



HAL
open science

A logic identifying isomorphic propositions

Alejandro Díaz-Caro, Gilles Dowek

► **To cite this version:**

Alejandro Díaz-Caro, Gilles Dowek. A logic identifying isomorphic propositions. 2018. hal-01109104v4

HAL Id: hal-01109104

<https://inria.hal.science/hal-01109104v4>

Preprint submitted on 8 Aug 2018 (v4), last revised 27 Aug 2018 (v5)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Logic Identifying Isomorphic Propositions[†]

ALEJANDRO DÍAZ-CARO^{1,2} and GILLES DOWEK³

¹ *Universidad Nacional de Quilmes. Roque Sáenz Peña 352, Bernal, Buenos Aires, Argentina.*

² *CONICET-Universidad de Buenos Aires – Instituto de Ciencias de la Computación (ICC).
Pabellón 1, Ciudad Universitaria, Ciudad Autónoma de Buenos Aires, Argentina.*

³ *Inria, LSV, ENS Paris-Saclay. 61 avenue du Président Wilson, 94235 Cachan, France.*

Received 8 August 2018

We define a fragment of propositional logic where isomorphic propositions, such as $A \wedge B$ and $B \wedge A$, or $A \Rightarrow (B \wedge C)$ and $(A \Rightarrow B) \wedge (A \Rightarrow C)$ are identified. We define System I, a proof language for this logic, and prove the strong normalisation of proof-reduction in this system.

1. Introduction

1.1. Identifying isomorphic propositions

Mathematical structures are defined by operations and properties of these operations. For instance, addition in a field is associative, commutative, etc. In contrast, conjunction in logic is neither associative nor commutative, implication does not distribute over conjunction, etc. So, the propositions form a free algebra defined by operations, but no properties. For instance, the propositions $A \wedge B$ and $B \wedge A$ are different, and they have different proofs: if $A \wedge B$ has a proof, then so does $B \wedge A$, but if r is a proof of $A \wedge B$, then it is not a proof of $B \wedge A$.

A first step towards considering $A \wedge B$ and $B \wedge A$ as the same proposition has been made by Rittri [1990], Bruce, Di Cosmo, and Longo [1992], and Di Cosmo [1995, 2005], who have defined a notion of isomorphic propositions and characterised them. Two propositions A and B are isomorphic if there exists two proofs of $A \Rightarrow B$ and $B \Rightarrow A$ whose composition, in both ways, is the identity.

For the fragment of propositional logic restricted to the operations \Rightarrow and \wedge , all the isomorphisms are consequences of the following four:

$$A \wedge B \equiv B \wedge A \tag{1}$$

$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C \tag{2}$$

$$A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C) \tag{3}$$

$$(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C \tag{4}$$

[†] Partially supported by projects ECOS-Sud A17C03 QuCa, PICT 2015-1208, and the LIA INFINIS.

For example, $(A \Rightarrow B \Rightarrow C) \equiv (B \Rightarrow A \Rightarrow C)$ is a consequence of (4) and (1).

In this paper, we go one step further and define a proof language, System I, for the fragment \Rightarrow, \wedge , such that when $A \equiv B$, then any proof of A is also a proof of B , so the propositions $A \wedge B$ and $B \wedge A$, for instance, are really identical, as they have the same proofs.

The idea of identifying some propositions has already been investigated, for example, in Martin-Löf's type theory [Martin-Löf, 1984], in the Calculus of Constructions [Coquand and Huet, 1988], and in Deduction modulo [Dowek et al., 2003, Dowek and Werner, 2003], where definitionally equivalent propositions, for instance $x \subseteq y$, $x \in P(y)$, and $\forall z (z \in x \Rightarrow z \in y)$ can be identified. But definitional equality does not handle all the isomorphisms. For example, $A \wedge B$ and $B \wedge A$ cannot be identified in these logics. Beside definitional equality, identifying isomorphic types in type theory, is also a goal of the univalence axiom [The Univalent Foundations Program, 2013].

1.2. Lambda-calculus

The proof-language of the fragment of propositional logic restricted to the operations \Rightarrow and \wedge is simply typed lambda-calculus extend with Cartesian product. So, System I is an extension of simply typed lambda-calculus where, for example, a pair of functions $\langle r, s \rangle$ can be applied to an argument t , yielding a term $\langle r, s \rangle t$ that is itself identified with the term $\langle rt, st \rangle$. To stress the associativity and commutativity of the notion of pair, we write $r \times s$ the term $\langle r, s \rangle$ and thus write this equivalence

$$(r \times s)t = rt \times st$$

Several similar equivalence rules on terms are introduced: one related to the isomorphism (1), the commutativity of the conjunction, $r \times s \rightleftharpoons s \times r$; one related to the isomorphism (2), the associativity of the conjunction, $(r \times s) \times t \rightleftharpoons r \times (s \times t)$; four to the isomorphism (3), the distributivity of implication with respect to conjunction, e.g. $(r \times s)t \rightleftharpoons rt \times st$; and one related to the isomorphisms (4), the currfication, $rst \rightleftharpoons r(s \times t)$.

One of the main difficulties in the design of System I is the design of the elimination rule for the conjunction. A rule like “if $r : A \wedge B$ then $\pi_1(r) : A$ ”, would not be consistent. Indeed, if A and B are two arbitrary types, s a term of type A and t a term of type B , then $s \times t$ has both types $A \wedge B$ and $B \wedge A$, thus $\pi_1(s \times t)$ would have both type A and type B . A solution is to consider explicitly typed (Church style) terms, and parametrise the projection by the type: if $r : A \wedge B$ then $\pi_A(r) : A$ and the reduction rule is then that $\pi_A(s \times t)$ reduces to s if s has type A .

This rule makes reduction non-deterministic. Indeed, in the particular case where A happens to be equal to B , then both s and t have type A and $\pi_A(s \times t)$ reduces both to s and to t . Notice that although this reduction rule is non-deterministic, it preserves typing, just like the calculus developed in [Dowek and Jiang, 2011], where the reduction is non-deterministic, but verifies subject reduction.

1.3. Non-determinism

Therefore, System I is one of the many non-deterministic calculi in the sense, for instance, of [Boudol, 1994, Bucciarelli et al., 2012, de'Liguoro and Piperno, 1995, Dezani-Ciancaglini et al., 1998, Pagani and Ronchi Della Rocca, 2010] and our pair-construction operator \times is also the parallel composition operator of a non-deterministic calculus.

In non-deterministic calculi, the non-deterministic choice is such that if r and s are two λ -terms, the term $r \oplus s$ represents the computation that runs either r or s non-deterministically, that is such that $(r \oplus s)t$ reduces either to rt or st . On the other hand, the parallel composition operator $|$ is such that the term $(r | s)t$ reduces to $rt | st$ and continue running both rt and st in parallel. In our case, given r and s of type $A \Rightarrow B$ and t of type A , the term $\pi_B((r \times s)t)$ is equivalent to $\pi_B(rt \times st)$, which reduces to rt or st , while the term $rt \times st$ itself would run both computations at the same time. Hence, our \times is equivalent to the parallel composition while the non-deterministic choice \oplus is decomposed into \times followed by π .

In System I, the non-determinism comes from the interaction of two operators, \times and π . This is similar to quantum computing where the non-determinism comes from the interaction of two operators, the first allowing to build a superposition, that is a linear combination, of two terms $\alpha.r + \beta.t$, and the measurement operator π . In addition, in such calculi, the distributivity rule $(r + s)t \rightleftharpoons rt + st$ is seen as the point-wise definition of the sum of two functions.

More generally, the calculus developed in this paper is also related to the algebraic calculi [Arrighi and Díaz-Caro, 2012, Arrighi et al., 2017, Arrighi and Dowek, 2008, 2017, Vaux, 2009, Díaz-Caro and Petit, 2012, Díaz-Caro and Dowek, 2017], some of which have been designed to express quantum algorithms. There is a clear link between the pair constructor \times and the projection π , with the superposition constructor $+$ and the measurement π on these calculi. In these cases, the pair $s + t$ is not interpreted as a non-deterministic choice but as a superposition of two processes running s and t , and the operator π is the projection related to the measurement, which is the only non-deterministic operator.

Outline

In Section 2, we define the notion of type isomorphism and prove elementary properties of this relation. In Section 3, we introduce System I. In Section 4, we prove its subject reduction, and its termination in Section 5. In Section 6, we prove that termination implies the consistency of the logic. Finally, in Section 7, we discuss how to use System I as a programming language.

2. Type isomorphisms

Types are defined by the following grammar

$$A, B, C, \dots ::= \tau \mid A \Rightarrow B \mid A \wedge B$$

where τ is the only atomic type.

The Isomorphisms (1), (2), (3), and (4) are made explicit by a congruence on types

$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ (A \wedge B) \wedge C &\equiv A \wedge (B \wedge C) \\ A \Rightarrow (B \wedge C) &\equiv (A \Rightarrow B) \wedge (A \Rightarrow C) \\ (A \wedge B) \Rightarrow C &\equiv A \Rightarrow B \Rightarrow C \end{aligned}$$

Definition 2.1 (Canonical form). A canonical form is inductively defined as follows: if S_1, \dots, S_n for $n > 0$ are either canonical forms or the empty multiset, then the multiset $[S_1 \Rightarrow \tau, \dots, S_n \Rightarrow \tau]$ is a canonical form. We may write τ for $\emptyset \Rightarrow \tau$. The size of a canonical form is defined by induction as: $size[S_1 \Rightarrow \tau, \dots, S_n \Rightarrow \tau] = \sum_{i=1}^n (size(S_i) + 1)$ where $size(\emptyset) = 0$.

Definition 2.2 (Canonical forms of a type). We associate to each type a canonical form as follows

$$\begin{aligned} \text{can}(\tau) &= [\tau] \\ \text{can}(A \Rightarrow B) &= \text{let } \text{can}(B) = [S_i \Rightarrow \tau]_{i=1}^n \\ &\quad \text{in } [(\text{can}(A) \uplus S_i) \Rightarrow \tau]_{i=1}^n \\ \text{can}(A \wedge B) &= \text{can}(A) \uplus \text{can}(B) \end{aligned}$$

We also define the inverse function as

$$\text{nac}([S_i \Rightarrow \tau]_i) = \bigwedge_i \text{nac}'(S_i \Rightarrow \tau)$$

for some fixed order, where

$$\begin{aligned} \text{nac}'(\tau) &= \tau \\ \text{If } S \neq \emptyset \text{ then } \text{nac}'(S \Rightarrow \tau) &= \text{nac}(S) \Rightarrow \tau \end{aligned}$$

Example 2.3.

$$\begin{aligned} \text{can}(\tau \Rightarrow (\tau \wedge \tau)) &= [[\tau] \Rightarrow \tau, [\tau] \Rightarrow \tau] \\ \text{nac}(\text{can}(\tau \Rightarrow (\tau \wedge \tau))) &= (\tau \Rightarrow \tau) \wedge (\tau \Rightarrow \tau) \\ \text{can}((\tau \wedge \tau) \Rightarrow \tau) &= [[\tau, \tau] \Rightarrow \tau] \\ \text{nac}(\text{can}((\tau \wedge \tau) \Rightarrow \tau)) &= (\tau \wedge \tau) \Rightarrow \tau \end{aligned}$$

Lemma 2.4. If $A \equiv B$, then $\text{can}(A) = \text{can}(B)$.

Proof. By structure on the derivation of the equivalence. □

Lemma 2.5. $\forall A, A \equiv \text{nac}(\text{can}(A))$.

Proof. By induction on A . □

In simply typed lambda-calculus, the size of a type verifies the property that the size of A and B are smaller than the size of $A \Rightarrow B$ and $A \wedge B$. Unfortunately, this notion of size is not preserved by equivalence. For instance the type $\tau \Rightarrow (\tau \wedge \tau)$ is equivalent to the bigger type $(\tau \Rightarrow \tau) \wedge (\tau \Rightarrow \tau)$. Thus, we define the measure $m(A)$ as $size(\text{can}(A))$.

Notice that, by Lemma 2.4, if $A \equiv B$, then $m(A) = m(B)$. In addition, the following lemma shows that the measure $m(A)$ verifies the usual properties.

Lemma 2.6.

- 1 $m(A \wedge B) > m(A)$
- 2 $m(A \Rightarrow B) > m(A)$

Proof.

- 1 It suffices to note that $\text{can}(A)$ is a strict submultiset of $\text{can}(A \wedge B)$.
- 2 Let $\text{can}(B) = [S_i \Rightarrow \tau]_{i=1}^n$. Then, $\text{can}(A \Rightarrow B) = [(\text{can}(A) \uplus S_i) \Rightarrow \tau]_{i=1}^n$. Hence, $m(A \Rightarrow B) \geq \text{size}(\text{can}(A) \uplus S_1) + 1 > \text{size}(\text{can}(A) \uplus S_1) \geq \text{size}(\text{can}(A)) = m(A)$. \square

In simply typed lambda calculus, the implication and the conjunction are constructors, that is $A \Rightarrow B$ is never equal to $C \wedge D$, and if $A \Rightarrow B = A' \Rightarrow B'$, then $A = A'$ and $B = B'$, and the same holds for conjunction. This is, of course, not the case in System I where, for example, $\tau \Rightarrow (\tau \wedge \tau) \equiv (\tau \Rightarrow \tau) \wedge (\tau \Rightarrow \tau)$, but the connectors still have some coherence properties such that

- If $A \Rightarrow B \equiv C_1 \wedge C_2$, then there exists B_1 and B_2 such that $B \equiv B_1 \wedge B_2$.
- If $(A \Rightarrow B) \equiv (A' \Rightarrow B')$ and $A \equiv A'$, then $B \equiv B'$.
- If $A \Rightarrow B \equiv (A \Rightarrow C) \wedge D$, then there exists E such that $D \equiv A \Rightarrow E$ and $B \equiv C \wedge E$.
- If $A \Rightarrow \tau \equiv B \Rightarrow C$ then $C \equiv A' \Rightarrow \tau$ and $A \equiv A' \wedge B$.
- If $A \wedge B \equiv C \wedge D$, then either $C \equiv A_1 \wedge B_1$ and $D \equiv A_2 \wedge B_2$, with $A \equiv A_1 \wedge A_2$ and $B \equiv B_2$, or similar cases.

We prove these properties in Lemmas 2.7, 2.8, 2.9, 2.10, and 2.12.

Lemma 2.7. If $A \Rightarrow B \equiv C_1 \wedge C_2$, then there exists B_1 and B_2 such that $B \equiv B_1 \wedge B_2$.

Proof. By Lemma 2.4, $\text{can}(A \Rightarrow B) = \text{can}(C_1 \wedge C_2) = \text{can}(C_1) \uplus \text{can}(C_2)$. Thus, the cardinality of $\text{can}(A \Rightarrow B)$ is at least 2. Notice that the cardinality of $\text{can}(A \Rightarrow B)$ is equal to the cardinality of $\text{can}(B)$. Therefore, the cardinality of $\text{can}(B)$ is at least 2, and we conclude by Lemma 2.5. \square

Lemma 2.8. If $(A \Rightarrow B) \equiv (A' \Rightarrow B')$ and $A \equiv A'$, then $B \equiv B'$.

Proof. Let $\text{can}(B) = [S_i \Rightarrow \tau]_i$ and $\text{can}(B') = [S'_j \Rightarrow \tau]_j$. By Lemma 2.4, $\text{can}(A \Rightarrow B) = \text{can}(A' \Rightarrow B')$. That is, $[(\text{can}(A) \uplus S_i) \Rightarrow \tau]_i = [(\text{can}(A') \uplus S'_j) \Rightarrow \tau]_j$. Since, by Lemma 2.4, $\text{can}(A) = \text{can}(A')$, we have $\text{can}(B) = [S_i \Rightarrow \tau]_j = [S'_j \Rightarrow \tau]_j = \text{can}(B')$. \square

Lemma 2.9. If $A \Rightarrow B \equiv (A \Rightarrow C) \wedge D$, then there exists E such that $D \equiv A \Rightarrow E$ and $B \equiv C \wedge E$.

Proof. Let $\text{can}(B) = [S_i \Rightarrow \tau]_{i=1}^n$. Then, $\text{can}(A \Rightarrow B) = [(\text{can}(A) \uplus S_i) \Rightarrow \tau]_{i=1}^n$. By Lemma 2.4, $\text{can}(A \Rightarrow B) = \text{can}((A \Rightarrow C) \wedge D) = \text{can}(A \Rightarrow C) \uplus \text{can}(D)$. Therefore, $\text{can}(D) \subseteq \text{can}(A \Rightarrow B)$ and $\text{can}(D) = [(\text{can}(A) \uplus S_i) \Rightarrow \tau]_{i \in R}$ for some $R \subseteq \{1, \dots, n\}$. Let $E = \text{nac}([S_i \Rightarrow \tau]_{i \in R})$. Therefore, $D \equiv A \Rightarrow E$. So, $A \Rightarrow B \equiv (A \Rightarrow C) \wedge (A \Rightarrow E)$, and by Lemma 2.8, $B \equiv C \wedge E$. \square

Lemma 2.10. If $A \Rightarrow \tau \equiv B \Rightarrow C$, then $C \equiv A' \Rightarrow \tau$ and $A \equiv A' \wedge B$.

