



UvA-DARE (Digital Academic Repository)

Rekenen-Informatica

Bergstra, J.A.; Bethke, I.; Ponse, A.

Publication date

2014

Document Version

Final published version

[Link to publication](#)

Citation for published version (APA):

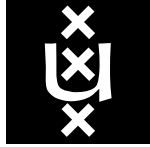
Bergstra, J. A., Bethke, I., & Ponse, A. (2014). *Rekenen-Informatica*. (TCS Electronic Report series; No. 1412). University of Amsterdam, Theory of Computer Science. <https://ivi.fnwi.uva.nl/tcs/pub/tcsreports/TCS1412.pdf>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.



University of Amsterdam
Theory of Computer Science

Rekenen-Informatica

J.A. Bergstra
I. Bethke
A. Ponse

J.A. Bergstra

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XH Amsterdam
the Netherlands

tel. +31 20 525.7591
e-mail: J.A.Bergstra@uva.nl

I. Bethke

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XH Amsterdam
the Netherlands

tel. +31 20 525.7583
e-mail: I.Bethke@uva.nl

A. Ponse

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XH Amsterdam
the Netherlands

tel. +31 20 525.7592
e-mail: A.Ponse@uva.nl

Rekenen-Informatica

Jan A. Bergstra, Inge Bethke & Alban Ponse

Instituut voor Informatica, Universiteit van Amsterdam*

18 oktober 2014

Samenvatting

Plaatsbepaling van Rekenen-Informatica als perspectief op elementair rekenen vanuit de informatica.

1 Inleiding

Rekenen-Informatica (hieronder R-I) is een term die wij willen gaan gebruiken voor een manier om naar elementair rekenen te kijken vanuit de informatica, en bij onze invulling daarvan, vanuit een specifiek deel van de theoretische informatica. R-I kan op allerlei andere wijzen worden vormgegeven, zodat wat we hier beschrijven ook een onderdeel van R-I zou kunnen zijn.

2 Rekenen-Informatica (R-I)

R-I vergt expliciete uitgangspunten. Hieronder een voorlopige formulering van deze uitgangspunten. Zoals opgemerkt, het kan ook anders, dan zou dit een hoofdstuk uit R-I kunnen zijn, niet het hele verhaal.

1. Om te beginnen een disclaimer. R-I betreft niet: nieuwe grondslagen van de wiskunde, ook in R-I kan men een realist zijn, of een nominalist, of een logicist, of een intuïtionist en ook met alle vertakkingen die er per categorie nog zijn.
2. Wel is er bij R-I sprake van een accentverschuiving ten opzichte van rekenen-wiskunde (wiskundige kijk op het rekenen, ook wel rekenen inclusief wiskunde); die verschuiving heeft de volgende aspecten:

*Science Park 904, 1098 XH Amsterdam

- (a) syntax bestaat en is zelfs alomtegenwoordig, vragen over de syntax die optreden bij onderwerpen in het rekenen doen steeds terzake,¹
 - (b) syntax vraagt om semantiek net zo zeer als omgekeerd semantiek om syntax vraagt,
 - (c) oneindige structuren zijn zinvol vanwege:
 - de mogelijkheid tot het systematiseren van (kennis over) een groot aantal eindige structuren,
 - elegantie, en de uitdrukingskracht die dat oplevert,
 - de intuïtieve kracht van asymptotiek.
3. Rekenen *partout* bestaat niet, het gebeurt altijd in een bepaalde structuur. Er bestaat geen “meest voor de hand liggende” structuur voor het rekenen.
 4. Het is vaak zinvol om bij structuren met een additioneel element a te werken (voor alle operaties) dat kan dienen als fout-waarde (*sink*).
 5. Een uniek domein van “de natuurlijke getallen” bestaat niet; al dan niet eindige structuren met beginstukken van een oneindige (zonder herhaling) opsombare collectie ervan bestaan wel.
 6. Getalsystemen zoals \mathbb{N} , \mathbb{Z} , \mathbb{Q} zijn abstracte datatypen. Bij elk van deze abstracte datatypen is een aantal voor de hand liggende concrete datatypen te vinden dat als realisatie van het abstracte datatype kan worden gezien. Van belang zijn hier met name de kanonieke termalgebra’s.
 7. Termherschrijfsystemen samen met bijbehorend herschrijfstrategieën (bijvoorbeeld *leftmost innermost*) leveren een bruikbare taal om algoritmen te noteren: “Functioneel programmeren in een gemakkelijk leesbare notatie”.

Met algebraïsche specificaties die een volledig termherschrijfsysteem vormen komt men aan concrete datatypen voor deze getalsystemen.

