Relating Staged Computation to the Record Calculus

Baris Aktemur

Özyeğin University, Turkey

Context

Program generation with quotations/antiquotations.

$$\begin{array}{l}\text{let } pow = \text{fix}\, f(n).\, \lambda y.\text{if } n = 0 \text{ then } 1 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{else } y \times f(n-1)y \\ \text{in } pow\ 3\end{array}$$

$$\begin{array}{l}\text{let } gpow = \text{fix}\, f(n).\, \text{if } n = 0 \text{ then } \langle 1 \rangle \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{else } \langle y \times \text{`}(f(n-1)) \rangle \\ \text{in run}\langle \lambda y.\, \text{`}(gpow\ 3) \rangle\end{array}$$

Context

Program generation with quotations/antiquotations.

$$\begin{array}{l}\text{let } pow = \text{fix}\, f(n).\, \lambda y.\text{if } n = 0 \text{ then } 1 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{else } y \times f(n-1)y \\ \text{in } pow\ 3\end{array}$$

$$\begin{array}{l}\text{let } gpow = \text{fix}\, f(n).\, \text{if } n = 0 \text{ then } \langle 1 \rangle \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{else } \langle y \times \text{`}(f(n-1)) \rangle \\ \text{in run}\langle \lambda y.\, \text{`}(gpow\ 3) \rangle \\ \qquad\qquad \overbrace{\langle y \times y \times y \times 1 \rangle}\end{array}$$

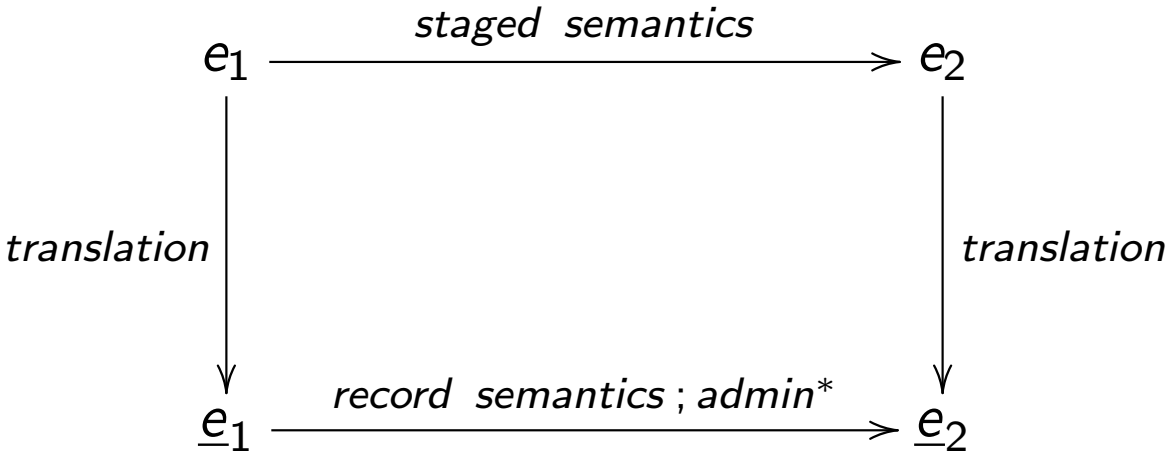
Context

Program generation with quotations/antiquotations.

$$\begin{array}{l}\text{let } pow = \text{fix}\, f(n).\, \lambda y.\text{if } n = 0 \text{ then } 1 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{else } y \times f(n-1)y \\ \text{in } pow\ 3\end{array}$$

