University of Amsterdam

Theory of Computer Science

Bitcoin, een "money-like informational commodity"

J.A. Bergstra

Report TCS1401 February 2014

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

Theory of Computer Science Electronic Report Series

Bitcoin, een "money-like informational commodity"

Jan Aldert Bergstra
Instituut voor Informatica, Universiteit van Amsterdam
Email: j.a.bergstra@uva.nl

February 6, 2014

Abstract

Een poging tot beschrijven en duiden van Bitcoin, een "money-like informational commodity". Voordacht College Tour, Spui 25, 5 februari 2014, 20.00-20.45.

Keywords and phrases: informational money, informational commodity, near-money, geld, double-spending attack, Bitcoin.

1 Inleiding

Satoshi Nakamoto is een naam die in 2013 bij het grote publiek is doorgebroken via zijn (haar, hun) geesteskind Bitcoin.[1]

Eind 2008 kwam de open source implementatie van de eerste Bitcoin client beschikbaar en begin 2009 werd het systeem operationeel met de creatie van het zogenaamde Genesis blok (block).

Hierover wil ik het nu hebben, zonder lichtbeelden, ik ben zelf helemaal niet visueel ingesteld, maar mijn tekst komt digitaal beschikbaar. Het bezwaar hiervan is dat u eventueel naar mij moet kijken, in plaats van naar de output van de beamer, ik kan mij daar slechts voor verontschuldigen.

Stel u voor dat men nieuw geld wil ontwerpen, en eigenlijk de afschaffing door president Nixon van de gouden standaard ongedaan wil maken. We willen "iGoud" ontwerpen, waarbij de "i" staat voor moderniseren en verbeteren.

[1] Wie of wat achter deze naam schuil is niet (algemeen) bekend. Ongevalideerde vermoedens zijn er te over.

iGoud willen we liefst per atoom, of althans op nano-schaal, kunnen verhandelen. Hoe veel tijd en geld we ook in mijnbouw steken iGoud moet schaars blijven. iGoud moet je goed en goedkoop kunnen bewaren, het moet niet kunnen roesten, verdampen, of verbranden. Een transactie van iGoud functioneert bij voorkeur zonder gepantserde auto's, en zonder gespecialiseerd personeel.

Dit is bij elkaar zo mooi dat we dan voor lief nemen dat we met iGoud geen kiezen kunnen vullen, geen trouwring kunnen maken, en geen electronica kunnen verbeteren. Van belang is alleen dat mensen graag iGoud hebben. Waarom ze het graag willen hebben maakt niet uit, maar een beroep op hebzucht, een volgens het kapitalisme verdienstelijke menselijke zwakte moet het wel doen. Dat is primair een zaak van marketing en PR. Wat iedereen graag wil hebben dat wil je zelf toch ook graag hebben.

1.1 Geld: wat is dat?

Ik heb jarenlang geprobeerd om te lezen wat ik kon vinden over de natuur van geld. Mijn idee is dat geld een onderdeel van een communicatieprotocol vormt en eigenlijk geheel een zaak van informatica is. Maar dat wil ik u niet aanpraten.

Ik kwam tamelijk toevallig op Bitcoin uit na een uitgebreide studie van Islamitisch geld; de stap van theoretische informatica naar Islamitisch geld is opmerkelijk klein. Islamitisch geld is nu zo'n 300 maal groter qua circulatie dan Bitcoin, ook iets om eens naar te kijken dus. Een connectie met Bitcoin is er ook, ik meen dat Bitcoin thans niet halal is (ook niet haram overigens), maar dat het uitermate bruikbaar is als implementatie van Islamitisch geld, omdat die problemen na enige aanpassingen van het Bitcoin ontwerp oplosbaar zijn.

