Compiler and Runtime Support for Continuation Marks

Matthew Flatt
University of Utah

R. Kent Dybvig
Cisco Systems, Inc.

Why Continuation Marks

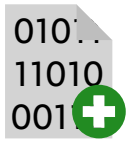
```
open "data.txt"
```

Why Continuation Marks

```
open "data.txt"
     'create
```

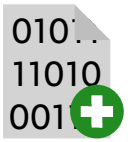
Why Continuation Marks

```
open "data.txt"
     'create
     'binary
```



Why Continuation Marks

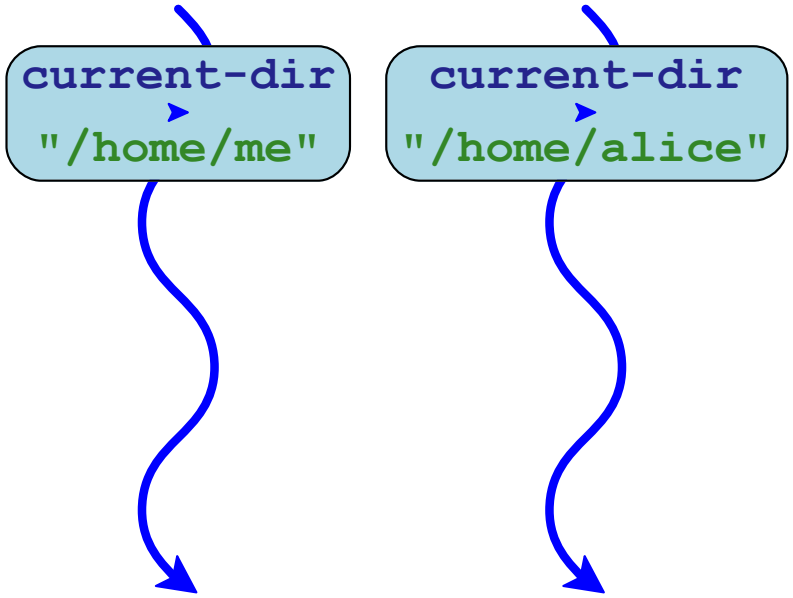
```
open "data.txt"
     'create
     'binary
```




```
current-dir
    ➤
"/home/me"
```

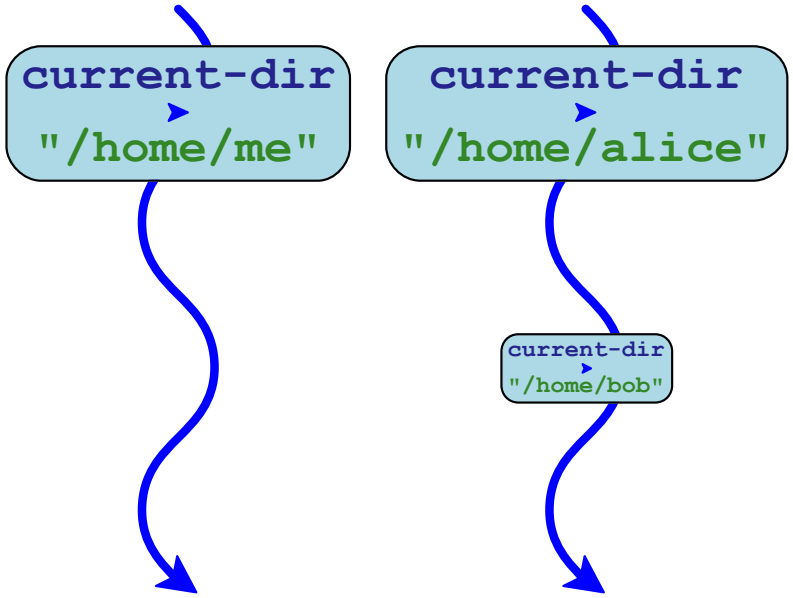
Why Continuation Marks

```
open "data.txt"
     'create
     'binary
```

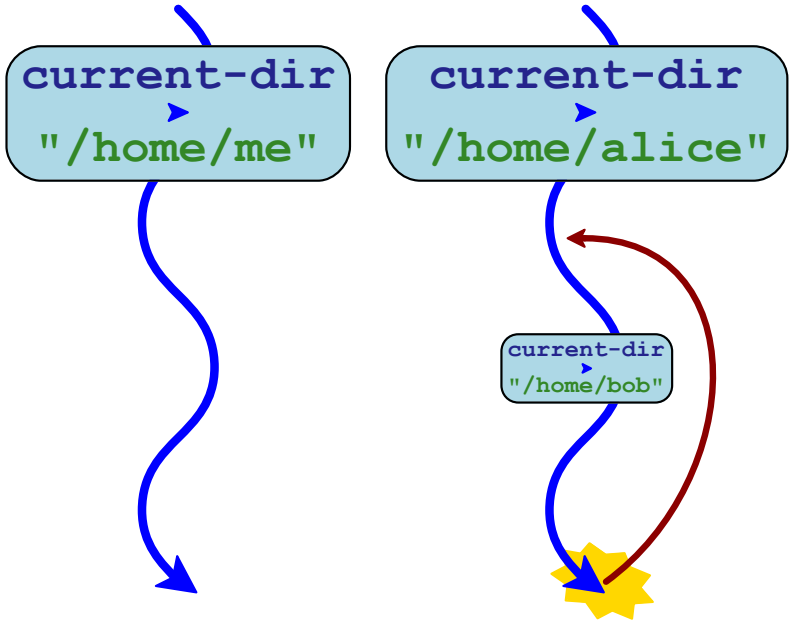
Why Continuation Marks

```
open "data.txt"
     'create
     'binary
```

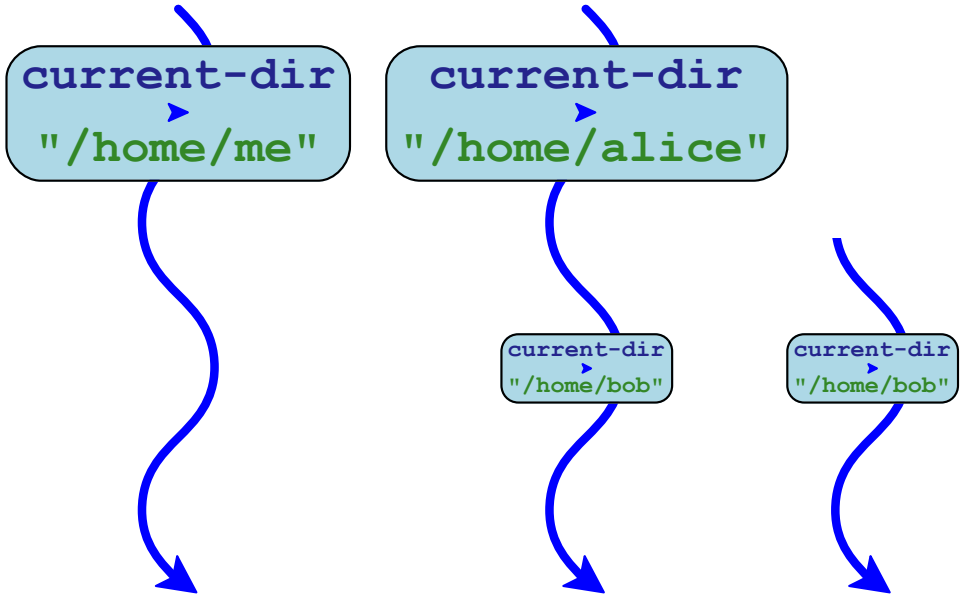
Why Continuation Marks

```
open "data.txt"
     'create
     'binary
```



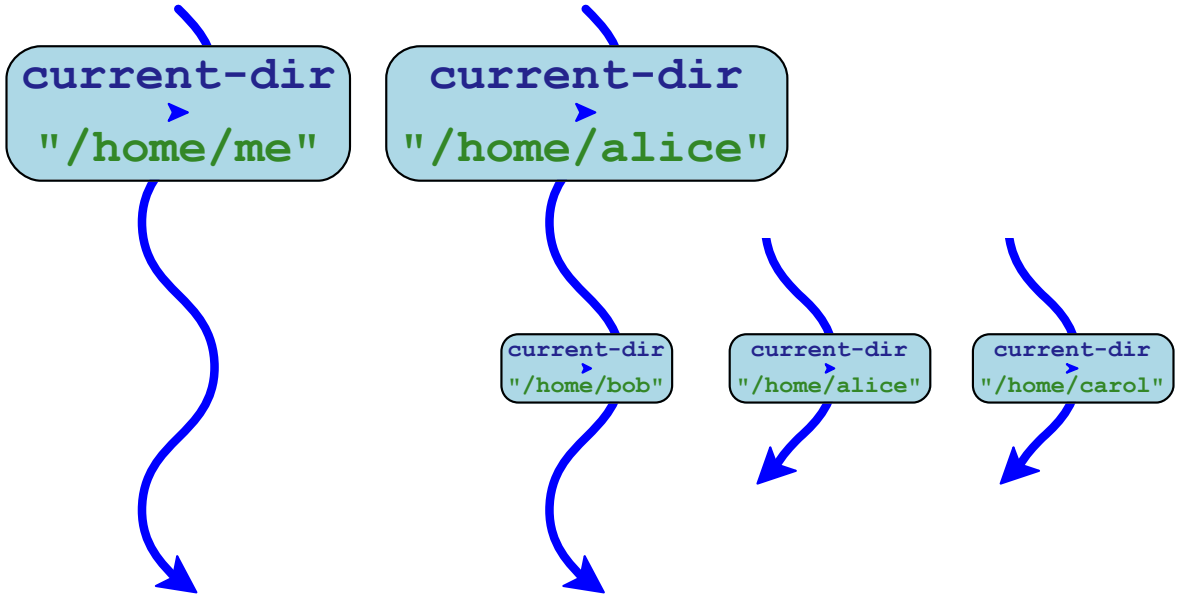
Why Continuation Marks

```
open "data.txt"
     'create
     'binary
```



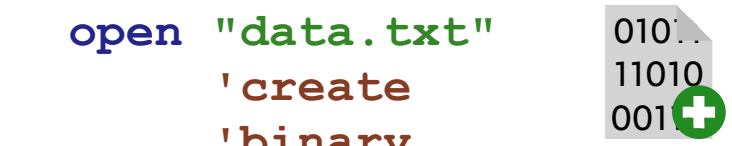
Why Continuation Marks

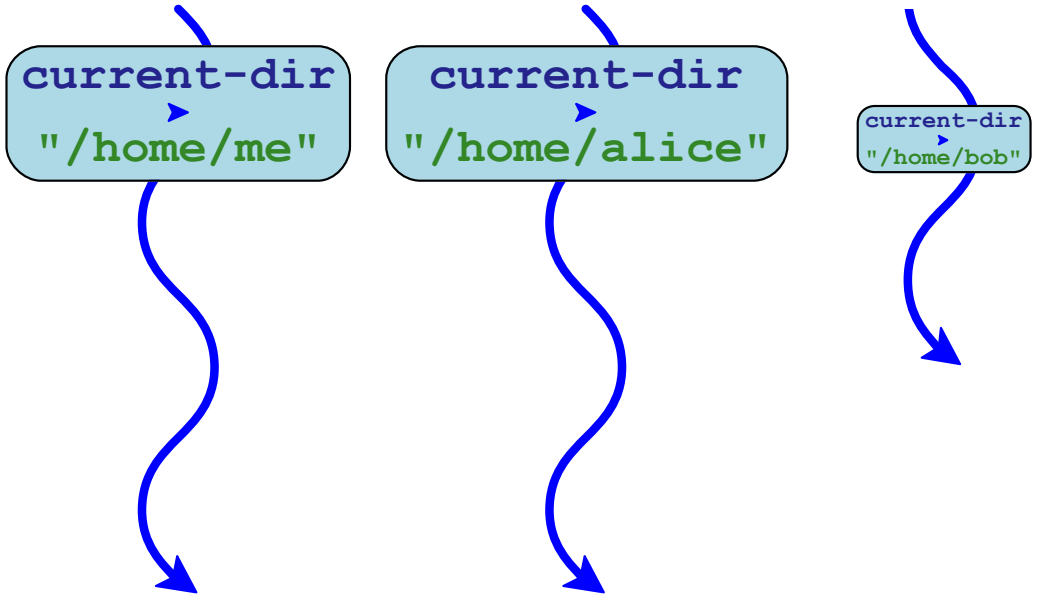
```
open "data.txt"
     'create
     'binary
```



Why Continuation Marks

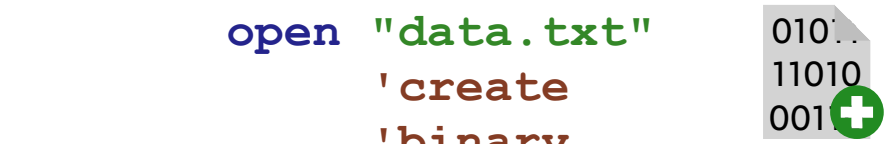
```
open "data.txt"
     'create
     'binary
```

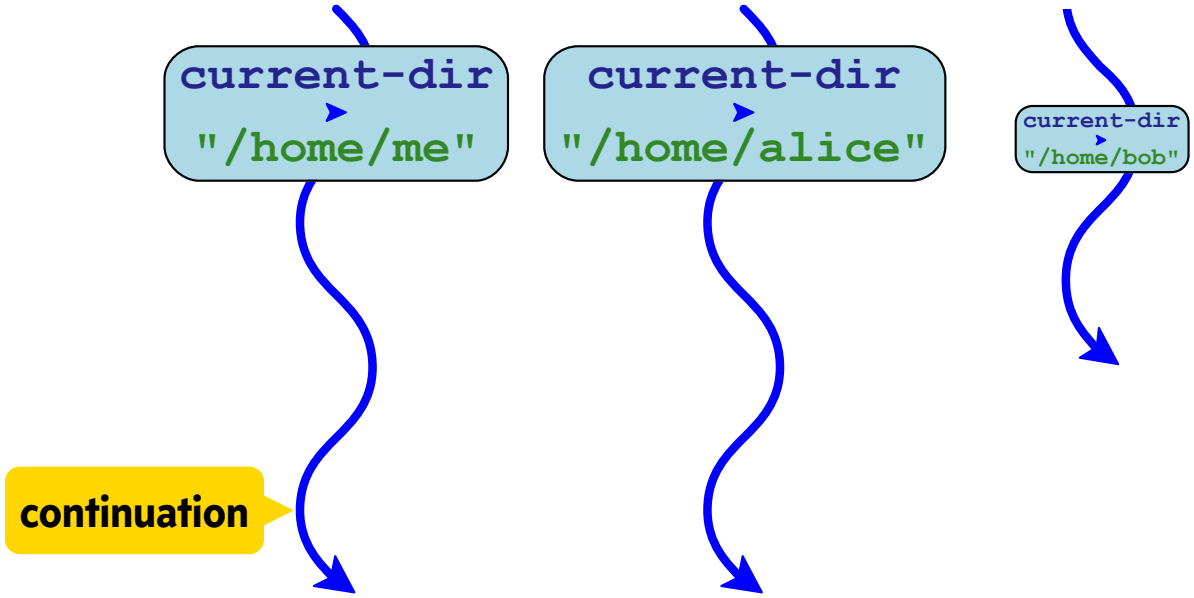




Why Continuation Marks

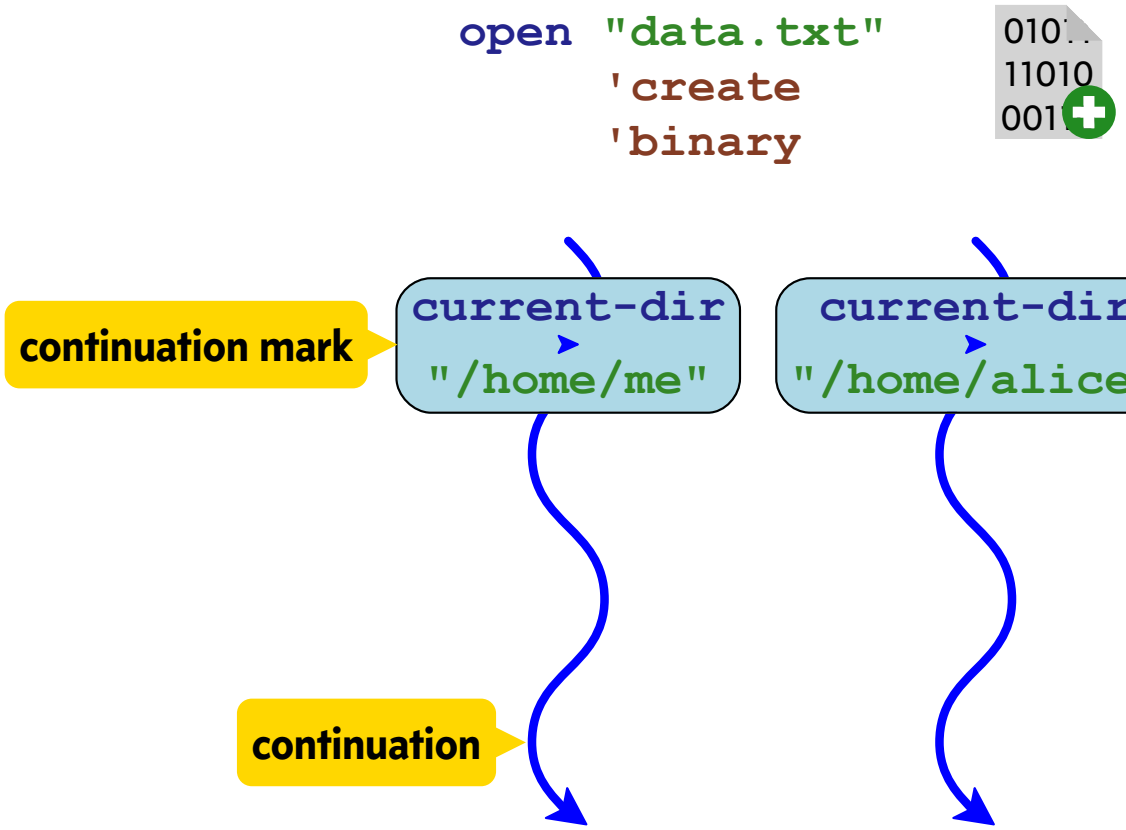
```
open "data.txt"
     'create
     'binary
```





Why Continuation Marks

```
open "data.txt"
     'create
     'binary
```



Continuation Mark Primitives

```
(parameterize ([current-directory "/home/alice"])
  (shuffle-files))
```

```
(define (path->compete-path path)
  (build-path (current-directory)
              path))
```

Continuation Mark Primitives

```
(with-mark
  current-directory "/home/alice"
  (shuffle-files))
```

```
(define (path->compete-path path)
  (build-path (current-directory)
              path))
```

Continuation Mark Primitives

```
(with-mark
  current-directory "/home/alice"
  (shuffle-files))
```

```
(define (path->compete-path path)
  (build-path (current-mark current-directory)
              path))
```

Continuation Mark Primitives

```
(with-mark
  current-directory "/home/alice"
  (shuffle-files))
```

```
(define (path->compete-path path)
  (build-path (first
               (current-marks current-directory))
              path))
```

Uses of Continuation Marks

- dynamic binding [ICFP'07]
- exception handlers
- debugging [ESOP'01, SLE'17]
- profiling [TOPLAS'19]
- generators
- serializable continuations [ICFP'09]
- security checks [OOPSLA'16]
- avoiding reundant contracts [OOPSLA'18]

Running **Racket** on **Chez Scheme**

Racket	delimited continuations with marks	state-of-the-art **functionality**
Chez Scheme	continuations via `call/cc`	state-of-the-art **performace**

Running **Racket** on **Chez Scheme**

good performance via `call/cc`

Racket	delimited continuations with marks	state-of-the-art **functionality**
Chez Scheme	continuations via `call/cc`	state-of-the-art **performace**

Running **Racket** on **Chez Scheme**

mediocre peformance via `call/cc`

Racket	delimited continuations with marks	state-of-the-art **functionality**
Chez Scheme	continuations via `call/cc`	state-of-the-art **performace**

Adding Marks to Chez Scheme

speedup relative to using `call/cc`

continuation marks	*contracts*	*applications*
×3 – ×22	**×3.4**	**×1.10 – ×1.25**

Notation

$(v_1\ (((\lambda\ (x)\ x)\ v_3)\ v_2))$

"Adding Delimited and Composable Control to a Production Programming Environment"
ICFP'07 Flatt, Yu, Findler, and Felleisen

Notation

$(v_1\ ((\lambda\ (x)\ x)\ v_3)\ v_2)$

Notation

$((\lambda\ (x)\ x)\ v_3)$

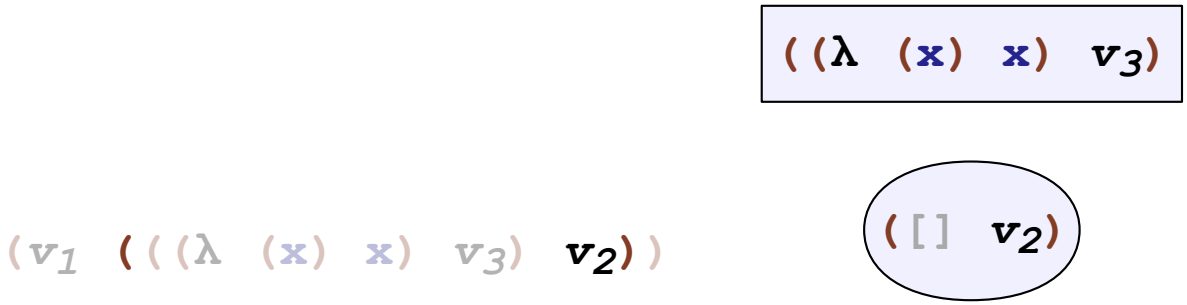
$(v_1\ ((\lambda\ (x)\ x)\ v_3)\ v_2))$

Notation

$((\lambda\ (x)\ x)\ v_3)$

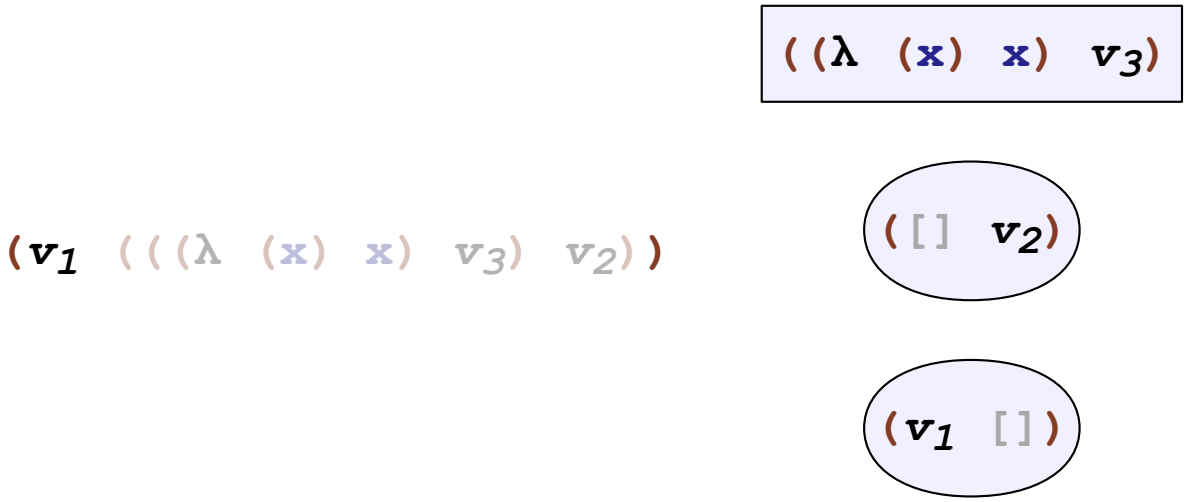
$(v_1\ ((\lambda\ (x)\ x)\ v_3)\ v_2)$

Notation



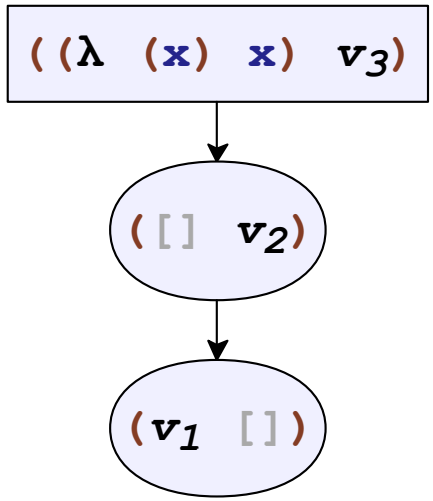
Notation

Notation

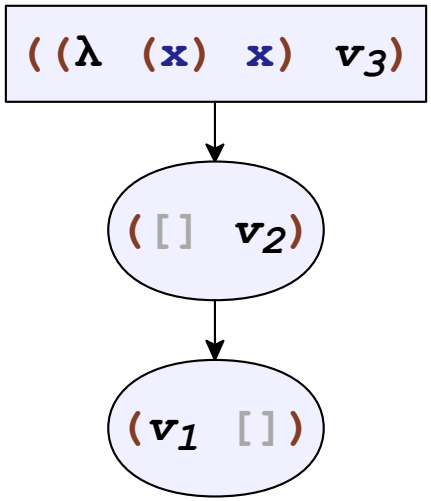


Notation

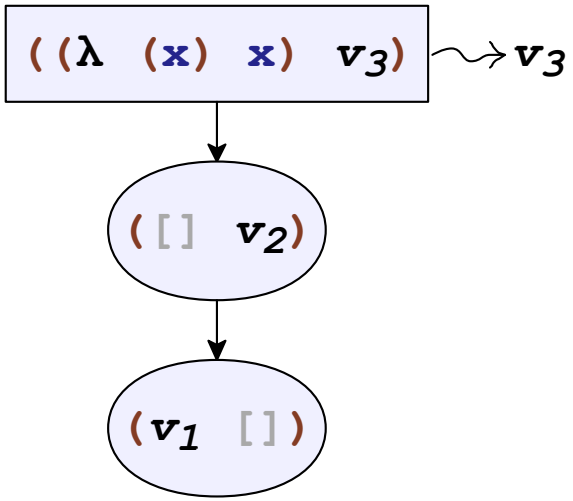
$(v_1\ ((\lambda\ (x)\ x)\ v_3)\ v_2)) =$



