The state of R7RS - Implementing MAL in portable Scheme

Vasilij Schneidermann

October 2017

Outline

Section 1

Intro

About

- Vasilij Schneidermann, 25
- Software developer at bevuta IT, Cologne
- Currently contracting at `$BIGCORP`
- `v.schneidermann@gmail.com`
- `https://github.com/wasamasa`
- `http://emacshorrors.com/`
- `http://emacsninja.com/`

Motivation

- CHICKEN isn't the only Scheme worth using
- R5RS without extensions is rather limiting
- R6RS failed (Chez not withstanding)
- What about R7RS?
- Is it usable?
- What's ahead of us?

Section 2

RNRS

General

- Started out as Report and Revised Report on the algorithmic language Scheme
- Revised Revised Report on the algorithmic language Scheme -> R2RS
- Community-driven standard
- Unanimous vote required to pass new revision

R5RS (1998)

- The baseline
- 50 pages
- Notable features: Numeric tower, continuations, multiple values, TCO, hygienic macros
- Not quite enough, implementations typically provide extensions (such as a module system, records, string ports, etc.)
- Most widely implemented standard

R6RS (2007)

- The controversial one
- Broke with unanimous vote
- 190 pages in total (standard, libraries, rationale, appendix)
- Adds libraries, bytevectors, UTF-8/16/32 support, two kinds of records, exceptions (with condition hierarchy), a second ports API, bitwise arithmetic, `syntax-case`, hash tables, enums, etc.

R7RS (2013-?)

- Aims to fix R6RS
- Almost compatible to R6RS (library definitions, `#!r6rs`, errata)
- R7RS-small: Like R5RS, removes the controversial R6RS parts
- R7RS-large: Provides a big standard library in form of SRFIs
- https://github.com/ecraven/r7rs-benchmarks
- https://github.com/ecraven/r7rs-coverage

R7RS-small (2013)

- 90 pages, minimalist
- Adds records, libraries, `cond-expand`, exceptions, bytevectors, string/bytevector ports, parameters, timing, etc.
- Many small fixes, some complications (number literal incompatibility, `#true` / `#false` literals)
- Considerable number of implementations exist by now

R7RS-large (ongoing)

- Organized by rainbow-colored dockets
- Might surpass CL standard in size (and time to completion)
- Red: Data structures (done)
- Orange: Numbers (TBD)
- Yellow: Syntax (TBD)
- Green: Non-portable (TBD)
- Blue: Further extensions (TBD)
- Indigo: Extensions of doubtful merit (TBD)

Section 3

Evaluation with MAL

What is MAL?

- Make A Lisp
- Guide for making a Lisp interpreter
- Clojure-like, minimalistic, bootstrappable language
- Implemented in 70+ languages
- TDD approach with $>$ 600 unit tests

Why MAL?

- Non-trivial, yet tractable exercise
- Implementing this should uncover plenty bugs
- R7RS-small covers just enough to make it possible
- I know it reasonably well (handed in four implementations so far)

Candidates

Chibi	Kawa	CHICKEN	Gauche	Picrin
Sagitarrius	Cyclone	Foment	Guile	Racket
Larceny	Mickey	Husk	Gerbil	Rapid

Rejects

- Guile (no R7RS mode or library support)
- Larceny (whacky tooling)
- Racket (inofficial support, issues with library loading)
- Mickey (completely broken)
- Picrin (no library loading support)
- Husk (superseded by Cyclone)
- Rapid (requires a R7RS implementation)
- Gerbil (R7RS support is WIP)

Chibi

- The recommended one for full compliance
- Comprehensive test suite (reused by others)
- No issues encountered
- Comes with a few dozen extensions
- Used for R7RS scripts, recommended for embedding
- Somewhat slow

Kawa

- Not really a GNU thing (as opposed to Guile)
- JVM language, good interop (only beaten by Clojure's syntax)
- Yet: Fast compiler, small boot time, decent speed, no issues
- Contains other language implementations, JEmacs
- Works for Android, Applets, . . .
- Why isn't it more popular?

Gauche

- Aims to be a practical Scheme distribution
- Bundled with many extensions, aimed at scripting
- No issues, good speed for an interpreter
- Been around for a long time
- I might have used this haven't I met. . .

CHICKEN

- My personal choice for writing CLI utils and small web apps
- Best in terms of speed from all candidates
- R7RS support implemented by an egg
- Rather tricky to use locally installed libraries, otherwise no issues
- Looking forward to CHICKEN 5 fixing this

Sagittarius

- A fork of Gauche
- Higher development activity, extra libraries, different build system
- Otherwise very similar to it

Cyclone

- Relatively new implementation with similar design choices as CHICKEN
  - Cheney on the MTA (with native thread support)
  - Generational GC (runs concurrently to threads)
  - Written purely in Scheme
  - Compiles to C
- Actively developed
- I've uncovered seven bugs so far and handed in one PR
- Not quite as fast as CHICKEN yet

Foment

- Someone's personal learning project
- As fast as Chibi
- I've reported five bugs and handed in one PR

Section 4

Outro

Summary

- Implementing MAL was successful
- I've found a new Scheme to play with (Kawa)
- Greatest difficulty: Loading libraries
- R7RS-small covers enough for writing useful programs/libraries
- R7RS-large is not completely bonkers

Further work to be done

- Further implementations:
  - Larceny
  - Gerbil
  - ...
- Testing Snow2
- Porting something more advanced (GRASS?)

Questions?