

Representing Control in the Presence of First-Class Continuations

Robert Hieb, R. Kent Dybvig, Carl Bruggeman

Thomas Mari

December 14, 2018

Objectives

Efficient implementation of first-class continuation used in Scheme.

Can be

- an argument
- returned from a function
- modified
- assigned to a variable

Things we want for the control flow:

- capturing a continuation
- reinstating a continuation

Data Structure for environment

Heap

- More flexible space
- Global variable and pointer
- slow access due to scattered data

Stack

- Local variables
- Quick allocation and deallocation

Continuation in Scheme : call/cc

```
( + ( * 3 4 ) 5 ) ; k : lambda (v) v
( * 3 4 ) ; k0 : lambda (v) ( + v 5 )
3 ; k1 : lambda (v) ( + ( * v 4 ) 5 )
4 ; k2 : lambda (v) ( + ( * 3 v ) 5 )
5 ; k3 : lambda (v) ( + ( * 3 4 ) v )
```

```
(+ (call/cc
  (lambda (k)
    (k (* 3 4))
  )
  5)
```

```
(define k0 #f)
(+ (call/cc
  (lambda (k)
    (begin
      (set! k0 k)
      (k (* 3 4)))
    )
  )
  5)
```

Others Approaches

Handling activation records = context

Heap approach

Cons :

- frame linkage costs
- garbage collection overhead
- decreased locality of reference.

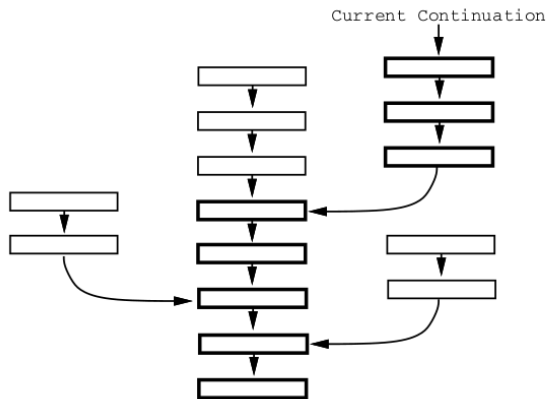
Naive Stack approach

Cons :

- unbounded copying overhead from copying creating continuation.

Control structure : Linked list of activation records

In the Heap

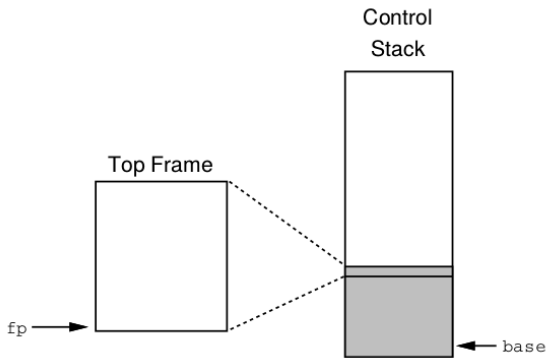


Continuation operation : Pointer operation (Quick)

