Directed Atomic λ-Calculus

David R. Sherratt and Willem Heijtjes

Department of Computer Science
University of Bath

08/03/2018
Swansea University Theory Tea
Theori Te Prifysgol Abertawe

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Before we begin...

Sharing

Use of a single representation for multiple instances of some subterm

Duplication

Extraction of an object from a sharing to its context within some term

$$t[x_1, x_2 \leftarrow s] \rightsquigarrow t\{s/x_1\}\{s/x_2\}$$

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Lazy Evaluation

$$(\lambda x.t)s \rightsquigarrow_{\beta} t\{s/x\}$$

- Delays evaluation of terms until a `value` is required
- Avoids duplication of work (does not repeat the same evaluation)

Lazy Evaluation

$$(\lambda x.t)s \rightsquigarrow_\beta t\{s/x\}$$

- Delays evaluation of terms until a `value` is required
- Avoids duplication of work (does not repeat the same evaluation)

NOT TRUE FOR FUNCTIONAL COMPILERS

$$(\lambda f.fI(fI))(\lambda w.(II)w)$$

$$I = \lambda x.x$$

Sharing Graphs

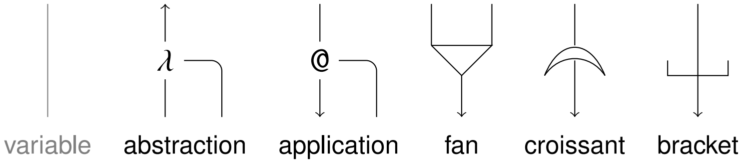
Optimal Reduction

To avoid useless repitition of work in the reduction of terms

- Concept of optimality formally introduced by Lévy (1970)
- Lamping defined an evaluator satisfying optimality requirements (1989/90) called *Sharing Graphs*
- Algorithm based on *Atomic Duplication*

Sharing Graphs

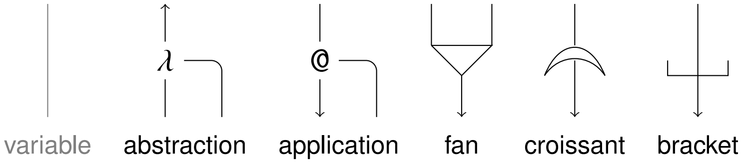
Sharing graph operators



- Each operator has a *principle port*

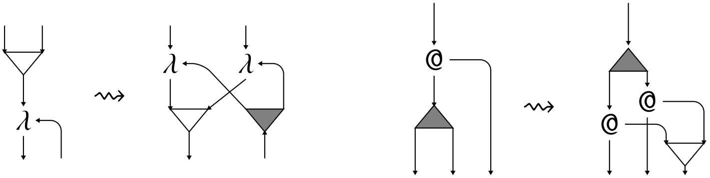
Sharing Graphs

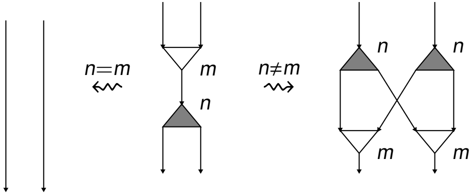
Sharing graph operators



- Each operator has a *principle port*
- Each operator has as index $i \in \mathbb{N}$ associated with it
- The index indicates the 'level' of a term
  e.g. $s^0(t^1 u^2)$

Sharing Graphs: Some Duplication





As an evaluator

As an evaluator, the work of maintaining the nodes outweighs the value of optimal reduction/sharing graphs

The atomic λ-calculus is the first use of atomic duplication in a term calculus

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

The atomic λ-calculus - Deep Inference

$$\begin{array}{c} A \\ \Downarrow \\ C \end{array} \;:=\; A \;\;\Big|\;\; \begin{array}{c} A_1 \\ \Downarrow \\ C_1 \end{array} \wedge \begin{array}{c} A_2 \\ \Downarrow \\ C_2 \end{array} \;\;\Big|\;\; \begin{array}{c} A_1 \\ \Uparrow \\ C_1 \end{array} \to \begin{array}{c} A_2 \\ \Downarrow \\ C_2 \end{array} \;\;\Big|\;\; \begin{array}{c} A \\ \Downarrow \\ \dfrac{B_1}{B_2}\, r \\ \Downarrow \\ C \end{array}$$

$$\frac{B}{A \to (A \wedge B)}\,\lambda \qquad \frac{A \wedge (A \to B)}{B}\,@ \qquad \frac{A}{A \wedge \ldots \wedge A}\,\triangle$$

Guglielmi, Gundersen, and Parigot. *A proof calculus which reduces syntactic bureaucracy.* (2010): 135-150.

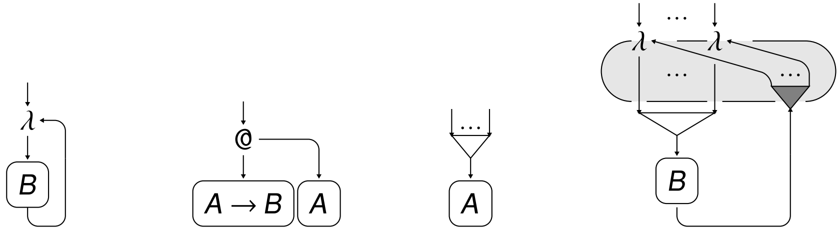
The atomic λ-calculus

$$\frac{A \to (B \wedge C)}{(A \to B) \wedge (A \to C)}\ d$$

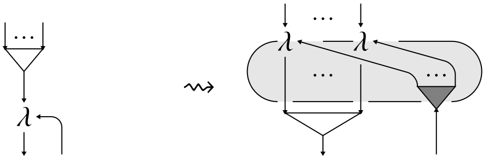
- A typed term calculus with explicit sharing
- Atomic duplication (on individual constructors)
- Achieves fully lazy sharing

