Formalizing contract law for smart contracts

T.F.E. (Eric) Tjong Tjin Tai
Department of Private Law, Tilburg Law School, Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
t.f.e.tjongtjintai@uvt.nl

Abstract

Smart contracts are presented as a self-executing, autonomous alternative to traditional contracts that require enforcement by court involvement. The first experiences with smart contracts show that contracts involve more than conditional execution. This paper proposes a method for formalizing contract law in order to make this as far as possible tractable for incorporation in smart contracts. Contract law can be viewed as a set of rules for resolving disputes while also protecting legitimate interests of parties. Examination of a simple example shows that formalization is complicated by several issues that arise from lay expectations regarding (smart) contracts. This exercise also shows implicit assumptions and characteristics of contract law, and challenges whether such characteristics are unavoidable.

1 Introduction

Smart contracts have been introduced as a concept in the mid-1990's [Sza94]. The idea of implementing contractual rules in a programming language has also been applied to financial contracts, using functional programming languages [Pey00]. With the advent of bitcoin and what is called blockchain technology, attempts have been made to construct a smart contract programming environment building on virtual currency (analogous to bitcoin), using procedural programming languages.[1]

The first experiences gained with smart contracts are not entirely satisfactory. In particular, it appears that smart contracts require additional mechanisms to correct mistakes and to resolve differences between contract parties. It is submitted that these are precisely the issues that contract law has created mechanisms for, based on centuries of practical experience. The question is how this experience can be used effectively for smart contracts. Answering this question also leads to further insights regarding the limitations of smart contracts and the nature of contract law.

2 The promise of smart contracts

2.1 Bitcoin and blockchain technology

We will presume that the reader is familiar with the basics of bitcoin and blockchain technology. Bitcoin is a virtual currency, based on a digital administration or ledger (the blockchain). The ledger is kept up to date by processing payment transactions sent by bitcoin ‘owners using a system of public key encryption. The objectivity of the ledger (the protection against fraudulent or incorrect verification of transactions) is ensured by a distributed system of verification, whereby it is not a priori known which network node will create the next ‘block of accepted transactions. This system revolves around the simultaneous notion of network acceptance of a block and the reward of the ‘winner with new bitcoin and/or transaction fees.

Several attack vectors have been discussed in literature, while furthermore the present system of verification of transactions is resource intensive and stands in need of improvement. These issues and disadvantages do not concern us here.

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

[1] For the following no extensive references will be provided. See [Tjo17] and [Ras16] for such references about bitcoin, blockchain, and the experiences with smart contracts. Most information has to be collected from Internet-based sources as there are as yet not many published academic sources available.

2.2 Smart contracts as an alternative to traditional contracts

Contemporary smart contracts build on the distributed mechanism of the blockchain by extending the bitcoin script language, in some instances into a Turing complete system, such as with Ethereum (discussed below, section 2.3). The advantage of an extended script language is that it allows complex payment transactions. An example is payment conditional on external input confirming receipt of an item, or sending out a signal that interrupts the starter of a car in case that an interim payment for a car has not been received [Sza94].

As bitcoing gained footing in the market place, it become possible to have actual self-executing programs and contracts tied to generally accepted currency. Furthermore the rise of Internet-connected objects (cf. the Internet of Things) makes it feasible to have a bitcoin-like network create off-line effects and receive real-world input. These developments have led to the claim that it would be possible to completely supplant traditional legal contracts by the use of smart contracts. This has given rise to several highly publicised start-up companies offering smart contracts as their business model. Smart contracts in this sense have been claimed to lead to 'code as law'. Programs in the script language, code, would be used instead of law; the code would serve the same function as law. This involves several claims.

- As the contract is constructed in a script language instead of in legally valid terms, this would allow the creation of contracts without the need of legal expertise.
- Payment due to the contract could be executed automatically in the virtual currency, which would bypass traditional banks or other payment intermediaries.
- Execution of the contract could be performed automatically without possibility of interference by any party, and without the need of legal enforcement.
- As execution is performed on the blockchain, which arguably is difficult to influence by ordinary state force, the code of smart contracts is immutable by law and thereby operates outside of and in place of the law.

2.3 Experiences with smart contracts

Several smart contract systems have been created by now. One of the most succesfull in terms of market adoption is the Ethereum framework. This system closely follows the model of bitcoin, albeit with certain modifications that would remedy the perceived weaknesses of bitcoin. Its script language, called Solidity, offered a Turing complete set of instructions. It contains a sophisticated set of mechanisms to allow programming of any conceivable contract. Ethereum has been started by selling a seed amount of its currency (called Ether) which was sold for over $ 18 million. At one point in 2016 the sum total of Ether was said to be worth over $ 1 billion.