Proof. Without loss of generality, let $C = C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D$ where $D \not\equiv D_1 \Rightarrow D_2$, and let $A' = \bigwedge_i C_i$. We have $C \equiv A' \Rightarrow D$, hence, $A \Rightarrow \tau \equiv (B \wedge A') \Rightarrow D$. If $D \equiv D_1 \wedge D_2$, then, $(B \wedge A') \Rightarrow D \equiv ((B \wedge A') \Rightarrow D_1) \wedge ((B \wedge A') \Rightarrow D_2)$, and hence, by Lemma 2.9, τ would be a conjunction. Hence, D is not a conjunction. Since D is neither a conjunction nor an implication, it has to be τ , and so we have $C \equiv A' \Rightarrow \tau$ and $A \Rightarrow \tau \equiv (B \wedge A') \Rightarrow \tau$. Thus, by Lemma 2.4, $\text{can}(A \Rightarrow \tau) = \text{can}((B \wedge A') \Rightarrow \tau)$. Hence, $[\text{can}(A) \Rightarrow \tau] = [\text{can}(B \wedge A') \Rightarrow \tau]$, so $\text{can}(A) = \text{can}(B \wedge A')$, and therefore $\text{nac}(\text{can}(A)) = \text{nac}(\text{can}(B \wedge A'))$. Hence, by Lemma 2.5, we have $A \equiv B \wedge A'$. \square

Lemma 2.11. Let R, S, T and U be four non-empty multisets such that $R \uplus S = T \uplus U$ then either

- there exists non-empty multisets R_1, R_2, S_1, S_2 such that $R = R_1 \uplus R_2, S = S_1 \uplus S_2, T = R_1 \uplus S_1, U = R_2 \uplus S_2$,
- there exists a non-empty multiset S_2 such that $S = T \uplus S_2, U = R \uplus S_2$,
- there exists a non-empty multiset S_1 such that $S = S_1 \uplus U, T = R \uplus S_1$,
- there exists a non-empty multiset R_2 such that $R = T \uplus R_2, U = R_2 \uplus S$,
- there exists a non-empty multiset R_1 such that $R = R_1 \uplus U, T = R_1 \uplus S$,
- $R = T, S = U$
- $R = U, S = T$

Proof. Let $R_1 = R \cap T, R_2 = R \cap U, S_1 = S \cap T, S_2 = S \cap U$.

We have $R_1 \uplus R_2 = R \cap (T \uplus U) = R \cap (R \uplus S) = R$. Similarly, $S_1 \uplus S_2 = S, R_1 \uplus S_1 = T, R_2 \uplus S_2 = U$.

If none of the multisets R_1, R_2, S_1, S_2 is empty, we get the first case.

If exactly one is empty, we get the four following cases. For instance if $R_1 = \emptyset$ then we get: there exists non-empty multisets R_2, S_1, S_2 such that $R = R_2, S = S_1 \uplus S_2, T = S_1, U = R_2 \uplus S_2$, that is, there exists a non-empty multiset S_2 such that $S = T \uplus S_2$ and $U = R \uplus S_2$. The three other cases are similar.

If two of them are empty, as the multisets $R_1 \uplus R_2, S_1 \uplus S_2, R_1 \uplus S_1$, and $R_2 \uplus S_2$ are non-empty the only possibilities are $R_2 = S_1 = \emptyset$ and $R_1 = S_2 = \emptyset$.

In the first case we get $R = R_1, S = S_2, T = R_1, U = S_2$, that is $R = T$ and $S = U$. In the second, we get $R = U$ and $S = T$.

As the multisets $R_1 \uplus R_2, S_1 \uplus S_2, R_1 \uplus S_1$, and $R_2 \uplus S_2$ are non-empty, it is not possible that three or four of these sets are empty. \square

Lemma 2.12. If $A \wedge B \equiv C \wedge D$ then either

- $A \equiv A_1 \wedge A_2, B \equiv B_1 \wedge B_2, C = A_1 \wedge B_1, D = A_2 \wedge B_2$,
- $B \equiv C \wedge B_2, D \equiv A \wedge B_2$,
- $B \equiv B_1 \wedge D, C \equiv A \wedge B_1$,
- $A \equiv C \wedge A_2, D \equiv A_2 \wedge B$,
- $A \equiv A_1 \wedge D, C \equiv A_1 \wedge B$,
- $A \equiv C, B \equiv D$,
- $A \equiv D, B \equiv C$.

Proof. Let $R = \text{can}(A), S = \text{can}(B), T = \text{can}(C)$, and $U = \text{can}(D)$. By Lemma 2.3,

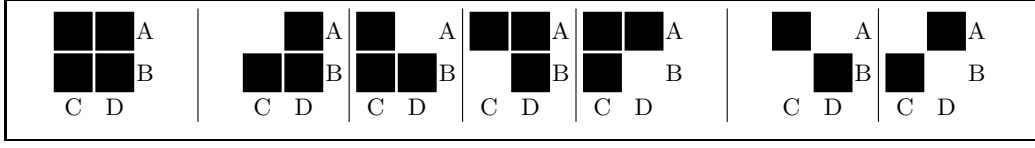


Fig. 1. Seven cases on the equivalence $A \wedge B \equiv C \wedge D$ as prescribed by Lemma 2.12.

we have $\text{can}(A \wedge B) = \text{can}(C \wedge D)$, that is $R \uplus S = T \uplus U$. We apply Lemma 2.11 leading to seven cases. In the first case, we call $A_1 = \text{nac}(R_1)$, $A_2 = \text{nac}(R_2)$, $B_1 = \text{nac}(S_1)$, $B_2 = \text{nac}(S_2)$. By Lemma 2.4, we have $A_1 \wedge A_2 = \text{nac}(R_1) \wedge \text{nac}(R_2) \equiv \text{nac}(R_1 \uplus R_2) = \text{nac}(R) = \text{nac}(\text{can}(A)) \equiv A$. Similarly, $B_1 \wedge B_2 \equiv B$, $A_1 \wedge B_1 \equiv C$ and $A_2 \wedge B_2 \equiv D$. The other cases are similar.

The seven cases prescribed by this lemma are graphically represented in Fig. 1. \square

3. System I

3.1. Syntax

We associate to each type A an infinite set of variables \mathcal{V}_A such that if $A \equiv B$ then $\mathcal{V}_A = \mathcal{V}_B$ and if $A \not\equiv B$ then $\mathcal{V}_A \cap \mathcal{V}_B = \emptyset$, and we use a presentation of typing rules without explicit context following Geuvers et al. [2010] and Park et al. [2014].

The set of terms is defined inductively by the grammar

$$r, s, t, u, v, w ::= x \mid \lambda x. r \mid r s \mid r \times s \mid \pi_A(r)$$

We may write x^A when $x \in \mathcal{V}_A$.

We define the set of free and bound variables of a term t as usual, denoted by $BV(t)$ and $FV(t)$. We say that a term r is closed whenever $FV(r) = \emptyset$. We consider implicit α -equivalence on terms where bound variables can be replaced by other variables of the same type. The substitution on terms is defined as usual in the λ -calculus.

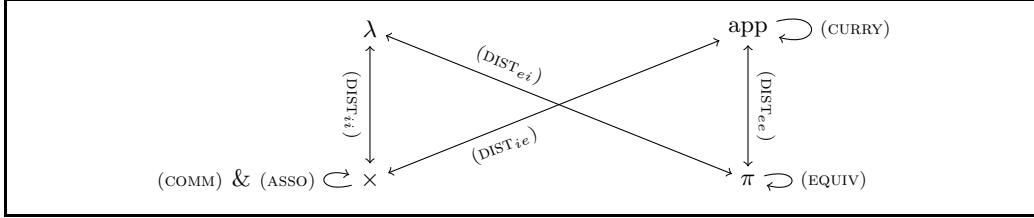
The type system is given in Table 1. Typing judgments are of the form $r : A$. We write the preconditions of a typing rule, at its left.

The operational semantics of the calculus is defined by two relations, the first is an equivalence relation, and the second is a reduction relation.

$\frac{}{x^A : A} \quad (ax)$	$[A \equiv B] \frac{r : A}{r : B} \quad (\equiv)$
$\frac{r : B}{\lambda x^A. r : A \Rightarrow B} \quad (\Rightarrow_i)$	$\frac{r : A \Rightarrow B \quad s : A}{r : A \Rightarrow B \quad s : A} \quad (\Rightarrow_e)$
$\frac{r : A \quad s : B}{r \times s : A \wedge B} \quad (\wedge_i)$	$\frac{rs : B}{r : A \wedge B} \quad (\wedge_e)$
$\frac{}{\pi_A(r) : A}$	

Table 1. *The type system.*

	$r \times s \rightleftharpoons s \times r$	(COMM)
	$(r \times s) \times t \rightleftharpoons r \times (s \times t)$	(ASSO)
	$\lambda x.(r \times s) \rightleftharpoons \lambda x.r \times \lambda x.s$	(DIST _{ii})
	$(r \times s)t \rightleftharpoons rt \times st$	(DIST _{ie})
	$\pi_{A \Rightarrow B}(\lambda x^A.r) \rightleftharpoons \lambda x^A.\pi_B(r)$	(DIST _{ei})
If $\left\{ \begin{array}{l} r : A \Rightarrow (B \wedge C) \\ s : A \end{array} \right\}$,	$\pi_{A \Rightarrow B}(r)s \rightleftharpoons \pi_B(rs)$	(DIST _{ee})
	$rst \rightleftharpoons r(s \times t)$	(CURRY)
	If $A \equiv B$, $\pi_A(r) \rightleftharpoons \pi_B(r)$	(EQUIV)
	$\frac{r \rightleftharpoons s}{C[r] \rightleftharpoons C[s]}$	(C \rightleftharpoons)

Table 2. *Symmetric relation.*Fig. 2. Graphical representation of the symmetric relation \rightleftharpoons from Table 2.

3.2. Equivalence relation

The symmetric relation \rightleftharpoons is defined in Table 2 where the set of contexts is defined by the grammar

$$C[\cdot] ::= [\cdot] \mid \lambda x.C[\cdot] \mid C[\cdot]r \mid rC[\cdot] \mid C[\cdot] \times r \mid r \times C[\cdot] \mid \pi_A(C[\cdot])$$

Each isomorphism, taken as equivalence between types, induces an equivalence between terms. Four possible rules exist however for the isomorphism (3), depending on which distribution is taken into account: elimination or introduction of conjunction, and elimination or introduction of implication.

In Fig. 2, the symmetric relation is graphically represented, taking into account the symbol in head position on each rule.

Because of the associativity property of \times , the term $r \times (s \times t)$ is the same as the term $(r \times s) \times t$, so we can just express it as $r \times s \times t$.

We write \rightleftharpoons^* for the transitive and reflexive closure of \rightleftharpoons . Note that \rightleftharpoons^* is an equivalence relation.

In Table 3 we introduce measures on terms P and M .

We want to prove that the equivalence class of a term is a finite set (Lemma 3.3). To do so, we need the following two auxiliary lemmas (Lemmas 3.1 and 3.2), showing that $P(t)$ and $M(t)$ are invariant with respect to \rightleftharpoons .

$P(x) = 0$	$M(x) = 1$
$P(\lambda x.r) = P(r)$	$M(\lambda x.r) = 1 + M(r) + P(r)$
$P(rs) = P(r)$	$M(rs) = M(r) + M(s) + P(r)M(s)$
$P(r \times s) = 1 + P(r) + P(s)$	$M(r \times s) = M(r) + M(s)$
$P(\pi_A(r)) = P(r)$	$M(\pi_A(r)) = 1 + M(r) + P(r)$

Table 3. *Measure on terms.*

Lemma 3.1. If $r \rightleftharpoons s$ then $P(r) = P(s)$.

Proof. By structural induction on relation \rightleftharpoons .

$$r \times s \rightleftharpoons s \times r: P(r \times s) = 1 + P(r) + P(s) = P(s \times r).$$

$$(r \times s) \times t \rightleftharpoons r \times (s \times t): P((r \times s) \times t) = 2 + P(r) + P(s) + P(t) = P(r \times (s \times t)).$$

$$\begin{aligned} \lambda x.(r \times s) \rightleftharpoons \lambda x.r \times \lambda x.s: P(\lambda x.(r \times s)) &= P(r \times s) \\ &= 1 + P(r) + P(s) = 1 + P(\lambda x.r) + P(\lambda x.s) \\ &= P(\lambda x.r \times \lambda x.s) \end{aligned}$$

$$\begin{aligned} (r \times s)t \rightleftharpoons rt \times st: P((r \times s)t) &= P(r \times s) \\ &= 1 + P(r) + P(s) = 1 + P(rt) + P(st) \\ &= P(rt \times st) \end{aligned}$$

$$\begin{aligned} \pi_{A \Rightarrow B}(\lambda x.r) \rightleftharpoons \lambda x.\pi_B(r): P(\pi_{A \Rightarrow B}(\lambda x.r)) &= P(\lambda x.r) \\ &= P(r) = P(\pi_B(r)) \\ &= P(\lambda x.\pi_B(r)) \end{aligned}$$

$$\begin{aligned} \pi_{A \Rightarrow B}(r)s \rightleftharpoons \pi_B(rs): P(\pi_{A \Rightarrow B}(r)s) &= P(\pi_{A \Rightarrow B}(r)) \\ &= P(r) = P(rs) \\ &= P(\pi_B(rs)) \end{aligned}$$

$$(rs)t \rightleftharpoons r(s \times t): P((rs)t) = P(rs) = P(r) = P(r(s \times t)).$$

$$\pi_A(r) \rightleftharpoons \pi_B(r), \text{ with } A \equiv B: P(\pi_A(r)) = P(r) = P(\pi_B(r)).$$

$C[r] \rightleftharpoons C[s]$ **with** $r \rightleftharpoons s$: Straightforward induction on the structure of $C[\cdot]$. For example, let $C[\cdot] = C'[\cdot] \times t$, then $P(C[r]) = 1 + P(C'[r]) + P(t)$, which, by the induction hypothesis, is equal to $1 + P(C'[s]) + P(t) = P(C[s])$. \square

Lemma 3.2. If $r \rightleftharpoons s$ then $M(r) = M(s)$.

Proof. By structural induction on relation \rightleftharpoons .

$$r \times s \rightleftharpoons s \times r: M(r \times s) = M(r) + M(s) = M(s \times r).$$

$$(r \times s) \times t \rightleftharpoons r \times (s \times t): M((r \times s) \times t) = M(r) + M(s) + M(t) = M(r \times (s \times t)).$$

$$\begin{aligned} \lambda x.(r \times s) \rightleftharpoons \lambda x.r \times \lambda x.s: M(\lambda x.(r \times s)) & \\ &= 2 + M(r) + M(s) + P(r) + P(s) \\ &= M(\lambda x.r \times \lambda x.s) \end{aligned}$$

$$\begin{aligned}
(r \times s)t \rightleftharpoons rt \times st: M((r \times s)t) &= M(r \times s) + M(t) + P(r \times s)M(t) \\
&= M(r) + M(s) + 2M(t) + P(r)M(t) + P(s)M(t) \\
&= M(rt) + M(st) \\
&= M(rt \times st) \\
\pi_{A \Rightarrow B}(\lambda x.r) \rightleftharpoons \lambda x.\pi_B(r): M(\pi_{A \Rightarrow B}(\lambda x.r)) &= 1 + M(\lambda x.r) + P(\lambda x.r) \\
&= 2 + M(r) + 2P(r) \\
&= M(\lambda x.\pi_B(r)) \\
\pi_{A \Rightarrow B}(r)s \rightleftharpoons \pi_B(rs): M(\pi_{A \Rightarrow B}(r)s) &= M(\pi_{A \Rightarrow B}(r)) + M(s) + P(\pi_{A \Rightarrow B}(r))M(s) \\
&= 1 + M(r) + P(r) + M(s) + P(r)M(s) \\
&= 1 + M(rs) + P(rs) \\
&= M(\pi(rs)) \\
(rs)t \rightleftharpoons r(s \times t): M((rs)t) &= M(rs) + M(t) + P(rs)M(t) \\
&= M(r) + M(s) + P(r)M(s) + M(t) + P(r)M(t) \\
&= M(r) + M(s \times t) + P(r)M(s \times t) \\
&= M(r(s \times t))
\end{aligned}$$

$$\pi_A(r) \rightleftharpoons \pi_B(r), \text{ with } A \equiv B: M(\pi_A(r)) = 1 + M(r) + P(r) = M(\pi_B(r)).$$

$C[r] \rightleftharpoons C[s]$ **with** $r \rightleftharpoons s$: Straightforward case by case on the structure of $C[\cdot]$. For example, let $C[\cdot] = \lambda x.C'[\cdot]$, then $M(C[r]) = 1 + M(C'[r]) + P(C'[r])$, which, by the induction hypothesis and Lemma 3.1, is equal to $1 + M(C'[s]) + P(C'[s]) = M(C[s])$.

