Theory and Applications of Categories, Vol. 8, No. 5, pp. 63–99.

FINITE SUM – PRODUCT LOGIC

J.R.B. COCKETT[1] AND R.A.G. SEELY[2]

ABSTRACT. In this paper we describe a deductive system for categories with finite products and coproducts, prove decidability of equality of morphisms *via* cut elimination, and prove a "Whitman theorem" for the free such categories over arbitrary base categories. This result provides a nice illustration of some basic techniques in categorical proof theory, and also seems to have slipped past unproved in previous work in this field. Furthermore, it suggests a type-theoretic approach to 2–player input–output games.

Introduction

In the late 1960's Lambek introduced the notion of a "deductive system", by which he meant the presentation of a sequent calculus for a logic as a category, whose objects were formulas of the logic, and whose arrows were (equivalence classes of) sequent derivations. He noticed that "doctrines" of categories corresponded under this construction to certain logics. The classic example of this was cartesian closed categories, which could then be regarded as the "proof theory" for the $\wedge - \Rightarrow$ fragment of intuitionistic propositional logic. (An excellent account of this may be found in the classic monograph [Lambek–Scott 1986].) Since his original work, many categorical doctrines have been given similar analyses, but it seems one simple case has been overlooked, *viz.* the doctrine of categories with finite products and coproducts (without any closed structure and without any extra assumptions concerning distributivity of the one over the other). We began looking at this case with the thought that it would provide a nice simple introduction to some techniques in categorical proof theory, particularly the idea of rewriting systems modulo equations, which we have found useful in investigating categorical structures with two tensor products ("linearly distributive categories" [Blute *et al.* 1996]). In addition, it also serves as a simple introduction to categorical cut elimination, in a style which has recently been studied by Joyal [Joyal 1995], and which he attributes to Whitman [Whitman 1941]. As a pedagogical tool, then, we think this case merits a closer look. There may be no surprises in the result, though there are some subtleties that make the proof of some interest.

In addition to an instructive case study in categorical proof theory there may be further

[1]Research partially supported by NSERC, Canada. [2]Research partially supported by Le Fonds FCAR, Québec, and NSERC, Canada. Diagrams in this paper were produced with the help of the TEXcad drawing program of G. Horn and the XY-pic diagram macros of K. Rose and R. Moore.

Received by the editors 2000 March 30 and, in revised form, 2001 February 20.

Transmitted by Michael Barr. Published on 2001 March 5.

2000 Mathematics Subject Classification: 03B70, 03F05, 03F07, 03G30 .

Key words and phrases: categories, categorical proof theory, finite coproducts, finite products, deductive systems .

reasons to be interested in this logic. There are suggestive connections with game theory, (see particularly the recent work of Luigi Santocanale [Santocanale 1999] on meet and join posets with fixed points). The types may be viewed as finite games which lack the usual requirement that play should alternate strictly between player (product structure) and opponent (coproduct structure). Games, in turn, are related to the protocols one expects to be obeyed by a communication channel. Thus, it is possible that this logic could be a useful foundation in understanding channel-based concurrent communication. In a sequel we intend to show that a type-theoretic foundation for games (*i.e.* for such communication) can be presented based on a variation of the type theory presented in this paper.

In the following we shall present a sequent calculus suitable for categories with finite products and coproducts (or "sums"). We adopt a slightly non-traditional presentation, in terms of products and coproducts indexed by arbitrary finite sets. This is easily seen to be equivalent to binary sums and products together with the nullary cases, the constants 0 and 1. We think this approach simplifies the details. Our axioms $A \vdash A$ are restricted to the cases where A is atomic; in a later section we show how $X \vdash X$ may be derived for arbitrary X. We show cut elimination for this system, giving eight cut elimination schema which suffice for the proof. Then for the categorical semantics, we show four equivalence schema which must be valid for the categorical properties to hold. We construct a term calculus for sequent derivations, and then show it is Church–Rosser, modulo these four equivalences. In order to be able to decide equality of maps, we must be able to determine if two (cut-free) terms of the same type (*i.e.* cut-free derivations of the same sequent) are related by the four equivalence schema.

We have various presentations of this decision procedure. The first, presented in the main body of the paper, is symmetric with respect to the duality between sum and product and involves a simple algorithm for deciding the equivalences. We present this procedure graphically using trees which, we hope, suggest the link to games, as these trees do indeed specify strategies. The second technique (in Appendix A) was suggested to us by Luigi Santocanale and orients the equivalences in such a way that, together with the original elimination schema, we have a confluent rewriting system. This rewriting system is a little curious, however, as it does not respect the sum–product duality (so there are actually two possible orientations of the equations). We have not made this approach our primary attack on the decision question, since although it provides an immediate way to establish decidability, we felt that it does not flow as naturally from the proof theory. Finally, we can view the terms as planar geometric diagrams ("cellular squares") (Appendix B) in a manner suggested to us by Peter Selinger, which gives an immediate decision procedure. This last method is the most intuitive but perhaps the least rigorous approach. Furthermore, in so far as we have tried to develop this idea, it appears that cellular squares fail to handle the units (*viz.* the initial and final objects); we believe, in fact, that cellular squares may be related to Girard's suggestion for handling the additives in linear logic [Girard 1995] which also fails to handle the (additive) units.

One point of interest is that although we do need to put an equivalence relation on derivations, induced by the cut-elimination steps, we do not need to suppose the general identity equations, nor the associativity of composition — these follow from the cut elimination process. We conclude the paper with the "Whitman" theorem for this logic, which gives a characterization of the free category with finite products and coproducts over a base category in terms of characterizations of the hom sets.

So, we present a correspondence between the doctrine of categories with finite sums and products, and a "Lambek style" deductive system $\Sigma\Pi$. This correspondence will no doubt seem quite like similar results proven over the past several decades (*e.g.* consider the rules for the additives in linear logic [Girard 1987]): we think some comments concerning its novelty will help the reader attune himself to some points that otherwise might slip past unnoticed.

First, note that unlike the work of Joyal's [Joyal 1995] on the bicompletion of categories, our construction of the free category does not use an inductive chain of constructions, alternating between completions and cocompletions. In this finite case, one can do both finite completions (but only for sums and products) in one step. Our proof of cut elimination and the Church–Rosser property does not make sense without finiteness however. Furthermore, unlike the related work on categories with finite products (or sums, but not both), with or without cartesian closedness, if we wish to maintain sum–product symmetry we cannot rely only on directed rewrites ("reductions") but must also make essential use of two-way rewrites (which we might call "equations"). Hence we need to use the theory of reduction systems modulo equations; in particular, we need to strengthen an old result of Huet's in this connection. Our use of such systems to provide a decision procedure for the equality of derivations is not particularly common in categorical cut elimination, but it was a key ingredient in our earlier work on linearly distributive categories and $*$-autonomous categories [Blute *et al.* 1996]. Its reappearance here in a slightly different guise is one of the highlights of the present paper, we think. Finally, we have already drawn the reader's attention to the fact that in setting up the equivalence relation for the category induced by the deductive system, we need not assume the categorical axioms of identity (apart from atomic instances) and associativity — these follow from the cut elimination process. This is not unusual in working with free logics, but may be somewhat less familiar to the reader in the present case of logic over an arbitrary category. Small points individually, but together they give this result a slightly different flavour from its long-familiar relations.

1. The sequent calculus

Define a sequent calculus for products and coproducts as follows. The propositions are either atoms (which we shall write as A, B, ..) or compound formulas (which we shall write as X, Y, ..). A compound formula is either an I-ary **sum**, where I is a finite set, written $\sum_{i\in I} X_i$, or a **product**, written $\prod_{i\in I} X_i$. The special cases of these sums and products for when the index set is empty, $I=\emptyset$, shall be written respectively as $\sum_\emptyset = 0$ and $\prod_\emptyset = 1$. For binary sums and products we shall write $X+Y$ and $X\times Y$ respectively.

A sequent will be an ordered pair of formulas, denoted in the usual manner with a "proof turnstile": $X \vdash Y$. The rules of inference for this logic are as follows:

$$\frac{}{A \vdash A}\ \textit{identity on atoms}$$

$$\frac{\{X_i \vdash Y\}_{i\in I}}{\sum_{i\in I} X_i \vdash Y}\ \textit{cotuple} \qquad \frac{\{X \vdash Y_i\}_{i\in I}}{X \vdash \prod_{i\in I} Y_i}\ \textit{tuple}$$

$$\frac{X \vdash Y_k}{X \vdash \sum_{i\in I} Y_i}\ \textit{injection} \qquad \frac{X_k \vdash Y}{\prod_{i\in I} X_i \vdash Y}\ \textit{projection}$$

$$\text{where } k \in I,\ I \neq \emptyset$$

$$\frac{X \vdash Y \quad Y \vdash Z}{X \vdash Z}\ \textit{cut}$$

Note that in the cotuple and tuple rules, I may be empty, though not in the injection and projection rules.

Here are some typical proofs in this logic:

1.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}}{A\times B \vdash A} \quad \dfrac{\dfrac{\dfrac{}{B \vdash B}}{B \vdash B+C}}{A\times B \vdash B+C}}{A\times B \vdash A\times(B+C)} \quad \dfrac{\dfrac{\dfrac{}{A \vdash A}}{A\times C \vdash A} \quad \dfrac{\dfrac{\dfrac{}{C \vdash C}}{A\times C \vdash C}}{A\times C \vdash B+C}}{A\times C \vdash A\times(B+C)}}{(A\times B)+(A\times C) \vdash A\times(B+C)}$$

Note that this proves one direction of the distributive law: it will be clear shortly that the other direction cannot be proven in the system.

2.

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}}{A\times(B+C) \vdash A} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash B}}{B \vdash B+C} \quad \dfrac{\dfrac{}{C \vdash C}}{C \vdash B+C}}{B+C \vdash B+C}}{A\times(B+C) \vdash B+C}}{A\times(B+C) \vdash A\times(B+C)}$$

This proves the identity inference! Note that the proof has a non-trivial structure which involves decomposing repeatedly the top level structure until the identities at the atomic level are reached.

Notice that this logic does not have the usual structural rules of thinning, exchange, and weakening. Clearly the projection and injection rules provide the effect of weakening (on either side of the "turnstile"). The effect of thinning is provided by the diagonal sequent which can be constructed by:

$$\frac{A \vdash A \quad A \vdash A}{A \vdash A\times A}$$

The codiagonal sequent is constructed, of course, by the dual proof. Finally, since we have *sets* of premises (in the tupling and cotupling rules) instead of *sequences* of premises, we can deduce exchange in the form of sequents expressing the commutativity of sum and product.

We shall denote this logic by $\Sigma\Pi$, and we shall consider various augmentations of this basic logic:

- The "initial logic" is the logic with no atoms: notice that this is still a non-trivial logic because of the symbols $\sum_\emptyset$ and $\prod_\emptyset$ from which more complex formulae can be constructed. We shall write this as $\Sigma\Pi_\emptyset$.

- The "pure logic" is the logic with an arbitrary set of atoms, A: we shall write this as $\Sigma\Pi_A$.

- The "free logic" is the logic with an arbitrary set of atoms and an arbitrary set of axioms relating those atoms. Although we could regard this as a graph, we shall also allow a notion of equality of paths, and so the atoms are actually the objects of a category and the axioms are maps in that category (with the "essential cuts" being provided by composition in that category). If we denote this category by $\mathbf{A}$, we shall denote the resulting logic by $\Sigma\Pi_{\mathbf{A}}$.

We may think of the atoms of a pure logic as forming a discrete category, the free logic on this discrete category is then just the "pure" logic. Clearly, therefore, each variant above includes the previous variants. We shall thus only deal with the last variant, unless otherwise noted, since it is the most general.

