Proof Normalisation in a Logic Identifying Isomorphic Propositions

Alejandro Díaz-Caro

Instituto de Ciencias de la Computación (CONICET-Universidad de Buenos Aires), Ciudad Autónoma de Buenos Aires, Argentina
Universidad Nacional de Quilmes, Bernal (Buenos Aires), Argentina
https://www-2.dc.uba.ar/staff/adiazcaro/
adiazcaro@icc.fcen.uba.ar

Gilles Dowek

Inria, LSV, ENS Paris-Saclay, France
http://www.lsv.fr/~dowek/
gilles.dowek@ens-paris-saclay.fr

Abstract

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 its normalisation and consistency.

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Type theory; Mathematics of computing → Lambda calculus

Keywords and phrases Simply typed lambda calculus, Isomorphisms, Logic, Cut-elimination, Proof-reduction

Digital Object Identifier 10.4230/LIPIcs.FSCD.2019.14

Related Version A variant of the system presented in this paper, has been published in https://arxiv.org/abs/1303.7334v1 (LSFA 2012 [13]), with a sketch of a subject reduction proof but no normalisation or consistency proofs.

Funding Partially supported by ECOS-Sud A17C03, PICT 2015-1208, and the French-Argentinian laboratory SINFIN.

1 Introduction

1.1 Identifying isomorphic propositions

In mathematics, addition is associative and commutative, multiplication distributes over addition, etc. In contrast, in logic conjunction is neither associative nor commutative, implication does not distribute over conjunction, etc. For instance, the propositions $A \wedge B$ and $B \wedge A$ are different: 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 in [6, 11, 12, 26], where a notion of isomorphic propositions has been defined: two propositions A and B are isomorphic if there exist 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:

$$A \wedge B \equiv B \wedge A \tag{1}$$
$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C \tag{2}$$
$$A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C) \tag{3}$$
$$(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C \tag{4}$$

For example, $(A \Rightarrow B \Rightarrow C) \equiv (B \Rightarrow A \Rightarrow C)$ is a consequence of (4) and (1) [6].

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 [23], in the Calculus of Constructions [8], and in Deduction modulo theory [17, 19], where definitionally equivalent propositions, for instance $A \subseteq B$, $A \in \mathcal{P}(B)$, and $\forall x\ (x \in A \Rightarrow x \in B)$ can be identified. But definitional equality does not handle isomorphisms. For example, $A \wedge B$ and $B \wedge A$ are not identified in these logics. Beside definitional equality, identifying isomorphic types in type theory, is also a goal of the univalence axiom [27].

Isomorphisms make proofs more natural. For instance, to prove $(A \wedge (A \Rightarrow B)) \Rightarrow B$ in natural deduction we need to introduce conjunctive hypothesis $A \wedge (A \Rightarrow B)$ which has to be decomposed into A and $A \Rightarrow B$, while using the isomorphism (4) allows to transform the goal to $A \Rightarrow (A \Rightarrow B) \Rightarrow B$ and introduce directly the hypotheses A and $A \Rightarrow B$, eliminating completely the need for conjunctive hypotheses.

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 extended with Cartesian product. So, System I is an extension of this calculus where, for example, a pair of functions $\langle r, s\rangle$ of type $(A \Rightarrow B) \wedge (A \Rightarrow C) \equiv A \Rightarrow (B \wedge C)$ can be applied to an argument t of type A, yielding a term $\langle r, s\rangle t$ of type $B \wedge C$. For example, the term $\langle \lambda x^\tau.x, \lambda x^\tau.x\rangle y$ has type $\tau \wedge \tau$. With the usual reduction rules of lambda calculus with pairs, such a term would be normal, but we can also extend the reduction relation, with an equation $\langle r, s\rangle t \rightleftarrows \langle rt, st\rangle$, such that this term is equivalent to $\langle (\lambda x^\tau.x)y, (\lambda x^\tau.x)y\rangle$ and thus reduces to $\langle y, y\rangle$. Taking too many of such equations may lead to non termination (Section 8.1), and taking too few multiplies undesired normal forms. The choice of the rules in this paper is motivated by the goal to have both termination of reduction (Section 5) and consistency (Section 6), that is, no normal closed term of atomic types.

To stress the associativity and commutativity of the notion of pair, we write $r \times s$ instead of $\langle r, s\rangle$ and thus write this equivalence as

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

Several similar equivalence rules on terms are introduced: one related to the isomorphism (1), the commutativity of the conjunction, $r \times s \rightleftarrows s \times r$; one related to the isomorphism (2), the associativity of the conjunction, $(r \times s) \times t \rightleftarrows r \times (s \times t)$; two to the isomorphism (3), the distributivity of implication with respect to conjunction, $\lambda x.(r \times s) \rightleftarrows \lambda x.r \times \lambda x.s$ and $(r \times s)t \rightleftarrows rt \times st$; and one related to the isomorphism (4), the currification, $rst \rightleftarrows r(s \times t)$.

One of the 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 type $A \wedge B$ and type $B \wedge A$, 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, like the calculus developed in [18], 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 [5,7,9,10,24] 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 $(r \mid s)t$ reduces to $rt \mid st$ 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 in parallel. 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 fist 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 $(r + s)t \rightleftarrows rt + st$ 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 [1–4,14,16,28], 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. In Section 5, we prove its strong normalisation. In Section 6, we prove its consistency. Finally, in Section 7, we discuss how System I could be used as a programming language.

2 Type isomorphisms

2.1 Types and isomorphisms

Types are defined by the following grammar

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

where τ is the only atomic type.

▶ **Definition 2.1** (Size of a type). *The size of a type is defined as usual by*

$$s(\tau) = 1$$
$$s(A \Rightarrow B) = s(A) + s(B) + 1$$
$$s(A \wedge B) = s(A) + s(B) + 1$$

▶ **Definition 2.2** (Congruence). *The isomorphisms* (1), (2), (3), *and* (4) *define a congruence on types.*

$$A \wedge B \equiv B \wedge A \tag{1}$$
$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C \tag{2}$$
$$A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C) \tag{3}$$
$$(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C \tag{4}$$

2.2 Prime factors

▶ **Definition 2.3** (Prime types). *A prime type is a type of the form $C_1 \Rightarrow \cdots \Rightarrow C_n \Rightarrow \tau$, with $n \geq 0$.*

A prime type is equivalent to $(\bigwedge_{i=1}^n C_i) \Rightarrow \tau$, which is either equivalent to τ or to $C \Rightarrow \tau$, for some C. For uniformity, we may write $\emptyset \Rightarrow \tau$ for τ.

We now show that each type can be decomposed into a conjunction of prime types. We use the notation $[A_i]_{i=1}^n$ for the multiset whose elements are $A_1, \dots, A_n$. We may write $[A_i]_i$ when the number of elements is not important. If $R = [A_i]_i$ is a multiset of types, then $\mathsf{conj}(R) = \bigwedge_i A_i$.

▶ **Definition 2.4.** *We write $[A_1, \dots, A_n] \sim [B_1, \dots, B_m]$ if $n = m$ and $B_i \equiv A_i$.*

▶ **Definition 2.5** (Prime factors). *The multiset of prime factors of a type A is inductively defined as follows*

$$\mathsf{PF}(\tau) = [\tau]$$
$$\mathsf{PF}(A \Rightarrow B) = [(A \wedge C_i) \Rightarrow \tau]_{i=1}^n \quad \textit{where } [C_i \Rightarrow \tau]_{i=1}^n = \mathsf{PF}(B)$$
$$\mathsf{PF}(A \wedge B) = \mathsf{PF}(A) \uplus \mathsf{PF}(B)$$

with the convention that $A \wedge \emptyset = A$.

Note that if $B \Rightarrow \tau \in \mathsf{PF}(A)$, then $s(B) < s(A)$.

▶ **Lemma 2.6.** *For all A, $A \equiv \mathsf{conj}(\mathsf{PF}(A))$.*

Proof. By induction on $s(A)$.

- If $A = \tau$, then $\mathsf{PF}(\tau) = [\tau]$, and so $\mathsf{conj}(\mathsf{PF}(\tau)) = \tau$.
- If $A = B \Rightarrow C$, then $\mathsf{PF}(A) = [(B \wedge C_i) \Rightarrow \tau]_i$, where $[C_i \Rightarrow \tau]_i = \mathsf{PF}(C)$. By the induction hypothesis, $C \equiv \bigwedge_i (C_i \Rightarrow \tau)$, hence, $A = B \Rightarrow C \equiv B \Rightarrow \bigwedge_i (C_i \Rightarrow \tau) \equiv \bigwedge_i (B \Rightarrow C_i \Rightarrow \tau) \equiv \bigwedge_i ((B \wedge C_i) \Rightarrow \tau)$.
- If $A = B \wedge C$, then $\mathsf{PF}(A) = \mathsf{PF}(B) \uplus \mathsf{PF}(C)$. By the induction hypothesis, $B \equiv \mathsf{conj}(\mathsf{PF}(B))$, and $C \equiv \mathsf{conj}(\mathsf{PF}(C))$. Therefore, $A = B \wedge C \equiv \mathsf{conj}(\mathsf{PF}(B)) \wedge \mathsf{conj}(\mathsf{PF}(C)) \equiv \mathsf{conj}(\mathsf{PF}(B \wedge C)) \equiv \mathsf{conj}(\mathsf{PF}(B) \uplus \mathsf{PF}(C)) = \mathsf{conj}(\mathsf{PF}(A))$. ◀

▶ **Lemma 2.7.** *If $A \equiv B$, then $\mathsf{PF}(A) \sim \mathsf{PF}(B)$.*

Proof. First we check that $\mathsf{PF}(A \wedge B) \sim \mathsf{PF}(B \wedge A)$ and similar for the other three isomorphisms. Then we prove by structural induction that if A and B are equivalent in one step, then $\mathsf{PF}(A) \sim \mathsf{PF}(B)$. We conclude by an induction on the length of the derivation of the equivalence $A \equiv B$. ◀

2.3 Measure of types

The size of a type is not preserved by equivalence. For instance, $\tau \Rightarrow (\tau\wedge\tau) \equiv (\tau \Rightarrow \tau)\wedge(\tau \Rightarrow \tau)$, but $s(\tau \Rightarrow (\tau \wedge \tau)) = 5$ and $s((\tau \Rightarrow \tau) \wedge (\tau \Rightarrow \tau)) = 7$. Thus, we define another notion of measure of a type.

▶ **Definition 2.8** (Measure of a type). *The measure of a type is defined as follows*

$$m(A) = \sum_i (m(C_i) + 1) \qquad \textit{where } [C_i \Rightarrow \tau]_i = \mathsf{PF}(A)$$

with the convention that $m(\emptyset) = 0$.

▶ **Lemma 2.9.** *If* $A \equiv B$*, then* $m(A) = m(B)$.

Proof. By induction on $s(A)$. Let $\mathsf{PF}(A) = [C_i \Rightarrow \tau]_i$ and $\mathsf{PF}(B) = [D_j \Rightarrow \tau]_j$. By Lemma 2.7, $[C_i \Rightarrow \tau]_i \sim [D_i \Rightarrow \tau]_i$. Without lost of generality, take $C_i \equiv D_i$. By the induction hypothesis, $m(C_i) = m(D_i)$. Then, $m(A) = \sum_i(m(C_i) + 1) = \sum_i(m(D_i) + 1) = m(B)$. ◀

The following lemma shows that the measure $m(A)$ verifies the usual properties.

