LE FORME DEL FALSO NEGLI SMART CONTRACT

Silvia Crafa, Cosimo Laneve, Giovanni Sartor
Università degli Studi di Padova; Alma Mater Studiorum - Università di Bologna,
INRIA Focus; Alma Mater Studiorum - Università di Bologna, European University Institute, Italia

INTRODUZIONE

Gli smart contract (contratti intelligenti) sono, in senso ampio, accordi suscettibili di attuazione automatica che non richiedono l'intervento di intermediari.

Essi sono infatti espressi usando linguaggi di programmazione, le cui istruzioni, qualora non necessitino di nuovi input esterni, possono essere eseguite da un sistema informatico. In questa tipologia di contratti, hanno assunto un'importanza sempre maggiore gli smart contract destinati ad essere eseguiti da sistemi distribuiti basati su blockchain. In tali sistemi, dati, programmi e risultati delle elaborazioni sono registrati su sequenze di blocchi non modificabili. Queste sequenze sono replicate su tutti i nodi del sistema, così da formare il "registro replicato condiviso" (distributed ledger) che funge da riferimento per tutti i partecipanti.

Gli smart contract possono corrispondere a contratti giuridicamente vincolanti, cioè ad accordi mediante i quali le parti modificano le rispettive posizioni giuridiche, creando, modificando o trasferendo diritti ed obblighi. Ciò avviene quando i risultati prodotti automaticamente dallo smart contract corrispondono ai risultati giuridici che le parti intendono conseguire, ad esempio, un prestito di denaro (contratto di mutuo), una scommessa, la concessione del diritto d'uso su un bene (contratto di licenza d'uso, comodato), il trasferimento della proprietà di un bene (vendita), ecc.

Una delle ragioni per le quali le parti possono decidere di dare al proprio accordo la forma dello smart contract consiste nella certezza sul contenuto del contratto (garantita dall'uso di un linguaggio informatico, dotato di una semantica univoca), e dalla certezza sull'adempimento del contratto (garantita dall'esecuzione automatica).

Tali certezze possono consentire scambi tra parti che non si conoscono o comunque non si fidano l'una dell'altra, anche in assenza di terze parti fidate, che

possano prevenire gli inadempimenti o rimediare ad essi (si pensi al ruolo svolto dagli intermediari negli scambi commerciali, e più in generale alla possibilità di ricorrere ad arbitri o giudici per risolvere controversie).

Il presente contributo esamina come i meccanismi computazionali degli smart contract non possano sempre garantire l'esatta corrispondenza tra il risultato prodotto dallo smart contract e l'intenzione delle parti.

Dapprima si considerano le forme di falsità in senso ampio – insincerità o comunque inefficacia – nelle enunciazioni "costitutive", intese a creare nuovi assetti normativi o a modificare quelli esistenti.

Quindi si specifica come tali forme possano presentarsi negli smart contract, data la loro natura di enunciazioni costitutive digitalmente eseguibili. A tal fine, si esamina il modo di operare degli smart contract, che disciplinano rapporti interpersonali mediante direttive rivolte a un sistema informatico.

Si approfondisce la discussione dei limiti entro i quali sia possibile un'analisi formale, che dia sufficienti garanzie della correttezza degli smart contract.

Infine, si esamina in quale misura i limiti della espressività degli smart contract, e la necessità di far riferimento a dati forniti da fonti esterne, possano consentire al falso di influire sull'operatività degli smart contract.

ENUNCIAZIONI PERFORMATIVE E CONTRATTI

Gli smart contract sono accordi il cui contenuto può essere eseguito automaticamente da un sistema informatico. Secondo la teoria degli atti linguistici, il linguaggio non si limita a descrivere il mondo, ma può tendere a modificarlo: si possono "fare cose con le parole" (1). In particolare, Searle ha distinto "due direzioni di adattamento" (directions of fit) tra linguaggio e mondo: dal mondo alla parola – *world-to-word* – e dalla parola al mondo – *word-to-world* – (11, 12).

Nel primo caso il parlante intende far sì che il messaggio linguistico si adegui al mondo, cioè a far sì che il contenuto del suo asserto corrisponda a quanto esiste indipendentemente dall'asserto stesso. È questo il caso per le enunciazioni che Searle chiama assertive, quelle tese a descrivere oggetti e situazioni.