It is worth noting that the inference system for $\Sigma\Pi$ is self-dual, that is, it has an obvious sum–product symmetry. Explicitly, we may swap the direction of the sequents while, at the same time, turning sums into products and products into sums to obtain the same system. This means that each proof has a dual interpretation and can be "reused" to prove a dual theorem. The systems we introduce in the main body of this paper are designed to maintain this symmetry.

There is a simple cut elimination theorem for the logic $\Sigma\Pi_{\mathbf{A}}$ which eliminates all cuts by rewriting the proof trees. Of course, the process will get stuck on the introduced atomic axioms or sequents. A cut between atomic axioms is an essential cut.

1.1. Proposition. [Cut elimination:] *Any proof in the free logic $\Sigma\Pi_{\mathbf{A}}$ can be transformed to a proof in which the only cuts are essential.*

Proof. We shall provide a family of rewrites for proofs and show that they terminate. Any canonical proof (that is a proof which cannot be further rewritten) for this set of rewrites will be a "cut eliminated" proof in the sense of having no inessential cuts.

We shall use the duality to reduce the number of rewrites we present.

Sequent–Identity (Identity–Sequent): This rewrite removes the cut below an identity axiom on the right:

$$\frac{\dfrac{\pi}{X \vdash A} \quad \overline{A \vdash A}}{X \vdash A} \quad \Longrightarrow \quad \frac{\pi}{X \vdash A}$$

Its dual rule removes the cut below an identity axiom on the left.

Sequent–Injection (Projection–Sequent): This rewrite moves a cut which is below an arbitrary sequent and an injection above the injection.

$$\frac{\dfrac{\pi}{Y \vdash Z} \quad \dfrac{\dfrac{\pi'}{Z \vdash X_k}}{Z \vdash \sum X_i}}{Y \vdash \sum X_i} \quad \Longrightarrow \quad \frac{\dfrac{\dfrac{\pi}{Y \vdash Z} \quad \dfrac{\pi'}{Z \vdash X_k}}{Y \vdash X_k}}{Y \vdash \sum X_i}$$

The dual rewrite moves a cut above a projection.

Cotupling–Sequent (Sequent–Tupling): This rewrite moves a cut which is below a cotupling and an arbitrary sequent above the cotupling.

$$\frac{\dfrac{\left\{\dfrac{\pi_i}{Y_i \vdash X}\right\}_{i\in I}}{\sum Y_i \vdash X} \quad \dfrac{\pi}{X \vdash Z}}{\sum Y_i \vdash Z} \quad \Longrightarrow \quad \frac{\left\{\dfrac{\dfrac{\pi_i}{Y_i \vdash X} \quad \dfrac{\pi}{X \vdash Z}}{Y_i \vdash Z}\right\}_{i\in I}}{\sum Y_i \vdash Z}$$

The dual moves the cut above tupling on the right.

Tupling–Projection (Injection–Cotupling): This rewrite moves the cut above tupling and projection:

$$\frac{\dfrac{\left\{\dfrac{\pi_i}{X \vdash Y_i}\right\}_{i\in I}}{X \vdash \prod Y_i} \quad \dfrac{\dfrac{\pi}{Y_k \vdash Z}}{\prod Y_i \vdash Z}}{X \vdash Z} \quad \Longrightarrow \quad \frac{\dfrac{\pi_k}{X \vdash Y_k} \quad \dfrac{\pi}{Y_k \vdash Z}}{X \vdash Z}$$

The dual of this rewrite moves the cut above injection and cotupling.

We have now accounted for all the ways in which compound formulas are introduced either on the left or right above a cut and have shown how to move the cut above these rules. Thus, a proof which cannot be rewritten further must have an axiom above the cut on each side. This is an essential cut.

It remains only to show that this process terminates. To show this we use a bag of "cut heights". In essence, each cut-elimination step removes a cut and replaces it by a finite bag of cuts at a lesser height. This shows that this rewriting terminates. (The technical details will be presented in Section 2 with the proof of the Church–Rosser property for the cut elimination process.) ∎

1.2. Identity derivations. The cut elimination process above introduces a number of proof identifications. All these identifications involve the cut rule as our purpose was to remove it: we shall shortly see that the system has some further proof identifications which do not involve the cut rule.

Our purpose is to view this proof system as a category where cut is the composition. The cut elimination process, above, therefore provides part of the dynamics of composition: the activity which takes place when two proofs are plugged together.

In order to prove that cut acts as a composition we must start by showing that there are identity derivations which behave in the correct manner. We define the identity derivations inductively by:

Atoms: The identity atomic sequent:

$$A \vdash A.$$

Sums: The identity derivation on sums is given by:

$$\frac{\left\{\dfrac{\dfrac{\iota_{X_i}}{X_i \vdash X_i}}{X_i \vdash \sum X_i}\right\}_i}{\sum X_i \vdash \sum X_i}$$

where the identity derivation ι_{X_i} of $X_i \vdash X_i$ is given by induction on the structure of X_i.

Products: The identity on products is given by the dual of the proof above.

1.3. Lemma. *The Sequent–Identity and Identity–Sequent cut-elimination reductions are derivable reductions for the general identity derivations as defined above.*

$$\frac{\dfrac{\pi}{X \vdash Y} \quad \dfrac{\iota_Y}{Y \vdash Y}}{X \vdash Y} \quad \Longrightarrow \quad \frac{\pi}{X \vdash Y}$$

and similarly for the dual rule.

Proof. We shall suppose the identity derivation is on the right; duality covers the other case. We argue by structural induction on the proof π. We assume, therefore that the result is true for any subformula of π.

1. The base case is a cut with an atomic sequent: here cut elimination removes the atomic identity so the result is immediate.

2. Next we suppose the identity is on a sum type:

$$\frac{\dfrac{\pi}{X \vdash \sum Y_i} \quad \dfrac{\iota}{\sum Y_i \vdash \sum Y_i}}{X \vdash \sum Y_i}$$

   There are three possibilities for the root inference of π.

(a) If the root inference is a cotupling, the cut elimination step moves the cut onto smaller proofs and so the inductive assumption does the job.

$$\dfrac{\dfrac{\left\{\dfrac{\pi_j}{X_j \vdash \sum Y_i}\right\}_{j\in J}}{\sum X_j \vdash \sum Y_i} \qquad \dfrac{\iota}{\sum Y_i \vdash \sum Y_i}}{\sum X_j \vdash \sum Y_i} \quad\Longrightarrow\quad \dfrac{\left\{\dfrac{\dfrac{\pi_j}{X_j \vdash \sum Y_i} \qquad \dfrac{\iota}{\sum Y_i \vdash \sum Y_i}}{X_j \vdash \sum Y_i}\right\}_{j\in J}}{\sum X_j \vdash \sum Y_i}$$

(b) The root inference is an injection.

$$\dfrac{\dfrac{\dfrac{\pi}{X \vdash Y_i}}{X \vdash \sum Y_i} \qquad \dfrac{\left\{\dfrac{\dfrac{\iota}{Y_i \vdash Y_i}}{Y_i \vdash \sum Y_i}\right\}_{i\in I}}{\sum Y_i \vdash \sum Y_i}}{X \vdash \sum Y_i} \quad\Longrightarrow\quad \dfrac{\dfrac{\dfrac{\pi}{X \vdash Y_i} \qquad \dfrac{\iota}{Y_i \vdash Y_i}}{X \vdash Y_i}}{X \vdash \sum Y_i}$$

(c) The root inference is a projection: in this case the cut step immediately moves the identity onto a smaller proof so we are done.

3. Next we suppose the identity is on a product type:

$$\dfrac{\dfrac{\pi}{Y \vdash \prod X_i} \qquad \dfrac{\iota}{\prod X_i \vdash \prod X_i}}{Y \vdash \prod X_i}$$

There are two possibilities for the root inference of π.

(a) The root inference is a tupling:

$$\dfrac{\dfrac{\left\{\dfrac{\pi_j}{Y \vdash X_j}\right\}_{j\in I}}{Y \vdash \prod X_i} \qquad \dfrac{\left\{\dfrac{\dfrac{\iota_i}{X_i \vdash X_i}}{\prod X_i \vdash X_i}\right\}_{i\in I}}{\prod X_i \vdash \prod X_i}}{Y \vdash \prod X_i} \quad\Longrightarrow\quad \dfrac{\left\{\dfrac{\dfrac{\left\{\dfrac{\pi_j}{Y \vdash X_j}\right\}_{j\in I}}{Y \vdash \prod X_i} \qquad \dfrac{\dfrac{\iota_i}{X_i \vdash X_i}}{\prod X_i \vdash X_i}}{Y \vdash X_i}\right\}_{i\in I}}{Y \vdash \prod X_i}$$

$$\Longrightarrow\quad \dfrac{\left\{\dfrac{\dfrac{\pi_i}{Y \vdash X_i} \qquad \dfrac{\iota_i}{X_i \vdash X_i}}{Y \vdash X_i}\right\}_{i\in I}}{Y \vdash \prod X_i}$$

(b) The root inference is a projection: here as before we can immediately move the identity onto a smaller proof tree.

∎

1.4. Permuting conversions. In order to obtain a normal form for sequent derivations, we shall want to prove the cut-elimination rewrites are Church–Rosser. However, a brief consideration of these rewrites will show some obviously problematic critical pairs; for example, given a derivation with a cotupling and a tupling immediately above a cut, there are two ways we can reduce the derivation: (Cotupling–Sequent) or (Sequent–Tupling), with no apparent way to resolve these rewrites. So in order to even hope for a Church–Rosser rewrite system, we shall need additional rewrites which allow us to interchange these two rules. Similar considerations for other critical pairs (*viz.* projection *vs.* tupling, cotupling *vs.* injection, and projection *vs.* injection) lead us to the following four conversions (which we shall denote by $\Longmapsto$).

- Projection/tuple interchange:

$$\dfrac{\left\{\dfrac{\dfrac{\pi_i}{X_k \vdash Y_i}}{\prod X_j \vdash Y_i}\right\}_i}{\prod X_j \vdash \prod Y_i} \quad \Longmapsto \quad \dfrac{\dfrac{\left\{\dfrac{\pi_i}{X_k \vdash Y_i}\right\}_i}{X_k \vdash \prod Y_i}}{\prod X_j \vdash \prod Y_i}$$

- Cotuple/injection interchange: this is dual to the previous proof equality.
- Projection/injection interchange:

$$\dfrac{\dfrac{\dfrac{\pi}{X_l \vdash Y_k}}{\prod X_j \vdash Y_k}}{\prod X_j \vdash \sum Y_i} \quad \Longmapsto \quad \dfrac{\dfrac{\dfrac{\pi}{X_l \vdash Y_k}}{X_l \vdash \sum Y_i}}{\prod X_j \vdash \sum Y_i}$$

- Tuple/cotuple interchange:

$$\dfrac{\left\{\dfrac{\left\{\dfrac{\pi_{ij}}{X_i \vdash Y_j}\right\}_i}{\sum X_i \vdash Y_j}\right\}_j}{\sum X_i \vdash \prod Y_j} \quad \Longmapsto \quad \dfrac{\left\{\dfrac{\left\{\dfrac{\pi_{ij}}{X_i \vdash Y_j}\right\}_j}{X_i \vdash \prod Y_j}\right\}_i}{\sum X_i \vdash \prod Y_j}$$

1.5. A term calculus. The proof theory of $\Sigma\Pi_{\mathbf{A}}$ is intended to be the free category with products and coproducts generated by $\mathbf{A}$. (Note that we are not assuming any sort of distributivity.) It will turn out if we define an equivalence relation on sequent derivations induced by the cut-elimination steps and the conversions above, that this is indeed the case. To this end, it will be convenient to have a more compact notation for sequent derivations: this leads us to impose a system of terms, typed by sequents, which in effect will also give us the categorical semantics for the logic. The term formation rules are given in Table 1.