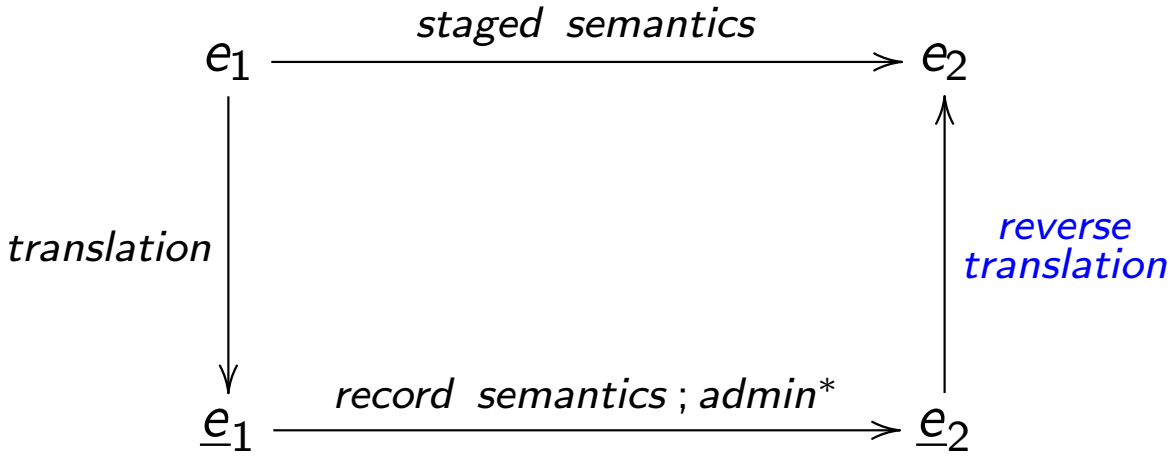
$$\begin{array}{l}\text{let } gpow = \text{fix}\, f(n).\, \text{if } n = 0 \text{ then } \langle 1 \rangle \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{else } \langle y \times \text{`}(f(n-1)) \rangle \\ \text{in } \underbrace{\text{run}\langle \lambda y.\, \text{`}(gpow\ 3) \rangle}_{\lambda y.y \times y \times y \times 1}\end{array}$$

Previously



- record calculus = lambda calculus + records
- *admin** = unrestricted side-effect-free reductions
- Benefit: Reuse a record type system to type-check staged expressions.

Today



- record calculus = lambda calculus + records
- *admin** = two kinds of reductions applied exhaustively
- Why: Reuse existing record calculus analyses (data-flow analysis, abstract interpretation, etc) to reason about program generators.

Outline

Staged vs. Record Calculus

$$\langle 2 + 3 \rangle \quad \lambda\rho.2 + 3$$

Staged vs. Record Calculus

$$\langle 2 + 3 \rangle \quad \lambda\rho.2 + 3$$

$$\langle x + 3 \rangle \quad \lambda\rho.\rho\cdot\mathtt{x} + 3$$

Staged vs. Record Calculus

$$\langle 2 + 3 \rangle \quad \lambda\rho.2 + 3$$

$$\langle x + 3 \rangle \quad \lambda\rho.\rho\cdot\mathtt{x} + 3$$

$$\langle \text{`}(c) + 3 \rangle \quad \lambda\rho.c(\rho) + 3$$

Staged vs. Record Calculus

$$
\begin{array}{rl}
\langle 2 + 3 \rangle & \lambda\rho.2 + 3 \\
\langle x + 3 \rangle & \lambda\rho.\rho{\cdot}\mathtt{x} + 3 \\
\langle \text{`}(c) + 3 \rangle & \lambda\rho.c(\rho) + 3 \\
\langle \lambda x.\, \text{`}(c) + 3 \rangle & \lambda\rho.\lambda x.c(\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) + 3
\end{array}
$$

Staged vs. Record Calculus

$$
\begin{array}{rl}
\langle 2 + 3 \rangle & \lambda\rho.2 + 3 \\
\langle x + 3 \rangle & \lambda\rho.\rho\cdot\mathtt{x} + 3 \\
\langle \,\text{`}(c) + 3 \rangle & \lambda\rho.c(\rho) + 3 \\
\langle \lambda x.\,\text{`}(c) + 3 \rangle & \lambda\rho.\lambda x.c(\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) + 3 \\
\mathsf{run}(\langle 2 + 3 \rangle) & (\lambda\rho.2 + 3)\{\}
\end{array}
$$

Staged vs. Record Calculus

$$
\begin{array}{rl}
\langle 2+3 \rangle & \lambda\rho.2+3 \\
\langle x+3 \rangle & \lambda\rho.\rho\cdot\mathtt{x}+3 \\
\langle \,\grave{}(c)+3 \rangle & \lambda\rho.c(\rho)+3 \\
\langle \lambda x.\,\grave{}(c)+3 \rangle & \lambda\rho.\lambda x.c(\rho \,\mathsf{with}\, \{\mathtt{x}=x\})+3 \\
\mathsf{run}(\langle 2+3 \rangle) & (\lambda\rho.2+3)\{\}
\end{array}
$$

$$
\begin{array}{rll}
\langle \ldots \,\grave{}(e) \ldots \rangle & (\lambda\rho.\ldots \underline{e}(\rho) \ldots) & \text{vs.} \\
 & (\lambda h.\lambda\rho.\ldots h(\rho) \ldots)\underline{e} &
\end{array}
$$

Related Work

- Davies and Pfenning [JACM 2001]
  - Translation that makes the order of evaluation explicit.
- Kameyama, Kiselyov, Shan [PEPM 2008]:
  - Translation to System F with tuples.
- Aktemur [2009]

