$\vau$-calculi and the theory of fexprs

John N. Shutt

WPI

October 2007

outline

outline

- fexprs

outline

- fexprs
- λ-calculus

outline

- fexprs
- λ-calculus
- $\vau$-calculus

outline

- fexprs
- λ-calculus
- $\vau$-calculus
- Lisp / λ-calculus correspondence

outline

- fexprs
- λ-calculus
- $\vau$-calculus
- Lisp / λ-calculus correspondence
- conclusion

terminology

terminology

combination — pair to be eval'd

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands
eval'd		arguments

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands
eval'd	combiner	arguments

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands
eval'd	combiner	arguments

(e.g.: Scheme procedure)

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands
eval'd	combiner	arguments

(e.g.: special form combiner)

(e.g.: Scheme procedure)

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands
eval'd	combiner	arguments

operative combiner — acts on its operands
(e.g.: special form combiner)

(e.g.: Scheme procedure)

terminology

combination — pair to be eval'd

	car	cdr
uneval'd	operator	operands
eval'd	combiner	arguments

operative combiner — acts on its operands
(e.g.: special form combiner)

applicative combiner — acts on its arguments
(e.g.: Scheme procedure)

fexprs

- first-class operatives

fexprs

- first-class operatives
  that act by evaluating their bodies

fexprs

- first-class operatives
  that act by evaluating their bodies
- alternative to macros

fexprs

- first-class operatives
  that act by evaluating their bodies
- alternative to macros
- abstractive power

λ-calculus

λ-calculus

- $T \;::=\; x \mid c \mid (TT) \mid (\lambda x.T)$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2] \qquad (\beta)$

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ (β)

λ-calculus

- $T \; ::= \; x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2] \qquad (\beta)$

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2] \qquad (\beta)$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ (β)
- $T_1 \longrightarrow_\beta T_2$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ (β)
- $T_1 \longrightarrow_\beta T_2$ compatibility

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ $\qquad (\beta)$
- $T_1 \longrightarrow_\beta T_2$ $\qquad$ compatibility

$$\begin{aligned} \forall\, C, T_k, \qquad & (T_1 \longrightarrow_\beta T_2) \\ \Rightarrow \quad & (C[T_1] \longrightarrow_\beta C[T_2]) \end{aligned}$$

λ-calculus

- $T ::= x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ $\qquad (\beta)$
- $T_1 \longrightarrow_\beta T_2$ compatibility

contextual equivalence

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ $\qquad (\beta)$
- $T_1 \longrightarrow_\beta T_2$ compatibility

contextual equivalence

$$\Rightarrow \forall\, C, \qquad C[T_1] \longrightarrow^*_\beta \text{observable}$$
$$\text{iff } C[T_2] \longrightarrow^*_\beta \text{observable}$$

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ $\qquad (\beta)$
- $T_1 \longrightarrow_\beta T_2$ compatibility

contextual equivalence

λ-calculus

- $T \; ::= \; x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ (β)
- $T_1 \longrightarrow_\beta T_2$ compatibility

contextual equivalence

```
((lambda (x) (+ x y)) 3)
```

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ (β)
- $T_1 \longrightarrow_\beta T_2$ compatibility

contextual equivalence

```
((lambda (x) (+ x y)) 3)
```

$$\Rightarrow \ ((\lambda x.(+\ x\ y))\ 3)$$

λ-calculus

- $T \ ::= \ x \mid c \mid (TT) \mid (\lambda x.T)$
- $(\lambda x.T_1)T_2 \longrightarrow T_1[x \leftarrow T_2]$ $\qquad (\beta)$
- $T_1 \longrightarrow_\beta T_2$ compatibility

