Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 360 (2006) 289–326

Theoretical Computer Science

www.elsevier.com/locate/tcs

Investigations on the Dual Calculus

Nikos Tzevelekos*

Computing Lab, Oxford University, Wolfson Building, Parks Road, Oxford OX13QD, UK

Received 7 June 2005; received in revised form 24 January 2006; accepted 24 April 2006

Communicated by G.D. Plotkin

Abstract

The Dual Calculus, proposed recently by Wadler, is the outcome of two distinct lines of research in theoretical computer science:

(A) Efforts to extend the Curry–Howard isomorphism, established between the simply-typed lambda calculus and intuitionistic logic, to classical logic.

(B) Efforts to establish the tacit conjecture that call-by-value (CBV) reduction in lambda calculus is dual to call-by-name (CBN) reduction.

This paper initially investigates relations of the Dual Calculus to other calculi, namely the simply-typed lambda calculus and the Symmetric lambda calculus. Moreover, Church–Rosser and Strong Normalization properties are proven for the calculus' CBV reduction relation. Finally, extensions of the calculus to second-order types are briefly introduced.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Dual Calculus; Classical lambda calculi; Curry–Howard isomorphism; Continuations

1. Introduction

1.1. Two lines of research leading to the Dual Calculus

The Dual Calculus, proposed by Wadler in [20], is the outcome of two distinct lines of research in theoretical computer science:

1. Efforts to extend the Curry–Howard isomorphism, established between the simply-typed lambda calculus and intuitionistic logic, to classical logic.
2. Efforts to establish the tacit conjecture that call-by-value (CBV) reduction in lambda calculus is dual to call-by-name (CBN) reduction.

Regarding the first line of investigation, the Curry–Howard isomorphism correlates two seemingly diverse scientific fields, namely proof theory and type theory. It states a correspondence between systems of formal logic and

* Tel.: +44 7792969142.
E-mail address: nikt@comlab.ox.ac.uk.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.04.009

computational calculi: logic formulas are related to types, and logic proofs are related to typed terms. More than that, proof normalization is related to term reduction. This correspondence allows to use methods and properties of the one field for the other, and it leads to a deeper understanding of foundational matters in theoretical computer science.

Traditionally, classical logic was not taken into account in the Curry–Howard isomorphism (see, for example, [11,17]). The first attempt to add classical constructs to a computational calculus is present in the work of Griffin [12], who defined a simply-typed lambda calculus in which the law of double-negation elimination was expressed in the typing rules. Griffin's rule would read:

> If M is a term of type $\neg\neg A$, then $\mathcal{C}(M)$ is a term of type A,

$\mathcal{C}$ is a control operator [1] which adds further expressive power to the simply-typed lambda calculus by allowing for some non-trivial jumps in computation. For example, using $\mathcal{C}$ we can define the `call/cc` operator of Scheme language.

After the work of Griffin, the view that classical constructs could be used to extend programming control features which would otherwise not be expressible in logical terms became increasingly widespread. Parigot [13] refined the idea of Griffin to a more concrete calculus, the $\lambda\mu$-calculus. This calculus is an extension of lambda calculus where one has the ability to name arbitrary subterms of a term by μ-variables and to abstract on them. Thus, operations can be applied directly to subterms of a term and control features such as $\mathcal{C}$ can be easily simulated in the $\lambda\mu$-calculus. Using this "naming mechanism", Parigot was able to derive a typed $\lambda\mu$-calculus corresponding to a natural deduction system with multiple conclusions. This latter system, called Classical Natural Deduction, is a system of classical logic.

A different approach was taken by Barbanera and Berardi [1], who proposed the Symmetric λ-calculus (Sλ-calculus), a classical simply-typed lambda calculus equipped with the following set of types:

$$\text{Type } D ::= \perp \mid A, \quad \text{m-Type } A, B ::= X \mid X^{\perp} \mid A \vee B \mid A \wedge B$$

with X standing for type variables. Thus, negation is a primitive type constructor in this calculus, yet constrained only to type variables. Negation is extended to all types by the usual De Morgan laws, and thus the authors manage to identify any m-type A with $A^{\perp\perp}$. Hence, having the law of double negation embedded in the syntax, this calculus corresponds to propositional classical logic. Nevertheless, it does not follow closely the computational paradigm, where double negation forms an embedding-projection pair, rather than an isomorphism. Further, investigation on the Sλ-calculus to second-order types was done by Parigot [14].

Regarding the second line of investigation, the notion of "duality" between CBV and CBN reduction was first suggested by Filinski [7]. Filinski defined a symmetric lambda calculus (SLC) in which there exist two distinct syntactic classes: *values* and *continuations*. The notion of a continuation was a well established one at the time:

> in any computation being part of a program there is some "rest of the program" ready to absorb the result of the given computation and *continue* with execution of following commands.

This "rest of the program" is called a continuation [18]. Thus, there is some kind of duality (or symmetry) between values and continuations in programming languages, in that values yield data, whereas continuations absorb data. This duality is part of SLC and Filinski suggests that a similar kind of duality holds between CBV and CBN reduction (or evaluation) strategies for SLC.

The suggestions of Filinski were established by Selinger in [16] (first published in 1998), who worked in the $\lambda\mu$-calculus. Selinger showed categorical duality between CBV and CBN reduction in the $\lambda\mu$-calculus by use of control categories to model the CBN semantics and co-control categories to model the CBV semantics.

Finally, Curien and Herbelin [4] defined the $\bar{\lambda}\mu\tilde{\mu}$-calculus, which is an extension of the $\lambda\mu$-calculus having duals for λ- and μ-abstraction. In order to type these dual abstractions, a difference $(-)$ type constructor is included in the typed version of the calculus, as difference is the De Morgan dual of implication—even though the operational interpretation of difference is not very intuitive. For this calculus it was shown that CBV is dual to CBN in a De Morgan fashion.

[1] In fact, $\mathcal{C}$ was introduced by Felleisen; see more, for example, in [6].

1.1.1. Summary

This paper investigates on the Dual Calculus of Wadler. In this first section we present the Dual Calculus and its reduction relations, and examine its relation to other computational calculi. In the second section we investigate syntactic properties of the Dual Calculus under CBV reduction, namely the Church–Rosser and Strong Normalization properties. These are original contributions and, in rough lines, follow and extend standard techniques from the literature. Finally, in the third section we introduce extensions of the calculus to second-order types.

1.2. The Dual Calculus

The lines of investigation presented above lead to the Dual Calculus [20], which epitomizes both properties of being the Curry–Howard equivalent of classical logic and of its CBV and CBN reduction relations being De Morgan duals. More than that, the calculus has the advantage of simplicity in syntax and operational semantics.

1.2.1. Definitions

The Dual Calculus (DuCa) consists of *types* and *objects*, in the same way that the simply-typed lambda calculus consists of types and terms. The types are the same as the formulas of propositional logic, whereas the objects are divided into *terms*, *coterms* and *statements*. The intended computational interpretation is this of terms being objects yielding data, whereas coterms absorb data. In fact, this is very similar to the notion of *values* and *continuations*, as presented in [7]. The statements of DuCa represent *cuts* of terms upon coterms, that is constructions consisting of a term and a coterm, where the term is yielding data to be absorbed by the coterm.

Below we give the definition of DuCa we will be using throughout this paper.

Definition 1.1 (*Dual Calculus, Walder [21]*). DuCa consists of Types and Objects. The set of objects is the union of the sets of Terms, Coterms and Statements:

$$\begin{array}{lll}
\text{Type} & A, B & ::= X \mid A\&B \mid A \vee B \mid \neg A \\
\text{Object} & G, H & ::= M \mid K \mid S \\
\text{Term} & M, N & ::= x \mid \langle M, N\rangle \mid \langle M\rangle\mathtt{inl} \mid \langle N\rangle\mathtt{inr} \mid [K]\mathtt{not} \mid (S).\alpha \\
\text{Coterm} & K, L & ::= \alpha \mid [K, L] \mid \mathtt{fst}[K] \mid \mathtt{snd}[L] \mid \mathtt{not}\langle M\rangle \mid x.(S) \\
\text{Statement} & S, T & ::= M \bullet K
\end{array}$$

The typing rules involve three forms of sequents:

$$\begin{array}{rl}
\text{Left sequent} & K : A \,\rule{1pt}{8pt}\, \Gamma \rightarrow \Theta \\
\text{Right sequent} & \Gamma \rightarrow \Theta \,\rule{1pt}{8pt}\, M : A \\
\text{Center sequent} & \Gamma \,\rule{1pt}{8pt}\, M \bullet K \mapsto \Theta
\end{array}$$

where Γ and Θ are *antecedent* and *succedent* sets, respectively:

$$\begin{array}{lll}
\text{Antecedent} & \Gamma, \Delta & ::= \{x_1 : A_1, \ldots , x_n : A_n\} \\
\text{Succecedent} & \Theta, \mathrm{I} & ::= \{\alpha_1 : B_1, \ldots , \alpha_m : B_m\}
\end{array}$$

with all x_i's and α_i's disjoint. We will usually omit the outer brackets in succedent and antecedent sets and use comma notation for disjoint union (e.g. $\Gamma, \Gamma' \equiv \Gamma \uplus \Gamma'$).

The typing rules of DuCa are:

$$\frac{}{\alpha : A \mid \Gamma \rightarrow \Theta, \alpha : A}\,\mathtt{id}L \qquad \frac{}{x : A, \Gamma \rightarrow \Theta \mid x : A}\,\mathtt{id}R$$

$$\frac{K : A \mid \Gamma \rightarrow \Theta}{\mathtt{fst}[K] : A\&B \mid \Gamma \rightarrow \Theta} \quad \frac{L : B \mid \Gamma \rightarrow \Theta}{\mathtt{snd}[L] : A\&B \mid \Gamma \rightarrow \Theta}\,\&L \qquad \frac{\Gamma \rightarrow \Theta \mid M : A \quad \Gamma \rightarrow \Theta \mid N : B}{\Gamma \rightarrow \Theta \mid \langle M, N\rangle : A\&B}\,\&R$$

$$\frac{\Gamma \rightarrow \Theta \mid M : A}{\Gamma \rightarrow \Theta \mid \langle M\rangle\mathtt{inl} : A \vee B} \quad \frac{\Gamma \rightarrow \Theta \mid N : B}{\Gamma \rightarrow \Theta \mid \langle N\rangle\mathtt{inr} : A \vee B}\,\vee R \qquad \frac{K : A \mid \Gamma \rightarrow \Theta \quad L : B \mid \Gamma \rightarrow \Theta}{[K, L] : A \vee B \mid \Gamma \rightarrow \Theta}\,\vee L$$

$$\frac{\Gamma \rightarrow \Theta \mid M : A}{\mathtt{not}\langle M\rangle : \neg A \mid \Gamma \rightarrow \Theta}\,\neg L \qquad \frac{K : A \mid \Gamma \rightarrow \Theta}{\Gamma \rightarrow \Theta \mid [K]\mathtt{not} : \neg A}\,\neg R$$

$$\frac{x : A, \Gamma \mid S \mapsto \Theta}{x.(S) : A \mid \Gamma \rightarrow \Theta}\,LI \qquad \frac{\Gamma \mid S \mapsto \Theta, \alpha : A}{\Gamma \rightarrow \Theta \mid (S).\alpha : A}\,RI$$

$$\frac{\Gamma \rightarrow \Theta \mid M : A \quad K : A \mid \Gamma \rightarrow \Theta}{\Gamma \mid M \bullet K \mapsto \Theta}\,\mathrm{Cut}$$

The above rules form the sequent calculus GW.

Note that there is no need to include rules for Weakening or Contraction, since these can be derived from the above.

The letters we use to denote components of DuCa are standard in this paper. As above, for types we use capital letters opening the alphabet ($A, B, C, \ldots$); for variables we use $x, y, z, \ldots$; for terms M, N; for covariables $\alpha, \beta, \gamma, \ldots$; for coterms K, L; for objects G, H; for antecedent sets Γ, Δ; for succedent sets Θ, I.

In DuCa, variable and covariable abstractions are performed by dots '.'. For example, the rule *LI* introduces the variable abstraction $x.(S)$. Consequently, we have the usual convention for free and bound occurrences of variables and covariables. For example, in $x.(S)$ all occurrences of x inside S are bound, while the occurrence of x right before the dot is transparent. Similar things hold for covariables.

The intended computational interpretation of objects in DuCa is as aforementioned: terms stand for computations yielding data, whereas coterms stand for computations absorbing data. Thus, sequents stand for computational scenarios where one supplies data to all variables (and coterms) of the sequent, and expects the computation to pass data to some covariable (or a term to yield data).

Under this interpretation, a term x trivially yields the data supplied to x and a term $\langle M, N\rangle$ yields a pair of data of type $A\&B$, consisting of the data yielded by terms M and N. Hence, the type conjunction $A\&B$ corresponds to product of types. Dually, a coterm α absorbs the data passed to α and a coterm $[K, L]$ absorbs a datum of type $A \vee B$, which is passed on to K or L according to this data being a left or right injection. Therefore, $A \vee B$ corresponds to sum of types. For further details on the intended computational interpretation see [20].

Note that the inference rules of GW are very similar to the inference rules of system LK of Gentzen [9] restricted to propositional logic.[2] This similarity is in fact a Curry–Howard isomorphism. We can see this more clearly if we introduce the following abbreviations:

$$\begin{aligned} A \supset B &\equiv \neg(A\&\neg B) \\ \lambda x.M &\equiv [z.(z \bullet \mathtt{fst}[x.(z \bullet \mathtt{snd}[\mathtt{not}\langle M\rangle])])]\mathtt{not} \\ N@L &\equiv \mathtt{not}\langle\langle N, [L]\mathtt{not}\rangle\rangle \end{aligned} \tag{1}$$

and thus $A \supset B$ is a type, $\lambda x.M$ a term, and $N@L$ a coterm of DuCa.

For these constructs we have the following familiar inference rules being derivable in GW:

$$\frac{\Gamma \rightarrow \Theta \mid M : A \quad K : B \mid \Gamma \rightarrow \Theta}{M@K : A \supset B \mid \Gamma \rightarrow \Theta}\supset L \qquad \frac{x : A, \Gamma \rightarrow \Theta \mid M : B}{\Gamma \rightarrow \Theta \mid \lambda x.M : A \supset B}\supset R$$

[2] One may also note that symbols for logical connectives follow Gentzen's formulations.

1.2.2. Reduction relations

In [20] two reduction relations are proposed for DuCa, representing CBV and CBN approaches. In fact, both these relations are restrictions of a basic reduction relation, which we present below.

Definition 1.2 (*Basic reduction R_b*). R_b is the one-step reduction relation yielded by the reduction rules listed below, when these are applied on subobjects of DuCa objects:

$$\begin{array}{lll}
(\beta\&_1) & \langle M, N\rangle \bullet \mathtt{fst}[K] & \to M \bullet K, \\
(\beta\&_2) & \langle M, N\rangle \bullet \mathtt{snd}[L] & \to N \bullet L \\
(\beta\vee_1) & \langle M\rangle\mathtt{inl} \bullet [K, L] & \to M \bullet K \\
(\beta\vee_2) & \langle N\rangle\mathtt{inr} \bullet [K, L] & \to N \bullet L \\
(\beta\neg) & [K]\mathtt{not} \bullet \mathtt{not}\langle M\rangle & \to M \bullet K \\
(\beta L) & M \bullet x.(S) & \to S\{M/x\} \\
(\beta R) & (S).\alpha \bullet K & \to S\{K/\alpha\} \\
 & & \\
(\eta L) & K & \to x.(x \bullet K) \\
(\eta R) & M & \to (M \bullet \alpha).\alpha.
\end{array}$$

$$\begin{array}{lll}
(\varsigma\&_1) & \langle M, N\rangle \bullet K & \to M \bullet x.(\langle x, N\rangle \bullet K), \\
(\varsigma\&_2) & \langle M, N\rangle \bullet K & \to N \bullet y.(\langle M, y\rangle \bullet K), \\
(\varsigma\vee_3) & \langle M\rangle\mathtt{inl} \bullet K & \to M \bullet x.(\langle x\rangle\mathtt{inl} \bullet K), \\
(\varsigma\vee_4) & \langle N\rangle\mathtt{inr} \bullet K & \to N \bullet y.(\langle y\rangle\mathtt{inr} \bullet K), \\
(\varsigma\vee_1) & M \bullet [K, L] & \to (M \bullet [\alpha, L]).\alpha \bullet K, \\
(\varsigma\vee_2) & M \bullet [K, L] & \to (M \bullet [K, \beta]).\beta \bullet L, \\
(\varsigma\&_3) & M \bullet \mathtt{fst}[K] & \to (M \bullet \mathtt{fst}[\alpha]).\alpha \bullet K, \\
(\varsigma\&_4) & M \bullet \mathtt{snd}[L] & \to (M \bullet \mathtt{snd}[\beta]).\beta \bullet L,
\end{array}$$

For $G, H \in$ DuCa, $(G, H) \in R_b$ is usually written as $G \longrightarrow^b H$.

In rules βL and βR we notice the introduction of *substitutions*: the statement $S\{M/x\}$ is obtained from S if we substitute [3] M for all the free occurrences of x in S, and similarly for $S\{K/\alpha\}$. In η- and ς-rules, x, y, α and β are all fresh.

Note that we will often use explicit notation in reduction arrows to indicate the reduction rule responsible for a reduction (reduction step). For example, if $G \longrightarrow^b H$ because of some redex $\langle M, N\rangle \bullet \mathtt{fst}[K]$ inside G reducing to $M \bullet K$, we may write $G \overset{\beta\&_1}{\longrightarrow}{}^b H$.

The β-reduction rules listed above correspond to familiar cut elimination techniques for LK (see, for example, [11]), and demonstrate further the Curry–Howard isomorphism. η-expansions are used to express the fact that all terms and coterms may be seen as abstractions. Finally, the utility of ς-rules will be seen clearly later, when we introduce CBV and CBN reduction relations (e.g. see comment following Proposition 1.8).

Note that for a reduction relation R denoted by $\longrightarrow$, we will denote:
- its reflexive closure by $\longrightarrow_=$,
- its transitive closure by $\longrightarrow_+$,
- its reflexive transitive closure by $\twoheadrightarrow$.

Now, due to duality being present in its reduction rules, R_b is not confluent. For example, $(x \bullet \alpha).\beta \bullet y.(z \bullet \gamma)$ reduces both to $x \bullet \alpha$ and to $z \bullet \gamma$. Confluent reduction relations can be obtained by placing restrictions on R_b. In particular, restrictions placed on R_b can lead to confluent CBV and CBN reduction relations.

Due to the computational interpretation of terms and coterms, values are a subset of terms. Dually, one defines covalues as coterms of special structure. [4]

Definition 1.3 (*Values and covalues*).

$$\begin{array}{lll}
\text{Value} & V, W & ::= x \mid \langle V, W\rangle \mid \langle V\rangle\mathtt{inl} \mid \langle W\rangle\mathtt{inr} \mid [K]\mathtt{not} \\
\text{Covalue} & P, Q & ::= \alpha \mid [P, Q] \mid \mathtt{fst}[P] \mid \mathtt{snd}[Q] \mid \mathtt{not}\langle M\rangle
\end{array}$$

Thus, *variable abstractions are not values*, and similarly for covariable abstractions. At this point, the reader is advised to keep in mind the letters used to denote values and covalues (as well as other syntactic classes), since these are standard in this paper.

Evidently, covalues are in a way the duals of values: covalues are to CBN what values are to CBV.

[3] Note that we assume a tacit *variable convention*, in the style of [2], so that such substitutions do not bound free variables inside M.

[4] The intuition behind these definitions can be found in [20].

Definition 1.4 (*CBV reduction R_v and CBN reduction R_n, Walder [20]*). The CBV reduction relation R_v [resp. CBN reduction relation R_n] is the one-step reduction relation yielded by the CBV [CBN] reduction rules listed below, when these are applied to subobjects of DuCa objects. For $G, H \in \texttt{DuCa}$, $(G, H) \in R_v$ [$(G, H) \in R_n$] is usually written as $G \longrightarrow^v H$ [$G \longrightarrow^n H$].

	CBV rules			CBN rules		
$(\beta\&_1)$	$\langle V, W\rangle \bullet \texttt{fst}[K]$	$\to$	$V \bullet K$	$\langle M, N\rangle \bullet \texttt{fst}[P]$	$\to$	$M \bullet P$
$(\beta\&_2)$	$\langle V, W\rangle \bullet \texttt{snd}[L]$	$\to$	$W \bullet L$	$\langle M, N\rangle \bullet \texttt{snd}[Q]$	$\to$	$N \bullet Q$
$(\beta\vee_1)$	$\langle V\rangle\texttt{inl} \bullet [K, L]$	$\to$	$V \bullet K$	$\langle M\rangle\texttt{inl} \bullet [P, Q]$	$\to$	$M \bullet P$
$(\beta\vee_2)$	$\langle W\rangle\texttt{inr} \bullet [K, L]$	$\to$	$W \bullet L$	$\langle N\rangle\texttt{inr} \bullet [P, Q]$	$\to$	$N \bullet Q$
$(\beta\neg)$	$[K]\texttt{not} \bullet \texttt{not}\langle M\rangle$	$\to$	$M \bullet K$	$[K]\texttt{not} \bullet \texttt{not}\langle M\rangle$	$\to$	$M \bullet K$
(βL)	$V \bullet x.(S)$	$\to$	$S\{V/x\}$	$M \bullet x.(S)$	$\to$	$S\{M/x\}$
(βR)	$(S).\alpha \bullet K$	$\to$	$S\{K/\alpha\}$	$(S).\alpha \bullet P$	$\to$	$S\{P/\alpha\}$
(ηL)	K	$\to$	$x.(x \bullet K)$	K	$\to$	$x.(x \bullet K)$
(ηR)	M	$\to$	$(M \bullet \alpha).\alpha$	M	$\to$	$(M \bullet \alpha).\alpha$
$(\varsigma\&_1)$	$\langle M, N\rangle \bullet K$	$\to$	$M \bullet x.(\langle x, N\rangle \bullet K)$	$M \bullet \texttt{fst}[K]$	$\to$	$(M \bullet \texttt{fst}[\alpha]).\alpha \bullet K$
$(\varsigma\&_2)$	$\langle V, N\rangle \bullet K$	$\to$	$N \bullet y.(\langle V, y\rangle \bullet K)$	$M \bullet \texttt{snd}[L]$	$\to$	$(M \bullet \texttt{snd}[\beta]).\beta \bullet L$
$(\varsigma\vee_1)$	$\langle M\rangle\texttt{inl} \bullet K$	$\to$	$M \bullet x.(\langle x\rangle\texttt{inl} \bullet K)$	$M \bullet [K, L]$	$\to$	$(M \bullet [\alpha, L]).\alpha \bullet K$
$(\varsigma\vee_2)$	$\langle N\rangle\texttt{inr} \bullet K$	$\to$	$N \bullet y.(\langle y\rangle\texttt{inr} \bullet K)$	$M \bullet [P, L]$	$\to$	$(M \bullet [P, \beta]).\beta \bullet L$

One clearly notes the duality in the definitions of R_v and R_n, which proposes that sums are treated as "duals" of products. This duality is manifested in [20], a paper titled "CBV is Dual to CBN".

Another property to be discussed further ahead is this of confluence, or Church–Rosser property (CR). Though it is clear that the case of the critical pair $(x \bullet \alpha).\beta \bullet y.(z \bullet \gamma)$ is now resolved in CBV or CBN, it is not clear that CR holds for these reduction relations. This is studied in Section 2, and it is shown that it holds indeed.

