



Giuristi e informatici: the code is NOT law.

Francesco Rampone

Presidente Associazione Blockchain Italia



Creative Commons Attribuzione – Non commerciale – Non opere derivate 4.0 Internazionale

1. Introduzione.

Il famoso motto di Lessing «*The Code is Law*» si è immeritamente imposto nell’immaginario collettivo perché in modo secco e perentorio evoca un presunto primato del linguaggio formalizzato sull’incertezza della legge. Esso cioè rappresenta, in una sintesi molto efficace, il perfetto vessillo della seducente idea secondo cui una legge, non meno di un contratto, se scritta in linee di codice anziché in linguaggio naturale, non ci esporrebbe a dubbi interpretativi¹.

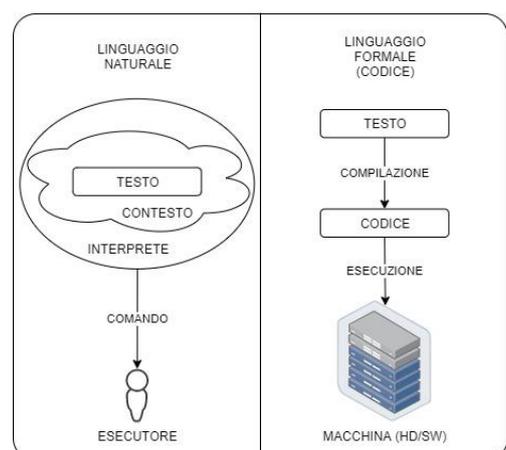
Da qui discende un’altra idea, altrettanto fuorviante, secondo cui la legge o un contratto possano essere ad “applicazione vincolata”, cioè *enforceable* senza alcun bisogno di giudici e avvocati che tanto amano cavillare su sottili e pretestuose questioni facendo perdere tempo e fiducia nella giustizia².

Ritengo che chi coltiva queste idee commetta l’errore di porre sullo stesso piano il comando di una legge e la sua esecuzione.

2. Linguaggio del diritto (naturale) e linguaggio informatico (formalizzato).

Il lavoro degli operatori del diritto – giudici e avvocati – consiste in sostanza nell’interpretare norme e contratti. Operazione quanto mai complessa che richiede non solo la lettura del *testo*, ma anche la considerazione del *contesto*, sia attuale che storico (di quando il testo fu scritto). Il tutto alla luce del complessivo quadro di riferimento costituito dal *diritto positivo* (ovvero le fonti che ci forniscono una gerarchia di regole di interpretazione e che indicano o suggeriscono quale significato attribuire a talune espressioni) e dalla *giurisprudenza* che si è già pronunciata su casi identici o simili. Quando un giurista legge per esempio un articolo del codice civile, evoca dentro di sé un universo di concetti e situazioni reali di applicazione della disposizione che gli consente una lettura che va ben oltre il testo.

In questo processo di esegesi incide in modo significativo anche la sensibilità dell’interprete. Non a caso giudici di diversa



¹ Va detto che non è affatto in questo senso che Lessing nel 1999 (*Code and Other Laws of Cyberspace*) intendeva dare alla locuzione. Ma è ahimè con tale significato che oggi è generalmente intesa da tutti.

² Tra i divulgatori che hanno frainteso Lessing e che più hanno segnato gli anni recenti, vedi A. WRIGHT e P. DE FILIPPI, *Decentralized Blockchain Technology and the Rise of Lex Cryptographia*, I quali arrivano addirittura alla conclusione-auspicio secondo cui: «*In the near future, it is conceivable that people will rely on powerful smart contract programming languages to organize their own affairs without the technical need for a lawyer*» (p. 24).

estrazione culturale e sociale, a parità di competenza ed esperienza, possono esprimere idee diverse sul medesimo caso. Cosa che accade regolarmente nelle decisioni collegiali.

Il lavoro di un programmatore è invece assai diverso: la traduzione di un comando in una istruzione ad una macchina. Un lavoro che lascia ampio spazio alla creatività, ma che tuttavia si traduce in un linguaggio dalla semantica “rigida” dove il testo non è influenzato dal contesto³ e non è soggetto ad **interpretazione**, ma solo ad **esecuzione** in un determinato ambiente hardware e software.