contextual equivalence

λ-calculus + `quote`

- $T_1 \longrightarrow_q T_2$ compatibility

contextual equivalence

$$(\texttt{quote}\ T_1) \equiv (\texttt{quote}\ T_2)$$

λ-calculus + quote

- $T_1 \longrightarrow_q T_2$

contextual equivalence

$$(\texttt{quote}\ T_1) \equiv (\texttt{quote}\ T_2)$$

λ-calculus $+$ quote

- $T_1 \longrightarrow_q T_2$

$$(\texttt{quote}\ T_1) \equiv (\texttt{quote}\ T_2)$$

explicit evaluation

- $T_1 \longrightarrow_{\wr} T_2$

$$(\texttt{quote}\ T_1) \equiv (\texttt{quote}\ T_2)$$

explicit evaluation

- $T_1 \longrightarrow_{\digamma} T_2$

$$T_1$$

explicit evaluation

- $T_1 \longrightarrow_{\eth} T_2$

$$T_1$$

$$[\text{eval}\ T_1\ e]$$

explicit evaluation

- $T_1 \longrightarrow_{\text{ɿ}} T_2$

$$\begin{array}{ll} T_1 & \longrightarrow^*_{\text{ɿ}} T_2 \\ [\text{eval}\ T_1\ e] & \end{array}$$

explicit evaluation

- $T_1 \longrightarrow_{\text{ʄ}} T_2$

$$T_1$$

$$[\text{eval}\ T_1\ e] \longrightarrow^{*}_{\text{ʄ}} T_2$$

explicit evaluation

- $T_1 \longrightarrow_{\natural} T_2$

$$(\texttt{quote}\ T_1) \equiv (\texttt{quote}\ T_2)$$

explicit evaluation

- $T_1 \longrightarrow_{?} T_2$ compatibility

contextual equivalence

$$(\texttt{quote}\ T_1) \equiv (\texttt{quote}\ T_2)$$

explicit evaluation

- $T_1 \longrightarrow_{\wr} T_2$ compatibility

contextual equivalence

```
((lambda (x) (+ x y)) 3)
```

explicit evaluation

- $T_1 \longrightarrow_{?} T_2$ compatibility

contextual equivalence

```
((lambda (x) (+ x y)) 3)
```

$$\Rightarrow ((\lambda x.(+\ x\ y))\ 3)$$

explicit evaluation

- $T_1 \longrightarrow_{\vartheta} T_2$ compatibility

contextual equivalence

$$\texttt{((lambda (x) (+ x y)) 3)}$$

$$\Rightarrow \text{[eval } \texttt{((lambda (x) (+ x y)) 3)} \; e\text{]}$$

explicit evaluation

- $T_1 \longrightarrow_{\wr} T_2$ compatibility

contextual equivalence

`((lambda (x) (+ x y)) 3)`

$\Rightarrow$ [eval `((lambda (x) (+ x y)) 3)` e]

explicit evaluation

- $T_1 \longrightarrow_{\wr} T_2$ compatibility

contextual equivalence

```
((lambda (x) (+ x y)) 3)
```

$$\Rightarrow ((\lambda x.(+\ x\ y))\ 3)$$

explicit evaluation

- $T_1 \longrightarrow_{ϝ} T_2$ compatibility

contextual equivalence

`((lambda (x) (+ x y)) 3)`

$\Rightarrow$ [eval `((lambda (x) (+ x y)) 3)` e]

explicit evaluation

- $T_1 \longrightarrow_{ʔ} T_2$ compatibility

contextual equivalence

`((lambda (x) (+ x y)) 3)`

$\Rightarrow$ [eval `((lambda (x) (+ x y)) 3)` e]

$\imath_x$-calculus

$\vau_x$-calculus

$S ::=$

$\imath_x$-calculus

$S ::= c$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$\daleth_x$-calculus

$$S ::= c \mid \langle \daleth x.T \rangle \mid \langle \daleth_2.T \rangle \mid \langle \daleth_0.T \rangle$$

$$T ::=$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$$T ::= x$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$$T ::= x \mid S$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$
$$T ::= x \mid S \mid (T \;.\; T)$$

ℓ_x-calculus

$$S ::= c \mid \langle \ell x.T \rangle \mid \langle \ell_2.T \rangle \mid \langle \ell_0.T \rangle$$

$$T ::= x \mid S \mid (T \,.\, T) \mid \langle T \rangle$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$$T ::= x \mid S \mid (T\ .\ T) \mid \langle T \rangle \mid [\text{eval}\ T]$$

