Bitcoin: Informational Money en het Einde van Gewoon Geld

Bergstra, J.A.

Publication date
2014
Document Version
Final published version

University of Amsterdam

Theory of Computer Science

Bitcoin: Informational Money en het Einde van Gewoon Geld

J.A. Bergstra

Report TCS1408 June 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: Informational Money en het einde van Gewoon Geld

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

June 2, 2014

Abstract

Deze paper bevat een verkorte beschrijving van de Nakamoto architectuur voor informational money, een positionering van Bitcoin, een "money-like informational commodity", als een verschijningsvorm van iGoud, en een bespiegeling over de vraag hoe hoe informational money het einde van Gewoon Geld in kan luiden.[1]

Keywords and phrases: informational money, informational commodity, P2P system, geld, iGoud, double-spending attack, Bitcoin.

1 Inleiding

Satoshi Nakamoto is een anonymus die in 2013 bij het grote publiek is doorgebroken via zijn geesteskind Bitcoin.[2]

Eind 2008 kwam de open source implementatie van de eerste Bitcoin client beschikbaar en begin 2009 werd het Bitcoin systeem operationeel na de introductie van het zogenaamde Genesis block.[3]

Stel u voor dat men nieuw geld wil ontwerpen,"iGoud", waarbij de "i" staat voor moderniseren en verbeteren.

iGoud wil men per atoom, of althans op nano-schaal, kunnen verhandelen. Hoe veel tijd en geld men ook in mijnbouw steekt iGoud moet schaars bli-

[1] Uitgewerkte versie van een voordrachtgehouden als onderdeel van "Kamermans Kermis: Het Einde" in De Balie, Amsterdam, 31 mei 2014. Deze tekst is deels gebaseerd op: voordacht in het kader van de College Tour, Spui 25, 5 februari 2014 (zie [2]).

[2] Zoe [17]. Wie of wat achter deze naam schuil is niet (algemeen) bekend. Ongevalideerde vermoedens daaromtrent komen en gaan.

[3] In het vervolg gebruik ik "block" en "blok" door elkaar.

jven. iGoud moet je goed en goedkoop kunnen bewaren, het moet niet kunnen roesten, verdampen, of verbranden. Een transactie van iGoud functioneert zonder gepantserde auto's, en zonder ondersteuning door gespecialiseerd personeel. iGoud is technologisch gedemocratiseerd, maar iGoud vult geen kiezen en deugt niet voor een trouwring.

1.1 Wat is geld?

Ik denk dat geld een onderdeel vormt van een communicatieprotocol en daarmee geheel een zaak van informatica is, maar dat wil ik u niet aanpraten.

Ik kwam tamelijk toevallig medio 2012 op Bitcoin uit na het bekijken van Islamitisch geld (zie [9, 10]). Islamitisch geld is nu zo'n 500 maal groter qua circulatie dan Bitcoin, en is alleen al om die reden ieders aandacht waard. Ik meen dat Bitcoin bruikbaar is als implementatie van de eisen die men stelt aan Islamitisch geld (zie [3]). Bitcoin is ook een kandidaat voor iGoud, het oordeel daarover is aan u, want er is nog geen wetenschap die deze vraag op doorslaggevende wijze beantwoorden kan.

Mijn zorg over de vraag wat geld is moet men niet verkeerd begrijpen; ik heb iets met zulke "domme vragen", zoals de vraag wat de uitdrukking 1 gedeeld door 0 voor kan stellen (zie [12, 8] en [5]), de vraag wat een algoritme is (zie [11]), en de vraag of en waarom 23 een natuurlijk getal is (zie [4]).

1.2 Bitcoin: ingebouwde schaarste

De eenheid van het Bitcoin systeem heet ook Bitcoin en de internationale afkorting ervoor is BTC. Er ontstaan maximaal 21 miljoen Bitcoins en dat duurt nog een aantal jaren. We zijn nu op de helft van de creatie van de Bitcoins. Elke Bitcoin kan men in 100 miljoen Satoshi's verdelen, samen dus uiteindelijk 2,1 maal 10 tot de macht 15 Satoshi's (ofwel zo'n 200.000 per wereldburger) waarvan een onbekend maar langzaam toenemend aantal in de loop der jaren zoek raakt en nooit meer gebruikt zal kunnen worden.[4] Transacties kun je op je laptop uitvoeren, en eigenlijk hoef je niet te weten hoe men een computer beveiligt. Gewoon na elke transactie een nieuwe laptop kopen en de oude echt

