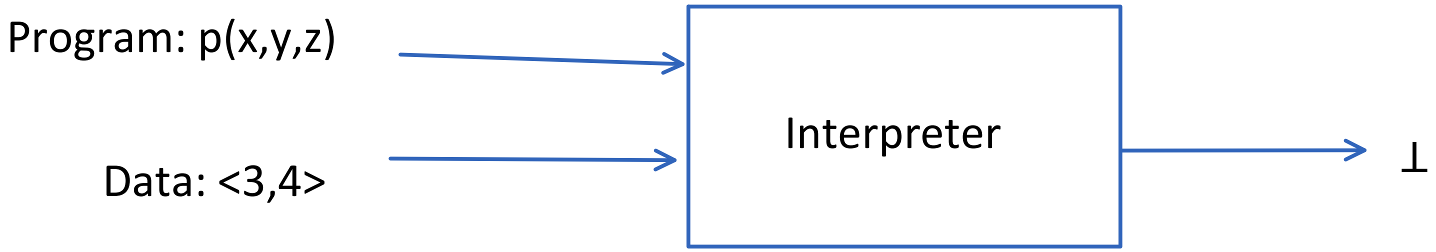
Partial evaluation

Monday, March 23, 2020 10:21 AM

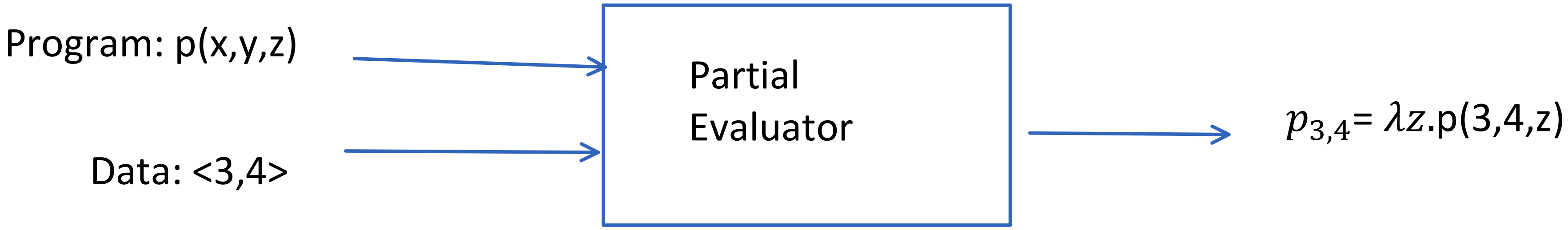


Program: p(x,y,z)

Data: <3,4>

Interpreter

⊥



Program: p(x,y,z)

Data: <3,4>

Partial Evaluator

$p_{3,4} = \lambda z.$p(3,4,z)

