THE LAMBDA CALCULUS AS A PROGRAMMING LANGUAGE.

λ-CALCULUS:

LOGIC ⟶ FUNCTIONAL PROGRAMMING
⟶ CATEGORY THEORY

- AN EXPRESSION LANGUAGE ("EVALUATE")
- FUNCTIONAL (NO SIDE EFFECTS)
- HIGHER-ORDER (FUNCTIONS OF FUNCTIONS)
- PROTOTYPICAL (BASIS FOR STUDYING LANGUAGE FEATURES)

ACTUAL PROGRAMMING LANGUAGES BASED ON λ-CALCULUS:

ML, Lisp, Scheme, Miranda, Haskell ...

Fields Institute Summer School
Logic and Foundations of Computation
University of Ottawa, May 2-20, 2003

Peter Selinger

University of Ottawa

THE UNTYPED LAMBDA CALCULUS

- NO TYPES
- NO BUILT-IN DATA OR OPERATIONS
- REPRESENT BOTH PROGRAMS AND DATA AS λ-TERMS

SYNTAX

TERMS $M, N ::= x \mid MN \mid \lambda x.M$

FREE VARIABLES $FV(x) = \{x\}$
$FV(MN) = FV(M) \cup FV(N)$
$FV(\lambda x.M) = FV(M) - \{x\}$

α-EQUIVALENCE: RENAMING OF BOUND VARIABLES

CONVENTIONS: $MNP = (MN)P$
$\lambda x.MN = \lambda x.(MN)$

RULES: EQUATIONAL

(β) $(\lambda x.M)N = M[x:=N]$ CAPTURE-AVOIDING

(η) $\lambda x.Mx = M$ where $x \notin FV(M)$

+ CONTEXTUAL CLOSURE

REDUCTION

(β) $(\lambda x.M)N \longrightarrow M[x:=N]$

(η) $\lambda x.Mx \longrightarrow M$ $\quad x \notin FV(M)$

+ CONTEXTUAL + REFLEXIVE TRANSITIVE CLOSURE

NORMAL FORM: A TERM WHICH CONTAINS NO **REDEX** (I.E. REDUCES NO FURTHER).

HOW PROGRAMMING?

① DATA REPRESENTATION

BOOLEANS:

$$\underline{T} = \lambda xy.x \qquad \underline{F} = \lambda xy.y$$

INTEGERS:

$$\underline{0} = \lambda fx.x$$
$$\underline{1} = \lambda fx.fx \qquad \text{"CHURCH NUMERALS"}$$
$$\underline{2} = \lambda fx.f(fx)$$
$$\vdots$$
$$\underline{n} = \lambda fx.f^{(n)}x \qquad \text{"ITERATORS"}$$

② FUNCTIONS

$\underline{if} = \lambda abc.abc$

$\underline{and} = \lambda ab.ab\underline{F}$

$\underline{or} = \lambda ab.a\underline{T}b$

$\underline{not} = \lambda a.a\underline{F}\underline{T}$

$\underline{succ} = \lambda nfx.f(nfx)$

$\underline{plus} = \lambda nm.n\ \underline{succ}\ m$

$\underline{times} = \lambda nm.n(\underline{plus}\ m)\underline{0}$

$\underline{exp} = \lambda nm.nm$

EXERCISE:

FIND TERMS $\underline{iszero}$ AND $\underline{pred}$ SUCH THAT

$\underline{iszero}\ \underline{0} = \underline{T}$, $\underline{iszero}\ \underline{n+1} = \underline{F}$, $\underline{pred}\ \underline{n+1} = \underline{n}$, $\underline{pred}\ \underline{0} = \underline{0}$

③ CONTROL STRUCTURES

LEMMA. LET $A = \lambda af.f(aaf)$ AND LET

$$\Theta = AA = (\lambda af.f(aaf))(\lambda af.f(aaf))$$

"TURING'S FIXPOINT COMBINATOR".

THEN FOR ANY TERM M:

$$\Theta M \longrightarrow M(\Theta M)$$

THUS, ΘM IS A **FIXPOINT** OF M.

- EVERY FUNCTION HAS A FIXPOINT!
- FIXPOINTS ARE USED TO **SOLVE EQUATIONS**:

TO SOLVE

$$x = \ldots.x\ldots.,$$

LET x BE A FIXPOINT OF

$$\lambda x.\ldots.x\ldots.$$

SO SET

$$x = \Theta(\lambda x.\ldots.x\ldots.)$$

RECURSION IS AN EXAMPLE OF SOLVING AN EQUATION.

FACTORIAL :

$\underline{fact}\ x = \underline{if}\ (\underline{iszero}\ x)\ \underline{1}\ (\underline{times}\ x\ (\underline{fact}\ (\underline{pred}\ x)))$

$\Leftrightarrow$

$\underline{fact} = \lambda x.\ \underline{if}\ (\underline{iszero}\ x)\ \underline{1}\ (\underline{times}\ x\ (\underline{fact}\ (\underline{pred}\ x)))$

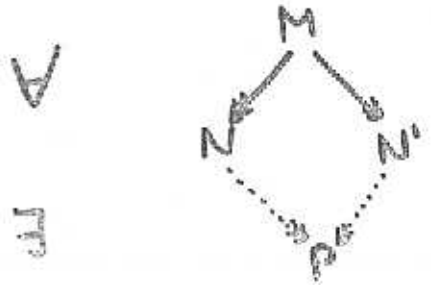
$\Leftarrow$

$\underline{fact} = \Theta\,(\lambda f.\lambda x.\ \underline{if}\ (\underline{iszero}\ x)\ \underline{1}\ (\underline{times}\ x\ (f\ (\underline{pred}\ x))))$

EXERCISE: VERIFY THAT

$\underline{fact}\ \underline{0} \longrightarrow \underline{1}$
$\underline{fact}\ \underline{1} \longrightarrow \underline{1}$
$\underline{fact}\ \underline{2} \longrightarrow \underline{2}$
$\underline{fact}\ \underline{3} \longrightarrow \underline{6}$
etc.

CHURCH-ROSSER PROPERTY, 1936:



COROLLARIES:

- IF $M =_{\beta\eta} N$ THEN THERE EXISTS P WITH $M \twoheadrightarrow_{\beta\eta} P$, $N \twoheadrightarrow_{\beta\eta} P$.
- NORMAL FORMS ARE UNIQUE.

TYPED LAMBDA CALCULUS

- HAS PRIMITIVE DATA + OPERATIONS BUILT IN
- USES TYPES TO ENSURE WELL-DEFINEDNESS OF OPERATIONS

SYNTAX

TYPES: $A, B ::= \text{base types} \mid A \to B \mid A \times B \mid 1$

TERMS: $M, N ::= x \mid MN \mid \lambda x^A.M \mid \langle M, N \rangle \mid \pi_i M \mid * \mid c^A$

TYPING JUDGEMENT:

$$\underbrace{x_1 : A_1, \ldots, x_n : A_n}_{\Gamma} \vdash M : B$$

TYPING RULES:

$$\Gamma, x:A \vdash x:A$$

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

$$\frac{\Gamma, x:A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \to B}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \times B}$$

$$\frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \pi_i M : A_i}$$

$$\Gamma \vdash * : 1$$

PLUS CONSTANTS, E.G.:

$Y : (A \to A) \to A$

$0 : \underline{\text{int}}$

$\underline{\text{plus}} : \underline{\text{int}} \to \underline{\text{int}} \to \underline{\text{int}}$

$\underline{\text{if}} : \underline{\text{bool}} \to (A \to A \to A)$

OPERATIONAL SEMANTICS

PROGRAMMING LANGUAGE =

SYNTAX + OPERATIONAL SEMANTICS

REDUCTION STRATEGY

CALL-BY-NAME (CBN):

SUBSTITUTE BEFORE EVALUATING THE ARGUMENT

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

$$\pi_i \langle M_1, M_2 \rangle \longrightarrow M_i$$

$$\frac{M \to M'}{MN \to M'N} \qquad \frac{M \to M'}{\pi_i M \to \pi_i M'}$$

PLUS SOME RULES FOR CONSTANTS E.G.

$$Y M \to M (Y M)$$

$$\frac{M \to M'}{\underline{\text{plus}}\ MN \to \underline{\text{plus}}\ M'N}$$

$$\frac{N \to N'}{\underline{\text{plus}}\ \underline{m}\ N \to \underline{\text{plus}}\ \underline{m}\ N'}$$

$$\underline{\text{plus}}\ \underline{m}\ \underline{n} \to \underline{m+n}$$

$$\underline{\text{if}}\ \underline{T} \to \lambda xy.x$$

$$\underline{\text{if}}\ \underline{F} \to \lambda xy.y$$

etc.