ꝩ_x-calculus

$$S ::= c \mid \langle \text{ꝩ} x.T \rangle \mid \langle \text{ꝩ}_2.T \rangle \mid \langle \text{ꝩ}_0.T \rangle$$

$$T ::= x \mid S \mid (T\ .\ T) \mid \langle T \rangle \mid [\text{eval}\ T] \mid [\text{combine}\ T\ T]$$

$\daleth_x$-calculus

$$S ::= c \mid \langle \daleth x.T \rangle \mid \langle \daleth_2.T \rangle \mid \langle \daleth_0.T \rangle$$
$$T ::= x \mid S \mid (T \;.\; T) \mid \langle T \rangle \mid [\text{eval}\ T] \mid [\text{combine}\ T\ T]$$

$[\text{eval}\ x]$

$\imath_x$-calculus

$$S ::= c \mid \langle \imath x.T \rangle \mid \langle \imath_2.T \rangle \mid \langle \imath_0.T \rangle$$

$$T ::= x \mid S \mid (T\ .\ T) \mid \langle T \rangle \mid [\text{eval}\ T] \mid [\text{combine}\ T\ T]$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$$T ::= x \mid S \mid (T \;.\; T) \mid \langle T \rangle \mid [\text{eval } T] \mid [\text{combine } T\; T]$$

$$[\text{eval } S] \longrightarrow S$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$$T ::= x \mid S \mid (T \;.\; T) \mid \langle T \rangle \mid [\text{eval } T] \mid [\text{combine } T \; T]$$

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \;.\; T_2)] \longrightarrow [\text{combine } [\text{eval } T_1] \; T_2]$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T \rangle \mid \langle \vau_2.T \rangle \mid \langle \vau_0.T \rangle$$

$$T ::= x \mid S \mid (T \;.\; T) \mid \langle T \rangle \mid [\text{eval } T] \mid [\text{combine } T\ T]$$

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \;.\; T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\ T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$\daleth_x$-calculus

$$S ::= c \mid \langle \daleth x.T \rangle \mid \langle \daleth_2.T \rangle \mid \langle \daleth_0.T \rangle$$

$$T ::= x \mid S \mid (T \;.\; T) \mid \langle T \rangle \mid [\text{eval } T] \mid [\text{combine } T\; T]$$

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \;.\; T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\; T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$$[\text{combine } \langle \daleth_0.T \rangle\; (\,)] \longrightarrow T$$

$\vau_x$-calculus

$$S ::= c \mid \langle \vau x.T\rangle \mid \langle \vau_2.T\rangle \mid \langle \vau_0.T\rangle$$

$$T ::= x \mid S \mid (T \;.\; T) \mid \langle T\rangle \mid [\text{eval}\; T] \mid [\text{combine}\; T\; T]$$

$$[\text{eval}\; S] \longrightarrow S$$

$$[\text{eval}\; (T_1 \;.\; T_2)] \longrightarrow [\text{combine}\; [\text{eval}\; T_1]\; T_2]$$

$$[\text{eval}\; \langle T\rangle] \longrightarrow \langle [\text{eval}\; T]\rangle$$

$$[\text{combine}\; \langle \vau_0.T\rangle\; (\,)] \longrightarrow T$$

$$[\text{combine}\; \langle \vau_2.T_1\rangle\; (T_2 \;.\; T_3)] \longrightarrow$$
$$[\text{combine}\; [\text{combine}\; T_1\; T_2]\; T_3]$$

$ꝩ_x$-calculus

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \;.\; T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\; T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$$[\text{combine } \langle ꝩ_0.T \rangle\; (\,)] \longrightarrow T$$

$$[\text{combine } \langle ꝩ_2.T_1 \rangle\; (T_2 \;.\; T_3)] \longrightarrow$$
$$[\text{combine } [\text{combine } T_1\; T_2]\; T_3]$$