[4]BTC verlies treedt op door het verloren gaan van sleutels. Het Bitcoin systeem kent geen boekhouding van de sleutels die nog bekend zijn en geen mechanisme om een sleutel definitief verloren geraakt te verklaren en de via die sleutel toegankelijke BTC amount weer in circulatie te brengen. 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 onder de aanname dat elke sleutel een positieve en op den duur stabiele kans heeft om per tijdseenheid verloren te raken. Het is net zo iets als het afkoelende heelal.

vernietigen volstaat. Maar je moet wel een brandkast hebben als je serieus met Bitcoin aan de gang gaat. En als je bezit (in BTC) bij overlijden naar één of meer overlevenden moet vererven vergt dat goed doordachte voorbereiding van alle betrokkenen want dat gaat beslist niet vanzelf.

2 Informationeel iGoud, wiskundig iGoud

Hoe maken we iGoud? Aan willekeurige keuzen valt niet te ontkomen. Ik bekijk enkele keuzen die voor Bitcoin werden gemaakt. Bitcoin is slechts één van onmetelijk veel mogelijke ontwerpen voor een informational money.[5] De combinatie van een aantal van deze keuzen is in [6] in meer detail beschreven onder de benaming Nakamoto architectuur, een architectuur voor informational money (of zo men wil informational currency).

Dragers van waarde worden bitreeksen van 128 bits. Die kan iedereen zelf aanmaken, dat vergt dan wel technieken uit de zogenaamde elliptic curve cryptography (derdegraads krommen, zie ook [16]). Het systeem kent dan aan zo'n reeks een nominale waarde toe (uitgedrukt in BTC), en transacties kunnen die waarde geheel of gedeeltelijk verplaatsen naar andere bitreeksen.[6] Zo'n bitreeks is net als een bankrekening. Zo ontstaat "informational money", geld dat alleen uit informatie bestaat.[7]

Hoe bewijs je nu je bezit van zo'n bitreeks, en je beschikkingsmacht over de waarde die het Bitcoin systeem daaraan toekent. We noemen de bitreeks voor het gemak een account. Tegelijk met dat account maak je een andere bitreeks, ook 128 bits lang en eveneens met behup van de derde graads-krommen uit de algebraïsche meetkunde. Deze tweede bitreeks (die een op en hoort bij het account) noemt men de sleutel van dat account.[8] 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 wel veilig en geheim bewaren, bijvoorbeeld op papier in een kluis. Wie de sleutel van een account kent beschikt over de waarde die hoort bij dat account en kan een ander

[5] Bitcoin heeft inmiddels een kleine 100 varianten, de zogenaamde altcoins.

[6] Transacties staan meestal in directe relatie met een tegenprestatie, bijvoorbeeld het leveren van diensten of goederen. Is tegenprestatie de overdracht van een hoeveelheid Gewoon Geld naar de agent die de transactie onderneemt dan ontstaat prijsvorming van de Bitcoin in termen van Gewoon Geld.

[7] In [13] betogen we dat Bitcoin op dit moment beter als een "money-like informational commodity" kan worden gekarakteriseerd dan als een "informational money". Voor wie "currency" prefereert boven "money" zou van een "currency-like informational commodity" gesproken kunnen worden.

[8] Essentieel aan de elliptic curve cryptografie is dat een algoritme om bij een account de bijbehorende sleutel te zoeken weliswaar bekend is maar dat geen algorithme bekend is waarmee men dat in pakweg 1000 jaar kan doen op de thans bestaande computers of computernetwerken.

die dezelfde sleutel ook kent deze beschikkingsmacht in enkele seconden ontnemen. Hieruit blijkt het belang van de fysieke kluis als exclusieve bewaarplaats voor deze informationele sleutels in complexe omstandigheden.

Het is in Bitcoin voor de hand liggend dat een gebruiker veelvuldig nieuwe account/sleutel-paren aanmaakt. De idee is dat er zoveel bitreeksen van 128 bits zijn dat dit in de praktijk geen begrenzing oplevert.

