Formalising law, or, the return of the Golem

Formalising law, or, the return of the Golem

Burkhard Schafer

Abstract: "Good old fashioned" AI, developed first in the 1980s but still an approach used in many contemporary legal apps and law chatbots, is often seen as less likely to create a dangerous "black box society" than machine learning based approaches. The chapter queries this notion by looking at the way i n which the very process of formalising the law rests on normative decisions and value commitments that can't simply be left to software developers. Using the literary figure of the Golem, it traces some of the normative decisions that any legal technology has to make, and posits some desiderata for an ethically responsible theory of legal formalisation

Keywords: Legal tech; AI and Law, Formal logic, legal reasoning, AI ethics

1. Introduction

Artificial Intelligence applications for law, for many decades (perceived as) a niche pursuit of academic researchers with a dearth of commercial success stories, have recently begun to capture the public imagination. Riding sometimes on the coat-tails of headline grabbing advances of AI in other fields, such as DeepMinds victory over the South Korean Go master Lee Sedol in the game of Go, the possibility of a "robo-judge" seems for many an inevitable and desirable future application of the technology (see e.g. Addady, 2016 ; Mills, 2016)[1].

Some of the headline grabbing applications promise significant improvements for the administration of justice, in particular improved access to justice. DoNotPay for instance, a

[1] See Law Society of England and Wales. (2018). Artificial Intelligence and the Legal Profession. Retrieved from https://www.lawsociety.org.uk/policy-campaigns/articles/artificial-intelligence-and-the-legal-profession-horizon-scanning-report.

legal chatbot, helped to contest 160,000 parking tickets in London and New York for free[2] (Sparkes, 2023, p. 8). If it were possible to "scale" this type of application across a broader range of legal disputes, we could see people historically excluded from professional legal advice on grounds of cost, complexity or other socio-cultural barriers being able to enforce their rights much more systematically.

At the same time, there is persistent concern that "robo-judges" could harm the justice system, not just by amplifying systematic biases and prejudices, but through an illegitimate power grab by software developers usurping the role of legislators and judges. AI developers create de-facto new laws without the accountability and contestability that the traditional process provides, and potentially also "design out" the human, empathetic element in legal decision (for an overview see Hildebrandt, 2015). As the judges in US case of Keppel v. BaRoss Builders put it:

"*Above all, it showed that a judge is a human being, not the type of unfeeling robot some would expect the judge to be*."[3]

A recent edited collection by Deakin and Markou (2022) brought this unease to the point of asking the question: Is Law computational? This chapter will contribute to this discussion through the lens of formalization of the law. It discusses if, and if so to what degree, law is amenable to formalisation of the type found in a broad range of legal technologies, or legal AIs. It will argue that while the question is much older than current interest in legal technology, and in many ways as old as law itself, it is nonetheless misleading.

Legal AI or computational law came historically both with a promise, and a conception of justice: If law could be applied with the cold rationality of a machine, following nothing but the logic of the rules, justice would be enhanced by reducing arbitrary, untransparent and discriminatory decisions. It would enhance justice by eliminating the impact of human biases, limitations of our memory, our short attention span or often failing reasoning capacity.

[2] https://donotpay.com/

[3] Keppel v. BaRoss Builders, Inc., 509 A.2d 51, 56

This vision of law as inherently computational, and that of the ideal judge as an automaton, a "mere voice of the law" predates computers by several centuries. The Enlightenment had developed the idea of a clockwork universe, governed by strict and mechanical rules that guarantee predictability and with that ultimately also human control. With the new mechanical world view came also a new capability to build automata. Technical skills merged with philosophical reflection in the work of Rene Descartes. Descartes (in)famously suggested that (non-human) animals are nothing more than complex machines. Thus, mechanism became the standard prism to see nature and organisms.

No-one more though took the automata model and applied it to humans than Julien Offrey de La Mettrie in his *L'homme machine*. Mettrie had previously advanced a proto-evolutionist argument that saw humans and animals as closely related, the latter just using a somewhat more complex mechanism. That mechanical argument took centre stage in *L'homme machine* (Mettrie, 1748/1994).

Crucially for our discussion, La Mettrie explicitly mentions the human ability to make ethical and legal judgements. In his account, legal reasoning that applies rules to facts is ultimately not different from any other attempt to reason about the world: "To be a machine, to feel, think, know good from evil like blue from yellow" (Mettrie, 1748/1994, p. 71). Colour recognition and moral discernment are equally within the capacity of deterministic machines, both are nothing but mechanical responses to material inputs. Again La Mettrie: "Even if man alone had received a share of natural law, would he be any less a machine for that? A few more wheels, a few more springs than in the most perfect animals." (Mettrie, 1748, p. 72; See also Campbell, 1970 and, for an application to law, Thomson, 2016)

It is important to note that La Mettries's vision of the ideal judge was born out of a normative agenda: with incompetent, corrupt, biased and cruel judges a lived experience for him and his contemporaries, the idea of a mechanistic application of the law becomes liberating. By putting ourselves under the law, our lives become plannable and predictable, just as the laws of nature allow planning and prediction. This mechanistic vision of the law then dominated legal theory especially in continental Europe during the 19th and early 20th century. "Mechanical jurisprudence", legal formalism and the conception of the judge as a mere passive mouth of the legislator became the dominant legal idea.

A century later, and again the confluence of technology with a specific vision of justice promises if not an end, then at least a much-reduced role, for human lawyers. (Susskind, 2008). After initial enthusiasm in the legal expert systems of the 1980s was interrupted by a short "AI winter", we are seeing a resurgence of interest in "law tech", the idea to assist, or maybe even to replace, lawyers and judges by automata (see e.g. Sourdin, 2018 ; Ulenaers, 2020).

This chapter will not focus so much on the *capabilities* of these systems, or their capabilities relative to a given body of legal knowledge. A comprehensive overview of their recent history can be found in Bench-Capon et al (2012). Rather, it looks at an often neglected aspect of developing (legal) AI, the early steps of the development cycle where decisions are made which aspects of a law to formalise, which language to choose for representation, how to document and justify these decisions and also how to document any risks that these design choices can create. This links the technical discussion on formalising and computerising law with the emerging discussion on the ethical and regulatory aspects of AI in general and legal AI in particular, such as the AI4People principles on responsible legal technology that the author helped to draft (Schafer *et al*., 2020).

In particular, it aims to sensitise the reader to the human element, and the normatively salient decisions, that often invisibly feed into the development of legal technology. The idea of the neutral, logical, mechanistic AI judge fails, or needs to be treated with caution, because it can hide the all too human aspects that went into its design. Currently, under the label of "black box society", this problem is discussed in the machine learning environment as a problem of the algorithm and its training data itself. By contrast, this paper will look into the design process, and a different type of "black box" – not impenetrable algorithms, but design decisions taken behind closed doors, when laws are formalised.

In Section 1 of the paper, some key technical terminology is introduced, using a small number of case studies as illustrative examples. In particular, we will look at recent attempts to formulate road traffic laws in a way that makes automated vehicles (AVs) law compliant by design. The dual aim of that section is to introduce some basic concepts of legal AI and legal formalisation, but also to sensitise the reader to the normative decisions that the program developer has to make.

In the second part of the paper, we will go back in time, to the oldest reflections on the mechanical nature of laws and rule following. In particular, we will use the ancient myth of the Golem to illustrate and elucidate some of the issues that we face when thinking of the law as a computational artefact.

In the final part, we return to the Golem, and ask what normative conclusions we can draw from this discussion, and how we can start developing a more responsible approach to legal AI and legal formalisation.

The Golem is chosen as a lens partly because the myth was always also one of law. We learn about the original Golem in the "Cases and Materials" of rabbinic law, the Talmud, its reported maker was an accomplished lawyer and law reformer, Abba Ben Rav Hamma. Moreover, in the subsequent Golem stories, we also see many of the tropes that still inform the discussion on autonomous systems and legal technology today. The Golem was built to abide by commands and to follow rules, and therefore would have required some internal representation of these rules that "make sense in Golem" and are executable by it. Today, autonomous systems such as self-driving cars have become a new "audience" for legal rules, and we encounter again the idea to program legal rules directly into their governing algorithms.

