



## Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	<b>Formal Methods</b>	
Corso di studio	Laurea Magistrale in Computer Science	
Anno Accademico	2022/23	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	6 CFU	
Settore Scientifico Disciplinare	INF/01	
Lingua di erogazione	Inglese	
Anno di corso	Primo	
Periodo di erogazione	1 <sup>^</sup> semestre	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	<a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/</a>	

## Docente/i

Nome e cognome	Berardina De Carolis
Indirizzo mail	berardina.decarolis@uniba.it
Telefono	-
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.560, 5 <sup>^</sup> piano.
Sede virtuale	Piattaforma ADA - <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a>
Sito web del docente	-
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Lunedì dalle 11:30 alle 13.00. In giorni diversi solo su appuntamento.

## Syllabus

<b>Obiettivi formativi</b>	<p>Definire e comprendere i diversi metodi formali, in particolare la logica (logica proposizionale, logica dei predicati, logica temporale). Capire come utilizzare le logiche temporali per il model checking. In seguito, gli studenti acquisiranno conoscenze sulle reti di Petri, il linguaggio formale più diffuso nella modellazione del comportamento dei sistemi. Oltre alle reti di Petri base, il corso esaminerà anche le reti di Petri temporizzate e le reti di Petri di alto livello.</p> <p>Durante le ore di laboratorio, gli studenti si eserciteranno con i metodi formali più importanti, utilizzando strumenti e sviluppando programmi. In particolare, YAKINDU sarà utilizzato per la formalizzazione e l'esecuzione di diagrammi di stato; NuSMV sarà utilizzato per il model checking e diversi strumenti saranno utilizzati per imparare la modellazione e la verifica formale con le reti di Petri.</p>
<b>Prerequisiti</b>	Si consiglia di riprendere i concetti di Teoria dei linguaggi formali dall'insegnamento del corso di linguaggi di programmazione.
<b>Contenuti di insegnamento (Programma)</b>	Formal Methods in Computer Science è un corso che mira a fornire agli studenti un'introduzione completa ai metodi formali e alla loro applicazione nelle specifiche e nella verifica di software. La prima parte del corso introduce alcuni fondamenti dei metodi formali, tra cui le macchine a stati finiti e le espressioni regolari. La seconda parte si concentra sulla logica per specificare le proprietà dei sistemi. Vengono trattate la logica proposizionale, la logica dei predicati, la logica temporale e il model checking. La terza parte presenta le reti di Petri, il linguaggio formale più diffuso nella modellazione del comportamento dei sistemi. Oltre alle reti di Petri regolari, questa parte esamina anche le reti di Petri temporizzate e le reti di Petri di alto livello.



	<p>Viene inoltre mostrata l'applicazione dei metodi formali alle applicazioni del mondo reale.</p> <p>Moduli:</p> <ol style="list-style-type: none"> <li>1. Panoramica riassuntiva dei principali metodi formali introdotti nei corsi della laurea triennale: <ul style="list-style-type: none"> <li>- Set Theory e Funzioni</li> <li>- Finite State Machine e State Chart</li> <li>- Espressioni regolari e linguaggi</li> </ul> </li> <li>2. Introduzione alla logica: <ul style="list-style-type: none"> <li>- Propositional Logic</li> <li>- Predicate Logic</li> <li>- Temporal Logic LTL</li> </ul> </li> <li>3. Formal Verification by Model Checking</li> <li>4. Reti di Petri <ul style="list-style-type: none"> <li>- Petri Nets</li> <li>- Timed Petri Nets</li> <li>- Colored Petri Nets</li> </ul> </li> </ol> <p>Laboratorio:</p> <p>YAKINDU -&gt; Tool per la formalizzazione e l'esecuzione di state charts; NuSMV per il model checking Diversi tool saranno usati per comprendere la modellazione e verifica formale di sistemi con le Reti di Petri Presentazione dei casi di studio su cui lavorare durante i laboratori</p> <p>Incontri di approfondimento su temi innovativi: Process Mining, Blockchain, Smart Home, Smart Environment</p>		
<b>Testi di riferimento</b>	<p>Formal Methods in Computer Science, Jiacun Wang, William Tepfenhart ISBN 9781498775328, Chapman &amp; Hall</p> <p>Slides del docente</p> <p>Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo <a href="https://opac.uniba.it/easyweb/w8018/index.php?">https://opac.uniba.it/easyweb/w8018/index.php?</a> e contattare la biblioteca per concordare il prestito.</p>		
<b>Note ai testi di riferimento</b>	<p>Il libro di testo e' integrato con gli appunti presi a lezione e con le slide del docente disponibili sulla piattaforma di e-learning Ada.</p>		
<b>Organizzazione della didattica</b>			
<b>Ore</b>			
Totali	Didattica frontale	Laboratorio e esercitazioni	Studio individuale
150 ore	32 ore	30 ore	88 ore
<b>CFU/ETCS</b>			
6 CFU	4 CFU	2 CFU	
<b>Metodi didattici</b>			
<p>Lezioni frontali, esercitazioni, laboratori ed attività autonome e di gruppo in aula e a casa (come descritto nel programma). Gli studenti non frequentanti possono lavorare singolarmente prendendo accordi con il docente.</p>			