Ethereum was constructed as a platform that allows setting up specific applications that allow concrete smart contracts. One such application was 'The DAO'. The DAO was intended to allow set-up of investment contracts. In 2016 the application suffered from exploitation of a weakness in the system which allowed the contract creator to completely drain the virtual wallet of all investors in the contract. The reaction of the participants in Ethereum was to modify the system with a 'hard fork', thereby allowing the investors to recoup the money that had become frozen in the contract.[2]

This experience gave further fuel to the discussion on weaknesses of a smart contract system such as Ethereum. It showed that smart contracts may be subject to an implicit governance mechanism, which also means that it is not entirely autonomous and may be tractable for legal enforcement, even if this may in practice often be complicated to achieve. Furthermore, the experience highlighted the need for a more nuanced set of instructions for dealing with various possibilities that might arise during the execution of a contract.

3 Legal analysis and smart contracts

3.1 Smart contracts as an alternative to traditional contracts

The claim of smart contracts was that they could supplant traditional legal contracts. This claim seems to rest on a several assumptions which are not supported by research into contract practice.[3]

First of all, the possibility of drafting contracts without recourse to legal language or 'legalese' is not new and indeed is accepted in most cases in most jurisdictions. Formal requirements for contracts are relatively rare. The practice of asking legal advice for contracts may be due more to considerations of risk management (imposed by investors, supervising authorities, or accountants) than to actual legal requirements. Indeed, many actual contracts such as a simple sale in a shop are oral, and many others are drafted in fairly informal language. Highly formalized contracts

[2] See [Tjo17] and [Ras16].

[3] See for example the overview and analysis of [Col99].

appear to be more specific to certain legal and commercial cultures rather than a general phenomenon of law and social practice.

Furthermore, smart contracts are claimed to be immutable, therefore immune to what is described as ‘corruption’ by lawyers who would change the meaning of contracts. However, immutability also means that contracts cannot keep up with changing circumstances: the need to do so was a driving social force for allowing post-fact change of contracts (i.e. hardship). Indeed, cases where a contract was later interpreted against the obvious or literal meaning of its wording quite often are based on various reasons having to do with reasonable expectations, fairness, protection of weaker parties, and other interests that appear to find social and political support (as can be deduced from similar statutory provisions adopted by democratically elected governments).

Similarly, enforcement of contracts occurs mostly without legal intervention, if only because it is too costly to litigate small claims. Admittedly enforcement may in certain circumstances be difficult to achieve with and without legal intervention, in particular where parties do not know each other and are geographically far removed from each other. This is indeed a possible failing of the existing system of contract law and legal enforcement.

The major selling point of smart contracts may actually be the converse of doing away with law.[4] It may be useful because it allows enforcement where normal legal enforcement is not practical and/or too costly. For example, in cases where the counterparty is not known or situated far away, litigation may not be a real option. Indeed, one of the arguments for smart contracts appears to be the certainty that one is going to get what was contracted for, i.e. that the contract is going to be performed to the letter.

However, to do so, parties will probably have expectations as to what this performance involves. This covers the behaviour of the contract in cases where the execution does not go fully as planned. Now the hypothesis defended here is that contract law actually is a collection of knowledge and experience about how to deal with various problems and issues that may arise during the life of a contract.

3.2 The role of contract law in contract practice

Every contract law book discusses a variety of what are called ‘doctrines’. Such doctrines are in essences sets of rules that govern a certain issue that crops up during contract execution and has in the past required court intervention, as parties could not resolve the dispute themselves. In the course of centuries of court procedures, lawyers have gained a large body of experiences of possible contract issues and disputes.

Furthermore the outcomes of court procedures have created a network of precedents, supplemented with statutory rules, which provide default rules that show acceptable solutions to such issues (at least, acceptable to lawyers).

These issues and rules do not merely represent predilections of lawyers but, it is submitted, are indicative of real interests of parties, which they attempted to vindicate in a court procedure with the mediation of lawyers.[5] Similarly, in a significant number of cases such rules have been adopted by statute law, on the basis of democratic processes (see section 3.1).