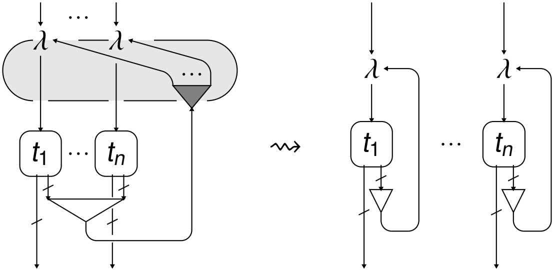
Graphical Intuition

$$\frac{B}{A \to (A \wedge B)}\,\lambda \qquad \frac{A \wedge (A \to B)}{B}\,@ \qquad \frac{A}{A \wedge \ldots \wedge A}\,\triangle \qquad \frac{A \to \dfrac{B}{(B \wedge \ldots \wedge B)}\,\triangle}{(A \to B) \wedge (A \to B)}\,d$$

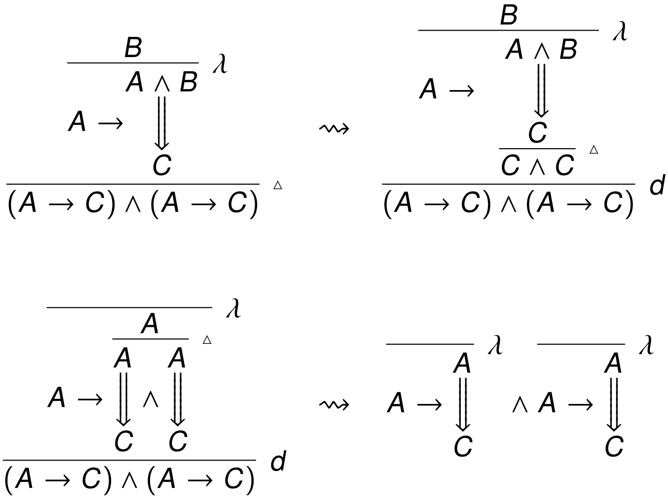


Differences to Sharing Graphs - Duplication of Abstraction





Duplication of Abstraction - Proof Transformations



The atomic λ-calculus: Term Calculus

An atomic lambda term is given by the mutually recursive grammar

$$t ::= x \mid \lambda x.t \mid (t)t \mid t[x_1, \ldots, x_n \leftarrow t] \mid t[x_1, \ldots, x_n \twoheadleftarrow \lambda y.t^n]$$

$$t^n ::= \langle t_1, \ldots, t_n \rangle \mid t^n[x_1, \ldots, x_n \leftarrow t] \mid t^n[x_1, \ldots, x_m \twoheadleftarrow \lambda y.t^m]$$

- Linear calculus - each variable occurs exactly once
- Variables bound by an abstraction/sharing/distributor must occur

Atomic λ-calculus: Fully Lazy Sharing

$$u[z_1, z_2 \leftarrow \lambda x.t[\leftarrow x]]$$

$$\rightsquigarrow\ u[z_1, z_2 \twoheadleftarrow \lambda x.\langle y_1, y_2\rangle[y_1, y_2 \leftarrow t[\leftarrow x]]]$$

$$\rightsquigarrow\ u[z_1, z_2 \twoheadleftarrow \lambda x.\langle y_1, y_2\rangle[y_1, y_2 \leftarrow t][\leftarrow x]]$$

$$\rightsquigarrow\ u[z_1, z_2 \twoheadleftarrow \lambda x.\langle y_1, y_2\rangle[\leftarrow x]][y_1, y_2 \leftarrow t]$$

$$\rightsquigarrow\ u\{\lambda x_1.y_1[\leftarrow x_1]/z_1\}\{\lambda x_2.y_2[\leftarrow x_2]/z_2\}[y_1, y_2 \leftarrow t]$$

Atomic λ-calculus: Implementation

$\lambda x.t[\vec{y} \leftarrow s] \rightsquigarrow (\lambda x.t)[\vec{y} \leftarrow s]$
only if $x \notin \mathrm{fv}(s)$

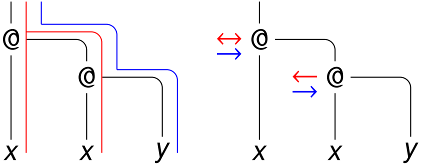
- Variable lookup operations are expensive
- A potential abstract machine would need to perform these checks, destroying the value of *full laziness*

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Director Strings: Local Reduction

$$(x)(xy)\{X/x\}\{Y/y\}$$



Director Strings: Why?

- Director strings provide the ideal information to perform an optimal performance of substitution.
- Many abstract machines have been designed using director strings. Director strings seem like a natural choice to use to help develop a computationally-efficient evaluation for functional programs.
- Effecient reduction strategies involving director strings have already been studied. These strategies are considered efficient in the sense of the number of rewrite steps (not just β-steps).

Outline

The directed atomic λ-calculus

$\Lambda_a^{\mathcal{D}}$: A calculus with local reduction rules that performs fully-lazy reduction

- Typed
- Strongly Normalizing
- Confluent

The directed atomic λ-calculus

$$(\!|-|\!) : \Lambda_a^{\mathcal{D}} \to \Lambda_a$$

$$[\![-]\!] : \Lambda_a \to \Lambda_a^{\mathcal{D}}$$

Theorem

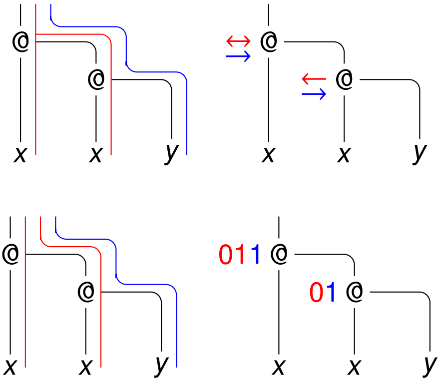
Given $\Lambda_a^{\mathcal{D}}$-terms t and s, if $t \rightsquigarrow s$ then there is a reduction rule in Λ_a such that $(\!|t|\!) \rightsquigarrow (\!|s|\!)$.