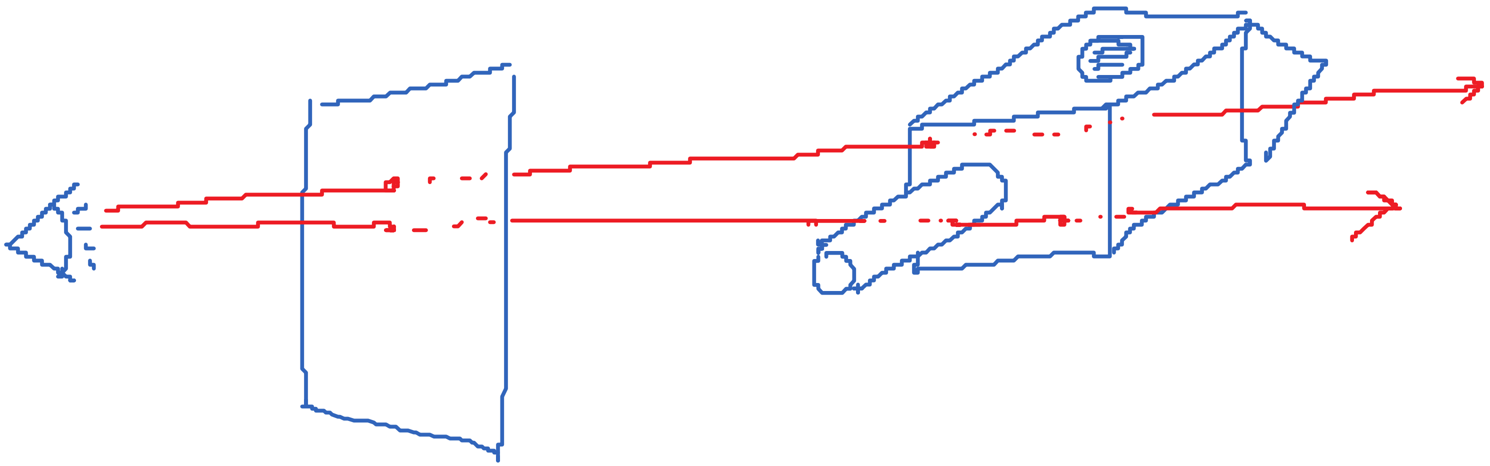
p(x,y,z) → Subject program

$p_{3,4}$

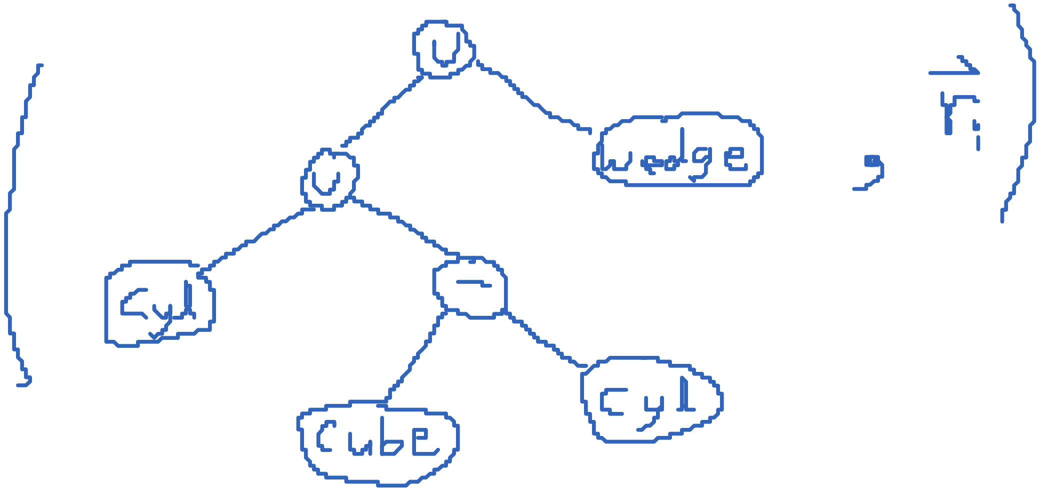
Residual program

Ray tracing of solid models

Monday, March 23, 2020 10:34 AM



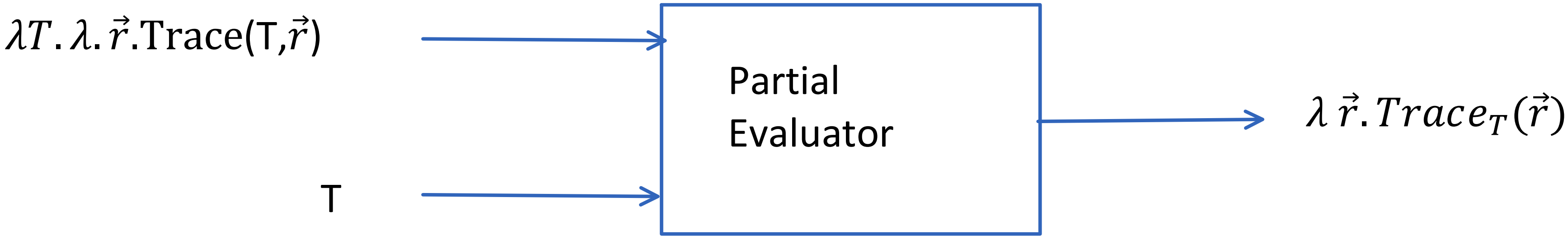
Trace



```
for each ray r⃗ do
    Trace (T,r⃗)
    Display
od
```

Ray Tracing and Partial Evaluation

Monday, March 23, 2020   10:43 AM



$\lambda T.\lambda.\vec{r}.\mathrm{Trace}(\mathrm{T},\vec{r})$

T

Partial Evaluator

$\lambda\,\vec{r}.Trace_T(\vec{r})$

```
for each ray $\vec{r}$ do
   $Trace_T(\vec{r})$
   Display
od
```

Other examples:

For many blocks b: Encrypt(key, b)

For many strings s: Write(fd, s)

For many substrings of s: Match(pat, s)

For many programs p, Parse(grammar, p)

- a hint that PE is relevant to compiling
- bison, yacc, etc.