Assuming that this description of contract law is by and large correct, this suggests a renewed investigation of both contract law and smart contracts. Contract parties would normally assume that the contract behaves in a way corresponding to expectations regarding traditional contracts. If the contract does not behave in that way, it would be considered to be defective. A contract provider could leave out certain parts of these expectations, but it would seem that he then should inform contracting parties about such omissions. If the contract provider does not do so, a prospective contracting party might hire a contract auditor to perform such checks. Such an auditor would then need some kind of reference to assess whether the contract will perform as expected.

Hence in the practice of smart contracts there appears to be a need for knowledge about the functions of contract law to safeguard the expectations and interests of parties, and the way in which these contract law rules might or might not be implemented in smart contracts. The present research offers an approach to obtain such knowledge.

We will describe a method of analysis which would both improve understanding of the functions of many rules of contract law, as well as allow a more sophisticated form of smart contracts, by incorporating the experience and knowledge available in contract law regarding the general expectations of the public regarding just outcomes.

3.3 Method of analysis of contract law

The proposed method involves a formalization of the system of contract law, with an eye to possibilities of automation. To do so, it is necessary to analyse the body of contract law in a way that recognises the various interests involved, and therefore needs to look further than the bare rules, instead looking at the back-

[4] See [Ras16] who stresses the point of self-enforcement.

[5] [Smi12] argues that the core of legal research consists of arguments for certain rules.

ground or ratio of the rules. Furthermore the analysis has to look for possibilities of automation. This would involve the following steps.

1. List and examine the various issues ('doctrines) of contract law, in the order of the life of a contract. This follows the usual order of textbooks on contract law: formation, content, and remedies. The detailed doctrines within these general parts have to be discussed. For now it remains an open question whether it is possible to clearly separate the doctrines, or whether these are correlated to some extent.

2. For the various doctrines, examine the relevant rules. These have to be described for now at a fairly abstract level. In order to do this, it may be useful to use principle-oriented descriptions of law (such as the Principles of European Contract Law or the Draft Common Frame of Reference) which abstract to some extent of detail historical rules of specific jurisdictions and opt for a more general, rationalised description of contract law. By and large such rules approximate the law in most jurisdictions, both common law and civil law. As smart contract claim not to be tied to a specific jurisdiction, arguably the precise details and differences between jurisdictions should not matter. It remains for now an open question how to determine the proper level of abstraction.

3. For each issue, examine the relevant rules and identify possible inputs or outputs required.

4. For each issue and/or rule, identify the relevant interests at stake (and possibly the way these are protected by the various rules involved).

5. For each rule assess the possibility of automation. Provisionally we may hypothesise that two options exists.

    (a) A simple evaluative rule using objectively qualified values and simple logical or mathematical operators. This lends itself to direct formulation in a programming language.

    (b) A complex evaluative rule of which the inputs are unclear, may refer to the outside, offline world, while the manner of 'weighing' the inputs is unclear, and furthermore the viewpoint of the other party would usually be required. This may be an open norm requiring balancing of the relative interests of parties. An example is a rule that states that only foreseeable damage can be awarded. It is unclear how to operationalize 'foreseeable', as this may involve various (unspecified) circumstances. Such a rule at present requires human judgement for assessment, and assumes input from both parties. In the future evaluation through a sophisticated AI may be feasible, which however would most likely have to be positioned outside the blockchain as it would require too much processing power to allow evaluation on the blockchain.

6. Finally the analysis can be extended by looking at possibilities for evading the need for applying evaluative rules (in particular complex evaluation, which may be difficult to operationalize). Various options may be available.

    (a) Agree on an overriding rule. This would mean that parties agree on a general rule in case such an issue arises which would preclude the default contract rule.

    (b) Redraw the contract boundaries in such a manner that the issue lies outside the contract. For example leave the determination of what constitutes fitness for purpose outside the contract by requiring parties to formalize this before entering into the contract.

    (c) Rely on third party input. This would mean that the contract uses input from outside, thereby delegating or outsourcing the complex evaluation or proof. Examples are expert advice, trusted third parties.

On the basis of a full assessment of contract law it is possible to formalize those parts that lend themselves to formalization. Some parts may be found to resist formalization: this assesment would identify those parts, compile the arguments why these would resist formalization, and list possible alternatives to formalization.

The end result would be a formalization of contract law (possibly in pseudocode form) that is as far as possible aligned with the expectations from social and business practice. Insofar as aspects have not been formalized, it will be made explicit why this is so, which explicates what parties will lose by entering in a smart contract.