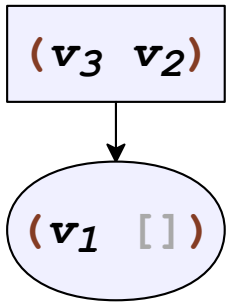
Reductions



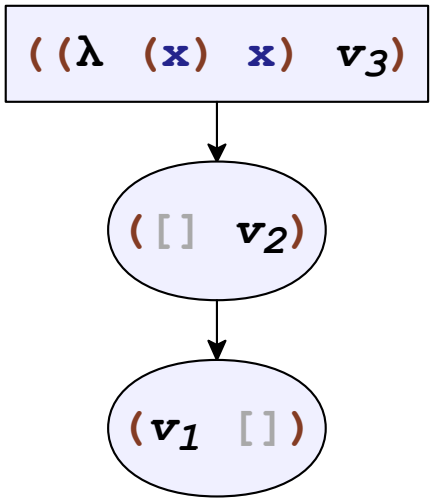
Reductions



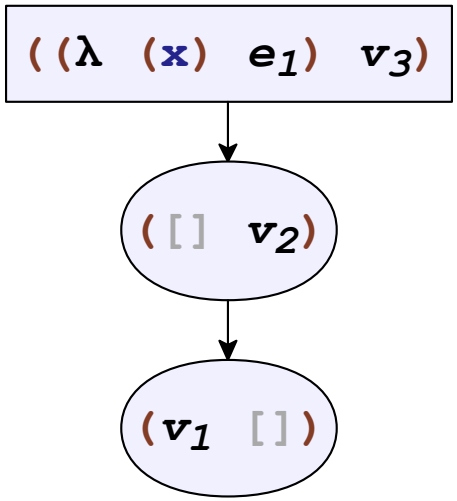
Reductions



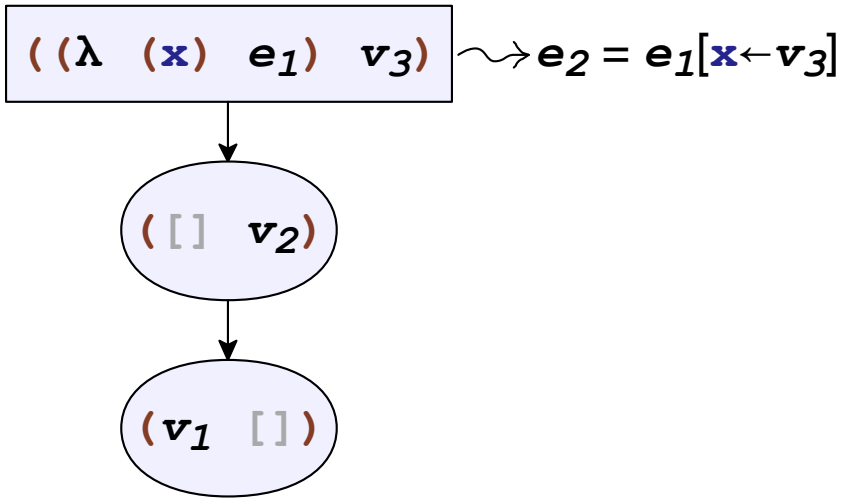
Reductions



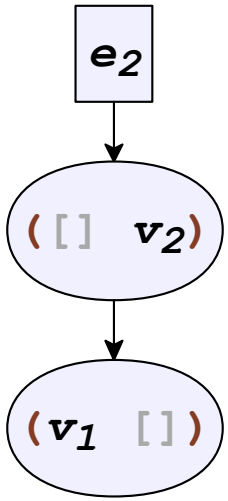
Tail Evaluation



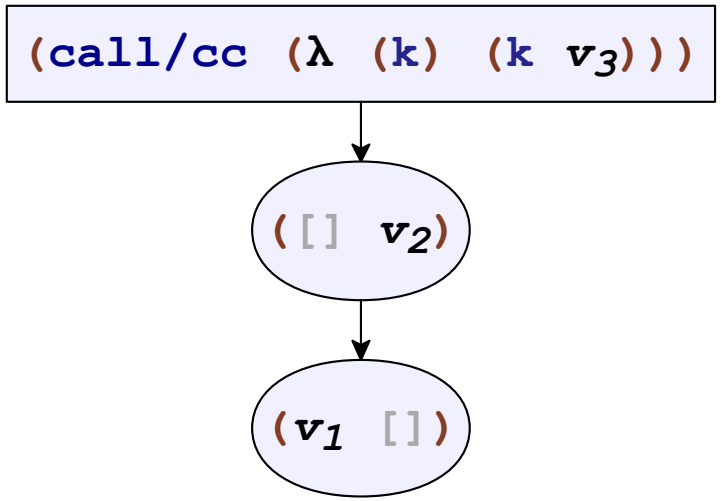
Tail Evaluation



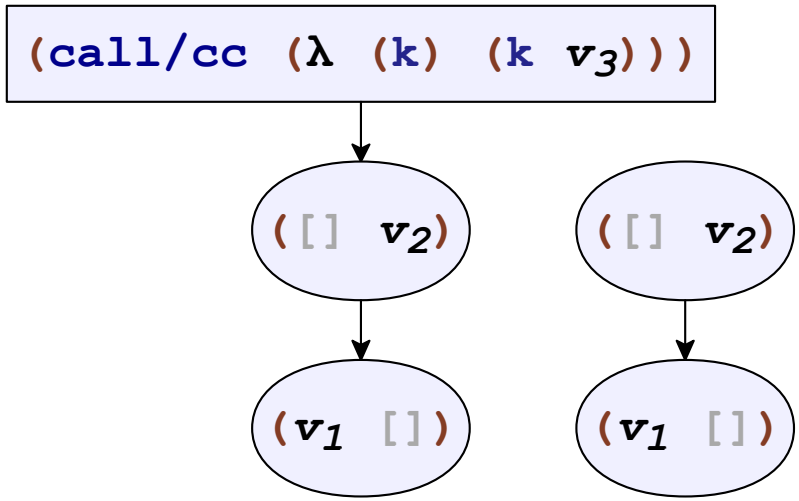
Tail Evaluation



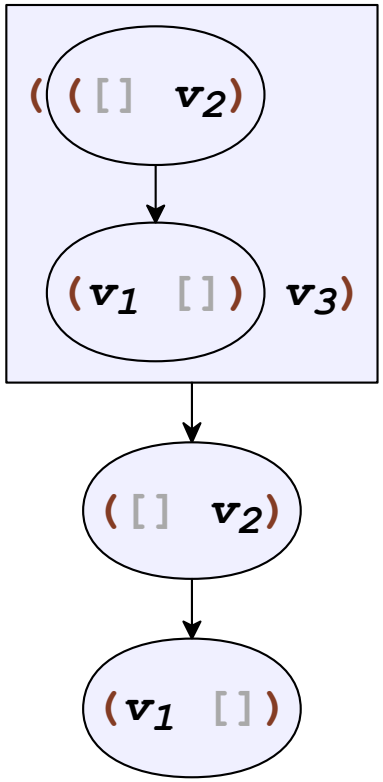
Continuations



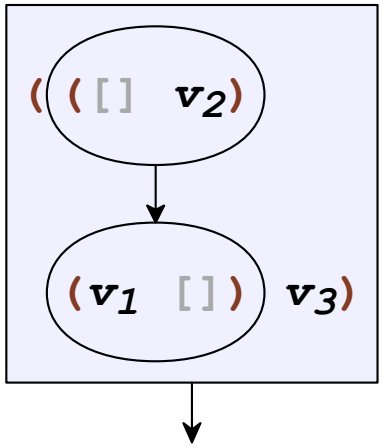
Continuations



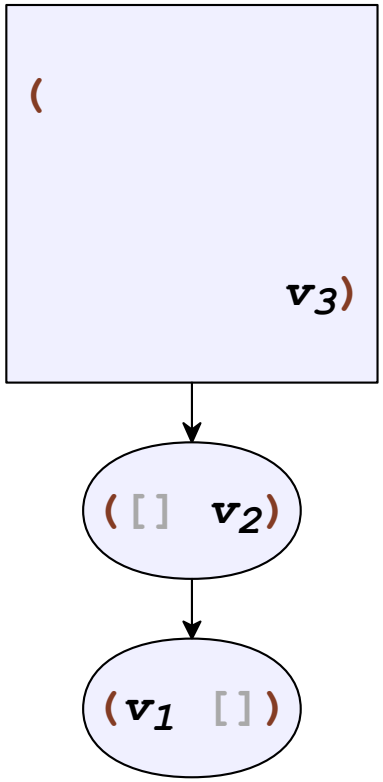
Continuations



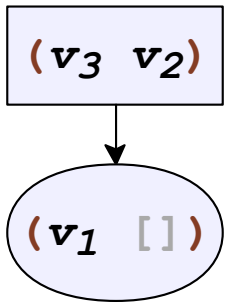
Continuations



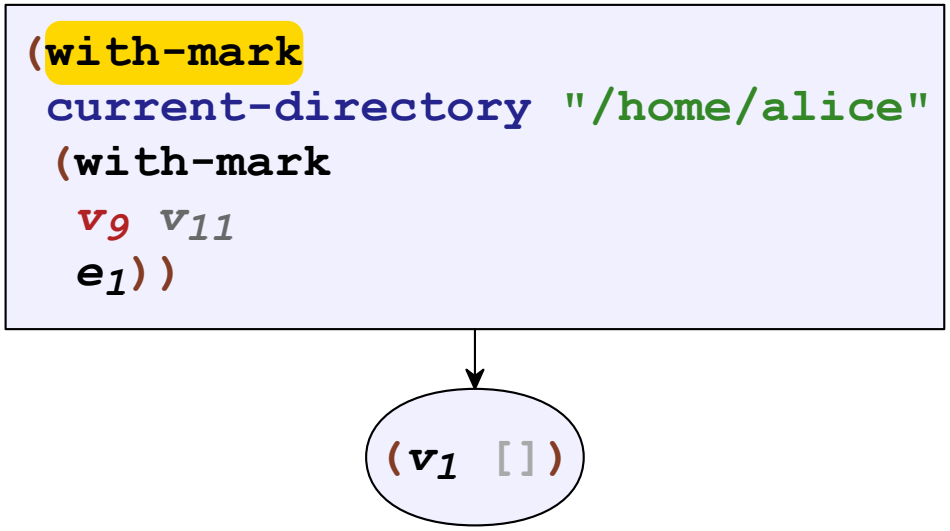
Continuations



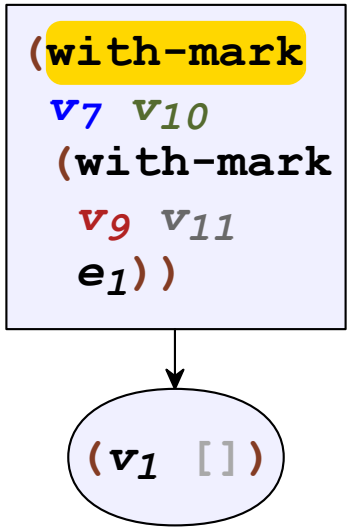
Continuations



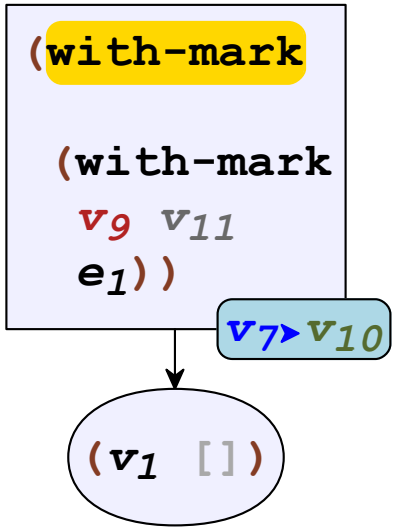
Continuation Marks



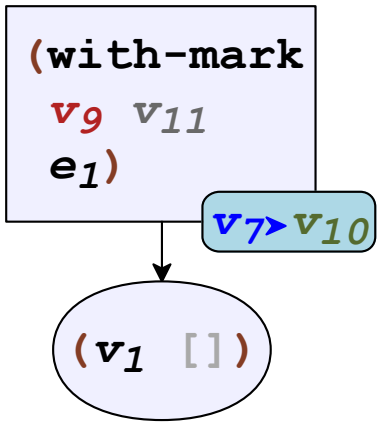
Continuation Marks



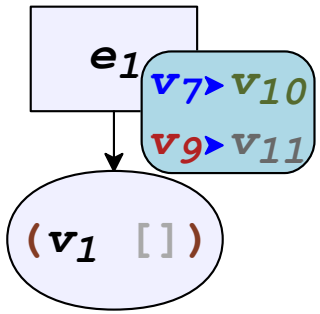
Continuation Marks



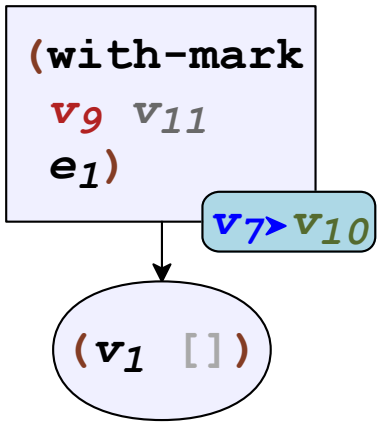
Continuation Marks



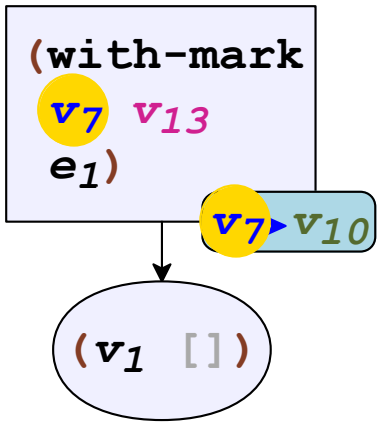
Continuation Marks



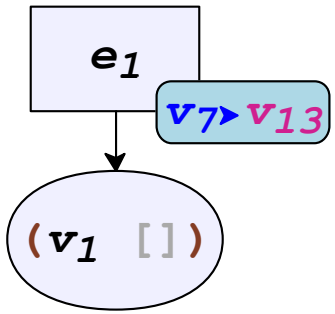
Continuation Marks



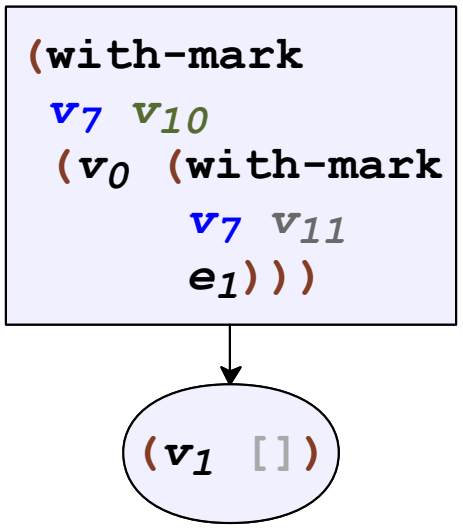
Continuation Marks



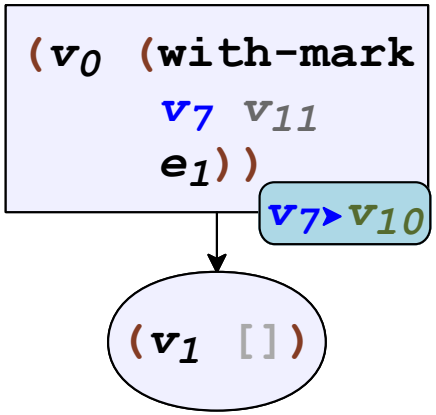
Continuation Marks



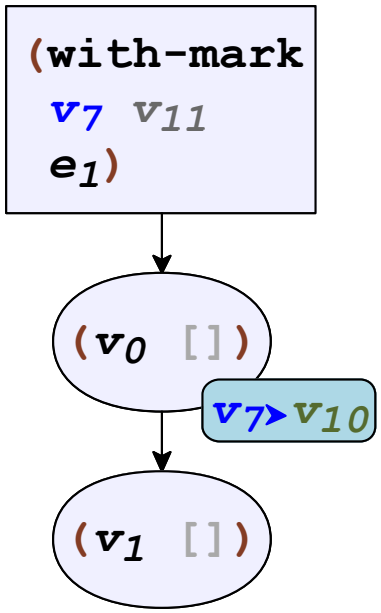
Continuation Marks



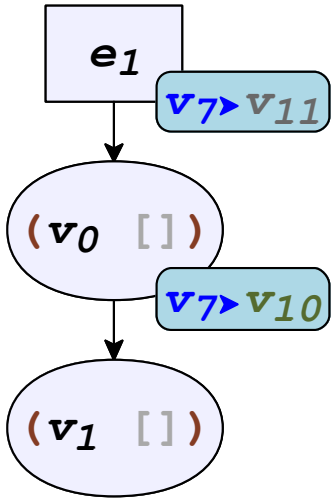
