

TCS-groep (Theory of Computer Science)

Bezoek studenten tbv Practicum Academische Vaardigheden

sectie *Theory of Computer Science*
Informatica Instituut, FNWI
Universiteit van Amsterdam

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

5 november 2014

Welkom & even voorstellen

Programma voor vandaag

- 1 Bitcoin: JAB (Slides 3–4)
- 2 Rekenen zonder getallen, demo (colleges TAP & CT): IB
(<http://tcs.science.uva.nl/WebToolset/parser>)
- 3 A small Factory, demo PSF-Toolkit + ACP (colleges TAP & CT): AP
(<https://staff.fnwi.uva.nl/b.diertens/psf/specifications/factory.html>)
- 4 Short-circuit logica (college TAP): AP (Slides 5–13)
- 5 Wat meer informatie over de sectie TCS (onderzoek en onderwijs)
(Slides 14–19)

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

Bitcoin

Jan Bergstra

sectie Theory of Computer Science
Informatica Instituut, FNWI
Universiteit van Amsterdam

j.a.bergstra@uva.nl

Informational money

- Theorie over Bitcoin: is het geld, wat voor geld.
- Bitcoin: a money-like informational commodity.
- Bitcoin: not a currency-like informational commodity.
- Bitcoin & Islamic Finance.
- Exclusively Informational Money (EXIM).
- Abstractie van de Nakamoto architecture (blockchain technology).
- Ethische aspecten van Bitcoin en andere “money-like informational commodities”

Short-Circuit Logica

Alban Ponse

sectie Theory of Computer Science
Instituut voor Informatica, FNWI
Universiteit van Amsterdam

<https://staff.fnwi.uva.nl/a.ponse/>

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

Stel: $(i==k)$ is de instructie die test of programma variable i de waarde k (een integer) heeft; assignment $[i=i+1]$ levert als test `true` op als i (initiële) waarde 2 heeft

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

$$((i==2) \text{ OR } [i=i+1]) \text{ AND } (i==2)$$

evalueert naar

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

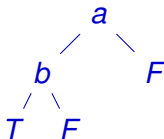
`false` met *strict evaluation* (StE) (alle atomen in een te evalueren expressie worden ge-evalueerd)

- VRAAG 1:** Welke logica karakterizeert SCE?
2: Welke logica karakterizeert StE?

SCE-voorbeeld: alle mogelijke evaluaties van de propositie

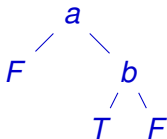
a AND b

worden gekarakteriseerd door de volgende **evaluatieboom**:

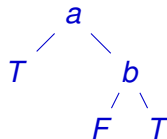


- 1 Takken vanuit een interne knoop naar **links (onder)**: het betreffende atoom evalueert naar T ("true");
takken naar **rechts**: atoom evalueert naar F ("false")
- 2 Een **evaluatie** van a AND b is een volledig pad;
Het blad waarin een evaluatie eindigt representeert de uiteindelijke waarde van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

Nog twee voorbeelden van SCE-evaluatiebomen die negatie NOT en OR illustreren:



Evaluatieboom van (NOT a) AND b



Evaluatieboom van a OR (NOT b)

(Takken naar links (onder) geven aan dat het betreffende atoom naar T evalueert, en takken naar rechts naar F)

Zeven eenvoudige axioma's (en sommige dualen):

$$\text{NOT } T = F$$

$$\text{NOT } F = T$$

$$x \text{ OR } y = \text{NOT } ((\text{NOT } x) \text{ AND } (\text{NOT } y))$$

$$\text{NOT } (\text{NOT } x) = x$$

$$T \text{ AND } x = x$$

$$F \text{ OR } x = x$$

$$x \text{ AND } T = x$$

$$x \text{ OR } F = x$$

$$F \text{ AND } x = F$$

$$T \text{ OR } x = T$$

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

Nog drie axioma's:

$$x \text{ AND } F = (\text{NOT } x) \text{ AND } F$$

(levert altijd F op)

$$(x \text{ AND } F) \text{ OR } y = (x \text{ OR } T) \text{ AND } y$$

(in deze context wordt y altijd ge-evalueerd)