This will allow us to make a number of interrelated arguments:

1. The question whether law "can" be formalised is misleading and in its simplicity potentially dangerous. Rather, we should ask: for a given normative conception of justice and a vision of a good legal system, how does a specific formal language and a specific approach to formalisation enhance or hinder achieving this vision in a specific intended application?
2. "Formalising" law is a process of translation, not dissimilar from any other translation between natural languages. This means among other things that every formalisation of the law is also an interpretation that will at best be "faithful to a degree" to the original - "traduttore tradittore", the translator is also always a traitor. But while translation studies have developed systematic, detailed and comprehensive rules on how to translate best, there is comparatively little research done about the process of formalisation. One attempt at such a systematic theory of formalisation was made by Georg Brun (2004), but it too was mainly a discussion of problems rather than a systematic attempt at resolving them.

3. As a consequence, the "translator", i.e., the programmer who formalises law, inevitably has to make choices. Some of these choices will have consequences that are normatively salient and affect either individual citizens or (our perception of) the functioning of the legal system as a whole. This means the question of appropriate formalisation and the ethics of legal technology are intertwined. In this respect too, legal formalisation can learn from translation studies, where discussions about the ethics of translation have become a mainstay of the meta-methodological debate since the 1980s (see e.g. Chesterman, 1997; Baker & Maier, 2011).
4. One emerging candidate that seems to sidestep the problems associated with legal formalisation, "law as code", i.e., the proposal to enact machine executable versions of legislation through the normal parliamentary process, mitigates some of the problems, but also creates new ones.

A short primer in legal formalisation

While most of the current interest in, and concerns about, legal AI centre on machine learning approaches (ML), this chapter focusses on examples of "good old-fashioned AI" (GOFAI), the symbolic-manipulation paradigm that rose to prominence in the 1980s, but still is the legal knowledge representation method behind a number of the most high profile and popular law tech apps these days.

The reasons for this choice are twofold. The first is that while ML based approaches to legal technology have received most of the public attention, a significant number of systems actually in use have at their core a GOFAI representation of legal knowledge or use GOFAI as "guardrails" to enforce law compliant behaviour of the underlying ML system. Secondly, there is a widespread misconception that GOFAI systems are less risky because of their inherent transparency, or do not pose the same regulatory and ethical challenges, as data driven approaches. One aim of this paper is to argue that this perception is mistaken, and that at key stages of the development also of GOFAI legal technology, design decisions are taken which can adversely affect citizens and their rights, and which are shielded in the design process from public scrutiny and legal contestability just as much as the "black box" systems of ML.

In the first step, we are now introducing some key concepts and ideas that are needed as background to follow the discussion. Despite differences in detail that we will discuss below, a GOFAI will have

A) a formal language into which the knowledge expressed in natural language has to be translated.
B) A set of inference or rewriting rules that tell us how to derive an output, the "inference engine".

A formal language means an explicitly defined alphabet together with "grammar rules" that tell us how to combine symbols from the alphabet into longer "well formed" expressions. The alphabet contains a set of symbols for logical constants (examples are "If-then" or "or") which have a fixed, explicit meaning and a set of non-logical symbols such as "F" or "p" , parameters for external objects and properties such as "driving fast" or "Peter". The grammar, or formation rules, then tell us that we can combine for instance the symbol c with the symbol F to form F(c) as a sentence in that language, here with the intended meaning, or interpretation, "Peter is driving fast". We will write such an intended interpretation as:

F: driving fast; p: Peter.

This interpretation depends entirely on the context, and F(p) could also stand for "2 is even" or "Cyanide is healthy". Some symbols however keep their meaning across all contexts, the logical *constants* such as → ("if -then") or ∀ ("For all.."). Their meaning is defined explicitly, for the "If-Then" through a truth table that tells us for all possible combinations how the truth value of the complex "If-Then" sentence can be derived from the truth value of its component parts – a sentence of the form "IF A then B" is true if and only if A is not false while B is true, regardless of the content of A and B.

For a logic-based AI, only these constants have meaning, so that we can think of First Order Predicate Logic (FOPL) also as the theory of the meaning of the words "all", "none" "if then" and "or". This is important to remember as an antidote to the "Eliza effect": a legal chatbot may give us the impression that it "understands" legal concepts such as property, crime, intent, or contract, by using these terms correctly in its answers. In reality, the AI "understands" just the logical constants and treats the non-logical terms as meaningless strings of symbols.

The choice of formal language prejudges how much of the natural language sentence we can analyse, and how much we have to leave unanalysed in the parameters. If our language is FOPL and the sentence we are interested in is "Peter *should* drive fast", there is no direct way to express the meaning of "should" and we then have to assign the sentence the same structure as above, but with a different intended interpretation:

F(p) F: should drive fast; p:Peter

However, logicians soon discovered that terms such as "should" or "must not" have similar invariant properties like the sentence connective "If then" or the quantifier "For all", an invariant meaning that can be made explicit and formally captured. So if in a given application, it is desirable to "unpack" the meaning of the *deontic operators* such as "should", it may be better to choose a richer language, a language of deontic logic, where additional symbols for the new logical constants like "should" or "is prohibited from…" are introduced and given a fixed and explicit meaning.

Similar decisions can be made for terms like "after" and "before", leading to temporal logics (Mackaay, 1990), "necessarily X" and "possibly X", leading to alethic modal logics, and "believes that X" and "doubts that X" leading to epistemic modal logics (see for an overview Prakken & Sartor, 2002). These, crucially, are methodological choices that are not "right" or "wrong" as such, but rather "useful" or "not useful" for a given intended application (Gabbay, 1992).

This gives us a first hint of the creative choices that are available to the programmer. As all choices, they can carry normative consequences and ethical or legal obligations: who makes the choice, on what authority, how can they be justified if contested, and how do we know they were "good" choices? These questions are the ultimate focus of this paper, as an often overlooked but crucial aspect in the question of how we should regulate AI in general and legal technology in particular.

While in the example above, the different languages had different expressive power, typically "adding to" classical logic, some formal languages that have been designed for legal users specifically are "doing the same thing" as a general-purpose language such as PROLOG.

Examples include PROLEG (PROlog-based LEGal reasoning support system, Satoh et al 2010) or Catala (Merigoux, Chataing & Protzenko, 2021).

Their aim is to make the process of formalisation, and also the checking for correctness, easier and more intuitive for lawyers who may be lacking the time, skill or experience to learn a "multipurpose" language. Here too the choice can be normatively salient, for instance to meet transparency duties of the developers: can they design the system in such a way that a domain expert who is not also a computer scientist can check the way the system is working? In that case we can ask if there should be a legal duty to use of all the available and equivalent formal languages ones that are most intuitive? This would allow a greater number of citizens, with little or no formal training, to scrutinize the formalism and to contest, if appropriate, its adequacy.

In addition to having a formal language that regiments how expressions are formed and knowledge is represented, an AI also needs a way to "do things" with these sentences. That is the role of the inference engine that prescribes how one string of symbols (the input) can be rewritten as another string of symbols (the output). Intuitively, in legal reasoning contexts we often think of this as an argument, were we move from a set of premises (input) to the conclusion (output), but the output can also be an answer to a question, or an instruction to a machine to perform an action (e.g., to lower the speed of the car in response to a sensor input).

A *legal* GOFAI then, following an influential definition by Trevor Bench-Capon, is an AI where some or all of the formulas that represent legal knowledge computationally are isomorphic to a corresponding legal normal in natural language. That means that their syntactic structures correspond to each other (Bench-Capon & Coenen, 1992). Let us illustrate this idea by looking at the legal norm 152 from the UK Highway Code:[4]

"You should drive slowly and carefully on streets where there are likely to be pedestrians, cyclists and parked cars"

To give a formal account of this sentence, we would first be rephrased as a rule: If there are pedestrians, cyclists and parked cars , then the driver must drive slowly and carefully"

[4] https://highwaycode.org.uk/rule-152/

We can then formalise this sentence in FOPL as

$$\forall x\, D(x) \wedge \exists y \left(P(y) \vee C(y) \vee PC\,(y)\right) \rightarrow SDS\,(x)$$

with D: drives, P: is a pedestrian; C: is a cyclist; PC: is a parked car; DS: should drive slowly "Read out" this formula now states roughly: For everyone, it holds that if they are driving, and there are pedestrians, cyclists or parked vehicles, then they must drive slowly.

This seems a good approximation of the legal norm that we try to model, at least at first sight.