1.3. Relation to other calculi

We briefly examine how DuCa is related to some standard computational calculi. In [20] Wadler introduces CPS translations taking objects of DuCa to a restriction of the simply-typed lambda calculus with sums and products. Inverse CPS translations are also introduced, and it is shown that the CPS translation is a *reflection*. In [21] translations to and from the $\lambda\mu$-calculus are introduced, and it is shown that these form an *equational correspondence*.

Below we introduce a simple β-reduction-preserving embedding of the simply-typed lambda calculus in the DuCa, which clarifies the fact that DuCa is an extension of the lambda calculus. Afterwards, we examine a possible translation from DuCa to the Symmetric λ-calculus and vice versa.

1.3.1. Embedding the lambda calculus

The three forms of reduction relations in DuCa, namely basic, CBV and CBN reduction, are also present in the simply-typed lambda calculus. Here, we will use the CBV versions of both calculi. Moreover, we will use, for space economy, the abbreviations of lambda abstraction and application for DuCa defined in (1). Note that under these abbreviations we have the following simulation of common β-reduction:

$$\lambda x.M \bullet V @ K \overset{\beta}{\twoheadrightarrow}{}^{v} V \bullet x.(M \bullet K).$$

Now, the definition of the simply-typed lambda calculus is standard [2].

Definition 1.5. The simply-typed lambda calculus consists of Types and Terms:

$$\begin{array}{lll} \text{Type } A, B & ::= & X \mid A \supset B \\ \text{Term } M, N & ::= & V \mid MN \\ \text{Value } V & ::= & x \mid \lambda x.M \end{array}$$

The set of terms is denoted by Λ. The typing rules for this calculus are:

$$\frac{}{x : A, \Gamma \vdash x : A}\,Ax \qquad \frac{\Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}\,App \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \supset B}\,Abs$$

where Γ is some set of assumptions $x_1 : B_1, \dots, x_n : B_n$.

The CBV reduction relation R_v^λ is yielded by the reduction rule

$$(\lambda x.M)V \to M\{V/x\}$$

being applied to subterms of terms. For $M, N \in \Lambda$, we usually write $M \longrightarrow^v N$ instead of $(M, N) \in R_v^\lambda$.

The translation of the simply-typed lambda calculus is defined below. Note that all elements (terms) of the source calculus are translated to terms of DuCa.

Definition 1.6. We define the following translation from simply-typed lambda calculus to DuCa:

$$\begin{array}{lll} (A)^D & \equiv & A \\ (x)^D & \equiv & x \\ (\lambda x.M)^D & \equiv & \lambda x.(M)^d \\ (V)^d & \equiv & ((V)^D \bullet \alpha).\alpha \\ (MN)^d & \equiv & ((M)^d \bullet (N)^d @ \alpha).\alpha \end{array}$$

Note that the α's in the last two lines above are fresh.

The reason for using this two-step definition for $(M)^d$ is mainly that thus, for every value V, $(V)^D$ is a value in DuCa. We can prove the following proposition.

Proposition 1.7.

1. *For any* $x, M, V, \gamma, L \in$ DuCa: $\lambda x.M \bullet (V \bullet \gamma).\gamma @ L \overset{\beta v}{\twoheadrightarrow}{}^v V \bullet x.(M \bullet L)$.
2. *For any* $M, V, x \in \Lambda$: $(M)^d\{(V)^D/x\} \equiv (M\{V/x\})^d$.
3. *For any* $M \in \Lambda$ *and* $\alpha \in$ DuCa: $((M)^d \bullet \alpha).\alpha \overset{\beta R}{\longrightarrow}{}^v (M)^d$.

Proof. The first claim is straightforward; the other two are proven by induction. □

Thus, we can prove that the translation defined produces the desired embedding.

Proposition 1.8 (*Embedding of CBV simply-typed lambda calculus*).
1. *If* $M \in \Lambda$ *and* $\Gamma \vdash M : A$ *is derivable, then* $(\Gamma)^D \rightarrow \mid (M)^d : (A)^D$ *is derivable.*
2. *For any* $x, M, V \in \Lambda$: $\quad ((\lambda x.M)V)^d \overset{\beta v}{\twoheadrightarrow}{}^v (M\{V/x\})^d$.

Proof. 1 is proven by induction on the derivation of the former sequent in the simply-typed lambda calculus. For 2, we have:

$$\begin{aligned}((\lambda x.M)V)^d &\equiv ((\lambda x.M)^d \bullet (V)^d @\alpha).\alpha \equiv ((\lambda x.(M)^d \bullet \beta).\beta \bullet ((V)^D \bullet \gamma).\gamma @\alpha).\alpha \\ &\xrightarrow{\beta R}{}^{v} (\lambda x.(M)^d \bullet ((V)^D \bullet \gamma).\gamma @\alpha).\alpha \\ &\underset{\text{prop.1.7(1)}}{\overset{\beta v}{\twoheadrightarrow}}{}^{v} ((V)^D \bullet x.((M)^d \bullet \alpha)).\alpha \xrightarrow{\beta L}{}^{v} ((M)^d\{(V)^D/x\} \bullet \alpha).\alpha \\ &\underset{\text{prop.1.7(2)}}{\equiv} ((M\{V/x\})^d \bullet \alpha).\alpha \underset{\text{prop.1.7(3)}}{\xrightarrow{\beta R}}{}^{v} (M\{V/x\})^d \qquad \square\end{aligned}$$

It is important to note that, under the defined translation, we need to use v-reductions in DuCa in order to preserve β-reductions of the simply-typed lambda calculus. Indeed, v-reductions are needed in Proposition 1.7, and, in particular, the $v\&_1$ rule is used (in the omitted proof). This fact gives us a hint on the role of v-rules in DuCa: they are rather complementary to β-rules, facilitating some β-reductions otherwise forbidden by CBV (or CBN) restrictions, than entirely novel rules.

1.3.2. Translating to and from the Symmetric Lambda Calculus

The Sλ-calculus of Barbanera and Berardi [1] is similar to DuCa in that it is a "classical" λ-calculus with symmetric abstractions. In fact, the two calculi are even more similar: below we will define typing- and reduction-preserving translations between them.

Definition 1.9. The terms of the Sλ-calculus are defined by

$$\text{Terms} \quad u, v, w ::= x \mid \mu x.u \mid u \bullet v \mid \langle u, v\rangle \mid \sigma_1(u) \mid \sigma_2(u).$$

The types are given by

$$\begin{aligned}\text{Type} \quad & D ::= \bot \mid A \\ \text{m-Type} \quad & A, B, C ::= X \mid X^\bot \mid A\&B \mid A \vee B\end{aligned}$$

For each m-type A we define $A^\bot$ by the usual De Morgan rules pushing negations inside propositional connectives, and using double-negation elimination for type variables.

The typing rules of the Sλ-calculus are listed below. These involve sequents of the form $\Gamma \vdash u : A$, where Γ is some context set $\{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\}$.

$$\frac{}{\Gamma, x : A \vdash x : A}\,\texttt{id}$$

$$\frac{\Gamma \vdash u : A \quad \Gamma \vdash v : B}{\Gamma \vdash \langle u, v\rangle : A\&B}\,\&I \qquad \frac{\Gamma \vdash u_i : A_i}{\Gamma \vdash \sigma_i(u_i) : A_1 \vee A_2}\,\vee I \;\; (i = 1, 2)$$

$$\frac{\Gamma, x : A \vdash u : \bot}{\Gamma \vdash \mu x.u : A^\bot}\,\neg I \qquad \frac{\Gamma \vdash u : A^\bot \quad \Gamma \vdash v : A}{\Gamma \vdash u \bullet v : \bot}\,\neg E$$

The reduction relation R is the one-step reduction relation yielded by the reduction rules listed below, when these are applied on subterms of terms:

$$\begin{array}{lll}(\beta L) & \mu x.u \bullet v & \to u\{v/x\} \\ (\beta R) & u \bullet \mu x.v & \to v\{u/x\} \\ (\beta\&_i) & \langle u_1, u_2\rangle \bullet \sigma_i v & \to u_i \bullet v \\ (\beta\vee_i) & \sigma_i u \bullet \langle v_1, v_2\rangle & \to u \bullet v_i \\ (\eta L) & \mu x.(u \bullet x) & \to u \\ (\eta R) & \mu x.(x \bullet u) & \to u\end{array}$$

with x taken fresh in the η rules, and $i = 1, 2$. For $(u, v) \in R$ we simply write $u \longrightarrow v$.

Below we define a translation of DuCa into the Sλ-calculus.

Definition 1.10. The translation $(\_)^o$ from objects of DuCa to terms of the Sλ-calculus is given by induction on the syntax, given some fixed interpretation from covariables (and variables) of DuCa to variables in the Sλ-calculus. It is accompanied by a types translation.

$$(X)^o \equiv X \qquad (\neg A)^o \equiv ((A)^o)^\perp \qquad (A\&B)^o \equiv (A)^o\&(B)^o \qquad (A \vee B)^o \equiv (A)^o \vee (B)^o$$

$$\begin{aligned}
(x)^o &\equiv x & (\alpha)^o &\equiv x_\alpha \\
(\langle M, N\rangle)^o &\equiv \langle (M)^o, (N)^o\rangle & ([K, L])^o &\equiv \langle (K)^o, (L)^o\rangle \\
(\langle M\rangle\mathtt{inl})^o &\equiv \sigma_1((M)^o) & (\mathtt{fst}[K])^o &\equiv \sigma_1((K)^o) \\
(\langle N\rangle\mathtt{inr})^o &\equiv \sigma_2((N)^o) & (\mathtt{snd}[L])^o &\equiv \sigma_2((L)^o) \\
([K]\mathtt{not})^o &\equiv \mu x.x \bullet (K)^o & (\mathtt{not}\langle M\rangle)^o &\equiv (M)^o \\
((S).\alpha)^o &\equiv \mu x_\alpha.(S)^o & (x.(S))^o &\equiv \mu x.(S)^o \\
(M \bullet K)^o &\equiv (M)^o \bullet (K)^o
\end{aligned}$$

Finally, let $(M : A)^o \equiv (M)^o : (A)^o$ and $(K : A)^o \equiv (K)^o : ((A)^o)^\perp$.

The translation defined above exploits duality of the calculus by identifying dual constructions for sum and product, and ignoring constructions for negation. One may notice a redundancy in translating $[K]\mathtt{not}$ to $\mu x.x \bullet (K)^o$, instead of $(K)^o$. We include this redundancy on purpose, in order to strongly preserve the $\beta\neg$ reductions of DuCa.

It is not difficult to show that typing is preserved.

Proposition 1.11. *Typing is preserved under the above defined translation*:

$$\begin{aligned}
\Gamma \rightarrow \Theta \mid M : A \quad &\Longrightarrow \quad (\Gamma)^o, (\Theta)^o \vdash (M : A)^o \\
K : A \mid \Gamma \rightarrow \Theta \quad &\Longrightarrow \quad (\Gamma)^o, (\Theta)^o \vdash (K : A)^o \\
\Gamma \mid S \mapsto \Theta \quad &\Longrightarrow \quad (\Gamma)^o, (\Theta)^o \vdash (S)^o : \perp
\end{aligned}$$

Proof. By induction on DuCa derivations. □

Now, consider the reduction relation $R_b^{\beta\eta'}$ yielded by restricting R_b to β- and η-rules, the latter turned to η-contractions instead of η-expansions.

Definition 1.12. $R_b^{\beta\eta'}$ is the reduction relation yielded by the same β-rules as R_b (Definition 1.2), plus the altered η-rules:

$$(\eta L')\ \ x.(x \bullet K) \rightarrow K, \quad (\eta R')\ \ (M \bullet \alpha).\alpha \rightarrow M.$$

For $G, H \in$ DuCa, $(G, H) \in R_b^{\beta\eta'}$ may be written as $G \xrightarrow{\beta\eta'}{}^{b} H$.

Then $R_b^{\beta\eta'}$ reductions are preserved by our translation.

Proposition 1.13. *For $G, H \in$ DuCa, if $G \xrightarrow{\beta\eta'}{}^{b} H$ then $(G)^o \longrightarrow (H)^o$.*

Proof. First we show by induction that for any $G, M, K, x, \alpha \in$ DuCa,

$$(G\{M/x\})^o \equiv (G)^o\{(M)^o/x\} \quad \text{and} \quad (G\{K/\alpha\})^o \equiv (G)^o\{(K)^o/x_\alpha\},$$

whence the statement straightforwardly follows. □

To devise a translation from the Sλ-calculus to DuCa is a more complicated task. This is mainly because of the identity $A \equiv A^{\perp\perp}$ on m-types of the former calculus not being preserved in the latter. Therefore, amongst other specifications, we have to “encode” somehow the double-negation rule inside our translation. Moreover, we need some extra η-reduction rules to be added to $R_b^{\beta\eta'}$ in order to preserve reductions.

Definition 1.14. $R_b^{\beta\eta+}$ is the reduction relation yielded by the same rules as $R_b^{\beta\eta'}$ plus the η-rules:

$$\begin{array}{lrcl} (\eta\neg) & \mathtt{not}\langle([\beta]\mathtt{not}\bullet\alpha).\beta\rangle & \to & \alpha \\ (\eta\&) & \langle(x\bullet\mathtt{fst}[\alpha]).\alpha,(x\bullet\mathtt{snd}[\beta]).\beta\rangle & \to & x \\ (\eta\vee) & [x.(\langle x\rangle\mathtt{inl}\bullet\alpha),y.(\langle y\rangle\mathtt{inr}\bullet\alpha)] & \to & \alpha \end{array}$$

For $G, H \in \mathtt{DuCa}$, $(G,H)\in R_b^{\beta\eta+}$ may be written as $G \xrightarrow{\beta\eta+}{}_b H$.

Below we define the translation from the Sλ-calculus to DuCa. Since DuCa is not symmetric we necessarily need type-information for Sλ-terms. Therefore, the translation is from typed Sλ-terms to typed DuCa objects, although in the definition we indicate types explicitly only when necessary.

Definition 1.15. The translation $(\_)^p$ from typed terms of the Sλ-calculus to typed objects of DuCa is defined by induction, given some fixed such translation for variables.

$$\begin{array}{llll} (X)^p & \equiv X & (X^\perp)^p & \equiv \neg X \\ (A_1\&A_2)^p & \equiv (A_1)^p\&(A_2)^p & (A_1\vee A_2)^p & \equiv (A_1)^p\vee(A_2)^p \end{array}$$

$$\begin{array}{llll} (x)^p & \equiv x & (\langle u,v\rangle)^p & \equiv \langle(u)^p,(v)^p\rangle \\ (\sigma_1(u))^p & \equiv \langle(u)^p\rangle\mathtt{inl} & (\sigma_2(v))^p & \equiv \langle(v)^p\rangle\mathtt{inr} \\ (u^A\bullet v^{A^\perp})^p & \equiv (u)^p\bullet((v)^p)^d & \multicolumn{2}{l}{\text{when } A\equiv X \text{ or } A\equiv A_1\&A_2} \\ (u^A\bullet v^{A^\perp})^p & \equiv (v)^p\bullet((u)^p)^d & \multicolumn{2}{l}{\text{when } A\equiv X^\perp \text{ or } A\equiv A_1\vee A_2} \\ (\mu x.u)^p & \equiv (x.(u)^p)^d & & \end{array}$$

$(\_)^d$ is a partial internal translation in DuCa from typed terms to typed coterms, and vice versa, which demorganly negates types. $(\_)^d$ is defined for terms typed with types

$$A,B ::= X \mid \neg X \mid A\vee B \mid A\&B \tag{2}$$

i.e. images of m-types under $(\_)^p$. On variables and covariables it is defined by induction on types; on other terms and coterms by use of $(\_)^d$ on variables and covariables.

$$\begin{array}{ll} (X)^d\equiv\neg X & (A_1\&A_2)^d\equiv(A_1)^d\vee(A_2)^d \\ (\neg X)^d\equiv X & (A_1\vee A_2)^d\equiv(A_1)^d\&(A_2)^d \end{array}$$

$$\begin{array}{lll} (x:X)^d & \equiv \mathtt{not}\langle x\rangle:\neg X & \\ (x:A_1\&A_2)^d & \equiv [x_1.(x\bullet\mathtt{fst}[K_1]),x_2.(x\bullet\mathtt{snd}[K_2])]:(A_1)^d\vee(A_2)^d & \text{where } K_i:A_i\equiv(x_i:(A_i)^d)^d \\ (\alpha:\neg X)^d & \equiv ([\beta]\mathtt{not}\bullet\alpha).\beta:X & \\ (\alpha:A_1\vee A_2)^d & \equiv \langle(\langle M_1\rangle\mathtt{inl}\bullet\alpha).\alpha_1,(\langle M_2\rangle\mathtt{inr}\bullet\alpha).\alpha_2\rangle:(A_1)^d\&(A_2)^d & \text{where } M_i:A_i\equiv(\alpha_i:(A_i)^d)^d \\ (M:A)^d & \equiv x'.(M\bullet K') \quad \text{where } K':A\equiv(x':(A)^d)^d & \\ (K:A)^d & \equiv (M'\bullet K).\alpha' \quad \text{where } M':A\equiv(\alpha':(A)^d)^d & \end{array}$$

Note that the cases for $\alpha : X$ and $\alpha : A_1\&A_2$ are given by the last line above, and similarly for $x : \neg X$ and $x : A_1 \vee A_2$.

In the above definition we explicitly use the fact that for any type A defined by (2) we have $((A)^d)^d \equiv A$. The internal DuCa translation has several useful properties.

Lemma 1.16. *Let $x, y, M, N, \alpha, K, L, \Gamma, \Theta \in$ DuCa, A be a type as in* (2) *and B be an m-type*:
1. *$(B)^p$ belongs to the types in* (2) *and* $(B^\perp)^p \equiv ((B)^p)^d$,
2. $\Gamma \twoheadrightarrow \Theta \mid M : A \implies (M:A)^d \mid \Gamma \twoheadrightarrow \Theta, K : A \mid \Gamma \twoheadrightarrow \Theta \implies \Gamma \twoheadrightarrow \Theta \mid (K:A)^d$,

3. $(A \equiv \neg X \vee A \equiv A_1 \vee A_2) \implies (M : A)^d \equiv (x : A)^d\{M/x\}, (M : A)^d\{N/y\} \equiv (M\{N/y\})^d$,
4. $((x : A)^d \equiv K : (A)^d \wedge (\alpha : (A)^d)^d \equiv M : A) \implies (K\{M/x\} \overset{\beta\eta+}{\twoheadrightarrow}{}^b \alpha \wedge M\{K/x\} \overset{\beta\eta+}{\twoheadrightarrow}{}^b x)$,
5. $((M : A)^d)^d \equiv N : A \implies N \overset{\beta\eta+}{\twoheadrightarrow}{}^b M, ((K : A)^d)^d \equiv L : A \implies L \overset{\beta\eta+}{\twoheadrightarrow}{}^b K$.

Proof. Item 1 is straightforward. 2 is shown by induction on the measures $(|M|, |A|_{\&\vee}, \kappa_M)$ and $(|K|, |A|_{\&\vee}, \kappa_K)$, where $|A|_{\&\vee}$ is the $(\&\vee)$-complexity of type A and $\kappa_M = 0$ if $M : A$ is $x : X$ or $x : A_1\&A_2$, otherwise $\kappa_M = 1$ (symmetrically for κ_K).

The first statement of 3 is straightforward by definition. The second statement follows:

$$(M : A)^d\{N/y\} \equiv (x : A)^d\{M/x\}\{N/y\} \equiv (x : A)^d\{M\{N/y\}/x\} \equiv (M\{N/y\})^d$$

4 is shown by induction on (the $(\&\vee)$-complexity of) A using the added η-reduction rules.
5 is shown by a long computation using (among others) 4. $\square$

Now, we can show that typing is preserved.

Proposition 1.17. *Typing is preserved under the above defined translation*:

$$\begin{array}{lll} \Gamma \vdash u : A & \implies & (\Gamma)^p \rightarrow \mid (u : A)^p \\ \Gamma \vdash u : \bot & \implies & (\Gamma)^p \mid (u)^p \mapsto \end{array}$$

Proof. By straightforward induction on Sλ-derivations using the first two items of the previous lemma. $\square$

Reductions are also preserved (note that $\overset{\beta\eta^+}{\longrightarrow}{}^b_+$ is the transitive closure of $\overset{\beta\eta^+}{\longrightarrow}{}^b$).

Proposition 1.18. *Let u^D, v^D be typed Sλ-terms, if $u \longrightarrow v$ then $(u^D)^p \overset{\beta\eta^+}{\longrightarrow}{}^b_+ (v^D)^p$.*

Proof. First we show that, for any Sλ-terms u and v, $(u)^p\{(v)^p/x\} \equiv (u\{v/x\})^p$. We proceed by induction on u; the interesting case is of u being a cut, $u \equiv u_1^A \bullet u_2^{A^\perp}$ with $A \equiv X$ or $A \equiv A_1\&A_2$ (the case of $A \equiv X^\perp$ or $A \equiv A_1 \vee A_2$ is treated symmetrically):

$$\begin{aligned} (u)^p\{(v)^p/x\} &\equiv (u_1^A \bullet u_2^{A^\perp})^p\{(v)^p/x\} \equiv ((u_1)^p \bullet ((u_2)^p)^d)\{(v)^p/x\} \\ &\equiv (u_1)^p\{(v)^p/x\} \bullet ((u_2)^p)^d\{(v)^p/x\} \overset{\text{lem 1.16(3)}}{\equiv} (u_1)^p\{(v)^p/x\} \bullet ((u_2)^p\{(v)^p/x\})^d \\ &\overset{IH}{\equiv} (u_1\{v/x\})^p \bullet ((u_2\{v/x\})^p)^d \equiv (u_1\{v/x\} \bullet u_2\{v/x\})^p \equiv (u\{v/x\})^p \end{aligned}$$

Next, it suffices to show that reduction rules are preserved. Note that below we use plain arrows for $\beta\eta^+$ reductions. The cases of rules βL and βR follow from the above result on substitution. Regarding $\beta\&_1$, we have that, if $A \equiv A_1\&A_2$ and $(x' : (A)^p)^d \equiv K' : ((A)^p)^d$ then,

$$\begin{aligned} (\langle u_1, u_2\rangle^A \bullet \sigma_1(v)^{A^\perp})^p &\equiv \langle (u_1)^p, (u_2)^p\rangle \bullet x'.(\langle (v)^p\rangle \mathtt{inl} \bullet K') \\ &\overset{K'\equiv\ldots}{\longrightarrow} \langle (u_1)^p, (u_2)^p\rangle \bullet x'.((v)^p \bullet x_1.(x' \bullet \mathtt{fst}[K_1])) \twoheadrightarrow (v)^p \bullet x_1.((u_1)^p \bullet K_1) \end{aligned}$$

where $(x_1 : ((A_1)^p)^d)^d \equiv K_1 : (A_1)^p$.

If $A_1 \equiv X^\perp$ or $A_1 \equiv B_1 \vee B_2$ then $(v)^p \bullet x_1.((u_1)^p \bullet K_1) \equiv (u_1 \bullet v)^p$, since $u_1 : A_1$. Otherwise, if $A_1 \equiv X$ or $A_1 \equiv B_1\&B_2$ then $(v)^p \bullet x_1.((u_1)^p \bullet K_1) \longrightarrow (u_1)^p \bullet K_1\{(v)^p/x_1\} \equiv (u_1 \bullet v)^p$.

The case of $\beta\vee_1$ is treated similarly.