Syntax

$$
\begin{aligned}
e \in \mathit{Exp} ::= {} & c \mid x \mid e\, e \\
& \mid \lambda x.e \mid \mathsf{fix}\, f(x).\, e \mid \mathsf{let}\, x = e\, \mathsf{in}\, e \\
& \mid \langle e \rangle \mid {\sim}(e) \mid \mathsf{run}(e) \mid \mathsf{lift}(e) \\
& \mid \ell \mid \mathsf{ref}\ e \mid !e \mid e{:=}e
\end{aligned}
$$

$$
\begin{aligned}
& \rho \in \mathit{RVar} \\
& w \in \mathit{Var} \cup \mathit{RVar} \\
e \in \mathit{RExp} ::= {} & c \mid x \mid \rho \mid e\, e \\
& \mid \lambda w.e \mid \mathsf{fix}\, f(x).\, e \mid \mathsf{let}\, x = e\, \mathsf{in}\, e \\
& \mid \{\} \mid e\, \mathsf{with}\, \{a = e\} \mid e{\cdot}a \\
& \mid \ell \mid \mathsf{ref}\ e \mid !e \mid e{:=}e
\end{aligned}
$$

Translation

- $\langle \ldots \lambda x.\, \backprime(e) \ldots \rangle$ translates to
  $(\lambda h.\lambda \rho. \ldots \lambda x.h(\rho \text{ with } \{\mathtt{x} = x\}) \ldots)\underline{e}$

Translation

- $\langle \ldots \lambda x.\, \backprime(e) \ldots \rangle$ translates to $(\lambda h.\lambda \rho. \ldots \lambda x.h(\rho \,\text{with}\, \{\mathtt{x} = x\}) \ldots)\underline{e}$
- A hole corresponds to a context:

$$K \in \mathit{Context} \quad ::= \quad (\lambda h.[\,])e$$

Translation

- $\langle \ldots \lambda x.\, \backtick(e) \ldots \rangle$ translates to
  $(\lambda h.\lambda \rho. \ldots \lambda x.h(\rho \text{ with } \{\mathtt{x} = x\}) \ldots)\underline{e}$
- A hole corresponds to a context:

$$K \in \mathit{Context} \quad ::= \quad (\lambda h.[\,])e \mid (\lambda h.K)e$$

- $\langle \ldots \backtick(e_1) \ldots \backtick(e_2) \ldots \rangle$ translates to
  $(\lambda h_1.(\lambda h_2.\lambda \rho. \ldots h_1(\rho) \ldots h_2(\rho) \ldots)\underline{e}_2)\underline{e}_1$

Translation

- $\langle \ldots \lambda x.\, \backprime(e) \ldots \rangle$ translates to
  $(\lambda h.\lambda \rho. \ldots \lambda x.h(\rho \text{ with } \{\mathtt{x} = x\}) \ldots)\underline{e}$
- A hole corresponds to a context:

$$K \in \mathit{Context} \quad ::= \quad (\lambda h.[\,])e \mid (\lambda h.K)e$$

- $\langle \ldots \backprime(e_1) \ldots \backprime(e_2) \ldots \rangle$ translates to
  $(\lambda h_1.(\lambda h_2.\lambda \rho. \ldots h_1(\rho) \ldots h_2(\rho) \ldots)\underline{e_2})\underline{e_1}$
- Antiquotations can be arbitrarily nested. A list of contexts:

$$\kappa_1^j = K_1 :: K_2 :: \ldots :: K_j$$

Translation

- $\langle \ldots \lambda x.\, \backprime(e) \ldots \rangle$ translates to
  $(\lambda h.\lambda \rho. \ldots \lambda x.h(\rho \text{ with } \{\mathtt{x} = x\}) \ldots)\underline{e}$
- A hole corresponds to a context:

$$K \in \mathit{Context} \quad ::= \quad (\lambda h.[\,])e \mid (\lambda h.K)e$$

- $\langle \ldots \backprime(e_1) \ldots \backprime(e_2) \ldots \rangle$ translates to
  $(\lambda h_1.(\lambda h_2.\lambda \rho. \ldots h_1(\rho) \ldots h_2(\rho) \ldots)\underline{e_2})\underline{e_1}$
- Antiquotations can be arbitrarily nested. A list of contexts:

$$\kappa_1^j = K_1 :: K_2 :: \ldots :: K_j$$

- $R \in \mathit{Env} ::= \{\} \mid \rho \mid R \text{ with } \{\mathtt{x} = x\}$