▶ **Lemma 2.10.**
1. $m(A \wedge B) > m(A)$
2. $m(A \Rightarrow B) > m(A)$
3. $m(A \Rightarrow B) > m(B)$

Proof.
1. $\mathsf{PF}(A)$ is a strict submultiset of $\mathsf{PF}(A \wedge B)$.
2. Let $\mathsf{PF}(B) = [C_i \Rightarrow \tau]_{i=1}^n$. Then, $\mathsf{PF}(A \Rightarrow B) = [(A \wedge C_i) \Rightarrow \tau]_{i=1}^n$. Hence, $m(A \Rightarrow B) \geq m(A \wedge C_1) + 1 > m(A \wedge C_1) \geq m(A)$.
3. $m(A \Rightarrow B) = \sum_i m(A \wedge C_i) + 1 > \sum_i m(C_i) + 1 = m(B)$. ◀

2.4 Decomposition properties on types

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

- If $A \Rightarrow B \equiv \bigwedge_{i=1}^n C_i$, then each C_i is equivalent to an implication $A \Rightarrow B_i$, where the conjunction of the B_i is equivalent to B.
- If $A \wedge B \equiv \bigwedge_i C_i$, then each C_i is a conjunction of elements, possibly empty, that contribute to A and to B.

We state these properties in Corollary 2.12 and Lemma 2.15.

▶ **Lemma 2.11.** *If* $A \Rightarrow B \equiv C_1 \wedge C_2$*, then* $C_1 \equiv A \Rightarrow B_1$ *and* $C_2 \equiv A \Rightarrow B_2$ *where* $B \equiv B_1 \wedge B_2$.

Proof. By Lemma 2.7, $\mathsf{PF}(A \Rightarrow B) \sim \mathsf{PF}(C_1 \wedge C_2) = \mathsf{PF}(C_1) \uplus \mathsf{PF}(C_2)$. Let $\mathsf{PF}(B) = [D_i \Rightarrow \tau]_{i=1}^n$, so $\mathsf{PF}(A \Rightarrow B) = [(A \wedge D_i) \Rightarrow \tau]_{i=1}^n$. Without lost of generality, take $\mathsf{PF}(C_1) \sim [(A \wedge D_i) \Rightarrow \tau]_{i=1}^k$ and $\mathsf{PF}(C_2) \sim [(A \wedge D_i) \Rightarrow \tau]_{i=k+1}^n$. Therefore, by Lemma 2.6, we have $A \Rightarrow B \equiv \bigwedge_{i=1}^k((A \wedge D_i) \Rightarrow \tau) \wedge \bigwedge_{i=k+1}^n((A \wedge D_i) \Rightarrow \tau) \equiv (A \Rightarrow \bigwedge_{i=1}^k(D_i \Rightarrow \tau)) \wedge (A \Rightarrow \bigwedge_{i=k+1}^n(D_i \Rightarrow \tau))$. Take $B_1 = \bigwedge_{i=1}^k D_i \Rightarrow \tau$ and $B_2 = \bigwedge_{i=k+1}^n D_i \Rightarrow \tau$. Remark that $C_1 \equiv A \Rightarrow B_1$, $C_2 \equiv A \Rightarrow B_2$ and $B \equiv B_1 \wedge B_2$. ◀

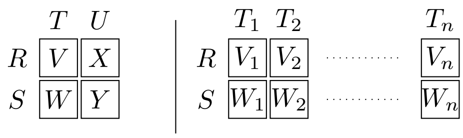

Figure 1 Lemma 2.13.

▶ **Corollary 2.12.** *If $A \Rightarrow B \equiv \bigwedge_{i=1}^n C_i$, then for $i \in \{1, \dots, n\}$, we have $C_i \equiv A \Rightarrow B_i$ where $B \equiv \bigwedge_{i=1}^n B_i$.*

Proof. By induction on n. By Lemma 2.11, $\bigwedge_{i=1}^{n-1} C_i \equiv A \Rightarrow B'$ and $C_n \equiv A \Rightarrow B_n$, with $B \equiv B' \wedge B_n$. By the induction hypothesis, for $i \leq n-1$, $C_i \equiv A \Rightarrow B_i$ where $B' \equiv \bigwedge_{i=1}^{n-1} B_i$. Hence, $B \equiv \bigwedge_{i=1}^n B_i$. ◀

▶ **Lemma 2.13.** *Let R, S, T and U be four multisets such that $R \uplus S = T \uplus U$, then there exist four multisets V, W, X, and Y such that $R = V \uplus X$, $S = W \uplus Y$, $T = V \uplus W$, and $U = X \uplus Y$, cf. Figure 1.*

Proof. Consider an element $a \in R \uplus S = T \uplus U$. Let r be the multiplicity of a in R, s its multiplicity in S, t its multiplicity in T, and u its multiplicity in U. We have $r + s = t + u$. If $r \leq t$ we put r copies of a in V, $t - r$ in W, 0 in X, and u in Y. Otherwise, we put t in V, 0 in W, $r - t$ in X, and s in Y. ◀

▶ **Corollary 2.14.** *Let R and S be two multisets and $(T_i)_{i=1}^n$ be a family of multisets, such that $R \uplus S = \biguplus_{i=1}^n T_i$. Then, there exist multisets $V_1, \dots, V_n, W_1, \dots, W_n$ such that $R = \biguplus_i V_i$ and $S = \biguplus_i W_i$ and for each i, $T_i = V_i \uplus W_i$, cf. Figure 1.*

Proof. By induction on n. We have $R \uplus S = \biguplus_{i=1}^{n-1} T_i \uplus T_n$. Then, by Lemma 2.13, there exist R', S', V_n, W_n such that $R = R' \uplus V_n$, $S = S' \uplus W_n$, $\biguplus_{i=1}^{n-1} T_i = R' \uplus S'$, and $T_n = V_n \uplus W_n$. By induction hypothesis, there exist $V_1, \dots, V_{n-1}$ and $W_1, \dots, W_{n-1}$ such that $R' = \biguplus_{i=1}^{n-1} V_i$, $S' = \biguplus_{i=1}^{n-1} W_i$ and each $T_i = V_i \uplus W_i$. Hence, $R = \biguplus_{i=1}^n V_i$ and $S = \biguplus_{i=1}^n W_i$. ◀

▶ **Lemma 2.15.** *If $A \wedge B \equiv \bigwedge_{i=1}^n C_i$ then there exists a partition $E \uplus F \uplus G$ of $\{1, \dots, n\}$ such that*
- *$C_i = A_i \wedge B_i$, when $i \in E$;*
- *$C_i = A_i$, when $i \in F$;*
- *$C_i = B_i$, when $i \in G$;*
- *$A = \bigwedge_{i \in E \uplus F} A_i$; and*
- *$B = \bigwedge_{i \in E \uplus G} B_i$.* ◀

Proof. Let $R = \mathsf{PF}(A)$, $S = \mathsf{PF}(B)$, and $T_i = \mathsf{PF}(C_i)$. By Lemma 2.7, we have $\mathsf{PF}(A \wedge B) \sim \mathsf{PF}(\bigwedge_i C_i)$, that is $R \uplus S \sim \biguplus_i T_i$. By Corollary 2.14, there exist V_i and W_i such that $R = \biguplus_{i=1}^n V_i$, $S = \biguplus_{i=1}^n W_i$, and $T_i \sim V_i \uplus W_i$. As T_i is non-empty, V_i and W_i cannot be both empty.
- If V_i and W_i are both non-empty, we let $i \in E$ and $A_i = \mathsf{conj}(V_i)$ and $B_i = \mathsf{conj}(W_i)$. By Lemma 2.6, $C_i \equiv \mathsf{conj}(T_i) \equiv \mathsf{conj}(V_i \uplus W_i) \equiv A_i \wedge B_i$.
- If V_i is non-empty and W_i is empty, we let $i \in F$, and $A_i = \mathsf{conj}(V_i) \equiv \mathsf{conj}(T_i) \equiv C_i$.
- If W_i is non-empty and V_i is empty, we let $i \in G$, and $B_i = \mathsf{conj}(W_i) \equiv \mathsf{conj}(T_i) \equiv C_i$.

As $V_i = \emptyset$ when $i \in G$, we have $A \equiv \mathsf{conj}(R) \equiv \mathsf{conj}(\biguplus_{i \in E \uplus F} V_i) \equiv \bigwedge_{i \in E \uplus F} A_i$.
As $W_i = \emptyset$ when $i \in F$, we have $B \equiv \mathsf{conj}(S) \equiv \mathsf{conj}(\biguplus_{i \in E \uplus G} W_i) \equiv \bigwedge_{i \in E \uplus G} B_i$. ◀

Table 1 The type system.

$$[x\in\mathcal{V}_A]\frac{}{x:A}\ (ax) \qquad [A\equiv B]\frac{r:A}{r:B}\ (\equiv)$$

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

Table 2 Symmetric relation.

$$r\times s \rightleftarrows s\times r \qquad (\textsc{comm})$$
$$(r\times s)\times t \rightleftarrows r\times(s\times t) \qquad (\textsc{asso})$$
$$\lambda x^A.(r\times s) \rightleftarrows \lambda x^A.r\times\lambda x^A.s \qquad (\textsc{dist}_\lambda)$$
$$(r\times s)t \rightleftarrows rt\times st \qquad (\textsc{dist}_{\mathsf{app}})$$
$$rst \rightleftarrows r(s\times t) \qquad (\textsc{curry})$$

3 System I

3.1 Syntax

We associate to each (up to equivalence) prime 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 $\mathcal{V}_A\cap\mathcal{V}_B=\emptyset$. The set of terms is defined inductively by the grammar

$$r,s,t,\dots \quad ::= \quad x \mid \lambda x.r \mid rs \mid r\times s \mid \pi_A(r)$$

We recall the type on binding occurrences of variables and write $\lambda x^A.t$ for $\lambda x.t$ when $x\in\mathcal{V}_A$. α-equivalence and substitution are defined as usual. The type system is given in Table 1. We use a presentation of typing rules without explicit context following [21, 25], hence the typing judgments have the form $r:A$. The preconditions of a typing rule is written on its left.

3.2 Operational semantics

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

▶ **Definition 3.1.** *The symmetric relation $\rightleftarrows$ is the smallest contextually closed relation defined by the rules given in Table 2.*

Each isomorphism induces an equivalence between terms. Two rules however correspond to the isomorphism (3), depending on which distribution is taken into account: elimination or introduction of implication. We write $\rightleftarrows^*$ for the transitive and reflexive closure of $\rightleftarrows$. Note that $\rightleftarrows^*$ is an equivalence relation.

Because of the associativity property of $\times$, the term $r\times(s\times t)$ is equivalent to the term $(r\times s)\times t$, so we can just write it $r\times s\times t$.

As explained in the introduction, variables of conjunctive types are useless, hence all variables have prime types. This way, there is no term $\lambda x^{\tau\wedge\tau}.x$, but a term $\lambda y^\tau.\lambda z^\tau.y\times z$ which is equivalent to $(\lambda y^\tau.\lambda z^\tau.y)\times(\lambda y^\tau.\lambda z^\tau.z)$.