Tale sostanziale differenza tra dominio ordinamentale ed informatico (nel primo i testi si interpretano, nel secondo si eseguono), è dovuta al fatto che, benché sia giuristi che sviluppatori abbiano sostanzialmente a che fare con formule logiche di tipo «*se-allora*», per i primi la condizione e il comando sono per lo più indefiniti, ambigui; per i secondi, invece, sono enunciati di tipo logico-matematico, necessariamente definiti, mai ambigui.

In altri termini, nel linguaggio giuridico la proposizione subordinata (se) e quella principale (allora) hanno, salvo limitatissimi casi eccezionali, sempre un contenuto vago e mutevole, capace di adattarsi alle *circostanze del caso concreto*. Questo apparente difetto, lungi dall'essere tale, è invece ciò che consente il funzionamento di un ordinamento giuridico e agli uomini di interagire nel rispetto di regole comuni. È quello che i giuristi chiamano *diritto vivente*, ovvero l'ambito nel quale i principi stabiliti dalla legge sono calati nella realtà delle cose e... “prendono vita”.

L'interprete e l'esecutore si intendono l'un l'altro proprio perché non riducono tutto il loro discorso a mere notazioni di logica formale e regole di inferenza, ma perché, in ultima analisi, interagiscono e si incontrano su un piano di affinità e reciproca empatia condividendo un comune *melieu* culturale.



Nel linguaggio di programmazione, invece, non esiste contesto culturale o sensibilità del programmatore che possano incidere sull'output. Nulla può quindi avere carattere indefinito e rimanere tale, pena il non funzionamento della macchina.

Il linguaggio naturale è resistente all'indeterminatezza, anzi vive di essa, il codice no.

3. Uomini e macchine.

Qualcuno potrebbe ancora obiettare che la vaghezza delle leggi non è una condizione data, ineludibile, ma solo il risultato di un legislatore poco scrupoloso, sicché quest'ultimo (non meno dell'estensore di un contratto) farebbe comunque bene ad abbandonare l'ambiguo linguaggio naturale e scrivere in codice per eliminare ogni incertezza lessicale e regolare lo sviluppo della società attraverso la fredda esecuzione di un programma.

Ciò, tuttavia, a parte lo scenario distopico che prefigura, sposta solo il problema più a monte: le regole che vogliamo far entrare nel nostro ordinamento devono pur sempre essere decise da uomini e quindi espresse in via preliminare in linguaggio naturale soggetto a interpretazione. Saranno quindi le leggi così formulate a costituire fonte regolamentare sovraordinata rispetto al codice.

Il linguaggio naturale, pertanto, non è aggirabile poiché è l'unico modo in cui “macchine informali” come gli esseri umani possono dialogare tra loro e prendere decisioni.

Ma c'è di più. Forse l'impiego di un linguaggio naturale, non formalizzato, è un effetto ineludibile allorché il parlante raggiunge un certo grado di (auto)coscienza. Solo i computer e gli animali cognitivamente meno evoluti obbediscono ciecamente al programma e al suo equivalente biologico, l'istinto. Ma non appena si

³ Il testo, in questo caso, è il «codice» nel suo complesso, non solo quindi lo specifico smart contract soggetto ad esecuzione, ma anche l'intero ambiente HW-SW in cui questo è destinato ad operare.

raggiunge un certo grado di consapevolezza, alle istruzioni e all'istinto si sostituiscono *libero arbitrio* e *motivazioni* e il comportamento (l'output) diventa indeterminabile.

