特別研究報告書

スマートコントラクトのガス消費量の Reource Aware ML を用いた静的解析

指導教員：末永　幸平　教授

京都大学工学部情報学科

小野　雄登

2021 年 2 月 2 日

スマートコントラクトのガス消費量の Reource Aware ML を用いた静的解析

小野　雄登

内容梗概

2009 年にビットコインを用いた取引がオープンソフトウェアで始まって以来，現在に至るまでにブロックチェーンを技術基盤とする様々な仮想通貨が開発されている．スマートコントラクトは，仮想通貨の取引における契約の締結や履行を自動化する仕組みであり，ブロックチェーン上で動作するプログラムとして実装される．

スマートコントラクトにはガスの概念が存在し，ガスはコントラクトの実行にかかる手数料を表している．コントラクトが実行される際に，コントラクトの各命令の評価毎に命令の内容に比例した量のガスが消費され，消費量の合計が許容ガス消費量を超えると，その命令が直ちに停止され，命令の実行による値の変更が取り消される．プログラムとして非効率なコントラクトが実行されると，想定される量以上のガスが消費されてしまうので，コントラクトのガス消費量を静的に解析することは，ユーザーが必要以上に手数料を支払わないために必要な技術であると考えられる．

本研究では，スマートコントラクトのガス消費量の静的な解析を行うプログラムを実装する．具体的には，スマートコントラクトを実装しているブロックチェーンプロトコルである Tezos において，スタックベースのプログラミング言語 Michelson で書かれたコントラクトのガス消費量を，プログラミング言語型のツールである Resource Aware ML(RAML) を用いて静的に解析する．RAML は，OCaml で用いられる文法を備えた関数型プログラミング言語で，入力として与えられたプログラムのリソース消費量の上界を，指定されたメトリックに従って自動的に，かつ静的に解析して，その結果を出力する．

実装の方針は，

1. Michelson の挙動を再現するためのライブラリを RAML で設計する．
2. 設計したライブラリを用いて，Michelson で書かれたコントラクトをエンコードする．
3. エンコードしたコントラクトを解析し，ガス消費量を見積もる．

という流れである．以下，各過程について説明する．

1. において，Michelson はスタック構造をもち，コントラクトに用いられる命令は，初期スタックを受け取ってスタックの内容を変更して返す関数として実装されている．この構造を RAML で設計するにあたって，スタックの要素をヴァリアント型 t として定義し，命令を (t list -> t list) 型をもつ関数として定義した．

2. において，Michelson のコントラクトは，初期スタックに入る値の型宣言と，初期スタックに対して順に適用される一連の命令によって構成されている．RAML においてこのコントラクトを，1. で設計したライブラリを用いて，初期スタックを表すリストに対して命令を順に関数適用するプログラムとして実装した．

3. において，2. で実装したプログラムを，RAML の steps メトリックを用いて評価ステップ数に関する解析を行った結果，基本的なスタックの操作に関する命令や，簡単な算術演算や条件分岐の命令のみを含むコントラクトについては解析が正しく行われたが，ループ命令や，リストや集合に対する再帰を行う命令を含むコントラクトについては解析が失敗するものも存在した．続いて，ガス消費量の見積もりについては，RAML の tick メトリックを用いた．tick メトリックは，リソース消費の値や発生するタイミングを，ユーザーが関数として定義することができるメトリックである．RAML で実装した各命令について，その命令のガス消費量に相当する値の tick 関数を定義し，tick メトリックを用いた解析を行い，コントラクトのガス消費量を見積もれるかどうかを検証した．結果として，コントラクトの実行において発生するガス消費のうち，プログラムの解釈実行を行う際に発生する interpreter cost について概ね正しく消費量を見積もることができた．

本研究においては，Michelson に実装されているコントラクトの命令のうち，主要なものについて RAML で実装した．残りの命令の実装については，今後の課題とする．また，ガス消費量の見積もりについては interpreter cost についてのみ取り組んだが，他の過程において発生するガス消費量の見積もり，ひいてはコントラクトの実行において発生するガス消費量全体の見積もりについても検討していきたい．