$\digamma_x$-calculus

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \mathbin{.} T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\ T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$$[\text{combine } \langle \digamma_0.T \rangle\ (\,)] \longrightarrow T$$

$$[\text{combine } \langle \digamma_2.T_1 \rangle\ (T_2 \mathbin{.} T_3)] \longrightarrow$$

$$[\text{combine } [\text{combine } T_1\ T_2]\ T_3]$$

$$[\text{combine } \langle T_0 \rangle\ (T_1\ \ldots\ T_n)] \longrightarrow$$

$\daleth_x$-calculus

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \mathbin{.} T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\ T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$$[\text{combine } \langle \daleth_0 . T \rangle\ (\,)] \longrightarrow T$$

$$[\text{combine } \langle \daleth_2 . T_1 \rangle\ (T_2 \mathbin{.} T_3)] \longrightarrow$$
$$[\text{combine } [\text{combine } T_1\ T_2]\ T_3]$$

$$[\text{combine } \langle T_0 \rangle\ (T_1\ \ldots\ T_n)] \longrightarrow$$
$$[\text{combine } T_0\ ([\text{eval } T_1]\ \ldots\ [\text{eval } T_n])]$$

ʔ_x-calculus

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \;.\; T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\; T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$$[\text{combine } \langle \text{ʔ}_0.T \rangle\; (\,)] \longrightarrow T$$

$$[\text{combine } \langle \text{ʔ}_2.T_1 \rangle\; (T_2 \;.\; T_3)] \longrightarrow$$
$$[\text{combine } [\text{combine } T_1\; T_2]\; T_3]$$

$$[\text{combine } \langle T_0 \rangle\; (T_1 \;\ldots\; T_n)] \longrightarrow$$
$$[\text{combine } T_0\; ([\text{eval } T_1] \;\ldots\; [\text{eval } T_n])]$$

$$[\text{combine } \langle \text{ʔ}x.T_1 \rangle\; T_2] \longrightarrow$$

$\daleth_x$-calculus

$$[\text{eval } S] \longrightarrow S$$

$$[\text{eval } (T_1 \mathbin{.} T_2)] \longrightarrow [\text{combine } [\text{eval } T_1]\ T_2]$$

$$[\text{eval } \langle T \rangle] \longrightarrow \langle [\text{eval } T] \rangle$$

$$[\text{combine } \langle \daleth_0 . T \rangle\ (\,)] \longrightarrow T$$

$$[\text{combine } \langle \daleth_2 . T_1 \rangle\ (T_2 \mathbin{.} T_3)] \longrightarrow$$
$$[\text{combine } [\text{combine } T_1\ T_2]\ T_3]$$

$$[\text{combine } \langle T_0 \rangle\ (T_1 \ \ldots\ T_n)] \longrightarrow$$
$$[\text{combine } T_0\ ([\text{eval } T_1]\ \ldots\ [\text{eval } T_n])]$$

$$[\text{combine } \langle \daleth x . T_1 \rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$\vau_x$-calculus

$$[\text{eval}\ S] \longrightarrow S$$

$$[\text{eval}\ (T_1\ .\ T_2)] \longrightarrow [\text{combine}\ [\text{eval}\ T_1]\ T_2]$$

$$[\text{eval}\ \langle T\rangle] \longrightarrow \langle[\text{eval}\ T]\rangle$$

$$[\text{combine}\ \langle\vau_0.T\rangle\ (\,)] \longrightarrow T$$

$$[\text{combine}\ \langle\vau_2.T_1\rangle\ (T_2\ .\ T_3)] \longrightarrow$$
$$[\text{combine}\ [\text{combine}\ T_1\ T_2]\ T_3]$$

$$[\text{combine}\ \langle T_0\rangle\ (T_1\ \ldots\ T_n)] \longrightarrow$$
$$[\text{combine}\ T_0\ ([\text{eval}\ T_1]\ \ldots\ [\text{eval}\ T_n])]$$

$$[\text{combine}\ \langle\vau x.T_1\rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$$((\lambda x.T_1)T_2) \longrightarrow T_1[x \leftarrow T_2]$$