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Linear Director Strings

$$t[x \to s] \rightsquigarrow t\{s/x\}$$



$$\alpha, \gamma ::= (\,0\,|\,1\,)^*$$

Invert

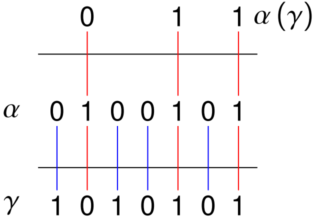
$$\overline{\alpha} = \alpha \text{ inverted}$$
$$\text{e.g.}\quad \overline{10100} = 01011$$

String Operations

Mask

$$
\begin{aligned}
0 \cdot \alpha\,(z \cdot \gamma) &= \alpha\,(\gamma) \\
1 \cdot \alpha\,(z \cdot \gamma) &= z \cdot \alpha\,(\gamma)
\end{aligned}
$$

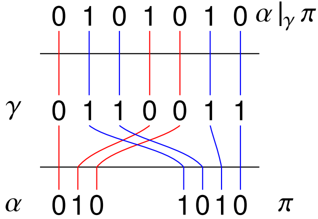
α	=	0	1	0	0	1	0	1
γ	=	1	0	1	0	1	0	1
$\alpha\,(\gamma)$	=		0			1		1
$\overline{\alpha}\,(\gamma)$	=	1		1	0		0	



Interleaving

$$
\begin{aligned}
(z \cdot \alpha) \,|_{0 \cdot \gamma}\, \pi &= z \cdot (\alpha \,|_\gamma\, \pi) \\
\alpha \,|_{1 \cdot \gamma}\, (z \cdot \pi) &= z \cdot (\alpha \,|_\gamma\, \pi)
\end{aligned}
$$

α	=	0			1	0		
π	=		1	0			1	0
γ	=	0	1	1	0	0	1	1
$\alpha \,\vert_\gamma\, \pi$	=	0	1	0	1	0	1	0



String Operations

Merge$_1$

$$
\begin{aligned}
(0 \cdot \alpha) \wr_1 (0 \cdot \gamma) &= 0 \cdot (\alpha \wr_1 \gamma) \\
(1 \cdot \alpha) \wr_1 \gamma &= 0 \cdot (\alpha \wr_1 \gamma) \\
\alpha \wr_1 (1 \cdot \gamma) &= 1 \cdot (\alpha \wr_1 \gamma) \quad \text{if } \alpha = 0 \cdot \alpha' \text{ or } |\alpha| = 0
\end{aligned}
$$

Merge$_2$

$$
\begin{aligned}
(0 \cdot \alpha) \wr_2 (0 \cdot \gamma) &= 0 \cdot (\alpha \wr_2 \gamma) \\
(1 \cdot \alpha) \wr_2 \gamma &= 1 \cdot (\alpha \wr_2 \gamma) \\
\alpha \wr_2 (1 \cdot \gamma) &= 0 \cdot (\alpha \wr_2 \gamma) \quad \text{if } \alpha = 0 \cdot \alpha' \text{ or } |\alpha| = 0
\end{aligned}
$$

String Operation

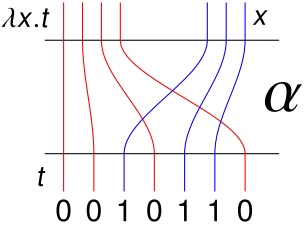
α	$=$	0	1	1		0	1			0	
γ	$=$	0			1	0		1	1	0	1
$\alpha \wr_1 \gamma$	$=$	0	0	0	1	0	0	1	1	0	1
$\alpha \wr_2 \gamma$	$=$	0	1	1	0	0	1	0	0	0	0

$$0\,1\,1\,0\,1\,0 \;\wr_1\; 0\,1\,0\,1\,1\,0\,1 \;=\; 0\,0\,0\,1\,0\,0\,1\,1\,0\,1$$

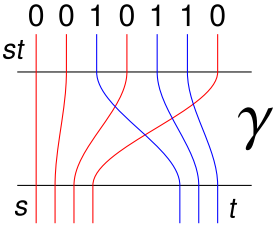
$$0\,1\,1\,0\,1\,0 \;\wr_2\; 0\,1\,0\,1\,1\,0\,1 \;=\; 0\,1\,1\,0\,0\,1\,0\,0\,0\,0$$

Director Strings in Terms

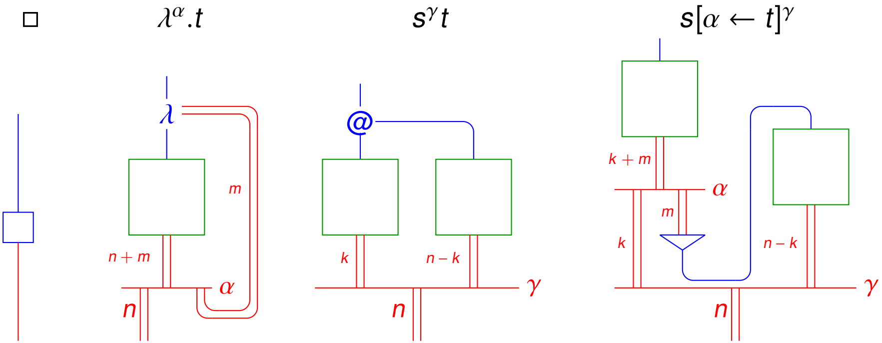
$\lambda^{\alpha}.t$



$s^{\gamma} t$



Director Strings in Terms



Directed Well-Formed Terms

Λ-term	$\Lambda_a^{\mathcal{D}-}$-term
$\lambda x.x$	$\lambda^1 \square$
$\lambda x.\lambda y.x$	$\lambda^1 \lambda^0 \square$
$\lambda x.\lambda y.\lambda z.(xz)(yz)$	$\lambda^1 \lambda^{01} \lambda^{0011} (\square^{01} \square)^{0101} (\square^{01} \square)$

