UNIVERSITEIT GENT

Studiefiche

Vanaf academiejaar 2026-2027

Compilers (E018520)

Cursusomvang *(nominale waarden; effectieve waarden kunnen verschillen per opleiding)*

Studiepunten 6.0 **Studietijd 180 u**

Aanbodsessies en werkvormen in academiejaar 2026-2027

A (semester 2)	Engels	Gent	zelfstandig werk	0.0u
			groepswerk	0.0u
			werkcollege	
			hoorcollege	

Lesgevers in academiejaar 2026-2027

De Sutter, Bjorn	TW06	Verantwoordelijk lesgever

Aangeboden in onderstaande opleidingen in 2026-2027	**stptn**	**aanbodsessie**
Educatieve Master of Science in de wetenschappen en technologie(afstudeerrichting informatica)	6	A
Master of Science in Computer Science Engineering	6	A
Master of Science in de industriële wetenschappen: informatica	6	A
Master of Science in de informatica	6	A
Uitwisselingsprogramma industriële wetenschappen: informatica	6	A
Uitwisselingsprogramma informatica (niveau master)	6	A

Onderwijstalen

Engels

Trefwoorden

contextvrije grammatica, afleidingsboom, parsers, abstracte-syntaxbomen, semantische analyse, basisblokken, instructieselectie, registertoewijzing, dataverloopanalyse, controleverloopanalyse, codegeneratie, optimalisatie, ondersteuning managed talen, geheugensanering, pijplijnen, inroosteren, technieken voor hoog-niveautalen, just-in-time-compilatie

Situering

Een computer werkt met machinetaal terwijl een programmeur schrijft in een hogere programmeertaal. Compilers vormen de verbinding tussen de programmeur en de computer. Een compiler zal het bronprogramma op een correcte en efficiënte manier vertalen naar machine-instructies. Dit gebeurt in twee grote stappen. in het front-end gedeelte worden lexicale en parsing technieken gebruikt om het programma om te vormen naar een intermediaire voorstelling. In het back-end gedeelte worden verschillende optimisatietechnieken en codegeneratietechnieken toegepast om het programma zo goed mogelijk af te beelden op de onderliggende architectuur. In dit vak worden de theoretische basisconcepten gekoppeld aan de implementatie van een wereldwijd gebruikte compiler en aan een aantal recente ontwikkelingen in het domein.

Inhoud

- Inleiding: Datastructuren voor compilers, Overzicht van fasen in de compilatie
- Lexicale analyse: Tokens en reguliere expressies, Eindige automaten
- Parsing: Contextvrije grammatica, Predictieve en LR-parsers
- Abstracte syntax: Semantische acties, Abstracte syntaxboom
- Semantische analyse: Symbooltabellen, Typechecking
- Activatierecords: Stapelvensters en locale variabelen, Doorgeven van parameters

- Intermediaire code: Voorstellingsboom, Vertaling
- Basisblokken en traces: Canonische bomen, Voorwaardelijke sprongen
- Instructieselectie: Algoritmen voor instructieselectie
- Liveness analyse en registerallocatie: Levensduurvergelijkingen, Graafkleuring en registertoewijzing
- Datastroomanalyse: Datastructuren voor datastroomvoorstelling, Datastroomalgoritmen
- Lusoptimisaties: Reduceerbare grafen, Lustransformaties, Controleverloopanalyse
- Codegeneratie: pijplijnen en inroosteren
- Optimalisatie van geheugengebruik: prefetching, lustransformaties
- Ondersteuning voor hoge-niveautalen: geheugensanering, polymorfisme, klassenhiërarchieën
- Just-in-time compilatietechnieken
- Capita selecta van state-of-the-art compilertechnieken

Begincompetenties

- Kennis van progammeertalen, programmeren in C
- Basiskennis van computerarchitectuur (de software-hardware-interface)

