Juvix: Efficient, dependently-typed smart contracts

Christopher Goes, Cryptium Labs

Oh no, blockchain!

- Scams
- Buggy programs
- Too many buzzwords
- (aside: terrible terminology)

Contrapositive: excellent use-case for formal verification

Language economics for smart contracts

- Correctness matters
  - No security-by-obscurity
  - Controlling funds, data, high-value transactions
- Execution efficiency matters
  - Must be replicated
- Compiler speed doesn’t matter much
- Developer accessibility, syntactic familiarity matter less

Core language

- Syntax, semantics from quantitative type theory (McBride, Atkey)
  - Combines dependent & linear types, dependent linear implication
  - Separates contemplation from computation
- Dependent types for property verification
- Linear types for efficient compilation, erasure
- Instantiated over Nat rig
  - More precision for optimizations

Optimal reduction

- Interaction system
  - Node types corresponding to atoms
  - Rewrite rules corresponding to reduction
- Lambda term translated to graph, rewritten, read-back to lambda term

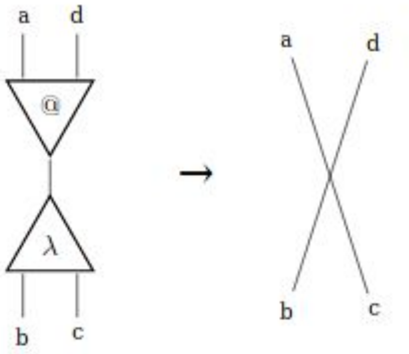

Figure 7: Lambda-application annihilation (beta reduction)

Optimal reduction

- Benefits
    - Optimal sharding
        - Efficient higher-order functions, lexical closures
        - Asymptotically better (# of β reductions) than call-by-name, call-by-value
    - No (separate) garbage collection
        - Encoded in graph rewrite rules
    - Automatic parallelism
- Constraints
    - Subset of lambda terms (abstract algorithm): typable in EAL
    - Elementary complexity class terms

Open questions

- Translation between QTT & EAL
- Tradeoffs between space & time in optimal reduction
- Lambda-encoding of user-defined data types
    - Deriving induction for recursive types
    - O(1) pattern matching (predecessor)
        - Current: Mendler encoding, Scott encoding, work by Aaron Stump

References

- Optimal Implementation of Functional Programming Languages - Asperti et al. [1998]
- I Got Plenty O’ Nuttin’ - Conor McBride [2016]
- Quantitative Type Theory - Robert Atkey [2018]
- The Calculus of Dependent Lambda Eliminations - Aaron Stump [2018]