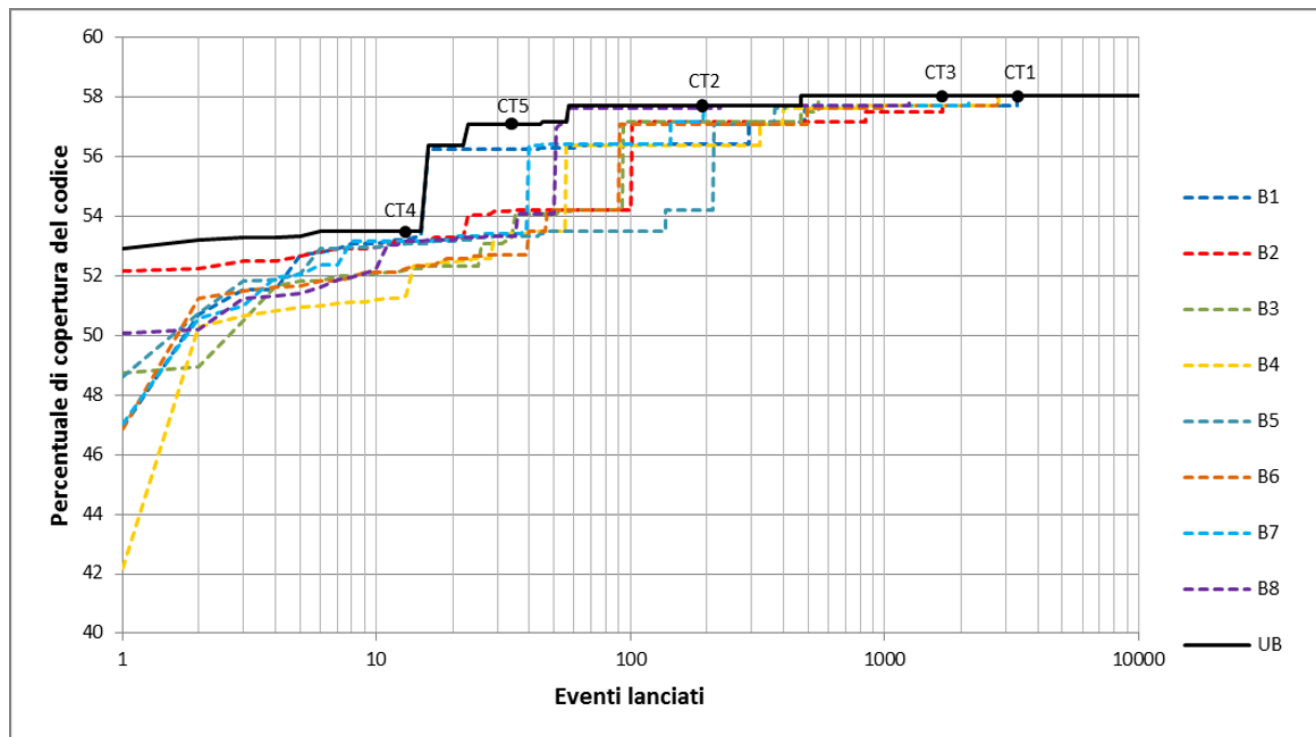
- $CT3 = 1 \Leftrightarrow \exists STP: |STP| = \left\lceil \frac{k}{2} \right\rceil, PCC(ST_i) = PUC(ST) \quad \forall ST_i \subseteq STP$

- $CT4 = 1 \Leftrightarrow \frac{TEST_{sv}}{TEST_{tot}} > \alpha$

- $CT5 = 1 \Leftrightarrow TIME > NCL \cdot TCL$

Criteri di terminazione

- Scelta dei criteri incentrata sull'effetto saturazione
- In quale punto ci conviene arrestare il processo?