Continuation Marks



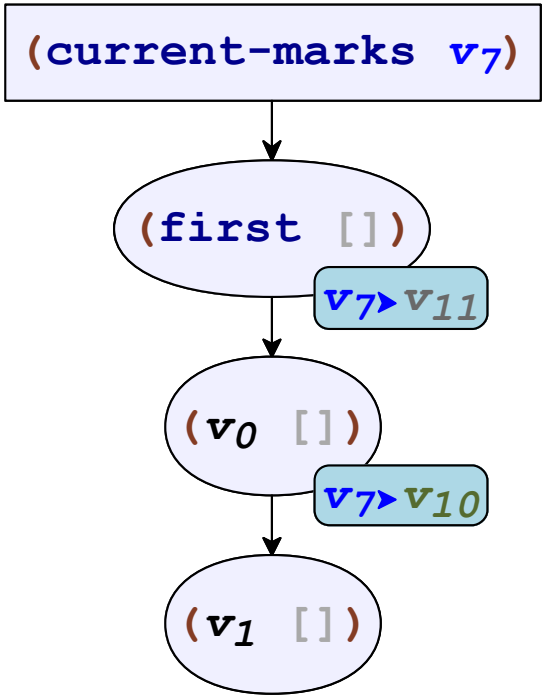
Continuation Marks



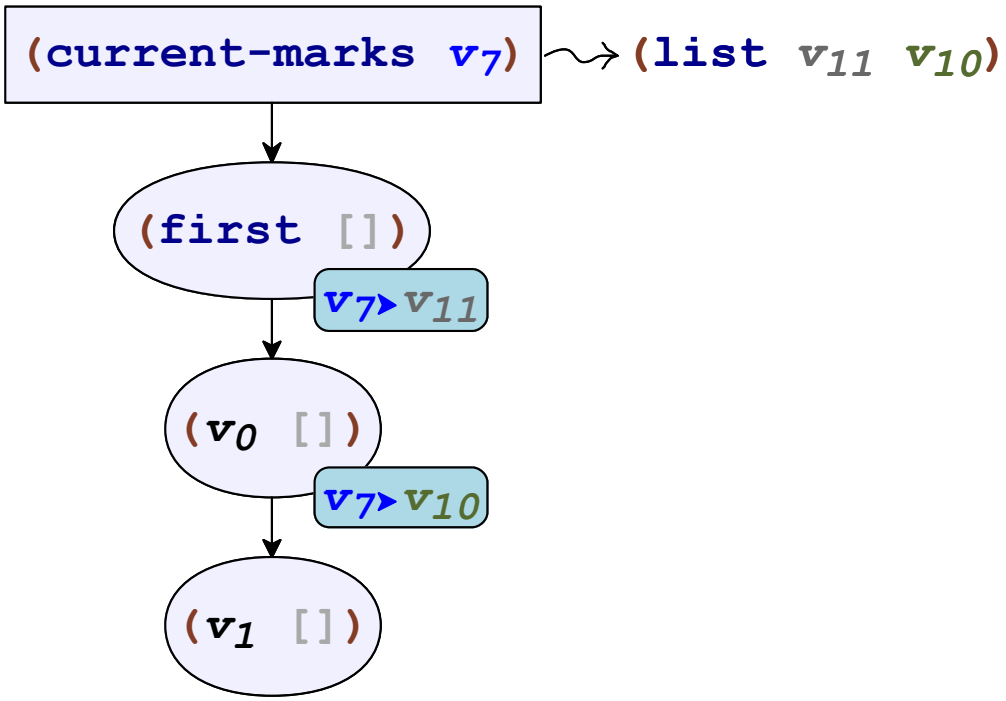
Continuation Marks



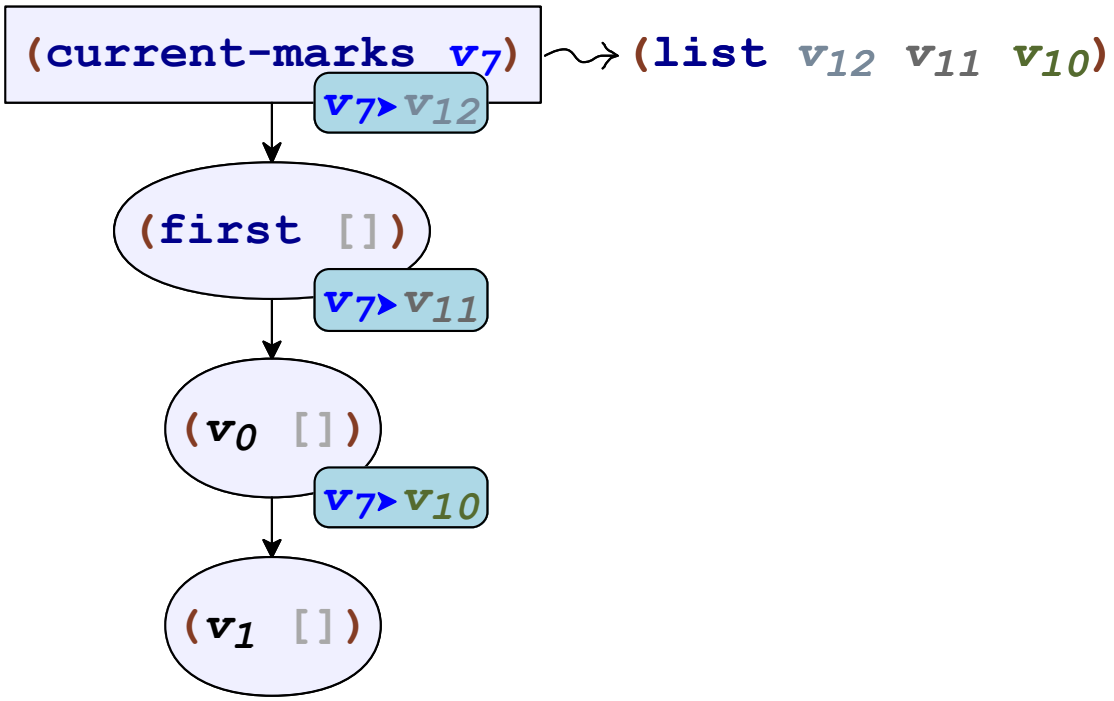
Continuation Marks



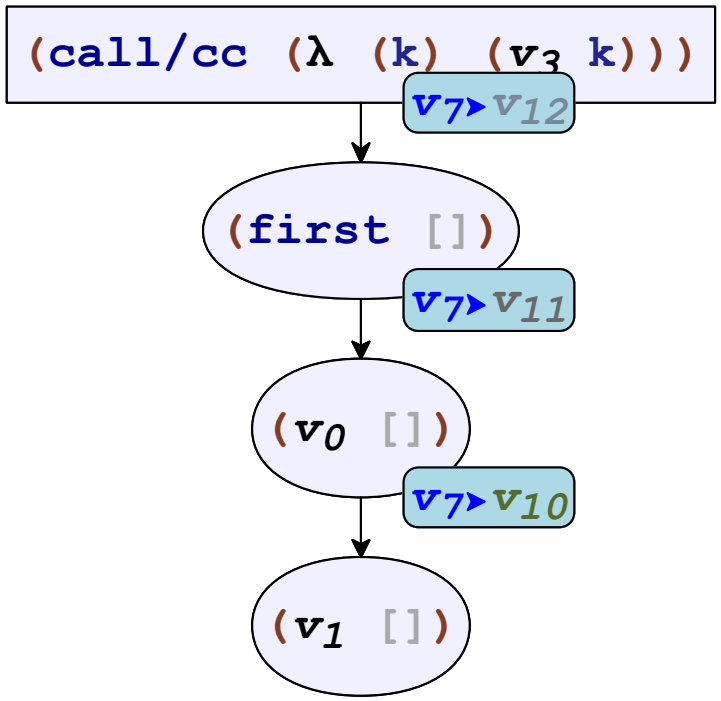
Continuation Marks



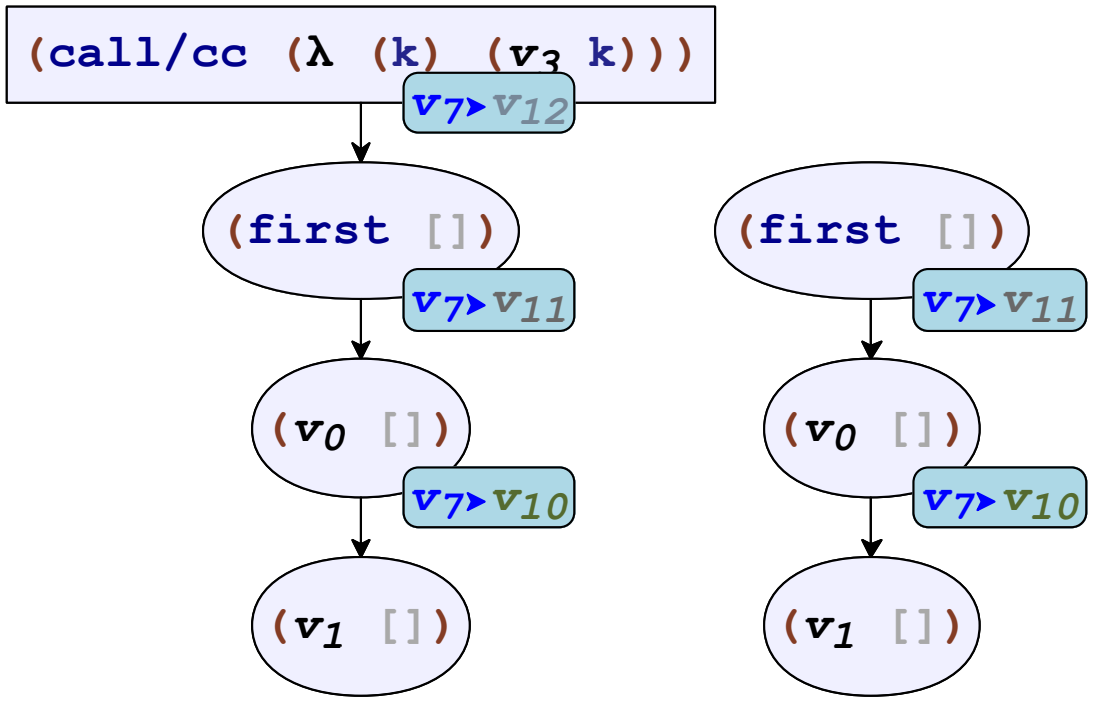
Continuation Marks



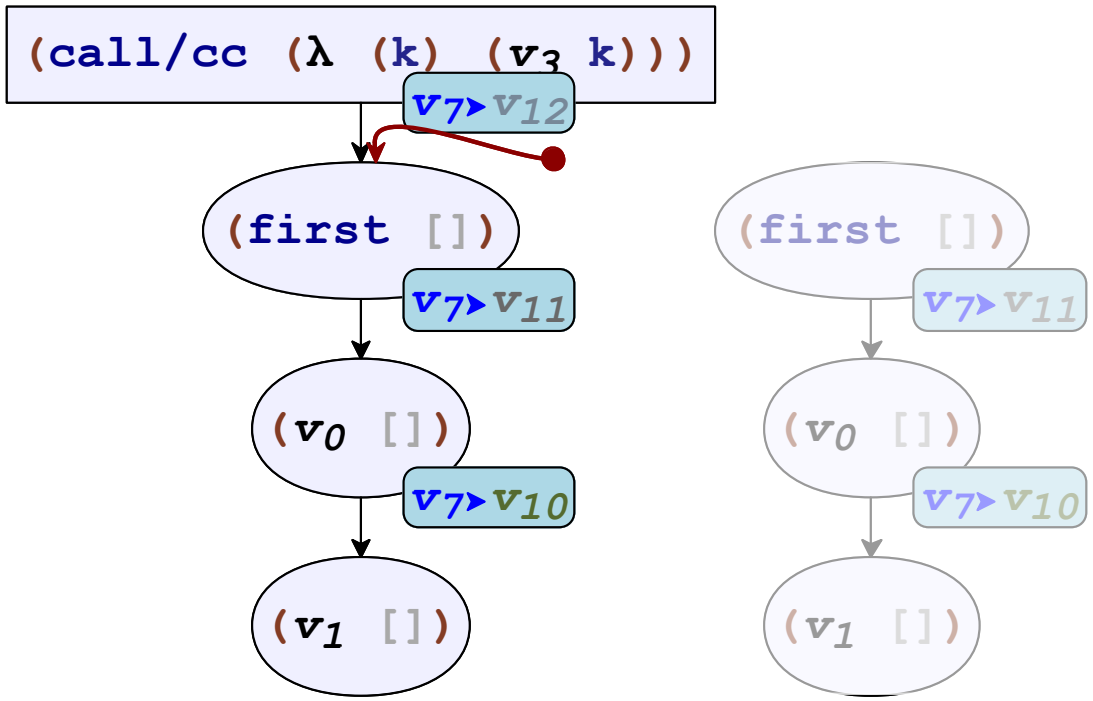
Capturing Marks



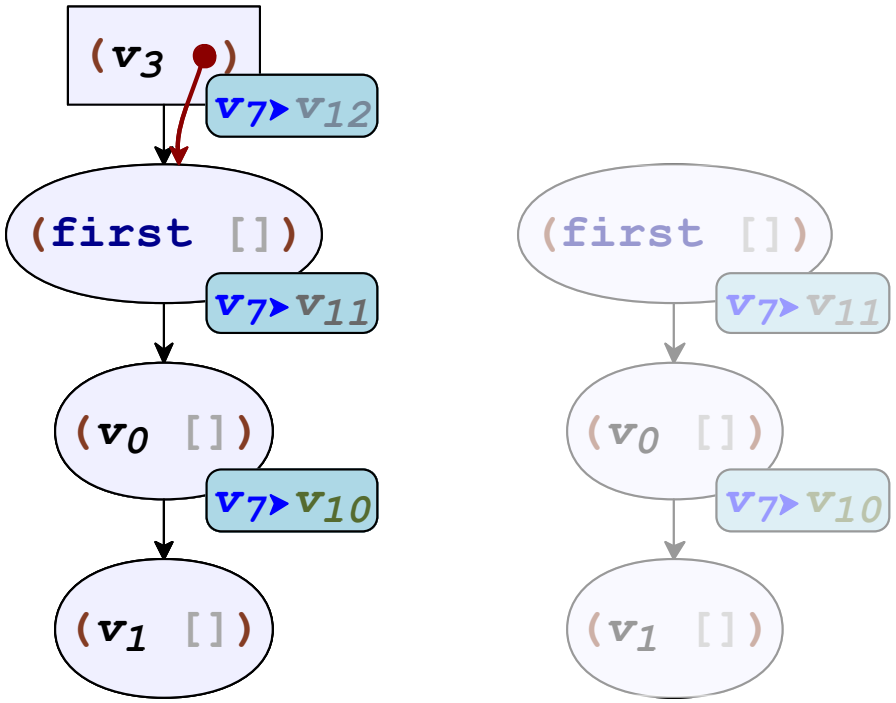
Capturing Marks



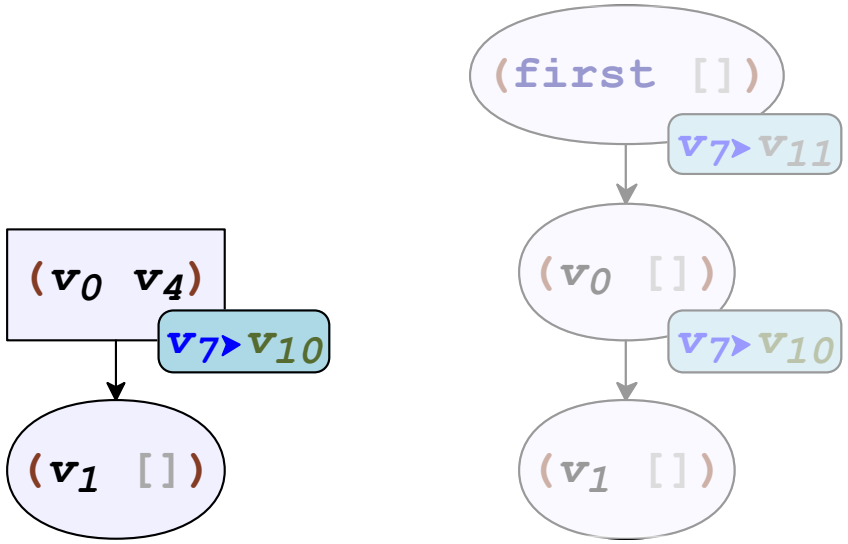
Heap-Allocated Frames



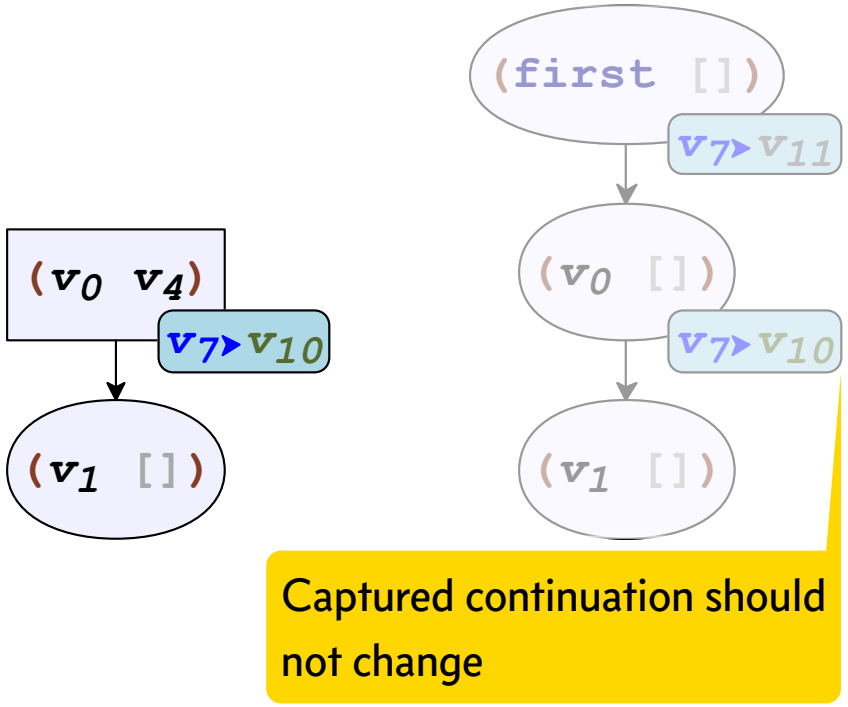
Heap-Allocated Frames



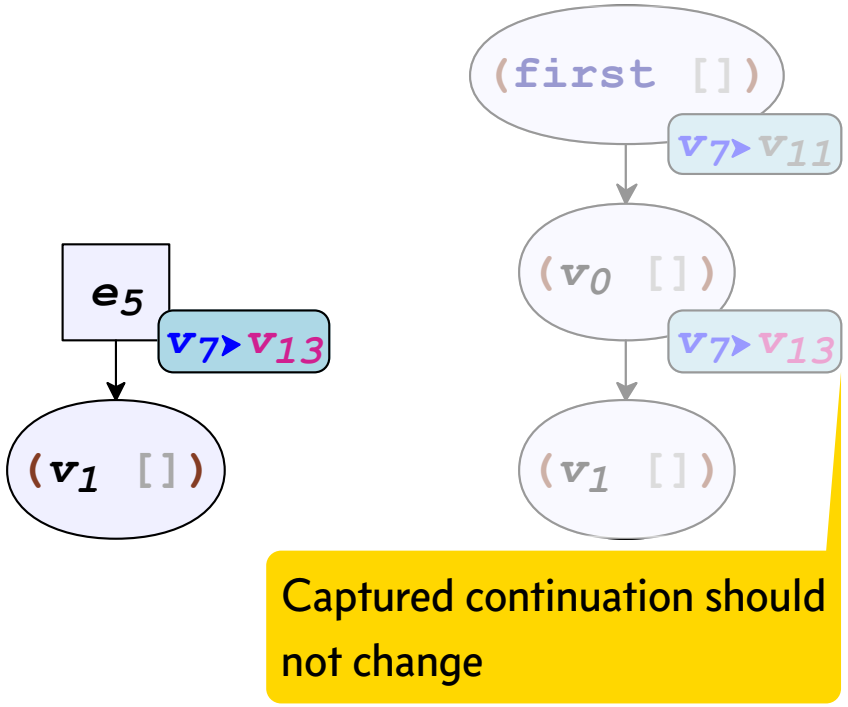
Heap-Allocated Frames



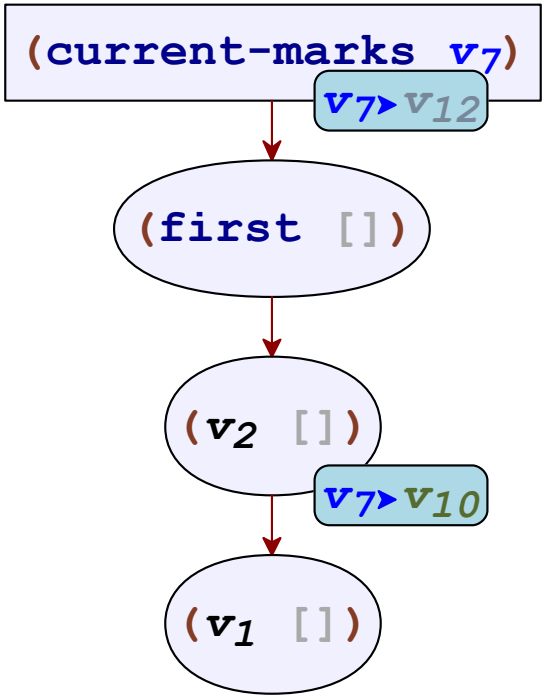
Heap-Allocated Frames



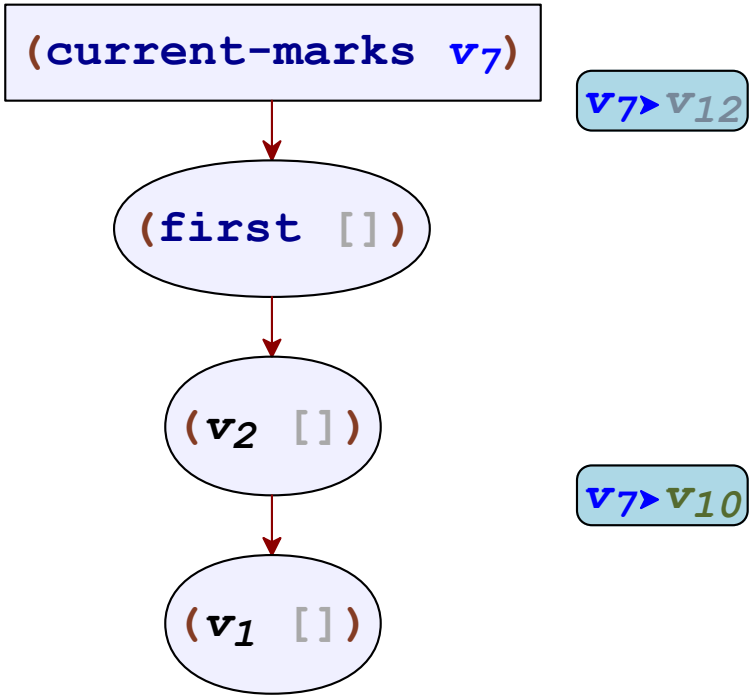
Heap-Allocated Frames



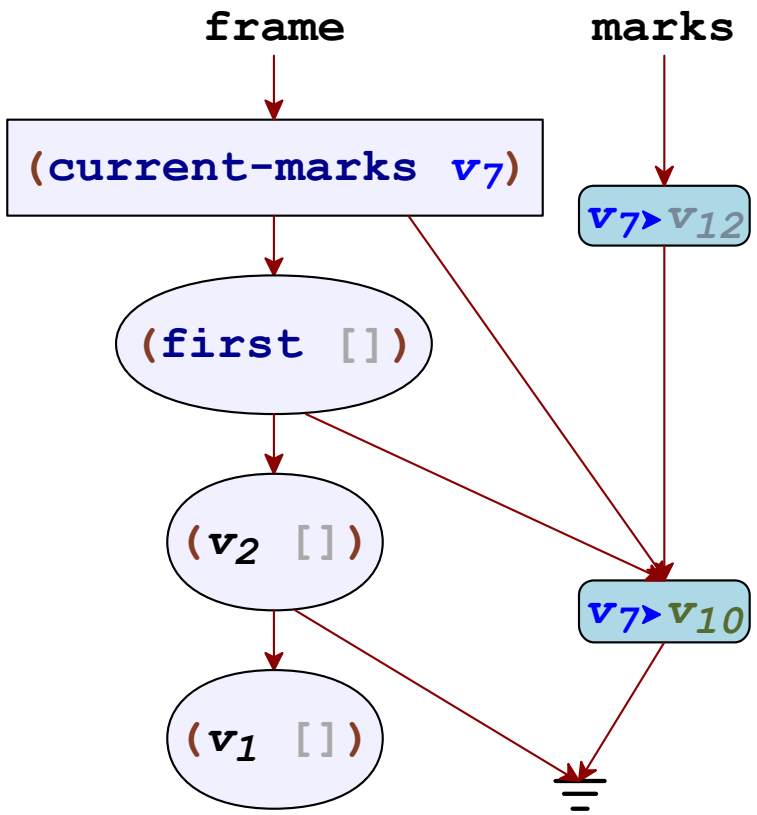
Functional Heap-Allocated Frames



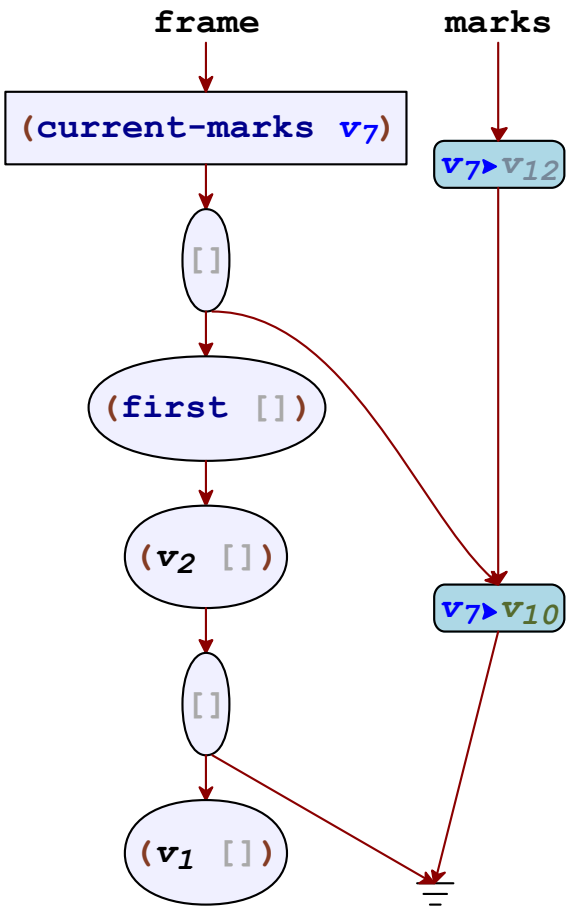
Functional Heap-Allocated Frames



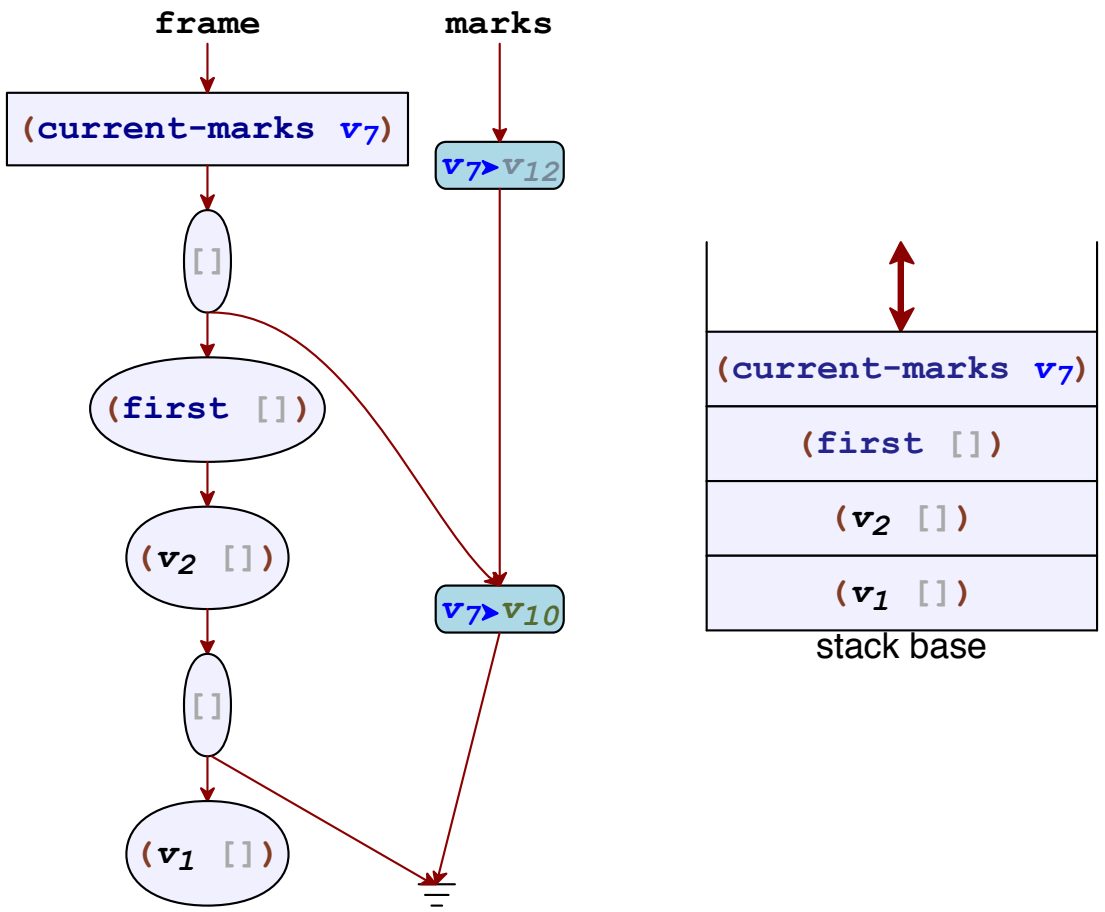
Functional Heap-Allocated Frames



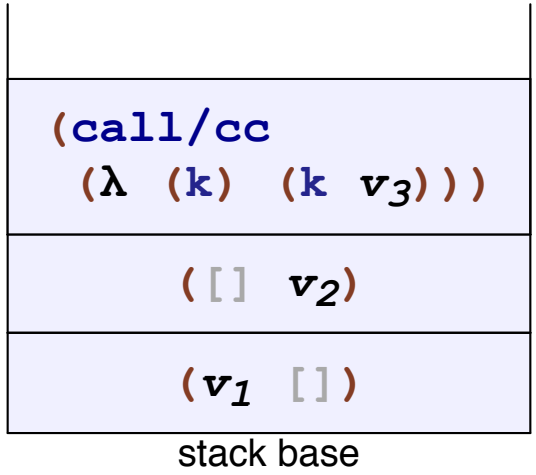
Functional Heap-Allocated Frames



Heap versus Stack

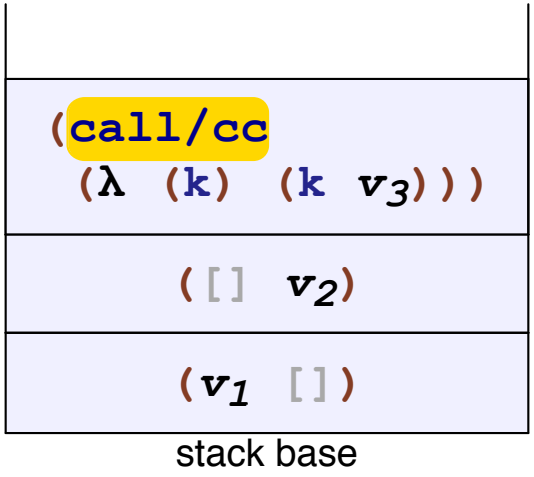


Stack-Allocated Frames

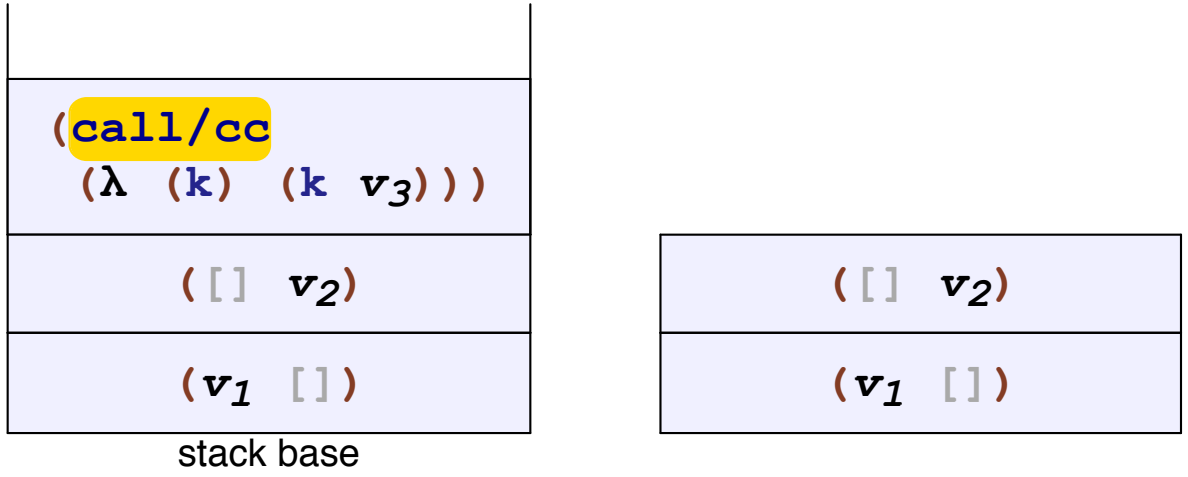


"Representing Control in the Presence of First-Class Continuations"
PLDI'90 Heib, Dybvig, and Bruggeman