The size of a term is not invariant through the equivalence $\rightleftarrows$. Hence, we introduce a measure $M(\cdot)$, which is given in Table 3.

Table 3 Measure on terms.

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

▶ **Lemma 3.2.** *If $r \rightleftarrows s$ then $P(r) = P(s)$.*

Proof. We check the case of each rule of Table 2, and then conclude by structural induction to handle the contextual closure.

- (COMM): $P(r \times s) = 1 + P(r) + P(s) = P(s \times r)$.
- (ASSO): $P((r \times s) \times t) = 2 + P(r) + P(s) + P(t) = P(r \times (s \times t))$.
- (DIST$_\lambda$): $P(\lambda x^A.(r \times s)) = 1 + P(r) + P(s) = P(\lambda x^A.r \times \lambda x^A.s)$.
- (DIST$_{\mathsf{app}}$): $P((r \times s)t) = 1 + P(r) + P(s) = P(rt \times st)$.
- (CURRY): $P((rs)t) = P(r) = P(r(s \times t))$. ◀

▶ **Lemma 3.3.** *If $r \rightleftarrows s$ then $M(r) = M(s)$.*

Proof. We check the case of each rule of Table 2, and then conclude by structural induction to handle the contextual closure.

- (COMM): $M(r \times s) = M(r) + M(s) = M(s \times r)$.
- (ASSO): $M((r \times s) \times t) = M(r) + M(s) + M(t) = M(r \times (s \times t))$.
- (DIST$_\lambda$): $M(\lambda x^A.(r \times s)) = 2 + M(r) + M(s) + P(r) + P(s) = M(\lambda x^A.r \times \lambda x^A.s)$
- (DIST$_{\mathsf{app}}$): $M((r \times s)t) = M(r) + M(s) + 2M(t) + P(r)M(t) + P(s)M(t) = M(rt \times st)$
- (CURRY): $M((rs)t) = M(r) + M(s) + P(r)M(s) + M(t) + P(r)M(t) = M(r(s \times t))$ ◀

▶ **Lemma 3.4.** *$M(\lambda x^A.r) > M(r)$, $M(rs) > M(r)$, $M(rs) > M(s)$, $M(r \times s) > M(r)$, $M(r \times s) > M(s)$, and $M(\pi_A(r)) > M(r)$.*

Proof. By induction on r, $M(r) \geq 1$. We conclude with a case inspection. ◀

We use the measure to prove that the equivalence class of a term is a finite set.

▶ **Lemma 3.5.** *For any term r, the set $\{s \mid s \rightleftarrows^* r\}$ is finite (modulo α-equivalence).*

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

By induction on n. For $n = 1$ the set $\{s \mid FV(s) \subseteq F$ and $M(s) \leq 1\}$ contains only the variables of F. Assume the property holds for n, then, by the Lemma 3.4 the set $H(n+1, F)$ is a subset of the finite set containing the variables of F, the abstractions $(\lambda x^A.r)$ for r in $H(n, F \cup \{x\})$, the applications (rs) for r and s in $H(n, F)$, the products $r \times s$ for r and s in $H(n, F)$, the projections $\pi_A(r)$ for r in $H(n, F)$. ◀

▶ **Definition 3.6.** *The reduction relation $\hookrightarrow$ is the smallest contextually closed relation defined by the rules given in Table 4. We write $\hookrightarrow^*$ for the transitive and reflexive closure of $\hookrightarrow$.*

Table 4 Reduction relation.

If $s : A$, $(\lambda x^A.r)s \hookrightarrow r[s/x]$ (β)

If $r : A$, $\pi_A(r \times s) \hookrightarrow r$ (π)

▶ **Definition 3.7.** *We write $\rightsquigarrow$ for the relation $\hookrightarrow$ modulo $\rightleftarrows^*$ (i.e. $r \rightsquigarrow s$ iff $r \rightleftarrows^* r' \hookrightarrow s' \rightleftarrows^* s$), and $\rightsquigarrow^*$ for its transitive and reflexive closure.*

Remark that, by Lemma 3.5, a term has a finite number of reducts in one step and these reducts can be computed.

3.3 Examples

▶ **Example 3.8.** Let $r : A$ and $s : B$. Then $(\lambda x^A.\lambda y^B.x)(r \times s) : A$ and

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

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

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

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

▶ **Example 3.9.** Let $s : A$ and $t : B$, and let $\mathbf{TF} = \lambda x^A.\lambda y^B.(x \times y)$.

Then $\mathbf{TF} : A \Rightarrow B \Rightarrow (A \wedge B) \equiv ((A \wedge B) \Rightarrow A) \wedge ((A \wedge B) \Rightarrow B)$. Therefore, $\pi_{(A\wedge B)\Rightarrow A}(\mathbf{TF}) : (A \wedge B) \Rightarrow A$. Hence, $\pi_{(A\wedge B)\Rightarrow A}(\mathbf{TF})(s \times t) : A$.

This term reduces as follows:

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

▶ **Example 3.10.** Let $\mathbf{T} = \lambda x^A.\lambda y^B.x$ and $\mathbf{F} = \lambda x^A.\lambda y^B.y$. The term $\mathbf{T} \times \mathbf{F} \times \mathbf{TF}$ has type $((A \wedge B) \Rightarrow (A \wedge B)) \wedge ((A \wedge B) \Rightarrow (A \wedge B))$.

Hence, $\pi_{(A\wedge B)\Rightarrow(A\wedge B)}(\mathbf{T} \times \mathbf{F} \times \mathbf{TF})$ is well typed and reduces non-deterministically either to $\mathbf{T} \times \mathbf{F}$ or to $\mathbf{TF}$. Moreover, as $\mathbf{T} \times \mathbf{F}$ and $\mathbf{TF}$ are equivalent, the non-deterministic choice does not play any role in this particular case. We will come back to the encoding of booleans in System I on Section 7.

4 Subject Reduction

The set of types assigned to a term is preserved under $\rightleftarrows$ and $\hookrightarrow$. Before proving this property, we prove the unicity of types (Lemma 4.1), the generation lemma (Lemma 4.2), and the substitution lemma (Lemma 4.3). We only state the lemmas in this section. The detailed proofs can be found in Appendix A.

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* $x \in \mathcal{V}_A$ *and* $x : B$*, 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 statement is proved by induction on the typing derivation. For the statement 1, we have $x \in \mathcal{V}_A$ and $x : B$. The only way to type this term is either by the rule (ax) or $(\equiv)$.
- In the first case, $A = B$, hence $A \equiv B$.
- In the second case, there exists B' such that $x : B'$ 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. ◀

▶ **Lemma 4.3** (Substitution). *If* $r : A$*,* $s : B$*, and* $x \in \mathcal{V}_B$*, then* $r[s/x] : A$.

Proof. By structural induction on r (cf. Appendix A). ◀

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

Proof. By induction on the rewrite relation (cf. Appendix A). ◀

5 Strong Normalisation

In this section we prove the strong normalisation of reduction $\rightsquigarrow$: 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 $|t|$ (recall that each term has a finite number of reducts).

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

In simply typed lambda calculus we can either define $[\![A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_n \Rightarrow \tau]\!]$ as the set of terms r such that for all $s \in [\![A_1]\!]$, $rs \in [\![A_2 \Rightarrow \cdots \Rightarrow A_n \Rightarrow \tau]\!]$ or, equivalently, as the set of terms r such that for all $s_i \in [\![A_i]\!]$, $rs_1 \dots s_n \in [\![\tau]\!] = \mathsf{SN}$. To prove that a term of the form $\lambda x^A.t$ is reducible, we need to use the so-called CR3 property [22], in the first case, and the property that a term whose all one-step reducts are in SN is in SN, in the second. In System I, an introduction can be equivalent to an elimination e.g. $rt \times st \rightleftarrows (r \times s)t$, hence, we cannot define a notion of neutral term and have an equivalent to the CR3 property. Therefore, we use the second definition.

Before we prove the normalisation of System I, we first reformulate the proof of strong normalisation of simply typed lambda-calculus along these lines.

5.1 Normalisation of simply typed lambda calculus

▶ **Definition 5.1** (Elimination context). *Consider an extension of simply typed lambda calculus where we introduce an extra symbol* $[]^A$*, called hole of type* A*.*

An elimination context with a hole $[]^{B_1\Rightarrow\cdots\Rightarrow B_n\Rightarrow\tau}$ *is a term* $K^\tau_{B_1\Rightarrow\cdots\Rightarrow B_n\Rightarrow\tau}$ *of type* τ *of the form* $[]^{B_1\Rightarrow\cdots\Rightarrow B_n\Rightarrow\tau}r_1\dots r_n$*. We write* $K^\tau_A[t]$*, for the term* $K^\tau_A[t/[]^A] = tr_1\dots r_n$*.*

▶ **Definition 5.2** (Terms occurring in an elimination context). $\mathcal{T}([]^A r_1\dots r_n) = \{r_1,\dots,r_n\}$*. Note that the types of the elements of* $\mathcal{T}([]^A r_1\dots r_n)$ *are smaller than* A*, and that if* $r_1,\dots,r_n \in \mathsf{SN}$*, then* $[]^A r_1\dots r_n \in \mathsf{SN}$*.*

▶ **Definition 5.3** (Reducibility). *The set* $[\![A]\!]$ *of reducible terms of type* A *is defined by structural induction on* A *as the set of terms* $t : A$ *such that for any elimination context* K^τ_A *such that the terms in* $\mathcal{T}(K^\tau_A)$ *are all reducible, we have* $K^\tau_A[t] \in \mathsf{SN}$*.*

▶ **Definition 5.4** (Reducible elimination context). *An elimination context* K^B_A *is reducible, if all the terms in* $\mathcal{T}(K^B_A)$ *are reducible.*

▶ **Lemma 5.5.** *For all* A*,* $[\![A]\!] \subseteq \mathsf{SN}$ *and all the variables of type* A *are in* $[\![A]\!]$*.*

Proof. By induction on A. ◀

▶ **Lemma 5.6** (Adequacy of application). *If* $r \in [\![A \Rightarrow B]\!]$ *and* $s \in [\![A]\!]$*, then* $rs \in [\![B]\!]$*.*

Proof. Let K^τ_B be a reducible elimination context. We need to prove that $K^\tau_B[rs] \in \mathsf{SN}$. As $s \in [\![A]\!]$, the elimination context $K'^\tau_{A\Rightarrow B} = K^\tau_B[[]^{A\Rightarrow B}s]$ is reducible, and since $r \in [\![A \Rightarrow B]\!]$, we have $K^\tau_B[rs] = K'^\tau_{A\Rightarrow B}[r] \in \mathsf{SN}$. ◀

▶ **Lemma 5.7** (Adequacy of abstraction). *If for all* $t \in [\![A]\!]$*,* $r[t/x] \in [\![B]\!]$*, then* $\lambda x^A.r \in [\![A \Rightarrow B]\!]$*.*

Proof. We need to prove that for every reducible elimination context $K^\tau_{A\Rightarrow B}$, $K^\tau_{A\Rightarrow B}[\lambda x^A.r] \in \mathsf{SN}$, that is that all its one step reducts are in SN. By Lemma 5.5, $x \in [\![A]\!]$, so $r \in [\![B]\!] \subseteq \mathsf{SN}$. Then, we proceed by induction on $|r| + |K^\tau_{A\Rightarrow B}|$. ◀

▶ **Definition 5.8** (Adequate substitution). *A substitution* σ *is adequate if for all* $x : A$*, we have* $\sigma(x) \in [\![A]\!]$*.*

▶ **Theorem 5.9** (Adequacy). *If* $r : A$*, the for all* σ *adequate, we have* $\sigma r \in [\![A]\!]$*.*

Proof. By induction on r. ◀

▶ **Theorem 5.10** (Strong normalisation). *If* $r : A$*, then* $r \in \mathsf{SN}$*.*

Proof. By Lemma 5.5, the idendity substitution is adequate. Thus, by Theorem 5.9 and Lemma 5.5, $r \in [\![A]\!] \subseteq \mathsf{SN}$. ◀

5.2 Reduction of a product

When simply-typed lambda-calculus is extended with pairs, proving that if $r_1 \in \mathsf{SN}$ and $r_2 \in \mathsf{SN}$ then $r_1 \times r_2 \in \mathsf{SN}$ is easy. However, in System I this property (Lemma 5.13) is harder to prove, as it requires a characterisation of the terms equivalent to the product $r_1 \times r_2$ (Lemma 5.11) and of all the reducts of this term (Lemma 5.12).