If we now have another sentence, one that says that Peter was indeed driving and there were pedestrians…,

$$D(p) \wedge \exists y \left(P(y) \vee C(y) \vee PC\,(y)\right)$$

We can derive that he had to drive slowly, by the inference rule of modus ponens

$$SDS(p)$$

If we now add another rule that describes the consequences for not driving slowly, e.g. a fine, and also tell the system in the same formal language that he was in fact driving fast, the AI can infer that he is now liable for this fine, and print out a fixed penalty notice. This is how in a nutshell a GOFAI legal AI works. This approach to legal AI rose to prominence in the 1990s onwards, but it is still today the way in which many legal apps and chatbots reason about the law.

As indicated above, in this formalisation we leave a crucial aspect of the "norm-likeness" of the rule implicit and hidden in the "SDS" part, it is not visible, from the systems perspective, that we are dealing here with a norm that directs behaviour, an "ought" rather than a description. So as formalised, the system could not infer automatically that from $SDS(p)$, which we interpreted as "the driver *should* drive slowly", we can also infer that it is therefore *prohibited not* to drive slowly, or with other words to drive fast. In some applications, we may want the program to perform this inference automatically, and one possibility is to use a richer logic, deontic modal logic, where the above formula would appear as

$$\forall x\, D(x) \wedge \exists y \left(P(y) \vee C(y) \vee PC\,(y)\right) \rightarrow \blacksquare SDS\,(x)$$

The ■ means here "ought to", an operator that applies to the sentence following it and modifies it. Just as it was possible to define formally and for all contexts the meaning of the IF THEN arrow through a truth table, the meaning of "ought" can be formally defined (for an overview see Meyer, 1993).

Whether the AI developer opts for this more expressive language or leaves the "ought" operator un-analysed and hidden in the "S" parameters will often be a question of convenience, driven by the needs of the application. It makes more sense for instance in a system that assists judges than one that regulates the driving of an AV. Importantly though, hidden behind these technical considerations is another deep jurisprudential issue: what, really, are the norms that we formalise?

In the influential Austinian understanding of norms, they are commands directed from the sovereign to the citizen, backed by sanctions (Austin, 1880). Here, their nature as "command" is essential, and we may want to represent it. In a very different approach, popularised by Herbert Hart (2012) in a common law environment, but also suggested by Karl Binding (1872) for the civilian tradition, legal rules often look descriptive for a reason. The law of homicide in many jurisdictions does not say: "Thou shallst not kill", it says "Someone who intentionally kills another human being without justification is a murderer", possibly with another sentence of the form "The punishment for murder is a prison sentence ranging from 3 years to life".

These sentences read on their surface like ordinary statements of facts, though we intuitively understand them as also implying that killing is wrong. For Binding and Hart (and Mettrie above), the main audience for legal norms are not citizens, but judges and other legal officials. Depending on the audience, just as with any translation, different formalisations can be more or less appropriate. This means that the decision of the AI developer, even if driven mainly by technical considerations, cannot but take sides in complex jurisprudential questions, every legal technology inevitably is aligned with some conceptions of justice and the law, and silent on others.

Traduttore traditore – the pitfalls of formalisation

While the above formalisation of a simple sentence may look straightforward, it is anything but – and in some sense cut a number of corners already. In what follows, I will disclose some of the "cheats" and discuss why they matter.

Cheat 1 „and", „or" and the issue of legislative intend

First, the natural language norm said "pedestrians, cyclists ***and*** parked cars" However, our reformulation changed this into a nonexclusive "or" ("v"). For the legally trained reader, the reason is obvious: the norm aims to protect vulnerable traffic participants such as cyclists. If we had used the "and", the rule would only "fire", i.e., allow us to infer the desired action, if there were *simultaneously* pedestrians, cyclists and parked cars on the road. Our Peter could speed at his heart's content around children playing in the street, as long as none of them is on a bike. The reformulation seems therefore to be perfectly adequate in the light of our background knowledge about traffic, UK legislators, their attitude to pedestrians and other such factors.

Yet still we should ask who, and with what authority, should make this decision during the development process, especially when the end-product were to be used to generate speeding fines in a semi-automated way. After all, we have chosen here a reading that is more burdensome on the driver. If the intended application is one of criminal law, issuing a speeding fine, the "in dubio pro reo" rule normally requires us to choose between different possible interpretations the one most advantageous to the accused. Do we need to find a legal precedent that authoritatively supports the way we formalised the sentence? Can any programmer make that decision, or does it need to be "signed off" by someone licensed to practice law? How should that design choice be documented, giving for instance the transparency duties in the proposed EU AI Act?

Conversely, if the intended application is as a "guardrail" that controls the AIs on an autonomous vehicle, where the outcome is merely to lower the cruising speed, the "in dubio" meta-rule is irrelevant. At worst the car drives in that case safer than the legislator required. We can see here a recurrent theme of this chapter: to determine if a formalisation of the law is adequate requires knowledge of the intended deployment of the AI, and can't be decided in isolation.

Cheat 2 From probabilities to facts

A second cheat was our omission of the word "likely" in "where there are *likely* to be pedestrians, cyclists and parked cars". As formalised, the rule only applies when there were actual pedestrians present. To formalise the idea that something was not the case, but "could have been the case" requires either a "calculus of probabilities (Robertson & Vignaux, 1993), or "alethic" additions to our language that express the idea that something was necessary, possible, impossible or likely to happen (see generally Hughes & Cresswell, 1996, historically for the relation between these and legal reasoning Lenzen, 2005)

Cheat 3 A world without pedestrians – what a thought

The final "cheat" happened when we translated the reformulated natural language sentence into a formal one. Our formal version of the rule is much more demanding than the natural language version in another aspect. Read literally, everyone has to drive slowly, always. That's because the condition $\exists y \left(P(y) \vee C(y) \vee PC\ (y)\right)$ is already fulfilled if there is a *single* pedestrian, *somewhere* on this planet. The "$\exists$" only means "there is at least one", and nothing in the rule as formulated expresses the idea that this one person needs to be anywhere nearby. Intuitively, we understand under which conditions the legal rule matters: the diver is in an inherently dangerous environment, with sufficient risks in close proximity to require extra care. The law however does not specify just how many pedestrians or cyclists are needed to constitute such a risk, or how close by they have to be, this is left implicit and relies for interpretation on the intuitive background knowledge and common sense of the norm addressee.

As written, the formalisation is therefore plain wrong. And yet, for many possible applications of a legal AI, it would still function perfectly. If the system were to be used for instance to assist a court in issuing speeding fines, ensuring correct operation is easy: The system requires input from the operators. They act in modern parlance as an "Oracle", the interface between the world and the algorithm. Remember that we said above the AI only understands the meaning of the logical constants. The rest is "invisible" to the systema and requires input from the user. In our case, that is the fact that indeed, Peter was driving and there were pedestrians and cyclists in the vicinity.

Now, no competent human operator would give the system as input "there are pedestrians.. " when the event in question happened at 4am on an empty countryside road, and then justify

this by pointing out that there was a pedestrian, but 500 miles away and 6 hours earlier. The competent human operator knows this is not what is meant, and in this way compensates for the shortcomings of the formalism.

But what about an incompetent operator who just "ticks boxes" unthinkingly and as a result tells the AI that yes, there are pedestrians (somewhere in this world)? To be able to contest the fine the system would generate as a result, the driver could request an explanation, but who exactly in our scenario owes the explanation, and can any single person satisfy the driver's query? The developer can point to the fact that while the rule was not a perfect translation of the legal text, it was a sufficiently adequate one, assuming a modicum of common sense by the operator. The operator can point to the fact that the data they inputted was literally true. The system itself could generate the formal proof, and demonstrate in this way that it was working correctly – the advantage of GOFAI is that from the machine perspective, it is highly transparent and interpretable precisely because it uses isomorphic, symbolic representations of the rules that inform its decisions. All the problematic mistakes and choices have been made long before it carried out that operation, or rather, many decisions by different actors, decisions that were individually not "that wrong", caused in their interaction the wrong outcome.

The situation becomes more complicated if the intended use is not as a legal decision-making system, but as a way to control an automated vehicle. In that case, there is no longer a place for human involvement, rather the car uses as input readings from its sensors. But AVs lack a human operator's common sense and background knowledge. So for this purpose, the above formalisation would indeed be inadequate. Rather, the rule would have to rephrased by something like:

"If you are in driving mode, and your speed is X and your sensors identify a pedestrian within n_1 meter distance, or a cyclist within n_2 meters distance, or a parked car within n_2 meters distance, then slow down the speed to Y"

The values for $n_1 \ldots n_3$, how many pedestrians, how far away etc, would need to be determined by *someone*, but they can't be read directly from the statute. Several new choices become available at this point:

The values could come from the law in a different way, for instance past court decisions in speed driving cases. The task then becomes to extract from the decided legal cases those features and aspects that disambiguate the meaning of the statute (Borges *et al*., 2023). This way, a considerably richer and more fine-grained model of the operating law is possible than using the governing statute only.