2.1 Nakamoto wilde meer

Een bank gebruikt traditioneel analoge handtekeningen om te bepalen wie een account mag gebruiken. Dat mechanisme is thans eigenlijk verouderd. Tegenwoordig werkt authenticatie via vrij eenvoudige interfaces op een computer maar slechts weinigen weten wat er echt aan de hand is wanneer je inlogt op de site van je bank. Waarom kijkt de maffia niet mee? En waarom vertrouwen we de bank? Het antwoord op de tweede vraag is bekend: wij vertrouwen De Nederlandse Bank (DNB) en die vertrouwt op haar beurt onze bank (of niet en dan houdt de DNB die deceptie jammergenoeg soms voor zichzelf uit piëtiet). Nakamoto was niet onder de indruk van deze stand van zaken.

Bitcoin vermijdt elke "single point of failure". Geen enkele partij staat zo sterk dat disfunctie van uitsluitend die partij het systeem onderuit kan halen. Althans dat was destijds de bedoeling; of dat doel met de introductie van Bitcoin ook bereikt wordt is de vraag, ik vermoed eigenlijk van niet. Bitcoin mining, waarover hieronder meer, schaalt niet tot wereldschaal zonder dat er partijen met te dominante invloed ontstaan. Zulke partijen worden vanzelf een single point of failure voor het gebruik van sommige accounts, en mogelijk zelfs voor het totaal van de Bitcoin gerelateerde activiteiten van individuele gebruikers. Er zijn ook andere problemen (zie [14]), maar Bitcoin kan zich wel ontwikkelen (software-evolutie), en die ontwikkeling kan veel problemen aan, dat is inmiddels wel bewezen.

2.2 De "double spending attack"

We bekijken nu drie personen A, B, en C. A wil iets van B kopen en wil daarvoor betalen door waarde p BTC van zijn account X naar een account Y van B te verplaatsen. A wil ook iets van C kopen. Waarom zou A niet de zelfde waarde (p BTC) van hetzelfde account (X) gebruiken en die naar account Z van C proberen te sturen? Dan is A weliswaar goedkoop uit maar na enige tijd blijkt dat B of C geloof gehecht heeft aan een transactie die bij nader inzien niet plaats

heeft gevonden (en die in een gewoon bancair transactiesysteem dan achteraf ongedaan zou worden gemaakt, iets wat in Bitcoin onmogelijk is). Dit is de double spending attack (zie ook [15]). De aanval wordt uitgevoerd door A met B of C als slachtoffer op termijn, aannemende dat het systeem goed functioneert. Wie van de twee de schade lijdt maakt A niet uit.

Hoe verhindert een gewone bank zoiets? Door na elke transactie een balans op te maken van de account X van A en de zaken (met name de transacties betreffende X) in volgorde af te wikkelen. Dan moeten alle transacties aangaande deze account X via die ene bank lopen: de bank is het single point of failure (althans vanuit het perspective van A als houder van account X).

2.3 De vernieuwing die het Bitcoin transactiesysteem introduceert

Wat is nieuw aan Bitcoin: het is de eerste realisatie van informational money als peer-to-peer (P2P) system, zonder single point of failure, en met adequate bescherming tegen de double spending attack. Het theoretische concept bestond al, in een tamelijk vage omschrijving waarop men m.i. geen patent zou kunnen baseren, maar dit idee bruikbaar uitprogrammeren, het resultaat daarvan vrij beschikbaar te maken en dat op zodanige wijze dat de open source software die zo is ontstaan door een zichzelf rond die software groeperende community goed onderhouden kan worden is een zeer verdienstelijke stap geweest van de anonieme Nakamoto.

3 De Nakamoto-architectuur

In [6, 7] wordt onder de benaming Nakamoto-architectuur een abstracte beschrijving van het Bitcoin mechanisme gegeven. Deze architectuur laat nog veel parameters vrij en Bitcoin ontstaat door zulke parameterwaarden te kiezen. De Nakamoto-architectuur zie ik als een architectuur voor informational money (informational currency zo men wil) en in het bijzonder als een architectuur voor transactiemechanismen en voor globale boekhouding. Hieronder volgt een sterk verkorte weergave daarvan.

3.1 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 oorspronkelijke idee is onhoudbaar gebleken. Er zijn nu twee klassen van deelnemers, gewone deelnemers, ook wel gebruikers genoemd, en buitengewone deelnemers, ook wel miners genoemd.