Regarding ηL, $(\mu x.(u^A \bullet x^{A^\perp}))^p \equiv (x.(u^A \bullet x^{A^\perp})^p)^d \equiv (M' \bullet x.(u^A \bullet x^{A^\perp})^p).\alpha'$, with $(\alpha' : (A)^p)^d \equiv M' : ((A)^p)^d$. Now, if $A \equiv X$ or $A \equiv B_1\&B_2$ then,

$$(\mu x.(u^A \bullet x^{A^\perp}))^p \equiv (M' \bullet x.((u)^p \bullet (x)^d)).\alpha' \longrightarrow ((u)^p \bullet (x)^d\{M'/x\}).\alpha' \overset{\text{lem 1.16(4)}}{\twoheadrightarrow} ((u)^p \bullet \alpha').\alpha' \longrightarrow (u)^p$$

Otherwise, if $A \equiv \neg X$ or $A \equiv B_1 \vee B_2$ then, taking $(x' : ((A)^p)^d)^d \equiv K' : (A)^p$,

$$(\mu x.(u^A \bullet x^{A^\perp}))^p \equiv (M' \bullet x.(x \bullet ((u)^p)^d)).\alpha' \longrightarrow (M' \bullet ((u)^p)^d).\alpha' \equiv (M' \bullet x'.((u)^p \bullet K')).\alpha'$$
$$\longrightarrow ((u)^p \bullet K'\{M'/x'\})).\alpha' \overset{\text{lem 1.16(4)}}{\twoheadrightarrow} ((u)^p \bullet \alpha').\alpha' \longrightarrow (u)^p$$

The case of ηR is similar. □

Finally, we note that the defined translations between types in DuCa and the Sλ-calculus form a projection-embedding pair. That is, for any m-type A, $((A)^p)^o \equiv A$ holds,[5] while the reverse direction, from DuCa into Sλ into DuCa, does not have this property: for $A \equiv \neg(X \& Y)$ we have $((A)^o)^p \equiv (\neg X \vee \neg Y) \not\equiv A$. The key point is that in DuCa we do not have an involutive negation.

2. Syntactic investigations

In this section we investigate some syntactic properties of the DuCa under CBV reduction. Because of duality with CBN, all results proven here have analogs for the CBN case.

First, we examine the untyped DuCa and prove confluence, or Church–Rosser property, under CBV. The proof follows, in rough lines, the proof of $\beta\eta$-reduction being CR in the lambda calculus, as it is presented in [2].

Another important syntactic property to examine is Strong Normalization (SN). For this, we investigate the typed version of DuCa under CBV reduction, with certain restrictions applied on reduction rules to avoid reduction loops. The proof of SN involves using translations of DuCa to other calculi, known to be SN, notably the CBV CPS translation of [20] and a translation to a simplified version of DuCa (called DuCa*), which we show to be SN.

The section ends with a consideration of a restricted version of CBV reduction which satisfies both the CR and SN properties.

2.1. Investigation of the Church–Rosser property

In this section we are interested in the untyped version of DuCa under CBV reduction. The untyped DuCa consists solely of Objects. The set of objects is the union of the set of Terms, Coterms and Statements:

Object	G, H	$::=$	$M \mid K \mid S$
Term	M, N	$::=$	$x \mid \langle M, N\rangle \mid \langle M\rangle\mathtt{inl} \mid \langle N\rangle\mathtt{inr} \mid [K]\mathtt{not} \mid (S).\alpha$
Coterm	K, L	$::=$	$\alpha \mid [K, L] \mid \mathtt{fst}[K] \mid \mathtt{snd}[L] \mid \mathtt{not}\langle M\rangle \mid x.(S)$
Statement	S, T	$::=$	$M \bullet K$
Value	V, W	$::=$	$x \mid \langle V, W\rangle \mid \langle V\rangle\mathtt{inl} \mid \langle W\rangle\mathtt{inr} \mid [K]\mathtt{not}.$

Recall also that the CBV reduction relation R_v is the one-step reduction relation yielded by the CBV reduction rules listed below, when these are applied to subobjects of DuCa objects (for $G, H \in$ DuCa, $(G, H) \in R_v$ is denoted by $G \longrightarrow^v H$).

$(\beta\&_1)$	$\langle V, W\rangle \bullet \mathtt{fst}[K]$	$\to$	$V \bullet K$	(ηL)	K	$\to$	$x.(x \bullet K),$
$(\beta\&_2)$	$\langle V, W\rangle \bullet \mathtt{snd}[L]$	$\to$	$W \bullet L$	(ηR)	M	$\to$	$(M \bullet \alpha).\alpha,$
$(\beta\vee_1)$	$\langle V\rangle\mathtt{inl} \bullet [K, L]$	$\to$	$V \bullet K$				
$(\beta\vee_2)$	$\langle W\rangle\mathtt{inr} \bullet [K, L]$	$\to$	$W \bullet L$	$(\nu\&_1)$	$\langle M, N\rangle \bullet K$	$\to$	$M \bullet x.(\langle x, N\rangle \bullet K),$
$(\beta\neg)$	$[K]\mathtt{not} \bullet \mathtt{not}\langle M\rangle$	$\to$	$M \bullet K$	$(\nu\&_2)$	$\langle V, N\rangle \bullet K$	$\to$	$N \bullet y.(\langle V, y\rangle \bullet K),$
(βL)	$V \bullet x.(S)$	$\to$	$S\{V/x\}$	$(\nu\vee_1)$	$\langle M\rangle\mathtt{inl} \bullet K$	$\to$	$M \bullet x.(\langle x\rangle\mathtt{inl} \bullet K),$
(βR)	$(S).\alpha \bullet K$	$\to$	$S\{K/\alpha\}$	$(\nu\vee_2)$	$\langle N\rangle\mathtt{inr} \bullet K$	$\to$	$N \bullet y.(\langle y\rangle\mathtt{inr} \bullet K).$

For the rest of this section, the restriction of R_v to $\beta\nu$-rules (i.e. β-rules and ν-rules) will be denoted by $\xrightarrow{\beta\nu}$, and called simply $\beta\nu$-reduction relation. Analogously, the restriction to η-rules will be denoted by $\xrightarrow{\eta}$, and called η-reduction relation.

[5] Note that the same, i.e. $((u : A)^p)^o \equiv u : A$, does not hold for typed Sλ-terms. In order to get the weaker $((u : A)^p)^o \longrightarrow u : A$ we would have to add η-reduction rules for & and $\vee$ in Sλ.

Let us recall the definition of the Church–Rosser property.

Definition 2.1 (*Church–Rosser properties*). Let $R \subset U^2$ be some reduction relation denoted by $\longrightarrow$, for some universe set U. Then,

- R satisfies the Diamond Property if, for all $M, N, K \in U$, if $N \longleftarrow M \longrightarrow K$, then there is some $L \in U$ such that $N \longrightarrow L \longleftarrow K$.
- R satisfies the Church–Rosser property, or is CR, if its transitive reflexive closure ($\twoheadrightarrow$) satisfies the diamond property.
- R satisfies the Weak Church–Rosser property (WCR) if, for all $M, N, K \in U$, if $N \longleftarrow M \longrightarrow K$, then there is some $L \in U$ such that $N \twoheadrightarrow L \twoheadleftarrow K$.

The purpose of this section is to show that R_v in the dual calculus is Church–Rosser. We follow the steps below:

- We show that the βv-reduction relation is CR (Lemma 2.6).
- We show that the η-reduction relation is CR (Lemma 2.7).
- We show that $\overset{\beta v}{\twoheadrightarrow}$ and $\overset{\eta}{\twoheadrightarrow}$ commute (Lemma 2.11).

For the first step, we define a parallel reduction relation $\longrightarrow_{\mathrm{p}}$, such that $\overset{\beta v}{\longrightarrow}_{=} \subseteq \longrightarrow_{\mathrm{p}} \subseteq \overset{\beta v}{\twoheadrightarrow}$.

Definition 2.2. $\longrightarrow_{\mathrm{p}}$ is defined as follows. Let $G, M, N, K, L, V, W, S \in$ DuCa; if $M \longrightarrow_{\mathrm{p}} M'$, $V \longrightarrow_{\mathrm{p}} M''$, $N \longrightarrow_{\mathrm{p}} N'$, $W \longrightarrow_{\mathrm{p}} N''$, $K \longrightarrow_{\mathrm{p}} K'$, $L \longrightarrow_{\mathrm{p}} L'$, $S \longrightarrow_{\mathrm{p}} S'$ then,

(pid)	$G \longrightarrow_{\mathrm{p}} G$		
(pβ&$_1$)	$\langle V, W\rangle \bullet \mathtt{fst}[K] \longrightarrow_{\mathrm{p}} M'' \bullet K'$	(p$\bullet$)	$M \bullet K \longrightarrow_{\mathrm{p}} M' \bullet K'$,
(pβ&$_2$)	$\langle V, W\rangle \bullet \mathtt{snd}[L] \longrightarrow_{\mathrm{p}} N'' \bullet L'$	(p$\langle,\rangle$)	$\langle M, N\rangle \longrightarrow_{\mathrm{p}} \langle M', N'\rangle$,
(p$\beta\vee_1$)	$\langle V\rangle\mathtt{inl} \bullet [K, L] \longrightarrow_{\mathrm{p}} M'' \bullet K'$	(p$\langle\rangle$inl)	$\langle M\rangle\mathtt{inl} \longrightarrow_{\mathrm{p}} \langle M'\rangle\mathtt{inl}$,
(p$\beta\vee_2$)	$\langle W\rangle\mathtt{inr} \bullet [K, L] \longrightarrow_{\mathrm{p}} N'' \bullet L'$	(p$\langle\rangle$inr)	$\langle M\rangle\mathtt{inl} \longrightarrow_{\mathrm{p}} \langle N'\rangle\mathtt{inl}$,
(p$\beta\neg$)	$[K]\mathtt{not} \bullet \mathtt{not}\langle M\rangle \longrightarrow_{\mathrm{p}} M' \bullet K'$	(p[]not)	$[K]\mathtt{not} \longrightarrow_{\mathrm{p}} [K']\mathtt{not}$,
(pβL)	$V \bullet x.(S) \longrightarrow_{\mathrm{p}} S'\{M''/x\}$	(p().)	$(S).\alpha \longrightarrow_{\mathrm{p}} (S').\alpha$,
(pβR)	$(S).\alpha \bullet K \longrightarrow_{\mathrm{p}} S'\{K'/\alpha\}$	(p[,])	$[K, L] \longrightarrow_{\mathrm{p}} [K', L']$,
(pv&$_1$)	$\langle M, N\rangle \bullet K \longrightarrow_{\mathrm{p}} M' \bullet x.(\langle x, N'\rangle \bullet K')$	(pfst[])	$\mathtt{fst}[K] \longrightarrow_{\mathrm{p}} \mathtt{fst}[K']$,
(pv&$_2$)	$\langle V, N\rangle \bullet K \longrightarrow_{\mathrm{p}} N' \bullet y.(\langle M'', y\rangle \bullet K')$	(psnd[])	$\mathtt{snd}[L] \longrightarrow_{\mathrm{p}} \mathtt{snd}[L']$,
(p$v\vee_1$)	$\langle M\rangle\mathtt{inl} \bullet K \longrightarrow_{\mathrm{p}} M' \bullet x.(\langle x\rangle\mathtt{inl} \bullet K')$	(pnot$\langle\rangle$)	$\mathtt{not}\langle M\rangle \longrightarrow_{\mathrm{p}} \mathtt{not}\langle M'\rangle$,
(p$v\vee_2$)	$\langle N\rangle\mathtt{inr} \bullet K \longrightarrow_{\mathrm{p}} N' \bullet y.(\langle y\rangle\mathtt{inr} \bullet K')$	(p.())	$x.(S) \longrightarrow_{\mathrm{p}} x.(S')$.

First, we show that parallel reduction preserves values.

Proposition 2.3. *Suppose $M \longrightarrow_{\mathrm{p}} N$. Then, M is a value iff N is.*

Proof. The forward direction by induction on M being a value, the reverse by induction on N being so. □

Moreover, parallel reduction satisfies the diamond property. To prove this, we need a substitution lemma.

Lemma 2.4 (*Substitution*). *Let $G, G', V, V', K, K' \in$ DuCa with $G \longrightarrow_{\mathrm{p}} G'$, $V \longrightarrow_{\mathrm{p}} V'$, $K \longrightarrow_{\mathrm{p}} K'$. Then, for any variable x and covariable α,*

$$G\{V/x\} \longrightarrow_{\mathrm{p}} G'\{V'/x\},\ G\{K/\alpha\} \longrightarrow_{\mathrm{p}} G'\{K'/\alpha\}.$$

Proof. The proof is by a straightforward induction on $G \in$ DuCa. The base cases (of $G \equiv y, G \equiv x$, $G \equiv \alpha$ or $G \equiv \beta$) are trivial, since $G \equiv G'$. The induction step is done by a long case analysis on G and all cases are straightforwardly proven using the IH. □

Lemma 2.5. *The relation $\longrightarrow_{\mathrm{p}}$ defined above satisfies the diamond property. That is, for all $G, G_1, G_2 \in$ DuCa, if $G_1\ {}_{\mathrm{p}}\!\longleftarrow G \longrightarrow_{\mathrm{p}} G_2$, then there exists $G_c \in$ DuCa such that $G_1 \longrightarrow_{\mathrm{p}} G_c\ {}_{\mathrm{p}}\!\longleftarrow G_2$.*

Proof. See the Appendix.

Therefore, the βv-reduction relation is CR.

Lemma 2.6. *The βv-reduction relation is CR; that is, for all $G, G_1, G_2 \in$ DuCa, if $G_1 \overset{\beta v}{\twoheadleftarrow} G \overset{\beta v}{\twoheadrightarrow} G_2$, then there exists $G_c \in$ DuCa such that $G_1 \overset{\beta v}{\twoheadrightarrow} G_c \overset{\beta v}{\twoheadleftarrow} G_2$.*

Proof. By definition of the parallel reduction we have that $\overset{\beta v}{\longrightarrow}_{=} \subseteq \longrightarrow_{\mathrm{p}} \subseteq \overset{\beta v}{\twoheadrightarrow}$.

Taking transitive closures in this formula, we have that $\longrightarrow_{\mathrm{p}+} \equiv \overset{\beta v}{\twoheadrightarrow}$. But, since $\longrightarrow_{\mathrm{p}}$ satisfies the diamond property, $\longrightarrow_{\mathrm{p}+}$ is CR, by a simple diagram chase. □

An easier result is that the η-reduction relation is CR.

Lemma 2.7. *The η-reduction relation is CR; that is, for all $G, G_1, G_2 \in$ DuCa, if $G_1 \overset{\eta}{\twoheadleftarrow} G \overset{\eta}{\twoheadrightarrow} G_2$, then there exists $G_c \in$ DuCa such that $G_1 \overset{\eta}{\twoheadrightarrow} G_c \overset{\eta}{\twoheadleftarrow} G_2$.*

Proof. It suffices to show that $\overset{\eta}{\longrightarrow}$ satisfies the diamond property, since then the claim follows by a simple diagram chase. But this is straightforward: for every context C,

$$\begin{tikzcd}
 & C\{K\} \arrow[r, "\eta"] \arrow[dl, "\eta"'] \arrow[dr, "\eta"] & C\{x.(x \bullet K)\} \arrow[dl, "\eta"'] \arrow[dr, "\eta"] & \\
C\{K'\} \arrow[r, "\eta"] & C\{x.(x \bullet K')\} & C'\{K\} \arrow[r, "\eta"] & C'\{x.(x \bullet K)\}
\end{tikzcd}$$

and similarly for $C\{M\}$. □

Now, regarding βv-reductions, we do the following distinction.

Definition 2.8. All βv-reductions are called *simple reductions*, except if they happen by application of βL or βR rules; these latter are called $\mathtt{sub}_{\leqslant 1}$ or $\mathtt{sub}_{>1}$ reductions:

$V \bullet x.(S) \overset{\beta L}{\longrightarrow} S\{V/x\}$ is a $\mathtt{sub}_{\leqslant 1}$ reduction if x occurs at most once in S, otherwise it is a $\mathtt{sub}_{>1}$ reduction. Similarly for the βR rule.

The following lemma concerns η-reductions that destroy values.

Lemma 2.9. *Let V be a value, M a non-value term, and K a coterm. Then,*

$$V \bullet K \overset{\eta}{\longrightarrow} M \bullet K \quad \textit{implies} \quad M \bullet K \overset{\beta v}{\twoheadrightarrow} V \bullet K,$$

where $M \bullet K \overset{\beta v}{\twoheadrightarrow} V \bullet K$ involves only simple or $\mathtt{sub}_{\leqslant 1}$ reductions.

Proof. First note that, if $\overset{\eta}{\longrightarrow}$ reduces the whole of V, we trivially have:

$$V \bullet K \overset{\eta}{\longrightarrow} (V \bullet x).x \bullet K \overset{\beta R}{\longrightarrow} V \bullet K,$$

where x is fresh, and thus the βR-reduction is $\mathtt{sub}_{\leqslant 1}$.

So suppose that $\xrightarrow{\eta}$ reduces inside V. Since V is turned to a non-value, V cannot be of the type $[L]\mathtt{not}$. Repeating this argument several times, we come to the conclusion that the η-reduction above is in fact:

$$E\{W\} \bullet K \xrightarrow{\eta} E\{(W \bullet \alpha).\alpha\} \bullet K$$

where

$$E ::= \{\} \mid \langle E, W'\rangle \mid \langle W', E\rangle \mid \langle E\rangle\mathtt{inl} \mid \langle E\rangle\mathtt{inr}$$

Therefore, we proceed by induction on V and a case analysis on E. The case where $E \equiv \{\}$ is dealt with above. It also includes the base case $V \equiv x$.

For the inductive step, we have the following reductions.

$$\begin{aligned} \langle E\{W\}, W'\rangle \bullet K &\xrightarrow{\eta} \langle E\{(W \bullet \alpha).\alpha\}, W'\rangle \bullet K \xrightarrow{\nu} E\{(W \bullet \alpha).\alpha\} \bullet y.(\langle y, W'\rangle \bullet K) \\ &\overset{\textbf{(IH)}\ \beta\nu}{\twoheadrightarrow} E\{W\} \bullet y.(\langle y, W'\rangle \bullet K) \xrightarrow{\beta L} \langle E\{W\}, W'\rangle \bullet K \end{aligned}$$

and similarly for the $\langle W', E\{W\}\rangle$ case. Also,

$$\begin{aligned} \langle E\{W\}\rangle\mathtt{inl} \bullet K &\xrightarrow{\eta} \langle E\{(W \bullet \alpha).\alpha\}\rangle\mathtt{inl} \bullet K \xrightarrow{\nu} E\{(W \bullet \alpha).\alpha\} \bullet y.(\langle y\rangle\mathtt{inl} \bullet K) \\ &\overset{\textbf{(IH)}\ \beta\nu}{\twoheadrightarrow} E\{W\} \bullet y.(\langle y\rangle\mathtt{inl} \bullet K) \xrightarrow{\beta L} \langle E\{W\}\rangle\mathtt{inl} \bullet K \end{aligned}$$

and similarly for the $\langle E\{W\}\rangle\mathtt{inr}$ case. □

Then, we can prove the following lemmata.

Lemma 2.10. *If* $G, G_1, G_2 \in \mathtt{DuCa}$, *and* $G_1 \xleftarrow{\beta\nu} G \xrightarrow{\eta} G_2$, *then either*

- $G_2 \overset{\beta\nu}{\twoheadrightarrow} G$, *by use of simple or* $\mathtt{sub}_{\leqslant 1}$ *reductions*, *or*
- *if* $G \xrightarrow{\beta\nu} G_1$ *is simple or* $\mathtt{sub}_{\leqslant 1}$, *then there exists* G_c *such that* $G_1 \xrightarrow{\eta}_= G_c \xleftarrow{\beta\nu} G_2$, *and* $G_2 \xrightarrow{\beta\nu} G_c$ *is simple or* $\mathtt{sub}_{\leqslant 1}$; *otherwise, if* $G \xrightarrow{\beta\nu} G_1$ *is* $\mathtt{sub}_{>1}$, *then there exists* G_c *such that* $G_1 \overset{\eta}{\twoheadrightarrow} G_c \xleftarrow{\beta\nu} G_2$, *and* $G_2 \xrightarrow{\beta\nu} G_c$ *is* $\mathtt{sub}_{>1}$.

Proof. In the Appendix.

Lemma 2.11 (*Commutativity*). $\overset{\beta\nu}{\twoheadrightarrow}$ *and* $\overset{\eta}{\twoheadrightarrow}$ *commute*; *that is, for all* $G, G', G'' \in \mathtt{DuCa}$, *if* $G' \overset{\beta\nu}{\twoheadleftarrow} G \overset{\eta}{\twoheadrightarrow} G''$, *then there exists some* $G_c \in \mathtt{DuCa}$ *such that* $G' \overset{\eta}{\twoheadrightarrow} G_c \overset{\beta\nu}{\twoheadleftarrow} G''$.

Proof. Suppose that

$$G' \equiv H_m \xleftarrow{\beta\nu} \cdots \xleftarrow{\beta\nu} H_2 \xleftarrow{\beta\nu} H_1 \xleftarrow{\beta\nu} G \xrightarrow{\eta} G_1 \xrightarrow{\eta} G_2 \xrightarrow{\eta} \cdots \xrightarrow{\eta} G_n \equiv G''$$

and assume $n > 0$ (the case $n = 0$ is trivial). We do induction on m; the base case, $m = 0$, is trivial.

So fix some $m > 0$. We claim that there exist $u_1, u_2, \ldots, u_n \in \mathtt{DuCa}$ such that,

$$\begin{tikzcd} G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0"] & G_1 \arrow[r, "\eta"] \arrow[d, two heads, "\beta\nu^1"] & \cdots \arrow[r, "\eta"] & G_n \arrow[d, two heads, "\beta\nu^n"] \\ H_1 \arrow[r, two heads, "\eta"] & u_1 \arrow[r, two heads, "\eta"] & \cdots \arrow[r, two heads, "\eta"] & u_n \end{tikzcd} \quad \text{hence} \quad \begin{tikzcd} G \arrow[r, two heads, "\eta"] \arrow[d, "\beta\nu"] & G_n \arrow[d, two heads, "\beta\nu"] \\ H_1 \arrow[r, two heads, "\eta"] & u_n \end{tikzcd}$$

Thus, applying the claim, we only need to prove commutativity for the reduction chain,

$$G' \equiv H_m \xleftarrow{\beta v} \cdots \xleftarrow{\beta v} H_2 \xleftarrow{\beta v} H_1 \overset{\eta}{\twoheadrightarrow} u_n$$

for which the IH on m applies.

