

Contents

Einleitung	2
Dezentrale Verifizierung	3
Digitale Geteiltes Objekt	4
Repräsentation	5
Akteure	5
Besitzverteilung	5
Kandidaten	5
Transaktionen	5
Konsens	6
Bewertung	6
Qualitätskriterien	6
Optimierung	6
Optimierung der Kandidatenmenge	6
Grammatikerweiterung S	7
Morphismus	7
Abstimmungen	9
2. Manipulation	9
Qualitätskritären	9
States	9
Transitionen	10
Delegationsprogrammierung	11
zu untersuchen	11
Umfang der Arbeit	11
Aussicht	12
Dezentralisierung	12
Quellen	13

Einleitung

Im Internet werden zunehmend Inhalte kollaborativ erzeugt. Dabei entsteht ein Inhalt durch Beiträge einzelner **Akteure**. Zentral ist die Frage nach dem **Besitz** des Inhaltes sowie die **Bewertung** der einzelnen Beiträge durch die Besitzer und damit ihren Anteil am **Gesamtergebnis**.

In dieser Arbeit möchte ich eine Meta-Programmiersprache $S(G)$ für eine beliebige eindeutige kontextfreie Grammatik G herausarbeiten, die die verteilte Steuerung Contextfreier Inhalte ermöglichen soll. Hierfür werden in (1) zwei Funktion angegeben. Eine erweitert eine gegebene kontextfreie Grammatik G zu einer $S(G)$, eine Konsensfunktion erzeugt aus einem Wort der Sprache $L(S(G))$, ein Wort in der ursprünglichen Sprache $L(G)$. Manipulationen eines Wortes einer $L(S(G))$ -Sprache sind bedingungen unterworfen, die in (2) angegeben sind.

Dezentrale Verifizierung

Das auf dem Bitcoin-Protokoll aufbauende Ethereum [Woo14], beschreibt ein Protokoll in einem Netzwerk zur dezentralen Verifizierung von turing-vollständigen Programmausführungen. Es stellt einen gebildeten Konsens von Programmen mit Aktuellen Zuständen bereit. Neue Inhalte wie neue Programme oder neue Interaktionen von Akteuren mit Programmen werden nach einer veröffentlichung und Validierung der Ausführung durch das Netzwerk mit dem neu berechnetem Zustand in den Konsens aufgenommen.

Akteure werden durch ein 20-byte Adresse identifiziert. Ihre Interaktion mit dem Netzwerk ist durch ein asymmetrisches Kryptosystem gesichert.

Das Netzwerk finanziert sich durch die Validierung der Programme. Für jeden Berechnungsschritt wird ein kleiner Betrag vom Interagierenden Akteur erhoben. Ist nicht genügend Wert verfügbar, terminiert die Transaktion und wird vom Netzwerk abgelehnt. So wird auch das Halteproblem umgangen.

Ein Beispiel für ein Programm währe eine dezentrale Währung, wie sie derzeit vom Bitcoin Protokoll repräsentiert wird (Quelle [But14]):

```
from = msg.sender
to = msg.data[0]
value = msg.data[1]

if self.storage[from] >= value:
    self.storage[from] = self.storage[from] - value
    self.storage[to] = self.storage[to] + value
```

Digitale Geteiltes Objekt

Digitale Geteilte Objekte müssen in digitaler repräsentation vorliegen, sowie für die Anteilseigner in verschlüsselter oder unverschlüsselter form zugänglich sein.

Beispiele wären:

- medialen Inhalte wie Bilder, Texte oder Videos
- Programme und contextuelle Angaben wie deren Compiler oder die Architektur
- Besitzrepräsentation wie Bitcoin Anteile oder Aktien

Sie können einerseits in direkter Form, oder als Referenz unter angabe einer *URI* angegeben werden.

Andererseits können digitale geteilte Objekte dynamische Inhalte repräsentieren, welche eine Syntax besitzen und durch ein Compiler Interpretiert werden können. Da *URI*'s durch reguläre Sprachen (*REG*) konstruiert werden können und $L(REG) \subsetneq L(CFG)$ ist. Reicht es zur betrachtung aller digitaler Objekte solche zu betrachten, die durch eine *CFG* Syntax besitzen, oder durch eine Solche referenziert werden können.