NOTE:

- NO CONTEXTUAL CLOSURE, REDUCE AT TOP LEVEL
- DON'T REDUCE UNDER A λ OR PAIR
- REDUCTION IS DETERMINISTIC.

A RESULT IS A TERM OF THE FORM:

$$x, \quad \lambda x.M, \quad \langle M,N\rangle, \quad *, \quad c,$$

$$f M_1 \dots M_n \quad \text{when } i < n = \text{arity}(f).$$

NOTE:

- A RESULT DOES NOT REDUCE
- A WELL-TYPED CLOSED TERM WHICH DOES NOT REDUCE IS A RESULT
- A WELL-TYPED CLOSED TERM (PROGRAM) DOES NOT GET STUCK
- A PROGRAM PRODUCES AT MOST ONE RESULT.

LEMMA

SUBJECT REDUCTION

$$\Gamma \vdash M : A, \quad M \to M' \Rightarrow \Gamma \vdash M' : A$$

CALL-BY-VALUE (CBV):

SUBSTITUTE AFTER EVALUATING THE ARGUMENT

VALUES

$$V ::= x \mid \lambda x.M \mid \langle V,V'\rangle \mid * \mid c \mid f c_1 \dots c_i \quad \text{when } i < n$$

REDUCTION RULES:

$$(\lambda x.M)V \to M[x := V]$$

$$\pi_i\langle V_1, V_2\rangle \to V_i$$

$$\frac{M \to M'}{MN \to M'N} \qquad \frac{N \to N'}{VN \to VN'}$$

$$\frac{M \to M'}{\langle M,N\rangle \to \langle M',N\rangle} \qquad \frac{N \to N'}{\langle V,N\rangle \to \langle V,N'\rangle}$$

$$\frac{M \to M'}{\pi_i M \to \pi_i M'}$$

PLUS SOME RULES FOR CONSTANTS E.G.

plus m n $\to$ $m+n$

$$Y(\lambda f x.M)z \to M[f := Y(\lambda f x.M), x := z]$$

$$Y : ((A \to B) \to (A \to B)) \to A \to B$$

NOTE: COMPARE WITH PHIL SCOTT'S LECTURES:

NNO (NATURAL NUMBERS OBJECT):

- PRIMITIVE RECURSION
- ALL FUNCTIONS ARE TOTAL
- STRONG NORMALIZATION
- EQUATIONAL WORLD

FIXPOINT OPERATOR:

- GENERAL RECURSION
- FUNCTIONS ARE PARTIAL
- DIVERGENCE
- OPERATIONAL WORLD

OPERATIONAL EQUIVALENCE

A PROGRAM IS A CLOSED TERM OF GROUND TYPE (E.G. OF TYPE BOOL)

TERMS M,N ARE OPERATIONALLY EQUIVALENT IF THEY BEHAVE "THE SAME" AS PART OF ANY PROGRAM.

FORMALLY:

TERMS M,N ARE OPERATIONALLY EQUIVALENT ($M =_{op} N$) IF FOR ALL GROUND-TYPE, CLOSING CONTEXTS $C[\,]$, AND ALL GROUND TYPE VALUES (CONSTANTS) c,

$$C[M] \twoheadrightarrow c \iff C[N] \twoheadrightarrow c$$

NOTES:
- CONTEXT $C[\,]$ REPRESENTS WHAT CAN BE OBSERVED ABOUT A TERM
- GENERALLY, $C[\,]$ IS FROM THE SAME LANGUAGE AS M,N.
- HOWEVER, IT COULD BE FROM A LARGER LANGUAGE, LEADING TO LESS OPERATIONAL EQ.

THE NOTION OF OPERATIONAL EQUIV. DEPENDS ON THE LANGUAGE OF THE CONTEXTS ALLOWED.

LARGER LANGUAGES HAVE LESS OPERAT. EQUIVALENCE.

EXAMPLE: $and_1 = \lambda ab.\ \underline{if}\ a\ b\ \underline{F}$

$and_2 = \lambda ab.\ \underline{if}\ b\ a\ \underline{F}$

IN A LANGUAGE WITHOUT DIVERGENCE (OR SIDE EFFECTS), HAVE

$$and_1 =_{op} and_2$$

HOWEVER, IF DIVERGENCE IS ADDED (E.G. A TERM Ω S.T. $\Omega \to \Omega$), THEN

$$and_1 \neq_{op} and_2$$

BECAUSE THE CONTEXT $C[\,] = [\,]F\Omega$ DISTINGUISHES THEM.

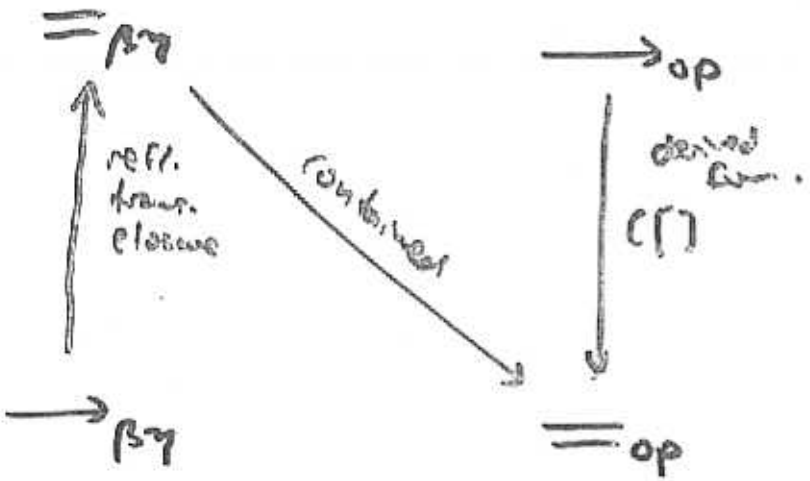
RELATIONSHIP BETWEEN $=_{\beta\eta}$ AND OPERATIONAL SEMANTICS.

CALL-BY-NAME

LEMMA (SOUNDNESS)

$$M =_{\beta\eta} N \Rightarrow M =_{op} N$$

- THUS $\beta\eta$-EQUIVALENCE IS A CONSERVATIVE APPROXIMATION TO OPERATIONAL EQUIVALENCE
- MOREOVER, THIS HOLDS W.R.T. ALL REASONABLE LANGUAGE EXTENSIONS
- COMPILER OPTIMIZATIONS AND PROGRAM TRANSFORMATIONS.



CALL BY VALUE

WE DO NOT HAVE $M =_{\beta\eta} N \Rightarrow M =_{op} N$.

EXAMPLE $(\lambda x.I)\Omega$ DIVERGES

$(\lambda x.I)(\lambda y.\Omega y) \to I$

- THUS $\Omega \neq_{op} \lambda y.\Omega y$

  (η-LAW IS UNSOUND)
- ALSO $(\lambda x.I)\Omega \neq_{op} I$

  (β-LAW IS UNSOUND).

CALL-BY-VALUE EQUIVALENCE
(= MOGGI'S COMPUTATIONAL λ-CALCULUS)

WRITE $\underline{\text{LET}}\ x = N\ \underline{\text{in}}\ M$ FOR $(\lambda x.M)N$

$$
\begin{aligned}
(\lambda x.M)V &= M[V/x] \\
\lambda x.Vx &= V \qquad x \notin FV(V) \\
\pi_i\langle V_1, V_2\rangle &= V_i \\
\langle \pi_1 V, \pi_2 V\rangle &= V \\
* &= V : 1
\end{aligned}
$$

$$
\begin{aligned}
\underline{\text{let}}\ x = M\ \underline{\text{in}}\ x &= M \\
\text{let } y = (\text{let } x = M \text{ in } N) \text{ in } P &= \\
&\quad \text{let } x = M \text{ in let } y = N \text{ in } P \\
MN &= \text{let } x = M \text{ in let } y = N \text{ in } xy \\
(M,N) &= \text{let } x = M \text{ in let } y = N \text{ in } \langle x, y\rangle \\
\pi_i M &= \text{let } x = M \text{ in } \pi_i x
\end{aligned}
$$

THE DISJUNCTIVE $\lambda\mu$-CALCULUS

CONTROL OPERATORS: ALTER THE "NORMAL" FLOW OF CONTROL.

"JUMP" TO SOME OTHER PART OF THE PROGRAM.