□

Lemma 3.3. For any term r , the set $\{s \mid s \rightleftharpoons^* r\}$ is finite (modulo α -equivalence and type equivalence).

Proof. As $\{s \mid s \rightleftharpoons^* r\} \subseteq \{s \mid FV(s) = FV(r) \text{ and } M(s) = M(r)\} \subseteq \{s \mid FV(s) \subseteq FV(r) \text{ and } M(s) \leq M(r)\}$ all we need to prove is that for all natural numbers n , for all finite sets of variables F , the set $H(n, F) = \{s \mid FV(s) \subseteq F \text{ and } M(s) \leq n\}$ is finite.

We first prove by induction on s that $M(s) \geq 1$ and then the property by induction on n . For $n = 1$ the set $\{s \mid FV(s) \subseteq F \text{ and } M(s) \leq 1\}$ contains only the variables of F . Assume the property holds for n , then $H(n+1, F)$ is a subset of the finite set containing the variables of F , some abstractions $(\lambda x.r)$ for r in $H(n, F \cup \{x\})$, some applications (rs) for r and s in $H(n, F)$, some products $r \times s$ for r and s in $H(n, F)$, some projections $\pi_A(r)$ for r in $H(n, F)$. □

In Lemma 3.4, we characterise the terms equivalent to a product. The first item of such a lemma is analogous to the seven cases depicted in Fig. 1. In Lemma 3.5, we

characterise the terms equivalent to a projection and, in Lemma 3.6, the terms equivalent to an abstraction.

Lemma 3.4. If $r \times s \rightleftharpoons^* t$ then one of the following cases happen

- 1 $t = u \times v$ where either
 - (a) $u \rightleftharpoons^* t_{11} \times t_{21}$ and $v \rightleftharpoons^* t_{12} \times t_{22}$ with $r \rightleftharpoons^* t_{11} \times t_{12}$ and $s \rightleftharpoons^* t_{21} \times t_{22}$, or
 - (b) $v \rightleftharpoons^* w \times s$ with $r \rightleftharpoons^* u \times w$, or any of the three symmetric cases, or
 - (c) $r \rightleftharpoons^* u$ and $s \rightleftharpoons^* v$, or the symmetric case.
- 2 $t = \lambda x.a$ and $a \rightleftharpoons^* a_1 \times a_2$ with $r \rightleftharpoons^* \lambda x.a_1$ and $s \rightleftharpoons^* \lambda x.a_2$.
- 3 $t = av$ and $a \rightleftharpoons^* a_1 \times a_2$, with $r \rightleftharpoons^* a_1v$ and $s \rightleftharpoons^* a_2v$.

Proof. By a double induction, first on $M(t)$ and then on the length of the relation \rightleftharpoons^* . Consider an equivalence proof $r \times s \rightleftharpoons^* t' \rightleftharpoons t$ with a shorter proof $r \times s \rightleftharpoons^* t'$. By the second induction hypothesis, the term t' has the form prescribed by the lemma. We consider the three cases and in each case, the possible rules transforming t' in t .

- 1 Let $r \times s \rightleftharpoons^* u \times v \rightleftharpoons t$. The possible equivalences from $u \times v$ are
 - $t = u' \times v$ or $u \times v'$ with $u \rightleftharpoons u'$ and $v \rightleftharpoons v'$, and so we are in case 1.
 - Associativity and commutativity preserve the conditions of case 1.
 - $t = \lambda x.(u' \times v')$, with $u = \lambda x.u'$ and $v = \lambda x.v'$. By the first induction hypothesis (since $M(u) < M(t)$ and $M(v) < M(t)$), one of the following cases happen
 - (a) $u \rightleftharpoons^* w_{11} \times w_{21}$ and $v \rightleftharpoons^* w_{12} \times w_{22}$, by the first induction hypothesis, $w_{ij} \rightleftharpoons^* \lambda x.t_{ij}$ for $i = 1, 2$ and $j = 1, 2$, with $u' \rightleftharpoons^* t_{11} \times t_{21}$ and $v' \rightleftharpoons^* t_{12} \times t_{22}$, so $u' \times v' \rightleftharpoons^* t_{11} \times t_{12} \times t_{21} \times t_{22}$. Hence, $r \rightleftharpoons^* \lambda x.(t_{11} \times t_{12})$ and $s \rightleftharpoons^* \lambda x.(t_{21} \times t_{22})$, and hence we are in case 2.
 - (b) $v \rightleftharpoons^* w \times s$ and $r \rightleftharpoons^* u \times w$. Since $v \rightleftharpoons^* \lambda x.v'$, by the first induction hypothesis, $w \rightleftharpoons^* \lambda x.t_1$ and $s \rightleftharpoons^* \lambda x.t_2$, with $v' \rightleftharpoons^* t_1 \times t_2$. Hence, $r \rightleftharpoons^* \lambda x.(u' \times t_1)$, and hence we are in case 2.
 - (c) $r \rightleftharpoons^* \lambda x.u'$ and $s \rightleftharpoons^* \lambda x.v$, and hence we are in case 2.
(the symmetric cases are analogous).
 - $t = (u' \times v')t'$, with $u = u't'$ and $v = v't'$. By the first induction hypothesis (since $M(u) < M(t)$ and $M(v) < M(t)$), one of the following cases happen
 - (a) $u \rightleftharpoons^* w_{11} \times w_{21}$, $v \rightleftharpoons^* w_{12} \times w_{22}$, $r \rightleftharpoons^* w_{11} \times w_{12}$, and $s \rightleftharpoons^* w_{21} \times w_{22}$. By the first induction hypothesis (since $M(w_{ij}) < M(t)$), $w_{ij} \rightleftharpoons^* t_{ij}t'$, for $i = 1, 2$ and $j = 1, 2$, where $u' \rightleftharpoons^* t_{11} \times t_{21}$, $v' \rightleftharpoons^* t_{12} \times t_{22}$, $r \rightleftharpoons^* w_{11} \times w_{12}$ and $s \rightleftharpoons^* w_{21} \times w_{22}$. Therefore, $u \rightleftharpoons^* t_{11}t' \times t_{21}t'$ and $v \rightleftharpoons^* t_{12}t' \times t_{22}t'$ with $r \rightleftharpoons^* (t_{11} \times t_{12})t'$ and $s \rightleftharpoons^* (t_{21} \times t_{22})t'$, and hence we are in case 3.
 - (b) $v't' \rightleftharpoons^* w \times s$ and $r \rightleftharpoons^* u't' \times w$. By the first induction hypothesis on $v't' \rightleftharpoons^* w \times s$ (since $M(w) < M(t)$ and $M(s) < M(t)$), we have $w \rightleftharpoons^* t_1t'$ and $s \rightleftharpoons^* t_2t'$ with $t_1 \times t_2 \rightleftharpoons^* v'$. Therefore, $v \rightleftharpoons^* t_1t' \times t_2t'$ with $r \rightleftharpoons^* (u \times t_1)t'$ and $s \rightleftharpoons^* t_2t'$, and $u' \times v' \rightleftharpoons^* u' \times t_1 \times t_2$, hence we are in case 3.
 - (c) $r \rightleftharpoons^* u't'$ and $s \rightleftharpoons^* v't'$, and hence we are in case 3.

(the symmetric cases are analogous).

- 2 Let $r \times s \rightrightarrows^* \lambda x.a \rightrightarrows t$, with $a \rightrightarrows^* a_1 \times a_2$, $r \rightrightarrows^* \lambda x.a_1$, and $s \rightrightarrows^* \lambda x.a_2$. Hence, possible equivalences from $\lambda x.a$ to t are
- $t = \lambda x.a'$ with $a \rightrightarrows^* a'$, hence $a' \rightrightarrows^* a_1 \times a_2$, and so we are in case 2.
 - $t = \lambda x.u \times \lambda x.v$, with $a_1 \times a_2 \rightrightarrows^* a = u \times v$. Hence, by the first induction hypothesis (since $M(a) < M(t)$), either
 - (a) $a_1 \rightrightarrows^* u$ and $a_2 \rightrightarrows^* v$, and so $r \rightrightarrows^* \lambda x.u$ and $s \rightrightarrows^* \lambda x.v$, or
 - (b) $v \rightrightarrows^* t_1 \times t_2$ with $a_1 \rightrightarrows^* u \times t_1$ and $a_2 \rightrightarrows^* t_2$, and so $\lambda x.v \rightrightarrows^* \lambda x.t_1 \times \lambda x.t_2$, $r \rightrightarrows^* \lambda x.u \times \lambda x.t_1$ and $s \rightrightarrows^* \lambda x.t_2$, or
 - (c) $u \rightrightarrows^* t_{11} \times t_{21}$ and $v \rightrightarrows^* t_{12} \times t_{22}$ with $a_1 \rightrightarrows^* t_{11} \times t_{12}$ and $a_2 \rightrightarrows^* t_{21} \times t_{22}$, and so $\lambda x.u \rightrightarrows^* \lambda x.t_{11} \times \lambda x.t_{21}$, $\lambda x.v \rightrightarrows^* \lambda x.t_{12} \times \lambda x.t_{22}$, $r \rightrightarrows^* \lambda x.t_{11} \times \lambda x.t_{12}$ and $s \rightrightarrows^* \lambda x.t_{21} \times \lambda x.t_{22}$.
- (the symmetric cases are analogous), and so we are in case 1.
- Notice that rule (DIST_{ei}) cannot apply because by the first induction hypothesis, $a \not\rightrightarrows^* \pi_C(t')$.

- 3 Let $r \times s \rightrightarrows^* aw \rightrightarrows t$, with $a \rightrightarrows^* a_1 \times a_2$, $r \rightrightarrows^* a_1w$, and $s \rightrightarrows^* a_2w$. The possible equivalences from aw to t are
- $t = a'w$ with $a \rightrightarrows^* a'$, hence $a' \rightrightarrows^* a_1 \times a_2$, and so we are in case 3.
 - $t = aw'$ with $w \rightrightarrows^* w'$ and so we are in case 3.
 - $t = uw \times vw$, with $a_1 \times a_2 \rightrightarrows^* a = u \times v$. Hence, by the first induction hypothesis (since $M(a) < M(t)$), either
 - (a) $a_1 \rightrightarrows^* u$ and $a_2 \rightrightarrows^* v$, and so $r \rightrightarrows^* uw$ and $s \rightrightarrows^* vw$, or
 - (b) $v \rightrightarrows^* t_1 \times t_2$ with $a_1 \rightrightarrows^* u \times t_1$ and $a_2 \rightrightarrows^* t_2$, and so $vw \rightrightarrows^* t_1w \times t_2w$, $r \rightrightarrows^* uw \times t_1w$ and $s \rightrightarrows^* t_2w$, or
 - (c) $u \rightrightarrows^* t_{11} \times t_{21}$ and $v \rightrightarrows^* t_{12} \times t_{22}$ with $a_1 \rightrightarrows^* t_{11} \times t_{12}$ and $a_2 \rightrightarrows^* t_{21} \times t_{22}$, and so $uw \rightrightarrows^* t_{11}w \times t_{21}w$, $vw \rightrightarrows^* t_{12}w \times t_{22}w$, $r \rightrightarrows^* t_{11}w \times t_{12}w$ and $s \rightrightarrows^* t_{21}w \times t_{22}w$.
- (the symmetric cases are analogous), and so we are in case 1.
- $t = a'(v \times w)$ with $a = a'v$, thus $a'v = a \rightrightarrows^* a_1 \times a_2$. Hence, by the first induction hypothesis, $a' \rightrightarrows^* a'_1 \times a'_2$, with $a_1 \rightrightarrows^* a'_1v$ and $a_2 \rightrightarrows^* a'_2v$. Therefore, $r \rightrightarrows^* a'_1(v \times w)$ and $s \rightrightarrows^* a'_2(v \times w)$, and so we are in case 3.
- Notice that rule (DIST_{ee}) cannot apply because by the first induction hypothesis, $a \not\rightrightarrows^* \pi_C(t')$. \square

Lemma 3.5. If $\pi_A(r) \rightrightarrows^* s$, then one of the following cases happens:

- 1 $s = \pi_B(t)$, where $r \rightrightarrows^* t$ and $A \equiv B$.
- 2 $s = \lambda x^B.t$, where $t \rightrightarrows^* \pi_C(u)$, $r \rightrightarrows^* \lambda x^B.u$, and $A \equiv B \Rightarrow C$.
- 3 $s = tu$, where $t \rightrightarrows^* \pi_C(v)$ and $r \rightrightarrows^* vu$, with $C \equiv B \Rightarrow A$.

Proof. By a double induction, first on $M(s)$ and then on the length of the relation \rightrightarrows^* . Consider an equivalence proof $\pi_A(r) \rightrightarrows^* s' \rightrightarrows s$ with a shorter proof $\pi_A(r) \rightrightarrows^* s'$.

By the second induction hypothesis, the term s' has the form prescribed by the lemma. We consider the three cases and in each case, the possible rules transforming s' in s .

- If $s' = \pi_B(t')$, where $r \rightleftarrows^* t'$ and $A \equiv B$, then the possible equivalences from $\pi_B(t')$ to s' are
 - $s = \pi_B(t)$, with $r \rightleftarrows^* t' \rightleftarrows t$, hence we are in case 1.
 - $s = \pi_C(r)$, with $A \equiv B \equiv C$, hence we are in case 1.
 - $s = \lambda x^C.\pi_D(u)$, with $r \rightleftarrows^* t' = \lambda x^C.u$ and $A \equiv B \equiv C \Rightarrow D$, hence, we are in case 2.
 - $s = \pi_{C \Rightarrow B}(v)u$, with $r \rightleftarrows^* t' = vu$, $C \Rightarrow A \equiv C \Rightarrow B$, hence we are in case 3.
- If $s' = \lambda x^B.t'$, where $t' \rightleftarrows^* \pi_C(u)$, $r \rightleftarrows^* \lambda x^B.u$, and $A \equiv B \Rightarrow C$. Hence, by the first induction hypothesis (since $M(t') < M(s)$), the possible equivalences from $\lambda x^B.t'$ are
 - $s = \lambda x^B.t$, with $r \rightleftarrows^* t' \rightleftarrows t$, hence we are in case 2.
 - $s = \pi_{B \Rightarrow C}(\lambda x^B.u')$, where $t' = \pi_C(u') \rightleftarrows^* \pi_C(u)$, hence, since $M(t') < M(s)$, we have $u \rightleftarrows^* u'$, and so $r \rightleftarrows^* \lambda x^B.u'$, hence we are in case 1.
- If $s' = t'u'$, where $t' \rightleftarrows^* \pi_C(v')$ and $r \rightleftarrows^* v'u'$, with $C \equiv B \Rightarrow A$, then the possible equivalences from $t'u'$ are
 - $s = tu'$ where $\pi_C(v') \rightleftarrows^* t' \rightleftarrows t$ and $r \rightleftarrows^* v'u'$, with $C \equiv B \Rightarrow A$, hence we are in case 3.
 - $s = t'u$ where $u' \rightleftarrows u$, $t' \rightleftarrows^* \pi_C(v')$, $r \rightleftarrows^* v'u' \rightleftarrows v'u$, with $C \equiv B \Rightarrow A$, hence we are in case 3.
 - $s = \pi_D(vu')$, where $t' = \pi_C(v)$ with $C \equiv F \Rightarrow D$, hence, since $M(t') < M(s)$, we have $v \rightleftarrows^* v'$, and so $r \rightleftarrows^* v'u' \rightleftarrows vu'$, and by Lemma 4.2, $D \equiv A$. Hence, we are in case 1.
 - $s = t_1(t_2 \times u')$, with $t' = t_1t_2 \rightleftarrows^* \pi_C(v')$. Since $M(t') < M(s)$, we have $t_1 \rightleftarrows^* \pi_D(v_1)$ and $v' \rightleftarrows^* v_1t_2$, with $D \equiv E \Rightarrow C \equiv E \Rightarrow B \Rightarrow A \equiv (E \wedge B) \Rightarrow A$, hence we are in case 3. \square

Lemma 3.6. If $\lambda x^A.r \rightleftarrows^* s$ then one of the following cases happens:

- 1 $s = \lambda x^A.u$ where $u \rightleftarrows^* r$.
- 2 $s = s_1 \times s_2$ where $s_1 \rightleftarrows^* \lambda x^A.r_1$, $s_2 \rightleftarrows^* \lambda x^A.r_2$ and $r \rightleftarrows^* r_1 \times r_2$.
- 3 $s = \pi_C(u)$ where $C \equiv A \Rightarrow B$, $u \rightleftarrows^* \lambda x^A.v$, and $r \rightleftarrows^* \pi_B(v)$.

