DSW algorithm

@raviqqe

May 31, 2026

Contents

- What I'm doing these days
- DSW algorithm
- Future work

Moco VM

- The new project I'm working on these days.
  - These daysと言い張る勇気

```
Author: Yota Toyama <raviqqe@gmail.com>
Date:   Thu May 14 20:46:46 2026 -0700

    Rename `vm` `machine` (#55)
```

- VM for multiple host/target languages.

Moco VM

- Host: Rust, Go, Assembly, etc.
- Target: JavaScript, Lua, etc.
  - Scheme?
- Spiritual successor of Ribbit VM
  - More portable than the VM of Stak Scheme
- DSW garbage collection

DSW algorithm

- Deutsch-Schorr-Waite algorithm
  - When I search "DSW garbage collection" in Google, I get local schedules of garbage collection...
- It seems to be very minor for the GC use.
  - The thread in Zulip: https://prog-lang-sys-ja.zulipchat.com/#narrow/channel/343500/topic/DSW.20algorithm
- The subject of program verification (correctness, termination)
- The algorithm keeps flipping pointer directions among objects back and forth.

Implementation

Mark

```rust
fn mark(&mut self) -> Result<(), Error> {
    let mut previous = V::default();
    let mut current = self.root;

    loop {
        let cons = Cons::from(current);
        let value = self.get(cons.index())?;

        if !value.is_marked() {
            if value.is_pointer() {
                self.set(cons.index(), previous.mark(true))?;
                previous = current;
                current = value;
            } else {
                self.set(cons.index(), value.mark(true))?;
            }
        } else if cons.index().is_multiple_of(2) {
            current = Cons::new(cons.index() + 1).into();
        } else if !previous.is_pointer() {
            break;
        } else {
            let previous_cons = Cons::from(previous);
            let current_cons = Cons::from(current);
            previous = self.get(previous_cons.index())?;

            self.set(
                previous_cons.index(),
                V::from(current_cons.set_index(current_cons.index() - 1)).mark(true),
            )?;

            current = previous_cons.into();
        }
    }

    Ok(())
}
```

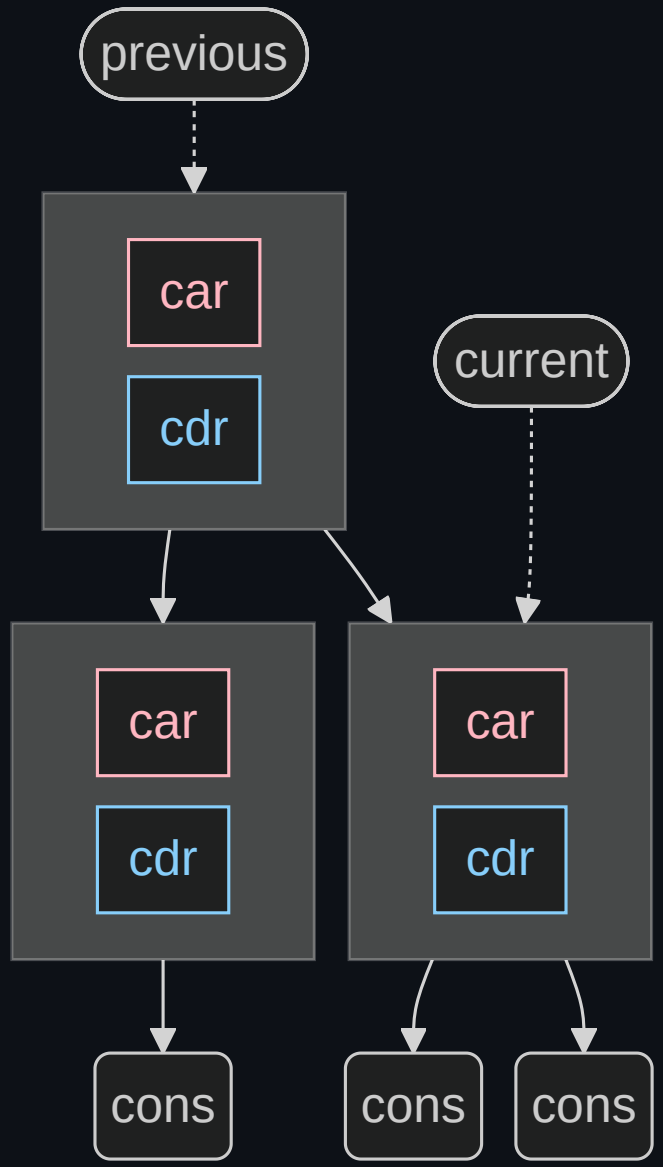
Sweep

```
fn sweep(&mut self) -> Result<(), Error> {
    self.free = Default::default();

    for index in (0..self.heap().len()).step_by(2) {
        let value = self.get(index)?;

        if value.is_marked() {
            for field in [0, 1] {
                let index = index + field;
                self.set(index, self.get(index)?.mark(false))?;
            }
        } else {
            self.set(index + 1, self.free)?;
            self.free = Cons::new(index).into();
        }
    }

    Ok(())
}
```


previous
car
cdr
current
car
cdr
car
cdr
cons
cons
cons

Future work

- Big integer in Stak Scheme?
  - The numeric tower is the final boss
  - Some SRFIs too?
- The Moco VM project
  - JavaScript?
  - Lua?
  - https://github.com/raviqqe/moco

Summary

- Building GC is fun!