We shall use the notation $\langle\rangle$ for the map from the empty sum and $()$ for the map to the empty product.

$$\frac{}{A \vdash_{1_A} A}\ identity$$

$$\frac{\{X_i \vdash_{f_i} Y\}_{i\in I}}{\sum_{i\in I} X_i \vdash_{\langle f_i\rangle_{i\in I}} Y}\ cotuple \qquad \frac{\{X \vdash_{f_i} Y_i\}_{i\in I}}{X \vdash_{(f_i)_{i\in I}} \prod_{i\in I} Y_i}\ tuple$$

$$\frac{X \vdash_f Y_k}{X \vdash_{b_k(f)} \sum_{i\in I} Y_i}\ injection \qquad \frac{X_k \vdash_f Y}{\prod_{i\in I} X_i \vdash_{p_k(f)} Y}\ projection$$

where $k \in I$, $I \neq \emptyset$

$$\frac{X \vdash_f Y \quad Y \vdash_g Z}{X \vdash_{f;g} Z}\ cut$$

Table 1: $\Sigma\Pi$ term formation rules

There is something still a bit odd from the categorical viewpoint about these terms — namely, the projections and injections seem to be a little unfamiliar in their presentation. This may be "remedied" as follows. Using the notation above, there are projection derivations $\prod_{i\in I} X_i \vdash_{p_k} X_k$ for $k \in I$ given by $p_k = p_k(\iota_{X_k})$. With these more "standard" projections, the general projection terms may be identified with $p_i\ ;\ f$. Note this is a valid identification, since there is a reduction of derivations

$$\frac{\dfrac{X_k \vdash X_k}{\prod X_i \vdash X_k} \quad X_k \vdash Y}{\prod X_i \vdash Y} \quad \Longrightarrow \quad \frac{X_k \vdash Y}{\prod X_i \vdash Y}$$

Using the terms defined in Table 1, we can summarize the cut-elimination reductions and the permuting conversions as in Table 2. (We omit typing information, since it may be inferred from the terms, and in any case we have displayed these as sequent derivations in the previous subsections.)

Note these come in dual pairs — apart from (11) and (12) which are self-dual — so we have four reductions, one conversion, and their duals, and two other conversions: essentially just 7 rewrites.

We ought to point out that we do allow the index sets I, J to be empty, except for the reductions (3), (4), (7), and (8) and except for the permuting conversion (11); in these cases, since reference is made to an element k or l, it does not make sense for the corresponding index set I or J to be empty. In (9), (10), the index set J for the named element k must not be empty, but the other index set I may be. In (12) either (or both or neither) index set may be empty. An explicit treatment of these nullary cases may be found in Appendix C.

It is an easy exercise to verify that these cut-elimination reductions and permuting

$$
\begin{array}{rclr}
f\,;1 & \Longrightarrow & f & (1)\\
1\,;f & \Longrightarrow & f & (2)\\
f\,;b_k(g) & \Longrightarrow & b_k(f\,;g) & (3)\\
p_k(f)\,;g & \Longrightarrow & p_k(f\,;g) & (4)\\
\langle f_i\rangle_{i\in I}\,;g & \Longrightarrow & \langle f_i\,;g\rangle_{i\in I} & (5)\\
f\,;(g_i)_{i\in I} & \Longrightarrow & (f\,;g_i)_{i\in I} & (6)\\
b_k(f)\,;\langle g_i\rangle_{i\in I} & \Longrightarrow & f\,;g_k & (7)\\
(f_i)_{i\in I}\,;p_k(g) & \Longrightarrow & f_k\,;g & (8)
\end{array}
\qquad
\begin{array}{rclr}
b_k(\langle f_i\rangle_{i\in I}) & \vDash\!\!\!\dashv & \langle b_k(f_i)\rangle_{i\in I} & (9)\\
(p_k(f_i))_{i\in I} & \vDash\!\!\!\dashv & p_k((f_i)_{i\in I}) & (10)\\
b_l(p_k(f)) & \vDash\!\!\!\dashv & p_k(b_l(f)) & (11)\\
(\langle f_{ij}\rangle_{i\in I})_{j\in J} & \vDash\!\!\!\dashv & \langle (f_{ij})_{j\in J}\rangle_{i\in I} & (12)
\end{array}
$$

Table 2: $\Sigma\Pi$ conversion rules

conversions are valid in any category with finite products and coproducts. We shall leave that for the reader, however, and instead concentrate on proving Church–Rosser for the system of cut-elimination rewrites, modulo the conversions.

2. Proof of the Church–Rosser property

We wish to show that, given any two $\Sigma\Pi$-morphisms related by a series of reductions and permuting conversions

$$t_1 \Longleftarrow t_2 \vDash\!\!\!\dashv t_3 \Longrightarrow \cdots\cdots \Longleftarrow t_{n-2} \vDash\!\!\!\dashv t_{n-1} \Longrightarrow t_n$$

there is an alternate way of arranging the reductions and equalities so that t_1 and t_n can be reduced to terms which are related by permuting conversions. This means there is a convergence of the following form:

$$
\begin{tikzcd}
t_1 \arrow[dr, Rightarrow, "*"'] & & t_n \arrow[dl, Rightarrow, "*"] \\
& t'_1 \vDash\!\!\!\dashv_{*}\; t'_n &
\end{tikzcd}
$$

When the rewriting system terminates (in the appropriate sense) this allows the decision procedure for the equality of terms to be reduced to the decision procedure for the permuting conversions. In order to test the equality of two terms one can rewrite both terms into a reduced normal form (one from which there are no further reductions). These will be equal if and only if the two reduced forms are equivalent through the permuting conversions alone.

In the current situation the reduction process is the cut elimination procedure. We have already argued informally that it is a terminating process: in this section we will formalize this. In the next section we shall discuss a decision procedure for the permuting conversions.

2.1. Resolving critical pairs locally. The core of the proof of Church–Rosser in this situation involves looking at all the possible critical pairs of reductions and conversions, although there are some significant subtleties in showing that this does actually achieve the desired result. In this subsection we shall show how to resolve all critical pairs of reductions, and likewise all critical pairs consisting of a reduction and a conversion, in such a way that the first rewrite following the conversion (in the latter cases) is a reduction (not a permuting conversion). We shall see in the subsequent subsections that this is sufficient to deduce the existence of normal forms modulo the conversions; once we establish the decision procedure for conversions, this will give us a decision procedure for the equivalence relation induced by the reductions and permuting conversions.

(1) - (2) obvious.

(1) - (4) $p_k(f \; ; \; 1) \Longleftarrow p_k(f) \; ; \; 1 \Longrightarrow p_k(f)$ is resolved by $p_k(f \; ; \; 1) \Longrightarrow p_k(f)$ (apply reduction (1) inside $p_k(\;)$).

(1) - (5) is handled similarly.

With the evident dualities, this handles all critical pairs with (1) and (2).

(3) - (4) We indicate the resolution of the critical pair by the following reduction diagram.

$$\begin{tikzcd}
 & p_k(f) \; ; \; b_l(g) \arrow[dl, Rightarrow, "(3)"'] \arrow[dr, Rightarrow, "(4)"] & \\
b_l(p_k(f) \; ; \; g) \arrow[d, Rightarrow, "b_l((4))"'] & & p_k(f \; ; \; b_l(g)) \arrow[d, Rightarrow, "p_k((3))"] \\
b_l(p_k(f \; ; \; g)) \arrow[rr, Leftrightarrow, "(11)"'] & & p_k(b_l(f \; ; \; g))
\end{tikzcd}$$

(3) - (5) is similar, using the conversion (9).

(3) - (9) $b_k(f \; ; \langle g_i \rangle_i) \Longleftarrow f \; ; \; b_k(\langle g_i \rangle_i) \Longleftrightarrow f \; ; \; \langle b_k(g_i) \rangle_i$

For this critical pair (a reduction *vs.* a permuting conversion) we have three cases to consider.

Case (i) $f = \langle f_j \rangle_j$ In this case we can resolve the critical pair as follows.

$$
\begin{tikzcd}[column sep=small]
& \langle f_j \rangle_j \,;\, b_k(\langle g_i \rangle_i) \arrow[dl, Rightarrow, "(3)"'] \arrow[dr, Leftrightarrow, "1;(9)"] & \\
b_k(\langle f_j \rangle_j \,;\, \langle g_i \rangle_i) \arrow[d, Rightarrow, "b_k((5))"'] & & \langle f_j \rangle_j \,;\, \langle b_k(g_i) \rangle_i \arrow[d, Rightarrow, "(5)"] \\
b_k(\langle f_j \,;\, \langle g_i \rangle_i \rangle_j) \arrow[d, Leftrightarrow, "(9)"'] & & \langle f_j \,;\, \langle b_k(g_i) \rangle_i \rangle_j \arrow[d, Leftrightarrow, "\langle 1;(9) \rangle_j"] \\
\langle b_k(f_j \,;\, \langle g_i \rangle_i) \rangle_j & & \langle f_j \,;\, b_k(\langle g_i \rangle_i) \rangle_j \arrow[ll, Rightarrow, "\langle (3) \rangle_j"']
\end{tikzcd}
$$

Case (ii) $f = b_j(f')$ In this case, the critical pair resolves itself easily using the reductions (7) and (3), to end up with $b_k(f' \,;\, f_j)$ as common reduct.

Case (iii) $f = p_j(f')$ Using reduction (4) we immediately reduce the right hand side to $p_j(f' \,;\, \langle b_k(g_i) \rangle_i)$, and then by (9) to $p_j(f' \,;\, b_k(\langle g_i \rangle_i))$ and by (3) to $p_j(b_k(f' \,;\, \langle g_i \rangle_i))$ which converts to $b_k(p_j(f' \,;\, \langle g_i \rangle_i))$ by (11).

(3) - (11) $b_l(f \,;\, p_k(g)) \Longleftarrow f \,;\, b_l(p_k(g)) \models\!\!\dashv f \,;\, p_k(b_l(g))$ This is resolved in a manner similar to the previous case. Again there are three cases: $f = (f_j)_j$, $f = \langle f_j \rangle_j$, and $f = p_j(f')$, which are resolved as above essentially using (11) in place of (9).

With the evident dualities, this handles all critical pairs with (3) and (4).

(5) - (6)

$$
\begin{tikzcd}[column sep=small]
& \langle f_i \rangle_i \,;\, (g_j)_j \arrow[dl, Rightarrow, "(6)"'] \arrow[dr, Rightarrow, "(5)"] & \\
(\langle f_i \rangle_i \,;\, g_j)_j \arrow[d, Rightarrow, "((5))_j"'] & & \langle f_i \,;\, (g_j)_j \rangle_i \arrow[d, Rightarrow, "\langle (6) \rangle_i"] \\
(\langle f_i \,;\, g_j \rangle_i)_j \arrow[rr, Leftrightarrow, "(12)"'] & & \langle (f_i \,;\, g_j)_j \rangle_i
\end{tikzcd}
$$

(5) - (9) $b_k(\langle f_i \rangle_i) \,;\, g \models\!\!\dashv \langle b_k(f_i) \rangle_i \,;\, g \Longrightarrow \langle b_k(f_i) \,;\, g \rangle_i$ This pair may be divided into three cases.

Case (i): $g = \langle h_j \rangle_j$ This is resolved by a simple use of (7) and (5).