Proof. By a double induction, first on $M(s)$ and then on the length of the relation \rightleftarrows^* . Consider an equivalence proof $\lambda x^A.r \rightleftarrows^* s' \rightleftarrows s$ with a shorter proof $\lambda x^A.r \rightleftarrows^* s'$. By the second induction hypothesis, the term s' has the form prescribed by the lemma. We consider the three cases and in each case, the possible rules transforming s' in s .

- If $s' = \lambda x^A.u$ where $u \rightleftarrows^* r$, then the possible equivalences from $\lambda x^A.u$ are
 - $s = \lambda x^A.u'$ with $r \rightleftarrows^* u \rightleftarrows u'$, hence we are in case 1.
 - $s = \lambda x^A.u_1 \times \lambda x^A.u_2$, with $u = u_1 \times u_2$, then take $s_1 = \lambda x^A.u_1$, $s_2 = \lambda x^A.u_2$, and we have $r \rightleftarrows^* u \rightleftarrows^* u_1 \times u_2$, so we are in case 2.
 - $s = \pi_C(\lambda x^A.v)$ with $C \equiv A \Rightarrow B$, $r \rightleftarrows^* u = \pi_B(v)$, hence we are in case 3.

- If $s' = s'_1 \times s'_2$ where $s'_1 \xleftrightarrow{*} \lambda x^A.r_1$, $s'_2 \xleftrightarrow{*} \lambda x^A.r_2$ and $r \xleftrightarrow{*} r_1 \times r_2$. Hence, by the first induction hypothesis (since $M(s'_1) < M(s)$ and $M(s'_2) < M(s)$), the possible equivalences from $s'_1 \times s'_2$ are
 - $s = s_1 \times s'_2$ with $\lambda x^A.r_1 \xleftrightarrow{*} s'_1 \xleftrightarrow{*} s_1$, so we are in case 2.
 - $s = s'_1 \times s_2$ with $\lambda x^A.r_2 \xleftrightarrow{*} s'_2 \xleftrightarrow{*} s_2$, so we are in case 2.
 - $s = s_2 \times s_1$, with $r \xleftrightarrow{*} r_1 \times r_2 \xleftrightarrow{*} r_2 \times r_1$, so we are in case 2.
 - $s = s_1 \times (s_2 \times s'_2)$, with $\lambda x^A.r_1 \xleftrightarrow{*} s'_1 = s_1 \times s_2$. Then, since $M(s'_1) < M(s)$, we have $r_1 \xleftrightarrow{*} r_{11} \times r_{12}$, and so $s_1 \xleftrightarrow{*} \lambda x^A.r_{11}$ and $s_2 \times s'_2 \xleftrightarrow{*} \lambda x^A.(r_{12} \times r_2)$. Hence, we are in case 2.
 - $s = \lambda x^A.(r_1 \times r_2)$, so we are in case 1.
- If $s' = \pi_C(u')$ where $C \equiv A \Rightarrow B$, $u' \xleftrightarrow{*} \lambda x^A.v$ and $r \xleftrightarrow{*} \pi_B(v)$. Hence, by the first induction hypothesis, (since $M(u') < M(s)$), the possible equivalences from $\pi_C(u')$ are
 - $s = \pi_C(u)$, where $\lambda x^A.v \xleftrightarrow{*} u' \xleftrightarrow{*} u$, then we are in case 3.
 - $s = \pi_D(u')$, where $A \Rightarrow B \equiv C \equiv D$, then we are in case 3.
 - $s = \lambda x^A.\pi_B(v)$, hence we are in case 1. □

3.3. Reduction relation

The reduction relation \hookrightarrow is given in Table 4. It contains a β reduction rule and a projection rule π for pairs.

We write \hookrightarrow^* for the transitive and reflexive closure of \hookrightarrow . We write \rightsquigarrow for the relation \hookrightarrow modulo $\xleftrightarrow{*}$ (i.e. $r \rightsquigarrow s$ iff $r \xleftrightarrow{*} r' \hookrightarrow s' \xleftrightarrow{*} s$), and \rightsquigarrow^* for its reflexive and transitive closure.

In Lemma 3.7, we characterise the reducts of a product.

Lemma 3.7. If $r_1 \times r_2 \xleftrightarrow{*} s \hookrightarrow t$, there exists u_1, u_2 such that $t \xleftrightarrow{*} u_1 \times u_2$ and either $(r_1 \rightsquigarrow u_1$ and $r_2 \rightsquigarrow u_2)$, or $(r_1 \rightsquigarrow u_1$ and $r_2 \xleftrightarrow{*} u_2)$, or $(r_1 \xleftrightarrow{*} u_1$ and $r_2 \rightsquigarrow u_2)$.

Proof. By induction on $M(r_1 \times r_2)$. By Lemma 3.4, s is either a product, an abstraction or an application with the conditions given in the lemma. The different terms s reducible by \hookrightarrow are

- $(\lambda x.a)s'$ that reduces by the (β) rule to $a[s'/x]$.
- $s_1 \times s_2$, $\lambda x.a$, as' , with a reduction in the subterm s_1 , s_2 , a , or s' .

Notice that rule (π) cannot apply since $s \not\xleftrightarrow{*} \pi_C(s')$.

We consider each case:

$\text{If } s : A, (\lambda x^A.r)s \hookrightarrow r[s/x] \qquad (\beta)$
$\text{If } r : A, \pi_A(r \times s) \hookrightarrow r \qquad (\pi)$
$\frac{r \hookrightarrow s}{C[r] \hookrightarrow C[s]} \qquad (C\hookrightarrow)$

Table 4. Reduction relation.

- $s = (\lambda x.a)s'$ and $t = a[s'/x]$. Using twice Lemma 3.4, we have $a \rightleftarrows^* a_1 \times a_2$, $r_1 \rightleftarrows^* (\lambda x.a_1)s'$ and $r_2 \rightleftarrows^* (\lambda x.a_2)s'$. Since $t \rightleftarrows^* a_1[s'/x] \times a_2[s'/x]$, we take $u_1 = a_1[s'/x]$ and $u_2 = a_2[s'/x]$.
- $s = s_1 \times s_2$, $t = t_1 \times s_2$ or $t = s_1 \times t_2$, with $s_1 \hookrightarrow t_1$ and $s_2 \hookrightarrow t_2$. We only consider the first case since the other is analogous. One of the following cases happen

- (a) $r_1 \rightleftarrows^* w_{11} \times w_{21}$, $r_2 \rightleftarrows^* w_{12} \times w_{22}$, $s_1 = w_{11} \times w_{12}$ and $s_2 = w_{21} \times w_{22}$. Hence, by the induction hypothesis, either $t_1 = w'_{11} \times w_{12}$, or $t_1 = w_{11} \times w'_{12}$, or $t_1 = w'_{11} \times w'_{12}$, with $w_{11} \hookrightarrow w'_{11}$ and $w_{12} \hookrightarrow w'_{12}$. We take, in the first case $u_1 = w'_{11} \times w_{21}$ and $u_2 = w_{12} \times w_{22}$, in the second case $u_1 = w_{11} \times w_{21}$ and $u_2 = w'_{12} \times w_{22}$, and in the third $u_1 = w'_{11} \times w_{21}$ and $u_2 = w'_{12} \times w_{22}$.

- (b) We consider two cases, since the other two are symmetric.

- $r_1 \rightleftarrows^* s_1 \times w$ and $s_2 \rightleftarrows^* w \times r_2$, in which case we take $u_1 = t_1 \times w$ and $u_2 = r_2$.
- $r_2 \rightleftarrows^* w \times s_2$ and $s_1 = r_1 \times w$. Hence, by the induction hypothesis, either $t_1 = r'_1 \times w$, or $t_1 = r_1 \times w'$ or $t_1 = r'_1 \times w'$, with $r_1 \hookrightarrow r'_1$ and $w \hookrightarrow w'$. We take, in the first case $u_1 = r'_1$ and $u_2 = w \times s_2$, in the second case $u_1 = r_1$ and $u_2 = w' \times s_2$, and in the third case $u_1 = r'_1$ and $u_2 = w' \times s_2$.

- (c) $r_1 \rightleftarrows^* s_1$ and $r_2 \rightleftarrows^* s_2$, in which case we take $u_1 = t_1$ and $u_2 = s_2$.

- $s = \lambda x.a$, $t = \lambda x.t'$, and $a \hookrightarrow t'$, with $a \rightleftarrows^* a_1 \times a_2$ and $s \rightleftarrows^* \lambda x.a_1 \times \lambda x.a_2$. Therefore, by the induction hypothesis, there exists u'_1, u'_2 such that either $(a_1 \rightsquigarrow u'_1$ and $a_2 \rightsquigarrow u'_2)$, or $(a_1 \rightleftarrows^* u'_1$ and $a_2 \rightsquigarrow u'_2)$, or $(a_1 \rightsquigarrow u'_1$ and $a_2 \rightleftarrows^* u'_2)$. Therefore, we take $u_1 = \lambda x.u'_1$ and $u_2 = \lambda x.u'_2$.
- $s = as'$, $t = t's'$, and $a \hookrightarrow t'$, with $a \rightleftarrows^* a_1 \times a_2$ and $s \rightleftarrows^* a_1s' \times a_2s'$. Therefore, by the induction hypothesis, there exists u'_1, u'_2 such that either $(a_1 \rightsquigarrow u'_1$ and $a_2 \rightsquigarrow u'_2)$, or $(a_1 \rightleftarrows^* u'_1$ and $a_2 \rightsquigarrow u'_2)$, or $(a_1 \rightsquigarrow u'_1$ and $a_2 \rightleftarrows^* u'_2)$. Therefore, we take $u_1 = u'_1s'$ and $u_2 = u'_2s'$.
- $s = as'$, $t = at'$, and $s' \hookrightarrow t'$, with $a \rightleftarrows^* a_1 \times a_2$ and $s \rightleftarrows^* a_1s' \times a_2s'$. By Lemma 3.4 several times, one the following cases happen

- (a) $a_1s' \rightleftarrows^* w_{11}s' \times w_{12}s'$, $a_2s' \rightleftarrows^* w_{21}s' \times w_{22}s'$, $r_1 \rightleftarrows^* w_{11}s' \times w_{21}s'$ and $r_2 \rightleftarrows^* w_{12}s' \times w_{22}s'$. We take $u_1 \rightleftarrows^* (w_{11} \times w_{21})t'$ and $r_2 \rightleftarrows^* (w_{12} \times w_{22})t'$.

- (b) $a_2s' \rightleftarrows^* w_1s' \times w_2s'$, $r_1 \rightleftarrows^* a_1s' \times w_2s'$ and $r_2 \rightleftarrows^* w_2s'$. So we take $u_1 = (a_1 \times w_1)t'$ and $u_2 = w_2t'$, the symmetric cases are analogous.

- (c) $r_1 \rightleftarrows^* a_1s'$ and $r_2 \rightleftarrows^* a_2s'$, in which case we take $u_1 = a_1t'$ and $u_2 = a_2t'$ the symmetric case is analogous. \square

3.4. Examples

Example 3.8. Let $s : A$ and $t : B$. Then,

$$\frac{\frac{\frac{\overline{x^{A \wedge B} : A \wedge B}^{(ax)}}{\lambda x^{A \wedge B}.x : (A \wedge B) \Rightarrow (A \wedge B)}^{(\Rightarrow_i)}}{\lambda x^{A \wedge B}.x : A \Rightarrow B \Rightarrow (A \wedge B)}^{(\equiv)}}{s : A}^{(\Rightarrow_e)} \quad \frac{\frac{(\lambda x^{A \wedge B}.x)s : B \Rightarrow (A \wedge B)}{(\lambda x^{A \wedge B}.x)s : (B \Rightarrow A) \wedge (B \Rightarrow B)}^{(\equiv)}}{\pi_{B \Rightarrow A}((\lambda x^{A \wedge B}.x)s) : B \Rightarrow A}^{(\wedge_e)} \quad t : B}{\pi_{B \Rightarrow A}((\lambda x^{A \wedge B}.x)s)t : A}^{(\Rightarrow_e)}$$

This term reduces as follows:

$$\begin{aligned} \pi_{B \Rightarrow A}((\lambda x^{A \wedge B}.x)s)t &\rightleftharpoons \pi_A((\lambda x^{A \wedge B}.x)st) \\ &\rightleftharpoons \pi_A((\lambda x^{A \wedge B}.x)(s \times t)) \\ &\hookrightarrow \pi_A(s \times t) \\ &\hookrightarrow s \end{aligned}$$

or, equivalently, as follows:

$$\begin{aligned} \pi_{B \Rightarrow A}((\lambda x^{A \wedge B}.x)s)t &\rightleftharpoons \pi_{A \Rightarrow B \Rightarrow A}(\lambda x^{A \wedge B}.x)st \\ &\rightleftharpoons \pi_{(A \wedge B) \Rightarrow A}(\lambda x^{A \wedge B}.x)st \\ &\rightleftharpoons \lambda x^{A \wedge B} \pi_A(x)st \\ &\rightleftharpoons \lambda x^{A \wedge B} \pi_A(x)(s \times t) \\ &\hookrightarrow \pi_A(s \times t) \\ &\hookrightarrow s \end{aligned}$$

Example 3.9. Let $r : A$ and $s : B$. Then $(\lambda x^A.\lambda y^B.x)(r \times s) : A$ and

$$(\lambda x^A.\lambda y^B.x)(r \times s) \rightleftharpoons (\lambda x^A.\lambda y^B.x)rs \hookrightarrow^* r$$

However, if $A \equiv B$, it is also possible to reduce in the following way

$$(\lambda x^A.\lambda y^B.x)(r \times s) \rightleftharpoons (\lambda x^A.\lambda y^A.x)(r \times s) \rightleftharpoons (\lambda x^A.\lambda y^A.x)(s \times r) \rightleftharpoons (\lambda x^A.\lambda y^A.x)sr \hookrightarrow^* s$$

Hence, the classical encoding of the projector also behaves non-deterministically.

Example 3.10. Let $s : A$ and $t : B$, and let $\mathbf{TF} = \lambda x^A.\lambda y^B.(x \times y)$. Then

$$\frac{\frac{\frac{x^A : A}{(ax)} \quad \frac{y^B : B}{(ax)}}{x^A \times y^B : A \wedge B} (\wedge_i)}{\lambda y^B.(x^A \times y) : B \Rightarrow (A \wedge B)} (\Rightarrow_i)}{\mathbf{TF} : A \Rightarrow B \Rightarrow (A \wedge B)} (\equiv)}{\frac{\mathbf{TF} : (A \Rightarrow B \Rightarrow A) \wedge (A \Rightarrow B \Rightarrow B)} (\wedge_e)}{\frac{\pi_{A \Rightarrow B \Rightarrow A}(\mathbf{TF}) : A \Rightarrow B \Rightarrow A}{\pi_{A \Rightarrow B \Rightarrow A}(\mathbf{TF})s : B \Rightarrow A} \quad s : A} (\Rightarrow_e)}{\frac{\pi_{A \Rightarrow B \Rightarrow A}(\mathbf{TF})s : B \Rightarrow A \quad t : B}{\pi_{A \Rightarrow B \Rightarrow A}(\mathbf{TF})st : A} (\Rightarrow_e)}$$

This term reduces as follows:

$$\pi_{A \Rightarrow B \Rightarrow A}(\mathbf{TF})st \rightleftharpoons \pi_{B \Rightarrow A}(\mathbf{TF}s)t \rightleftharpoons \pi_A(\mathbf{TF}st) \hookrightarrow^* \pi_A(s \times t) \hookrightarrow s$$

Example 3.11. Let $\mathbf{T} = \lambda x^A.\lambda y^B.x$ and $\mathbf{F} = \lambda x^A.\lambda y^B.y$. Then

$$\frac{\frac{\mathbf{T} : A \Rightarrow B \Rightarrow A \quad \mathbf{F} : A \Rightarrow B \Rightarrow B}{\mathbf{T} \times \mathbf{F} : (A \Rightarrow B \Rightarrow A) \wedge (A \Rightarrow B \Rightarrow B)} (\wedge_i)}{\mathbf{T} \times \mathbf{F} \times \mathbf{TF} : ((A \Rightarrow B \Rightarrow A) \wedge (A \Rightarrow B \Rightarrow B)) \wedge ((A \Rightarrow B \Rightarrow A) \wedge (A \Rightarrow B \Rightarrow B))} (\wedge_i)$$

So, $\pi_{(A \Rightarrow B \Rightarrow A) \wedge (A \Rightarrow B \Rightarrow B)}(\mathbf{T} \times \mathbf{F} \times \mathbf{TF})$ is well typed and reduces non-deterministically either to $\mathbf{T} \times \mathbf{F}$ or to \mathbf{TF} . Moreover, notice that $\mathbf{T} \times \mathbf{F}$ and \mathbf{TF} are equivalent. Hence, in this very particular case, the non-deterministic choice does not play any role. We will come back to the encoding of booleans in System I on Section 7.2.

