Testing a self-hosted compiler in Stak

@raviqqe

December 10, 2023

Contents

- Self hosting in Stak
- Testing a self-hosted Stak
- Progress
- Future work

Self hosting in Stak

- Stak is a Scheme compiler + bytecode virtual machine (VM.)
- Self hosting is a fun activity to write a language processor by the language the processor can process.
  - e.g. A C compiler written in C can compile itself. And then, the compiled compiler can compile any C source codes.
- Stak's compiler is now self-hosted.

Design bug in the language

Problem

- I assumed that some identifiers can be reserved for the special use by a compiler and a VM.
- But then, those reserved identifiers cannot be used by the compiler when it is self-hosted.
  - Paradox!
- There are 4 reserved symbols.
  - `$$rib`, `$$false`, `$$true`, `$$null`
- They do not have string representation because they are initialized as primitives by a VM.

Design bug in the language

(Temporary) solution

- Handle those reserved identifiers in a special way.
- When we convert strings to symbols of reserved identifiers, we do not try to find them in a symbol table.
- But we rather directly map them to symbols in environment.

```
(define (string->symbol x)
  (cond
    ((equal? x "$$rib")
      '$$rib)

    ...

    (else
      (let ((pair (member x symbol-table (lambda (x y) (equal? x (symbol->string y))))))
        ...))))
```

Testing a self-hosted compiler

- Self-hosted compilers have stages.
  - Stage 1: Compile a compiler by an external compiler.
  - Stage 2: Compile a compiler by a stage-1 compiler.
  - Stage 3: Compile a compiler by a stage-2 compiler.
- Compiling Scheme source codes in Stak involves the following components:
  - Source codes
  - Bytecodes of the compiler
  - A VM
- Source codes and a VM are always the same regardless of the compiler.
- We can just test equivalence of the bytecodes of the compiler!

Testing a self-hosted compiler

Examples

- They are all implemented on GitHub Actions.
- A job to test a self-hosted compiler
  - `self_host_test.sh`
  - A `self_host_test` job
- A job to run integration tests with a self-hosted compiler.
  - A `self_host_integration_test` job

Progress

- Self-hosted compiler

Future work

- `eval` procedure
- Library system

Summary

- Self hosting is fun! 😄
- Testing self hosting is kind of fun. 😃