Berekenbaarheidstheorie Samenvatting

Daan Schipper

16 april 2014

Beslisbare talen

- $A_{DFA} = \{\langle B, w\rangle \mid B \text{ is een DFA die } w \text{ accepteert}\}$

  A_{DFA} is beslisbaar door de volgende TM M:
  $M =$ "Op invoer $\langle B, w\rangle$, waar B een DFA is en w een woord:

  1. Simuleer w op B
  2. Als B accepteert, *accepteer*; als B verwerpt, *verwerp*"

- $A_{NFA} = \{\langle B, w\rangle \mid B \text{ is een NFA die } w \text{ accepteert}\}$

  A_{NFA} is beslisbaar door de volgende TM N:
  $N =$ "Op invoer $\langle B, w\rangle$, waar B een NFA is en w een woord:

  1. Converteer NFA B naar een equivalente DFA C met de procedure voor deze conversie beschreven in Theorem 1.39 in Sipser.
  2. Voer TM M uit op de input $\langle C, w\rangle$.
  3. Als M accepteert, *accepteer*; anders *verwerp*."

- $A_{REX} = \{\langle R, w\rangle \mid R \text{ is een reguliere expressie dat woord } w \text{ genereert}\}$

  A_{REX} is beslisbaar door de volgende TM P:
  $P =$ "Op invoer $\langle R, w\rangle$, waar R een reguliere expressie is en w een woord:

  1. Converteer de reguliere expressie R naar een equivalente NFA A met de procedure voor deze conversie beschreven in Theorem 1.54 in Sipser.
  2. Voer TM N uit op invoer $\langle A, w\rangle$.
  3. Als N accepteert, *accepteer*; als N verwerpt, *verwerp*."

- $E_{DFA} = \{\langle A\rangle \mid A \text{ is een DFA en } L(A) = \emptyset\}$

  E_{DFA} is beslisbaar door de volgende TM T:
  $T =$ "Op invoer $\langle A\rangle$, waar A een DFA is:

  1. Markeer de start configuratie van A.
  2. Herhaal totdat er geen nieuwe configuratie worden gemarkeerd:

3. Markeer elke configuratie die een transitie naar zich heeft van een configuratie die al gemarkeerd is
4. Als geen accepterende configuratie is gemarkeerd, *accepteer*; anders, *verwerp*"

- $EQ_{DFA} = \{\langle A, B\rangle \mid A$ en B zijn DFAs en $L(A) = L(B)\}$

  EQ_{DFA} is beslisbaar door de volgende Turinmachine F:
  $F =$ "Op invoer $\langle A, B\rangle$, waar A en B DFAs zijn:

  1. Construeer DFA C met behulp van het symmetrisch verschil.
  2. Voer TM T uit op input $\langle C\rangle$.
  3. Als T accepteert, *accepteer.* Als T verwerpt, *verwerp.*"

- $A_{CFG} = \{\langle G, w\rangle \mid G$ is een CFG dat woord w genereert$\}$

  A_{CFG} is beslisbaar door de volgende TM S:
  $S =$ "Op invoer $\langle G, w\rangle$, waar G een CFG is en w een string:

  1. Converteer G naar een equivalente grammatica in Chomsky normaal vorm.
  2. Noteer alle afleidingen met $2n - 1$ stappen, waar n de lengte is van w; behalve als $n = 0$, noteer dan alle afleidingen met één stap.
  3. Als een van deze afleidingen w genereert, *accepteer*; zo niet, *verwerp.*"

- $E_{CFG} = \{\langle G\rangle \mid G$ is een CFG en $L(G) = \emptyset\}$

  E_{CFG} is beslisbaar door de volgende TM R:
  $R =$ "Op invoer $\langle G\rangle$, waar G een DFG is:

  1. Markeer al eindige symbolen in G
  2. Herhaal totdat geen nieuwe variabelen worden gemarkeerd:
  3. Markeer elke variabele A waar G een regel $A \rightarrow U_1U_2 \dots U_k$ en ieder symbool $U_1, \dots, U_k$ is al gemarkeerd.
  4. Als de start variabele niet gemarkeerd is, *accepteer*; anders, *verwerp.*"