Hence, it suffices to prove our claim. By hypothesis, $H_1 \xleftarrow{\beta v^0} G \xrightarrow{\eta} G_1$. Now, by Lemma 2.10, if $\xrightarrow{\beta v^0}$ is a simple or $\mathtt{sub}_{\leqslant 1}$ reduction, then one of the following diagrams must be the case:

$$\begin{tikzcd}
G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0"] & G_1 \arrow[d, two heads, "\beta\nu^{1'}"] \\
H_1 \arrow[dr, "\eta(\mathtt{id})", "="'] & G \arrow[d, "\beta\nu^0"] \\
 & u_1 \equiv H_1
\end{tikzcd}
\qquad \text{or} \qquad
\begin{tikzcd}
G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0"] & G_1 \arrow[d, "\beta\nu^1"] \\
H_1 \arrow[r, "\eta", "\equiv"'] & u_1
\end{tikzcd}$$

In both cases $\overset{\beta v^{1'}}{\twoheadrightarrow}$ and $\xrightarrow{\beta v^1}$ include only simple or $\mathtt{sub}_{\leqslant 1}$ reductions. Thus, we can reuse this reasoning repeatedly and finally get the following diagram, which proves the claim in this case.

$$\begin{tikzcd}
G \arrow[r, two heads, "\eta"] \arrow[d, "\beta\nu^0"] & G_{n-1} \arrow[r, "\eta"] \arrow[d, "\beta\nu"] & G_n \arrow[d, two heads, "\beta\nu"] \\
H_1 \arrow[dddd, "\eta"] & w_1 \arrow[r, "\eta", "\equiv"'] \arrow[d, "\beta\nu"] & w_1' \arrow[d, two heads, "\beta\nu"] \\
 & w_2 \arrow[r, "\eta", "\equiv"'] \arrow[d, "\beta\nu"] & w_2' \arrow[d, two heads, "\beta\nu"] \\
 & \vdots \arrow[d, "\beta\nu"] & \vdots \arrow[d, two heads, "\beta\nu"] \\
 & v_{n-1} \equiv w_k \arrow[r, "\eta", "\equiv"'] & w_k' \equiv u_n
\end{tikzcd}$$

Now suppose $\xrightarrow{\beta v^0}$ is a $\mathtt{sub}_{>1}$ reduction. Then, by Lemma 2.10, one of the first two diagrams below must be the case.

$$\boxed{\begin{tikzcd}
G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0"] & G_1 \arrow[d, two heads, "\beta\nu^{1'}"] \\
H_1 \arrow[dr, "\eta(\mathtt{id})"] & G \arrow[d, "\beta\nu^0"] \\
 & u_1 \equiv H_1
\end{tikzcd}
\quad\Bigg|\quad
\begin{tikzcd}
G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0"] & G_1 \arrow[d, "\beta\nu^1"] \\
H_1 \arrow[r, two heads, "\eta"] & u_1
\end{tikzcd}
\quad\Bigg|\quad
\begin{tikzcd}
G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0"] & G_1 \arrow[d, two heads, "\beta\nu(\text{simple or } \mathtt{sub}_{\leq 1})"] \\
H_1 \arrow[dr, two heads, "\eta"] & w_1 \arrow[d, "\beta\nu(\mathtt{sub}_{>1})"] \\
 & u_1
\end{tikzcd}}$$

with $\overset{\beta v^{1'}}{\twoheadrightarrow}$ including only simple or $\mathtt{sub}_{\leqslant 1}$ reductions and $\xrightarrow{\beta v^1}$ being a $\mathtt{sub}_{>1}$ reduction. Hence, both diagrams have the form of the third diagram above.

Reasoning thus repeatedly and handling $G_i \overset{\beta v}{\twoheadrightarrow} w_i$ in the same way as $G \overset{\beta v^0}{\longrightarrow} H_1$ previously (where $\overset{\beta v^0}{\longrightarrow}$ was simple or $\mathtt{sub}_{\leqslant 1}$), we have the following diagram:

$$\begin{tikzcd}
G \arrow[r, "\eta"] \arrow[d, "\beta\nu^0(\mathtt{sub}_{>1})"] & G_1 \arrow[r, "\eta"] \arrow[d, two heads, "\beta\nu"] & G_2 \arrow[r, "\eta"] \arrow[d, two heads, "\beta\nu"] & \cdots \arrow[r, "\eta"] & G_{n-1} \arrow[r, "\eta"] \arrow[ddddd, two heads, "\beta\nu"] & G_n \arrow[ddddd, two heads, "\beta\nu"] \\
H_1 \arrow[dr, two heads, "\eta"] & w_1 \arrow[r, "\eta", "\equiv"' near end] \arrow[d, "\beta\nu(\mathtt{sub}_{>1})"] & w_1' \arrow[d, two heads, "\beta\nu"] & & & \\
& u_1 \arrow[dr, two heads, "\eta"] & w_2 \arrow[r, "\eta", "\equiv"' near end] \arrow[d, "\beta\nu(\mathtt{sub}_{>1})"] & \cdots & & \\
& & u_2 \arrow[dr, two heads, "\eta"] & & & \\
& & & \ddots \arrow[dr, two heads, "\eta"] & & \\
& & & & w_{n-1} \arrow[r, "\eta", "\equiv"' near end] \arrow[d, "\beta\nu(\mathtt{sub}_{>1})"] & w_{n-1}' \arrow[d, two heads, "\beta\nu"] \\
& & & & u_{n-1} \arrow[dr, two heads, "\eta"] & w_n \arrow[d, "\beta\nu(\mathtt{sub}_{>1})"] \\
& & & & & u_n
\end{tikzcd}$$

which proves the claim and the lemma. □

Combining the results above we prove confluence for the CBV reduction relation.

Theorem 2.12. *R_v is Church–Rosser.*

Proof. By Lemmata 2.6, 2.7 and 2.11, as in [2]. □

2.2. Strong normalization

In this section we are interested in the typed version of the Dual Calculus (DuCa) under CBV reduction, and our aim is to prove that the calculus is strongly normalizing. Of course, we have to place certain restrictions on the reduction rules, and particularly on η- and v-rules, as divergence is otherwise evident. We will call the resulting reduction relation $R_v^{\mathtt{SN}}$.

Regarding the restrictions on v-rules, divergencies may arise, for example, in the following cases.

$$\langle x\rangle\mathtt{inl} \bullet K \longrightarrow x \bullet y.(\langle y\rangle\mathtt{inl} \bullet K) \longrightarrow x \bullet y.(y \bullet z.(\langle z\rangle\mathtt{inl} \bullet K)) \longrightarrow \cdots$$

A reasonable restriction to avoid the above would be to restrict v-rules to non-value terms. Nevertheless, η-rules can help in the breaking of this restriction, as in

$$\langle x\rangle\mathtt{inl} \bullet K \longrightarrow \langle (x \bullet \alpha).\alpha\rangle\mathtt{inl} \bullet K \longrightarrow (x \bullet \alpha).\alpha \bullet y.(\langle y\rangle\mathtt{inl} \bullet K) \longrightarrow \cdots$$

Moreover, η-rules can diverge on their own:

$$K \longrightarrow x.(x \bullet K) \longrightarrow x.(x \bullet y.(y \bullet K)) \longrightarrow \cdots$$
$$K \longrightarrow x.(x \bullet K) \longrightarrow y.(y \bullet x.(x \bullet K)) \longrightarrow \cdots$$

Other divergences arise by the ability to create β-redices that can be immediately contracted.

A set of restrictions which resolves all of the above—and other—cases is proposed below.

Definition 2.13. $R_v^{\mathtt{SN}}$ is the one-step reduction relation yielded by the following rules, when these are applied to subobjects of $\mathtt{DuCa}$ objects, respecting restrictions $\mathbf{R} := \mathbf{R}v \cup \mathbf{R}\eta \cup \mathbf{R}\eta'$:

$$\begin{array}{lllllll}
(\beta\&_1) & \langle V, W\rangle \bullet \mathtt{fst}[K] & \to V \bullet K & \quad & (\eta L) & K & \to x.(x \bullet K),\\
(\beta\&_2) & \langle V, W\rangle \bullet \mathtt{snd}[L] & \to W \bullet L & & (\eta R) & M & \to (M \bullet \alpha).\alpha,\\
(\beta\vee_1) & \langle V\rangle\mathtt{inl} \bullet [K, L] & \to V \bullet K, & & & & \\
(\beta\vee_2) & \langle W\rangle\mathtt{inr} \bullet [K, L] & \to W \bullet L & & (\nu\&_1) & \langle M, N\rangle \bullet K & \to M \bullet x.(\langle x, N\rangle \bullet K),\\
(\beta\neg) & [K]\mathtt{not} \bullet \mathtt{not}\langle M\rangle & \to M \bullet K & & (\nu\&_2) & \langle V, M\rangle \bullet K & \to M \bullet y.(\langle V, y\rangle \bullet K),\\
(\beta L) & V \bullet x.(S) & \to S\{V/x\} & & (\nu\vee_1) & \langle M\rangle\mathtt{inl} \bullet K & \to M \bullet x.(\langle x\rangle\mathtt{inl} \bullet K),\\
(\beta R) & (S).\alpha \bullet K & \to S\{K/\alpha\} & & (\nu\vee_2) & \langle M\rangle\mathtt{inr} \bullet K & \to M \bullet y.(\langle y\rangle\mathtt{inr} \bullet K).
\end{array}$$

Rv. In the v-rules, M is a non-value term.
Rη. In the η-rules, M and K are non-abstraction objects (i.e. not $(S).\alpha$ or $x.(S)$).
Rη'. η-rules are not allowed to be applied to terms [resp. coterms] that are immediately followed by [immediately follow] a cut '$\bullet$', or to values that are immediate subjects to $\langle\_, N\rangle$, $\langle V, \_\rangle$, $\langle\_\rangle\mathtt{inl}$ or $\langle\_\rangle\mathtt{inr}$.
For $G, H \in \mathtt{DuCa}$, $(G, H) \in R_v^{\mathtt{SN}}$ is written simply as $G \longrightarrow H$.

We will prove that these restrictions are adequate for our purpose: $R_v^{\mathtt{SN}}$ is indeed SN. The steps we follow for this proof are the following.

- We devise a simplified version[6] of DuCa, called Dual Calculus* (DuCa*), and prove SN for DuCa* using the method of *reducibility sets*.
  DuCa* is accompanied by a reduction relation that uses only βR and βL rules. In DuCa* we have a "neutralizing dot" symbol '$_\odot$' for variable and covariable abstractions, over which β-reductions are disallowed. Moreover, in the absence of all β-rules apart from βR and βL, we use the more general notion of neutral terms instead of values (see Definition 2.16).
  We also introduce a translation from DuCa to DuCa* which η- and v-expands objects and preserves β-reductions. The translation thus identifies terms with their η-reducts and converts v-reductions to β-reductions.
- We present the CBV CPS translation of the Dual Calculus, which was introduced in [20]. Under this translation reductions are preserved and in some cases one-step reductions are preserved or lengthened. Moreover, the reduction relation of the target calculus is SN.
- Using the results of the previous steps, we prove SN.

Note 2.14. Having introduced a translation from DuCa to the Sλ-calculus in Section 1.3.2 which preserves almost all one-step reductions, it would be expected to try and prove SN for DuCa through the known SN result for Sλ-calculus (as proven in [1]). The problem that would arise is that here we use η-expanding rules in our reduction relation, in contrast to Section 1.3.2 and [1], where η-contracting ones are used. Moreover, Sλ does not contain v-rules. For these reasons, the proof in [1] cannot be easily adjusted to our purposes, so we prefer following the above plan (which is also more fun).

Let us recall the definition of Strong Normalization.

Definition 2.15. Let $R \subset U^2$ be some reduction relation on some universe U. Then,

- if $G \in U$, then G is Strongly Normalizing (under R), or simply SN, if there is no infinite R-reduction sequence starting from G.
- R is Strongly Normalizing if all elements of U are SN.

Moreover, if $G \in U$ is SN, then $l(G)$ is the length of the longest R-reduction path starting from G.

2.2.1. *The reduction relation of* DuCa* *is* SN

We introduce an auxiliary calculus similar to DuCa.

[6] Note that we use DuCa* only as a step for proving SN for DuCa; otherwise DuCa* has no use.

Definition 2.16 (`DuCa`*, *the sequent calculus* `GW`* *and the reduction relation* R^*). `DuCa`* is a typed calculus consisting of Types and Objects. The set of objects is the union of the sets of Terms, Coterms and Statements:

Type	A, B	$::=$	$X \mid A\&B \mid A \vee B \mid \neg A$
Object	G, H	$::=$	$M \mid K \mid S$
Term	M, N	$::=$	$M_{\mathrm{n}} \mid (S).\alpha$
Neutral Term	M_{n}	$::=$	$(S)_{\odot}\alpha$
Coterm	K, L	$::=$	$K_{\mathrm{n}} \mid x.(S)$
Neutral Coterm	K_{n}	$::=$	$x_{\odot}(S)$
Statement	S, T	$::=$	$M \bullet K$

An object G of `DuCa`* is *neutral* if it is a *neutral term*, or a *neutral coterm*, or a *statement*.

R^* is the one-step reduction relation yielded by the following rules, when these are applied to subobjects of `DuCa`* objects, respecting restrictions $\mathbf{R}\eta'$:

$$(\beta L)\ M_{\mathrm{n}} \bullet x.(S) \rightarrow S\{M_{\mathrm{n}}/x\}$$
$$(\beta R)\ (S).\alpha \bullet K \rightarrow S\{K/\alpha\}$$

$\mathbf{R}\eta'$. The η-rules are not allowed to be applied to terms [resp. coterms] that are immediately followed by [immediately follow] some cut '$\bullet$'.

For $G, H \in$ `DuCa`, $(G, H) \in R^*$ is written as $G \longrightarrow H$.

The typing rules for `DuCa`* are the same as those of `DuCa` (i.e. of system `GW`), with the addition of RI$_\odot$ and LI$_\odot$ rules introducing '$_\odot$':

$$\frac{x : A, \Gamma \,\rule{2pt}{8pt}\, S \mapsto \Theta}{x_{\odot}(S) : A \,\rule{2pt}{8pt}\, \Gamma \rightarrow \Theta} LI_{\odot} \qquad \frac{\Gamma \,\rule{2pt}{8pt}\, S \mapsto \Theta, \alpha : A}{\Gamma \rightarrow \Theta \,\rule{2pt}{8pt}\, (S)_{\odot}\alpha : A} RI_{\odot}$$

The addition of these rules yields the sequent calculus `GW`*.

Note that by the above definition neutral objects are preserved by reduction: if G is neutral and $G \longrightarrow G'$, then G' is neutral.

As noted above, `DuCa`* differs from `DuCa` in the addition of '$_\odot$' and the usage of the more general notion of neutral terms instead of values. In particular, '$_\odot$' is a "neutralizing dot" for abstractions, as we cannot apply β-rules over it; for example:

$$M_{\mathrm{n}} \bullet x.(S) \rightarrow S\{M_{\mathrm{n}}/x\} \quad \text{but } M_{\mathrm{n}} \bullet x_{\odot}(S) \not\rightarrow S\{M_{\mathrm{n}}/x\}$$

On the other hand, in the absence of extra β-rules, as in the `DuCa`, we can simplify the distinctions inside terms and make use of neutral terms instead of values.

Now, we introduce notation regarding derivable sequents that type objects of `DuCa`*. We also introduce notation for "reduction" between such sequents.

Definition 2.17. Let $G \in$ `DuCa`* ; we introduce the set T_G of sequents typing G as follows. For any sequent s,

- $\sigma \in T_G(A, \Gamma, \Theta)$ if σ is derivable in `GW`* and either

  $\sigma \equiv \Gamma \rightarrow \Theta \,\rule{2pt}{8pt}\, M : A$ and G is the term M, or
  $\sigma \equiv K : A \,\rule{2pt}{8pt}\, \Gamma \rightarrow \Theta$ and G is the coterm K, or
  $\sigma \equiv \Gamma \,\rule{2pt}{8pt}\, S \mapsto \Theta$ and G is the statement S.

- $\sigma \in T_G(A)$ if $\sigma \in T_G(A, \Gamma, \Theta)$ for some Γ, Θ.
- $\sigma \in T_G(\Gamma, \Theta)$ if $\sigma \in T_G(A, \Gamma, \Theta)$ for some type A.
- $\sigma \in T_G$ if $\sigma \in T_G(A, \Gamma, \Theta)$ for some A, Γ, Θ.

We say that G is *typed* if $T_G \neq \emptyset$.

Moreover, if $G, G' \in$ `DuCa`*, $\sigma \in T_G(A, \Gamma, \Theta)$, $\sigma' \in T_{G'}(A, \Gamma, \Theta)$ and $G \longrightarrow G'$ then $\sigma \longrightarrow \sigma'$.

According to the above, if $\sigma \in T_G$ and $\sigma \in T_H$, then $G \equiv H$. Moreover, for all statements S and types A, $T_S = T_S(A)$. This reflects the fact that *statements are not assigned types*, but are typed iff their components are typed with the same type. Further, an object G of the calculus can be assigned more than one type and, on the other hand, we allow for objects G with $T_G = \emptyset$. For example, $\langle x, x\rangle \bullet [a, a]$ is not typed.

Now, subject reduction holds for DuCa*, along with two other useful properties.

Proposition 2.18 (*Weakening, M for x, subject reduction*). *Let $G, G', M, K \in$ DuCa*,*

1. *If $T_G(A, \Gamma, \Theta) \neq \emptyset$ then $T_G(A, \Gamma \uplus (x : B), \Theta) \neq \emptyset$ and $T_G(A, \Gamma, \Theta \uplus (\alpha : B)) \neq \emptyset$, any fresh variable x and fresh covariable α.*
2. *If $T_G(A, \Gamma \uplus (x : B), \Theta) \neq \emptyset$ and $T_M(B, \Gamma, \Theta) \neq \emptyset$ then $T_{G\{M/x\}}(A, \Gamma, \Theta) \neq \emptyset$, and if $T_G(A, \Gamma, \Theta \uplus (\alpha : B)) \neq \emptyset$ and $T_K(B, \Gamma, \Theta) \neq \emptyset$ then $T_{G\{K/\alpha\}}(A, \Gamma, \Theta) \neq \emptyset$.*
3. *If $G \longrightarrow G'$ and $\sigma \in T_G(A, \Gamma, \Theta)$, some sequent σ, then there exists a sequent $\sigma' \in T_{G'}(A, \Gamma, \Theta)$ and thus $\sigma \longrightarrow \sigma'$.*

Proof. 1 (weakening) is proven by straightforward induction on the sequent typing G, and the same holds for 2 (M for x). For 3 (subject reduction), the case of G reducing to G' by η-rules is straightforward. As far as β-rules are concerned, the claim is proven by substituting proofs follows from 1 and 2. □

Neutral objects of DuCa* are like boxes the inside of which cannot be accessed by outer reductions; neutral terms are like variables with extra structure and neutral coterms are like covariables with extra structure. In particular, substituting a neutral term [resp. coterm] for a variable [covariable] does not produce new redices apart from those inside the term [coterm]. This remark is implicitly used in the proof of the following lemma.

Lemma 2.19. *If M is a neutral term and S a statement, and both M and S are* SN, *then $M \bullet x.(S)$ is* SN *for any variable x.*

If K is a neutral coterm and S a statement, and both K and S are SN, *then $(S).\alpha \bullet K$ is* SN *for any covariable α.*

Proof. We prove only the first claim; the second is proven similarly.

We do induction on $l(M) + l(S)$. Let $S_0 \equiv M \bullet x.(S)$. If $S_0 \longrightarrow S^?$, then either $S^? \equiv M' \bullet x.(S')$, with $M \longrightarrow M'$ or $S \longrightarrow S'$, or $S^? \equiv S\{M/x\}$.

In the former case, we have $l(M') + l(S') < l(M) + l(S)$, so $M' \bullet x.(S')$ is SN by the IH.

Now, if we prove that in the latter case $S\{M/x\}$ is SN, then S_0 reduces only to SN objects, so S_0 is SN. In order to prove that, we show something stronger:

For any statement $S \in$ SN and variable x, if we mark the occurrences of x inside S by $1, 2, \ldots, n$, then for any tuple $M_1, \ldots, M_n$ of neutral SN terms,

$$S_1 \equiv S\{M_1/_1x, M_2/_2x, \ldots, M_n/_nx\} \in \mathtt{SN}$$

where $M_i/_ix$ denotes the substitution of M_i for the i-occurrence of x in S.

The proof of this claim is by induction on $l(S)$. For the base case, that is of S being in normal form, we have that the redexes inside S_1 are exactly those inside the M_i's, since S does not contain any redexes and all M_i's are neutral. But then S_1 is SN, since all M_i's are SN.

For the inductive step, assume $l(S) > 0$ and suppose that there is some infinite reduction sequence starting from S_1. Then, since the M_i's are SN, in this sequence it must be the case that,

$$S_1 \twoheadrightarrow S_2 \longrightarrow S_1', \text{ with } S_2 \equiv S\{M_1'/_1x, \ldots, M_n'/_nx\}\text{ , some } M_i \twoheadrightarrow M_i',$$
$$i = 1, \ldots, n, \text{ and } S_1' \equiv S'\{N_1/_1x, \ldots, N_{n+k}/_{n+k}x\}$$

where the reduction $S \longrightarrow S'$ produces $k \in \mathbb{Z}$ new occurrences of x. In S_1', all x's are replaced by some of the M_i''s (denoted by N_j, $j = 1, \ldots, n+k$). Then, by IH, since $N_1, \ldots, N_{n+k}$ are all SN and neutral, and S' is SN with $l(S') < l(S)$, S_1' is SN, contradiction to this being an infinite reduction sequence. Hence, S_1 is SN, and thus our initial $S\{M/x\}$ is SN. □

A similar idea is applied in the following lemma.

Lemma 2.20. *Let S be a statement and α a covariable. If $S\{K/\alpha\}$ is* SN, *for some coterm K, then S is* SN.

Proof. See the Appendix.

Back to the sequent calculus, we order derivable sequents by their degree.

Definition 2.21. Let $G \in \mathtt{DuCa}^*$ and $\sigma \in T_G$. Then, the degree $d(\sigma)$ of σ is

$$d(\sigma) := (c(\sigma), \neg\mathtt{cut}(\sigma))$$

where $c(\sigma)$ is an integer and $\neg\mathtt{cut}(\sigma)$ a boolean. These are given by:
- if $\sigma \equiv K : A \,\rule{2pt}{7pt}\, \Gamma \rightarrow \Theta$, then $c(\sigma) = c(\Gamma) + c(\Theta)$, $\neg\mathtt{cut}(\sigma) = 1$,
- if $\sigma \equiv \Gamma \rightarrow \Theta \,\rule{2pt}{7pt}\, M : A$, then $c(\sigma) = c(\Gamma) + c(\Theta)$, $\neg\mathtt{cut}(\sigma) = 1$,
- if $\sigma \equiv \Gamma \,\rule{2pt}{7pt}\, S \mapsto \Theta$, then $c(\sigma) = c(\Gamma) + c(\Theta)$, $\neg\mathtt{cut}(\sigma) = 0$,
- if $\Gamma \equiv x_1 : A_1, x_2 : A_2, \ldots , x_n : A_n$, then $c(\Gamma) = c(A_1) + c(A_2) + \cdots + c(A_n)$,
- if $\Theta \equiv \alpha_1 : B_1, \alpha_2 : B_2, \ldots , \alpha_n : B_m$, then $c(\Theta) = c(B_1) + c(B_2) + \cdots + c(B_m)$,
- if A is some type, then $c(A)$ is its complexity, that is the number of connectives contained in A.

We order degrees lexicographically.

We define the set of reducible sequents, which is a subset of derivable sequents.