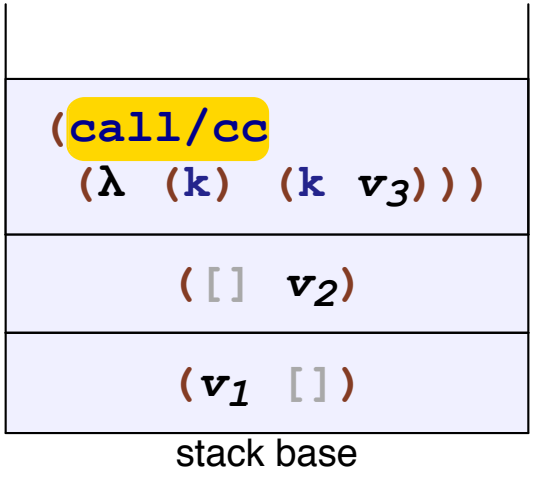
Capturing Stack-Based Continuations



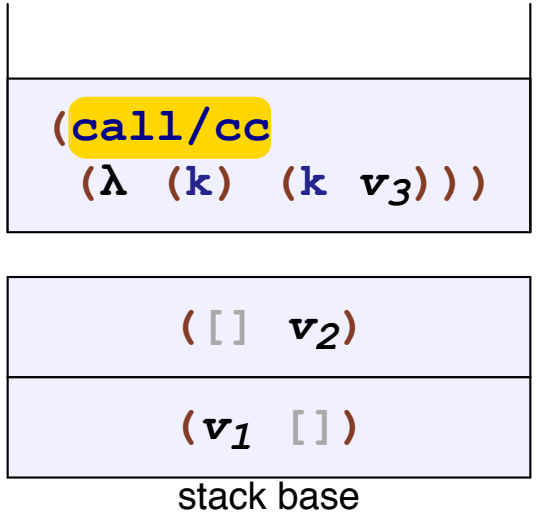
Capturing Stack-Based Continuations



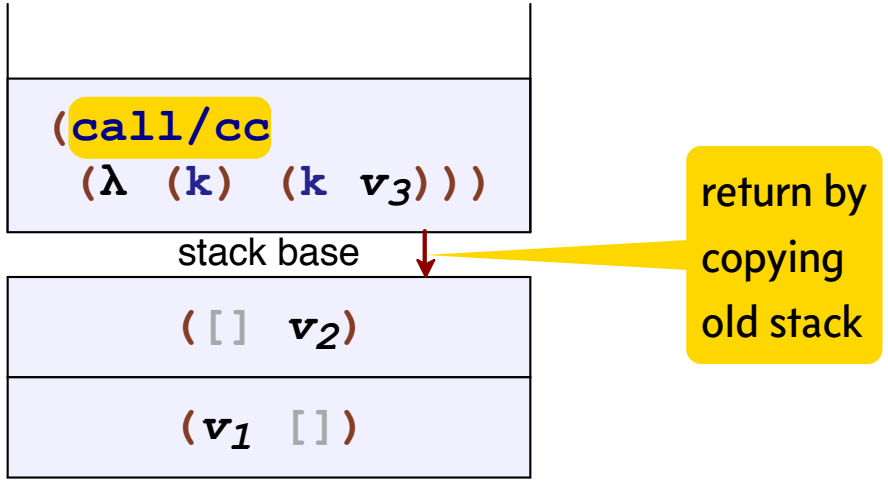
Capturing Stack-Based Continuations



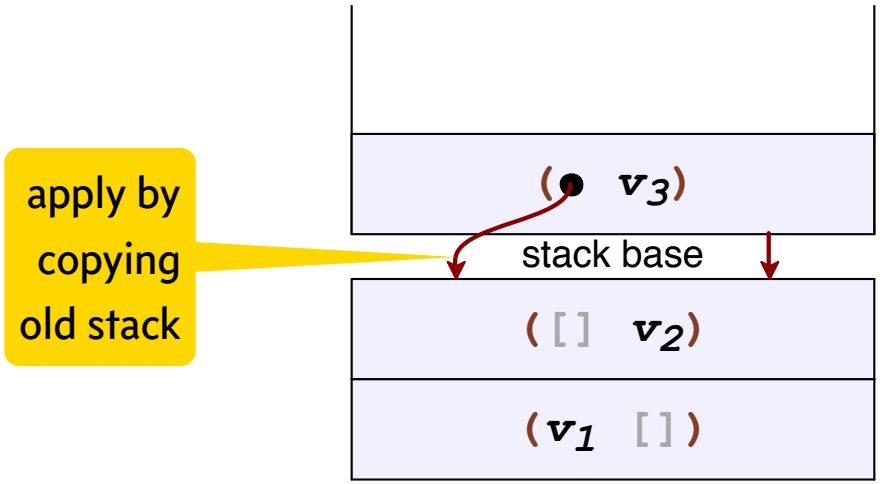
Capturing Stack-Based Continuations



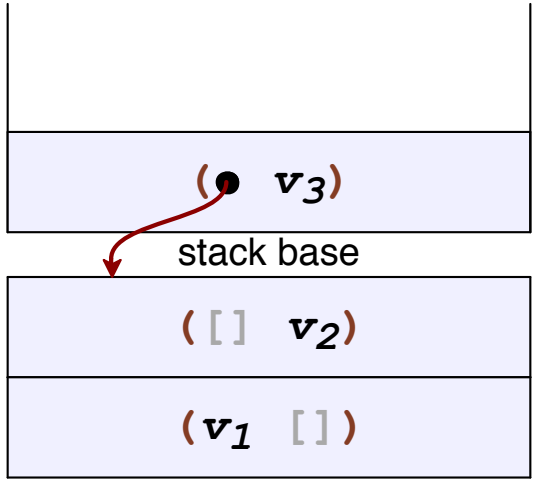
Capturing Stack-Based Continuations



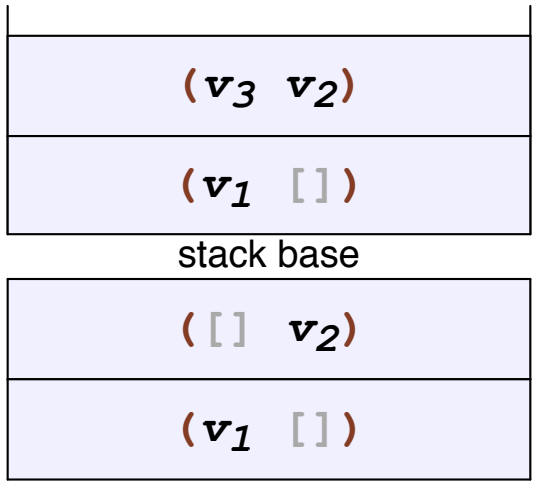
Capturing Stack-Based Continuations



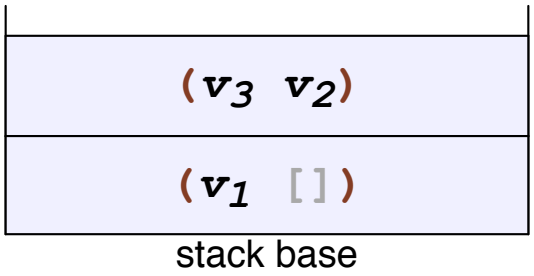
Applying Stack-Based Continuations



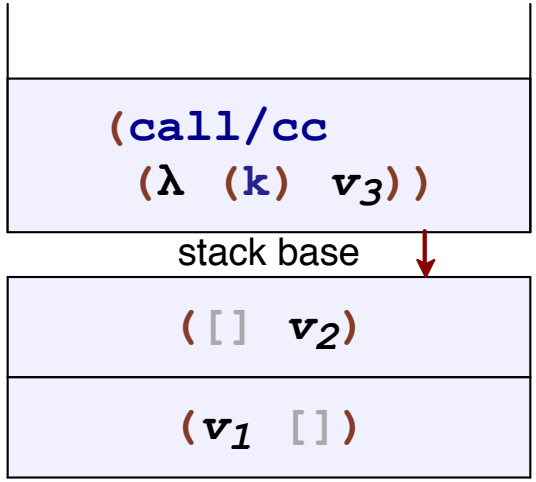
Applying Stack-Based Continuations



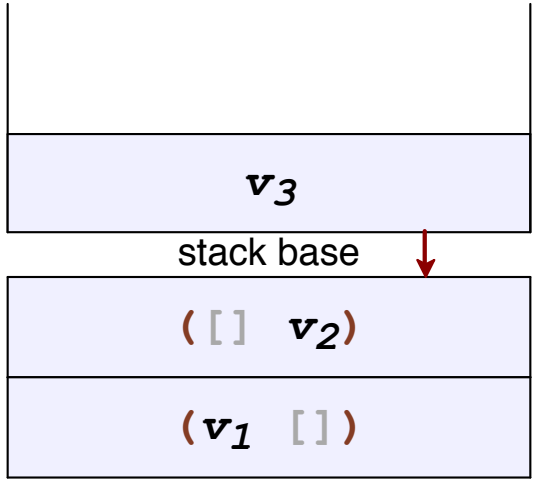
Applying Stack-Based Continuations



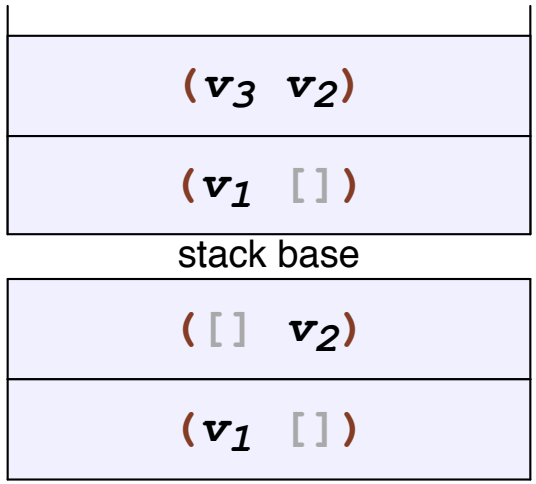
Returning After Capture



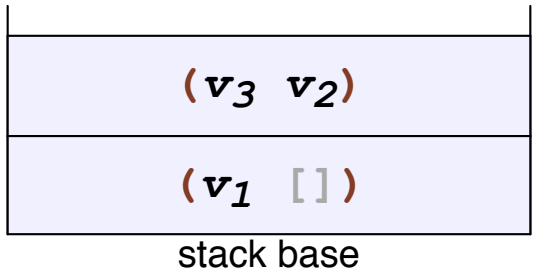
Returning After Capture



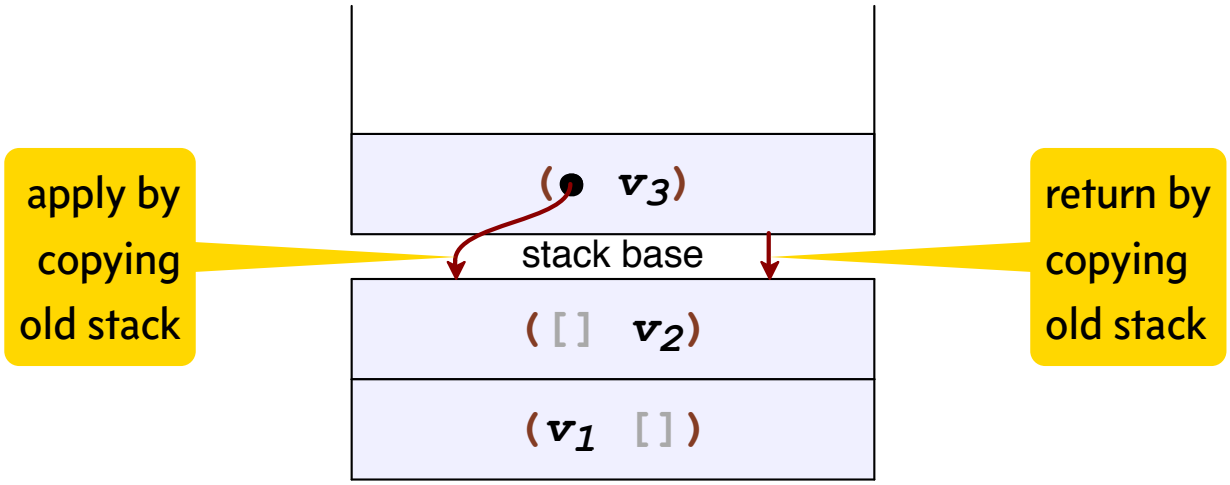
Returning After Capture



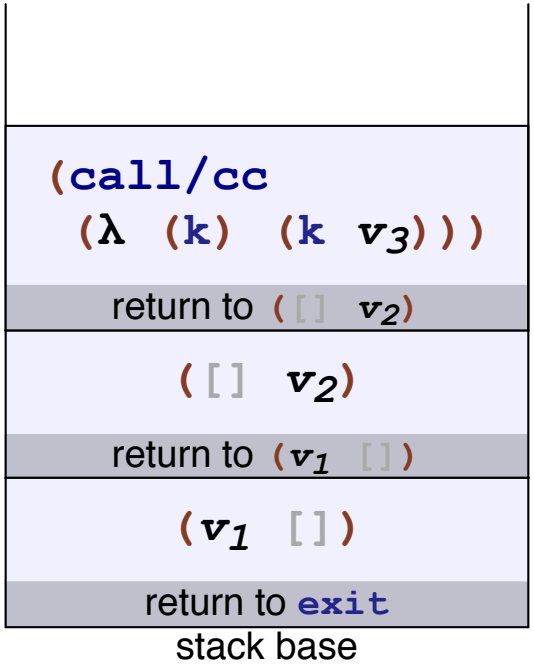
