

Sectie TCS (Theory of Computer Science)

Bezoek studenten tbv Practicum Academische Vaardigheden

Alban Ponse

Sectie *Theory of Computer Science*

Instituut voor Informatica, FNWI

Universiteit van Amsterdam

<https://ivi.fnwi.uva.nl/tcs/>

19 november 2019

Programma voor vandaag

WELKOM en even voorstellen

- 1 Short-circuit-logica (onderzoek & afstudeerprojecten)
(Slides 3–10)
- 2 Wat meer informatie over de sectie TCS (onderzoek en onderwijs)
(Slides 11–15)

Deze slides:

<https://staff.fnwi.uva.nl/a.ponse/PAV-TCS2019.pdf>

Short-Circuit-Logica

Imperative programming. Stel

$$a = (x > 3)$$

$$b = (f(x) == 0)$$

$$c = (g(x) > 3)$$

zijn condities (Boolese expressies) in programma's P_1 and P_2 :

P_1 : ... if (a AND (b OR c)) then (P) else (Q)

P_2 : ... if ((a AND b) OR (a AND c)) then (P) else (Q)

VRAAG: Leveren P_1 en P_2 hetzelfde resultaat (na uitvoering)?

Systematische analyse

Algemener: stel een programma bevat het volgende fragment

if (a AND (b OR c)) then (..) else (..)

VRAAG: Welke logische wetten op condities als hierboven zijn geldig?
Die van de propositielogica?

Bijvoorbeeld: geldt links-distributiviteit van AND over OR, dus

$$x \text{ AND } (y \text{ OR } z) = (x \text{ AND } y) \text{ OR } (x \text{ AND } z) \quad ?$$

Of bijvoorbeeld $x \text{ AND } x = x$?

Voorbeeld: stel integer variable `i` heeft waarde `2`

`(i==2)` instructie die test of integer variable `i` de waarde `2` heeft
`[i:=i+1]` assignment die als test `true` oplevert

Verschillende vormen van sequentiële evaluatie van `AND` en `OR` :

`((i==2) OR [i:=i+1]) AND (i==2)`

evalueert naar

`true` met *short-circuit evaluation* (SCE) (stop zodra het kan)

`false` met *full evaluation* (FE) (alle atomen worden ge-evalueerd)

VRAAG 1. Wat zijn de logische wetten voor SCE?

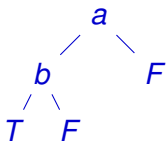
2. Wat zijn de logische wetten voor FE?

Schrijf $\&\&$ (ipv **AND**) om SCE voor te schrijven.

Voorbeeld: alle mogelijke evaluaties van de propositie

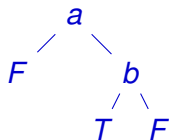
$a \ \&\& \ b$

worden gekarakteriseerd door de volgende **evaluatieboom**:

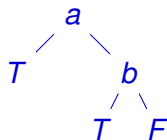


- 1 Takken vanuit een interne knoop naar **links (onder)**: het atoom evalueert naar T ("true");
Takken naar **rechts**: het atoom evalueert naar F ("false")
- 2 Een **evaluatie** van $a \ \&\& \ b$ is een volledig pad;
Het blad waarin een evaluatie eindigt geeft de uiteindelijke waarde aan van de evaluatie van $a \ \&\& \ b$

SCE-evaluatiebomen die negatie \sim en disjunctie $||$ illustreren:



Evaluatieboom van $\sim a \ \&\& \ b$



Evaluatieboom van $a \ || \ b$

Takken naar **links (onder)**: de *literal* (a of $\sim a$) evalueert naar T
naar **rechts (onder)**: ... F

Zeven eenvoudige axioma's (en sommige dualen):

$$\sim T = F$$

$$\sim F = T$$

$$x \ || \ y = \sim(\sim x \ \&\& \ \sim y)$$

$$\sim(\sim x) = x$$

$$T \ \&\& \ x = x$$

$$F \ || \ x = x$$

$$x \ || \ F = x$$

$$x \ \&\& \ T = x$$

$$F \ \&\& \ x = F$$

$$T \ || \ x = T$$

$$(x \ \&\& \ y) \ \&\& \ z = x \ \&\& \ (y \ \&\& \ z)$$

Nog drie axioma's:

$$\sim x \ \&\& \ F = x \ \&\& \ F$$

(levert altijd F op)

$$(x \ \&\& \ F) \ || \ y = (x \ || \ T) \ \&\& \ y$$

(in deze context wordt y altijd ge-evalueerd)

$$(x \ \&\& \ y) \ || \ (z \ \&\& \ F) = (x \ || \ (z \ \&\& \ F)) \ \&\& \ (y \ || \ (z \ \&\& \ F))$$

($||$ rechts-distribueert over $\&\&$
als 't rechterargument F oplevert)

Stelling. (Daan Staudt, 2012) De tien beschreven axioma's vormen een (equationele) axiomatizing van SCE-evaluatiebomen:

voor alle condities P en Q ,

$$E \vdash P = Q \iff \text{eval-boom}(P) = \text{eval-boom}(Q)$$

Bewijs.