- Ieder contextvrije taal is beslisbaar

- $A_{LBA} = \{\langle M, w\rangle \mid M$ is een LBA dat woord w accepteert$\}$

  A_{LBA} is beslibaar door de volgende TM L:
  $L =$ "Op invoer $\langle M, w\rangle$, waar M een LBA is en w een woord:

  1. Simuleer M op w voor qng^n stappen of totdat het stopt.
  2. Als M is gestopt, *accepteer* als het heeft geaccepteerd en *verwerp* als het heeft verworpen. Als het niet gestopt is, *verwerp.*"

Onbeslisbare Talen

- $A_{TM} = \{\langle M, w\rangle \mid M \text{ is een TM en } M \text{ accepteert } w\}$

  Bewijs uit het ongerijmde. Neem aan dat H een beslisser is voor A_{TM}. Dus H is een TM, waar

  $$H(\langle M, w\rangle) = \begin{cases} \textit{accepteer} & \text{als } w \text{ wordt geaccepteert door } M \\ \textit{verwerp} & \text{als } w \text{ niet wordt geaccepteerd door } M \end{cases}$$

  Vervolgens construeren we D, die het tegengestelde doet van H.
  $D =$ "Op invoer $\langle M\rangle$, waar M een Turinmachine is:

  1. Voer H uit op invoer $\langle M, \langle M\rangle\rangle$.
  2. Output het tegenovergestelde van wat H output. Dus als H accepteert, *verwerp*; en als H verwerpt, *accepteer*

  Samengevat:

  $$D(\langle M\rangle) = \begin{cases} \textit{accepteer} & \text{als } \langle M\rangle \text{ niet wordt geaccepteert door } M \\ \textit{verwerp} & \text{als } \langle M\rangle \text{ wordt geaccepteerd door } M \end{cases}$$

  Als $\langle D\rangle$ wordt gesimuleerd op D krijgen we

  $$D(\langle D\rangle) = \begin{cases} \textit{accepteer} & \text{als } \langle D\rangle \text{ niet wordt geaccepteert door } D \\ \textit{verwerp} & \text{als } \langle DM\rangle \text{ wordt geaccepteerd door } D \end{cases}$$

  Ongeacht wat D doet, het wordt gedwongen om het tegenovergestelde te doen, wat duidelijk een contradictie is. Dus noch TM D noch TM H kunnen bestaan.

- $HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is een TM en } M \text{ stopt op invoer } w\}$

  Bewijs met directe reductie naar A_{TM} uit het ongerijmde. Neem aan dat TM R $HALT_{TM}$ beslist. Construeer TM S om A_{TM} te beslissen, als volgt:
  $S =$ "Op invoer $\langle M, w\rangle$, een codering van een TM M en een woord w:

  1. Voer TM R uit op invoer $\langle M, w\rangle$.
  2. Als R verwerpt, *verwerp*.
  3. Als R accepteert, simuleer M op w totdat het stopt.
  4. Als M heeft geaccepteerd, *accepteer*; als M heeft verworpen, *verwerp*."

  Het is duidelijk dat R $HALT_{TM}$ beslist als S A_{TM} beslist. Omdat A_{TM} onbeslisbaar is, moet $HALT_{TM}$ ook onbeslisbaar zijn.

- $E_{TM} = \{\langle M \rangle \mid M \text{ is een Tm en } L(M) = \emptyset\}$

  Bewijs met directe reductie naar A_{TM} uit het ongerijmde. Definieer M_1 als volgt: M_1 = "Op invoer x:

  1. Als x $\neq$ w, *verwerp*
  2. Als x = w, voer M uit op invoer w en *accepteer* als M accepteert."

  Neem aan dat TM RE_{TM} beslist en definieer TM S die A_{TM} beslist als volgt:
  S = "Op invoer $\langle M, w \rangle$, een codering van en TM M en een woord w:

  1. Gebruik de beschrijving van M en w om de TM M_1 te construeren.
  2. Voer R uit op invoer $\langle M_1 \rangle$.
  3. Als R accepteert, *verwerp*; als R verwerpt, *accepteer*."

  Als R een beslisser zou zijn voor E_{TM}, dan zou S een beslisser zijn voor A_{TM}. Aangezien een beslisser voor A_{TM} niet kan bestaan, is E_{TM} niet beslisbaar.