Case (ii): $g = (h_j)_j$ We have $\langle b_k(f_i) \,;\, (h_j)_j \rangle_i \Longrightarrow \langle (b_k(f_i) \,;\, h_j)_j \rangle_I \models\!\!\dashv (\langle b_k(f_i) \,;\, h_j \rangle_i)_j$ and $b_k(\langle f_i \rangle_i) \,;\, (h_j)_j \Longrightarrow (b_k(\langle f_i \rangle_i) \,;\, h_j)_j \Longrightarrow (\langle b_k(f_i) \rangle_i \,;\, h_j)_j \Longrightarrow (\langle b_k(f_i) \,;\, h_j \rangle_i)_j$.

Case (iii): $g = b_j(h)$ This pair may be resolved directly. $\langle b_k(f_i) \,;\, b_j(h) \rangle_i \Longrightarrow \langle b_j(b_k(f_i) \,;\, h) \rangle_i \models\!\!\dashv b_j(\langle b_k(f_i) \,;\, h \rangle_i) \Longleftarrow b_j(\langle b_k(f_i) \rangle_i \,;\, h) \models\!\!\dashv b_j(b_k(\langle f_i \rangle_i) \,;\, h) \Longleftarrow b_k(\langle f_i \rangle_i) \,;\, b_j(h)$.

(5) - (12) $(\langle f_{ij}\rangle_i)_j \ ; \ g \mathrel{\vDash\!\!\dashv} \langle(f_{ij})_j\rangle_i \ ; \ g \Longrightarrow \langle(f_{ij})_j \ ; \ g\rangle_i$ This pair may be divided into three cases.

Case (i): $g = p_k(h)$ This is resolved by (8) and (5): $\langle(f_{ij})_j \ ; p_k(h)\rangle_i \Longrightarrow \langle f_{ik} \ ; \ h\rangle_i$ and $(\langle f_{ij}\rangle_i)_j \ ; p_k(h) \Longrightarrow \langle f_{kj}\rangle_i \ ; h \Longrightarrow \langle f_{ik} \ ; \ h\rangle_i$

Case (ii): $g = b_k(h)$ This is resolved by (3) and (5):
$(\langle f_{ij}\rangle_i)_j \ ; b_k(h) \Longrightarrow b_k((\langle f_{ij}\rangle_i)_j \ ; \ h) \mathrel{\vDash\!\!\dashv} b_k(\langle(f_{ij})_j\rangle_i \ ; \ h) \Longrightarrow b_k(\langle(f_{ij})_j \ ; \ h\rangle_i)$
$\mathrel{\vDash\!\!\dashv} \langle b_k((f_{ij})_j \ ; \ h)\rangle_i \Longleftarrow \langle(f_{ij})_j \ ; b_k(h)\rangle_i$

Case (iii): $g = (h_k)_k$ This is resolved simply with (6) and (12).

With the evident dualities, this handles all critical pairs with (5) and (6).

(7) - (9)

$$
\begin{tikzcd}
 & b_k(\langle f_i\rangle_i) \ ; \ \langle g_j\rangle_j \arrow[dl, dash, "(9)"'] \arrow[dr, Rightarrow, "(7)"] & \\
\langle b_k(f_i)\rangle_i \ ; \ \langle g_j\rangle_j \arrow[d, Rightarrow, "(5)"'] & & \langle f_i\rangle_i \ ; \ g_k \arrow[d, Rightarrow, "(5)"] \\
\langle b_k(f_i) \ ; \ \langle g_j\rangle_j\rangle_i \arrow[rr, Rightarrow, "\langle(7)\rangle_i"'] & & \langle f_i \ ; \ g_k\rangle_i
\end{tikzcd}
$$

(7) - (11)

$$
\begin{tikzcd}
 & b_i(p_j(g)) \ ; \ \langle f_k\rangle_k \arrow[dl, dash, "(11);1"'] \arrow[dr, Rightarrow, "(7)"] & \\
p_j(b_i(g)) \ ; \ \langle f_k\rangle_k \arrow[d, Rightarrow, "(4)"'] & & p_j(g) \ ; \ f_i \arrow[d, Rightarrow, "(4)"] \\
p_j(b_i(g) \ ; \ \langle f_k\rangle_k) \arrow[rr, Rightarrow, "p_j((7))"'] & & p_j(g \ ; \ f_i)
\end{tikzcd}
$$

(7) - (12)

$$
\begin{tikzcd}
 & b_k(h) \ ; \ \langle(f_{ij})_j\rangle_i \arrow[dl, dash, "1;(12)"'] \arrow[dr, Rightarrow, "(7)"] & \\
b_k(h) \ ; \ (\langle f_{ij}\rangle_i)_j \arrow[d, Rightarrow, "(6)"'] & & h \ ; \ (f_{kj})_j \arrow[d, Rightarrow, "(6)"] \\
(b_k(h) \ ; \ \langle f_{ij}\rangle_i)_j \arrow[rr, Rightarrow, "((7))_j"'] & & (h \ ; \ f_{kj})_j
\end{tikzcd}
$$

With the evident dualities, this handles all critical pairs with (7) and (8).

To see that this is sufficient to obtain the Church–Rosser property, we must look a little more carefully at rewrite systems modulo equations.

2.2. Confluence modulo equations. We consider the general theory of rewrite systems modulo equations; note that the case we have in mind will have the cut elimination reductions (1)–(8) as "reductions", and the permuting conversions (9)–(12) as "equations".

In Huet [Huet 1980] it is proven that a rewrite system modulo equations, which is noetherian (modulo those equations), is confluent modulo equations if and only if it is "locally confluent modulo the equations." By locally confluent Huet meant that each (one step) divergence of the form

$$\begin{tikzcd}[column sep=small] & t_0 \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\ t_1 & & t_2 \end{tikzcd} \qquad \text{or} \qquad \begin{tikzcd}[column sep=small] & t_0 \arrow[dl, Rightarrow, no head, "|"-{pos=0}, "|"{pos=1}] \arrow[dr, Rightarrow] & \\ t_1 & & t_2 \end{tikzcd}$$

should have a convergence of the form

$$\begin{tikzcd} t_1 \arrow[dr, Rightarrow, "*"'] & & t_2 \arrow[dl, Rightarrow, "*"] \\ & t'_1 \arrow[r, Rightarrow, no head, "*"'] & t'_2 \end{tikzcd}$$

To suit the resolutions of the last section, we shall use a more general form of this result which allows a significantly more permissive form for the resolutions of the divergences. Henceforth, in this paper we shall say that a system is **locally confluent modulo equations** in case we have the following convergences (respectively) for the two divergences above:

$$\begin{tikzcd}[column sep=small] t_1 \arrow[dr, Rightarrow, tail, "*"'] & & t_2 \arrow[dl, Rightarrow, tail, "*"] \\ & t' & \end{tikzcd} \qquad \text{and} \qquad \begin{tikzcd} t_1 \arrow[d, Rightarrow] & & t_2 \arrow[ddl, Rightarrow, tail, "*"] \\ t'_1 \arrow[dr, Rightarrow, "*"'] & & \\ & t' & \end{tikzcd}$$

where the new arrow $t_1 \Longmapsto t_2$ indicates either an equality or a reduction in the indicated direction. Note that this is precisely our previously noted condition that the rewrite immediately following the equation (in our case, permuting conversion) is a reduction, after which equations or reductions are permitted.

2.3. Proposition. *Suppose $(N, \mathcal{R}, \mathcal{E})$ is a rewriting system with equations equipped with a measure on the terms $\alpha \colon N \longrightarrow W$ where W is a well-ordered set and*

$$t_1 \Longrightarrow t_2 \quad \textit{implies} \quad \alpha(t_1) > \alpha(t_2)$$

$$t_1 \models\!\!=\!\!\dashv t_2 \quad \textit{implies} \quad \alpha(t_1) = \alpha(t_2)$$

then the system is confluent modulo equations if and only if it is locally confluent modulo equations.

Notice that this really does subsume Huet's result, since if the system is noetherian modulo the equations then the well-ordering W can be provided by the quotient of the terms by the equations with respect to the evident reduction ordering. We have chosen to express the result in this form as we shall shortly provide an explicitly measure α for our system.

PROOF. If the system is confluent modulo equations it is certainly locally confluent modulo equations. Conversely suppose we have a chain of reductions and equations (as above) then we may associate with it the bag of measures of the terms in the sequence:

$$[\alpha(t_1), \alpha(t_2), ...]$$

These bags are well-ordered under the usual bag ordering [Dershowitz–Manna 1976].

The crux of the argument is to show that replacing any local divergence in this chain by a local confluence will result in a new chain whose bag measure is strictly smaller. However, this can be seen by inspection as we are removing the apex of the divergence and replacing it by the bag of values associated to the terms on the interior of the convergence: these all will have strictly smaller associated α-measures.

Thus, each rewriting reduces this bag ordering and thus any sequence of rewriting on such a chain must terminate. However, it can only terminate when there are no local divergences to resolve. This easily implies that the end result must be a confluence modulo the equations. ∎

In our application of this result $\alpha: T \longrightarrow \text{bag}(\mathbf{N}^2)$ is the map which associates to each $\Sigma\Pi$-morphism a bag of cut costs. We shall argue that, as each reduction strictly reduces these cut costs while each equality leaves it stationary, this is a criterion of the desired form. The construction of this cost criterion is our next task.

2.4. The cut measure on $\Sigma\Pi$-morphisms. The measure on the terms in this calculus is forced to be fairly complex because a cut elimination step can duplicate subterms, *e.g.* reductions (5) and (6). The measure of the cuts in subterms which are duplicated must be reduced even though they may not be directly involved in the reduction step. To further complicate matters we also have to take into account the effect of singleton and empty tuples and cotuples: here reductions (5) and (6) may no longer duplicate subterms but our measure still has to give a strict reduction in these situations.

We shall use two different measures to quantify the cut costs: **duplicity** and **height**. The first, roughly speaking, gives an upper bound on the number of times the term can be duplicated in the reduction (or cut elimination) process; the second measures how far the cut is from the leaves. Our aim is to show that the lexicographical combination of these costs is strictly reduced on the principal and on all duplicated cuts, that it is non-increasing on other cuts, and that it is unchanged by equalities.

We shall first describe the duplicity cost of a cut. This is calculated in two phases: the first from the leaves downward (synthesized) and the second from the root upward (inherited). In the first phase we calculate the width at each subterm as follows:

- $\mathrm{width}[a] = 1$ where a is an atomic map (or an identity),
- $\mathrm{width}[\langle f_i\rangle_{i\in I}] = \mathrm{width}[(f_i)_{i\in I}] = \max(\sum_{i\in I}\mathrm{width}[f_i], 1)$,
- $\mathrm{width}[b_k(f)] = \mathrm{width}[p_k(f)] = \mathrm{width}[f]$,
- $\mathrm{width}[f\ ;\ g] = \mathrm{width}[f]\cdot\mathrm{width}[g]$.

We observe the rewritings and equalities never increase the width of a term:

2.5. Lemma.

(i) *If* $t_1 \Longrightarrow t_2$ *then* $\mathrm{width}[t_1] \geq \mathrm{width}[t_2]$,

(ii) *If* $t_1 \models\!\!\!=\!\!\!\dashv t_2$ *then* $\mathrm{width}[t_1] = \mathrm{width}[t_2]$.

Proof.

(i) Clearly rewrites (1), (2), (3), and (4) do not affect the width. For rewrite (5) and the set I non-empty we have:

$$\begin{aligned}
\mathrm{width}[\langle f_i\rangle_{i\in I}\ ;\ g] &= \mathrm{width}[\langle f_i\rangle_{i\in I}]\cdot\mathrm{width}[g]\\
&= (\sum_{i\in I}\mathrm{width}[f_i])\cdot\mathrm{width}[g]\\
&= \sum_{i\in I}(\mathrm{width}[f_i]\cdot\mathrm{width}[g])\\
&= \sum_{i\in I}\mathrm{width}[f_i\ ;\ g]\\
&= \mathrm{width}[\langle f_i\ ;\ g\rangle_{i\in I}]
\end{aligned}$$

When the set I is empty the width will decrease when g has significant width. The situation for (6) is the same.