Translation

- $[\![e]\!]_{R_0,\ldots,R_n} = (\underline{e}, K_1 :: K_2 :: \ldots :: K_j)$

$$\frac{[\![e]\!]_{R_0,\ldots,R_n,\rho} = (\underline{e}, \mathtt{nil}) \quad \rho \in \mathit{RVar} \text{ is fresh}}{[\![\langle e \rangle]\!]_{R_0,\ldots,R_n} = (\lambda\rho.\underline{e}, \mathtt{nil})}$$

$$\frac{[\![e]\!]_{R_0,\ldots,R_n,\rho} = (\underline{e}, K :: \kappa) \quad \rho \in \mathit{RVar} \text{ is fresh}}{[\![\langle e \rangle]\!]_{R_0,\ldots,R_n} = (K[\lambda\rho.\underline{e}], \kappa)}$$

$$\frac{[\![e]\!]_{R_0,\ldots,R_n} = (\underline{e}, \kappa) \quad h \in \mathit{Var} \text{ is fresh}}{[\![\text{`}(e)]\!]_{R_0,\ldots,R_n,R_{n+1}} = (h(R_{n+1}), ((\lambda h.[\,])\underline{e}) :: \kappa)}$$

$$\frac{[\![e_1]\!]_{R_0,\ldots,R_n} = (\underline{e}_1, \kappa) \quad [\![e_2]\!]_{R_0,\ldots,R_n} = (\underline{e}_2, \kappa')}{[\![e_1\, e_2]\!]_{R_0,\ldots,R_n} = (\underline{e}_1\underline{e}_2, \mathit{zip}(\kappa, \kappa'))}$$

- $\mathit{Close}([\![e]\!]_{R_0,\ldots,R_n}) = K_j[\cdots K_2[K_1[\underline{e}]]\cdots]$ gives a closed expression.

Admin Reductions

$$\langle \lambda x.\, `(\langle x \rangle) + 1 \rangle \xrightarrow{0} \langle \lambda x.x + 1 \rangle$$

Admin Reductions

$$\langle \lambda x.\, `(\langle x \rangle) + 1 \rangle \xrightarrow{0} \langle \lambda x.x + 1 \rangle$$

$$(\lambda h.\lambda \rho.\lambda x.h(\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) + 1)(\lambda \rho.\rho \cdot \mathtt{x})$$

$$\xrightarrow{\mathcal{R}} (\lambda \rho.\lambda x.(\lambda \rho.\rho \cdot \mathtt{x})(\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) + 1)$$

Admin Reductions

$$\langle \lambda x.\, `(\langle x \rangle) + 1 \rangle \xrightarrow{0} \langle \lambda x.x + 1 \rangle$$

$$(\lambda h.\lambda \rho.\lambda x.h(\rho \text{ with } \{\mathtt{x} = x\}) + 1)(\lambda \rho.\rho \cdot \mathtt{x})$$

$$\xrightarrow{\mathcal{R}} (\lambda \rho.\lambda x.(\lambda \rho.\rho \cdot \mathtt{x})(\rho \text{ with } \{\mathtt{x} = x\}) + 1)$$

Admin Reductions

$$\langle \lambda x.\, \backprime(\langle x \rangle) + 1 \rangle \xrightarrow{0} \langle \lambda x.x + 1 \rangle$$

$$(\lambda h.\lambda \rho.\lambda x.h(\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) + 1)(\lambda \rho.\rho \cdot \mathtt{x})$$

$$\xrightarrow{\mathcal{R}} (\lambda \rho.\lambda x.(\lambda \rho.\rho \cdot \mathtt{x})(\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) + 1)$$

$$\xrightarrow{\mathcal{A}} (\lambda \rho.\lambda x.((\rho \,\mathsf{with}\, \{\mathtt{x} = x\}) \cdot \mathtt{x}) + 1)$$

Admin Reductions

$$\langle \lambda x.\, `(\langle x \rangle) + 1 \rangle \xrightarrow{0} \langle \lambda x.x + 1 \rangle$$

$$(\lambda h.\lambda \rho.\lambda x.h(\rho \text{ with } \{\mathtt{x} = x\}) + 1)(\lambda \rho.\rho \cdot \mathtt{x})$$

$$\xrightarrow{\mathcal{R}} (\lambda \rho.\lambda x.(\lambda \rho.\rho \cdot \mathtt{x})(\rho \text{ with } \{\mathtt{x} = x\}) + 1)$$

$$\xrightarrow{\mathcal{A}} (\lambda \rho.\lambda x.((\rho \text{ with } \{\mathtt{x} = x\}) \cdot \mathtt{x}) + 1)$$

$$\xrightarrow{\mathcal{A}} (\lambda \rho.\lambda x.x + 1)$$

Admin Reductions

$$(\lambda\rho.e)R \xrightarrow{\mathcal{A}} e[\rho\backslash R]$$
$$R\cdot \mathtt{x} \xrightarrow{\mathcal{A}} R(\mathtt{x})$$

Semantic Relation

Theorem

$Close([\![e]\!]_{R_0,\ldots,R_n})$ is in admin-normal form.