With respect to the result for legal scholarship, such a formalization of contract law would improve understanding of the rules and justification of contract law. Simultaneously a formalized system of contract law would offer an improved basis for examining and criticizing actual legal systems.

3.4 Example: withholding performance

In order to explain the approach described above, an example may be useful. For the sake of brevity the method described above will be followed only in general, as a detailed application would take far too much space and time. The main aspects of the approach will be followed.

Contract law recognises the right to withhold performance in case the other party has not performed its obligations properly. Take as an example the sale of a car, whereby payment is to be made in several interim payments (hereafter called ‘term’). The buyer of the car may withhold payment of a term if the car turns out to have be damaged in an accident before the sale, because of which the car is worth far less than if it had not been damaged. The seller had warranted that the car had suffered no accidents.[6] Assume that the warranty explicitly allows the buyer to withhold performance.[7]

This rule of withholding payment can be justified as it protects the buyer, as he doesn't have to continue making payments for a defective product (pending correct performance by the seller). It would be possible to agree that withholding payment is not allowed, but that would remove this protection for the buyer. It doesn't seem possible to move this outside the contract, as the defect may turn up only some time after start of the contract and receipt of the car.

For this example we can take the formulation from the Draft Common Frame of Reference.[8]

> **III.-3:401: Right to withhold performance of reciprocal obligation**
> (1) A creditor who is to perform a reciprocal obligation at the same time as, or after, the debtor performs has a right to withhold performance of the reciprocal obligation until the debtor has tendered performance or has performed.
> (2) A creditor who is to perform a reciprocal obligation before the debtor performs and who reasonably believes that there will be non-performance by the debtor when the debtors performance becomes due may withhold performance of the reciprocal obligation for as long as the reasonable belief continues. However, the right to withhold performance is lost if the debtor gives an adequate assurance of due performance.
> (3) A creditor who withholds performance in the situation mentioned in paragraph (2) has a duty to give notice of that fact to the debtor as soon as is reasonably practicable and is liable for any loss caused to the debtor by a breach of that duty.
> (4) The performance which may be withheld under this Article is the whole or part of the performance as may be reasonable in the circumstances.

This distinguishes between withholding after breach (non-performance) (1), and withholding in anticipation of breach (2), (3). Furthermore (4) allows for partial withholding.

For now let's concentrate on art. III.-3:401(1). How can we develop this in a smart contract procedure? If we take the example of a car which won't start if no payment is made in time, the structure of the rule for payment enforcement is as follows (Algorithm 1).

Algorithm 1 Payment enforcement

```
procedure ENFORCE($p, term, car$)
    if $p < term$ then                    ▷ Term fully paid?
        InterruptStarter($car$)
    end if
end procedure
```

In order to incorparate withholding payment, this rule has to be modified (Algorithm 2).[9]

Algorithm 2 Payment enforcement extended

```
procedure ENFORCE($p, term, car$)
    if $p < term$ then                    ▷ Term fully paid?
        if WithholdingAllowed($car$) = FALSE then
            InterruptStarter($car$)
        else
            NotifySeller($car, PaymentWithheld$)    ▷ Notification is required
        end if
    end if
end procedure
```

In case the buyer may withhold payment, the seller has to be notified, as follows from a different part of contract law (see art. III.-3:106 and 3:107 DCFR). Incidentaly, this shows that interpreting the law is complicated by the fact that relevant rules may be found in other parts of the law.

[6] We assume a warranty. In some jurisdictions no warranty is required as this would be covered by an implicit duty of disclosure, while other (common law) jurisdictions do not recognise a general duty to disclose.

[7] In some jurisdictions a warranty requires that contracting parties make explicit which remedies can be invoked in case of violation of the warranty.

[8] This specific formulation is taken from the Draft Common Frame of Reference [DCFR], which is not a rule of positive law as such, but rather is based on the general common denominator of European jurisdictions.

[9] This only offers the bare bones of incorporating withholding performance.

The full rule for withholding performance after breach of contract is rather complicated, as it depends on whether there are reciprocal obligations. For our case we can assume reciprocity between term payment and delivery of a proper car. If we concentrate on developing the single rule for withholding performance for this contract only, the structure is as follows (Algorithm 3).