De gewone deelnemers gebruiken Bitcoin voor belegging, speculatie, betaling (transactie), transfer, en voor een veelheid van meer of minder creatieve vormen van bedrog. De buitengewone deelnemers zijn de zogenaamde miners (mijnbouwers). Deze miners vormen inmiddels een aparte kaste van gebruikers. De zogenaamde mining die zij uitvoeren vraagt om speciaal ontworpen en toenemend kostbare hardware.[9] De principiële taak van het collectief van de miners is er voor te zorgen dat double spending attacks worden vermeden en wel op zodanige wijze dat geen enkele van hen individueel, noch een coalitie van miners, zich tot een single point of failure kan ontwikkelen.

3.1.1 Werken met het systeem: profiteren van de functionaliteit

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

Alle verhalen over de onveiligheid van Bitcoin zijn uit de lucht gegrepen. Er is geen succesvolle aanval bekend. Maar aan onbetrouwbare of incompetente tussenpersonen en "dienstverleners" is in de wereld van Bitcoin beslist geen gebrek.[10]

3.1.2 Werken voor het systeem: verhinderen van "double spending"

De buitengewone deelnemers die ten gunste van het Bitcoin-systeem werken verhinderen in onderlinge samenwerking en met gelijktijdige competitie de double spending attack. Zij worden daarbij beloond door het systeem met oude zowel als met nieuwe Bitcoins. Nieuwe Bitcoins komen uitsluitend langs deze route

[9] Welk open source programma creëert in enkele jaren tijd een eigen hardware-industrie? Een hoogst opmerkelijk succes. Mining is een economisch bepaalde activiteit: de miner zoekt een niche waar een kostenprofiel betreffende elektriciteit, koeling, de operationele kosten van een rekeninstallatie, en de financiering en vervanging van hardware, in een adequaat gewogen mix gunstig afsteekt tegen het kostenprofiel waarmee concurrerende miners werken.

[10] Het optreden van fraude is een teken van maatschappelijk succes van de onderliggende activiteit. Dat ligt in de wetenschap ook zo want alleen waar iets te verdienen valt fraudeert men. De inmiddels beroemde fraudeurs bewijzen het grote publiek dat de wetenschap nu "echt belangrijk" is geworden. Elke zogenaamde Bitcoinfraude versterkt Bitcoin.

als beloning voor miners beschikbaar, vandaar de term mining. Om die beloning te verdienen moet een miner, naast het controleren van een reeks van voorgestelde maar nog niet gevalideerde transacties een combinatorische puzzel zo goed mogelijk en zo snel mogelijk oplossen. Wie de beste oplossing het eerste vindt en rondstuurt en direct daarna de erkenning van een meerderheid van alle deelnemers (gewoon en buitengewoon) verkrijgt van kwaliteit en tempo van de oplossing wint te de competitie om de constructie van het volgende blok in de zogenaamde blockchain. De winnaar vangt de uitgeloofde premie. Die premie is voor alle blokken dezelfde gedurende een tijdsinterval van enkele jaren maar wordt met enige regelmaat gehalveerd. Dat leidt ertoe dat de creatie van Bitcoins asymptotisch tot een eindpunt convergeert. Deze winnaar wordt behalve met nieuwe Bitcoins ook beloond door deelnemers die hun transacties in het voorgestelde blok opgenomen (en daarmee goedgekeurd) zien: die deelnemers betalen de winnende miner een zelf te bepalen fee die in een transactie vooraf wordt vastgelegd. Die fee stamt af van al bestaande (ofwel oude) Bitcoins. een gebruiker die een transactie voorstelt en die daarbij te weinig fee ter beschikking stelt loopt het risico dat de betreffende transactie door de meest kansrijke en daamee meest invloedrijke miners wordt genegeerd met als gevolg dat validatie van de transactie kan uitblijven of onaangenaam lang op zich kan laten wachten.[11]

3.1.3 Een comptetitieronde per 10 minuten

Een gebruiker die een transactie (als actie) wil uitvoeren maakt op eigen apparatuur een transactie (nu gezien als informationeel item) aan, dat is een pakket van data met een in het Bitcoin protocol vast omschreven formaat. Vervolgens stuurt de gebruiker deze transactie via het rond aan alle andere gebruikers.[12] Transacties die men rondstuurt als gebruiker zijn eerst kandidaat transacties die nog moeten worden goedgekeurd (gevalideerd). Als onderdeel van de validatie wordt tijdens de constructie van een blok door een miner eventuele double spending gedetecteerd en wordt hoogstens een enkele transactie van de zelfde amount toegestaan in een correct blok.[13]

Nu wordt het verhaal echter geheel anders dan men in de wereld van het

[11] Deze gebruiker moet er dan overigens rekening mee houden dat op een willekeurig moment in de toekomst die tot dan toe "vergeten" transactie als nog wordt gevalideerd, tenzij hij een "mislukte" double spending attack op zichzelf uitvoert met een meer attractieve alternatieve transactie die een hogere fee levert.

