

Operational properties of interest

- Termination and Confluence

Existence and uniqueness of normal forms

- How to Compute

reduction strategies with good properties:

- standardization,
- normalization

Strategies

1

Normalization

► **Def.** $(A, \rightarrow)$ is strongly (weakly, uniformly) normalizing if each $t \in A$ is, where the three normalization notions are as follows.

- t is strongly $\rightarrow$ -normalizing: every maximal $\rightarrow$ -sequence from t ends in a normal form.
- t is weakly $\rightarrow$ -normalizing: there exist a $\rightarrow$ -sequence from t which ends in a normal form.
- t is uniformly $\rightarrow$ -normalizing: t weakly $\rightarrow$ -normalizing implies t strongly $\rightarrow$ -normalizing.

If terms are not strongly normalizing, how do we compute a normal form, or even test if any exists? This is the problem tackled by *normalization*. By repeatedly performing only specific steps $\rightarrow$, we are guaranteed that a normal form will eventually be computed, if any exists.

3

Factorization (aka weak Standardization)

another commutation!

5

2

Normalizing strategies

► **Def.** $(A, \rightarrow)$ is strongly (weakly, uniformly) normalizing if each $t \in A$ is, where the three normalization notions are as follows.

- t is strongly $\rightarrow$ -normalizing: every maximal $\rightarrow$ -sequence from t ends in a normal form.
- t is weakly $\rightarrow$ -normalizing: there exist a $\rightarrow$ -sequence from t which ends in a normal form.
- t is uniformly $\rightarrow$ -normalizing: t weakly $\rightarrow$ -normalizing implies t strongly $\rightarrow$ -normalizing.

► **Def.**

- $\rightarrow$ is a **strategy** for $\rightarrow$ if $\rightarrow \subseteq \rightarrow$, and it has the same normal forms as $\rightarrow$.
- It is a **normalizing strategy** for $\rightarrow$ if whenever $t \in A$ has $\rightarrow$ -normal form, then every maximal $\rightarrow$ -sequence from t ends in normal form.

4

Factorization

(aka Semi-Standardization, Postponement, or often simply Standardization)

- most basic property about *how to compute*

$$t \xrightarrow{\beta}^* u \Rightarrow t \xrightarrow{h}^* \cdot \xrightarrow{-h}^* u \quad \text{head factorization}$$

A **key building-block** in proofs of more sophisticated *how-to-compute* properties:

- allows immediate proofs of **normalization**
(a reduction strategy reaches a normal form, whenever one exists)
- simplest way to prove **standardization**, by using Mitschke's argument
(left-to-right standardization = iterate head factorization)

7

Factorization

(aka Semi-Standardization, Postponement, or often simply Standardization)

Melliès 97:

the **meaning of factorization** is that the **essential** part of a computation can always be separated from its junk.

Assume computations consists of

- steps $\rightarrow_e$ which are in some sense *essential*, and
- steps $\rightarrow_i$ which are not.

Factorization says that every rewrite sequence can be reorganized/factorized as a sequence of **essential steps followed by inessential ones**.

$$t \rightarrow^* u \Rightarrow t \xrightarrow[e]{*} \cdot \xrightarrow[i]{*} u \quad \text{e-factorization}$$

8

Call-by-Name and Call-by-Value λ -calculus: **terms**

Terms and *values* are generated by the following grammars

$$\begin{array}{ll} V & ::= x | \lambda x.M \quad (\text{Values, } \mathcal{V}) \\ M & ::= x | c | \lambda x.M | MM \quad (\text{Terms}) \end{array}$$

where x ranges over a countable set of *variables*, and c over a disjoint (possibly empty) set $\mathcal{O}$ of constants.

- If the set of constants is empty, the calculus is *pure*, and the set of terms is denoted Λ .
- Otherwise, the calculus is called *applied*, and the set of terms is often indicated as $\Lambda_{\mathcal{O}}$.