$\vau_x$-calculus

$$T ::= x \mid c \mid \langle \vau x.T \rangle \mid [\text{combine}\ T\ T]$$

$$T ::= x \mid c \mid (\lambda x.T) \mid (TT)$$

$$[\text{combine}\ \langle \vau x.T_1 \rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$$((\lambda x.T_1)T_2) \longrightarrow T_1[x \leftarrow T_2]$$

$\vau_x$-calculus

$$T ::= x \mid c \mid \langle \vau x.T \rangle \mid [\text{combine}\ T\ T]$$

$$T ::= x \mid c \mid (\lambda x.T) \mid (TT)$$

$$[\text{combine}\ \langle \vau x.T_1 \rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$$((\lambda x.T_1)T_2) \longrightarrow T_1[x \leftarrow T_2]$$

$$\forall\, T_1 \in \Lambda,\quad T_1 \longrightarrow_{\vau x} T_2 \quad \text{iff} \quad T_1 \longrightarrow_{\beta} T_2$$

$\vau_x$-calculus

$$T ::= x \mid c \mid \langle \vau x.T \rangle \mid [\text{combine}\ T\ T]$$

$$T ::= x \mid c \mid (\lambda x.T) \mid (TT)$$

$$[\text{combine}\ \langle \vau x.T_1 \rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$$((\lambda x.T_1)T_2) \longrightarrow T_1[x \leftarrow T_2]$$

$$\forall\, T_1 \in \Lambda,\quad T_1 \longrightarrow_{\vau x} T_2 \quad \text{iff} \quad T_1 \longrightarrow_{\beta} T_2$$

ꝩ_x-calculus

$$T ::= x \mid c \mid \langle \text{ꝩ} x.T \rangle \mid [\text{combine}\ T\ T]$$

$$T ::= x \mid c \mid (\lambda x.T) \mid (TT)$$

$$[\text{combine}\ \langle \text{ꝩ} x.T_1 \rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$$((\lambda x.T_1)T_2) \longrightarrow T_1[x \leftarrow T_2]$$

$$\forall\, T_1 \in \Lambda,\quad T_1 \longrightarrow_{\text{ꝩ} x} T_2 \quad \text{iff} \quad T_1 \longrightarrow_{\beta} T_2$$

$\text{\textfrakturF}_x$-calculus

$$T ::= x \mid c \mid \langle \text{\textfrakturF} x.T \rangle \mid [\text{combine}\ T\ T]$$

$$T ::= x \mid c \mid (\lambda x.T) \mid (TT)$$

$$[\text{combine}\ \langle \text{\textfrakturF} x.T_1 \rangle\ T_2] \longrightarrow T_1[x \leftarrow T_2]$$

$$((\lambda x.T_1)T_2) \longrightarrow T_1[x \leftarrow T_2]$$

$$\forall\, T_1 \in \Lambda,\quad T_1 \longrightarrow_{\text{\textfrakturF} x} T_2 \ \text{ iff }\ T_1 \longrightarrow_{\beta} T_2$$

simulation

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3)) \longrightarrow_{\beta} ((\lambda x.(* \; x \; x)) \; 5)$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3)) \longrightarrow^{+}_{\beta} (* \; 5 \; 5)$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3)) \longrightarrow^{+}_{\beta} 25$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{combine} \; \langle \vau x.[\text{combine} * \boldsymbol{(} \; x \;\; x \; \boldsymbol{)}]\rangle \\ [\text{combine} + \boldsymbol{(} \; 2 \;\; 3 \; \boldsymbol{)}]]$$

simulation

$$((\lambda x.(*\ x\ x))\ (+\ 2\ 3))$$

$$[\text{combine}\ \langle \vau x.[\text{combine} * (\, x\ \ x\,)]\rangle$$
$$[\text{combine} + (\,2\ \ 3\,)]]$$

```
((lambda (x) (* x x)) (+ 2 3))
```