Theorem

If $e_1 \xrightarrow{n} e_2$, then $Close([\![e_2]\!]_{\vec{R}_n}) \xrightarrow{\mathcal{R};\mathcal{A}^*} Close([\![e_2]\!]_{\vec{R}_n})$ where the admin reductions are exhaustive.

$$\begin{array}{ccc} e_1 & \xrightarrow{\quad n \quad} & e_2 \\ {\scriptstyle [\![\cdot]\!]_{\vec{R}_n}}\downarrow & & \downarrow{\scriptstyle [\![\cdot]\!]_{\vec{R}_n}} \\ \underline{e}_1 & \xrightarrow{\quad \mathcal{R};\mathcal{A}^* \quad} & \underline{e}_2 \end{array}$$

Reverse Translation

- Convert record expressions back to staged.

$$
\begin{array}{rcl}
\lambda x.x & \leftrightarrow & \lambda x.x \\
\langle 0 \rangle & \leftrightarrow & \lambda \rho.0 \\
\langle x \rangle & \leftrightarrow & \lambda \rho.\rho \cdot x \\
\langle \backprime(\langle 3 \rangle) + 4 \rangle & \leftrightarrow & (\lambda h.\lambda \rho.h(\rho) + 4)(\lambda \rho.3)
\end{array}
$$

- Annotate lambda bindings to distinguish ordinary variables from bindings for holes: $\lambda x.x \leftrightarrow \lambda_x x.x$
- $[\![\underline{e}, \mathcal{H}]\!]^{-1} = e$, where $\mathcal{H}$ is a function $\{h_i : e_i\}$.

Definition

$$[\![\rho \cdot \mathrm{x}, \mathcal{H}]\!]^{-1} = x$$

$$[\![\lambda_x x.e, \mathcal{H}]\!]^{-1} = \lambda x.[\![e, \mathcal{H}]\!]^{-1}$$

$$[\![(\lambda \rho.e), \mathcal{H}]\!]^{-1} = \langle [\![e, \mathcal{H}]\!]^{-1} \rangle$$

$$[\![(\lambda h.e)e', \mathcal{H}]\!]^{-1} = [\![e, (\mathcal{H} \cup \{h : e'\})]\!]^{-1}$$

$$[\![h\ R, \mathcal{H}]\!]^{-1} = \text{`}([\![\mathcal{H}(h), \mathcal{H}]\!]^{-1})$$

$$[\![e_1\, e_2, \mathcal{H}]\!]^{-1} = [\![e_1, \mathcal{H}]\!]^{-1}\ [\![e_2, \mathcal{H}]\!]^{-1}$$

if $\nexists e'$ such that $e_1 = \lambda h.e'$, and $e_2 \notin \mathit{Env}$.

- A context $(\lambda h.[\,])e$ corresponds to a function $\{h : e\}$. So, a context list κ is converted to a function:

$$\begin{aligned}
\overline{\texttt{nil}} &= \varnothing \\
\overline{(\lambda h.[\,])e :: \kappa'} &= \{h : e\} \cup \overline{\kappa'} \\
\overline{(\lambda h.K)e :: \kappa'} &= \{h : e\} \cup \overline{K :: \kappa'}
\end{aligned}$$

Theorem

Let $[\![e]\!]_{\vec{R}_n} = (\underline{e}, \kappa)$. *Then* $[\![\underline{e}, \mathcal{H}]\!]^{-1} = e$ *for any* $\mathcal{H} \supseteq \overline{\kappa}$.

