Adding Delimited and Composable Control to a Production Programming Environment

Matthew Flatt University of Utah

Gang Yu Institute of Software, Chinese Academy of Sciences

Robert Bruce Findler University of Chicago

Matthias Felleisen Northeastern University

Scheme provides first-class *continuations*

Continuations are useful for implementing a **web server**

Web Servlet with Continuations

```
(define (paper-search-servlet)
  (let ([terms (get-search-terms)])
    (find-paper terms)))
```

Web Servlet with Continuations

send back HTML form, then
wait for answer as new request

```
(define (paper-search-servlet)
  (let ([terms (get-search-terms)])
    (find-paper terms)))
```

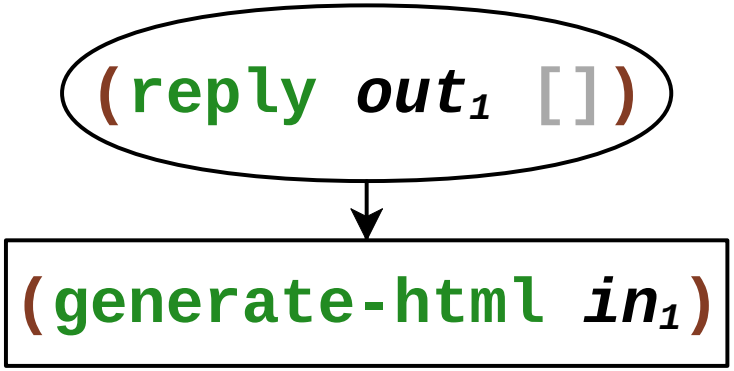
Implementing a Web Server

```
(serve out₁ in₁)
```

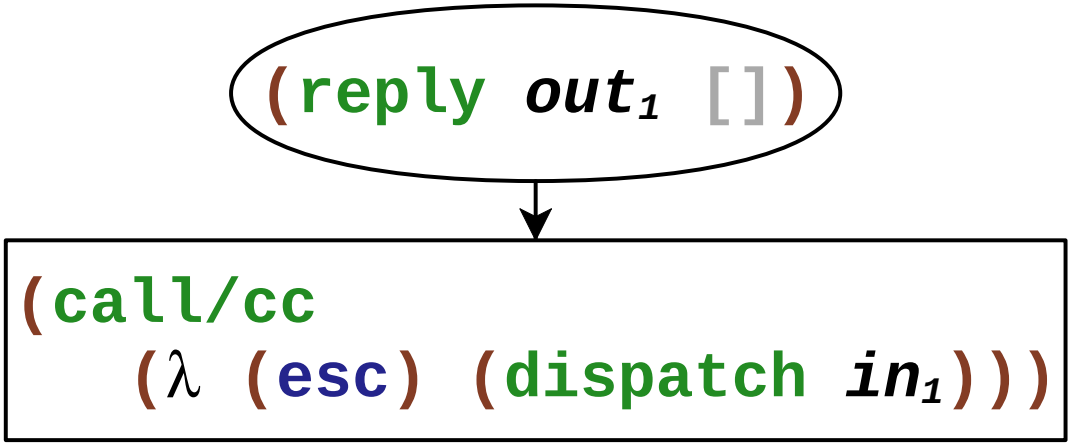
Implementing a Web Server

```
(reply out1 (generate-html in1))
```

Implementing a Web Server

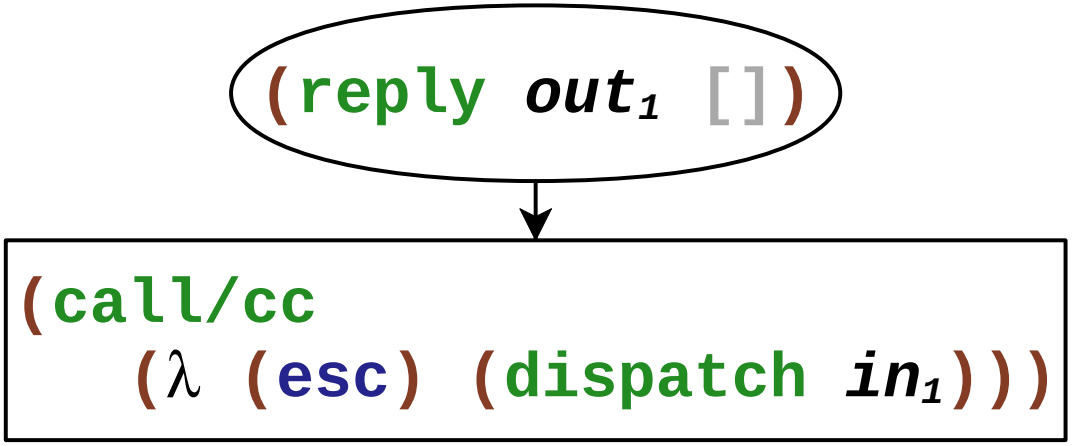


Implementing a Web Server



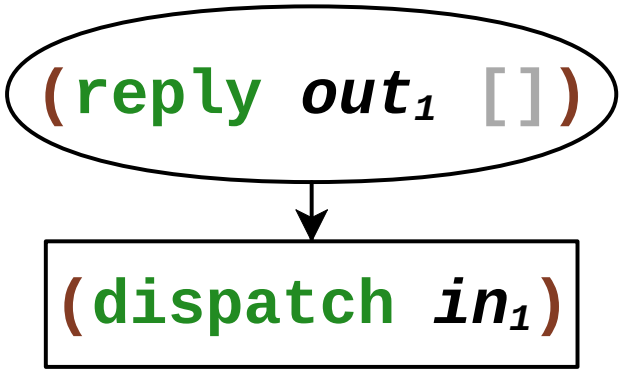
Implementing a Web Server

esc = (reply out_1 [])



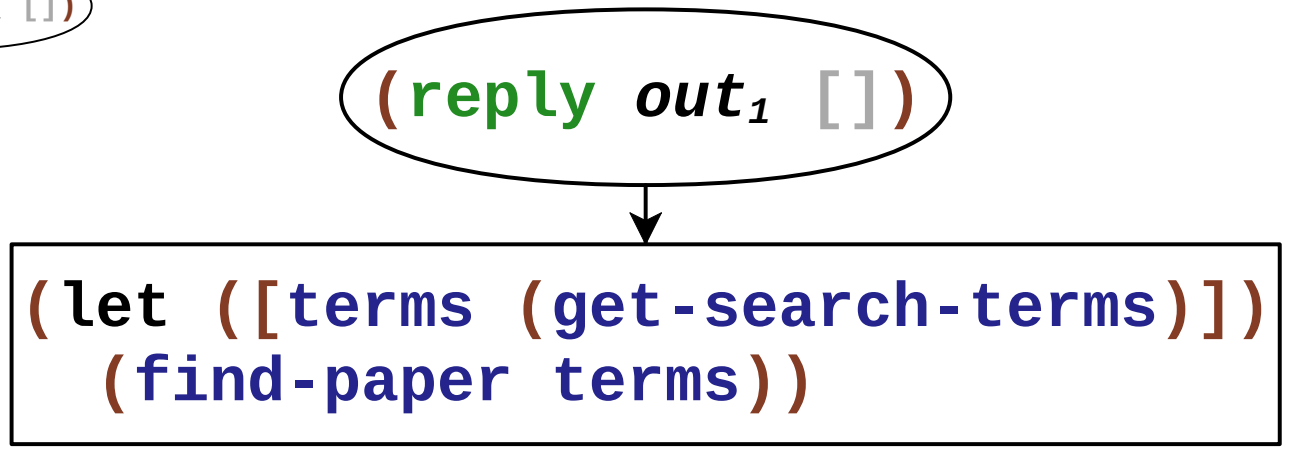
Implementing a Web Server

```
esc = (reply out₁ [])
```



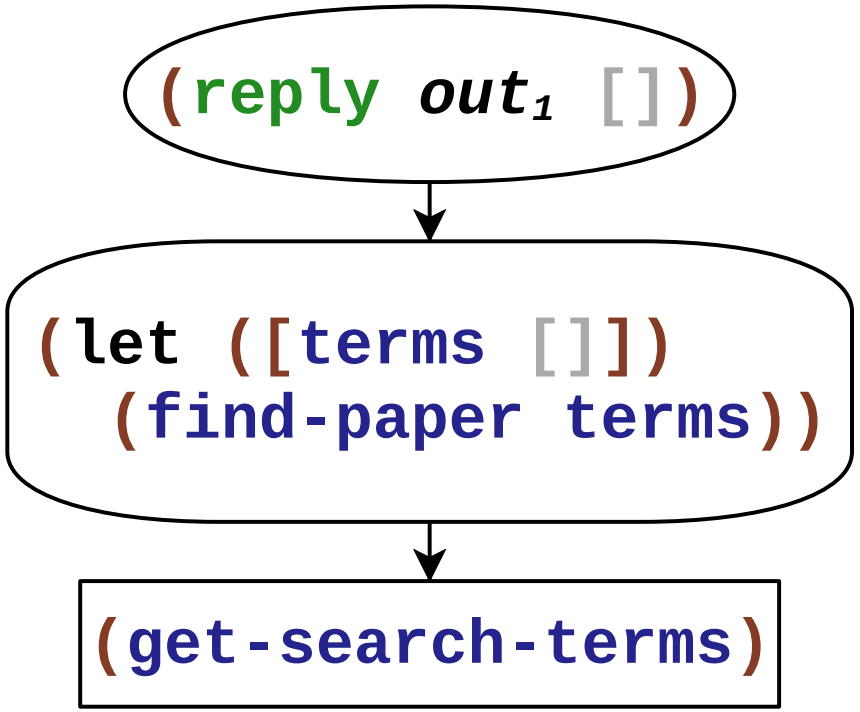
Implementing a Web Server

esc = (reply out_1 [])



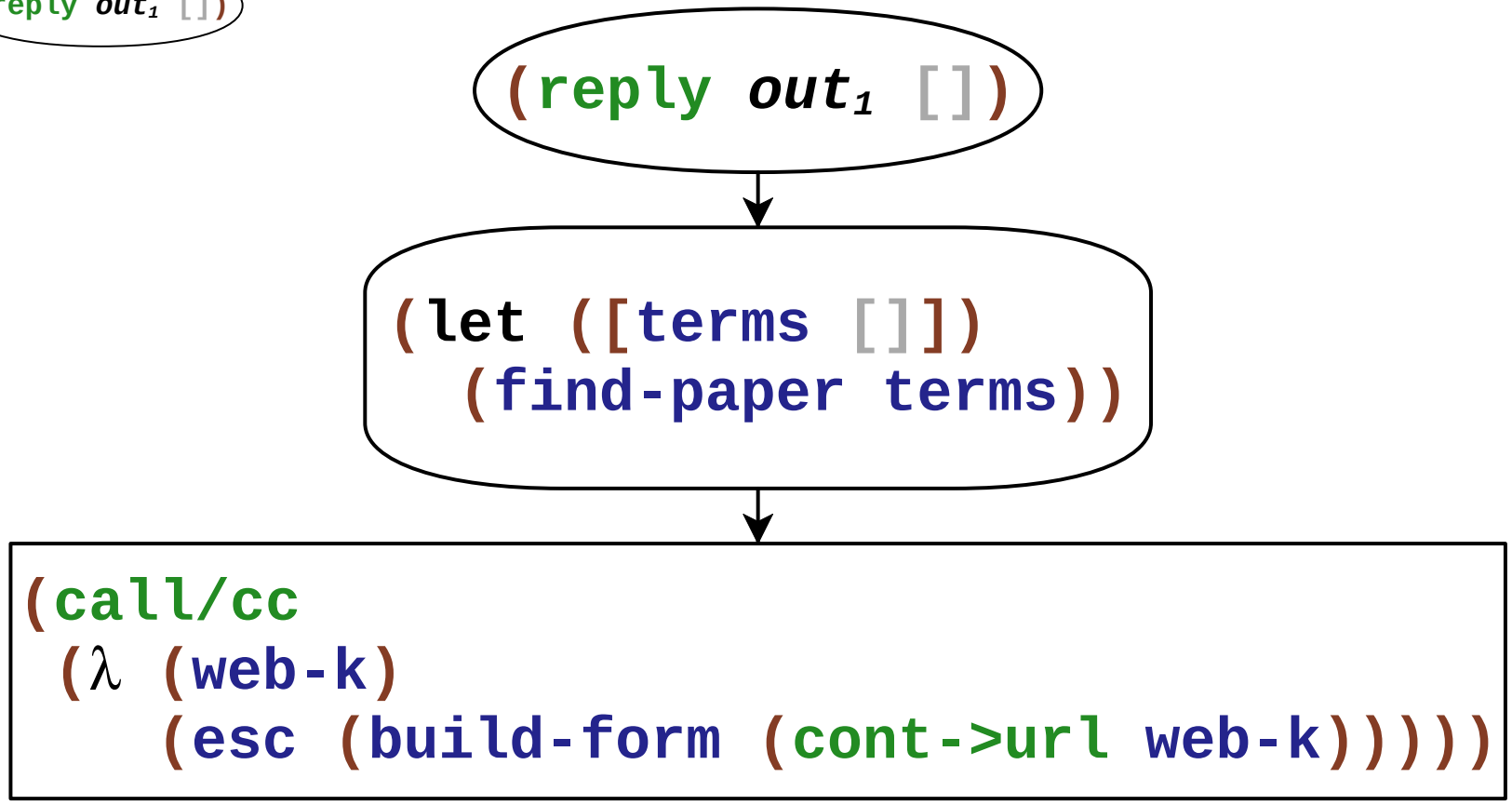
Implementing a Web Server

esc = (reply out_1 [])

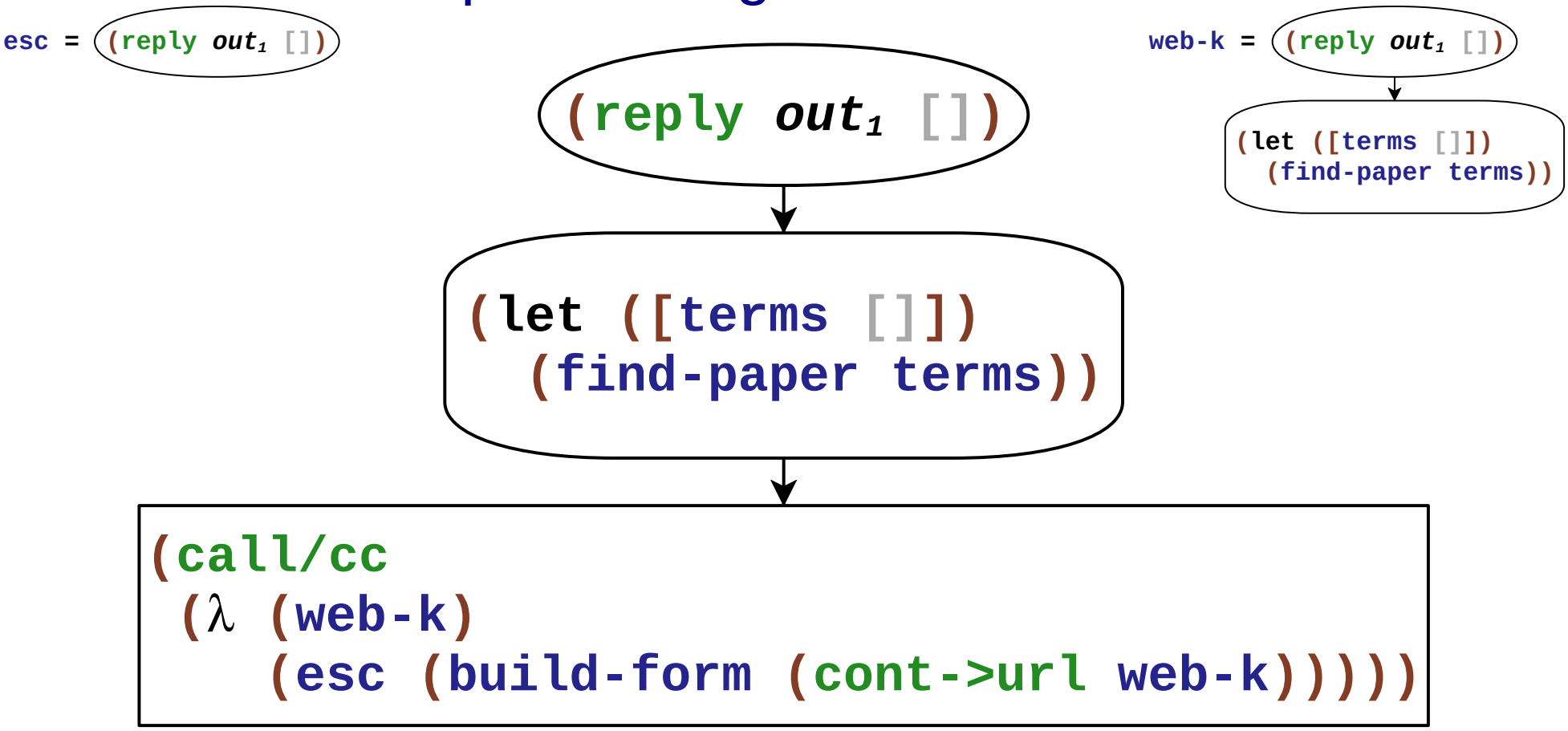


Implementing a Web Server

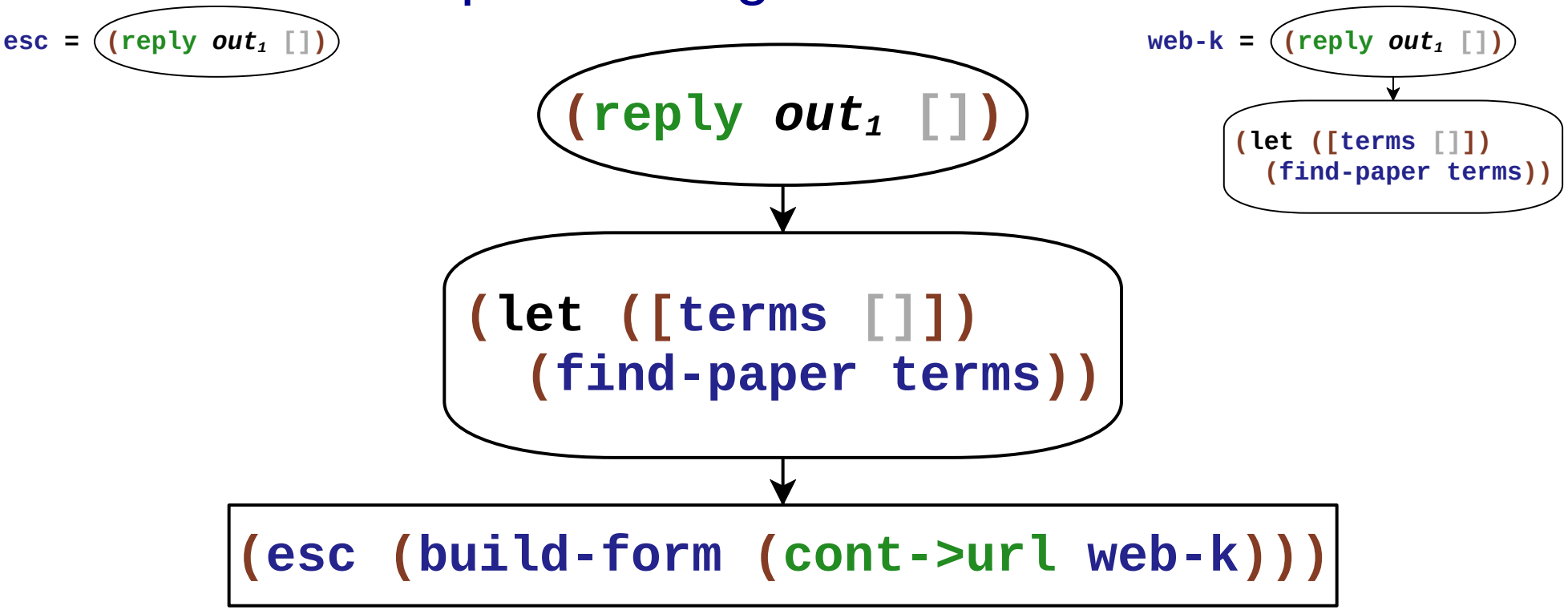
esc = (reply out_1 [])