However, in this approach legal knowledge is still understood as a set of law-like rules, the use of case law for disambiguation happens "behind closed doors" by the development team during the process of reformulating the law prior to formalising it. The resulting formalised rules do not carry any indices that make the cases from which they originate visible and explicit. Contesting their correctness now requires to consult material external to the software and the statute. This could be any documentation that describes the choices that the developers made in selecting and analysing the cases. Or the challenger needs to carry out their own legal analysis of the case law and develop an alternative formalisation from scratch

A different approach is to make the reasoning with precedents an explicit part of the formal representation, shift it from the preparatory stage of building a legal AI to the formal system itself. This is the approach taken in case-based reasoning systems, which emerged contemporaneously with rule-based systems from the 1980s onwards (see e.g. Ashley, 1992 , 2002). While rule-based systems such as the TAXMAN (McCarty, 1976), Divorce Advisor (Duguid, Edwards & Kingston, 2001) or ADVOCATE (Schafer & Bromby, 2005) focus on the application of an abstract legal rule to the facts of a case, case-based reasoning systems for legal applications such as CATO (Rissland, Ashley & Branting, 2005) or IBP (Brüninghaus & Ashley, 2003) analyse the way in which a past decision is used in a process of analogical reasoning to guide the decision in the new case.

For this task, there has to be a way to formalise not just legal rules, but also court decisions. In a legal CBR system, cases are formally represented through a more complex structure that contains the names of the parties, the outcome or disposition of by the judge, and a range of "factors" that describe the fact situation that yielded the outcome (see for an overview the papers in Atkinson, 2009; for an application to statutory interpretation see Araszkiewicz, 2013). "Factors" are factual aspects of a case that are pertinent for the decision to some degree or other, and are likely to appear across a range of similar cases. In our example, relevant factors could be "number of pedestrians" and "closeness to the car", but also whether or not it

was raining on the day, the road conditions, or whether the driver had reasons to believe that the pedestrian had seen them. Not a relevant factor would be, arguably, the hair colour of the driver, their hometown or their gender (for the importance for legal justification see Atkinson, Bench-Capon & Bollegala, 2020). So while the driver in our case may well have been a male redhead from Edinburgh, this would not be formally represented in the case structure; that there was a slight drizzle on the day, but visibility otherwise good, might be. The precedent case (PC) and the current case (CC) are then both formally represented in such a case template. The reasoner then calculates if the overlap between the factors of each case is strong enough, quantitatively and qualitatively to transfer the decision reached in the PC to the CC.

CBR is still using symbolic representations of "the law", but it differs from rule-based systems in its understanding of what "the" law is. In some ways, it is already closer to data driven approaches that use machine learning, and also shares some of the problems with them. For instance, which factors to represent in a case as important, and which one to disregard and render invisible inevitably reflects the views of the formaliser. This constitutes a very similar inroad for biases as those found in ML.

Rule based systems have another advantage. They make it easy to ascertain if the AI is correct at least in this sense: We can determine if all the rules in the AI are authoritative. For this, we simply link the formal rule to its corresponding natural language rule (regardless of whether we think it is *the best* way to represent that rule). Second, we can also determine if the knowledge base is complete. As there is a finite number of rules in a given statute, we can check if each of them has a formal counterpart.

With CBR, we can still check if each of the cases in the knowledge base is authoritative, that is it has been decided by a competent court and not overruled by a higher court. But it becomes normally impossible to also check for completeness. In many jurisdictions, only a small number of court decisions gets published. Appeal court cases are much more likely to get published than cases of courts in the first instance, but in order for a dispute to reach the appeal courts, the parties must have the social capital and financial resources to continue litigating. Other biases can be the result of uneven distribution of digital equipment and skills between courts in different regions of a country.

Another decision is whether to include cases from other jurisdictions. In the paper by Borges et al cited above, only German court decisions were used. For a German lawyer, this is such an obvious choice that it is not even worth discussing, *of course* for a formal representation of a German statute, only domestic courts are relevant. But for a UK lawyer, the answer would be far from obvious. The common law used foreign decisions much more liberally, at least as persuasive, if not binding, precedent. The reason for this different attitude is again deeply connected to historically grown conceptions of law and justice: a child of the modern nation state imposed by a central authority for the former, an organically grown expression of informal conceptions of justice rooted in human nature for the latter.

For a computer programmer to decide which foreign cases, if any, to include is therefore far from trivial, the choice inevitably reflects deeply ingrained commitments to a vision of the law.

Whatever approach we chose though, we will have changed the meaning of the original natural language sentence. The law, as formulated, was left intentionally vague and "open textured". While there are clear examples where the driver should have slowed down, there will also be borderline cases. By giving precise values to the various parameters (the "n"s) we introduce precision that the original was lacking. This increase in precision through formalisation has been lauded by some as an added benefit, as it achieves the value of predictability and legal certainty (Allen, 1956), the very thing Le Mettrie hoped to achieve through legal automation. But we can also see vagueness as a necessary human element that allows us to mitigate the harshness of the law by mercy. If we consider vagueness not as a bug to be fixed, but a positive feature of law, we may want a formalism that has a greater degree of flexibility. It is for this reason that at some point fuzzy logic was seen as a better medium for legal formalisation, as it reduces the "increased precision" that formalisation otherwise would bring (see e.g. Mazzarese, 1993; Philipps & Sartor, 1999).

The point for us is to emphasise that what may look superficially like a mere technological question happily left to computer scientists, in reality reflects deeply ingrained and culturally mediated normative assumptions about justice and the nature of the justice system. Depending on these philosophical commitments, vagueness in law is a problem that the AI developer should fix, or an important aspect of a humanist perception of law that could be distorted by legal AIs.

Cheat 4: just passing through your country…

We note in the passing another "cheat" with our formalisation above – typical for most legal AIs, it does not represent from which jurisdiction it comes. It is another aspect of the law that we "intuitively know" and never consider necessary to state explicitly, or only in the manual. As in the examples above, the formal rule, as stated, is false – but normally, we can rely on the user to compensate for this loss in translation. If a system is developed by and for German lawyers, for adjudicating solely German cases, we may not need to formally represent the jurisdiction. But if we build an AV that may cross borders while in operation, introduction of formal parameters and indices for jurisdictions may be needed, and new sets of formal rules that determine which country's laws apply.

We have used this example of a very small fragment of road traffic law to introduce some basic issues and vocabulary. The theme that emerged is that while the aim of legal technology is to automate legal reasoning, the process of formalising law is not in turn a mere mechanical, automated process. Instead, it is an exercise in normative reasoning that touches upon intuitions about justice, fairness and the nature of legal rules. What counts as an adequate formalisation can differ between jurisdictions as much as between intended applications. And every design choice will inevitably reflect often implicit assumptions about the nature of justice and the role of law in society – which becomes an issue if these are a) outsourced to software developers and b) remain in the "black box" that is the development process.

Law's Golems

This section will be using the myth of the Golem as a foil for contemporary discussions on legal AI to deepen some of the ideas introduced above.

The Golem story resonates with current discussions on law and robotics on several levels. The first Golem was also a "black box". It was not given a voice, and with the inability to speak came an inability to account for its actions. So when another rabbi, Rav Zeira, asked the Golem what it was and why it was doing what it did, it could not answer. The penalty for failing the first ever Turing test was sharp and fast: "You were created by the sages; return to your dust".

Today we are too worried about explainable or interpretable AI: it is not enough for many autonomous systems that they deliver the right result, we also want to know why exactly they

behaved as they did. Increasingly, for many AI-enabled or supported applications, this is becoming also a legal requirement. We encounter it in the duty to give reasons for fully automated decision making under the GDPR, though its full scope there is contested (See for an overview Kaminski, 2021). Even more detailed requirements are stipulated in the proposal for an EU AI Act (Hacker & Passoth, 2022).

The story of the Golem has been retold countless times over the centuries. In them, the Golem is typically performing any task given to it, but performing it literally and unthinkingly.

