Voortgezette Informatica

2024

Intro

Tot dusver:

- programmeren in Python (en MatLab)
- nadruk op onder de knie krijgen programmeren
- implementeren van kleine problemen (TCP1, TCP2)
- oefening in data-science (TCP2)
- geen fundamentele beschouwingen over computers, programmeertalen, ed

Voortgezette Informatica

Docent: dr. Jan Martin Jansen

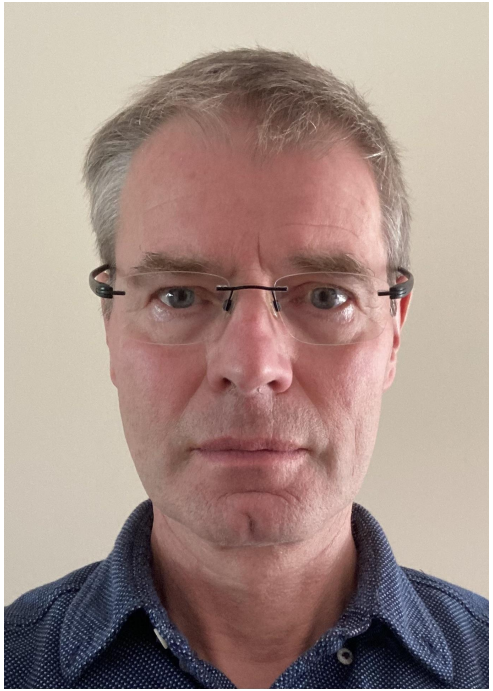
Toetsing

- Inleveropdrachten (onv/vol)
- Schriftelijk Tentamen(3 uur)
- Geslaagd: Opdracht voldoende en tentamen > 5.5

6 weken: 10 lessen (2 lessen uitloop)

Alle materiaal online in ELO

Oefentools ook online



Vragen?

- wat is een computer?
- wat is een programmeertaal?
- wat is een algoritme?
- wat hebben ze met elkaar te maken?

Meer uitgebreid: doelstellingen Voortgezette Informatica

- wat is nu eigenlijk een computer?
- kunnen we daar een abstract (meer wiskundig) model voor maken?
- welke alternatieve modellen zijn er?
- in hoeverre verschillen deze modellen in wat er mee berekend kan worden?
- is er een soort optimum: een model waarmee alles kan wat überhaupt met een computer berekend kan worden?
- zijn er, op het eerste oog, berekenbare problemen, waarvan geen algoritme op een computer mogelijk is?
- wat is de rol van programmeertalen?
- wanneer zijn 2 talen equivalent?
- hoe zit het met algoritmen?

Voorlopige antwoorden

Een algoritme is een stappenplan om een berekening uit te voeren.

Een programmeertaal is een formalisme om algoritmen in uit te drukken.

Computers kunnen algoritmen uitvoeren die zijn uitgedrukt in een programmeertaal.

Een programmeertaal is dus de verbindende factor tussen algoritme en computer

Onderwerpen in dit vak

1. eindige automaten en reguliere expressies
2. stapelautomaten en contextvrije grammatica’s
3. Turingmachines en berekenbaarheid
4. afbeelding van programmeertalen op computers
5. beschouwingen over recursie en iteratie

Werkwijze

Niet te veel focus op de theorie

Veel praktische voorbeelden

Veel opgaven en soms wat programmeren

Leerdoelen Voortgezette Informatica 2024

Na afloop van de cursus kan de student:

1. uitleggen wat er met de begrippen: computer, programma en algoritme bedoeld wordt met voorbeelden
2. beschrijven welk soort problemen met een eindige automaat kunnen worden opgelost
3. een deterministische eindige automaat kunnen omzetten naar een Python programma
4. een eindige automaat construeren bij een gegeven reguliere expressie en omgekeerd
5. voor een beschreven contextvrije taal een grammatica construeren
6. voor een contextvrije grammatica de bijbehorende stapelautomaat construeren
7. ontleedbomen voor een grammatica kunnen maken
8. fundamentele computermodellen zoals Turing machines beschrijven en de (on)mogelijkheden hiervan aangeven
9. voor een eenvoudige probleem een Turingmachine construeren
10. uitleggen wat met de universaliteit van Turing machines bedoeld wordt
11. aangeven welke soorten problemen wel of niet m.b.v. een rekenautomaat (computer) opgelost kunnen worden en minimaal 2 voorbeelden geven van niet berekenbare problemen
12. het halting-probleem kunnen beschrijven
13. beschrijven wat de essentiële componenten van een moderne computerarchitectuur zijn en waar ze voor dienen aan de hand van een virtueel model computer met een eigen instructieset (jmvm)
14. aanduiden wat de rol van de stack en de heap in moderne computerarchitectuur is en dit met voorbeelden kunnen illustreren
15. eenvoudige python programma’s naar jmvm assembly kunnen omzetten
16. het verband kunnen aangeven tussen iteratie en recursie en een iteratie kunnen omzetten naar recursie