[12] De vraag hoe het systeem op eerlijke wijze er voor zorgt of kan zorgen dat alle gebruikers min of meer gelijktijdig deze transactie waar kunnen nemen is heel serieus. DoS aanvallen kunnen juist dat aspect van het systeem onplezierig verstoren met onaangename gevolgen voor de gebruiker.

[13] Bij een double spending attack wordt dus minstens een van beide transacties niet succesvol gevalideerd.

Gewone Geld zou verwachten. Elke 10 minuten wordt er een wereldwijde competitie uitgeschreven waarmee wordt bepaald welke miner een blok mag aanleveren dat wordt opgenomen in de blockchain, de linear geordende keten van blokken te beginnen met het Genesis blok die de gehele transactiehistorie van het Bitcoinsysteem codeert en waarover de community van gebruikers op dat moment volgens het Bitcoin protocol overeenstemmming heeft. Zo'n blok bevat enkele duizenden transacties die gevalideerd worden geacht juist omdat ze in een blok uit de blockchain voorkomen. Iedereen kan de blockchain bekijken en zodra deelnemers (gebruikers) er een van of naar hunzelf uitgevoerde transactie in tegenkomen dan kunnen zij aannemen dat die transactie ook langdurig stand zal houden, zij niet met 100% zekerheid voor altijd. Alle deelnemers worden geacht steeds toegang tot de blockchain te hebben bij voorbeeld door er zelf een instantie van bij te houden.

Het oplossen van de aan de miners toegespeelde puzzel is bij de nu bekende stand der wetenschap een domme maar goed parralleliseerbare zoekpartij per computer met het karakter van een loterij. Er valt weinig diepzinnigs over te bedenken, maar nog wel genoeg verstandigs om van mining een discipline voor technische specialisten te maken. Wanneer een miner een oplossing heeft gevonden of claimt te hebben gevonden en deze met een kandidaat blok de wereld (van alle deelnemers) rond stuurt dan kan iedere deelnemer heel snel en goedkoop uitrekenen of dat inderdaad een oplossing is en hoe goed die oplossing is in vergelijking met andere oplossingen.

Per 10 minuten proberen alle deelnemers 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.

3.1.4 Confirmatie van transacties

Wanneer agent A een transactie T naar B heeft gestuurd is het voor beide partijen (A en B) van belang om 10 minuten te wachten op een nieuw blok in de blockchain en te bekijken of de transactie T daarin verschijnt. Voor de zekerheid kan met beter ook het volgende blok afwachten. Zolang het blok met transactie T erin in de blockchain blijft staan weet men zeker dat er geen double spending attack is uitgevoerd die met T incompatibel is. Is ook een volgend blok aangetroffen dan kan men er met zeer hoge waarschijnlijkheid vanuit gaan dat het vorige blok in nu en in de toekomst in Bitcoin onaantastbaar is. Op deze wijze kunnen A en B de transactie T op basis van de validerende activiteiten van miners die tot opneming in een blok uit de blokchain hebben geleid, ook

confirmeren.