In Lemma 5.11, we characterise the terms equivalent to a product.

▶ **Lemma 5.11.** *If $r \times s \rightleftarrows^* t$ then either*

1. $t = u \times v$ *where either*

   a. $u \rightleftarrows^* t_{11} \times t_{21}$ *and* $v \rightleftarrows^* t_{12} \times t_{22}$ *with* $r \rightleftarrows^* t_{11} \times t_{12}$ *and* $s \rightleftarrows^* t_{21} \times t_{22}$*, or*

   b. $v \rightleftarrows^* w \times s$ *with* $r \rightleftarrows^* u \times w$*, or any of the three symmetric cases, or*

   c. $r \rightleftarrows^* u$ *and* $s \rightleftarrows^* v$*, or the symmetric case.*

2. $t = \lambda x^A.a$ *and* $a \rightleftarrows^* a_1 \times a_2$ *with* $r \rightleftarrows^* \lambda x^A.a_1$ *and* $s \rightleftarrows^* \lambda x^A.a_2$*.*

3. $t = av$ *and* $a \rightleftarrows^* a_1 \times a_2$*, with* $r \rightleftarrows^* a_1 v$ *and* $s \rightleftarrows^* a_2 v$*.*

Proof. By a double induction, first on $M(t)$ and then on the length of the relation $\rightleftarrows^*$ (cf. Appendix B.1). ◀

In Lemma 5.12, we characterise the reducts of a product.

▶ **Lemma 5.12.** *If $r_1 \times r_2 \rightleftarrows^* s \hookrightarrow t$, there exists u_1, u_2 such that $t \rightleftarrows^* u_1 \times u_2$ and either ($r_1 \rightsquigarrow u_1$ and $r_2 \rightsquigarrow u_2$), or ($r_1 \rightsquigarrow u_1$ and $r_2 \rightleftarrows^* u_2$), or ($r_1 \rightleftarrows^* u_1$ and $r_2 \rightsquigarrow u_2$).*

Proof. By induction on $M(r_1 \times r_2)$. ◀

▶ **Lemma 5.13.** *If $r_1 \in \mathsf{SN}$ and $r_2 \in \mathsf{SN}$, then $r_1 \times r_2 \in \mathsf{SN}$.*

Proof. By Lemma 5.12, 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. ◀

5.3 Reduction of a term of a conjunctive type

The next lemma takes advantage of the fact that all the variables have prime types to prove that all terms of conjunctive type, even open ones, reduce to a product. For instance, instead of the term $\lambda x^{\tau\wedge\tau}.x$, of type $((\tau \wedge \tau) \Rightarrow \tau) \wedge ((\tau \wedge \tau) \Rightarrow \tau)$, we must write $\lambda y^\tau.\lambda z^\tau.y \times z$, which is equivalent to $(\lambda y^\tau.\lambda z^\tau.y) \times (\lambda y^\tau.\lambda z^\tau.z)$.

▶ **Lemma 5.14.** *If $r : \bigwedge_{i=1}^n A_i$, then $r \rightsquigarrow^* \prod_{i=1}^n r_i$ where $r_i : A_i$.*

Proof. By induction on r.

- $r = x$, then it has a prime type, so take $r_1 = r$.
- Let $r = \lambda x^C.s$. Then, by Lemma 4.2, $s : D$ with $C \Rightarrow D \equiv \bigwedge_i A_i$. So, by Corollary 2.12, $D \equiv \bigwedge_i D_i$, and so, by the induction hypothesis, $s \rightsquigarrow^* \prod_i s_i$. Therefore, $\lambda x^C.s \rightsquigarrow^* \prod_i \lambda x^C.s_i$.
- Let $r = st$. Then, by Lemma 4.2, $s : C \Rightarrow \bigwedge_i A_i$, so $s : \bigwedge_i (C \Rightarrow A_i)$. Therefore, by the induction hypothesis, $s \rightsquigarrow^* \prod_i s_i$, and so $st \rightsquigarrow^* \prod_i s_i t$.
- Let $r = s \times t$. Then, by Lemma 4.2, $s : B$ and $t : C$, with $B \wedge C \equiv \bigwedge_i A_i$. By Lemma 2.15, there exists a partition $E \uplus F \uplus G$ of $\{1, \dots, n\}$ such that $A_i \equiv B_i \wedge C_i$, when $i \in E$; $A_i \equiv B_i$, when $i \in F$; $A_i \equiv C_i$, when $i \in G$; $B \equiv \bigwedge_{i \in E \uplus F} B_i$; and $C \equiv \bigwedge_{i \in E \uplus G} C_i$. By the induction hypothesis, $s \rightsquigarrow^* \prod_{i \in E \uplus F} s_i$ and $t \rightsquigarrow^* \prod_{i \in E \uplus G} t_i$. If $i \in E$, we let $r_i = s_i \times t_i$, if $i \in F$, we let $r_i = s_i$, if $i \in G$, we let $r_i = t_i$. We have $r = s \times t \rightsquigarrow^* \prod_{i=1}^n r_i$.
- Let $r = \pi_{\bigwedge_i A_i}(s)$. Then, by Lemma 4.2, $s : \bigwedge_i A_i \wedge B$, and hence, by the induction hypothesis, $s \rightsquigarrow^* \prod_i s_i \times t$ where $s_i : A_i$ and $t : B$, hence $r \rightsquigarrow \prod_i s_i$. ◀

▶ **Corollary 5.15.** *If $r : A \wedge B$, then $r \rightsquigarrow^* r_1 \times r_2$ where $r_1 : A$ and $r_2 : B$.*

Proof. Let $\mathsf{PF}(A) = [A_i]_i$, $\mathsf{PF}(B) = [B_j]_j$, by Lemma 2.6, $A \wedge B \equiv \bigwedge_i A_i \wedge \bigwedge_j B_j$. Then, by Lemma 5.14, $r \rightsquigarrow^* \prod_i r_{1i} \times \prod_j r_{2j}$. Take $r_1 = \prod_i r_{1i}$ and $r_2 = \prod_j r_{2j}$. ◀

5.4 Reducibility

▶ **Definition 5.16** (Elimination context). *Consider an extension of the language where we introduce an extra symbol* $[]^A$*, called hole of type* A*. We define the set of elimination contexts with a hole* $[]^A$ *as the smallest set such that:*

- $[]^A$ *is an elimination context of type* A*,*
- *if* $K_A^{B\Rightarrow C}$ *is an elimination context of type* $B \Rightarrow C$ *with a hole of type* A*, and* $r : B$ *then* $K_A^{B\Rightarrow C} r$ *is an elimination context of type* C *with a hole of type* A*,*
- *and if* $K_A^{B\wedge C}$ *is an elimination context of type* $B\wedge C$ *with a hole of type* A*, then* $\pi_B(K_A^{B\wedge C})$ *is an elimination context of type* B *with a hole of type* A*.*

We write $K_A^B[t]$ *for* $K_A^B[t/[]^A]$*, where* $[]^A$ *is the hole of* K_A^B*. In particular,* t *may be an elimination context.*

▶ **Example 5.17.** Let $K_\tau^\tau = []^\tau$ and $K'^{\tau}_{\tau\Rightarrow(\tau\wedge\tau)} = K_\tau^\tau[\pi_\tau([]^{\tau\Rightarrow(\tau\wedge\tau)}x)]$. Then $K'^{\tau}_{\tau\Rightarrow(\tau\wedge\tau)} = \pi_\tau([]^{\tau\Rightarrow(\tau\wedge\tau)}x)$, and $K'^{\tau}_{\tau\Rightarrow(\tau\wedge\tau)}[\lambda y^\tau.y\times y] = \pi_\tau((\lambda y^\tau.y\times y)x)$.

▶ **Definition 5.18** (Terms occurring in an elimination context). *Let* K_A^B *be an elimination context. The multiset of terms occurring in* K_A^B *is defined as*

$$\mathcal{T}([]^A) = \emptyset; \qquad \mathcal{T}(K_A^{B\Rightarrow C} r) = \mathcal{T}(K_A^{B\Rightarrow C}) \uplus \{r\}; \qquad \mathcal{T}(\pi_B(K_A^{B\wedge C})) = \mathcal{T}(K_A^{B\wedge C})$$

We write $|K_A^B|$ *for* $\sum_{i=1}^n |r_i|$ *where* $[r_1,\dots,r_n] = \mathcal{T}(K_A^B)$*.*

▶ **Example 5.19.** $\mathcal{T}([]^A rs) = [r,s]$ and $\mathcal{T}([]^A(r\times s)) = [r\times s]$. Remark that $K_A^B[t] \rightleftarrows^* K'^B_A[t]$ does not imply $\mathcal{T}(K_A^B) \sim \mathcal{T}(K'^B_A)$.

Remark that if K_A^B is a context, $m(B) \leq m(A)$ and hence if a term in $\mathcal{T}(K_A^B)$ has type C, then $m(C) < m(A)$.

▶ **Lemma 5.20.** *Let* K_A^τ *be an elimination context such that* $\mathcal{T}(K_A^\tau) \subseteq \mathsf{SN}$*, let* $\mathsf{PF}(A) = [B_1,\dots,B_n]$*, and let* $x_i \in \mathcal{V}_{B_i}$*. Then* $K_A^\tau[x_1\times\cdots\times x_n] \in \mathsf{SN}$*.*

Proof. By induction on the number of projections in K_A^τ.