TO THE SIMPLY-TYPED λ-CALCULUS, ADD:

- A TYPE $\bot$
  (EMPTY TYPE, TYPE OF A "FUNCTION WHICH DOES NOT RETURN")
- A SET OF "CHANNEL NAMES" $\alpha, \beta, \gamma, \ldots$
  ("CHANNELS", "SIGNALS", "LABELS"...)
- A TERM CONSTRUCTOR $[\alpha]M$
  ("COMPUTE M, THEN SEND THE RESULT ON CHANNEL α")
- A TERM CONSTRUCTOR $\mu\alpha.M$
  (BINDER. "COMPUTE M WHILE LISTENING ON α. ANY VALUE RECEIVED ON α IMMEDIATELY BECOMES THE VALUE OF $\mu\alpha.M$")

CHANNELS α, β, γ ARE TYPED:
THE TYPE OF α IS THE TYPE OF VALUES THAT CAN BE "SENT" ON α.

TYPING JUDGMENTS

$x_1 : B_1, \ldots, x_n : B_n \vdash M : A \mid \alpha_1 : A_1, \ldots, \alpha_m : A_m$

OR BRIEFLY: $\Gamma \vdash M : A \mid \Delta$

$$\frac{\Gamma \vdash M : A \mid \Delta, \alpha : A}{\Gamma \vdash [\alpha] M : \bot \mid \Delta, \alpha : A}$$

$$\frac{\Gamma \vdash M : \bot \mid \Delta, \alpha : A}{\Gamma \vdash \mu \alpha^A . M : A \mid \Delta}$$

ADD DISJUNCTION TYPES $A \vee B$

AND TERMS $[\alpha, \beta] M$, $\mu(\alpha, \beta) . M$

$$\frac{\Gamma \vdash M : A \vee B \mid \Delta, \alpha : A, \beta : B}{\Gamma \vdash [\alpha, \beta] M : \bot \mid \Delta, \alpha : A, \beta : B}$$

$$\frac{\Gamma \vdash M : \bot \mid \Delta, \alpha : A, \beta : B}{\Gamma \vdash \mu(\alpha^A, \beta^B) . M : A \vee B \mid \Delta}$$

NOTE: NOT THE SAME AS "SUM TYPE" $A + B$ DEFINED VIA inl, inr, case.

CONTROL OPERATORS GIVE
CLASSICAL LOGIC. [GRIFFIN '90]

WHY ?

EVIL KING

POOR SHEPHERD

A : PHILOSOPHER'S STONE

⊥ : GOLD

$$A \vee (A \to \bot) \quad ?$$

$$A \vee (A \to \bot)$$

$$\mu(\alpha^{A}, \beta^{A\to\bot}).\left[[\beta]\left(\lambda x^{A}.\,[\alpha]\,x\right)\right]$$

$\lambda x^{A}.\,[\alpha]\,x$: maximize $A \to \bot$

EQUATIONAL THEORY.

THERE ARE <u>SEPARATE</u> EQUATIONAL THEORIES FOR THE CALL-BY-NAME AND FOR THE CALL-BY-VALUE $\lambda\mu$-CALC.:

<u>CBN</u> [PARIGOT '92]

$$(\mu\alpha^{A\to B}.M)N = \mu\beta^{B}.M\left[\frac{[\beta](-)N}{[\alpha](-)}\right]$$

AND OTHERS...

<u>CBV</u> [ONG, STEWART '97]

$$V^{A\to B}(\mu\alpha^{A}.N) = \mu\beta^{B}.N\left[\frac{[\beta]V(-)}{[\alpha](-)}\right]$$

AND OTHERS...

$\mathbb{A}, \mathbb{B}, \mathbb{C}$ CATEGORIES.

A <u>BINOIDAL FUNCTOR</u> F FROM $\mathbb{A}, \mathbb{B}$ TO $\mathbb{C}$ IS GIVEN BY TWO FUNCTORS

$$F_0 : \mathbb{A} \times |\mathbb{B}| \longrightarrow \mathbb{C}$$
$$F_1 : |\mathbb{A}| \times \mathbb{B} \longrightarrow \mathbb{C}$$

SUCH THAT $F_0(A,B) = F_1(A,B)$ FOR ALL OBJECTS A, B.

HAVE $F(A,B)$, $F(f,B)$, $F(A,g)$ BUT <u>NOT</u> $F(f,g)$

THE FOLLOWING MAY NOT COMMUTE:

$$\begin{array}{ccc} F(A,B) & \xrightarrow{F(f,B)} & F(A',B) \\ {\scriptstyle F(A,g)}\downarrow & \not\# & \downarrow{\scriptstyle F(A',g)} \\ F(A,B') & \xrightarrow{F(f,B')} & F(A',B') \end{array}$$

<u>EXAMPLE</u> "CATEGORY OF CONTINUATIONS"

LET $\mathbb{C}$ BE A CATEGORY WITH FINITE PRODUCTS, A DISTINGUISHED OBJECT R, AND EXPONENTIALS OF THE FORM R^A.

LET $R^{\mathbb{C}}$ DENOTE THE FULL SUBCATEGORY OF $\mathbb{C}$ OF OBJECTS OF THE FORM R^A.

DEFINE AN <u>OBJECT MAP</u> $R^A \parr R^B := R^{A \times B}$

FUNCTORIAL IN R^A VIA $R^{A\times B} \cong (R^A)^B$
FUNCTORIAL IN R^B VIA $R^{A\times B} \cong (R^B)^A$

<u>BUT NOT JOINTLY</u>.

HAVE $\bot := R^1$

$$\bot \parr R^A = R^1 \parr R^A = R^{1\times A} \cong R^A$$

$$(R^A \parr R^B) \parr R^C \cong R^{A\times B\times C} \cong R^A \parr (R^B \parr R^C)$$

ETC.

CONTROL CATEGORIES

DEFINITION

A **CONTROL CATEGORY** IS A CARTESIAN-CLOSED CATEGORY $\mathbf{P}$ TOGETHER WITH A SYMMETRIC PRETENSOR $\parr$, $\bot$ AND A MONOID

$$\nabla_A : A \parr A \to A$$
$$i_A : \bot \to A$$

ON EACH OBJECT A, SUCH THAT

1. DISTRIBUTIVITY

$$(A \times B) \parr C \cong (A \parr C) \times (B \parr C)$$
$$1 \parr C \cong 1$$

AND

2. EXPONENTIAL STRENGTH

$$B^A \parr C \cong (B \parr C)^A, \quad \leftarrow \text{CLASSICAL LOGIC}$$

SATISFYING A NUMBER OF COHERENCE AND NATURALITY CONDITIONS.

N.B.: **ALGEBRAIC** STRUCTURE

Table 1. *The signature of control categories*

Nullary morphism constructors:

$id:$	$A \to A$
$\diamond:$	$A \to 1$
$\pi_1:$	$A \times B \to A$
$\pi_2:$	$A \times B \to B$
$\epsilon:$	$B^A \times A \to B$
$a:$	$(A \parr B) \parr C \to A \parr (B \parr C)$
$l:$	$A \to A \parr \bot$
$c:$	$A \parr B \to B \parr A$
$i:$	$\bot \to A$
$\nabla:$	$A \parr A \to A$
$d:$	$(A \parr C) \times (B \parr C) \to (A \times B) \parr C$
$s^{-1}:$	$(B \parr C)^A \to B^A \parr C$

Object constructors:

$$1 \quad \bot \quad A \times B \quad A \to B \quad A \parr B$$

Binary and unary morphism constructors:

$$\frac{f : A \to B \qquad g : B \to C}{g \circ f : A \to C}$$
$$\frac{f : A \to B \qquad g : A \to C}{\langle f, g \rangle : A \to B \times C}$$
$$\frac{f : A \times B \to C}{f^* : A \to C^B}$$
$$\frac{f : A \to B}{f \parr C : A \parr C \to B \parr C}$$