Ein **geteiltes digitales Objekt** ist ein Tupel $O_G = (A, K, w, rate, consens)$ bestehend aus:

1. Eine endliche Menge von Akteuren A
2. Eine eindeutige **Besitzverteilung** von Akteuren zum Objekt

$$w : A \rightarrow \mathbb{N}$$

3. eine endliche Menge von validen Kandidaten mit mindestes einem Element

$$K_G \subseteq \{w|S \rightarrow_P^* w \wedge w \in T^*\} \wedge |K_G| \geq 1$$

4. eine Bewertung der Kandidaten durch die Akteure.

$$rate : A \times K \rightarrow [0, 1]$$

5. einer Konsensfunktion

$$consens : A \times K \times w \times rate \rightarrow L(G)$$

Sei **SOBJ_G** die Menge aller geteilten Objekte in der Grammatik G sowie **KAND_G** $:= \{K_G | K_G \subseteq L(G)\}$ die Menge aller möglichen Kandidatenmengen.

Repräsentation

Akteure

Da ein Geteiltes Objekt auch Werte in Form einer Kryptowährung wie Bitcoin beinhalten kann, strategische wertvolle Attribute in einem Netzwerk besitzt, wie Reputation oder Namensrechte, geschlossene Informationen beinhaltet auf die nur die Eigentümer zugreifen können oder Gewinnüberschüsse an die Besitzer ausschüttet, besitzt das Objekt und somit auch seine Anteile auf einem freien Markt einen realen und spekulativen Wert. Somit können wir davon ausgehen, dass es im Interesse jedes rationalen Akteurs ist, seinen Besitzanteil zu maximieren. (**AMAX**)

Besitzverteilung

Die Besitzverteilung wird ähnlich dem Aktienmarkt Modelliert. Dabei hat ein geteiltes Objekt eine bestimmte Anzahl an Teilen (geschrieben $|O| \in \mathbb{N}$), die unter den Akteuren aufgeteilt sind.

$$|O| := \sum_{a \in A} w(a)$$

Der Besitz eines Objekts O wird durch das Recht definiert, auf dieses zugreifen zu können und es kontrollieren zu können [Wal04]. Besitzen mehrere Akteure ein Objekt, so gibt der Anteil am Objekt an, zu welcher Gewichtung jeder einzelne Akteur über das Objekt mitbestimmen kann.

Eine Konsensfunktion entscheidet über die Ausführung der Kontrolle. Dabei kann eine Zustimmung von $\frac{1}{2}|O|$ eine Ausführung bedingen. Man spricht von einer **majorität**. Eine Zustimmung von $\frac{2}{3}|O|$ heißt **super majorität**.

Kandidaten

Eine wichtige Frage ist, ob alle syntaktisch validen Kandidaten zugelassen werden, oder nur semantisch valide. Falls nur semantisch valide zugelassen werden stellt sich die Frage wo geschieht diese Validierung.

Transaktionen

Transaktionen sind Manipulationen des derzeitigen Zustandes des Objektes. Sie werden **immer** von einem Akteur a ausgelöst und besitzen die Form (a, O') . O' ist dabei das manipulierte geteilte Objekt.

Wir betrachten vorerst nur die Manipulation der Kandidatenmenge K sowie der Kandidatenbewertung *rate*.

Konsens

Bewertung

Qualitätskriterien

Für die Implementation ist auf folgende Qualitätskriterien zu achten:

RESMIN:

Es sind möglichst wenig Ressourcen (Speicher und Rechenleistung) vom Ethereum-Netzwerk notwendig, um Transaktionen zu validieren.

INTMIN:

Eine Akteur soll möglichst wenig Interaktionen benötigen, um auf eine gewünschte, von ihm erreichbare Manipulation zu kommen.

Optimierung

Optimierung der Kandidatenmenge

Die Idee hinter der Optimierung ist nah am Parsebaum der Grammatik zu arbeiten und die Grammatik so zu erweitern, dass die Wörter der Kandidatenmenge in einem Wort der erweiterten Grammatik kodiert werden können. Redundant auftauchende Stellen zwischen den Wörtern werden zusammengefasst. An jedem Ableitungsschritt wird eine Mehrdeutigkeit zugelassen. Die Kandidatenmenge ergibt sich durch die verschiedenen Kombination der mehrdeutigen Ableitung.

Dazu muss es eine Grammatikerweiterung $S : CFG \rightarrow CFG$ gefunden werden. Um zu zeigen, dass es sich bei S um die gesuchte Funktion handelt, muss die Existenz der Funktionen $\phi : KAND_G \rightarrow L(S(G))$ und $\phi^{-1} : L(S(G)) \rightarrow KAND_G$ gezeigt werden, sowie folgende Bedingune erfüllt sein:

$$K_G \subseteq \phi \circ \phi^{-1}(K_G) \quad (1)$$

$$|\phi(K_G)| \lesssim \sum_{k \in K_G} |k| \quad (2)$$

In Bedingung (1) ist es zulässig, dass neue Kandidaten hinzu kommen, solange sie valide Wörter in der $L(G)$ Sprache bilden.

