IN2505-II Berekenbaarheidstheorie

College 1

Hans Tonino

Algoritmiekgroep
Faculteit EWI – TU Delft

7 april 2009

IN2505 II Berekenbaarheidstheorie

- Docent: Hans Tonino
- Colleges/oefeningen: dinsdag 5 + 6 (EWI-A), vrijdag 1 + 2 (AULA-A)
- Boek: Michael Sipser, *Introduction to the Theory of Computation*, Second, international edition, Thomson, Boston, 2006. ISBN 0-619-21764-2.
- Practicum: ca. 20 uur; 5 opgaven, in groepjes van 2
- Website: Blackboard IN2505 II (2008-2009)

Practicum

- Assistenten: Bert Wolters (coördinatie), Pieter Senster, Ot ten Thije
- Aanwezig: donderdagmiddag 13.00-17.00 vrijdagmiddag 13.00-17.00 uur.
- Elektronisch reserveersysteem: afspraken van 15 min. voor hulp en aftekenen
- Vragen (bij mij): donderdagmiddag 14.00-16.00; alleen voor vragen, geen opgaven aftekenen.
- Zaal DW 1-180 (do) en DW 1-160 (vr), Drebbelweg 5

Practicum: Inschrijven

- Alleen of in groepje van 2.
- email naar: in2505@gmail.com
- Vermeld daarbij: voorletters, achternaam, studienummer ST of MKT
- Eerste opgave krijg je dan toegestuurd.

Practicum: LET OP

- Harde deadline practicum: einde kwartaal
- Zie Blackboard voor deadlines per opgave (komt nog)
- Geen herkansing in de zomer!
- Lever nette uitwerkingen in (pdf of uitgeprint)

Globale indeling stof

- Hoofdstuk 0: Wiskundige voorkennis (komt vanzelf aan de orde)
- Hoofdstuk 3: Turing machines & de Church-Turing-these
- Hoofdstuk 4: Beslisbaarheid, Diagonalisatie & het Halting-probleem
- Hoofdstuk 5: Reduceerbaarheid
- Hoofdstuk 6: Speciaal onderwerp – Recursiestelling,

Algemene leerdoelen

De student

- heeft kennis van de elementaire begrippen uit de berekenbaarheidstheorie en toepassingen daarvan;
- heeft kennis van de uit deze theorie voortvloeiende beperkingen van de informatica;
- kent technieken om deze beperkingen in speciale gevallen aan te tonen en is in staat deze toe te passen;
- is in staat zelf eenvoudige verbanden tussen de genoemde begrippen te vinden.

Kernvraag van dit college

Welke problemen kunnen (niet) algoritmisch worden opgelost?

Deelvragen van kernvraag

- Wat is een probleem?
- Wat is een algoritme?

Wat is een probleem?

- Antwoord ligt in het begrip taal.
- Gegeven een alfabet Σ en een taal $L \subseteq \Sigma^*$. Het met L geassocieerde probleem is de vraag of voor een willekeurig woord $w \in \Sigma^*$ geldt $w \in L$.

Wat is een algoritme?

- Intuïtief: een recept dat in een aantal stappen het antwoord geeft.
- Verschillende formele definities voorgesteld, ook wel berekeningsmodellen genoemd.
- Informatici: denk aan een programma in een programmeertaal!

Berekeningsmodellen

- Turingmachines (Alan Turing, 1936)
- Recursieve functies (o.a. Gödel)
- Random Access Machines
- Combinatoren en λ-calculus (Alonso Church, 1936)
- . . . andere berekenbaarheidsmodellen

Toepassingen modellen

- Turingmachines: Tijd- en ruimtegebruik
  - Berekeningsstappen die even duur zijn.
  - Dataopslag die uniform is.
  - Lijkt meest op "computer".
- Recursieve functies en λ-calculus: Mathematische eigenschappen eenvoudiger te beschrijven.

Opgave

Waarom zit dit college in het curriculum?

Antwoord

- Academisch gevormde informatici dienen te weten wat de grondslagen van "rekenen" zijn.
- Zij dienen ook te weten wat binnen het bereik van dit rekenen valt en wat daarbuiten.
- Dit laatste heeft in sommige gevallen een concrete toepassing.

Recapitulatie alfabet (1)

- Een alfabet is een eindige verzameling van symbolen ofwel karakters.
  Notatie: $\Sigma = \{a, b, c, d, e\}$, $\Gamma = \{0, 1\}$.
- Een woord of string over Σ is een eindige reeks symbolen uit Σ.
- De lengte van een woord x, notatie $|x|$, is gedefinieerd als het aantal symbolen in x.
- Het lege woord ϵ is gedefinieerd als het woord met lengte 0.

Alfabet (2)

- Σ^* duidt de verzameling van alle woorden over Σ aan.
- $\Sigma^+ = \Sigma^* - \{\epsilon\}$.