Table 2. *The typing rules for the $\lambda\mu$-calculus*

$$(var)\ \frac{}{\Gamma \vdash x : A \mid \Delta} \quad \text{if } x{:}A \subset \Gamma$$
$$(conv)\ \frac{}{\Gamma \vdash c^A : A \mid \Delta}$$
$$(*)\ \frac{}{\Gamma \vdash * : \top \mid \Delta}$$
$$(pair)\ \frac{\Gamma \vdash M : A \mid \Delta \qquad \Gamma \vdash N : B \mid \Delta}{\Gamma \vdash (M, N) : A \wedge B \mid \Delta}$$
$$(\pi_1)\ \frac{\Gamma \vdash M : A \wedge B \mid \Delta}{\Gamma \vdash \pi_1 M : A \mid \Delta}$$
$$(\pi_2)\ \frac{\Gamma \vdash M : A \wedge B \mid \Delta}{\Gamma \vdash \pi_2 M : B \mid \Delta}$$
$$(app)\ \frac{\Gamma \vdash M : A \to B \mid \Delta \qquad \Gamma \vdash N : A \mid \Delta}{\Gamma \vdash MN : B \mid \Delta}$$
$$(abs)\ \frac{\Gamma, x{:}A \vdash M : B \mid \Delta}{\Gamma \vdash \lambda x^A . M : A \to B \mid \Delta}$$
$$(name)\ \frac{\Gamma \vdash M : A \mid \Delta}{\Gamma \vdash [\alpha]M : \bot \mid \Delta} \quad \text{if } \alpha{:}A \subset \Delta$$
$$(\mu)\ \frac{\Gamma \vdash M : \bot \mid \alpha{:}A, \Delta}{\Gamma \vdash \mu\alpha^A . M : A \mid \Delta}$$
$$(weaken)\ \frac{\Gamma \vdash M : A \mid \Delta}{\Gamma' \vdash M : A \mid \Delta'} \quad \text{if } \Gamma \subseteq \Gamma', \Delta \subseteq \Delta'$$
$$(name')\ \frac{\Gamma \vdash M : A \vee B \mid \Delta}{\Gamma \vdash [\alpha, \beta]M : \bot \mid \Delta} \quad \text{if } \alpha{:}A, \beta{:}B \subset \Delta,$$
$$(\mu')\ \frac{\Gamma \vdash M : \bot \mid \alpha{:}A, \beta{:}B, \Delta}{\Gamma \vdash \mu(\alpha^A, \beta^B).M : A \vee B \mid \Delta}$$

EXAMPLE (CONTINUED):

LET $\mathbb{C}$ AND $R^{\mathbb{C}}$ BE AS BEFORE; ASSUME IN ADDITION THAT $\mathbb{C}$ HAS FINITE (DISTRIBUTIVE) SUMS.

$$R^A \,\&\, R^B = R^{A \times B}$$
$$\underline{1} = R^{\emptyset}$$
$$R^A \times R^B = R^{A+B}$$
$$! = R^0$$
$$(R^B)^{R^A} = R^{B \times R^A}$$

$\nabla : R^A \,\&\, R^A \to R^A$ IS $R^{\Delta} : R^{A \times A} \to R^A$

$i : \underline{1} \to R^A$ IS $R^* : R^1 \to R^A$

DISTRIBUTIVITY: $R^{(A+B)\times C} \cong R^{A \times C + B \times C}$

$R^{0 \times C} \cong R^0$

EXP. STRENGTH: $R^{(B \times R^A) \times C} \cong R^{(B \times C) \times R^A}$

$\rightsquigarrow R^{\mathbb{C}}$ IS A CONTROL CATEGORY.

MAIN THEOREM OF CONTROL CATEGORIES

(STRUCTURE THEOREM)

EVERY CONTROL CATEGORY $\mathbb{P}$ IS EQUIVALENT TO A CATEGORY OF CONTINUATIONS $R^{\mathbb{C}}$.

[cf. HOFMANN '98, FÜHRMANN '98]

THE INTERPRETATION OF THE CALL-BY-NAME λμ-CALCULUS IN A CONTROL CATEGORY.

TYPES:

$$\begin{aligned}
[\![\iota]\!] &= !\\
[\![A \times B]\!] &= [\![A]\!] \times [\![B]\!]\\
[\![\bot]\!] &= \bot\\
[\![A \vee B]\!] &= [\![A]\!] \parr [\![B]\!]\\
[\![A \to B]\!] &= [\![B]\!]^{[\![A]\!]}
\end{aligned}$$

TERMS:

$$[\![x_1 : B_1, \dots, x_n : B_n \triangleright M : A \triangleright \alpha_1 : A_1, \dots, \alpha_m : A_m]\!]$$

$$: B_1 \times \dots \times B_n \to A \parr A_1 \parr \dots \parr A_m$$

E.G.

$$[\![\Gamma \triangleright MN : B \triangleright \Delta]\!] = \Gamma \xrightarrow{\langle M, N \rangle} (B^A \parr \Delta) \times (A \parr \Delta) \xrightarrow{\varepsilon} B \parr \Delta$$

$$[\![\Gamma \triangleright \lambda x : A . M : B \triangleright \Delta]\!] = \Gamma \xrightarrow{M^*} (B \parr \Delta)^A \cong B^A \parr \Delta$$

$$[\![\Gamma \triangleright [\alpha] M : \bot \triangleright \alpha : A, \Delta]\!] = \Gamma \xrightarrow{M} A \parr A \parr \Delta \xrightarrow{\nabla} A \parr \Delta$$

$$[\![\Gamma \triangleright \mu \alpha : A . M : A \triangleright \Delta]\!] = \Gamma \xrightarrow{M} \bot \parr A \parr \Delta \cong A \parr \Delta$$

Table 3. *The CPS translation of the call-by-name λμ-calculus*

$\underline{x}$	$=$	$\lambda k^{K_A} . \bar{x} k$	where $x : A()$
$\underline{\epsilon^A}$	$=$	$\lambda k^{K_A} . \bar{\epsilon} k$	
$\underline{*}$	$=$	$\lambda k^{K_\top} . \sqsubset_R k$	
$\underline{(M,N)}$	$=$	$\lambda k^{K_{A \wedge B}} . [\underline{M}, \underline{N}] k$	where $M : A, N : B$
$\underline{\pi_1 M}$	$=$	$\lambda k^{K_A} . \underline{M} (\mathrm{in}_L k)$	where $M : A \wedge B$
$\underline{\pi_2 M}$	$=$	$\lambda k^{K_B} . \underline{M} (\mathrm{in}_R k)$	where $M : A \wedge B$
$\underline{MN}$	$=$	$\lambda k^{K_B} . \underline{M} (\underline{N}, k)$	where $M : A \to B, N : A$
$\underline{\lambda x^A . M}$	$=$	$\lambda \langle \bar{x}, k \rangle^{K_{A \to B}} . \underline{M} k$	where $M : B$
$\underline{[\alpha] M}$	$=$	$\lambda k^{K_\bot} . \underline{M} \bar{\alpha}$	where $M : A$
$\underline{\mu \alpha^A . M}$	$=$	$\lambda \bar{\alpha}^{K_A} . \underline{M} *$	where $M : \bot$
$\underline{[\alpha, \beta] M}$	$=$	$\lambda k^{K_\bot} . \underline{M} (\bar{\alpha}, \bar{\beta})$	where $M : A \vee B$
$\underline{\mu [\alpha^A, \beta^B] . M}$	$=$	$\lambda \langle \bar{\alpha}, \bar{\beta} \rangle^{K_{A \vee B}} . \underline{M} *$	where $M : \bot$

Table 4. *The interpretation of the call-by-name λμ-calculus in a control category*

$$\begin{aligned}
[\Gamma \vdash x_i : B_i \mid \Delta]_n &= \Gamma \xrightarrow{\pi_i} B_i \xrightarrow{in} B_i \parr \Delta\\
[\Gamma \vdash \epsilon^A : A \mid \Delta]_n &= \Gamma \xrightarrow{\circ} 1 \xrightarrow{\delta} A \xrightarrow{in} A \parr \Delta\\
[\Gamma \vdash * : \top \mid \Delta]_n &= \Gamma \xrightarrow{\circ} 1 \xrightarrow{in} \bot \parr \Delta\\
[\Gamma \vdash (M,N) : A \wedge B \mid \Delta]_n &= \Gamma \xrightarrow{\langle [M]_n, [N]_n \rangle} (A \parr \Delta) \times (B \parr \Delta) \xrightarrow{\delta} (A \times B) \parr \Delta\\
[\Gamma \vdash \pi_1 M : A \mid \Delta]_n &= \Gamma \xrightarrow{[M]_n} (A \times B) \parr \Delta \xrightarrow{\pi_1 \parr \Delta} A \parr \Delta\\
[\Gamma \vdash \pi_2 M : B \mid \Delta]_n &= \Gamma \xrightarrow{[M]_n} (A \times B) \parr \Delta \xrightarrow{\pi_2 \parr \Delta} B \parr \Delta\\
[\Gamma \vdash MN : B \mid \Delta]_n &= \Gamma \xrightarrow{\langle [M]_n, [N]_n \rangle} (B^A \parr \Delta) \times (A \parr \Delta) \xrightarrow{\delta} (B^A \times A) \parr \Delta \xrightarrow{\epsilon \parr \Delta} B \parr \Delta\\
[\Gamma \vdash \lambda x^A . M : A \to B \mid \Delta]_n &= \Gamma \xrightarrow{[M]^*_n} (B \parr \Delta)^A \xrightarrow{\cong} B^A \parr \Delta\\
[\Gamma \vdash [\alpha] M : \bot \mid \Delta]_n &= \Gamma \xrightarrow{[M]_n} A_i \parr \Delta \xrightarrow{\alpha_i \parr \Delta} \Delta \parr \Delta \xrightarrow{\nabla} \Delta \xrightarrow{in} \bot \parr \Delta\\
[\Gamma \vdash \mu \alpha^A . M : A \mid \Delta]_n &= \Gamma \xrightarrow{[M]_n} \bot \parr A \parr \Delta \xrightarrow{\cong} A \parr \Delta\\
[\Gamma \vdash [\alpha_i, \alpha_j] M : \bot \mid \Delta]_n &= \Gamma \xrightarrow{[M]_n} A_i \parr A_j \parr \Delta \xrightarrow{\alpha_i \parr \alpha_j \parr \Delta} \Delta \parr \Delta \parr \Delta \xrightarrow{\nabla \parr \Delta, \nabla} \Delta \xrightarrow{in} \bot \parr \Delta\\
[\Gamma \vdash \mu [\alpha^A, \beta^B] . M : A \vee B \mid \Delta]_n &= \Gamma \xrightarrow{[M]_n} \bot \parr A \parr B \parr \Delta \xrightarrow{\cong} (A \parr B) \parr \Delta
\end{aligned}$$

