



UNIVERSITÀ DI PISA

SOFTWARE VALIDATION AND VERIFICATION

FABIO GADDUCCI

| | |
|-----------------|-------------|
| Anno accademico | 2023/24 |
| CdS | INFORMATICA |
| Codice | 660AA |
| CFU | 9 |

| Moduli | Settore/i | Tipo | Ore | Docente/i |
|--------------------------------------|-----------|---------|-----|----------------|
| SOFTWARE VALIDATION AND VERIFICATION | INF/01 | LEZIONI | 72 | FABIO GADDUCCI |

Obiettivi di apprendimento

Conoscenze

Lo scopo del corso è quello di introdurre tecniche per la verifica e la validazione delle proprietà del software, sia estraendo un modello da un programma e verificandolo, sia testando il software prima della successiva distribuzione, sia dotando il software di strumenti in grado di monitorare la sua esecuzione.

Modalità di verifica delle conoscenze

L'esame consiste in una prova scritta (ed eventualmente una orale) e un seminario su un argomento avanzato presentato durante il corso. La prova scritta e l'eventuale prova orale hanno l'obiettivo di accertare che lo studente abbia acquisito sufficiente familiarità con i concetti di base legati alla logica temporale lineare e alla sua verifica. Il seminario quello di verificare la capacità dello studente nell'affrontare temi allo stato dell'arte riguardo l'uso di metodi formali per la specifica e la verifica del software.

Capacità

Al termine del corso lo studente sarà in grado di utilizzare tecniche formali e tool allo stato dell'arte per quel che riguarda la specifica e la verifica software, con particolare attenzione per il model checking.

Modalità di verifica delle capacità

Alcune lezioni sono completamente dedicate alla soluzione guidata di esercizi al fine di far meglio comprendere sia le potenzialità espressive (rispetto alle proprietà del software) dei linguaggi di specifica introdotti nel corso che le difficoltà che si incontrano nello sviluppare tool per la verifica automatica di tali proprietà.

Comportamenti

Al termine del corso lo studente avrà acquisito maggiore consapevolezza sull'importanza di descrivere formalmente le proprietà desiderate del software e posto in contesto alcune delle tecniche attualmente utilizzate per quel che riguarda sia la specifica che la verifica di dette proprietà.

Modalità di verifica dei comportamenti

Le esercitazioni permettono di verificare la sensibilità del gruppo rispetto all'importanza nella pratica dei temi trattati durante il corso, mentre la prova seminariale consente di evidenziare la capacità del singolo nell'inquadramento generale dello specifico argomento trattato.

Prerequisiti (conoscenze iniziali)

Conoscenze di base di logica proposizionale e di teoria degli automi.

Indicazioni metodologiche

Il corso si basa su lezioni frontali e sessioni di esercizi.

Programma (contenuti dell'insegnamento)

Specifiche delle proprietà del software

- Asserzioni



UNIVERSITÀ DI PISA

- Proprietà degli invarianti, safety e liveness, fairness

Model Cheking di Proprietà Linear Time

- Sistemi di transizione e grafi del programma
- Controllo delle proprietà di safety regolari
- Controllo delle proprietà omega-regolari con gli automi di Büchi

Logiche Linear Time

- Forme normali positive
- Fairness
- Model checking di formule LTL

Logiche Computational Tree

- Model checking di formule CTL e CTL*

Bibliografia e materiale didattico

Christel Baier and Joost-Pieter Katoen. Principles of Model Checking. MIT Press, 2008.

Indicazioni per non frequentanti

Slide e testi di esercizi sono resi disponibili sul sito del corso.

Modalità d'esame

Verifica intermedia (in alternativa, una prova orale) e un seminario su uno degli argomenti avanzati discussi durante il corso.

Pagina web del corso

<http://pages.di.unipi.it/gadducci/SVV-23/>

Altri riferimenti web

Nessuno.

Note

Nessuna.

Ultimo aggiornamento 26/07/2023 11:21