Si consideri per esempio l'asserto secondo il quale Tizio, una delle parti di un contratto (es. una compravendita), è maggiorenne, o l'asserto secondo il quale l'oggetto del contratto è in buone condizioni.

Alle enunciazioni che hanno una direzione di adattamento dal mondo alla parola, si contrappongono quelle che hanno invece una direzione dalla parola al mondo. Mediante queste ultime il parlante tende a modificare la realtà sociale (giuridico-istituzionale). Questa direzione, secondo Searle, è comune alle enunciazioni che egli chiama direttive (ordini: "Fai…!"), commissive (promesse: "Mi impegno a fare…!"), e dichiarative (enunciazioni costitutive: "Ti regalo…", "Da oggi anche tu sei socio dell'associazione…", "Ti nomino vincitore di…").

Nel presente contributo, dedicato ai contratti, ci limitiamo a considerare promesse e dichiarazioni quali possibili contenuti di dichiarazioni contrattuali.

Coi contratti, infatti, le parti possono assumere obblighi di comportamento nei confronti della controparte (e.g., l'obbligo di consegnare un bene, o di pagare un prezzo), e realizzare costitutivamente dei cambiamenti nelle loro posizioni giuridiche (e.g., trasferire la proprietà di un bene dall'uno all'altro, concedere una licenza per l'uso di un software, ecc.).

Possiamo interpretare anche gli enunciati promissori secondo la prospettiva della realizzazione costitutiva. L'effetto di una promessa è la creazione dell'obbligo di adempiere alla stessa. Consideriamo per esempio il contratto che si forma con la proposta di Tizio "Caia, ti cedo la mia automobile se tu mi dai 3000 euro, 1000 oggi e 2000 entro dieci giorni" e con l'accettazione di Caia "D'accordo, accetto". Possiamo riformulare il contratto nella forma della seguente dichiarazione performativa: "Con la presente dichiarazione il sottoscritto Tizio vende questa automobile e in cambio la sottoscritta Caia gli consegnerà 1000 Euro di acconto subito e 2000 Euro entro 10 giorni a saldo". Tale enunciazione può essere vista come una dichiarazione costitutiva intesa a realizzare due effetti giuridici: il trasferimento della proprietà dell'automobile da Tizio a Caia, e la creazione dell'obbligo, in capo a Caia, di pagare subito 1000 euro a Tizio e il resto entro 10 giorni. Passiamo ora a considerare in quale senso la suddetta dichiarazione possa dirsi falsa. Trattandosi di una dichiarazione costitutiva, con direzione word-to-world (tesa a modificare la realtà istituzionale), non possiamo determinarne la verità considerando come stian le cose nel mondo nel momento in cui essa viene compiuta. Quindi non è una enunciazione che possa dirsi vera o falsa in senso stretto, cioè nel senso di una sua corrispondenza alla realtà di fatto. Tuttavia, la dichiarazione potrebbe non essere sincera.

Per esempio, Tizio potrebbe dichiarare di vendere un'automobile non sua, per rendersi irreperibile una volta ricevuto l'acconto.

Oppure la dichiarazione potrebbe essere in tutto o in parte inefficace, nel senso di non realizzare il risultato atteso. Ad esempio, Tizio potrebbe erroneamente ritenere di essere il proprietario dell'automobile, che invece appartiene ad altra persona. In questo caso, non si effettuerebbe il trasferimento della proprietà della stessa. Oppure, pur essendosi sinceramente impegnata a pagare il prezzo, Caia potrebbe scoprire che il suo conto in banca è purtroppo a zero, così da non essere in grado di pagare il debito appena contratto.

Quindi nel primo caso, la dichiarazione sarebbe stata "falsa" in quanto insincera, cioè effettuata senza l'intenzione che ne seguissero gli effetti cui essa appare diretta (anzi nella consapevolezza che quegli effetti non si potranno realizzare per la mancanza dei loro presupposti). Negli altri casi la dichiarazione sarebbe stata "falsa" in quanto inefficace, pur essendo stata compiuta nell'erronea credenza che ne esistessero i presupposti o che gli obblighi da essa statuita sarebbero stati adempiuti.

Nelle sezioni seguenti esamineremo la tecnologia degli smart contract e se essi possono dar luogo a falsi nelle dichiarazioni costitutive, interrogandoci sui limiti e le implicazioni di questo fatto.