Terms are identified up to renaming of bound variables, where λx is the only binder constructor. $P\{Q/x\}$ is the capture-avoiding substitution of Q for the free occurrences of x in P .

10

Reduction = contextual closure of a rule

Contexts (with one hole $\langle \rangle$) are generated as follows. $\mathbf{C}\langle M \rangle$ stands for the term obtained from $\mathbf{C}$ by replacing the hole with the term M (possibly capturing free variables of M).

$$\mathbf{C} ::= () \mid () \mid M\mathbf{C} \mid \mathbf{C}M \mid \lambda x.\mathbf{C} \quad (\text{Contexts})$$

- A rule ρ is a binary relation on Λ_O , which we also denote $\mapsto_\rho$, writing $R \mapsto_\rho R'$. R is called a ρ -redex.

The best known rule is β :

$$(\lambda x.M)N \mapsto_{\beta} M\{N/x\}$$

- A **reduction step** $\rightarrow_\rho$ is the closure under context $\mathbf{C}$ of ρ .
Explicitly, $T \rightarrow_\rho T'$ holds if $T = \mathbf{C}(\llbracket R \rrbracket)$, $T' = \mathbf{C}(\llbracket R' \rrbracket)$, and $R \rightarrow_\rho R'$.

12

Call-by-Name and Call-by-Value λ -calculus

CALL-BY-NAME, CALL-BY-VALUE AND THE λ -CALCULUS

G. D. PLOTKIN

*Department of Machine Intelligence, School of Artificial Intelligence, University of Edinburgh
Edinburgh, United Kingdom*

Contributed by R. Milner

Abstract. This paper examines the old question of the relationship between *INTJ* and *INTJ*, using the distinction between call-by-value and call-by-name. It is held that the relationship should be mediated by a transformation theorem. Since this leads to difficulties, a new *lambda*-calculus is introduced whose evaluation theorem gives a good correspondence with *lambda*-calculus as given by the *RECD* machine, but without the latter flaws. Thus a call-by-name version of *INTJ* is introduced which is the analogue of *lambda*-calculus as with the usual *lambda*-calculus. The relation between call-by-value and call-by-name is then studied by giving simulations of one language by the other and interpretations of each calculus in the other. There are obtained several applications of the conversion technique. Some examples are placed throughout in sections of optional reading (or optional skipping). If users are so present need is satisfied they are immediately ready for the corresponding language. Unfortunately, occasional apolo-

9

Contexts (with one hole $\langle \rangle$) are generated as follows. $\mathbf{C}\langle M \rangle$ stands for the term obtained from $\mathbf{C}$ by replacing the hole with the term M (possibly capturing free variables of M).

$$\mathbf{C} ::= () \mid () \mid M\mathbf{C} \mid \mathbf{C}M \mid \lambda x.\mathbf{C} \quad (\text{Contexts})$$

11

Call-by-Name vs Call-by-Value λ -calculus

- The λ -calculus can be seen both as an equational theory on terms and as an abstract model of computation.

- With the functional paradigm point of view, the meaning of any λ -term is the value it evaluates to.

13

Call-by-Name and Call-by-Value λ -calculus

CbN and CbV Calculi.

- The (pure) **Call-by-Name** calculus $\Lambda^{\text{cbn}} = (\Lambda, \rightarrow_{\beta})$ is the set of terms equipped with the contextual closure of the β -rule.

$$(\lambda x.M)N \mapsto_{\beta} M\{N/x\}$$

- The (pure) **Call-by-Value** calculus $\Lambda^{\text{cbv}} = (\Lambda, \rightarrow_{\beta_v})$ is the same set equipped with the contextual closure of the β_v -rule.

$$(\lambda x.M)V \mapsto_{\beta_v} M\{V/x\} \text{ where } V \in \mathcal{V}$$

weak evaluation

- According to the function paradigm of computation the goal of every computation is to determine its value
- Since functions are seen as values, it is natural to consider *weak evaluation*. In practical implementations, weak evaluation is more realistic than the full beta reduction

14

CbV: Weak Reduction

Weak reductions in CbV

The result of interest are **values** (*i.e.* functions).