4. Subject Reduction

System I has the subject reduction property, that is, the set of types assigned to a term is preserved under \rightleftharpoons and \hookrightarrow . Before proving such a property, we need to prove the unicity of types (Lemma 4.1), a generation lemma (Lemma 4.2), and a substitution lemma (Lemma 4.3).

The following lemma states that a term can be typed only by equivalent types.

Lemma 4.1 (Unicity). If $r : A$ and $r : B$, then $A \equiv B$.

Proof.

- If the last rule of the derivation of $r : A$ is (\equiv) , then we have a shorter derivation of $r : C$ with $C \equiv A$, and, by the induction hypothesis, $C \equiv B$, hence $A \equiv B$.
- If the last rule of the derivation of $r : B$ is (\equiv) we proceed in the same way.
- All the remaining cases are syntax directed. □

Lemma 4.2 (Generation).

- 1 If $x^A : B$, then $A \equiv B$.
- 2 If $\lambda x^A.r : B$, then $B \equiv A \Rightarrow C$ and $r : C$.
- 3 If $rs : B$, then $r : A \Rightarrow B$ and $s : A$.
- 4 If $r \times s : A$, then $A \equiv B \wedge C$ with $r : B$ and $s : C$.

5 If $\pi_A(r) : B$, then $A \equiv B$ and $r : B \wedge C$.

Proof. Each of the statement is proved by induction on the typing derivation. For the statement 1, we have $x^A : B$. The only way to type this term is either by the rule (ax) or (\equiv) .

— In the first case, we have $A = B$.

— In the second case, there exists B' such that $x^A : B'$ has a shorter derivation, and $B \equiv B'$. By the induction hypothesis $A \equiv B' \equiv B$.

For the statement 2, we have $\lambda x^A.r : B$. The only way to type this term is either by rule (\Rightarrow_i) , (\equiv) .

— In the first case, we have $B = A \Rightarrow C$ for some, C and $r : C$.

— In the second, there exists B' such that $\lambda x^A.r : B'$ has a shorter derivation, and $B \equiv B'$. By the induction hypothesis, $B' \equiv A \Rightarrow C$ and $r : C$. Thus, $B \equiv B' \equiv A \Rightarrow C$.

The three other statements are similar. \square

Lemma 4.3 (Substitution). If $r : A$ and $s : B$, then $r[s/x^B] : A$.

Proof. By structural induction on r .

— Let $r = x$. By Lemma 4.2, $A \equiv B$, thus $s : A$. We have $x[s/x^B] = s$, so $x[s/x^B] : A$.

— Let $r = y$, with $y \neq x$. We have $y[s/x^B] = y$, so $y[s/x^B] : A$.

— Let $r = \lambda y^C.r'$. By Lemma 4.2, $A \equiv C \Rightarrow D$, with $r' : D$. By the induction hypothesis $r'[s/x^B] : D$, and so, by rule (\Rightarrow_i) , $\lambda y^C.r'[s/x^B] : C \Rightarrow D$. Since $\lambda y^C.r'[s/x^B] = (\lambda y^C.r')[s/x^B]$, using rule (\equiv) , $(\lambda y^C.r')[s/x^B] : A$.

— Let $r = r_1 r_2$. By Lemma 4.2, $r_1 : C \Rightarrow A$ and $r_2 : C$. By the induction hypothesis $r_1[s/x^B] : C \Rightarrow A$ and $r_2[s/x^B] : C$, and so, by rule (\Rightarrow_e) , $(r_1[s/x^B])(r_2[s/x^B]) : A$. Since $(r_1[s/x^B])(r_2[s/x^B]) = (r_1 r_2)[s/x^B]$, we have $(r_1 r_2)[s/x^B] : A$.

— Let $r = r_1 \times r_2$. By Lemma 4.2, $r_1 : A_1$ and $r_2 : A_2$, with $A \equiv A_1 \wedge A_2$. By the induction hypothesis $r_1[s/x^B] : A_1$ and $r_2[s/x^B] : A_2$, and so, by rule (\wedge_i) , $(r_1[s/x^B]) \times (r_2[s/x^B]) : A_1 \wedge A_2$. Since $(r_1[s/x^B]) \times (r_2[s/x^B]) = (r_1 \times r_2)[s/x^B]$, using rule (\equiv) , we have $(r_1 \times r_2)[s/x^B] : A$.

— Let $r = \pi_A(r')$. By Lemma 4.2, $r' : A \wedge C$. Hence, by the induction hypothesis, $r'[s/x^B] : A \wedge C$. Hence, by rule \wedge_e , $\pi_A(r'[s/x^B]) : A$. Since $\pi_A(r'[s/x^B]) = \pi_A(r')[s/x^B]$, we have $\pi_A(r')[s/x^B] : A$. \square

Theorem 4.4 (Subject reduction). If $r : A$ and $r \hookrightarrow s$ or $r \rightleftharpoons s$ then $s : A$.

Proof. By induction on the rewrite relation.

$r \times s \rightleftharpoons s \times r$: If $r \times s : A$, then by Lemma 4.2, $A \equiv A_1 \wedge A_2 \equiv A_2 \wedge A_1$, with $r : A_1$ and $s : A_2$. Then,

$$\frac{\frac{s : A_2 \quad r : A_1}{s \times r : A_2 \wedge A_1} (\wedge_i)}{s \times r : A} (\equiv)$$

$(r \times s) \times t \rightleftharpoons r \times (s \times t)$:

(\rightarrow) If $(r \times s) \times t : A$, then by Lemma 4.2, $A \equiv (A_1 \wedge A_2) \wedge A_3 \equiv A_1 \wedge (A_2 \wedge A_3)$, with $r : A_1$, $s : A_2$ and $t : A_3$. Then,

$$\frac{\frac{\frac{s : A_2 \quad t : A_3}{s \times t : A_2 \wedge A_3} (\wedge_i)}{r : A_1 \quad s \times t : A_2 \wedge A_3} (\wedge_i)}{r \times (s \times t) : A_1 \wedge (A_2 \wedge A_3)} (\wedge_i) \quad (\equiv) \\ \frac{}{r \times (s \times t) : A} (\equiv)$$

(\leftarrow) Analogous to (\rightarrow).

$\lambda x^B.(r \times s) \rightleftharpoons \lambda x^B.r \times \lambda x^B.s$:

(\rightarrow) If $\lambda x^B.(r \times s) : A$, then by Lemma 4.2, $A \equiv (B \Rightarrow (C_1 \wedge C_2)) \equiv ((B \Rightarrow C_1) \wedge (B \Rightarrow C_2))$, with $r : C_1$ and $s : C_2$. Then,

$$\frac{\frac{\frac{r : C_1}{\lambda x^B.r : B \Rightarrow C_1} (\Rightarrow_i) \quad \frac{s : C_2}{\lambda x^B.s : B \Rightarrow C_2} (\Rightarrow_i)}{\lambda x^B.r \times \lambda x^B.s : (B \Rightarrow C_1) \wedge (B \Rightarrow C_2)} (\wedge_i)}{\lambda x^B.r \times \lambda x^B.s : A} (\equiv)$$

(\leftarrow) If $\lambda x^B.r \times \lambda x^B.s : A$, then by Lemma 4.2, $A \equiv ((B \Rightarrow C_1) \wedge (B \Rightarrow C_2)) \equiv (B \Rightarrow (C_1 \wedge C_2))$, with $r : C_1$ and $s : C_2$. Then,

$$\frac{\frac{\frac{r : C_1 \quad s : C_2}{r \times s : C_1 \wedge C_2} \wedge_i}{\lambda x^B.(r \times s) : B \Rightarrow (C_1 \wedge C_2)} (\Rightarrow_i)}{\lambda x^B.(r \times s) : A} (\equiv)$$

$(r \times s)t \rightleftharpoons rt \times st$:

(\rightarrow) If $(r \times s)t : A$, then by Lemma 4.2, $r \times s : B \Rightarrow A$, and $t : B$. Hence, by Lemma 4.2 again, $B \Rightarrow A \equiv C_1 \wedge C_2$, and so by Lemma 2.7, $A \equiv A_1 \wedge A_2$, with $r : B \Rightarrow A_1$ and $s : B \Rightarrow A_2$. Then,

$$\frac{\frac{\frac{r : B \Rightarrow A_1 \quad t : B}{rt : A_1} (\Rightarrow_e) \quad \frac{s : B \Rightarrow A_2 \quad t : B}{st : A_2} (\Rightarrow_e)}{rt \times st : A_1 \wedge A_2} (\wedge_i)}{rt \times st : A} (\equiv)$$

(\leftarrow) If $rt \times st : A$, then by Lemma 4.2, $A \equiv A_1 \wedge A_2$ with $r : B \Rightarrow A_1$, $s : B' \Rightarrow A_2$, $t : B$ and $t : B'$. By Lemma 4.1, $B \equiv B'$. Then

$$\frac{\frac{\frac{\frac{s : B' \Rightarrow A_2}{s : B \Rightarrow A_2} (\equiv)}{r : B \Rightarrow A_1 \quad s : B \Rightarrow A_2} (\wedge_i)}{r \times s : (B \Rightarrow A_1) \wedge (B \Rightarrow A_2)} (\wedge_i)}{r \times s : B \Rightarrow (A_1 \wedge A_2)} (\equiv) \quad t : B \quad (\Rightarrow_e) \\ \frac{}{(r \times s)t : A_1 \wedge A_2} (\equiv) \quad (\Rightarrow_e) \\ \frac{}{(r \times s)t : A} (\equiv)$$

$\pi_{B \Rightarrow C}(\lambda x^B.r) \rightleftharpoons \lambda x^B.\pi_C(r)$:

(\rightarrow) If $\pi_{B \Rightarrow C}(\lambda x^B.r) : A$, then by Lemma 4.2, $A \equiv (B \Rightarrow C)$ and $\lambda x^B.r : (B \Rightarrow C) \wedge D$.

Thus, by Lemma 4.2, $(B \Rightarrow C) \wedge D \equiv B \Rightarrow F$, and $r : F$. Hence, by Lemma 2.9, $D \equiv B \Rightarrow D'$, and $F \equiv C \wedge D'$. Hence, $r : C \wedge D'$. Therefore, by rule (\wedge_e) , $\pi_C(r) : C$, so

$$\frac{\frac{\pi_C(r) : C}{\lambda x^B . \pi_C(r) : B \Rightarrow C} (\Rightarrow_i)}{\lambda x^B . \pi_C(r) : A} (\equiv)$$

(\Leftarrow) If $\lambda x^B . \pi_C(r) : A$, then by Lemma 4.2, $A \equiv B \Rightarrow C$ and $\pi_C(r) : C$, so by Lemma 4.2 again, $r : C \wedge D$. Hence,

$$\frac{\frac{\frac{r : C \wedge D}{\lambda x^B . r : B \Rightarrow (C \wedge D)} (\equiv)}{\lambda x^B . r : (B \Rightarrow C) \wedge (B \Rightarrow D)} (\Rightarrow_i)}{\frac{\pi_{B \Rightarrow C}(\lambda x^B . r) : B \Rightarrow C}{\pi_{B \Rightarrow C}(\lambda x^B . r) : A} (\equiv)} (\wedge_e)$$

$\pi_{B \Rightarrow C}(r)s \rightleftharpoons \pi_C(rs)$ **with** $r : B \Rightarrow (C \wedge D)$ **and** $s : B$:

(\rightarrow) If $\pi_{B \Rightarrow C}(r)s : A$, then by Lemma 4.1, $A \equiv C$.

$$\frac{\frac{r : B \Rightarrow (C \wedge D) \quad s : B}{rs : C \wedge D} (\Rightarrow_e)}{\frac{\pi_C(rs) : C}{\pi_C(rs) : A} (\equiv)} (\wedge_e)$$

(\Leftarrow) If $\pi_C(rs) : A$, then by Lemma 4.1, $A \equiv C$.

$$\frac{\frac{\frac{r : B \Rightarrow (C \wedge D)}{r : (B \Rightarrow C) \wedge (B \Rightarrow D)} (\equiv)}{\pi_{B \Rightarrow C}(r) : B \Rightarrow C} (\wedge_e)}{\frac{\pi_{B \Rightarrow C}(r)s : C}{\pi_{B \Rightarrow C}(r)s : A} (\equiv)} (\Rightarrow_e) \quad s : B$$

$rst \rightleftharpoons r(s \times t)$:

(\rightarrow) If $rst : A$, then by Lemma 4.2, $r : B \Rightarrow C \Rightarrow A \equiv ((B \wedge C) \Rightarrow A)$, $s : B$ and $t : C$.

Then,

$$\frac{\frac{r : B \Rightarrow C \Rightarrow A}{r : (B \wedge C) \Rightarrow A} (\equiv) \quad \frac{s : B \quad t : C}{s \times t : B \wedge C} (\wedge_i)}{r(s \times t) : A} (\Rightarrow_e)$$

(\Leftarrow) If $r(s \times t) : A$, then by Lemma 4.2, $r : (B \wedge C) \Rightarrow A \equiv (B \Rightarrow C \Rightarrow A)$, $s : B$ and $t : C$. Then

$$\frac{\frac{r : (B \wedge C) \Rightarrow A}{r : B \Rightarrow C \Rightarrow A} (\equiv) \quad s : B}{rs : C \Rightarrow A} (\Rightarrow_e) \quad t : C}{rst : A} (\Rightarrow_e)$$

$\pi_B(r) \rightleftharpoons \pi_C(r)$ **with** $B \equiv C$: If $\pi_B(r) : A$, then by Lemma 4.2, $C \equiv B \equiv A$ and $r : A \wedge D$.

Then

$$\frac{\frac{\frac{r : A \wedge D}{r : C \wedge D} (\equiv)}{\pi_C(r) : C} (\wedge_e)}{\pi_C(r) : A} (\equiv)$$

$(\lambda x^B.r)s \hookrightarrow r[s/x]$ **with** $s : B$: If $(\lambda x^B.r)s : A$, then by Lemma 4.2, $\lambda x^B.r : B \Rightarrow A$, and by Lemma 4.2 again, $r : A$. Then by Lemma 4.3, $r[s/x^B] : A$.

$\pi_B(r \times s) \hookrightarrow r$ **with** $r : B$: If $\pi_B(r \times s) : A$, then by Lemma 4.2, $A \equiv B$, and so, by rule (\equiv) , $r : A$.

Contextual cases: Let $t \rightarrow r$, where \rightarrow is either \rightrightarrows or \hookrightarrow .

$\lambda x^B.t \rightarrow \lambda x^B.r$: If $\lambda x^B.t : A$, then by Lemma 4.2, $A \equiv (B \Rightarrow C)$ and $t : C$, hence by the induction hypothesis, $r : C$ and so $\lambda x^B.r : B \Rightarrow C \equiv A$.

$ts \rightarrow rs$: If $ts : A$ then by Lemma 4.2, $t : B \Rightarrow A$ and $s : B$, hence by the induction hypothesis, $r : B \Rightarrow A$ and so $rs : A$.

$st \rightarrow st$: If $st : A$ then by Lemma 4.2, $s : B \Rightarrow A$ and $t : B$, hence by the induction hypothesis $r : B$ and so $sr : A$.

$t \times s \rightarrow r \times s$: If $t \times s : A$ then by Lemma 4.2, $A \equiv A_1 \wedge A_2$, $t : A_1$, and $s : A_2$, hence by the induction hypothesis, $r : A_1$ and so $r \times s : A_1 \wedge A_2 \equiv A$.

$s \times t \rightarrow s \times r$: Analogous to previous case.

$\pi_B(t) \rightarrow \pi_B(r)$: If $\pi_B(t) : A$ then by Lemma 4.2, $A \equiv B$ and $t : B \wedge C$, hence by the induction hypothesis $r : B \wedge C$. Therefore, $\pi_B(r) : B \equiv A$. \square

5. Strong Normalisation

Now we prove the strong normalisation of reduction \rightsquigarrow . That is, that every reduction sequence fired from a typed term eventually terminates. The set of strongly normalising terms with respect to reduction \rightsquigarrow is written SN. The size of the longest reduction issued from t is written $|t|$.