Cons : maintaining pointers on each call of functions

Control structure : Record in the Stack

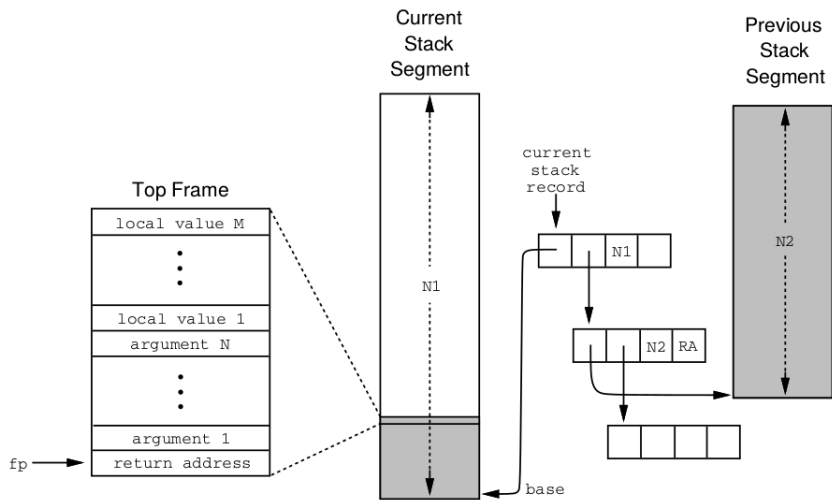
In the Stack



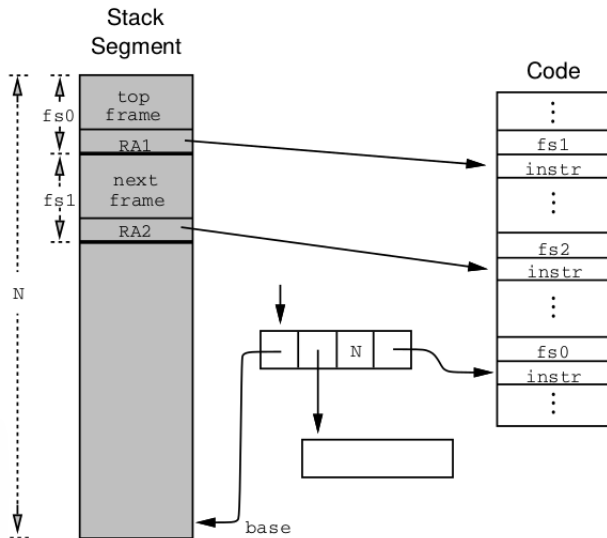
Continuation operation : Saving and copying stack frames

Cons : cost of continuation proportional to the size of the stack

Mixed Heap-stack Control structure

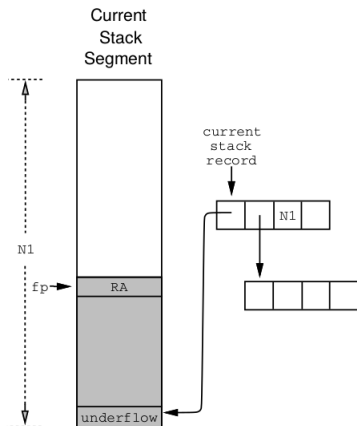


Semantic of the Control structure

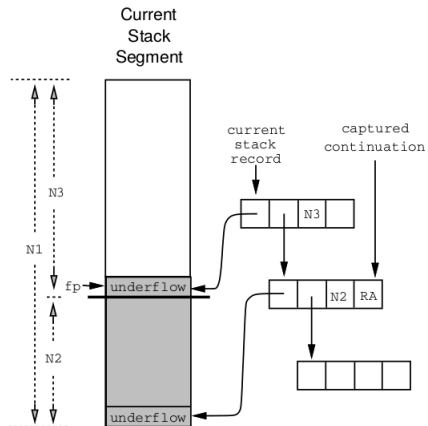


Capturing Continuation

Before Capture

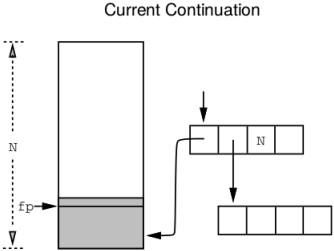


After Capture

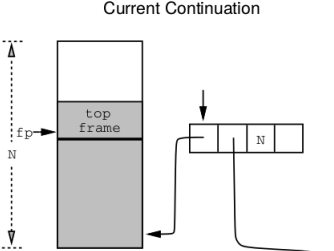


Reinstating Continuation

Before Reinstatement



After Reinstatement



Reinstating Continuation

Advantages of the approach :

- Constant time capture of continuation
- Bounded time copy for reinstating continuation
- Efficient Stack overflow recovery
- less heap space