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

(OR rechts-distribueert over AND
als 't rechterargument F oplevert)

Stelling 1. (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

TCS-groep (Theory of Computer Science)

Wat meer over onderzoek (*in English*) en onderwijs

Alban Ponse

sectie Theory of Computer Science
Informatica Instituut, FNWI
Universiteit van Amsterdam

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

Sectie

sectie TCS:	Theory of Computer Science	(sinds 2009)
groepsleider:	Prof.dr. J.A. (Jan) Bergstra Dr. A. (Alban) Ponse	(tot 12-2011) (sinds 2012)
verdere leden:	Dr. I. (Inge) Bethke Dr. B. (Bert) Bredeweg Dr.ir. B. (Bob) Diertens Dr. P.H. (Piet) Rodenburg	(wetensch. progr.) (tot 07-2011)
<i>associated people:</i>	Dr. G. (Gerard) Alberts Prof.dr. C.A. (Kees) Middelburg Dr. P.H. (Piet) Rodenburg Dr. S.F.M. (Bas) van Vlijmen	

Research themes

1 Instruction sequences

- many journal articles
- 2003: Bethke & Ponse; textbook (In Dutch)
- 2012: Bergstra & Middelburg; 2nd text book

2 Meadows

- a modelling of the rational, real and complex numbers with the property that the multiplicative inverse of 0 exists

3 Proposition algebra and short-circuit logics

- central question: What are the operators and logics underlying conditionals and short-circuit evaluation?
- [ACM Transactions on Computational Logic, 12(3), Article 21 (36 pp), 2011]

FOKKE & SUKKE
KNOW WHAT SCIENCE IS ABOUT

A MOST IMPRESSIVE
DEMONSTRATION, COLLEAGUE...

BUT WILL IT WORK
IN THEORY?



Research themes Qualitative Reasoning Group

Bert Bredeweg

- 4 Representation and reasoning
 - automated problem solving methods for QR (qualitative reasoning)
- 5 Interface and interaction
 - development of workbenches for building, simulating, and inspecting qualitative models
- 6 Knowledge capture
 - empowers domain experts in better organizing their knowledge

Output in 2009-2011 of these related themes (63 items in total):

11	journal articles
16	conference proceedings
1	Book chapter
1	Ed. collection
12	Technical reports
22	Invited talks

Notable results 2009-2011

- Various toolkits, workbenches, portals for *Edu + Research*, including

PSF Toolkit (Process Specification Formalism), latest update in 2012

PGA Toolset (ProGram Algebra), latest update in 2007; web interface in 2010

QRM portal (Qualitative Reasoning and Modelling), latest update in 2012

DynaLearn (version 1.0.2), latest update in 2012

- Journals: special issues (TCS '11, Ecological Inf. '09) and editorships:

Journal of Logic and Algebraic Programming

Journal of Applied Logic

Scientific Annals of Computer Science

Advances in Software Engineering

Informatics and Logic

Scholarly Research Exchange

Science of Computer Programming

Computer Science Reviews

Journal of Universal Computer Science

The Open Software Engineering Journal

Journal of Algorithms in Cognition

Ecological Informatics

- Organization of TCSA 2009, 2010, 2011, 2013 (now biannual)
(Theoretical Computer Science day Amsterdam)

De TCS-groep en het onderwijs

- Automaten en Formele Talen (Ba Inf, Ba Wis + Inf)
- Discrete Wiskunde en Logica (Ba Inf)
- (TAP) Theoretische Aspecten van Programmatuur (Ba Inf)
- Afstudeerprojecten (Ba Informatica, Ma Logic)
- Examencommissie Ba Inf + Ma Computational Science / Examencommissie Ma SNE

- (CT) Concurrency Theory (Ma Computer Science, Ma Logic)
- Concurrent System Design by Abstraction (Ma Computer Science)
- Afstudeerprojecten (Ba Inf, Ma Logic, Ma Computer Science)
- Webklas Informatica (sinds 2004, VO scholieren ++)