For (7) and (8) the width will decrease when the set I is not a singleton and will remain the same otherwise.

(ii) (9), (10), and (11) do not affect the width as they involve the projections and injections. Similarly (12) is a double sum when the sets are non-empty; when either is empty the width is 1.

∎

Next we associate with each subterm of t a duplicity as follows:

- $\mathrm{dupl}[t] = 1$ the duplicity of the whole term is 1,
- If the duplicity of a subterm of the form $\langle f_i \rangle_{i\in I}$ or $(f_i)_{i\in I}$ is n then $\mathrm{dupl}[f_i] = n$ for each $i \in I$
- If the duplicity of a subterm of the form $b_k(t')$ or $p_k(t')$ is n then $\mathrm{dupl}[t'] = n$.
- If the duplicity of $f \,;\, g$ is n then $\mathrm{dupl}[f] = n \cdot \mathrm{width}[g]$ and $\mathrm{dupl}[g] = n \cdot \mathrm{width}[f]$.

Notice that the duplicity of all subterms of a cut-eliminated term is 1 as the only way that duplicity can be increased is through the presence of a cut. We define the **duplicity of a cut** to be the product of the widths of its subterms with the duplicity:

$$\mathrm{cutdupl}[f \,;\, g] = \mathrm{dupl}[f \,;\, g] \cdot \mathrm{width}[f] \cdot \mathrm{width}[g].$$

We observe the following:

2.6. Lemma.

(i) *If* $t_1 \Longrightarrow t_2$ *then the bag of the cut duplicities of* t_1 *is greater or equal to that of* t_2*,*

(ii) *If* $t_1 \Longrightarrow t_2$ *is a reduction which duplicates subterms (that is an application of (5) or (6) with the set* I *having more than one element) then the bag of the cut duplicities of* t_1 *is strictly greater than that of* t_2*,*

(iii) *If* $t_1 \models\!\!=\!\!\dashv t_2$ *then the bag of the cut duplicities of* t_1 *is the same as that of* t_2*.*

Proof.

(i) Clearly (1) and (2) strictly reduce the bag, (7) and (8) will generally reduce the bag when the set I is not a singleton (but will certainly never increase the bag), and (3) and (4) never affect the bag. This leaves (5) and (6): we shall consider (5) in detail ((6) is dual).

The rewrite is

$$\langle f_i \rangle_{i\in I} \,;\, g \Longrightarrow \langle f_i \,;\, g \rangle_{i\in I}$$

Notice that on the left hand side the duplicity of g is

$$\mathrm{dupl}[g] = n \cdot \sum_{i\in I} \mathrm{width}[f_i]$$

where n is the duplicity at the root of the rewriting. On the right hand side g occurs in (possibly) multiple places but the duplicity of the cuts in the g in each $f_i \,;\, g$ depends multiplicatively on the root duplicity value $n \cdot \mathrm{width}[f_i]$. So the sum of the duplicities of these cuts does not change. This in turn means that the bag will not increase, since if there is duplication, each new cut must have strictly lower duplicity (as their sum is invariant).

Finally we must consider the principal cut of the reduction (when I is non-empty):

$$\begin{aligned} \mathrm{cutdupl}[\langle f_i \rangle_{i\in I} \,;\, g] &= \mathrm{dupl}[\langle f_i \rangle_{i\in I} \,;\, g] \cdot \mathrm{width}[\langle f_i \rangle_{i\in I}] \cdot \mathrm{width}[g] \\ &= \mathrm{dupl}[\langle f_i \rangle_{i\in I} \,;\, g] \cdot (\sum_{i\in I} \mathrm{width}[f_i]) \cdot \mathrm{width}[g] \\ &= \sum_{i\in I} \mathrm{dupl}[\langle f_i \rangle_{i\in I} \,;\, g] \cdot \mathrm{width}[f_i] \cdot \mathrm{width}[g] \\ &= \sum_{i\in I} \mathrm{dupl}[\langle f_i \,;\, g \rangle_{i\in I}] \cdot \mathrm{width}[f_i] \cdot \mathrm{width}[g] \\ &= \sum_{i\in I} \mathrm{dupl}[f_i \,;\, g] \cdot \mathrm{width}[f_i] \cdot \mathrm{width}[g] \\ &= \sum_{i\in I} \mathrm{cutdupl}[f_i \,;\, g] \end{aligned}$$

Which shows that the sum of the duplicity of the introduced cuts does not exceed that of the original cut. Therefore, again the bag of duplicities does not increase.

(ii) It is an easy observation now from the proof of (i) that when the rewrite duplicates terms there is a strict reduction in the bag (as all costs are at least 1).

(iii) Notice that the duplicity of subterms is not affected by any of the equalities.

■

The other measure we shall use is the height of the cut. The problem with the duplicity is that it may not decrease with a rewriting when there is no duplication of subterms. This means that we need a second measure essentially to catch the case when there is no duplication. We define the height of a term as:

- $\mathrm{hgt}[a] = 1$ where a is an atomic map (or an identity),
- $\mathrm{hgt}[\langle f_i \rangle_{i\in I}] = \mathrm{hgt}[(f_i)_{i\in I}] = \max\{\mathrm{hgt}[f_i] | i \in I\} + 1$,
- $\mathrm{hgt}[b_k(f)] = \mathrm{hgt}[p_k(f)] = \mathrm{hgt}[f] + 1$,
- $\mathrm{hgt}[f \,;\, g] = \max\{\mathrm{hgt}[f], \mathrm{hgt}[g]\} + 1$.

We then say that the **height of a cut** is

$$\mathrm{cuthgt}[f \,;\, g] = \mathrm{hgt}[f] + \mathrm{hgt}[g]$$

2.7. Lemma.

(i) *If* $t_1 \Longrightarrow t_2$ *then*

- $\mathrm{hgt}[t_1] \geq hgt[t_2]$,
- *The height of each non-principal cut does not increase,*
- *The height of any cut produced from the principal cut is strictly less than the height of the principal cut.*

(ii) *If* $t_1 \models\!\!\!=\!\!\!\dashv t_2$ *then* $\mathrm{hgt}[t_1] = \mathrm{hgt}[t_2]$ *and the height of each cut in* t_1 *and* t_2 *is unchanged.*

Proof. By inspection the height of a term across a rewrite does not increase. This means that cuts below a redex will not increase their cut height on a rewriting. Similarly, cuts whose terms do not contain the redex of the rewriting will not change cost. Finally observe that the cuts which replace principal cuts always have smaller height. ■

Now we define $\alpha(t)$ (the measure we actually want!) to be the bag whose elements are, for each cut in t, the pair consisting of the cut duplicity followed by the height. We order these pairs lexicographically. Notice that the lexicographical ordering is a well-ordering and consequently these bags are well ordered.

2.8. Lemma. $\alpha\colon T \longrightarrow \mathrm{bag}(\mathbf{N}^2)$, *as defined above, has the following properties:*

$$t_1 \Longrightarrow t_2 \quad \textit{implies } \alpha(t_1) > \alpha(t_2)$$

$$t_1 \models\!\!\!=\!\!\!\dashv t_2 \quad \textit{implies } \alpha(t_1) = \alpha(t_2).$$

Proof. Clearly the equalities do not affect these bags. The reductions (3), (4), (7), and (8) do not affect the duplicity of any cuts nor do they cause any duplication of subterms. Thus, the number of cuts is unchanged, however, their cut heights are strictly decreased. In (5) and (6) when there is no duplication (the set I is a singleton) the cut heights decrease. When there is duplication (the set I has more than one element) the duplicity of all the cuts affected decrease. When the set I is empty then all the cuts of g (and the principal cut) are removed, thus there is an obvious strict decrease. Similarly as (1) and (2) remove a cut these rules give a strict decrease.

Neither duplicity nor height are affected by the equalities thus these bags are not affected by the equalities. ■

This completes the proof of the proposition:

2.9. Proposition. $\Sigma\Pi_{\mathbf{A}}$ *under the rewrites (1) – (8) is confluent modulo the equations (9) – (12).*

3. Deciding the ΣΠ-conversions

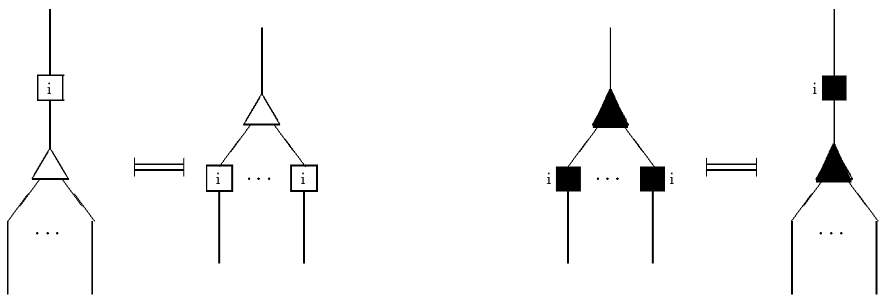
From the above, it is clear that given any two derivations, deciding their equivalence (or equality in the term model category) reduces to deciding equivalence of cut-free proofs. To this end, we must replace any cuts involving atomic formulas with the atomic sequents given by the appropriate composition in the generating category **A**. In effect this makes the decision procedure a relative one depending on a decision procedure for **A**.

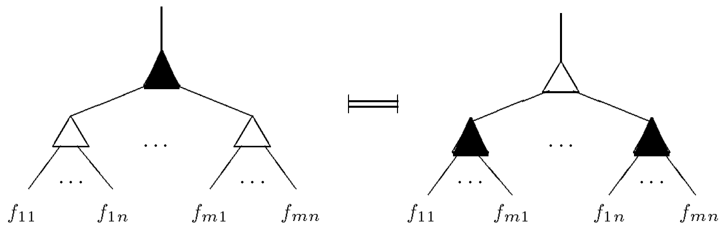
We have two graphically-inspired approaches to this decision procedure. We shall present one here, which allows further generality and so seems superior; further, it easily allows an interpretation in a compositional variant of Blass games, where the types are thought of as 2–player input–output games, and the terms as strategies (or also as single-channel communications between processes). An alternate approach is presented in Appendix B. It is also possible to orient the conversions (left to right as presented above) to have a Church–Rosser system. This is discussed in Appendix A.

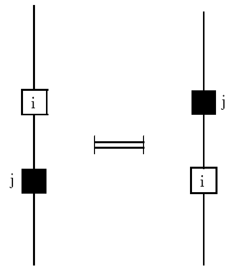
Before proceeding to a detailed presentation of the decision procedure, it will perhaps be helpful to consider the process graphically, with an example. The decision procedure operates on pairs of terms representing cut-free derivations of a given sequent. We use one term as a "template" for transforming the other term into one of the same shape. The key idea is to try to force the second term to start with the same proof rule as the template; if this is possible, then proceed inductively with the subterms. If this fails then the two terms are not equivalent. By using one term as a template in this manner one provides an order to the search for the conversions which can make the terms equivalent.

This can be described using the term calculus displayed above, but is clearer with a simple graphical representation of the terms along the following lines. With a term we can associate a term-graph, whose nodes represent the subterms of the term. We shall denote tupling by a black triangle, which has "output" edges for each component of the tuple, and cotupling by similar white triangles. We shall denote projections by decorated black boxes, the decoration indicating which component is being projected; similar white boxes denote injections. Atomic sequents will be represented by oval nodes containing the atomic term, as will identities on atomic formulas.