LA DIGITALIZZAZIONE E GLI SMART CONTRACT

Come detto in precedenza, gli smart contract sono programmi informatici intesi a dare attuazione all'accordo tra le parti, senza l'intervento di intermediari, e che vengono eseguiti su blockchain, cioè su una sorta di computer globale decentralizzato, il cui funzionamento è garantito dal controllo contemporaneo di migliaia di utenti. Prima di tutto quindi gli smart contract sono programmi software, cioè sequenze di istruzioni scritte in uno specifico linguaggio (di programmazione), che per sua natura è conciso e non ambiguo in modo da indicare con esattezza quali calcoli deve effettuare una macchina. La blockchain funge da memoria non modificabile di questo computer globale, quale libro mastro visibile a tutti, che sintetizza lo stato del mondo (gli scambi e interazioni tra le parti) contabilizzandolo in termini di dati e transazioni.

Nei termini di Searle, possiamo dire dunque che uno smart contract raccoglie delle enunciazioni direttive e al tempo stesso costitutive che modificano quella rappresentazione del mondo che è la blockchain. Si tratta di enunciazioni

direttive in quanto comandi diretti verso la blockchain, e al tempo stesso costitutive, in quanto l'esecuzione di tali comandi dà luogo alle modificazioni da essi stabilite senza intervento umano. Le istruzioni inserite negli smart contract possono essere anche di tipo commissivo (tra una settimana eseguirò…) e di tipo assertivo (il possessore di questo account è…), ma la natura eseguibile del software (o meglio, la sua esecuzione automatica) trasforma ogni promessa e dichiarazione in una direttiva per la macchina, che dà luogo automaticamente alle modifiche stabilite. D'altra parte, la distinzione di Searle diventa particolarmente interessante nello scarto tra il mondo e la sua digitalizzazione.

La rivoluzione digitale prevede infatti di digitalizzare sempre più ogni processo reale, al fine di rendere il processo sempre più efficiente e automatico (persino autonomo), tramite software e algoritmi sempre più sofisticati. Nascono dunque due nuove relazioni, che coinvolgono le parole del linguaggio digitale: *world-to-code* e *code-to-world*.

La prima direzione, dal mondo al software, in analogia con le enunciazioni assertive di Searle, tese a descrivere oggetti e situazioni, ha a che fare con il processo di datificazione della realtà che il software intende modificare.

Ad esempio, nel sistema degli smart contract e della blockchain, è necessario "datificare Tizio e Caia", cioè è necessario un sistema di identità digitali (più o meno anonime a piacere), la cui creazione e gestione apre a numerose questioni relative al concetto di verità e falsità (e.g., furto di identità, sostituzione di persona, errore o omissione nell'identità digitale). Il caso delle enunciazioni che hanno una direzione *code-to-world* diventa ancora più delicato: rispetto al linguaggio parlato, il software inserito in un sistema socio-tecnologico ha un enorme potere performativo sulla realtà. L'ampia discussione attuale sulla non neutralità della tecnologia digitale e il suo impatto sulla società e sui comportamenti delle persone rientra in questa direzione. Ma anche circostanziando il discorso al solo caso degli smart contract, possiamo notare come l'esecuzione automatica del loro codice sia in grado a tutti gli effetti di modificare la realtà sociale, ad esempio autorizzando o meno un accesso ad un servizio di affitto di smart bike in città, oppure rendendo non più accessibile la criptovaluta contenuta in un certo portafoglio (wallet) digitale. Inoltre, la rapida evoluzione degli smart contract (e in parallelo la crescente digitalizzazione della pubblica amministrazione) sta rendendo il codice software sempre più capace di modificare anche la realtà giuridica-istituzionale, tramite la digitalizzazione di contratti legali (smart legal

contract) (2, 3, 7, 13). È interessante inoltre osservare che gli smart contract trovano la loro principale applicazione come intermediazione tra due parti che non si fidano l'una dell'altra. La questione della fiducia è infatti al cuore della tecnologia blockchain, nata proprio per eliminare il ruolo di intermediario di fiducia tradizionalmente svolto dalle istituzioni. Uno sguardo più consapevole ci fa oggi affermare che, più che eliminare la fiducia tramite una *dis-intermediazione*, la blockchain effettua una *alter-mediazione*, riassegnando la fiducia a tutta una serie di soggetti (miner, programmatori, fondazioni, aziende di servizi) che, in modo più opaco, realizzano, gestiscono e abilitano il funzionamento di questa piattaforma tecnologica. Ciò non toglie che, quando Tizio e Caia non si fidano l'un l'altro, l'uso degli smart contract rappresenta una soluzione efficace grazie all'esecuzione automatica delle clausole previste, ad esempio in una scommessa.

