UNIVERSITÀ degli STUDI di CATANIA

DIPARTIMENTO DI INGEGNERIA ELETTRICA ELETTRONICA E INFORMATICA
Corso di laurea magistrale in Ingegneria informatica
Anno accademico 2015/2016 - 2° anno

LINGUAGGI E TRADUTTORI

ING-INF/05 - 9 CFU - 1° semestre

Docente titolare dell'insegnamento

VINCENZA CARCHIOLO

Email: vincenza.carchiolo@unict.it
Edificio / Indirizzo: Edificio 3 - piano V - stanza 11
Telefono: 095 7382359
Orario ricevimento: martedi e venerdi dalle ore 11 alle ore 12

OBIETTIVI FORMATIVI

Conoscere le tecniche di analisi e traduzione automatica dei linguaggi di programmazione e dei linguaggi naturali. Sapere progettare compilatori e strumenti che consentono la generazione automatica dei diversi moduli funzionali di un compilatore. Avere la capacita di applicare metodi e tecnologie proprie della progettazione dei compilatori a una vasta gamma di problemi che riguardano più in generale la progettazione e lo sviluppo di software.

CONTENUTI DEL CORSO

Struttura di un compilatoreAnalisi lessicale - Analisi sintattica - Analisi semantica - Generazione del codice intermedio - Ottimizzazione del codice - Generazione del codice - Gestione della tabella dei simboli - Raggruppamento delle fasi in passate - Strumenti di sviluppo per i compilatori - Evoluzione dei linguaggi di programmazione - Applicazioni della tecnologia dei compilatori - Fondamenti dei linguaggi di programmazioneAnalisi LessicaleRuolo dell’analizzatore lessicale - Buffering dell’ingresso - Descrizione dei token - Estensioni delle espressioni regolari - Riconoscimento dei token - Il generatore di analizzatori lessicali Lex - Automi a stati finiti - Dalle espressioni regolari agli automi -Progettazione di un generatore di analizzatori lessicali - Ottimizzazione dei riconoscitori basati su DFAAnalisi SintatticaIntroduzione - Grammatiche libere dal contesto - Scrittura di una grammatica -Parsing top-down - Parsing bottom-up - Introduzione al parsing LR: LR(0), SLR(1), LR(1), LALR(1) - Uso delle grammatiche ambigue - Generatori di parser – Gestione degli errori sintatticiTraduzione guidata dalla sintassiDefinizioni guidate dalla sintassi - Ordine di valutazione delle SDD - Applicazioni della traduzione guidata dalla sintassi - Schemi di traduzione postfissi - Implementazione di SDD L-attribuiteGenerazione del codice intermedioVarianti degli alberi sintattici -- Codice a tre indirizzi - Tipi e dichiarazioni -Traduzione delle espressioni - Controllo dei tipi - Flusso di controllo - Backpatching - Statement switch - Codice intermedio delle procedureSupporto run-timeOrganizzazione della memoria - Allocazione della memoria a stack - Accesso ai dati non locali sullo stack - Gestione dello heap - Introduzione alla garbage collection - Introduzione alla garbage collection basata su tracce - Concetti avanzati di garbage collectionGenerazione del codiceProgettazione di un generatore di codice - Il linguaggio target - Indirizzi nel codice target - Blocchi base e diagrammi di

flusso - Ottimizzazione dei blocchi base - Semplice generatore di codice - Ottimizzazione a finestra - Allocazione e assegnamento dei registri - Selezione delle istruzioni mediante riscrittura di alberi - Generazione di codice ottimo per le espressioni - Generazione di codice e programmazione dinamica - Ottimizzazioni indipendenti dall'architetturaPrincipali opportunità di ottimizzazione – Introduzione all'analisi data-flow - Fondamenti dell'analisi data-flow - Propagazione delle costanti - Eliminazione della ridondanza parziale - Cicli nei diagrammi di flusso - Analisi basata sulle regioni -Analisi simbolicaParallelismo a livello d'istruzioneArchitetture dei microprocessori - Vincoli di scheduling del codice –Scheduling nei blocchi base - Scheduling globale del codice - Software pipeliningOttimizzazione di parallelismo e localitàConcetti di base - Multiprocessori - Parallelismo nelle applicazioni - Granularità del parallelismo - Parallelismo a livello di ciclo - Località dei dati - Parallelizzazione per righe - Spazio delle iterazioni - Indici di array affini - Riuso dei dati - Analisi della dipendenza dai dati negli array - Sincronizzazione tra cicli paralleli - Pipelining -Ottimizzazione della località Analisi interproceduraleConcetti fondamentali e motivazione - Rappresentazione logica dei diagrammi di flusso - Datalog - Effetti dell'invocazione di un metodo - Binary decision diagram

---

TESTI DI RIFERIMENTO

Aho Alfres V. – Lam Monica S. – Sethi Ravi - Ulmann J. Compilers: principles, techniques & tools, Second Edition, Addison Wesley
Steven Muchnick, Advanced Compiler Design Implementation, Morgan Kaufmann.
Fischer, Le Blanch, "Crafting a compiler", The Benjamin Cummings Company, Inc

---