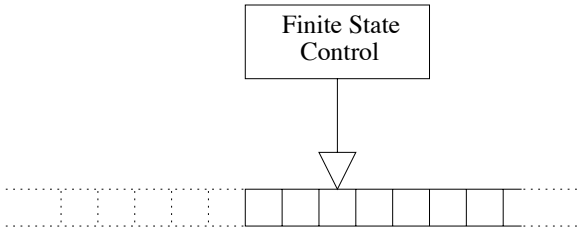
Woorden

- Concatenatie
- Prefix, suffix
- Deelwoord (substring), deelreeks (subsequence)
- Omgekeerd woord x^R.
- Palindroom: $x = x^R$

Taal

- Een taal L over een alfabet Σ is een deelverzameling van Σ^*, ofwel $L \subseteq \Sigma^*$.
- Verschil tussen $L = \emptyset$ en $L = \{\epsilon\}$

Schema Turingmachine



Andere literatuur:
tape ook links onbegrensd

Sipser: tape links begrensd

Formele definitie Turingmachine (Def. 3.3)

Een Turingmachine is een 7-tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ met:

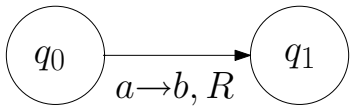
- Q is een eindige verzameling toestanden;
- Σ, het invoeralfabet, is een eindige verzameling die het 'blank'-symbool $\sqcup$ niet bevat;
- Γ, het tape-alfabet, is een eindige verzameling, met $\Sigma \subseteq \Gamma$ en $\sqcup \in \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is de transitiefunctie.
- $q_0, q_{accept}, q_{reject} \in Q$ zijn respectievelijk de begintoestand, de accepterende en de verwerpende toestand met $q_{accept} \neq q_{reject}$.

De transitiefunctie δ

$\delta(q, a) = (r, b, L)$ betekent: als de TM zich in toestand q bevindt en een a leest, dan overschrijft zij deze met b, gaat over in toestand r en verschuift de lees/schrijfkop één positie naar links (L).

δ wordt meestal weergegeven met behulp van een transitiediagram (voorbeelden: zie Sipser p. 146 e.v.).

Diagram Turingmachine



Notatie voor: $\delta(q_0, a) = (q_1, b, R)$.

Andere mogelijkheid voor labels op pijlen: $a, b \to L$.

Voorbeeld TM

Zij $\Sigma = \{a, b, c\}$ en

$$L = \{w \in \Sigma^* \mid w \text{ begint met } a \text{ en eindigt op } c\}.$$

Maak een TM die L "beslist" (diagram + definitie)

Configuraties

Een configuratie van een TM wordt bepaald door:

- haar toestand,
- haar tape-inhoud, en
- de positie van haar lees/schrijfkop.

Notatie configuratie: $u\,q\,v$.

Dit betekent dat de TM zich in toestand q bevindt, de tape-inhoud gelijk is aan uv en dat de kop zich onder het meest linkse symbool van v bevindt.

(Sipser p. 142 e.v.)

Opleveren

Gegeven een TM M met transistiefunctie δ.

Men zegt dat configuratie $ua\,q_i\,bv$ de configuratie $u\,q_j\,acv$ oplevert, als $\delta(q_i, b) = (q_j, c, L)$.

Evenzo levert de configuratie $ua\,q_i\,bv$ de configuratie $uac\,q_j\,v$ op, als $\delta(q_i, b) = (q_j, c, R)$.

(Sipser p. 143)

Speciale configuraties

Een startconfiguratie is een configuratie waarbij de toestand gelijk is aan q_0.

Een accepterende (verwerpende) configuratie is een configuratie waarbij de toestand q_{accept} (q_{reject}) is. Dit zijn de stopconfiguraties.

NB: Het is mogelijk dat een TM op een invoer nooit in een stopconfiguratie terecht komt!

(Sipser p. 143)

Accepterende TM

Een TM M accepteert het woord w d.e.s.d.a. er een reeks configuraties $C_1, C_2, \ldots, C_k$ bestaat, zodanig dat:

- C_1 is een startconfiguratie;
- iedere C_i levert C_{i+1} op $(1 \leq i < k)$; en
- C_k is een accepterende configuratie.

De taal van M is de verzameling woorden die M accepteert, notatie $L(M)$.

(Sipser p. 143/144)

Turing-herkenbaar (Def. 3.5)

Een taal L is Turing-herkenbaar (ook wel: recursief opsombaar) als er een TM M bestaat zodanig dat $L(M) = L$.

Zo'n TM M wordt een herkenner van L genoemd.

NB: Het is nog steeds mogelijk dat M voor bepaalde invoer niet in een stopconfiguratie terecht komt, dus in een 'loop' raakt.

Turing-beslisbaar (Def. 3.6)

Een taal L is Turing-beslisbaar, ofwel kortweg beslisbaar (ook wel: recursief), als er een TM M bestaat zodanig dat $L(M) = L$ en zodanig dat M voor iedere invoer in een stopconfiguratie terecht komt.

Zo'n TM M wordt een beslisser van L genoemd.

Beschrijvingen van TM's

- Formele beschrijving: alle details in termen van δ of diagram.
- Implementatiebeschrijving: beschrijft kopbewegingen en hoe data op de tape(s)* worden geplaatst.
- *High-level* beschrijving: het globale idee zonder implementatiedetails.

* We zullen ook varianten van TM's bespreken die meerdere tapes bezitten.

(Sipser p. 159)

Opgave

Zij $\Sigma = \{a, b\}$ en

$$L = \{w \in \Sigma^* \mid n_a(w) = 2 \cdot n_b(w)\},$$

waarin $n_a(w)$ resp. $n_b(w)$ voor het aantal a's en b's in w staan.

Geef een implementatiebeschrijving van een TM die L beslist.

Aanbevolen opgaven

Sipser p. 161 e.v.: 3.1, 3.2, 3.5, 3.8.