IN5100:
Formal Semantics of Blockchain Languages and Virtual Machines in $\mathbb{K}$

Peter Ölveczky

October 18, 2023

Curriculum

The main technical paper is

> Everett Hildenbrandt and others: *KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine*, which appeared in the proceedings of 2018 IEEE 31st Computer Security Foundations Symposium (CSF'18), IEEE. Available (from UiO at least) at `https://ieeexplore.ieee.org/document/8429306`. (PDF at `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8429306`) Let me know if you cannot access the paper for some reason.

Two other, shorter, invited papers are:

> Xiaohong Chen and Grigore Rosu: *The K Vision for the Future of Programming Language Design and Analysis*. In Formal Methods in Outer Space 2021 (Klaus Havelund Festschrift), pages 3–9, Springer LNCS volume 13065. `https://link.springer.com/chapter/10.1007/978-3-030-87348-6_1`

and

> Grigore Rosu: *Formal Design, Implementation and Verification of Blockchain Languages*. In proceedings of FSCD 2018, Leibniz International Proceedings in Informatics, 2018. Open access, available at `https://drops.dagstuhl.de/opus/volltexte/2018/9172/`.

In addition, Grigore Rosu's slides for his FACS 2018 invited talk are very useful: can be reached ("slides") from `http://sevlab.postech.ac.kr/facs18invited-speakers/#keynote2`.

Note that it is not expected that you have a (detailed) knowledge of $\mathbb{K}$, so please do not focus on the details of the semantics, but on the overall picture; in some sense, the real curriculum is defined by good answers to the "questions" below.

1 What to know?

As mentioned, don't worry about details about the formal semantics of EVM in $\mathbb{K}$.

Some questions on what to know about the first of the above papers (KEVM ...) are (roughly in order of appearance in the paper):

- (Section I) What is Ethereum, Ether, and EVM?
- What are "smart contracts" (see also II.B)?
- How much money is lost/stolen/etc by the various flaws and attacks mentioned in the paper?
- What is the goal of the $\mathbb{K}$ semantic framework?
- (I.A) Why are formal methods particularly good targets for formal methods?
- (I.C) What are some claimed benefits of a formalization of the EVM semantics in $\mathbb{K}$?
- (II.A) What are the differences mentioned here between Bitcoin and Ethereum?
- (II.B and elsewhere) What language(s) are smart contracts written in (as explained in this paper)?
- (II.C) What is *gas* (in this context), and why is it needed?
- (II.D) What are the stated goals of $\mathbb{K}$?
- (III.B) In the semantics, what aspects of the VM execution state must be maintained/stored in the semantics? Apart from having a very rough idea of this, don't worry about details of the semantics. So: read Section III very quickly for a very quick overview.
- (IV) They compare the performance of using KEVM with what? On what "applications"? What are the results of the performance comparisons?
- (V.A) Very briefly: what is *reachability logic*? What does the reachability claim $\phi \Longrightarrow \psi$ mean?
- (V.B) What is the ERC20 standard?
- What is the claimed advantage of performing the analysis at the EVM level instead of at "high-level language" level?
- What are the disadvantages? How do the authors address this disadvantage?
- (V.B.2) What kind of error in HKG Token led to the re-issuance of an entire token?
- (V.C) What does the "requirement" on page 213 (code on top of right column) seem to specify?
- (VI) What other important outcomes came out of this project/work?

Some questions to ponder about when reading the other papers:

- (1) Why is "standard" "post-mortem" formal semantics of programming languages not sufficient for blockchain programming languages and virtual machines?
- Why target the blockchain as an application domain for $\mathbb{K}$?
- (2) Very roughly: what is $\mathbb{K}$? (No need to read the middle paragraph in Section 2.)
- (3) What had been achieved by 2018 in this project?
- What is the motivation/goal of the K project?