Floating-point numbers in Stak Scheme

@raviqqe

September 1, 2024

Contents

- Stak Scheme
- Numeric tower in R7RS
- Implementation in Stak Scheme
- Future work

Stak Scheme

- A bytecode compiler and virtual machine (VM) for Scheme
    - The compiler is written in Scheme.
    - The VM is written in Rust.
- It aims to support R7RS-small.

Progress

- Floating-point number

Numeric tower in R7RS

- R7RS defines a numeric tower.
  - `number?`
  - `complex?`
  - `real?`
  - `rational?`
  - `integer?`

```
> 1+2i
1+2i
> 3.14
3.14
> 1/3
1/3
> 42
42
```

Numeric tower in R7RS

- `exact` and `inexact` conversion

```
> (inexact 1/3)
0.3333333333333333
> (exact 3.14)
7070651414971679/2251799813685248
```

Implementation in Stak Scheme

- Stak Scheme supports numbers represented by 63-bit integers or 64-bit floating-point numbers internally.
- They are switched by Rust's `feature` system.
- The bytecode compiler compiles floating-point numbers in source codes into integer components.
  - Signs, mantissae, and exponents
  - Smaller mantissae and exponents take less space in bytecodes.
  - Generated bytecodes re-calculate floating-point numbers on initialization.

Implementation in Stak Scheme

Problems

Degraded precision in bytecodes

- Floating-point numbers' integer components are calculated by normal `exp` and `log` procedures.
  - We can't calculate accurate mantissae and exponents.

Demo

Future work

- Optimize bytecode decoding

Summary

- Building floating-point numbers is fun!