Returning After Capture



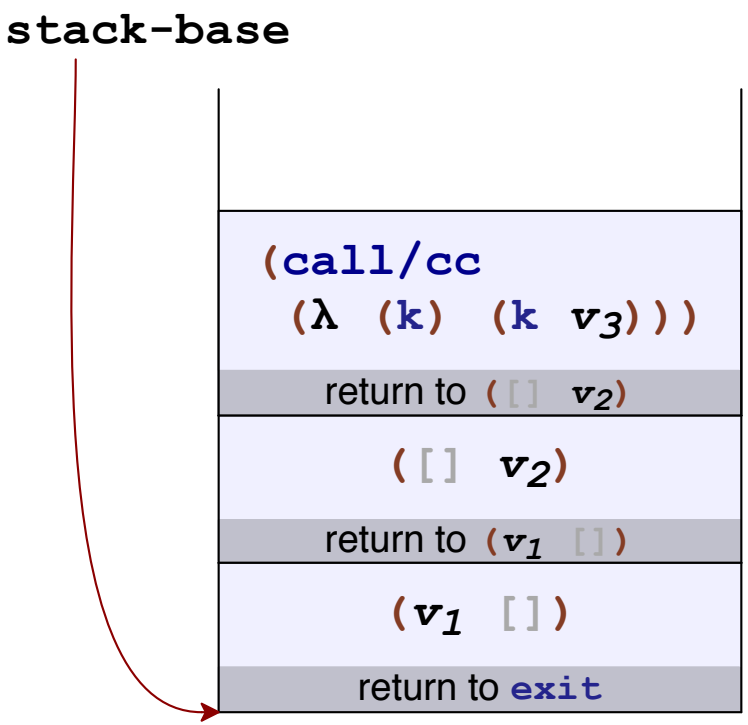
Two Kinds of Continuation Pointers



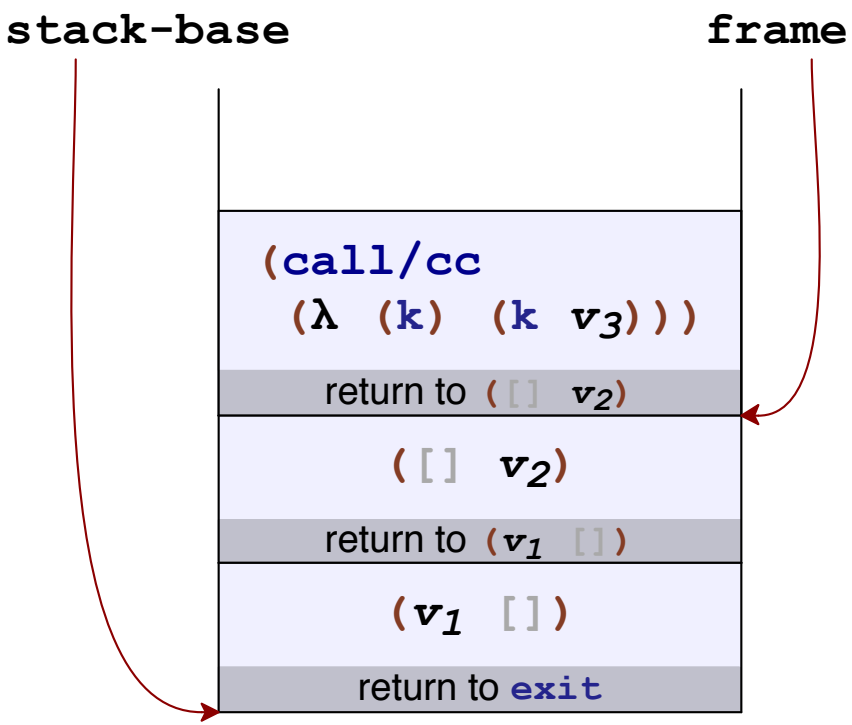
Returns in Stack Frames



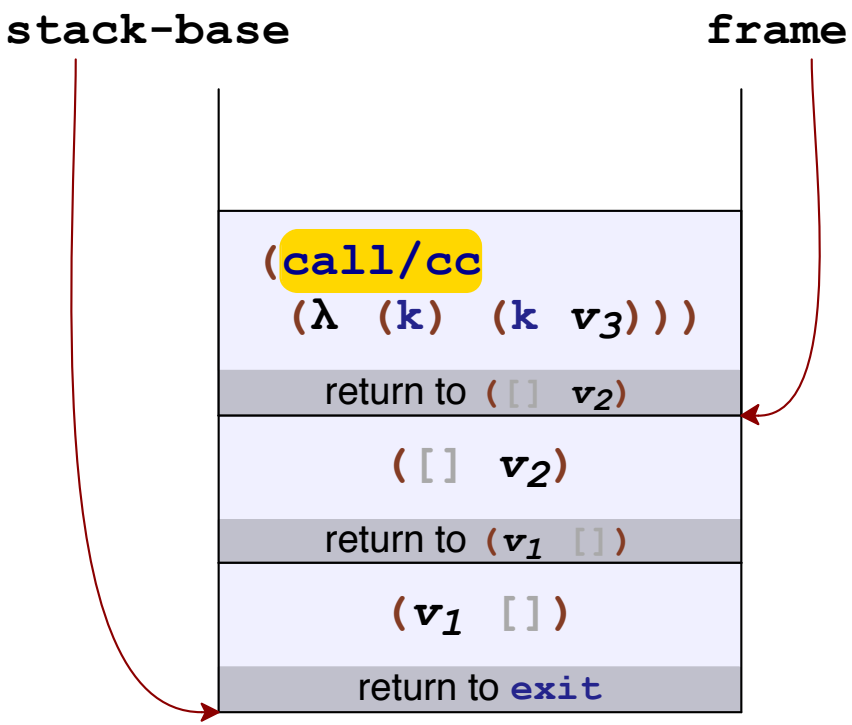
Returns in Stack Frames



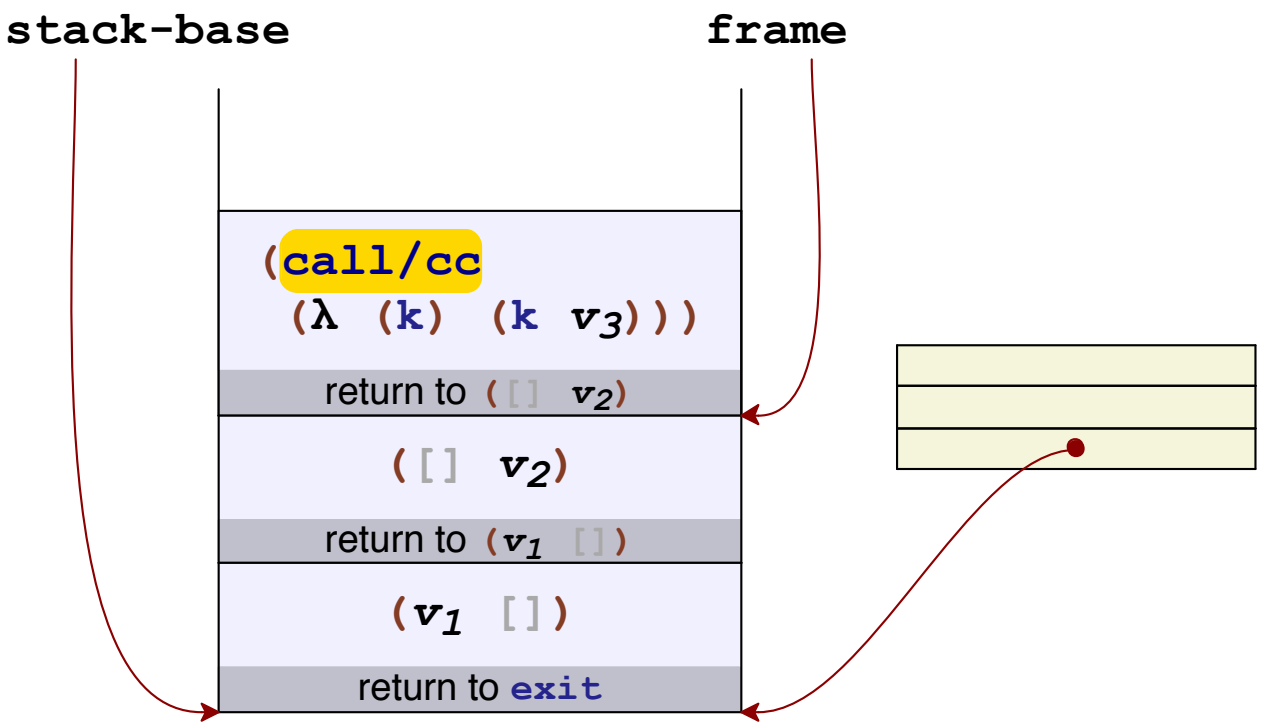
Returns in Stack Frames



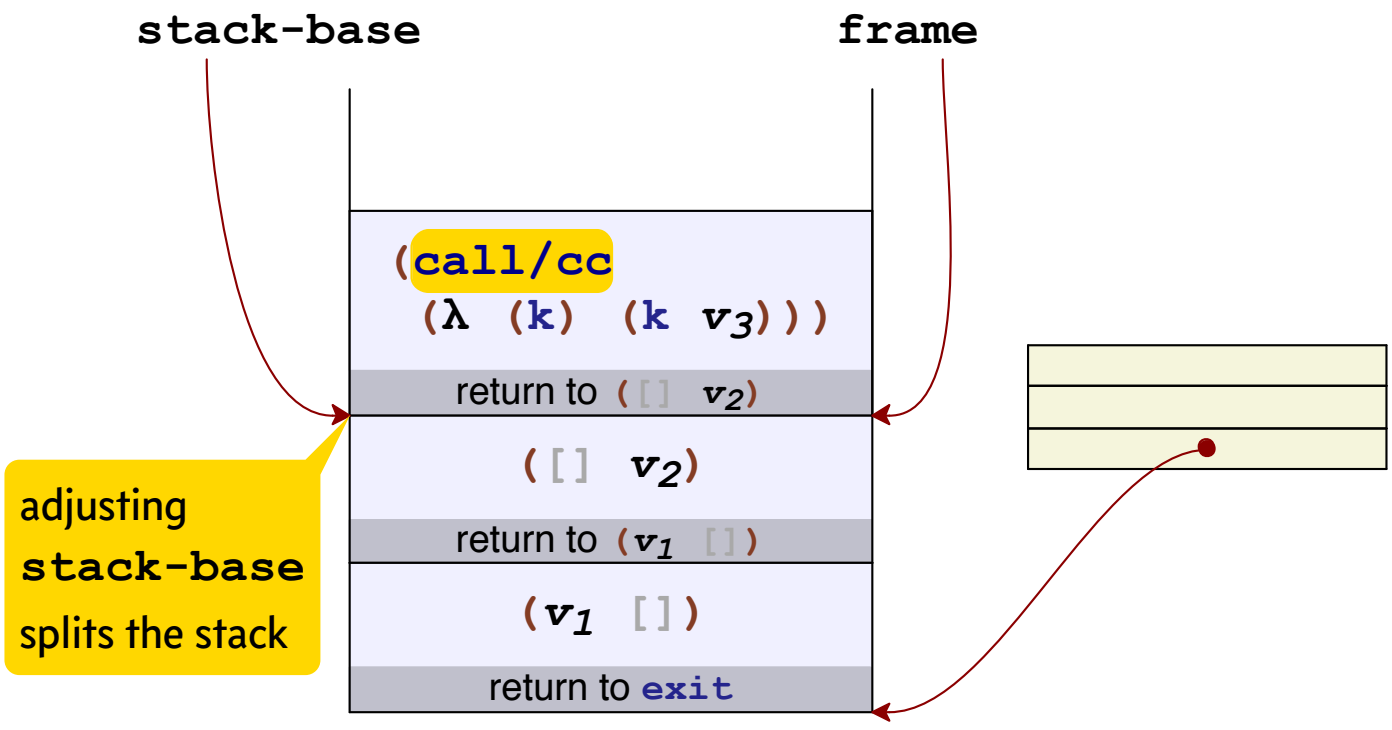
Capturing Stack-Based Continuations



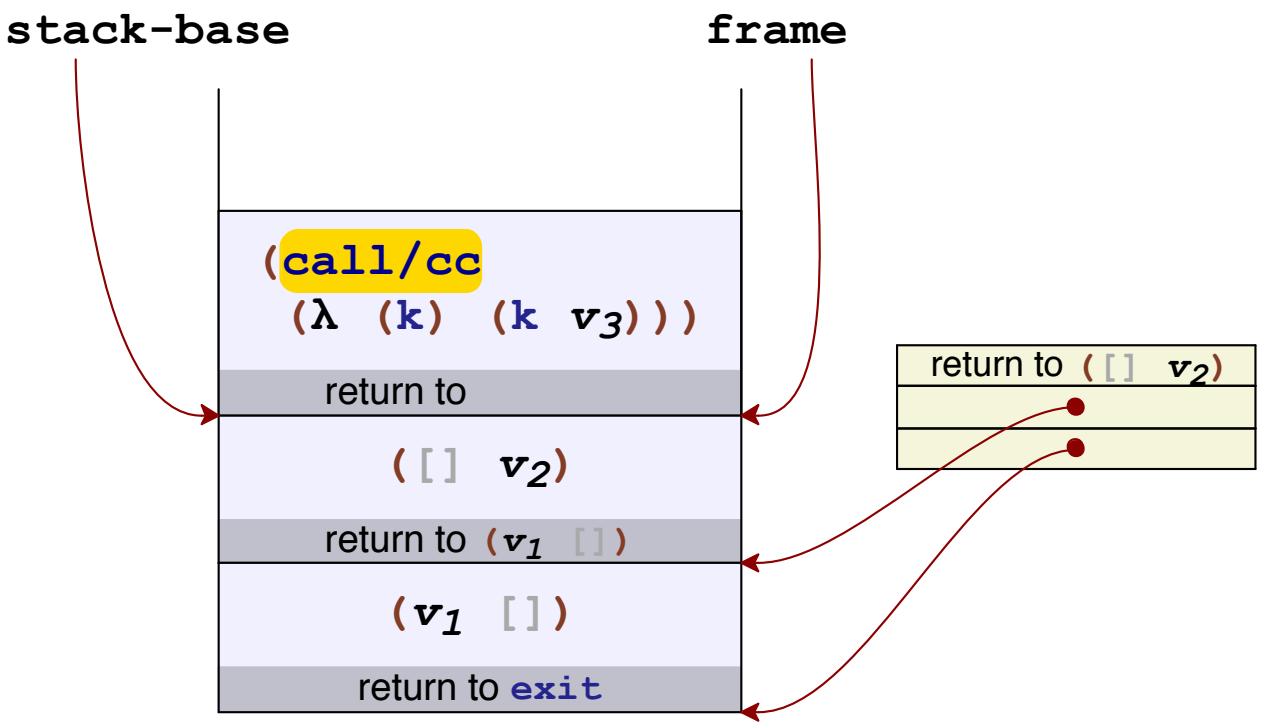
Capturing Stack-Based Continuations



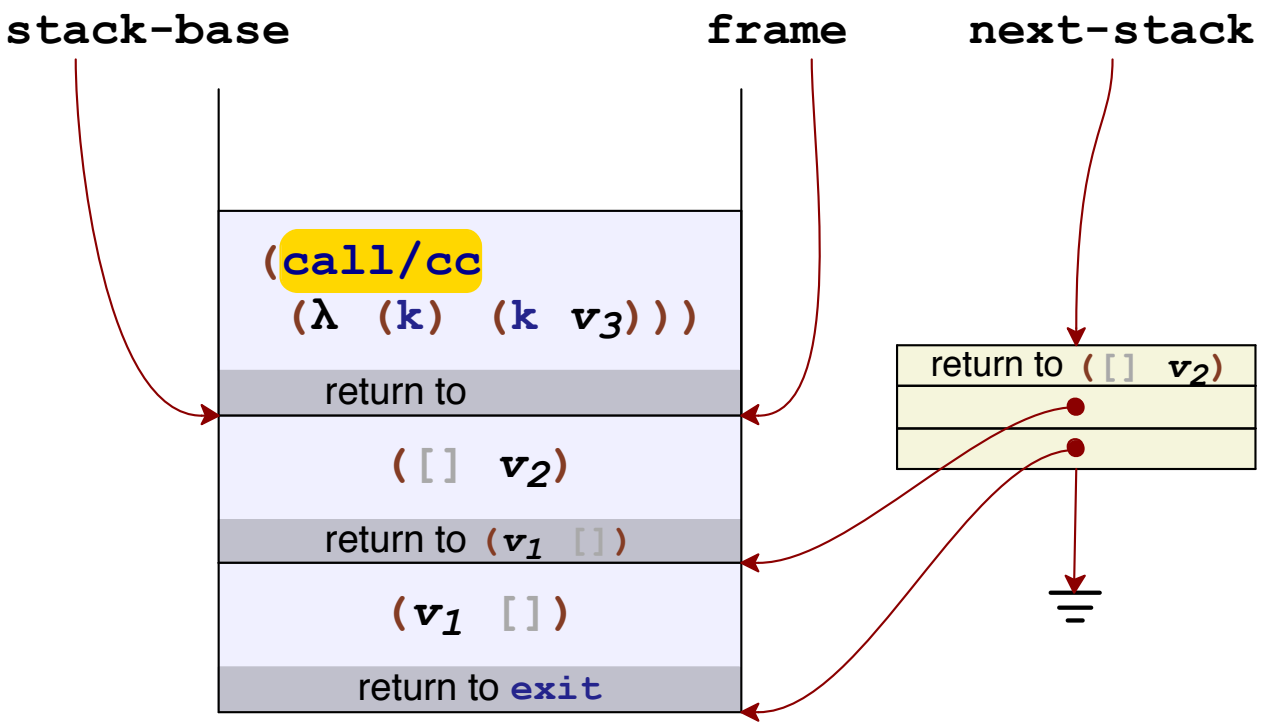
Capturing Stack-Based Continuations



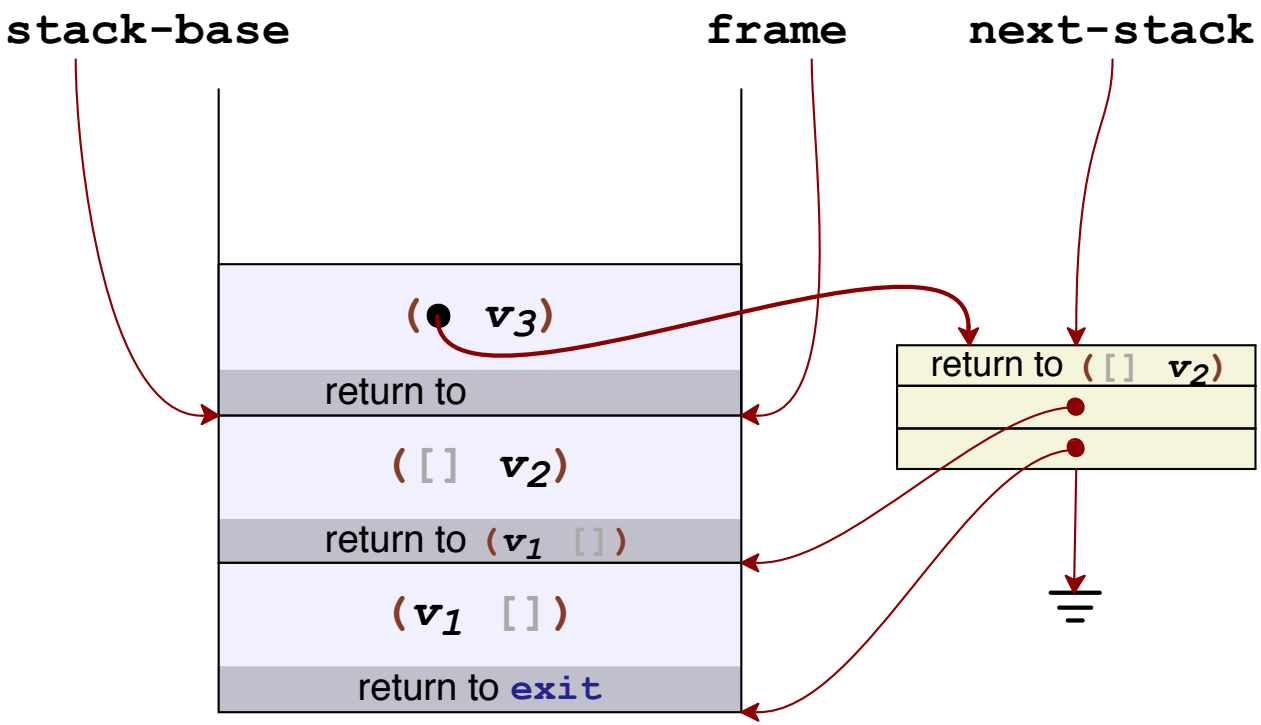
Capturing Stack-Based Continuations



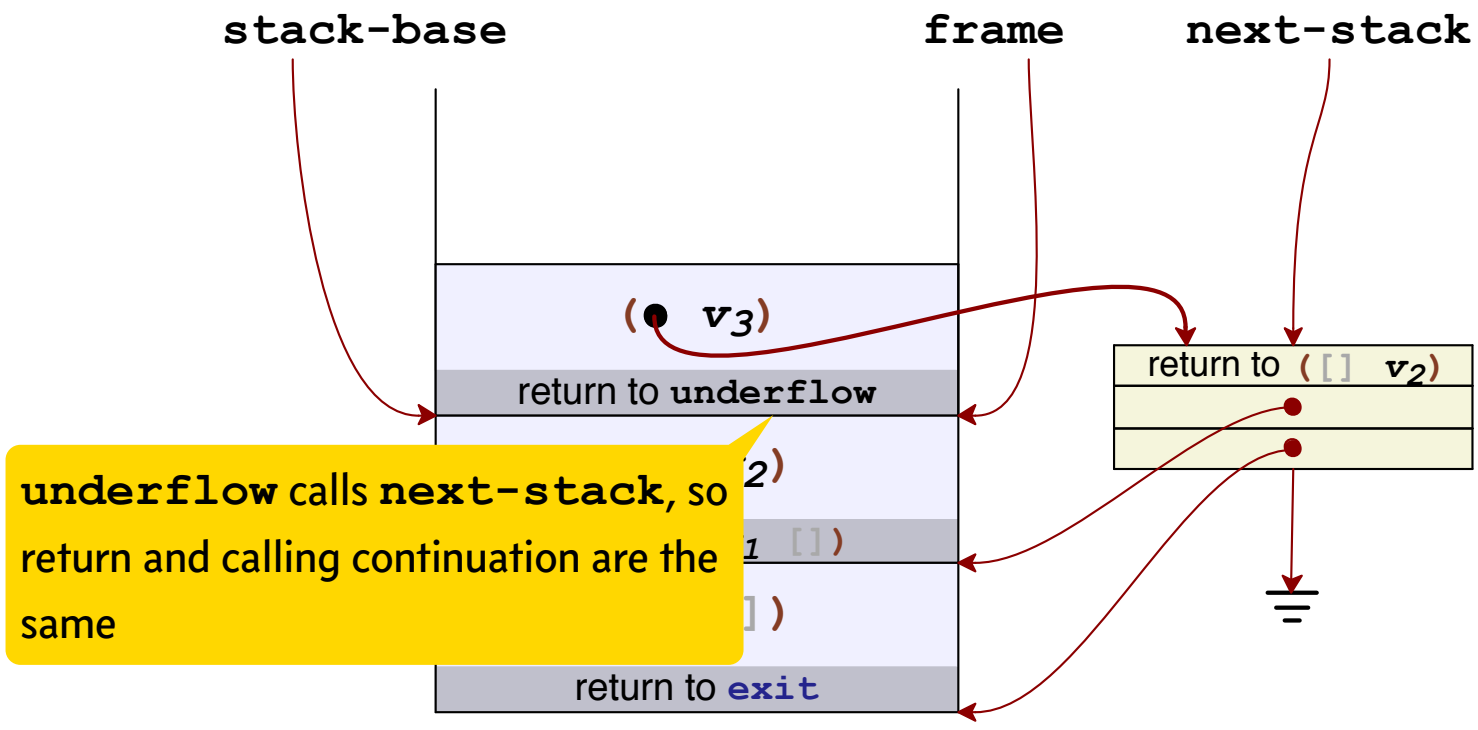
Capturing Stack-Based Continuations



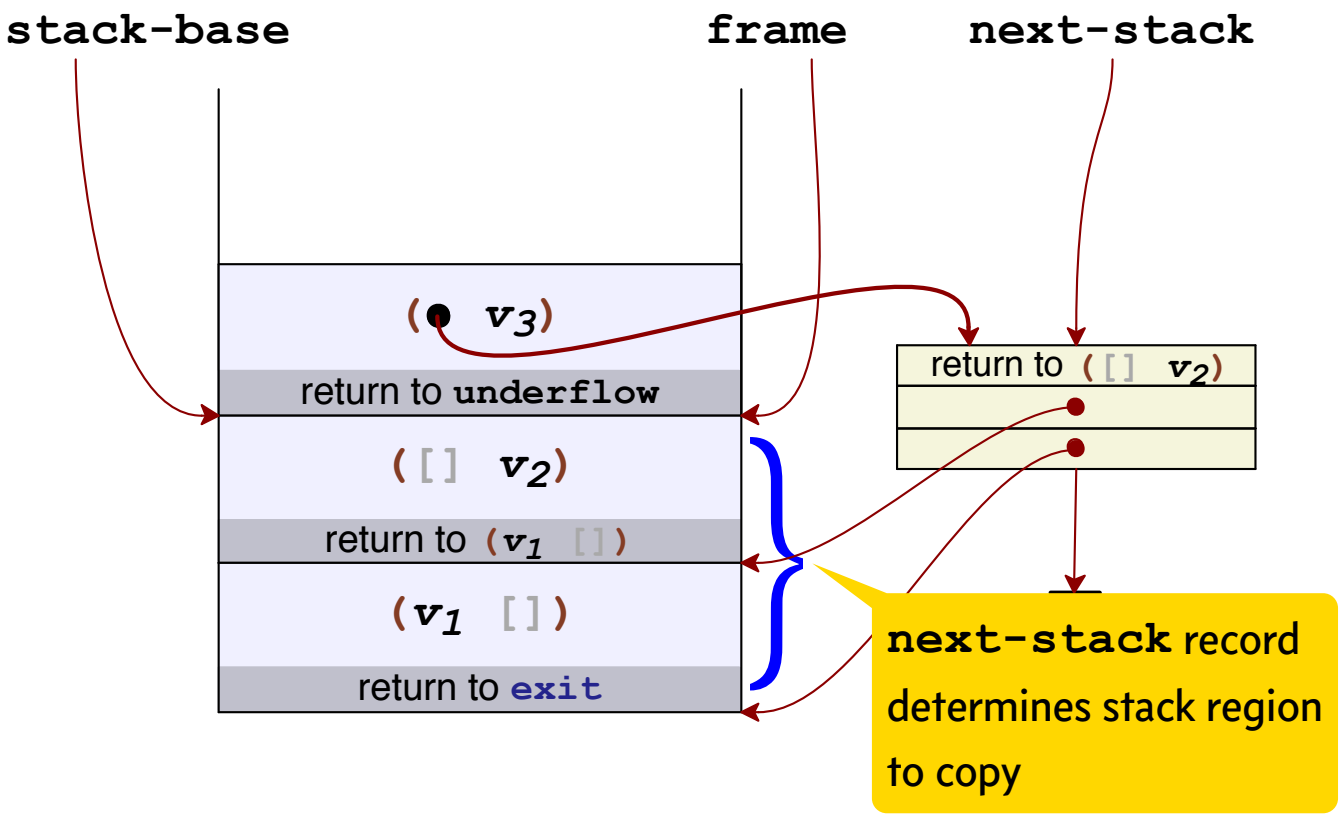
Capturing Stack-Based Continuations



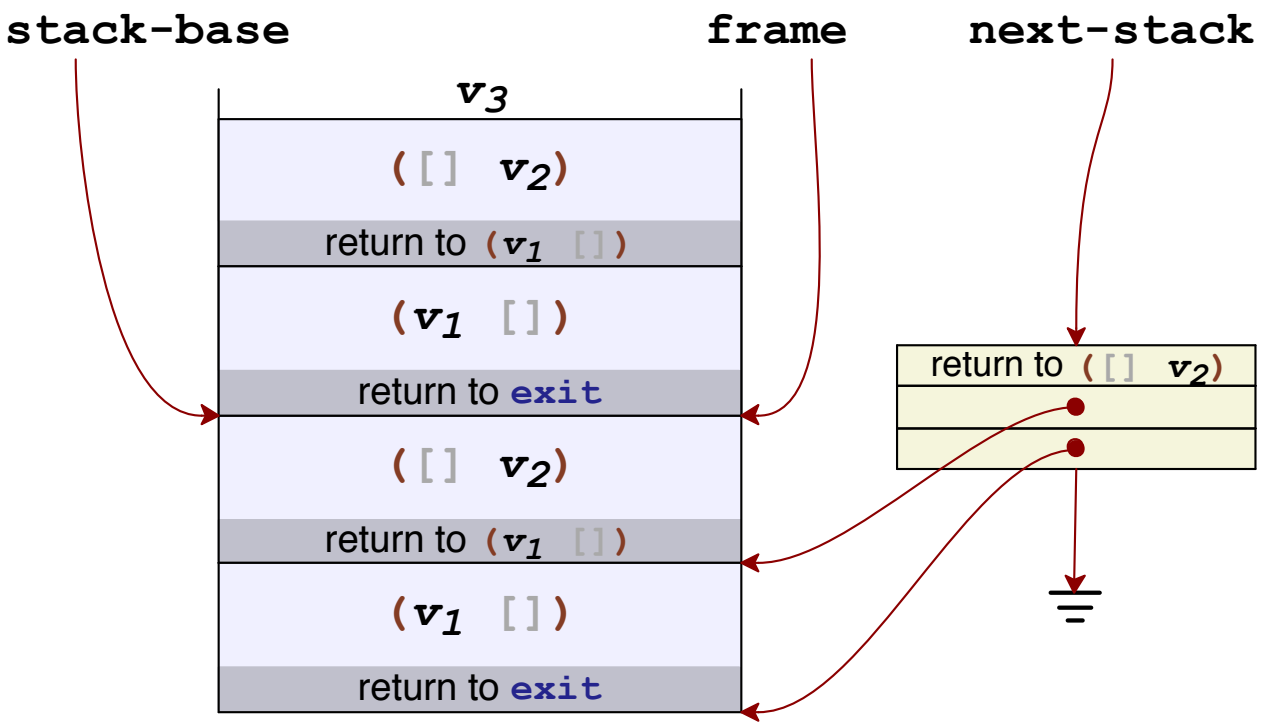
Capturing Stack-Based Continuations



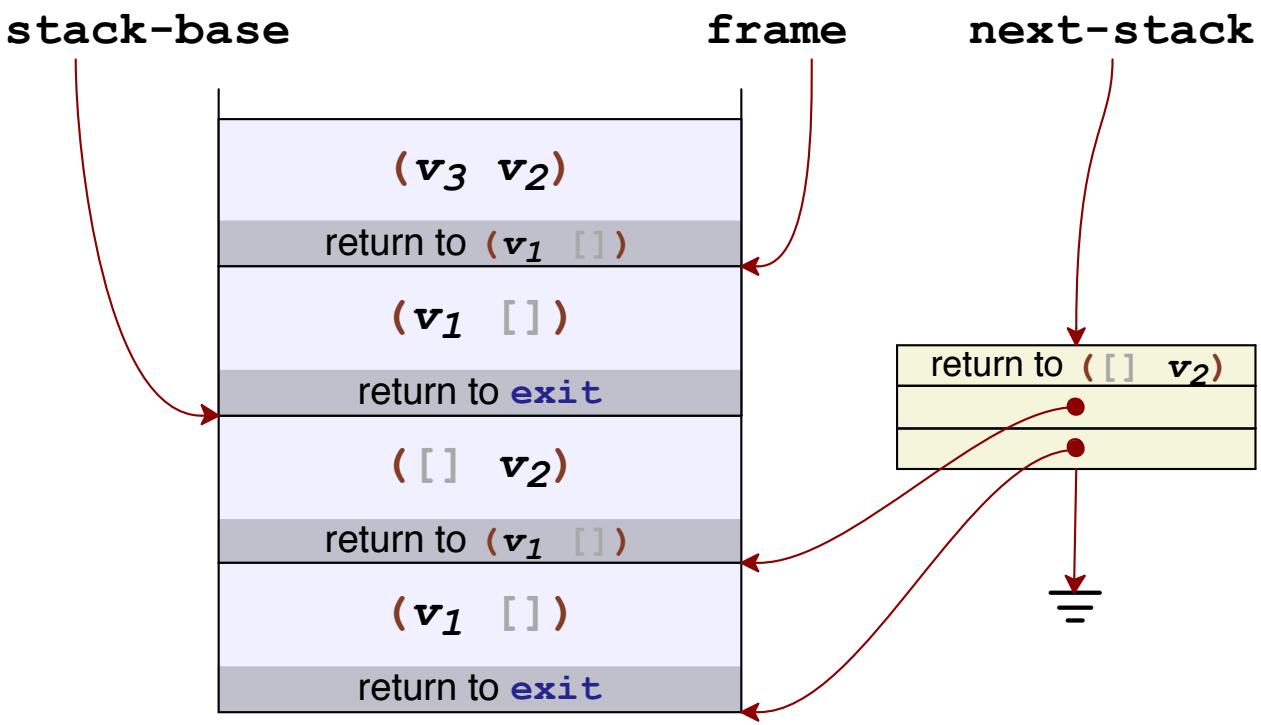
Applying Stack-Based Continuations



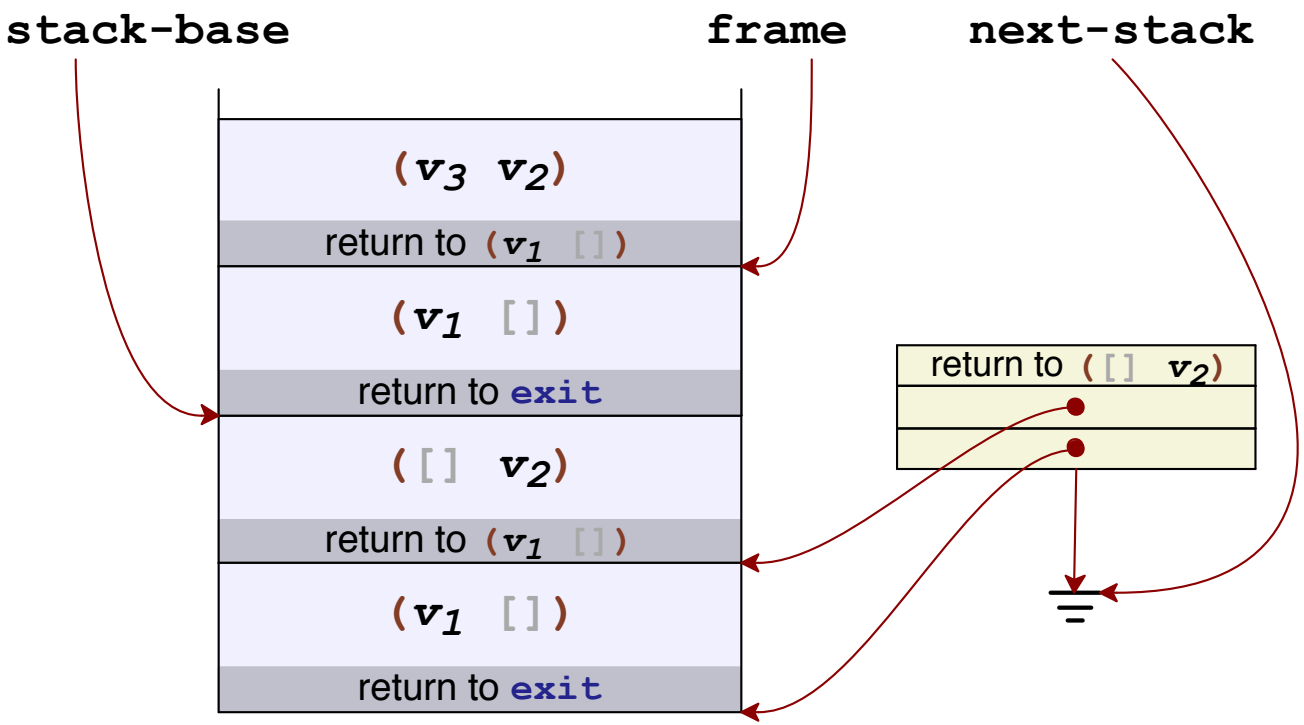