THEOREM

THE INTERPRETATION OF THE CALL-BY-NAME $\lambda\mu$-CALCULUS IN A CONTROL CATEGORY IS SOUND AND COMPLETE.

MOREOVER, THE CATEGORICAL STRUCTURE IS DEFINABLE BY SYNTAX, AND SO THE CBN $\lambda\mu$-CALCULUS IS AN INTERNAL LANGUAGE FOR CONTROL CATEGORIES.

INTERPRETATION IN A CAT OF CONTINUATIONS R.C. YIELDS HOFMANN-STREICHER CPS TRANSFORM FOR CALL-BY-NAME.

CONTROL CATEGORIES

↓

CPS TRANSLATION

↓

KRIVINE'S ABSTRACT MACHINE

↓

IMPLEMENTATION (ASSEMBLY LANGUAGE)

Table 4: The transitions of the abstract machine

CPS		Abstract Machine		
$\overline{x}k$	$\to \overline{\sigma x}k$	$\{x, \sigma, k\}$	$\to \{M, \tau, k\},$	where $\sigma(x) = M^\tau$.
$\overline{\langle M, N\rangle}(\mathrm{inl}\ k)$	$\to \overline{M}k$	$\{\langle M, N\rangle, \sigma, \mathrm{fst}::k\}$	$\to \{M, \sigma, k\}$	
$\overline{\langle M, N\rangle}(\mathrm{inr}\ k)$	$\to \overline{N}k$	$\{\langle M, N\rangle, \sigma, \mathrm{snd}::k\}$	$\to \{N, \sigma, k\}$	
$\overline{\pi_1 M}k$	$\to \overline{M}(\mathrm{inl}\ k)$	$\{\pi_1 M, \sigma, k\}$	$\to \{M, \sigma, \mathrm{fst}::k\}$	
$\overline{\pi_2 M}k$	$\to \overline{M}(\mathrm{inr}\ k)$	$\{\pi_2 M, \sigma, k\}$	$\to \{M, \sigma, \mathrm{snd}::k\}$	
$\overline{MN}k$	$\to \overline{M}(\overline{N}, k)$	$\{MN, \sigma, k\}$	$\to \{M, \sigma, N^\sigma::k\}$	
$\overline{\lambda x^A.M}(N, k)$	$\to \overline{M[N/x]}k$	$\{\lambda x.M, \sigma, N^\tau::k\}$	$\to \{M, \sigma(x \to N^\tau), k\}$	
$\overline{[\alpha]M}k$	$\to \overline{M}\alpha$	$\{[\alpha]M, \sigma, k\}$	$\to \{M, \sigma', k'\},$	where $\sigma(\alpha) = k'$,
$\overline{\mu\alpha.M}k$	$\to \overline{M[k/\alpha]}*$	$\{\mu\alpha.M, \sigma, k\}$	$\to \{M, \sigma(\alpha \to k), nil\}$	

After having changed the notation for continuations, we will now also change the notation for computations, i.e., for translated terms. In order to avoid having to do substitutions, we introduce the notion of a closure. A *closure* is a pair M^σ of a term M and an environment σ. An *environment* for M is a map from the free variables of M to closures, and from the free names of M to continuation (i.e. stacks). An environment

CPS → ABSTRACT MACHINE

KRIVINE MACHINE:

STATES $\langle M, \sigma, k\rangle$, WHERE:

M : A TERM

σ : AN ENVIRONMENT (MAPS VARIABLES TO CLOSURES, NAMES TO STACKS)

k : A STACK.

CPS

$\overline{x}k \to xk$

$\overline{\langle M,N\rangle}(\mathrm{inl}\ k) \to \overline{M}k$

$\overline{\langle M,N\rangle}(\mathrm{inr}\ k) \to \overline{N}k$

$\overline{\pi_1 M}k \to \overline{M}(\mathrm{inl}\ k)$

$\overline{\pi_2 M}k \to \overline{M}(\mathrm{inr}\ k)$

$\overline{MN}k \to \overline{M}(\overline{N}, k)$

$\overline{\lambda x^A.M}(N,k) \to \overline{M[N/x]}k$

$\overline{[\alpha]M}k \to \overline{M}\alpha$

$\overline{\mu\alpha.M}k \to \overline{M[k/\alpha]}*$

MACHINE

$\langle x,\sigma,k\rangle \to \langle M,\tau,k\rangle$ $(\sigma(x)=M^\tau)$

$\langle MN,\sigma,k\rangle \to \langle M,\sigma,N^\sigma::k\rangle$

$\langle \lambda x.M,\sigma,N^\tau::k\rangle \to \langle M,\sigma(x\to N^\tau),k\rangle$

$\langle \pi_i M,\sigma,k\rangle \to \langle M,\sigma,i::k\rangle$

$\langle [\alpha]M,\sigma,k\rangle \to \langle M,\sigma',k'\rangle$ $(\sigma(\alpha)=k')$

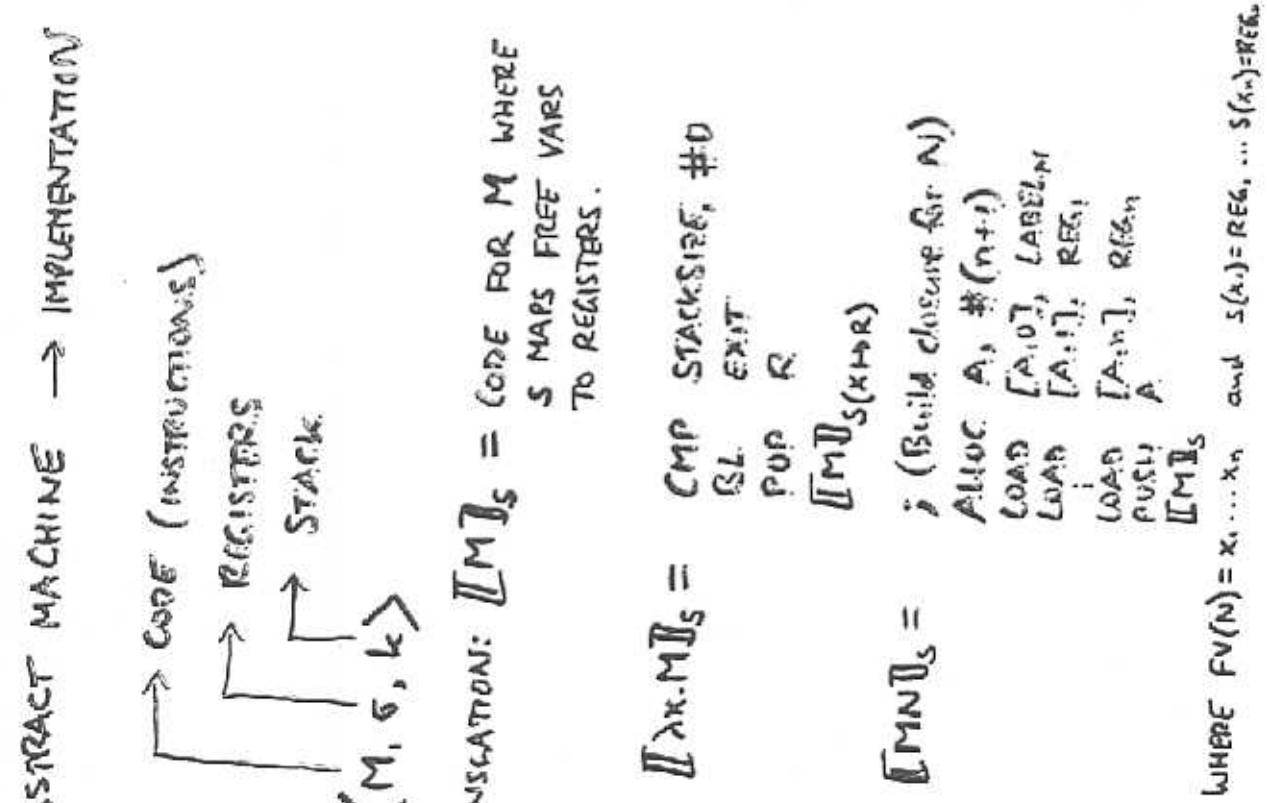
$\langle \mu\alpha.M,\sigma,k\rangle \to \langle M,\sigma(\alpha\to k),nil\rangle$

