Stak Scheme updates

@raviqqe

October 10, 2026

Contents

- Stak Scheme
- Updates
  - Floating-point numbers
  - I/O
  - Library system
  - R7RS compatibility
- 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 implements the R7RS-small standard.

Floating-point numbers

Floating-point number literals

- Bytecode used to pack a sign, an exponent, and a mantissa into one integer.
  - The integer must fit in the 52-bit mantissa of `f64`.
    - Some VMs represent all numbers as `f64`.
  - Mantissas were limited to 38 bits (~11 decimal digits).
- Now, a mantissa is encoded as a separate integer.
  - Literals keep their full 53-bit mantissas.

Decimal representation

- `number->string` now prints digits up to the precision of each number.

Expression	Old	New
`(write 3.141592653589793)`	`3.141592654`	`3.141592653589793`
`(write 0.1234567890123456)`	`0.12345679`	`0.1234567890123456`

- Other fixes
  - `round` with large numbers
  - Division by zero in the integer and 62-bit float configurations

I/O

I/O

- New I/O benchmarks on CodSpeed
  - In-memory ports and files
  - ASCII and UTF-8 strings
- UTF-8 encoding and decoding in ports rewritten
  - `read-string` from files got 3x faster.
- Bug fixes
  - `peek-char` and `peek-u8` reordering pending input
  - UTF-8 strings in `open-input-string`
- `(stak io)` split into byte and character (`(stak io utf8)`) libraries

Library system

Library load paths

- `(import (foo bar))` loads `foo/bar.sld` in library load paths.
- `-I` (prepend) and `-A` (append) options from SRFI 138
  - Both in the `stak` and `stak-compile` commands
- `include` in libraries resolves paths relative to their files.
- Circular imports are detected.

Library load paths

```
; lib/foo/bar.sld
(define-library (foo bar)
  (export hello)
  (import (scheme base) (scheme write))
  (begin
    (define (hello)
      (display "Hello, world!")
      (newline))))
```

```
; main.scm
(import (foo bar))

(hello)
```

```
stak -I lib main.scm
```

R7RS compatibility

R7RS compatibility

- `include-library-declarations` syntax (`,@` for `include`)
- `cond-expand` with `(library ...)` checks available libraries.
  - It used to check a fixed list of library names.
- `exit` with integers
- Reader
  - Datum comments
  - Hexadecimal characters

```
(write '(1 #;2 3)) ; -> (1 3)
(write-char #\x3bb) ; -> λ
```

`snow-chibi`

- A request for snow-chibi support
  - `snow-chibi install --impls=stak '(srfi 180)'`
- Packages on snow-fort need the features above.
- Remaining
  - Default library directories
  - `cond-expand` in `define-library`

Compiler on Gauche

- The compiler is written in portable R7RS.
  - CI compiles all Scheme files with Chibi Scheme, Gauche, Guile, and Stak Scheme, and compares their bytecode byte by byte.
- Gauche crashed on the compiler for a year.
  - A bug in its `inline-locals` optimization with nested closures
  - Reported and fixed upstream on October 4
  - CI uses `gosh -fno-inline-locals` until the next release.

Future work

Future work

- `snow-chibi` support
  - Default library directories
  - `cond-expand` in `define-library`
- Big integers

Big integers?

- Investigated with AI
  - Bignums as ribs with an unused type tag
  - Arithmetic implemented in Scheme
- Options
  - An opt-in `(stak bignum)` library: zero cost for other programs
  - Transparent promotion with overflow checks in Scheme: 1.6-2.5x slower arithmetic
  - Primitives calling Scheme procedures on overflow: no cost on the fast path
- The numeric tower is still the final boss.

Summary

- The full R7RS-small compatibility is near! 🚀