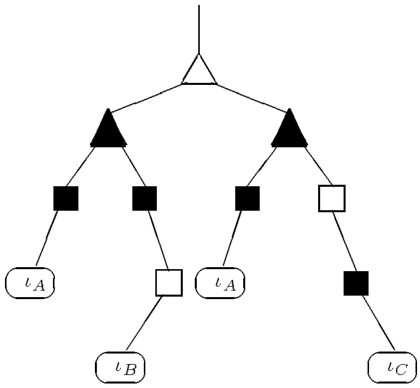
With these conventions, the four permuting conversions may be represented by the following graph equivalences.

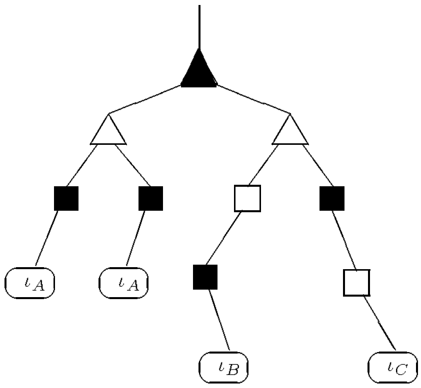






To illustrate the graphical representation, the derivation (1) at the start of the paper can be represented either by the term $\langle (p_1(\iota_A), p_2(b_1(\iota_B))), (p_1(\iota_A), b_2(p_2(\iota_C)))\rangle$, or equivalently by the following graph on the left, where for simplicity we have indicated the injection and projection indices by an output edge sloped to the left or the right. Note that clearly the graph is a quite direct representation of the derivation tree.



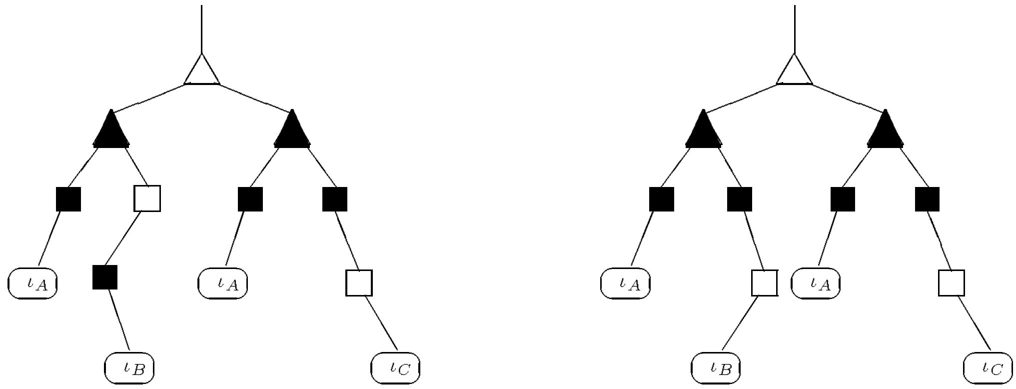


Notice that the derivation (1) at the start of the paper is equivalent to the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}}{A \times B \vdash A} \quad \dfrac{\dfrac{}{A \vdash A}}{A \times C \vdash C}}{(A \times B) + (A \times C) \vdash A} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash B}}{A \times B \vdash B}}{A \times B \vdash B + C} \quad \dfrac{\dfrac{\dfrac{}{C \vdash C}}{C \vdash B + C}}{A \times C \vdash B + C}}{(A \times B) + (A \times C) \vdash B + C}}{(A \times B) + (A \times C) \vdash A \times (B + C)}$$

which is given by the term $(\langle p_1(\iota_A), p_1(\iota_A)\rangle, \langle b_1(p_2(\iota_B)), p_2(b_2(\iota_C))\rangle)$, or equivalently by the graph on the right above.

We shall illustrate the decision procedure with this example. Take the left graph as template. The first step in the procedure is to see if the graph on the right can start the same way as the graph on the left. This means we have to move a white triangle up into the topmost level. We search down the paths of the right tree until we find a white triangle that can be moved upwards in the necessary manner — in this case we quickly find one at the second level. Moving it up gives us the graph on the left below.



Moving down a level we repeat (inductively) the process for all subtrees starting at second level nodes. Of course in this case this is already done, so we proceed down to the third level, where we check the subtrees left to right (say). The first node is "correct" but the second is not — we must move a black (projection) box up into this position, replacing the white (injection) box. Looking down the path(s) from this node, we search for the first black box which can be moved up — we find it one level down, and so move it up, which produces the tree at the right above. The third node is correct, so we turn to the last node. This requires a move of a white box up into the third level, analogous to the previous move; this produces the required graph, (*i.e.* we have transformed the original right graph into the "template" graph), and so completes the proof that the two original derivations are equivalent. In general, the decision procedure proceeds in this recursive manner.

3.1. Deciding the conversions: the details. A term is **Π-inert** if it can be built inductively by the rules:

- A variable $x_1, \ldots$ is Π-inert,
- $b_k(t)$ is Π-inert whenever t is Π-inert,
- $(t_i)_{i \in I}$ is Π-inert whenever each t_i is Π-inert.

Similarly a term is **Σ-inert** if it can be built inductively by the rules:

- A variable $x_1, \ldots$ is Σ-inert,
- $p_k(t)$ is Σ-inert whenever t is Σ-inert,
- $\langle t_i \rangle_{i \in I}$ is Σ-inert whenever each t_i is Σ-inert.

We observe:

3.2. LEMMA. *If t is a Π or Σ-inert term then there is no equality which applies to it.*

The result is immediate as the redex for an equality always involves a mixture of Σ-inert and Π-inert structure. Of course, this is the point of inert terms!

Let C be the constructors $b_k(\ )$, $p_k(\ )$, $\langle\ \rangle_{i\in I}$, or $(\ )_{i\in I}$, We shall say that a term starts with constructor C in case the first constructor in the term is C.

If t is a term the C-**prefix** of t, $\mathrm{prefix}_C[t]$, is defined as:

- If t starts with C then $\mathrm{prefix}_C[t] = \_$ (where $\_$ is the "anonymous" variable, that is a distinct variable which has not been used before and will not be used again).
- If t does not start with C then:
  - if $t = b_k(t')$ then $\mathrm{prefix}_C[b_k(t')] = b_k(\mathrm{prefix}_C[t'])$,
  - if $t = p_k(t')$ then $\mathrm{prefix}_C[p_k(t')] = p_k(\mathrm{prefix}_C[t'])$,
  - if $t = \langle t_i\rangle_{i\in I}$ then $\mathrm{prefix}_C[\langle t_i\rangle_{i\in I}] = \langle \mathrm{prefix}_C[t_i]\rangle_{i\in I}$,
  - if $t = (t_i)_{i\in I}$ then $\mathrm{prefix}_C[(t_i)_{i\in I}] = (\mathrm{prefix}_C[t_i])_{i\in I}$.

3.3. LEMMA. *Suppose t starts with constructor C then in any series of equalities*

$$t \models\!\!=\!\!\dashv t_1 \models\!\!=\!\!\dashv t_2 \models\!\!=\!\!\dashv \cdots\cdots\cdots \models\!\!=\!\!\dashv t_{n-2} \models\!\!=\!\!\dashv t_{n-1} \models\!\!=\!\!\dashv t_n\ ,$$

the C-prefix of each t_i is (either Π or Σ) inert.

PROOF. We shall say the C-**frontier** of a term with a C-prefix is those first occurrences across the term of the constructor C. Thus the C-frontier is the last constructor (necessarily C) of the C-prefix. Suppose that $t = b_k(t')$ and $t_i = w[b_k(t_1)/x_1, b_k(t_2)/x_2, ...]$ where w is Σ-inert. Either $t_i \models\!\!=\!\!\dashv t_{i+1}$ is an application of an equality at the $b_k(\ )$-frontier of the inert term w or beyond. If it is beyond the frontier then $\mathrm{prefix}_{b_k(\ )}[t_i] = \mathrm{prefix}_{b_k(\ )}[t_{i+1}]$; if it is on the frontier either it moves structure out of the inert term by shrinking the frontier (in which case $\mathrm{prefix}_{b_k(\ )}(t_{i+1})$ is certainly still inert (if smaller)), or it moves structure in to the prefix by expanding the frontier. However, only Σ-inert structure can be moved over $b_k(\ )$, so again $\mathrm{prefix}_{b_k(\ )}[t_{i+1}]$ is Σ inert.

Similar arguments hold for the remaining constructors. ∎

In a series of equalities emanating from a term which starts with C we may distinguish the steps which increase the C-inert prefix, $t_i \rightarrowtail\!\!\Rightarrow t_{i+1}$, those which decrease the C-inert prefix, $t_i \Leftarrow\!\!\prec t_{i+1}$, and those steps which do not affect the C-inert prefix, $t_i \models\!\!=\!\!\dashv t_{i+1}$.

3.4. LEMMA.

$$t_i \overset{\alpha}{\rightarrowtail\!\!\Rightarrow} t_{i+1} \overset{\beta}{\models\!\!=\!\!\dashv} t_{i+2}$$

can be rearranged as

$$t_i \overset{\beta'}{\models\!\!=\!\!\dashv} t'_{i+1} \overset{\alpha'}{\rightarrowtail\!\!\Rightarrow} t_{i+2}.$$

PROOF. The redex of β cannot be within the inert tree, nor by assumption is it on the frontier. Thus, it must be independent of α: so the equalities can be rearranged. ∎

This means that we can rearrange the steps in any proof of equality so that no C-inert prefix-increasing step happens before a step which does not affect the inert prefix. However, we cannot take these increasing steps past an inert prefix-*decreasing* step. However, a decreasing step is only possible if there has already been the corresponding (reverse) increasing step. However, this means we can cancel the decreasing step with the increasing step. We conclude:

3.5. LEMMA. *In any series of steps*

$$t \mathrel{\overset{*}{\vDash\!\!\dashv}} t_1 \mathrel{\overset{*}{\underset{\alpha}{\rightarrowtail}}} t_2 \mathrel{\underset{\beta}{\leftarrowtail}} t_3$$

where the starting construct of t determined the direction indicated on the arrows the decreasing step β can be cancelled.

This means:

3.6. PROPOSITION. *Any proof of equality from t to t' can be rearranged as*

$$t \mathrel{\overset{*}{\underset{\alpha}{\vDash\!\!\dashv}}} t_1 \mathrel{\overset{*}{\underset{\beta}{\rightarrowtail}}} t'$$

where the initial equalities do not touch the root constructor.

Notice that the second part of this proof, β, is essentially unique. There can be independent expansions of the inert frontier which can be reordered but every proof must do the same expansions. Notice that this part of the proof β can be read in reverse as a procedure which pulls the root structure of t to the root of t'.

3.7. LEMMA.

(i) *The structure $C = b_k(\ )$ or $C = (\quad)$ can be pulled to the root of t if and only if the C-prefix of t is Σ-inert,*

(ii) *The structure $C = p_k(\ )$ or $C = \langle\quad\rangle$ can be pulled to the root of t if and only if the C-prefix of t is Π-inert.*

PROOF. We have already observed that such a pulling up process results in an inert prefix (of the appropriate sort). Conversely given an inert prefix of the appropriate sort clearly means that that C-frontier can be contracted. ∎

Thus, we can test quite easily to see whether the last stage in this proof is possible. Because the equalities in the first part of this proof, α, do not touch the root constructor each equality must apply to one of the arguments of that constructor. Thus, for each argument we then have an equality proof: but each of these proofs can now be "normalized" into the form of the corollary. This gives us a normal form for (directed) equality proofs and whence an algorithm for determining equality which amounts to trying to grow such a proof by progressively matching the structure of t starting from the root by pulling up that structure to the root of the second term.