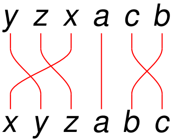
Term	Constraints
$\square$	$n = 1$
$\lambda^\alpha t_{n+m}$	$\lvert\alpha\rvert = n + m, \lvert\alpha\rvert_1 = m$
$t_k{}^\gamma t_{n-k}$	$\lvert\gamma\rvert = n, \lvert\gamma\rvert_0 = k$
$t_{k+m}[\alpha \leftarrow t_{n-k}]^\gamma$	$\lvert\gamma\rvert_0 = \lvert\alpha\rvert_0 = k, \lvert\alpha\rvert_1 = m$

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Permutation Equivalence

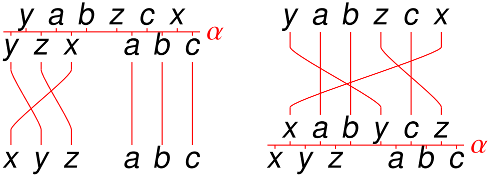
$\lambda^{01}\square^{01}\square$ $\qquad$ $\lambda^{10}\square^{10}\square$



$$\begin{pmatrix} y & z & x & a & c & b \\ x & y & z & a & b & c \end{pmatrix}$$

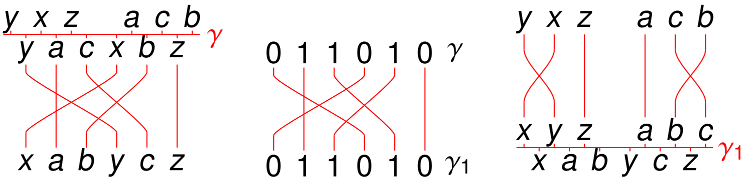
Permutation Equivalence: Permuting

$\lambda^{\alpha}.t$ $\qquad$ $t[\alpha \leftarrow s]^{\gamma}$



Permutation Equivalence: Permuting

$s^{\gamma} t$ $\qquad$ $t[\alpha \leftarrow s]^{\gamma}$



Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Substitution

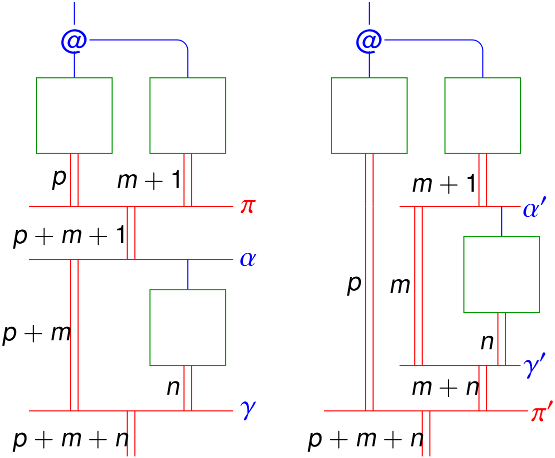
$$t\{s/\alpha\}^{\gamma} \qquad |\alpha|_1 = 1$$

$$(s^{\pi} t)\{u/\alpha\}^{\gamma} \rightsquigarrow s^{\pi'} t\{u/\alpha'\}^{\gamma'} \qquad \alpha(\pi) = 1$$

Substitution

$$t\{s/\alpha\}^{\gamma} \qquad |\alpha|_1 = 1$$

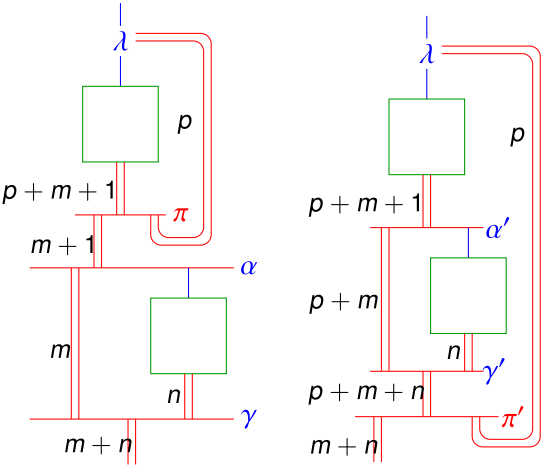
$$(s^{\pi} t)\{u/\alpha\}^{\gamma} \rightsquigarrow s^{\pi'} t\{u/\alpha'\}^{\gamma'} \qquad \alpha(\pi) = 1$$



π'	$\overline{\alpha}(\pi)\,\|_{\gamma}\,\vec{1}$
α'	$\pi(\alpha)$
γ'	$\pi'(\gamma)$

Substitution

$$(\lambda^{\pi} t)\{u/\alpha\}^{\gamma} \rightsquigarrow \lambda^{\pi'} t\{u/\alpha'\}^{\gamma'}$$



$\pi' \quad \overline{\alpha'}\,(\pi) \wr_2 \gamma$

$\alpha' \quad \alpha\,|_{\pi}\,\vec{0}$

$\gamma' \quad \overline{\alpha'}\,(\pi) \wr_1 \gamma$

Beta Rule

$$(\lambda^{\alpha}.t)^{\gamma} s \rightsquigarrow_{\beta} t[\alpha \leftarrow s]^{\gamma}$$

$$t[\alpha \leftarrow s]^{\gamma} \rightsquigarrow t\{s/\alpha\}^{\gamma} \qquad |\,\alpha\,|_1 = 1$$

$$t[\pi \leftarrow s]^{\sigma}[\alpha \leftarrow u]^{\gamma} \sim t[\alpha' \leftarrow u]^{\gamma'}[\pi' \leftarrow s]^{\sigma'} \qquad \text{if } |\,\alpha\,(\sigma)\,|_1 = 0$$