- $REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is een TM en } L(M) \text{ is een reguliere taal}\}$

  Bewijs met directe reductie naar A_{TM} uit het ongerijmde. Laat R een TM zijn die $REGULAR_{TM}$ beslist en construeer TM S die A_{TM} beslist als volgt:
  S = "Op invoer $\langle M, w \rangle$, waar M een TM is en w een woord:

  1. Construeer de volgende TM M_2.
     M_2 = "Op invoer x:
     1. Als x van de vorm 0^n1^n is, *accepteer*.
     2. Als x niet deze vorm heeft, voer M uit op invoer w en *accepteer* als w wordt geaccepteerd door M.
  2. Voer R uit op invoer $\langle M_2 \rangle$.
  3. Als R accepteert, *accepteer*; als R verwerpt, *verwerp*."

  Als M w accepteert, accepteert M_2 elk woord $\Rightarrow L(M_2)$ is regulier. Als M niet w accepteert, accepteert M_2 alleen woorden van de vorm $0^n1^n \Rightarrow L(M_2)$ is niet regulier. Dus M_2 kunnen we 'voeren' aan R.

  Als R een beslisser zou zijn voor $REGULAR_{TM}$, dan zou S een beslisser zijn voor A_{TM}. Aangezien een beslisser voor A_{TM} niet kan bestaan, is $REGULAR_{TM}$ niet beslisbaar.

- $EQ_{TM} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ en } M_2 \text{ zijn TMs en } L(M_1) = L(M_2)\}$

  Bewijs met directe reductie naar E_{TM} uit het ongerijmde. Laat R een TM zijn die EQ_{TM} beslist en construeer TM S die E_{TM} beslist als volgt:
  S = "Op invoer $\langle M \rangle$, waar M een TM is:

1. Voer R uit op invoer $\langle M, M_1 \rangle$, waar M_1 een TM is die alle invoeren verwerpt.
2. Als R accepteert, *accepteer*; als R verwerpt, *verwerp*."

Als R een beslisser zou zijn voor EQ_{TM}, dan zou S een beslisser zijn voor E_{TM}. Aangezien een beslisser voor E_{TM} niet kan bestaan, is EQ_{TM} niet beslisbaar.

- $E_{LBA} = \{\langle M \rangle \mid M \text{ is een LBA waar } L(M) = \emptyset\}$

  Bewijs met directe reductie naar A_{TM} uit het ongerijmde.
  Laat R een TM zijn die E_{LBA} beslist en construeer TM S die A_{TM} beslist als volgt:
  $S =$ "Op invoer $\langle M, w \rangle$, waar M een TM is en w een woord:

  1. Construeer LBA B van M en w. LBA B werkt als volgt. Wanneer het een invoer x ontvangt, accepteert B als x een accepterende berekeningsgeschiedenis is van M op w.
  2. Voer R uit op invoer $\langle B \rangle$.
  3. Als R verwerpt, *accepteer*; als R accepteert, *verwerp*."

  Als $\langle B \rangle$ wordt geaccepteerd door R, dan geldt $L(B) = \emptyset$. Dus, M heeft geen accepterende berekeningsgeschiedenis van w en accepteert Mw niet. Derhalve, S verwerpt $\langle M, w \rangle$. Evenzo, als $\langle B \rangle$ wordt verworpen door R, dan is de taal van B niet leeg. Het enige woord dat B kan accepteren is een accepterende berekeningsgeschiedenis voor M op w. Derhalve, M moet w accepteren. Derhalve, S accepteert $\langle M, w \rangle$.
  Als R een beslisser zou zijn voor E_{LBA}, dan zou S een beslisser zijn voor A_{TM}. Aangezien een beslisser voor A_{TM} niet kan bestaan, is E_{LBA} niet beslisbaar.

- $ALL_{CFG} = \{\langle G \rangle \mid G \text{ is een CFG en } L(G) = \Sigma^*\}$

  Bewijst met directe reductie naar A_{TM} uit het ongerijmde.

- EQ_{CFG} Bewijs met directe reductie naar ALL_{CFG} uit het ongerijmde. Laat R een TM zijn die EQ_{CFG} beslist en construeer TM S die ALL_{CFG} beslist als volgt: $S =$ "Op invoer $\langle G \rangle$, waar G een CFG is:

  1. Voer R uit op invoer $\langle G, G_1 \rangle$ waar G_1 een CFG is die Σ^* genereert.
  2. Als R accepteert, *accepteer*; als R verwerpt, *verwerp*."