In languages, in general the reduction is *weak*, that is, it does not reduce in the body of a function.

There are three main weak schemes: left, right and in arbitrary order.

Left contexts **L**, *right* contexts **R**, and (arbitrary order) *weak* contexts **W** are defined by

$$\begin{aligned} \mathbf{L} &::= () \mid \mathbf{L}M \mid V\mathbf{L} \\ \mathbf{R} &::= () \mid M\mathbf{R} \mid \mathbf{R}V \\ \mathbf{W} &::= () \mid \mathbf{W}M \mid M\mathbf{W} \end{aligned}$$

Q: Which reductions are deterministic?

Given a rule $\mapsto$ on Λ , *weak reduction* $\mapsto_{\mathbf{W}}$ is the closure of $\mapsto$ under context **W**.

A step $T \rightarrow S$ is non-weak, written $T \not\rightarrow_{\mathbf{W}} S$ if it is not weak. Similarly for left ($\mapsto_{\mathbf{L}}$ and $\not\rightarrow_{\mathbf{L}}$), and right ($\mapsto_{\mathbf{R}}$ and $\not\rightarrow_{\mathbf{R}}$).

► **Fact 3 (Weak normal forms).** Given M a closed term, M is $\rightarrow$ -normal iff M is a value.

18

- Classical key result (e.g. in Barendregt 84 book)

in Call-by-Name:

- Head Factorization: $\rightarrow_{\beta}^* \subseteq \rightarrow_{\mathbf{h}\beta}^* \cdot \rightarrow_{\mathbf{h}}^* \beta^*$.
- Head Normalization: M has hnf if and only if $M \rightarrow_{\mathbf{h}\beta}^* S$ (for some $S \in \mathcal{H}$).

- Classical key result [Plotkin 75]

in Call-by-Value:

- Let $s \in \{w, l, r\}$
- weak factorization of $\rightarrow_{\beta_v}$: $\rightarrow_{\beta_v}^* \subseteq \rightarrow_{s\beta_v}^* \cdot \rightarrow_{s\mathbf{h}}^* \beta^*$.
- Convergence: $T \rightarrow_{\beta_v} W (W \in \mathcal{V})$ if and only if $T \rightarrow_{s\beta_v}^* V (V \in \mathcal{V})$

17

CbV: Left contexts **L**, right contexts **R**, and (arbitrary order) weak contexts **W** are defined by

$$\begin{aligned} \mathbf{L} &::= () \mid \mathbf{L}M \mid V\mathbf{L} \\ \mathbf{R} &::= () \mid M\mathbf{R} \mid \mathbf{R}V \\ \mathbf{W} &::= () \mid \mathbf{W}M \mid M\mathbf{W} \end{aligned}$$

The closure under **L** (resp. **W**, **R**) context is noted $\mapsto_{\mathbf{L}}$ (resp. $\mapsto_{\mathbf{W}}$, $\mapsto_{\mathbf{R}}$)

► **Fact 3 (Weak normal forms).** Given M a closed term, M is $\rightarrow$ -normal iff M is a value.

Question.: which of the above reductions are deterministic?

Questions. [True or False?] Let M be a closed term.

- $M \rightarrow_{\mathbf{L}}^* V$ iff $M \rightarrow_{\mathbf{L}}^* V$. True?
- $M \rightarrow_{\mathbf{W}}^* V$ iff $M \rightarrow_{\mathbf{L}}^* V$. True?
- Assume you proved $M \rightarrow_{\mathbf{L}}^k V$ (runtime is k). Does the sequence of $\rightarrow_{\mathbf{L}}$ -steps also terminates? Can we say how long does it take?
- Assume you proved $M \rightarrow_{\mathbf{L}}^k V$ (runtime is k). What about $\rightarrow_{\mathbf{W}}$?