In the Golem of Prague, this task is to protect the Jewish community. Denied a proper legal status by the Christian majority, and subject to constant discrimination and harassment, the best they can hope for is a benevolent tyrant. But even a benevolent tyrant is a tyrant, and what he gives as protection on a whim he can equally rescind on a whim. In such an environment, "living life lawfully" becomes impossible, the pursuit of a coherent life plan and a life with integrity brittle and fragile (Bankowski, 2001; Bankowski & Schafer, 2007). In such a chaotic environment, advocating for "mechanistic" legal rule following is not a bug that de-humanises the legal system, it is a design feature that holds the promise of justice, and with that freedom, for all. Humans like Emperor Rudolf may be unpredictable, but with a Golem, we can know, by design, exactly what it will be doing.

Or can we? Because of course, this is not how the best-known of the Golem stories end. Rather, the Golem becomes inevitably dangerous for its human owner. In one such story, the owner forgets to switch the Golem off on a Friday evening. As a result, it continues to perform its assigned task also when Sabbath begins. This however breaks Jewish law, the law it is designed to follow, always and unwaveringly. It now faces a normative conflict: The law "obey the command given to you by your owner" conflicts with another rule it is programmed to abide by, "Respect the sabbath". Ultimately, this destroys the Golem, when the internal rule conflict becomes too much to bear. The underlying idea had seemed sound – ensure the safety of an autonomous device by – literally – hard baking the legal rules into its clay, the "Shem". However, all this assumes that our legal rules are consistent, and while this is an aspiration for modern legal systems, in reality we know of course that rules often (seem to) contradict each other.

We find the same problem with the golems of our age, autonomous systems. Let us now modify our AV example above. Imagine the following: A police officer spots an abandoned car parked in front of a primary school. On inspection, he realizes that it contains a bomb, programmed to go off soon. He can't safely diffuse it in situ, the only way to avoid the death of hundreds of innocent people is to risk his own life and drive the car as fast as he can away from bystanders. In this situation, we would not want the AV to tell him "I can't let you do that, Dave" and artificially slow down the car to local speed limits. Golem-like rule following, without human override, is one of the ways in which the historical man of clay and its modern reincarnation can cause harm.

Can we just add another rule, one that says: "allow to drive in an emergency as fast as possible"? If we use PROLOG or a similar programming language based on classical logic, the outcome would be similar to that in the Golem story. Classical logic, and the AIs that are built with it, are as unforgiving towards contradictions as the Golem was, a problem known since the Middle Ages as "logical explosion" (Priest & Routley, 1982). From any contradiction in the program, every statement becomes provable, as a counterintuitive and undesirable side result of how formal logic works. From the contradictory set of rules {"drive below local speed limits if there are pedestrians nearby"; "drive as fast as possible if the car contains a bomb"} the car could also derive, as counterintuitive as this sounds, "drive on the pavement and aim to hit as many people as possible".

Law for Golems

So, what should Rabbi Loew have done, what should a modern AV designer do, to keep an automaton law-compliant and safe? There are a couple of strategies, each with its own advantages and disadvantages.

Making legal formalisms Golem proof

First, the designer could have disambiguated the law *before* formalizing it. Rather than using literal, direct translations of two contradictory norms, they should have looked for the intended meaning behind the rules, and found a translation that avoids the conflict. Maybe the second rule could be formulated as an exception to the first rule:

"If the speed limit is X, *AND the driver does not push the "override button",* reduce speed as soon as the speed sensor gives the reading "X+n"".

This rule is not found anywhere in the UK Road Traffic Act, but it is how the legislator would have wanted the law to be understood. To achieve this result in a legal AI, it may be necessary to use a more expressive formalism as well, in the case at hand one of the many forms of "non-monotonic" logic that can express the idea that often, prima facie applicable rules can be "defeated" if an exception applies (see e.g. Johnston & Governatori, 2003).

For our mini-formula introduced above, this typically means to replace the "if then" arrow with a new constant, "~", so that "A ~ B" intuitively reads: "If A, then *typically* B" or "If A, then B *unless* challenged by a sound counter-argument".

Sometimes, the relevant exceptions can be found in the relevant legislation itself. Very often, legal rules explicitly refer to other norms, and a faithful formalization needs to preserve this element. Even more often, a law may state explicitly in one of its introductory sections that all the norms that follow should be constructed as an exception to another law, or that conversely, they do not apply when a named, higher ranking law also applies. In this case it is not enough to formalize every individual rule in isolation, rather the programmer has to read the norm in its context and ensure that references to other parts of the same law, or references to other laws, are formally represented. This is how lawyers are trained to read a statute, and replicating this process in the design stage of building a legal AI seems normatively unproblematic, though we move progressively away from a simplistic notion of legal formalization and rule isomorphism that allowed us to read the correctness of a proposed legal AI directly from its code. On automatic identification and formal representation of legal cross-references see e.g. de Maat Winkels and van Engers (2006) or Maxwell, at al (2012).

Neither strategy would work however in our example. Here, the higher-ranking norms that allow the violation of the Road Traffic Act are not mentioned directly anywhere in the statute. Rather, lawyers understand the hierarchy of norms and values that turn our laws from a mere list of rules into an organized system, and know for instance that the general rules regarding the "necessity defence" trumps in our case the UK Road Traffic Act. This indicates a much more significant problem in the task to formalize law: The legal system is first and foremost a *system*, it aims to promote a coherent set of values, and as a result the meaning we assign to a

given legal norm may depend on the meaning we assign to any other norm in the system. Some jurisdictions give this insight itself the status of a meta-norm, as an instruction to choose that interpretation of a norm that fits best with the interpretation of all other norms (Felix, 1998).

Here we can see a parallel to translations between natural languages – should a translator aim for a literal sentence-by-sentence translation, or one she thinks most closely matches the effect the author wanted to have on their audience overall?

A good translator will at the very least avoid introducing additional ambiguities. If e.g. a term for a character trait in the original language has two possible translations in the target language, one with positive and one with negative connotations, they will choose the one that is consistent with the way the person is described elsewhere in the same book. Would they, or should they, go further than that and also aim at maximum consistency across different novels about the same character (e.g., the Sherlock Holmes character across the stories), even if this risks to "repair" a real inconsistency in the original? Or is the risk too great that this obscures a character development that was intended by the author? How much should they consider the background knowledge of the target audience, which may have been different from the one that the original author had in mind?

In translation studies, the technical skills, professional and ethical implications of these decisions have created a comprehensive body of knowledge, most recently also as a response to the rise of the machine translation (see e.g. Coban, 2015; Floros, 2020). There is regrettably not a similar body of knowledge when it comes to formalizing law, despite the similarities.

In law, these issues raise additional questions of legitimacy and transparency. The programmer takes on a role that normally, society assigns to the legislator or the judiciary. This may be less of an issue if the aim is to build a car that adheres to road traffic law. Just as we as citizens have to decide if a given law applies to a situation and in that sense constantly "interpret" the law, the programmer has to decide what action a given situation requires of the car.

The situation is different when the AI replaces judicial or other law-based decision making by a public authority, such as a decision whether a citizen is entitled to certain benefits. Here we face the danger that software developers, in the process of formalization, take on a role that they are neither qualified nor authorized to do - deciding "what the law is".

Making Golems inconsistency-tolerant

So far we discussed how during the formalization process, the "raw data" of the law has to be reformulated and "cleaned up" first, not unlike the way machine learning approaches to AI first require extensive data preparation that also happens "behind closed doors".

A very different strategy is to keep the inconsistency, at least initially, represent it faithfully in code, and prevent logical explosion through a modified logic.

There are a number of formal systems that have been designed to achieve that ability. *Paraconsistent logics* for instance tolerate local inconsistencies, and allows to represent how these can be resolved over time (see generally Priest, 2002; for an application to law see Ausin & Pena, 2000).

Rather than asking the *programmer* to sanitize the law prior to formalization, the process of disambiguation itself gets represented *in* the program This may also involve formalizing those meta-rules that lawyers deploy to resolve inconsistencies, for instance the legal rule "lex priori derogat lex posterior". Rules like this are then not used informally by the programmer *prior* to formalization, but become part of the computer representation of the law. This gives a richer – and less idealised - account of the law. It also transfers part of the "invisible" process of reformulating the law by the developer into an explicit and visible part of the operation of the legal AI. As with classical logic, the vocabulary of paraconsistent logic too can be extended with deontic operators, so that we can express inconsistent obligations (McGinnis, 2007).