Definition 2.22. The set of reducible sequents Red is defined by induction on the degree of (derivable) sequents:
- if $d(\sigma) = (0, n)$, $n \in \{0, 1\}$ and $\sigma \in T_G$, then $\sigma \in \mathtt{Red} \iff G \in \mathtt{SN}$
- if $d(\sigma) = (c, n)$, $n \in \{0, 1\}$, $c > 0$ and $\sigma \in T_G$, then $\sigma \in \mathtt{Red} \iff cl(\sigma) \subset \mathtt{Red}$
- if $d(\sigma) = (c, 1)$, $c > 0$, $\sigma \equiv \Gamma \rightarrow \Theta \,\rule{2pt}{7pt}\, M : A$, $\sigma \in T_M$, then $cl(\sigma)$ is the set:

$$\sigma_2 \equiv \Gamma, \Gamma_0 \,\rule{2pt}{7pt}\, M \bullet K_0 \mapsto \Theta, \Theta_0$$

  if $\sigma_1 \equiv K_0 : A \,\rule{2pt}{7pt}\, \Gamma_0 \rightarrow \Theta_0 \in (\mathtt{Red} \cap T_{K_0})$, $d(\sigma_1) < d(\sigma)$, $d(\sigma_2) < d(\sigma)$, and if M is neutral, then K_0 is neutral.[7]
- if $d(\sigma) = (c, 1)$, $c > 0$, $\sigma \equiv K : A \,\rule{2pt}{7pt}\, \Gamma \rightarrow \Theta$, $\sigma \in T_K$, then $cl(\sigma)$ is the set:

$$\sigma_2 \equiv \Gamma, \Gamma_0 \,\rule{2pt}{7pt}\, M_0 \bullet K \mapsto \Theta, \Theta_0$$

  if $\sigma_1 \equiv \Gamma_0 \rightarrow \Theta_0 \,\rule{2pt}{7pt}\, M_0 : A \in (\mathtt{Red} \cap T_{M_0})$, $d(\sigma_1) < d(\sigma)$, $d(\sigma_2) < d(\sigma)$, and M_0 is neutral.
- if $d(\sigma) = (c, 0)$, $c > 0$, $\sigma \equiv \Gamma \,\rule{2pt}{7pt}\, S \mapsto \Theta$, $\sigma \in T_S$, then $cl(\sigma)$ is the union of the sets:

$$\sigma_1 \equiv \Gamma \rightarrow \Theta - \{\beta : B\} \,\rule{2pt}{7pt}\, (S)_{\odot}\beta : B, \text{ if } \beta : B \in \Theta,\ c(B) > 0$$
$$\sigma_2 \equiv y_{\odot}(S) : B \,\rule{2pt}{7pt}\, \Gamma - \{y : B\} \rightarrow \Theta, \text{ if } y : B \in \Gamma,\ c(B) > 0$$

Note that the above definition is valid. In all cases, the question of $\sigma \in \mathtt{Red}$ reduces to questions of $\sigma' \in \mathtt{Red}$, with $d(\sigma') < d(\sigma)$. For example, in the last case we have that, for all such σ_1, $d(\sigma_1) < d(\sigma)$, since we subtract a non-base type B from Θ, and σ being derivable implies σ_1 is derivable.

Note also that, if $\sigma \longrightarrow \sigma'$, then $d(\sigma) = d(\sigma')$.

The following proposition assures that for any type A we can find terms and coterms typed with A using only variables and covariables of base type.

Proposition 2.23. *For any type A there exist derivable sequents*:

$$\sigma_1 \equiv \Gamma_1 \rightarrow \Theta_1 \,\rule{2pt}{7pt}\, M : A$$
$$\sigma_2 \equiv K : A \,\rule{2pt}{7pt}\, \Gamma_2 \rightarrow \Theta_2$$

such that $d(\sigma_1) = d(\sigma_2) = (0, 1)$ and M, K are neutral and SN.

[7] This is in fact an abbreviation for

$$cl(\sigma) := \{\sigma_2 \mid \exists A, K_0, \Gamma_0, \Theta_0, \sigma_1.[\,(\sigma_2 \equiv \Gamma, \Gamma_0 \,\rule{2pt}{7pt}\, M \bullet K_0 \mapsto \Theta, \Theta_0) \wedge (\sigma_1 \equiv K_0 : A \,\rule{2pt}{7pt}\, \Gamma_0 \rightarrow \Theta_0) \wedge (\sigma_1 \in (\mathtt{Red} \cap T_{K_0})) \wedge (d(\sigma_1) < d(\sigma)) \wedge (d(\sigma_2) < d(\sigma)) \wedge (M \text{ neutral} \implies K_0 \text{ neutral})]\}$$

Note that σ, σ_1 being derivable implies that σ_2 is derivable, thus $d(\sigma_2)$ is defined.

Proof. We derive σ_1 by

$$\dfrac{\dfrac{x : X \rightarrow \alpha : X, \beta : A \mid x : X \quad \alpha : X \mid x : X \rightarrow \alpha : X, \beta : A}{x : X \mid x \bullet \alpha \mapsto \alpha : X, \beta : A}}{x : X \rightarrow \alpha : X \mid (x \bullet \alpha)_{\odot}\beta : A}$$

and similarly σ_2. □

The following lemma shows the relation between the reducibility set Red of sequents and the set SN of strongly normalizing objects of DuCa*.

Lemma 2.24. *Let σ be some derivable sequent, then,*
CR1: *If $\sigma \in (T_G \cap \mathtt{Red})$, some G, then G is* SN.
CR3: *If $\sigma \in T_G$, some neutral G, and $\sigma \longrightarrow \sigma'$ implies that $\sigma' \in \mathtt{Red}$, then $\sigma \in \mathtt{Red}$.*
This implies:
CR3′: *If $\sigma \in T_G$, G neutral and* SN, *then $\sigma \in \mathtt{Red}$.*
CR2: *If $\sigma \in (\mathtt{Red} \cap T_G)$, some G, and $\sigma \longrightarrow \sigma'$, then $\sigma' \in \mathtt{Red}$.*

Proof. See the Appendix.

A straightforward corollary of the lemma is the following.

Corollary 2.25. *Let G be some object of* DuCa*, *then*:
- *If G is neutral and* SN, *then $T_G \subset \mathtt{Red}$.*
- *If $(T_G \cap \mathtt{Red}) \neq \emptyset$ and G is neutral, then $T_G \subset \mathtt{Red}$.*
- *If K is some coterm and $(T_K \cap \mathtt{Red}) \neq \emptyset$, then $T_K \subset \mathtt{Red}$.*

Proof. The first claim is clear from CR3′.

For the second, if $\sigma \in (T_G \cap \mathtt{Red})$, then, by CR1, G is SN, so $T_G \subset \mathtt{Red}$ by first claim.

For the last claim, if K is neutral, then we use the previous claim. Otherwise, assume $K \equiv x.(S)$ and take some $\sigma \equiv x.(S) : A \mid \Gamma \rightarrow \Theta \in T_K$. Then, $\sigma \in \mathtt{Red}$ iff for all neutral M_0 and $\sigma_1 \equiv \Gamma_0 \rightarrow \Theta_0 \mid M_0 : A \in (\mathtt{Red} \cap T_{M_0})$ with $d(\sigma_1) < d(\sigma)$:

if $\sigma_2 \equiv \Gamma, \Gamma_0 \mid M_0 \bullet x.(S) \mapsto \Theta, \Theta_0$ and $d(\sigma_2) < d(\sigma)$, then $\sigma_2 \in \mathtt{Red}$.

For this, it suffices to show that $M_0 \bullet x.(S)$ is SN, by first claim. But $x.(S)$ is SN, by hypothesis and CR1, and thus S is SN. Moreover, by CR1, M_0 is also SN, thus, by Lemma 2.19, $M_0 \bullet x.(S)$ is SN. □

We can prove the following lemma for sequents typing non-neutral terms.

Lemma 2.26. *Let $\sigma \equiv \Gamma \rightarrow \Theta \mid (S).\alpha : A$ be some derivable sequent. If, for all coterms L with $T_L(A) \cap \mathtt{Red} \neq \emptyset$, we have $T_{S\{L/\alpha\}} \subset \mathtt{Red}$, then $\sigma \in \mathtt{Red}$.*

Proof. Assume the hypothesis. $\sigma \in \mathtt{Red}$ iff for all coterms K_0 and sequents $\sigma_1 \in (T_{K_0} \cap \mathtt{Red})$ with $d(\sigma_1) < d(\sigma)$ and $\sigma_1 \equiv K_0 : A \mid \Gamma_0 \rightarrow \Theta_0$:

$$\text{if } \sigma_2 \equiv \Gamma, \Gamma_0 \mid (S).\alpha \bullet K_0 \mapsto \Theta, \Theta_0 \text{ and } d(\sigma_2) < d(\sigma), \text{ then } \sigma_2 \in \mathtt{Red}.$$

Now take any such σ_1, K_0, σ_2. By Corollary 2.25, $\mathtt{Red} \supset T_\alpha \neq \emptyset$, $\therefore T_S \subset \mathtt{Red}$, by hypothesis. Since $(S).\alpha$ is typed, S is also typed, thus, by CR1, $S \in \mathtt{SN}$. Since K_0 is also SN, by CR1, we show by induction on $l(S) + l(K_0)$ that, if $\sigma_1 \in (T_{K_0}(A) \cap \mathtt{Red})$ and for all coterms L with $T_L(A) \cap \mathtt{Red} \neq \emptyset$ we have $T_{S\{L/\alpha\}} \subset \mathtt{Red}$, then $\sigma_2 \in \mathtt{Red}$.

So suppose that $\sigma_2 \longrightarrow \sigma_2'$. By CR3, it suffices to show that $\sigma_2' \in \mathtt{Red}$. Now, σ_2' may be:
- $\sigma_2' \equiv \Gamma, \Gamma_0 \mid S\{K_0/\alpha\} \mapsto \Theta, \Theta_0$, where, since $\sigma_1 \in (T_{K_0}(A) \cap \mathtt{Red})$, we have, by hypothesis, $T_{S\{K_0/\alpha\}} \subset \mathtt{Red}$, $\therefore \sigma_2' \in \mathtt{Red}$.
- $\sigma_2' \equiv \Gamma, \Gamma_0 \mid (S').\alpha \bullet K_0' \mapsto \Theta, \Theta_0$, where $S \longrightarrow S'$ or $K_0 \longrightarrow K_0'$.

By CR2, $\sigma_1' \equiv K_0' : A \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} \Gamma_0 \rightarrow \Theta_0 \in \texttt{Red}$. Thus, if we show that $T_{S'\{L/\alpha\}} \subset \texttt{Red}$, for all relevant L, then we can use the IH on $l(S) + l(K_0)$ and get $\sigma_2' \in \texttt{Red}$.

Now take some relevant L. By Corollary 2.25, it suffices to show that $S'\{L/\alpha\}$ is $\texttt{SN}$. Since $(S).\alpha$ and L are typed with A, $(S).\alpha \bullet L$ is also typed, $\therefore S\{L/\alpha\}$ is typed, $\therefore S\{L/\alpha\} \in \texttt{SN}$, by hypothesis and CR1. But $S \longrightarrow S'$ implies that $S\{L/\alpha\} \longrightarrow S'\{L/\alpha\}$, $\therefore S'\{L/\alpha\}$ is $\texttt{SN}$. □

Now, strong normalization of $\texttt{DuCa*}$ follows from the next theorem.

Theorem 2.27. *Let G be some element of* $\texttt{DuCa*}$ *with free variables amongst $x_1, x_2, \dots, x_n$ and covariables amongst $\alpha_1, \alpha_2, \dots, \alpha_m$.*

Then, for all coterms L_i with $T_{L_i} \cap \texttt{Red} \neq \emptyset$, $i = 1, \dots, m$, and neutral terms N_j with $T_{N_j} \cap \texttt{Red} \neq \emptyset$, $j = 1, \dots, n$,

$$T_{G\{f\}} \subset \texttt{Red}, \quad \textit{where} \quad f := N_1/x_1, \dots, N_n/x_n, L_1/\alpha_1, \dots, L_m/\alpha_m$$

Proof. Note first that if $G\{f\}$ is not typed, then the claim trivially holds. Assume then that $T_{G\{f\}} \neq \emptyset$. We do induction on G. The base cases, that is of $G \equiv x$ or $G \equiv \alpha$, are clear by Corollary 2.25.

For the inductive step, we do a case analysis on G, showing only the most difficult cases. Let $\sigma \in T_{G\{f\}}(A)$, some type A:

- $G \equiv \langle M, N\rangle$. Then $G\{f\} \equiv \langle M\{f\}, N\{f\}\rangle$, thus both $M\{f\}, N\{f\}$ are typed. By IH, $T_{M\{f\}} \subset \texttt{Red}$ and $T_{N\{f\}} \subset \texttt{Red}$, so, by CR1, both $N\{f\}$ and $M\{f\}$ are $\texttt{SN}$.
  But then $\langle M\{f\}, N\{f\}\rangle \equiv \langle M, N\rangle\{f\}$ is $\texttt{SN}$, so, by Corollary 2.25, $\sigma \in \texttt{Red}$.
- $G \equiv (S)_\odot\alpha$. Then $G\{f\} \equiv (S\{f, \alpha/\alpha\})_\odot\alpha$, thus $S\{f, \alpha/\alpha\}$ is typed. By IH, $T_{S\{f,\alpha/\alpha\}} \subset \texttt{Red}$, therefore, by CR1, $S\{f, \alpha/\alpha\}$ is $\texttt{SN}$.
  But then $(S\{f, \alpha/\alpha\})_\odot\alpha$ is $\texttt{SN}$, so, by Corollary 2.25, $\sigma \in \texttt{Red}$.
- $G \equiv (S).\alpha$. By IH, $T_{S\{f,L/\alpha\}} \subset \texttt{Red}$, for any coterm L with $T_L \cap \texttt{Red} \neq \emptyset$. Then, $T_{S\{f,L/\alpha\}} \subset \texttt{Red}$ for any coterm L with $T_L(A) \cap \texttt{Red} \neq \emptyset$. Now $S\{f, L/\alpha\} \equiv (S\{f\})\{L/\alpha\}$, since f is a substitution in $(S).\alpha$ and it does not introduce new α's. Moreover, $\sigma \equiv \Gamma \rightarrow \Theta \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} (S\{f\}).\alpha : A$, some Γ, Θ, so, by Lemma 2.26, $\sigma \in \texttt{Red}$.
- $G \equiv x.(S)$. By definition and Corollary 2.25, it suffices to show that, for every relevant neutral M_0 and $\sigma_1 \in (T_{M_0}(A) \cap \texttt{Red})$, $M_0 \bullet (x.(S)\{f\})$ is $\texttt{SN}$. Now $G\{f\} \equiv x.(S\{f, x/x\})$, thus $S\{f, x/x\}$ is typed and, by IH and CR1, $S\{f, x/x\} \in \texttt{SN}$. Moreover, $M_0 \in \texttt{SN}$ by CR1, therefore, $M_0 \bullet (x.(S)\{f\})$ is $\texttt{SN}$ by Lemma 2.19.
- $G \equiv M \bullet K$. Then $G\{f\} \equiv M\{f\} \bullet K\{f\}$, thus both $M\{f\}, K\{f\}$ are typed: say with type B. There are two subcases:
  - $G \equiv (S).\alpha \bullet K$. Let
    $$\sigma \equiv \Gamma \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} ((S).\alpha \bullet K)\{f\} \mapsto \Theta$$
    Since σ is derivable, σ_1 and σ_2 are also derivable:
    $$\sigma_1 \equiv \Gamma \rightarrow \Theta \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} (S).\alpha\{f\} : B$$
    $$\sigma_2 \equiv K\{f\} : B \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} \Gamma \rightarrow \Theta$$
    By IH, $T_{(S).\alpha\{f\}} \subset \texttt{Red}$, so $\sigma_1' \in (\texttt{Red} \cap T_{(S).\alpha\{f\}})$, where
    $$\sigma_1' \equiv \Gamma \rightarrow \Theta, \beta : A^b \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} (S).\alpha\{f\} : B$$
    with A^b some big type and β fresh.
    Now note that, by IH, $\sigma_2 \in \texttt{Red}$ and, because of A^b, $d(\sigma_2) < d(\sigma_1')$. Then, by definition, $\sigma_1' \in \texttt{Red}$ implies $\sigma' \in (T_{((S).\alpha\bullet K)\{f\}} \cap \texttt{Red})$, where
    $$\sigma' \equiv \Gamma \mathbin{\rule[-0.2ex]{0.2em}{1.4ex}} ((S).\alpha \bullet K)\{f\} \mapsto \Theta, \beta : A^b$$
    with $d(\sigma') = (c, 0) < (c, 1) = d(\sigma_1')$, some c. But then, by Corollary 2.25, $\sigma \in \texttt{Red}$.
  - $G \equiv M_0 \bullet K$, M_0 neutral: treated dually as the above case. □

Corollary 2.28. *If $G \in$* $\texttt{DuCa*}$ *is typed, then G is* $\texttt{SN}$.

Proof. Straightforward from the previous theorem and CR1. □

2.2.2. Translation from DuCa *to* DuCa*

Although objects of DuCa are also objects of DuCa*, we devise a transition function from the former calculus to the latter so that we can apply some preprocessing. In particular, the translation function deliberately η- and v-expands terms and coterms, to remedy the absence of η- and v-rules in DuCa*.

Definition 2.29. We define functions $(\_)^{\textcircled{\eta}}$ and $(\_)^{\textcircled{v}}$ from objects of DuCa to objects of DuCa* by mutual recursion, as follows (note below that β, γ, y, z are fresh in their contexts):

$$(K)^{\textcircled{\eta}} \equiv \begin{cases} y.(y \bullet (L)^{\textcircled{v}}) & \text{if } K \equiv y.(y \bullet L) \\ y.(y \bullet (K)^{\textcircled{v}}) & \text{otherwise} \end{cases} \qquad (M)^{\textcircled{\eta}} \equiv \begin{cases} ((N)^{\textcircled{v}} \bullet \beta).\beta & \text{if } M \equiv (N \bullet \beta).\beta \\ ((M)^{\textcircled{v}} \bullet \beta).\beta & \text{ow} \end{cases}$$

$$\begin{aligned}
(M \bullet K)^{\textcircled{v}} &\equiv (M)^{\textcircled{v}} \bullet (K)^{\textcircled{v}} \\
(x.(S))^{\textcircled{v}} &\equiv x.((S)^{\textcircled{v}}) & (\alpha)^{\textcircled{v}} &\equiv \alpha \\
(\mathtt{fst}[K])^{\textcircled{v}} &\equiv \mathtt{fst}[(K)^{\textcircled{\eta}}] & (\mathtt{snd}[K])^{\textcircled{v}} &\equiv \mathtt{snd}[(K)^{\textcircled{\eta}}] \\
([K, L])^{\textcircled{v}} &\equiv [(K)^{\textcircled{\eta}}, (L)^{\textcircled{\eta}}] & ([K]\mathtt{not})^{\textcircled{v}} &\equiv [(K)^{\textcircled{\eta}}]\mathtt{not} \\
((S).\alpha)^{\textcircled{v}} &\equiv ((S)^{\textcircled{v}}).\alpha & (x)^{\textcircled{v}} &\equiv x \\
(\langle V\rangle\mathtt{inl})^{\textcircled{v}} &\equiv \langle (V)^{\textcircled{v}}\rangle\mathtt{inl} & (\langle V\rangle\mathtt{inr})^{\textcircled{v}} &\equiv \langle (V)^{\textcircled{v}}\rangle\mathtt{inr} \\
(\langle V, W\rangle)^{\textcircled{v}} &\equiv \langle (V)^{\textcircled{v}}, (W)^{\textcircled{v}}\rangle & (\mathtt{not}\langle M\rangle)^{\textcircled{v}} &\equiv \mathtt{not}\langle (M)^{\textcircled{\eta}}\rangle
\end{aligned}$$

$$\left.\begin{aligned}
(\langle V, M\rangle)^{\textcircled{v}} &\equiv \big((M)^{\textcircled{\eta}} \bullet z.(\langle (V)^{\textcircled{v}}, z\rangle \bullet \beta)\big).\beta \\
(\langle M, N\rangle)^{\textcircled{v}} &\equiv \big((M)^{\textcircled{\eta}} \bullet y.\big(\big((N)^{\textcircled{\eta}} \bullet z.(\langle y, z\rangle \bullet \alpha)\big).\alpha \bullet \beta\big)\big).\beta \\
(\langle M\rangle\mathtt{inl})^{\textcircled{v}} &\equiv (\,(M)^{\textcircled{\eta}} \bullet y.(\,\langle y\rangle\mathtt{inl} \bullet \beta\,)\,).\beta \\
(\langle M\rangle\mathtt{inr})^{\textcircled{v}} &\equiv (\,(M)^{\textcircled{\eta}} \bullet z.(\,\langle z\rangle\mathtt{inr} \bullet \beta\,)\,).\beta
\end{aligned}\right\} M \text{ non-value}$$

For any $G \in$ DuCa, its translation in DuCa* is $(G)^{\textcircled{\eta}}$.

Note that we do not really translate types; G is typed by A iff $(G)^{\textcircled{\eta}}$ is.

Lemma 2.30. *For any* $G, K, V, \alpha, x \in$ DuCa,

$$(G)^{\textcircled{v}}\{(V)^{\textcircled{v}}/x\} \equiv (G\{V/x\})^{\textcircled{v}} \quad \textit{and} \quad (G)^{\textcircled{v}}\{(K)^{\textcircled{v}}/\alpha\} \equiv (G\{K/\alpha\})^{\textcircled{v}}$$

Proof. By straightforward induction on G. Observe that a term M is a value iff $M\{V/x\}$ and $M\{K/\alpha\}$ are, and that M is of the form $(N \bullet \beta).\beta$ iff $M\{V/x\}$ and $M\{K/\alpha\}$ are. Similar observations for coterms. □

The translation conveniently preserves reductions.

Proposition 2.31. *For any* $G, H \in$ DuCa,

$$\begin{aligned}
G \xrightarrow{\eta} H &\implies (G)^{\textcircled{\eta}} \equiv (H)^{\textcircled{\eta}} \\
G \xrightarrow{v} H &\implies (G)^{\textcircled{\eta}} \longrightarrow_{+} (H)^{\textcircled{\eta}} \\
G \xrightarrow{\beta L/\beta R} H &\implies (G)^{\textcircled{\eta}} \longrightarrow_{+} (H)^{\textcircled{\eta}}
\end{aligned}$$

Proof. For the first implication suppose the reduction is ηR. Then, for some non-abstraction term M, $G \equiv C_0\{M\} \longrightarrow C_0\{(M \bullet \alpha).\alpha\} \equiv H$. Because of the restrictions in $R_v^{\mathtt{SN}}$, $C_0\{M\}$ must be in one of the forms:

$$\begin{cases} M,\ C\{\langle M, N\rangle\},\ C\{\langle N, M\rangle\},\ C\{\langle M\rangle\mathtt{inl}\},\ C\{\langle M\rangle\mathtt{inr}\},\ C\{\mathtt{not}\langle M\rangle\} & \text{if } M \text{ non-value,} \\ V,\ C\{\langle N', V\rangle\},\ C\{\mathtt{not}\langle V\rangle\} & \text{if } M \text{ a value } V \end{cases}$$

with N term, N' non-value term and C context. Now observe that in every case there is some context C^* in DuCa* such that $(C_0\{M\})^{\textcircled{n}} \equiv C^*\{(M)^{\textcircled{n}}\}$ and $C^*\{((M \bullet \alpha).\alpha)^{\textcircled{n}}\} \equiv (C_0\{(M \bullet \alpha).\alpha\})^{\textcircled{n}}$. But, since M is non-abstraction, $(M)^{\textcircled{n}} \equiv ((M)^{\textcircled{v}} \bullet \alpha).\alpha \equiv ((M \bullet \alpha).\alpha)^{\textcircled{n}}$.

The case of the reduction being ηL is similar.