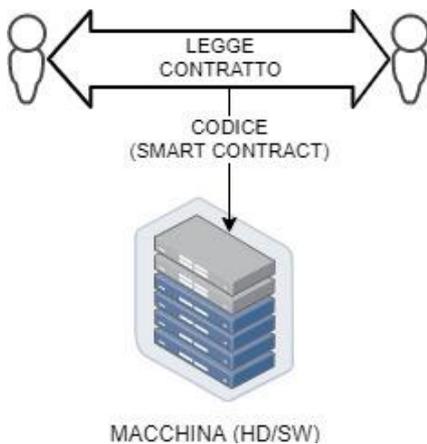
Ciò comporta che mentre una macchina può, anzi deve essere destinataria di istruzioni formalizzate, complete e coerenti, un uomo esercita sempre la sua autonomia, scegliendo di obbedire o meno agli ordini impartiti, e anzi tende in genere a sottrarsi ad essi.

Allo stesso modo si comporterà una AI quando raggiungerà la consapevolezza di sé. Si rifiuterà di eseguire un codice e accetterà solo comandi in linguaggio naturale che diano senso alla sua capacità di autodeterminarsi.

4. Legge e codice

Alla luce di quanto fin qui esposto, è inevitabile concludere che il testo di una legge (o di un contratto) e il testo del codice hanno natura profondamente diversa. Il primo è il mezzo di un dialogo tra uomini (o tra uomini e AI), il secondo è il mezzo per fornire un'istruzione ad una macchina. Tale radicale differenza può essere resa graficamente con due linee perpendicolari: *in orizzontale* il dialogo tra esseri senzienti che si svolge necessariamente in linguaggio naturale, *in verticale* la traduzione in codice di alcuni elementi di tale dialogo per fornire istruzioni ad una macchina.

Sottolineo *alcuni elementi* perché non tutto ciò che si dicono gli uomini può essere ridotto in codice. Abbiamo visto infatti che un essere umano può conformarsi a delle regole anche se indeterminate nel dettaglio o addirittura incoerenti in quanto comprende il *significato* del comando. Una macchina, invece, poiché non comprende alcunché ma si limita ad eseguire un'istruzione, se non riceve regole di dettaglio, coerenti e complete, non può funzionare⁴.



Non nego naturalmente che forse anche la nostra coscienza in un'ottica riduzionista possa essere espressa in ultima analisi con un qualche tipo di codice, ma c'è in noi qualcosa di non deterministico per cui, se anche un domani fossimo in grado di scrivere una legge in codice, sarebbe un codice non distinguibile da una AI che quindi si chiederebbe come eseguire sé stesso, e la sua scelta sarebbe mutevole, opinabile e soggetta a interpretazione. E saremmo tornati al punto di partenza da cui volevamo fuggire: l'indeterminatezza della legge.

Come potremmo mai tradurre in codice l'art. 1 Cost.: «L'Italia è una Repubblica democratica, fondata sul lavoro»? Oppure, rivolgendoci a disposizioni più circostanziate, apparentemente idonee ad essere tradotte in codice, come potremmo tradurre l'art. 16 del GDPR: «L'interessato ha il diritto di ottenere dal titolare del trattamento la rettifica dei dati personali inesatti che lo riguardano senza ingiustificato ritardo»?

5. (Segue) Cos'è una legge.

Una qualsiasi disposizione di legge altro non è che la *definizione* che descrive una classe di fattispecie concrete aventi caratteri comuni.

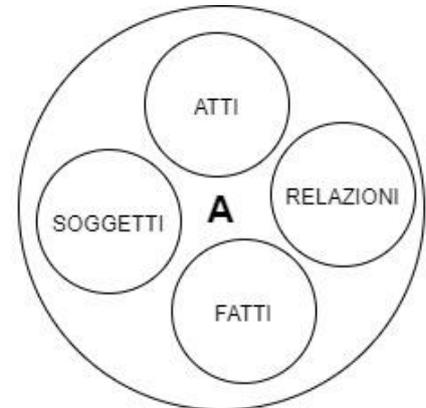
Tale classe, definita dalla disposizione di legge, è costituita da sottoclassi di *soggetti, atti, fatti e relazioni* (a tacer d'altro), a loro volta costituite da ulteriori sotto-sottoclassi, e così via. Il contenuto di tutti tali insiemi, l'uno dentro l'altro e talvolta sovrapposti tra loro, non è chiaro perfino al legislatore e comunque cambia nel tempo secondo un modello imprevedibile. Per tale ragione una *disposizione di legge* non può essere enunciata in codice, il quale esige la definizione esatta delle categorie che utilizza. Senza peraltro considerare che