Give the Golem a voice: From monological to dialogical formalization

So far, we have treated the law in our examples as a set of instructions directed at humans, machines or human-like machines. The Golem is told which laws to abide by, and not having a voice, they are not open for debate. Similarly, an AV that is designed to be law compliant will simply reduce its speed if the road traffic law so requires. The AV in this case is unlikely to explain itself, let alone argue with the driver or developer about the merits of the law.

More ambitious but still "monological", are legal expert systems that assist legal decision makers. They will produce a decision as output, but also give a valid legal reason for it. In our example the output can read as: You are given 3 penalty points on your license, *because* the law says in sec 152 of the Road Traffic Act that in the presence of pedestrians, a driver must lower their driving speed appropriately, and you did not do so".

But are these adequate justifications? It tells me why I was found guilty, but it does not tell me why *my* arguments – for instance that speeding was necessary to prevent even more danger – were rejected. What this indicates is that a monological understanding of the law omits elements that are important, maybe even constitutive, in other contexts.

We mentioned above the different conceptions of law found in Austin and Hart. If we understand law just as a command, directed at a citizen or Golem, by an all-powerful sovereign, thinking of it as a monologue makes sense. But if laws are mainly instructions to officials, in particular judges, then the contested nature of law becomes more prominent. In the trial, it is essential that both sides have a voice, and the process of deliberative evaluation of their respective arguments by the judge is constitutive for a fair trial. "E xplaining" the decision now also has to mean to explain why some arguments failed. Such a richer notion of the trial and the type of explanation it generates can be found e.g. in Brownlee's (2011) development of Duff's communicative theory of the trial. Going back to our speed driving example from above, we now shift from an AI that simply reasons:

1) You should drive slowly and carefully on streets where there are likely to be pedestrians, cyclists and parked cars"
2) there were pedestrians, cyclists and parked cars
Therefore
3) you should drive slowly

to one that can make a disagreement between two parties (Defendant D and prosecutor P) explicit and reason about the argument they are making:

P: Proposition 1: You should drive slowly and carefully on streets where there are likely to be pedestrians, cyclists and parked cars. There were pedestrians, cyclists and parked cars but you did not drive slowly, therefore you did something you should not have done

D: Proposition 2 While true, I transported a bomb away from pedestrians, and there is a general necessity exception that allows to break the law if needed to save life ("attacks" the antecedent of proposition 1)
P: Proposition 3: While 2 is generally true, it only applies when the value of the law that was broken is less important than the value that was protected. Here however speed limits are there to protect life, and balancing life against life is not permitted under the necessity defence (attacks the attack-proposition 2)

While early legal AIs in the 80s and 90s followed the first, monological model of legal reasoning, the limitations of this approach became quickly visible: much of the law is in the form of debate and disagreement, and if the AI developer has to resolve all these disagreements even before the formalisation can begin, they

a) act outside their competence
b) potentially usurp the rule of the judges
c) fail to produce *adequate* explanations of the decision
d) simply miss much of what makes law unique.

Formal systems were therefore developed that again extend the simple language we encountered above. Formal dialogue systems come in a huge variety of forms, all with different expressive power and abilities, but typically share the idea that an "attack" relation and a "defence" relation between arguments can be formally expressed (see eg. Prakken & Sartor , 2015; Walton, 2005; for an application to case based reasoning see Prakken *at al*., 2015)

The output is then not any longer the "one right answer" but rather a complex map of interrelated arguments. While it also becomes possible to formally define the "winner" of the debate as the party which has at least one undefeated argument, it nonetheless makes it easier for the losing side to contest the decision.

New laws for old Golems

So far, our strategy was to either remedy any inconsistencies prior to formalisation, or to make the reasoning about inconsistencies explicit within the AI. A more radical strategy is to simply choose one of the rules and disregard the conflicting norm altogether.

Maybe the benefits of reducing road traffic accidents through AVs that strictly adhere to speed limits are worth the loss of flexibility in a very small number of exceptional cases. Maybe the protection that the Golem provides is worth the theoretical risk of its malfunctioning, provided we can mitigate the harm. After all, we are also happy to use ATM machines that will never give us money that is not in our account – even if in a hypothetical case, it may be needed to pay off a kidnapper who threatens to kill his hostage. A human bank manager might have been swayed by this argument, still, as a society, we consider the advantage of ubiquitous access to money higher, and deal with the exceptional cases through alternative strategies.

So far, we treated this as a problem of the AI: If the formal language that the computer understands is insufficiently expressive for a given task, the "fault" is with the approach to computation, and we have to develop better and more expressive languages. But it is of course also possible to invert this argument: maybe, if our legal system is riddled with contradictions and ambiguities, that is the problem to fix? Maybe a good legal system *should* be easily rendered computational, for the very same reasons that we develop legal AI in the first place: because this is the way to achieve justice and in particular formal equality. What the critics consider as pathological "legalism", may just be a particularly radical form of legality (see e.g. Bankowski & Schafer, 2007; Diver, 2021).

Legal AI from its inception was closely linked to such a vision of formalism. Early legal technology in the 1980s and 90s in particular often took the formalist account of law as *descriptively* correct and mirrored it in the design of expert systems. The lack of success of these systems was then seen as a consequence of this association of legal technology with an (inadequate) legal theory (Zeleznikow & Hunter, 1995; Zeleznikow, 2019). Legal expert systems failed, in this analysis, due to descriptive inadequacy – lawyers could not use them because the reality of the law was too different.

But there was of course always an alternative: formalism might fail as a descriptive theory of the law, and as a result legal AI may distort its subject matter *as it currently is*. But maybe the law *ought* to be logical, simple and rule based. Rather than developing more and more complex formalisms to capture the law *as it is*, maybe we should change the law to make it more amenable to formalization.

This was the idea behind the "EDV- compliant legal drafting" movement in Germany, promoted by academics like Herbert Fiedler and implemented in some low-level legislative projects (so Fiedler, 1976).[5] The idea was to incorporate a future formalization already at the drafting stage, and write legislation simple enough for 1970s natural language parsers.

The project was ultimately unsuccessful, also because the limitations of the technology at that time would have meant to force legislators to use highly unintuitive language, while any possible benefit remained purely speculative.

Today we see a similar idea in the "Law as Code" movement (programmatically Lessig, 2009). Law as code complements the idea of using "code as legal enforcement" by changing the way legislators enact and promulgate rules, so that in addition to the natural language version directed at citizens, an "authorized" translation into software code is enacted as well (Waddington, 2021; Waddington, forthcoming).[6]

This approach to formalizing law addresses one of the normative concerns expressed in this chapter. As every formalization of the law also distorts its meaning, with what authority can AI developers make the necessary decisions? How can they formalize the law without usurping the role of the legislator? With an approved formalization enacted in parallel with the natural language text, this danger can be mitigated.

However, as we have seen, the formal version of the law is inevitably more austere, simple and rigid – it serves one specific aspect of our intuition about justice, but in the past, this intuition was always balanced with other, competing values. So, we have rules, but also discretion, harsh punishment but also amnesties, we treat like cases alike, but try to be responsive to the particularities of a given case. Over time, we can see the influence of these philosophies wax and vane, but they never entirely excluded their opposite, not even during the high days of formalism in the 19th century. Or, the Supreme Court of Alabama opined in Alan v. State (1973):

---

[5] For an implementation see e.g the practice guidance of the ministries of Niedersachen: . "Grundsätze für die Fassung automationsgerechter Vorschriften" der Niedersachs. Ministerien from the 1.6.1970

[6] For a practical project see the Jersey legislative drafting project https://www.gov.je/Government/NonexecLegal/StatesGreffe/SiteAssets/pages/legislativedraftingoffice/Introduction%20to%20the%20Computer-Readable%20Legislation%20Project.pdf

"We have not, and hopefully never will reach the stage in Alabama at which a stone-cold computer is draped in a black robe, set up behind the bench, and plugged in to begin service as Circuit Judge."[7]

Very clearly, "narrow, legalistic ways" are not enough, and equally, "being stone-cold" is not an endearing quality in a judge. Humans are good at balancing conflicting normative ideals and reason which and about rules that are in conflict But as we saw above, legal AI is much less accommodating. The problem then with formalized law is not necessarily that it is wrong, or does not serve a normative good, but that it has to do so at the exclusion of all other aspects of our legal ideals. If we change what we expect of the justice system – or even worse, when the prevalence of legal technology subliminally and clandestinely lowers our expectation of what we can demand of justice – then formalizing the (new conception of) law becomes easy(er).