Corollary:

- $[\![\underline{e}, \overline{\kappa}]\!]^{-1} = e$
- $[\![\mathit{Close}([\![e]\!]_{\vec{R}_n}), \varnothing]\!]^{-1} = e$

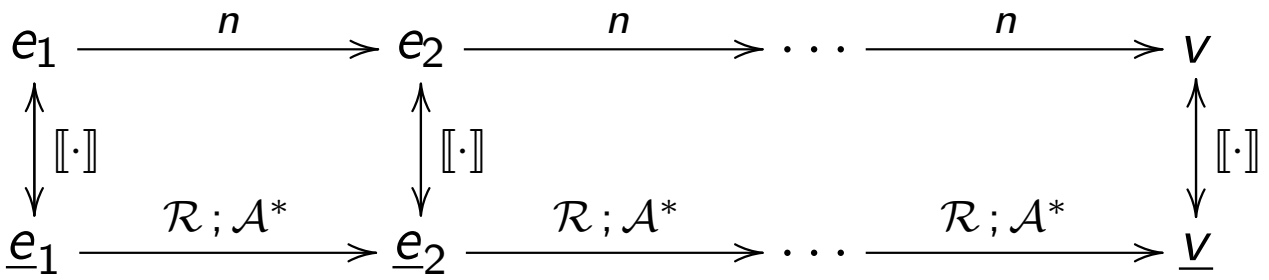
Semantic Relation

Corollary

If $e_1 \xrightarrow{n} e_2$, *then* $\mathit{Close}(\llbracket e_1 \rrbracket_{\vec{R}_n}) \xrightarrow{\mathcal{R}\,;\mathcal{A}^*\,;\llbracket\cdot\rrbracket^{-1}} e_2$, *where the admin reductions are exhaustive.*

$$\begin{array}{ccc} e_1 & \xrightarrow{\quad n \quad} & e_2 \\ {\scriptstyle \llbracket\cdot\rrbracket_{\vec{R}_n}}\downarrow & & \uparrow{\scriptstyle \llbracket\cdot\rrbracket^{-1}} \\ \underline{e}_1 & \xrightarrow{\quad \mathcal{R}\,;\mathcal{A}^* \quad} & \underline{e}_2 \end{array}$$

Discussion



- Idea: use record calculus analyses to reason about staged programs
- Beware: The relation is unidirectional.
  - $\langle 42 \rangle 3$ is stuck.
  - $(\lambda\rho.42)3 \xrightarrow{\mathcal{R}} 42$
- Distinguishing record variables in the beta-reduction yields an iff relation (I think).

$$\frac{v \text{ is a record value}}{(\lambda\rho.e)v \xrightarrow{\mathcal{R}} e[\rho \backslash v]} \qquad \frac{v \text{ is not a record value}}{(\lambda x.e)v \xrightarrow{\mathcal{R}} e[x \backslash v]}$$

(Reverse) Translation in Action

$$
\begin{array}{l}
\text{let } pow = \text{fix}\, gen(n).\, \text{if } n = 0 \text{ then } \langle 1 \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{else } \langle y \times \text{`}(gen(n-1)) \rangle \\
\text{in run}\langle \lambda y.\, \text{`}(pow\ 3) \rangle
\end{array}
$$

$$
\begin{array}{l}
\text{let } pow = \text{fix}\, gen(n).\, \text{if } n = 0 \text{ then } \lambda\rho.1 \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{else } (\lambda h.(\lambda\rho.\ \rho\cdot \mathrm{y} \times h\ \rho))(gen(n-1)) \\
\text{in let } h' = (\lambda h.\lambda\rho.\lambda y.h\ (\rho \,\text{with}\, \{\mathrm{y} = y\})) \\
\qquad\qquad\quad (pow\ 3) \\
\quad \text{in } h'\ \{\}
\end{array}
$$

(Reverse) Translation in Action

$$
\begin{array}{l}
\text{let } \mathit{pow} = \text{fix}\, \mathit{gen}(n).\, \text{if } n = 0 \text{ then } \langle 1 \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{else } \langle y \times \text{`}(\mathit{gen}(n-1)) \rangle \\
\text{in run}\langle \lambda y.\, \text{`}(\mathit{pow}\ 3) \rangle
\end{array}
$$

$$
\begin{array}{l}
\text{let } \mathit{pow} = \text{fix}\, \mathit{gen}(n).\, \text{if } n = 0 \text{ then } \lambda\rho.1 \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{else } (\lambda h.(\lambda\rho.\ \rho \cdot \mathrm{y} \times h\ \rho))(\mathit{gen}(n-1)) \\
\text{in let } h' = (\lambda h.\lambda\rho.\lambda y.h\ (\rho \,\text{with}\, \{\mathrm{y} = y\})) \\
\qquad\qquad\quad (\mathit{pow}\ 3) \\
\quad \text{in } h'\ \{\}
\end{array}
$$