D'altra parte, l'automatismo dell'esecuzione e dei suoi risultati (cioè l'effetto della direzione *code-to-world*) può rappresentare un problema quando sussista un errore nelle premesse del contratto (cioè nella direzione *world-to-code*).

Rimanendo nel caso della scommessa tra Tizio e Caia, se c'è stato un disallineamento tra i termini della scommessa su cui si erano accordati e la traduzione in codice dello smart contract di tali termini (ad es. un errore di programmazione), possiamo dire che lo smart contract è una dichiarazione costitutiva "falsa"?

Approfondiremo questa domanda nella prossima sezione, limitandoci qui ad osservare che per dirimere le controversie, la visione originale della blockchain – basata sul dogma secondo cui la fiducia è programmata (*hardwired*) negli algoritmi di intermediazione – ricorre all'approccio *code-is-law*, cioè è il codice dello smart contract, che è sempre pubblicamente disponibile, che fa fede nell'accordo delle parti. In quest'ottica un problema nella programmazione, che determina un comportamento inatteso dello smart contract, rappresenta una *caratteristica* del codice e non un errore. Come insegna l'esperienza concreta, ovviamente questo approccio non è soddisfacente nella pratica: quando sono in gioco ingenti volumi di denaro, nessuno è davvero disponibile a considerare un errore di sicurezza nella programmazione come parte del contratto che ha sottoscritto. Ancora una volta, cosa è vero e cosa è falso, in un contratto così come nella sua versione digitalizzata, non è facilmente caratterizzabile.

Di certo la traduzione di intenzioni, promesse, azioni e oggetti in codice informatico, per quanto pubblico e non ambiguo – per la macchina – non risolve il problema, ma lo sposta in un'altra dimensione.

SMART CONTRACT INSINCERI O INEFFICACI?

Entriamo ora nel dettaglio del tema della falsità degli smart contract, quali enunciazioni costitutive digitalizzate e eseguibili. Come discusso in precedenza, una dichiarazione costitutiva non può considerarsi vera o falsa in relazione alla sua corrispondenza alla realtà di fatto, ma piuttosto può essere "falsa" in quanto insincera, cioè fatta nella consapevolezza che gli effetti dichiarati non si potranno realizzare, oppure in quanto inefficace, cioè compiuta nell'erronea credenza di poterne realizzare i risultati.

Insincerità e inefficacia possono presentarsi in diversi modi nel contesto degli smart contract, che per la loro natura presentano due aspetti specifici. Innanzitutto, il contenuto contrattuale è espresso mediante un programma informatico, che può non rappresentare compiutamente ed esattamente le intenzioni di entrambe le parti (specialmente quando non siano dotate di competenze tecniche approfondite). Inoltre, i contratti formulati nella forma di smart contract si caratterizzano per un intreccio di azioni e conseguenze sul registro condiviso, cioè sulla catena dei blocchi (*on-chain*) e al di fuori di esso (*off-chain*). Ad esempio, la vendita di un'automobile coinvolge il trasferimento della proprietà di un bene fisico (off-chain) in cambio della promessa di trasferimento di denaro in criptovaluta (on-chain). Alcune pattuizioni contrattuali – come i trasferimenti di criptovalute o di altri beni digitali – possono essere direttamente attuabili on-chain. In questo caso vi sarà la garanzia della loro realizzazione, sempre che i beni da trasferire siano resi disponibili al contratto. Anche le dichiarazioni costitutive che mirano a modificare la fruibilità di oggetti materiali, come la disponibilità di un'automobile o di una casa in affitto temporaneo, possono trovare altresì attuazione automatica grazie all'uso di oggetti digitali, i cosiddetti *token* (es. codici univoci per accedere a servizi, NFT o altri tipi di beni digitali sia fungibili che non fungibili), messi a disposizione dalla tecnologia blockchain (6). Questi token non sono altro che specifici smart contract, scritti secondo un preciso standard, che utilizzano lo stato della blockchain per realizzare una precisa gestione dell'accesso, eventualmente esclusivo o in qualche modo limitato o condizionato, a beni e servizi. Un altro elemento caratteristico di questa tecnologia che mescola elementi on-chain e off-chain è la presenza dei cosiddetti *oracoli*, cioè dei servizi web (software accessibile via internet) fruibili automaticamente dagli smart contract per recuperare informazioni sul mondo