1.2 Wat is geld: zorgen van een onpraktisch mens

Dat ik pieker over de vraag wat geld is moet men niet verkeerd begrijpen, dat ligt ook aan mijzelf, ik lig wakker over veel simpeler zaken, die de voortgang van het mensdom werkelijk niet betreffen. Veel tijd steek in de vraag wat de uitdrukking 1 gedeeld door 0 wel kan betekenen. Die vraag kun je in de gewone wiskunde eigenlijk niet stellen (dat zie ik als een "ontdekking" van mijzelf), dan kun je rekenen met 1 gedeeld door 0 is 0 (een uitvinding die door velen is gedaan), en de wiskunde wordt dan best mooi, weer een "ontdekking" (en die komt uit Japan in de jaren 70). Daaraan werk ik met mijn collega's weer verder sinds 2007 (zie [11, 8] en [5]).

1.3 Wat is geld: ontdekking versus uitvinding

Dit is een kip of ei kwestie van enige allure. Een antropoloog komt bij een nieuwe cultuur en ziet dat daar goederen worden geruild, maar sommige goederen alleen om te ruilen. Verder ziet hij dat in die cultuur voor zulke goederen geen speciale naam bestaat: hij zegt, die voer ik in, "geld" en meldt het thuisfront dat hij "geld" heeft ontdekt en dat men dat ook maar eens moet proberen.

Als uitvinding werkt het anders: moe geworden van de complexe handelsconstructies zoekt men (een overheidscommissie) een ruilmiddel zonder verder belang, daar zoekt men ook een naam voor: geld. Nu is het geld uitgevonden. En dat het geld ergens ooit werd uitgevonden is iets dat men kan ontdekken, of proberen te ontdekken. En de theorie dat geld ergens ontdekt werd kan men uitvinden (en dan onderzoeken of de hypothese klopte). Deze overwegingen zijn oud: Karl Menger uit Wenen (de vader van de vader van de verzamelingenleer), het begin van de Austrian economics. Diens opvolger Hayek, ruim een eeuw later, wordt in bijna elke conceptuele paper over Bitcoin aangehaald.

1.4 Bitcoin: schaarste is ingebouwd

Er ontstaan maximaal 21 miljoen Bitcoins (BTCs) en dat duurt nog zo'n 30 jaar, we zijn nu op de helft van de creatie van de Bitcoins. Elke Bitcoin (BTC) kan men in 100 miljoen Satoshi's verdelen, samen dus uiteindelijk 2,1 maal 10 tot de macht 15 Satushi's (ofwel zo'n 200.000 per wereldburger) waarvan een onbekend en langzaam toenemend aantal in de loop der jaren zoek raakt en nooit meer gebruikt zal kunnen worden.[2] Transacties kun je op je laptop uitvoeren, en eigenlijk hoef je niet te weten wat een firewall is, je hoeft helemaal niet te weten hoe men een computer beveiligt. Maar je moet liever wel een brandkast hebben als je met Bitcoin aan de gang gaat, en als je bezit (in Bitcoin) bij overlijden naar je partner of kinderen wilt doen vererven: let op, let op, want dat gaat helemaal niet vanzelf.

Bitcoin als iGoud, wat moeten we daarvan denken. Dat moet iedereen zelf beoordelen, dat is zeker geen duidelijke zaak.

[2]Het BTC volume neemt dus eerst toe tot een maximum waar creatie en verlies in evenwicht zijn en neemt daarna geleidelijk af, aannemende dat verlies steeds weer blijft optreden. Op de lange duur (duizenden jaren) is er niets meer over, dit ander de aanname dat elke sleutel een positieve en op den duur stabiele kans heeft om per tijdseenheid verloren te raken.

2 Informationeel iGoud, wiskundig iGoud

Hoe maken we iGoud zonder kansloze alchemie. Aan willekeurige keuzen valt niet te ontkomen. Dragers van waarde worden bitreeksen van 128 bits. Die kan iedereen zelf aanmaken, dat vergt dan wel hulpmiddel uit de zogenaamde elliptic curve cryptography (derdegraads krommen). Prachtige klassieke wiskunde. Het systeem kent dan aan zo'n reeks een waarde toe, en transacties kunnen die waarde geheel of gedeeltelijk verplaatsen naar andere bitreeksen. Zo'n bitreeks is net als een bankrekening. Alleen je kunt zo'n bitreeks zelf aanmaken, dat zelfaanmaken zou een handelsbank overigens ook gewoon kunnen "leveren".