Eindcompetenties

1 De structuur van een compiler begrijpen en duiden.
2 De constructie en werking van automata voor lexicale analyze begrijpen en duiden.
3 De constructie en werking van verschillende parsertechnieken en hun complexiteit begrijpen, duiden, en toepassen.
4 Typechecking op abstracte syntaxbomen begrijpen en duiden.
5 Diverse representaties van code, data, afhankelijkheden en eigenschappen begrijpen, duiden, en opbouwen.
6 Afhankelijkheidsanalyse voor levensduur-analyse en code-optimisatie gebruiken, duiden, en toepassen.
7 Instructieselectie-algoritmen vergelijken en toepassen.
8 Graafkleuring algoritmes voor registerallocatie begrijpen, duiden, en toepassen.
9 Intermediaire voorstellingen voor de generatie van machine-onafhankelijke code begrijpen, duiden, en hanteren.
10 Diverse analyses van lussen en lustransformaties voor code-optimisatie begrijpen, duiden, en toepassen.
11 Diverse dataverloopanalyses en erop gebaseerde code-optimalisaties begrijpen, duiden, en toepassen.
12 Het statisch inroosteren van code en pijplijnen van lussen begrijpen, duiden, en toepassen.
13 De implementatie van ondersteunende technieken voor hogere programmeertalen, zoals geheugensanering, polymorfisme, en overerving, begrijpen, duiden, en toepassen.
14 Het programmeren van basisfunctionaliteit van compilers (lexer, parser, semantische analyse, codegeneratie, etc.) in een labo-omgeving.
15 Het programmeren van basisfunctionaliteit in een wereldwijd populaire, complexe compiler, gebruikmakend van de interne interfaces ervan.

Creditcontractvoorwaarde

Toelating tot dit opleidingsonderdeel via creditcontract is mogelijk na gunstige beoordeling van de competenties

Examencontractvoorwaarde

Dit opleidingsonderdeel kan niet via examencontract gevolgd worden

Didactische werkvormen

Groepswerk, Werkcollege, Hoorcollege, Zelfstandig werk

Toelichtingen bij de didactische werkvormen

In de hoorcolleges wordt de theoretische onderbouw en interne werking van compilers besproken, en wordt de werking van basisalgoritmes ingeoefend.
De studenten krijgen voor de "practica" zeven programmeeropdrachten die ze deels tijdens begeleide werkcolleges (online of in de laptopklassen), deels daarbuiten moeten uitvoeren. Ze moeten hun software-oplossing, alsook soms een klein verslag, indienen. Ze werken in groepjes van twee. Samenwerking met andere groepen is niet toegestaan. Na het indienen van de werkstukken kunnen de

studenten individueel uitgenodigd worden tot een kort evaluatiegesprek waarop ze over hun werkstukken ondervraagd worden om te checken of ze de beoogde competenties verworven hebben. Een aantal practica handelen om onderdelen van LLVM, een van de meest gebruikte compilers ter wereld.

Studiemateriaal

Type: Handboek

Naam: Moderne Compilerimplementatie in C
Richtprijs: € 70
Optioneel: ja
Taal : Engels
Auteur : Andrew W. Appel
ISBN : 978-0-52160-765-0
Aantal pagina's : 556
Alternatief : Slides en eigen notities
Oudst bruikbare editie : 2004
Online beschikbaar : Nee
Beschikbaar in de bibliotheek : Ja
Beschikbaar via studentenvereniging : Ja
Gebruik en levensduur binnen het opleidingsonderdeel : intensief
Gebruik en levensduur binnen de opleiding : eenmalig
Gebruik en levensduur na de opleiding : niet
Bijkomende info: De inhoud van deze cursus sluit nauw aan bij dit boek. Ter voorbereiding op het examen volstaan de beschikbare slides en de aantekeningen van de student uit de hoorcolleges en oefenzittingen. Dit boek biedt uitgeschreven uitleg van de cursusinhoud voor studenten die dergelijk aanvullend studiemateriaal verkiezen. Tweemaal in de cursus wordt de studenten gevraagd zelf een korte sectie uit het boek te bestuderen. Hiervoor is het lenen van een exemplaar van een bibliotheek of een collega voldoende.

