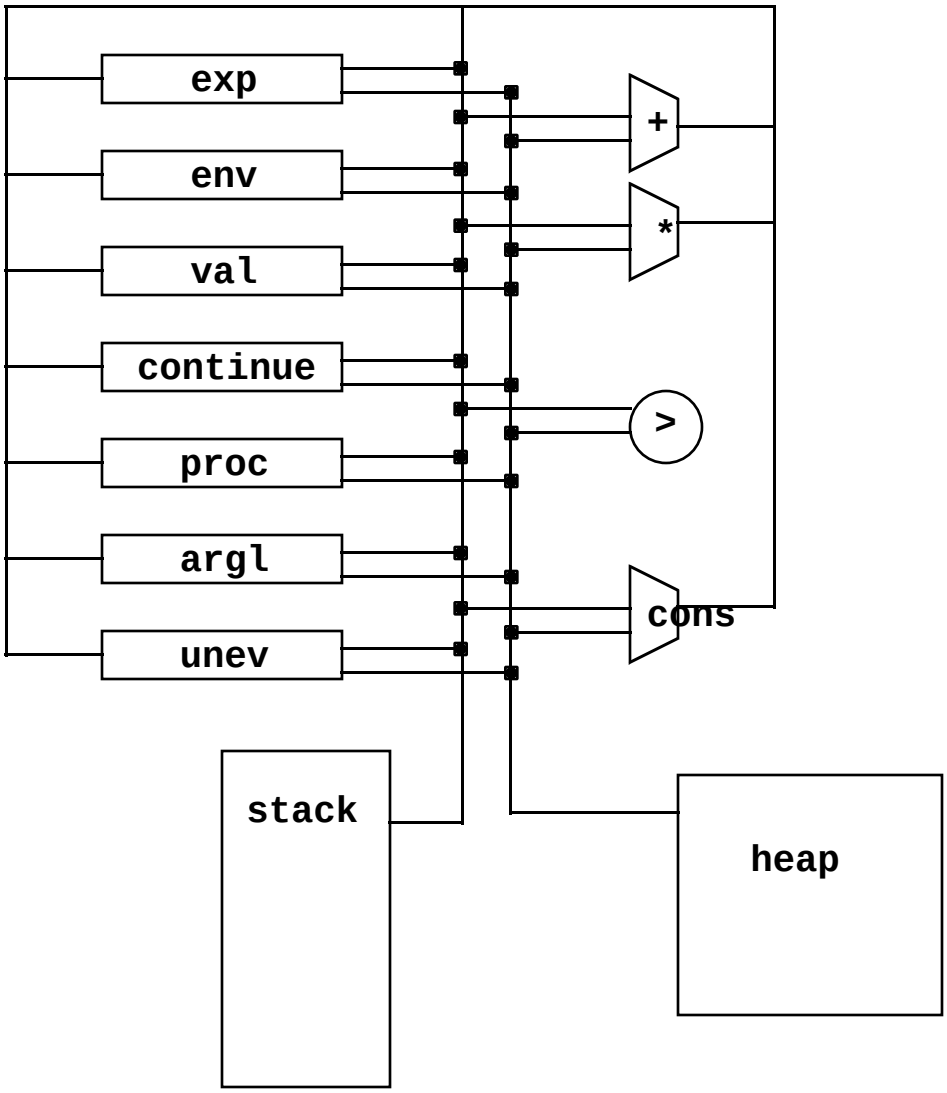
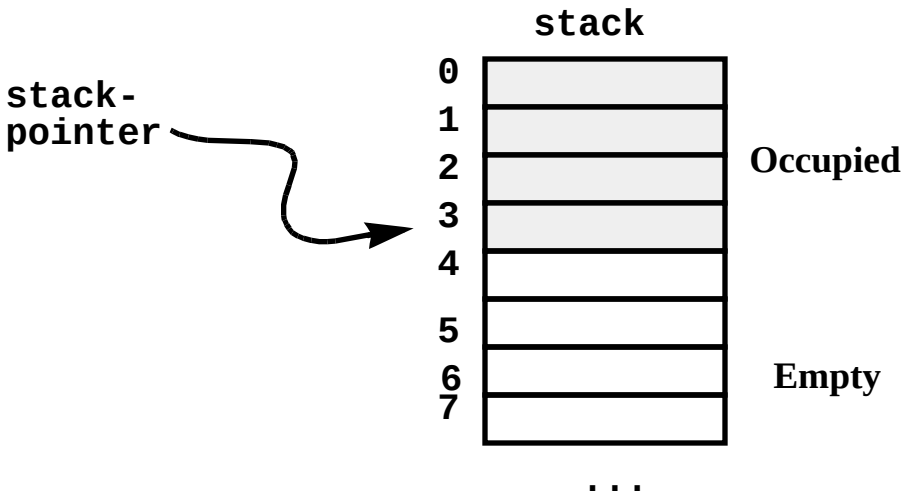
Scheme Interpreter - Register Machine

Data Flow:



Vector Implementation of the Stack




```
(save <value>)
```

becomes

```
(assign stack-pointer
         (op +) (reg stack-pointer) (const 1))
(perform (op vector-set!)
         (reg stack) (reg stack-pointer) <value>)
```

```
(restore <regname>)
```

becomes

```
(assign <regname>
        (op vector-ref) (reg stack) (reg stack-pointer))
(assign stack-pointer
       (op -) (reg stack-pointer) (const 1))
```

Implementing the Pair Abstraction

Pair accessors:

```
(assign <regname> (op car) <source>)
(assign <regname> (op cdr) <source>)
```

becomes

```
(assign <regname>
        (op vector-ref) (reg the-cars) <source>)
(assign <regname>
        (op vector-ref) (reg the-cdrs) <source>)
```

where <source> is now treated as an offset into the heap.

Pair allocation:

```
(assign <regname> (op cons) <value1> <value2>)
```

becomes

```
(perform (op vector-set!)
         (reg the-cars) (reg free) <value1>)
(perform <op vector-set!)
         (reg the-cdrs) (reg free) <value2>)
(assign <regname> (reg free))
(assign free (op incr) (reg free))
```

Mark/Sweep

Mark Phase:

- If pair
  - If already marked, then return
  - else
    -- set the mark
    -- mark car
    -- mark cdr
- Else not a pair, so return

Sweep Phase:

- Set **`free`** to **`E0`**
- Start scanning at end of memory
- Scan loop
  - If **`scan`** pointer is before start of memory, then we're done
  - If mark set for **`scan`** cell
    -- clear mark
    -- move **`scan`** pointer back one
    -- continue at scan loop
  - Else mark is not set
    -- set cdr of scan cell to **`free`**
    -- set **`free`** to **`scan`**
    -- move **`scan`** pointer back one
    -- continue at scan loop

Stop & Copy

Two Parts:

1. Move cells in old memory to front of new memory
2. Update pointers in new memory to point to *new* locations of cells

Sweep Phase:

- Set **`free`** to **`E0`**
- Start scanning at end of memory
- Scan loop
  - If **`scan`** pointer is before start of memory, then we're done
  - If **`scan`** cell already marked, then return
  - If mark set for **`scan`** cell
    -- clear mark
    -- move **`scan`** pointer back one
    -- continue at scan loop
  - Else mark is not set
    -- set cdr of scan cell to **`free`**
    -- set **`free`** to **`scan`**
    -- move **`scan`** pointer back one
    -- continue at scan loop