- If K_A^τ does not contain any projection, then it has the form $[]^A\ r_1\ ...\ r_m$. Let C_i be the type of r_i, we have $A \equiv C_1 \Rightarrow ... \Rightarrow C_n \Rightarrow \tau$, thus A is prime, $n=1$, and we need to prove that $x_1\ r_1\ ...\ r_m$ is in SN which is a consequence of the fact that r_1, ..., r_n are in SN.
- Otherwise, $K_A^\tau = K'^\tau_B[\pi_B([]^A r_1\dots r_m)]$, and $K_A^\tau[x_1\times\cdots\times x_n] = K'^\tau_B[\pi_B((x_1\times\cdots\times x_n)r_1\dots r_m)] \rightleftarrows^* K'^\tau_B[\pi_B(x_1 r_1\dots r_m\times\cdots\times x_n r_1\dots r_m)]$. We prove that this term is in SN by showing, more generally, that if s_i are reducts of $x_i r_1\dots r_m$, then $K'^\tau_B[\pi_B(s_1\times\cdots\times s_n)] \in \mathsf{SN}$. To do so, we show, by induction on $|K'^\tau_B| + |s_1\times\cdots\times s_n|$, that all the one step reducts of this term are in SN.
  - If the reduction takes place in one of the terms in $\mathcal{T}(K'^\tau_B)$ or in one of the s_i, we apply the induction hypothesis.
  - Otherwise, the reduction is a (π) reduction of $\pi_B(s_1\times\cdots\times s_n)$ yielding, without lost of generality, a term of the form $K'^\tau_B[s_1\times\cdots\times s_q]$. This term is a reduct of $K'^\tau_B[(x_1\times\cdots\times x_q)r_1\dots r_m]$. As the context $K'^\tau_B[([]^C\ r_1\dots r_m)]$ contains one projection less than K_A^τ this term is in SN. Hence so does its reduct $K'^\tau_B[s_1\times\cdots\times s_q]$. ◀

▶ **Definition 5.21** (Reducibility). *The set* $[\![A]\!]$ *of reducible terms of type* A *is defined by induction on* $m(A)$ *as the set of terms* $t : A$ *such that for any elimination context* K_A^τ *such that the terms of* $\mathcal{T}(K_A^\tau)$ *are all reducible, we have* $K_A^\tau[t] \in \mathsf{SN}$*.*

▶ **Definition 5.22** (Reducible elimination context). *An elimination context K_A^B is reducible, if all the terms in $\mathcal{T}(K_A^B)$ are reducible.*

From now on we consider all the elimination contexts to be reducible.

The following lemma is a trivial consequence of the definition of reducibility.

▶ **Lemma 5.23.** *If $A \equiv B$, then $[\![A]\!] = [\![B]\!]$.*

▶ **Lemma 5.24.** *For all A, $[\![A]\!] \subseteq \mathsf{SN}$ and $[\![A]\!] \neq \emptyset$.*

Proof. By induction on $m(A)$. By the induction hypothesis, for all the B such that $m(B) < m(A)$, $[\![B]\!] \neq \emptyset$. Thus, there exists an elimination context K_A^τ. Hence, if $r \in [\![A]\!]$, $K_A^\tau[r] \in \mathsf{SN}$, hence $r \in \mathsf{SN}$.

We then prove that if $\mathsf{PF}(A) = [B_1, \dots, B_n]$ and $x_i \in \mathcal{V}_{B_i}$ then $\prod_i x_i \in [\![A]\!]$. By the induction hypothesis, $\mathcal{T}(K_A^\tau) \subseteq \mathsf{SN}$, hence, by Lemma 5.20, $K_A^\tau[x_1 \times \cdots \times x_n] \in \mathsf{SN}$. ◀

5.5 Adequacy

We finally prove the adequacy theorem (Theorem 5.30) showing that every typed term is reducible, and the strong normalisation theorem (Theorem 5.31) as a consequence of it.

▶ **Lemma 5.25** (Adequacy of projection). *If $r \in [\![A \wedge B]\!]$, then $\pi_A(r) \in [\![A]\!]$.*

Proof. We need to prove that $K_A^\tau[\pi_A(r)] \in \mathsf{SN}$. Take $K'^\tau_{A\wedge B} = K_A^\tau[\pi_A[]^{A\wedge B}]$ and since $r \in [\![A \wedge B]\!]$, we have $K_A^\tau[\pi_A(r)] = K'^\tau_{A\wedge B}[r] \in \mathsf{SN}$. ◀

▶ **Lemma 5.26** (Adequacy of application). *If $r \in [\![A \Rightarrow B]\!]$, and $s \in [\![A]\!]$, then $rs \in [\![B]\!]$.*

Proof. We need to prove that $K_B^\tau[rs] \in \mathsf{SN}$. Take $K'^\tau_{A\Rightarrow B} = K_B^\tau[[]^{A\Rightarrow B}s]$ and since $r \in [\![A \Rightarrow B]\!]$, we have $K_B^\tau[rs] = K'^\tau_{A\Rightarrow B}[r] \in \mathsf{SN}$. ◀

▶ **Lemma 5.27** (Adequacy of product). *If $r \in [\![A]\!]$ and $s \in [\![B]\!]$, then $r \times s \in [\![A \wedge B]\!]$.*

Proof. We need to prove that $K_{A\wedge B}^\tau[r \times s] \in \mathsf{SN}$. We proceed by induction on the number of projections in $K_{A\Rightarrow B}^\tau$. Since the hole of $K_{A\wedge B}^\tau$ has type $A \wedge B$, and $K_{A\wedge B}^\tau[t]$ has type τ for any $t : A$, we can assume, without lost of generality, that the context $K_{A\wedge B}^\tau$ has the form $K'^\tau_C[\pi_C([]^{A\wedge B}t_1 \dots t_n)]$. We prove that all $K'^\tau_C[\pi_C(rt_1 \dots t_n \times st_1 \dots t_n)] \in \mathsf{SN}$ by showing, more generally, that if r' and s' are two reducts of $rt_1 \dots t_n$ and $st_1 \dots t_n$, then $K'^\tau_C[\pi_C(r' \times s')] \in \mathsf{SN}$. For this, we show that all its one step reducts are in SN, by induction on $|K'^\tau_C| + |r'| + |s'|$. Full details are given in Appendix B.2. ◀

▶ **Lemma 5.28** (Adequacy of abstraction). *If for all $t \in [\![A]\!]$, $r[t/x] \in [\![B]\!]$, then $\lambda x^A.r \in [\![A \Rightarrow B]\!]$.*

Proof. By induction on $M(r)$. In the case $r \not\rightleftarrows^* r_1 \times r_2$, we need to prove that for any elimination context $K_{A\Rightarrow B}^\tau$, we have $K_{A\Rightarrow B}^\tau[\lambda x^A.r] \in \mathsf{SN}$, and we do so by a second induction on $|K_{A\Rightarrow B}^\tau| + |r|$ to show that all the one step reducts of $K_{A\Rightarrow B}^\tau[\lambda x^A.r]$ are in SN. Full details are given in Appendix B.2. ◀

▶ **Definition 5.29** (Adequate substitution). *A substitution σ is adequate if for all $x \in \mathcal{V}_A$, we have $\sigma(x) \in [\![A]\!]$.*

▶ **Theorem 5.30** (Adequacy). *If $r : A$, then for all σ adequate, we have $\sigma r \in [\![A]\!]$.*

Proof. By induction on r. Full details are given in Appendix B.2. ◀

▶ **Theorem 5.31** (Strong normalisation). *If $r : A$, then $r \in \mathsf{SN}$.*

Proof. By Lemma 5.24, the identity substitution is adequate. Thus, by Theorem 5.30 and Lemma 5.24, $r \in [\![A]\!] \subseteq \mathsf{SN}$. ◀

6 Consistency

▶ **Lemma 6.1.** *For any term $r : A$ there exists an elimination context K_B^A and a term $s : B$, which is not an elimination, such that $r = K_B^A[s]$.*

Proof. We proceed by structural induction on r.

- If r is a variable, an abstraction, or a product, we take $s = r$ and $K_A^A = []^A$.
- If r is an application $r_1 r_2$, by the induction hypothesis, $r_1 = K_A^{C \Rightarrow B}[s]$, we take $K_A'^B = K_A^{C\Rightarrow B} r_2$.
- If r is a projection $\pi_A(r')$, by the induction hypothesis, $r' = K_B^{A\wedge C}[s]$, we take $K_B'^A = \pi_A(K_B^{A\wedge C})$. ◀

▶ **Corollary 6.2.** *There is no closed normal term of type τ.*

Proof. Let $r : \tau$ be a closed normal term. By Lemma 6.1, any $r = K_A^\tau[s]$, where s is not an elimination. Since the term is closed, s is not a variable. Thus it is either and abstraction or a product.

- If A is prime, then, K_A^τ cannot contain a projection, so by rule (curry) we have $K_A^\tau \rightleftarrows^* []^A t$, with $t : B$, and s has the form $\lambda x^C.s'$ with $s' : D \Rightarrow \tau$. We have $B \equiv C \wedge D$. By Corollary 5.15, and since t is normal, $t \rightleftarrows^* t_1 \times t_2$ where $t_1 : C$ and $t_2 : D$, so $K_A^\tau \rightleftarrows^* []^A t_1 t_2$, hence, $r = K_A^\tau[\lambda x^C.s'] \rightleftarrows^* (\lambda x^C.s')t_1 t_2$ is not normal.
- Otherwise, $K_A^\tau = K_B'^\tau[\pi_B([]^A t_1 \dots t_n)]$, with $[]^A t_1 \dots t_n : B \wedge C$. Then, by Corollary 5.15, and since $st_1 \dots t_n$ is normal, $st_1 \dots t_n \rightleftarrows^* s_1 \times s_2$, thus $r = K_A^\tau[s] \rightleftarrows^* K_B'^\tau[\pi_B(s_1 \times s_2)]$, which is not normal. ◀

7 Computing with System I

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 to encode the pairs $\langle r, s\rangle : A \times A$ as $\lambda x^{\mathbb{1}}.r \times \lambda x^{\mathbb{2}}.s : \mathbb{1} \Rightarrow A \wedge \mathbb{2} \Rightarrow A$ and the projection $\pi_1\langle r, s\rangle$ as $\pi_{\mathbb{1}\Rightarrow A}(\lambda x^{\mathbb{1}}.r \times \lambda x^{\mathbb{2}}.s)y^{\mathbb{1}}$ (similarly for π_2), 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 of $\times$, the encoding can be easily extended to lists.

Example 3.10 on booleans overlooks an interesting fact: If $A \equiv B$, then both $\mathbf{T}$ and $\mathbf{F}$ behave as a non-deterministic projector. Indeed, $\mathbf{T}rs \hookrightarrow^* r$, but also $(\lambda x^A.\lambda y^A.x)rs \rightleftarrows (\lambda x^A.\lambda y^A.x)(r \times s) \rightleftarrows (\lambda x^A.\lambda y^A.x)(s \times r) \rightleftarrows (\lambda x^A.\lambda y^A.x)sr \hookrightarrow^* s$. 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, we 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: $\mathbb{tt} := \lambda x^A.\lambda y^{(\tau\Rightarrow\tau)\Rightarrow A}.x$ and $\mathbb{ff} := \lambda x^{(\tau\Rightarrow\tau)\Rightarrow A}.\lambda y^A.\{x\}^{\tau\Rightarrow\tau}$. Hence, it is possible to encode an if-then-else conditional expression as $\mathsf{If}\ \mathsf{c}\ \mathsf{then}\ \mathsf{r}\ \mathsf{else}\ \mathsf{s} := \mathsf{cr}[\mathsf{s}]^{\tau\Rightarrow\tau}$. Thus, $\mathbb{tt}r[s]^{\tau\Rightarrow\tau} \hookrightarrow^* r$, while $\mathbb{ff}r[s]^{\tau\Rightarrow\tau} \rightleftarrows^* \mathbb{ff}[s]^{\tau\Rightarrow\tau}r \hookrightarrow^* \{[s]^{\tau\Rightarrow\tau}\}^{\tau\Rightarrow\tau} \hookrightarrow s$.