Risultati di apprendimento previsti	
<b>Conoscenza e capacità di comprensione</b>	Conoscere i concetti fondamentali inerenti i metodi formali in computer science. Conoscere i principali metodi e tecniche per la modellazione di sistemi e programmi mediante metodi formali. Comprendere quali sono i pro e i contro di ogni metodo proposto. Comprendere la capacità di ogni metodo proposto di verificare alcune proprietà del sistema modellato.
<b>Conoscenza e capacità di comprensione applicate</b>	Conoscere e mettere in pratica i metodi appresi attraverso laboratori ed esercitazioni su ognuno di essi.
<b>Competenze trasversali</b>	<i>Autonomia di giudizio</i> Acquisire autonomia di giudizio sulle scelte relative alla modellazione e verifica delle proprietà di un sistema/software attraverso il caso di studio scelto. <i>Abilità comunicative</i> Essere in grado di comunicare in modo appropriato i metodi e le tecniche applicate nel caso di studio durante la presentazione dello stesso. <i>Capacità di apprendere in modo autonomo</i> Sviluppare capacità di progettare ed applicare in autonomia i metodi appresi nel contesto del corso approfondendo in maniera autonoma i concetti e metodi presentati.
Valutazione	
<b>Modalità di verifica dell'apprendimento</b>	La verifica dell'apprendimento terra' in considerazione sia gli aspetti teorici, attraverso la prova scritta, che pratici, attraverso la presentazione e discussione di un progetto assegnato. Verranno valutate le capacità di problem solving e di utilizzo delle metodologie e strumenti descritti durante il corso per la modellazione e verifica di proprietà del sistema/software scelto. Il progetto di gruppo sarà svolto durante le ore di laboratorio ed integrato nelle ore di studio individuale. Il carico di lavoro di lavoro sarà adeguato al numero di componenti del gruppo (max 3). Sarà consentito anche lo svolgimento individuale del progetto su richiesta dello studente. La prova scritta puo' essere svolta previa consegna della documentazione relativa al progetto di gruppo.  I risultati saranno comunicati attraverso la piattaforma Esse3.  Lo studente potrà decidere di sostenere le due prove in due appelli differenti. I risultati delle prove saranno conservati fino al termine dell'AA in corso.
Criteri di valutazione	<ul style="list-style-type: none"><li>• <i>Conoscenza e capacità di comprensione:</i> Lo studente dovrà dimostrare di conoscere e di aver compreso i concetti fondamentali alla base dei metodi formali, i principali metodi applicabili alla progettazione e verifica formale dei sistemi.</li><li>• <i>Conoscenza e capacità di comprensione applicate:</i> Lo studente dovrà dimostrare di essere in grado di applicare i metodi formali in contesti applicativi specifici dell'informatica.</li><li>• <i>Autonomia di giudizio:</i> Lo studente dovrà dimostrare di saper formulare un proprio giudizio sulle scelte relative al progetto sviluppato.</li></ul>



	<ul style="list-style-type: none"><li>• <b>Abilità comunicative:</b> Lo studente dovrà dimostrare di saper comunicare le conoscenze acquisite nonché motivare le proprie scelte relative ai vari step dello sviluppo del progetto in modo appropriato.</li><li>• <b>Capacità di apprendere:</b> Lo studente dovrà dimostrare di aver acquisito la capacità di approfondire in autonomia gli argomenti attinenti i metodi formali e la loro applicazione.</li></ul>
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	Saranno valutati i risultati di apprendimento previsti. Il voto massimo sarà 30/30 e lode. La prova di laboratorio consisterà nel presentare il progetto sviluppato nel corso delle lezioni e avrà una valutazione da 6 a 10/10. Mentre la prova scritta, che mira a testare le conoscenze teoriche, avrà un punteggio minimo di 12 e massimo di 20/10. La lode sarà attribuita se in entrambe le prove di dimostra di avere conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.
<b>Altro</b>	<p>Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti:</p> <ul style="list-style-type: none"><li>• <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li><li>• <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica">https://www.uniba.it/it/ricerca/dipartimenti/informatica</a></li><li>• <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></li></ul> <p>I programmi degli insegnamenti sono disponibili qui:</p> <ul style="list-style-type: none"><li>• <a href="https://programmi.di.uniba.it/">https://programmi.di.uniba.it/</a></li></ul> <p>Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:</p> <ul style="list-style-type: none"><li>• <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li></ul> <hr/> <p>Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.</p> <p>Gli studenti potranno unirsi al team del corso A.A. 2022/23 <b>bq32s42</b> utilizzato per scopi didattici, al quale aderisce anche il docente: <a href="https://t.me/+THISCYIG1nBz_OO9">https://t.me/+THISCYIG1nBz_OO9</a></p> <p>Link al corso sulla piattaforma e-learning del dipartimento ADA: <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></p>



UNIVERSITÀ  
DEGLI STUDI DI BARI  
ALDO MORO

DIPARTIMENTO  
DI  
INFORMATICA

--	--