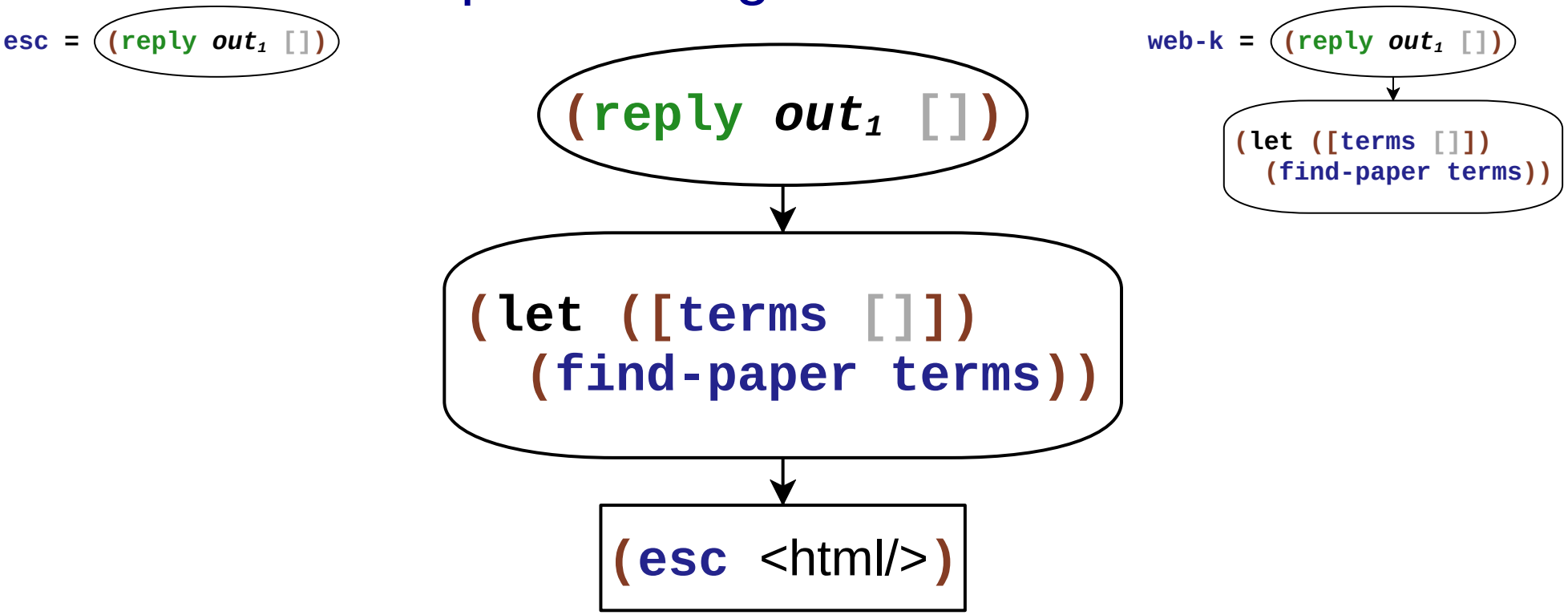
Implementing a Web Server



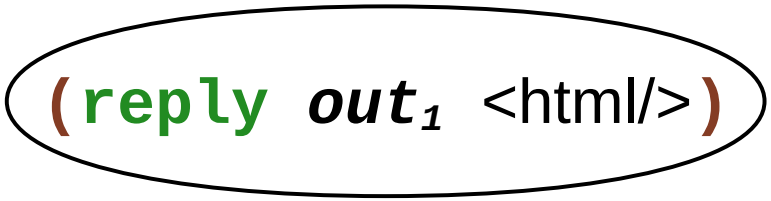
Implementing a Web Server

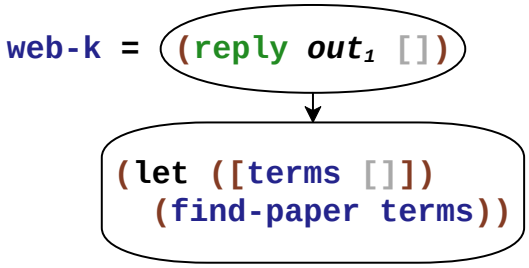


Implementing a Web Server

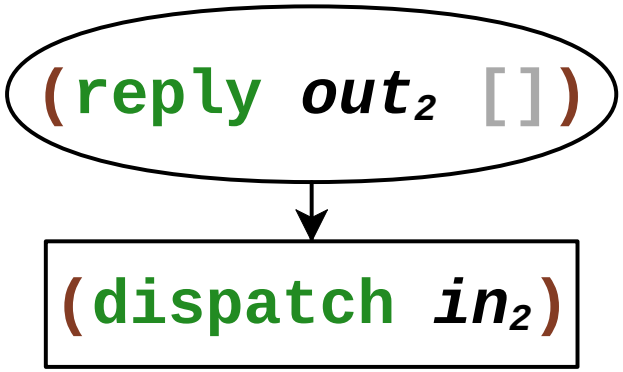


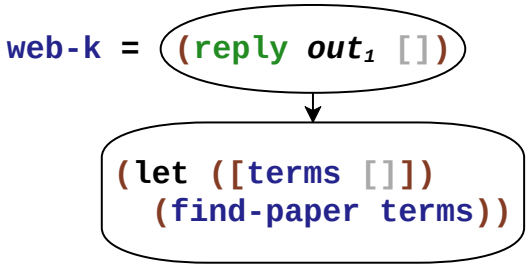
Implementing a Web Server



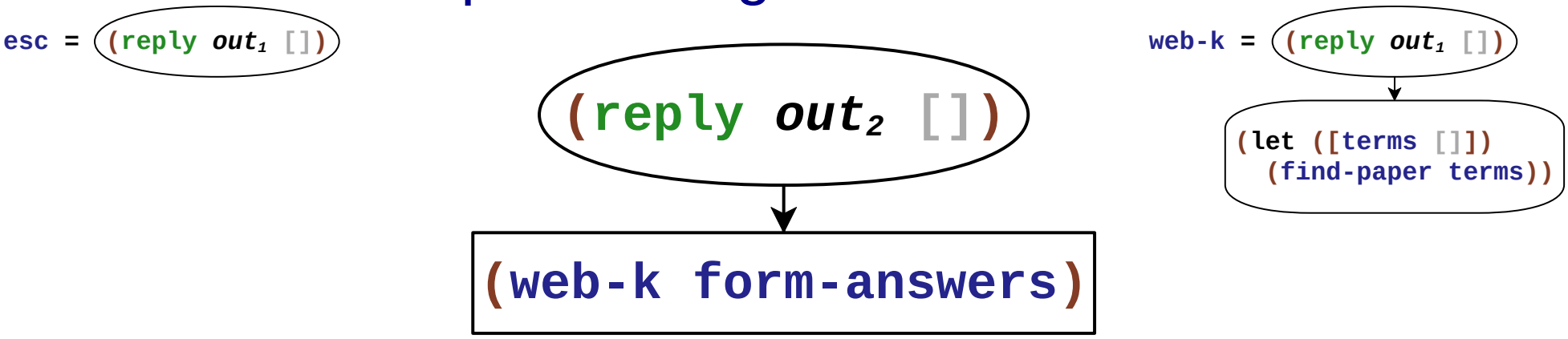


Implementing a Web Server

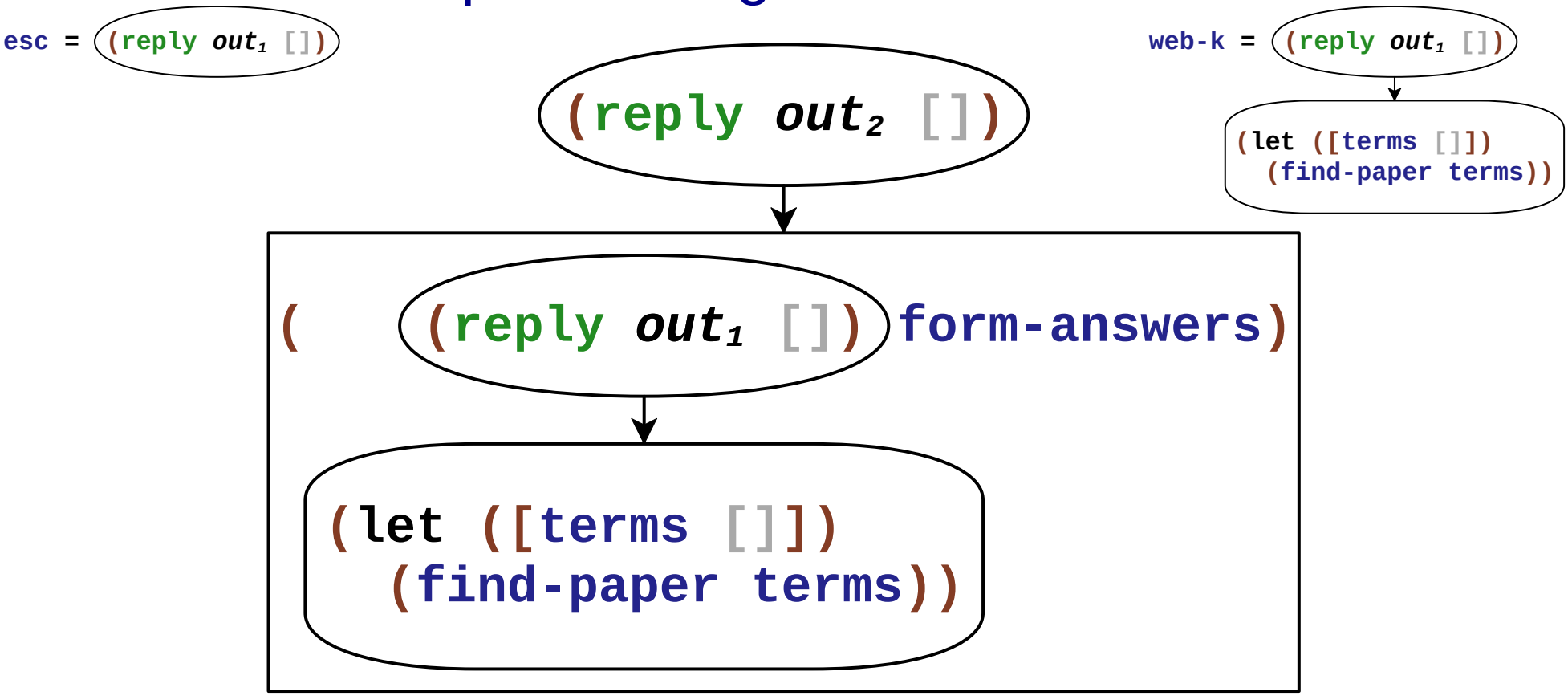




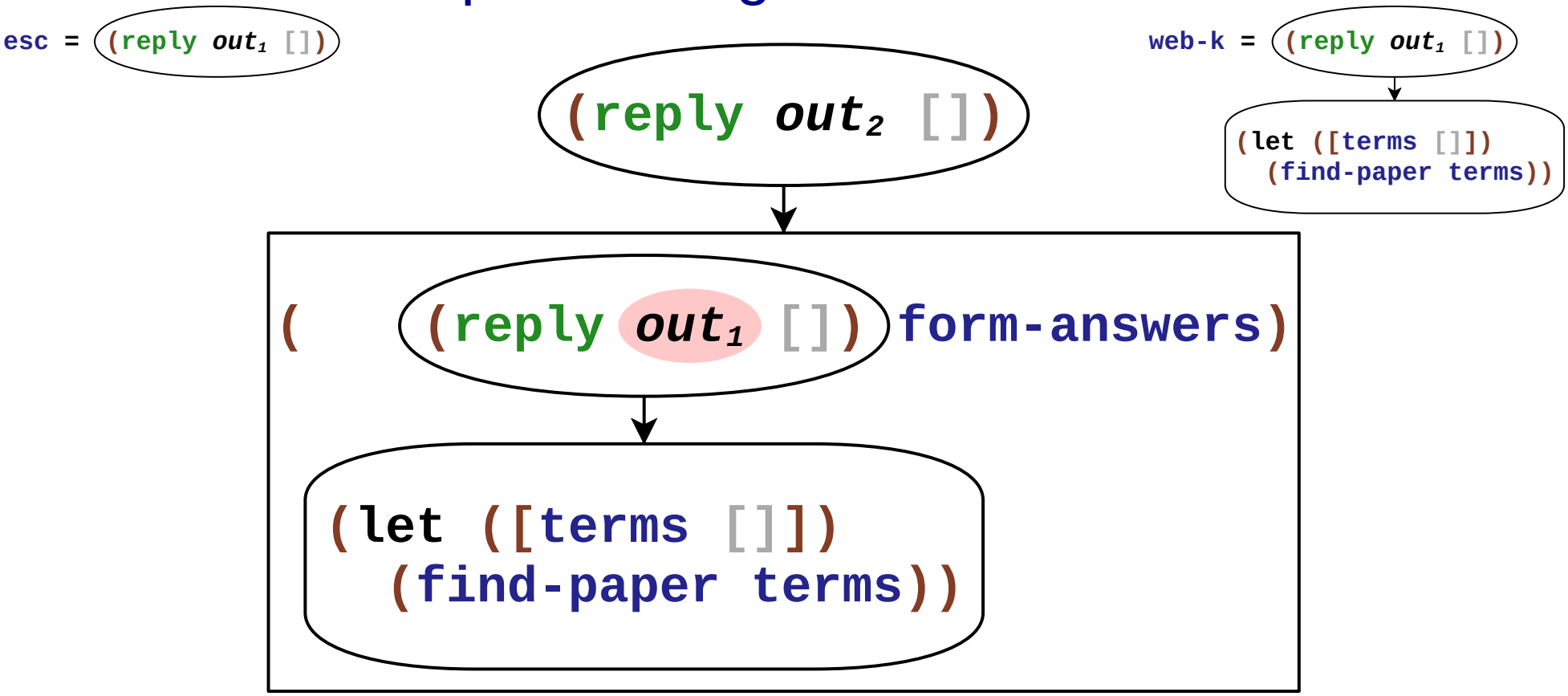
Implementing a Web Server



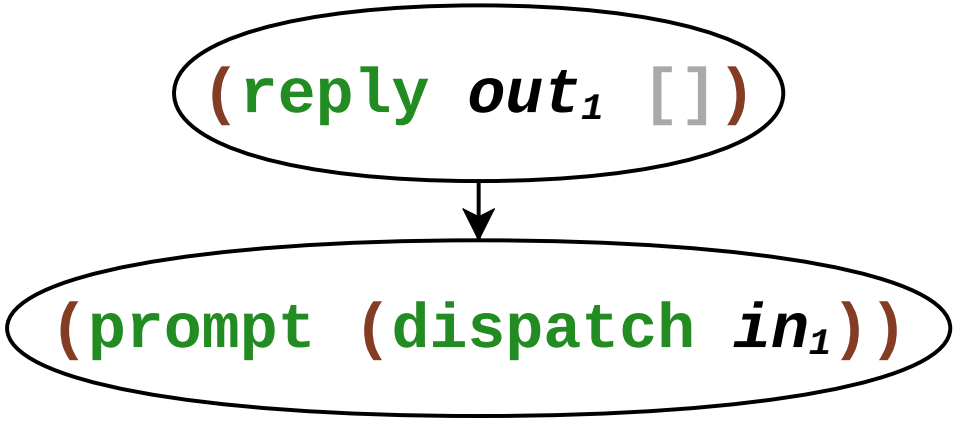
Implementing a Web Server



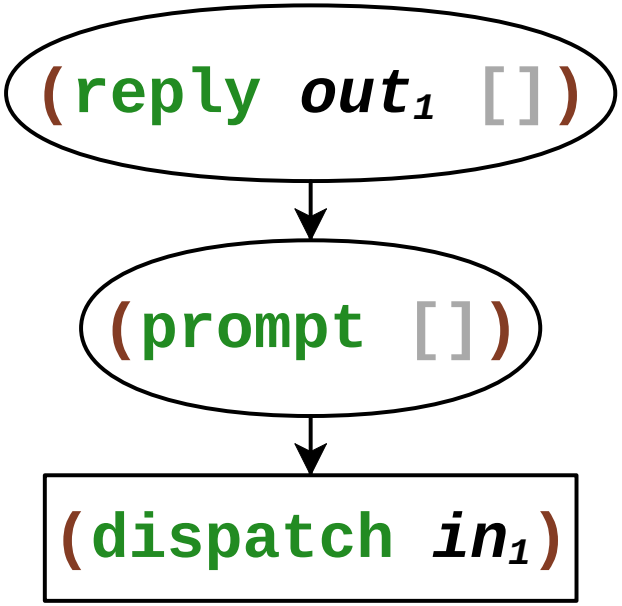
Implementing a Web Server



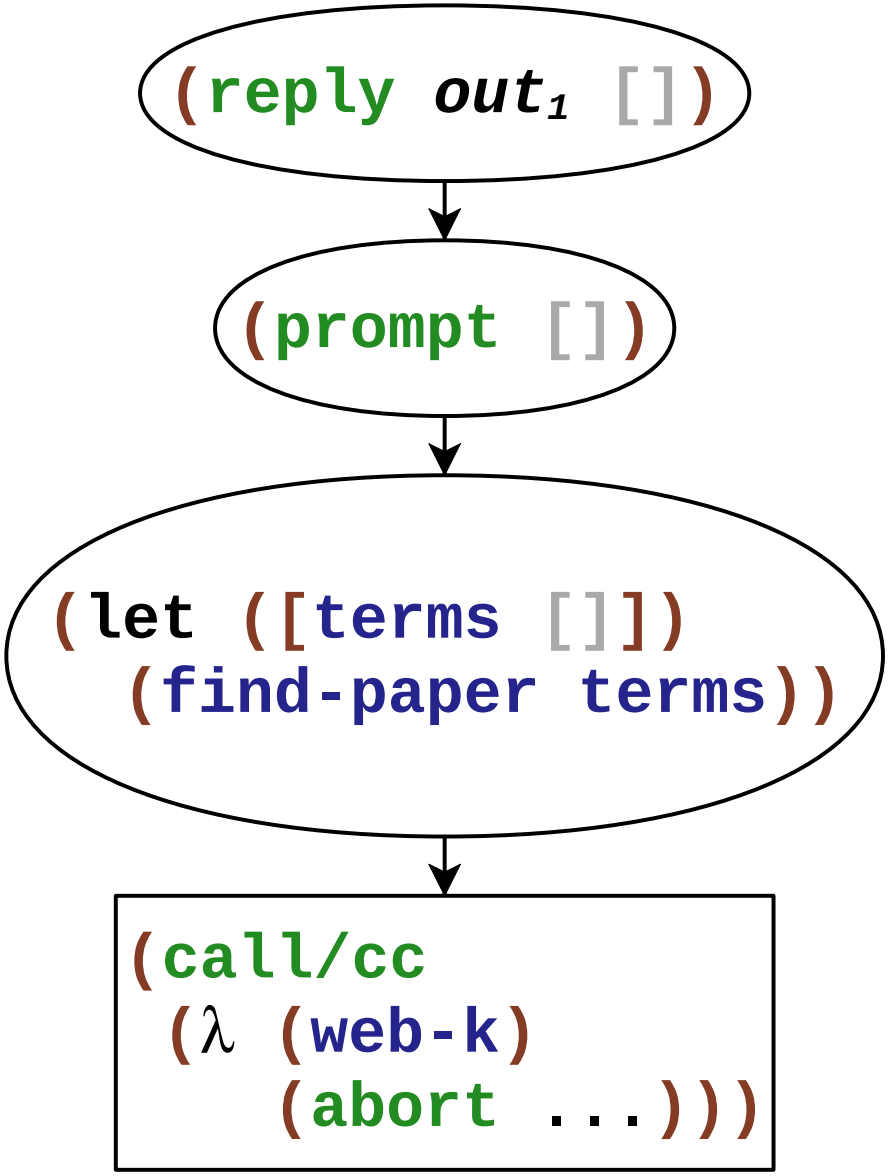
Web Server with Prompt



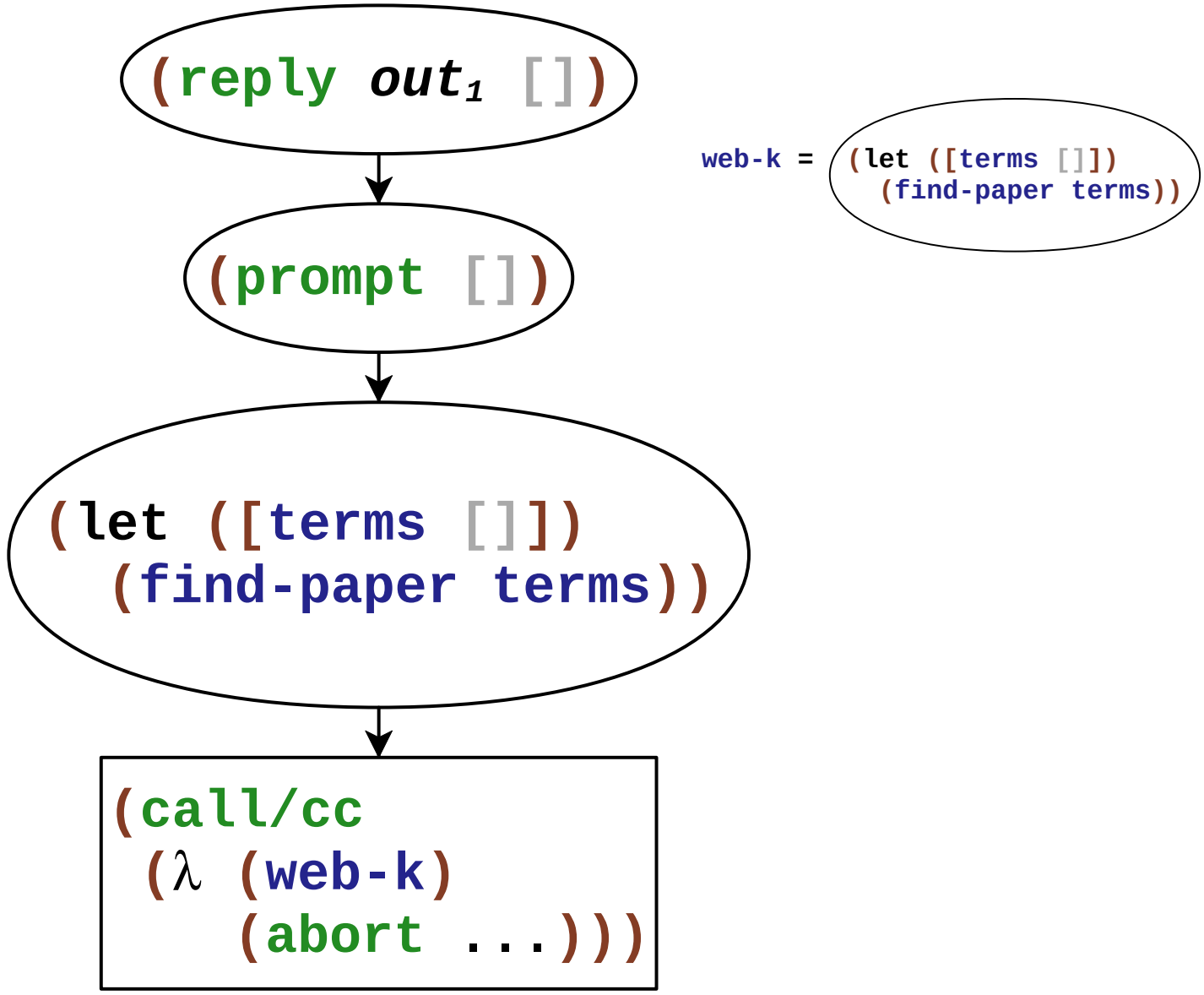
Web Server with Prompt



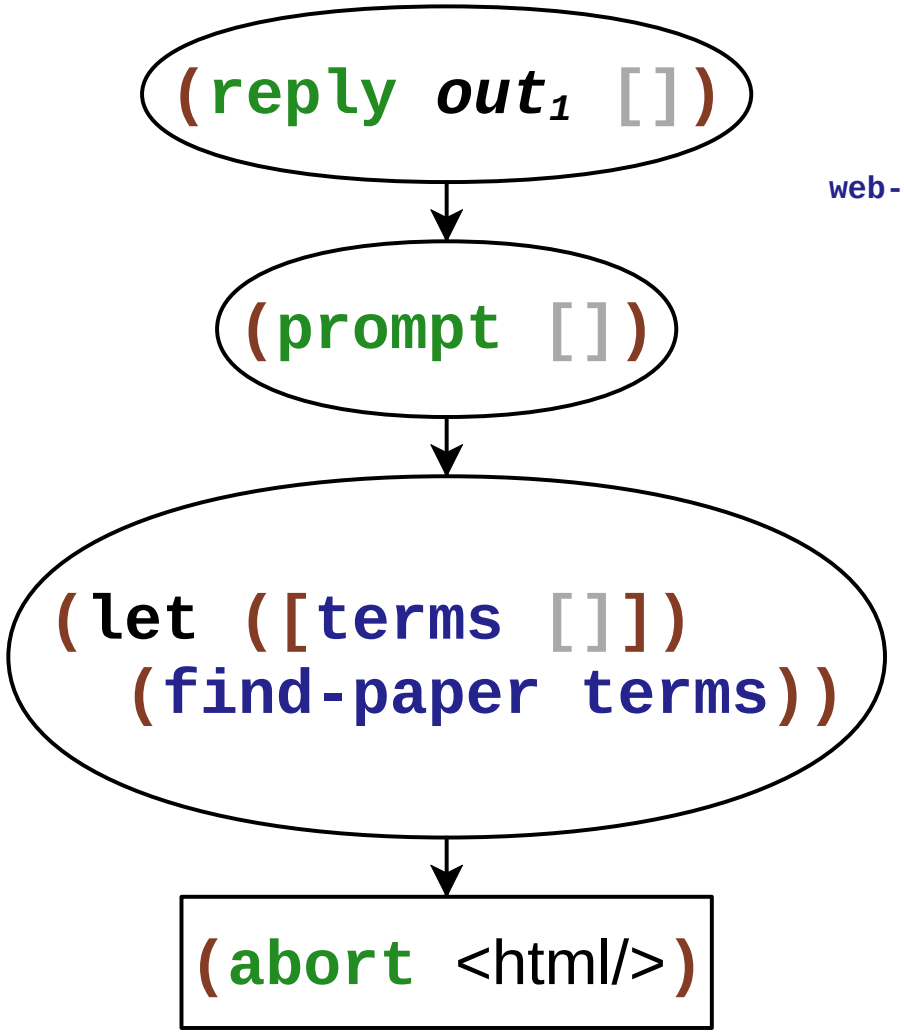
Web Server with Prompt



Web Server with Prompt



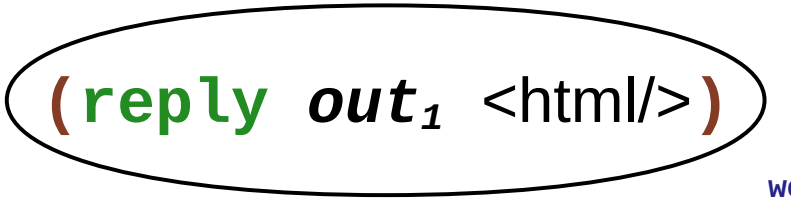
Web Server with Prompt



web-k =

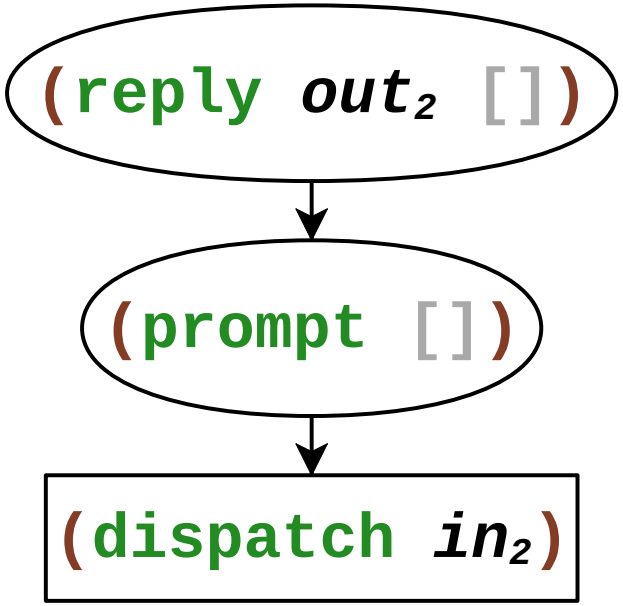
```
(let ([terms []])
  (find-paper terms))
```

Web Server with Prompt



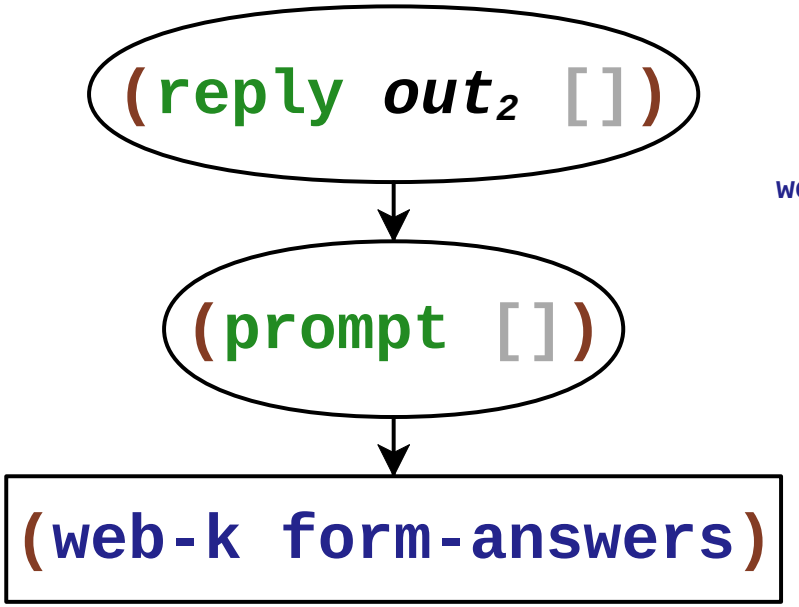

```
web-k = (let ([terms []])
           (find-paper terms))
```

Web Server with Prompt



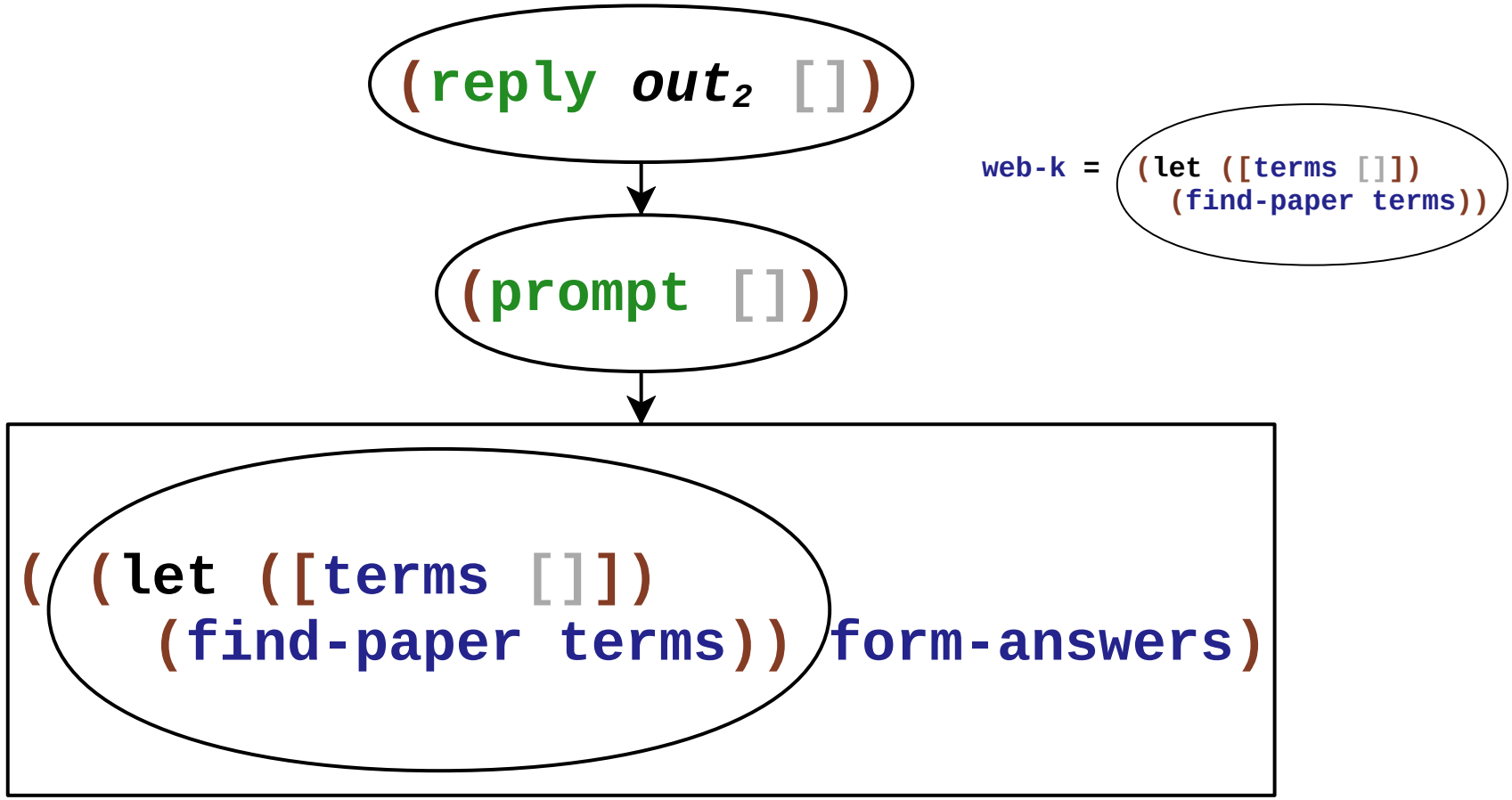

```
web-k = (let ([terms []])
           (find-paper terms))
```

Web Server with Prompt

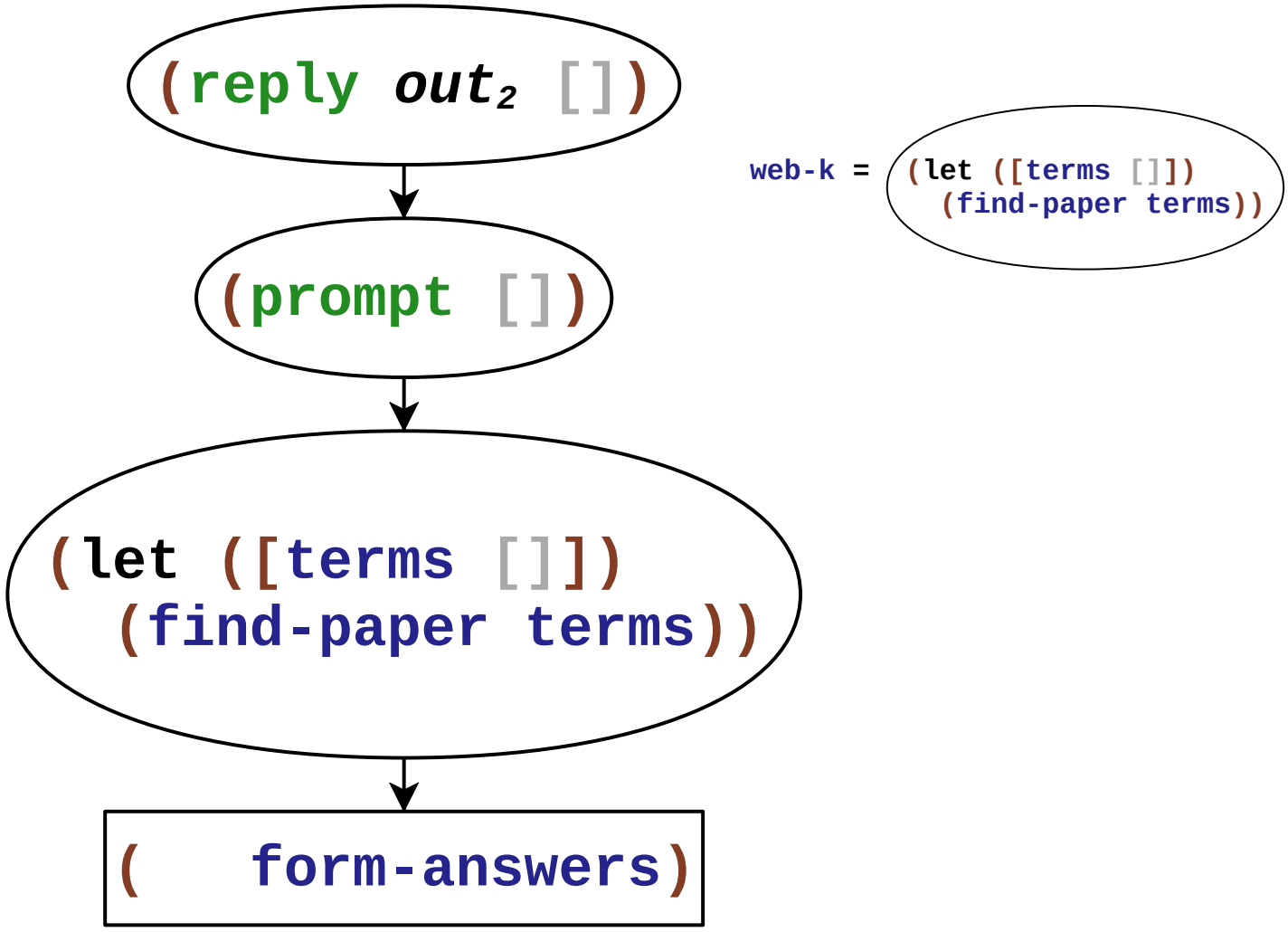



```
web-k = (let ([terms []])
           (find-paper terms))
```

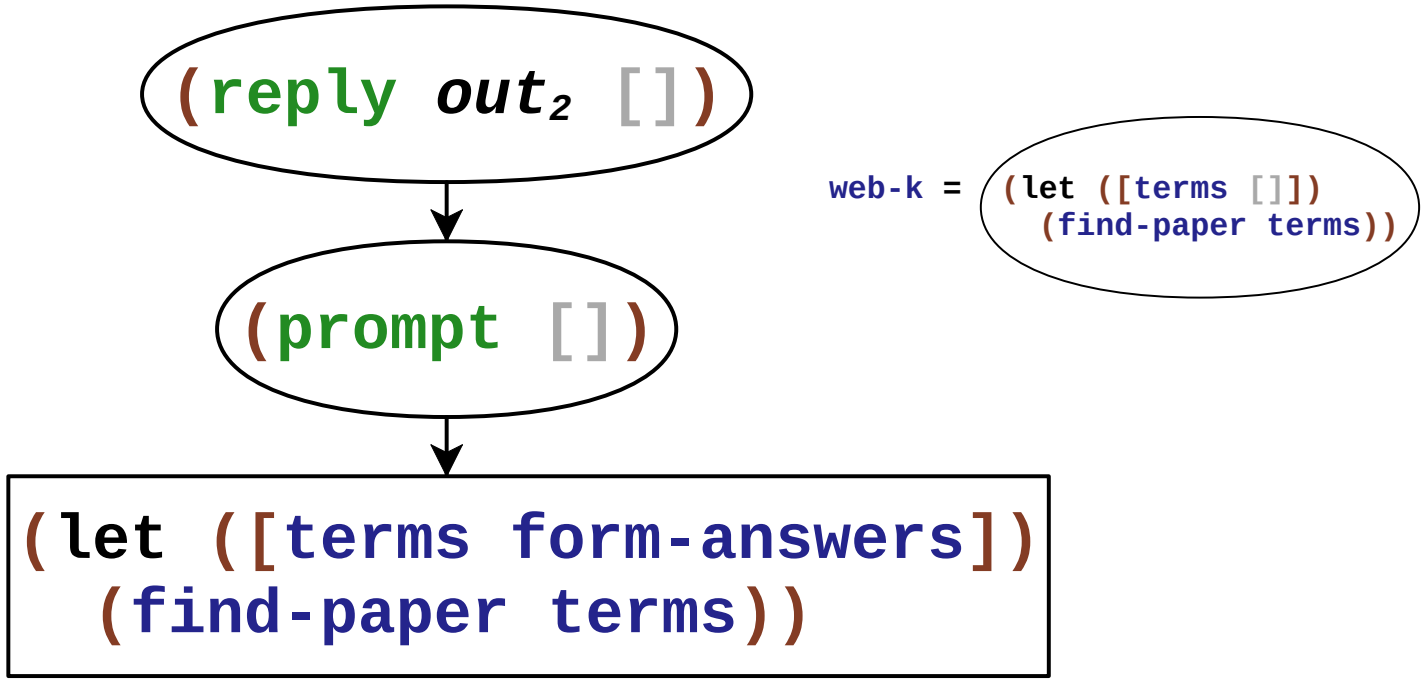
Web Server with Prompt



Web Server with Prompt



Web Server with Prompt



Rolling Your Own vs. Language Extension

Delimited control via `call/cc` doesn't work right with

- exceptions
- dynamic binding
- `dynamic-wind`

Our goal: delimited control integrated with existing constructs

Papers on Delimited Continuations

Felleisen 88
Hieb and Dybvig 90
Queinnec and Serpette 91
Queinnec 93
Wadler 94
Rehof and Sørensen 94
Gunter et al. 95
Rehof 01
Kameyama and Hasegawa 03
Shan 04
Kiselyov et al. 06
Biernacki et al. 06
Danvy and Filinski 90
Sitaram and Felleisen 90
Sitaram 93
Moreau and Queinnec 94
deGroote 94
Gunter et al. 95
Thielecke 97
Gasbichler and Sperber 02
Ariola et al. 04
Saurin 05
Dybvig et al. 06