  Als R een beslisser zou zijn voor EQ_{CFG}, dan zou S een beslisser zijn voor ALL_{CFG}. Aangezien een beslisser voor ALL_{CFG} niet kan bestaan, is EQ_{TM} niet beslisbaar.

Definities

Alfabet
: Een **alfabet** is een eindige verzameling Σ van symbolen ofwel karakters.

Woord/String
: 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.

Lege Woord
: Het **lege woord** ϵ is gedefinieerd als het woord met lengte 0.

Taal
: Een **taal** L over een alfabet Σ is een deelverzameling van Σ^*, ofwel $L \subseteq \Sigma^*$.

Turingmachine
: Een **Turingmachine** is een 7-tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, waar Q, Σ, Γ eindige verzamelingen zijn en:

  - Q is de verzameling **toestanden**
  - Σ het **invoeralfabet** dat niet het **lege symbool** $\sqcup$ bevat.
  - Γ is het **tape-alfabet**, waar $\sqcup \in \Gamma$ en $\Sigma \subseteq \Gamma$.
  - $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$ is de **transitiefunctie**.
  - q_0, q_{accept}, $q_{reject} \in Q$ zijn respectievelijk de **begintoestand**, de **accepterende** en de textbfverwerpende toestand met $q_{accept} \neq q_{reject}$.

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

Turing-herkenbaar
: Een taal L is **Turingherkenbaar** als er een TM M bestaat zodanig dat $L(M) = L$. Zo'n TM M wordt een **herkenner** van L genoemd.

Turing-beslisbaar
: Een taal L is **Turing-beslisbaar** als er een TM M bestaat zodanig dat $L(M) = L$ en M voor iedere invoer in een stopconfiguratie terecht komt. Zo'n TM M wordt een **beslisser** van L genoemd.

Multitape Turingmachine
: Een TM kan meerdere tapes hebben. Een k-tape TM kan worden gesimuleerd door een standaard één-tape TM.

Opsommer
: Een TM M is een **opsommer** van taal L als deze, gegeven het lege woord op de invoertape, alle woorden van L print, gescheiden door spaties en in een door de machine bepaalde volgorde.

Symmetrisch Verschil van $L(A)$ en $L(B)$

De verzameling van de elementen die tot een van de twee verzamelingen behoren, maar niet allebei.

$$L(C) = \Big(L(A) \cap \overline{L(B)}\Big) \cup \Big(\overline{L(A)} \cap L(B)\Big)$$

Injectief (one-to-one)

Een functie f is injectief als het nooit twee verschillende elementen projecteert op dezelfde positie - als $a \neq b$, dan $f(a) \neq f(b)$.

Surjectief (onto)

Een functie f is surjectief als het ieder element van B projecteert - er bestaat voor iedere $b \in B$ een $a \in A$ met $f(b) = a$.

Bijectief (correspondence)

Een functie f is bijectief als het zowel injectief als surjectief is.

Oneindig Aftelbaar

Een verzameling A is oneindig aftelbaar als er een bijectie tussen A en $\mathbb{N}$ bestaat.

Aftelbaar

Een verzameling A is aftelbaar als het of eindig is of als het oneindig aftelbaar is.

Overaftelbaar

Een verzameling A is overaftelbaar als het niet aftelbaar is

Diagonalisatie Methode

Standaard opstelling van een diagonalisatie methode: Stel V is aftelbaar. Dan bestaat er een aftelling $f_0, f_1, \ldots, f_n$ van V. Definieer $g(x)$, de diagonaal, zo dat geldt:

1. $g \in V$
2. $g \notin V$

Dit is een tegenspraak, dus V is niet aftelbaar.

co-Turingherkenbaar

Een taal is co-Turingherkenbaar als het het complement is van een Turingherkenbare taal

Accepterende Berekeningsgeschiedenis

Een accepterende berekeningsgeschiedenis van M op w is een reeks configuraties $C_0, C_1, \ldots, C_n$ zodanig dat:

i. C_0 is de start configuratie van M m.b.t. w;
ii. C_i levert C_{i+1} op voor $0 \leq i < n$; en
iii. C_n is een accepterende configuratie van M.