Algorithm 3 Withholding performance

```
procedure WithholdingAllowed($car$)
    if $NonPerformance$(car) = TRUE then
        return TRUE
    else
        return FALSE
    end if
end procedure
```

This simple algorithm is a close approximation of what the rule says: if the other party has not performed, you may withhold payment. However, this simplicity is only apparent. We have evaded the question as to how to determine whether the obligation to provide a car has been performed correctly, by putting this in a new procedure (still to be defined). In case of the damaged car (as explained above) this would require some additional verification by a trusted third party. Hence it appears that in this case either one has to assume the truth of a claim by the buyer about the damage, or one has to rely on an input by a third party as to the claim of the buyer.The deeper issue is that this requires an extensive judgement as to all possibilities of non-performance, which seems to require an external off-line input.

This very brief exercise shows that

- implementing a general rule for withholding performance is complicated by the need to determine reciprocal obligations,
- it is feasible to simplify the general rule by only allowing specific cases,
- the rule relies in the end on judgement by third parties of off-line events, which moves the computational aspect to the real world and lessens the claim of a self-contained environment,
- the rule can be contracted away, which however removes protection of the buyer,
- the rule cannot be considered apart from the broader issue of enforcement in case of non-performance, and several issues may have to be implemented not in the function of withholding, but rather in the environment calling this function (see the modification to *Enforce()*),
- real-world implementation requires implementation of additional rules about (for example) notification, which may be found elsewhere in the law and not in the part of the law about withholding.

3.5 A more extensive example

A more extensive example would have to incorporate significantly more parts of contract law. To show how complicated matters may become we can attempt to implement the full section on Withholding payment. A simplified implementation is the following Algorithm 4, which is provided without detailed comment for the sake of brevity.[10]

Algorithm 4 Withholding performance full

```
procedure WithholdingAllowed($car$)    ▷ First part is (1), after breach
    if BuyerComplaint($car, breach$) = TRUE then
        if $NonPerformance(car)$ = TRUE then
            return ReasonableAmount($term, Facts$)
        else
            NotifyBuyer($CorrectPerformance$);
            return 0
        end if
    else    ▷ Now anticipatory breach (2)
        if BuyerAnticipatoryComplaint($car, Facts$) = TRUE then
            if Reasonable($car, Facts$)=TRUE then
                return ReasonableAmount($term, Facts$)
            else
                NotifyBuyer($UnreasonableWithholding$);
                return 0
            end if
        else
            NotifyBuyer($NoNotification$)
            return 0
        end if
        return 0
    end if
end procedure
```

Several comments are in order.

Instead of returning TRUE or FALSE, this procedure now returns the proportion to which withholding performance is allowed. The procedure on enforcement of performance would have to be modified to incorporate this change. Note that in this way the buyer is not informed whether the enforcement is due to an incorrect claim of breach, due to an unreasonable amount of withholding payment, or simply due to non-payment of a term. In normal contract practice the seller, when invoking a remedy, would be expected to make clear

[10] The passing of parameters is not consistent, as this is only an example analysis for which these details are not decisive.

in what way he disputes the standpoint of the buyer. This dialogical process appears to be exceedingly difficult to program into the procedural form of a smart contract.

This procedure only checks for Nonperformance if no full payment is received. This is at odds with actual contract practice and expectations: normally the buyer would complain beforehand, and only then withhold payment. The procedure could be formulated in various other ways and with a different order of checks, but that would complicate this significantly (hence is not done for this example).

The procedure does not take into account that checking for nonperformance may take time if third parties are involved. In order to take this into account, it would be required to invoke a period after the appeal to withholding, during which withholding can be done legitimately, until the check of Nonperformance has been completed. This would then have to be refined for the case that the check of Nonperformance is not possible due to uncooperative behaviour of the buyer. Furthermore one would expect to skip the check of Nonperformance in case the seller acknowledged the Nonperformance.

Note that the notification of the breach to the seller is now incorparated as a requirement prior to withholding payment.

Note also that this procedure does not yet take into account the possibility that the seller has ‘tendered’ performance (as stated in the DCFR). This would usually occur some time after first withholding payment, hence would require periodic checks of whether it is still allowed to withhold payment. This does not sit easy with the simple assumption that one only has to check at the time agreed on whether the payment has been made.

The algorithm has several unresolved issues:

1. it is unclear how to submit the facts of the breach or anticipated breach: this may effectively require a complete description of the real-world situation,

2. it is unclear how to determine whether withholding on anticipatory breach is reasonable,

3. it is unclear how to determine which proportion of the term payment is reasonable to withhold.