Hoe bewijs je nu je bezit van zo'n bitreeks, en van de waarde die het systeem daaraan toekent. We noemen de bitreeks een account. Tegelijk met dat account maak je een andere bitreeks, ook met de derde graads-krommen uit de algebraïsche meetkunde, dat is de sleutel. Met die sleutel kun je een handtekening plaatsen onder elk bericht, en dus ook onder een overschrijving vanuit het account waarbij de sleutel hoort. Die sleutel moet je dan veilig en geheim bewaren (bijvoorbeeld op papier in een kluis). Wie de sleutel van een account kent, beschikt over de waarde die hoort bij het account. Ziehier het nut van de kluis als plek voor deze sleutels bij overmacht, geweld of bij overlijden. Zoiets werkt dan meestal goed.

2.1 Nakamoto wilde meer

Een bank gebruikt handtekeningen om te bepalen wie een account mag gebruiken. Dat is verouderd. Tegenwoordig werkt dit via vrij eenvoudige interfaces op een computer, en slechts weinigen begrijpen wat er echt aan de hand is wanneer je inlogt op de site van je bank. Waarom kijkt de NSA niet mee, of de maffia? En waarom vertrouwen we de bank? Die vraag mag je in Nederland niet stellen, het antwoord is bekend: wij vertrouwen de Nederlandse Bank. Nakamoto was niet onder de indruk.

Bitcoin vermijdt elke "single point of failure". Geen enkele partij staat zo sterk dat disfunctie van uitsluitend die partij de zaak kan frauderen. Althans dat is de bedoeling, of dat doel door Bitcoin ook bereikt is onderwerp van vele papers die nu verschijnen. Ik vermoed van niet. Bitcoin mining schaalt niet tot wereldschaal zonder dat er partijen met dominante invloed ontstaan. Er zijn ook andere problemen, maar dat is de gewone wapenwedloop in de techniek, en Bitcoin kan zich wel ontwikkelen (software-evolutie), dat is inmiddels al bewezen.

2.2 De "double spending attack"

Op dit punt aangekomen stellen we ons drie personen voor A, B, en C. A wil iets van B kopen en wil daarvoor betalen door waarde van zijn account naar een account van B te verplaatsen. A wil ook iets van C kopen, waarom niet de zelfde waarde van hetzelfde account gebruiken? Hoe verhindert een bank dit? Door na elke transactie een balans op te maken en de zaak duidelijk in volgorde af te wikkelen. Maar dan moeten ALLE transacties aangaande aan account via die ene bank lopen: de bank is het single point of failure.

Wat is nieuw aan Bitcoin: de eerste realisatie van informational money, zonder single point of failure, met adequate bescherming tegen de double spending attack. In theory was dat er al, maar dit bruikbaar uitprogrammeren is een zeer verdienstelijke stap, wie die stap ook heeft gezet.

2.3 Peer-to-peer systeem: werken met en werken voor het systeem

Oorspronkelijk is Bitcoin gedacht als een P2P-systeem waarin alle deelnemers (clients, of gebruikers van clients) dezelfde taken, rechten, en mogelijkheden hebben. Dat oorspronkelijk idee is volstrekt niet houdbaar gebleken. Een aparte kaste van gebruikers, die de zogenaamde mining uitvoeren, gebruikt daarvoor nu speciaal ontworpen en zeer kostbare hardware. Welk open source programma creëert in enkele jaren tijd een eigen hardware-industrie? Een opmerkelijk succes. Mining is niet meer voor de kleine jongens, het is een miljoenenbedrijf geworden (in Euro's gemeten).