reale da usare per verificare le condizioni contrattuali. Ad esempio, uno smart contract finanziario può modificare la sua esecuzione a seconda del valore di un certo bene finanziario in un dato giorno dell'anno, oppure uno smart contract assicurativo può far scattare automaticamente un indennizzo per il ritardo di un volo aereo. In questi casi la veridicità dello smart contract – la corrispondenza delle sue premesse e operazioni all'intenzione delle parti – è strettamente legata alla verità, i.e., la correttezza dell'informazione fornita dall'oracolo, per esempio, sul valore del bene o sull'effettivo ritardo dell'aereo, cioè è legato alla verità degli oracoli. Il fatto che la verità di un contratto automatico dipenda dalla verità di un servizio software esterno è particolarmente critico per una tecnologia che mira a rimuovere gli intermediari di fiducia.

Oltre al problema della verità dei dati – e quindi delle enunciazioni assertive – che stanno alla base delle esecuzioni automatiche, esistono diverse forme di falsità che dipendono dal fatto che uno smart contract non può realizzare in modo completamente automatico quelle prestazioni che richiedono uno specifico comportamento delle parti o di terzi.

Insincerità e inefficacia possono presentarsi negli smart contract innanzitutto quando le condizioni previste per la realizzazione del contratto dipendono da attività successiva delle parti. Infatti, nonostante l'automaticità dell'esecuzione delle clausole dello smart contract, insincerità e inefficacia potrebbero verificarsi quando una parte non mette a disposizione, o sottrae dal proprio conto, le somme o i token necessari all'esecuzione del contratto. Inefficacia o insincerità potrebbero dipendere altresì dal fatto che una parte non voglia o possa realizzare quanto previsto off-chain, come la chiusura manuale di un lucchetto o il mantenimento della riservatezza di qualche dato.

Un diverso tipo di falsità riguarda la possibile divergenza tra gli effetti ottenuti dell'esecuzione del contratto e quanto una (o più) delle parti riteneva si sarebbe dovuto verificare. Una piena comprensione del contenuto e degli effetti di un contratto è già difficile con i contratti scritti in linguaggio naturale, ma la codifica del contratto in un software scritto in un linguaggio di programmazione, per quanto non ambiguo e perfettamente ispezionabile, resta lontano dall'essere pienamente trasparente ed intellegibile. Ci può quindi essere insincerità, quando si inganna una controparte sugli effetti automatici del contratto, facendole ritenere che l'esecuzione automatica avrebbe avuto effetti diversi da quelli che invece avrà (e.g. viene calcolato un interesse diverso da quello concordato a voce).

Oppure più facilmente ci può essere inefficacia perché lo smart contract contiene un errore di programmazione che non era stato rilevato e che determina l'interruzione dell'esecuzione in modo brusco e non previsto. Ancora più sottile è il caso di un errore di programmazione di tipo logico, cioè il caso in cui l'esecuzione automatica procede e giunge al termine, ma il risultato prodotto non è quello atteso perché era stata inserita qualche istruzione sintatticamente corretta ma logicamente errata, e.g. un errato controllo di una condizione oppure un errore nella descrizione di un calcolo scambiando una somma con una sottrazione. In altri termini, la falsità di un'enunciazione costitutiva digitale – nel senso della sua divergenza dall'intenzione delle parti – può dipendere dagli errori di programmazione, e quindi la fiducia nel contratto dipende dalla fiducia nella loro assenza.