Papers on Design

Felleisen 88
Danvy and Filinski 90
Hieb and Dybvig 90
Sitaram and Felleisen 90
Queinnec and Serpette 91
Sitaram 93
Moreau and Queinnec 94
Gunter et al. 95
Kiselyov et al. 06
Dybvig et al. 06

Papers on Implementation

Gasbichler and Sperber 02

Dybvig et al. 06

Implementations

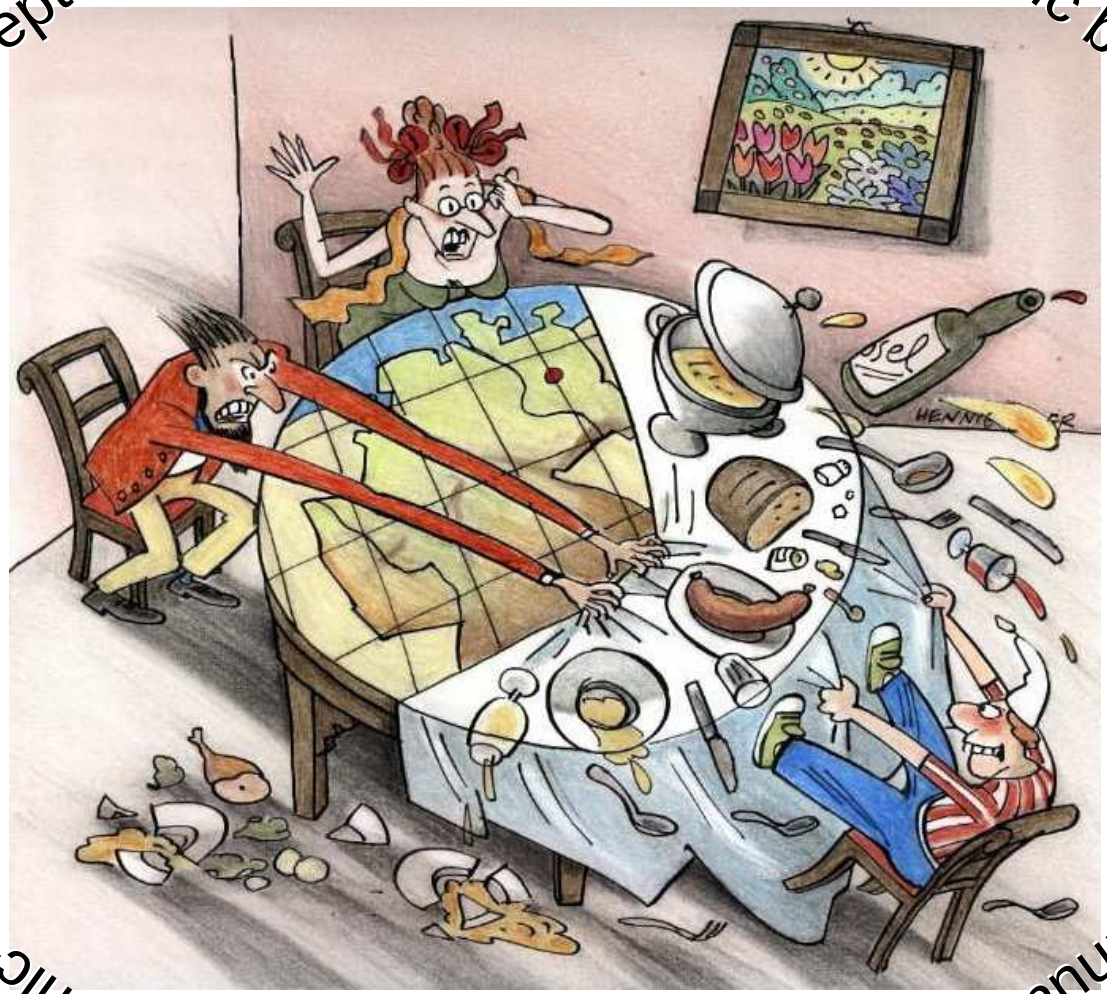
Implementations (Now)

Balance

exceptions

dynamic binding

prompts



abort

dynamic-wind

continuations

Contributions

- Comprehensive design
- Formal model
- Implementation

Contributions

- Comprehensive design

graphical intuition

- Formal model

- Implementation

Notation

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

Notation

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

Notation

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

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

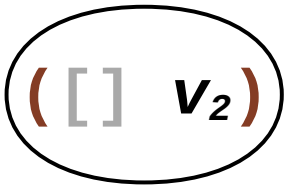
Notation

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

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

Notation

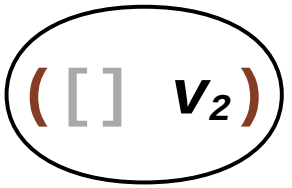
$(v_1\ (((\lambda\ (x)\ x)\ v_3)\ v_2))$

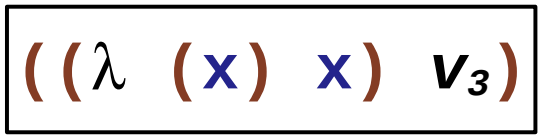


$((\lambda\ (x)\ x)\ v_3)$

Notation

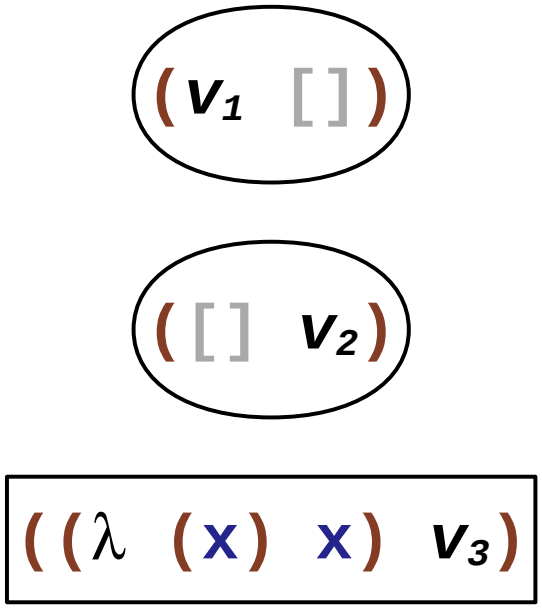
$(v_1\ (((\lambda\ (x)\ x)\ v_3)\ v_2))$





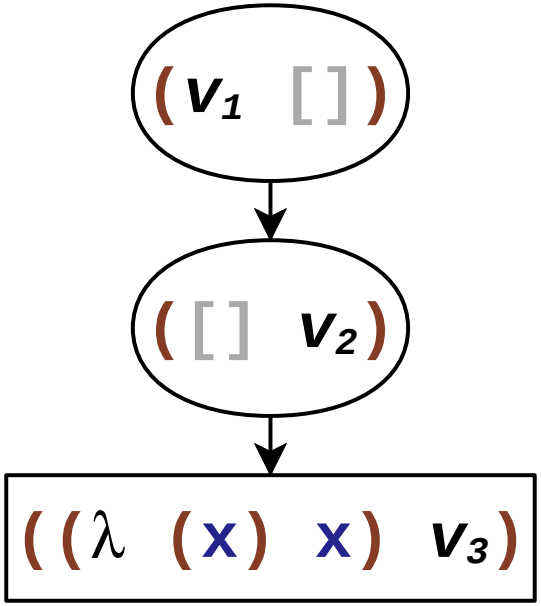
Notation

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

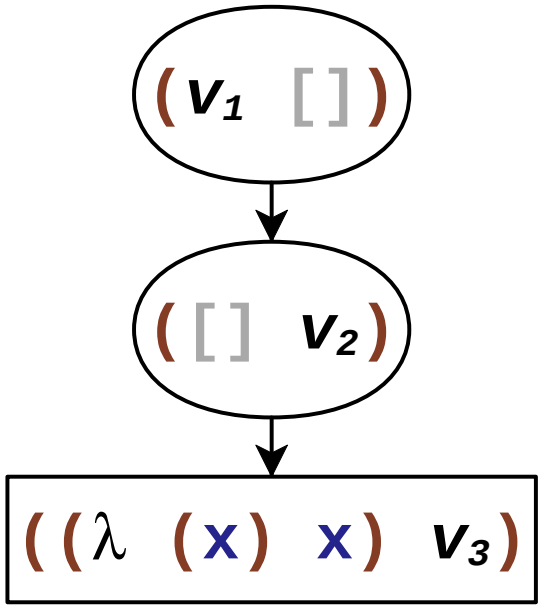


Notation

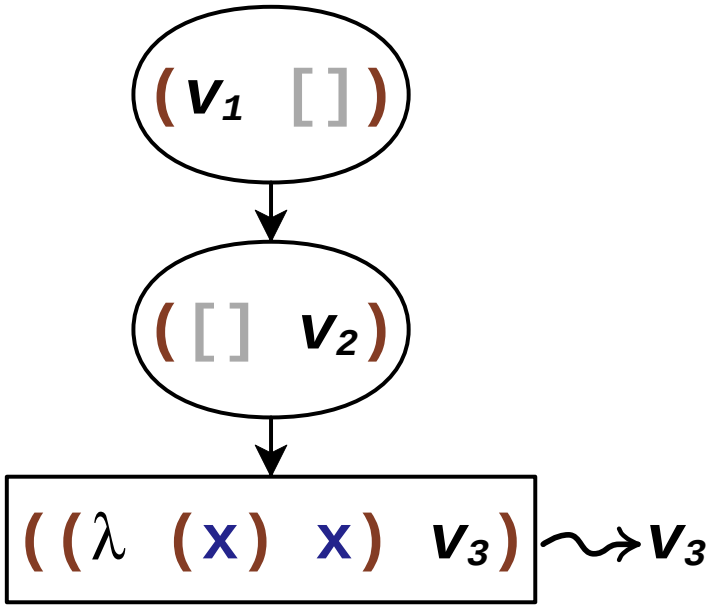
$(v_1\ (((\lambda\ (x)\ x)\ v_3)\ v_2)) =$