Applying Stack-Based Continuations



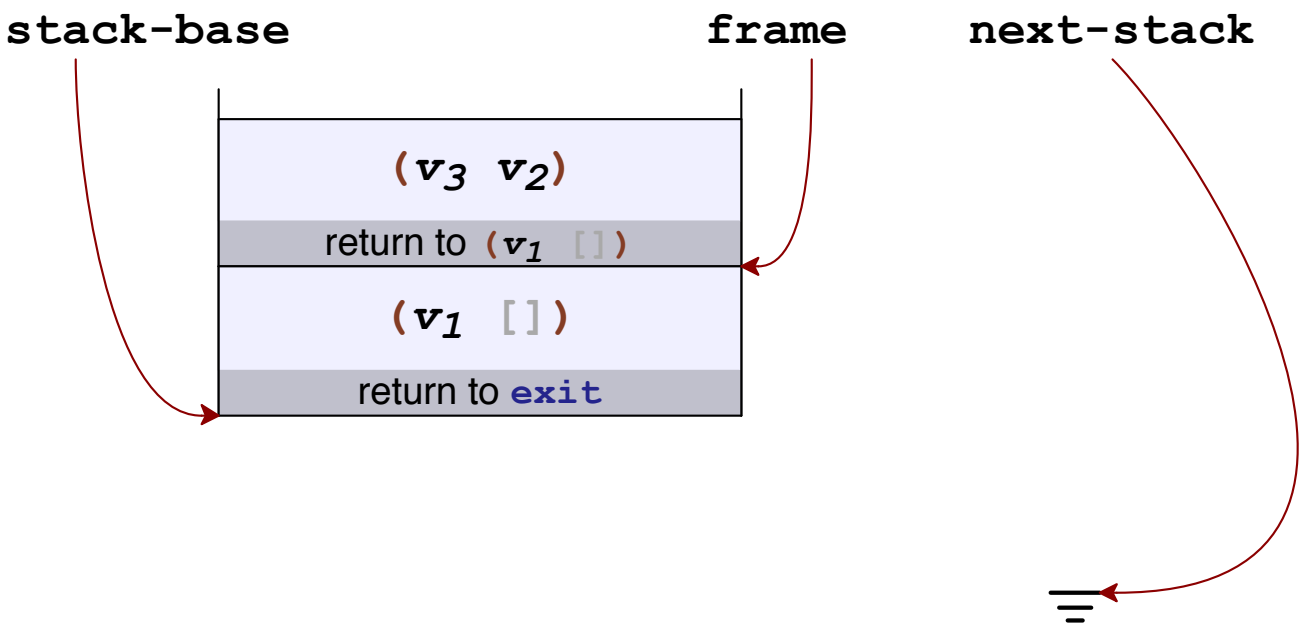
Applying Stack-Based Continuations



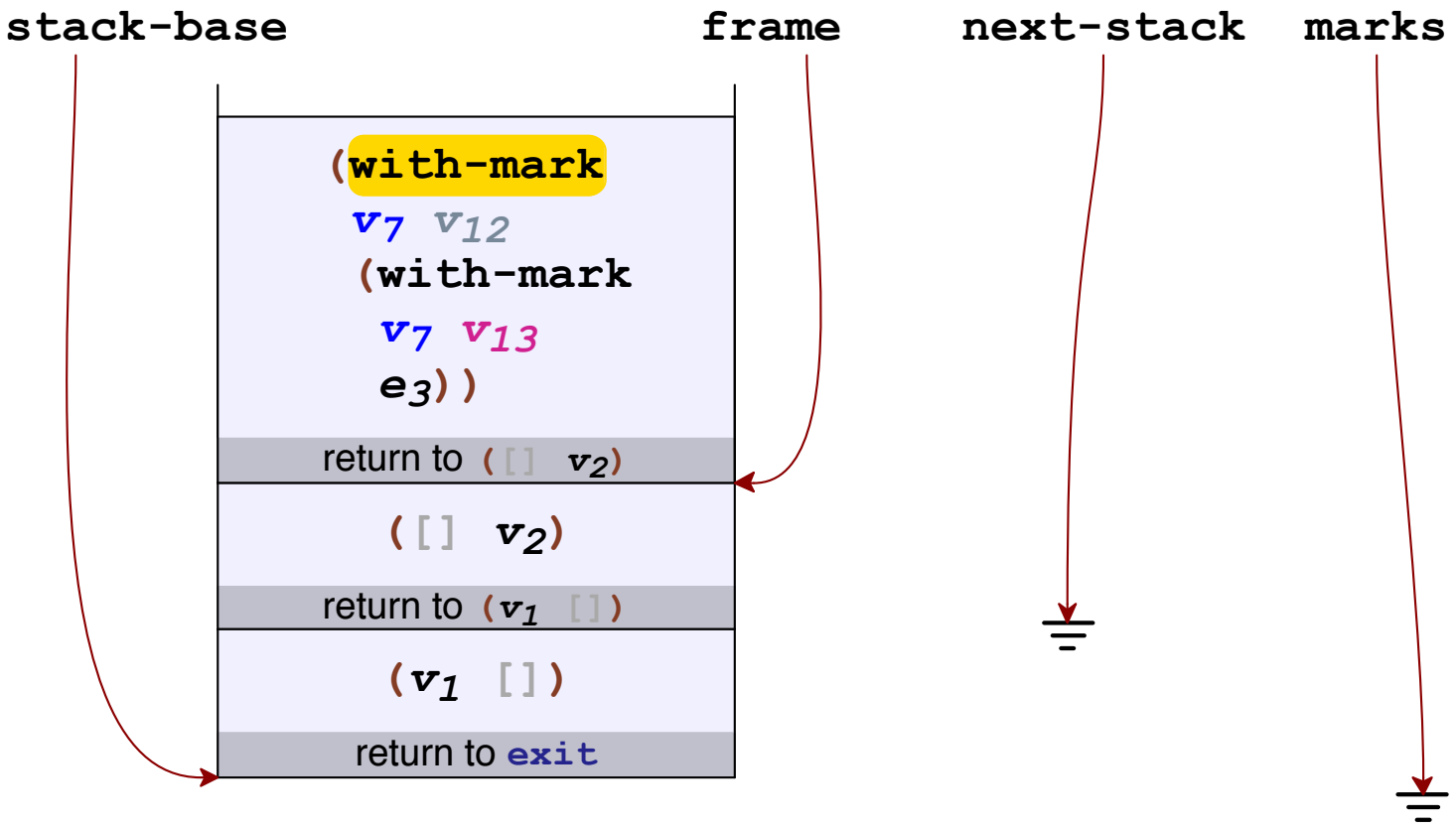
Applying Stack-Based Continuations



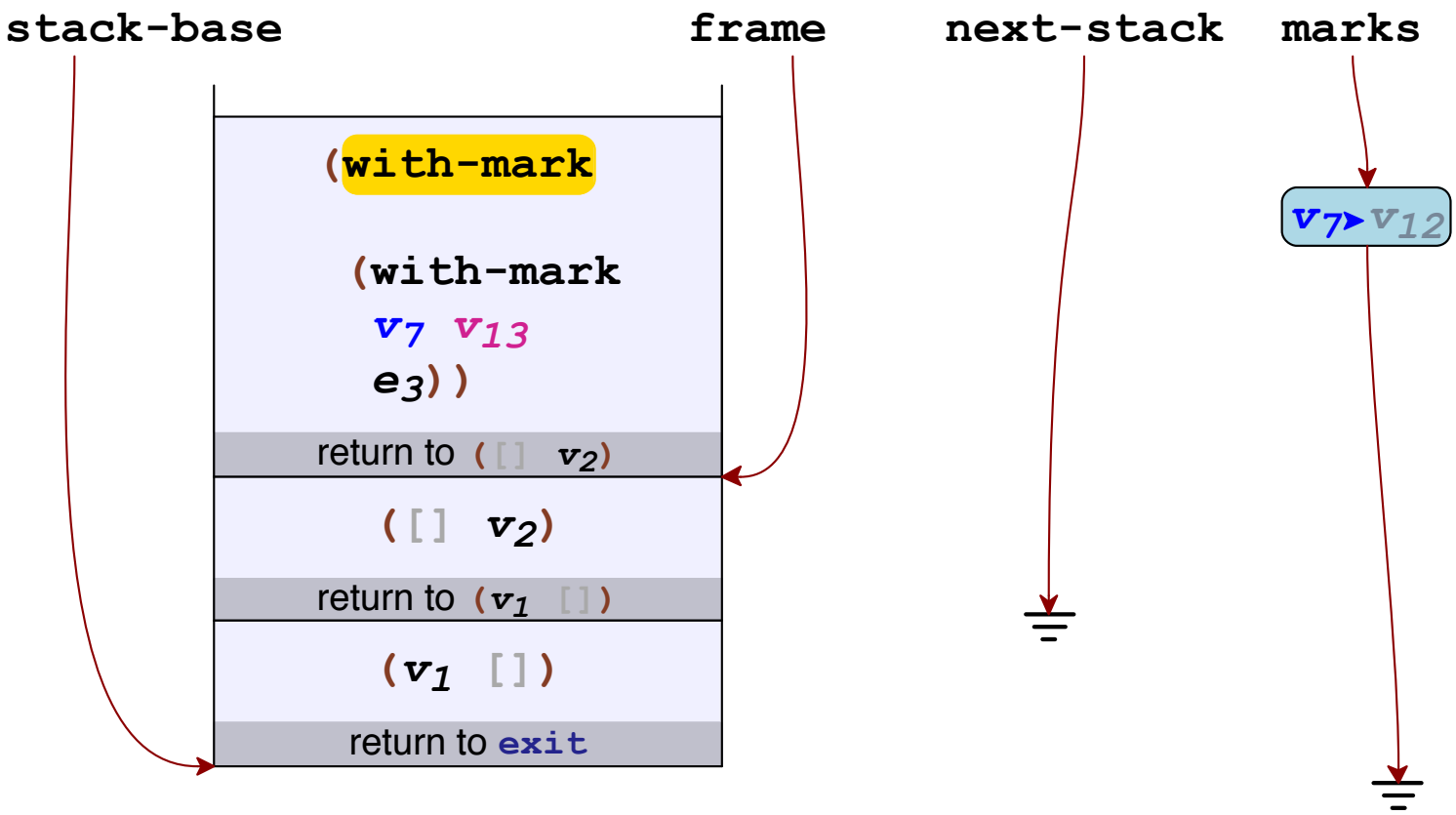
Applying Stack-Based Continuations



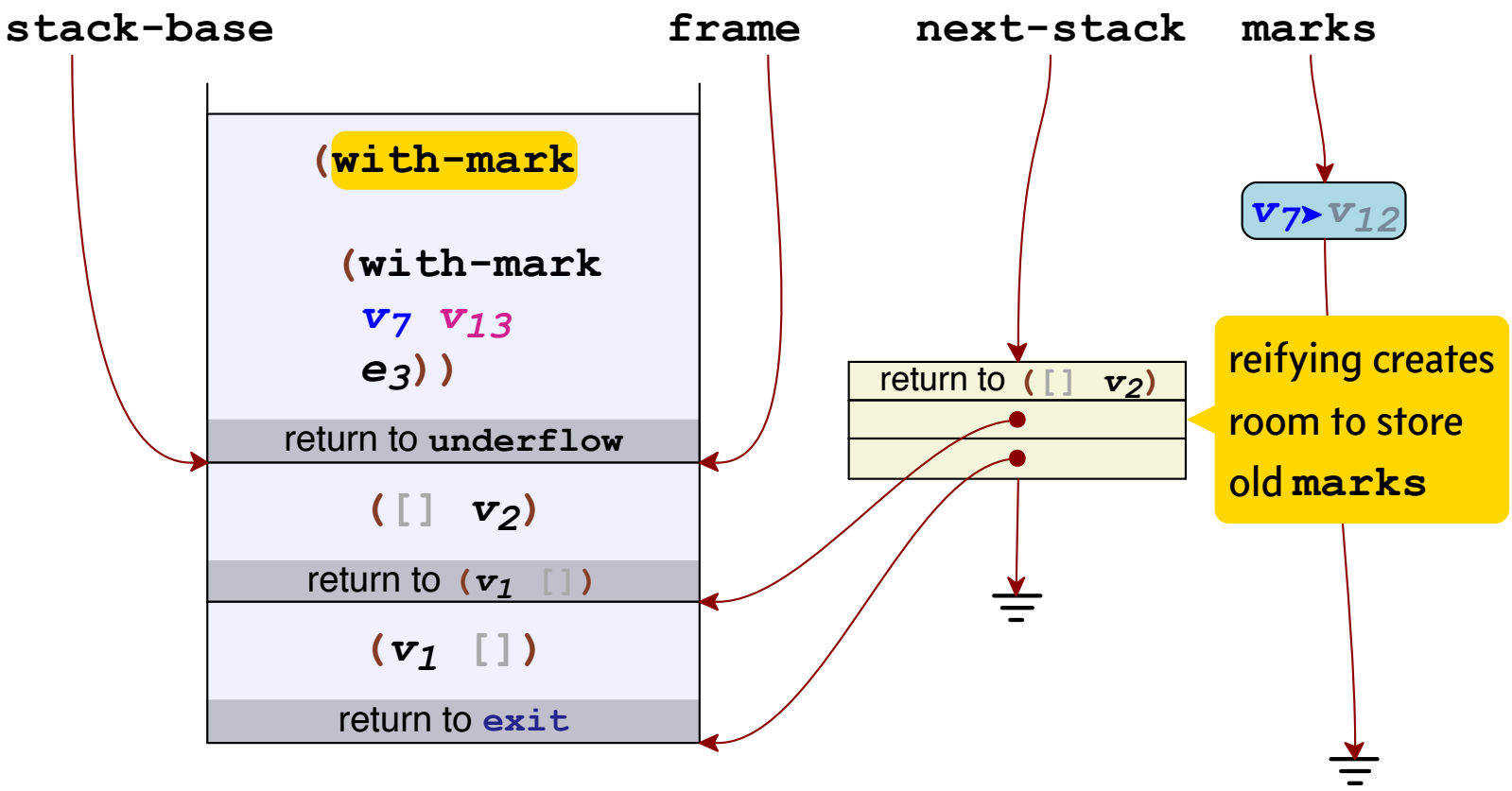
new! Setting Continuation Marks



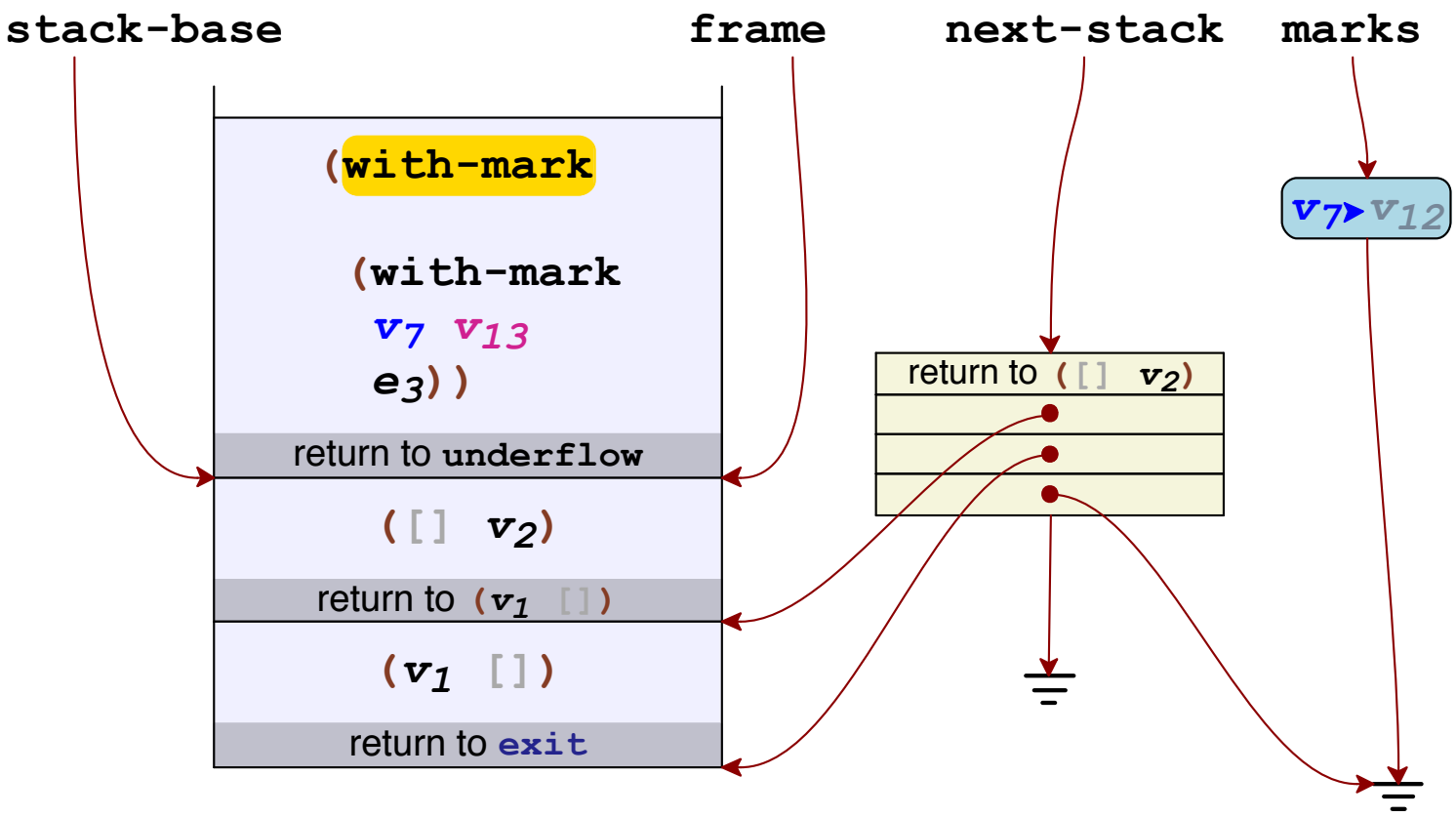
Setting Continuation Marks



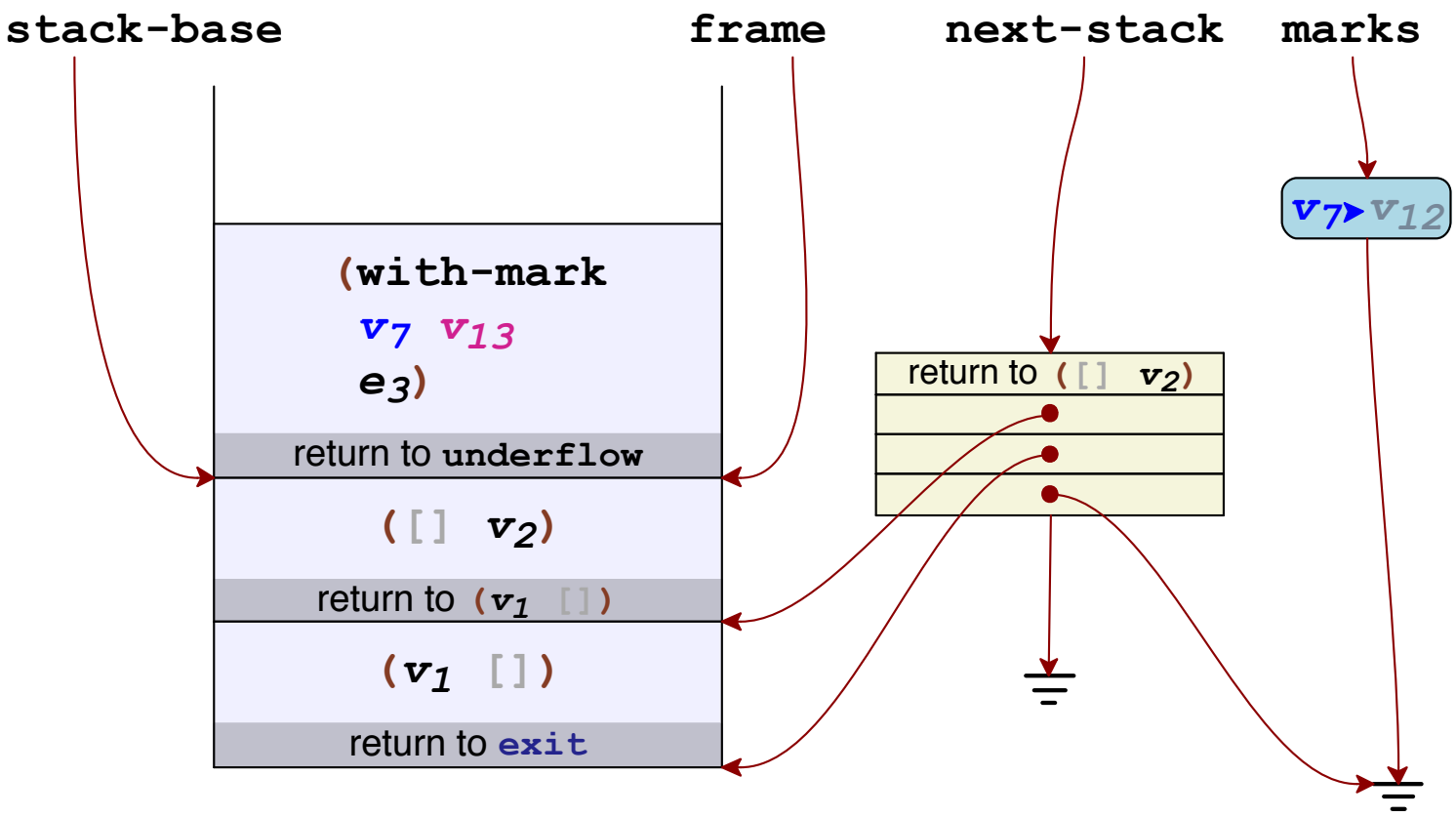
Setting Continuation Marks



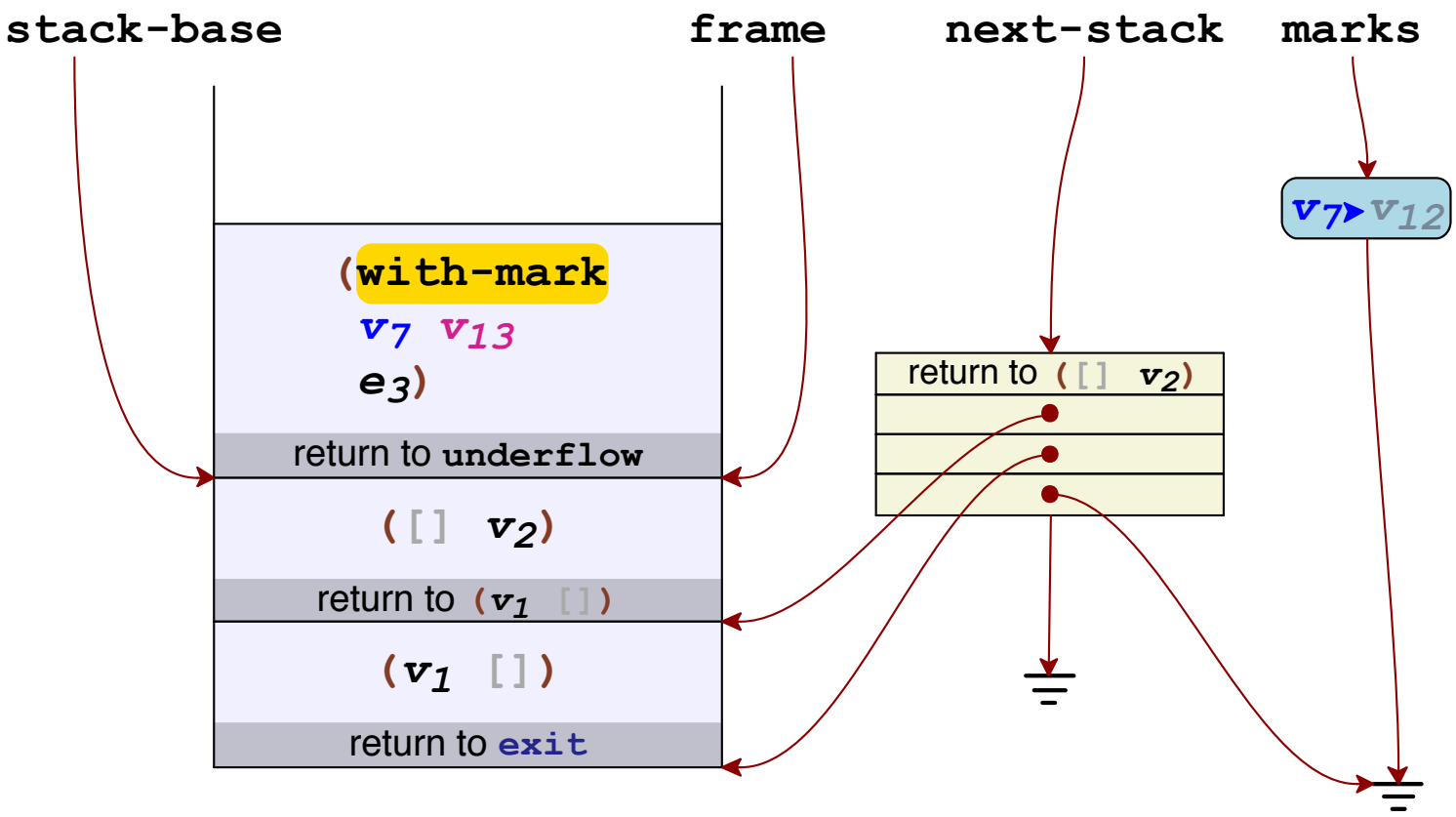
Setting Continuation Marks



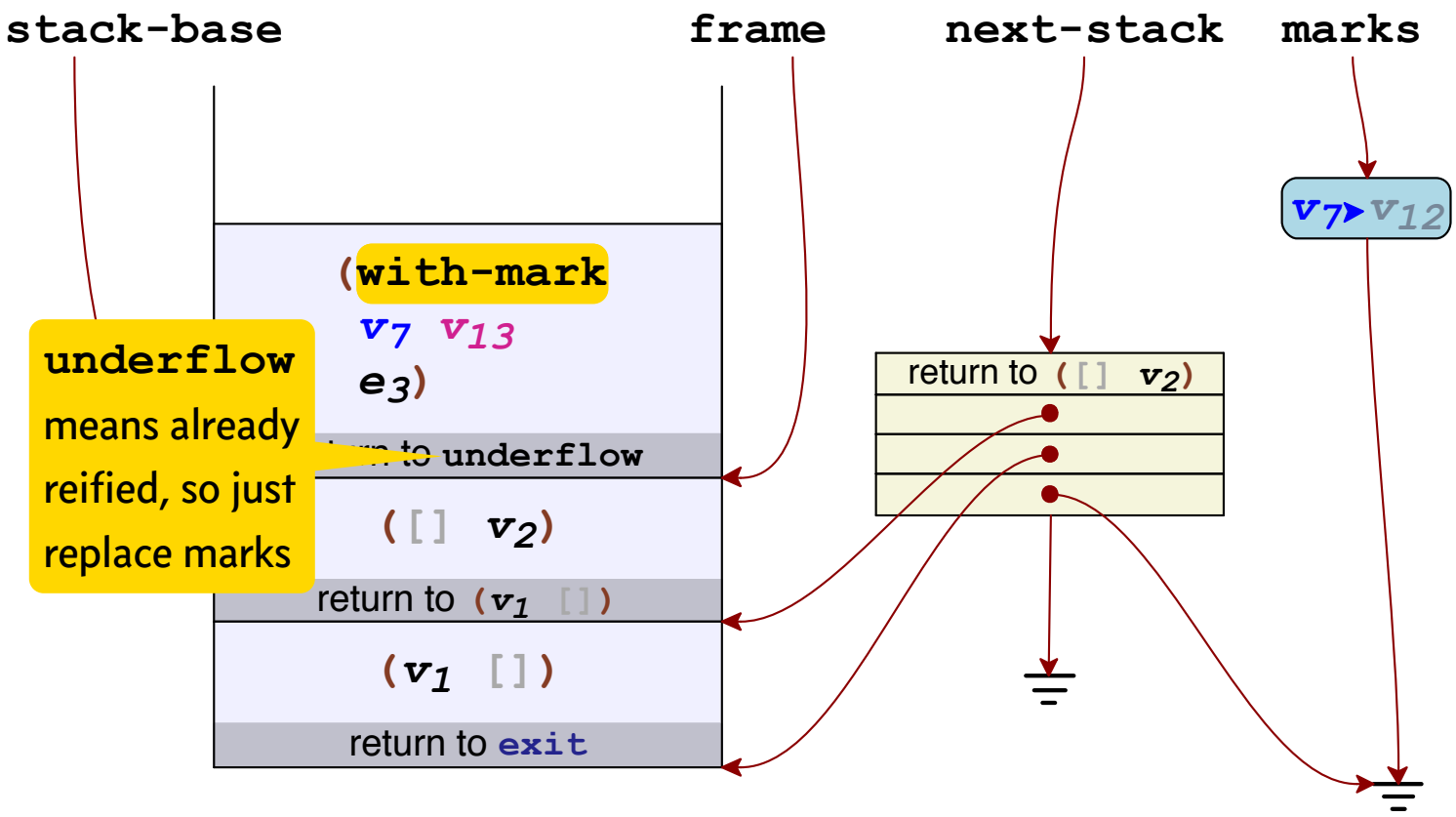
Setting Continuation Marks



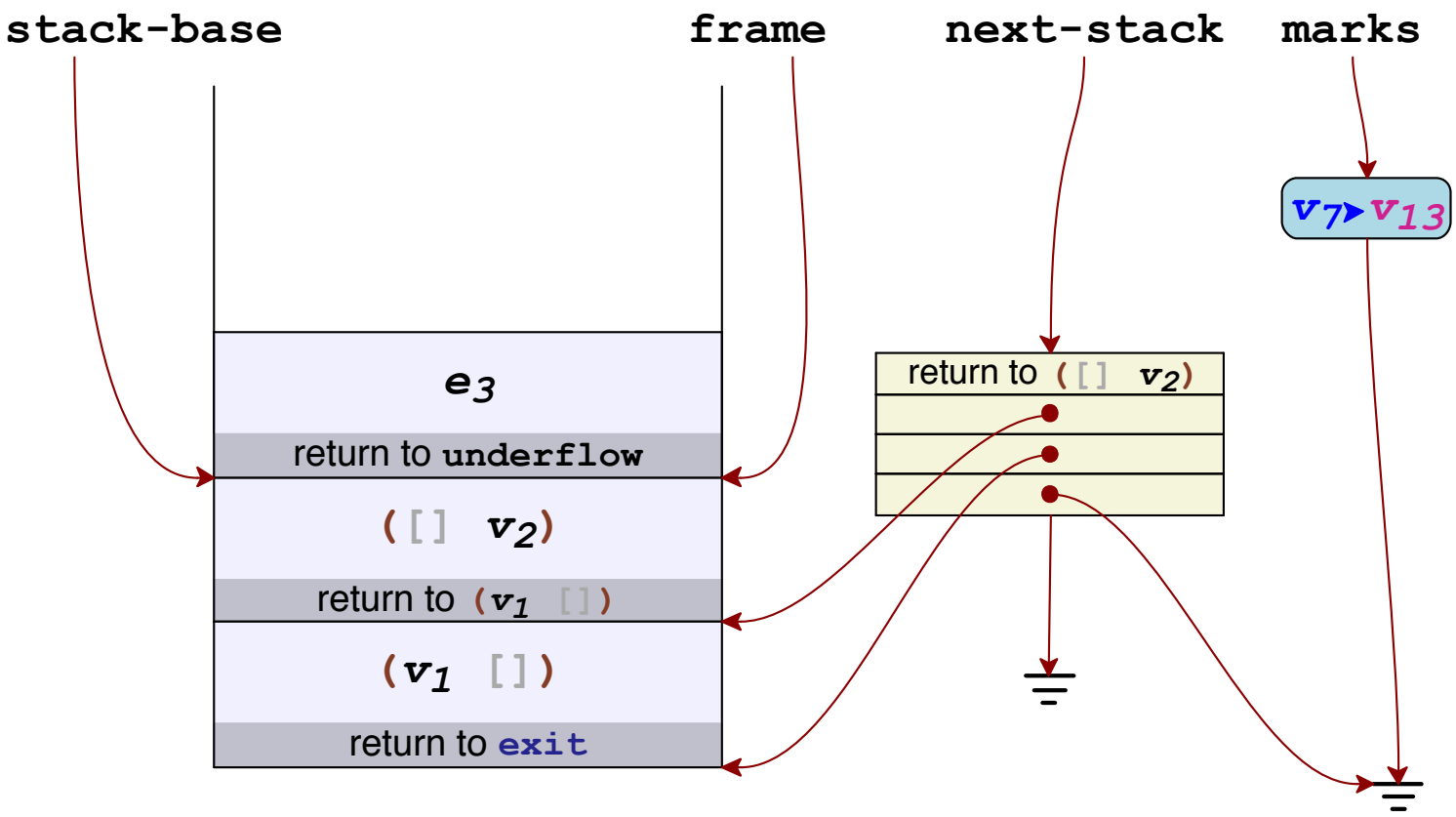
Setting Continuation Marks



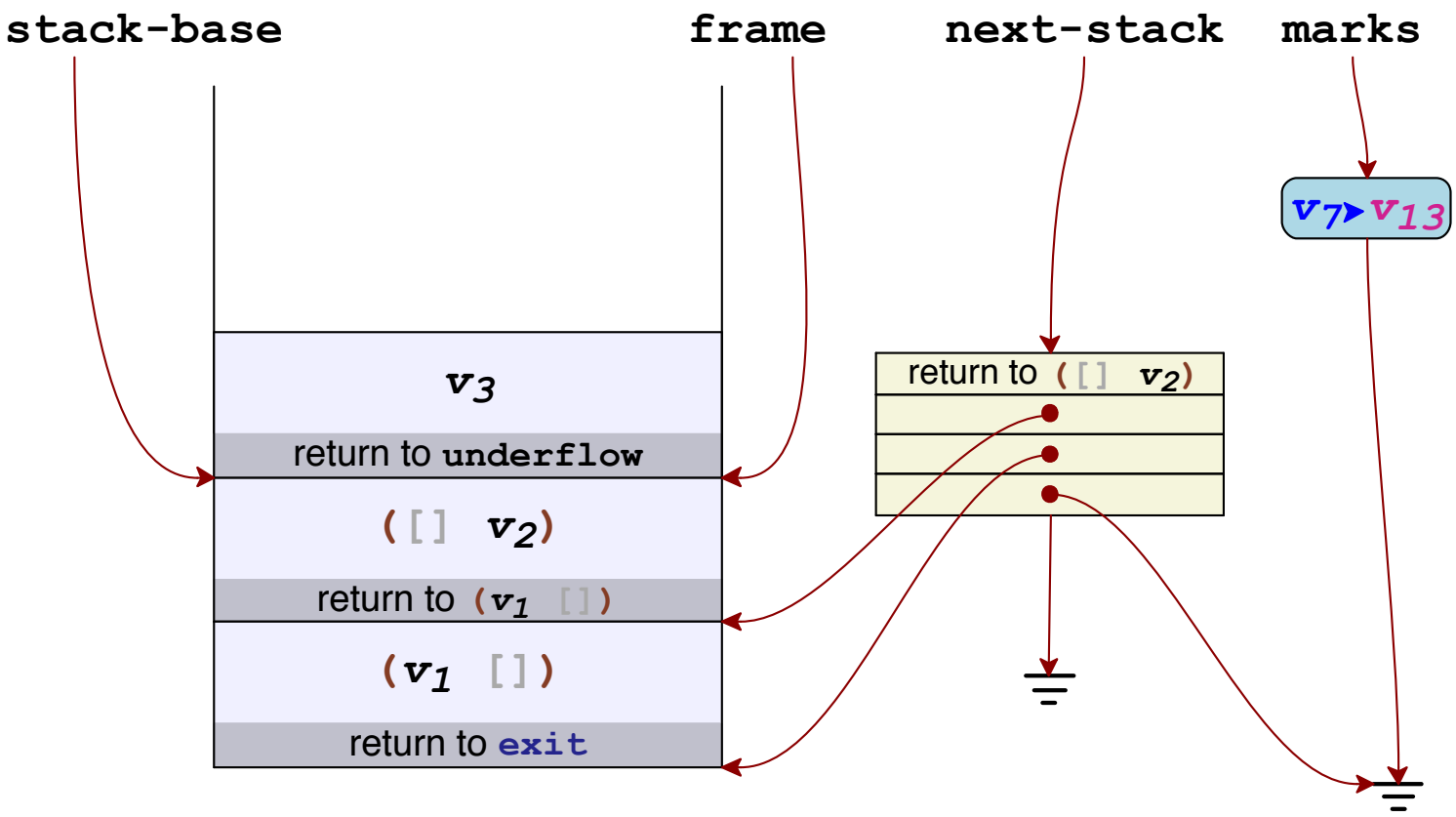
Setting Continuation Marks



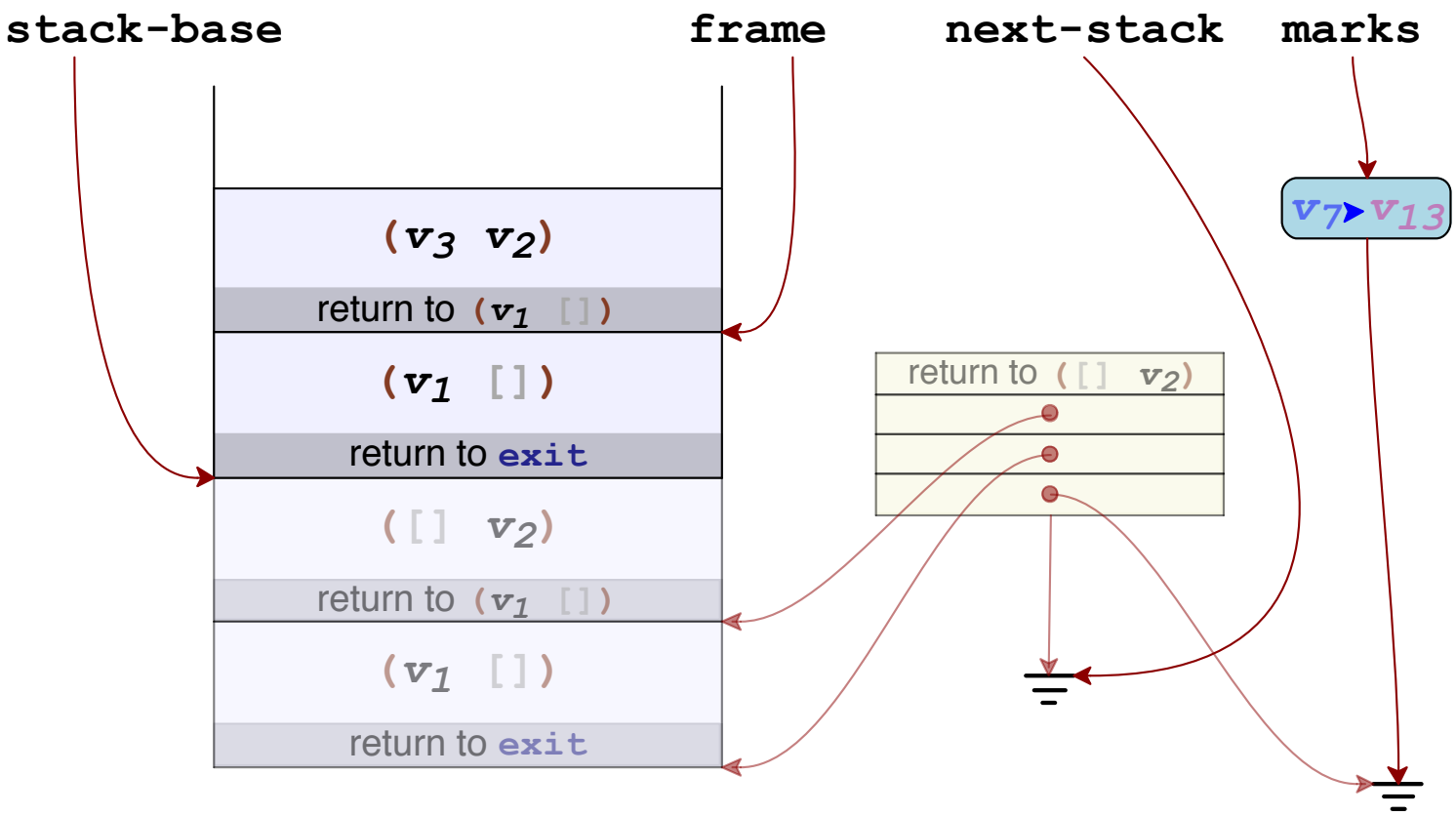
Setting Continuation Marks



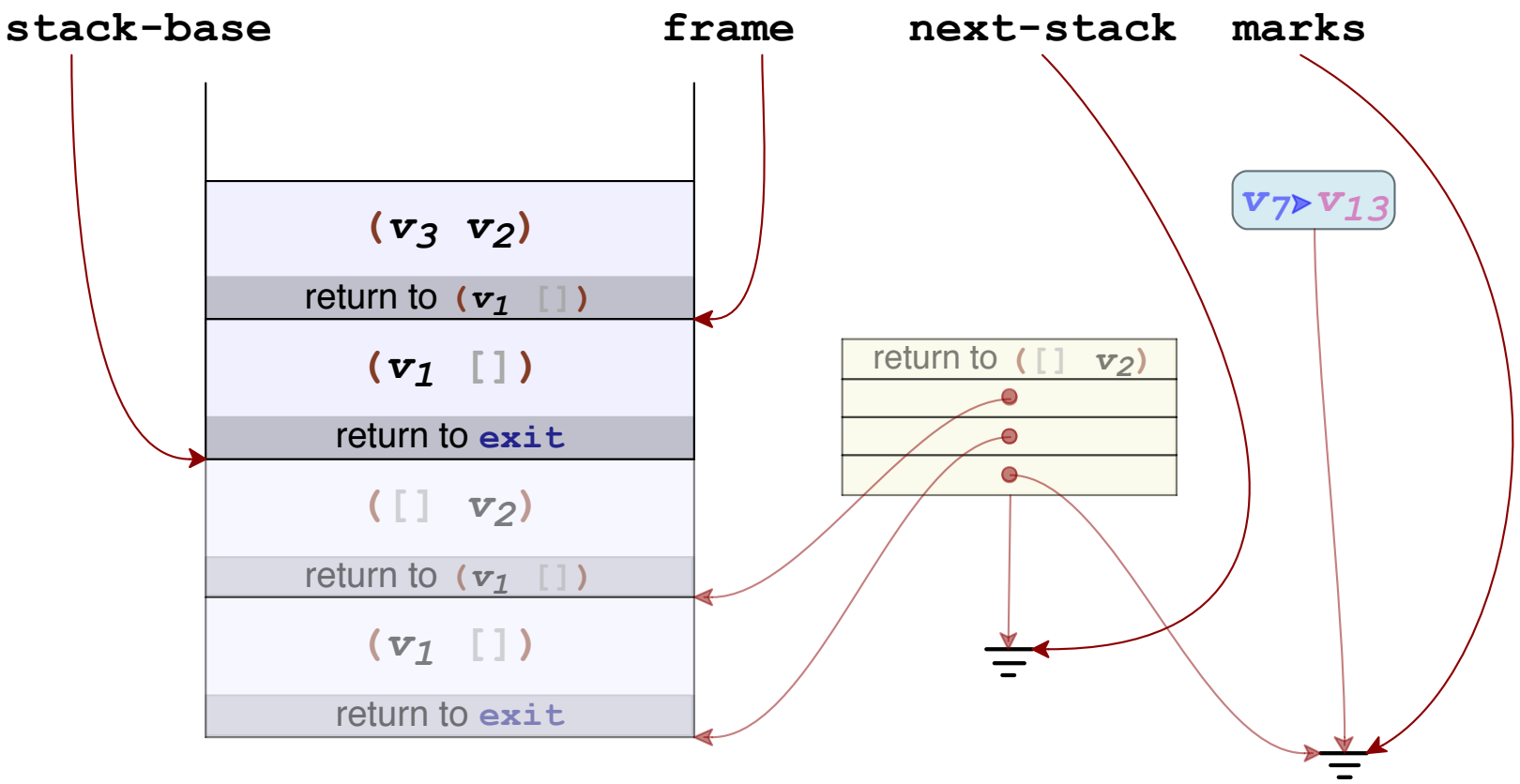
Setting Continuation Marks



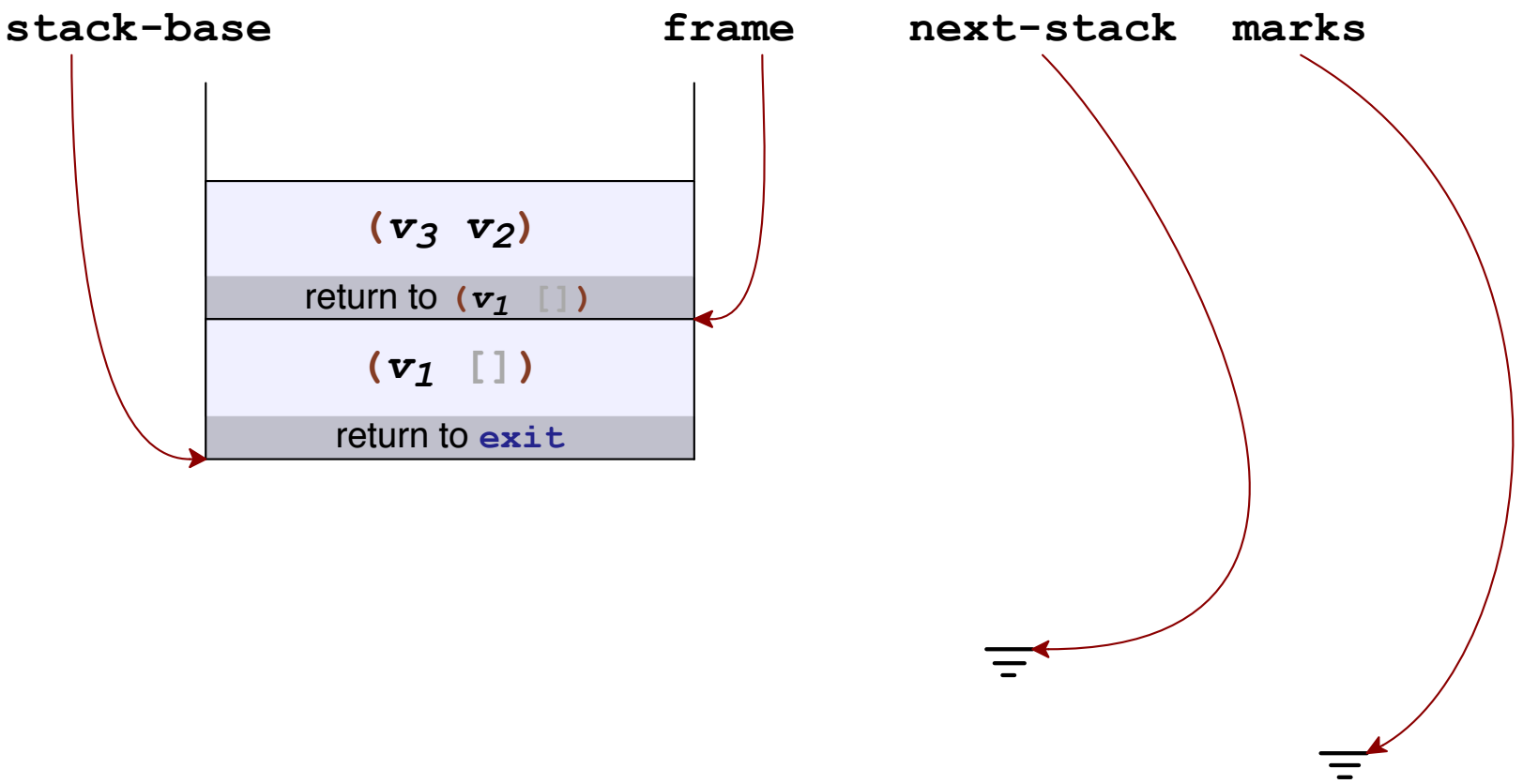