3.1.5 Gedeeltelijk afbreken van de blockchain

Een miner kan ook een reeks van blokken tegelijk van een betere oplossing van de betrokken puzzels voorzien. Dan kan deze miner ook de inhoud van die blokken wijzigen en juist andere transacties van een paar dat een double spending attack vertegenwoordigt toelaten in nieuw te maken blokken met overeenkomstige rangnummers. Zo kan een miner naar believen de historie van Bitcoin herschrijven, of althans de abstractie van die historie die door de blockhain wordt gegeven. Dit verklaart het hierboven genoemde caveat dat de transacties in de blockchain niet meer ongedaan gemaakt kunnen worden. Dat in theorie altijd nog gebeuren. Dit vergt ongelofelijk veel rekenwerk, maar wie meer dan de helft van de rekencapaciteit van alle miners onder controle heeft en wie er lang genoeg aan wil werken kan het gehele systeem kraken, ofwel alle gevalideerde blokken door andere blokken vervangen en elke double spending attack desgewenst anders oplossen. Er is tot dusverre bij mijn weten nog nooit een blok afgekeurd dat niet het laatste in de keten was. Bij het laatste blok is zo nu en dan een afkeuring niet te vermijden omdat minder slagvaardige miners nu eenmaal relatief slechte oplossingen kunnen rondsturen. Die moet men wel afkeuren zodra betere oplossingen worden rondgestuurd. Als zich echter na enige tijd een winnaar uitkristalliseert dan houdt die toewijzing in de praktijk ook stand.

3.1.6 Automatische schaling met technische vooruitgang

De competitie tussen de miners schaalt met hun technische vooruitgang. Dit is een zeer opmerkelijke en cruciale 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. Het gemiddelde probleemoplossend vermogen inzake de combinatorische puzzels waar het bij mining om gaat is sinds 2009 verbluffend toegenomen.

4 Bitcoin en het Ponzi schema

Het Ponzi mechanisme (ook wel piramidespel genoemd) verklaart waarde van een commodity vanuit een serieuze vorm van handel, maar levert het waardebehoud slechts op basis van handel die al uitgaat van een verwachte waardestijging. Zo'n

setting laat op termijn een groot aantal deelnemers in een markt achter met items waarvoor de verwachting van waardebehoud niet meer gerechtvaardigd blijkt. Wie Bitcoin een Ponzi schema noemt, zou de NL huizenmarkt tussen pakweg 1998 en 2008 toch ook zo moeten zien. Ik geloof dat Bitcoin in een Ponzi fase kan verkeren, maar dat is niet noodzakelijk altijd het geval. Een zuiver Ponzi schema komt nooit op verdedigbare bodemwaarden uit, iets wat met Bitcoin wel degelijk mogelijk is. Daar ligt ook een overeenkomst met de huizenmarkt.

4.1 Informational money en het einde van Gewoon Geld

Er bestaat niet zoiets als Gewoon Geld. Het zogenaamde gewone geld gaat door een voortgaande evolutie en de vrijwel constante vorm van de bankbiljetten levert een optische illusie van behoud van concept en mechanisme. Vrijwel niemand in ons land weet hoe het geld thans werkt en wat de overgang van Gulden naar Euro conceptueel en in termen van de architectuur van transacties en van bezit en eigendom van assets in Euro betekende. Hoe werkt de ECB, en hoe werkt de DNB, en hoe werken beide samen, en welke rol spelen de handelsbanken daarin, en waarom hebben we niet als particulier direct een account (in EUR) bij de ECB. Waarom zijn we nog steeds zo afhankelijk van banken en van hun fractional reserve banking. De actuele antwoorden op zulke vragen migreren voortdurend door een formidabel grote design space van "Gewoon Geld" en de potentiële antwoorden op deze vragen zijn een klassiek onderwerp wetenschappelijk onderzoek.

4.2 Gewoon Geld is ongewoon complex

Bitcoin is bij nadere beschouwing eigenlijk een wonder van eenvoud in vergelijking met hedendaags Gewoon Geld, het heeft ook veel minder functies (zie [6] inzake de ethiek van Bitcoin). Alleen al de relatie tussen politiek en Gewoon Geld is zo complex, en de mechanica van de hiërarchie van banken is dat ook. Digitaal Gewoon Geld vergt formidabele inspanningen inzake informatiebeveiliging, veel meer dan Bitcoin ooit gaat vragen. Zit information security niet aan de basis van een ontwerp dan faalt het op den duur en niet zo zuinig ook. Daar zit een zwakte van het gehele internet en ook van het hedendaagse "gewone" digitale geld.

4.3 Van een horizontale verkaveling naar een verticale verkaveling

Zoals in [6] is beschreven maken we nu misschien een transitie mee van een globale architectuur waarin Conventional Monies (verschijningsvormen van Gewoon Geld) die per geografische eenheid in een mix van samenwerking en competitie naast elkaar staan (horizontale verkaveling) worden vervangen door een familie van Informational Monies die elk volstrekt internationaal zijn en die een functionele verkaveling vertonen (verticale verkaveling). Bitcoin is eigenlijk niet geschikt voor het kopen van een enkele kop koffie, maar juist wel voor de transfer van een heel groot bedrag. De ingebouwde beveiliging, die een internationale competitie oproept ter validatie van elke transactie hoe klein ook is vanzelfsprekend niet het laatste woord op dit gebied.