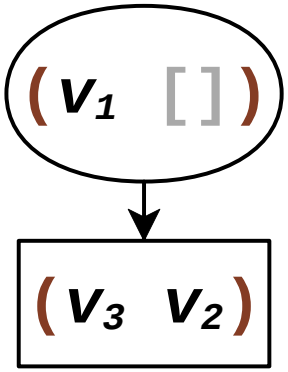
Reductions



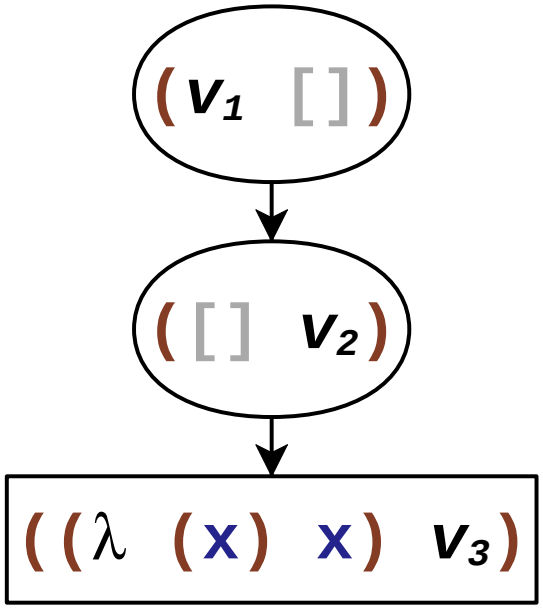
Reductions



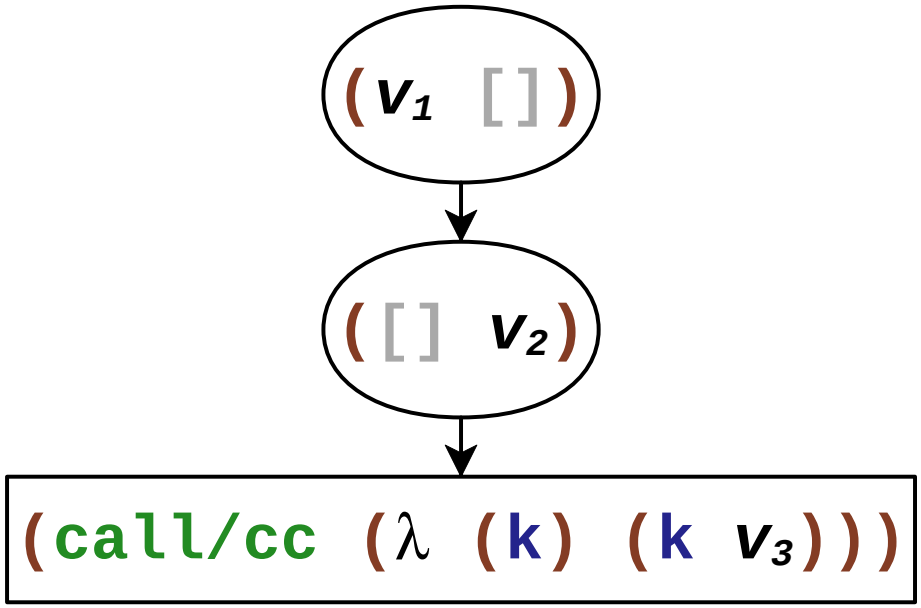
Reductions



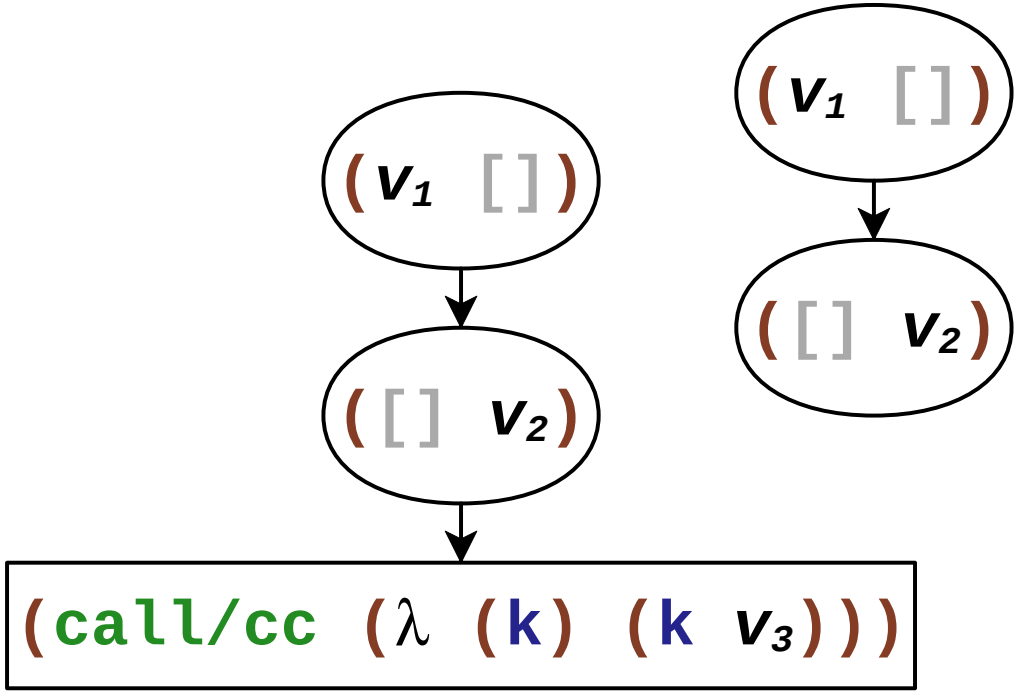
Reductions



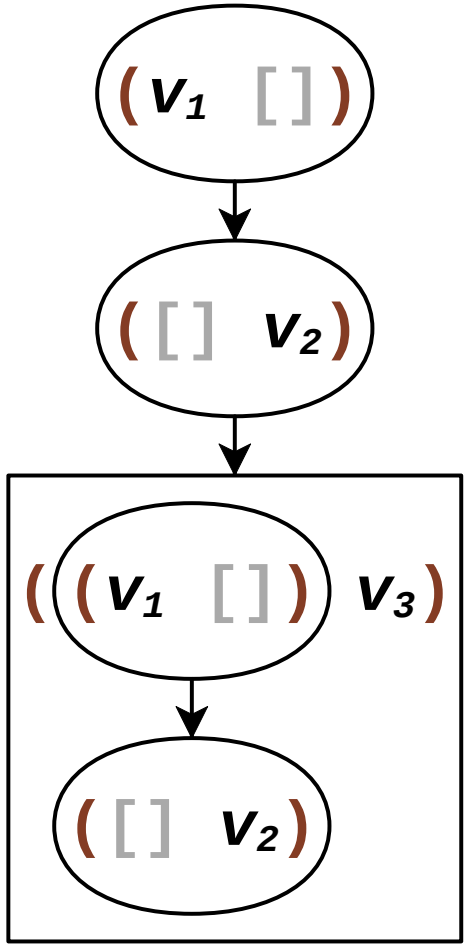
Continuations



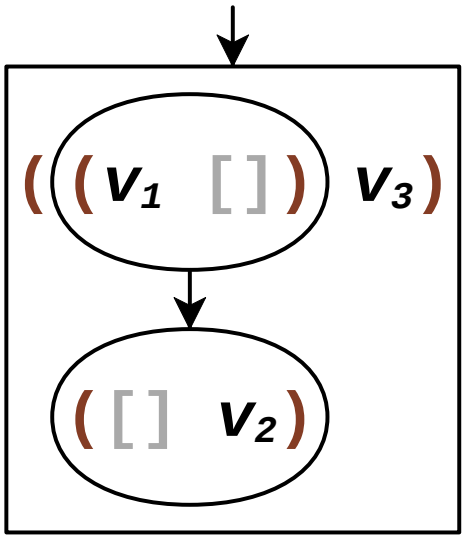
Continuations



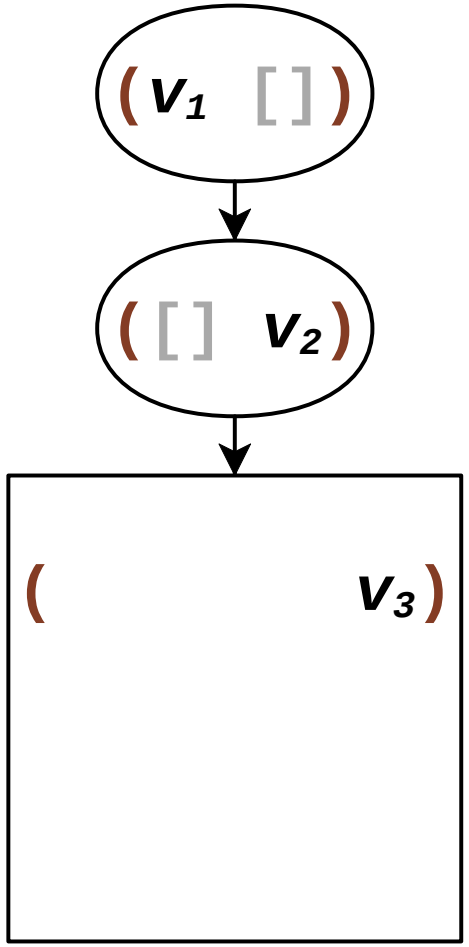
Continuations



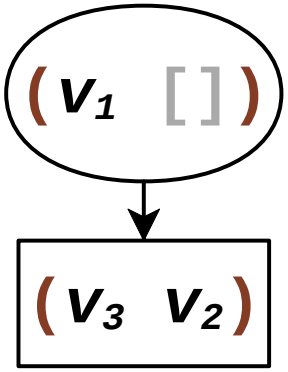
Continuations



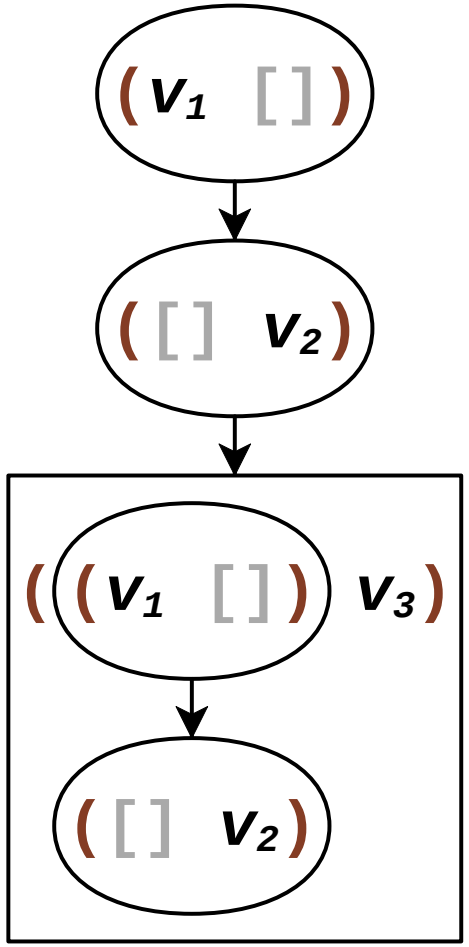
Continuations



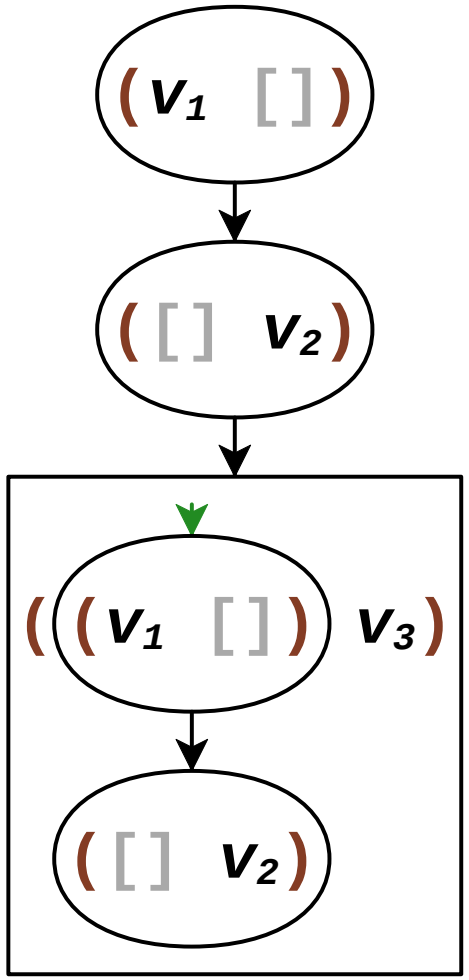
Continuations



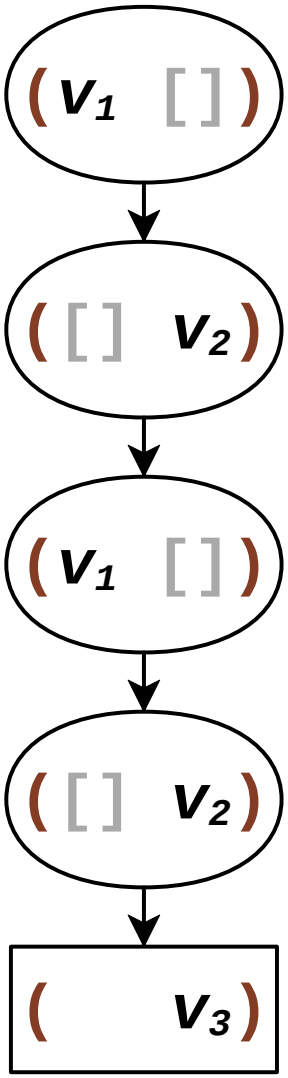
Continuations



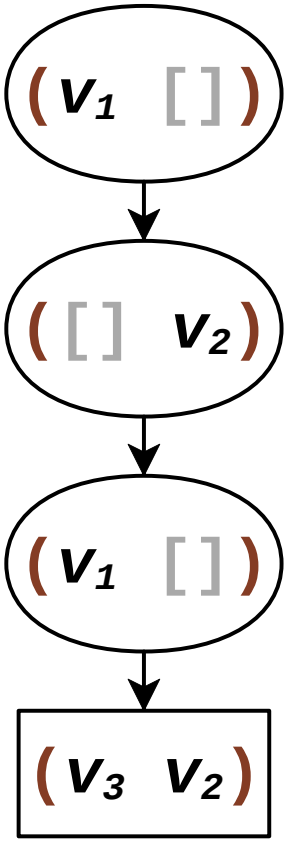
Composable Continuations



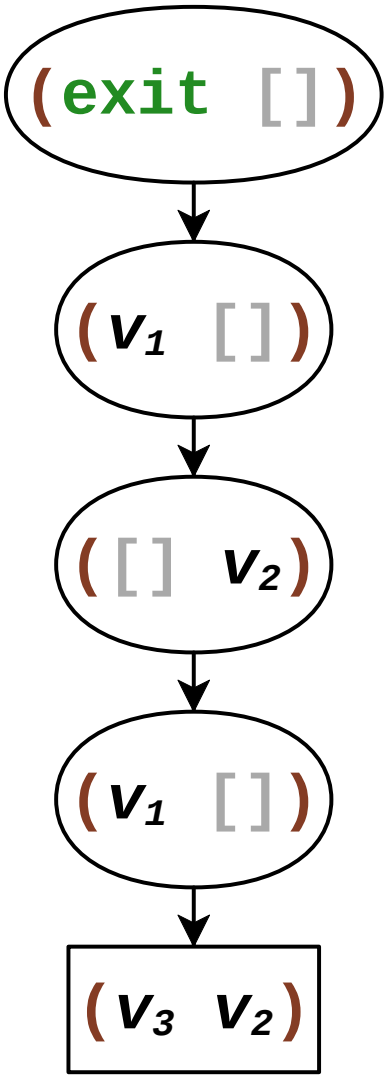
Composable Continuations



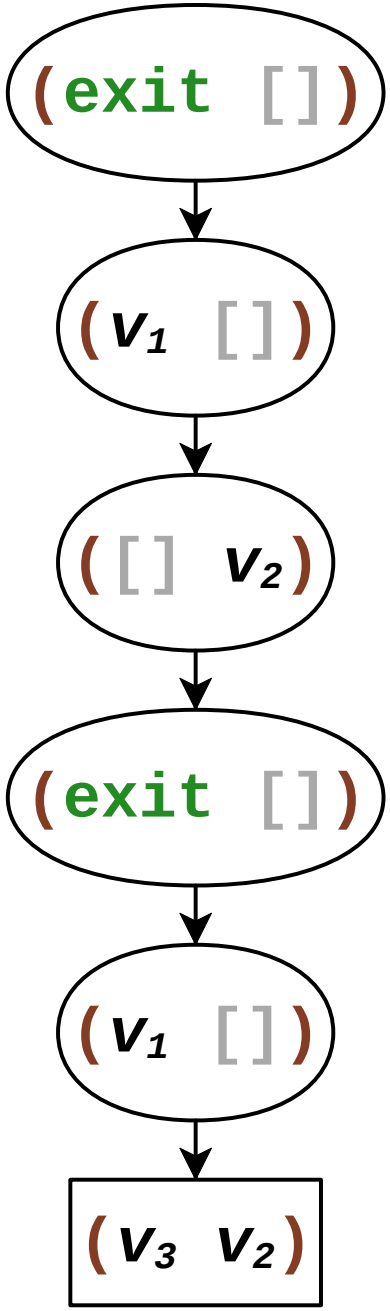
Composable Continuations



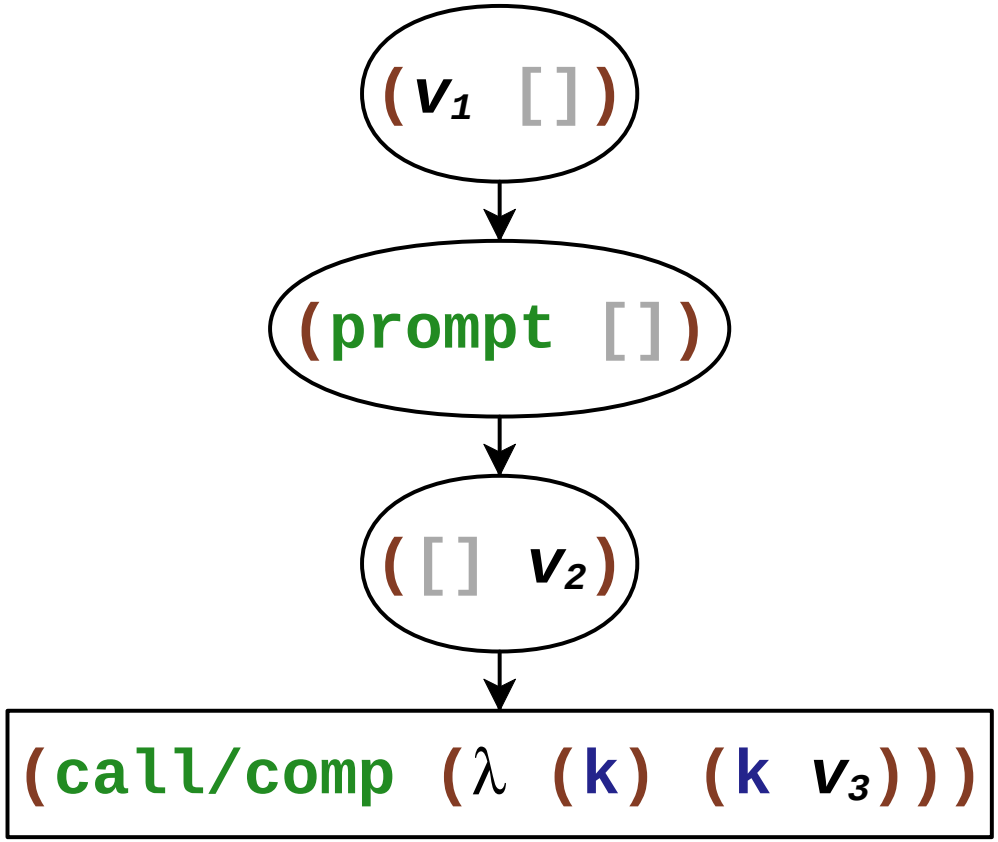
Composable Continuations



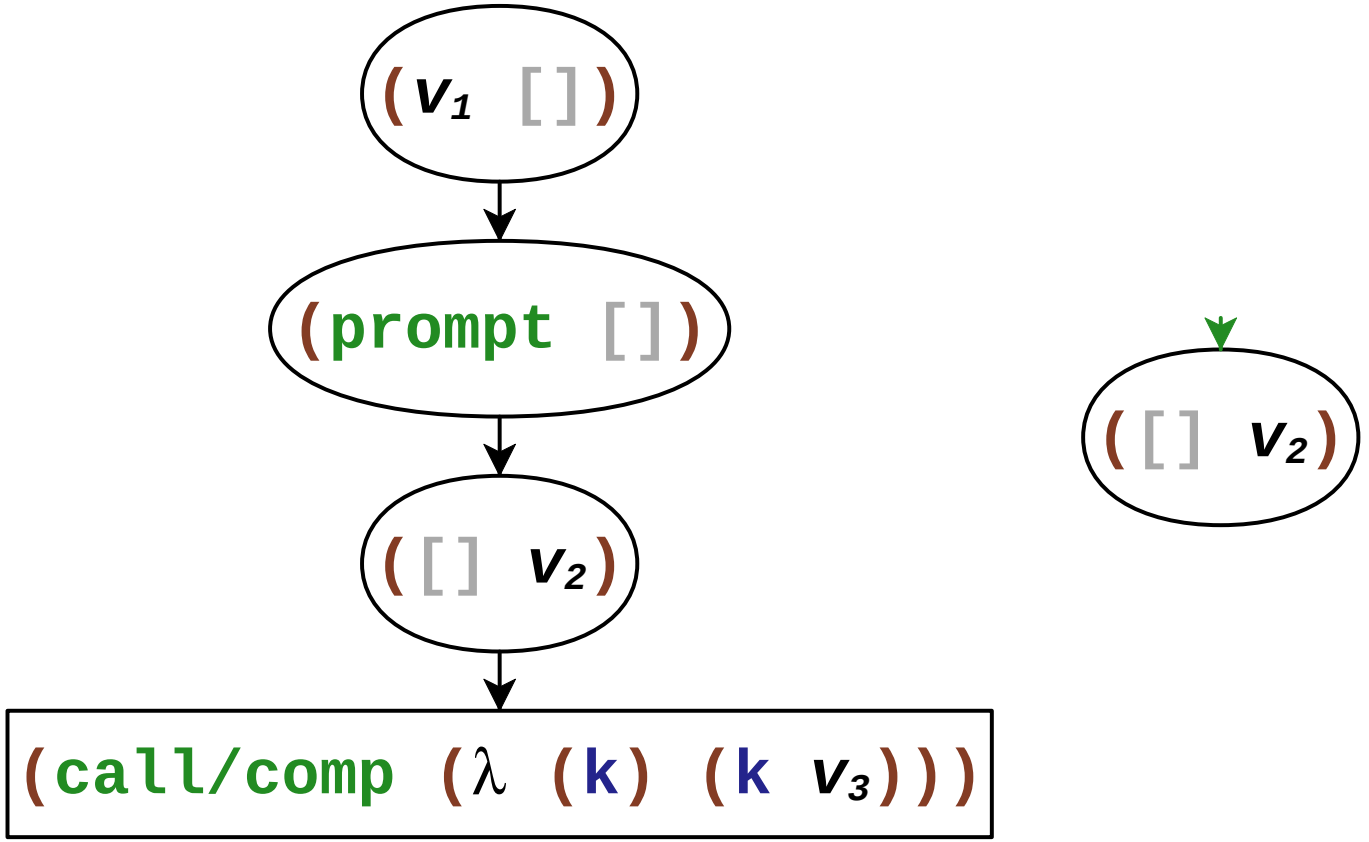
Composable Continuations



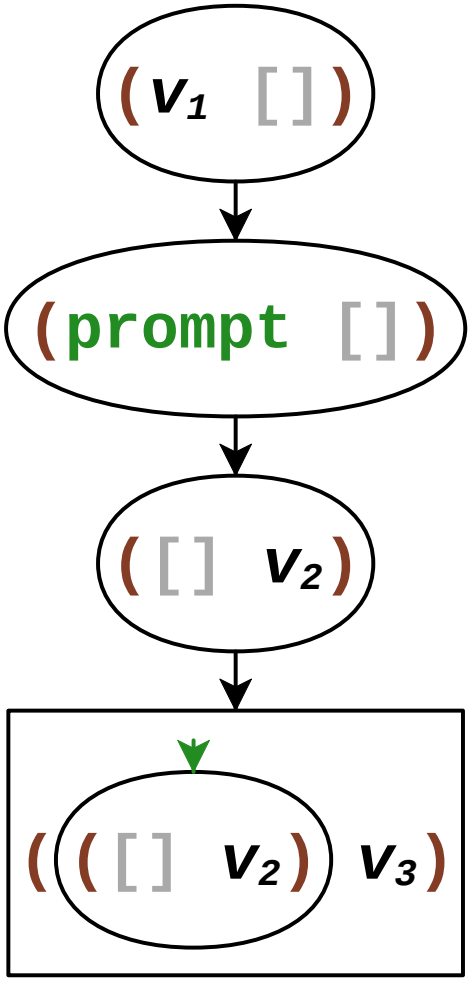
Delimited Capture



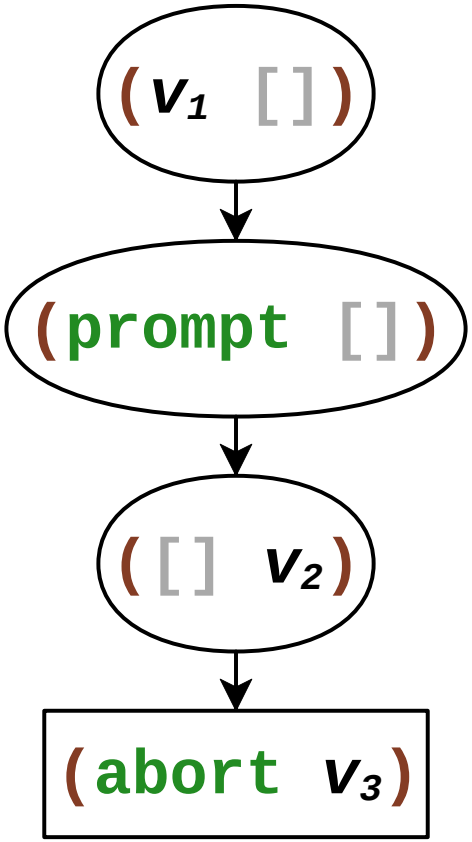
Delimited Capture



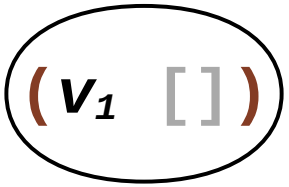
Delimited Capture



Delimited Abort



Delimited Abort



v_3

Delimited Abort

Splitting Capture and Abort

- **`call/comp`** : capture current continuation
- **`abort`** : abort current continuation

- $\mathcal{F}$: capture and abort current continuation

Splitting Capture and Abort

- **`call/comp`** : capture current continuation
- **`abort`** : abort current continuation

- $\mathcal{F}$: capture and abort current continuation

```
F = (λ (f) (call/comp
             (λ (k) (abort (λ () (f k))))))
```

Splitting Capture and Abort

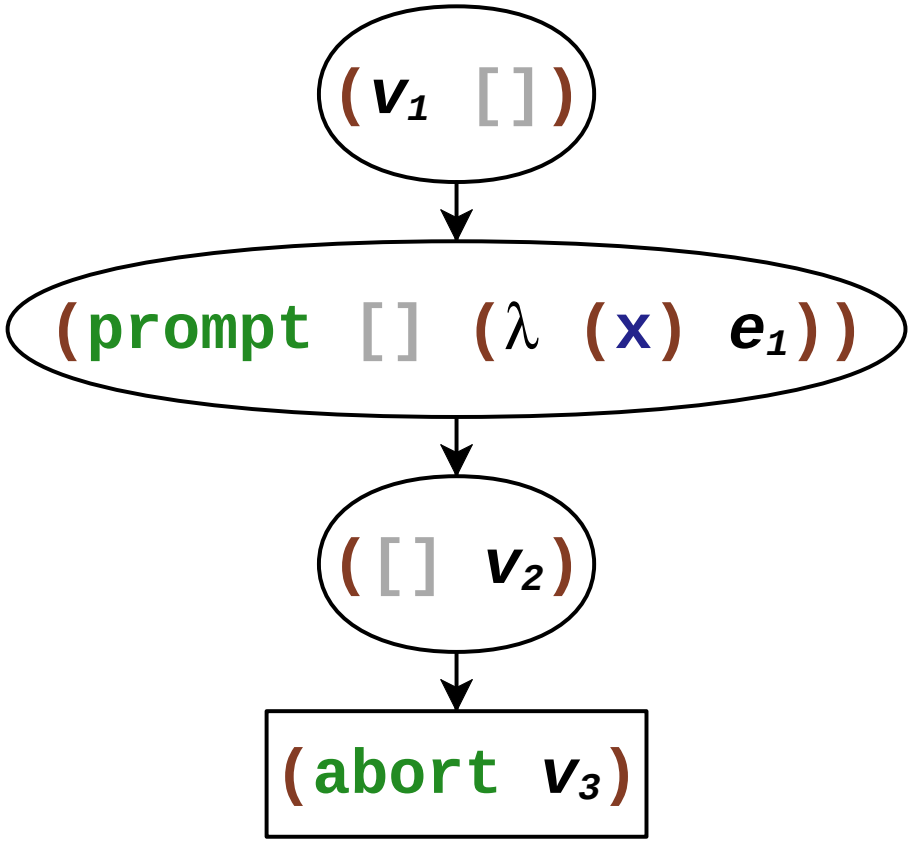
- **`call/comp`** : capture current continuation
- **`abort`** : abort current continuation

- $\mathcal{F}$: capture and abort current continuation

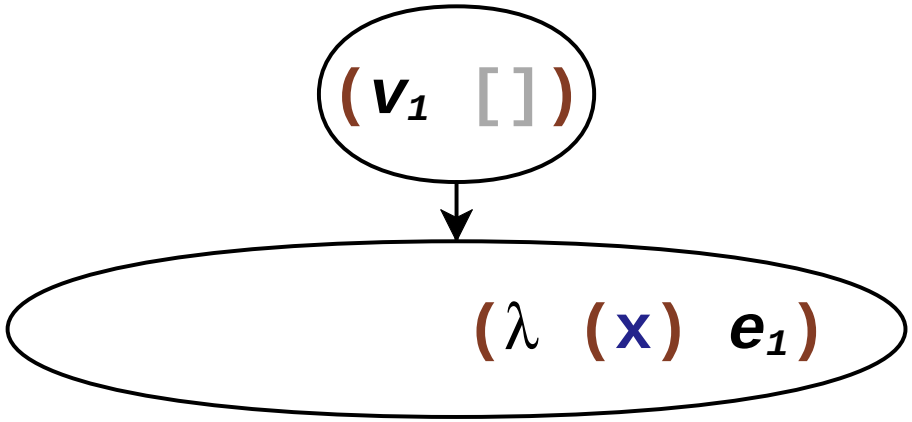
```
F = (λ (f) (call/comp
             (λ (k) (abort (λ () (f k))))))
```

delay `(f k)`
until after abort

Prompt with Handler

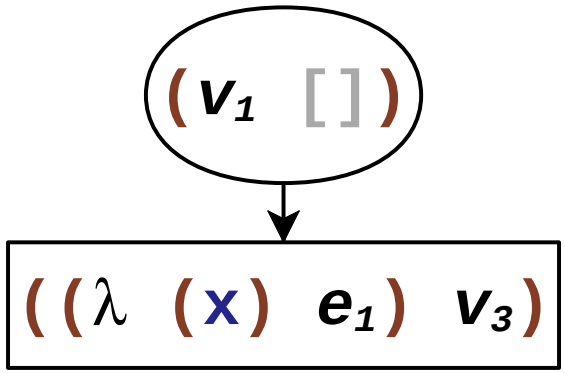


Prompt with Handler

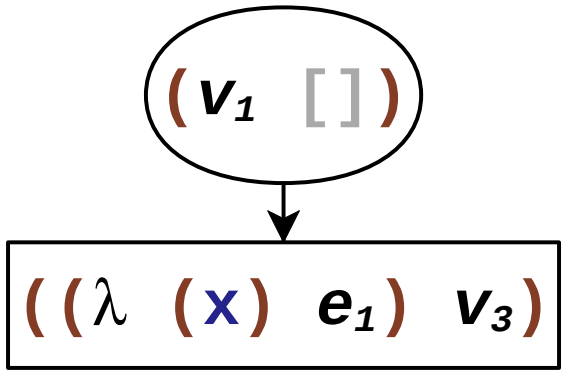


v_3

Prompt with Handler



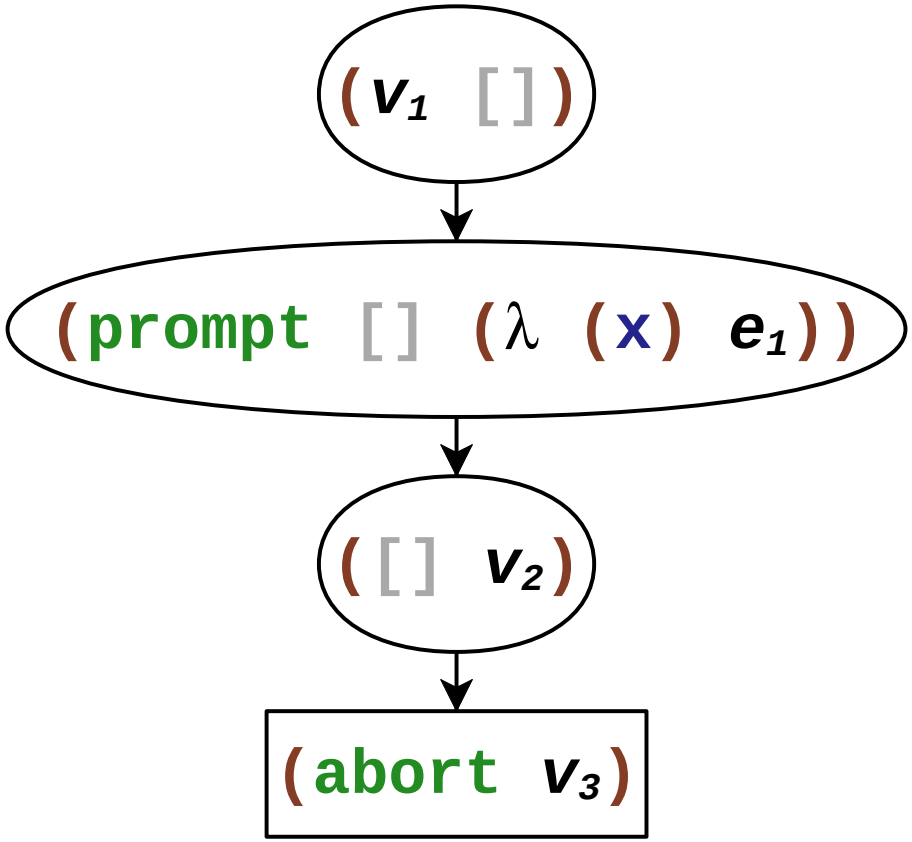
Prompt with Handler



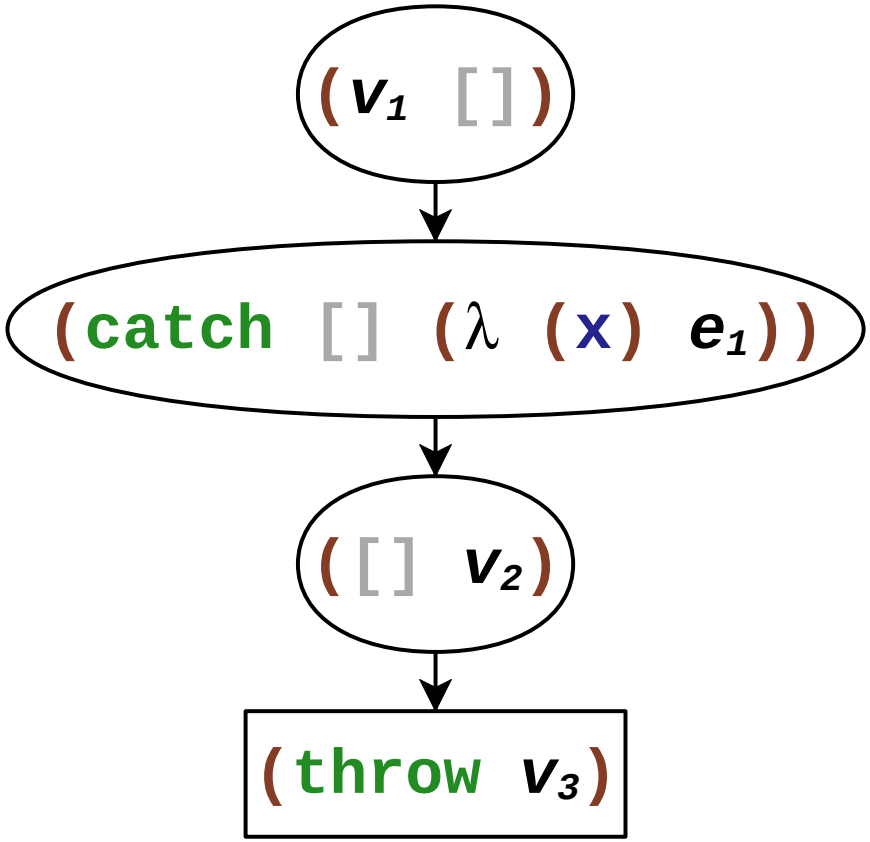

```
(prompt e1) = (prompt e1 (λ (t) (t)))

F = (λ (f) (call/comp
             (λ (k) (abort (λ () (f k))))))
```