2.3.1 Werken met het systeem: profiteren van de functionaliteit

Dit is relatief simpel, wie Bitcoin gebruikt kan accounts maken, de sleutels daarbij maken, die in wallets (portemonee's), of in de brandkast, bewaren, en kan transacties verkrijgen en versturen. Dit alles tegen zeer lage kosten, en in een wereld waar rente niet bestaat, en waar (zie [6]) het klassieke onderscheid tussen bezit en eigendom niet meer zo voor de hand ligt.

Alle verhalen over de onveiligheid van Bitcoin zijn tot dusverre uit de lucht gegrepen. Er is geen succesvolle aanval bekend. Aan onbetrouwbare of incompetente tussenpersonen die je het geld al dan niet op legale wijze uit de zak kunnen kloppen, is in de wereld van Bitcoin geen gebrek, dat is bij gewoon geld overigens net zo. De presentie van fraude is een teken van maatschappelijk succes. Dat is in de wetenschap ook zo, alleen waar iets te verdienen valt fraudeert men.

De inmiddels beroemde fraudeurs bewijzen het grote publiek dat de wetenschap nu “echt belangrijk” is geworden. En elke zogenaamde Bitcoinfraude versterkt Bitcoin!

2.3.2 Werken voor het systeem: verhinderen van “double spending”

Deelnemers die voor het systeem werken verhinderen in onderlinge samenwerking en gelijktijdige competitive de double spending attack. Zij worden daarbij beloond door het systeem, extra nieuwe Bitcoins komen beschikbaar, vandaar de term mining. Zij worden ook beloond door deelnemers die hun transacties goedgekeurd zien: die betalen een zelf te bepalen fee. Wie te weinig fee beschikbaar stelt loopt het risico door de miners te worden overgeslagen.

Op dit punt aangekomen wordt het verhaal ingewikkeld en wil ik geen inzicht claimen dat ik niet heb. De basis-idee is nog wel te bevatten: transacties die men rondstuurt als gebruiker zijn eerst kandidaat transacties die nog moeten worden gevalideerd, daarbij wordt double spending gedetecteerd en wordt maximaal een enkele transactie van de zelfde amount toegestaan na validatie. Maar dan wordt het ineens heel complex: elke 10 minuten is er een wereldwijde competitie die bepaalt wie van de miners een blok mag aanleveren dat wordt opgenomen in de zogenaamde blockchain (linear geordende keten van blokken te beginnen met het Genesis blok). Zo’n blok bevat enkele duizenden transacties. Iedereen kan de blockchain bekijken en als deelnemers er een van of naar hunzelf uitgevoerde transactie in tegenkomen dat kunnen zij aannemen (maar niet met 100% zekerheid) dat die transactie ook stand zal houden. Alle deelnemers worden geacht permanent toegang tot de blockchain te hebben.

De competitie tussen de miners houdt rekening met het feit dat deze technisch steeds sterker worden. Dit is een zeer opmerkelijke en volstrekt essentieel gebleken flexibiliteit van het Bitcoin protocol. De software van elke miner genereert steeds complexere puzzels (rekening houdend met metingen aan de resultaten van de miners) die de miners per 10 minuten zo snel mogelijk moeten oplossen.

Van belang is hier dat het oplossen van zo’n puzzel bij de nu bekende stand der wetenschap een zoekpartij is met het karakter van een loterij. Er valt weinig verstandigs over te bedenken (maar genoeg om van mining een vak voor specialisten te maken). Wanneer een miner een oplossing heeft gevonden of claimt te hebben gevonden dan kan iedere gebruiker heel snel uitrekenen of dat inderdaad een oplossing is en hoe goed die oplossing is in vergelijking met andere oplossingen. Per 10 minuten proberen alle gebruikers de beste oplossing van de

puzzel aan te wijzen die als bijlage bij een correct blok is bijgeleverd dat door een miner wereldwijd wordt rondgestuurd.