(Reverse) Translation in Action

$$\mathsf{run}\langle \lambda y.\, \text{`}(\langle y \times \text{`}(\langle y \times \text{`}(\langle y \times \text{`}(\langle 1 \rangle)\rangle)\rangle)\rangle)\rangle$$

$$\begin{aligned}
&\mathsf{let}\ h' = (\lambda h.\lambda\rho.\lambda y.h\ (\rho\ \mathsf{with}\ \{\mathrm{y} = y\})) \\
&\qquad\qquad \big((\lambda h.\lambda\rho.\rho\cdot\mathrm{y}\times h\rho)((\lambda h.\lambda\rho.\rho\cdot\mathrm{y}\times h\rho)((\lambda h.\lambda\rho.\rho\cdot\mathrm{y}\times h\rho)(\lambda\rho.1)))\big) \\
&\mathsf{in}\ h'\ \{\}
\end{aligned}$$

$$\langle y \times \text{`}(\langle y \times \text{`}(\langle y \times \text{`}(\langle 1 \rangle)\rangle)\rangle)\rangle$$
$$\xrightarrow{0} \langle y \times \text{`}(\langle y \times \text{`}(\langle y \times 1 \rangle)\rangle)\rangle$$
$$\xrightarrow{0} \langle y \times \text{`}(\langle y \times y \times 1 \rangle)\rangle$$
$$\xrightarrow{0} \langle y \times y \times y \times 1 \rangle$$

$$(\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.1)))$$
$$\xrightarrow{\mathcal{R}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times (\lambda\rho.1)\ \rho))$$
$$\xrightarrow{\mathcal{A}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times 1))$$
$$\xrightarrow{\mathcal{R}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times (\lambda\rho.\rho\cdot \mathrm{y} \times 1)\ \rho)$$
$$\xrightarrow{\mathcal{A}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times 1)$$
$$\xrightarrow{\mathcal{R}} (\lambda\rho.\rho\cdot \mathrm{y} \times (\lambda\rho.\rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times 1)\ \rho)$$
$$\xrightarrow{\mathcal{A}} (\lambda\rho.\rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times 1)$$

$$\langle y \times \text{`}(\langle y \times \text{`}(\langle y \times \text{`}(\langle 1 \rangle)\rangle)\rangle)\rangle$$
$$\xrightarrow{0} \langle y \times \text{`}(\langle y \times \text{`}(\langle y \times 1 \rangle)\rangle)\rangle$$
$$\xrightarrow{0} \langle y \times \text{`}(\langle y \times y \times 1 \rangle)\rangle$$
$$\xrightarrow{0} \langle y \times y \times y \times 1 \rangle$$

$$(\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.1)))$$
$$\xrightarrow{\mathcal{R}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times (\lambda\rho.1)\ \rho))$$
$$\xrightarrow{\mathcal{A}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)((\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times 1))$$
$$\xrightarrow{\mathcal{R}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times (\lambda\rho.\rho\cdot \mathrm{y} \times 1)\ \rho)$$
$$\xrightarrow{\mathcal{A}} (\lambda h.\lambda\rho.\rho\cdot \mathrm{y} \times h\ \rho)(\lambda\rho.\rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times 1)$$
$$\xrightarrow{\mathcal{R}} (\lambda\rho.\rho\cdot \mathrm{y} \times (\lambda\rho.\rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times 1)\ \rho)$$
$$\xrightarrow{\mathcal{A}} (\lambda\rho.\rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times \rho\cdot \mathrm{y} \times 1)$$

$$\mathsf{run}\langle \lambda y.\, \text{`}(\langle y \times \text{`}(\langle y \times \text{`}(\langle y \times \text{`}(\langle 1 \rangle)\rangle)\rangle)\rangle)\rangle$$

$$\begin{array}{l}\mathsf{let}\ h' = (\lambda h.\lambda\rho.\lambda y.h\ (\rho\,\mathsf{with}\,\{\mathrm{y} = y\})) \\ \qquad\qquad \big((\lambda h.\lambda\rho.\rho{\cdot}\mathrm{y}\times h\rho)((\lambda h.\lambda\rho.\rho{\cdot}\mathrm{y}\times h\rho)((\lambda h.\lambda\rho.\rho{\cdot}\mathrm{y}\times h\rho)(\lambda\rho.1)))\big) \\ \mathsf{in}\ h'\ \{\}\end{array}$$

$$\mathsf{run}\langle \lambda y.\, \text{`}(\langle y \times y \times y \times 1 \rangle) \rangle$$

$$\begin{array}{l} \mathsf{let}\ h' = (\lambda h.\lambda \rho.\lambda y.h\ (\rho\, \mathsf{with}\, \{\mathrm{y} = y\})) \\ \qquad\qquad (\lambda \rho.\rho \cdot \mathrm{y} \times \rho \cdot \mathrm{y} \times \rho \cdot \mathrm{y} \times 1) \\ \mathsf{in}\ h'\ \{\} \end{array}$$