3 Theoretische grondslagen van R-I

De meest voor de hand liggende filosofie van de wiskunde die aan R-I ten grondslag ligt is *structuralisme*. Dan ziet men het natuurlijke getal 7 niet als een uniek bepaalde entiteit die op zichzelf kan staan maar als een functor (functie op een klasse) die aan structuren die aan de axioma’s voor de natuurlijke getallen voldoen de daarin optredende interpretatie van de 7e opvolger van 0 toekent. Zoals [12] uiteenzet roept structuralisme ook vragen op die geen eenvoudig antwoord hebben.

¹Als men syntax geheel mathematiseert, dan wordt het een onderdeel van de wiskunde dat zelf weer om syntax vraagt. De zo dreigende *infinite regress* wordt vermeden door vragen over metasyntax, de syntax van de theorie over de syntax, minder ernstig op te nemen dan die over syntax.

Uit de structuralistische visie ontlenen we de volgende gezichtspunten, die we als naïef structuralisme willen opvoeren:

1. Atomaire (of ook primitieve) entiteiten die “logisch” via een rol in een groter geheel gekarakteriseerd worden bestaan alleen in een grotere structuur waarin ook soortgelijke entiteiten voorkomen.
2. De grotere structuren (elk model van een per structuur vooraf vast te leggen verzameling beweringen) waarin de atomaire entiteiten “wonen” zijn niet uniek.
3. De daarin voorkomende entiteiten, zelfs die met een rolbeschrijving die een uniek element van de structuur aanwijzen zijn toch zelf niet uniek, want afhankelijk van de keuze van de grotere structuur.²
4. Gestructureerde objecten, waaronder de structuren die atomaire objecten huisvesten, kunnen wel een zekere uniciteit hebben, althans modulo isomorfie (een notie die voor atomaire objecten irrelevant is).

4 Uitwerking van R-I

Uitwerking van R-I kan op vele manieren. De auteurs hebben een route gekozen waarin de elementaire getsystemen \mathbb{N} , \mathbb{Z} , \mathbb{Q} als abstracte datatypen worden gezien. Dit leidt er onder meer toe dat deling een functiesymbool toegewezen krijgt (éénplaatsig voor een inverse, of tweeplaatsig met een deelstreep), waarmee uitdrukkingen als

$$0^{-1} \quad \text{en} \quad \frac{1}{0}$$

ontstaan en “deze syntax om semantiek vraagt” (deze verschillende functiesymbolen worden nader bestudeerd in [3]). De introductie van een functiesymbool voor deling (onvermijdelijk leidend tot de vraag naar de betekenis van delen door nul) markeren we in navolging van [9] door te spreken over een weide (*meadow*) in plaats van een lichaam (*field*).

Er zijn verschillende oplossingen voor de vraag hoe om te gaan met 0^{-1} :

1. De keuze $0^{-1} = 0$ leidt tot de zogenaamde *involutive meadows* die voldoen aan $(x^{-1})^{-1} = x$ (zie [9] en bijvoorbeeld ook [1]).
2. De keuze $0^{-1} = 1$ levert een *one-based non-involutive meadow* (zie [4]).
3. De keuze $0^{-1} = a$ met a een fout-waarde levert de zogenaamde *common meadows* (zie [6]).

²Dit geldt a fortiori wanneer de structuren komen uit een klasse van modellen van een theorie die zowel eindige als oneindige leden heeft.

De voor- en nadelen van deze keuzen zijn niet eenduidig af te wegen. Zo blijkt dat de involutive meadows heel hanteerbaar zijn in de context van kansrekening (zie [5]), terwijl de common meadows een goed uitgangspunt lijken te bieden voor de formalisering van rekenen-(school)wiskunde. Bij elk van deze wijzen voor het omgaan met deling door nul ontstaat een variant van de theorie van de lichamen die zich goed met vergelijkingen laat beschrijven. Die formalisering leidt weer tot allerlei vragen voor verder onderzoek. Typische onderzoeksvragen zijn bijvoorbeeld:

- Het bestaan van een automatische procedure die voor een willekeurige gesloten vergelijking bepaalt of die correct is of niet. Dit zogenaamde ‘woordprobleem’ wordt in [10] bewezen voor involutive meadows.
- Het bestaan van volledige axiomatizeringen (zie bijvoorbeeld [2]).
- Het bestaan van eindige structuren voor de verschillende varianten (zie bijvoorbeeld [11]).

Naast het ontwikkelen van theorievorming over het delen door nul werken we aan het ontwerpen van *ground complete* termherschrijfsystemen voor relevante datatypen zoals \mathbb{N} , \mathbb{Z} , \mathbb{Q} (zie bijvoorbeeld [7]). Binnen dit thema zijn er nog ruimschoots open vragen aanwezig die voortgaand onderzoek motiveren. Een algemene stelling uit [8] die de mogelijkheid van het vinden van zulke termherschrijfsystemen aangeeft is een uitgangspunt bij dit werk.