Een miner kan ook een reeks van blokken tegelijk van een betere oplossing voorzien. Zo kan een miner naar believen de historie herschrijven. Dit vergt ongelofelijk veel rekenwerk, maar wie meer dan de helft van de rekencapaciteit van alle miners onder controle heeft en wie heel lang kan wachten kan het gehele systeem kraken. Er is tot dusverre bij mijn weten nog nooit een enkel blok afgekeurd dat niet het laatste in de keten was. Bij het laatste blok is afkeuren niet te vermijden omdat incompetente miners nu eenmaal bar slechte oplossingen kunnen rondsturen. Die moet men wel afkeuren. Maar als zich dan een winnaar uitkristalliseert dan houdt dat in de praktijk ook stand. Doet het dat niet dan krijg je een zogenaamde harde vork, het schrikbeeld van de Bitcoin Foundation.

3 Besluit

Conclusies zijn moeilijk te trekken. Maar dat Bitcoin qua technologie iets levert dat stand houdt durf ik wel als vermoeden uit te spreken. De blockchain techniek is op allerlei punten toepasbaar, niet alleen bij geld. Is Bitcoin geld? Dat is minstens zo zeer een vraag over geld als over Bitcoin. Maar zover zijn we nu in mijn ogen nog niet en ik zou Bitcoin als volgt willen typeren: Bitcoin is een "money-like informational commodity" (MLIC). Argumenten daarvoor vindt men (binnenkort) in [12].

Veel commentaar wekt de suggestie dat de waarde van de Bitcoin met zo'n 650 Euro op dit moment veel te hoog zou zijn. Zo'n commentaar vergt eigenlijk steeds dezelfde repliek: maak een model dat de waarde van zoiets als een Bitcoin bepaalt, en pas die theorie dan toe. Ik meen (zie ook [6]) dat de huidige koers ten opzichte van de Euro past bij een subjectieve kans van 1 op 5000 dat Bitcoin als systeem de komende 20 jaar "overleeft". Iedere inschatting van de waarde van de BTC is onvermijdelijk afhankelijk van de bepaling/keuze van één of meer subjectieve kansen betreffende het wel of niet optreden van zekere voor Bitcoin relevante omstandigheden in de toekomst. Ik denk dat 1 op 5000 eerder te laag dan te hoog is zodat de huidige koers zelfs laag genoemd kan worden.

Hieronder noem nog enkele tamelijk willekeurige bijeengeveegde items uit de inmiddels al volstrekt onoverzichtelijke en dagelijks groeiende literatuur over Bitcoin (en enig eigen werk in de context van deze voordracht). In [6] en[7] vindt men meer verwijzingen betreffende Bitcoin, in [4] enige opmerkingen over "wat is geld".

References

[1] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in Bitcoin. `in: Proc. 2012 ACM Conf. on Computer and Communications Security` (2012).

[2] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten Have a snack, pay with Bitcoins. *13th International Conference of Peer-to-Peer Networks,* (2013).

[3] Simon Barber, Xavier Boyen, Elain Shi, and Ersin Uzun. Bitter to better–how to make Bitcoin a better currency. *In: A.D. Keromytis (ed.): FC 2012*, LNCS 7397, 399–414 (2012).

[4] Jan A. Bergstra. Formaleuros, formalbitcoins, and virtual monies. `arxiv.org/abs/1008.0616v2 [cs.CY]` (2013).

[5] Jan A. Bergstra, Inge Bethke, and Alban Ponse. Cancellation meadows: a generic basis theorem and some applications. *The Computer Journal*, 56(1): 3–14, `doi:10.1093/comjnl/bsx147` (2013).

[6] Jan A. Bergstra and Karl de Leeuw. Bitcoin and Beyond: Exclusively Informational Money. `arXiv:1304.4758v2 [cs.CY]` (2013).

[7] Jan A. Bergstra and Karl de Leeuw. Questions related to Bitcoin and other Informational Money. `arXiv:1305.5956v2 [cs.CY]` (2013).

