Formalizing Modal Embeddings of Call-by-Name and Call-by-Value

Floris Reuvers

Supervisor: dr. N.M. van der Weide
Second Reader: dr. E.G.M. Hubbers

June 17, 2025

Radboud University

Research Question

How can the unification of call-by-name and call-by-value evaluation strategies using modal logic be formalised in Agda?

Introduction

Context

- Functional Programming Languages
- Evaluation Strategies
  - Call-by-value (cbv): OCaml/F#/SML
  - Call-by-name (cbn)/Lazy Evaluation: Haskell/Clean
- Idea of cbn and cbv
  - cbv: first evaluate the argument, then the beta
  - cbn: evaluate the beta

Call-by-name

Let f be defined as

$f(x) = x * x$

Call-by-name

Let f be defined as

$f(x) = x * x$

cbn and cbv evaluation of f (3 + 3)

Call-by-name

Let f be defined as

$f(x) = x * x$

cbn and cbv evaluation of f (3 + 3)

cbn: **cvb:**

Call-by-name

Let f be defined as

$f(x) = x * x$

cbn and cbv evaluation of f (3 + 3)

cbn:	**cvb:**
$f(3+3) \to (3+3) * (3+3)$	$f(3+3) \to f(6)$

Call-by-name

Let f be defined as

$f(x) = x * x$

cbn and cbv evaluation of f (3 + 3)

cbn:

$$\begin{aligned} f(3+3) &\to (3+3)*(3+3) \\ &\to 6*(3+3) \end{aligned}$$

cvb:

$$\begin{aligned} f(3+3) &\to f(6) \\ &\to 6*6 \end{aligned}$$

Call-by-name

Let f be defined as

$f(x) = x * x$

cbn and cbv evaluation of f (3 + 3)

cbn:	**cvb:**
$f(3+3) \to (3+3)*(3+3)$	$f(3+3) \to f(6)$
$\to 6*(3+3)$	$\to 6*6$
$\to 6*6$	$\to 36$

Call-by-name

Let f be defined as

$f(x) = x * x$

cbn and cbv evaluation of f (3 + 3)

cbn:

$$
\begin{aligned}
f(3+3) &\to (3+3)*(3+3) \\
&\to 6*(3+3) \\
&\to 6*6 \\
&\to 36
\end{aligned}
$$

cvb:

$$
\begin{aligned}
f(3+3) &\to f(6) \\
&\to 6*6 \\
&\to 36
\end{aligned}
$$

Reasons to unify cbn and cbv

- Reason 1
- Reason 2
- Reason 3

Approaches to unification

- Modal logic
- Linear logic
- Thunks

Background

Grammar Lambda Calculus (λ-calculus)

$$A \quad ::= \quad X \mid A \to A' \qquad M, N, P, Q \quad ::= \quad x \mid \lambda x.M \mid MN$$

Example terms

Grammar Lambda Calculus (λ-calculus)

$$A \quad ::= \quad X \mid \; A \to A' \qquad M, N, P, Q \quad ::= \quad x \mid \; \lambda x.M \mid \; MN$$

Example terms

x

Grammar Lambda Calculus (λ-calculus)

$$A \quad ::= \quad X \mid A \to A' \qquad M, N, P, Q \quad ::= \quad x \mid \lambda x.M \mid MN$$

Example terms

x

$\lambda x.x$

Grammar Lambda Calculus (λ-calculus)

$$A \quad ::= \quad X \mid A \to A' \qquad M, N, P, Q \quad ::= \quad x \mid \lambda x.M \mid MN$$

Example terms

x

$\lambda x.x$

$\lambda y.yz$

Grammar Lambda Calculus (λ-calculus)

$$A \quad ::= \quad X \mid A \to A' \qquad M, N, P, Q \quad ::= \quad x \mid \lambda x.M \mid MN$$

Example terms

x

$\lambda x.x$

$\lambda y.yz$

$(\lambda y.y)z$

Grammar Lambda Calculus (λ-calculus)

$$A \quad ::= \quad X \mid A \to A' \qquad M, N, P, Q \quad ::= \quad x \mid \lambda x.M \mid MN$$

Example terms

x

$\lambda x.x$

$\lambda y.yz$

$(\lambda y.y)z$

$\lambda x.\lambda t.xt(\lambda s.w)wy$

Call-by-name and call-by-value λ-calculus

Lorem ipsum

Closure Rules

Here we define closure rules

Evaluation Relations

Lorem ipsum

Grammer

$$A \quad ::= \quad X \mid B \to A \mid B \qquad\qquad B \quad ::= \quad \Box A$$

$$M, N, P, Q \quad ::= \quad \varepsilon(x) \mid \lambda x.M \mid MN \mid \mathsf{box}(N)$$

Evaluation Relations of λ_b

Lorem ipsum

Embeddings into λ_b

Girard's Translation

Lorem ipsum

Gödel's Translation

Lorem ipsum

Challenges of Formalisation

Overview Challenges

- Variables
- Ill typed terms
- Formal definition of raise

De Bruijn Indices

Lorem ipsum

Restriction to well-typed terms

Lorem ipsum

Formal definition of raise

Lorem ipsum

Propositions

Girard's Translation

Lorem ipsum

Gödel's Translation

Lorem ipsum

Conclusion

Conclusion

Lorem ipsum