Conclusion:

We have now discussed all the points that our initial question raised. Is law computational, or more specifically, is it formalizable?

And it should be clear by now what our answer is: as worded the question is meaningless, as the answers will be both trivial and misleading.

Can law be formalized? Yes of course!

Define a function f so that the first sentence of a statute is assigned the propositional variable A, the second sentence the propositional variable B, the third sentence…..

Perfect formalization, and no problematic judgement calls by the formalizer needed to. But of course utterly useless for any practical purpose. So maybe we need more:

Can *law* be formalized? Yes of course!

---

[7] Allan v State 290 Ala 339 342 (1973)

We simply need to simplify the law, and lower our expectations of what the justice system does. We may even have good normative reasons for this.

Both answers are correct, and both are misleading, at least if the purpose of formalization is to build useful tools. Law is trivially formalizable, and even risk-free but the outcome is useless. Or the law can be formalized as long as we change our expectations of what the law ought to deliver – the easiest, but also the most dangerous option.

What we want in reality is something in the middle, a formalisation that preserves aspects of the law we find interesting or relevant, without distorting the law that it represents too much. For this we need to know who will operate it (and bring their understanding to the task, complementing the machine) , in what context and with what aims. "Interesting", "relevant" and "too much" are irreducible human judgements.

So we need to ask a different question, one that is also reflected in the ethics principles for legal technology by AI4People. Not: "can law be formalized". But: "For a given intended application, and given the skills and knowledge of the human who will eventually operate the system, can we formally explicate enough aspects of the meaning of the legal terms that interest us so that the inevitable loss of meaning is not so harmful, to individuals or the justice system, that it outweighs the benefits of automation?"

No formalization is value free; it always requir es normative decisions and commitment to a specific vision of justice. No formalization is "provably the right one", but only ever "good enough for a given objective". This evaluation needs to consider how the system will operate in practice, and at which points human input and judgement will be required – don't evaluate legal AI, evaluate the socio-legal systems into which AIs are embedded.

Law is based on the notion of contestability. AI applications in the legal domain are potentially undermining this contestability. This is the fear of the "black box" society, where intelligible (and hence contestable" human decisions are replaced by modern day oracles, whose pronouncements can only be interpreted by the high priest of technology.

Our discussion however showed a very different form of "black boxing" that may be more difficult to remedy. Hidden in the early stages of the development process, crucial normative

decisions are already taken when the law is translated into machine readable form. To allow contestability of the value judgments and design decisions that come with it requires

- A theory of formalization as a precondition for standards for both the formalization process and, crucially, its documentation
- An understanding of the ethical and legal implication of the choices that are taken in the formalization process , which requires subject expertise in law

If we take the metaphor that underpinned this chapter seriously , we can see the emergence of a framework that could address this issue. Translation studies as a profession has not only developed methods, standards and concepts that allow to evaluate and critique translations, it is also engaged in a process of the ethical meta reflection that sensitises practitioners to the ethical implications of the translation choices. A similar professionalisation, including ethics training and appropriate certification, will also be needed for developers of GOFAI legal technology to ensure that we can reap the benefits of automation while respecting the rule of law ideal.

Acknowledgment

Work on this paper was supported by EPSRC grant EP/T027037/1 AISEC: AI Secure and Explainable by Construction. I benefited greatly from my discussions with Laurence Diver and Pauline McBride, though all mistakes are my own. As per University of Edinburgh policy, for the purpose of open access, the authors have applied a 'Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

Bibliography

Addady, M. (2016, May 12). *Meet Ross, the World's First Robot Lawyer, Fortune* . Retrieved from https://fortune.com/2016/05/12/robot-lawyer/

Allen, L. E. (1956). Symbolic logic: A razor-edged tool for drafting and interpreting legal documents. *Yale Law Journal*, *66*, 833 -879

Araszkiewicz, M. (2013). Towards Systematic Research on Statutory Interpretation in AI and Law. In K. D. Ashley (Ed.). *JURIX, Proceedings of the 26th conference*. Amsterdam: IOS 15-24

Ashley, K. D. (1992). Case-based reasoning and its implications for legal expert systems. *Artificial Intelligence and Law*, *1*(2-3), 113-208

Ashley, K. D. (2002). An AI model of case-based legal argument from a jurisprudential viewpoint. *Artificial Intelligence and Law*, *10*(1-3), 163-218
Atkinson, K. (Ed.). (2009). *Modelling Legal Cases. Proceedings of the Workshop co-located with the 12th International Conference on Artificial Intelligence and Law*. Barcelona: Huygens
Atkinson, K., Bench-Capon, T. & Bollegala, D. (2020). Explanation in AI and law: Past, present and future. *Artificial Intelligence*, *289*, 103387
Ausín, F. J. & Peña, L. (2000). Paraconsistent deontic logic with enforceable rights. In D. Batens, Ch. Mortensen, G. Priest & J.-P. van Bendegem (Ed.). Frontiers of Paraconsistent Logic (pp. 29-47). Baldford: Research Studies Pres
Austin, J. (1880). *Lectures on jurisprudence, or, The philosophy of positive law*. London: John Murray
Baker, M. & Maier, C. (2011). Ethics in interpreter & translator training: Critical perspectives. *The interpreter and translator trainer*, *5*(1), 1-14
Bańkowski, Z. (2001). *Living Lawfully*, Dordrecht: Springer
Bańkowski, Z. & Schafer, B. (2007). Double-click justice: Legalism in the computer age. *Legisprudence*, *1*(1), 31-49
Bench-Capon, T. J. & Coenen, F. P. (1992). Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, *1*, 65-86
Bench-Capon, T., Araszkiewicz, M., Ashley, K., Atkinson, K., Bex, F., Borges, F. & Wyner, A. Z. (2012). A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law. *Artificial Intelligence and Law*, *20*, 215-319
Binding, K. (1872). *Die Normen und ihre Übertretung: eine Untersuchung über die rechtmässige Handlung und die Arten des Delikts. Erster Band, Normen und Strafgesetze*. Leipzig: Wilhelm Engelmann
Borges, G., Wüst, C., Sasdelli, D., Margvelashvili, S. & Klier-Ringle, S. (2023). Making the Implicit Explicit: The Potential of Case Law Analysis for the Formalization of Legal Norms. In G. Borges, K. Satoh & E. Schweighofer (Eds.). Proceedings of the International Workshop on Methodologies for Translating Legal Norms into Formal Representations. Retrieved from https://research.nii.ac.jp/~ksatoh/LN2FRproceedings.pdf
Branting, L. K., Pfeifer, C., Brown, B., Ferro, L., Aberdeen, J., Weiss, B. & Liao, B. (2021). Scalable and explainable legal prediction. *Artificial Intelligence and Law*, *29*, 213-238
Brownlee, K. (2011). The offender's part in the dialogue. In R. Cruft , M.H. Kramer &
M. R. Reiff (Eds.). *Crime, punishment, and responsibility: The jurisprudence of Antony Duff* (pp. 54-67). Oxford: Oxford University Press
Brun, G. (2003). *Die richtige Formel: Philosophische Probleme der logischen Formalisierung*. Berlin: de Gruyter
Brüninghaus, S. & Ashley, K. D. (2003). Combining case-based and model-based reasoning for predicting the outcome of legal cases. In K. Ashely, D. Bridge (Eds.) *Case-Based Reasoning Research and Development: 5th International Conference on Case-Based Reasoning, ICCBR 2003 Trondheim, Norway, June 23–26, 2003 Proceedings 5* (pp. 65-79). Berlin: Springer
Campbell, B. (1970). La Mettrie: the robot and the automaton. *Journal of the History of Ideas*, *31*(4), 555-572
Chesterman, A. (1997). *Memes of Translation: The spread of ideas in translation theory* (Vol. 22). Amsterdam: John Benjamins Publishing

Coban, F. (2015). Analysis and training of the required abilities and skills in translation in the light of translation models and general theories of translation studies. *Procedia-Social and Behavioral Sciences*, *197*, 707-714

Cresswell, M. J. & Hughes, G. E. (2012). *A new introduction to modal logic*. London: Routledge

de Maat, E., Winkels, R. & van Engers, T. (2006). Automated detection of reference structures in law. In T. van Engers (Ed.). *Legal knowledge and information systems* (pp. 41-50). Amsterdam: IOS Press

Deakin, S. & Markou, C. (Eds.). (2020). *Is law computable?: critical perspectives on law and artificial intelligence*. London: Bloomsbury Publishing