[8] J.A. Bergstra and C.A. Middelburg. Inversive meadows and divisive meadows. *Journal of Applied Logic*, 9(3): 203–220 (2011).

[9] J.A. Bergstra and C.A. Middelburg. Preliminaries to an investigation of reduced product set finance. *JKAU: Islamic Economics*, 24(1):175–210 (2011).

[10] J.A. Bergstra and C.A. Middelburg. Interest prohibition and financial product innovation. *In: Finance Islamique: Regard(s) sur une Finance Alternative, Mazars Hadj Ali*, 274–284 (2012).

[11] J.A. Bergstra and J.V. Tucker. The rational numbers as an abstract data type. *Journal of the ACM*, 54 (2), Article 7 (2007).

[12] Jan A. Bergstra and Peter Weijland. Bitcoin: a money-like informational commodity. In preparation (2014).

[13] Jerry Brito and Andrea Castillo. Bitcoin, a Primer for Policymakers. *Mercatus Center, George Mason University,* (2013).

[14] Nicolas T. Courtois, Marek Grajek, and Rahul Naik. The Unreasonable Fundamental Incertitudes Behind Bitcoin Mining. arXiv preprint `arXiv:1310.7935`, (2013).

[15] Gerion Entrup. Bitcoin, Der Stärkere gewinnt. *Thesis Leibniz Universität Hannover, Institut für Theoretische Informatik,* `http://www.thi.uni-hannover.de/fileadmin/forschung/arbeiten/entrup-ba.pdf` (September 2013).

[16] Luciano Floridi. Philosophy of Information. *Oxford University Press*, ISBN 978-0-19-923239-0 (2011).

[17] Reuben Grinberg. Bitcoin: an alternative digital currency. *Hastings Sci. and Tech. Law J.*, 159–208 (2012).

[18] Jarek Gryz. Privacy as informational commodity. *Proc IACAP,* `philpapers.org`, (2013).

[19] Brian P. Hanley. The False Premises and Promises of Bitcoin. arXiv preprint `arXiv:1312.2048` (2013).

[20] Matthias Herrmann. Implementation, evaluation, and detection of a double-spend attack on Bitcoin. *MSc Thesis, ETH Zürich* (2012).

[21] Danny Yuxing Huang. Profit-driven abuses of virtual currencies. `http://sysnet.ucsd.edu/ dhuang/pmwiki/uploads/Main/huang-research-exam.pdf` UCSD, (2013).

[22] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ECDSA). *IJCS*, 1,36–63 (2001).

[23] Nikolei M. Kaplanov. Nerdy money: Bitcoin, the private digital currency, and the case against its regulation. *Temple University Legal Studies Research Paper* `http://ssrn.com/abstract=2115203` (2012)

[24] Bill Maurer, Taylor C. Nelms, and Lana Swartz. "When perhaps the real problem is money itself!": the practical materiality of Bitcoin. *Social Semiotics*, DOI:10.1080/10350330.2013.777594 (2013).

[25] Satoshi Nakamoto. Bitcoin: a peer-to-peer electronic cash system. `http://Bitcoin.org/Bitcoin.pdf` (2008).

[26] Ian Steadman. Wary of Bitcoin? A guide to some other cryptocurrencies. `http://www.wired.co.uk/news/archive/2013-05/7/alternative-cryptocurrencies-guide/page/4` (2013).

Electronic Reports Series of section Theory of Computer Science

Within this series the following reports appeared.

[TCS1301] B. Diertens, *The Refined Function-Behaviour-Structure Framework*, section Theory of Computer Science - University of Amsterdam, 2013.

[TCS1202] B. Diertens, *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-Orientation*, 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 Architecures*, 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: www.science.uva.nl/research/prog/

Electronic Report Series

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 107
1098 XG Amsterdam
the Netherlands

www.science.uva.nl/research/prog/