Partial Evaluation Themes

Monday, March 23, 2020 10:53 AM

- Reducing/eliminating interpretation overhead
- Transforming interpretation of data into control state
    - Correctness issue: need to be able to match states in subject program with states in the residual program
- Rate of change of arguments
- Language translation
- Uniform approach to several compiler optimizations
    - constant folding
    - loop unrolling
- An account of compiling (translation) and compiler generation

Some Notation

Monday, March 23, 2020 11:14 AM

$$[\![p]\!]_L\,[i] = o$$

$$[\![p]\!]\,[\underset{\substack{\uparrow\\ \text{static}\\ \text{(or supplied)}}}{s},\ \underset{\substack{\nwarrow\\ \text{dynamic (or delayed)}}}{d}] = v$$

$$[\![pe]\!]\,[p,\ s] = p_s\,,\quad \text{s.t.}\quad [\![p_s]\!]\,[d] = [\![p]\!]\,[s,d] = v$$

Futamura Projections

Monday, March 23, 2020   11:20 AM

(0) $[\![pe]\!][p, s] = p_s$ s.t. $[\![p_s]\!][d] = [\![p]\!][s, d]$

$$[\![pe^{\overset{L1}{\cancel{L1 \to L1}}}_{L}]\!]_L [p_{L1}, s] = (p_s)_{\boxed{L1}}$$

?

(1) $[\![int]\!][q, i] = [\![q]\!][i]$

$$[\![int^{L1}_{L}]\!]_L [q_{L1}, i] = [\![q_{L1}]\!]_{L1} [i]$$

(2) $[\![pe]\!][int, q] = int_q$ s.t. $[\![int_q]\!][i] = [\![int]\!][q, i] = [\![q]\!][i]$

$$[\![pe^{L1}_{L}]\!]_L [int^{L2}_{L1}, q_{L2}] = (int_q)_{L1}$$

$$[\![(int_q)_{L1}]\!]_{L1} [i] = [\![int^{L2}_{L1}]\!]_{L1} [q_{L2}, i] = [\![q_{L2}]\!]_{L2} [i]$$

1st Futamura projection

L1

=

L2

translation/compiling

More Futamura Projections

Monday, March 23, 2020 11:34 AM

(3) $[\![pe]\!][pe, int] = pe_{int}$ s.t. $[\![\underline{pe_{int}}]\!][q] = [\![pe]\!][int, q] = int_q$ (a compiled/translated version of q)

2nd Futamura projection

compiler ↓ input: program q

output: compiled version of q

$$[\![pe^{L1}_{L}]\!]\left[pe^{\boxed{L2}}_{\boxed{L1}}, int^{\boxed{L3}}_{\boxed{L2}}\right] = (pe_{int})^{?\,\boxed{L3 \to L2}}_{\boxed{L1}}$$

$$[\![(pe_{int})^{L3\to L2}_{L1}]\!]_{L}[q_{\underline{L3}}] = [\![\underset{\updownarrow}{pe^{L2}_{L1}}]\!]_{L1}[int^{L3}_{L2}, q_{L3}]$$

$$= (int_q)_{\underline{L2}}$$

(4) $[\![pe]\!][pe, pe] = pe_{pe}$ s.t. $[\![\underline{pe_{pe}}]\!][int] = [\![pe]\!][pe, int]$

$= pe_{int}$

$=$ compiler

3rd Futamura projection

compiler-compiler

lang. Spec.

Futamura projections as a right-shift

Monday, March 23, 2020 11:41 AM

$$\begin{tikzcd}
\llbracket int \rrbracket \arrow[dr] & {[q,} \arrow[dr] & {i]} & & \\
\llbracket pe \rrbracket \arrow[dr] & {[int,} \arrow[dr] & {q]} & = int_q & \text{compiling} \\
\llbracket pe \rrbracket \arrow[dr] & {[pe,} \arrow[dr] & {int]} & = pe_{int} & \text{compiler} \\
\llbracket pe \rrbracket & {[pe,} & {pe]} & = pe_{pe} & \text{compiler-compiler}
\end{tikzcd}$$

Does a partial evaluator exist?

Monday, March 23, 2020 12:12 PM

λ- calculus

$$\lambda x.\ \overbrace{\cdots\ x\ \cdots}^{\text{scope}}$$

a λ-term defines a 1-argument anonymous function

$$(\lambda x.\ M)(N) = M[x \leftarrow N]$$

$$\lambda \langle x,y \rangle.\ \cdots$$

$$\lambda x.\lambda y.\ \cdots$$

$$pe =_{df} \lambda h.\ \lambda x.\ \lambda y.\ h(\langle x,y \rangle)$$

$$pe\ \ f\ \ s \to (\lambda h.\lambda x.\lambda y.\ h(\langle x,y\rangle))\ \ f\ \ s = \lambda y.\ f(\langle s,y\rangle)$$

trivial partial evaluator

challenge: create a non-trivial partial evaluator (next few lectures)