Now, suppose the reduction happens because of some statement S v-reducing to T: $G \equiv C\{S\} \xrightarrow{v} C\{T\} \equiv H$. Then, we can see that there exists a context C^* in DuCa* such that $(G)^{\textcircled{n}} \equiv C^*\{S^{\textcircled{v}}\}$ and $(H)^{\textcircled{n}} \equiv C^*\{T^{\textcircled{v}}\}$. Therefore, it suffices to show that $(S)^{\textcircled{v}} \longrightarrow_+ (T)^{\textcircled{v}}$. But this follows from the definition of $(\_)^{\textcircled{v}}$. For example,

$$\begin{aligned}
(\langle M, V\rangle \bullet K)^{\textcircled{v}} \equiv\ & \big((M)^{\textcircled{n}} \bullet y.\big(\big((V)^{\textcircled{n}} \bullet z.(\langle y, z\rangle \bullet \alpha)\big).\alpha \bullet \beta\big)\big).\beta \,\underline{\bullet}\, (K)^{\textcircled{v}} \\
\longrightarrow\ & (M)^{\textcircled{n}} \bullet y.\big(\big((V)^{\textcircled{n}} \bullet z.(\langle y, z\rangle \bullet \alpha)\big).\alpha \,\underline{\bullet}\, (K)^{\textcircled{v}}\big) \\
\longrightarrow\ & \underline{(M)^{\textcircled{n}}} \bullet y.\big(\underline{(V)^{\textcircled{n}}} \bullet z.(\langle y, z\rangle \bullet (K)^{\textcircled{v}})\big) \\
\longrightarrow_+ & (M)^{\textcircled{v}} \bullet y.\big((V)^{\textcircled{v}} \,\underline{\bullet}\, z.(\langle y, z\rangle \bullet (K)^{\textcircled{v}})\big) \\
\longrightarrow\ & (M)^{\textcircled{v}} \bullet y.\big(\langle y, (V)^{\textcircled{v}}\rangle \bullet (K)^{\textcircled{v}}\big) \equiv (M \bullet y.(\langle y, V\rangle \bullet K))^{\textcircled{n}}
\end{aligned}$$

Also, for N non-value we have,

$$\begin{aligned}
(\langle M, N\rangle \bullet K)^{\textcircled{v}} \equiv\ & \big((M)^{\textcircled{n}} \bullet y.\big(\big((N)^{\textcircled{n}} \bullet z.(\langle y, z\rangle \bullet \alpha)\big).\alpha \bullet \beta\big)\big).\beta \,\underline{\bullet}\, (K)^{\textcircled{v}} \\
\longrightarrow\ & \underline{(M)^{\textcircled{n}}} \bullet y.\big(\big((N)^{\textcircled{n}} \bullet z.(\langle y, z\rangle \bullet \alpha)\big).\alpha \bullet (K)^{\textcircled{v}}\big) \\
\longrightarrow_= & (M)^{\textcircled{v}} \bullet y.\big(\big((N)^{\textcircled{n}} \bullet z.(\langle y, z\rangle \bullet \alpha)\big).\alpha \bullet (K)^{\textcircled{v}}\big) \equiv (M \bullet y.(\langle y, N\rangle \bullet K))^{\textcircled{n}}
\end{aligned}$$

The cases for the other v-rules are proven similarly.

Finally, suppose the reduction happens because of some statement S βLR-reducing to T: $G \equiv C\{S\} \xrightarrow{\beta L/\beta R} C\{T\} \equiv H$. Then, $(G)^{\textcircled{n}} \equiv C^*\{S^{\textcircled{v}}\}$ and $(H)^{\textcircled{n}} \equiv C^*\{T^{\textcircled{v}}\}$, for some context C^*. Because of the previous lemma $(S)^{\textcircled{v}} \longrightarrow_+ (T)^{\textcircled{v}}$, hence $(G)^{\textcircled{n}} \longrightarrow_+ (H)^{\textcircled{n}}$. $\square$

2.2.3. *The call-by-value CPS translation*

CPS translations are very useful when examining extensions of lambda calculi, because they supply us with a way of projecting given properties of the source calculus to a well-behaved lambda calculus. In our case, using a CPS translation of the Dual Calculus enables us to prove that the reduction relation yielded by some β-reduction rules is strongly normalizing, since these reduction rules are projected in the target calculus.

The CBV CPS translation of the Dual Calculus we will be using was defined in [20]. We quote that definition.

Definition 2.32 (*CBV CPS translation, Walder [20]*). The CBV CPS translation is defined for types and objects of the DuCa as follows:

$$\begin{array}{llll}
(X)^V & \equiv X & (x)^V & \equiv x \\
(A\&B)^V & \equiv (A)^V \times (B)^V & (\langle V, W\rangle)^V & \equiv \langle (V)^V, (W)^V\rangle \\
(A \vee B)^V & \equiv (A)^V + (B)^V & (\langle V\rangle \mathtt{inl})^V & \equiv \mathtt{inl}(V)^V \\
(\neg A)^V & \equiv (A)^V \to R & (\langle W\rangle \mathtt{inr})^V & \equiv \mathtt{inr}(W)^V \\
 & & ([K]\mathtt{not})^V & \equiv (K)^v
\end{array}$$

$$(M \bullet K)^v \equiv (M)^v (K)^v$$

$$\begin{array}{ll}
(x)^v & \equiv \lambda\gamma.\gamma x \\
(\langle M, N\rangle)^v & \equiv \lambda\gamma.(M)^v(\lambda x.(N)^v(\lambda y.\gamma\langle x, y\rangle)) \\
(\langle M\rangle \mathtt{inl})^v & \equiv \lambda\gamma.(M)^v(\lambda x.\gamma(\mathtt{inl} x)) \\
(\langle N\rangle \mathtt{inr})^v & \equiv \lambda\gamma.(N)^v(\lambda y.\gamma(\mathtt{inr} y)) \\
([K]\mathtt{not})^v & \equiv \lambda\gamma.\gamma(\lambda z.(K)^v z) \\
((S).\alpha)^v & \equiv \lambda\alpha.(S)^v
\end{array}$$

$$\begin{array}{ll}
(\alpha)^v & \equiv \boldsymbol{\lambda} z.\alpha z \\
([K,L])^v & \equiv \boldsymbol{\lambda} z.\texttt{case}\ z\ \texttt{of}\ \texttt{inl}x \Rightarrow (K)^v x,\ \texttt{inr}y \Rightarrow (L)^v y \\
(\texttt{fst}[K])^v & \equiv \boldsymbol{\lambda} z.\texttt{case}\ z\ \texttt{of}\ \langle x, -\rangle \Rightarrow (K)^v x \\
(\texttt{snd}[L])^v & \equiv \boldsymbol{\lambda} z.\texttt{case}\ z\ \texttt{of}\ \langle -, y\rangle \Rightarrow (L)^v y \\
(\texttt{not}\langle M\rangle)^v & \equiv \boldsymbol{\lambda} z.(\lambda \gamma.(M)^v \gamma) z \\
(x.(S))^v & \equiv \boldsymbol{\lambda} x.(S)^v
\end{array}$$

Note that boldface lambda-abstractions are administrative, that is, they are reduced automatically on translation.

As follows from the following definition, the target calculus is a restriction of the simply-typed lambda calculus with products and sums.

Definition 2.33 (*The target calculus, Walder [20]*). The CPS target calculus is a typed calculus containing values, terms, coterms and statements:

$$\begin{array}{lll}
\text{Type} & A, B & ::= X \mid A \times B \mid A + B \mid A \to R \\
\\
\text{Value} & V, W & ::= x \mid \langle V, W\rangle \mid \texttt{inl}V \mid \texttt{inr}W \mid K \\
\text{Term} & M, N & ::= \lambda\alpha.S \\
\text{Coterm} & K, L & ::= \lambda x.S \\
\text{Statement} & S, T & ::= \alpha V \mid \texttt{case}\ V\ \texttt{of}\ \langle x, -\rangle \Rightarrow S \mid \texttt{case}\ V\ \texttt{of}\ \langle -, y\rangle \Rightarrow T \mid \\
 & & \quad\ \texttt{case}\ V\ \texttt{of}\ \texttt{inl}x \Rightarrow S,\ \texttt{inr}y \Rightarrow T \mid M\,V
\end{array}$$

The reduction relation is defined by the following reduction rules.

$$\begin{array}{lll}
(\beta\times_1) & \texttt{case}\ \langle V, W\rangle\ \texttt{of}\ \langle x, -\rangle \Rightarrow S & \to S\{V/x\} \\
(\beta\times_2) & \texttt{case}\ \langle V, W\rangle\ \texttt{of}\ \langle -, y\rangle \Rightarrow T & \to T\{W/y\} \\
(\beta+_1) & \texttt{case}\ \texttt{inl}V\ \texttt{of}\ \texttt{inl}x \Rightarrow S,\ \texttt{inr}y \Rightarrow T & \to S\{V/x\} \\
(\beta+_2) & \texttt{case}\ \texttt{inr}W\ \texttt{of}\ \texttt{inl}x \Rightarrow S,\ \texttt{inr}y \Rightarrow T & \to T\{W/y\} \\
(\beta\to) & (\lambda\alpha.S)(\lambda x.T) & \to S\{T\{-/x\}/\alpha -\}
\end{array}$$

Note that in the above definition $S\{T\{-/x\}/\alpha-\}$ stands for S with all occurrences of the form $\alpha\, V$ replaced by $T\{V/x\}$.

Calculi of this type have been investigated in depth and many nice properties are known to hold. One such property is strong normalization.

Proposition 2.34. *The target calculus of the CBV CPS translation is* SN *under the given reduction relation.*

Proof. It suffices to show that the target calculus is a restriction of the lambda calculus with sums and products of Dougherty [5], since the latter was shown to be SN.

But this clearly holds. For example, $\texttt{case}\ V\ \texttt{of}\ \langle x, -\rangle \Rightarrow S$ is an abbreviation for $(\lambda x.S)\pi_1 V$ and $\texttt{case}\ V\ \texttt{of}\ \texttt{inl}x \Rightarrow S,\ \texttt{inr}y \Rightarrow T$ is an abbreviation for $[\lambda x.S, \lambda y.T]V$. Moreover, the reductions of the target calculus are valid in the calculus of Dougherty. □

A very handy property of the CPS translation is that it preserves R_v-reductions.

Proposition 2.35 (*Walder [20]*). *Let M, N, K, L, S, T be in the Dual Calculus. Then,*

$$\begin{array}{ll}
M \longrightarrow^v N & \Longrightarrow (M)^v \twoheadrightarrow (N)^v \\
K \longrightarrow^v L & \Longrightarrow (K)^v \twoheadrightarrow (L)^v \\
S \longrightarrow^v T & \Longrightarrow (S)^v \twoheadrightarrow (T)^v
\end{array}$$

In particular, if the left-hand side reduction is of type $\beta L, \beta R, \eta$ or ν, then, in the right-hand side, $\twoheadrightarrow$ can be replaced by $\equiv$. Otherwise, it can be replaced by $\longrightarrow_+$.

Proof. The proposition is proven in [20]. The last part of it is not completely stated in that paper, yet it is straightforward from the definition of the CPS translation and the results that proceed this proposition in [20]. □

Wadler goes further by defining an inverse CPS translation, which translates elements M of the target calculus to objects $(M)_v$ of the Dual Calculus. Finally, he proves that the CPS translation is a *reflection*.

2.2.4. Strong normalization of call-by-value reduction in DuCa

Using the results of the previous sections, we show that the CBV reduction relation $R_v^{\mathtt{SN}}$ is strongly normalizing in DuCa. The SN result is the following.

Theorem 2.36. *For any $G \in$ DuCa, there is no infinite $R_v^{\mathtt{SN}}$-reduction sequence starting from G.*

Proof. Let $G \in$ DuCa and suppose that there is some infinite $R_v^{\mathtt{SN}}$-reduction sequence starting from G: say $G \longrightarrow G_1 \longrightarrow G_2 \longrightarrow \cdots$. Then, by Proposition 2.35, there is a sequence:

$$(G)^v \twoheadrightarrow (G_1)^v \twoheadrightarrow (G_2)^v \twoheadrightarrow \cdots$$

in the target calculus of the CPS translation. By Proposition 2.34, the target calculus is SN, so there is some last element in the sequence, say M_n. Therefore, there is some $1 \leqslant m \leqslant n$ and some i_m such that, for all $i \geqslant i_m$, $(G_i)^v \equiv M_m$. But then, by Proposition 2.35, in the sequence

$$G_{i_m} \longrightarrow G_{i_m+1} \longrightarrow G_{i_m+2} \longrightarrow \cdots$$

all reductions are instances of βL, βR, η or ν. Now, by Proposition 2.31, this produces a sequence

$$(G_{i_m})^{\textcircled{\eta}} \twoheadrightarrow (G_{i_m+1})^{\textcircled{\eta}} \twoheadrightarrow (G_{i_m+2})^{\textcircled{\eta}} \twoheadrightarrow \cdots$$

in DuCa*. By Corollary 2.28, DuCa* is SN, so there is some last element in the sequence, say G_l. Therefore, there is some $1 \leqslant k \leqslant l$ and some j_k such that, for all $j \geqslant j_k$, $(G_j)^{\textcircled{\eta}} \equiv G_k$. But then, by Proposition 2.31, in the sequence

$$G_{j_k} \longrightarrow G_{j_k+1} \longrightarrow G_{j_k+2} \longrightarrow \cdots$$

all reductions are instances of ηL or ηR. This is a contradiction to the fact that, under $R_v^{\mathtt{SN}}$, every η-reduction reduces by one the η-redices of the object it is applied to. □

2.3. A call-by-value reduction to satisfy both SN *and* CR

In this section we present a restricted version of the CBV reduction relation in DuCa which satisfies both CR and SN. The version of the reduction relation we use is $R_v^{\mathtt{SN}}$ of the previous section with some more restrictions on the η-rules. The added restrictions cope with separations of the following form. For M a non-value term,

$$M \bullet K \;\longleftarrow\; [K]\mathtt{not} \bullet \mathtt{not}\langle M\rangle \;\longrightarrow\; [x.(x \bullet K)]\mathtt{not} \bullet \mathtt{not}\langle M\rangle \;\longrightarrow\; M \bullet x.(x \bullet K)$$

The reduction relation with the new restrictions is called $R_v^{\mathtt{SN}\prime}$.

Definition 2.37. The reduction rules of $R_v^{\mathtt{SN}\prime}$ are those of $R_v^{\mathtt{SN}}$ (Definition 2.13), while the restrictions of $R_v^{\mathtt{SN}\prime}$ are those of $R_v^{\mathtt{SN}}$ with the addition of:
Rη''. η-rules are not allowed to be applied to coterms that are immediate subjects to [_]not.

Note that values followed by a cut cannot be reduced to non-values using $R_v^{\mathtt{SN}\prime}$.

It is clear that, since $R_v^{\mathtt{SN}\prime}$ is a restriction of $R_v^{\mathtt{SN}}$ and the latter is SN, $R_v^{\mathtt{SN}\prime}$ is SN. Therefore, we need only to show satisfaction of the CR property, which is much easier with SN at hand. Indeed, it suffices to prove WCR (Weak CR), because of the known result (see [2]):

$$\mathtt{SN} \wedge \mathtt{WCR} \Rightarrow \mathtt{CR} \tag{3}$$

Below, βv-reduction relation is $R_v^{\mathtt{SN}'}$ restricted to β- and v-rules, and η-reduction relation is $R_v^{\mathtt{SN}'}$ restricted to η-rules.

Lemma 2.38. *The βv- and η-reduction relations are* WCR.

Proof. For both cases we do a case analysis on some $G_1 \longleftarrow G \longrightarrow G_2$. The proofs are straightforward, as restrictions on v- and η-rules do not allow for non-trivial cases. □

We now prove WCR for $R_v^{\mathtt{SN}'}$.

Lemma 2.39. $R_v^{\mathtt{SN}'}$ *satisfies the* WCR; *that is, for all $G, G_1, G_2 \in$* DuCa, *if $G_1 \longleftarrow G \longrightarrow G_2$, then there exists some $G_c \in$* DuCa *such that $G_1 \longrightarrow\!\!\!\!\!\rightarrow G_c \longleftarrow\!\!\!\!\!\leftarrow G_2$.*

Proof. The proof is by a case analysis on $G \longrightarrow G_1$ and the possible combinations for $G \longrightarrow G_2$. By Lemma 2.38, we can omit the cases of both reductions being βv or both being η. Therefore, by symmetry, we may assume that $G \longrightarrow G_1$ is a βv-reduction and $G \longrightarrow G_2$ an η-reduction. In the following diagrams we do the case analysis on $G \longrightarrow G_1$, which is always the topmost horizontal reduction. C is some context and note that we have omitted the trivial cases of $G \longrightarrow G_2$ affecting solely C and not its content.

For $G \longrightarrow G_1$ being $\beta\&_1$:

$$\begin{tikzcd}
C\{\langle V,W\rangle \bullet \mathtt{fst}[K]\} \arrow[r, "\beta"] \arrow[d, "\eta"] \arrow[dd, bend right=60, "\eta"'] & C\{V \bullet K\} \arrow[d, "\eta", "="'] \\
C\{\langle V',W'\rangle \bullet \mathtt{fst}[K']\} \arrow[r, "\beta"] & C\{V' \bullet K'\} \\
C\{\langle V,W\rangle \bullet \mathtt{fst}[x.(x \bullet K)]\} \arrow[r, "\beta"] & C\{V \bullet x.(x \bullet K)\} \arrow[uu, bend right=60, "\beta"']
\end{tikzcd}$$

The cases of $\beta\&_2$, $\beta\vee_1$, $\beta\vee_2$ and $\beta\neg$ are similar.

For $G \longrightarrow G_1$ being βR we have:

$$\begin{tikzcd}
& C\{(S).\alpha \bullet K\} \arrow[r, "\beta"] \arrow[dl, "\eta"] \arrow[dr, "\eta"] & C\{S\{K/\alpha\}\} \arrow[dl, "\eta"] \arrow[dr, "\eta"] & \\
C\{(S).\alpha \bullet K'\} \arrow[r, "\beta"] & C\{S\{K'/\alpha\}\} & C\{(S').\alpha \bullet K\} \arrow[r, "\beta"] & C\{S'\{K/\alpha\}\}
\end{tikzcd}$$

and also the particular case when $S\{K/\alpha\}$ cannot reduce to $S'\{K/\alpha\}$, which occurs when

$$\begin{tikzcd}
C\{\ (C_0\{\alpha\}).\alpha \bullet y.(S)\ \} \arrow[r, "\beta"] \arrow[d, "\eta"] & C\{\ (C_0\{\alpha\})\{y.(S)/\alpha\}\ \} \arrow[d, shift right, "\times" marking] \\
C\{\ (C_0\{x.(x \bullet \alpha)\}).\alpha \bullet y.(S)\ \} \arrow[r, "\beta"] & C\{\ (C_0\{x.(x \bullet \alpha)\})\{y.(S)/\alpha\}\ \} \arrow[u, shift right, "\beta"']
\end{tikzcd}$$

where $C_0\{\alpha\}$ is some statement S_0, and we use the fact that, by alpha-conversion, we have

$$x.(x \bullet y.(S)) \longrightarrow x.(S\{x/y\}) \equiv y.(S)$$

The case of βL is similar and simpler.

For $G \longrightarrow G_1$ being $\nu\&_1$:

$$\begin{tikzcd}
C\{\langle M,N\rangle \bullet K\} \arrow[r, "\nu"] \arrow[d, "\eta"] \arrow[dd, bend right, "\eta"'] \arrow[ddd, bend right=60, "\eta"'] & C\{M \bullet x.(\langle x,N\rangle \bullet K)\} \arrow[d, "\eta"] \arrow[ddd, bend left=50, "\nu", "N\text{ non-value}"'] \\
C\{\langle M',N'\rangle \bullet K'\} \arrow[r, "\nu"] & C\{M' \bullet x.(\langle x,N'\rangle \bullet K')\} \\
C\{\langle (M \bullet \alpha).\alpha, N\rangle \bullet K\} \arrow[r, "\nu"] & C\{(M \bullet \alpha).\alpha \bullet x.(\langle x,N\rangle \bullet K)\} \arrow[uu, bend right, "\beta"'] \\
C\{\langle M,(N \bullet \alpha).\alpha\rangle \bullet K\} \arrow[r, "\nu"] & C\{M \bullet x.(\langle x,(N \bullet \alpha).\alpha\rangle \bullet K)\} \arrow[dl, "\nu"', "N\text{ value}"] \\
C\{M \bullet x.((N \bullet \alpha).\alpha \bullet y.(\langle x,y\rangle \bullet K))\} \arrow[r, "\beta"] & C\{M \bullet x.(N \bullet y.(\langle x,y\rangle \bullet K))\} \arrow[uuuu, bend right=70, "\beta"']
\end{tikzcd}$$

The cases of $\nu\&_2$, $\nu\vee_1$ and $\nu\vee_2$ are similar and simpler. □

We conclude with the main result of this subsection.

Theorem 2.40. $R_v^{\mathtt{SN}\prime}$ *is both* SN *and* CR.

Proof. Since $R_v^{\mathtt{SN}\prime}$ is a restriction of $R_v^{\mathtt{SN}}$, every reduction sequence of the former is also a reduction sequence of the latter. By Theorem 2.36, $R_v^{\mathtt{SN}}$ is SN, therefore $R_v^{\mathtt{SN}\prime}$ is SN.

Furthermore, by (3) and Lemma 2.39, $R_v^{\mathtt{SN}\prime}$ is CR. □

3. A glimpse into the second-order case

Girard proposed an extension of the simply-typed lambda calculus, called polymorphic lambda calculus (system **F** [11,17], or $\lambda 2$ [3]), which is isomorphic to second-order propositional intuitionistic logic in Curry–Howard style. The extension from the simply-typed lambda calculus to **F** is a very strong one, with regard to the functions that we can represent in each calculus. In [8] it is shown that the functions which are representable in the simply-typed lambda calculus form a proper subset of the elementary functions.[8] This is indeed a very "small" class of functions. On the other hand, in [11] it is shown that the functions representable in **F** are exactly those which are provably total[9] in second-order Peano Arithmetic. This is a substantially "larger" class of functions.

Consequently, it is interesting and natural to study second-order extensions for the Dual Calculus. We briefly introduce two different such extensions.

3.1. The natural extension

We extend the Dual Calculus to second-order propositional classical logic by adding typing rules for second-order quantifiers over types.

Definition 3.1 (DuCa2). DuCa2 consists of Types and Objects. The set of objects is the union of the sets of Terms, Coterms and Statements:

Type	A, B	$::=$	$X \mid A\&B \mid A \vee B \mid \neg A \mid \forall X.A \mid \exists X.A$
Object	G, H	$::=$	$M \mid K \mid S$
Term	M, N	$::=$	$x \mid \langle M, N\rangle \mid \langle M\rangle\mathtt{inl} \mid \langle N\rangle\mathtt{inr} \mid [K]\mathtt{not} \mid (S).\alpha \mid \mathsf{a}\cdot M \mid \mathsf{e}\cdot M$
Coterm	K, L	$::=$	$\alpha \mid [K, L] \mid \mathtt{fst}[K] \mid \mathtt{snd}[L] \mid \mathtt{not}\langle M\rangle \mid x.(S) \mid \mathsf{a}\cdot K \mid \mathsf{e}\cdot K$
Statement	S, T	$::=$	$M \bullet K$

[8] The class of elementary functions is the smallest class of functions which contains projections, successor, $+$, $\dot{-}$ and $\times$, and is closed under composition and bounded sums and products (note $x \dot{-} y$ is $x - y$ for $x \geqslant y$, otherwise 0).

[9] A function f is provably total in a theory T, if there is an algorithm A computing f for which T proves that A terminates on all inputs.

The typing rules are the same as those of DuCa (i.e. of GW) plus the second-order rules:

$$\frac{K : A\{B/X\} \,\|\, \Gamma \to \Theta}{\mathsf{a}\cdot K : \forall X.A \,\|\, \Gamma \to \Theta}\,\forall L \qquad \frac{\Gamma \to \Theta \,\|\, M : A}{\Gamma \to \Theta \,\|\, \mathsf{a}\cdot M : \forall X.A}\,\forall R$$

$$\frac{K : A \,\|\, \Gamma \to \Theta}{\mathsf{e}\cdot K : \exists X.A \,\|\, \Gamma \to \Theta}\,\exists L \qquad \frac{\Gamma \to \Theta \,\|\, M : A\{B/X\}}{\Gamma \to \Theta \,\|\, \mathsf{e}\cdot M : \exists X.A}\,\exists R$$

where in $\forall R$ and $\exists L$ we have $X \notin FTV(\Gamma, \Theta)$. The typing rules form the sequent calculus GW2.

The basic reduction relation is $R2_b$, which extends R_b by the rules:

(βa) $\mathsf{a}\cdot M \bullet \mathsf{a}\cdot K \to M \bullet K$
(βe) $\mathsf{e}\cdot M \bullet \mathsf{e}\cdot K \to M \bullet K$

For $G, H \in$ DuCa2, we denote $(G, H) \in R2_b$ by $G \longrightarrow H$.