3.6 Further remarks

This tiny part of contract law may already seem to be a mess. One might argue that it is not useful and indeed not necessary to fully capture the legal rules in their current form. It might be possible to simplify the rules while still keeping the essence of the protection of the interest concerned.

Admittedly the current structure of law may not be the most elegant formulation of the essence of these rules. Nonetheless it does not follow that these rules can be rigourously deleted or simplified. The law has several aspects, which are not always recognised, which may explain the complicated structure of legal rules and may prove an obstacle to formalization.

Many contract law rules proscribe procedural steps involving interaction, communication between parties. This complicates formulation in the form of purely functional programs. On the other hand, the legal rules do not have a fixed chronological structure such as a procedural language would imply. Legal rules apply when applicable, not when they are called at the specific moment in a procedural script.

This seemingly odd structure may be explained by a dual aspect of legal rules. On the one hand, rules simply tell the court how to decide the case post-facto, given the facts about how parties have acted in the past. On the other hand, rules have an expressive aspect in that they allow parties to deduce how they should act in certain situations. A smart contract would fit the latter approach. Incidentally, this differs from attempts at automated adjudication which takes the former approach.[11] Hence those earlier attemps have only limited relevance for smart contracts.

However, some parts of contract law assume that there is sufficient time to determine the truth of the matter, such as whether a product is defective and whether an obligation has been performed correctly. In a smart contract environment it is hard to incorporate this time-consuming process of determining the actual state of affairs. An interim solution would be required, or some other alternative that is acceptable to parties.

An alternative that is used at present with online trading platforms and intermediaries such as eBay and PayPal is that in case of a dispute usually a simple default rule is chosen, and parties are refered to the standard judicial process to investigate the details of their case. This option is not available to ‘pure’ smart contracts if these assert that they are outside the law and do not require legal support or intervention for proper operation or enforcement.

Incidentally, the fact that legal rules often assume that the moment of evaluation is the court procedure allows legal rules to use complex evaluative terms, ‘open norms’, by which complicated issues do not have to be spelled out or resolved fully at the moment of contracting, but instead can be defered to the later judgement by the court.[12] In this respect legal rules can actually be simpler than explicated antecedent rules.

[11] About which see for example [Tho16] with many references.

[12] One may compare this to what is called ‘lazy evaluation’ in compilers.

These issues explain part of the complicated structure of contract law, and are not evidently easy to resolve. Programmers of smart contract might wish to evade these issues or simplify the rules dealing with these issues. Users of smart contracts will then have to accept the limitations of smart contracts in providing everything that contract law can provide regarding the protection of party interests.

4 Conclusions

Smart contracts offer an opportunity for a novel re-investigation of contract law, by explicating the reasoning behind contract law rules and by elucidating the precise structure of contract rules by way of formalization in pseudocode. An analysis along the lines sketched in this paper would provide improved opportunities for realising smart contracts while optimally safeguarding party interests.

Acknowledgements

I am grateful to Hans Weigand and Joris Hulstijn (TiSEM, Tilburg University) for pointing me to sources on the early history of smart contracts.

References

[Col99] H. Collins. *Regulating Contracts.* Oxford, Oxford University Press 1999.

[DCFR] Chr. von Bar and E. Clive. *Draft Common Frame of Reference: Principles, Definitions and Model Rules of European Private Law.* Oxford, Oxford University Press 2010.

[Pey00] S. Peyton Jones, J.-M. Eber and J. Seward. Composing Contracts: An Adventure in Financial Engineering. In *ACM SIGPLAN International Conference on Function Programming* (ICFP) 2000, 280-292.

[Ras16] M. Raskin. The Law of Smart Contracts. Available at: https://ssrn.com/abstract=2842258, 22 September 2016 (last revision 13 December 2016).

[Smi12] J. Smits. *The Mind and Method of the Legal Academic.* Cheltenham, Elgar 2012.

[Sza94] N. Szabo. The Idea of Smart Contracts. Available at http://szabo.best.vwh.net/smart.contracts.html. Year of publication is said to be 1994, but I have been unable to verify this. See also http://www.alamut.com/subj/economics/nick_szabo/smartContracts.html mentioning Extropy nr. 16, 1996.

[Tho16] J. Thornton. Cost, Accuracy, and Subjective Fairness in Legal Information Technology: A Response to Technological Due Process Critics. *New York University Law Review* vol. 91 (2016), 1821-1850.

[Tjo17] T.F.E. Tjong Tjin Tai. Smart contracts en het recht. *Nederlands Juristenblad* 2017/146, 176-182.