8 Conclusion, Discussion and Future Work

In this paper we have defined System I, a proof system for propositional logic, where isomorphic propositions have the same proofs.

8.1 Non-terminating extension

As mentioned in the introduction, the choice of rules is subtle. Indeed, as well known, the strong normalisation of simply typed lambda calculus is not a very robust property: minor modifications of typing or reduction rules can lead to non-terminating calculi, see for instance [18]. In System I, we have the rule $(\textsc{dist}_{\mathsf{app}})$ to deal with the equivalence $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$, and we could have also considered a rule such as $\pi_A(rs) \rightleftarrows \pi_{B\Rightarrow A}(r)s$ [13]. However, adding such a rule leads to a non-terminating calculus, as shown by the following example. Let $\delta = \lambda x^{(\tau\Rightarrow\tau)\wedge\tau}.\pi_{\tau\Rightarrow\tau}(x)\pi_\tau(x) : ((\tau \Rightarrow \tau) \wedge \tau) \Rightarrow \tau$, $\delta' = \delta((z^{\tau\Rightarrow\tau\Rightarrow\tau}y^\tau) \times y^\tau) : \tau$, and $\Omega = \delta((z^{\tau\Rightarrow\tau\Rightarrow\tau}y^\tau) \times \delta') : \tau$. Then, we have

$$\begin{aligned}
&\Omega\\
&\hookrightarrow \pi_{\tau\Rightarrow\tau}((zy) \times \delta')\pi_\tau((zy) \times \delta') \hookrightarrow \pi_{\tau\Rightarrow\tau}((zy) \times \delta')\delta' = \pi_{\tau\Rightarrow\tau}((zy) \times (\delta((zy) \times y)))\delta'\\
&\rightleftarrows \pi_{\tau\Rightarrow\tau}((zy) \times (\delta(zy)y))\delta' \rightleftarrows \pi_{\tau\Rightarrow\tau}((z \times (\delta(zy)))y)\delta' \overset{(\textsc{wrong-rule})}{\rightleftarrows^*} \pi_{\tau\Rightarrow\tau}((z \times (\delta(zy)))\delta')y\\
&\rightleftarrows \pi_{\tau\Rightarrow\tau}((z\delta') \times (\delta(zy)\delta'))y \rightleftarrows \pi_{\tau\Rightarrow\tau}((z\delta') \times (\delta((zy) \times \delta')))y = \pi_{\tau\Rightarrow\tau}((z\delta') \times \Omega)y
\end{aligned}$$

8.2 Other Related Work

Apart from the related work already discussed in the introduction, in a work by Garrigue and Aït-Kaci [20], the selective λ-calculus has been presented, where only the isomorphism

$$(A \Rightarrow B \Rightarrow C) \equiv (B \Rightarrow A \Rightarrow C). \tag{5}$$

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 currification 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).

8.3 Towards 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 $\top \Rightarrow A \equiv A$. 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 $(\lambda x^A.r)s \hookrightarrow r[\star/x]$", where $\star : \top$ is the introduction rule of $\top$.

8.4 Eta-expansion rule

In [15] we have given an implementation embedded in Haskell of an extended fragment of the system as presented in [13], which is an early version of System I. In such an implementation, we have added some rules in order to have only introductions as normal forms. For example, "If $s : B$ then $(\lambda x^A.\lambda y^B.r)s \hookrightarrow \lambda x^A.((\lambda y^B.r)s)$. Such a rule, among others introduced in this implementation, is a particular case of a more general η-expansion rule. Indeed, with the rule" "If $t : A \Rightarrow B$ then $t \hookrightarrow \lambda x^A.tx$" we can derive $(\lambda x^A.\lambda y^B.r)s \hookrightarrow \lambda z^A.(\lambda x^A.\lambda y^B.r)sz \rightleftarrows^* \lambda z^A.(\lambda x^A.\lambda y^B.r)zs \hookrightarrow \lambda z^A.((\lambda y^B.r[z/x])s)$.

Indeed, we conjecture that System I extended with an η-expansion rule would lead to a system where there is no closed elimination term in normal form. Such an extension is left for future work.

References

1 Pablo Arrighi and Alejandro Díaz-Caro. A System F Accounting for Scalars. *Logical Methods in Computer Science*, 8(1:11), 2012.

2 Pablo Arrighi, Alejandro Díaz-Caro, and Benoît Valiron. The Vectorial Lambda-Calculus. *Information and Computation*, 254(1):105–139, 2017.

3 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.

4 Pablo Arrighi and Gilles Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1:8), 2017.

5 Gérard Boudol. Lambda-Calculi for (Strict) Parallel Functions. *Information and Computation*, 108(1):51–127, 1994.

6 Kim B. Bruce, Roberto Di Cosmo, and Giuseppe Longo. Provable isomorphisms of types. *Mathematical Structures in Computer Science*, 2(2):231–247, 1992.

7 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.

8 Thierry Coquand and Gérard Huet. The Calculus of Constructions. *Information and Computation*, 76(2–3):95–120, 1988.

9 Ugo de'Liguoro and Adolfo Piperno. Non Deterministic Extensions of Untyped λ-calculus. *Information and Computation*, 122(2):149–177, 1995.

10 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.

11 Roberto Di Cosmo. *Isomorphisms of types: from λ-calculus to information retrieval and language design*. Progress in Theoretical Computer Science. Birkhauser, 1995.

12 Roberto Di Cosmo. A short survey of isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5):825–838, 2005.

13 Alejandro Díaz-Caro and Gilles Dowek. Non determinism through type isomorphism. In Delia Kesner and Petrucio Viana, editors, *Proceedings of LSFA 2012*, volume 113 of *EPTCS*, pages 137–144, 2013.

14 Alejandro Díaz-Caro and Gilles Dowek. Typing quantum superpositions and measurement. In Carlos Martín-Vide, Roman Neruda, and Miguel A. Vega-Rodríguez, editors, *Proceedings of TPNC 2017*, volume 10687 of *LNCS*, pages 281–293, 2017.

15 Alejandro Díaz-Caro and Pablo E. Martínez López. 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.

16 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.

17 Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1):33–72, 2003.

18 Gilles Dowek and Ying Jiang. On the expressive power of schemes. *Information and Computation*, 209:1231–1245, 2011.

19 Gilles Dowek and Benjamin Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.

20 Jacques Garrigue and Hassan Aït-Kaci. The typed polymorphic label-selective λ-calculus. In *Proceedings of POPL 1994*, ACM SIGPLAN, pages 35–47, 1994.

21 Herman Geuvers, Robbert Krebbers, James McKinna, and Freek Wiedijk. Pure Type Systems without Explicit Contexts. In Karl Crary and Marino Miculan, editors, *Proceedings of LFMTP 2010*, volume 34 of *EPTCS*, pages 53–67, 2010.

22 Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, 1989.

23 Per Martin-Löf. *Intuitionistic type theory*. Studies in proof theory. Bibliopolis, 1984.

24 Michele Pagani and Simona Ronchi Della Rocca. Linearity, non-determinism and solvability. *Fundamental Informaticae*, 103(1–4):173–202, 2010.

25 Jonghyun Park, Jeongbong Seo, Sungwoo Park, and Gyesik Lee. Mechanizing Metatheory without Typing Contexts. *Journal of Automated Reasoning*, 52(2):215–239, 2014.

26 Mikael Rittri. Retrieving library identifiers via equational matching of types. In *Proceedings of CADE 1990*, volume 449 of *LNCS*, pages 603–617, 1990.

27 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

28 Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

A Detailed proofs of Section 4

▶ **Lemma 4.3** (Substitution). *If $r : A$, $s : B$, and $x \in \mathcal{V}_B$, then $r[s/x] : A$.*

Proof. By structural induction on r.

- Let $r = x$. By Lemma 4.2, $A \equiv B$, thus $s : A$. We have $x[s/x] = s$, so $x[s/x] : A$.
- Let $r = y$, with $y \neq x$. We have $y[s/x] = y$, so $y[s/x] : A$.
- Let $r = \lambda y^C.r'$. By Lemma 4.2, $A \equiv C \Rightarrow D$, with $r' : D$. By the induction hypothesis $r'[s/x] : D$, and so, by rule $(\Rightarrow_i)$, $\lambda y^C.r'[s/x] : C \Rightarrow D$. Since $\lambda y^C.r'[s/x] = (\lambda y^C.r')[s/x]$, using rule $(\equiv)$, $(\lambda y^C.r')[s/x] : A$.
- Let $r = r_1 r_2$. By Lemma 4.2, $r_1 : C \Rightarrow A$ and $r_2 : C$. By the induction hypothesis $r_1[s/x] : C \Rightarrow A$ and $r_2[s/x] : C$, and so, by rule $(\Rightarrow_e)$, $(r_1[s/x])(r_2[s/x]) : A$. Since $(r_1[s/x])(r_2[s/x]) = (r_1 r_2)[s/x]$, we have $(r_1 r_2)[s/x] : A$.
- 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 $r_1[s/x] : A_1$ and $r_2[s/x] : A_2$, and so, by rule $(\wedge_i)$, $(r_1[s/x]) \times (r_2[s/x]) : A_1 \wedge A_2$. Since $(r_1[s/x]) \times (r_2[s/x]) = (r_1 \times r_2)[s/x]$, using rule $(\equiv)$, we have $(r_1 \times r_2)[s/x] : A$.
- Let $r = \pi_A(r')$. By Lemma 4.2, $r' : A \wedge C$. Hence, by the induction hypothesis, $r'[s/x] : A \wedge C$. Hence, by rule $\wedge_e$, $\pi_A(r'[s/x]) : A$. Since $\pi_A(r'[s/x]) = \pi_A(r')[s/x]$, we have $\pi_A(r')[s/x] : A$. ◀

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

Proof. By induction on the rewrite relation.