Beta Rule

$$(\lambda^{\alpha}.t)^{\gamma}s \rightsquigarrow_{\beta} t[\alpha \leftarrow s]^{\gamma}$$

$$t[\alpha \leftarrow s]^{\gamma} \rightsquigarrow t\{s/\alpha\}^{\gamma} \qquad |\alpha|_1 = 1$$

$$t[\pi \leftarrow s]^{\sigma}[\alpha \leftarrow u]^{\gamma} \sim t[\alpha' \leftarrow u]^{\gamma'}[\pi' \leftarrow s]^{\sigma'} \qquad \text{if } |\alpha(\sigma)|_1 = 0$$

$$\begin{array}{cl} & t[001100 \leftarrow s]^{10000}[01100 \leftarrow u]^{0010} \\ {}_{\beta}\!\leftsquigarrow & ((\lambda^{001100}t)^{10000}s)[01100 \leftarrow u]^{0100} \\ = & ((\lambda^{001100}t)[1100 \leftarrow u]^{010})^{1000}s \\ = & (\lambda^{11000}(t[110000 \leftarrow u]^{00010}))^{1000}s \\ \rightsquigarrow_{\beta} & t[110000 \leftarrow u]^{00010}[11000 \leftarrow s]^{1000} \end{array}$$

Preservation of well-formedness

Let t_n be a well-formed term with n occurrences of free variables. If term s_m can be obtained from t_n, $t_n \rightsquigarrow^* s_m$, by following the beta rule and substitution, then s_m is also a well-formed term. Furthermore, $n = m$.

Preservation of well-formedness: Proof

Case: Abstraction $(\lambda^{\pi} t)\{u/\alpha\}^{\gamma} = \lambda^{\pi'} t\{u/\alpha'\}^{\gamma'}$ where $|\vec{0}| = |\pi|_1$

$$
\begin{array}{lcl}
\pi' & = & \overline{\alpha|_\pi \vec{0}}\,(\pi) \wr_2 \gamma \\
\alpha' & = & \alpha|_\pi \vec{0} \\
\gamma' & = & \overline{\alpha|_\pi \vec{0}}\,(\pi) \wr_1 \gamma
\end{array}
$$

The rule preserves well-formedness if

$$
\begin{array}{rcl}
|\gamma| & = & |\pi'|_0 \\
\\
|\gamma'| & = & |\pi'| \\
|\gamma'|_0 & = & |\alpha'|_0 \\
\\
|\pi|_1 & = & |\pi'|_1 \\
|\gamma|_1 & = & |\gamma'|_1 \\
|\pi| & = & |\alpha'| \\
|\alpha|_1 & = & |\alpha'|_1 = 1
\end{array}
$$

Outline

Lifting Closures

$$\lambda^{\pi} t[\alpha \leftarrow u]^{\gamma} \rightsquigarrow_S (\lambda^{\pi'} t)[\alpha' \leftarrow u]^{\gamma'}$$

$$(s[\alpha \leftarrow u]^{\gamma})^{\pi} t \rightsquigarrow_S (s^{\gamma'_1 \pi'} t)[\alpha' \leftarrow u]^{\gamma'}$$

$$s^{\pi} t[\alpha \leftarrow u]^{\gamma} \rightsquigarrow_S (s^{\pi'} t)[\alpha' \leftarrow u]^{\gamma'}$$

$$s[\alpha_2 \leftarrow t[\alpha \leftarrow u]^{\gamma}]^{\pi} \rightsquigarrow_S s[\alpha_2 \leftarrow t]^{\pi'}[\alpha' \leftarrow u]^{\gamma'}$$

$$t[\alpha_2 \twoheadleftarrow \lambda^{\sigma}.d[\alpha \leftarrow u]^{\gamma}]^{\pi} \rightsquigarrow_S t[\alpha_2 \twoheadleftarrow \lambda^{\sigma'}.d]^{\pi'}[\alpha' \leftarrow u]^{\gamma'}$$

$$t[\alpha \leftarrow \square]^{\gamma}[\alpha_1 \leftarrow u]^{\gamma_1} \rightsquigarrow_S t[\alpha' \leftarrow u]^{\gamma'}$$

Atomic Duplication: Application

$$u[\vec{x} \leftarrow st] \rightsquigarrow_S u\{y_i z_i / x_i\}_{\forall_i . x_i \in \vec{x}}[\vec{y} \leftarrow s][\vec{z} \leftarrow t]$$

$$u[\alpha \leftarrow s^{\gamma_1} t]^{\gamma} \rightsquigarrow_S u\{\Box^{01}\Box/\alpha^1\}^{\gamma^1} \ldots \{\Box^{01}\Box/\alpha^n\}^{\gamma^n}[\alpha'_1 \leftarrow s]^{\gamma'_1}[\alpha'_2 \leftarrow t]^{\gamma'_2}$$

Atomic Duplication: Application

$$u[\alpha \leftarrow s^{\gamma_1} t]^{\gamma} \rightsquigarrow_S u\{\square^{01}\square/\alpha^1\}^{\gamma^1} \ldots \{\square^{01}\square/\alpha^n\}^{\gamma^n}[\alpha'_1 \leftarrow s]^{\gamma'_1}[\alpha'_2 \leftarrow t]^{\gamma'_2}$$

$$u\{\square^{01}\square/\alpha^1\}^{\gamma^1} \ldots \{\square^{01}\square/\alpha^n\}^{\gamma^n}$$