Bij eindige structuren is het altijd mogelijk een *ground complete* termherschrijfsysteem te vinden ter beschrijving. Interessant wordt het pas wanneer men de omvang van het stelsel van vergelijkingen (te lezen als herschrijfgeregels) wil beperken. Ook op het gebied van de beschrijving van eindige structuren met termherschrijfsystemen valt nog veel onderzoek te doen.

5 Besluit

Wij hebben een omschrijving van Rekenen-Informatica gegeven, alsmede een omschrijving van de wijze waarop de auteurs, werkend in verschillende combinaties met ook andere co-auteurs zoals John Tucker, Kees Middelburg en Piet Rodenburg, Rekenen-Informatica proberen te ontwikkelen. Deze omschrijving zou men een manifest kunnen noemen. Mocht een Engelstalig equivalent voor Rekenen-Informatica gewenst zijn, dan komen wij op *Arithmetic-Informatics*.

Qua toepassing denken we met name aan het rekenonderwijs. Allereerst kan langs deze lijn via het rekenonderwijs een kennismaking met een deel van de theoretische informatica plaatsvinden, en in tweede instantie kunnen ook voor het rekenonderwijs zelve mogelijk langs de lijnen van Rekenen-Informatica nog nieuwe perspectieven ontstaan.

Referenties

- [1] Bergstra, J.A., Bethke, I., and Ponse, A. (2013). Cancellation meadows: a generic basis theorem and some applications. *The Computer Journal*, 56(1):3–14, doi:10.1093/comjnl/bsx147.
- [2] Bergstra, J.A., Bethke, I., and Ponse, A. (2014). Equations for formally real meadows. arXiv:1310.5011v3 [math.RA, cs.LO], this version (v3): 11 February 2014. (First version (v1): 18 October 2013.)
- [3] Bergstra, J.A. and Middelburg, C.A. (2011). Inversive meadows and divisive meadows. *Journal of Applied Logic*, 9(3):203–220.
- [4] Bergstra, J.A. and Middelburg, C.A. (2014). Division by zero in non-involutive meadows. arXiv:1406.2092v1 [math.RA], 9 June 2014.
- [5] Bergstra, J.A. and Middelburg, C.A. (2014). A thread algebra with probabilistic features. arXiv:1409.6873v2 [cs.LO], 26 September 2014.
- [6] Bergstra, J.A. and Ponse, A. (2014). Division by zero in common meadows. arXiv:1406.6878v1 [math.RA], 26 June 2014.
- [7] Bergstra, J.A. and Ponse, A. (2014). Three datatype defining rewrite systems for datatypes of Integers each extending a datatype of Naturals. arXiv:1406.3280v2 [math.RA], 21 August 2014.
- [8] Bergstra, J.A. and Tucker, J.V. (1995). Equational specifications, complete term rewriting systems, and computable and semicomputable algebras. *Journal of the ACM*, 42(6):1194–1230.
- [9] Bergstra, J.A. and Tucker, J.V. (2007). The rational numbers as an abstract data type. *Journal of the ACM*, 54(2), Article 7 (25 pages).
- [10] Bethke, I. and Rodenburg, P.H. (2010). The initial meadows. *Journal of Symbolic Logic*, 75(3):888–895.
- [11] Bethke, I., Rodenburg, P.H., and Sevenster, A. (2009). The structure of finite meadows. arXiv:0903.1196v1 [cs.LO], 6 March 2009. To appear in *Journal of Logical and Algebraic Methods in Programming*, DOI: 10.1016/j.jlamp.2014.08.004
- [12] Parsons, C. (1990). The structuralist view of mathematical objects. *Synthese*, 84(3):303–346.

Electronic Reports Series of section Theory of Computer Science

Within this series the following reports appeared.

