20-7

hi jay!

Control flow analysis

Control Flow Graph
Basic Blocks
- Segments of code without transfers of control.

2 Techniques
- Continuation Passing Style (CPS)
- Non-standard Abstract Semantics

- all functions get a new argument, K

All transfers of control are <u>uniform</u>

(λ (v) ......)

```
(define (f x)                  →   (define (f x k)
  (+ (g (h x)) (g x)))               (h x (λ (hx)
                                       (+ (g hx) (g x)))))

(define (f x k)
  (h x (lambda (hx)
     (g hx (lambda (ghx)
         (+ ghx (g x)))))))

(define (f x k)                        define (h x k)
  (h x (λ (hx)
     (g hx (λ (ghx)
        (g x, (λ (gx)
           (+ ghx gx)))))))
```

Stop continuation: $(\lambda(v)\ v)$

NSAS - key behind abstraction is that the accuracy is traded for compile-time computability.

$$Expr ::= i \mid (+\ e_1\ e_2) \mid (*\ e_1\ e_2) \mid (-\ e)$$

$$e_1, e_2 \in Expr$$
$$\underline{i} \in Integer$$

S

$$[\![i]\!] = i$$
$$[\![(+\ e_1\ e_2)]\!] = [\![e_1]\!] + [\![e_2]\!]$$
$$[\![(*\ e_1\ e_2)]\!] = [\![e_1]\!] * [\![e_2]\!]$$
$$[\![(-\ e)]\!] = -[\![e]\!]$$

$$S[\![(+\ 2\ (*\ 3\ (-4)))]\!] = -10$$

Non-standard Semantics

$$S_{sign}[\![e]\!] = \begin{cases} - & S[\![e]\!] < 0 \\ 0 & S[\![e]\!] = 0 \\ + & S[\![e]\!] > 0 \end{cases}$$

$$S_{sign}[\![(+\ 2\ (*\ 3\ (-\ 4)))]\!] = -$$

Non-standard Abstract Semantics

$\hat{S}sign$

$$\hat{S}sign[\![i]\!] = Ssign[\![i]\!]$$

$$\hat{S}sign[\![(+\ e_1\ e_2)]\!] =$$

$\hat{S}sign[\![e_1]\!]$ \ $\hat{S}sign[\![e_2]\!]$	{-}	{0}	{+}	{-, 0, +}
{-}	{-}	{-}	{-, 0, +}	{-, 0, +}
{0}	{-}	{0}	{+}	{-, 0, +}
{+}	{-, 0, +}	{+}	{+}	{-, 0, +}
{-, 0, +}	{-, 0, +}	{-, 0, +}	{-, 0, +}	{-, 0, +}

$$\hat{S}sign[\![(*\ e_1\ e_2)]\!] =$$

$\hat{S}sign[\![e_1]\!]$ \ $\hat{S}sign[\![e_2]\!]$	-	0	+	{-, 0, +}
{-}	{+}	{0}	{-}	{-, 0, +}
{0}	{0}	{0}	{0}	{0}
{+}	{-}	{0}	{+}	{-, 0, +}
{-, 0, +}	{-, 0, +}	{0}	{-, 0, +}	{-, 0, +}

$$\hat{S}sign[\![(+\ 2\ (*\ 3\ (-\ 4)))]\!] =$$

(+ {+} (× {+} (- {+})))

(+ {+} (× {+} {-}))

(+ {+} {-})

{-, 0, +}

Contours

Variable environment
Contour environment.

(define (f x K)
  (zero