IN2505-II Berekenbaarheidstheorie

College 11

Hans Tonino

Algoritmiekgroep
Faculteit EWI – TU Delft

27 mei 2009

Paradox van zelfreproductie

1. Levende wezens zijn machines.
2. Levende wezens kunnen zich reproduceren.
3. Machines kunnen zich niet reproduceren.

(Sipser 6.1, p. 221 e.v.)

Zelfreproductie?

De uitspraken 1 en 3 zijn samen in strijd met uitspraak 2.

...maar is uitspraak 3 te rechtvaardigen?
Kunnen machines zich niet reproduceren?

Programma's en zelfreproductie

... maar programma's kunnen zichzelf reproduceren.

De overvloed aan computervirussen bewijst dit.

De *recursiestelling* "zegt" dat ieder programma kan beschikken over zijn eigen code.

Programma's die teksten printen

Wat doet het volgende "programma"?

Print de volgende tekst: "Het regent."

... en wat doet het volgende programma?

Print het volgende twee keer, de tweede keer tussen aanhalingstekens: "Het regent."

Programma dat eigen code print

Let nu op: Wat is het resultaat van de instructie?

Print het volgende twee keer, de tweede keer tussen aanhalingstekens: "Print het volgende twee keer, de tweede keer tussen aanhalingstekens:"

Berekenbaarheid van een TM P_w die w print

Er bestaat een berekenbare functie $q : \Sigma^* \to \Sigma^*$, zodanig dat voor ieder woord $w \in \Sigma^*$ het resultaat $q(w)$ de beschrijving van een TM P_w is die w op de tape print en stopt. (Lem. 6.1)

Idee bewijs: de machine Q die q berekent, leest invoer w en construeert hieruit een TM P_w door w te "coderen" in de transitiefunctie (machinetabel) van deze TM.

Definitie van Q

Q = **"**Op invoerwoord w:

1. Construeer de volgende TM P_w

P_w = **"**Op invoer x:

1. Wis x van de tape.

2. Schrijf w op de tape.

3. Stop.**"**

2. Geef als output $\langle P_w \rangle$ en stop.**"**

Samenstelling TM's

Als A en B TM's zijn, dan duidt AB (ook wel $A \rightarrow B$) de *samenstelling* van A en B aan:

AB = "Op invoerwoord w:

1. Draai A op invoer w.
2. Als A in een accepterende toestand is gekomen, draai B met als invoer wat A op de tape heeft gezet.
3. *accepteer* als B accepteert; *verwerp* als A of B verwerpt."

Idee TM *SELF* die eigen code print

SELF bestaat uit de *samenstelling* AB van twee delen A en B:

- A print $\langle B \rangle$ op haar tape. Dus $A = P_{\langle B \rangle}$.
- B print de code van A daarvóór met zodanige aanpassingen dat de code van de samenstelling AB, ofwel $\langle AB \rangle$, op de tape komt te staan.

Vraag: Kun je stellen $B = P_{\langle A \rangle}$?

Antwoord

Neen, dat zou een circulaire definitie opleveren.

Oplossing: B *reconstrueert* A uit de uitvoer $\langle B \rangle$ van A, door Q te gebruiken.

Definitie *SELF*

$SELF = AB$ waarbij:

$A = P_{\langle B \rangle}$,

en

B = "Op invoer $\langle M \rangle$:

1. Bereken $q(\langle M \rangle)$ m.b.v. Q en noem deze TM M', m.a.w. $\langle M' \rangle = q(\langle M \rangle)$.
2. Bepaal de *samenstelling* $M'M$ en codeer dit tot $\langle M'M \rangle$.
3. Print dit resultaat en stop."

Let op: B verwijst op geen enkele manier naar A.

Werking *SELF*

- Eerst werkt A $(= P_{\langle B \rangle})$ die $\langle B \rangle$ op de tape plaatst.
- Dan vervolgt B. Die leest zijn eigen beschrijving $\langle B \rangle$ van de tape, en reconstrueert A door $q(\langle B \rangle)$ uit te rekenen met behulp van Q. Dat kan omdat $q(\langle B \rangle) = \langle P_{\langle B \rangle} \rangle = \langle A \rangle$.
- Daarna combineert B het resultaat van deze berekening (ofwel A) met B (die was in de vorm $\langle B \rangle$ als input gegeven) tot $\langle AB \rangle$ (d.w.z. de code van de samenstelling AB van A en B).
- Tenslotte print B deze beschrijving en stopt.

De recursiestelling (Th. 6.3)

Zij T een TM die een functie $t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ berekent. Dan bestaat er een TM R die een functie $r : \Sigma^* \rightarrow \Sigma^*$ berekent zodanig dat voor iedere $w \in \Sigma^*$ geldt:

$$r(w) = t(\langle R \rangle, w).$$

De machine R kan dus over haar eigen code beschikken!

Bewijs

De machine R wordt als volgt verkregen door de volgende 3 delen A, B en T samen te stellen tot ABT:

1. $A = P_{\langle BT \rangle}$. Dus A plaatst $\langle BT \rangle$ op de tape.
2. B leest $\langle BT \rangle$ van de tape en berekent hieruit $\langle A \rangle$ met behulp van Q. Vervolgens combineert B dit resultaat $\langle A \rangle$ met de invoer $\langle BT \rangle$ tot één machine ABT en plaatst de beschrijving $\langle ABT \rangle$ op de tape.
3. Tenslotte vervolgt T.