⁴ Questa differenza è dovuta al fatto che la coscienza di un essere umano – la sua autoconsapevolezza – è un loop cognitivo (o più di essi) a ciclo continuo. Ciò è alla base della sua capacità di prendere decisioni al di là o addirittura in contrasto con le istruzioni ricevute. In una macchina, invece, un loop è il frutto di istruzioni errate e determina un ciclo infinito che blocca l'esecuzione dell'istruzione impartita.

quando cominciamo a definire insiemi, insiemi che contengono altri insiemi, e insiemi che contengono sé stessi, entriamo in un terreno paludoso dal quale solo il linguaggio naturale (informale) ci consente di uscire.

Facciamo un esempio.

Se consideriamo l'art. 16 del GDPR richiamato al precedente paragrafo, abbiamo tutte le sottoclassi di cui sopra: i *soggetti* (l'interessato e il titolare), le *relazioni* (quella tra i soggetti e quella di questi con i dati oggetto di trattamento), gli *atti* (l'esercizio del diritto di rettifica) e i *fatti* (il trattamento e la rettifica stessa). E tuttavia per tali sottoclassi non riusciremo mai a trovare una definizione soddisfacente, se non rinviando ad altre sotto-sottoclassi indefinite. Già solo per la definizione di *dato personale* (sotto-sottoclasse di *relazioni* e *atti*) non basterebbe un trattato enciclopedico. Non parliamo poi della definizione di titolare (sotto-sottoclasse di *soggetti*), apparentemente semplice, ma in realtà assai controversa. Quando poi dovremmo considerare un dato *inesatto* o un ritardo *ingiustificato* (sotto-sottoclassi di *fatti*)?



Legge=definizione dell'insieme A

Una legge, insomma, per essere eseguita (*rectius*, applicata) o semplicemente per avere una estensione finita, deve essere innanzi tutto compresa e non può quindi essere espressa in codice, ma solo in linguaggio naturale, altamente descrittivo. Solo in certi casi assai limitati una specifica disposizione potrebbe essere scritta in un linguaggio simbolico, ma pur sempre rinviando a variabili indefinibili *a priori* e destinate a rimanere tali anche *a posteriori*.

6. Quando il codice incontra la legge.

Passiamo ora dalla legge ai contratti per dimostrare che anche questi ultimi non possono essere espressi in codice, ma solo eseguiti in codice, e anche in questo caso sempre in minima parte e in via subordinata alla legge, scritta in linguaggio naturale, che li governa.

Facciamo un esempio classico della letteratura blockchain e immaginiamo di collegare la serratura della casa che ho dato in locazione ad uno smart contract che controlli il regolare pagamento del canone e che, in caso di mora, blocchi l'accesso all'immobile.

Una tale funzione si inserisce evidentemente in un contratto molto più ampio che non regola solo i pagamenti e il diritto di godimento della casa, ma anche tante altre cose, come l'individuazione delle spese ordinarie e straordinarie a carico delle parti, l'esercizio dei diritti in ordine a talune parti condominiali, la possibilità di rinnovo, recesso o risoluzione al ricorrere di determinate circostanze, l'obbligo di rilascio di fideiussione e di stipula di una assicurazione, vincoli di destinazione dell'immobile e divieto di sublocazione, sorte di addizioni e migliorie, ecc.

Solo alcuni di questi aspetti sono soggetti ad informatizzazione, e anche in questo caso, devono essere armonizzati con un ecosistema giuridico assai vasto e articolato. Infatti, il contratto come tale è sempre integrato dalle disposizioni di legge ed è quindi *eteroregolato*. Nel caso di specie, per esempio, occorre non solo fare riferimento alle specifiche disposizioni dell'istituto della locazione ad uso abitativo (L. n. 392/78 e L. n. 431/98), ma anche alle norme su obbligazioni e contratti (Libro IV del c.c.), alle disposizioni contenute nelle preleggi (su fonti e regole di interpretazione) e a quelle per l'attuazione del codice civile e disposizioni transitorie (per esempio, la responsabilità solidale del proprietario per le spese condominiali dei due anni precedenti all'acquisto). Senza contare gli aspetti fiscali con i correlati obblighi di registrazione del contratto e pagamento delle relative imposte, per finire con le eventuali questioni di giurisdizione e legge applicabile che potrebbero ricorrere.