Sperimentazione

- **Obiettivi finali:**

- **O1 - valutare la capacità d'interruzione dei criteri**
- **O2 - confrontare le prestazioni della tecnica di testing casuale al variare del criterio di terminazione**

- **Domande di ricerca:**

- **D1 – A quale livello di copertura i criteri prescelti riescono a fermare i test?**
- **D2 - Quanto differiscono le prestazioni della tecnica di test al variare del criterio di terminazione (sia come copertura, sia come tempo trascorso)?**

Processo di testing

- **Struttura iterativa: loop finché la condizione di terminazione non viene soddisfatta**

```
1: procedure TESTPROC(app,nses,tse,precond,postcond,tc[])
2:   cycle ← 0;
3:   loop START :
4:     cycle ← cycle + 1;
5:     for (i = 0; i < nse; i++) do
6:       setPrecond(app,precond);
7:       seed ← genRandSeed();
8:       cov ← executeTest(app,tse,seed);
9:       covses[i] ← mergeCovses(cov,covses[i]);
10:      covun ← mergeCovun(covses[i],covun);
11:      setPostcond(app,postcond);
12:    end for
13:    output[cycle] ← genOutput(cycle,covses[],covun);
14:    for (j = 0; j < lenght(tc[]); j++) do
15:      tc[j] ← FALSE;
16:      check ← evalTC(tcform[j],output[cycle]);
17:      if check = TRUE then tc[j] ← TRUE;
18:    end if
19:  end for
20:  tercond ← evalTerCond(tc[]);
21:  if tercond = TRUE then goto EXIT;
22:  else goto START;
23:  end if
24:  EXIT:
25: end loop
26: end procedure
```

Software per il testing

- **Randoop**: generatore di casi di unità di test per programmi Java



- **EMMA**: strumento per l'analisi della copertura del codice sorgente di applicazioni Java



- **Starter**: file batch modellato sul processo di testing per l'avvio e l'arresto automatico dei test in base ai criteri scelti (con esecuzione automatica di Randoop e di EMMA)



Impostazioni della sperimentazione

- Test lanciati tramite file batch esterno (con una o più chiamate a Starter)
- 12 programmi testati, selezionati all'interno del corpus SF110 di Evosuite
- 8 sessioni di test per processo, con un periodo di 20 secondi tra due valutazioni della copertura e dei criteri di terminazione
- Terminazione del processo quando tutti i criteri risultano soddisfatti

Applicazione	Nome	Descrizione	N. classi	N. metodi	N. blocchi	N. linee
APP1	Battlecry	Generatore casuale di brani musicali	15	130	9550	1831
APP2	dcParseArgs	Libreria Java per la gestione di args[]	6	21	654	110
APP3	FalseLight	Sistema di monitoraggio delle reti	14	32	1189	200
APP4	JCLO	Analizzatore di variabili in classi Java	4	43	1094	202
APP5	JGAAP	Framework grafico	23	95	3140	638
APP6	JIPA	Interprete Java	5	36	1488	228
APP7	Noen	Framework per analisi dinamiche	18	71	1353	-
APP8	Playgame ²	Videogioco	8	87	3964	808
APP9	SAXPath	Strumento per applicazioni XPath	12	257	3115	955
APP10	Sugar	Strumento per programmatori Java	14	80	2089	407
APP11	Transmitter Locator	Rilevatore di trasmettitori	6	39	1097	238
APP12	Tullibee	Componente per il trading online	19	204	8402	1798

Metriche delle 12 applicazioni sotto esame

Analisi dei risultati (obiettivo O1)