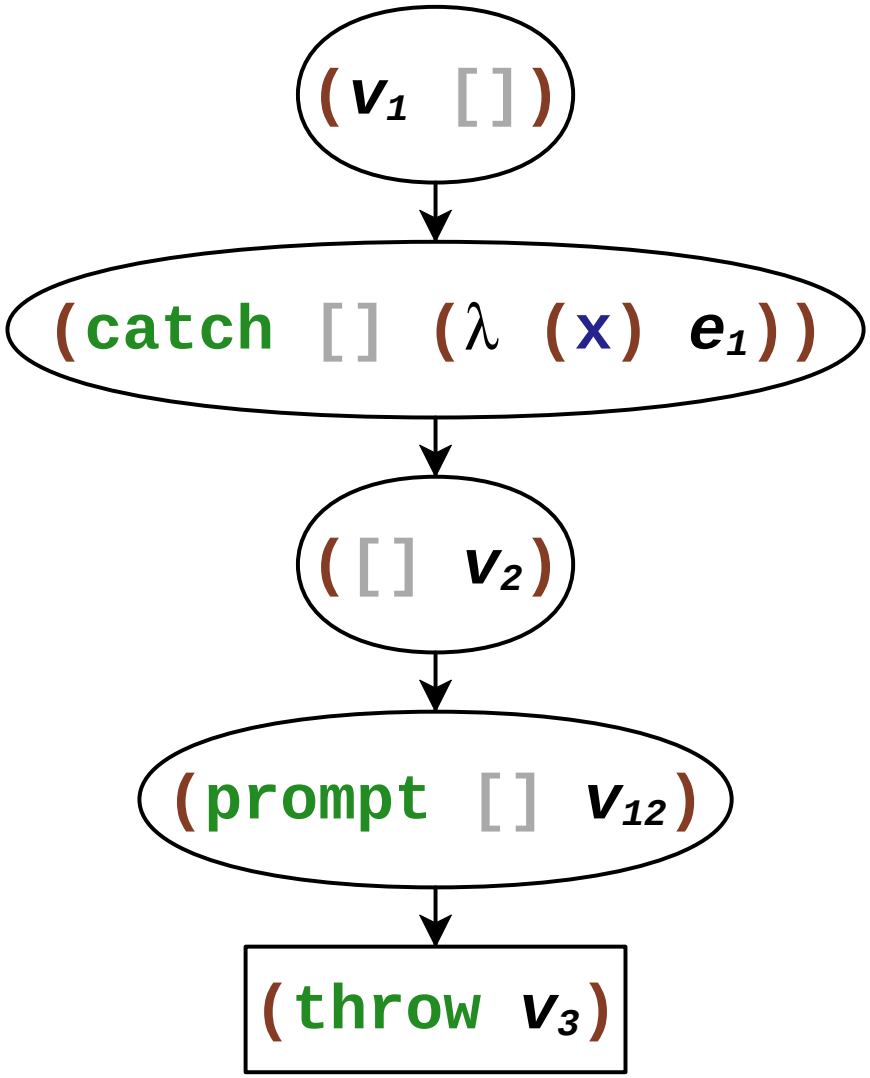
Prompt with Handler



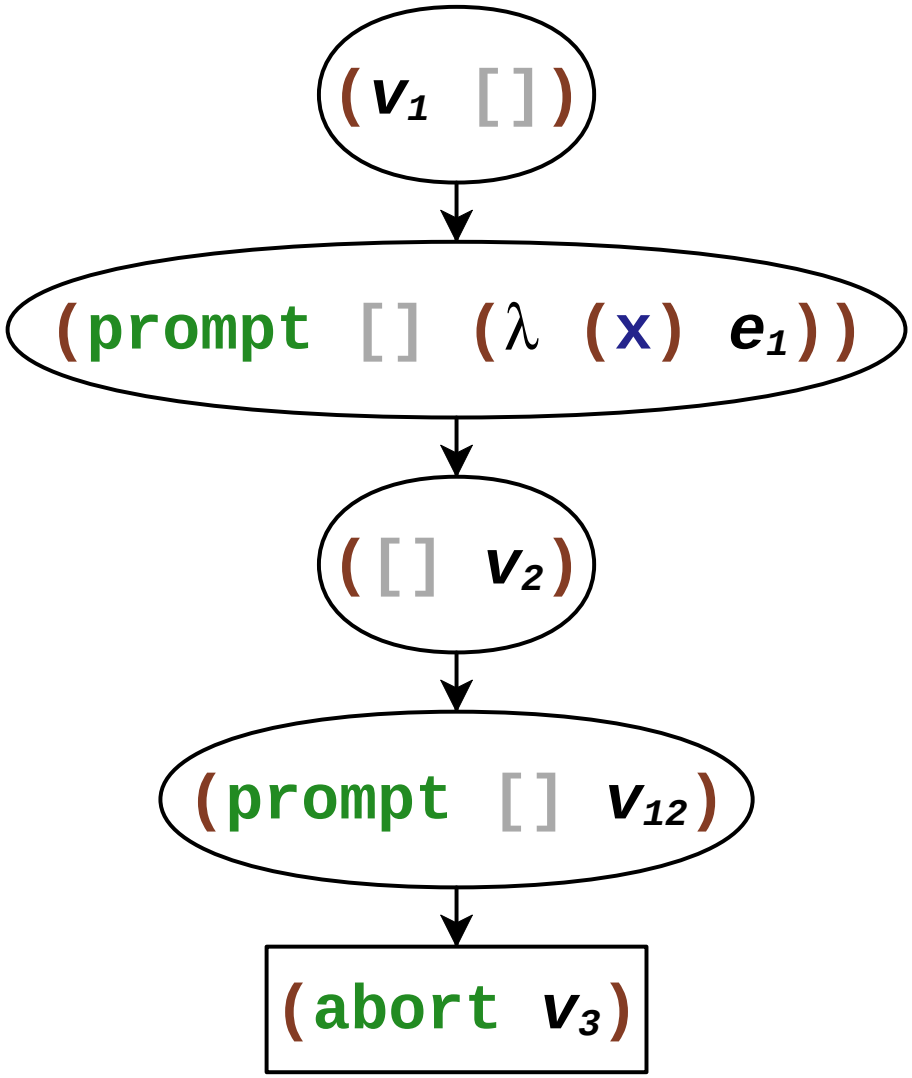
Catch and Throw



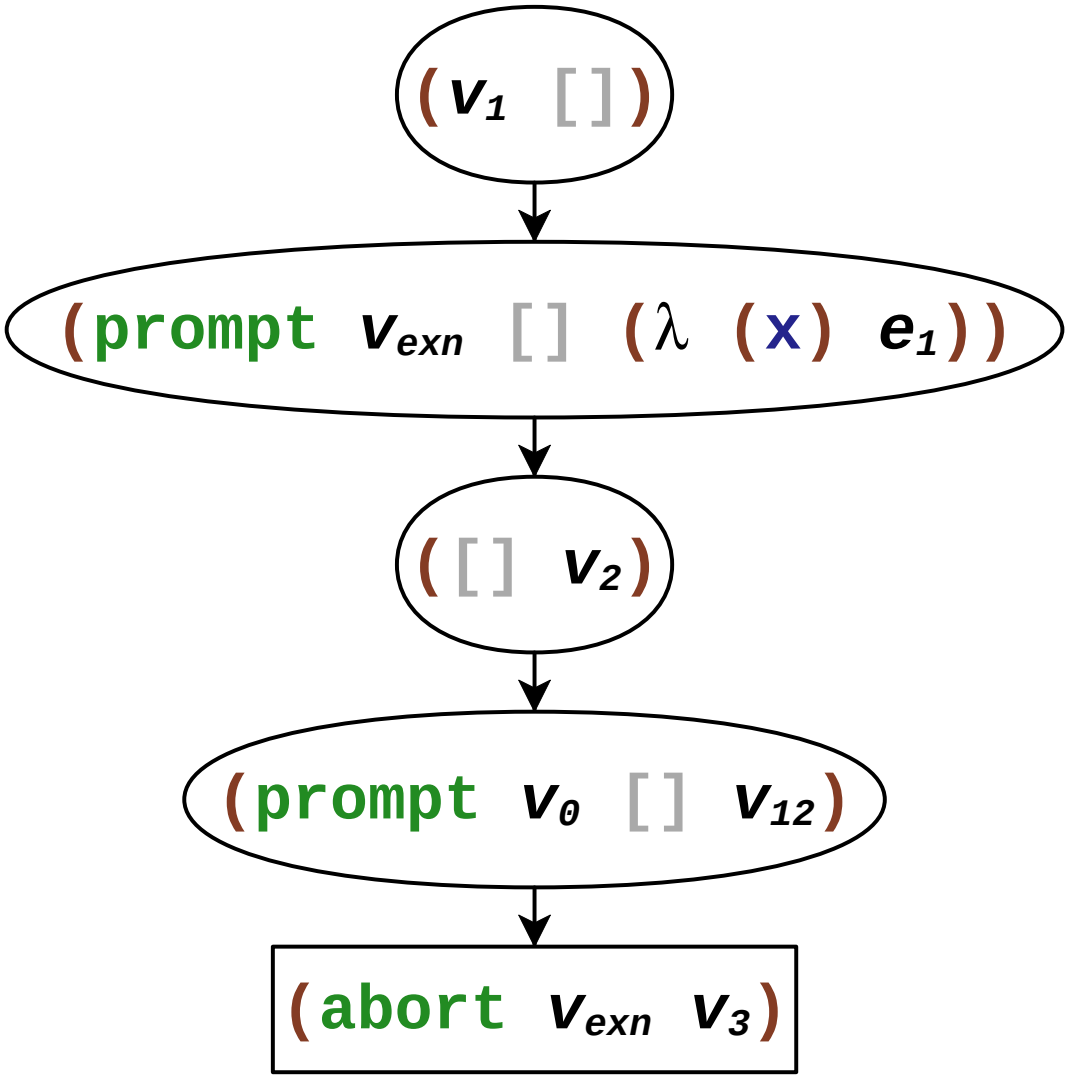
Catch and Throw



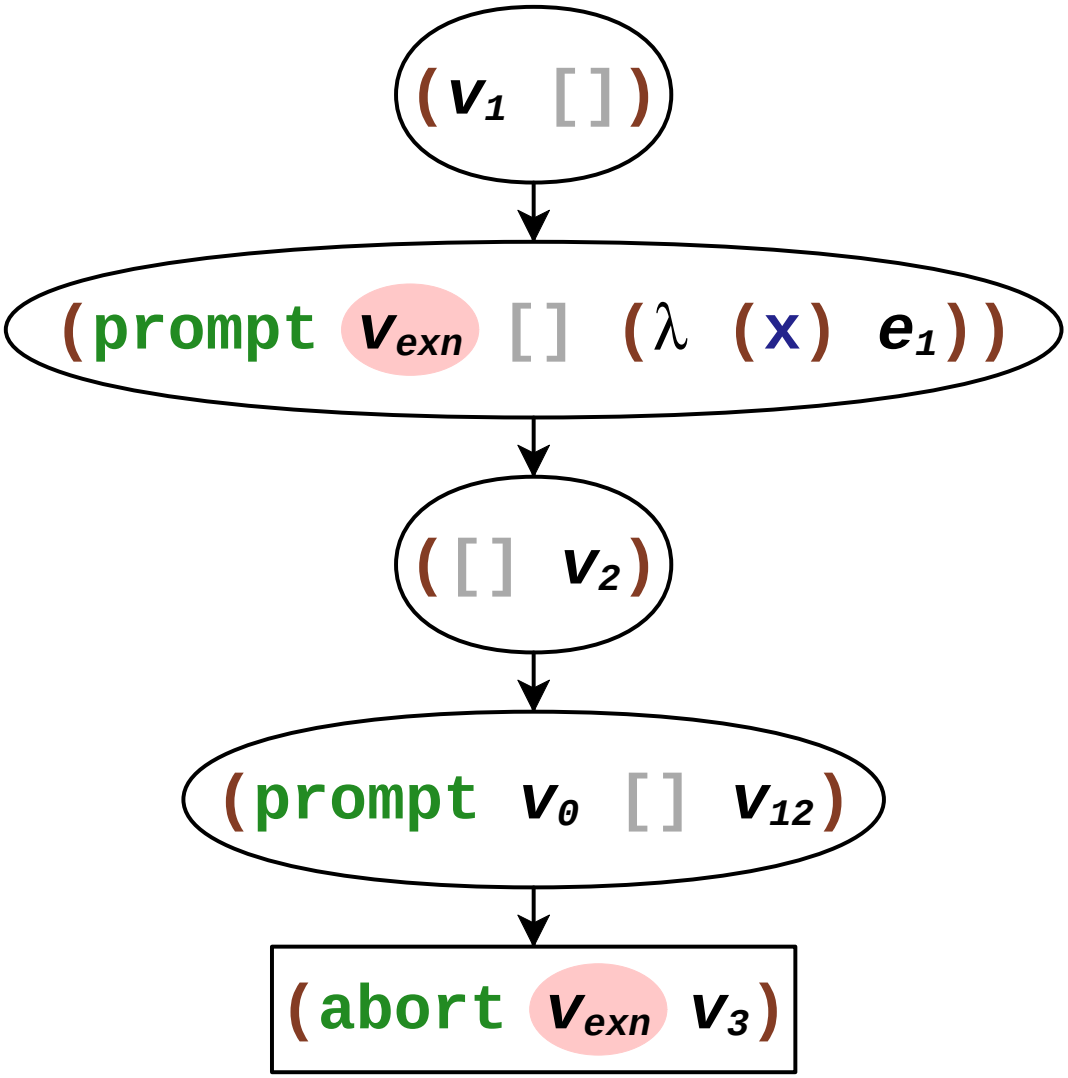
Catch and Throw



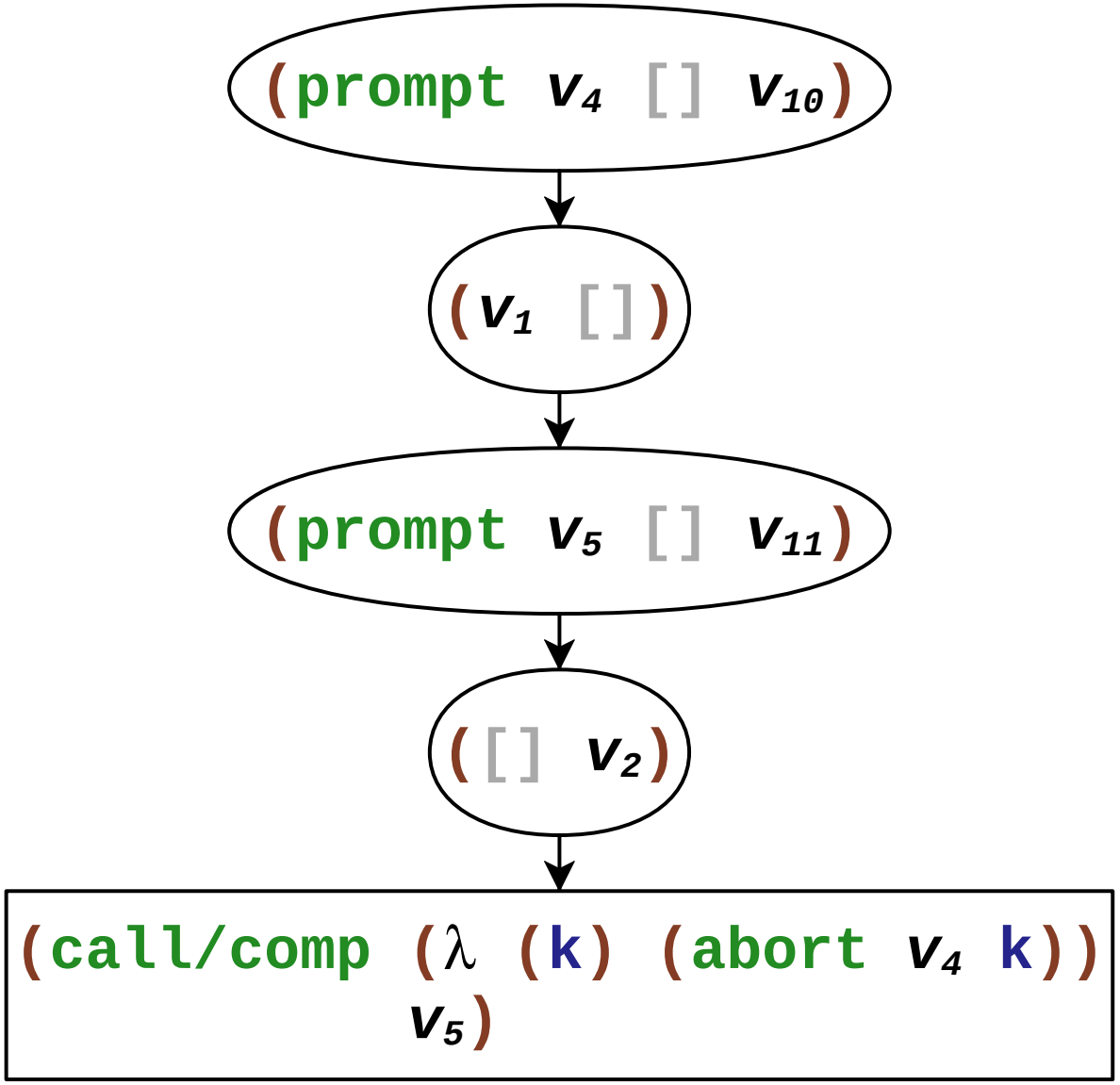
Catch and Throw



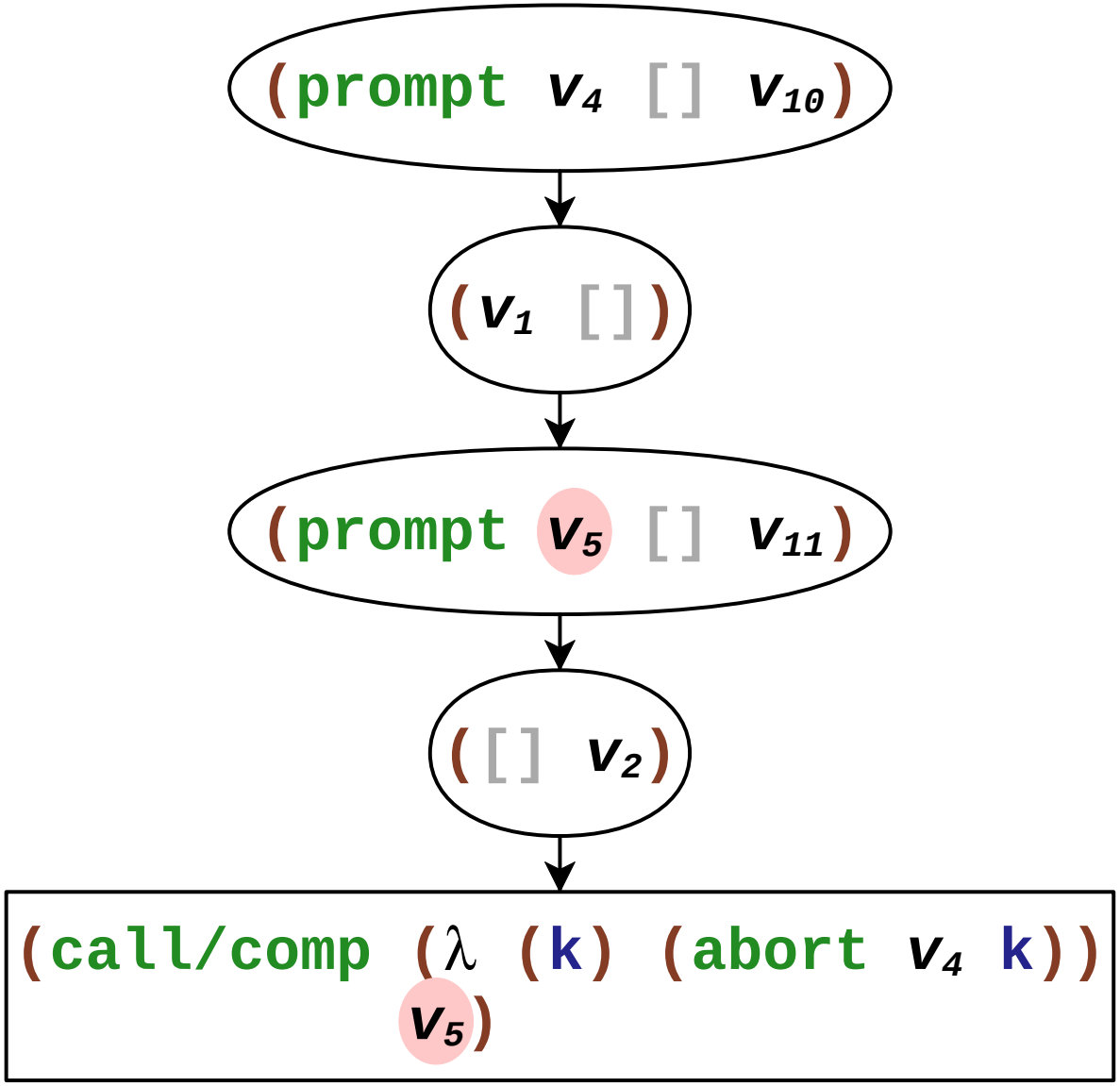
Tagged Prompts



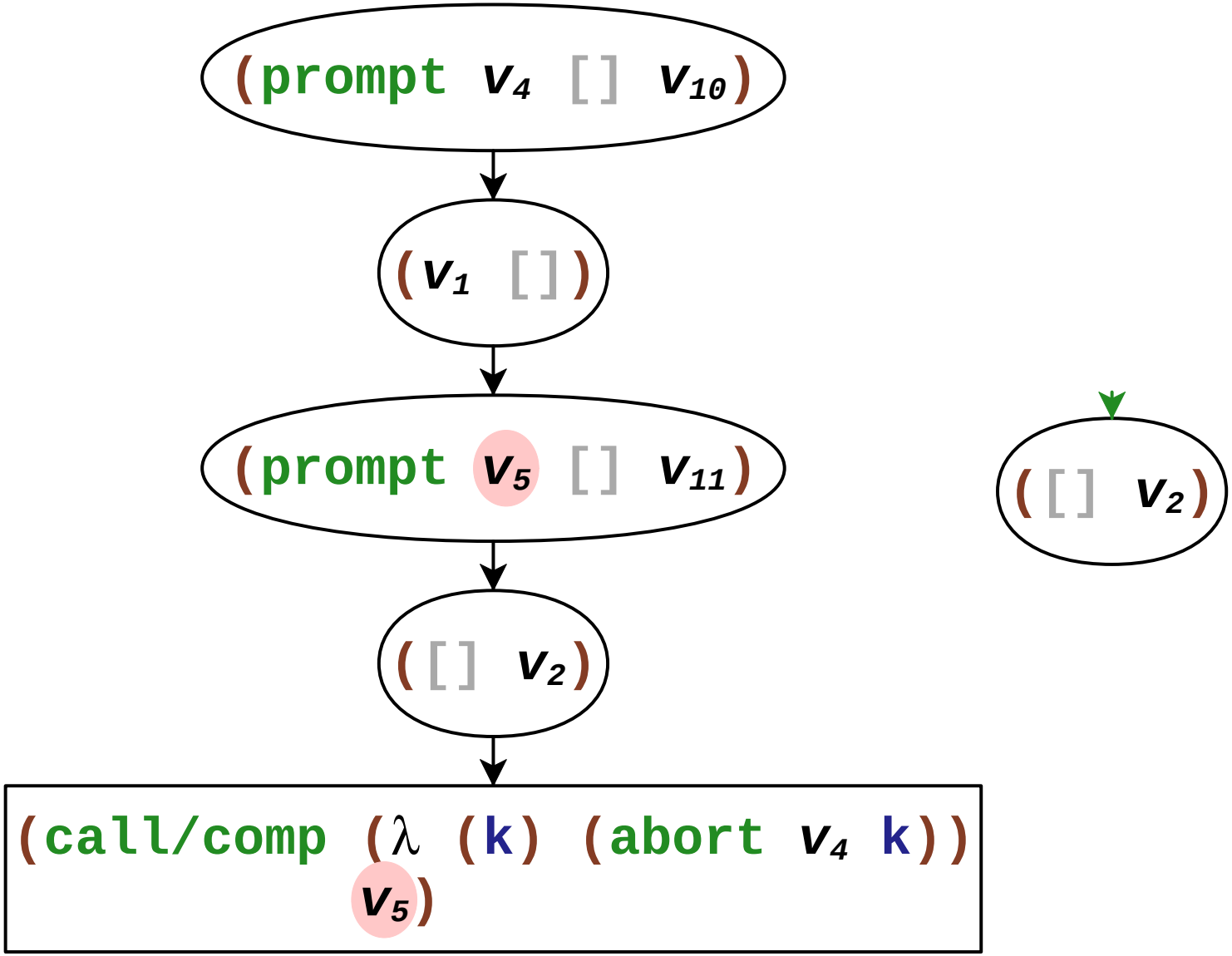
Tagged Prompts



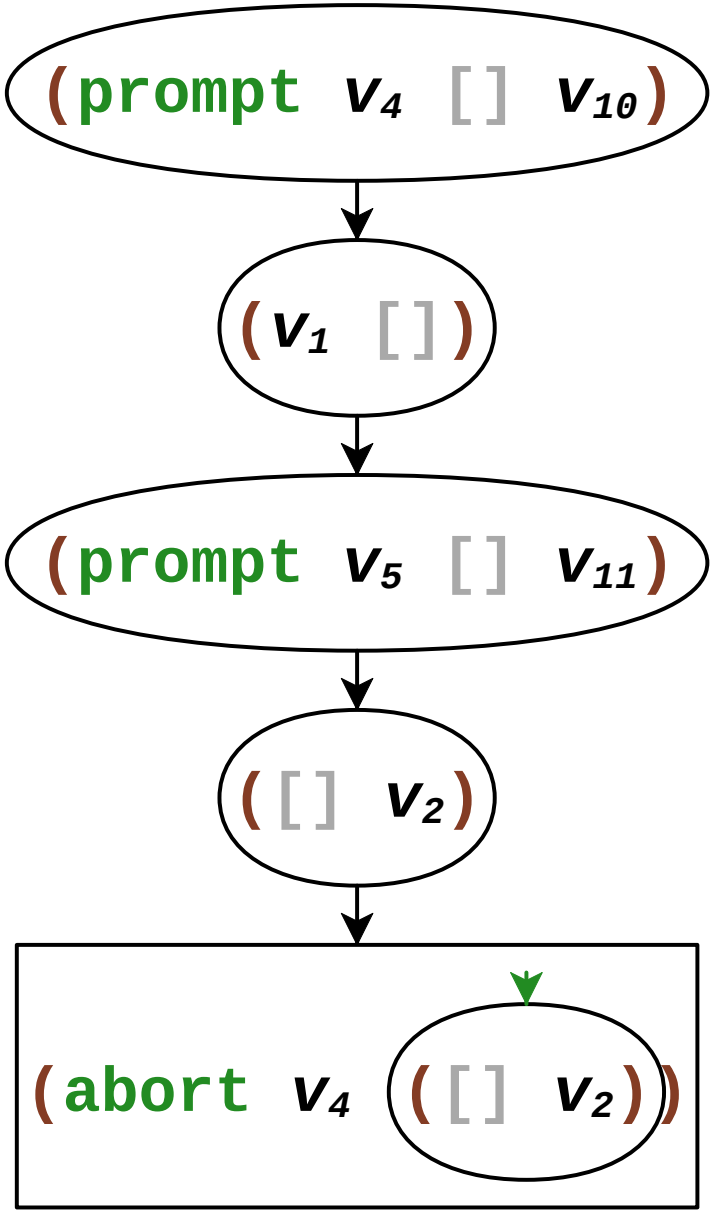
Tagged Prompts



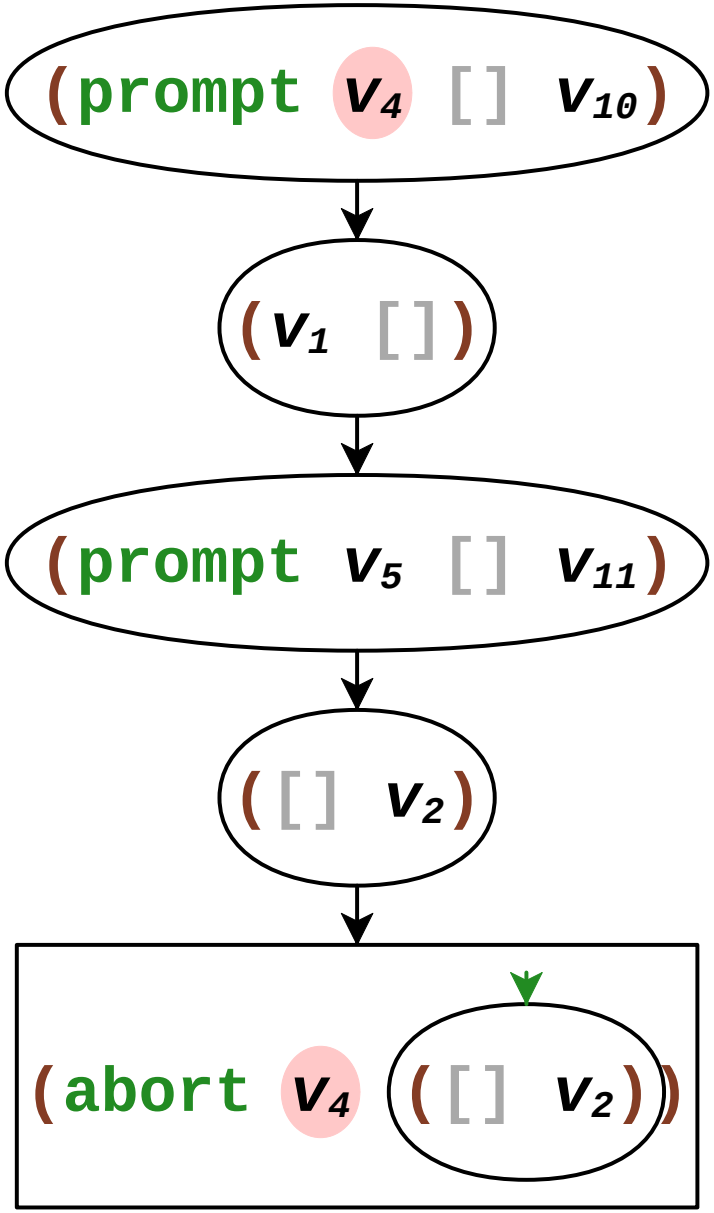
Tagged Prompts



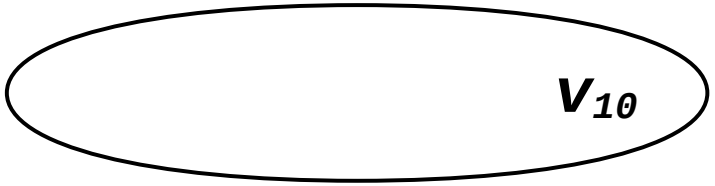
Tagged Prompts

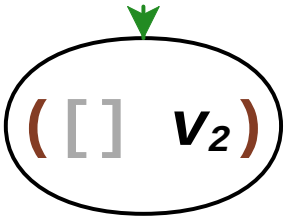


Tagged Prompts

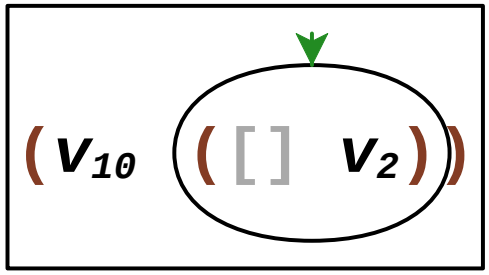


Tagged Prompts





Tagged Prompts



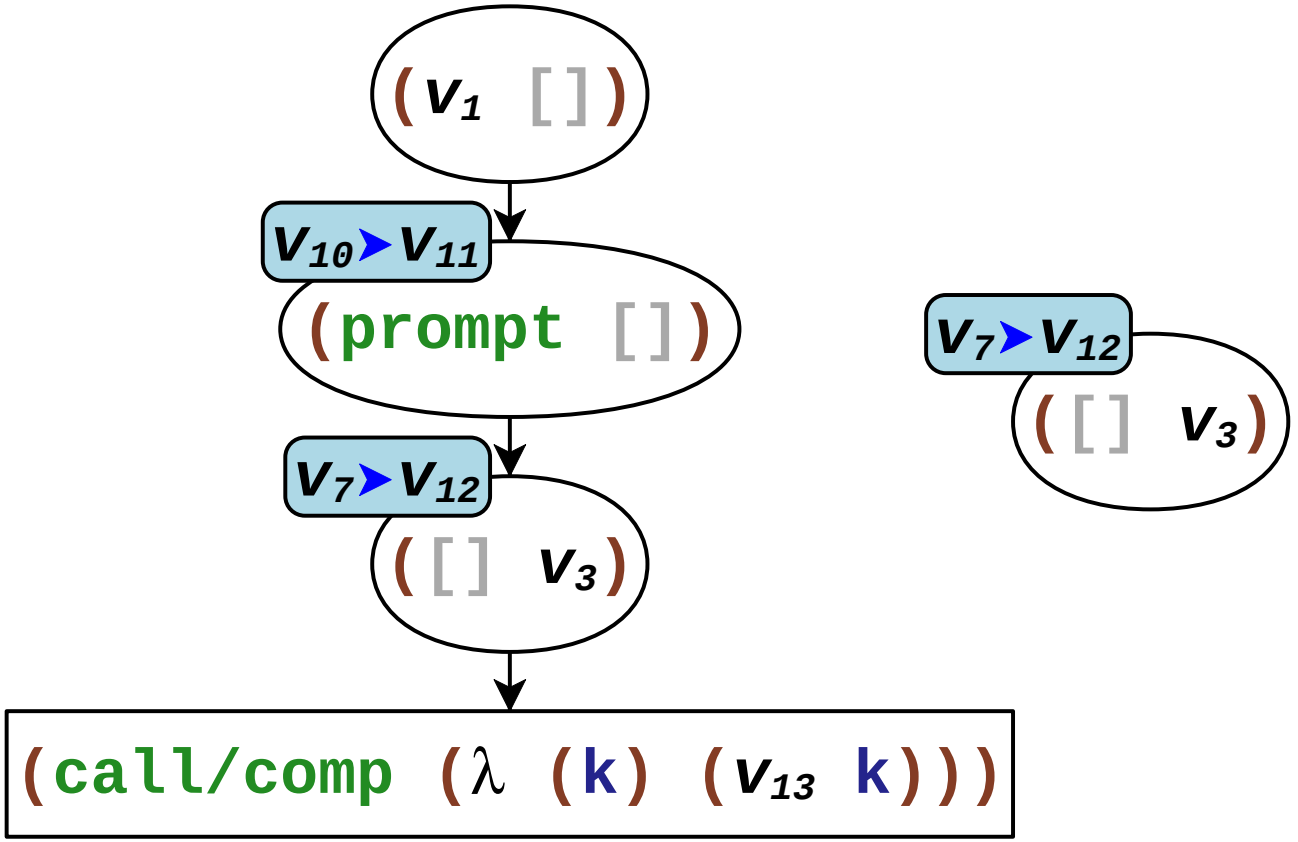
Continuations Summary

- `prompt` with handler and tag
- `abort` with tag
- `call/comp` with tag
- continuation composition

- plain `call/cc` uses default tag

[Sitaram PLDI'93]

Continuation Marks



Dynamic Binding Summary

- `call/cm` to add marks
- `current-marks` to get marks, up to a tag

- capture marks in `call/comp`
- splice marks in continuation composition

Side Effects and Control

```
(define (with-resource work-thunk)
  (begin
    (grab-resource)
    (work-thunk)
    (release-with-resource)))
```

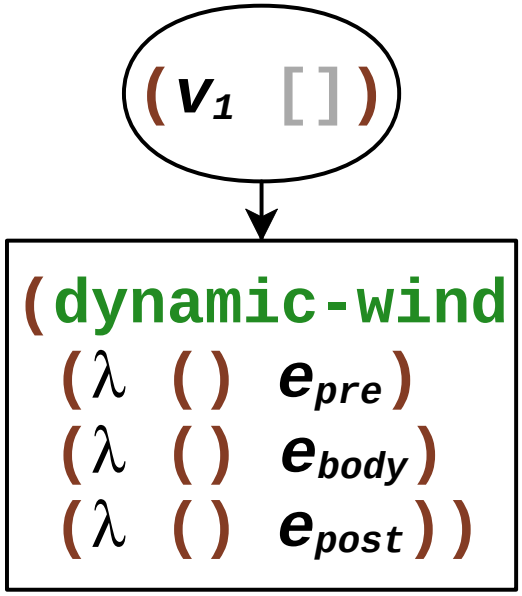
Dynamic Wind

```
(define (with-resource work-thunk)
  (dynamic-wind
   (λ () (grab-resource))
   (λ () (work-thunk))
   (λ () (release-with-resource))))
```

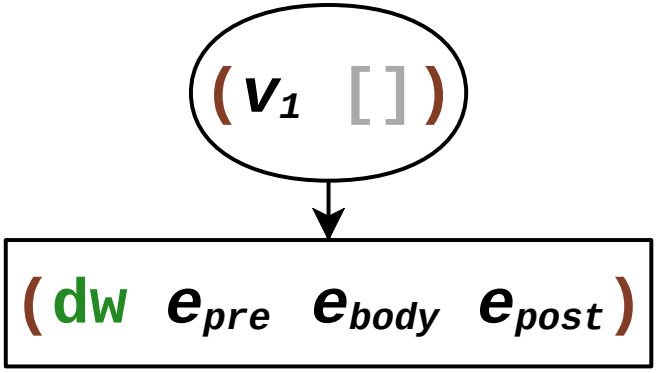
Dynamic Wind

```
(dynamic-wind
 (λ () e_pre)
 (λ () e_body)
 (λ () e_post))
```