Instruction	Meaning
MOVE w, v	Store the value v in location w
ADD w, v	Add v to w
PUSH v	Push the value v onto the stack
POP w	Pop a value from the stack and store it in w
CMP v_1, v_2	Compare the two values, and remember the result
BNE v	If previous CMP resulted in not equal, jump to location v
BGE v	If previous CMP resulted in greater than or equal jump to location v
JUMP v	Jump to address v
CALL v	Call subroutine at address v
ALLOC w, v	Allocate v words of memory and store a pointer to them in w
SAVE w	Make a stack closure from the current stack, and store a pointer to it in w
RESTORE v	Replace the current stack by a copy of the stack closure pointed to by v
EXIT v	Exit with result v

Table 5: Instruction set of the idealized assembly language

there are infinitely many *registers* $R_1, R_2, \ldots$, as well as four distinguished registers SP, SS, C, and V, each of which can hold a word.

We assume that there are infinitely many addressable memory cells, each of which holds a word. A *memory reference* takes the form $[R, n]$, where R is a register and n is a literal integer. The expression $[R, n]$ refers to the contents of the memory cell at



Lambda calculus

$[x]_s =$

```
MOVE   C, s(x)
JUMP   [C, 0]
```

$[\lambda x.M]_s =$

```
      CMP    SS, #0
      BNE    l
      EXIT   "function"
l :   POP    R
      [M]_{s(x↦R)}
```

(where l is a fresh label and R is a fresh register.)

$[MN]_s =$

```
      ; build closure for N
      ALLOC  R, #(n + 1)
      MOVE   [R, 0], #l
      MOVE   [R, 1], s(x_1)
      ...
      MOVE   [R, n], s(x_n)
      PUSH   R
      [M]_s
l :   [N]_{(x_1↦[C,1],...,x_n↦[C,n])}
```

(where l is a fresh label, R is a fresh register, and $FV(N) \cup FN(N) = \{x_1, \ldots, x_n\}$.)

$[\langle M, N\rangle]_s =$

```
      CMP    SS, #0
      BNE    l_1
      EXIT   "pair"
l_1 : POP    R
      CMP    R, #1
      BNE    l_2
      [M]_s
l_2 : [N]_s
```

(where l_1, l_2 are fresh labels, and R is a fresh register.)

$[\pi_i M]_s =$

```
      PUSH   #i
      [M]_s
```

$[*]_s =$

```
      EXIT   "unit"
```

$\lambda\mu$-Calculus

$[\mu\alpha.M]_s =$

```
      ; build a stack closure
      SAVE   R
      ; clear the stack
      MOVE   SP, #0
      [M]_{s(α↦R)}
```

(where R is a fresh register.)

$[[\alpha]M]_s =$

```
      RESTORE s(α)
      [M]_s
```

Classical disjunction ($\lambda\mu\nu$-calculus)

$[\nu\alpha.M]_s =$

```
      CMP    SS, #0
      BNE    l
      EXIT   "disjunction"
l :   POP    R
      [M]_{s(α↦R)}
```

(where l is a fresh label and R is a fresh register.)

$[\langle\alpha\rangle M]_s =$

```
      PUSH   s(α)
      [M]_s
```

Table 6: The compilation of terms

Basic constants

$[n]_s =$

```
      MOVE   V, #n
      CMP    SS, #0
      BNE    l
      EXIT   V
l :   POP    R
      JUMP   R
```

(where l is a fresh label and R is a fresh register.)

$[true]_s = [1]_s$

$[false]_s = [0]_s$

"Pure" basic functions

$[f]_s =$

```
      ; check for sufficient arguments
      CMP    SS, #n
      BGE    f_0
      EXIT   "function"
f_0 : MOVE   C, [SP, 1]
      PUSH   #f_1
      JUMP   [C, 0]
```

(repeat following code for $j = 1 \ldots n-1$)

```
f_j : MOVE   [SP, j], V
      MOVE   C, [SP, j + 1]
      PUSH   #f_{j+1}
      JUMP   [C, 0]
```

```
f_n : MOVE   [SP, n], V
      ; n values are now on top of stack
      CALL   native_f
      POP    V
      CMP    SS, #0
      BNE    l
      EXIT   V
f_{n+1} : POP  R
      JUMP   R
```

(where $n \geq 1$ is the arity of f, $f_0, \ldots f_{n+1}$ are fresh labels, and R is a fresh register.)

Some "impure" basic functions

$[if]_s =$

```
      CMP    SS, #1
      BGE    if_3
      EXIT   "function"
if_0 : POP   C
      PUSH   #if_1
      JUMP   [C, 0]
if_1 : CMP   V, #0
      BNE    if_2
      [λxy.y]_0
if_2 : [λxy.x]_0
```

(where $if_0, \ldots if_{n+1}$ are fresh labels.)

$[print]_s =$

```
      CMP    SS, #1
      BGE    print_0
      EXIT   "function"
print_0 : POP  C
      PUSH   #print_1
      JUMP   [C, 0]
print_1 : PUSH V
      CALL   native_print
      [λx.x]_0
```

(where $print_0$, $print_1$ are fresh labels.)

Table 7: The compilation of terms, continued

TOWARDS DUALITY : CO-CONTROL CATEGORIES.

DEFINITION A CO-CONTROL CATEGORY IS THE DUAL OF A CONTROL CATEGORY. WE WRITE :

CONTROL CAT		CO-CONTROL CAT
1	$\Rightarrow$	0
$A \times B$	$\Rightarrow$	$A + B$
$\bot$	$\Rightarrow$	I
$A \parr B$	$\Rightarrow$	$A \otimes B$
B^A	$\Rightarrow$	B_A
$B \multimap A$	$\Rightarrow$	$A \multimap B$

$\bot^{A^B} = B \multimap A = I_{A_B} = \neg(A \otimes \neg B)$

IN A CO-CONTROL CATEGORY, WE WRITE $\neg A$ FOR $\blacksquare_A$.

REMARK A CO-CONTROL CATEGORY IS A $\neg\otimes$-CATEGORY OF THIELECKE, ENHANCED WITH FINITE COPRODUCTS.

... a product $A \otimes B$ with unit I, and a weak closed structure $A \multimap B$. The following table lists some notation that we are going to use for objects and morphisms of a co-control category:

On objects		On morphisms	
Control	Co-control	Control categories	Co-control categories
1	0	$\Diamond : A \to 1$	$\Box : 0 \to A$
$A \times B$	$A + B$	$i : \bot \to A$	$\iota : A \to I$
$\bot$	I	$\nabla : A \parr A \to A$	$\Delta : A \to A \otimes A$
$A \parr B$	$A \otimes B$	$\epsilon : B^A \times A \to B$	$\flat : B \to B_A + A$
B^A	B_A	$f^* : C \to B^A$	$\sharp f : B_A \to C$
$B \circ\!\!- A$	$A \multimap B$	$coapp : B \to (B \circ\!\!- A) \parr A$	$app : (A \multimap B) \otimes A \to B$
		$cocurry(f) : (B \circ\!\!- A) \to C$	$curry(f) : C \to (A \multimap B)$

We use the usual notation for coproducts. Following Thielecke (1997), the dual of the map θ is called *thunk*, and the dual of ∂ is called *force*. The remaining structural maps keep the same names as their duals.

Every co-control category is a $\otimes\neg$-category in the sense of Thielecke (1997), where $\neg A$ is defined as $I_A \cong A \multimap 0$. Notice that one has $B_A \cong (\neg A) \otimes B$ and $A \multimap B \cong \neg(A \otimes \neg B)$; thus each two of the constructs B_A, $A \multimap B$, and $\neg A$ can be defined in terms of the third. Conversely, one can show that every $\otimes\neg$-category can be fully and faithfully embedded in a co-control category. Thus, co-control categories can be seen as a natural extension of $\otimes\neg$-categories with finite coproducts. The presence of finite coproducts is important for the duality result in Section 8.