Let us call the procedure to pull a constructor up to the top of a term `pullup(C,t)`: it will return `success(t´)` if the constructor can be pulled up and `fail` otherwise. We shall schematically represent a term with the construction C at the root by `C(t1,..,tn)`. We then have:

3.8. ALGORITHM. *The following algorithm determines whether $t = t'$ with respect to the $\Sigma\Pi$ equalities:*

```
def PSequel(C(t_1,...,t_n),t') =
            (case pullup(C,t') of
                     success(C(t'_1,..,t'_n)) => PSequal(t_1,t'_1)
                                                 and ... and PSequal(t_n,t'_n)
                   | fail => false )
  | PSequal(atom(t),atom(t')) = atomequal(t,t')
```

The fact that proofs of equality have the normal form discussed above suffices to show that this algorithm is correct.

4. Categorical semantics

We now wish to establish that $\Sigma\Pi_{\mathbf{A}}$ is a category; note that in particular, we have neither assumed nor shown that the composition given by cut is associative. In this section we shall establish this and show that $\Sigma\Pi_{\mathbf{A}}$ is the free category with products and coproducts generated by **A**. Following Joyal, we will summarize this by a "Whitman theorem" which characterizes the free category with products and coproducts up to equivalence.

The main logical result that allows all these results to follow, more or less as immediate corollaries, is the following consequence of cut elimination.

4.1. PROPOSITION. *In $\Sigma\Pi_{\mathbf{A}}$:*

(i) *Any cut-free derivation of a sequent $\sum X_i \vdash Y$ is equivalent to one whose principal rule is cotupling,*

(ii) *Any cut-free derivation of a sequent $X \vdash \prod Y_j$ is equivalent to one whose principal rule is tupling,*

(iii) *Any cut-free derivation of a sequent $\prod X_i \vdash \sum Y_j$ has its principal rule either a projection or an injection,*

(iv) *Any cut-free derivation of a sequent $A \vdash \sum Y_j$, where A is an atom, has its principal rule an injection,*

(v) *Any cut-free derivation of a sequent $\prod X_i \vdash A$, where A is an atom, has its principal rule a projection.*

(vi) *Any cut-free derivation of a sequent $A_1 \vdash A_2$, where A_1 and A_2 are atoms, must be an axiom (*i.e. *a morphism of* **A***).*

Notice that this result can be extended to arbitrary derivations (not only cut-free ones) using the cut elimination procedure. For example, any derivation of a sequent $\sum X_i \vdash Y$ can be transformed to one whose principal rule is cotupling.

PROOF. The last four statements are immediate from an inspection of which (non-cut) sequent rules could be applied to the given situation. The first two statements are dual, and so it suffices to prove the first statement.

We shall argue by induction on the structure of the second formula, Y.

- If Y is an atom then the only rule which can be applied is the cotuple rule, so this must be the principal rule.
- If $Y = \prod_{j\in I} Y_j$ then the principal rule of the derivation must either be a cotupling or a tupling. If it is a tupling

$$\frac{\left\{\sum X_i \vdash Y_j\right\}_{j\in J}}{\sum X_i \vdash \prod_{j\in J} Y_j}\ \textit{tuple}$$

  then, using our inductive hypothesis, the proofs above this may be transformed to have cotupling as their principal rules. This allows us to use (12) to transform the derivation so that cotupling is the principal rule for the original proof.
- If $Y = \sum_{j\in J} Y_j$ then the principal rule of the derivation must either be a cotupling or an injection. If it is an injection

$$\frac{\sum X_i \vdash Y_k}{\sum X_i \vdash \sum_{j\in J} Y_j}\ \textit{injection}$$

  then by induction the proof above may be transformed to have its principal rule a cotupling. However, this allows us to use (9) to transform the cotupling into the principal rule for the whole proof.

■

4.2. Associativity of cut. The reduction rules and the permuting conversions together define an equivalence relation (which we shall denote $\sim$) on derivations of a sequent. Our categorical semantics will have derivations modulo this equivalence as morphisms. We have already established that there are identity morphisms (Lemma 1.3); now we can use Proposition 4.1 to establish that the composition given by cut is associative. (The analogous result for several related systems is established by Došen in [Došen 1999].)

4.3. LEMMA. *Given derivations $W \vdash_f X$, $X \vdash_g Y$, $Y \vdash_h Z$, the derivations $W \vdash_{(f;g);h} Z$ and $W \vdash_{f;(g;h)} Z$ are $\sim$-equivalent.*

PROOF. We prove this by structural induction on f, g, h; without loss in generality we may assume f, g, h are all cut-free.

Case i: $W = \sum W_i$ and $f = \langle f_i\rangle_i$. In this case note that $(f\ ;\ g)\ ;\ h \Longrightarrow (\langle f_i\ ;\ g\rangle_i)\ ;\ h \Longrightarrow \langle (f_i\ ;\ g)\ ;\ h\rangle_i$ and $f\ ;\ (g\ ;\ h) \Longrightarrow \langle f_i\ ;\ (g\ ;\ h)\rangle_i$ and so by induction these are equivalent. Note that the case where $Z = \prod Z_i$ and $h = (h_i)_i$ is dual.

Since any derivation of a sequent $\sum W_i \vdash X$ may be written equivalently to end with a cotuple, if $W = \sum W_i$ we are done. Similarly if $Z = \prod Z_i$ we may suppose h is a tuple, and so we are done.

Case ii: $W = \prod W_i$ and $f = p_k(f')$. In this case $(f \,;\, g) \,;\, h \Longrightarrow p_k(f' \,;\, g) \,;\, h \Longrightarrow p_k((f' \,;\, g) \,;\, h)$ and $f \,;\, (g \,;\, h) \Longrightarrow p_k(f' \,;\, (g \,;\, h))$ and again these are equivalent by induction. The case where $Z = \sum Z_i$ and $h = b_i(h')$ is dual.

Case iii: $W = \prod W_i$, $X = \sum X_j$ and $f = b_k(f')$. Here we may suppose that $g = \langle g_j \rangle_j$ (since X is a sum), so $(f \,;\, g) \,;\, h \Longrightarrow (f' \,;\, g_k) \,;\, h$ and $f \,;\, (g \,;\, h) \Longrightarrow b_k(f') \,;\, \langle g_j \,;\, h \rangle_j \Longrightarrow f' \,;\, (g_k \,;\, h)$ so by induction these are equivalent. The case where $h = p_k(h')$ is dual.

Case iv: $W = \prod W_i$, $X = \prod Y_j$ and $f = (f_i)_i$. Here we consider two subcases: (a) $g = b_k(g')$ (and so $Y = \sum Y_l$ and we may suppose that $h = \langle h_l \rangle_l$), and (b) $g = (g_j)_j$ (and then the only case we need worry about is if $h = p_k(h')$, since we have done the other possible cases).

In case (a), $(f \,;\, g) \,;\, h \Longrightarrow b_k(f \,;\, g') \,;\, \langle h_l \rangle_l \Longrightarrow (f \,;\, g') \,;\, h_k$ and $f \,;\, (g \,;\, h) \Longrightarrow f \,;\, (g' \,;\, h_k)$ and so by induction these are equivalent.

In case (b) $(f \,;\, g) \,;\, h \Longrightarrow (f \,;\, g_j)_j \,;\, p_k(h') \Longrightarrow (f \,;\, g_k) \,;\, h'$ and $f \,;\, (g \,;\, h) \Longrightarrow f \,;\, (g_k \,;\, h')$ and so by induction these are equivalent.

This concludes all the essential cases; if any of W, X, Y, Z are atomic, a quick check of the possibilities will show that one ends up with a case essentially just like one of the cases above. So this concludes the proof. ■

We can now immediately conclude the following.

4.4. Proposition. $\Sigma\Pi_{\mathbf{A}}$ *is a category, whose objects are the formula of the logic, and whose morphisms are $\sim$–equivalence classes of derivations.*

4.5. Categorical products and coproducts. Clearly $\Sigma\Pi_{\mathbf{A}}$ is a category which has additional structure: namely finite products and coproducts. Moreover, this additional structure is free. These facts are direct corollaries of Proposition 4.1.

4.6. Proposition. $\Sigma\Pi_{\mathbf{A}}$ *is the free category generated from* $\mathbf{A}$ *with finite products and coproducts.*

There is a potential set-theoretic problem concerning the size of $\Sigma\Pi_{\mathbf{A}}$ since the index sets for the products and coproducts are arbitrarily chosen. This is a side-effect of our somewhat non-traditional syntax (presentation) for finite sums and products. We shall take the view that there is a small way of specifying "all" the finite sets so that these size problems do not arise and to say a category has all finite coproducts and products is to say that there is a specified method of getting from these finitely indexed sets of objects to their products and coproducts. Essentially this means that freeness can be given "on the nose" rather than up to equivalence.

PROOF. First we must establish that $\Sigma\Pi_{\mathbf{A}}$ has the necessary products and coproducts. To do this we shall use Proposition 4.1 to establish the necessary hom-set bijections:

$$\mathrm{Hom}\left(\textstyle\sum_{i\in I} X_i, Y\right) \cong \prod_{i\in I} \mathrm{Hom}(X_i, Y) \qquad \text{and} \qquad \mathrm{Hom}\left(X, \textstyle\prod_{i\in I} Y_i\right) \cong \prod_{i\in I} \mathrm{Hom}(X, Y_i)$$

where $\prod_i H_i$ (for sets H_i) is the product in the category of sets. We know each morphism in $\mathrm{Hom}(\sum_{i\in I} X_i, Y)$ can be expressed in a cut free manner so that the principal rule is a cotupling. With this presentation of the morphism we may associate a morphism of $\prod_{i\in I} \mathrm{Hom}(X_i, Y)$. We must show that this correspondence respects the equivalence relation $\sim$. However, this is actually immediate from our decision procedure since, once the cotuple structure is made principal, equality is determined by equality of the arguments. The inverse of this correspondence is provided by cotupling derivations. This therefore provides the desired bijection. It remains to establish that it is natural in Y: this, however, is immediate in view of (5). Products are treated dually. Thus, $\Sigma\Pi_{\mathbf{A}}$ has coproducts and products.

To show that $\Sigma\Pi_{\mathbf{A}}$ is the free category generated from $\mathbf{A}$ with coproducts and products it suffices to observe that all the identities (1) – (12) must hold in any category with coproducts. This we have already observed is an easy exercise which we leave to the reader. ∎

The coproducts and products of $\Sigma\Pi_{\mathbf{A}}$ satisfy a further important property.

4.7. LEMMA. *(Softness of finite sums and products) The following diagram is a pushout in the category of sets.*

$$\begin{tikzcd}
\sum_{ij} \mathrm{Hom}(X_i, Y_j) \arrow[r] \arrow[d] & \sum_i \mathrm{Hom}\left(X_i, \sum_j Y_j\right) \arrow[d] \\
\sum_j \mathrm{Hom}\left(\prod_i X_i, Y_j\right) \arrow[r] & \mathrm{Hom}\left(\prod_i X_i, \sum_j Y_j\right)
\end{tikzcd}$$

PROOF. This is almost immediate from Proposition 4.1: the necessary identification of derivations of $\prod_i X_i \vdash \sum_j Y_j$ whose last step was an injection with those whose last step was a projection needs the permuting conversion (11). ∎

This property is called "softness" by Joyal [Joyal 1995, and other references]. Here we are using the version suitable for finite products and coproducts rather than for arbitrary colimits and limits. This was an essential component in establishing what Joyal called a "Whitman theorem"; we follow Joyal's lead in paying tribute to [Whitman 1941], who established the lattice case of the theorem. This result is, in a strong sense, a categorical formulation of cut-elimination.