TODO: diskutieren ob hier nicht die stärkere bedingung $\phi \circ \phi^{-1} = id_{KAND_G}$ besser wäre

In Bedingung (2) werden Edge-Cases ausgeschlossen, die nur bei kleinen diversen lösungen auftauchen können. (z.B.: $\{a b\}$)

Grammatikerweiterung S

Sei $G = (N, T, S, P)$ eine eindeutige contextfreie Grammatik.

Eine Kompression der Kandidatenmenge wird definiert als:

$S(G) := G' = (N', T', S, P')$, mit:

$N' := N \cup \{O\}$ mit $O \notin N$

$T' := T \cup \{[,]\}$ mit $[,] \notin T$

$P' := P \cup P_{Options}$

$P_{Options} := \{R \rightarrow [O], O \rightarrow rO \mid R \rightarrow r \in P \text{ mit } r \in (N \cup T)^*\} \cup \{O \rightarrow \varepsilon\}$

Morphismus

Algorithmus

Sei $O^R := \{w | R \rightarrow^* w\}$. Wir definieren $O_r \subseteq O^R$ als die Teilmenge der Wörter von O^R , die durch die Regelausführung $(R \rightarrow r) \in P$ gebildet wurden.

Algorithm 1 $\phi : KAND_G \rightarrow L(S(G))$

```

1: function  $\phi(K_G)$ 
2:   return COMBINE( $K_G, S$ )
3: end function
4:
5: function COMBINE( $O^R, R$ )
6:   if  $|O^R| = 1$  then return  $o \in O^R$ 
7:   else
8:      $W \leftarrow \{\}$ 
9:     for  $(R \rightarrow r) \in P$  do
10:       $O_r \subseteq O^R$ 
11:       $W \leftarrow W \cup \text{PARSE}(O_r, r)$ 
12:     end for
13:     if  $|W| = 1$  then
14:       return  $w \in W$ 
15:     else
16:       return  $[w_0 w_1 \dots w_{|W|}]$ 
17:     end if
18:   end if
19: end function
20:
21: function PARSE( $O_r, r = s_0 s_1 \dots s_n$ )
22:    $o \leftarrow \epsilon$ 
23:   for  $i \leftarrow 0 \dots n$  do
24:     if  $s_i \in T$  then
25:        $o \leftarrow o \circ s_i$ 
26:     else
27:        $o \leftarrow o \circ \text{COMBINE}(O_i^s, s)$ 
28:     end if
29:   end for
30: end function

```

In Zeile (27) steht O^s für die Menge der teilwörter von O, die an der Stelle von s, durch die regel R erzeugt werden.

TODO: ϕ^{-1} Definieren

TODO: Definition und Algorithmus für O_i^s angeben.

TODO: Beweisen das der Algorithmus die Kritären Erfüllt

Abstimmungen

Meine erste Idee war es. Abstimmungen an Optionen zu binden. Dadurch würde sich ein Präferenzprofil für einen Akteur ergeben, bei dem der Akteur die Kandidaten am besten Bewertet, bei denen die meisten Optionen enthalten sind, für die er sich ausgesprochen hat. Ein solches vorgehen ist zwar bei kontextfreien inhalten sinnvoll, da jedoch durch Programme ein kontextsensitiver Inhalt dargestellt ist, muss ein beliebiges Präferenzprofil über der Kanddatenmenge möglich sein, dieses ist jedoch nicht möglich, wenn die Abstimmungen an Optionen gebunden werden.

2. Manipulation

Die Idee hinter der manipulation ist des es sich bei den Wörter um eine Persistente Datenstruktur handelt. Demnach können Optionen nur hinzu kommen, jedoch nicht gelöscht werden. Jeder Akteur kann Delegationen und Stimmen abgeben sowie die eigenen löschen, hat jedoch keine Macht über die Stimmen und Delegationen der Anderen.

Die Manipulation wird durch ein **initialisiertes Transitionssystem** T beschrieben.

Sei $G = (N, T, S, P)$ eine eindeutige Grammatik sowie $G' = S(G) = (N', T', S', P')$ die dazugehörige SCFG.

Qualitätskriterien

1. operationen wie *vote*, *delegate*, *addOption* sind effizient in bezug auf Laufzeit und Speicherverbrauch.
2. Die Verifikation der Transaktion durch das Netzwerk soll möglichst effizient sein.