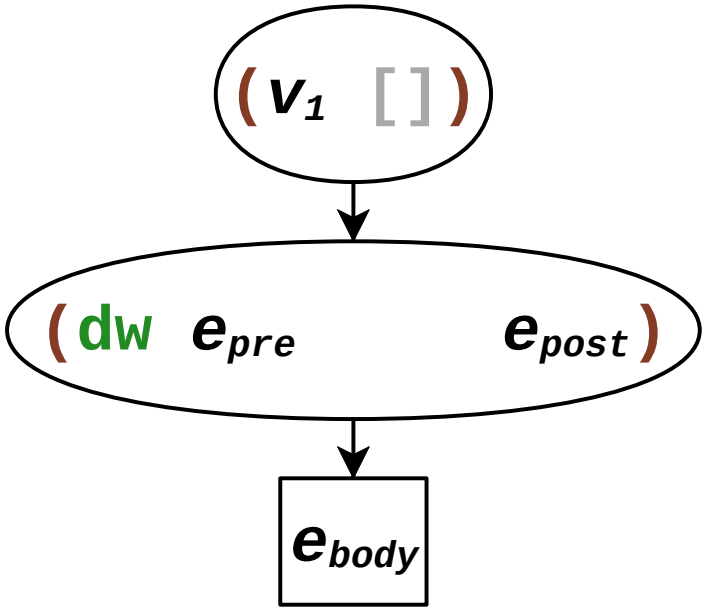
Dynamic Wind



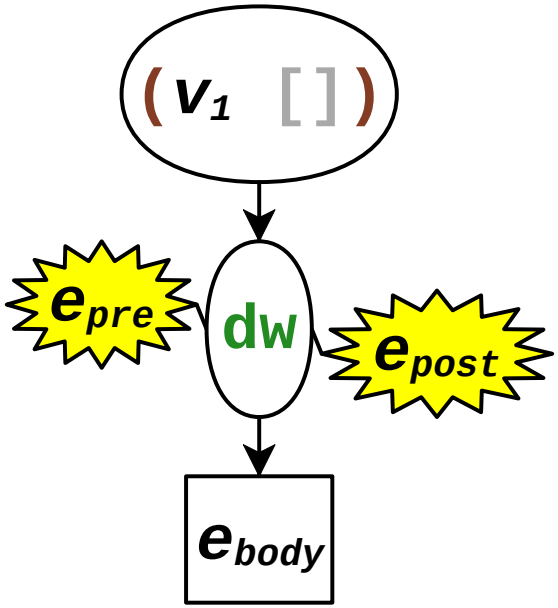
Dynamic Wind



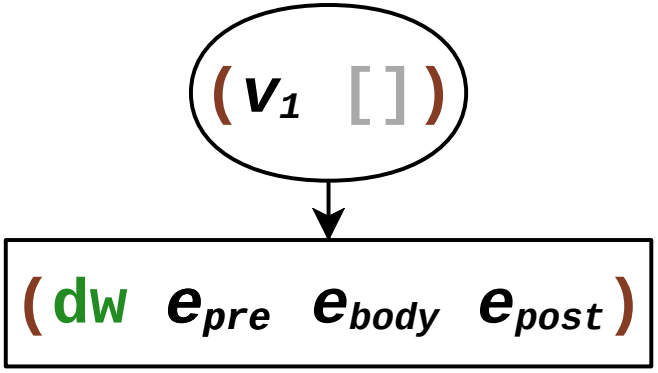
Dynamic Wind



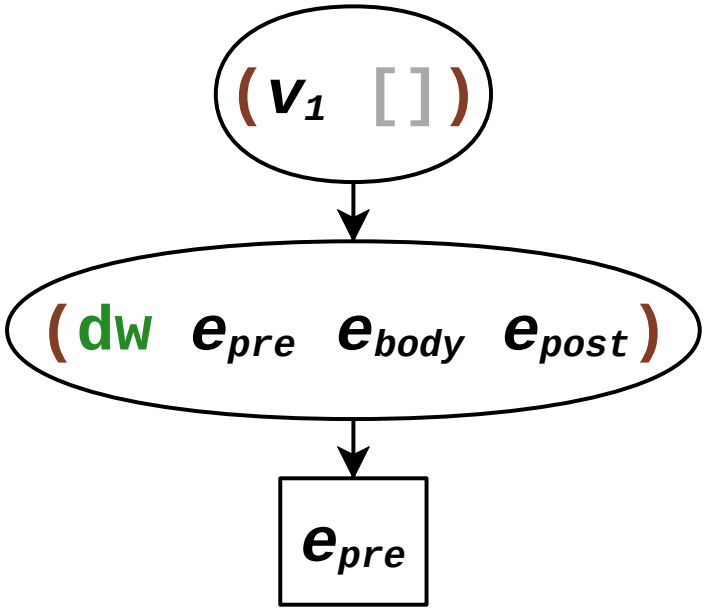
Dynamic Wind



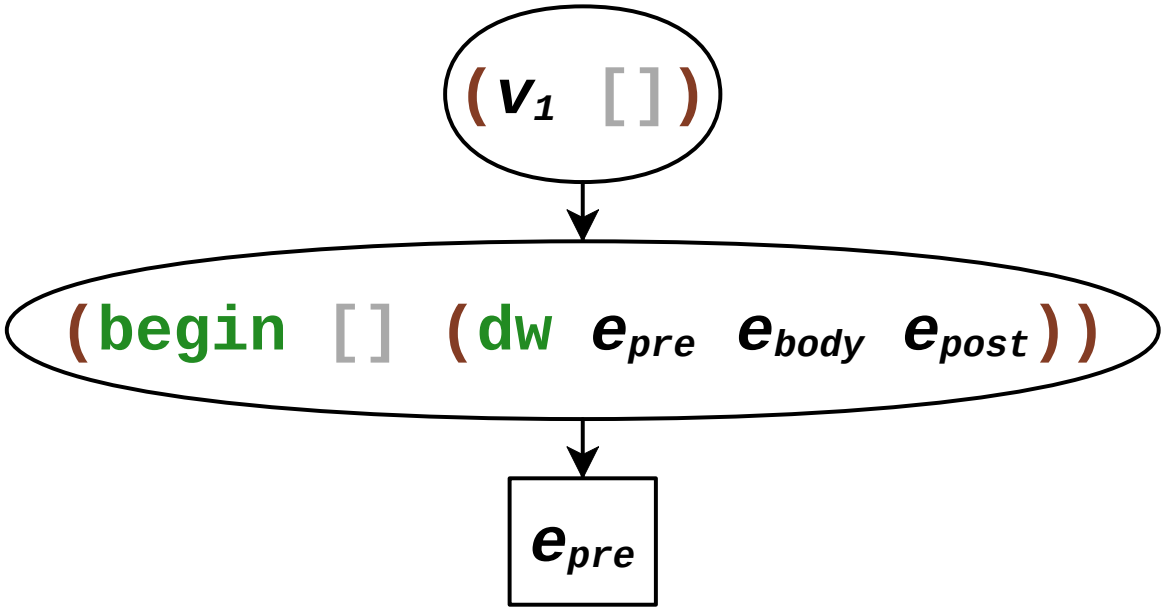
Dynamic Wind Evaluation



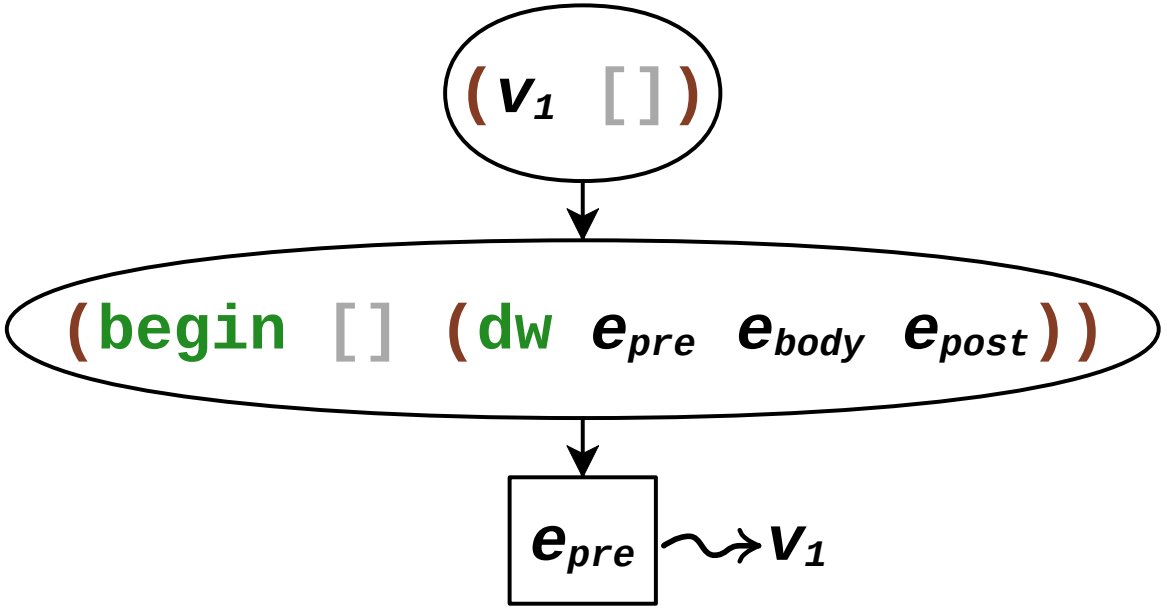
Dynamic Wind Evaluation



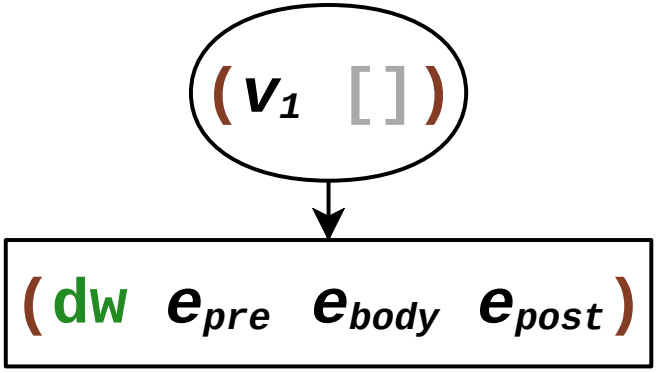
Dynamic Wind Evaluation



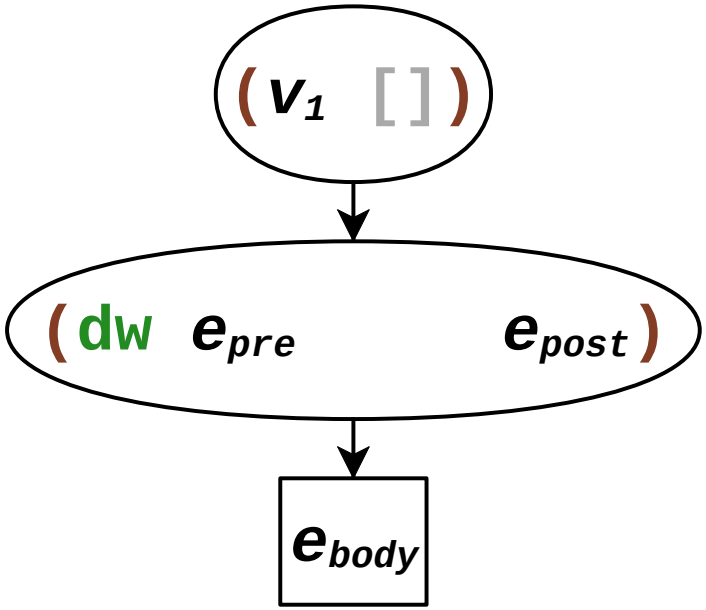
Dynamic Wind Evaluation



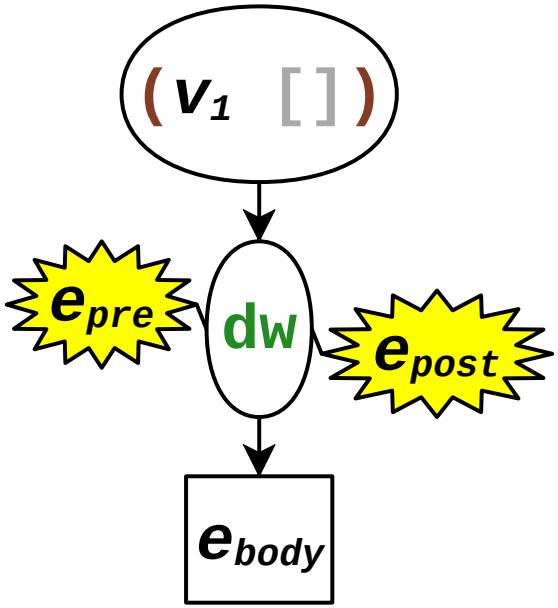
Dynamic Wind Evaluation



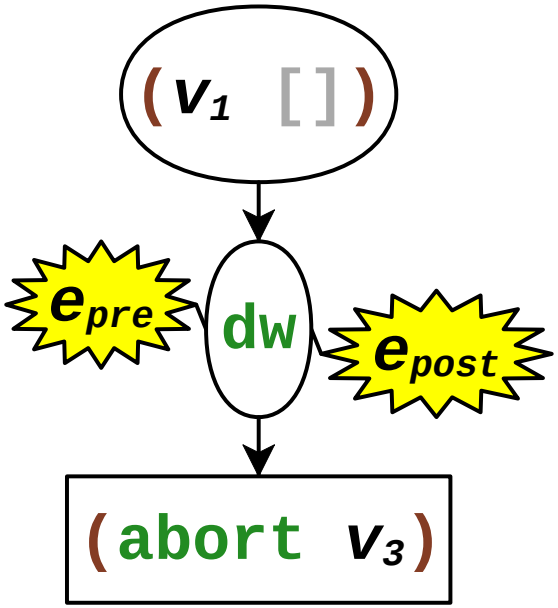
Dynamic Wind Evaluation



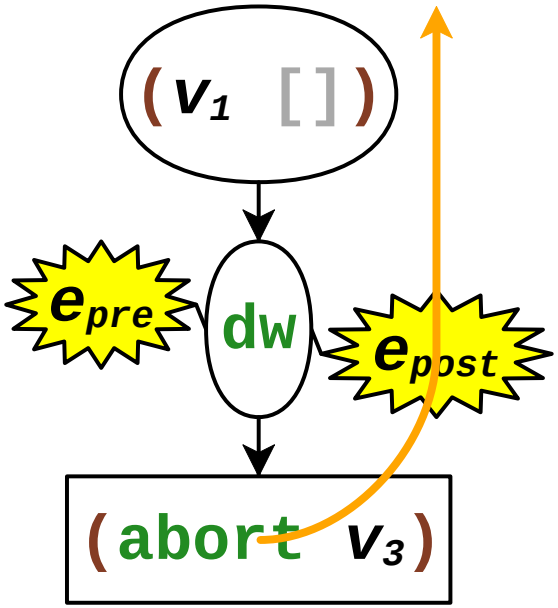
Dynamic Wind and Jumps



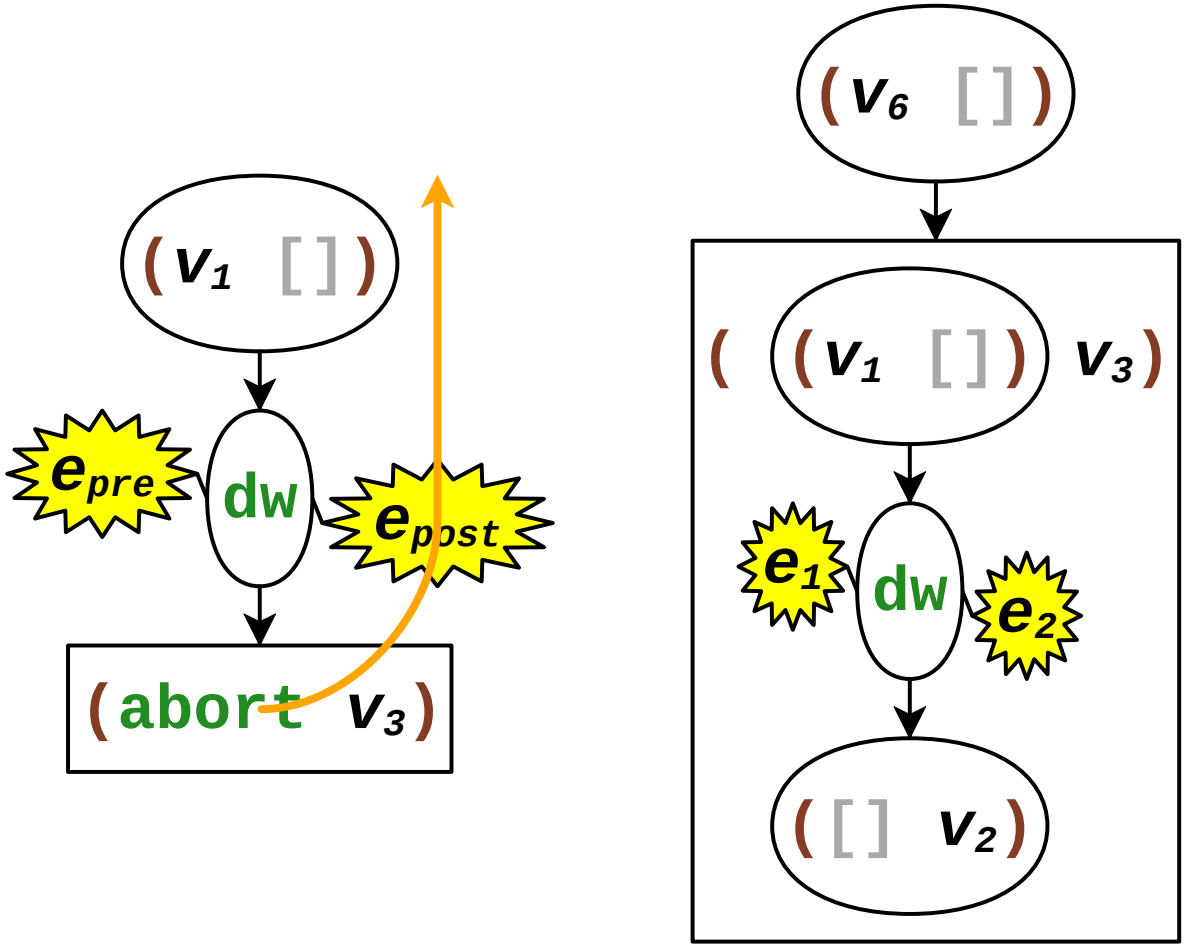
Dynamic Wind and Jumps



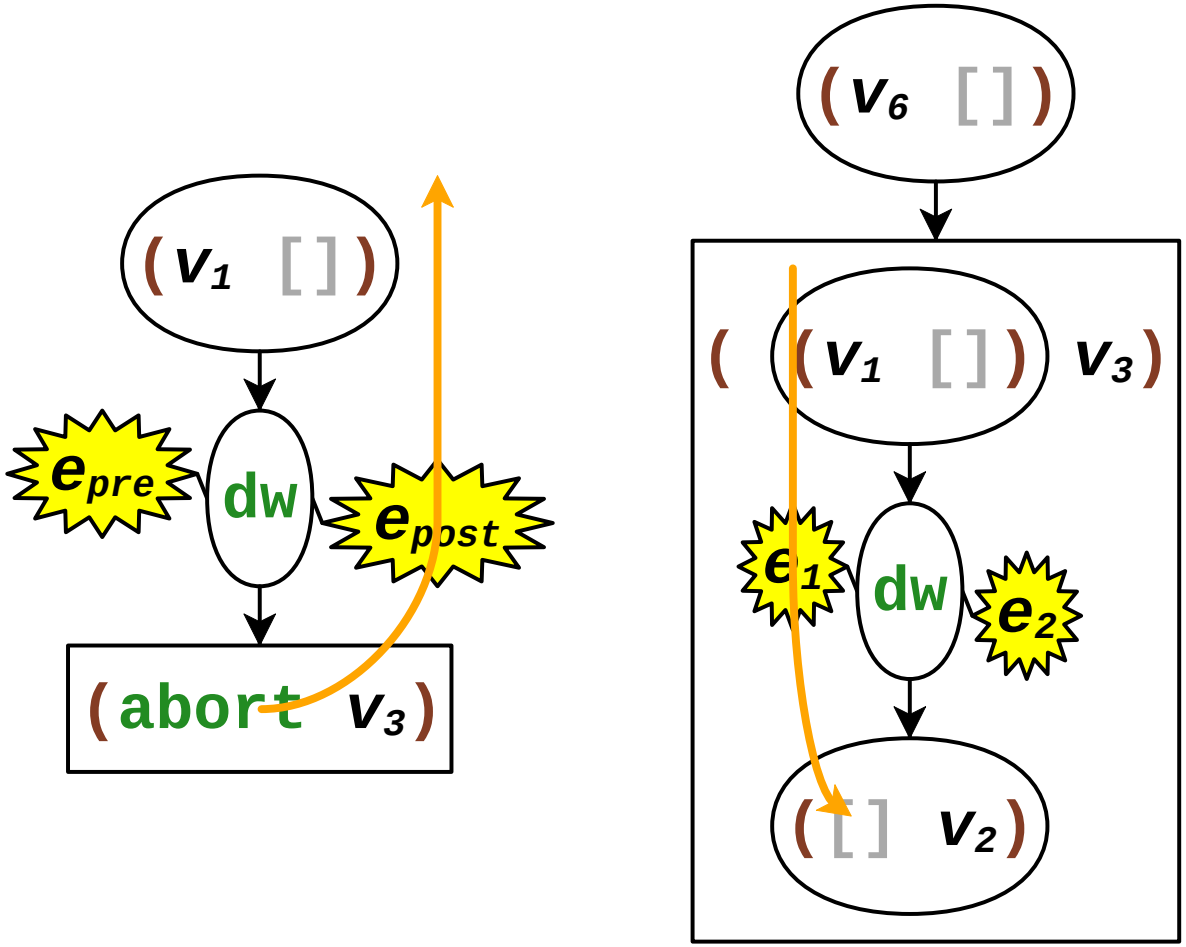
Dynamic Wind and Jumps



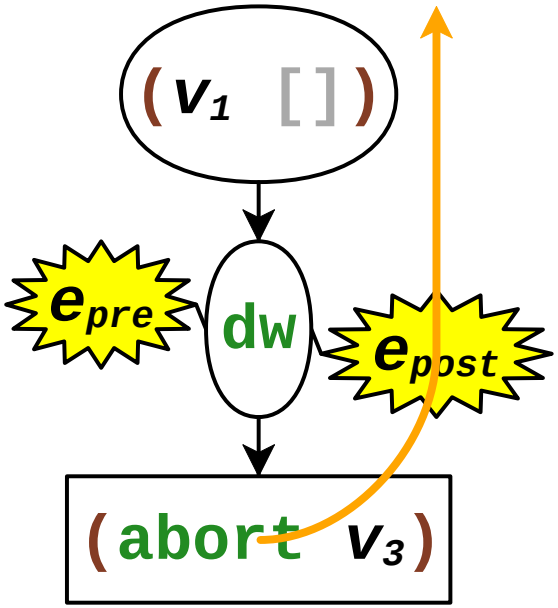
Dynamic Wind and Jumps



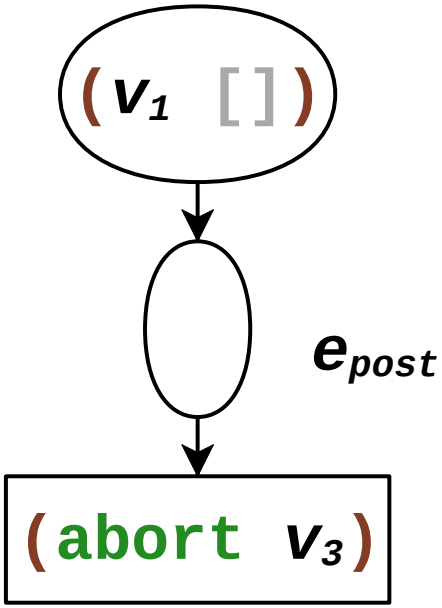
Dynamic Wind and Jumps



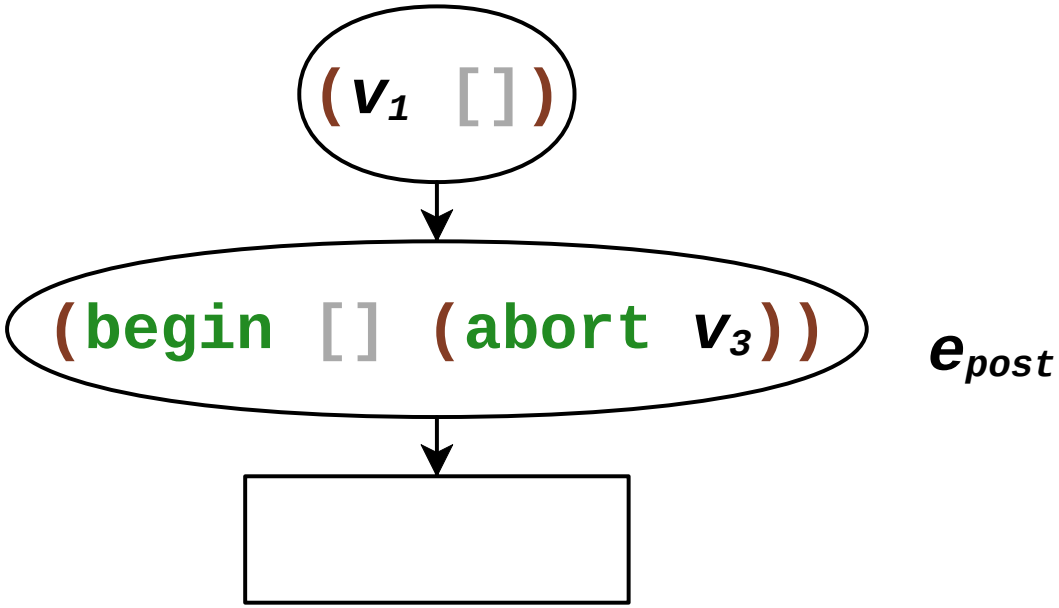
Dynamic Wind and Abort Evaluation



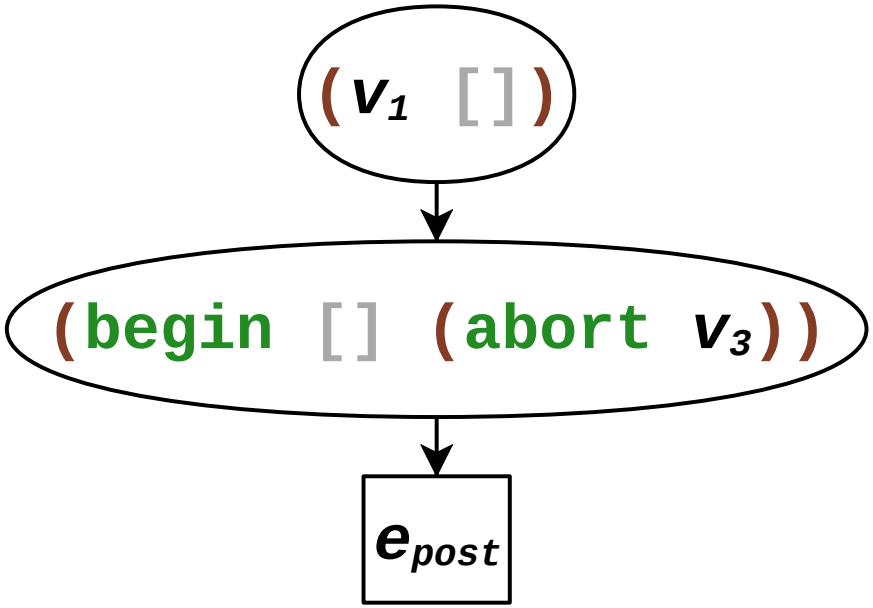
Dynamic Wind and Abort Evaluation



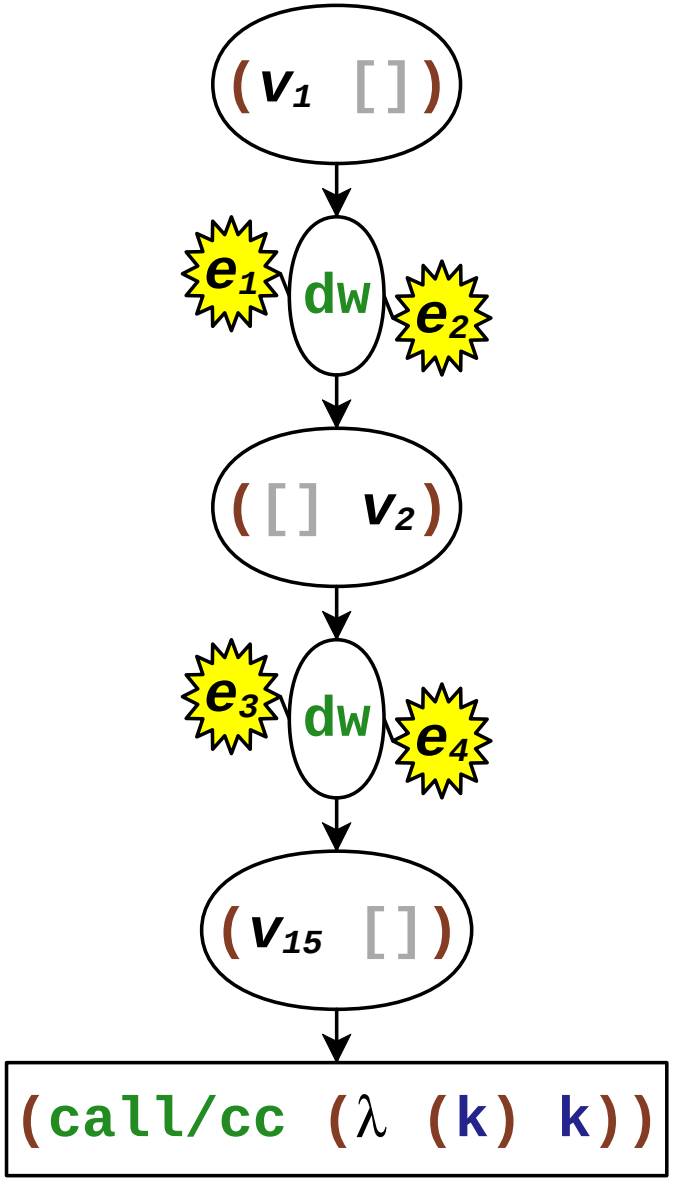
Dynamic Wind and Abort Evaluation



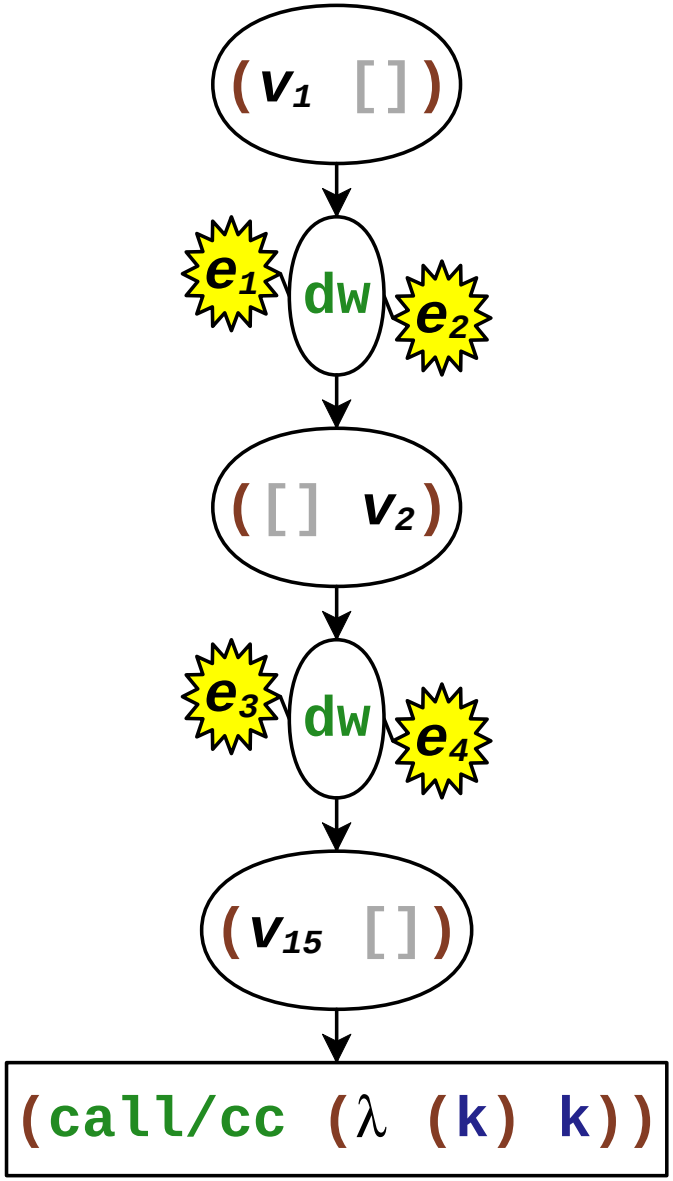