Static Analysis for Gas Consumption of Smart Contracts Using Resource Aware ML

Yuto Ono

Abstract

スマートコントラクトのガス消費量の **Reource Aware ML** を用いた静的解析

目次

1 序論

2009 年にビットコインを用いた取引がオープンソフトウェアで始まって以来，現在に至るまでにブロックチェーンを技術基盤とする様々な仮想通貨が開発されている．取引の記録をブロックとしてネットワーク上に記憶するという性質上，ブロックチェーンはデータ改竄に対する優れた耐性を持ち，仮想通貨の取引を支えるコア技術となっている．ブロックチェーン上で用いられる技術としてスマートコントラクトがある．スマートコントラクトは，仮想通貨の取引における契約の締結や履行を自動化する仕組みであり，ブロックチェーン上で動作するプログラムとして扱われる．第 3 者を介さずに，また相手の信頼を必要とせずに取引を行うことができ，決済期間の短縮や手数料の削減などの効果が期待できる．

Tezos はスマートコントラクトを用いたブロックチェーンを技術基盤とする仮想通貨の 1 つで，コントラクトはスタックベースのプログラミング言語 Michelson で書かれている．Tezos のスマートコントラクトにはガスの概念が存在する．ガスはコントラクトの実行にかかる手数料を表しており，コントラクトを実行するユーザーがマイナーと呼ばれるブロックの創始者に対して支払われる．コントラクトの実行に際して，コントラクトの各命令の評価毎に命令の実行内容に比例した量のガスが消費され，消費量の合計が許容ガス消費量を超えると，その命令が直ちに停止され，命令の実行による値の変更が取り消される．ガスの消費量の計算は複雑で，前もってガスの消費量を正確に見積もることは難しいとされているが，プログラムとして非効率なコントラクトを一度実行してしまうと，想定以上にガスが消費されてしまう．そのためにガスの消費量の静的な解析は，ユーザーが必要以上に手数料を払わないために必要な技術であると考えられる．

本研究では，プログラミング言語型のツールである Resource Aware ML（以下，RAML と略記する.）を用いて，Tezos のスマートコントラクトのガス消費量の静的な解析を行うプログラムを実装した．RAML は，OCaml の文法を用いたプログラムを入力として受け取り，プログラムのリソース消費量の限界値を指定されたメトリックに従って自動的に，かつ静的に計算し，その解析結果を多項式の値として出力するツールである．プログラムの実装の方法としては，Michelson プログラムのスタック構造を RAML において Ocaml のリスト構造

を用いたプログラムとして実装した．実装したプログラムに対して，RAML のメトリックの 1 つである tick メトリックを用いて，解析結果として得られるプログラムのリソース消費量からコントラクトのガス消費量を見積もれるかどうかを tick メトリックは，リソース消費の値や発生するタイミングを，ユーザーが関数として定義することができるメトリックである．

2 背景知識

2.1 tezos と Michelson について
2.2 コントラクトのガス消費の仕組み
2.3 Resource Aware ML について

3 RAML での Michelson プログラムの実装

4 検証結果と考察

5 改善点

6 結論

謝辞

参考文献

[1] Caplener, H. D. and Janku, J. A.: Improved Modeling of Computer Hardware Systems, *Computer Design*, Vol. 12, pp. 59–64 (1973).

[2] Beizer, B.: Towards a New Theory of Sequential Switching Networks, *IEEE Trans. Computers*, Vol. C–19, pp. 936–956 (1970).

[3] 村上伸一: 微分方程式の解曲線の表示, 情報処理, Vol. 14, pp. 231–238 (1970).

[4] 平井有三, 福島邦彦: 両眼視差抽出機構の神経回路網モデル, 信学論 (D), Vol. 56–D, pp. 465–472 (1973).

[5] Baraff, D.: Curved Surfaces and Coherence for Non-penetrating Rigid Body Simulation, *SIGGRAPH '90 Proceedings* (Beach, R. J.(ed.)), Dallas, Texas, ACM, Addison-Wesley, pp. 19–28 (1990).

[6] 對馬雄次ほか: ボリュームレンダリング専用並列計算機のアーキテクチャ,

並列処理シンポジウム JSPP'94, pp. 89–96 (1994).

[7] Barnett, S. and Storey, C.: *Matrix Methods in Stablity Theory*, Nelson, London (1970).

[8] J. E. ホップクロフト, J. D. ウルマン（木村, 野崎訳）: 言語理論とオートマトン, サイエンス社, chapter 6 (1972).

[9] 寺沢寛一: 自然科学者のための数学概論, 岩波書店, pp. 325–328 (1955).