5.1. Reducibility

To prove that every term is in SN, we associate, as usual, a set $\llbracket A \rrbracket$ of strongly normalising terms to each type A . A term $r : A$ is said to be reducible if $r \in \llbracket A \rrbracket$. We then prove an adequacy theorem stating that every term is reducible.

In Simply Typed Lambda Calculus we can either define $\llbracket A_1 \Rightarrow A_2 \Rightarrow \dots \Rightarrow A_n \Rightarrow \tau \rrbracket$ as the set of terms r such that for all $s \in \llbracket A_1 \rrbracket$, $rs \in \llbracket A_2 \Rightarrow \dots \Rightarrow A_n \Rightarrow \tau \rrbracket$ or, equivalently, as the set of terms r such that for all $s_i \in \llbracket A_i \rrbracket$, $rs_1 \dots s_n \in \llbracket \tau \rrbracket = \text{SN}$. To prove that a term of the form $\lambda x^A.t$ is reducible, we need to use the so-called CR3 property [Girard et al., 1989], in the first case, and the property that a term whose all one-step reducts are in SN is in SN, in the second case. In System I an introduction can be equivalent to an elimination e.g. $\pi_{A \Rightarrow B}(\lambda x^A.r) \rightrightarrows \lambda x^A.\pi_B(r)$, hence, we cannot define a notion of neutral term and have an equivalent to the CR3 property. Therefore, we use the second definition.

Because, in System I, projection is also an elimination operator, the definition of reducibility should be: for all reducible terms s_i , the term $(\pi_{B_n}(\dots(\pi_{B_1}(r)))s_1 \dots s_n$ is in SN (note that we do not need to consider terms where applications and projections are interleaved, such as $(\pi_B(rs_1))s_2$, because this term is equivalent to $\pi_{C \Rightarrow B}(r)s_1s_2$).

For instance, to prove that the term

$$((\lambda z^\tau . \lambda x^\tau . x)w^\tau) \times ((\lambda z^\tau . \lambda y^\tau . \lambda y'^{\tau \Rightarrow \tau} . y)w^\tau)$$

is reducible, we must prove, among other things, that for all $s_1 \in \llbracket \tau \rrbracket$ and $s_2 \in \llbracket \tau \Rightarrow \tau \rrbracket$, we have

$$\pi_{\tau \Rightarrow (\tau \Rightarrow \tau) \Rightarrow \tau}(((\lambda z^\tau . \lambda x^\tau . x)w^\tau) \times ((\lambda z^\tau . \lambda y^\tau . \lambda y'^{\tau \Rightarrow \tau} . y)w^\tau))s_1s_2 \in \text{SN}$$

The usual proof would proceed by proving that all the one-step reducts of this term are in SN, considering two cases: first the projection and then the reduction in a subterm, for which an induction hypothesis applies. But the situation in System I is more complex, as before reducing one of the subterms, we can move s_1 inside and also move w outside the projection. Fig. 3 shows a fragment of the one-step \rightsquigarrow -reduction tree of this term where the dashed arrows are \rightleftharpoons^* -equivalences and the full arrow \hookrightarrow -reductions.

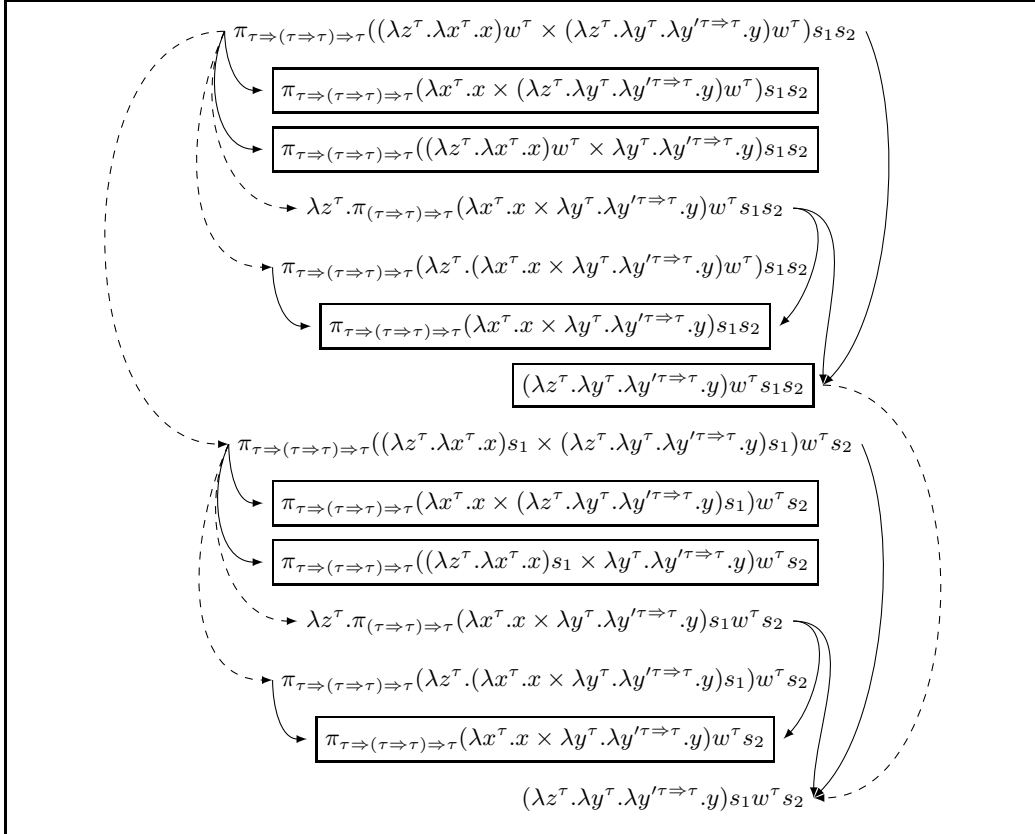


Fig. 3. One-step reductions of $\pi_{\tau \Rightarrow (\tau \Rightarrow \tau) \Rightarrow \tau}(((\lambda z^\tau . \lambda x^\tau . x)w^\tau) \times ((\lambda z^\tau . \lambda y^\tau . \lambda y'^{\tau \Rightarrow \tau} . y)w^\tau))s_1s_2$.

To circumvent this problem, rather than $(\pi_{B_n}(\dots(\pi_{B_1}(r))))s_1 \dots s_n$, we prove the strong normalisation of terms of the form

$$rs_1 \dots s_n$$

Although this choice simplifies the adequacy proof in general, it creates a difficulty in the case of projection. To address this issue, we prove a strengthened adequacy theorem where, instead of using a simple induction, we use a course of values induction on the size $M(r)$ of r .

Finally, the lemma according to which if r and s are in **SN** then $r \times s$ also, that is often trivial, is more challenging in System I, because $r \times s$ may have many \rightrightarrows^* -equivalent terms. For example, a product can be transformed to an application: $rt \times st \rightrightarrows^* (r \times s)t$. Hence, to prove this property, we need Lemma 3.7, which characterise the reducts of a product.

Definition 5.1 (Reducibility). Let A be a type. The set of terms $\llbracket A \rrbracket$ is defined by induction on $m(A)$ as follows:

The term $r \in \llbracket A \rrbracket$ if and only if $r : A$ and for all B_1, \dots, B_n, C such that $A \equiv B_1 \Rightarrow \dots \Rightarrow B_n \Rightarrow C$ where $C \not\equiv C_1 \Rightarrow C_2$, and for all $s_i \in \llbracket B_i \rrbracket$, we have

$$rs_1 \dots s_n \in \mathbf{SN}$$

The following three lemmas are direct consequences from the definition, and the fourth a consequence from Lemma 3.7.

Lemma 5.2 (CR1). For all A , $\llbracket A \rrbracket \subseteq \mathbf{SN}$. □

Lemma 5.3 (CR2). If $r \rightrightarrows^* s$ or $r \rightsquigarrow^* s$, and $r \in \llbracket A \rrbracket$, then $s \in \llbracket A \rrbracket$. □

Lemma 5.4. If $A \equiv B$, then $\llbracket A \rrbracket = \llbracket B \rrbracket$. □

Lemma 5.5. If $r_1 \in \mathbf{SN}$ and $r_2 \in \mathbf{SN}$, then $r_1 \times r_2 \in \mathbf{SN}$.

Proof. By Lemma 3.7, from a reduction sequence starting from $r_1 \times r_2$ we can extract one starting from r_1 , or r_2 or both. Hence, this reduction sequence is finite. □

5.2. Prime terms

Because of rule (CURRY) we shall need in the normalisation proof properties such as u_1, \dots, u_n are reducible if and only if $\prod_{i=1}^n u_i$ is reducible. The direct sense is a consequence of the adequacy of the product, and the converse should be a consequence of the adequacy of projection and Lemma 5.3. Unfortunately, this very property is needed in the proof of the adequacy of the projection. To circumvent this problem, we introduce the notion of prime term as a term which is not equivalent to a product, and we shall prove that if all prime factors of $\prod_{i=1}^n u_i$ are reducible, then u_1, \dots, u_n are product of reducible terms (Lemma 5.13).

Definition 5.6 (Prime term). A term p is prime if there are no terms r_1, r_2 such that $p \rightrightarrows^* r_1 \times r_2$.

Lemma 5.7 (Factorisation). For every term r there exists a multiset of prime terms $[p_1, \dots, p_n]$ such that $r \rightleftharpoons^* \prod_i p_i$.

Proof. By induction on $M(r)$ considering the cases where r is prime and where r is a product. \square

Lemma 5.8 (Simplification). If $r \times s \rightleftharpoons^* r \times t$ then $s \rightleftharpoons^* t$.

Proof. By induction on $M(r)$. We consider the seven cases of the item 1 of Lemma 3.4 (cf. Fig. 1). Two cases are impossible, in two other cases we get $s \rightleftharpoons^* t$ directly, in the three remaining ones we get $s \rightleftharpoons^* t$ as a consequence of the induction hypothesis. \square

Lemma 5.9 (Euclid's lemma). If $r \times s \rightleftharpoons^* t \times p$ where p is a prime term, then either $r \rightleftharpoons^* p$, or $r \rightleftharpoons^* p \times r'$, or $s \rightleftharpoons^* p$, or $s \rightleftharpoons^* p \times r'$.

Proof. As p is prime, only four cases out of the seven of item 1 of Lemma 3.4 are possible. \square

Corollary 5.10. If $p \times r \rightleftharpoons^* \prod_{q \in Q} q$ where Q is a multiset of prime terms. Then, there exists $q' \in Q$ such that $p \rightleftharpoons^* q'$.

Proof. By induction on the number of elements of Q . \square

Definition 5.11 (Equivalence of multisets). We write $[r_1, \dots, r_n] \rightleftharpoons^* [s_1, \dots, s_m]$ if there exists terms t_1, \dots, t_m such that $s_i \rightleftharpoons^* t_i$ and $[t_1, \dots, t_m] = [r_1, \dots, r_n]$.

Lemma 5.12 (Unique factorisation). If P and Q are two multisets of prime terms such that $\prod_{p \in P} p \rightleftharpoons^* \prod_{q \in Q} q$, then $P \rightleftharpoons^* Q$.

Proof. By induction on the number of elements in P . If $P = [p]$, then as p is prime, $Q = [q]$ and $p \rightleftharpoons^* q$. Otherwise, we have $(\prod_{p \in P \setminus \{p'\}} p) \times p' = \prod_{q \in Q} q$. By Corollary 5.10, there exists $q' \in Q$, such that $p' \rightleftharpoons^* q'$. Thus, by Lemma 5.8 $\prod_{p \in P \setminus \{p'\}} p \rightleftharpoons^* \prod_{q \in Q \setminus \{q'\}} q$. We conclude with the induction hypothesis. \square

Now we can prove the lemma announced in the beginning of this section.

Lemma 5.13. If $r \times s \rightleftharpoons^* \prod_{p \in P} p$ where P is a multiset of reducible prime terms, then r is equivalent to a product of reducible terms.

Proof. By Lemma 5.7, $r \rightleftharpoons^* \prod_{r' \in R} r'$ and $s \rightleftharpoons^* \prod_{s' \in S} s'$ where the R and the S are multisets of prime terms. By Lemma 5.12, $R \uplus S \rightleftharpoons^* P$, hence the elements of R are reducible. \square

To be able to use Lemma 5.13, we need some times to restrict the s_i in the definition of reducibility (Definition 5.1) to be prime terms. The next lemma shows that this can be done without lost of generality.

Lemma 5.14. Let $r : A$. If for all B_1, \dots, B_n, C such that $A \equiv B_1 \Rightarrow \dots \Rightarrow B_n \Rightarrow C$ and $C \not\equiv C_1 \Rightarrow C_2$, for all s_i prime such that $s_i \in \llbracket B_i \rrbracket$, we have $rs_1 \dots s_n \in \text{SN}$ then, $r \in \llbracket A \rrbracket$.

Proof. We need to prove that for all t_1, \dots, t_n such that $t_i \in \llbracket B_i \rrbracket$, we have that $rt_1 \dots t_n \in \text{SN}$. By Lemma 5.7, each t_i can be written as $\prod_{j=1}^{m_i} s_{ij}$ with s_{ij} prime. Thus, by Lemma 4.2, $B_i \equiv \bigwedge_{j=1}^{m_i} D_{ij}$, $A \equiv D_{11} \Rightarrow \dots \Rightarrow D_{nm_n} \Rightarrow C$, and $r \in \llbracket D_{11} \Rightarrow \dots \Rightarrow D_{nm_n} \Rightarrow C \rrbracket$. Hence, $rt_1 \dots t_n \stackrel{\text{def}}{=} rs_{11} \dots s_{nm_n} \in \text{SN}$. \square

5.3. Adequacy

We finally prove the adequacy theorem (Theorem 5.22) showing that every typed term is reducible and the strong normalisation theorem (Theorem 5.24) as a consequence of it.

The adequacy of product is a consequence of Lemma 5.5. The adequacy of projection uses the course of values induction scheme. The adequacy of abstraction, as usual, uses the substitution σ from the adequacy theorem. The adequacy of application is a direct consequence of the definition.

Definition 5.15 (Adequate substitution). A substitution σ is adequate if for all x^A , we have $\sigma(x^A) \in \llbracket A \rrbracket$.

Lemma 5.16 (Adequacy of product). If $r \in \llbracket A \rrbracket$ and $s \in \llbracket B \rrbracket$, then $r \times s \in \llbracket A \wedge B \rrbracket$.

Proof. Let $A \wedge B \equiv C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D$ where $D \not\equiv D_1 \Rightarrow D_2$. We must show that for all $t_i \in \llbracket C_i \rrbracket$, we have $(r \times s)t_1 \dots t_n \in \text{SN}$. By Lemma 2.7, $D \equiv D_1 \wedge D_2$, $A \equiv C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D_1$, and $B \equiv C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D_2$. From $r \in \llbracket A \rrbracket$, we deduce $rt_1 \dots t_n \in \text{SN}$. From $s \in \llbracket B \rrbracket$, we deduce $st_1 \dots t_n \in \text{SN}$. Therefore, by Lemma 5.5, we have $rt_1 \dots t_n \times st_1 \dots t_n \stackrel{\text{def}}{=} (r \times s)t_1 \dots t_n \in \text{SN}$. \square

The next lemma characterises the terms equivalent to $\pi_A(ts'_1 \dots s'_m)s_1 \dots s_n$, which is needed for the adequacy of projection.

Lemma 5.17. If $r \stackrel{\text{def}}{=} \pi_A(ts'_1 \dots s'_m)s_1 \dots s_n$ where t is not equivalent to an application, then, either

- $r = \pi_C(t')u_1 \dots u_p$, where $t' \stackrel{\text{def}}{=} tu'_1 \dots u'_q$ and $\prod_k u'_k \times \prod_h u_h \stackrel{\text{def}}{=} \prod_j s'_j \times \prod_i s_i$; or
- $r = (\lambda x^B.t')u_1 \dots u_p$, where $t \stackrel{\text{def}}{=} \lambda x^B.t''$, $t' \stackrel{\text{def}}{=} \pi_C(t'')$, and $\prod_h u_h \stackrel{\text{def}}{=} \prod_j s'_j \times \prod_i s_i$.

Proof. We show that if r is in one of the two cases and $r \stackrel{\text{def}}{=} r'$, then r' is also in one of those cases.

- Let $r = \pi_C(t')u_1 \dots u_p$ with $t' \stackrel{\text{def}}{=} tu'_1 \dots u'_q$, with t is not equivalent to an application.
 - If r' is obtained by an equivalence on one of the u_j , or t' , or by rule (CURRY), then r' is in the same category.
 - If r' is obtained by rule (DIST_{ee}), hence it is in the same category.
 - If r' is obtained by rule (DIST_{ei}), hence, by Lemma 3.6, $q = 0$, thus, r' is in the second category.
 - If r' is obtained by rule (EQUIV), hence it is in the same category.
- Let $r = (\lambda x^B.t')u_1 \dots u_p$, where $t \stackrel{\text{def}}{=} \lambda x^B.t''$, $t' \stackrel{\text{def}}{=} \pi_C(t'')$, and $\prod_h u_h \stackrel{\text{def}}{=} \prod_j s'_j \times \prod_i s_i$.