19

CbN: Head Reduction

Head reduction in CbN

Head reduction is the closure of β under head context

$$\lambda x_1 \dots x_n. () \mid M_1 \dots M_k$$

Head normal forms (hnf), whose set is denoted by $\mathcal{H}$, are its normal forms.

- Given a rule ρ , we write $\rightarrow_{\mathbf{h}\rho}$ for its closure under head context.
- A step $\rightarrow_{\rho}$ is non-head, written $\rightarrow_{\mathbf{h}\rho}$ if it is not head.

What about?

$$\mathbf{H} ::= () \mid \lambda x. \mathbf{H} \mid \mathbf{H}M$$

20

21

CbN Head Factorization

Head Factorization

Head factorization allows for a characterization of the terms which have head normal form, that is M has *hnf* if and only if $\rightarrow_{\text{h}}$ -reduction from M terminates.

► Theorem 2 (Head Factorization).

- Head Factorization: $\rightarrow_{\text{h}}^* \subseteq \rightarrow_{\text{h}}^* \cdot \rightarrow_{\text{h}}^* \cdot \rightarrow_{\text{h}}^*$.
- Head Normalization: M has hnf if and only if $M \rightarrow_{\text{h}}^* S$ (for some $S \in \mathcal{H}$).

22

CbV Weak Factorization

Weak Factorization.

Let $s \in \{w, l, r\}$

- weak factorization of $\rightarrow_{\beta_v}$: $\rightarrow_{\beta_v}^* \subseteq \rightarrow_{\beta_v}^* \cdot \rightarrow_{\beta_v}^*$.
- Convergence: $T \rightarrow_{\beta_v} W (W \in \mathcal{V})$ if and only if $T \rightarrow_{\beta_v}^* V (V \in \mathcal{V})$

► Corollary 4. Given M a closed term, M has a β_v -reduction to a value, if and only if the $\rightarrow_{\beta_v}$ -reduction from M terminates.

23

Question. Let M be a closed term. We say that M returns a value when $M \rightarrow_{\beta_v}^* V$, for some V . Prove :

- M returns value, if and only if $\rightarrow_{\beta_v}$ -reduction from M terminates.

24

Basic properties of the contextual closure

25

Basic properties of contextual closure

If a step $T \rightarrow_{\gamma} T'$ is obtained by closure under *non-empty context* of a rule $\mapsto_{\gamma}$, then T and T' have the same shape, i.e. both terms are an application (resp. an abstraction, a variable).

► Fact 5 (Shape preservation).

- Assume $T = \mathbf{C}(\langle R \rangle) \rightarrow \mathbf{C}(\langle R' \rangle) = T'$ and that the context $\mathbf{C}$ is non-empty. Then T and T' have the same shape.
- Hence, for any internal step $M \rightarrow_{\beta} M'$ ($s \in \{h, w, l, r, \dots\}$) M and M' have the same shape.

The following is an easy to verify consequence.

► Lemma 6 (Redexes preservation).

1. CbN: Assume $T \rightarrow_{\beta} S$. T is a β -redex iff so is S .
2. CbV. Assume $T \rightarrow_{\beta_v} S$. T is a β_v -redex iff so is S .

26

Internal steps preserve head and weak normal nf

Fixed a set of redexes $\mathcal{R}$, M is w-normal (resp. h-normal) if there is no redex $R \in \mathcal{R}$ such that $M = \mathbf{W}(\langle R \rangle)$ (resp. $M = \mathbf{H}(\langle R \rangle)$)

► Lemma 7 (Surface normal forms). 1. CbN. Let $\mathcal{R}$ be the set of β -redexes.

Assume $M \rightarrow_{\beta} M'$. M is h-normal $\Leftrightarrow M'$ is h-normal.

2. CbV. Let $\mathcal{R}$ be the set of β_v -redexes.

Assume $M \rightarrow_{\beta_v} M'$. M is w-normal $\Leftrightarrow M'$ is w-normal.

27