\implies (*Soundness*): eenvoudig

\impliedby (*Volledigheid*): ingewikkeld

Ook de connectieven die *full evaluation* (FE) voorschrijven zijn te axiomatiseren, **en** eenvoudig definieerbaar mbv SCE-connectieven:

$$x \& y = (x \mid\mid (y \&\& F)) \&\& y$$

Typisch voorbeeld: $F \&\& y = y \&\& F$.

Meer recent onderzoek:

- Short-Circuit-Logica met ook een waarheidswaarde *undefined*.
Typisch voorbeeld:

$$\sim(b = 0) \&\& (a/b > 18.5)$$

en $F \&\& U = F$, terwijl $U \&\& x = U$.

- Short-Circuit-Logica met extra connectieven zoals bv de XOR en de NAND, en/of met extra identificaties, bijvoorbeeld

$$x \&\& x = x$$

(non-commutative propositional logic, e.g. as in decision trees).

De sectie TCS, Onderzoek en Onderwijs

Website: <https://ivi.fnwi.uva.nl/tcs/>

Leden:

Dr. I. (Inge) Bethke

Dr. B. (Bert) Bredeweg (Lector op de HvA, 1 dag op Uva)

MA. D.S. (Damien) Fleur (PhD bij Bredeweg)

Dr. A. (Alban) Ponse (groepsleider sinds 2012)

Dr. P.H. (Piet) Rodenburg

Guests:

Prof.dr. J.A. (Jan) Bergstra (emeritaat sinds jan. 2017)

Dr.ir. B. (Bob) Diertens (wetenschappelijk systeemontwikkelaar)

Dr. K.M.M. (Karl) de Leeuw (*History of Information Security and Cryptology*)

Dr. S.F.M. van Vlijmen (Managing editor *Applied Soft Computing*)

Relaties: VU-onderzoeksgroep *Theoretical Computer Science*

<http://www.cs.vu.nl/~tcs/>

Dr. G. (Gerard) Alberts *KdV Instituut*

Main Research themes

1 Instruction sequences

Central question: What is a program?

2 Meadows

Modelling of the rational, real and complex numbers with the property that the multiplicative inverse of 0 exists

<https://meadowsite.wordpress.com/>

3 Proposition algebra and short-circuit logics

Central question: What are the operators and logics underlying conditionals and short-circuit evaluation?

FOKKE & SUKKE
KNOW WHAT SCIENCE IS ABOUT

A MOST IMPRESSIVE
DEMONSTRATION, COLLEAGUE...

BUT WILL IT WORK
IN THEORY?



Theorie-ontwikkeling && verspreiding via *journals* en *proceedings*, en (natuurlijk) ook via het onderwijs

- ▶ Automaten en Formele Talen (Bethke; Ba Inf)
- ▶ Discrete Wiskunde en Logica (Bethke; Ba Inf)
- ▶ Theoretische Aspecten van Programmatuur (Ponse; Ba Inf)
- ▶ Technology-enhanced Learning (Bredeweg; Ma IS en AI)
- ▶ Concurrency Theory (Ponse; Ma CS, PDCS, Logic)
- ▶ Protocol Validation (Ponse; Ma CS, PDCS, Logic)
- ▶ Recursion Theory (Rodenburg; Ma Logic, CS, PDCS)
- ▶ Lambda Calculus (Rodenburg; Ma Logic, CS, PDCS)
- ▶ Science in Perspective (Rodenburg; Ma Physics and Astronomy, etc)
- ▶ Research projects (6 EC) (Ma CS, PDCS Logic)
- ▶ Afstudeerprojecten (18 EC: Ba Inf en KI;
36 EC: Ma AI; 30 EC: Ma CS en Logic)

TCS & onderwijs-gerelateerde activiteiten

- ▶ Bethke:
 - opleidingsdirecteur Ba Informatica
- ▶ Ponse:
 - lid Examensubcommissie SE/SNE
 - UvA-contactpersoon Ma Computer Science (joint-degree)
 - adviserend lid OC Computer Science + PDCS @ VU
- ▶ Rodenburg:
 - vice-voorzitter facultaire ondernemingsraad
 - lid van de Centrale ondernemingsraad
- ▶ Webklas Informatica **Wat is een programma?**
(sinds 2004, VO scholieren ++)