- (COMM): 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, $s \times r : A_2 \wedge A_1 \equiv A$.
- (ASSO):
  ($\rightarrow$) 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, $r \times (s \times t) : A_1 \wedge (A_2 \wedge A_3) \equiv A$.
  ($\leftarrow$) Analogous to ($\rightarrow$).
- (DIST$_\lambda$):
  ($\rightarrow$) 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, $\lambda x^B.r \times \lambda x^B.s : (B \Rightarrow C_1) \wedge (B \Rightarrow C_2) \equiv A$.
  ($\leftarrow$) 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, $\lambda x^B.(r \times s) : B \Rightarrow (C_1 \wedge C_2) \equiv A$.
- (DIST$_{\mathsf{app}}$):
  ($\rightarrow$) 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 2.11, $A \equiv A_1 \wedge A_2$, with $r : B \Rightarrow A_1$ and $s : B \Rightarrow A_2$. Then, $rt \times st : A_1 \wedge A_2 \equiv A$.
  ($\leftarrow$) 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 $(r \times s)t : A_1 \wedge A_2 \equiv A$.
- (CURRY):
  ($\rightarrow$) 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, $r(s \times t) : A$.
  ($\leftarrow$) 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 $rst : A$.
- (β): 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, $r[s/x^B] : A$.
- (π): If $\pi_B(r \times s) : A$, then by Lemma 4.2, $A \equiv B$, and so, by rule ($\equiv$), $r : A$.
- Contextual closure: Let $t \to r$, where $\to$ is either $\rightleftarrows$ or $\hookrightarrow$.
  - Let $\lambda x^B.t \to \lambda x^B.r$: 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 $\lambda x^B.r : B \Rightarrow C \equiv A$.
  - Let $ts \to rs$: 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 $rs : A$.
  - Let $st \to st$: If $st : A$ then by Lemma 4.2, $s : B \Rightarrow A$ and $t : B$, hence by the induction hypothesis $r : B$ and so $sr : A$.
  - Let $t \times s \to r \times s$: 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 $r \times s : A_1 \wedge A_2 \equiv A$.
  - Let $s \times t \to s \times r$: Analogous to previous case.
  - Let $\pi_B(t) \to \pi_B(r)$: 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, $\pi_B(r) : B \equiv A$. ◀

B Detailed proofs of Section 5

B.1 Detailed proofs of Section 5.2

▶ **Lemma 5.11.** *If $r \times s \rightleftarrows^* t$ then either*

1. *$t = u \times v$ where either*
   a. *$u \rightleftarrows^* t_{11} \times t_{21}$ and $v \rightleftarrows^* t_{12} \times t_{22}$ with $r \rightleftarrows^* t_{11} \times t_{12}$ and $s \rightleftarrows^* t_{21} \times t_{22}$, or*
   b. *$v \rightleftarrows^* w \times s$ with $r \rightleftarrows^* u \times w$, or any of the three symmetric cases, or*
   c. *$r \rightleftarrows^* u$ and $s \rightleftarrows^* v$, or the symmetric case.*

2. $t = \lambda x^A.a$ *and* $a \rightleftarrows^* a_1 \times a_2$ *with* $r \rightleftarrows^* \lambda x^A.a_1$ *and* $s \rightleftarrows^* \lambda x^A.a_2$.
3. $t = av$ *and* $a \rightleftarrows^* a_1 \times a_2$, *with* $r \rightleftarrows^* a_1 v$ *and* $s \rightleftarrows^* a_2 v$.

Proof. By a double induction, first on $M(t)$ and then on the length of the relation $\rightleftarrows^*$. Consider an equivalence proof $r \times s \rightleftarrows^* t' \rightleftarrows t$ with a shorter proof $r \times s \rightleftarrows^* t'$. 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 $r \times s \rightleftarrows^* u \times v \rightleftarrows t$. The possible equivalences from $u \times v$ are
   - $t = u' \times v$ or $u \times v'$ with $u \rightleftarrows u'$ and $v \rightleftarrows v'$, and so the term t is in case 1.
   - Rules (COMM) and (ASSO) preserve the conditions of case 1.
   - $t = \lambda x^A.(u' \times v')$, with $u = \lambda x^A.u'$ and $v = \lambda x^A.v'$. By the first induction hypothesis (since $M(u) < M(t)$ and $M(v) < M(t)$), either
     a. $u \rightleftarrows^* w_{11} \times w_{21}$ and $v \rightleftarrows^* w_{12} \times w_{22}$, by the first induction hypothesis, $w_{ij} \rightleftarrows^* \lambda x^A.t_{ij}$ for $i = 1, 2$ and $j = 1, 2$, with $u' \rightleftarrows^* t_{11} \times t_{21}$ and $v' \rightleftarrows^* t_{12} \times t_{22}$, so $u' \times v' \rightleftarrows^* t_{11} \times t_{12} \times t_{21} \times t_{22}$. Hence, $r \rightleftarrows^* \lambda x^A.(t_{11} \times t_{12})$ and $s \rightleftarrows^* \lambda x^A.(t_{21} \times t_{22})$, and hence the term t is in case 2.
     b. $v \rightleftarrows^* w \times s$ and $r \rightleftarrows^* u \times w$. Since $v \rightleftarrows^* \lambda x^A.v'$, by the first induction hypothesis, $w \rightleftarrows^* \lambda x^A.t_1$ and $s \rightleftarrows^* \lambda x^A.t_2$, with $v' \rightleftarrows^* t_1 \times t_2$. Hence, $r \rightleftarrows^* \lambda x.(u' \times t_1)$, and hence the term t is in case 2.
     c. $r \rightleftarrows^* \lambda x^A.u'$ and $s \rightleftarrows^* \lambda x^A.v$, and hence the term t is 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 $M(v) < M(t)$), either
     a. $u \rightleftarrows^* w_{11} \times w_{21}$, $v \rightleftarrows^* w_{12} \times w_{22}$, $r \rightleftarrows^* w_{11} \times w_{12}$, and $s \rightleftarrows^* w_{21} \times w_{22}$. By the first induction hypothesis (since $M(w_{ij}) < M(t)$), $w_{ij} \rightleftarrows^* t_{ij}t'$, for $i = 1, 2$ and $j = 1, 2$, where $u' \rightleftarrows^* t_{11} \times t_{21}$, $v' \rightleftarrows^* t_{12} \times t_{22}$, $r \rightleftarrows^* w_{11} \times w_{12}$ and $s \rightleftarrows^* w_{21} \times w_{22}$. Therefore, $u \rightleftarrows^* t_{11}t' \times t_{21}t'$ and $v \rightleftarrows^* t_{12}t' \times t_{22}t'$ with $r \rightleftarrows^* (t_{11} \times t_{12})t'$ and $s \rightleftarrows^* (t_{21} \times t_{22})t'$, and hence the term t is in case 3.
     b. $v't' \rightleftarrows^* w \times s$ and $r \rightleftarrows^* u't' \times w$. By the first induction hypothesis on $v't' \rightleftarrows^* w \times s$ (since $M(w) < M(t)$ and $M(s) < M(t)$), we have $w \rightleftarrows^* t_1 t'$ and $s \rightleftarrows^* t_2 t'$ with $t_1 \times t_2 \rightleftarrows^* v'$. Therefore, $v \rightleftarrows^* t_1 t' \times t_2 t'$ with $r \rightleftarrows^* (u \times t_1)t'$ and $s \rightleftarrows^* t_2 t'$, and $u' \times v' \rightleftarrows^* u' \times t_1 \times t_2$, hence the term t is in case 3.
     c. $r \rightleftarrows^* u't'$ and $s \rightleftarrows^* v't'$, and hence we are in case 3.

     (the symmetric cases are analogous).
2. Let $r \times s \rightleftarrows^* \lambda x^A.a \rightleftarrows t$, with $a \rightleftarrows^* a_1 \times a_2$, $r \rightleftarrows^* \lambda x^A.a_1$, and $s \rightleftarrows^* \lambda x^A.a_2$. Hence, possible equivalences from $\lambda x.a$ to t are
   - $t = \lambda x^A.a'$ with $a \rightleftarrows^* a'$, hence $a' \rightleftarrows^* a_1 \times a_2$, and so the term t is in case 2.
   - $t = \lambda x^A.u \times \lambda x^A.v$, with $a_1 \times a_2 \rightleftarrows^* a = u \times v$. Hence, by the first induction hypothesis (since $M(a) < M(t)$), either
     a. $a_1 \rightleftarrows^* u$ and $a_2 \rightleftarrows^* v$, and so $r \rightleftarrows^* \lambda x^A.u$ and $s \rightleftarrows^* \lambda x^A.v$, or
     b. $v \rightleftarrows^* t_1 \times t_2$ with $a_1 \rightleftarrows^* u \times t_1$ and $a_2 \rightleftarrows^* t_2$, and so $\lambda x^A.v \rightleftarrows^* \lambda x.t_1 \times \lambda x^A.t_2$, $r \rightleftarrows^* \lambda x^A.u \times \lambda x^A.t_1$ and $s \rightleftarrows^* \lambda x^A.t_2$, or
     c. $u \rightleftarrows^* t_{11} \times t_{21}$ and $v \rightleftarrows^* t_{12} \times t_{22}$ with $a_1 \rightleftarrows^* t_{11} \times t_{12}$ and $a_2 \rightleftarrows^* t_{21} \times t_{22}$, and so $\lambda x^A.u \rightleftarrows^* \lambda x^A.t_{11} \times \lambda x^A.t_{21}$, $\lambda x.v \rightleftarrows^* \lambda x^A.t_{12} \times \lambda x^A.t_{22}$, $r \rightleftarrows^* \lambda x^A.t_{11} \times \lambda x^A.t_{12}$ and $s \rightleftarrows^* \lambda x^A.t_{21} \times \lambda x^A.t_{22}$.

     (the symmetric cases are analogous), and so the term t is in case 1.
3. Let $r \times s \rightleftarrows^* aw \rightleftarrows t$, with $a \rightleftarrows^* a_1 \times a_2$, $r \rightleftarrows^* a_1 w$, and $s \rightleftarrows^* a_2 w$. The possible equivalences from aw to t are

- $t = a'w$ with $a \rightleftarrows^* a'$, hence $a' \rightleftarrows^* a_1 \times a_2$, and so the term t is in case 3.
- $t = aw'$ with $w \rightleftarrows^* w'$ and so the term t is in case 3.
- $t = uw \times vw$, with $a_1 \times a_2 \rightleftarrows^* a = u \times v$. Hence, by the first induction hypothesis (since $M(a) < M(t)$), either
  a. $a_1 \rightleftarrows^* u$ and $a_2 \rightleftarrows^* v$, and so $r \rightleftarrows^* uw$ and $s \rightleftarrows^* vw$, or
  b. $v \rightleftarrows^* t_1 \times t_2$ with $a_1 \rightleftarrows^* u \times t_1$ and $a_2 \rightleftarrows^* t_2$, and so $vw \rightleftarrows^* t_1 w \times t_2 w$, $r \rightleftarrows^* uw \times t_1 w$ and $s \rightleftarrows^* t_2 w$, or
  c. $u \rightleftarrows^* t_{11} \times t_{21}$ and $v \rightleftarrows^* t_{12} \times t_{22}$ with $a_1 \rightleftarrows^* t_{11} \times t_{12}$ and $a_2 \rightleftarrows^* t_{21} \times t_{22}$, and so $uw \rightleftarrows^* t_{11} w \times t_{21} w$, $vw \rightleftarrows^* t_{12} w \times t_{22} w$, $r \rightleftarrows^* t_{11} w \times t_{12} w$ and $s \rightleftarrows^* t_{21} w \times t_{22} w$.

  (the symmetric cases are analogous), and so the term t is in case 1.
- $t = a'(v \times w)$ with $a = a'v$, thus $a'v = a \rightleftarrows^* a_1 \times a_2$. Hence, by the first induction hypothesis, $a' \rightleftarrows^* a'_1 \times a'_2$, with $a_1 \rightleftarrows^* a'_1 v$ and $a_2 \rightleftarrows^* a'_2 v$. Therefore, $r \rightleftarrows^* a'_1(v \times w)$ and $s \rightleftarrows^* a'_2(v \times w)$, and so the term t is in case 3. ◀

▶ **Lemma 5.12.** *If $r_1 \times r_2 \rightleftarrows^* s \hookrightarrow t$, there exists u_1, u_2 such that $t \rightleftarrows^* u_1 \times u_2$ and either ($r_1 \rightsquigarrow u_1$ and $r_2 \rightsquigarrow u_2$), or ($r_1 \rightsquigarrow u_1$ and $r_2 \rightleftarrows^* u_2$), or ($r_1 \rightleftarrows^* u_1$ and $r_2 \rightsquigarrow u_2$).*

Proof. By induction on $M(r_1 \times r_2)$. By Lemma 5.11, 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

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

Notice that rule (π) cannot apply since $s \not\rightleftarrows^* \pi_C(s')$.