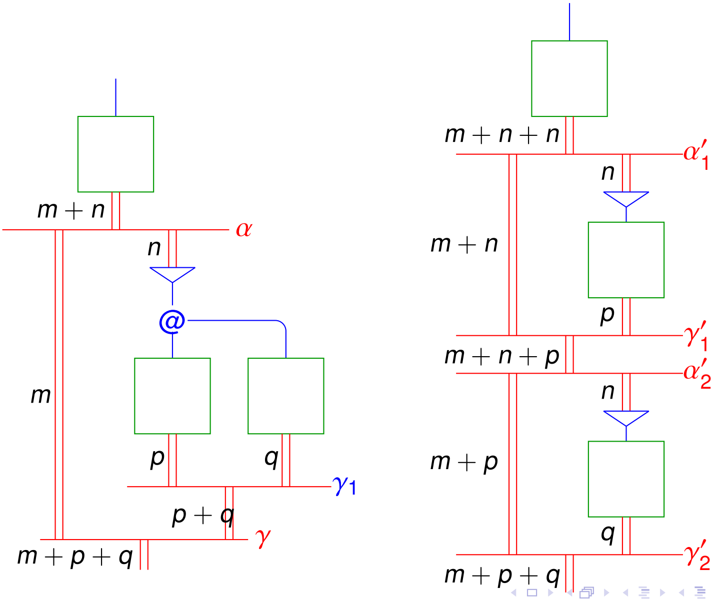
Atomic Duplication: Application

$$u[\alpha \leftarrow s^{\gamma_1} t]^{\gamma} \rightsquigarrow_S u\{\square^{01}\square/\alpha^1\}^{\gamma^1} \ldots \{\square^{01}\square/\alpha^n\}^{\gamma^n}[\alpha'_1 \leftarrow s]^{\gamma'_1}[\alpha'_2 \leftarrow t]^{\gamma'_2}$$

$$u\{\square^{01}\square/\alpha^1\}^{\gamma^1} \ldots \{\square^{01}\square/\alpha^n\}^{\gamma^n}$$

α	0	1	0	1	1	0					
α^1	0	1	0	0	0	0					
γ^1	0	·	0	0	0	0	1	1			
α^2	0	·	0	1	0	0	0	0			
γ^2	0	·	0	·	0	0	0	0	1	1	
α^3	0	·	0	·	1	0	0	0	0	0	
γ^3	0	·	0	·	·	0	0	0	0	0	1	1

Atomic Duplication: Application



Atomic Duplication: Application

$$u[\alpha \leftarrow s^{\gamma_1} t]^{\gamma} \rightsquigarrow_S u\{\square^{01}\square/\alpha^1\}^{\gamma^1} \ldots \{\square^{01}\square/\alpha^n\}^{\gamma^n}[\alpha'_1 \leftarrow s]^{\gamma'_1}[\alpha'_2 \leftarrow t]^{\gamma'_2}$$

$$\begin{aligned}
\alpha'_1 &= \vec{0} \cdot 1\vec{0} \text{ where } |\vec{0}| = |\alpha|_0, |1\vec{0}| = 2|\alpha|_1 \\
\gamma'_1 &= (\vec{1}\,|_\gamma\, \overline{\gamma_1})(\gamma) \cdot \vec{0} \text{ where } |\vec{0}| = |\alpha|_1 \\
\alpha'_2 &= \vec{0} \cdot \vec{1} \text{ where } |\vec{0}| = |\gamma_1|_0 + |\alpha|_0 \text{ and } |\vec{1}| = |\alpha|_1 \\
\gamma'_2 &= \vec{0}\,|_\gamma\, \gamma_1
\end{aligned}$$

Atomic Duplication: Distributor

$$u[\vec{x} \leftarrow \lambda y.t] \leftrightsquigarrow_S u[\vec{x} \twoheadleftarrow \lambda y.\langle \vec{z} \rangle[\vec{z} \leftarrow t]]$$

$$u[\alpha \leftarrow \lambda^{\alpha_1} t]^{\gamma} \rightsquigarrow_S u[\alpha \twoheadleftarrow \lambda^{\alpha_1}.d[\vec{m} \leftarrow t]^{\vec{1}}]^{\gamma}$$

$|\vec{1}| = |\alpha_1|$

$|\vec{m}| = |\vec{m}|_1 = |\alpha_1|_1$

$d = \langle \square, \langle \ldots \rangle^{\gamma} \rangle^{0 \cdot \vec{1}}$ where $|\, 0 \cdot \vec{1} \,| = |\, \vec{m} \,|$

Atomic Duplication: Distributor

$$u[\vec{x} \twoheadleftarrow \lambda y.\langle \vec{t}\rangle[\vec{z} \leftarrow y]] \rightsquigarrow_S u\{\lambda y_i.t[\vec{a} \leftarrow y_i]/x_i\}_{1\geq i\geq|\vec{x}|}$$
$$\text{where } \vec{a} = \vec{z} \cap \mathsf{fv}(t_i)$$

$$u[\alpha_1 \twoheadleftarrow \lambda^{\alpha_2}.\langle t, \langle\ldots\rangle^{\alpha'_3}\rangle^{\alpha_3}[\alpha_4 \leftarrow \square]^{\gamma_1}]^{\gamma_2} \rightsquigarrow_S$$
$$u\{\lambda^{\alpha'_2} t[\alpha'_4 \leftarrow \square]^{\gamma'_1}/\alpha'_1\}^{\gamma'_2}[\alpha''_1 \twoheadleftarrow \lambda^{\alpha''_2}\langle\ldots\rangle^{\alpha'_3}[\alpha''_4 \leftarrow \square]^{\gamma''_1}]^{\gamma''_2}$$

$$\begin{aligned}
\alpha'_4 &= \overline{\alpha_3}(\alpha_4) \\
\alpha'_2 = \gamma'_1 &= \vec{0}\cdot 1 \text{ where } |\vec{0}| = |\alpha'_4|_0 \\
\text{let } \pi &= \vec{1}\,|_{\gamma_2}\, \overline{\alpha_4}(\overline{\alpha_3}) \\
\gamma'_2 &= \pi(\gamma_2) \wr_2 \overline{\alpha'_1}(\alpha_1) \\
\alpha''_1 &= \pi(\gamma_2) \wr_1 \overline{\alpha'_1}(\alpha_1) \\
\alpha''_4 &= \alpha_3(\alpha_4) \\
\alpha''_2 = \gamma''_1 &= \vec{0}\cdot 1 \text{ where } |\vec{0}| = |\alpha_3(\alpha_4)|_0 \\
\gamma''_2 &= \vec{0}\,|_{\gamma_2}\, \overline{\alpha_4}(\alpha_3) \text{ where } |\vec{0}| = |\gamma_2|_0
\end{aligned}$$