R is dus de machine ABT, en T heeft op zijn tape zowel de beschrijving $\langle ABT \rangle$ van R staan, als de invoer w.

Voorbeeld 1 toepassing Recursiestelling

Een programma dat zijn eigen code print:

SELF = "Op iedere invoer:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle SELF \rangle$.

2. Print $\langle SELF \rangle$."

Verklaring

De recursiestelling wordt als volgt toegepast. Zij T de TM die de functie:

$$t(u, v) = u$$

berekent. Uit de recursiestelling volgt dat er een TM *SELF* moet bestaan die de functie r berekent waarvoor geldt:

$$r(v) = t(\langle SELF\rangle, v) = \langle SELF\rangle.$$

Met andere woorden: *SELF* berekent de functie $r(v)$ die voor elke v als uitvoer $\langle SELF\rangle$ heeft.

Voorbeeld 2

A_{TM} is onbeslisbaar. (Th. 6.5)

Bewijs: Zij H een beslisser voor A_{TM}. Construeer B:

B = "Op invoer w:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle B\rangle$.
2. Draai H op $\langle B, w\rangle$.
3. *Accepteer*, als H verwerpt, en *verwerp*, als H accepteert."

Dus: $\boxed{B \text{ accepteert } w \quad \Leftrightarrow \quad B \text{ verwerpt } w}$ Tegenspraak.

Minimale TM's

Als M een TM is, dan verstaat men onder de lengte van de beschrijving $\langle M\rangle$ van M het aantal symbolen van het woord $\langle M\rangle$. Men zegt dat M een minimale TM is, als er geen equivalente TM bestaat waarvan de beschrijving korter is. (Def. 6.4)

Definieer:

$$MIN_{\mathsf{TM}} = \{\langle M\rangle \mid M \text{ is een minimale TM}\}.$$

Voorbeeld 3

MIN_{TM} is niet Turing-herkenbaar. (Th. 6.7)

Bewijs: Stel E is een opsommer van MIN_{TM}. Construeer C:

C = "Op invoer w:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle C\rangle$.
2. Draai E totdat er een TM D verschijnt met een langere beschrijving dan C.
3. Simuleer D op invoer w."

Vervolg voorbeeld 3

MIN_{TM} bevat oneindig veel TM's. Dit betekent dat stap 2 een keer eindigt met een D. Omdat D vervolgens gesimuleerd wordt door C, is deze equivalent met C.

Aangezien E een opsommer van MIN_{TM} is, zou D minimaal moeten zijn. Maar de beschrijving van C is korter. Tegenspraak.

Opgave

Laat met behulp van de recursiestelling zien dat $HALT_{\mathsf{TM}}$ niet beslisbaar is.

Oplossing

Uit het ongerijmde: stel H is een beslisser van $HALT_{\mathsf{TM}}$. Definieer:

D = "Op invoer w:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle D \rangle$.

2. Draai H op $\langle D, w \rangle$.

3. *Accepteer* als H verwerpt, anders *loop*."

Nu geldt: $\boxed{D \text{ stopt op } w \quad \Leftrightarrow \quad D \text{ stopt niet op } w}$

Tegenspraak.

Vastepunt-versie recursiestelling (Th. 6.8)

Zij $t : \Sigma^* \to \Sigma^*$ een berekenbare functie. Dan bestaat er een TM F zodanig dat $t(\langle F \rangle)$ een TM beschrijft die equivalent is met F.

Bewijs: Definieer F als volgt:

F = "Op invoer w:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle F \rangle$.

2. Bereken $t(\langle F \rangle)$. Zij $\langle G \rangle$ het resultaat.

3. Simuleer G op invoer w."

Dan geldt dat $\langle F \rangle$ en $t(\langle F \rangle) = \langle G \rangle$ equivalente TM's beschrijven.

Paradox van Berry

Zij k het kleinste natuurlijke getal dat niet in minder dan 1000 woorden in de Nederlandse taal kan worden gedefinieerd.

?

Opgave

Beschouw de volgende functie $b : \mathbb{N} \to \mathbb{N}$ gedefinieerd door: $b(n)$ is het grootste natuurlijke getal dat door een TM M met lengte n kan worden berekend.

Omdat TM's op symboolreeksen werken, representeren we b als een functie $b : \Sigma^* \to \Sigma^*$ voor een geschikt alfabet Σ. Hierbij representeren we natuurlijke getallen als elementen van Σ^*.

Laat zien dat de functie b niet berekenbaar is.

b is een variant van de zogenaamde busy-beaver-functie.

Oplossing (1)

Zij B een TM die b berekent. Definieer nu een TM F:

F = "Op iedere invoer:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle F \rangle$.

2. Zij n de lengte van $\langle F \rangle$. Bereken $\langle m \rangle = b(\langle n \rangle)$ m.b.v. B.

3. Print $\langle m + 1 \rangle$."

Wat concluderen we hieruit?

Oplossing (2)

Zij n de lengte van F. Dan print F het getal $b(n) + 1$. Nu is $b(n)$ volgens de definitie van b het grootste getal dat een programma met lengte n kan afdrukken. Echter F heeft lengte n en print een getal dat groter is dan $b(n)$, namelijk $b(n) + 1$. Tegenspraak.