第三回報告書

2025 年 12 月

髙橋秀明

コロンビア大学博士課程二年の高橋秀明です。第三回目となる本報告書では、この半年間の授業および研究の進捗について報告させていただきます。

1. 授業

今学期は、計算量理論を学ぶ「Introduction to Computational Complexity」と、ブロックチェーン・web.3.0 の詳細を実装・プロジェクトを通して学ぶ「Engineering Blockchain and Web3 Apps」を受講しました。

1.1. Introduction to Computational Complexity

Introduction to Computational Complexity では、計算量理論の基礎から応用までを体系的に学びました。P・NP といった基本的な計算量クラスから始まり、最終的には PCP 定理 や Communication Complexity など、より発展的な理論を扱う構成となっていました。計算量理論は、「問題を解くことの難しさ」と「解の正しさを検証することの難しさ」などの関係を明らかにする分野です。特に PCP（Probabilistically Checkable Proof）は、証明全体を精査せずとも、その一部を確率的に検査することで高い信頼度で正しさを判定できるという枠組みを与えます。この考え方は、私の研究テーマである ゼロ知識証明（Zero-Knowledge Proof; ZKP） の理論的基盤の一つでもあり、現在の研究内容と密接に関連している点で、非常に有意義な講義でした。成績評価は宿題および試験に基づくもので、内容を十分に理解していれば対応可能な難易度であったと感じています。

1.2. Engineering Blockchain and Web3 Apps

Engineering Blockchain and Web3 Apps はプロジェクトベースで評価が行われる科目であり、ブロックチェーンおよび Web3 技術について、実装を通じて理解を深めることを目的としています。私は本講義において、ゼロ知識証明の回路を検証可能な形で記述するためのプログラミング言語「 Runwai 」 の開発に取り組みました

(https://github.com/Koukyosyumei/Runwai)。本プロジェクトは以前より構想していたものであり、興味を共有する 4 名のクラスメイトとともに、言語設計、理論的性質の検討、実装、簡単な応用例の作成までを行いました。

ゼロ知識証明では、証明したい計算の正しさを、**有限体上の多項式制約**として表現する必要があります。例えば、x^2 + 2x + 1 = 0 mod 17 のような制約がその一例です。従来は、各アプリケーションごとにこのような制約を人手で設計する必要があり、設計ミスや検証不足が実用化の大きな障壁となっていました。

近年では、この問題に対処するため、「任意のプログラムの実行をゼロ知識証明可能にする」汎用的な基盤である zkVM（ゼロ知識仮想マシン） が多数提案されており、これを用いてアプリケーションごとに別個に制約を実装する必要なく、ZKP を適用することが目指されています。この分野は近年とても注目が集まっており、RISC-0、SP1、Jolt などを含め、すでに 20 以上のプロジェクトが存在します。

一方で、zkVM 自身の正しさを定義・保証するためにも、依然として多項式制約が用いられており、その設計および検証は依然として困難な課題です。この課題に対し、渡したとのプロジェクトでは 篩型（Refinement Type） を用いた新しい関数型プログラミング言語を設計しました。

篩型とは、通常の型（整数型や真理値型など）に加えて、その値が満たすべき条件を型として記述できる仕組みです。例えば、{v: Int : v % 2 = 0}は「偶数である整数」を表す型となります。、

Runwai では、有限体上の多項式制約を記述するための関数型言語を設計し、その制約が満たすべき意味的性質を篩型として明示的に表現できるようにしました。さらに、本言語を Lean4（対話型定理証明支援系）上で実装することで、Lean4 の型検査および定理証明機能を用いて、制約が意図した性質を満たしているかを機械的に検証できる仕組みを実現しました。

```
#runwai_register chip IsZero(trace: [[Field: 3]: n], i : {v: UInt | v < n})
  -> {Unit| trace [i][1] == if trace [i][0] == Fp 0 then {Fp 1} else {Fp 0}} {
  let x = trace [i][0];
  let y = trace [i][1];
  let inv = trace [i][2];
  let u₁ = assert_eq(y, ((Fp 0 - x) * inv) + Fp 1);
  let u₂ = assert_eq(x*y, Fp 0);
  u₂
}
```

図 1: Runwai の例

2. 研究

2.1 zkFuzz プロジェクト

昨年度より取り組んできた研究テーマである、Fuzzing（ランダム入力を用いた動的解析手法）をゼロ知識証明に適用し、回路中のバグを検出する研究について、以下の論文が国際会議に採択されました。

- H. Takahashi, J. Kim, S. Jana and J. Yang, "zkFuzz: Foundation and Framework for Effective Fuzzing of Zero-Knowledge Circuits," in 2026 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2026, pp. 919-938, doi: 10.1109/SP63933.2026.00049.

IEEE S&P は、コンピュータセキュリティ分野において最も権威のある国際会議の一つであり、博士課程における目標の一つであった同会議への主著論文採択を、在籍 1 年目で達成できたことは大きな成果であると考えています。また、本研究の内容を広く共有する目的で、プロジェクトのウェブサイトも作成しました。https://zkfuzz.xyz/

2.2 現在の研究と今後の展開

現在は、zkFuzz で培った動的解析の手法を発展させ、一定の理論的保証を伴う検証手法を取り入れた形で zkVM のバグ検出を行う研究に取り組んでいます。
これまでに複数の zkVM 実装に対して調査を行い、合計 8 件の不具合を発見しています。現在は、手法の整理、実装の改善、および論文執筆を進めている段階です。

また、11 月にはゼロ知識証明に関する国内イベント Zero Knowledge Frontier Japan に登壇者として招待され、自身の研究について発表を行いました。

さらに、最近は OSS への貢献活動も再開し、Lean4 を用いて検証可能な形でゼロ知識証明回路を記述するための言語 clean において、基本的な制約に関する Soundness（健全性） および Completeness（完全性） の証明に取り組んでいます。

3. 最後に

改めまして、留学を支援していただいている船井情報科学振興財団の皆様に感謝申し上げます。ご期待に応えることができるよう、より一層努力を重ねたいと思います。