Optimal Synthesis of Michelson Bytecode

Intership at Nomadic Labs

Tianchi YU

yu.tianchi@nomadic-labs.com

Tuteurs : **Richard Bonichon, Yann Régis-Gianas**

27 April 2021

Catalog

Smart Contract and *gas*

Smart Contracts consumes *gas*, which is a kind of abstract resource and purchased with crypto-money.

What are the roles of gas:

- discouraging denial-of-service attacks
- incentivizing honest programs to run efficiently

Michelson

Michelson is the domain-specific language used to write smart contracts on the Tezos blockchain. Two main properties of Michelson is:

- stack based
- strongly typed

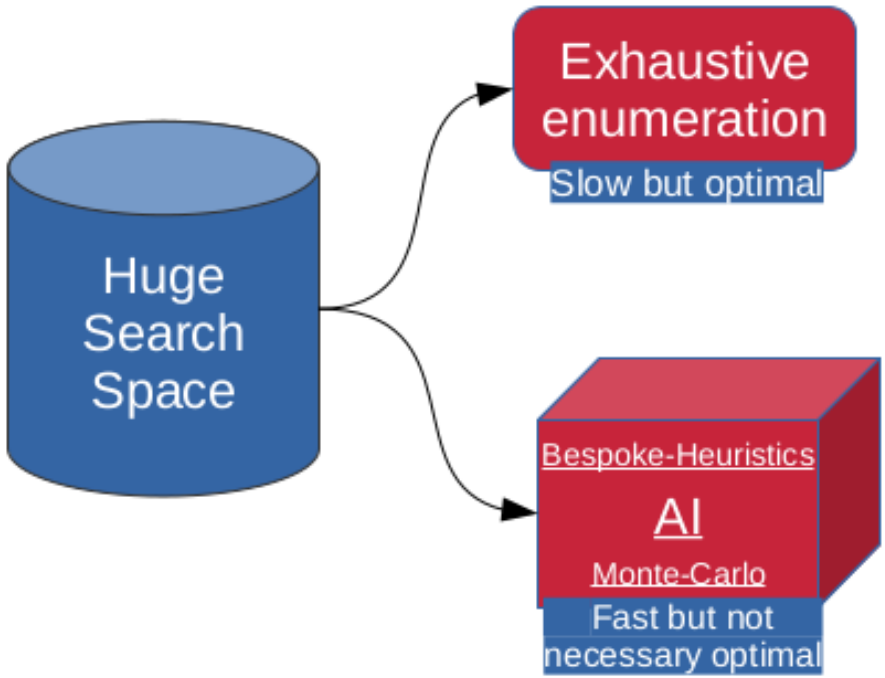
Simple example

parameter unit;
storage unit;
code CDR; NIL operation; PAIR;

Optimization & Super-optimization

Goal of optimization: reducing the gas required by the smart contracts.

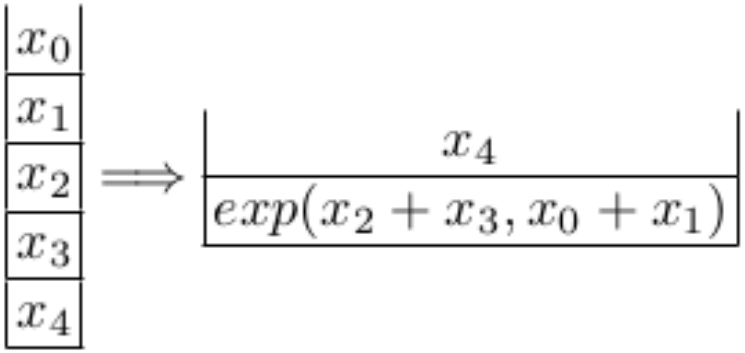
Super-optimization[1]: finding global optimizations to program structure whihc might be missed by local transformations



Max-SMT Method

MAX-SMT[2]: they use basic blocks and try to give a semantic model for these basic blocks and assign a cost using the cost model. (Two phases)

1. Extracting Stack Functional Specifications(SFS)



2. The Synthesis of Optimized Blocks

Encode the problem as a Max-SMT problem - using Max-SMT optimizer as a black box with an important gain in efficiency.

Our purpose :

To build AI-based method, which aims to find the optimal Smart Contracts in a fully blackbox way. Possibly partially referencing MAX-SMT(specially the first step), but we want to realize the optimization by

Our purpose :

To build AI-based method, which aims to find the optimal Smart Contracts in a fully blackbox way. Possibly partially referencing MAX-SMT(specially the first step), but we want to realize the optimization by

- **Synthesizing an expression/ an objective function** : I/O relations;

Our purpose :

To build AI-based method, which aims to find the optimal Smart Contracts in a fully blackbox way. Possibly partially referencing MAX-SMT(specially the first step), but we want to realize the optimization by

- **Synthesizing an expression/ an objective function** : I/O relations;
- **Solving through search heuristics** : S-metaheuristics

Example tool: *Xyntia*[3] - AI-based blackbox deobfuscator

Further steps

- *Benchmark Collection*, means we could benchmark the optimization performance of different metaheuristics.

Further steps

- *Benchmark Collection*, means we could benchmark the optimization performance of different metaheuristics.
- *Translation Validation*, verifies the semantic equivalence after the optimization.

Expectation

Our blackbox method could

- optimize the Michelson byte-code
- be efficient / fast
- perform solidly with different metaheuristics
- be verified by semantic equivalence

References


Rudy Bunel, Alban Desmaison, M. Pawan Kumar, Philip H.S. Torr. “Learning to Superoptimize Programs”. In: **ICLR**. (2017), DOI: {arxiv-1611.01787v3}.


Elvira Albert,Pablo Gordillo,Albert Rubio,Maria A. Schett. “Synthesis of Super-Optimized Smart Contracts Using Max-SMT”. In: **Computer Aided Verification**. 32nd International Conference. Los Angeles, CA, USA, July 2020.


Grégoire Menguy,Sébastien Bardin,Richard Bonichon,Cauim de Souza Lima. “AI-based Blackbox Code Deobfuscation Understand, Improve and Mitigate”. In: (2021), DOI: {arxiv-2102.04805}.

Thank you !

Contact:

yu.tianchi@nomadic-labs.com