Outline

1. Why
   - Efficient reduction of λ-terms
   - Atomic λ-calculus
   - Director Strings
2. What
   - The directed atomic λ-calculus
3. How
   - Prerequisites
   - Permutation Equivalence
   - Beta Rule
   - Atomic Duplication
   - Translations

Translation into linear $\Lambda_a^{\mathcal{D}}$-terms

The translation $[\![-]\!] : \Lambda_a^{\mathcal{D}} \rightarrow \Lambda_a^{\mathcal{D}}$ translates $\Lambda_a^{\mathcal{D}}$-terms into linear $\Lambda_a^{\mathcal{D}}$-terms

$$
\begin{array}{rcl}
[\![\square]\!] & = & \square \\
[\![s^{\gamma} t]\!] & = & [\![s]\!]^{\gamma} [\![t]\!] \\
[\![\lambda^{\alpha} t]\!] & = & \lambda^{\vec{0} \cdot 1} [\![t]\!] [\alpha \leftarrow \square]^{\vec{0} \cdot 1} \\
 & & \text{if } |\alpha|_1 \neq 1 \text{ where } |\vec{0}| = |\alpha|_0 \\
 & = & \lambda^{\alpha} [\![t]\!] \\
 & & \text{if } |\alpha|_1 = 1 \\
[\![s[\alpha \leftarrow t]^{\gamma}]\!] & = & [\![s]\!] [\alpha \leftarrow [\![t]\!]]^{\gamma} \\
[\![s[\alpha \twoheadleftarrow \lambda^{\alpha_1}.d]^{\gamma}]\!] & = & [\![s]\!] [\alpha \twoheadleftarrow \lambda^{\vec{0} \cdot 1}.[\![d]\!] [\alpha_1 \leftarrow \square]^{\vec{0} \cdot 1}]^{\gamma} \\
 & & \text{if } |\alpha_1|_1 \neq 1 \text{ where } |\vec{0}| = |\alpha_1|_0 \\
 & = & [\![s]\!] [\alpha \twoheadleftarrow \lambda^{\alpha}.[\![d]\!]]^{\gamma} \\
 & & \text{if } |\alpha_1|_1 = 1 \\
[\![\langle t, \langle \ldots \rangle^{\gamma_1} \rangle^{\gamma}]\!] & = & \langle [\![t]\!], [\![\langle \ldots \rangle^{\gamma_1}]\!] \rangle^{\gamma}
\end{array}
$$

Translation $\Lambda_a^{\mathcal{D}} \to \Lambda_a$

The interpretation $[\![-]\!] : \Lambda_a^{\mathcal{D}} \times \mathcal{N}^* \to \Lambda_a$ is defined by

$$
\begin{array}{rcl}
[\![\square]\!]_x & = & x \\
[\![s^\gamma t]\!]_{\vec{x}} & = & [\![s]\!]_{\overline{\gamma}(\vec{x})} [\![t]\!]_{\gamma(\vec{x})} \\
[\![\lambda^\alpha t]\!]_{\vec{x}} & = & \lambda y . [\![t]\!]_{\vec{x}\,|_\alpha\, y} \\
 & & \text{where } y \text{ is a fresh variable} \\
[\![s[\alpha \leftarrow t]^\gamma]\!]_{\vec{x}} & = & [\![s]\!]_{\overline{\gamma}(\vec{x})\,|_\alpha\, \vec{y}}[\vec{y} \leftarrow [\![t]\!]_{\gamma(\vec{x})}] \\
 & & \text{where } \vec{y} \text{ is a vector of fresh unique variables} \\
[\![s[\alpha \twoheadleftarrow \lambda^{\alpha_1}.d]^\gamma]\!]_{\vec{x}} & = & [\![s]\!]_{\overline{\gamma}(\vec{x})\,|_\alpha\, \vec{y}}[\vec{y} \twoheadleftarrow \lambda z . [\![d]\!]_{\gamma(\vec{x})\,|_{\alpha_1}\, z}] \\
 & & \text{where } \vec{y} \text{ is a vector of fresh unique variables} \\
 & & \text{and } z \text{ is a fresh variable} \\
[\![\langle t, \langle \ldots \rangle^{\gamma_1} \rangle^\gamma]\!]_{\vec{x}} & = & \langle [\![t]\!]_{\overline{\gamma}(\vec{x})}, [\![\langle \ldots \rangle^{\gamma_1}]\!]_{\gamma(\vec{x})} \rangle
\end{array}
$$