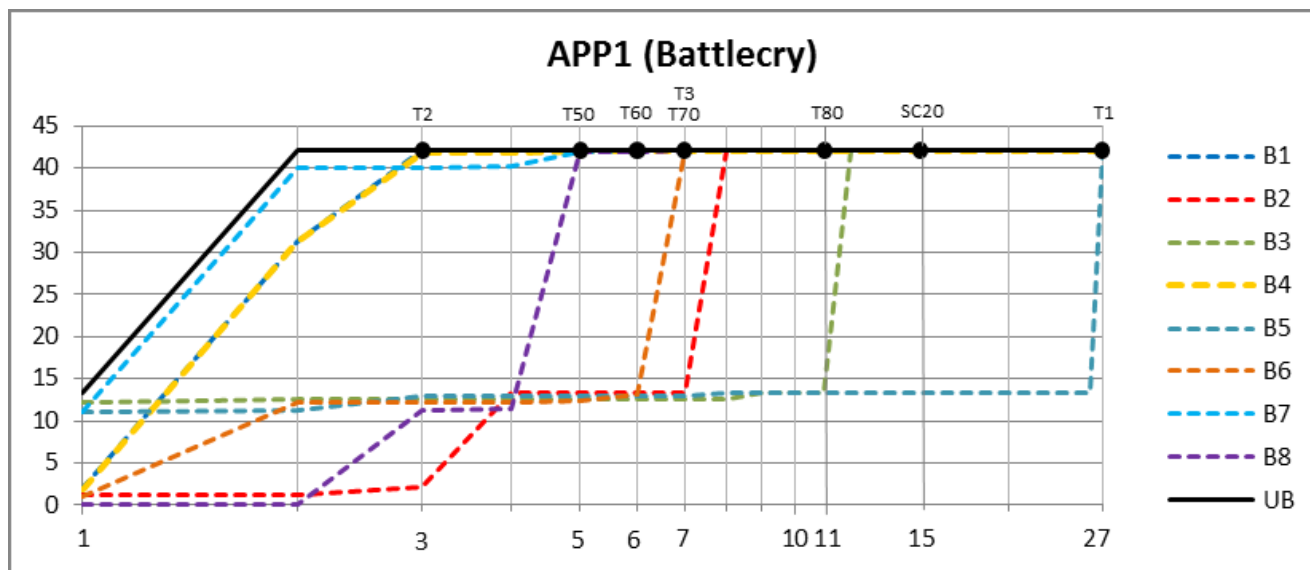
- Dai 5 tipi di criteri adottati sono stati ricavati 10 criteri di terminazione: T1 (CT1), T2 (CT2), T3 (CT3), T50 (CT4 con $\alpha=0,5$), T60 (CT4 con $\alpha=0,5$), T70 (CT4 con $\alpha=0,7$), T80 (CT4 con $\alpha=0,8$), SC20 (CT5 con $TCL=20$), SC40 (CT5 con $TCL=40$) e SC60 (CT5 con $TCL=60$)
- Copertura del codice compresa in generale tra il 40 e il 60% del totale
- La copertura più elevata è sempre stata registrata adottando il criterio T1

App.	T1	T2	T3	T50	T60	T70	T80	SC20	SC40	SC60
APP1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP2	0,0	1,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP4	0,0	10,2	8,3	10,2	10,2	10,2	10,2	10,2	10,2	10,2
APP5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP8	0,0	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,0	0,0
APP9	0,0	0,3	0,0	4,6	0,3	0,3	0,3	1,7	0,9	0,9
APP10	0,0	1,0	0,0	1,0	1,0	1,0	0,0	1,0	0,3	0,0
APP11	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
APP12	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6	0,0	0,0

Differenze (in percentuale) tra le coperture individuate dai 10 criteri adottati ed i livelli massimi di copertura

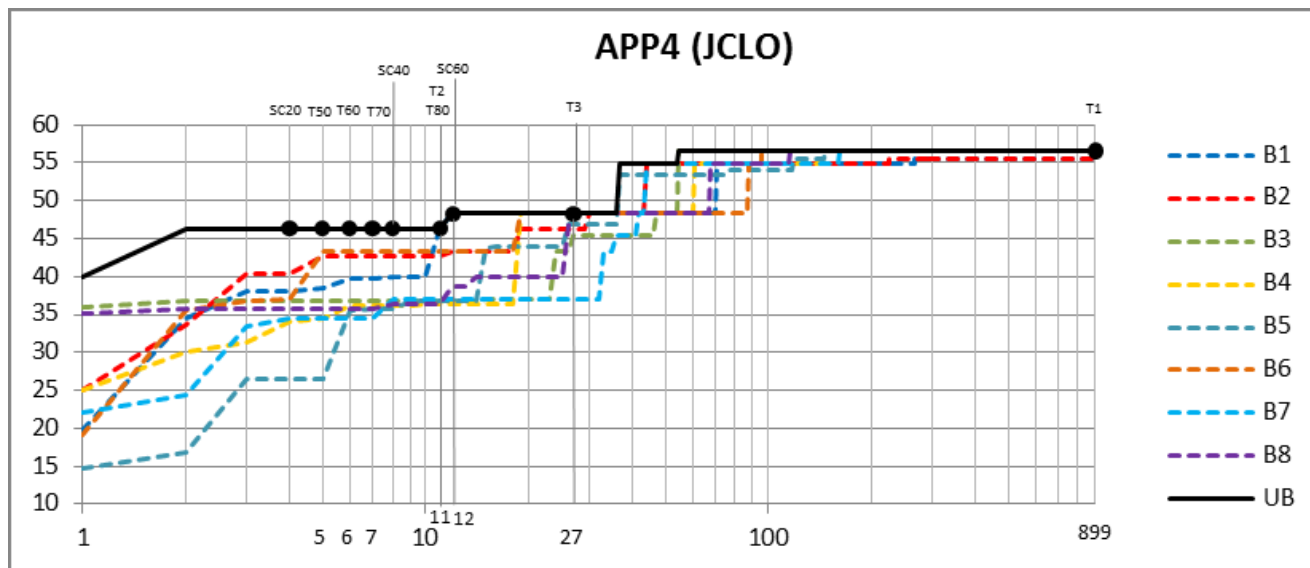
Analisi dei risultati (obiettivo O2)