States

Ein Zustand der ist eine Struktur der form $(U, compile, w)$

U ist das Universum der Struktur und wird definiert durch:

$$U = \{w | S' \rightarrow_p^* w \wedge w \in T'^*\}$$

compile ist die in (1.2) definierte Funktion der Form: $compile : L(G') \rightarrow L(G)$

$w \in U$ ist das derzeitige Wort

Transitionen

TODO: von delegationen und abstimmungen befreien

Bei der Interaktion eines Programms, ist der Akteur bekannt und wird durch ein Public Key ausgewiesen. Der Akteur kann in einem Interaktionsschritt das derzeitige Wort aktualisieren, wenn es bestimmte Bedingungen erfüllt. Eine Interaktion ist also ein Tupel (h, w') so, dass h der Public Key des Akteurs ist, sowie w' das neue Wort.

Erzeugen einer Optionsmenge

Sei $\alpha, \beta, o_1, o_2 \in T'^*$ sowie $R \in N$ und $V \rightarrow_P^* v$ Sei $w = \alpha o_1 \beta \oplus [v]$ Sei weiter $R \rightarrow_P^* o_1$ sowie $R \rightarrow_P^* o_2$ Dann ist $w := \alpha[o_1 \oplus [v]o_2 \oplus []]\beta$ eine valide Transition.

Erweitern einer Optionsmenge

Sei $w = \alpha[o]\beta \wedge S' \rightarrow_P^* \alpha R \beta$ mit $O \rightarrow_P^* o, o \in T'^*, R \in N$ Dann ist $w := \alpha[or \oplus []]\beta$ mit $R \rightarrow_P^* r, r \in T'^*$ eine valide Transition.

Hinzufügen einer Stimme

Sei $w = \alpha r \oplus [v]\beta$

Dann ist $w := \alpha r \oplus [v[hn]]\beta$ mit $n = float$ eine valide Transition.

Löschen einer Stimme

Sei $w = \alpha[hn]\beta$

Dann ist $w := \alpha\beta$ mit eine valide Transition.

Hinzufügen einer Delegation

Analog der Stimme

Löschen einer Delegation

Analog der Stimme

Transaktion

Delegationsprogrammierung

Context Delegation

A/B Tester

zu untersuchen

- wie werden neue anteile vergeben

Umfang der Arbeit

Konzeptuell

- Überarbeitung der Spracherweiterung
 - macht eine Spracherweiterung überhaupt sinn?
- Die Manipulation der Besitzverteilung w sowie der Grammatik G eines Objektes O wird in Aussicht gestellt. Ihre untersuchung ist jedoch nicht gegenstand dieser Arbeit.

Implementation

- Eine Zentralisierte Version der $L(S(G))$ -Sprache sowie des Transitionsystems wird implementiert
 - JISON, ein Bison/Flex ähnlicher javascript LALR(1) parser generator wird so erweitert, dass bei Eingabe einer beliebigen Grammatik G dieser einen Parser erzeugt welcher Wörter aus der $L(S(G))$ Sprache als Eingabe bekommt und ein öffentliches Transitionsystem erzeugt.
 - Ein Initialscript, welches aus einem Wort der $L(G)$ Sprache und einer Besitzgewichtung gegeben als $\{(a, n) | a \in 20byteHexString \wedge n \in \mathbb{N}\}$ ein valides initiales Wort in der $L(S(G))$ Sprache erzeugt
 - Akteure können durch eine Öffentliche Schnittstelle(Webseite) in die Historie einsehen, sowie das Wort valide manipulieren.
 - Die Konsensfunktion wird ebenfalls Implementiert.
- Es wird keine Dezentrale version Implementiert.

- Die Delegationen sowie die Abstimmungen werden direkt Implementiert und können jederzeit beteiligten Akteuren zugeordnet werden. Eine anonyme Lösung, mithilfe einer homomorphen Verschlüsselung oder eines zero knowledge proofs wird zwar diskutiert, jedoch weder konzipiert noch implementiert

Todos

Notes

■ TODO: diskutieren ob hier nicht die stärkere bedingung $\phi \circ \phi^{-1} = id_{KAND_G}$ besser wäre	6
■ TODO: ϕ^{-1} Definieren	8
■ TODO: Definition und Algorithmus für O_i^s angeben.	8
■ TODO: Beweisen das der Algorithmus die Kriterien Erfüllt	8
■ TODO: von delegationen und abstimmungen befreien	10

Aussicht

Dezentralisierung

- änderungen der Grammatik berücksichtigen

Quellen

References

- [But14] V Buterin. A next-generation smart contract and decentralized application platform. (January):1–36, 2014.
- [Wal04] Jeremy Waldron. Property and Ownership. September 2004.
- [Woo14] Gavin Wood. ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER. 2014.