Verwerpende Berekeningsgeschiedenis

Een verwerpende berekeningsgeschiedenis is hetzelfde gedefinieerd als een accepterende berekeningsgeschiedenis, behalve dat C_n een verwerpende configuratie is.

Lineair Begrensde Automaat (LBA)

Een LBA is een TM waarbij de lees/schrijfkop niet het deel van de tape mag/kan verlaten waarop de invoer staat/stond.
Zij M een LBA met q toestanden en g symbolen in het tape-alfabet. Dan zijn er precies qng^n verschillende configuraties van M voor een tape van lengte n.

Berekenbare Functie

Een functie $f : \Sigma^* \to \Sigma^*$ is een berekenbare functie als een TM M, op elke invoer w, stop met alleen $f(w)$ op de tape.

Mapping Reducibility

Een taal A is mapping reducible naar taal B, genoteerd als $A \leq_m B$, als er een berekenbare functie $f : \Sigma^* \to \Sigma^*$ bestaat, waar voor elke w,

$$w \in A \longleftrightarrow f(w) \in B$$

- Als $A \leq_m B$ en B is beslisbaar, dan is A beslisbaar.
- Als $A \leq_m B$ en B is onbeslisbaar, dan is A onbeslisbaar.
- Als $A \leq_m B$ en B is herkenbaar, dan is A herkenbaar.
- Als $A \leq_m B$ en B is niet herkenbaar, dan is A herkenbaar.

Standaard opstelling van een mapping reductie:
Bewijs dat L onbeslisbaar is, waar

$$L = \{\langle M \rangle \mid M \text{ is een TM en er geldt eigenschap P}\}$$

We voeren de mapping reductie $HALT_{TM} \leq_m L$ uit. Definieer de volgende TM N voor een gegeven Turingmachine M met invoer w:
$N =$ "Op invoer x:

1. Simuleer M op w.
2. Magie waardoor M de eigenschap P heeft.

Het bewijs dat f voldoet aan de eisen van mapping reductie. We moeten voldoen aan twee eisen:

- f is een berekenbare functie. De volgende Turingmachine F berekent f:
  $F =$ "Op invoer $\langle M, w \rangle$:
  1. Construeer N m.b.v. M en w.
  2. Print $\langle N, w \rangle$ en stop."
- f is een mapping reductie voor $HALT_{TM} \leq_m$, immers:
  * Als $\langle M, w \rangle \in HALT_{TM}$, dan zal M op invoer w stoppen. Dit betekent dat N aan stap 2 toekomt. Verdere redenatie waaruit volgt dat $f(\langle M, w \rangle) = \langle N, w \rangle \in L$
  * Als $\langle M, w \rangle \notin HALT_{TM}$, dan zal M niet op invoer w stoppen. Dit betekent dat N nooit aan stap 2 toekomt. Verdere redenatie waaruit volgt dat $f(\langle M, w \rangle) = \langle N, w \rangle \notin L$

  Hieruit volgt dat L onbeslisbaar is.

Stellingen

- **Church-Turingthese**: iets is berekenbaar als er een TM voor gevonden kan worden.
- Voor iedere niet-deterministische TM bestaat een equivalente deterministische TM.
- Een taal is beslisbaar dan en slechts dan als het Turingherkenbaar is en co-Turingherkenbaar
- **Stelling van Rice**: laat L een taal zijn van de vorm

$$L = \{\langle M \rangle \mid M \text{ heeft de eigenschap } P\}, \text{ waar}$$

  1. P is niet trivaal; er bestaat tenminste een TM M zodat $\langle M \rangle \in L$, en tenminste een TM N zodat $\langle N \rangle \notin L$.
  2. P is inderdaad een eigenschap van de *talen* van TMs; wanneer $L(M) = L(N)$, geldt $\langle M \rangle \in M$ dan en slecht dan als $\langle N \rangle \in L$.

  Dan is L onbeslisbaar.

Handige dingetjes

- Een voorbeeld van een functie $f : \mathbb{N} \to \mathbb{N}$ die niet *injectief* maar wel *surjectief* is: $f(n) = \lfloor \frac{n}{2} \rfloor$ (hier geldt $f(4) = f(5) = 2$).