Type: Slides

Naam: Compilers - Hoorcolleges Theorie
Richtprijs: Gratis of betaald door opleiding
Optioneel: nee
Taal : Engels
Aantal slides : 455
Oudst bruikbare editie : 2025-2026
Beschikbaar op Ufora : Ja
Online beschikbaar : Ja
Beschikbaar in de bibliotheek : Nee
Beschikbaar via studentenvereniging : Nee
Bijkomende info: -

Referenties

- Modern compiler implementation in C, Andrew W. Appel with Maia Ginsburg, Cambridge University Press, ISBN 052158390X, LCCN QA76.73.C15.A63
- LLVM Manual, http://llvm.org/docs/

Vakinhoudelijke studiebegeleiding

Evaluatiemomenten

periodegebonden en niet-periodegebonden evaluatie

Evaluatievormen bij periodegebonden evaluatie in de eerste examenperiode

Schriftelijke evaluatie

Evaluatievormen bij periodegebonden evaluatie in de tweede examenperiode

Schriftelijke evaluatie

Evaluatievormen bij niet-periodegebonden evaluatie

Mondelinge evaluatie, Werkstuk

Tweede examenkans in geval van niet-periodegebonden evaluatie

Examen in de tweede examenperiode is enkel mogelijk in gewijzigde vorm

Toelichtingen bij de evaluatievormen

De niet-periodegebonden evaluatie bestaat uit zeven werkstukken in de vorm van software, sommige met korte verslagen erbij, die gespreid zijn over het semester.

Het verslag en elk werkstuk moet voor de start van het volgende practicum of op een opgegeven datum ingediend worden. Na het indienen van het verslag volgt er mogelijk ook een mondelinge evaluatie waarop de student ondervraagd wordt over de gebruikte methodiek en de behaalde resultaten.
Voor de werkstukken werken de studenten mogelijk in groepjes van twee. Aan de hand van een peer assessment over hun bijdrage aan het werkstuk en verslag, en eventueel na een gesprek hierover bij tegenstrijdige assessments, wordt bepaald of beide studenten al dan niet dezelfde punten krijgen.
Als er een mondelinge evaluatie plaats vindt, is dit individueel. Elke student in een groep is verantwoordelijke voor de hele ingediende oplossing, en kan over alles daarin ondervraagd worden.
Niet op tijd ingediende verslagen/oplossingen zonder geldige reden van afwezigheid (zoals een doktersbriefje) worden met een nul gekwoteerd voor dat onderdeel van de niet-periodegebonden evaluatie.
Er mag geen code gegenereerd door AI-tools ingediend worden.
Wanneer code ingediend wordt die afgeleid is van code uit publieke documentatie (bv. tutorials), moet dit aangegeven worden door de indiener. Het werk wordt dan meer beoordeeld op basis van de gemaakte aanpassingen (volledigheid, correctheid, ...) dan op het geheel van de ingediende code.
De niet-periodegebonden evaluatie verloopt gelijkaardig voor de tweede examenperiode. De student wordt dan geëvalueerd op basis van nieuwe werkstukken met een gelijkaardige belasting als in de eerste periode, maar moet die zeker individueel maken. De verslagen ervan zullen een week voor het examen ingediend moeten worden. Vlak na dat schriftelijk periodegebonden examen volgt dan een mondelinge evaluatie over de ingediende werkstukken.
De periodegebonden evaluatie is schriftelijk met gesloten boek.

Eindscoreberekening

Verdeling van de punten: 3/5 voor de periodegebonden evaluatie en 2/5 voor de niet-periodegebonden evaluatie.
Speciale voorwaarde: men moet minstens 10/20 halen op de niet-periodegebonden evaluatie en op de periodegebonden evaluatie om voor dit opleidingsonderdeel te kunnen slagen. Indien dit niet het geval is voor één of beide delen en de met de gewichten berekende totaalscore 10/20 of meer bedraagt, wordt de finale totaalscore 9/20.
Indien een student in de eerste examenperiode geslaagd is voor een van beide onderdelen, kan die ervoor kiezen om die punten over te dragen naar de tweedekansexamenperiode.

Faciliteiten voor werkstudenten

Mogelijkheid tot vrijstelling van aanwezigheid met vervangende opdracht na overleg met verantwoordelijke lesgever. Mogelijkheid tot mondeling examen met schriftelijke voorbereiding op ander tijdstip binnen het academiejaar. Mogelijkheid tot feedback na afspraak tijdens en na kantooruren.