Diver, L. (2021). Computational legalism and the affordance of delay in law. *Journal of Cross-disciplinary Research in Computational Law*, *1*(1). Retrieved from https://journalcrcl.org/crcl/article/view/3

Duguid, S., Edwards, L. & Kingston, J. (2001). A web-based decision support system for divorce lawyers. *International Review of Law, Computers & Technology*, *15*(3), 265-279

Felix, D. (1998). *Einheit der Rechtsordnung: zur verfassungsrechtlichen Relevanz einer juristischen Argumentationsfigur*. Stuttgart: Mohr Siebeck

Fiedler, H. (1976). Automationsgerechte Rechtssetzung im Rahmen der Gesetzgebungstheorie. In: J. Rödig, E. Altmann, E. Baden, H. Kindermann, R. Motsch & G. Thieler-Mevissen (Eds.). *Studien zu einer Theorie der Gesetzgebung*. Berlin: Springer(pp 666-678)

Floros, G. (2020). Ethics in translator and interpreter education. In M. Zhou (Ed.) . *The Routledge handbook of translation and ethics* (pp. 338-350)

Gabbay, D. M. (1992). How to construct a logic for your application. In H. Ohlbach (Ed.). *GWAI-92: Advances in Artificial Intelligence, 1992 Proceedings. Lecture Notes in Computer Science* (Vol. 671, pp. 1-29). Berlin, Heidelberg: Springer

Hacker, P., Passoth, J.H. (2022). Varieties of AI Explanations Under the Law. From the GDPR to the AIA, and Beyond. In: A. Holzinger, *et al* . (Eds.). *xxAI – Beyond Explainable AI. Lecture Notes in Computer Science* (Vol. 13200). Cham: Springer. Retrieved from https://doi.org/10.1007/978-3-031-04083-2_17

Hart, H. L. A. (2012). *The Concept of Law*. Oxford : OUP

Hildebrandt, M. (2015). *Smart technologies and the end (s) of law: novel entanglements of law and technology*. Cheltenham: Edward Elgar Publishing

Johnston, B. & Governatori, G. (2003, June). Induction of defeasible logic theories in the legal domain. In G. Sartor (Ed.). *Proceedings of the 9th international conference on Artificial intelligence and law*. New York: ACM (pp 204-213). Retrieved from https://doi.org/10.1145/1047788.1047834

Kaminski, M. E. (2021). The right to explanation, explained. In S.K. Sandeen, C.W. Rademacher & A. Ohly . *Research handbook on information law and governance* (pp. 278-299). Cheltenham: Edward Elgar Publishing

La Mettrie, J. O. D. (1994). *Man a Machine and Man a Plant*, trans. R. A. Watson & M. Rybalka. Indianapolis: Hackett Publishing

Lenzen, W. (2005). Leibniz on alethic and deontic modal logic. In D. Berlioz, F. Nef (Eds.). *Leibniz et les puissances du langage* (pp. 341-362). Paris: J Vrien

Lessig, L. (1999). *Code and Other Laws of Cyberspace*. New York: Basic Books

Mackaay, E., Poulin, D., Frémont, J., Bratley, P. & Deniger, C. (1990). The logic of time in law and legal expert systems. *Ratio juris*, *3*(2), 254-271

Maxwell, J. C., Antón, A. I., Swire, P., Riaz, M. & McCraw, C. M. (2012). A legal cross-references taxonomy for reasoning about compliance requirements. *Requirements Engineering*, *17*, 99-115

Mazzarese, T. (1993). Fuzzy Logic and Judicial Decision-Making: A New Perspective on the Alleged Norm-Irrationalism. *Informatica e diritto*, *2*(2), 13-36
McCarty, L. T. (1976). Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning. *Harv. L. Rev.*, *90*, 837
McGinnis, C. N. (2007). *Paraconsistency and deontic logic: Formal systems for reasoning with normative conflicts*. PhD, University of Minnesota. University of Minnesota ProQuest Dissertations Publishing Retrieved from https://www.proquest.com/docview/304840273?fromunauthdoc=true
Merigoux, D., Chataing, N. & Protzenko, J. (2021). Catala: a programming language for the law. *Proceedings of the ACM on Programming Languages*, *5*, 1-29
Meyer, J. J. C. (1993). Deontic logic: A concise overview. In J.J.C. Meyer & R. J. Wieringa (Eds.). *Deontic Logic in Computer Science: Normative System Specification* (pp. 3-16). Chichster: Wiley
Mills, M. (2016). *Artificial Intelligence In Law: The State Of Play*. Retrieved from https://britishlegalitforum.com/wp-content/uploads/2016/12/Keynote-Mills-AI-in-Law-State-of-Play-2016.pdf
Philipps, L. & Sartor, G. (1999). From legal theories to neural networks and fuzzy reasoning. *Artificial Intelligence and La*, *7*, 115
Prakken, H. & Sartor, G. (2002). The role of logic in computational models of legal argument: a critical survey. In F. Sadri (Ed.). *Computational logic: Logic programming and beyond* (pp. 342-381). Berlin: Springer
Prakken, H. & Sartor, G. (2015). Law and logic: A review from an argumentation perspective. *Artificial intelligence*, *227*, 214-245
Prakken, H., Wyner, A., Bench-Capon, T. & Atkinson, K. (2015). A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. *Journal of Logic and Computation*, *25*(5), 1141-1166
Priest, G. (2002). Paraconsistent logic. In D.M. Gabbay & F. Guenthner (Eds.). *Handbook of philosophical logic* (pp. 287-393). Berlin: Springer
Priest, G. & Routley, R. (1982). Lessons from pseudo scotus. *Philosophical Studies*, *42*(2), 189-199
Rissland, E. L., Ashley, K. D. & Branting, L. K. (2005). Case-based reasoning and law. *The Knowledge Engineering Review*, *20*(3), 293-298
Robertson, B. & Vignaux, G. A. (1993). Probability—the logic of the law. *Oxford Journal of Legal Studies*, *13*(4), 457-478
Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y. & Takano, C. (2011). PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In T. Onada, D. Bekki, E. McCready (Eds.) *New Frontiers in Artificial Intelligence: JSAI-isAI 2010 Workshops Tokyo, Japan, November 18-19, 2010, Revised Selected Papers 2* (pp. 153-164). Berlin : Springer
Schafer, B. & Bromby, M. (2005). Wie Tajomaru seine NemeSys fand: Expertensysteme zum Augenzeugenbeweis. In B. Schünemann, M.-T. Tinnefeld, R. Wittman (Eds.). *Gerechtigkeitswissenschaft* (pp. 259 -277). Berlin: Berliner Wissenschaftsverlag
Schafer, B. *et al.* (2020). Legal Services Industry. In *AI4People 7 AI Global frameworks* (pp. 171-209). Retrieved from https://ai4people.eu/wp-content/pdf/AI4People7AIGlobalFrameworks.pdf
Sourdin, T. (2018). Judge v Robot?: Artificial intelligence and judicial decision-making. *University of New South Wales Law Journal*, *41*(4), 1114-1133
Sparkes, M. (2023). AI will advise a defendant in court. *New Scientist*, *257* (3421)
Susskind, R. (2008). *The end of lawyers* (pp. 121-123). Oxford: Oxford University Press

Thomson, A. (2016). French eighteenth-century materialists and natural law. *History of European ideas*, *42*(2), 243-255
Ulenaers, J. (2020). The impact of artificial intelligence on the right to a fair trial: Towards a robot judge?. *Asian Journal of Law and Economics*, *11*(2)
Waddington, M. (2021). Rules as Code . *Law in Context*,Vol 37 179-186. Retrieved from https://journals.latrobe.edu.au/index.php/law-in-context/article/view/134
Waddington, M. (2022). Rules as Code: Drawing out the logic of legislation for drafters and computers. In C. Stefanou (Ed.). *Modern Legislative Drafting-A Research Companion, Routledge* (Forthcoming). Retrieved from https://ssrn.com/abstract=4299375 or http://dx.doi.org/10.2139/ssrn.4299375
Walton, D. (2005). *Argumentation methods for artificial intelligence in law*. Berlin: Springer
Zeleznikow, J. (2019). Reflections on my journey in using information technology to support legal decision making-from legal positivism to legal realism. *Law in context*, *36*(1), 80-92
Zeleznikow, J. & Hunter, D. (1995). Reasoning paradigms in legal decision support systems.

Artificial intelligence review, *9*, 361-385