Removing Continuation Marks



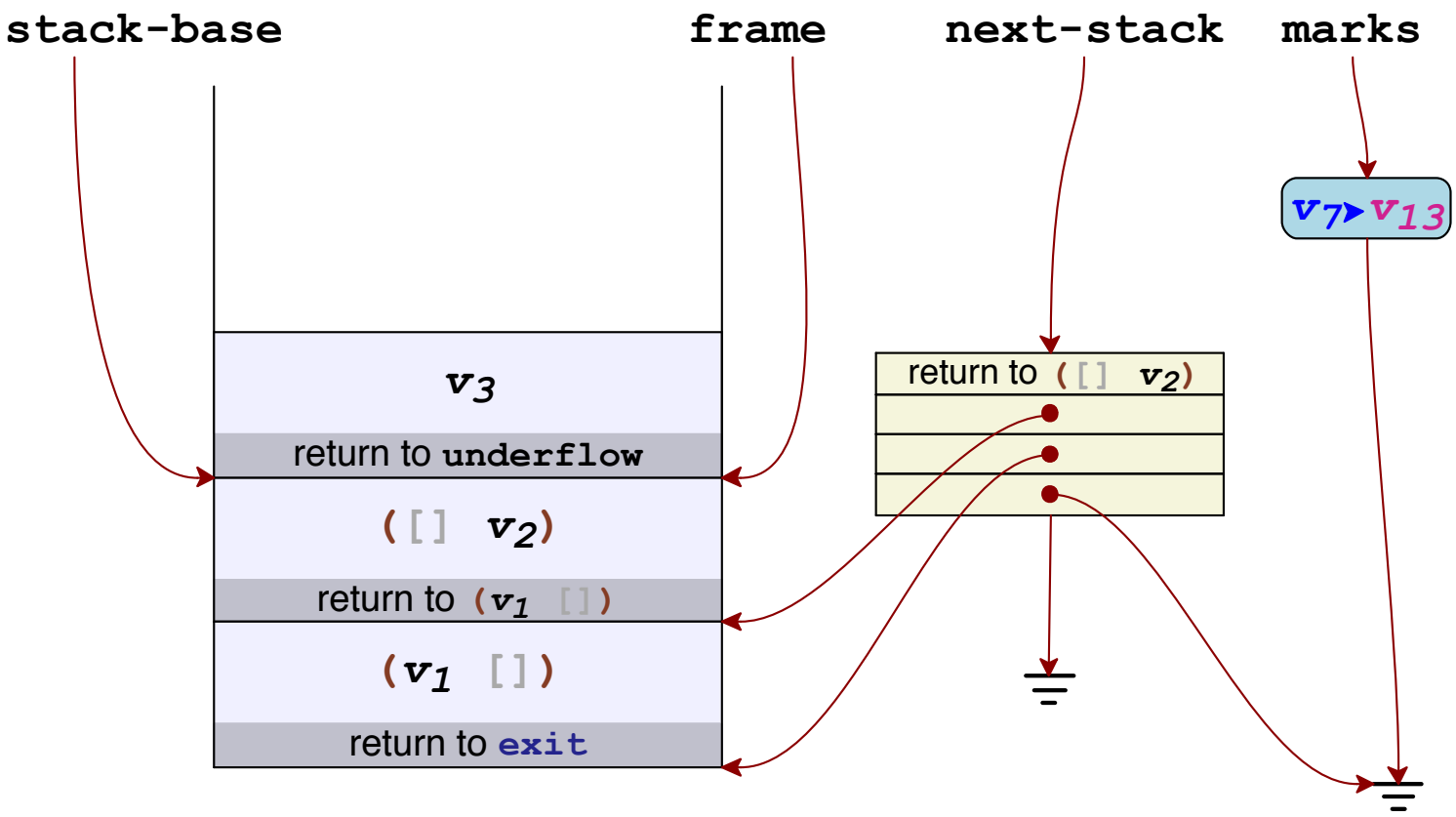
Removing Continuation Marks



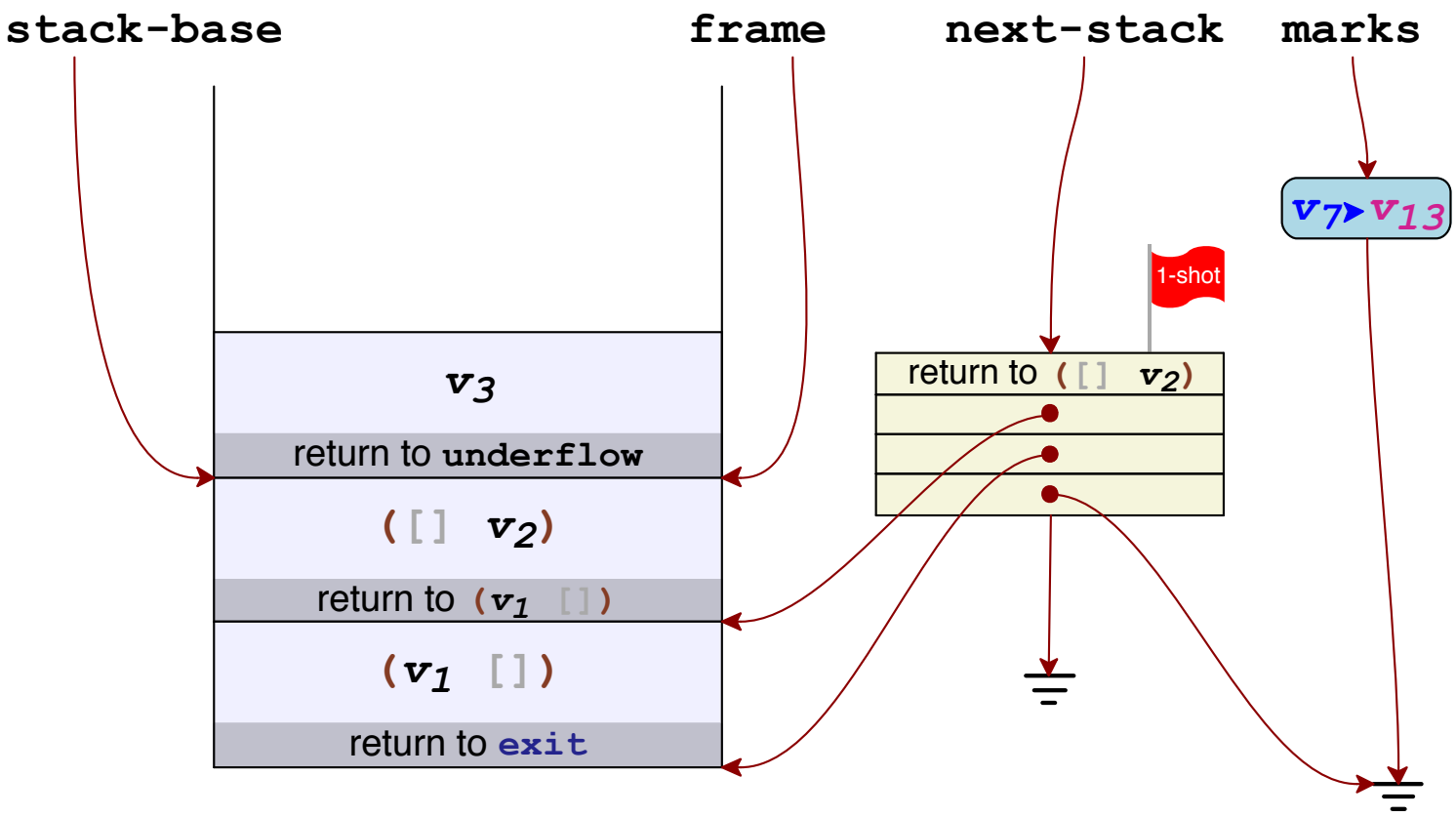
Removing Continuation Marks



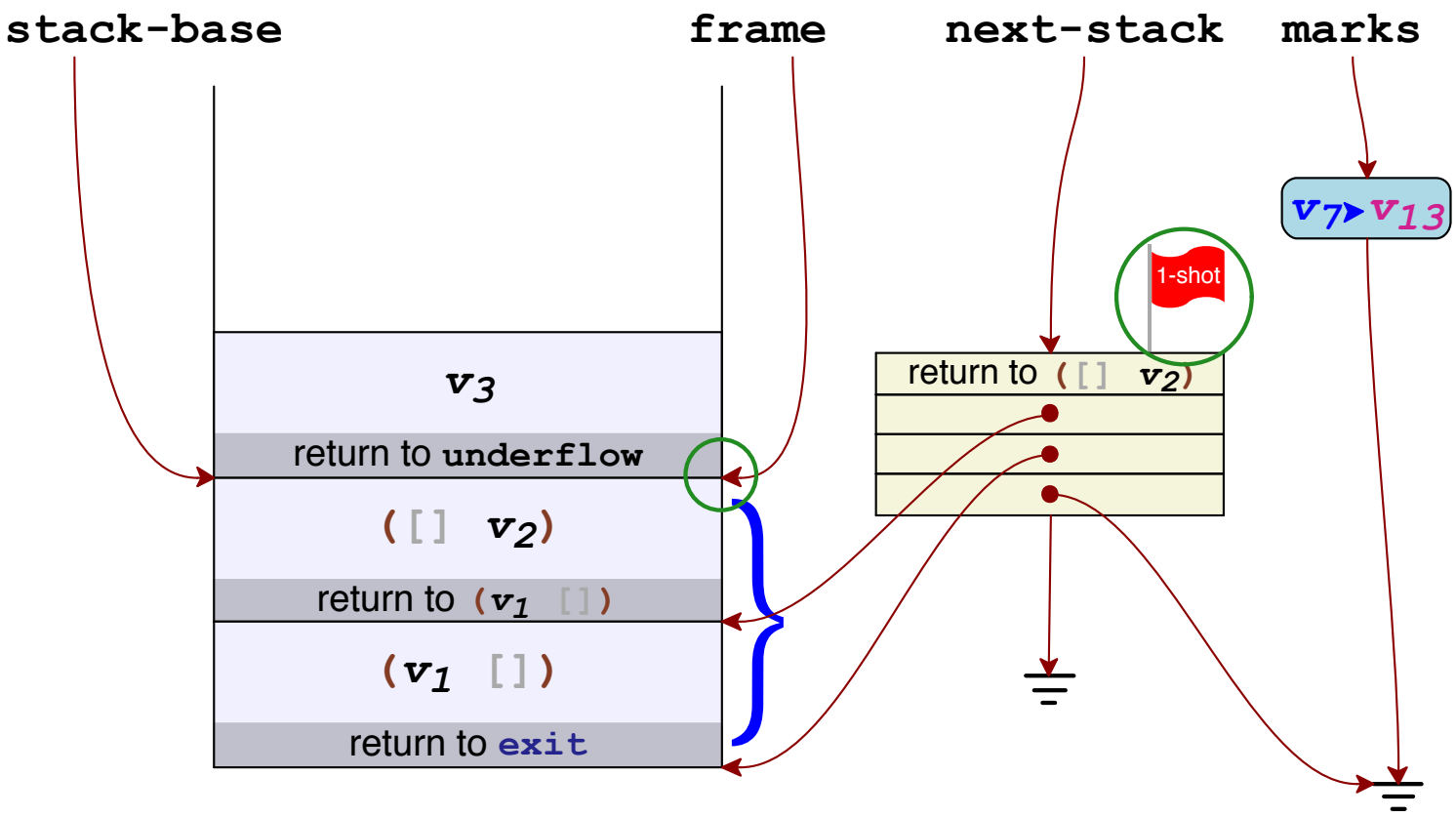
Optimistic One-Shot Continuations



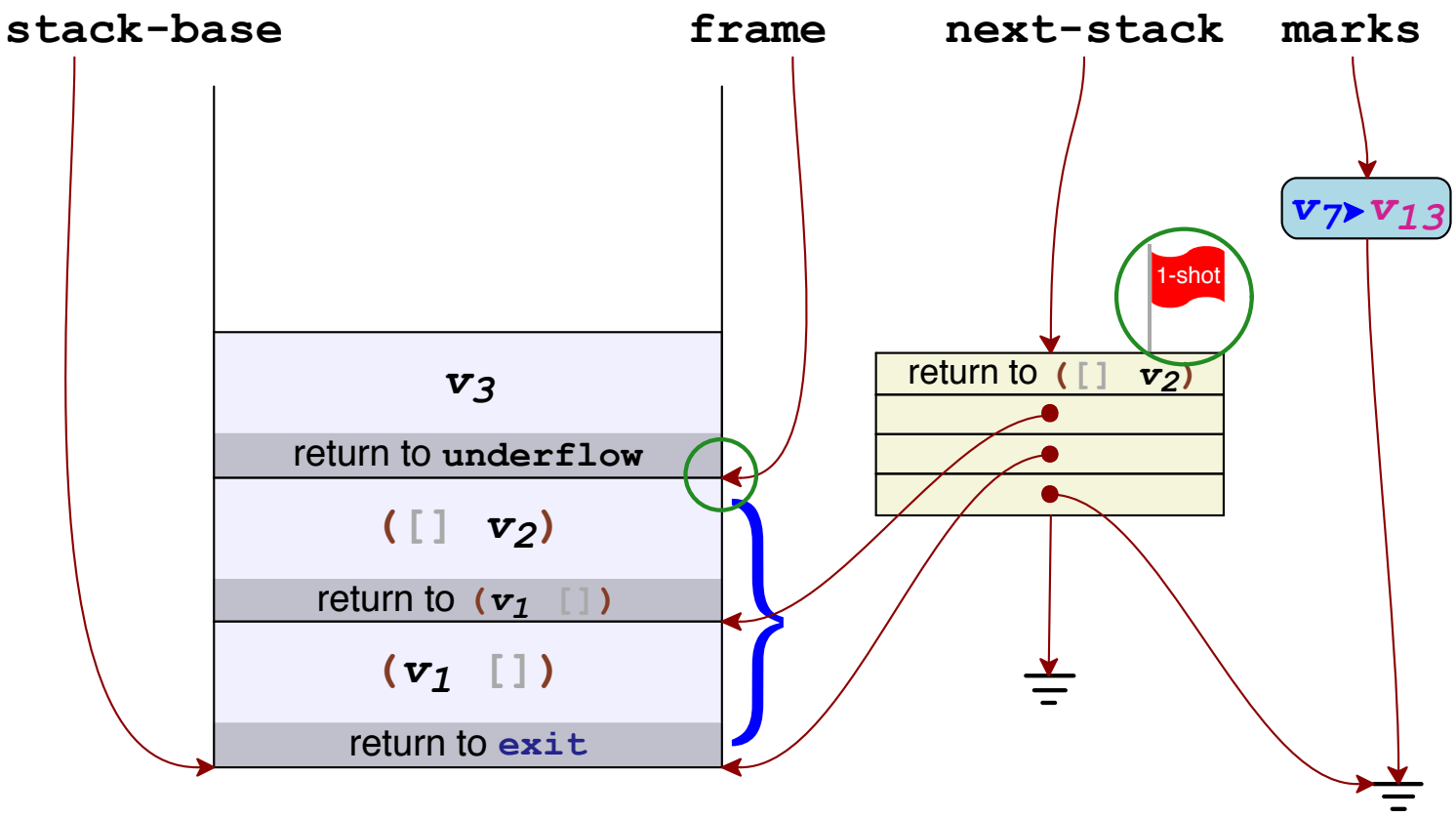
Optimistic One-Shot Continuations



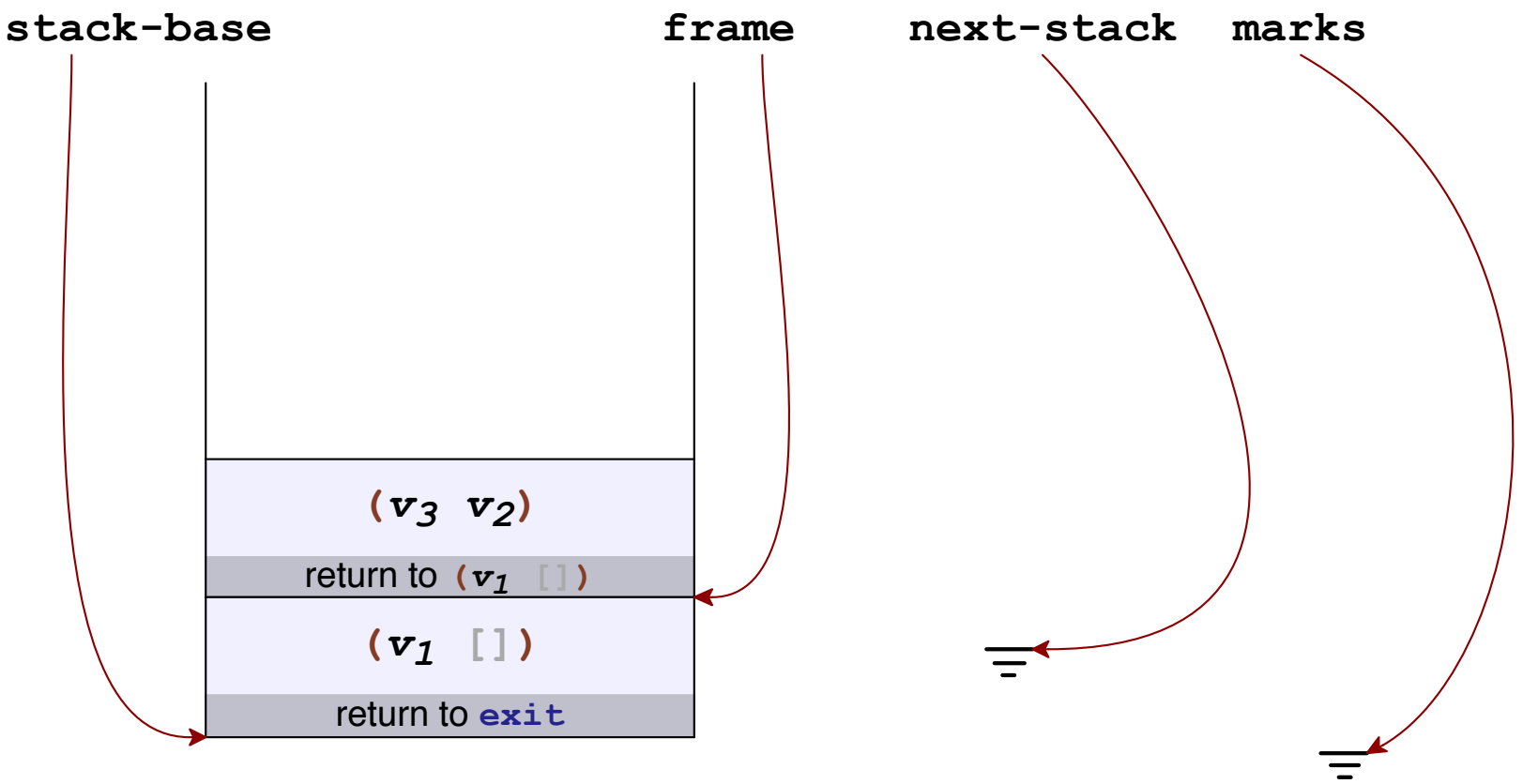
Optimistic One-Shot Continuations



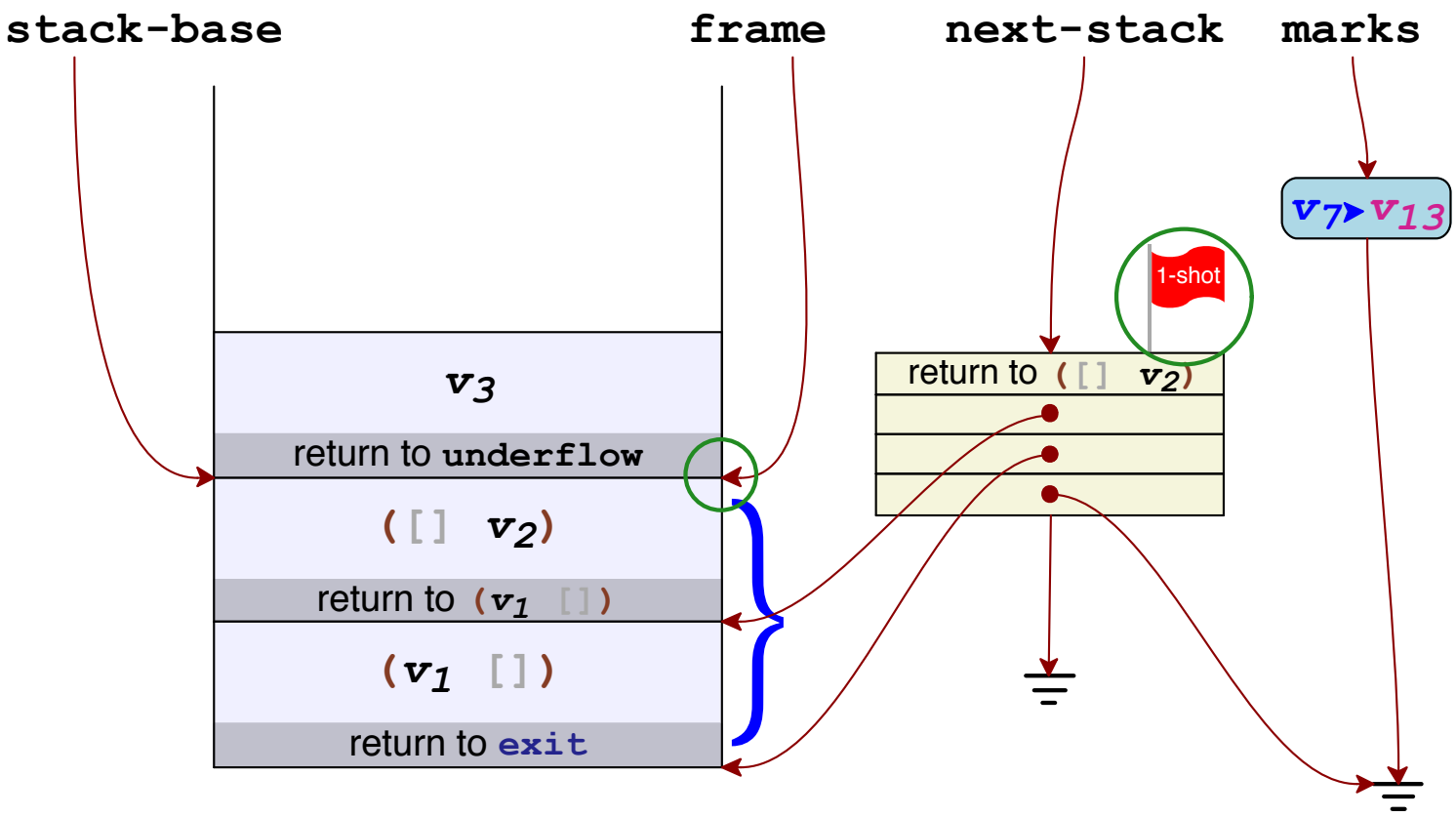
Optimistic One-Shot Continuations



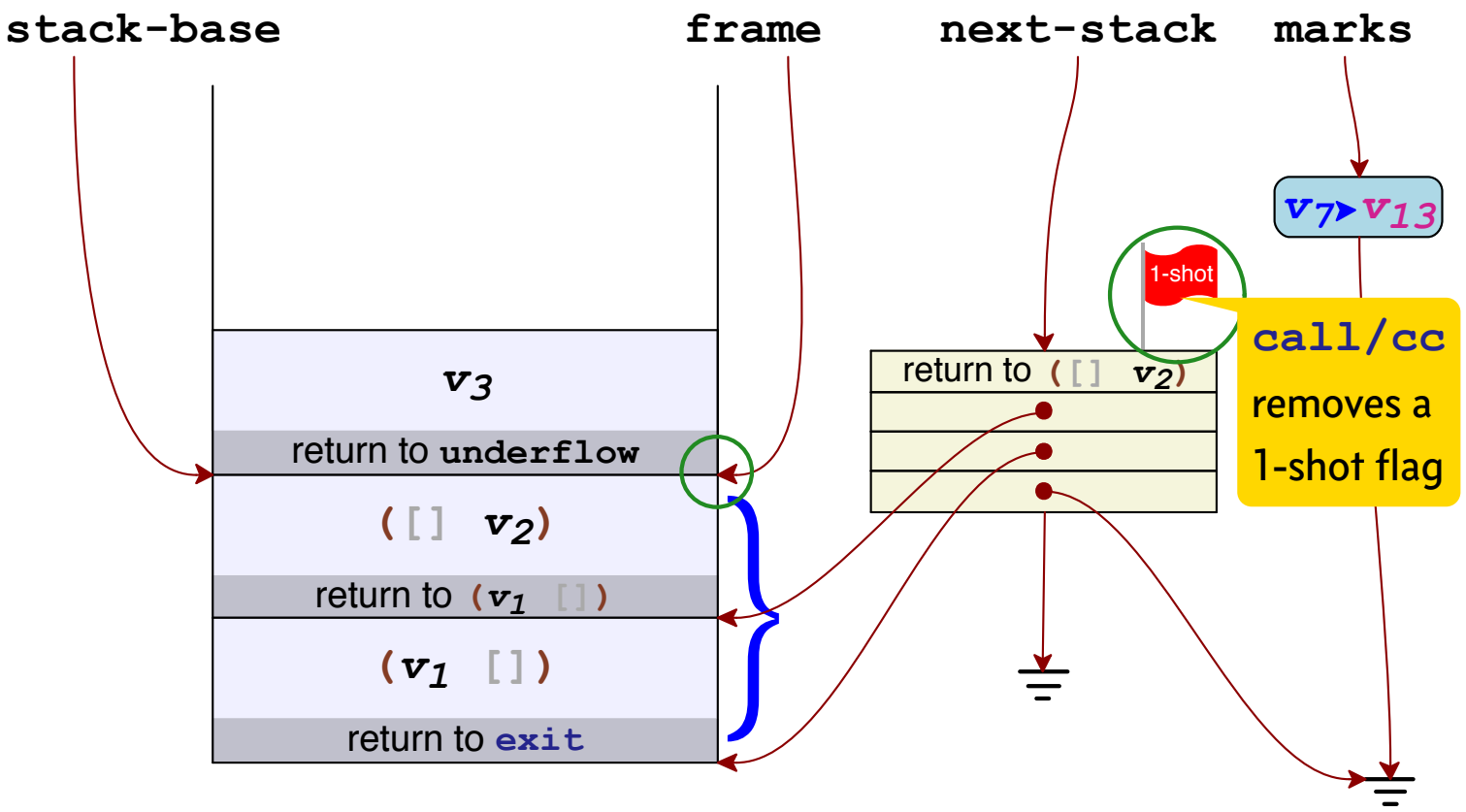
Optimistic One-Shot Continuations



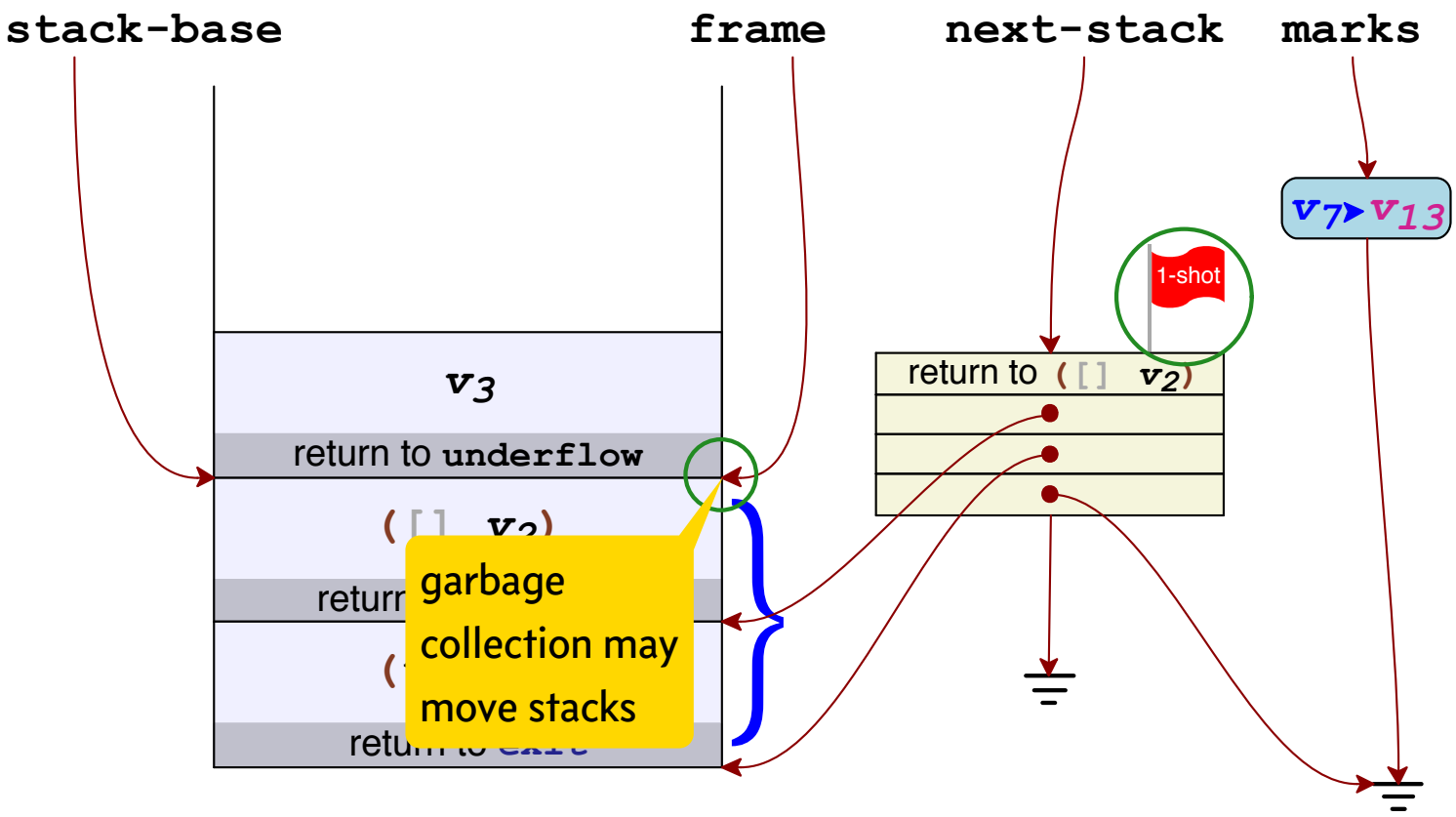
Optimistic One-Shot Continuations



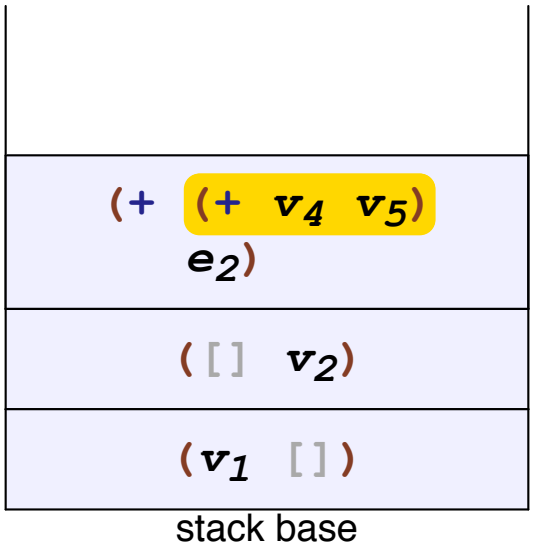
Optimistic One-Shot Continuations



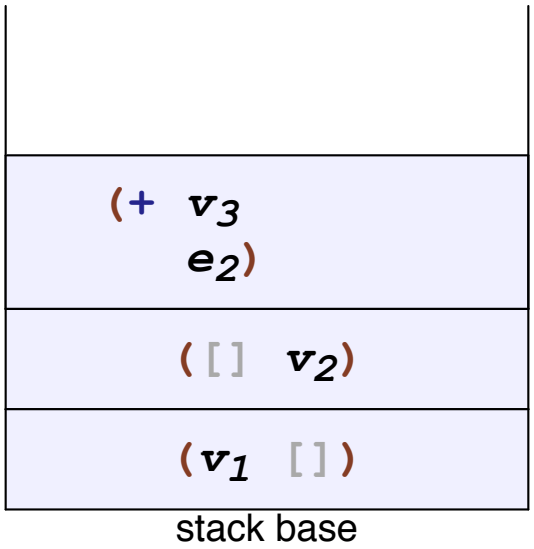
Optimistic One-Shot Continuations



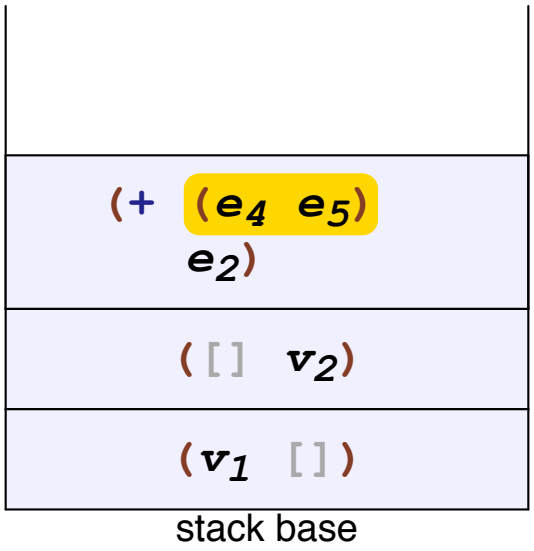
Optimization for Primitives: Don't Create Frame



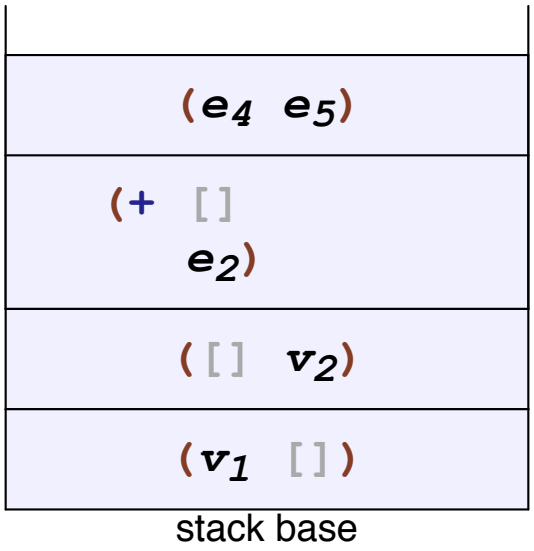
Optimization for Primitives: Don't Create Frame