Translation $\Lambda_a \to \Lambda_a^{\mathcal{D}}$

The interpretation $(\!|-|\!) : \Lambda_a \to \Lambda_a^{\mathcal{D}} \times \mathcal{N}^*$

$$
\begin{array}{rcl}
(\!|x|\!) & = & (\square, x) \\
(\!|st|\!) & = & (s'^{(\vec{0}\cdot\vec{1})} t', \vec{s} \cdot \vec{t}) \\
 & & \text{where } (s', \vec{s}) = (\!|s|\!) \text{ and } (t', \vec{t}) = (\!|t|\!) \\
 & & |\vec{0}| = |\vec{s}| \text{ and } |\vec{1}| = |\vec{t}| \\
(\!|\lambda x.t|\!) & = & (\lambda^{\alpha} t', \overline{\alpha}(\vec{t})) \\
 & & \text{where } (t', \vec{t}) = (\!|t|\!),\ |\alpha|_1 = 1,\ \alpha(\vec{t}) = x \\
(\!|s[\vec{y} \leftarrow t]|\!) & = & (s'[\alpha \leftarrow t']^{\vec{0}\cdot\vec{1}}, \overline{\alpha}(\vec{s}) \cdot \vec{t}) \\
 & & \text{where } (s', \vec{s}) = (\!|s|\!),\ (t', \vec{t}) = (\!|t|\!) \\
 & & \alpha \text{ is such that } \alpha(\vec{t}) = \vec{y},\ |\vec{0}| = |\overline{\alpha}(\vec{s})| \text{ and } |\vec{1}| = |\vec{t}|
\end{array}
$$

Translation $\Lambda_a \to \Lambda_a^{\mathcal{D}}$

$$
\begin{aligned}
(\!|s[\vec{y} \twoheadleftarrow \lambda z.d]|\!) &= (s'[\alpha \twoheadleftarrow \lambda^{\alpha_1}.d']^{\vec{0}\cdot\vec{1}}, \overline{\alpha}(\vec{s}) \cdot \overline{\alpha_1}(\vec{d})) \\
&\quad \text{where } (s', \vec{s}) = (\!|s|\!),\ (d', \vec{d}) = (\!|d|\!), \\
&\quad \alpha \text{ and } \alpha_1 \text{ are such that } \alpha(\vec{s}) = \vec{y},\ \alpha_1(\vec{d}) = z \\
&\quad |\vec{0}| = |\alpha|_0 \text{ and } |\vec{1}| = |\alpha_1|_0 \\
(\!|\langle t, \langle \ldots \rangle \rangle|\!) &= (\langle t', \langle \ldots \rangle^{\gamma} \rangle^{\vec{0}\cdot\vec{1}}, \vec{t} \cdot \vec{s}) \\
&\quad \text{where } (t', \vec{t}) = (\!|t|\!) \text{ and } (\langle \ldots \rangle^{\gamma}, \vec{s}) = (\!|\langle \ldots \rangle|\!) \\
&\quad |\vec{0}| = |\vec{t}| \text{ and } |\vec{1}| = |\vec{s}|
\end{aligned}
$$

Strongly Normalizing

Given linear $\Lambda_a^{\mathcal{D}}$-term s, if $s \rightsquigarrow_S t$ then there is a rule in the atomic λ-calculus such that $[\![s]\!]_{\vec{x}} \rightsquigarrow [\![t]\!]_{\vec{x}}$.

$$\begin{array}{ccc} s & \rightsquigarrow_S & t \\ \downarrow & & \downarrow \\ [\![s]\!]_{\vec{x}} & \rightsquigarrow_S & [\![t]\!]_{\vec{x}} \end{array}$$

Some useful Lemma's for proving this

$$|\alpha(\gamma)|_0 = |\overline{\gamma}(\alpha)|_1$$

Some useful Lemma's for proving this

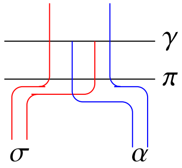
$$|\alpha(\gamma)|_0 = |\overline{\gamma}(\alpha)|_1$$

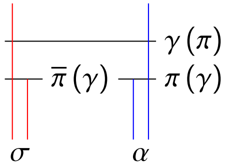
$$|\alpha(\gamma)|_0 = |\alpha(\overline{\gamma})|_1 = |\overline{\gamma}(\overline{\alpha})|_0 = |\overline{\gamma}(\alpha)|_1$$

$\|\alpha(\gamma)\|_0$		$\|\overline{\gamma}(\alpha)\|_1$	
$\alpha = 1$	$\gamma = 1$	$\gamma = 1$	$\alpha = 1$
$\alpha = 1$	$\gamma = 0$	$\gamma = 0$	$\alpha = 1$
$\alpha = 0$	$\gamma = 1$	$\gamma = 1$	$\alpha = 0$
$\alpha = 0$	$\gamma = 0$	$\gamma = 0$	$\alpha = 0$

Some useful Lemma's for proving this

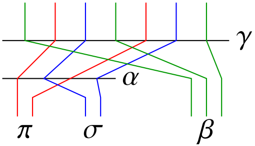
$$\gamma\,(\sigma\,|_\pi\,\alpha) = \overline{\pi}\,(\gamma)\,(\sigma)\,|_{\gamma\,(\pi)}\,\pi\,(\gamma)\,(\alpha)$$

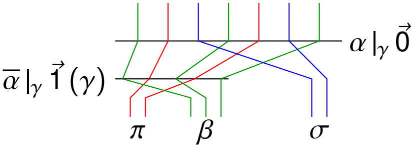




Some useful Lemma's for proving this

$$(\pi \,|_\alpha\, \sigma)\,|_\gamma\, \beta = (\pi \,|_{\overline{\alpha}\,|_\gamma\, \vec{1}\,(\gamma)}\, \beta)\,|_{\alpha\,|_\gamma\, \vec{0}}\, \sigma$$





Summary

- Explain how to implement full laziness using local reduction rules
- Achieved by keeping track of the free variables in all subterms

Summary

- Explain how to implement full laziness using local reduction rules
- Achieved by keeping track of the free variables in all subterms

- Outlook
  - Not clear the book-keeping required can be implemented efficiently as it is

Summary

- Explain how to implement full laziness using local reduction rules
- Achieved by keeping track of the free variables in all subterms

- Outlook
  - Not clear the book-keeping required can be implemented efficiently as it is - the use of a canonical form and a reduction strategy may solve this
  - Represent the distributor as an environment in an abstract machine
  - To formalize *atomic graphs*, the graphical equivalent of the atomic λ-calculus