- If r' is obtained by an equivalence on one of the u_j , or t' , or by rule (CURRY), then r' is in the same category.
- Rule (DIST_{ii}) cannot be applied since, by Lemma 3.5, t' is not equivalent to a product.
- If r' is obtained by applying rule (DIST_{ei}), then $t' = \pi_C(t''')$. By Lemma 3.5, $t''' \rightleftharpoons^* t''$, then $r' \rightleftharpoons^* \pi_D(\lambda x^B.t'')u_1 \dots u_p \rightleftharpoons^* \pi_D(t)u_1 \dots u_p$, where $D \equiv B \Rightarrow C$, hence we are in the first category. \square

Lemma 5.18 (Adequacy of projection). If $r \in \llbracket A \wedge B \rrbracket$ and all the terms t , such that $M(t) \leq M(r)$, are reducible, then $\pi_A(r) \in \llbracket A \rrbracket$.

Proof. Let $A \equiv C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D$ where $D \not\equiv D_1 \Rightarrow D_2$. Using Lemma 5.14, we must show that, for all s_i prime such that $s_i \in \llbracket C_i \rrbracket$, we have $\pi_A(r)s_1 \dots s_n \in \text{SN}$. We prove that all the one-step reducts of this term are in SN. Consider $\pi_A(r)s_1 \dots s_n \rightleftharpoons^* u \hookrightarrow v$ where $r \rightleftharpoons^* ts'_1 \dots s'_m$, with t not equivalent to an application. Then, by Lemma 5.17, the possible u are:

- $\pi_C(t')u_1 \dots u_p$, where $t' \rightleftharpoons^* tu'_1 \dots u'_q$ and $\prod_k u'_k \times \prod_h u_h \rightleftharpoons^* \prod_j s'_j \times \prod_i s_i$. By Lemma 5.7, each s'_j decomposes into products of prime factors. Since these factors have a size smaller than $M(s'_j) < M(r)$, the hypothesis applies and each of them is reducible. Hence, $\prod_k u'_k \times \prod_h u_h$ is equivalent to a product of reducible prime terms. By Lemmas 5.13 and 5.16, each u'_k and u_h is reducible. Since $M(t) \leq M(r)$, t is also reducible, and so $t' \rightleftharpoons^* tu'_1 \dots u'_q \in \text{SN}$. Also, by Lemma 5.16 and 5.2, $\prod_h u_h \in \text{SN}$. We show by induction on $|t'| + |\prod_h u_h|$ that all the one-step reducts of $\pi_C(t')u_1 \dots u_p$ are in SN. The possible reductions are:
 - A projection of $\pi_C(t')$ to t'_1 . Then $t' = t'_1 \times t'_2$. Then, by Lemma 3.4, $t \rightleftharpoons^* t_1 \times t_2$ and $t'_1 \rightleftharpoons^* t_1 u'_1 \dots u'_q$. Since $M(t_1) \leq M(t) \leq M(r)$, it is reducible, and so $t_1 u'_1 \dots u'_q u_1 \dots u_p \rightleftharpoons^* t'_1 u_1 \dots u_p \in \text{SN}$.
 - A reduction on any of t', u_1, \dots, u_p (possibly using rule (CURRY) before reducing), then the induction hypothesis applies.
- $(\lambda x^B.t')u_1 \dots u_p$, where $t \rightleftharpoons^* \lambda x^B.t''$, $t' \rightleftharpoons^* \pi_C(t'')$, and $\prod_h u_h \rightleftharpoons^* \prod_j s'_j \times \prod_i s_i$. Every \hookrightarrow -reduct of this term, is equivalent to a \hookrightarrow -reduct of $\pi_C((\lambda x^B.t'')u_1)u_2 \dots u_p$, which in the previous item has been shown to be in SN. \square

The next lemma characterises the terms equivalent to $r \rightleftharpoons^* (\lambda x^A.p)s_1 \dots s_n$, which is needed for the adequacy of abstraction.

Lemma 5.19. If $r \rightleftharpoons^* (\lambda x^A.p)s_1 \dots s_n$ where p is a prime term, then r is either

- $(\lambda x^A.q)u_1 \dots u_p$, with $\prod_i s_i \rightleftharpoons^* \prod_j u_j$ and $p \rightleftharpoons^* q$, or
- $\pi_{B_1}(\dots(\pi_{B_m}(t')u_{h+1} \dots u_k)\dots)u_{l+1} \dots u_p$, with $B_i \equiv A \Rightarrow C_i$, $\prod_i s_i \rightleftharpoons^* \prod_j u_j$, $t' \rightleftharpoons^* (\lambda x^A.t)u_1 \dots u_h$, and $p \rightleftharpoons^* \pi_{C_1}(\dots \pi_{C_m}(t))$, with $m > 0$.

Proof. Since p is a prime term, the only rules that can be applied are either (CURRY), (DIST_{ei}), (DIST_{ee}), (EQUIV), or equivalences over p, s_1, \dots, s_n . \square

Lemma 5.20 (Adequacy of abstraction). If for all ρ adequate, $\rho r \in \llbracket B \rrbracket$, then for all σ adequate, we have $\lambda x^A . \sigma r \in \llbracket A \Rightarrow B \rrbracket$.

Proof. Let $A \Rightarrow B \equiv C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D$ where $D \not\equiv D_1 \Rightarrow D_2$. Using Lemma 5.14, we need to prove that for all s_i prime such that $s_i \in \llbracket C_i \rrbracket$, we have $(\lambda x^A . \sigma r)_{s_1 \dots s_n} \in \text{SN}$.

By Lemma 5.2 and Lemma 5.5, we have $\prod_{i=1}^n s_i \in \text{SN}$. Also, by Lemma 5.2, we have $\sigma r \in \text{SN}$. We proceed by induction on the lexicographic order of $\langle |\sigma r| + \prod_{i=1}^n |s_i|, M(\sigma r) \rangle$.

- If $\sigma r \rightrightarrows^* r_1 \times r_2$, then we can use rules (DIST_{i_i}) and (DIST_{i_e}) to obtain $(\lambda x^A . r_1)_{s_1 \dots s_n} \times (\lambda x^A . r_2)_{s_1 \dots s_n}$, since $|r_1| \leq |\sigma r|$, $|r_2| \leq |\sigma r|$, $M(r_1) < M(\sigma r)$, and $M(r_2) < M(\sigma r)$, the induction hypothesis applies so $(\lambda x^A . r_1)_{s_1 \dots s_n} \in \text{SN}$ and $(\lambda x^A . r_2)_{s_1 \dots s_n} \in \text{SN}$. So, by Lemma 5.5, $(\lambda x^A . r_1)_{s_1 \dots s_n} \times (\lambda x^A . r_2)_{s_1 \dots s_n} \in \text{SN}$.
- If σr is a prime term, then we prove that all the one-step reductions starting in $(\lambda x^A . \sigma r)_{s_1 \dots s_n}$ are in SN. Consider the reduction $(\lambda x^A . \sigma r)_{s_1 \dots s_n} \rightrightarrows^* u \hookrightarrow v$. Then, by Lemma 5.19, the possible u are:
 - $(\lambda x^A . t')_{u_1 \dots u_p}$, with $\sigma r \rightrightarrows^* t'$ and $\prod_i s_i \rightrightarrows^* \prod_j u_j$. By Lemmas 5.13 and 5.16, the u_j are reducible. Then, v is obtained either by reducing t' or any of u_1, \dots, u_p , in which case the induction hypothesis applies, or, β -reduce obtaining $t'[u_1/x]u_2 \dots u_p \rightrightarrows^* \sigma r[u_1/x]u_2 \dots u_p$. Therefore, $\rho = [u_1/x] \circ \sigma$ is adequate, and since $\rho r \in \llbracket B \rrbracket$, then $\sigma r[u_1/x]u_2 \dots u_p \in \text{SN}$.
 - $\pi_{B_1}(\dots(\pi_{B_m}(t')_{u_{h+1} \dots u_k})\dots)_{u_{l+1} \dots u_p}$, with $B_i \equiv A \Rightarrow C_i$, $\prod_i s_i \rightrightarrows^* \prod_j u_j$, $t' \rightrightarrows^* (\lambda x^A . t)_{u_1 \dots u_h}$, and $p \rightrightarrows^* \pi_{C_1}(\dots \pi_{C_m}(t))$, with $m > 0$. Every \hookrightarrow -reduct of this term, is equivalent to a \hookrightarrow -reduct of $(\lambda x^A . \pi_{C_1}(\dots(\pi_{C_m}(t))_{u_1 \dots u_p})$, which in the previous item has been shown to be in SN. \square

Lemma 5.21 (Adequacy of application). If $r \in \llbracket A \Rightarrow B \rrbracket$, and $s \in \llbracket A \rrbracket$, then $rs \in \llbracket B \rrbracket$.

Proof. Let $B \equiv C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D$ where $D \not\equiv D_1 \Rightarrow D_2$. Hence, since $r \in \llbracket A \Rightarrow B \rrbracket$, we have that for all $s' \in \llbracket A \rrbracket$, $t_i \in \llbracket C_i \rrbracket$, $rs't_1 \dots t_n \in \text{SN}$. Since $s \in \llbracket A \rrbracket$, we have $rst_1 \dots t_n \in \text{SN}$, and so $rs \in \llbracket C_1 \Rightarrow \dots \Rightarrow C_n \Rightarrow D \rrbracket = \llbracket B \rrbracket$. \square

Theorem 5.22 (Adequacy). If $r : A$, then for all σ adequate, we have $\sigma r \in \llbracket A \rrbracket$.

Proof. By course of values induction on $M(r)$.

- If r is a variable x^A , then, since σ is adequate, we have $\sigma r \in \llbracket A \rrbracket$.
- If r is a product $s \times t$, then by Lemma 4.2, $s : B$, $t : C$, and $A \equiv B \wedge C$, then $M(s) < M(r)$ and $M(t) < M(r)$, hence by the induction hypothesis, $\sigma s \in \llbracket B \rrbracket$ and $\sigma t \in \llbracket C \rrbracket$. By Lemma 5.16, $(\sigma s \times \sigma t) \in \llbracket B \wedge C \rrbracket$, hence by Lemma 5.4, $\sigma r \in \llbracket A \rrbracket$.
- If r is a projection $\pi_A(s)$, then by Lemma 4.2, $s : A \wedge B$, and since $M(s) < M(r)$, by the induction hypothesis, $\sigma s \in \llbracket A \wedge B \rrbracket$ and for all terms s' such that $M(s') \leq M(s) < M(r)$, $\sigma s'$ is reducible. By Lemma 5.18, $\pi_A(\sigma s) \in \llbracket A \rrbracket$, hence $\sigma r \in \llbracket A \rrbracket$.
- If r is an abstraction $\lambda x^B . s$, with $s : C$, then by Lemma 4.2, $A \equiv B \Rightarrow C$, then $M(s) < M(r)$, hence by the induction hypothesis, for all $\rho, \rho s \in \llbracket C \rrbracket$. Hence, by Lemma 5.20, $\lambda x^B . \rho s \in \llbracket B \Rightarrow C \rrbracket$, hence, by Lemma 5.4, $\sigma r \in \llbracket A \rrbracket$.

— If r is an application st , then by Lemma 4.2, $s : B \Rightarrow A$ and $t : B$, then since $M(s) < M(r)$ and $M(t) < M(r)$, by the induction hypothesis, $\sigma s \in \llbracket B \Rightarrow A \rrbracket$ and $\sigma t \in \llbracket B \rrbracket$. Hence, by Lemma 5.21, we have $\sigma s \sigma t \in \llbracket A \rrbracket$, so $\sigma r \in \llbracket A \rrbracket$. \square

Lemma 5.23 (Inhabitation). For all $x \in \mathcal{V}_A$, $x \in \llbracket A \rrbracket$.

Proof. Let $A \equiv B_1 \Rightarrow \dots \Rightarrow B_n \Rightarrow C$ where $C \not\equiv C_1 \Rightarrow C_2$. We need to prove that for all $s_i \in \llbracket C_i \rrbracket$, we have $xs_1 \dots s_n \in \text{SN}$. By Lemma 5.5, $\prod_{i=1}^n s_i \in \text{SN}$. Therefore, we proceed by induction on $|\prod_{i=1}^n s_i|$. Each of the reducts of $xs_1 \dots s_n$ is equivalent to xr where $\prod_{i=1}^n s_i \rightsquigarrow r$. Therefore, the induction hypothesis applies. \square

Theorem 5.24 (Strong normalisation). If $r : A$, then $r \in \text{SN}$.

Proof. By Lemma 5.23, the identity substitution is adequate. Thus, by Theorem 5.22 and Lemma 5.2, $r \in \llbracket A \rrbracket \subseteq \text{SN}$. \square

6. Consistency

To prove that there is no closed term of type τ we introduce the notion of simple term, prove that every closed term can be translated to a closed simple term of the same type, and that there is no closed simple term of type τ .

Definition 6.1 (Simple terms). A term is simple if all its variables have a type $A \not\equiv A_1 \wedge A_2$.

Definition 6.2 (Measure N). We introduce a new measure N on terms counting the number of conjunctions in the types of bound variables, as follows.

$$\begin{aligned} N(x) &= 0 \\ N(\lambda x^A.r) &= \#\text{can}(A) - 1 + N(r) \\ N(rs) &= N(r) + N(s) \\ N(r \times s) &= N(r) + N(s) \\ N(\pi_A(r)) &= N(r) \end{aligned}$$

where $\#\text{can}(A)$ is the cardinal of the non-empty multiset $\text{can}(A)$.

Lemma 6.3. $N(\lambda x^{A \wedge B}.r) = 1 + N(\lambda y^A.\lambda z^B.r[y \times z/x])$

Proof. First we prove by induction on $M(r)$ that $N(r) = N(r[y \times z/x])$. The only interesting case is the case $r = x$, in which case $N(x) = 0 = N(y \times z)$.

Then

$$\begin{aligned} N(\lambda x^{A \wedge B}.r) &= \#\text{can}(A) + \#\text{can}(B) - 1 + N(r) \\ &= 1 + \#\text{can}(A) - 1 + \#\text{can}(B) - 1 + N(r[y \times z/x]) \\ &= 1 + N(\lambda y^A.\lambda z^B.r[y \times z/x]) \end{aligned} \quad \square$$

Definition 6.4 (Translation). We define the following translation from closed terms to closed simple terms by induction on the lexicographic order on $\langle N(r), M(r) \rangle$.

$$\begin{aligned} |x| &= x \\ |\lambda x^{A \wedge B}.r| &= |\lambda y^A.\lambda z^B.r[y \times z/x]| \\ \text{If } A \not\equiv A_1 \wedge A_2, |\lambda x^A.r| &= \lambda x^A.|r| \\ |rs| &= |r||s| \\ |r \times s| &= |r| \times |s| \\ |\pi_A(r)| &= \pi_A(|r|) \end{aligned}$$

Lemma 6.5. If $r : A \wedge B$ is a simple normal term, then $r \rightleftharpoons^* r_1 \times r_2$ where $r_1 : A$ and $r_2 : B$.

Proof. We proceed by induction on $M(r)$.

- r cannot be a variable, since it has type $A \wedge B$ and it is simple.
- Let $r \rightleftharpoons^* \lambda x^C.r'$. Then, by Lemma 4.2, $r' : D$ with $C \Rightarrow D \equiv A \wedge B$. So, by Lemma 2.9, $D \equiv D_1 \wedge D_2$, and so, by the induction hypothesis, $r' \rightleftharpoons^* r_1 \times r_2$. Therefore, $\lambda x^C.r' \rightleftharpoons^* \lambda x^C.r_1 \times \lambda x^C.r_2$.
- Let $r \rightleftharpoons^* st$. Then, by Lemma 4.2, $s : C \Rightarrow (A \wedge B)$, so $s : (C \Rightarrow A) \wedge (C \Rightarrow B)$. Therefore, by the induction hypothesis, $s \rightleftharpoons^* s_1 \times s_2$, and so $st \rightleftharpoons^* s_1 t \times s_2 t$.
- Let $r \rightleftharpoons^* r_1 \times r_2$. Then, by Lemma 4.2, $r_1 : C$ and $r_2 : D$, with $A \wedge B \equiv C \wedge D$. By Lemma 2.12 (cf. Fig. 1), either
 - $C \equiv A_1 \wedge B_1$ and $D \equiv A_2 \wedge B_2$, with $A \equiv A_1 \wedge A_2$ and $B \equiv B_1 \wedge B_2$, or
 - $C \equiv A_1$ and $D \equiv A_2 \wedge B$, with $A \equiv A_1 \wedge A_2$, or
 - $D \equiv A_1$ and $C \equiv A_2 \wedge B$, with $A \equiv A_1 \wedge A_2$, or
 - $C \equiv B_1$ and $D \equiv B_2 \wedge A$, with $B \equiv B_1 \wedge B_2$, or
 - $D \equiv B_1$ and $C \equiv B_2 \wedge A$, with $B \equiv B_1 \wedge B_2$, or
 - $C \equiv A$ and $D \equiv B$, or
 - $C \equiv B$ and $D \equiv A$.