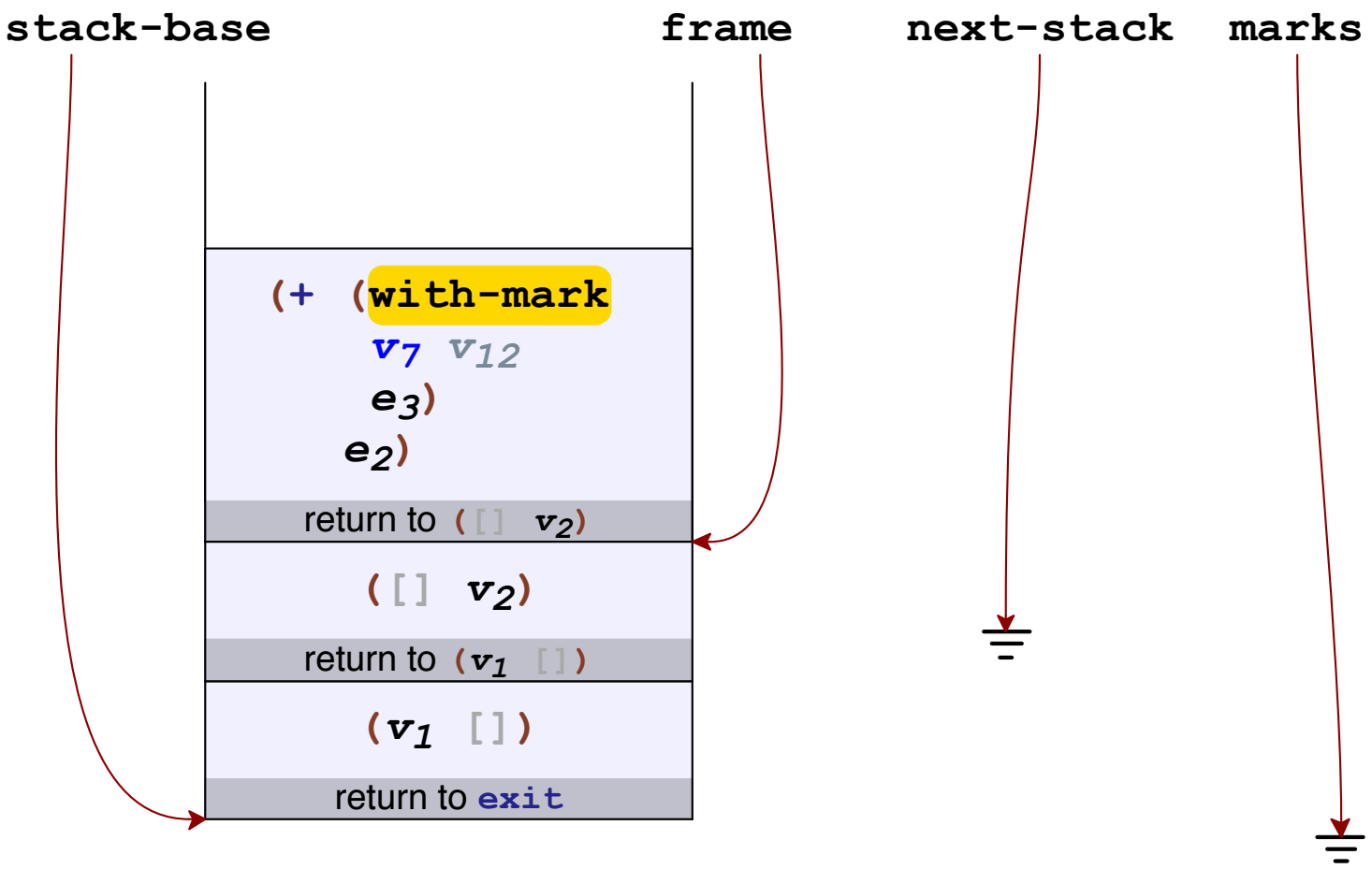
Create Frame for Nested Call



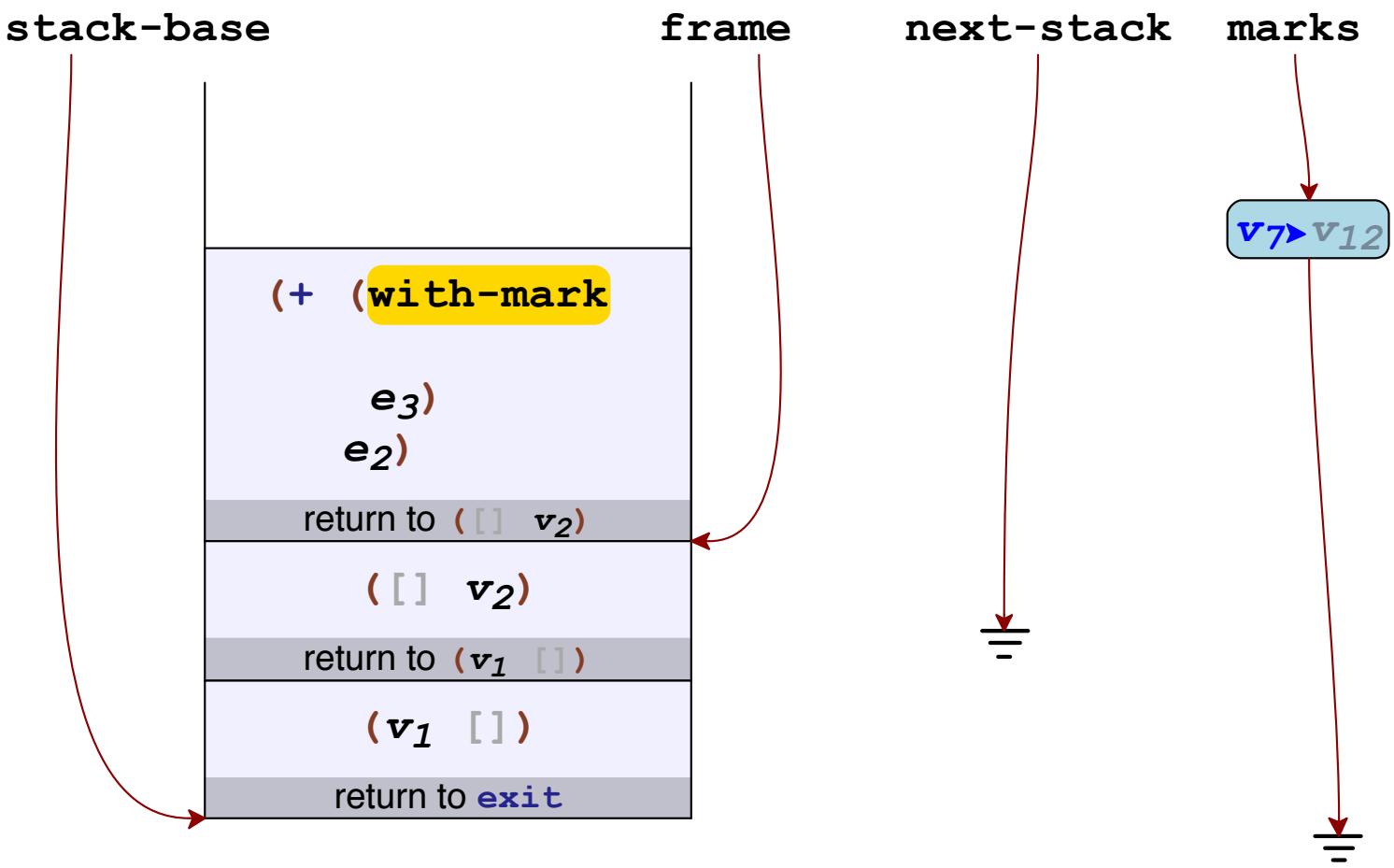
Create Frame for Nested Call



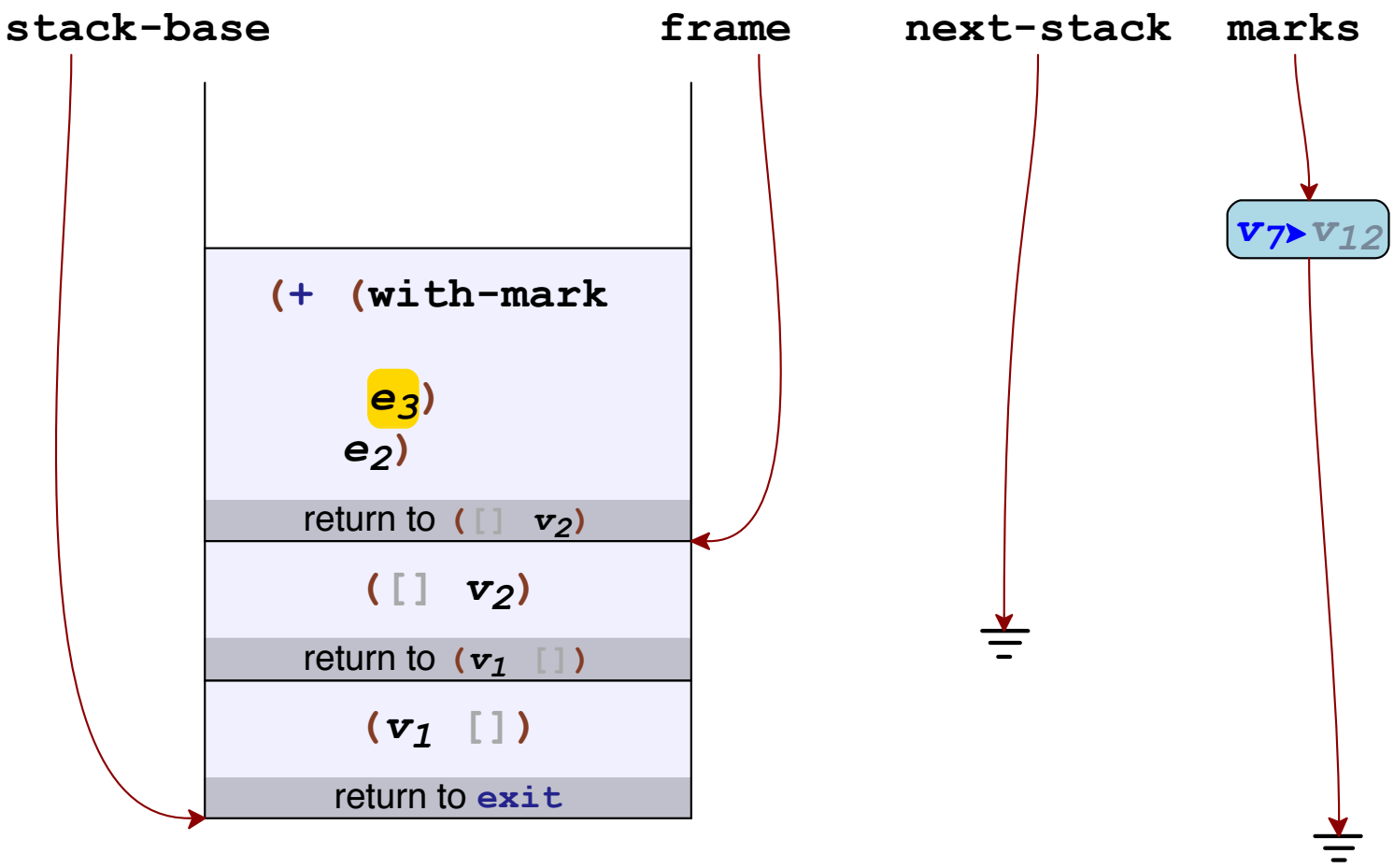
Optimzation for Non-Tail Marking



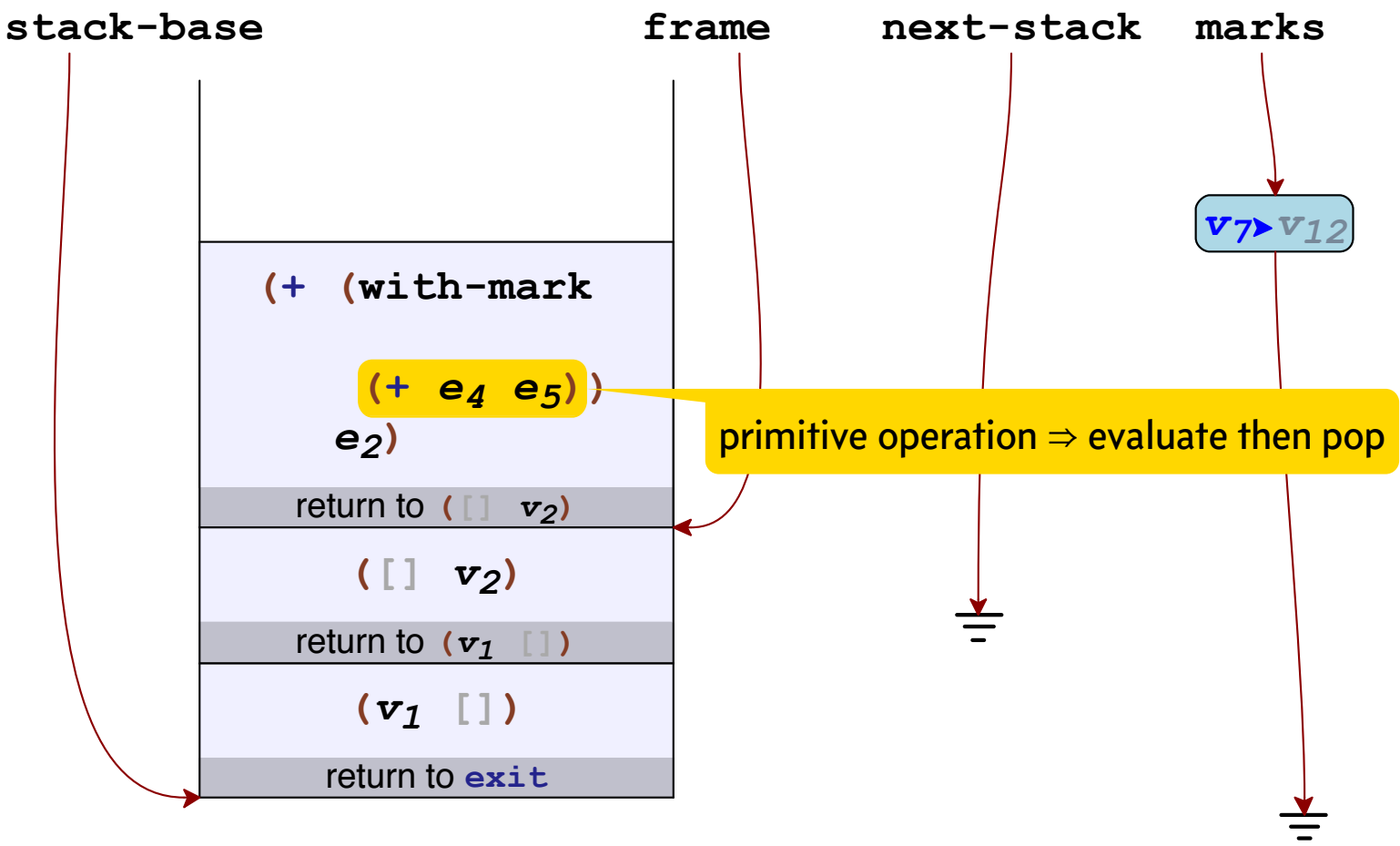
Optimzation for Non-Tail Marking



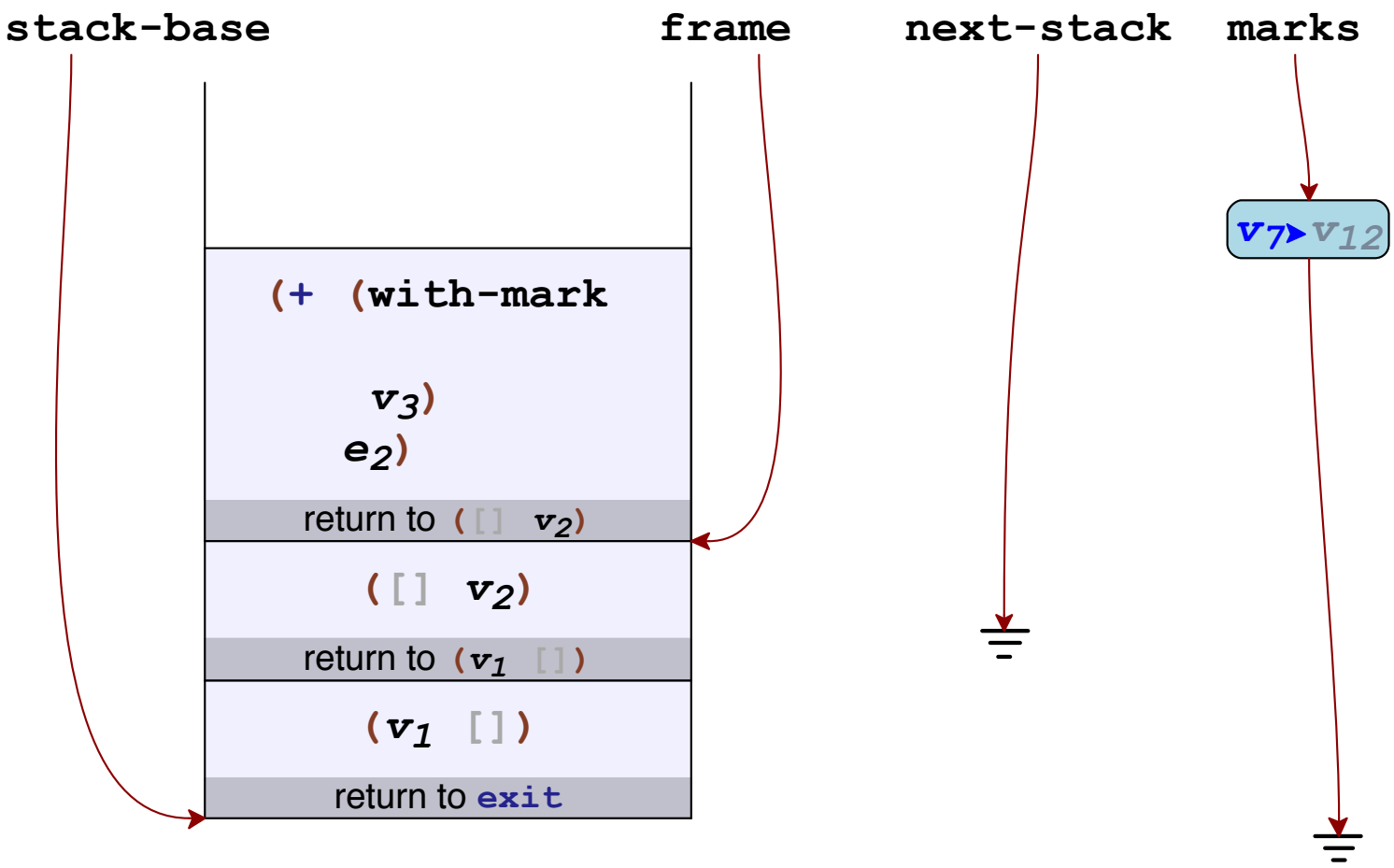
Optimzation for Non-Tail Marking



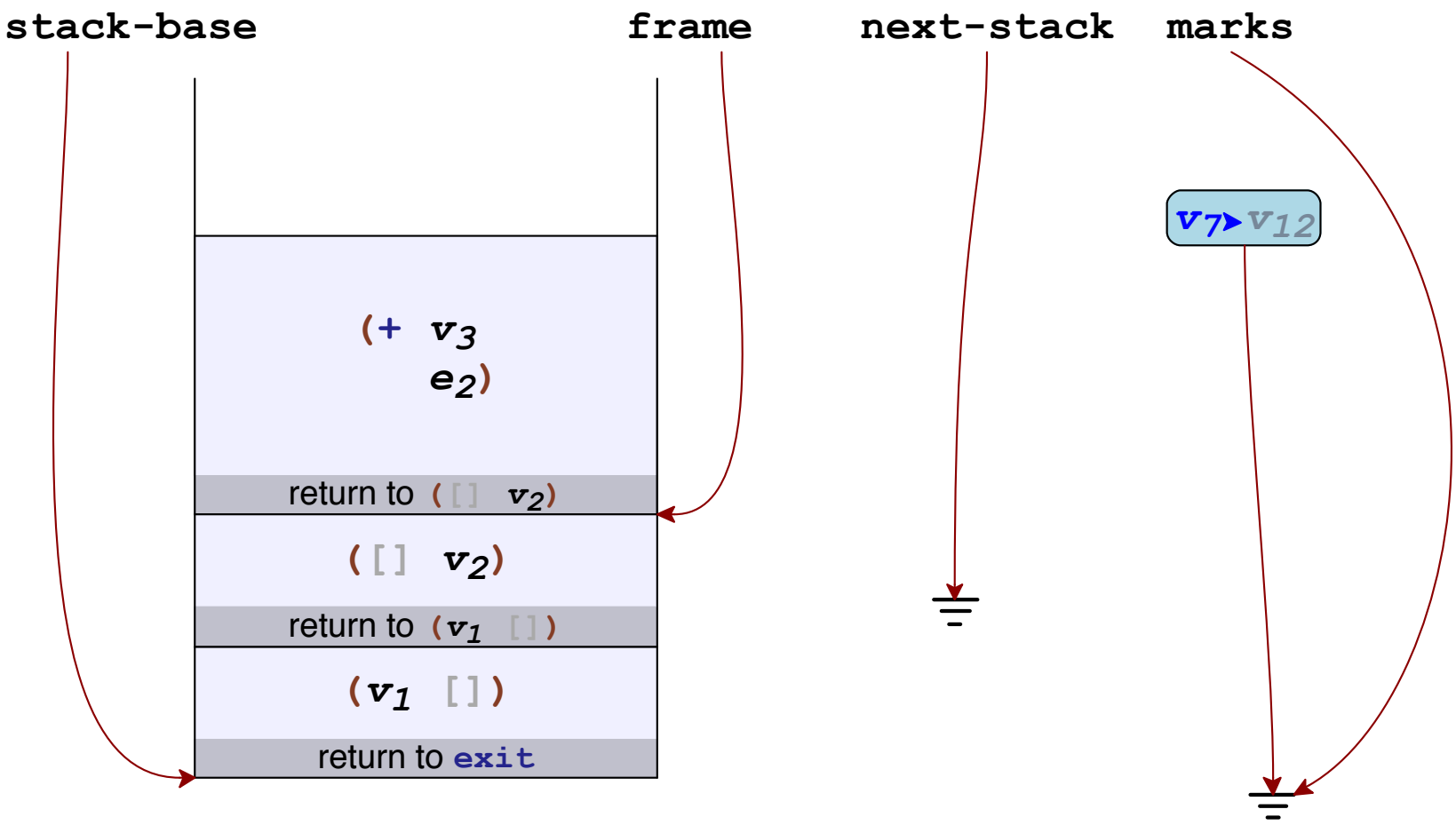
Non-Tail Marking with Primitive Operation



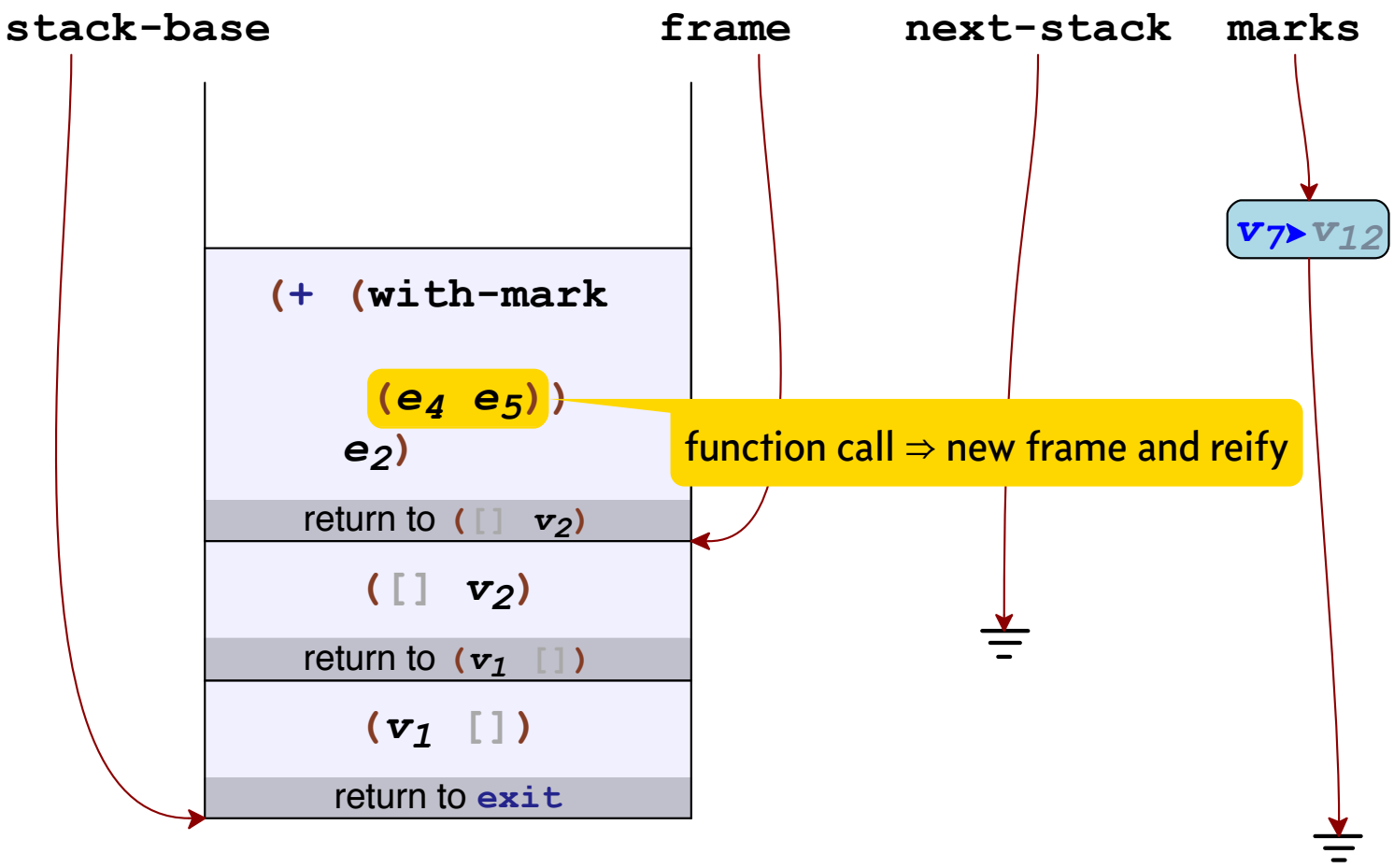
Non-Tail Marking with Primitive Operation



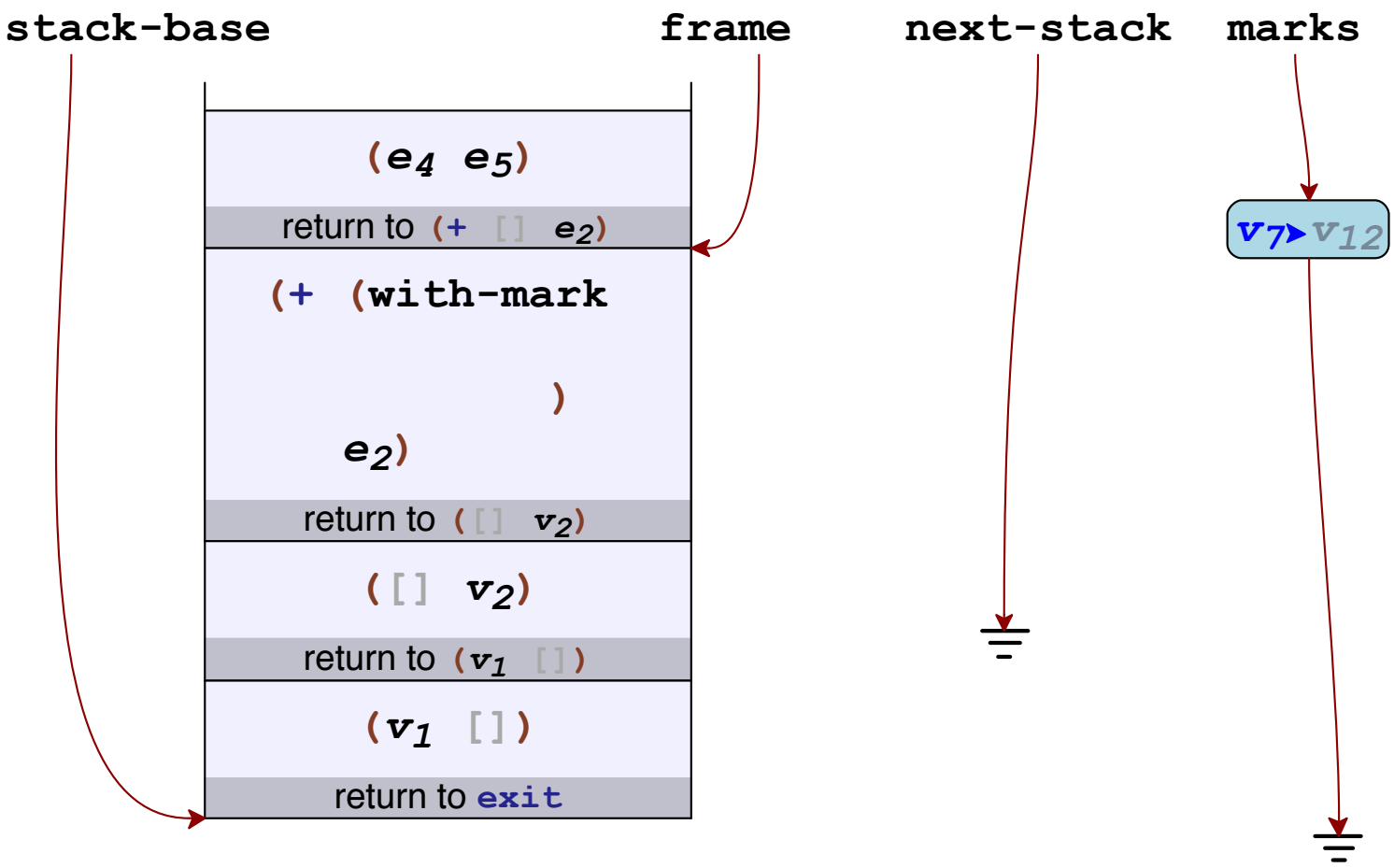
Non-Tail Marking with Primitive Operation



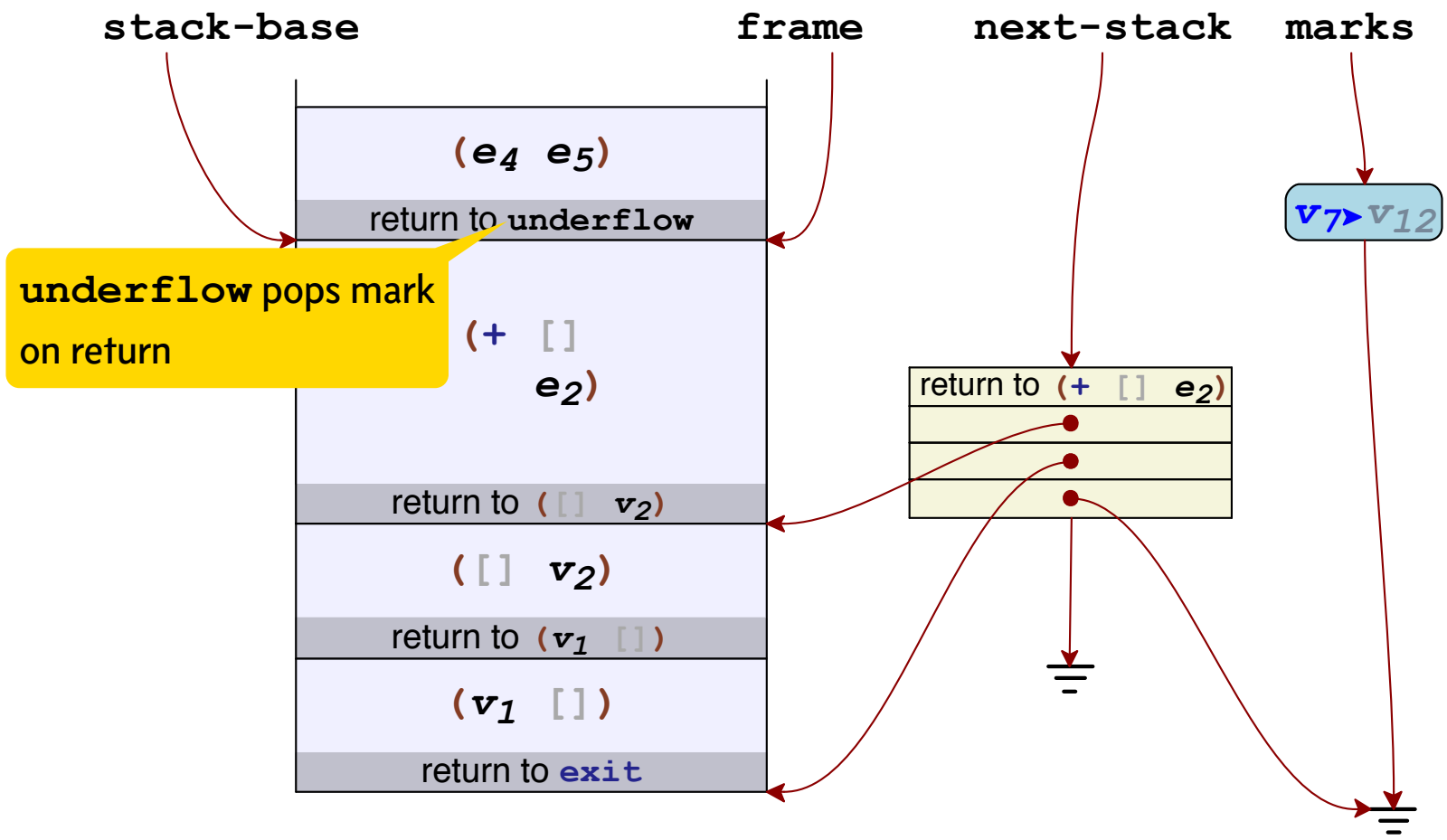
Non-Tail Marking with Non-Tail Call



Non-Tail Marking with Non-Tail Call



Non-Tail Marking with Non-Tail Call



Optimization Effectiveness

speedup relative to optimization disabled

	contracts	*applications*	
Optimisitic 1-shot	×**1.4**	×**1.04** – ×**1.05**	±0.03
Intraframe `with-mark`	×**2.0**	×**1.01** – ×**1.07**	±0.03
Non-marking primitives	×**1.4**	×**1.00** – ×**1.04**	±0.03
Runtime + optimizations	×**3.4**	×**1.10** – ×**1.25**	±0.03

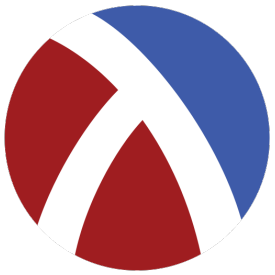
Continuation Marks

Enable

- dynamic scope
- control introspection

Compatible with

- optimizing compilation
- delimited control, concurrency, parallelism



Implement using

- stack-based continuations
- specific optimizations