Un ultimo tipo di falsità, ancora una volta tipico della scrittura del software, è quello in cui lo smart contract non contiene apparentemente errori logici e dà luogo ad esecuzioni normalmente corrette e conformi alle attese, ma può rispondere in modo anomalo e inatteso a certi input. Una parte malevola o un terzo possono approfittare di questo fatto, inducendo il comportamento anomalo e sfruttandolo a proprio vantaggio. In questo caso cioè lo smart contract corrisponde effettivamente al contratto concordato e formulato dalle parti, ma il suo funzionamento lascia spazio a comportamenti che non corrispondono alle intenzioni delle stesse. In altri termini, il codice contiene un problema di sicurezza che lo espone ad attacchi.

Questo è quanto si è verificato nel 2016 allo smart contract DAO (10), in cui l'attaccante ha sfruttato una peculiarità di Solidity (5), il linguaggio di programmazione della blockchain Ethereum (nello specifico l'esecuzione automatica di una funzionalità di default – il fallback), per sottrarre diversi milioni di euro (nella criptovaluta Ether) durante un procedimento automatizzato di crowdfunding (raccolta di fondi).

La possibilità di esecuzione automatica delle statuizioni contrattuali non garantisce l'assenza di errori di programmazione o di comportamenti inattesi del codice. Anzi, rende più difficile rimediare agli errori, soprattutto nei contesti in cui tipicamente si ricorre agli smart contract, cioè in quelli in cui manchi la fiducia reciproca e verso gli intermediari. Per questo è importante che la scrittura degli smart contract sia accompagnata dall'uso di strumenti informatici utili alla *verifica* e all'analisi della loro correttezza.

ANALISI DI SMART CONTRACT

In generale, non è possibile definire un procedimento automatico – un algoritmo – che prenda in input un programma e ne individui con esattezza tutti gli eventuali comportamenti scorretti o, meglio, non conformi a una certa intenzione o specifica. Questo asserto esprime un teorema ben noto agli informatici (il teorema di Rice) che, seppur incontrovertibile, ammette comunque la possibilità di definire algoritmi *imprecisi*, cioè che, tramite approssimazione, possano ritornare risultati erronei ma in qualche modo informativi.

Se si accetta che gli algoritmi di verifica possono compiere errori di valutazione, allora la dimensione dei linguaggi di programmazione diventa abilitante per una tecnica che non è invece possibile con i linguaggi naturali con cui sono tradizionalmente scritti i contratti legali. Chiariamo la questione con un esempio.

Supponiamo di avere scritto uno smart contract che sposta Bitcoin da un attore A ad un attore B, ed entrambi gli attori intendono evitare bonifici maggiori di 1000 euro. Poiché una volta abilitato (tramite dispiegamento sulla blockchain) lo smart contract diventa definitivo e non più modificabile, prima di tale attivazione gli attori in gioco vorrebbero verificare l'assenza di possibili esecuzioni che effettuano bonifici superiori a 1000 euro. Osserviamo che, in generale, per garantire questa proprietà non è sufficiente ispezionare il codice, perché l'entità del bonifico potrebbe essere descritta da una variabile il cui valore dipende da svariate condizioni. Si potrebbe quindi progettare un algoritmo V che prende in input uno smart contract SC e verifica se tutte le esecuzioni possibili di SC prevedono solo bonifici al di sotto di 1000 euro oppure no. Per quanto realizzabile, per il teorema di Rice, V sarebbe necessariamente impreciso, cioè potrebbe essere il caso che:

a. V affermi la correttezza di SC (le sue esecuzioni hanno solo bonifici al di sotto di 1000 euro) ma esiste almeno un'esecuzione con bonifici al di sopra di 1000 euro. Questo caso è detto falso positivo.
b. V affermi che SC non è corretto (esiste almeno una esecuzione con bonifici al di sopra di 1000 euro) ma tutte le sue esecuzioni hanno in realtà bonifici al di sotto di 1000. Questo caso è detto falso negativo.

Evidentemente, gli attori A e B si sentirebbero garantiti da un verificatore V che ammette falsi negativi ma che esclude falsi positivi. Cioè, da un verificatore V che quando dichiara che SC è corretto, allora davvero non ci saranno mai

esecuzioni con bonifici al di sopra di 1000 euro. Rispetto a quanto detto nella sezione precedente, il passaggio al linguaggio digitale consente di passare da una questione di *fiducia* ad una questione di *garanzia*. Mentre per un sistema informale, come tutti quelli basati su linguaggi naturali, questa garanzia non può essere data e per redimere controversie interpretative bisogna ricorrere alla giurisprudenza, così non è per un sistema formale, come gli smart contract e i linguaggi di programmazione, in generale.