Door een overgang van hoizontale naar verticale verkaveling van de monies kan het Gewone Geld verdwijnen net zoals de gewone krant verdwijnt en het gewone boek verdwijnt. Die processen duren steeds langer dan men denkt, maar het is net als bij de treinen, het treinkaartje verdwijnt pas vele jaren nadat nadat men ziet hoe de informatietechnologie het kan vervangen.

De papieren krant houdt stand, maar voor hoelang. Het papieren geld gaat ons verlaten, maar wanneer. Maar die stap is nog een evolutieslag van het Gewone Geld. Modern Informational Money in de stijl van Bitcoin integreert de gehele technologische keten van opslag en transactie in een enkele smartphone. Dat is met Gewoon Geld zoals het er nu bij staat onmogelijk. Voor de hand ligt dat Gewoon Geld delen van de Nakamoto-architectuur gaat integreren. De vraag of Gewoon Geld verdwijnt (vanwege Bitcoin) hangt dan samen met de uiteindelijke impact van die integratieslag.

4.4 Bitcoin versus Keynes

Bitcoin-style informational money is niet compatibel met Keynisanisme in het gebruik en het beheer van geld. Werkgelegenheid, industriepolitiek, sociaal beleid, natuurbehoud en innovatie, het moet allemaal worden benaderd zonder ingrepen in geldvolumes en transactiemechanismen. Misschien is dat op den duur ook wel zo eenvoudig.

5 Besluit

Dat Bitcoin qua technologie een vernieuwing voorstelt die stand zal houden vermoed ik wel. Dat is niet strijdig met een eclipseren van Bitcoin als concrete informational money. Het is goed denkbaar dat pas een opvloger van Bitcoin de slag kan winnen. Zo is het in de ICT zo vaak gegaan.

De blockchain techniek is op elke infomational commodity toepasbaar, niet alleen op informational monies.

Soms wekt iemand de indruk dat de handelswaarde van de Bitcoin met zo'n 450 Euro op dit moment onverklaarbaar hoog zou zijn. Zo'n commentaar vergt eigenlijk steeds hetzelfde antwoord: maak eerst een theoretisch model dat de waarde van een Bitcoin verklaart, schat daarin de relevante parameters, en pas die theorie daarna onbevooroordeeld toe (zie ook [6]). Elke inschatting van de waarde van de Bitcoin is onvermijdelijk afhankelijk van de bepaling/keuze van één of meer subjectieve kansen betreffende het wel of niet in de toekomst optreden van een reeks van denkbare en voor Bitcoin relevante omstandigheden.

In [6] en[7] vindt men meer verwijzingen betreffende Bitcoin, in [1] enige opmerkingen over mogelijke definities van geld. Er is over Bitcoin inmiddels ook een zeer informatieve pagina op Wikipedia met een uitgebreide literatuurlijst.

References

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

[2] Jan A. Bergstra. Bitcoin, een "money-like informational commodity". University of Amsterdam, Informatics Institute, Section Theory of Computer Science, `Report TCS1401` (Februari 2014).

[3] Jan A. Bergstra. Bitcoin and Islamic Finance (version 2). University of Amsterdam, Informatics Institute, Section Theory of Computer Science, `Report TCS1406v2` (May 2014).

[4] Jan A. Bergstra. Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers. University of Amsterdam, Informatics Institute, Section Theory of Computer Science, `Report TCS1407v1` (May 2014).

[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 C.A. Middelburg. On algorithmic equivalence of instruction sequences for computing bit string functions. `arXiv:1402.4950v2 [cs.LO]` (2014).

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

[13] Jan A. Bergstra and Peter Weijland. Bitcoin: a money-like informational commodity. University of Amsterdam, Informatics Institute, Section Theory of Computer Science, `Report TCS1402` (Februari 2014).

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

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

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

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

Electronic Reports Series of section Theory of Computer Science

Within this series the following reports appeared.

[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. 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 904
1098 XG Amsterdam
the Netherlands

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