We consider each case:

- $s = (\lambda x^A.a)s'$ and $t = a[s'/x]$. Using twice Lemma 5.11, we have $a \rightleftarrows^* a_1 \times a_2$, $r_1 \rightleftarrows^* (\lambda x^A.a_1)s'$ and $r_2 \rightleftarrows^* (\lambda x^A.a_2)s'$. 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 $u_2 = s_2$.
- $s = \lambda x^A.a$, $t = \lambda x^A.t'$, and $a \hookrightarrow t'$, with $a \rightleftarrows^* a_1 \times a_2$ and $s \rightleftarrows^* \lambda x^A.a_1 \times \lambda x^A.a_2$. 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 $u_1 = \lambda x^A.u'_1$ and $u_2 = \lambda x^A.u'_2$.
- $s = as'$, $t = t's'$, and $a \hookrightarrow t'$, with $a \rightleftarrows^* a_1 \times a_2$ and $s \rightleftarrows^* a_1 s' \times a_2 s'$. 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 $u_1 = u'_1 s'$ and $u_2 = u'_2 s'$.

- $s = as'$, $t = at'$, and $s' \hookrightarrow t'$, with $a \rightleftarrows^* a_1 \times a_2$ and $s \rightleftarrows^* a_1 s' \times a_2 s'$. By Lemma 5.11 several times, one the following cases happen
  (a) $a_1 s' \rightleftarrows^* w_{11}s' \times w_{12}s'$, $a_2 s' \rightleftarrows^* w_{21}s' \times w_{22}s'$, $r_1 \rightleftarrows^* w_{11}s' \times w_{21}s'$ 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) $a_2 s' \rightleftarrows^* w_1 s' \times w_2 s'$, $r_1 \rightleftarrows^* a_1 s' \times w_2 s'$ and $r_2 \rightleftarrows^* w_2 s'$. So we take $u_1 = (a_1 \times w_1)t'$ and $u_2 = w_2 t'$, the symmetric cases are analogous.
  (c) $r_1 \rightleftarrows^* a_1 s'$ and $r_2 \rightleftarrows^* a_2 s'$, in which case we take $u_1 = a_1 t'$ and $u_2 = a_2 t'$ the symmetric case is analogous. ◀

B.2 Detailed proofs of Section 5.5

▶ **Lemma 5.27** (Adequacy of product). *If $r \in [\![A]\!]$ and $s \in [\![B]\!]$, then $r \times s \in [\![A \wedge B]\!]$.*

Proof. We need to prove that $K^{\tau}_{A\wedge B}[r \times s] \in \mathsf{SN}$. We proceed by induction on the number of projections in $K^{\tau}_{A\wedge B}$. Since the hole of $K^{\tau}_{A\wedge B}$ has type $A \wedge B$, and $K^{\tau}_{A\wedge B}[t]$ has type τ for any $t : A$ there is at least one projection.

As $r \in [\![A]\!]$, for any elimination context K'^{τ}_{A}, we have $K'^{\tau}_{A}[r] \in \mathsf{SN}$, but then if $A \equiv B_1 \Rightarrow \cdots \Rightarrow B_n \Rightarrow C$, we also have $K''^{\tau}_{C}[rt_1 \dots t_n] \in \mathsf{SN}$, thus $rt_1 \dots t_n \in [\![C]\!]$. Similarly, since $s \in [\![B]\!]$, $st_1 \dots t_n$ is reducible.

We prove that $K'^{\tau}_{C}[\pi_C(rt_1 \dots t_n \times st_1 \dots t_n)] \in \mathsf{SN}$ by showing, more generally, that if r' and s' are two reducts of $rt_1 \dots t_n$ and $st_1 \dots t_n$, then $K'^{\tau}_{C}[\pi_C(r' \times s')] \in \mathsf{SN}$. For this, we show that all its one step reducts are in SN, by induction on $|K'^{\tau}_{C}| + |r'| + |s'|$.

- If the reduction takes place in one of the terms in $\mathcal{T}(K'^{\tau}_{C})$, in r', or in s', we apply the induction hypothesis.
- Otherwise, the reduction is a (π) reduction of $\pi_C(r' \times s')$, that is, $r' \times s' \rightleftarrows^* v \times w$, the reduct is v, and we need to prove $K'^{\tau}_{C}[v] \in \mathsf{SN}$. By Lemma 5.11, we have either:
  - $v \rightleftarrows^* r_1 \times s_1$, with $r' \rightleftarrows^* r_1 \times r_2$ and $s' \rightleftarrows^* s_1 \times s_2$. In such a case, by Lemma 5.25, v is the product of two reducible terms, so since there is one projection less than in $K^{\tau}_{A\wedge B}$, the first induction hypothesis applies.
  - $v \rightleftarrows^* r' \times s_1$, with $s' \rightleftarrows^* s_1 \times s_2$. In such a case, by Lemma 5.25, v is the product of two reducible terms, so since there is one projection less than in $K^{\tau}_{A\wedge B}$, the first induction hypothesis applies.
  - $v \rightleftarrows^* r_1 \times s'$, with $r' \rightleftarrows^* r_1 \times r_2$. In such a case, by Lemma 5.25, v is the product of two reducible terms, so since there is one projection less than in $K^{\tau}_{A\wedge B}$, the first induction hypothesis applies.
  - $v \rightleftarrows^* r'$, in which case, $C \equiv A$, and since $r \in [\![A]\!]$, we have $K'^{\tau}_{A}[r'] \rightleftarrows^* K'^{\tau}_{A}[v] \in \mathsf{SN}$.
  - $v \rightleftarrows^* r_1$ with $r' \rightleftarrows^* r_1 \times r_2$, in which case, since $r \in [\![A]\!]$, we have $K'^{\tau}_{C}[\pi_C(r')] \in \mathsf{SN}$ and $K'^{\tau}_{C}[\pi_C(r')] \rightsquigarrow K'^{\tau}_{C}[v]$ hence $K'^{\tau}_{C}[v] \in \mathsf{SN}$.
  - $v \rightleftarrows^* s'$, in which case, $C \equiv B$, and since $s \in [\![B]\!]$, we have $K'^{\tau}_{B}[s'] \rightleftarrows^* K'^{\tau}_{B}[v] \in \mathsf{SN}$.
  - $v \rightleftarrows^* s_1$ with $s' \rightleftarrows^* s_1 \times s_2$, in which case, since $s \in [\![B]\!]$, we have $K'^{\tau}_{C}[\pi_C(s')] \in \mathsf{SN}$ and $K'^{\tau}_{C}[\pi_C(s')] \rightsquigarrow K'^{\tau}_{C}[v]$ hence $K'^{\tau}_{C}[v] \in \mathsf{SN}$. ◀

▶ **Lemma 5.28** (Adequacy of abstraction). *If for all $t \in [\![A]\!]$, $r[t/x] \in [\![B]\!]$, then $\lambda x^A.r \in [\![A \Rightarrow B]\!]$.*

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

If $r \rightleftarrows^* r_1 \times r_2$, by Lemma 4.2, $B \equiv B_1 \wedge B_2$ with $r_1 : B_1$ and $r_2 : B_2$. and so by Lemma 4.3, $r_1[t/x] : B_1$ and $r_2[t/x] : B_2$. Since $r[t/x] \in [\![B]\!]$, we have $r_1[t/x] \times r_2[t/x] \in [\![B]\!]$. By Lemma 5.25, $r_1[t/x] \in [\![B_1]\!]$ and $r_2[t/x] \in [\![B_2]\!]$. By the induction hypothesis, $\lambda x^A.r_1 \in$

$[\![A \Rightarrow B_1]\!]$ and $\lambda x^A.r_2 \in [\![A \Rightarrow B_2]\!]$, then by Lemma 5.27, $\lambda x^A.r \rightleftarrows^* \lambda x^A.r_1 \times \lambda x^A.r_2 \in [\![(A \Rightarrow B_1) \wedge (A \Rightarrow B_2)]\!]$, and by Lemma 5.23, $[\![(A \Rightarrow B_1) \wedge (A \Rightarrow B_2)]\!] = [\![A \Rightarrow B]\!]$.

If $r \not\rightleftarrows^* r_1 \times r_2$, we need to prove that for any elimination context $K^\tau_{A\Rightarrow B}$, we have $K^\tau_{A\Rightarrow B}[\lambda x^A.r] \in \mathsf{SN}$.

Since r and all the terms in $\mathcal{T}(K^\tau_{A\Rightarrow B})$ are reducible, then they are in SN, by Lemma 5.24. We proceed by induction on $|K^\tau_{A\Rightarrow B}|+|r|$ to show that all the one step reducts of $K^\tau_{A\Rightarrow B}[\lambda x^A.r]$ are in SN. Since r is not a product, the only one step reducts are the following.

- If the reduction takes place in one of the terms in $\mathcal{T}(K^\tau_{A\Rightarrow B})$ or r, we apply the induction hypothesis.
- If $K^\tau_{A\Rightarrow B}[\lambda x^A.r] = K'^\tau_B[(\lambda x^A.r)s]$ and it reduces to $K'^\tau_B[r[s/x]]$, as $r[s/x] \in [\![B]\!]$, we have $K'^\tau_B[r[s/x]] \in \mathsf{SN}$. ◀

▶ **Theorem 5.30** (Adequacy)**.** *If $r : A$, then for all σ adequate, we have $\sigma r \in [\![A]\!]$.*

Proof. By induction on r.

- If r is a variable $x \in \mathcal{V}_A$, then, since σ is adequate, we have $\sigma r \in [\![A]\!]$.
- If r is a product $s \times t$, then by Lemma 4.2, $s : B$, $t : C$, and $A \equiv B \wedge C$, then by the induction hypothesis, $\sigma s \in [\![B]\!]$ and $\sigma t \in [\![C]\!]$. By Lemma 5.27, $(\sigma s \times \sigma t) \in [\![B \wedge C]\!]$, hence by Lemma 5.23, $\sigma r \in [\![A]\!]$.
- If r is a projection $\pi_A(s)$, then by Lemma 4.2, $s : A \wedge B$, and by the induction hypothesis, $\sigma s \in [\![A \wedge B]\!]$. By Lemma 5.25, $\pi_A(\sigma s) \in [\![A]\!]$, hence $\sigma r \in [\![A]\!]$.
- If r is an abstraction $\lambda x^B.s$, with $s : C$, then by Lemma 4.2, $A \equiv B \Rightarrow C$, hence by the induction hypothesis, for all σ, and for all $t \in [\![B]\!]$, $(\sigma s)[t/x] \in [\![C]\!]$. Hence, by Lemma 5.28, $\lambda x^B.\sigma s \in [\![B \Rightarrow C]\!]$, hence, by Lemma 5.23, $\sigma r \in [\![A]\!]$.
- If r is an application st, then by Lemma 4.2, $s : B \Rightarrow A$ and $t : B$, then by the induction hypothesis, $\sigma s \in [\![B \Rightarrow A]\!]$ and $\sigma t \in [\![B]\!]$. Hence, by Lemma 5.26, we have $\sigma r = \sigma s \sigma t \in [\![A]\!]$. ◀