Questi sistemi, possedendo una semantica precisa e non ambigua, determinata attraverso tecniche matematiche oppure dalla loro mera esecuzione su un calcolatore, ammettono la possibilità di sviluppare tecniche formali per cui quella garanzia può essere ottenuta attraverso *dimostrazioni* matematicamente solide.

La costruzione di verificatori e la dimostrazione delle loro proprietà sono oggetto di ricerca, a partire dal fondamentale lavoro di Hoare negli anni Sessanta (8). Molte tecniche formali sono state sviluppate per l'analisi di sistemi, dando luogo a numerosi risultati e aprendo importanti nuovi filoni di ricerca per continuare ad espandere l'espressività e la precisione dei verificatori realizzabili.

Nel contesto della blockchain, una formalizzazione della semantica del nocciolo del linguaggio Solidity è stata definita in (4), mentre un esempio di verificatore che consente di controllare i movimenti di criptovalute in maniera automatica ed in cui il risultato è garantito matematicamente è stato definito in (9).

Per riassumere, i linguaggi formali in cui le frasi – i programmi e gli smart contract, in particolare – hanno una semantica precisa, possono sottendere (lo sviluppo di) sistemi in grado di individuare proprietà rilevanti o eventuali errori in maniera automatizzabile e matematicamente corretta. In questo contesto, la controversia è considerevolmente ridotta e, qualora l'algoritmo di verifica di una proprietà non ammetta falsi positivi, la verità diventa un concetto assoluto.

ESPRESSIVITÀ DEGLI SMART CONTRACT

Mentre le applicazioni di smart contract riguardano di solito transazioni finanziarie (con trasferimenti di criptovalute), recentemente sono state sviluppate applicazioni che esprimono gli elementi caratteristici dei contratti legali considerando che, secondo i moderni sistemi legislativi, vale il principio della "libertà della forma", cioè le parti sono libere di definire il loro accordo nel linguaggio che preferiscono. In questa sezione discutiamo come siano rappresentabili in

termini di smart contract gli elementi caratteristici dei contratti legali. Per una trattazione completa si guardi (3).

Un semplice, e molto usato, contratto legale è il comodato d'uso, in cui un bene, ad esempio un armadietto in palestra, viene concesso per uso gratuito per un determinato tempo. La trasposizione in smart contract di un tale contratto già presenta tutte le caratteristiche descritte in precedenza. Avendo come oggetto un bene fisico e il suo uso, l'esecuzione del contratto coinvolge sia elementi on-chain che off-chain. La rappresentazione su blockchain della disponibilità dell'armadietto e della possibilità di accedervi richiede l'uso di soluzioni tecnologiche come uno lucchetto intelligente (smart lock) o altri dispositivi IoT il cui codice di accesso può essere associato ad un token memorizzato su blockchain.

Di conseguenza, la verità delle enunciazioni costitutive che scattano con la consegna del bene (il comodatario può usare l'armadietto ed il comodante non può impedirglielo) dipendono dall'effettivo funzionamento delle soluzioni tecnologiche adottate.

Meno problematica risulta invece la trasposizione in smart contract di permessi e divieti, che si rappresentano in termini di esecuzione opportunamente condizionata di corrispondenti funzionalità del software. Anche la promessa, cioè la creazione di un obbligo, come la restituzione dell'armadietto entro il tempo limite, si può tradurre impostando un controllo software automatico che, allo scadere del tempo limite, verifica lo stato del bene – così come rappresentato sulla blockchain – e adotta le conseguenze opportune. Vale a dire che la digitalizzazione degli enunciati promissori comporta la definizione di una specifica procedura che controlla, nel momento opportuno, l'assolvimento dell'obbligo associato, e in caso di promessa non mantenuta mette in atto una specifica contromisura, e.g., trattiene denaro (in criptovaluta) o blocca l'accesso al bene.

I maggiori limiti degli smart contract nella gestione automatica di un contratto di comodato hanno a che fare con la gestione delle violazioni previste dalle norme giuridiche. Ad esempio, se il comodatario viola i suoi obblighi (di custodia e cura con diligenza, di uso del bene solo secondo quanto previsto, il divieto di concessione del godimento ad un terzo senza consenso del comodante); il comodante può chiedere l'immediata restituzione della cosa, oltre al risarcimento del danno (art. 1804). Viceversa, se il bene aveva dei vizi che recano danni al comodatario e il comodante lo sapeva ma non ha avvertito, il comodante deve risarcire il comodatario (art. 1812).