- [TCS1411] J.A. Bergstra, *Bitcoin: not a Currency-like Informational Commodity*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1409v2] J.A. Bergstra and A. Ponse, *Three Datatype Defining Rewrite Systems for Datatypes of Integers each extending a Datatype of Naturals (version 2)*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1410] J.A. Bergstra and A. Ponse, *Division by Zero in Common Meadows*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1407v3] J.A. Bergstra, *Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers (version 3)*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1409] J.A. Bergstra and A. Ponse, *Three Datatype Defining Rewrite Systems for Datatypes of Integers each extending a Datatype of Naturals*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1406v3] J.A. Bergstra, *Bitcoin and Islamic Finance (version 3)*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1407v2] J.A. Bergstra, *Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers (version 2)*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1408] J.A. Bergstra, *Bitcoin: Informational Money en het Einde van Gewoon Geld*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1407] J.A. Bergstra, *Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1406v2] J.A. Bergstra, *Bitcoin and Islamic Finance (version 2)*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1406] J.A. Bergstra, *Bitcoin and Islamic Finance*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1405] J.A. Bergstra, *Rekenen in een Conservatieve Schrapwet Weide*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1404] J.A. Bergstra, *Division by Zero and Abstract Data Types*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1403] J.A. Bergstra, I. Bethke, and A. Ponse, *Equations for Formally Real Meadows*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1402] J.A. Bergstra and W.P. Weijland, *Bitcoin, a Money-like Informational Commodity*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1401] J.A. Bergstra, *Bitcoin, een "money-like informational commodity"*, section Theory of Computer Science - University of Amsterdam, 2014.
- [TCS1301] B. Dierkens, *The Refined Function-Behaviour-Structure Framework*, section Theory of Computer Science - University of Amsterdam, 2013.
- [TCS1202] B. Dierkens, *From Functions to Object-Orientation by Abstraction*, section Theory of Computer Science - University of Amsterdam, 2012.

- [TCS1201] B. Diertens, *Concurrent Models for Object Execution*, section Theory of Computer Science - University of Amsterdam, 2012.
- [TCS1102] B. Diertens, *Communicating Concurrent Functions*, section Theory of Computer Science - University of Amsterdam, 2011.
- [TCS1101] B. Diertens, *Concurrent Models for Function Execution*, section Theory of Computer Science - University of Amsterdam, 2011.
- [TCS1001] B. Diertens, *On Object-Oriented*, section Theory of Computer Science - University of Amsterdam, 2010.

Within former series (PRG) the following reports appeared.

- [PRG0914] J.A. Bergstra and C.A. Middelburg, *Autosolvability of Halting Problem Instances for Instruction Sequences*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0913] J.A. Bergstra and C.A. Middelburg, *Functional Units for Natural Numbers*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0912] J.A. Bergstra and C.A. Middelburg, *Instruction Sequence Processing Operators*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0911] J.A. Bergstra and C.A. Middelburg, *Partial Komori Fields and Imperative Komori Fields*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0910] J.A. Bergstra and C.A. Middelburg, *Indirect Jumps Improve Instruction Sequence Performance*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0909] J.A. Bergstra and C.A. Middelburg, *Arithmetical Meadows*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0908] B. Diertens, *Software Engineering with Process Algebra: Modelling Client / Server Architectures*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0907] J.A. Bergstra and C.A. Middelburg, *Inversive Meadows and Divisive Meadows*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0906] J.A. Bergstra and C.A. Middelburg, *Instruction Sequence Notations with Probabilistic Instructions*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0905] J.A. Bergstra and C.A. Middelburg, *A Protocol for Instruction Stream Processing*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0904] J.A. Bergstra and C.A. Middelburg, *A Process Calculus with Finitary Comprehended Terms*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0903] J.A. Bergstra and C.A. Middelburg, *Transmission Protocols for Instruction Streams*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0902] J.A. Bergstra and C.A. Middelburg, *Meadow Enriched ACP Process Algebras*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0901] J.A. Bergstra and C.A. Middelburg, *Timed Tuplix Calculus and the Wesseling and van den Berg Equation*, Programming Research Group - University of Amsterdam, 2009.
- [PRG0814] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences for the Production of Processes*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0813] J.A. Bergstra and C.A. Middelburg, *On the Expressiveness of Single-Pass Instruction Sequences*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0812] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences and Non-uniform Complexity Theory*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0811] D. Staudt, *A Case Study in Software Engineering with PSF: A Domotics Application*, Programming Research Group - University of Amsterdam, 2008.

- [PRG0810] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Poly-Threading*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0809] J.A. Bergstra and C.A. Middelburg, *Data Linkage Dynamics with Shedding*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0808] B. Diertens, *A Process Algebra Software Engineering Environment*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0807] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *Tuplix Calculus Specifications of Financial Transfer Networks*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0806] J.A. Bergstra and C.A. Middelburg, *Data Linkage Algebra, Data Linkage Dynamics, and Priority Rewriting*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0805] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *UvA Budget Allocatie Model*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0804] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Sequential Poly-Threading*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0803] J.A. Bergstra and C.A. Middelburg, *Thread Extraction for Polyadic Instruction Sequences*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0802] A. Barros and T. Hou, *A Constructive Version of AIP Revisited*, Programming Research Group - University of Amsterdam, 2008.
- [PRG0801] J.A. Bergstra and C.A. Middelburg, *Programming an Interpreter Using Molecular Dynamics*, Programming Research Group - University of Amsterdam, 2008.

The above reports and more are available through the website: ivi.fnwi.uva.nl/tcs/

Electronic Report Series

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XG Amsterdam
the Netherlands

ivi.fnwi.uva.nl/tcs/