THE INTERPRETATION OF THE CALL-BY-VALUE λμ-CALCULUS IN A CO-CONTROL CATEGORY.

Types:

$$[\![\,,\,]\!] = \mathbb{I}$$
$$[\![A\times B]\!] = [\![A]\!]\otimes[\![B]\!]$$
$$[\![\top]\!] = 0$$
$$[\![A\vee B]\!] = [\![A]\!]+[\![B]\!]$$
$$[\![A\to B]\!] = [\![A]\!]\multimap[\![B]\!]$$

Terms:

$$[\![\ x_1:B_1,\ldots,x_n:B_n \rhd M:A \rhd \alpha_1:A_1,\ldots,\alpha_m:A_m\ ]\!]$$
$$: B_1\otimes\cdots\otimes B_n \to A + A_1 + \cdots + A_m$$

E.g.

$$*[\![\Gamma\rhd MN\rhd\Delta]\!] = \Gamma_\Delta \xrightarrow{\Delta} \Gamma_\Delta\otimes\Gamma_\Delta \xrightarrow{{}^*[\![\Gamma]\!]\otimes id} (A\multimap B)\otimes\Gamma_\Delta \xrightarrow{id\otimes{}^*[\![N]\!]} (A\multimap B)\otimes A \xrightarrow{app} B$$

$$*[\![\Gamma\rhd\lambda x^A.M\rhd\Delta]\!] = \Gamma_\Delta \xrightarrow{curry(f)} A\multimap B \xrightarrow{{}^*[\![M]\!]} B$$
where $f=\Gamma_\Delta\otimes A\xrightarrow{\sim}(\Gamma\otimes A)_\Delta$

$$[\![\Gamma\rhd[\alpha]M:\bot\rhd\Delta]\!] = \Gamma\xrightarrow{M} A+A+\Delta \xrightarrow{\nabla} A+\Delta$$

$$[\![\Gamma\rhd\mu\alpha^A.M:A\rhd\Delta]\!] = \Gamma\xrightarrow{M} B+A+\Delta \xrightarrow{\sim} A+\Delta$$

Table 8. *The interpretation of the call-by-value λμ-calculus in a co-control category*

$$
\begin{array}{lcl}
[\![\Gamma\vdash x_i:B_i \mid \Delta]\!]_v & = & \Gamma\xrightarrow{w} B_i\xrightarrow{[i]} B_i+\Delta\\
[\![\Gamma\vdash v:A \mid \Delta]\!]_v & = & \Gamma\xrightarrow{!} I\xrightarrow{x} A\xrightarrow{[i]} A+\Delta\\
[\![\Gamma\vdash *:\top \mid \Delta]\!]_v & = & \Gamma\xrightarrow{!} I\xrightarrow{[i]} I+\Delta\\
{}^*[\![\Gamma\vdash\langle M,N\rangle:A\wedge B \mid \Delta]\!]_v & = & \Gamma_\Delta\xrightarrow{\Delta}\Gamma_\Delta\otimes\Gamma_\Delta\xrightarrow{{}^*[\![M]\!]_v\otimes id} A\otimes\Gamma_\Delta\xrightarrow{id\otimes{}^*[\![N]\!]_v} A\otimes B\\
[\![\Gamma\vdash\pi_1 M:A \mid \Delta]\!]_v & = & \Gamma\xrightarrow{[\![M]\!]_v}(A\otimes B)+\Delta\xrightarrow{\pi+id} A+\Delta\\
[\![\Gamma\vdash\pi_2 M:B \mid \Delta]\!]_v & = & \Gamma\xrightarrow{[\![M]\!]_v}(A\otimes B)+\Delta\xrightarrow{\pi+id} B+\Delta\\
{}^*[\![\Gamma\vdash MN:B \mid \Delta]\!]_v & = & \Gamma_\Delta\xrightarrow{\Delta}\Gamma_\Delta\otimes\Gamma_\Delta\xrightarrow{({}^*[\![M]\!]_v\otimes id)\,;\,(id\otimes{}^*[\![N]\!]_v)}(A\multimap B)\otimes A\xrightarrow{app} B\\
{}^*[\![\Gamma\vdash\lambda x^A.M:A\to B \mid \Delta]\!]_v & = & \Gamma_\Delta\xrightarrow{curry(f)}A\multimap B,\ \text{where } f=\Gamma_\Delta\otimes A\xrightarrow{\cong}(\Gamma\otimes A)_\Delta\xrightarrow{{}^*[\![M]\!]_v} B\\
[\![\Gamma\vdash[\alpha_i]M:\bot \mid \Delta]\!]_v & = & \Gamma\xrightarrow{[\![M]\!]_v}A_i+\Delta\xrightarrow{in_i+\Delta}\Delta+\Delta\xrightarrow{\nabla}\Delta\xrightarrow{\cong}0+\Delta\\
[\![\Gamma\vdash\mu\alpha^A.M:A \mid \Delta]\!]_v & = & \Gamma\xrightarrow{[\![M]\!]_v}0+A+\Delta\xrightarrow{\cong}A+\Delta\\
[\![\Gamma\vdash[\alpha_i,\alpha_j]M:\bot \mid \Delta]\!]_v & = & \Gamma\xrightarrow{[\![M]\!]_v}A_i+A_j+\Delta\xrightarrow{in_i+in_j+\Delta}\Delta+\Delta+\Delta\xrightarrow{\nabla+\Delta\,;\,\nabla}\Delta\xrightarrow{\cong}0+\Delta\\
[\![\Gamma\vdash\mu(\alpha^A,\beta^B).M:A\vee B \mid \Delta]\!]_v & = & \Gamma\xrightarrow{[\![M]\!]_v}0+A+B+\Delta\xrightarrow{\cong}(A+B)+\Delta
\end{array}
$$

THEOREM SOUND + COMPLETE AND:

THE CALL-BY-VALUE $\lambda_{\mu\tilde{\mu}}$-CALCULUS IS AN INTERNAL LANGUAGE FOR CO-CONTROL CATEGORIES.

AGAIN, BY INTERPRETING THE LANGUAGE IN A CAT OF CONTINUATIONS (P.C.), WE OBTAIN A CPS TRANSLATION.

$\overline{x}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\,.\,k\,\overline{x}$	where $x : A$
$\overline{v^{\mathrm{A}}}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\,.\,k\,v$	
$\overline{*}$	$=$	$\lambda k^{\mathrm{K}}\top\,.\,k_*$	
$\overline{\langle M, N\rangle}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\wedge\mathrm{B}\,.\,\overline{M}(\lambda m^{\forall}\mathrm{A}\,.\,\overline{N}(\lambda n^{\forall}\mathrm{B}\,.\,k(m,n)))$	where $M : A, N : B$
$\overline{\pi_1 M}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\,.\,\overline{M}(\lambda m^{\forall}\mathrm{A}\wedge\mathrm{B}\,.\,k\pi_1 m)$	where $M : A \wedge B$
$\overline{\pi_2 M}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{B}\,.\,\overline{M}(\lambda m^{\forall}\mathrm{A}\wedge\mathrm{B}\,.\,k\pi_2 m)$	where $M : A \wedge B$
$\overline{MN}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{B}\,.\,\overline{M}(\lambda m^{\forall}\mathrm{A}\to\mathrm{B}\,.\,\overline{N}(\lambda n^{\forall}\mathrm{A}\,.\,m(n,k)))$	where $M : A \to B, N : A$
$\overline{\lambda x^{\mathrm{A}}\,.\,M}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\to\mathrm{B}\,.\,k(\lambda(\tilde{x}, c)^{\forall}\mathrm{A}\times^{\mathrm{K}}\mathrm{B}\,.\,\overline{M}c)$	where $M : B$
$\overline{[\alpha]M}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\,.\,\overline{M}\tilde{\alpha}$	where $M : A$
$\overline{\mu\alpha^{\mathrm{A}}\,.\,M}$	$=$	$\lambda\tilde{\alpha}^{\mathrm{K}}\mathrm{A}\,.\,\overline{M}*$	where $M : \bot$
$\overline{[\alpha,\beta]M}$	$=$	$\lambda k^{\mathrm{K}}\mathrm{A}\,.\,\overline{M}[\tilde{\alpha},\tilde{\beta}]$	where $M : A \vee B$
$\overline{\mu(\alpha^{\mathrm{A}}, \beta^{\mathrm{B}})\,.\,M}$	$=$	$\lambda[\tilde{\alpha},\tilde{\beta}]^{\mathrm{K}}\mathrm{A}\vee\mathrm{B}\,.\,\overline{M}*$	where $M : A$

Table 7. *The CPS translation of the call-by-value $\lambda\mu$-calculus*

Table 8. *The interpretation of the call-by-value $\lambda\mu$-calculus in a co-control category*

SUMMARY:

- THE CALL-BY-NAME $\lambda\mu$-CALCULUS IS AN INTERNAL LANGUAGE FOR CONTROL CATEGORIES
- THE CALL-BY-VALUE $\lambda\mu$-CALCULUS IS AN INTERNAL LANGUAGE FOR CO-CONTROL CATEGORIES

COROLLARY

IN THE PRESENCE OF PRODUCTS, DISJUNCTION, AND CONTROL,

CBN AND CBV ARE ISOMORPHIC,

IN A NON-TRIVIAL WAY.