$$\begin{aligned}
&\mathsf{run}\langle \lambda y.\,\text{`}(\langle y \times y \times y \times 1\rangle)\rangle \\
&\quad \xrightarrow{0} \mathsf{run}(\lambda y.y \times y \times y \times 1) \\
&\quad \xrightarrow{0} \lambda y.y \times y \times y \times 1
\end{aligned}$$

$$\begin{aligned}
&\mathsf{let}\ h' = (\lambda h.\lambda\rho.\lambda y.h\ (\rho\,\mathsf{with}\,\{\mathrm{y} = y\})) \\
&\qquad\quad (\lambda\rho.\rho\cdot\mathrm{y} \times \rho\cdot\mathrm{y} \times \rho\cdot\mathrm{y} \times 1) \\
&\mathsf{in}\ h'\ \{\} \\
&\xrightarrow{\mathcal{R}} \mathsf{let}\ h' = (\lambda\rho.\lambda y.(\lambda\rho.\rho\cdot\mathrm{y}\times\rho\cdot\mathrm{y}\times\rho\cdot\mathrm{y}\times 1)(\rho\,\mathsf{with}\,\{\mathrm{y}=y\}))\ \mathsf{in}\ h'\,\{\} \\
&\xrightarrow{\mathcal{A}} \mathsf{let}\ h' = (\lambda\rho.\lambda y.(\rho\,\mathsf{with}\,\{\mathrm{y}=y\})\cdot\mathrm{y}\times(\rho\,\mathsf{with}\,\{\mathrm{y}=y\})\cdot\mathrm{y}\times(\rho\,\mathsf{with}\,\{\mathrm{y}=y\})\cdot\mathrm{y}\times 1)\ \mathsf{in}\ h'\ \{\} \\
&\xrightarrow{\mathcal{A}^*} \mathsf{let}\ h' = (\lambda\rho.\lambda y.\ y \times y \times y \times 1)\ \mathsf{in}\ h'\ \{\} \\
&\xrightarrow{\mathcal{R}} (\lambda\rho.\lambda y.\ y \times y \times y \times 1)\{\} \\
&\xrightarrow{\mathcal{A}} \lambda y.\ y \times y \times y \times 1
\end{aligned}$$

$$\mathsf{run}\langle \lambda y.\, `(\langle y \times y \times y \times 1\rangle)\rangle$$
$$\xrightarrow{0} \mathsf{run}(\lambda y.y \times y \times y \times 1)$$
$$\xrightarrow{0} \lambda y.y \times y \times y \times 1$$

$$\mathsf{let}\ h' = (\lambda h.\lambda\rho.\lambda y.h\ (\rho\,\mathsf{with}\,\{\mathrm{y} = y\}))$$
$$(\lambda\rho.\rho\cdot\mathrm{y} \times \rho\cdot\mathrm{y} \times \rho\cdot\mathrm{y} \times 1)$$
$$\mathsf{in}\ h'\ \{\}$$
$$\xrightarrow{\mathcal{R}} \mathsf{let}\ h' = (\lambda\rho.\lambda y.(\lambda\rho.\rho\cdot\mathrm{y}\times\rho\cdot\mathrm{y}\times\rho\cdot\mathrm{y}\times 1)(\rho\,\mathsf{with}\,\{\mathrm{y}=y\}))\ \mathsf{in}\ h'\,\{\}$$
$$\xrightarrow{\mathcal{A}} \mathsf{let}\ h' = (\lambda\rho.\lambda y.(\rho\,\mathsf{with}\,\{\mathrm{y}=y\})\cdot\mathrm{y}\times(\rho\,\mathsf{with}\,\{\mathrm{y}=y\})\cdot\mathrm{y}\times(\rho\,\mathsf{with}\,\{\mathrm{y}=y\})\cdot\mathrm{y}\times 1)\ \mathsf{in}\ h'\ \{\}$$
$$\xrightarrow{\mathcal{A}^*} \mathsf{let}\ h' = (\lambda\rho.\lambda y.\ y \times y \times y \times 1)\ \mathsf{in}\ h'\ \{\}$$
$$\xrightarrow{\mathcal{R}} (\lambda\rho.\lambda y.\ y \times y \times y \times 1)\{\}$$
$$\xrightarrow{\mathcal{A}} \lambda y.\ y \times y \times y \times 1$$

Conclusion

- Can convert record expressions back to staged expressions.
- Only two kinds of admin reductions.
  - Apply exhaustively.
- Future direction: Use record calculus analyses to reason about program generators.