Dynamic Wind and Abort Evaluation



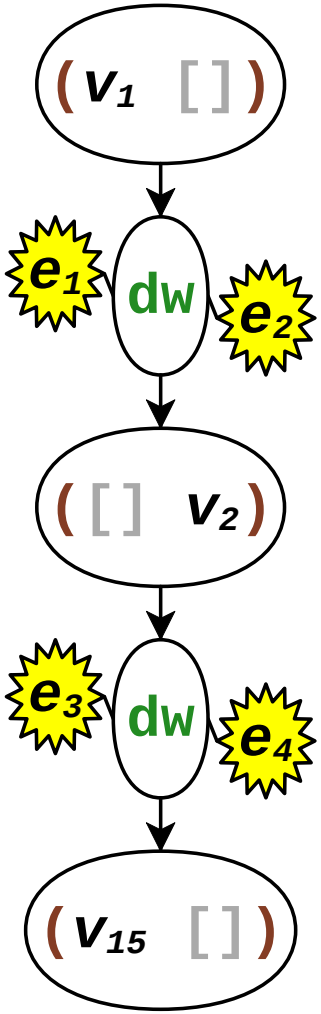
Dynamic Wind and Call/cc



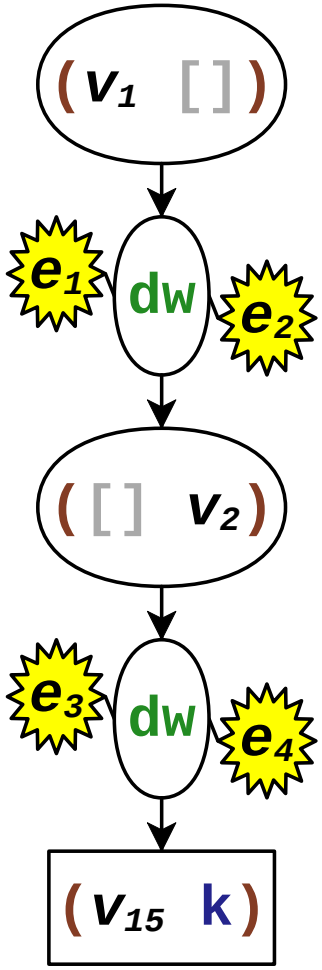
Dynamic Wind and Call/cc



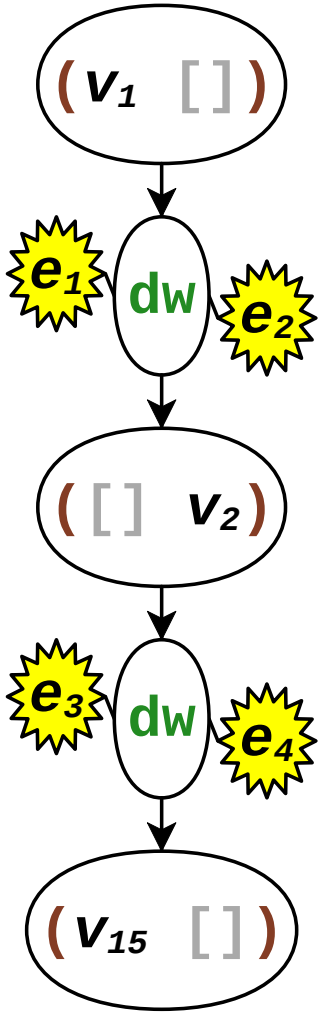
k =



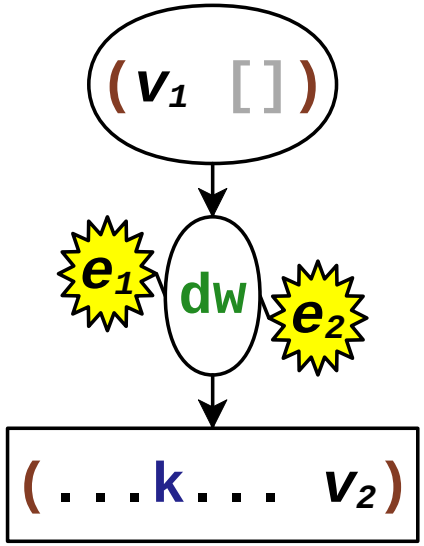
Dynamic Wind and Call/cc



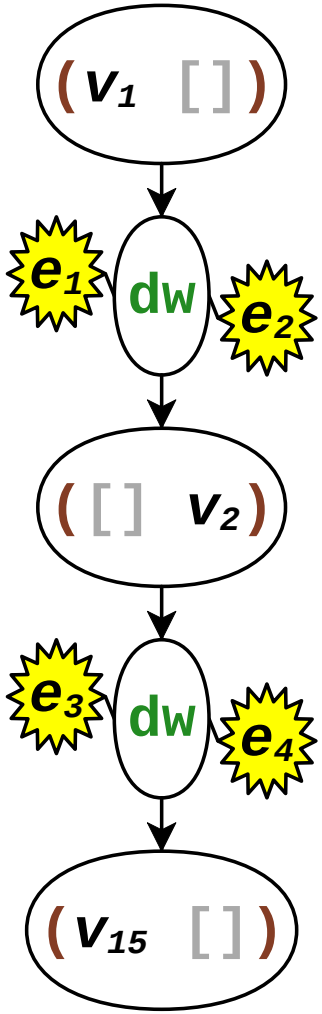
k =



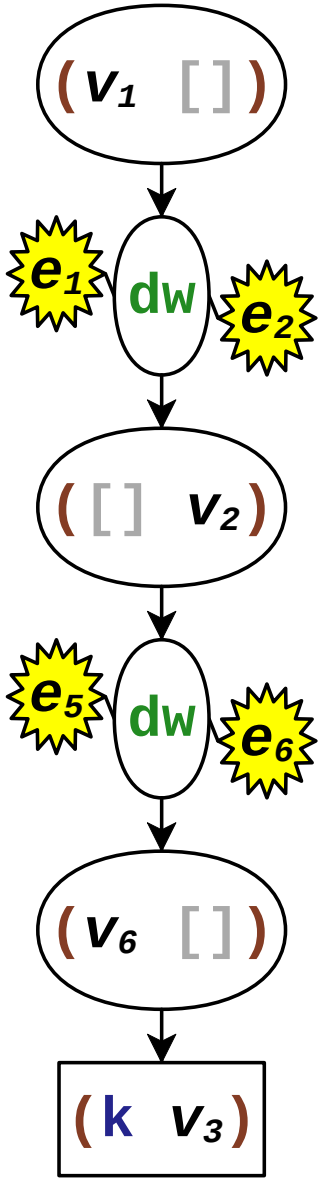
Dynamic Wind and Call/cc



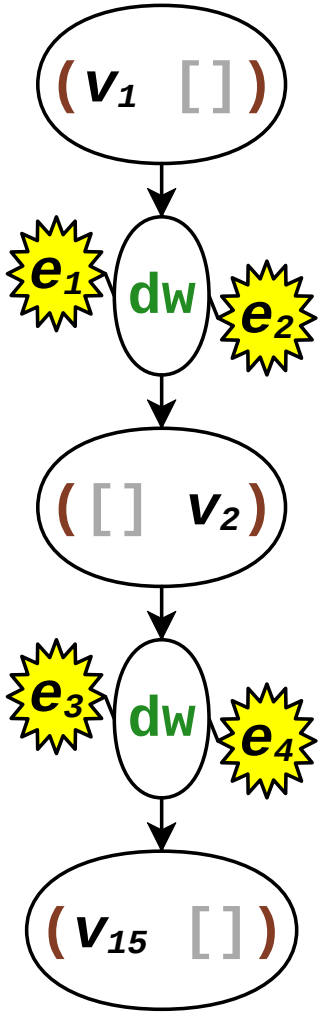
k =



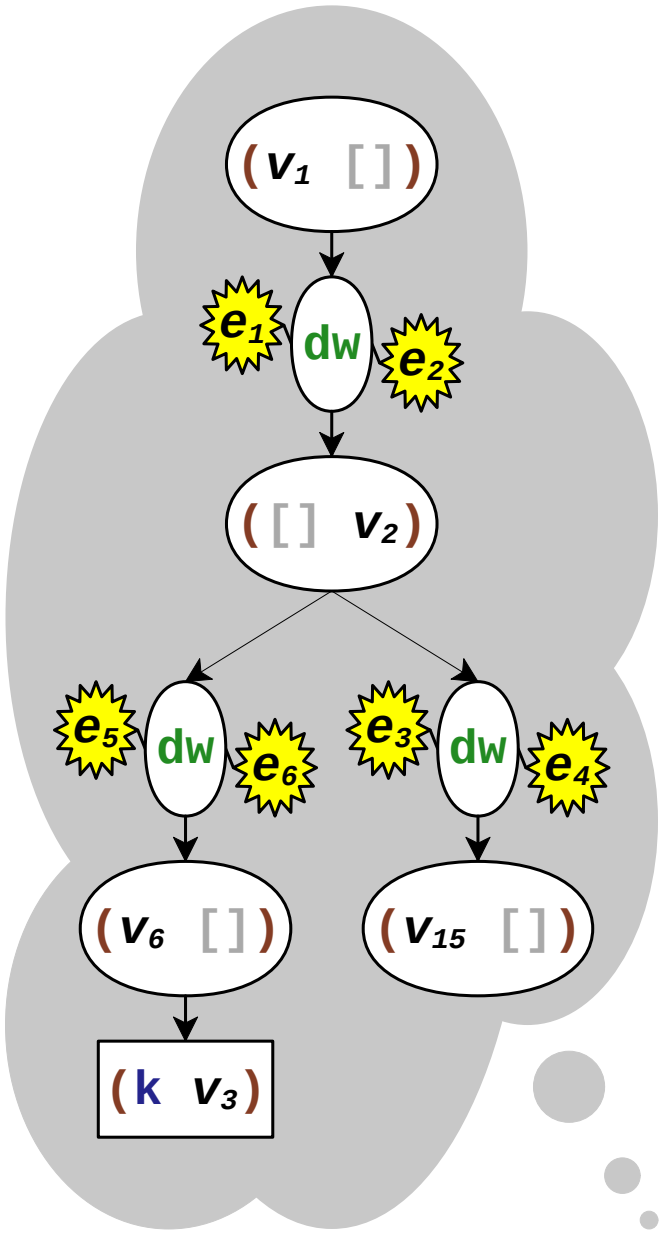
Dynamic Wind and Call/cc

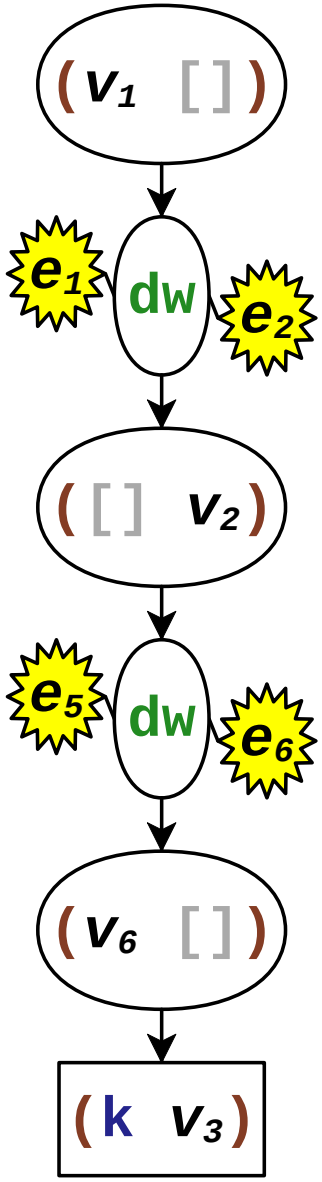


k =

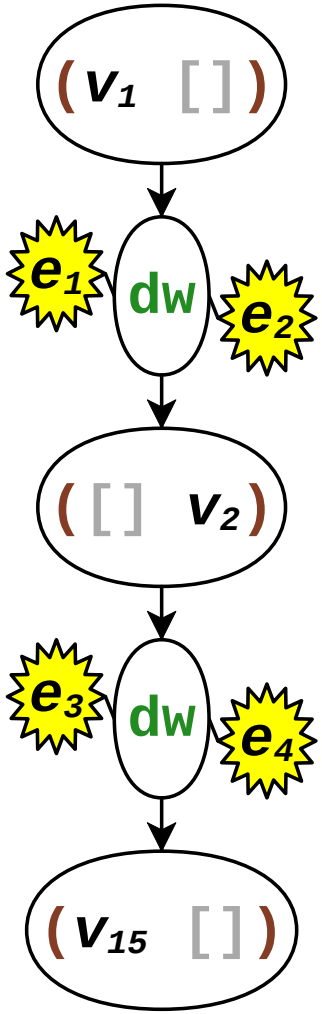


Dynamic Wind and Call/cc

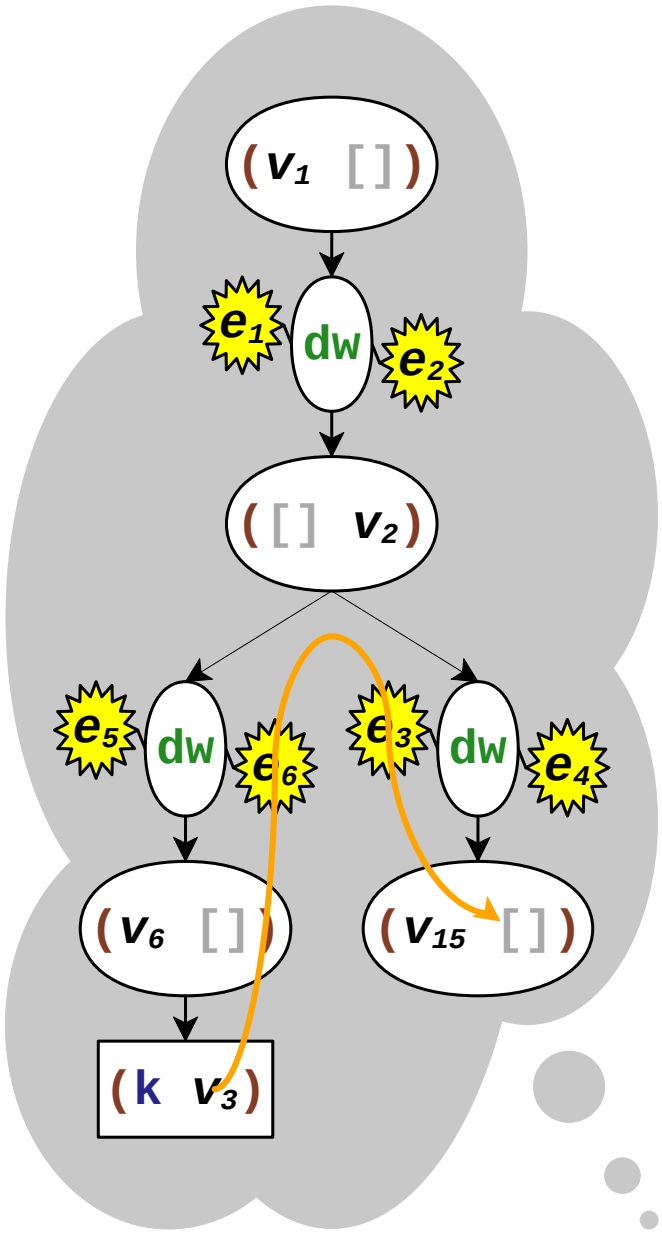


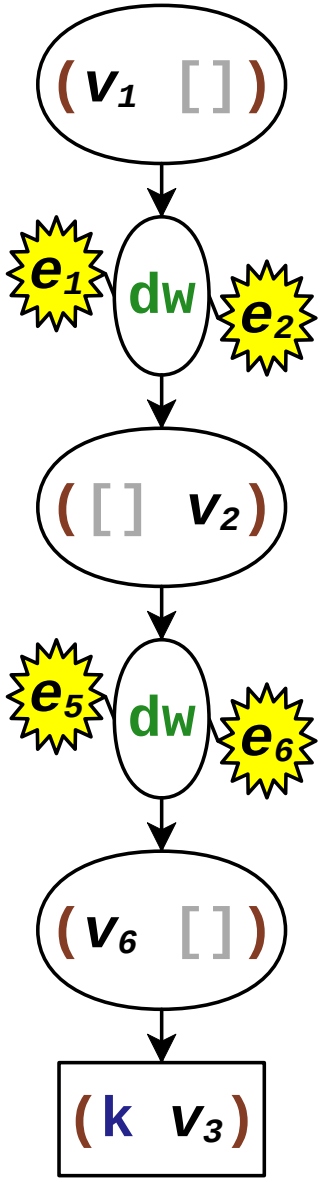


k =

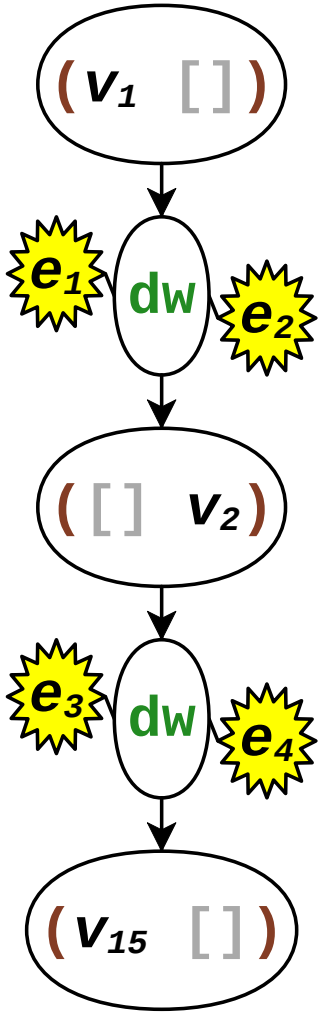


Dynamic Wind and Call/cc

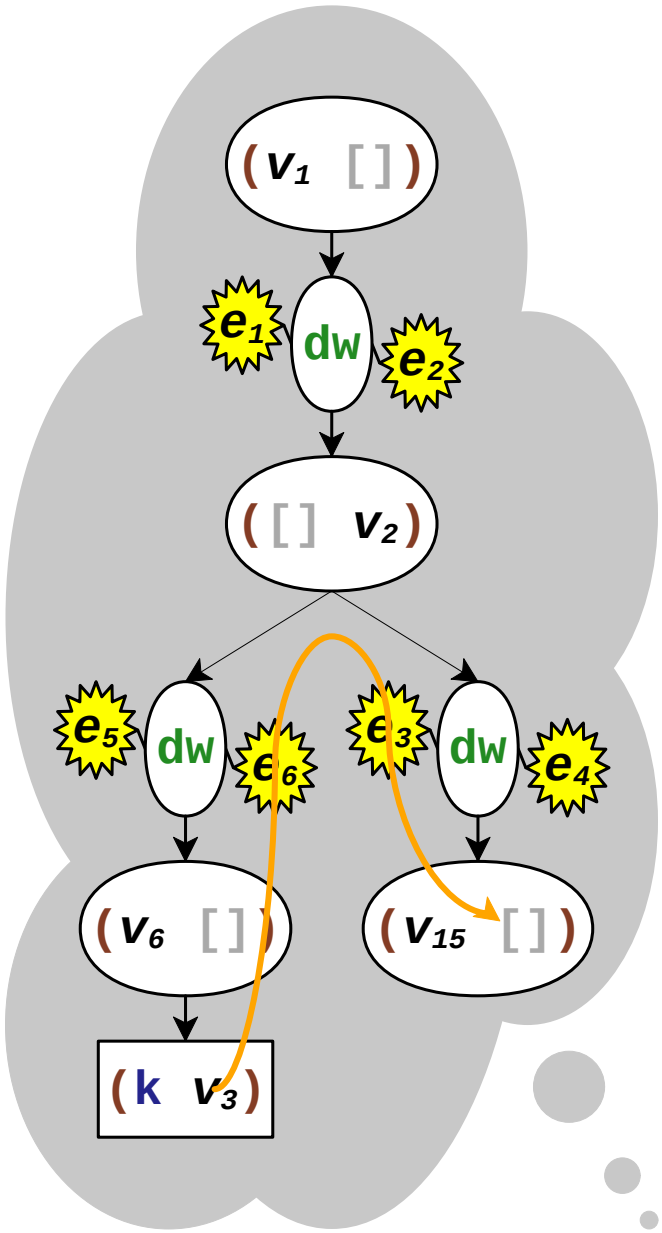


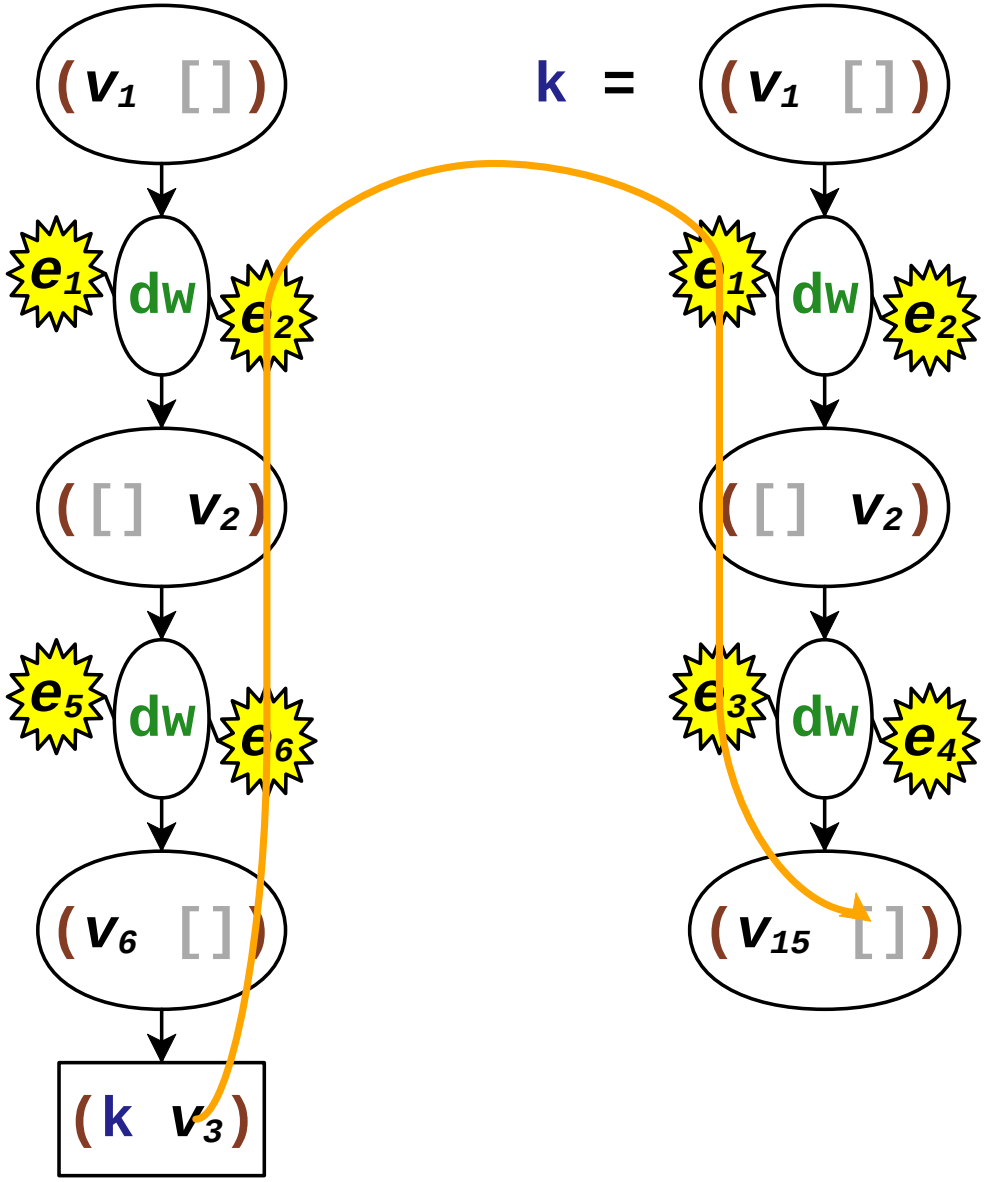


k =

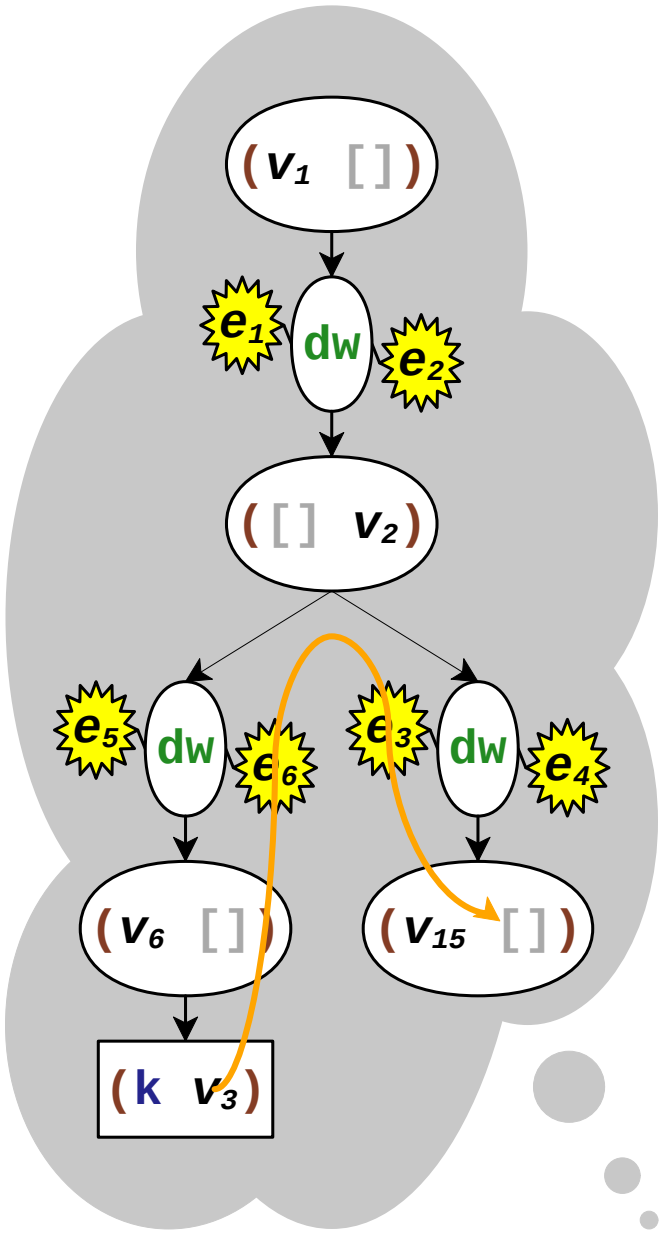


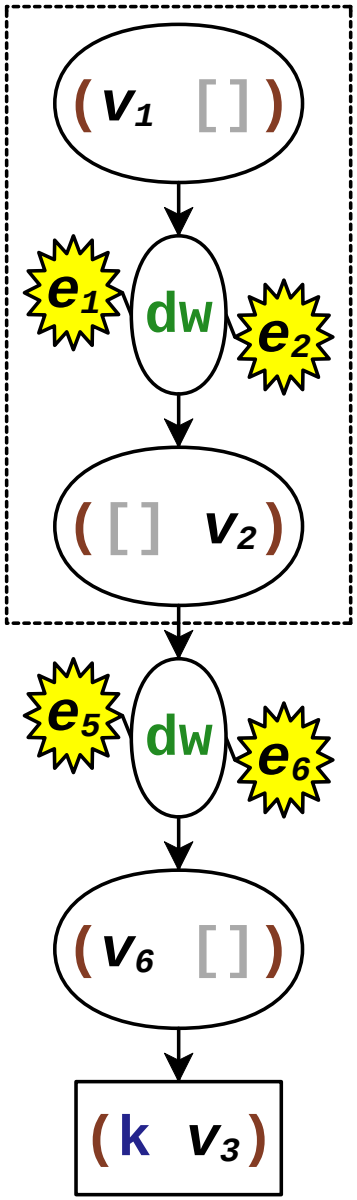
Dynamic Wind and Call/cc



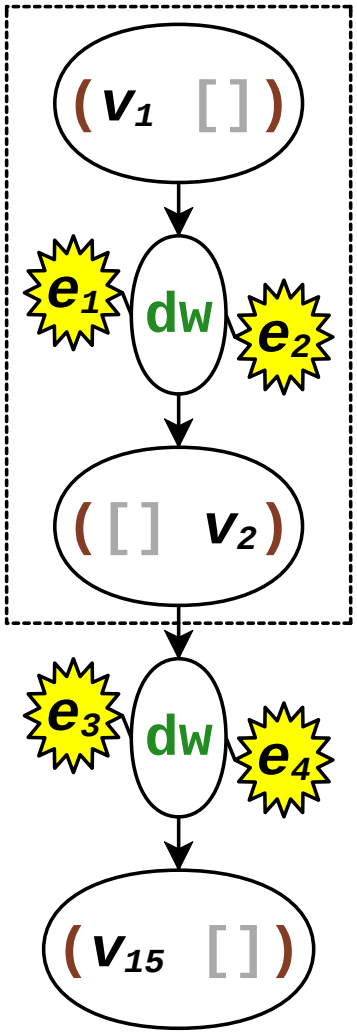


Dynamic Wind and Call/cc

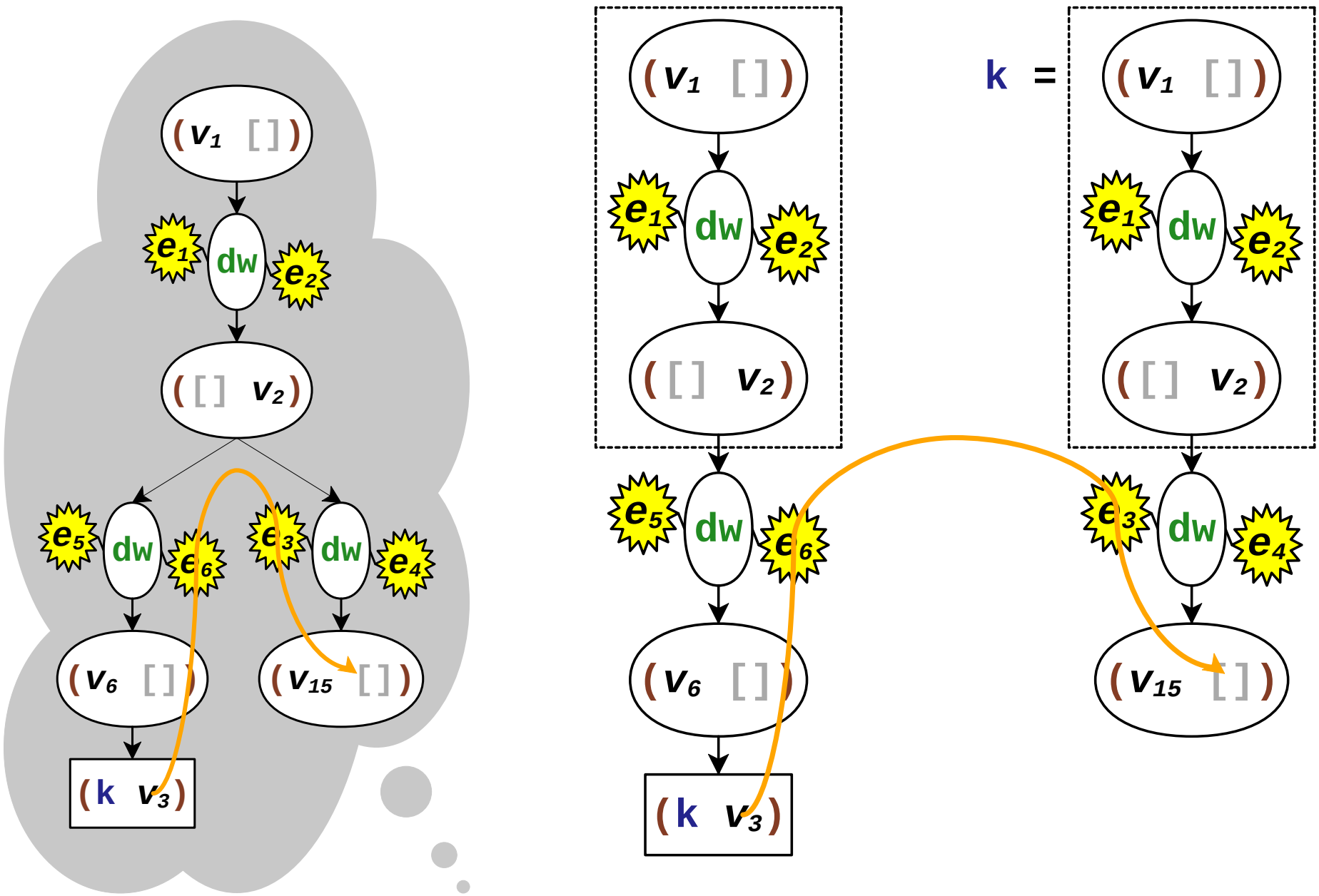




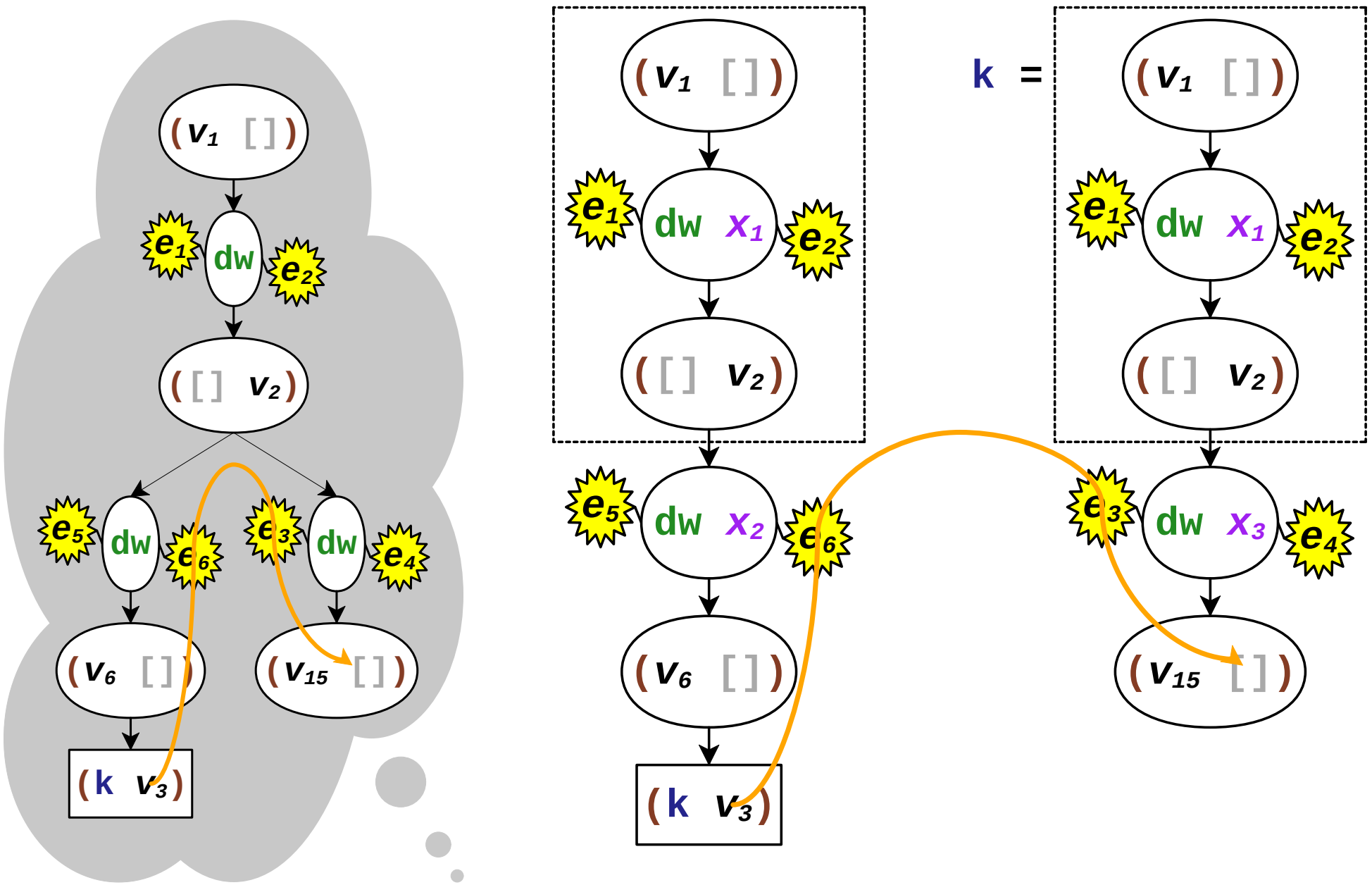
k =



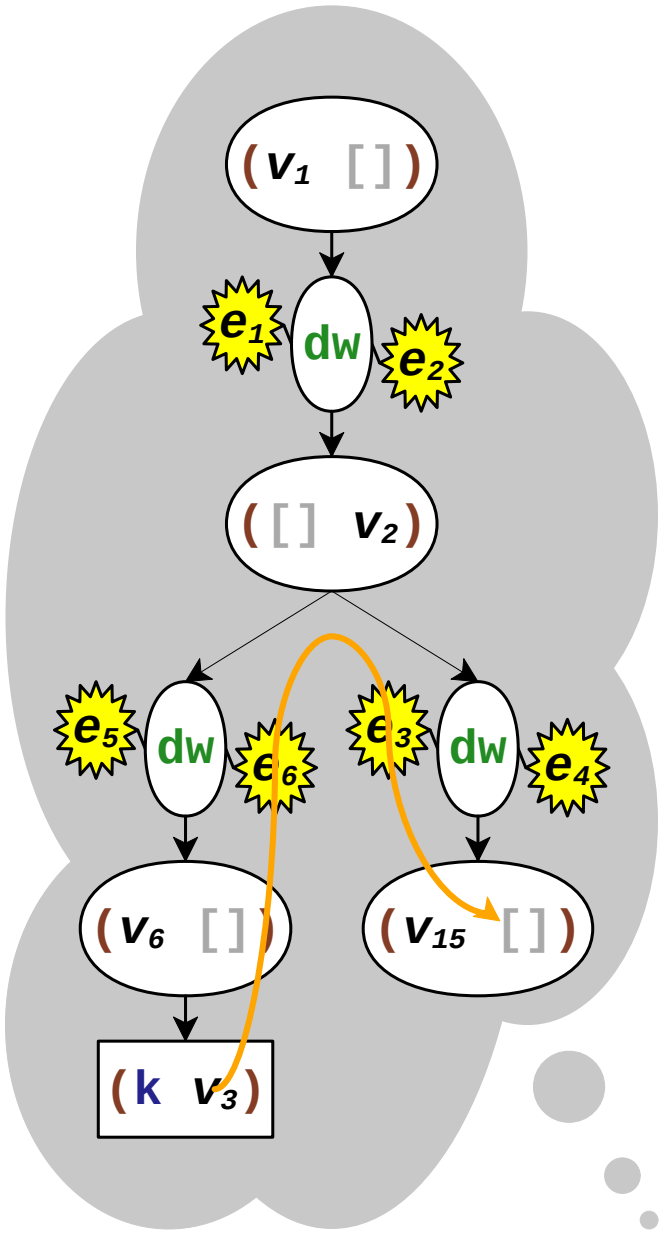
Dynamic Wind and Call/cc