(CF. FILINSKI 1989)

Table 11. *The syntactic translation from call-by-value to call-by-name*

On types:

$$
\begin{array}{lcl}
(\!|\sigma|\!) & = & \sigma, \quad \text{where } \sigma \text{ is a type constant} \\
(\!|\top|\!) & = & \bot \\
(\!|A \wedge B|\!) & = & (\!|A|\!) \vee (\!|B|\!) \\
(\!|\bot|\!) & = & \top \\
(\!|A \vee B|\!) & = & (\!|A|\!) \wedge (\!|B|\!) \\
(\!|A \to B|\!) & = & ((\!|B|\!) \to (\!|A|\!)) \to \bot
\end{array}
$$

On terms:

$$
\begin{array}{lcl}
(\!|x|\!) & = & \lambda\kappa^{(\!|A|\!)}.[x]\kappa \\
(\!|e^A|\!) & = & \lambda\kappa^{(\!|A|\!)}.[e^{(\!|A|\!)}]\kappa \\
(\!|\star|\!) & = & \lambda\kappa^{(\!|\top|\!)}.\kappa \\
(\!|\langle M,N\rangle|\!) & = & \lambda\kappa^{(\!|A\wedge B|\!)}.(\!|M|\!)(\mu x^{(\!|A|\!)}.(\!|N|\!)(\mu y^{(\!|B|\!)}.[x,y]\kappa)) \\
(\!|\pi_1 M|\!) & = & \lambda\kappa^{(\!|A|\!)}.(\!|M|\!)(\mu(x^{(\!|A|\!)},y^{(\!|B|\!)}).[x]\kappa) \\
(\!|\pi_2 M|\!) & = & \lambda\kappa^{(\!|B|\!)}.(\!|M|\!)(\mu(x^{(\!|A|\!)},y^{(\!|B|\!)}).[y]\kappa) \\
(\!|\lambda x^A.M|\!) & = & \lambda\kappa^{(\!|A\to B|\!)}.\kappa(\lambda\beta^B.\mu x^A.(\!|M|\!)\beta) \\
(\!|MN|\!) & = & \lambda\kappa^{(\!|B|\!)}.(\!|M|\!)(\lambda\gamma^{(\!|B|\!)\to(\!|A|\!)}.(\!|N|\!)(\gamma\kappa)) \\
(\!|[\alpha]M|\!) & = & \lambda\kappa^{(\!|\bot|\!)}.(\!|M|\!)\alpha \\
(\!|[e^A]M|\!) & = & \lambda\kappa^{(\!|\bot|\!)}.(\!|M|\!)e^{(\!|A|\!)} \\
(\!|\mu\alpha^A.M|\!) & = & \lambda\kappa^{(\!|A|\!)}.(\lambda\alpha^{(\!|A|\!)}.(\!|M|\!)\star)\kappa \\
(\!|\langle\alpha,\beta\rangle M|\!) & = & \lambda\kappa^{(\!|\top|\!)}.(\!|M|\!)\langle\alpha,\beta\rangle \\
(\!|\mu[\alpha^A,\beta^B].M|\!) & = & \lambda\kappa^{(\!|A\vee B|\!)}.(\lambda\alpha^A.\lambda\beta^B.(\!|M|\!)\star)(\pi_1\kappa)(\pi_2\kappa)
\end{array}
$$

Table 12. *The syntactic translation from call-by-name to call-by-value*

On types:

$$
\begin{array}{lcl}
(\!|\sigma|\!) & = & \sigma, \quad \text{where } \sigma \text{ is a type constant} \\
(\!|\top|\!) & = & \bot \\
(\!|A \wedge B|\!) & = & (\!|A|\!) \vee (\!|B|\!) \\
(\!|\bot|\!) & = & \top \\
(\!|A \vee B|\!) & = & (\!|A|\!) \wedge (\!|B|\!) \\
(\!|A \to B|\!) & = & ((\!|A|\!) \to \bot) \wedge (\!|B|\!)
\end{array}
$$

On terms:

$$
\begin{array}{lcl}
(\!|x|\!) & = & \lambda\kappa^{(\!|A|\!)}.[x]\kappa \\
(\!|e^A|\!) & = & \lambda\kappa^{(\!|A|\!)}.[e^{(\!|A|\!)}]\kappa \\
(\!|\star|\!) & = & \lambda\kappa^{(\!|\top|\!)}.\kappa \\
(\!|\langle M,N\rangle|\!) & = & \lambda\kappa^{(\!|A\wedge B|\!)}.(\!|M|\!)(\mu x^{(\!|A|\!)}.(\!|N|\!)(\mu y^{(\!|B|\!)}.[x,y]\kappa)) \\
(\!|\pi_1 M|\!) & = & \lambda\kappa^{(\!|A|\!)}.(\!|M|\!)(\mu(x^{(\!|A|\!)},y^{(\!|B|\!)}).[x]\kappa) \\
(\!|\pi_2 M|\!) & = & \lambda\kappa^{(\!|B|\!)}.(\!|M|\!)(\mu(x^{(\!|A|\!)},y^{(\!|B|\!)}).[y]\kappa) \\
(\!|\lambda x^A.M|\!) & = & \lambda\kappa^{(\!|A\to B|\!)}.(\pi_1\kappa)(\mu x^{(\!|A|\!)}.(\!|M|\!)(\pi_2\kappa)) \\
(\!|MN|\!) & = & \lambda\kappa^{(\!|B|\!)}.(\!|M|\!)(\langle\!|N|\!|,\kappa\rangle) \\
(\!|[\alpha]M|\!) & = & \lambda\kappa^{(\!|\bot|\!)}.(\!|M|\!)\alpha \\
(\!|[e^A]M|\!) & = & \lambda\kappa^{(\!|\bot|\!)}.(\!|M|\!)e^{(\!|A|\!)} \\
(\!|\mu\alpha^A.M|\!) & = & \lambda\kappa^{(\!|A|\!)}.(\lambda\alpha^{(\!|A|\!)}.(\!|M|\!)\star)\kappa \\
(\!|\langle\alpha,\beta\rangle M|\!) & = & \lambda\kappa^{(\!|\top|\!)}.(\!|M|\!)\langle\alpha,\beta\rangle \\
(\!|\mu[\alpha^A,\beta^B].M|\!) & = & \lambda\kappa^{(\!|A\vee B|\!)}.(\lambda\alpha^A.\lambda\beta^B.(\!|M|\!)\star)(\pi_1\kappa)(\pi_2\kappa)
\end{array}
$$

Some remarks.

a) In presence of first-class continuations, function types are definable from negation:

CBV: $A \to B \cong \neg(A \wedge \neg B)$

CBN: $A \to B \cong \neg A \vee B$

b) Compare with "intuitionistic" sum type $A+B$ defined via inl/inr/case:

CBV: $A + B \cong A \vee B$

CBN: $A + B \cong \neg\neg A \vee \neg\neg B$
$\cong \neg(\neg A \wedge \neg B)$

c) "names" vs. variables:

CBV: $$\frac{x:\neg A,\ \Gamma \vdash M:B \mid \Delta}{\Gamma \vdash M:B \mid \Delta, \alpha:A}$$

CBN: $$\frac{x:A,\ \Gamma \vdash M:B \mid \Delta}{\Gamma \vdash M:B \mid \Delta, \alpha:\neg A}$$

Some recent work:

- Laurent ['00]:

  Control cats $\Leftrightarrow$
  proof nets for polarized linear logic

- Hasegawa, Kakutani ['01];
  based on Filinski ['89]:

  CBV recursion $\Leftrightarrow$ CBV iteration
  $\Leftrightarrow$ CBN recursion

- Melliès, Selinger ['01]:

  control cats from game semantics via "polarized categories"
  (also explains Blass games).