This extension of DuCa to second-order follows Parigot's extension of the Sλ-calculus to second-order introduced in [14]. Its characteristic feature is the use of "trivial witnesses" at the level of objects, which distinguish objects with types quantified by introduction rules from those with types quantified by axioms. We choose this presentation, instead of Church-style passing of type-variables in terms, for simplicity.

In fact, the use of such witnesses is essential for subject reduction to hold, since without witnesses the following derivation would be legal.

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \,\|\, S \mapsto \Theta, \alpha : A}}{\dfrac{\Gamma \to \Theta \,\|\, (S).\alpha : A}{\Gamma \to \Theta \,\|\, (S).\alpha : \forall X.A}} \qquad \dfrac{\dfrac{\vdots}{x : \forall X.A, \Gamma \,\|\, S' \mapsto \Theta}}{x.(S') : \forall X.A \,\|\, \Gamma \to \Theta}}{\Gamma \,\|\, (S).\alpha \bullet x.(S') \mapsto \Theta}$$

Now, $(S).\alpha \bullet x.(S') \longrightarrow S\{x.(S')/\alpha\}$, yet we cannot always type $S\{x.(S')/\alpha\}$. [10]

Since subject reduction is problematic for the extension without witnesses, it is useful to show it for DuCa2. First we show the following lemmata. [11]

Lemma 3.2 (*Type substitution*). *Take any $G \in$ DuCa2. If there exists some sequent $\sigma \in T_G(A, \Gamma, \Theta)$, derived say by a derivation $\mathcal{D}$, then for any type B there exists some sequent $\sigma' \in T_G(A\{B/X\}, \Gamma\{B/X\}, \Theta\{B/X\})$ derived by a derivation $\mathcal{D}'$, such that $\mathcal{D}$ and $\mathcal{D}'$ have the same tree structure.*

Proof. In the Appendix.

Lemma 3.3 (*Weakening, M for x*). *Let $G, M, K \in$ DuCa2,*
1. *If $T_G(A, \Gamma, \Theta) \neq \emptyset$ then $T_G(A, \Gamma \uplus (x : B), \Theta) \neq \emptyset$ and $T_G(A, \Gamma, \Theta \uplus (\alpha : B)) \neq \emptyset$, any fresh variable x and fresh covariable α.*
2. *If $T_G(A, \Gamma \uplus (x : B), \Theta) \neq \emptyset$ and $T_M(B, \Gamma, \Theta) \neq \emptyset$ then $T_{G\{M/x\}}(A, \Gamma, \Theta) \neq \emptyset$, and if $T_G(A, \Gamma, \Theta \uplus (\alpha : B)) \neq \emptyset$ and $T_K(B, \Gamma, \Theta) \neq \emptyset$ then $T_{G\{K/\alpha\}}(A, \Gamma, \Theta) \neq \emptyset$.*

Proof. For 1 we do induction on the derivation of $\sigma \in T_G(A, \Gamma, \Theta)$ and case analysis on the last rule in it, using Lemma 3.2 to deal with the $\forall R$, $\exists L$ rules where B may contain the quantified type variable. For 2 we do similar induction. □

Proposition 3.4 (*Subject reduction*). *Suppose that $G, H \in$ DuCa2 and that $G \longrightarrow H$. If $T_G(A_0, \Gamma_0, \Theta_0) \neq \emptyset$, some A_0, Γ_0, Θ_0, then $T_H(A_0, \Gamma_0, \Theta_0) \neq \emptyset$.*

[10] In fact, there are also other cases in which Subject Reduction fails if we don't use witnessess, but they can be remedied by use of Strong Contraction rules (see [22] for details).

[11] Recall the notation introduced in Definition 2.17 for sequents typing objects of DuCa*. Here we use the analogs for DuCa2.

Proof. In the Appendix.

Because of the similarities between DuCa2 and the second-order Sλ-calculus, it is natural to consider translations between the two calculi. Although we will not examine these in full detail here, we note the following.

Note 3.5. Translations $(\_)^o$ (from DuCa to Sλ, Definition 1.10) and $(\_)^p$ (from Sλ to DuCa, Definition 1.15) readily extend to second order. Moreover, it is not difficult to show that the former translation preserves typing judgements and reductions. The same does not hold for the latter, as typing judgements are not preserved. This is because of the "substitution lemma" failing on types: $(A\{B/X\})^p \not\equiv (A)^p\{(B)^p/X\}$, e.g.

$$\neg Y \vee \neg Z \equiv (Y^\perp \vee Z^\perp)^p \equiv (X^\perp\{Y\&Z/X\})^p \not\equiv (X^\perp)^p\{(Y\&Z)^p/X\} \equiv \neg(Y\&Z)$$

As a result, typing judgements for existential quantification in second-order Sλ are not preserved in DuCa2 under $(\_)^p$.

3.2. Simulation of second-order quantifiers

In second-order classical propositional logic the quantification over propositional variables is in fact a quantification over true and false propositions. That is, if $\perp$ is some contradiction (for example, $\perp \equiv X_0 \,\&\, \neg X_0$) then, for any formula A, $\forall X.A$ is logically equivalent to $A\{\perp/X\} \,\&\, A\{\neg\perp/X\}$. A similar property holds for existential quantification, so quantifiers can be simulated in the logic.[12]

Below we are going to define a construction for quantification over types by using the above remark explicitly, so universal quantification will be the abbreviation of a conjunction and existential quantification the abbreviation of a disjunction. Therefore, universal types will be assigned to product syntactic constructs, while existential types to sum constructs. For this purpose, some new construction rules for terms and coterms will be defined, so as to capture quantification in the cases where the existing construction rules are not enough.

The resulting calculus is non-polymorphic: in the logic we can prove $A\{B/X\}$ given $A\{\top/X\}$ and $A\{\perp/X\}$, yet in the syntax we cannot always derive, for example, $M : A\{B/X\}$ given derivations for $M : A\{\top/X\}$ and $M : A\{\perp/X\}$. We will return to this in the end of this section. We begin with some definitions.

Definition 3.6. Let F be a type constant. Define the set of Types:

$$\text{Type} \quad A, B ::= \mathsf{F} \mid X \mid A\&B \mid A \vee B \mid \neg A$$

We define the following abbreviations, for any type A and $X \in TVar$,

$$\forall X.A \equiv A\{\top/X\} \,\&\, A\{\perp/X\}, \quad \exists X.A \equiv A\{\top/X\} \vee A\{\perp/X\}$$

and

$$\top \equiv \mathsf{F} \vee \neg\mathsf{F}, \quad \perp \equiv \mathsf{F}\&\neg\mathsf{F}$$

Defining $\top, \perp$ in terms of another constant F we get the properties of $\top, \perp$ for free, using the properties of $\&, \vee, \neg$. Note that under the above abbreviations alpha-equivalence is valid, that is, for all types A and $X, Y \in TVar$ with $Y \notin FTV(A)$, $\forall X.A \equiv \forall Y.A\{Y/X\}$ and $\exists X.A \equiv \exists Y.A\{Y/X\}$.

DuCa2$^\mathtt{C}$ is the resulting second-order calculus.

Definition 3.7. DuCa2$^\mathtt{C}$ consists of Types as in Definition 3.6 and Objects. The set of objects is the union of the sets of Terms, Coterms and Statements:

$$\begin{array}{lll} \text{Object} & G, H & ::= M \mid K \mid S \\ \text{Term} & M, N & ::= x \mid \langle M, N\rangle \mid \langle M\rangle\mathtt{inl} \mid \langle N\rangle\mathtt{inr} \mid [K]\mathtt{not} \mid (S).\alpha \mid \langle M\rangle\mathtt{in} \\ \text{Coterm} & K, L & ::= \alpha \mid [K, L] \mid \mathtt{fst}[K] \mid \mathtt{snd}[L] \mid \mathtt{not}\langle M\rangle \mid x.(S) \mid \mathtt{one}[K] \\ \text{Statement} & S, T & ::= M \bullet K \end{array}$$

[12] In [15] it is shown that in intuitionistic propositional logic second-order quantifiers can also be modelled in first-order constructions. That result is far deeper than its classical counterpart.

The typing rules are the same as those of DuCa (i.e. of system GW) with the addition of

$$\frac{K : A\{B/X\} \mid \Gamma \to \Theta}{\mathtt{one}[K] : \forall X.A \mid \Gamma \to \Theta}\,\forall L \qquad \frac{\Gamma \to \Theta \mid M : A\{B/X\}}{\Gamma \to \Theta \mid \langle M\rangle\mathtt{in} : \exists X.A}\,\exists R$$

The typing rules form the sequent calculus $\mathtt{GW2^C}$.

The basic reduction relation is $R2^{\mathtt{C}}_b$, which extends trivially R_b to the syntax of $\mathtt{DuCa2^C}$.

The intuition behind the new construction rules is that $\langle M\rangle\mathtt{in}$ produces a sum value built up from M being *either its first or its second element*, whereas $\mathtt{one}[K]$ absorbs a product value and offers *one of its elements* to K.

Now, suppose we can derive $\Gamma \to \Theta \mid M : A$ and $X \notin FTV(\Gamma, \Theta)$. Then it is not difficult to see that we can also derive $\Gamma \to \Theta \mid M : A\{\top/X\}$ and $\Gamma \to \Theta \mid M : A\{\bot/X\}$ and thus derive $\Gamma \to \Theta \mid \langle M, M\rangle : A\{\top/X\}\&A\{\bot/X\}$. Therefore, a proof of $\forall X.A$ is a construction merging together a proof of $A\{\top/X\}$ and one of $A\{\bot/X\}$. In fact, this is the analog of the Brouwer–Heyting–Kolmogorov interpretation of the proof of $\forall X.A$ in intuitionistic logic. Similar remarks can be made for the usual $\exists L$ rule.

Let us now formulate the above remark formally.

Proposition 3.8. *The following are derived rules of* $\mathtt{DuCa2^C}$:

$$\frac{K : A \mid \Gamma \to \Theta}{[K, K] : \exists X.A \mid \Gamma \to \Theta}\,\exists L \qquad \frac{\Gamma \to \Theta \mid M : A}{\Gamma \to \Theta \mid \langle M, M\rangle : \forall X.A}\,\forall R$$

where $X \notin FTV(\Gamma, \Theta)$.

Proof. First prove Weakening and Type Substitution for the calculus, from which the proposition follows straightforwardly □

It is easy to show that subject reduction holds under $R2^{\mathtt{C}}_b$ for $\mathtt{DuCa2^C}$.

Proposition 3.9 (*Subject reduction*). *Suppose that* $G, H \in \mathtt{DuCa2^C}$ *and that* $G \longrightarrow H$. *If* $T_G(A_0, \Gamma_0, \Theta_0) \neq \emptyset$, *some* A_0, Γ_0, Θ_0, *then* $T_H(A_0, \Gamma_0, \Theta_0) \neq \emptyset$.

Proof. The proof is by a case analysis on the rule used for the reduction and is similar (and simpler) to the proof of Proposition 3.4. □

Thus, we have introduced an extension of the dual calculus that corresponds to second-order classical propositional logic and is well-behaved in that it satisfies subject reduction. As we mentioned above, the calculus is not polymorphic. For example, abstractions of universal type can be derived only in the form $\langle (S).\alpha, (S).\alpha\rangle$, as below.

$$\frac{\dfrac{\Gamma \to \Theta \mid (S).\alpha : A}{\Gamma \to \Theta \mid \langle (S).\alpha, (S).\alpha\rangle : \forall X.A}\,X \notin FTV(\Gamma, \Theta) \qquad \dfrac{K : A\{B/X\} \mid \Gamma \to \Theta}{\mathtt{one}[K] : \forall X.A \mid \Gamma \to \Theta}}{\Gamma \mid \langle (S).\alpha, (S).\alpha\rangle \bullet \mathtt{one}[K] \mapsto \Theta}$$

However, $(S).\alpha$ cannot be applied on K, as polymorphism would demand.

A possible solution would be to add some obvious reduction rules in $R2^{\mathtt{C}}_b$ to enable this:

$$\begin{array}{lll} (\beta\&_c) & \langle M, M\rangle \bullet \mathtt{one}[K] & \to M \bullet K \\ (\beta\vee_c) & \langle M\rangle\mathtt{in} \bullet [K, K] & \to M \bullet K \end{array}$$

But now subject reduction breaks down; for example, $x : \bot \mid \langle\langle x\rangle\mathtt{in}, \langle x\rangle\mathtt{in}\rangle \bullet \mathtt{one}[\alpha] \mapsto \alpha : B \vee \bot$ is derivable for any type B, yet $x : \bot \mid \langle x\rangle\mathtt{in} \bullet \alpha \mapsto \alpha : B \vee \bot$ is derivable only if $B \equiv \top$ or $B \equiv \bot$.

4. Conclusion

In this paper we examined an extension of the lambda calculus, the Dual Calculus of Wadler. The calculus has two very interesting properties, namely that it corresponds to classical propositional logic under Curry–Howard isomorphism, and that its CBV and CBN reduction relations are De Morgan duals. We studied two basic syntactic properties relative to CBV reduction in the dual calculus: Church–Rosser property for the untyped calculus, and Strong Normalization for the typed calculus. Finally, we briefly introduced two extensions of the calculus to second-order quantified types, which correspond to second-order classical propositional logic.

The paper leaves space for a deeper investigation on the second-order calculus, in particular, on its Church–Rosser and Strong Normalization properties. On the other hand, an aspect we did not examine at all is that of denotational semantics for the dual calculus. It would be very interesting to study such semantics given the "classical" orientation of the calculus.

Acknowledgments

I would like to thank Prof. Samson Abramsky for his encouragement and guidance. Moreover, I would like to thank Prof. Phil Wadler and the anonymous reviewer for their constructive comments. I also acknowledge financial support by the Eugenides Foundation in Athens.

Appendix A. Some proofs from Sections 2 and 3

Proof of Lemma 2.5. The proof is by induction on $G \in \mathtt{DuCa}$ and case analysis on the two reductions. The pid cases are trivial. We examine the other cases. Note that we may use the previous lemma on substitution without mentioning it.

The cases for $G \longrightarrow_{\mathrm{p}} G_1$ are:

➛ $\mathrm{p}\beta\&_1$: then $G \equiv \langle V, W\rangle \bullet \mathtt{fst}[K]$ and $G_1 \equiv V' \bullet K'$; for some $V \longrightarrow_{\mathrm{p}} V'$, $W \longrightarrow_{\mathrm{p}} W'$,$K \longrightarrow_{\mathrm{p}} K'$.
By inspection of the other rules we have the following choices for the rule in $G \longrightarrow_{\mathrm{p}} G_2$:

- $\mathrm{p}\beta\&_1$: in this case $G_2 \equiv V'' \bullet K''$; for some $V \longrightarrow_{\mathrm{p}} V''$, $W \longrightarrow_{\mathrm{p}} W''$, $K \longrightarrow_{\mathrm{p}} K''$. By IH, there exist V_c, K_c such that $V', V'' \longrightarrow_{\mathrm{p}} V_c$ and $K', K'' \longrightarrow_{\mathrm{p}} K_c$. Thus, $G_1, G_2 \longrightarrow_{\mathrm{p}} V_c \bullet K_c$, by use of p•.
- $\mathrm{p}\nu\&_1$: then $G_2 \equiv V'' \bullet x.(\langle x, W''\rangle \bullet K^?)$; for some $V \longrightarrow_{\mathrm{p}} V''$, $W \longrightarrow_{\mathrm{p}} W''$ and $\mathtt{fst}[K] \longrightarrow_{\mathrm{p}} K^?$.
By inspection, $K^? \equiv \mathtt{fst}[K'']$ for some $K \longrightarrow_{\mathrm{p}} K''$, and so, by IH, there exist V_c, K_c with $V', V'' \longrightarrow_{\mathrm{p}} V_c$ and $K', K'' \longrightarrow_{\mathrm{p}} K_c$; thus,

$$\langle x, W''\rangle \bullet \mathtt{fst}[K''] \overset{\mathrm{p}\beta\&_1}{\longrightarrow}_{\mathrm{p}} x \bullet K_c$$
$$\therefore\ G_2 \equiv V'' \bullet x.(\langle x, W''\rangle \bullet \mathtt{fst}[K'']) \overset{\mathrm{p}\beta L}{\longrightarrow}_{\mathrm{p}} V_c \bullet K_c$$
$$\text{and } G_1 \equiv V' \bullet K' \overset{\mathrm{p}\bullet}{\longrightarrow}_{\mathrm{p}} V_c \bullet K_c$$

- $\mathrm{p}\nu\&_2$: then $G_2 \equiv W'' \bullet y.(\langle V'', y\rangle \bullet \mathtt{fst}[K''])$ and we follow the same steps as above.
- p•: then $G_2 \equiv M^? \bullet K^?$, with $\langle V, W\rangle \longrightarrow_{\mathrm{p}} M^?$, $\mathtt{fst}[K] \longrightarrow_{\mathrm{p}} K^?$. By inspection, $M^? \equiv \langle V'', W''\rangle$ and $K^? \equiv \mathtt{fst}[K'']$, some $V \longrightarrow_{\mathrm{p}} V''$, $W \longrightarrow_{\mathrm{p}} W''$, $K \longrightarrow_{\mathrm{p}} K''$. Hence, using the IH,
$G_2 \equiv \langle V'', W''\rangle \bullet \mathtt{fst}[K''] \longrightarrow_{\mathrm{p}} V_c \bullet K_c \;{}_{\mathrm{p}}\!\!\longleftarrow V' \bullet K' \equiv G_1$.

➛ $\mathrm{p}\beta\&_2, \mathrm{p}\beta\vee_1, \mathrm{p}\beta\vee_2, \mathrm{p}\beta\neg$: proven similarly.

➛ $\mathrm{p}\beta L$: then $G \equiv V \bullet x.(S)$ and $G_1 \equiv S'\{V'/x\}$; for some $V \longrightarrow_{\mathrm{p}} V'$, $S \longrightarrow_{\mathrm{p}} S'$.
By inspection, we have the following choices for $G \longrightarrow_{\mathrm{p}} G_2$:

- $\mathrm{p}\beta L$: then $G_2 \equiv S''\{V''/x\}$ and the result is straightforward from Lemma 2.4.
- $\mathrm{p}\nu\&_1$: then $G \equiv \langle V, W\rangle \bullet x.(S)$, $G_1 \equiv S'\{\langle V', W'\rangle/x\}$ and $G_2 \equiv V'' \bullet y.(\langle y, W''\rangle \bullet x.(S''))$; for some $V \longrightarrow_{\mathrm{p}} V', V''$, $W \longrightarrow_{\mathrm{p}} W', W''$, $S \longrightarrow_{\mathrm{p}} S', S''$. Then, by IH, there exist $V', V'' \longrightarrow_{\mathrm{p}} V_c$, $W', W'' \longrightarrow_{\mathrm{p}} W_c$, $S', S'' \longrightarrow_{\mathrm{p}} S_c$, and thus,

$$\langle y, W''\rangle \bullet x.(S'') \overset{\mathrm{p}\beta L}{\longrightarrow} S_c\{\langle y, W_c\rangle/x\}$$
$$\therefore\ V'' \bullet y.(\langle y, W''\rangle \bullet x.(S'')) \overset{\mathrm{p}\beta L}{\longrightarrow} (S_c\{\langle y, W_c\rangle/x\})\{V_c/y\} \overset{y\text{ fresh}}{\equiv} S_c\{\langle V_c, W_c\rangle/x\}$$
$$\text{and } S'\{\langle V', W'\rangle/x\} \longrightarrow_{\mathrm{p}} S_c\{\langle V_c, W_c\rangle/x\}, \text{ by Lemma 2.4}$$

- $\mathrm{p}\nu\&_2, \mathrm{p}\nu\neg_1, \mathrm{p}\nu\neg_2$: proven similarly.

• p•: this case is straightforward by applying Lemma 2.4.
➛ $\mathtt{p}\beta R$: this case is proven in a similar, yet simpler, way as $\mathtt{p}\beta L$.
➛ $\mathtt{p}v\&_1$: then $G \equiv \langle M, N\rangle \bullet K$ and $G_1 \equiv M' \bullet x.(\langle x, N'\rangle \bullet K')$; for some $M \longrightarrow_{\mathtt{p}} M'$, $N \longrightarrow_{\mathtt{p}} N'$ and $K \longrightarrow_{\mathtt{p}} K'$. Then, the possible choices for $G \longrightarrow_{\mathtt{p}} G_2$ which have not been considered above in symmetry are:
• $\mathtt{p}v\&_2$: then M is a value and $G_2 \equiv N'' \bullet y.(\langle M'', y\rangle \bullet K'')$; for some $M \longrightarrow_{\mathtt{p}} M''$, $N \longrightarrow_{\mathtt{p}} N''$ and $K \longrightarrow_{\mathtt{p}} K''$. Using the IH, we have that,

$$\begin{aligned} \langle x, N'\rangle \bullet K' &\overset{\mathtt{p}v\&_2}{\longrightarrow} N_c \bullet y.(\langle x, y\rangle \bullet K_c) \\ \therefore\ G_1 &\overset{\mathtt{p}\beta L}{\longrightarrow} N_c \bullet y.(\langle M_c, y\rangle \bullet K_c) \\ \text{and clearly } G_2 &\longrightarrow_{\mathtt{p}} N_c \bullet y.(\langle M_c, y\rangle \bullet K_c) \end{aligned}$$

• p•: this case is straightforward.
➛ $\mathtt{p}v\&_2$, $\mathtt{p}v\vee_1$, $\mathtt{p}v\vee_2$: proven similarly.
➛ p•: this case is simple, since all reductions to which it may be combined are already checked in the previous cases, by symmetry.
➛ $\mathtt{p}\langle,\rangle$, $\mathtt{p}\langle\rangle\mathtt{inl}$, $\mathtt{p}\langle\rangle\mathtt{inr}$, $\mathtt{p}[]\mathtt{not}$, $\mathtt{pnot}\langle\rangle$, $\mathtt{p}().$, $\mathtt{p}.()$, $\mathtt{pfst}[]$, $\mathtt{psnd}[]$: these cases are trivial, since no other reduction can be combined to any one of them. □

Proof of Lemma 2.10. We do a case analysis on the reduction $G \xrightarrow{\beta v} G_1$, which we label with an index: $G \xrightarrow{\beta v^0} G_1$.

In the following, C denotes some context and G is $C\{G'\}$, with G' being the redex of $\xrightarrow{\beta v^0}$ (so G is, in fact, a statement). The cases of $G \xrightarrow{\eta} G_2$ being an η-reduction that can be trivially "reverted" by a one-step β-reduction, for example, $C\{V \bullet x.(S)\} \xrightarrow{\eta} C\{V \bullet y.(y \bullet x.(S))\} \xrightarrow{\beta} C\{V \bullet x.(S)\}$, are trivial and omitted for economy. For the same reason, the cases of this η-reduction affecting solely C and not its content, for example, $C\{V \bullet x.(S)\} \xrightarrow{\eta} C'\{V \bullet x.(S)\}$, are also omitted.

Hence, we have the following diagrams.

For $\xrightarrow{\beta v^0}$ being βL:

$$\begin{tikzcd} & C\{V \bullet x.(S)\} \arrow[r, "\beta\nu^0"] \arrow[dl, "\eta"] \arrow[d, "\eta"] \arrow[dd, bend right, "\eta"'] & C\{S\{V/x\}\} \arrow[d, "\eta"] \arrow[dd, bend left, two heads, "\eta^1"] \\ C\{M \bullet x.(S)\} \arrow[ur, two heads, "\beta\nu"] & C\{V \bullet x.(S')\} \arrow[r, "\beta\nu"] & C\{S'\{V/x\}\} \\ & C\{V' \bullet x.(S)\} \arrow[r, "\beta\nu^1"] & C\{S\{V'/x\}\} \end{tikzcd}$$

where M is a non-value, and for it Lemma 2.9 is applied. Note in $\overset{\eta^1}{\twoheadrightarrow}$ that only in the occasion where $\xrightarrow{\beta v^0}$ is $\mathtt{sub}_{>1}$ are there more than one η-steps required. In this case, $\xrightarrow{\beta v^1}$ is also $\mathtt{sub}_{>1}$.