In the first case, by the induction hypothesis, $r_1 \rightleftharpoons^* s_1 \times t_1$ and $r_2 \rightleftharpoons^* s_2 \times t_2$, with $s_i : A_i$ and $t_i : B_i$. Thus, $r \rightleftharpoons^* s_1 \times s_2 \times t_1 \times t_2$. We take $s = s_1 \times s_2$ and $t = t_1 \times t_2$. The other cases are similar.

- Let $r \rightleftharpoons^* \pi_{A \wedge B}(r')$. Then, by Lemma 4.2, $r' : (A \wedge B) \wedge C$, and hence, by the induction hypothesis, $r' \rightleftharpoons^* r_1 \times r_2$ where $r_1 : A \wedge B$, hence r is not in normal form. \square

Lemma 6.6. There is not closed normal simple term of type τ .

Proof. We proceed by cases:

- A variable is not closed.
- By Lemma 4.2, an abstraction has a type equivalent to an arrow, so it cannot be τ .
- Let $rs : \tau$. Without loss of generality, let $r \rightleftharpoons^* r' r_1 \dots r_n$ where r' is not an application, where, by Lemma 4.2, $r'_i : A_i$, $s : A_{n+1}$, and $r : A \Rightarrow \tau$ with $A \equiv \bigwedge_{i=1}^{n+1} A_i$. We proceed by cases on r' :

- r' cannot be a variable, since the term is closed.
 - Let $r' \rightleftharpoons^* \lambda x^B.r''$, with r'' of type C , then, by Lemma 4.2, $A \Rightarrow \tau \equiv B \Rightarrow C$, therefore, by Lemma 2.10 $C \equiv A' \Rightarrow \tau$, with $A \equiv A' \wedge B$. In this case, $\prod_i r_i \times s : A' \wedge B$. So by Lemma 6.5, $\prod_i r_i \times s \rightleftharpoons^* t_1 \times t_2$ with $t_1 : B$. Therefore, r is not normal.
 - r' cannot be an application, by hypothesis.
 - r' cannot be a projection. Indeed, if $r' \rightleftharpoons^* \pi_{A \Rightarrow \tau}(r'')$, then by Lemma 4.2, r'' has type $(A \Rightarrow \tau) \wedge B$, and so, by Lemma 6.5, r'' is equivalent to a product. Therefore, r' is not normal.
 - r' cannot be a product, since it has type $A \Rightarrow \tau$ and Lemma 4.2 prescribe a conjunction.
- By Lemma 4.2, a product has a type equivalent to a conjunction, so it cannot be τ .
- By Lemma 4.2, if $\pi_\tau(r) : \tau$, then $r : \tau \wedge B$, and so, by Lemma 6.5, $r \rightleftharpoons^* r_1 \times r_2$ with $r_1 : \tau$. Therefore, $\pi_\tau(r)$ is not normal. \square

Theorem 6.7. There is no closed term of type τ .

Proof. It is easy to see that the translation from terms to simple terms preserves the types, and reduction preserves simplicity. Hence, if there is a close term of type τ , there must be a simple close term of type τ , however, by Theorem 5.24, Theorem 4.4, and Lemma 6.6, we arrive to a contradiction. \square

Notice that, although we do not have closed terms of type τ , we do not have the usual property that every closed normal terms is equivalent to an introduction corresponding to its head symbol. For example, the term

$$(\lambda x^\tau.\lambda y^{\tau \Rightarrow \tau}.x)(\lambda z^\tau.z) : \tau \Rightarrow \tau$$

is in normal form, although it is an application and not an abstraction, and the term

$$\lambda x^{\tau \wedge \tau}.x : ((\tau \wedge \tau) \Rightarrow \tau) \wedge ((\tau \wedge \tau) \Rightarrow \tau)$$

is in normal form, although it is an abstraction and not a product. However, with extra η and δ rules [Díaz-Caro and Martínez López, 2015], the first term would expand to

$$\lambda w^\tau.((\lambda x^\tau.\lambda y^{\tau \Rightarrow \tau}.x)(\lambda z^\tau.z)w) \rightsquigarrow^* \lambda w^\tau.w$$

and the second term would expand to

$$\lambda x^{\tau \wedge \tau}.(\pi_\tau(x) \times \pi_\tau(x)) \rightleftharpoons (\lambda x^{\tau \wedge \tau}.\pi_\tau(x)) \times (\lambda x^{\tau \wedge \tau}.\pi_\tau(x))$$

7. Computing with System I

7.1. Pairs (and lists)

Because the symbol \times is associative and commutative, System I does not contain the usual notion of pairs. However, it is possible to encode a deterministic projection, even if we have more than one term of the same type. An example, although there are various possibilities, is given in the following table:

Standard	Encoding
$r \times s : A \wedge A$	$\lambda x^{\mathbb{1}}.r \times \lambda x^{\mathbb{2}}.s : \mathbb{1} \Rightarrow A \wedge \mathbb{2} \Rightarrow A$
$\pi_{\mathbb{1}}(r \times s)$	$\pi_{\mathbb{1} \Rightarrow A}(\lambda x^{\mathbb{1}}.r \times \lambda x^{\mathbb{2}}.s)y^{\mathbb{1}}$

where types $\mathbb{1}$ and $\mathbb{2}$ are any two different types. This example uses free variables, but it is easy to close it, e.g. use $\lambda y.y$ instead of $y^{\mathbb{1}}$ in the second line.

Moreover, this technique is not limited to pairs. Due to the associativity nature of \times , the encoding can be easily extended to lists.

7.2. Booleans

Example 3.11 on booleans overlooks an interesting fact: If $A \equiv B$, then both \mathbf{T} and \mathbf{F} behaves as a non-deterministic projector. Indeed, $\mathbf{T}rs \hookrightarrow^* r$, but also $(\lambda x^A.\lambda y^B.x)rs \rightleftharpoons (\lambda x^A.\lambda y^A.x)rs \rightleftharpoons (\lambda x^A.\lambda y^A.x)(r \times s) \rightleftharpoons (\lambda x^A.\lambda y^A.x)(s \times r) \rightleftharpoons (\lambda x^A.\lambda y^A.x)sr \hookrightarrow^* s$.

Similarly, $\mathbf{F}rs \hookrightarrow^* s$ and also $\mathbf{F}rs \rightsquigarrow^* r$. Hence, $A \Rightarrow A \Rightarrow A$ is not suitable to encode the type `Bool`. The type $A \Rightarrow A \Rightarrow A$ has only one term in the underlying equational theory.

Fortunately, there are ways to construct types with more than one term. First, let us define the following notation. For any t , let write $[t]^{\tau \Rightarrow \tau}$, the *canon* of t , that is, the term $\lambda z^{\tau \Rightarrow \tau}.t$, where $z^{\tau \Rightarrow \tau}$ is a fresh variable not appearing in t . Also, for any term t of type $(\tau \Rightarrow \tau) \Rightarrow A$, we write $\{t\}^{\tau \Rightarrow \tau}$, the *cocanon*, which is the inverse operation, that is, $\{[t]^{\tau \Rightarrow \tau}\}^{\tau \Rightarrow \tau} \hookrightarrow t$ for any t of type A . For the cocanon it suffices to take $\{t\}^{\tau \Rightarrow \tau} = t(\lambda x^{\tau}.x)$. Therefore, the type $((\tau \Rightarrow \tau) \Rightarrow A) \Rightarrow A \Rightarrow A$ has the following two different terms:

$$\begin{aligned} \mathbf{tt} &:= \lambda x^A.\lambda y^{(\tau \Rightarrow \tau) \Rightarrow A}.x \\ \mathbf{ff} &:= \lambda x^{(\tau \Rightarrow \tau) \Rightarrow A}.\lambda y^A.\{x\}^{\tau \Rightarrow \tau} \end{aligned}$$

Hence, it is possible to encode an if-then-else conditional expression in the following way:

$$\text{If } c \text{ then } r \text{ else } s := cr[s]^{\tau \Rightarrow \tau}$$

So, $\mathbf{tt}r[s]^{\tau \Rightarrow \tau} \hookrightarrow^* r$, while $\mathbf{ff}r[s]^{\tau \Rightarrow \tau} \rightleftharpoons^* \mathbf{f}[s]^{\tau \Rightarrow \tau}r \hookrightarrow^* \{[s]^{\tau \Rightarrow \tau}\}^{\tau \Rightarrow \tau} \hookrightarrow s$.

7.3. A deterministic subsystem

In section 7.1 we have seen how to encode a pair, transforming the non-deterministic projection into a deterministic one via an encoding. Another possibility, is to remove the non-deterministic behaviour of this calculus by dropping the isomorphisms (1) and (2), as well as rules `(COMM)` and `(ASSO)`.

Despite that such a modification would simplify the calculus—indeed, the projection can be taken as the standard projection—the resulting calculus would still count with distribution of application over conjunction and currfication, two interesting features, in particular, for programming languages. The former allows to execute a function only partially, when not all its results are needed. The latter can also be used to optimise

programs when there are multiple calls to the same function, but one of its arguments is fixed.

8. Conclusion, Discussion and Future Work

In this paper we defined System I, a proof system for propositional logic with an associative and commutative conjunction, and a distributive implication with respect to it, where equivalent propositions get the same proofs.

8.1. Other Related Work

Apart from the related work already discussed in the introduction, in a work by Garrigue and Ait-Kaci [1994], the selective λ -calculus has been presented, where only the isomorphism

$$(A \Rightarrow B \Rightarrow C) \equiv (B \Rightarrow A \Rightarrow C). \quad (5)$$

has been treated, which is complete with respect to the function type. In System I we also consider the conjunction, and hence four isomorphisms. Isomorphism (5) is a consequence of curriffication and commutation, that is $A \wedge B \equiv B \wedge A$ and $(A \wedge B) \Rightarrow C \equiv (A \Rightarrow B \Rightarrow C)$.

The selective λ -calculus includes labellings to identify which argument is being used at each time. Moreover, by considering the Church encoding of pairs, isomorphism (5) implies isomorphism (1) (commutativity of \wedge). However, their proposal is different to ours. In particular, we track the term by its type, which is a kind of labelling, but when two terms have the same type, then we leave the system to non-deterministically choose any proof. One of our main novelties is, indeed, the non-deterministic projector. However, we can also get back determinism, by encoding a labelling, as discussed in Section 7, or by dropping some isomorphisms (namely, associativity and commutativity of conjunction).

8.2. Future Work

8.2.1. Adding more connectives A subtle question is how to add a neutral element of the conjunction, which will imply more isomorphisms, e.g. $A \wedge \top \equiv A$, $A \Rightarrow \top \equiv \top$ and $\top \Rightarrow A \equiv A$. In System I adding the equation $\top \Rightarrow \top \equiv \top$ would make it possible to derive $(\lambda x^\top .xx)(\lambda x^\top .xx) : \top$, however this term is not the classical Ω , it is typed by \top , and imposing some restrictions on the beta reduction, it could be forced not to reduce to itself but to discard its argument. For example: “If $A \equiv \top$, then $(\lambda x^A .r)s \hookrightarrow \lambda x^A .r$, in other case, do the standard beta-reduction”.

8.2.2. Probabilistic and quantum computing A second line is the probabilistic interpretation of the non-determinism in our calculus. In [Díaz-Caro and Dowek, 2014] a probability space over the set of non-deterministic execution traces is defined. This way, System I is transformed into a probabilistic calculus instead of just a non-deterministic one, providing an alternative way for more complex constructions. Moreover, the original motivation

behind the linear algebraic extension of lambda calculus [Arrighi and Dowek, 2017] and its *vectorial* type system [Arrighi et al., 2017] was to encode quantum computing on it by considering not only non-deterministic superpositions, but formal linear combinations of terms. A projection depending on scalars could lead to a measurement operator in a future design. This is a promising future direction.

References

- Pablo Arrighi and Alejandro Díaz-Caro. A System F accounting for scalars. *Logical Methods in Computer Science*, 8(1:11), 2012.
- Pablo Arrighi and Gilles Dowek. Linear-algebraic lambda-calculus: higher-order, encodings, and confluence. In Andrei Voronkov, editor, *Proceedings of RTA 2008*, volume 5117 of *LNCS*, pages 17–31, 2008.
- Pablo Arrighi and Gilles Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1:8), 2017.
- Pablo Arrighi, Alejandro Díaz-Caro, and Benoît Valiron. The vectorial lambda-calculus. *Information and Computation*, 254(1):105–139, 2017.
- G erard Boudol. Lambda-calculi for (strict) parallel functions. *Information and Computation*, 108(1):51–127, 1994.
- Kim B. Bruce, Roberto Di Cosmo, and Giuseppe Longo. Provable isomorphisms of types. *Mathematical Structures in Computer Science*, 2(2):231–247, 1992.
- Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. A relational semantics for parallelism and non-determinism in a functional setting. *Annals of Pure and Applied Logic*, 163(7):918–934, 2012.
- Thierry Coquand and G erard Huet. The calculus of constructions. *Information and Computation*, 76(2–3):95–120, 1988.
- Ugo de’Liguoro and Adolfo Piperno. Non deterministic extensions of untyped λ -calculus. *Information and Computation*, 122(2):149–177, 1995.
- Mariangiola Dezani-Ciancaglini, Ugo de’Liguoro, and Adolfo Piperno. A filter model for concurrent λ -calculus. *SIAM Journal on Computing*, 27(5):1376–1419, 1998.
- Roberto Di Cosmo. *Isomorphisms of types: from λ -calculus to information retrieval and language design*. Progress in Theoretical Computer Science. Birkhauser, 1995.
- Roberto Di Cosmo. A short survey of isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5):825–838, 2005.
- Alejandro D iaz-Caro and Gilles Dowek. The probability of non-confluent systems. In Mauricio Ayala-Rincn, Eduardo Bonelli, and Ian Mackie, editors, *Proceedings of DCM 2013*, volume 144 of *EPTCS*, pages 1–15, 2014.
- Alejandro D iaz-Caro and Gilles Dowek. Typing quantum superpositions and measurement. In Carlos Mart ın-Vide, Roman Neruda, and Miguel A. Vega-Rodr ıguez, editors, *Proceedings of TPNC 2017*, volume 10687 of *LNCS*, pages 281–293, 2017.
- Alejandro D iaz-Caro and Pablo E. Mart ınez L opez. Isomorphisms considered as equalities: Projecting functions and enhancing partial application through an implementation of λ^+ . In *Proceedings of the 27th Symposium on the Implementation and Application of Functional Programming Languages*, IFL ’15, pages 9:1–9:11. ACM, 2015.

- Alejandro Díaz-Caro and Barbara Petit. Linearity in the non-deterministic call-by-value setting. In Luke Ong and Ruy de Queiroz, editors, *Proceedings of WoLLIC 2012*, volume 7456 of *LNCS*, pages 216–231, 2012.
- Gilles Dowek and Ying Jiang. On the expressive power of schemes. *Information and Computation*, 209:1231–1245, 2011.
- Gilles Dowek and Benjamin Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.
- Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1):33–72, 2003.
- Jacques Garrigue and Hassan Aït-Kaci. The typed polymorphic label-selective λ -calculus. In *Proceedings of POPL 1994*, ACM SIGPLAN, pages 35–47, 1994.
- Herman Geuvers, Robbert Krebbers, James McKinna, and Freek Wiedijk. Pure type systems without explicit contexts. In Karl Crary and Marino Miculan, editors, *Proceedings of LFMTTP 2010*, volume 34 of *EPTCS*, pages 53–67, 2010.
- Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, 1989.
- Per Martin-Löf. *Intuitionistic type theory*. Studies in proof theory. Bibliopolis, 1984.
- Michele Pagani and Simona Ronchi Della Rocca. Linearity, non-determinism and solvability. *Fundamental Informaticae*, 103(1–4):173–202, 2010.
- Jonghyun Park, Jeongbong Seo, Sungwoo Park, and Gyesik Lee. Mechanizing metatheory without typing contexts. *Journal of Automated Reasoning*, 52(2):215–239, 2014.
- Mikael Rittri. Retrieving library identifiers via equational matching of types. In *Proceedings of CADE 1990*, volume 449 of *LNCS*, pages 603–617, 1990.
- The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. <http://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.
- Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.