Chiaramente, violazioni sulla custodia e cura con diligenza, così come la concessione del godimento a terzi o la presenza di vizi del bene, sono molto difficilmente controllabili mediante un software. Anche in questo caso soluzioni tecnologiche di sorveglianza e monitoraggio potrebbero essere adottate, ma non risolverebbero completamente il problema.

Riassumendo, le criticità – e dunque i limiti della veridicità degli smart contract – restano legate al complesso intreccio tra elementi on-chain e off-chain.

Osserviamo che i limiti della capacità della blockchain di descrivere correttamente lo stato delle cose si lega strettamente al problema della verità delle enunciazioni assertive e alla problematicità della relazione world-to-code (e world-to-word) di cui abbiamo già discusso.

CONCLUSIONI

Nel nostro contributo abbiamo esaminato come si possa parlare, seppure in senso esteso, di verità e falsità di enunciazioni tese a modificare la realtà (con direzione word-to-world), come le enunciazioni direttive, commissive e costitutive. In questi casi il concetto di falsità può essere ricondotto a quelli di insincerità e di inefficacia.

Abbiamo poi esaminato come questa tematica si configuri in modo nuovo rispetto agli smart contract, che sembrano risolvere i problemi inerenti all'insincerità e all'inefficacia grazie all'esecuzione automatica delle clausole contrattuali, che diventano direttive rivolte al sistema informatico, la cui esecuzione costituisce i risultati attesi dalle parti. Dal nostro esame è emerso come problemi di sincerità ed effettività si possano manifestare in modi nuovi, correlati al mezzo utilizzato per esprimere l'accordo (linguaggi di programmazione) e per dare attuazione ad esso (il sistema informatico distribuito). Importanti differenze sono emerse tra prestazioni da eseguirsi all'interno del sistema informatico (on-chain) o al di fuori di esso (off-chain).

Abbiamo quindi considerato come il tema della fiducia, per taluni versi superato dagli smart contract, possa ripresentarsi in forme nuove, e come strumenti di verifica automatica possano fornire garanzie di correttezza.

In conclusione, gli smart contract non eliminano il problema della falsità ma piuttosto ne ridefiniscono le forme, e richiedono nuovi metodi per affrontarlo.

Bibliografia

1. J.L. Austin, *How to do things with words*, Oxford, Oxford University Press 1962.
2. Open Source Contributors 2018. The Accord Project. https://accordproject.org.
3. S. Crafa, C. Laneve, G. Sartor, Pacta sunt servanda: legal contracts in Stipula. arXiv:2110.11069 Ottobre 2021. https://arxiv.org/abs/2110.11069.
4. S. Crafa, et al., *Is Solidity solid enough?*, in *Lecture Notes in Computer Science*, n. 11599 (2020), pp. 138-153.
5. C. Dannen, *Introducing ethereum and solidity: Foundations of cryptocurrency and blockchain programming for beginners*, New York, Apress 2017.
6. Ethereum Foundation 2015-21. Token Standards. https://ethereum.org/en/developers/docs/standards/tokens/.
7. Lexon Foundation 2019. Lexon Home Page. http://www.lexon.tech.
8. Hoare, C.A.R., *An axiomatic basis for computer programming*, in *Commun. ACM.*, vol. 12, n. 10 (1969), pp. 576-580.
9. C. Laneve, C. Sacerdoti Coen, *Analysis of smart contracts balances*, in *Blockchain: Research and Applications*, 2021.
10. I. Mehar, et al., *Understanding a revolutionary and flawed grand experiment in blockchain: The DAO Attack*, in *Journal of Cases on Information Technology*, n. 21 (Gennaio 2019), pp. 19-32.
11. J.R. Searle, *Speech acts: An essay in the philosophy of language*, Cambridge, Cambridge University Press 1969.
12. J.R. Searle, D. Vandervecken, *Foundations of illucutionary logic*, Cambridge, Cambridge University Press 1985.
13. A. Wright, et al., OpenLaw Web Site, 2019. https://www.openlaw.io.