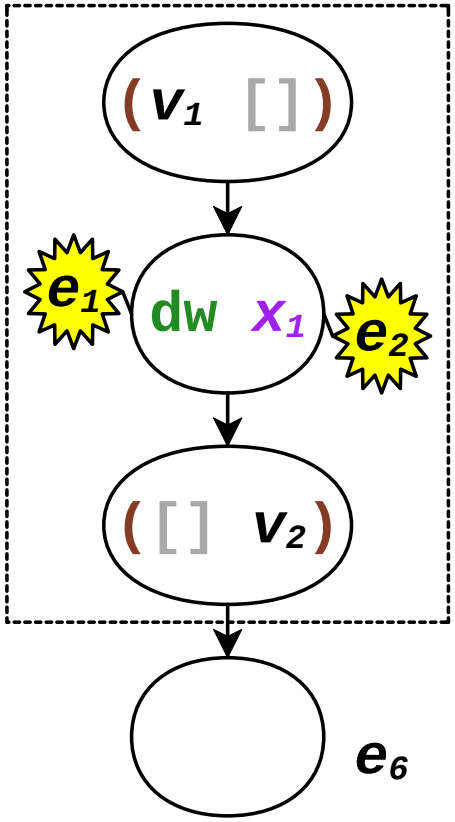


Dynamic Wind and Call/cc

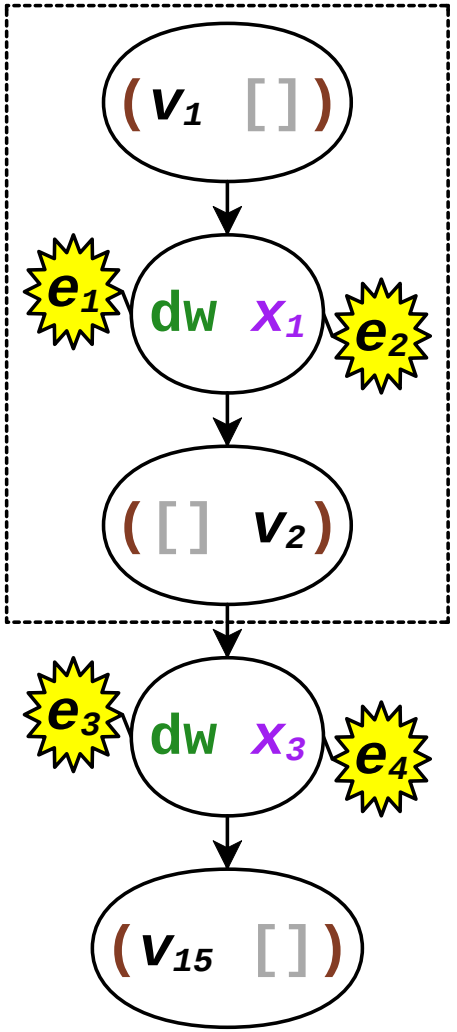


Dynamic Wind and Call/cc

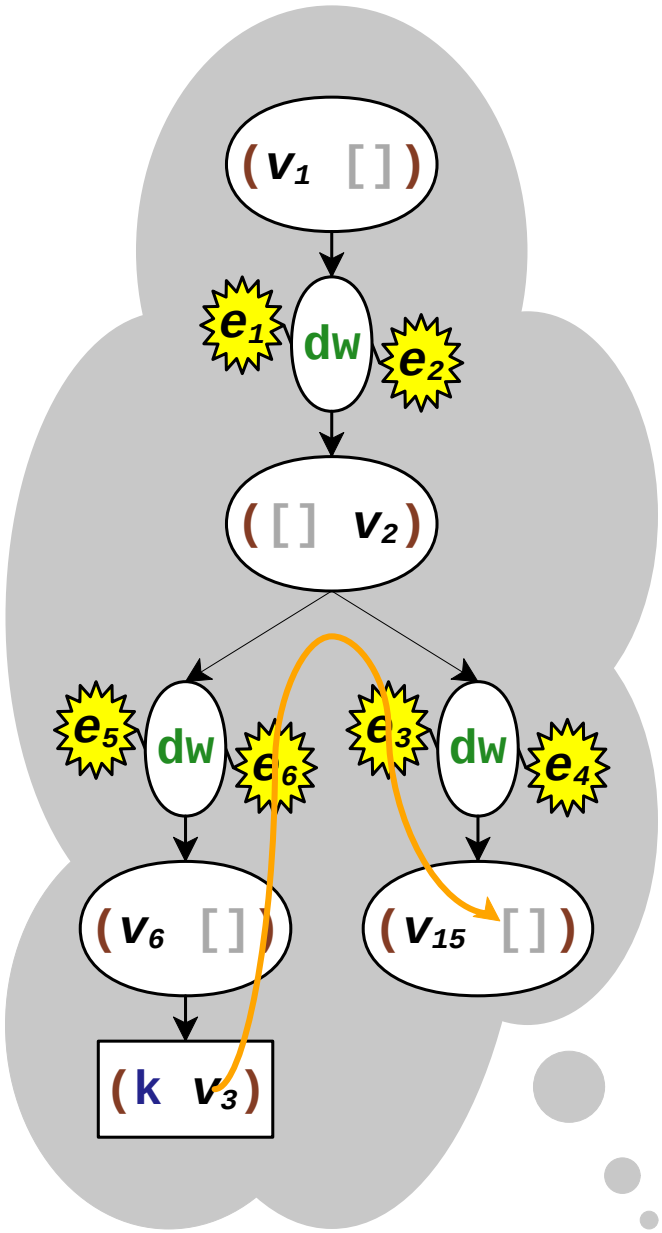


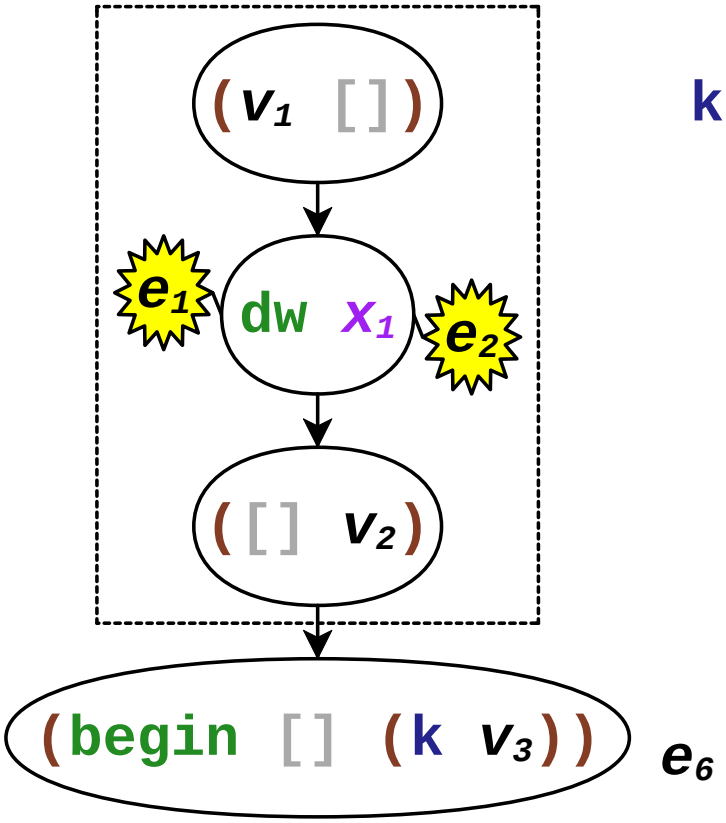


k =

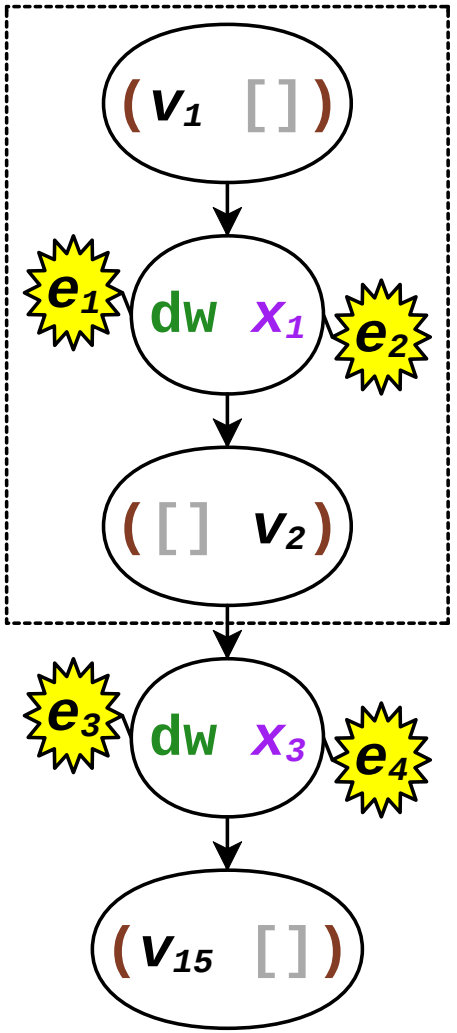


Dynamic Wind and Call/cc

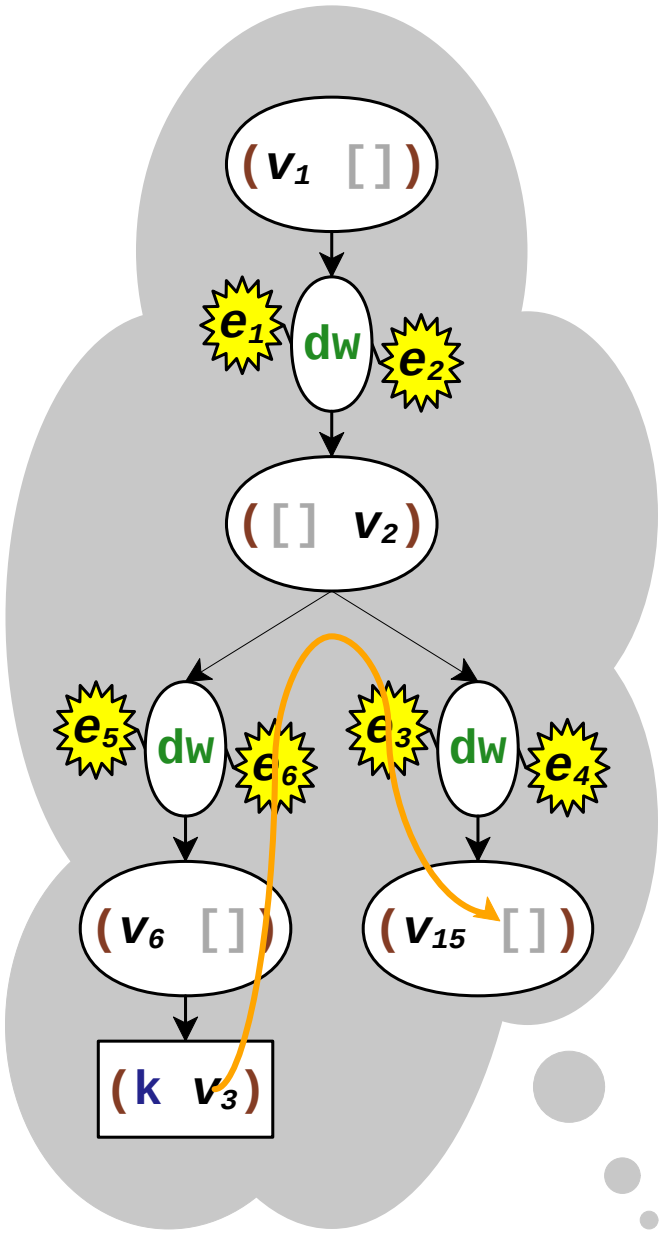


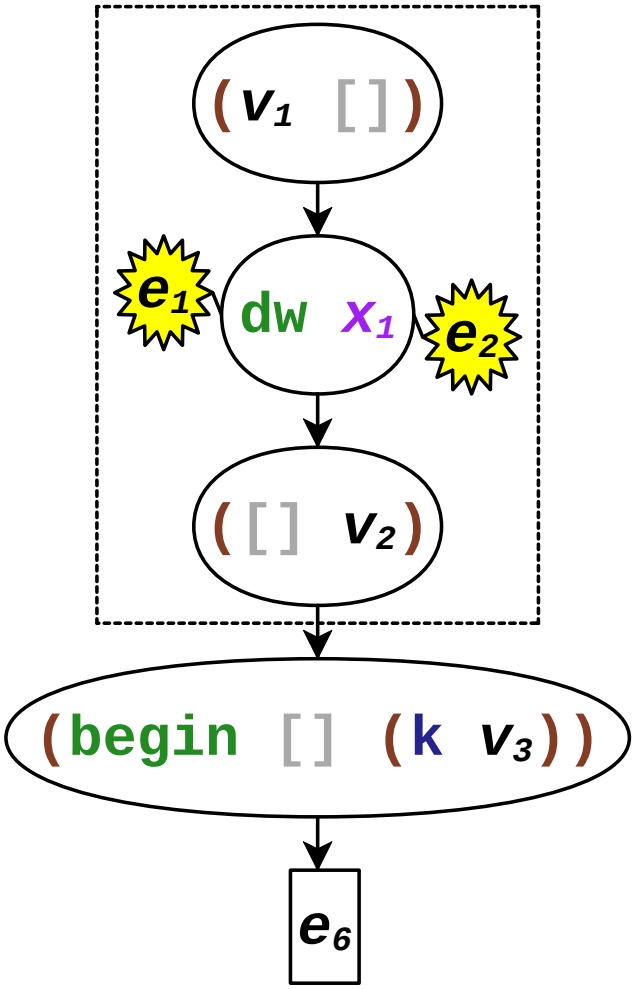


k =

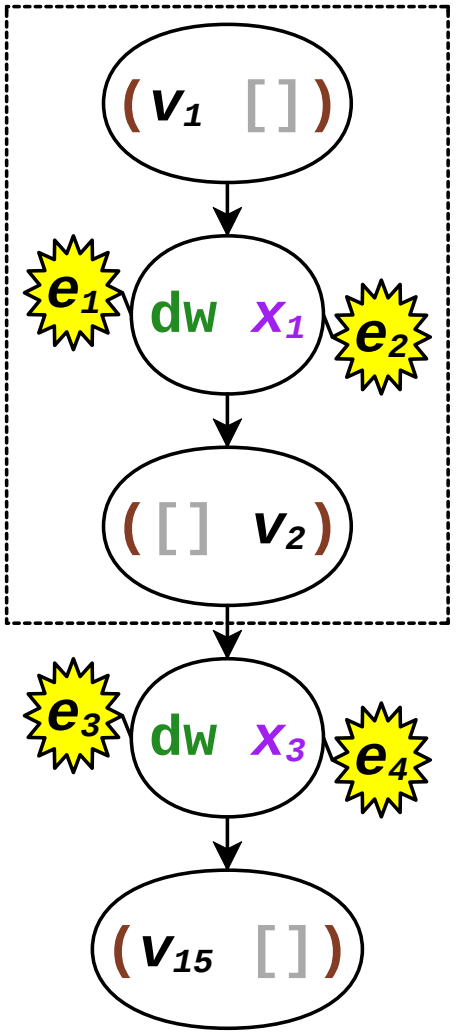


Dynamic Wind and Call/cc

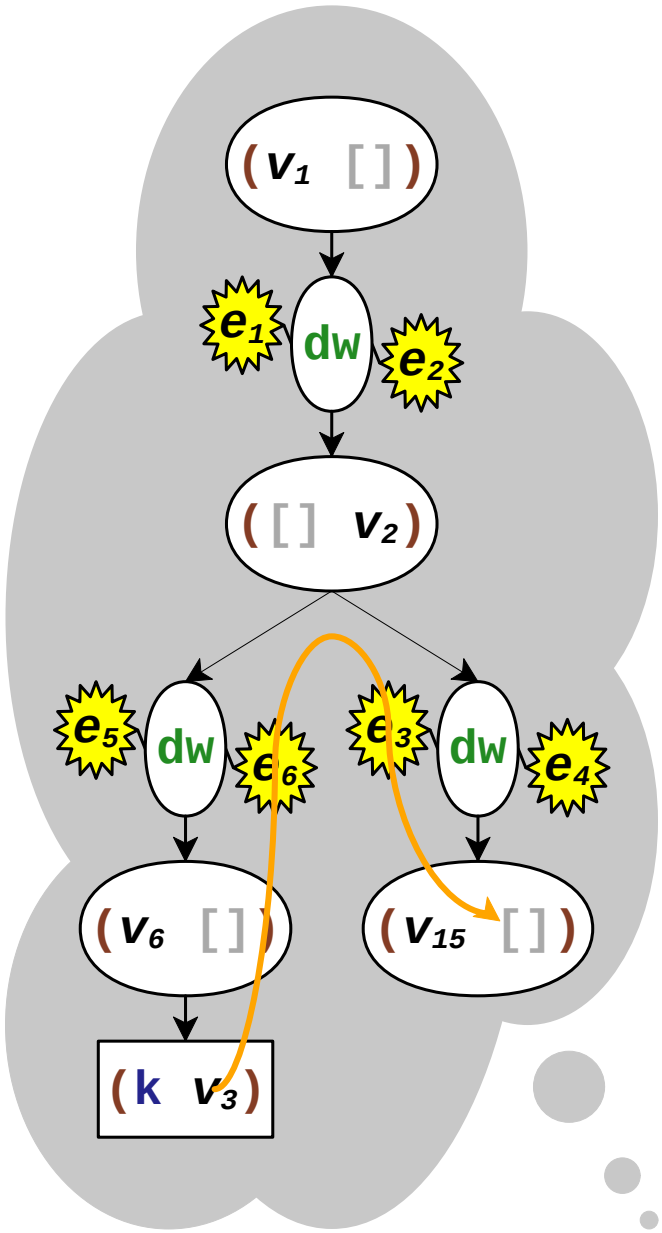


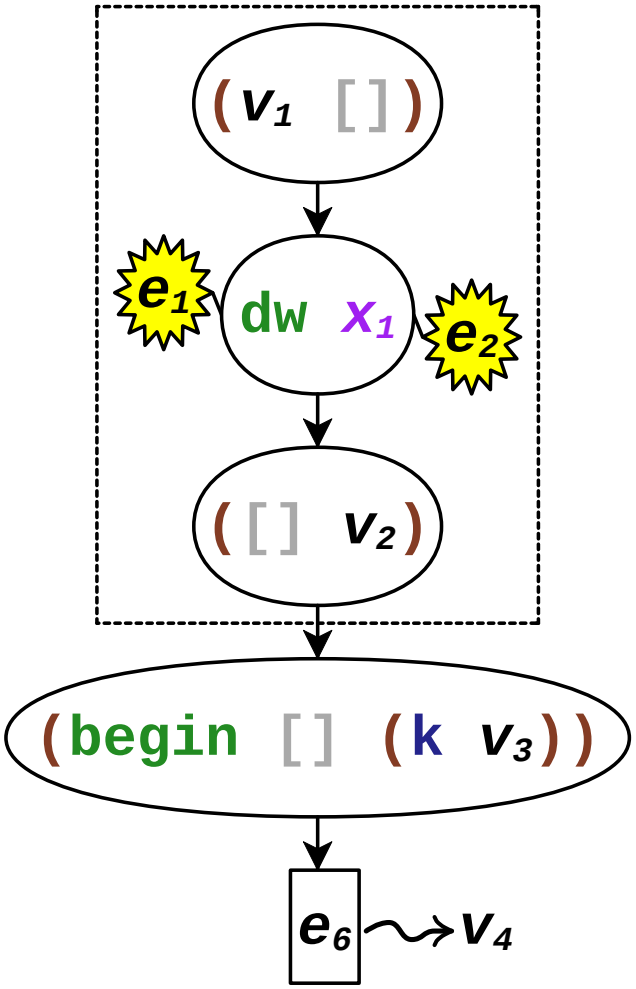


k =

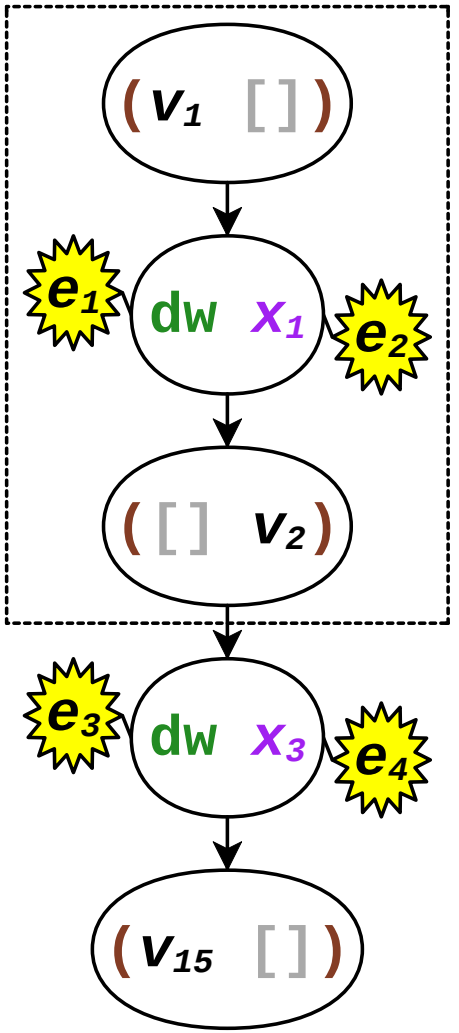


Dynamic Wind and Call/cc

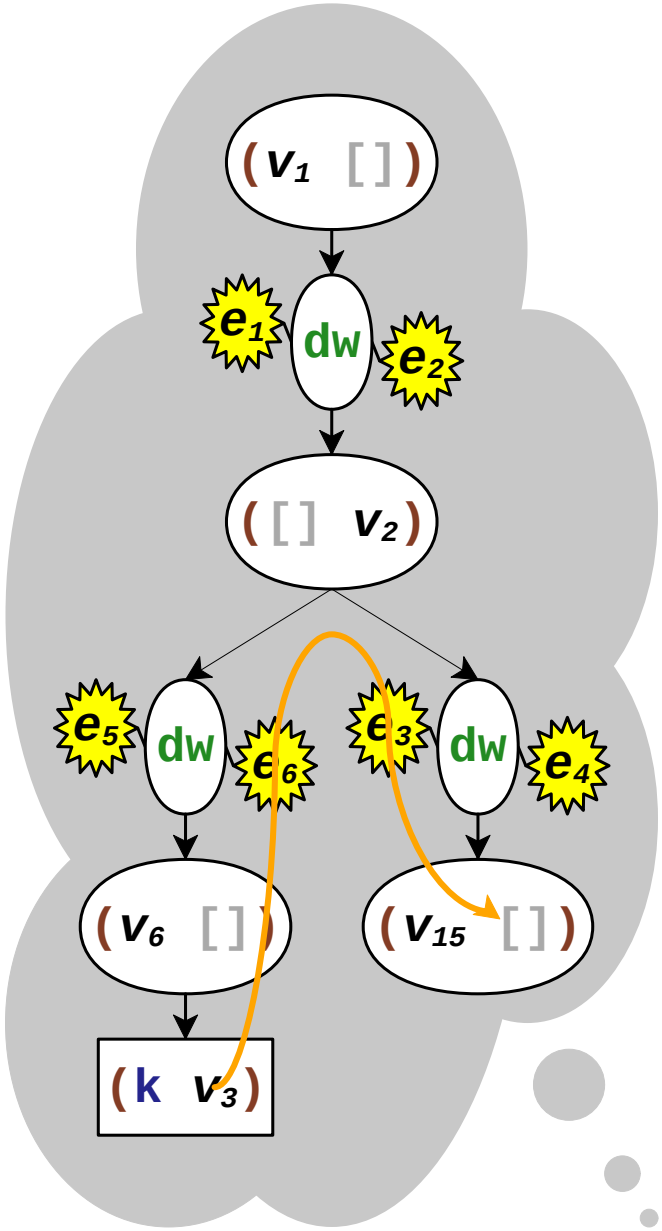


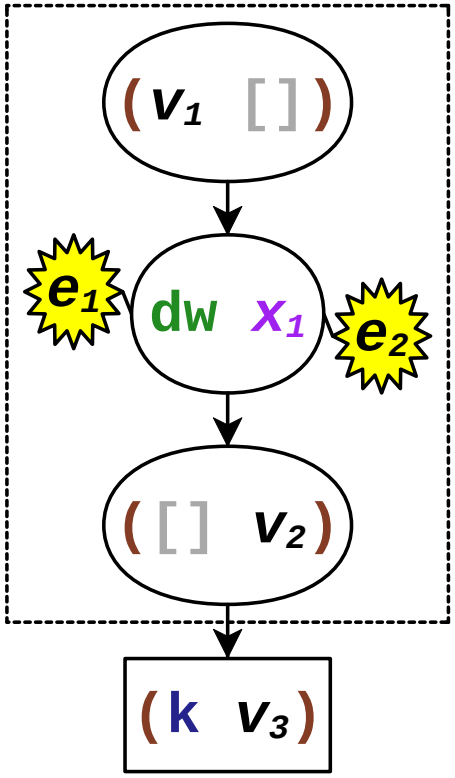


k =

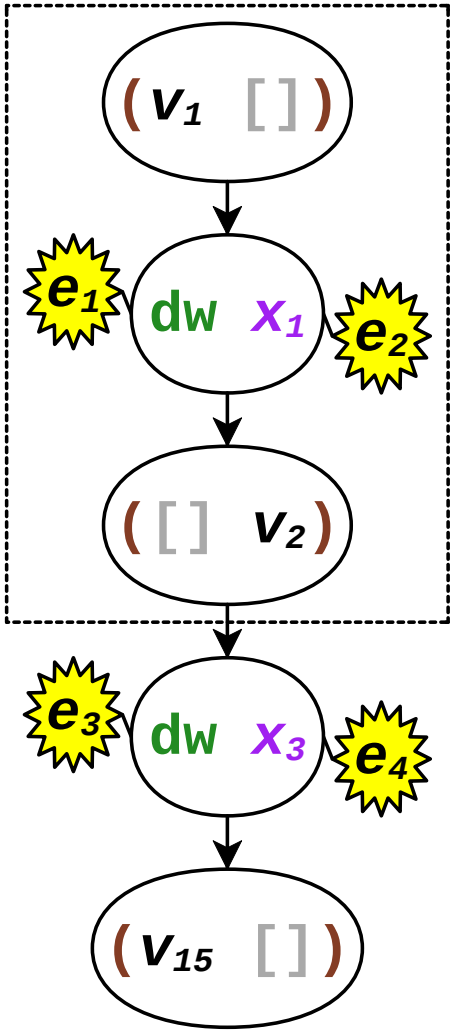


Dynamic Wind and Call/cc





k =



Dynamic Wind Summary

- `dynamic-wind` generates `dw`
- `call/cc` detects sharing in continuation jumps

- capture `dw` thunks in `call/comp` and `call/cc`
- run post thunks in `abort`
- run pre thunks in continuation composition

To Continue...

- Read the paper

  ICFP'07 Flatt, Yu, Findler, and Felleisen

- Run the Redex model:

  `http://www.cs.utah.edu/plt/delim-cont/`

- Download the implementation:

  `http://www.plt-scheme.org/`