- I grafici hanno «fotografato» due tipi di andamenti negli esperimenti:
 - processo breve con progresso rapido e quasi continuo della copertura
 - processo lungo con progressi lenti e discontinui della copertura
- Privilegiare l'accuratezza di copertura o la velocità per arrestare i test?



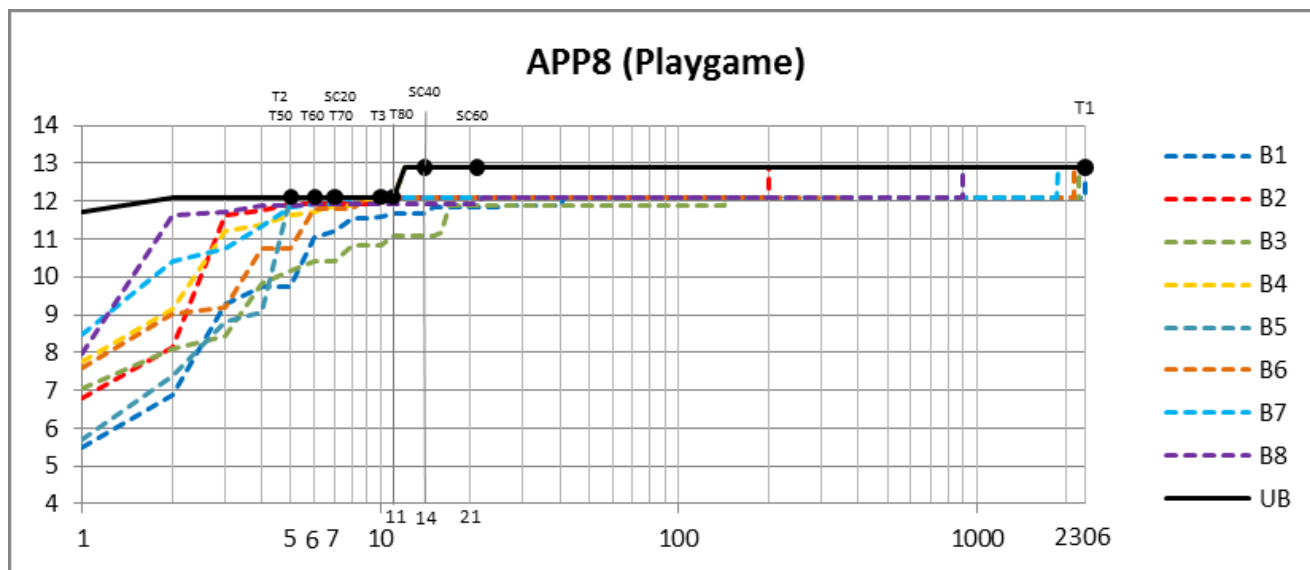
Analisi dei risultati (obiettivo O2)

- I grafici hanno «fotografato» due tipi di andamenti negli esperimenti:
 - processo breve con progresso rapido e quasi continuo della copertura
 - processo lungo con progressi lenti e discontinui della copertura
- Privilegiare l'accuratezza di copertura o la velocità per arrestare i test?



Analisi dei risultati (obiettivo O2)

- I grafici hanno «fotografato» due tipi di andamenti negli esperimenti:
 - processo breve con progresso rapido e quasi continuo della copertura
 - processo lungo con progressi lenti e discontinui della copertura
- Privilegiare l'accuratezza di copertura o la velocità per arrestare i test?



Conclusioni

- CT1 ottimo in copertura, ma più gravoso per tempistica (viceversa per CT2)
- CT3 a metà strada tra CT1 e CT2 (con gli aspetti positivi di entrambe)
- CT4 e CT5 si posizionano attorno alle altre 3 in base ai valori di α e TCL

Sviluppi futuri

- Rendere più intuitivo ed efficiente il file batch Starter
- Raccogliere campioni più significativi di applicazioni da testare
- Parallelizzare i test con l'impiego di più calcolatori distribuiti