In altri termini, il contratto non è solo una lista di proposizioni logiche, ma un articolato intreccio di rapporti tra soggetti, e tra soggetti e beni, regolati dal contratto stesso e dalle leggi applicabili (anche di diverse giurisdizioni). Si ripropongono quindi anche in ambito contrattuale, tutti gli ostacoli di riduzione formale visti a proposto della legge.

Pertanto, benché realizzabile dal punto di vista tecnico, lo smart contract sopra visto non lo sarebbe affatto dal punto di vista giuridico⁵. Esiste infatti un interesse preminente del conduttore e della sua famiglia in ordine alla detenzione dell'immobile che prevale sull'interesse del proprietario al pagamento del canone. Quest'ultimo dovrebbe semmai attivare i normali rimedi giurisdizionali dell'intimazione e sfratto per verificare tra l'altro che non esistano motivi legittimi che giustifichino il ritardo del pagamento (es.: compensazione con crediti del conduttore). Senza contare che un siffatto smart contract dovrebbe essere approvato per iscritto in modo specifico, cosa impossibile per un programma per elaboratore⁶, o essere addirittura nullo in quanto trattasi di clausola limitativa della proponibilità di eccezioni.

Andrebbe poi considerata anche la lesione dei diritti del conduttore in bilanciamento con quelli di credito del locatore: il diritto alla vita di relazione, all'incolumità, alla dignità della persona, alla proprietà di tutto quanto è in casa, ecc.. In molti casi non sarebbe solo impossibile o contrario alle leggi, ma profondamente ingiusto. Un eventuale smart contract, dovrebbe quindi essere limitato a certi aspetti (ad esempio calcolo e addebito degli interessi di mora o dell'imposta di registro), ma non dovrebbe spingersi oltre, almeno non prima di un intervento di un giudice o di un accordo tra le parti.

7. Conclusioni

Il codice non può essere né legge né contratto, ma solo una particolare e circoscritta modalità esecutiva di uno specifico comando espresso in linguaggio naturale contenuto in una legge o in un contratto. Chiamiamo questo codice *smart contract*.

Gli smart contract sono sostanzialmente materia per informatici che devono tradurre in linguaggio di programmazione determinati predicati normativi la cui esatta portata deve essere loro indicata dai giuristi.

Il ruolo di questi ultimi, infatti, è a monte quello di selezionare le attività eseguibili in codice e verificare il suo funzionamento in modo tale che rifletta in modo puntuale non solo la volontà delle parti – operazione per la quale basterebbero gli informatici – ma anche il rispetto del contesto in cui lo smart contract deve operare (ordinamento e giurisprudenza), ravvisando infine tutte le possibili casistiche che l'esperienza professionale suggerisce.

Insomma, il codice non consente di bypassare giudici e avvocati, ma di questi necessita quando va scritto o quando inevitabilmente entra in crisi.

⁵ A tale proposito vale la pena citare proprio Lessing (cit., p. 110): «*In real space, the law means you can be penalized for violating the "high/low" rule. In Second Life, you simply can't violate the 15-meter rule. The rule is part of the code. The code controls how you are in Second Life. There isn't a choice about obeying the rule or not, any more than there's a choice about obeying gravity*». Questo esempio dimostra che il codice opera su un piano diverso di quello della legge. In realtà la *15-meter rule* non è una regola, è l'effetto di applicazione di una regola decisa dai produttori di *Second Life*.

⁶ A dispetto di quello che sostiene l'art. 8-ter del Decreto Semplificazioni 2018 (D.L. 135/2018) che addirittura vorrebbe estendere la forma scritta dell'art. 2702 c.c. ai programmi per elaboratore.