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{combine} \; \langle \vau x.[\text{combine} * \mathbf{(} \, x \;\; x \, \mathbf{)}]\rangle \; [\text{combine} + \mathbf{(} \, 2 \;\; 3 \, \mathbf{)}]]$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{combine} \; \langle \vau x.[\text{combine} * (x \;\; x)]\rangle \; [\text{combine} + (2 \;\; 3)]]$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow_{\vau}^{+} \texttt{25}$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

[eval `((lambda (x) (* x x)) (+ 2 3))` e_0]

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow_{\vau}$$

$$[\text{combine} \; [\text{eval} \; \texttt{(lambda(x)(* x x))} \; e_0] \; \texttt{((+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

[eval `((lambda (x) (* x x)) (+ 2 3))`
e_0] $\longrightarrow_{\jmath}$

[combine [eval `(lambda(x)(* x x))` e_0]
`((+ 2 3))`
e_0]

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\mathrm{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{ꝩ}$$

$$[\mathrm{combine} \; \langle\langle ꝩ_2.\langle ꝩ x.\langle ꝩ_0.[\mathrm{eval} \; \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle (e_0)]\rangle\rangle\rangle\rangle \; \texttt{((+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

[eval `((lambda (x) (* x x)) (+ 2 3))`
e_0] $\longrightarrow^{+}_{\vau}$

[combine $\langle\langle \vau_2 . \langle \vau x . \langle \vau_0 .$ [eval `(* x x)`
$\langle\!\langle$ `x` $\leftarrow x \rangle\!\rangle (e_0)] \rangle\rangle\rangle\rangle$

`((+ 2 3))`

e_0]

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\mathrm{eval}\, \texttt{((lambda (x) (* x x)) (+ 2 3))}\; e_0] \longrightarrow^{+}_{\vau}$$

$$[\mathrm{combine}\; \langle\langle \vau_2 . \langle \vau x . \langle \vau_0 . [\mathrm{eval}\, \texttt{(* x x)} \, \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle (e_0)] \rangle\rangle\rangle\rangle \; \texttt{((+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\imath}$$

$$[\text{combine} \; \langle\langle \imath_2 . \langle \imath x . \langle \imath_0 . [\text{eval} \; \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle (e_0)] \rangle\rangle\rangle\rangle \; \texttt{((+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\imath}$$

$$[\text{combine} \; \langle\langle \imath_2.\langle \imath x.\langle \imath_0.[\text{eval} \; \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle(e_0)]\rangle\rangle\rangle\rangle \; \texttt{((+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \, \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\imath}$$

$$[\text{combine} \; \langle\langle \imath_2 . \langle \imath x . \langle \imath_0 . [\text{eval} \, \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle (e_0)] \rangle\rangle\rangle\rangle \; \texttt{((+ 2 3))} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\text{ꝩ}}$$

$$[\text{combine} \; \langle \text{ꝩ}_2 . \langle \text{ꝩ}x . \langle \text{ꝩ}_0 . [\text{eval} \; \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle (e_0)] \rangle \rangle \rangle \; \texttt{(5)} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\imath}$$

$$[\text{combine} \; \langle \imath_2 . \langle \imath x . \langle \imath_0 . [\text{eval} \; \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle (e_0)] \rangle \rangle \rangle \; \texttt{(5)} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\vau}$$

$$[\text{combine} \; \langle \vau_2.\langle \vau x.\langle \vau_0.[\text{eval} \; \texttt{(* x x)} \; \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle(e_0)]\rangle\rangle\rangle \; \texttt{(5)} \; e_0]$$

simulation

$$((\lambda x.(*\ x\ x))\ (+\ 2\ 3))$$

$$[\text{eval}\ \texttt{((lambda (x) (* x x)) (+ 2 3))}\ e_0] \longrightarrow^{+}_{\imath}$$

$$[\text{combine}\ \langle \imath_2.\langle \imath x.\langle \imath_0.[\text{eval}\ \texttt{(* x x)}\ \langle\!\langle \texttt{x} \leftarrow x \rangle\!\rangle(e_0)]\rangle\rangle\rangle\ \texttt{(5)}\ e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