For $\xrightarrow{\beta v^0}$ being βR:

$$\begin{tikzcd} & C\{(S).\alpha \bullet K\} \arrow[r, "\beta\nu^0"] \arrow[dl, "\eta"'] \arrow[dr, "\eta"] & C\{S\{K/\alpha\}\} \arrow[dl, "\eta"] \arrow[dr, two heads, "\eta^1"] & \\ C\{(S').\alpha \bullet K\} \arrow[r, "\beta\nu"] & C\{S'\{K/\alpha\}\} & C\{(S).\alpha \bullet K'\} \arrow[r, "\beta\nu^1"] & C\{S\{K'/\alpha\}\} \end{tikzcd}$$

where the same comments as above apply for $\overset{\eta^1}{\twoheadrightarrow}$, $\xrightarrow{\beta v^1}$.

For $\beta\&_1$:

$$\begin{tikzcd}
 & C\{\langle V,W\rangle \bullet \mathtt{fst}[K]\} \arrow[r, "\beta\nu^0"] \arrow[dl, "\eta"] \arrow[d, "\eta"] & C\{V \bullet K\} \arrow[d, "\eta", "="'] \\
C\{M \bullet \mathtt{fst}[K]\} \arrow[ur, two heads, "\beta\nu"] & C\{\langle V',W'\rangle \bullet \mathtt{fst}[K']\} \arrow[r, "\beta\nu"] & C\{V' \bullet K'\}
\end{tikzcd}$$

where M is a non-value. The cases of $\beta\&_2$, $\beta\vee_1$, $\beta\vee_2$ are similar and this of $\beta\neg$ is similar but simpler.

For $\nu\&_2$:

$$\begin{tikzcd}
 & C\{\langle V,N\rangle \bullet K\} \arrow[r, "\beta\nu^0"] \arrow[dl, "\eta"] \arrow[d, "\eta"] & C\{N \bullet y.(\langle V,y\rangle \bullet K)\} \arrow[d, "\eta"] \\
C\{\langle M,N\rangle \bullet K\} \arrow[d, "\nu\&_1"] & C\{\langle V',N'\rangle \bullet K'\} \arrow[r, "\beta\nu"] & C\{N' \bullet y.(\langle V',y\rangle \bullet K')\} \\
C\{M \bullet x.(\langle x,N\rangle \bullet K)\} \arrow[d, "\beta\nu\ (\text{Lemma 2.9})"] & & \\
C\{V \bullet x.(\langle x,N\rangle \bullet K)\} \arrow[uuur, "\beta L"] & &
\end{tikzcd}$$

where M is a non-value, and (since $V \xrightarrow{\eta} M$) $V \bullet x.(\langle x, N\rangle \bullet K) \xrightarrow{\eta} M \bullet x.(\langle x, N\rangle \bullet K)$, whence Lemma 2.9 can be applied. The cases of $\nu\&_1$, $\nu\vee_1$, $\nu\vee_2$ are similar but simpler. □

Proof of Lemma 2.20. We prove something stronger:

For any statement S and covariable α, if we mark the occurrences of α in S by $1, 2, \ldots, n$ and there exist coterms $K_1, \ldots, K_n$ such that $S^p \equiv S\{K_1/_1\alpha, \ldots, K_n/_n\alpha\} \in \mathtt{SN}$, then $S \in \mathtt{SN}$.

Above, $K_i/_i\alpha$ denotes the substitution of K_i for the i-occurrence of α in S.

The proof of this claim is by induction on $l(S^p)$. The base case is this of S^p being in normal form, in which case S is clearly also in normal form.

For the inductive step, suppose there is some infinite reduction sequence from S. Then, in this sequence it must be the case that $S \longrightarrow S'$, with the reduction producing $k \in \mathbb{Z}$ new occurrences of α. But now,

$$S^p \;\equiv\; S\{K_1/_1\alpha, \ldots, K_n/_n\alpha\} \longrightarrow S'\{K'_1/_1\alpha, \ldots, K'_{n+k}/_{n+k}\alpha\} \;\equiv\; S'^p$$

where the K'_i's are selected from the K_i's and, since S^p is $\mathtt{SN}$, S'^p is $\mathtt{SN}$. Moreover, $l(S'^p) < l(S^p)$, thus $S' \in \mathtt{SN}$ by IH, a contradiction.

Hence, S is $\mathtt{SN}$. □

Proof of Lemma 2.24. For CR[1, 2, 3], we do induction on $d(\sigma)$. Note that CR3′ is implied by CR3 by induction on $l(G)$, where G is neutral and $\mathtt{SN}$, using the fact that all its reducts are also neutral and $\mathtt{SN}$ and have the same degree as G.

The base case, of $d(\sigma) = (0, n)$, is straightforward:

- CR1 is a tautology.
- For CR3, if $\sigma' \in \mathtt{Red}$, then G' is $\mathtt{SN}$. So G reduces only to strongly normalizing objects, thus G is $\mathtt{SN}$, $\therefore \sigma \in \mathtt{Red}$.
- For CR2, if $\sigma \in (T_G \cap \mathtt{Red})$, then G is $\mathtt{SN}$ and so is any G' to which it may reduce.

Now assume $d(\sigma) = (c, n),\ c > 0$ and $\sigma \in T_G$:

CR1: [If $\sigma \in (T_G \cap \mathtt{Red})$, some G, then G is $\mathtt{SN}$]

• If $G \equiv S$, some $S \equiv M \bullet K$, and $\sigma \equiv \Gamma \,\mathbf{I}\, S \mapsto \Theta$, then, since $c > 0$, there exists some $\beta : B \in \Theta$ or $y : B \in \Gamma$ with $c(B) > 0$, and thus, for example, in the former case:

$$\sigma' \;\equiv\; \Gamma \;\rightarrow\; \Theta - \{\beta : B\} \,\mathbf{I}\, (S)_{\odot}\beta : B \;\in\; \mathtt{Red}$$

Then, by IH, $(S)_{\odot}\beta \in \mathtt{SN}$, $\therefore S \in \mathtt{SN}$.

• If $G \equiv M$ and $\sigma \equiv \Gamma \rightarrow \Theta \mid M : A$, then, by Proposition 2.23, there exists some derivable sequent $\sigma_1 \equiv K_0 : A \mid \Gamma_0 \rightarrow \Theta_0$, for some K_0 neutral and SN, with $d(\sigma_1) = (0, 1) < d(\sigma)$. By definition, $\sigma_1 \in \mathtt{Red}$. Then,

$$\sigma_2 \equiv \Gamma, \Gamma_0 \mid M \bullet K_0 \mapsto \Theta, \Theta_0$$

has $d(\sigma_2) = (c, 0) < (c, 1) = d(\sigma)$. Since $\sigma \in \mathtt{Red}$, by definition, $\sigma_2 \in \mathtt{Red}$,
$\therefore$ by IH, $M \bullet K_0 \in \mathtt{SN}$, $\therefore M \in \mathtt{SN}$.
• If $G \equiv K$, we work dually as above.
CR3: [If $\sigma \in T_G$, G neutral, and $\sigma \longrightarrow \sigma'$ implies $\sigma' \in \mathtt{Red}$, then $\sigma \in \mathtt{Red}$]
• If $G \equiv M$ and $\sigma \equiv \Gamma \rightarrow \Theta \mid M : A$, then we need to show that for all relevant K_0 and $\sigma_1 \in (T_{K_0} \cap \mathtt{Red})$, we have $\sigma_2 \in \mathtt{Red}$, where:

$$\sigma_2 \equiv \Gamma, \Gamma_0 \mid M \bullet K_0 \mapsto \Theta, \Theta_0, \quad \text{with } d(\sigma_2) < d(\sigma)$$
$$\sigma_1 \equiv K_0 : A \mid \Gamma_0 \rightarrow \Theta_0, \quad d(\sigma_1) < d(\sigma)$$

Since $d(\sigma_1) < d(\sigma)$, by IH on CR1, K_0 is SN. Thus, prove that $\sigma_2 \in \mathtt{Red}$ by induction on $l(K_0)$. Since M, K_0 are neutral, $\sigma_2 \longrightarrow \sigma_2'$ implies:

$$\sigma_2' \equiv \Gamma, \Gamma_0 \mid M' \bullet K_0' \mapsto \Theta, \Theta_0$$

where either $M \longrightarrow M'$, or $K_0 \longrightarrow K_0'$.
In the former case, $\sigma_2' \in \mathtt{Red}$ by hypothesis.
In the latter case, by IH on CR2, $\sigma_1' \equiv K_0' : A \mid \Gamma_0 \rightarrow \Theta_0 \in \mathtt{Red}$ and $l(K_0') < l(K_0)$, thus the IH on $l(K_0)$ applies and $\sigma_2' \in \mathtt{Red}$.
In any case, $\sigma_2 \longrightarrow \sigma_2'$ implies $\sigma_2' \in \mathtt{Red}$, $\therefore \sigma_2 \in \mathtt{Red}$, by IH on CR3.
• If $G \equiv K$, we work dually as above.
• If $G \equiv S \equiv M \bullet K$, and, say $\sigma \equiv \Gamma \mid S \mapsto \Theta \in T_S$, then $\sigma \in \mathtt{Red}$ iff for all $x : B \in \Gamma$, $\alpha : C \in \Theta$, with $c(B) > 0$, $c(C) > 0$, we have $\sigma_1, \sigma_2 \in \mathtt{Red}$, where,

$$\sigma_1 \equiv \Gamma \rightarrow \Theta - \{\alpha : C\} \mid (S)_{\odot}\alpha : C$$
$$\sigma_2 \equiv x_{\odot}(S) : B \mid \Gamma - \{x : B\} \rightarrow \Theta$$

But $\sigma_1 \longrightarrow \sigma_1'$ implies $\sigma_1' \equiv \Gamma \rightarrow \Theta - \{\alpha : C\} \mid (S')_{\odot}\alpha : C$, some $S \longrightarrow S'$, and, by hypothesis, $\sigma' \equiv \Gamma \mid S' \mapsto \Theta \in \mathtt{Red}$, thus $\sigma_1' \in \mathtt{Red}$, $\therefore$ by IH, $\sigma_1 \in \mathtt{Red}$.
Similarly, $\sigma_2 \in \mathtt{Red}$; hence, $\sigma \in \mathtt{Red}$.
CR2: [If $\sigma \in (\mathtt{Red} \cap T_G)$, some G, and $\sigma \longrightarrow \sigma'$, then $\sigma' \in \mathtt{Red}$]
• If $G \equiv M$ and $\sigma \equiv \Gamma \rightarrow \Theta \mid M : A$, then $\sigma \longrightarrow \sigma'$ implies that $\sigma' \equiv \Gamma \rightarrow \Theta \mid M' : A$, with $M \longrightarrow M'$.
Now, $\sigma \in \mathtt{Red}$; therefore, for all relevant K_0, $\sigma_1 \in (T_{K_0} \cap \mathtt{Red})$ and

$$\sigma_2 \equiv \Gamma, \Gamma_0 \mid M \bullet K_0 \mapsto \Theta, \Theta_0 \quad \text{with } d(\sigma_2) < d(\sigma),$$

we have that $\sigma_2 \in \mathtt{Red}$. Moreover,

$$\sigma_2 \longrightarrow \sigma_2' \equiv \Gamma, \Gamma_0 \mid M' \bullet K_0 \mapsto \Theta, \Theta_0$$

so, by IH, $\sigma_2' \in \mathtt{Red}$ for all such σ_2', and thus $\sigma' \in \mathtt{Red}$.
• If $G \equiv K : A$, we work dually as above.
• If $G \equiv M \bullet K \equiv S$, $\sigma \equiv \Gamma \mid S \mapsto \Theta$, then $\sigma \longrightarrow \sigma'$ implies $\sigma' \equiv \Gamma \mid S' \mapsto \Theta$, some $S \longrightarrow S'$. Since $\sigma \in \mathtt{Red}$, for all $y : B \in \Gamma$, $\beta : C \in \Theta$ with $c(B) > 0$, $c(C) > 0$, we have $\sigma_1, \sigma_2 \in \mathtt{Red}$, where,

$$\sigma_1 \equiv \Gamma \rightarrow \Theta - \{\beta : C\} \mid (S)_{\odot}\beta : C$$
$$\sigma_2 \equiv y_{\odot}(S) : B \mid \Gamma - \{y : B\} \rightarrow \Theta$$

By IH, since $\sigma_1 \longrightarrow \sigma_1' \equiv \Gamma \rightarrow \Theta - \{\beta : C\} \mid (S')_{\odot}\beta : C$, we have $\sigma_1' \in \mathtt{Red}$, and similarly $\sigma_2' \in \mathtt{Red}$, for all relevant σ_1', σ_2', $\therefore \sigma' \in \mathtt{Red}$. □

Proof of Lemma 3.2. Let $\sigma \in T_G(A, \Gamma, \Theta)$; we do induction on the derivation of σ. By a case analysis on the last rule of the derivation, the only non-straightforward cases are those of rules with quantifiers.

Assume the last rule is

$$\dfrac{K : A'\{C/Y\} \mid \Gamma \rightarrow \Theta}{\mathsf{a}\cdot K : \forall Y.A' \mid \Gamma \rightarrow \Theta}\,\forall L$$

By IH, we can derive

$$K : (A'\{C/Y\})\{B/X\} \mid \Gamma\{B/X\} \rightarrow \Theta\{B/X\}$$

Now, $(A'\{C/Y\})\{B/X\} \equiv ((A'\{Z/Y\})\{B/X\})\{C\{B/X\}/Z\}$, some fresh Z, and $\forall Z.(A'\{Z/Y\})\{B/X\} \equiv (\forall Z.A'\{Z/Y\})\{B/X\} \equiv (\forall Y.A')\{B/X\}$; thus, by $\forall L$ we derive

$$\mathsf{a}\cdot K : (\forall Y.A')\{B/X\} \mid \Gamma\{B/X\} \rightarrow \Theta\{B/X\}$$

as required. Now suppose the last rule is

$$\dfrac{\Gamma \rightarrow \Theta \mid M : A'}{\Gamma \rightarrow \Theta \mid \mathsf{a}\cdot M : \forall Y.A'}\,\forall R$$

with $Y \notin FTV(\Gamma, Th)$. Choose some fresh Z, by applying twice the IH we can derive

$$\Gamma \rightarrow \Theta \mid M : A'\{Z/Y\}$$
$$\text{and}\ \ \Gamma\{B/X\} \rightarrow \Theta\{B/X\} \mid M : (A'\{Z/Y\})\{B/X\}$$

and by $\forall R$ we derive

$$\Gamma\{B/X\} \rightarrow \Theta\{B/X\} \mid \mathsf{a}\cdot M : \forall Z.(A'\{Z/Y\})\{B/X\}$$

where $\forall Z.(A'\{Z/Y\})\{B/X\} \equiv (\forall Z.A'\{Z/Y\})\{B/X\} \equiv (\forall Y.A')\{B/X\}$, as required.

The cases of $\exists R$, $\exists L$ are dealt with similarly. □

Proof of Proposition 3.4. The proof is by a case analysis on the rule used for the reduction and it is rather straightforward. We are going to show some characteristic cases.

Suppose that $G \longrightarrow H$ by use of the $\beta\&_1$ rule. Then, it suffices to show that $T_{\langle M,N\rangle\bullet\mathtt{fst}[K]}(A, \Gamma, \Theta) \neq \emptyset$ implies $T_{M\bullet K}(A, \Gamma, \Theta) \neq \emptyset$. Suppose that $\Gamma \mid \langle M, N\rangle \bullet \mathtt{fst}[K] \mapsto \Theta$ is derivable. By inspection of the rules, its derivation must end with

$$\dfrac{\dfrac{K : B \mid \Gamma \rightarrow \Theta}{\mathtt{fst}[K] : B\&C \mid \Gamma \rightarrow \Theta} \qquad \dfrac{\Gamma \rightarrow \Theta \mid M : B \quad \Gamma \rightarrow \Theta \mid N : C}{\Gamma \rightarrow \Theta \mid \langle M, N\rangle : B\&C}}{\Gamma \mid \langle M, N\rangle \bullet \mathtt{fst}[K] \mapsto \Theta}$$

and the claim straightforwardly follows.

Suppose that $G \longrightarrow H$ by use of the $\nu\&_1$ rule. Then it suffices to show that $T_{\langle M,N\rangle\bullet K}(A, \Gamma, \Theta) \neq \emptyset$ implies $T_{M\bullet x.(\langle x,N\rangle\bullet K)}(A, \Gamma, \Theta) \neq \emptyset$. Let $\Gamma \mid \langle M, N\rangle \bullet K \mapsto \Theta$ be derivable. By inspection of the typing rules, its derivation must end with

$$\dfrac{K : B_1\&B_2 \mid \Gamma \rightarrow \Theta \qquad \dfrac{\Gamma \rightarrow \Theta \mid M : B_1 \quad \Gamma \rightarrow \Theta \mid N : B_2}{\Gamma \rightarrow \Theta \mid \langle M, N\rangle : B_1\&B_2}}{\Gamma \mid \langle M, N\rangle \bullet K \mapsto \Theta}$$

and the claim follows, using also Weakening.

Suppose that $G \longrightarrow Q$ by use of the βR rule. Then it suffices to show that $T_{(S).\alpha\bullet K}(A, \Gamma, \Theta) \neq \emptyset$ implies $T_{S\{K/\alpha\}}(A, \Gamma, \Theta) \neq \emptyset$. Let $\Gamma \mid (S).\alpha \bullet K \mapsto \Theta$ be derivable. By inspection of the typing rules, its derivation must end with

$$\dfrac{K : B \mid \Gamma \rightarrow \Theta \qquad \dfrac{\Gamma \mid S \mapsto \Theta, \alpha : B}{\Gamma \rightarrow \Theta \mid (S).\alpha : B}}{\Gamma \mid (S).\alpha \bullet K \mapsto \Theta}$$

and the claim follows from Lemma 3.3.

Finally, suppose that $G \longrightarrow Q$ by use of the $\beta\mathsf{a}$ rule. Then it suffices to show that $T_{\mathsf{a}\cdot M \bullet \mathsf{a}\cdot K}(A, \Gamma, \Theta) \neq \emptyset$ implies $T_{M \bullet K}(A, \Gamma, \Theta) \neq \emptyset$. Let $\Gamma \mid \mathsf{a}\cdot M \bullet \mathsf{a}\cdot K \mapsto \Theta$ be derivable. By inspection of the typing rules, its derivation must end with

$$\dfrac{\dfrac{\Gamma \to \Theta \mid M : A'}{\Gamma \to \Theta \mid \mathsf{a}\cdot M : \forall X.A'}\forall R \qquad \dfrac{K : A'\{B/X\} \mid \Gamma \to \Theta}{\mathsf{a}\cdot K : \forall X.A' \mid \Gamma \to \Theta}\forall L}{\Gamma \mid \mathsf{a}\cdot M \bullet \mathsf{a}\cdot K \mapsto \Theta}$$

where $X \notin FTV(\Gamma, \Theta)$. The claim follows from Lemma 3.2. □

References

[1] F. Barbanera, S. Berardi, A symmetric lambda calculus for classical program extraction, Inform. and Comput. 125 (2) (1996) 103–117.
[2] H.P. Barendregt, The Lambda Calculus. Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics, Vol. 103, North-Holland, Amsterdam, 1984.
[3] H.P. Barendregt, Lambda calculi with types, in: Abramsky, Gabbay, Maibaum (Eds.), Handbook of Logic in Computer Science, Vol. 2. Background: Computational Structures, Oxford University Press, Inc., Oxford, 1992, pp. 117–309.
[4] P.-L. Curien, H. Herbelin, The duality of computation, in: Proc. Fifth ACM SIGPLAN Internat. Conf. on Functional Programming, ACM Press, New York, 2000, pp. 233–243.
[5] D.J. Dougherty, Some lambda calculi with categorical sums and products, in: C. Kirchner (Ed.), Proc. Fifth Internat. Conf. on Rewriting Techniques and Applications (RTA), Lecture Notes in Computer Science, Vol. 690, Springer, Berlin, 1993, pp. 137–151.
[6] M. Felleisen, R. Hieb, The revised report on the syntactic theories of sequential control and state, Theoret. Comput. Sci. 103 (2) (1992) 235–271.
[7] A. Filinski, Declarative continuations: an investigation of duality in programming language semantics, in: D.H. Pitt et al. (Ed.), Category Theory and Computer Science, Lecture Notes in Computer Science, Vol. 389, Springer, Berlin, 1989, pp. 224–249.
[8] S. Fortune, D. Leivant, M. O'Donnell, The expressiveness of simple and second-order type structures, J. ACM 30 (1) (1983) 151–185.
[9] G. Gentzen, Untersuchungen über das logische schließen, Mathematische Zeitschrift 39 (1935) 176–210, 405–431. English translation in G. Gentzen, Investigations into logical deductions, 1935, in: M.E. Szabo (Ed.), The Collected Works of Gerhard Gentzen, North-Holland, Amsterdam, 1969, pp. 68–131.
[11] J.-Y. Girard, P. Taylor, Y. Lafont, Proofs and Types, Cambridge Tracts in Theoretical Computer Science, Vol. 7, Cambridge University Press, Cambridge, 1989.
[12] T.G. Griffin, A formulae-as-types notion of control, in: Conf. Record 17th Annu. ACM Symp. on Principles of Programming Languages, POPL'90, San Francisco, CA, USA, 17–19 January 1990, ACM Press, New York, 1990, pp. 47–57.
[13] M. Parigot, $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction, in: A. Voronkov (Ed.), Proc. Internat. Conf. on Logic Programming and Automated Reasoning, St. Petersburg, Russia, July 1992, Lecture Notes in Computer Science, Vol. 624, Springer, Berlin, pp. 190–201.
[14] M. Parigot, Strong normalization of second order symmetric lambda-calculus, in: Proc. 20th Conf. on Foundations of Software Technology and Theoretical Computer Science, Springer, Berlin, 2000, pp. 442–453.
[15] A.M. Pitts, On an interpretation of second order quantification in first order intuitionistic propositional logic, J. Symbolic Logic 57 (1) (1992) 33–52.
[16] P. Selinger, Control categories and duality: on the categorical semantics of the lambda–mu calculus, Math. Struct. Comput. Sci. 11 (2) (2001) 207–260.
[17] M.H. Sørensen, P. Urzyczyn, Lectures on the Curry–Howard isomorphism. Available as DIKU Rapport 98/14, 1998.
[18] C. Strachey, C.P. Wadsworth, Continuations: a mathematical semantics for handling full jumps. Technical Monograph PRG-11, Oxford University Computing Laboratory Programming Research Group, 1974. Republished in C. Strachey, C.P. Wadsworth, Continuations: a mathematical semantics for handling full jumps, Higher-Order Symbolic Comput. 13(1–2) (2000) 135–152.
[20] P. Wadler, Call-by-value is dual to call-by-name, ICFP 2003, Uppsala, Sweden, 25–29 August 2003.
[21] P. Wadler, Call-by-value is dual to call-by-name—reloaded, in: Jürgen Giesl (Ed.), RTA, Lecture Notes in Computer Science, Vol. 3467, Springer, Berlin, 2005, pp. 185–203.
[22] N. Tzevelekos, Investigations on the Dual Calculus, RR-04-21, Oxford University Computing Laboratory, 2004.