[eval `((lambda (x) (* x x)) (+ 2 3))`

$e_0] \longrightarrow^{+}_{ꝩ}$

[combine $\langle ꝩ_2.\langle ꝩx.\langle ꝩ_0.$[eval `(* x x)`

$\langle\!\langle$`x`$\leftarrow x\rangle\!\rangle(e_0)]\rangle\rangle\rangle$

`(5)`

$e_0]$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\vau}$$

$$[\text{combine} \; \langle \vau_2.\langle \vau x.\langle \vau_0.[\text{combine} \; * \; \texttt{(} x \; x \texttt{)} \; \langle\!\langle\rangle\!\rangle] \rangle\rangle\rangle \; \texttt{(5)} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\mathrm{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\digamma}$$

$$[\mathrm{combine} \; [\mathrm{combine} \; \langle \digamma x.\langle \digamma_0.[\mathrm{combine} * \texttt{(} x \; x \texttt{)} \, \langle\!\langle\rangle\!\rangle]\rangle\rangle \; \texttt{5} \; e_0] \; \texttt{()} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\text{ꝩ}}$$

$$[\text{combine} \; \langle \text{ꝩ}_0.[\text{combine} * \texttt{(5 5)} \langle\!\langle\rangle\!\rangle] \rangle \; \texttt{( )} \; e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\text{eval}\,\texttt{((lambda (x) (* x x)) (+ 2 3))}\; e_0] \longrightarrow^{+}_{ʔ}$$

$$[\text{combine}\;\langle ʔ_0.\texttt{25}\rangle\;\texttt{()}\;e_0]$$

simulation

$$((\lambda x.(* \; x \; x)) \; (+ \; 2 \; 3))$$

$$[\mathrm{eval} \; \texttt{((lambda (x) (* x x)) (+ 2 3))} \; e_0] \longrightarrow^{+}_{\digamma}$$

`25`

conclusion

conclusion

- implicit evaluation

conclusion

- implicit evaluation $\qquad \text{eval} = \longrightarrow^*$

conclusion

- implicit evaluation $\qquad \text{eval} = \longrightarrow^*$
  - fexprs $\Rightarrow$ trivial theory

conclusion

- implicit evaluation $\quad \text{eval} = \longrightarrow^*$
  - fexprs $\Rightarrow$ trivial theory
- explicit evaluation

conclusion

- implicit evaluation $\quad \text{eval} = \longrightarrow^*$
  - — fexprs $\Rightarrow$ trivial theory
- explicit evaluation $\quad \text{eval} \subset \longrightarrow^*$

conclusion

- implicit evaluation $\quad \text{eval} = \longrightarrow^*$
  - — fexprs $\Rightarrow$ trivial theory
- explicit evaluation $\quad \text{eval} \subset \longrightarrow^*$
  - — fexprs + nontrivial theory

conclusion

- implicit evaluation $\quad \text{eval} = \longrightarrow^*$
  - — fexprs $\Rightarrow$ trivial theory
- explicit evaluation $\quad \text{eval} \subset \longrightarrow^*$
  - — fexprs + nontrivial theory
- ϝ-calculus

conclusion

- implicit evaluation $\quad \text{eval} = \longrightarrow^*$
  — fexprs $\Rightarrow$ trivial theory
- explicit evaluation $\quad \text{eval} \subset \longrightarrow^*$
  — fexprs + nontrivial theory
- ϑ-calculus contains λ-calculus

conclusion

- implicit evaluation $\quad \text{eval} = \longrightarrow^*$
  — fexprs $\Rightarrow$ trivial theory
- explicit evaluation $\quad \text{eval} \subset \longrightarrow^*$
  — fexprs + nontrivial theory
- $\vau$-calculus contains λ-calculus
- λ-calculus is about fexprs

conclusion

- implicit evaluation $\qquad \text{eval} = \longrightarrow^*$

  — fexprs $\Rightarrow$ trivial theory

- explicit evaluation $\qquad \text{eval} \subset \longrightarrow^*$

  — fexprs + nontrivial theory

- $\vau$-calculus contains λ-calculus

- λ-calculus is about fexprs

`http://www.cs.wpi.edu/~jshutt/kernel.html`