4.8. THEOREM. $\Sigma\Pi_{\mathbf{A}}$ *has the following properties:*

- $\Sigma\Pi_{\mathbf{A}}$ *is "soft", meaning that it satisfies Corollary 4.7.*

- *For any atomic A,*

$$\mathrm{Hom}(A, \textstyle\sum_j Y_j) \cong \displaystyle\sum_j \mathrm{Hom}(A, Y_j) \quad \textit{and} \quad \mathrm{Hom}(\textstyle\prod_i X_i, A) \cong \displaystyle\sum_i \mathrm{Hom}(X_i, A)$$

- $\Sigma\Pi_{\mathbf{A}}$ *is generated by* $\mathbf{A}$ *under finite sums and products.*

- *The inclusion* $\mathbf{A} \overset{i}{\hookrightarrow} \Sigma\Pi_{\mathbf{A}}$ *is full and faithful.*

Moreover, these properties characterize $\mathbf{A} \overset{i}{\hookrightarrow} \Sigma\Pi_{\mathbf{A}}$ *up to equivalence of categories.*

Proof. It is clear from the above that $\Sigma\Pi_{\mathbf{A}}$ satisfies these properties. So suppose that $\mathbf{A} \xrightarrow{F} \mathbf{B}$ also satisfies the properties. Freeness guarantees that there is a comparison functor $\Sigma\Pi_{\mathbf{A}} \xrightarrow{F^*} \mathbf{B}$, and it follows from cut-elimination (in particular, from Proposition 4.1 — this is essentially Joyal's main argument) that F^* must be full, faithful, and essentially surjective. We sketch the argument.

F^* is full, because any $b\colon F^*(X) \longrightarrow F^*(Y)$ can be "decomposed" into a "$\Sigma\Pi$-word" of simpler functions, depending on the form of $F^*(X)$ and $F^*(Y)$, using the Whitman properties (the hypothesis of the theorem). Then by induction the simpler functions must be in the image of F^*, and since F^* preserves sums and products we can use the same $\Sigma\Pi$-word to place b in the image of F^*. (A formal proof by induction can be constructed along these lines.)

To show F^* faithful, consider a parallel pair of arrows $F^*(f), F^*(g)\colon F^*(X) \longrightarrow F^*(Y)$ so that $F^*(f) = F^*(g)$: as above we can decompose $F^*(f)$ and $F^*(g)$ into words. Since $F^*(f), F^*(g)$ are typed the same, both are decomposed into different substitution instances of the same $\Sigma\Pi$-word. Since F^* is full, these words involve simpler functions of the form $F^*(h)$. By induction the corresponding subterms h are equal, and so then too $f = g$.

To see that F^* is essentially surjective, notice that since $\mathbf{B}$ is generated from $\mathbf{A}$, each object of $\mathbf{B}$ is isomorphic to an object in the image of F^*. ■

Appendices

A. The equivalences as rewrites

In Section 3 we gave a decision procedure for the $\Sigma\Pi$ conversions, in a fashion that maintained the symmetry of the logic. Following a suggestion of Luigi Santocanale, we also note that it is possible to orient the conversions as rewrites, in a manner that produces a confluent and terminating rewrite system. These rewrites are given as follows.

$$\boxed{\begin{array}{rcl} b_k(\langle f_i\rangle_{i\in I}) & \Longrightarrow & \langle b_k(f_i)\rangle_{i\in I} \\ (p_k(f_i))_{i\in I} & \Longrightarrow & p_k((f_i)_{i\in I}) \\ b_l(p_k(f)) & \Longrightarrow & p_k(b_l(f)) \\ (\langle f_{ij}\rangle_{i\in I})_{j\in J} & \Longrightarrow & \langle (f_{ij})_{j\in J}\rangle_{i\in I} \end{array}}$$

These are slightly unsatisfactory, in that they do not respect the symmetry we have been emphasizing, but since they do not have any critical pairs among themselves, and introduce no intractable critical pairs when combined with the cut elimination rewrites, they do lead to a very simple decision procedure. Note that since they do not exhibit the sum–product symmetry, they can be dualized to produce another such rewrite system. It would be nice to see one or the other of these systems arise from natural logical considerations such as produced the other rewrites.

B. Cellular Squares

In this section we illustrate a second alternate approach to deciding equivalence of derivation terms (in the cut-free case only), which was suggested to the authors by Peter Selinger. We do this by introducing an alternate presentation of these terms, one which is particularly good at catching the identities (9, 10, 11, 12). This method does not seem to be able to handle the units 0 and 1 however; a similar disadvantage occurred with Girard's approach [Girard 1995], an approach that seems to have some connections with these cellular squares.

Atomic derivations will be represented by labelled squares. We then introduce four operations on arrays of rectangular cells as illustrated in Figure 1.

Some notes concerning Figure 1: we have illustrated only the binary case, since it allows some simplifications. In general, we would have to decorate the shaded vertical and horizontal bars to indicate which projection is involved — here we have indicated that by shading to the left (for the first projection) or to the right (for the second projection), and similarly for the injections. Likewise we would concatenate more rectangles for tupling and cotupling. Notice also that the dimensions of the cells provide clues as to what operations have been applied — these are not topological diagrams, invariant under stretching. The shaded bars are placed *inside* the rectangles concerned, whereas the concatenated cells are placed adjacent to one another. This may be illustrated by the cellular "square" in Figure 2, which represents the derivation (1) at the beginning of the paper, or equivalently, the term $\langle (p_1(\iota_A), p_2(b_1(\iota_B))), (p_1(\iota_A), b_2(p_2(\iota_C)))\rangle$. Note that the lower two squares have horizontal bars, since they are injections (as well as projections — all four squares have vertical bars).

Of course, we must then also describe the permuting conversions — these are listed in Figures 3, 4. We have illustrated only some cases — in the cases of Equations (9), (10), there is a restriction that the same projection (or injection) must be involved on each side of the equation. In Equation (11), any projections (or injections) may be involved. In these four equations, we have exaggerated the joins to indicate the order of the operations concerned.

Since it is clear that once drawn, these conversions really are identities (one can no longer distinguish the order in which the relevant operations were applied), it is clear that equivalent terms will have the same cellular square representation. But the converse is also true: if one considers the various pairs of operations that may be applied in building

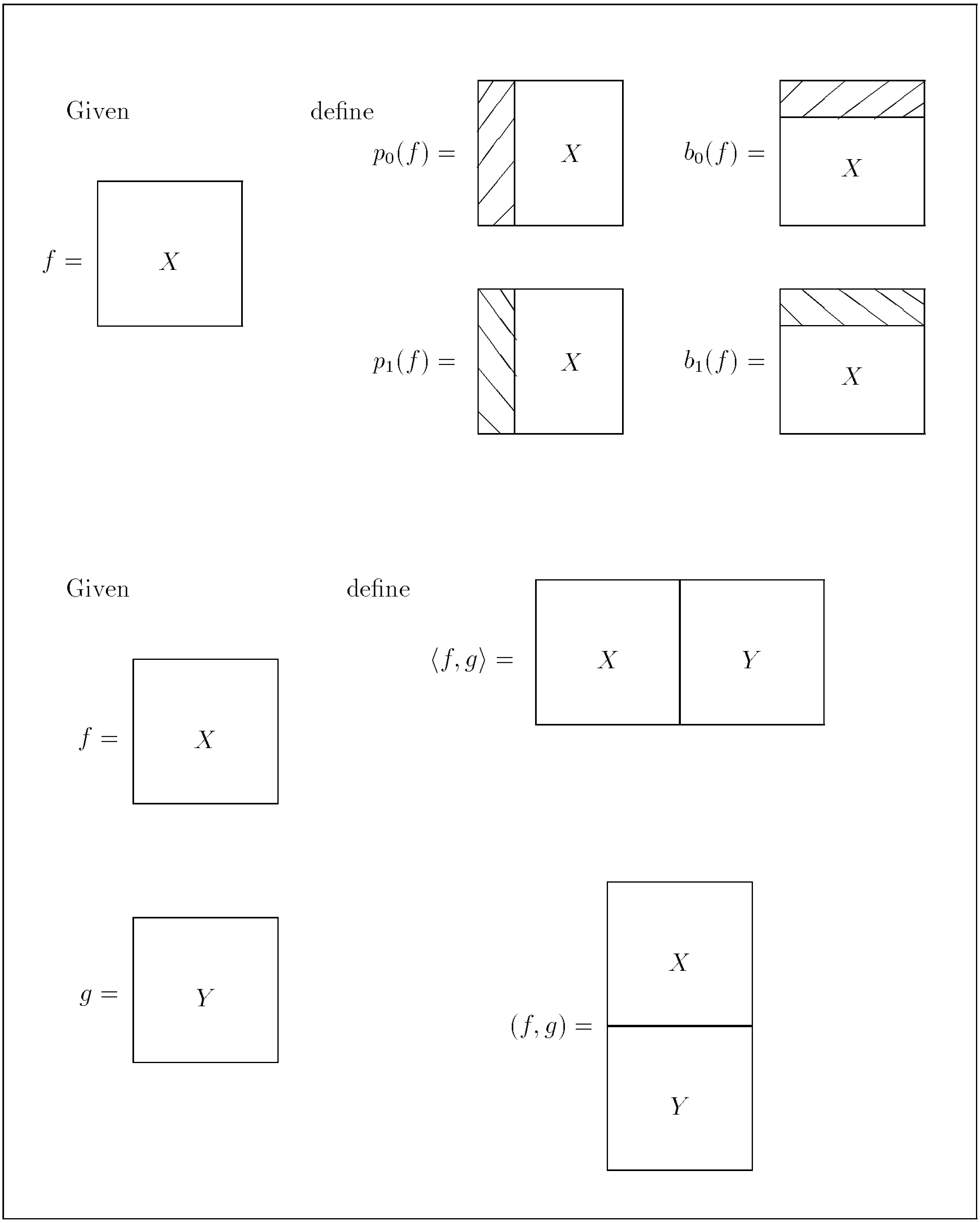
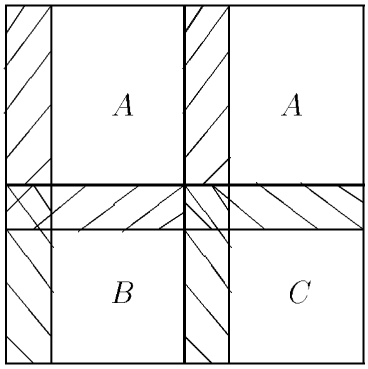


Figure 1: Cellular Squares



$$\langle(p_1(\iota_A), p_2(b_1(\iota_B))), (p_1(\iota_A), b_2(p_2(\iota_C)))\rangle$$

Figure 2: An example of a cellular "square"

up a cellular square, it will be apparent that the only cases where there is any ambiguity are the four considered here, and so if two terms with the same type have the same cellular square representation, they must indeed be equivalent. Note the restriction that the terms have the same type is necessary, with the presentation given here: for example, $\langle p_i(f), g\rangle$ and $p_i(\langle f, g\rangle)$ might both be represented by a pair of squares, one on the left with a vertical bar, and its neighbour on the right without a bar. These terms are differently typed, however, and so cannot be equal. This ambiguity is caused by the fact that the scope of the bars is only half indicated; we could amplify the notation by including some marker for the other half of the scope, but this seems an unnecessary complication since this representation of terms need only be applied once the terms are known to have the same type.

C. The nullary cases

There may be some confusion about how to apply our term calculus and the reductions and permuting conversions in the case $I = \emptyset$: we make those special cases explicit here. In the following, we use the abbreviations $\sum_\emptyset = 0$ and $\prod_\emptyset = 1$.

First, the nullary versions of cotupling and tupling:

$$\frac{}{0 \vdash_{\langle\rangle} Y}\ \mathit{cotuple} \qquad \frac{}{X \vdash_{()} 1}\ \mathit{tuple}$$

In fact, the notation here could be improved: although with the typing the notation is unambiguous, adding the typing to the terms makes it easier to read them as free-standing entities. So we shall write the terms above as $\langle\rangle_Y$ and $()_X$ respectively.

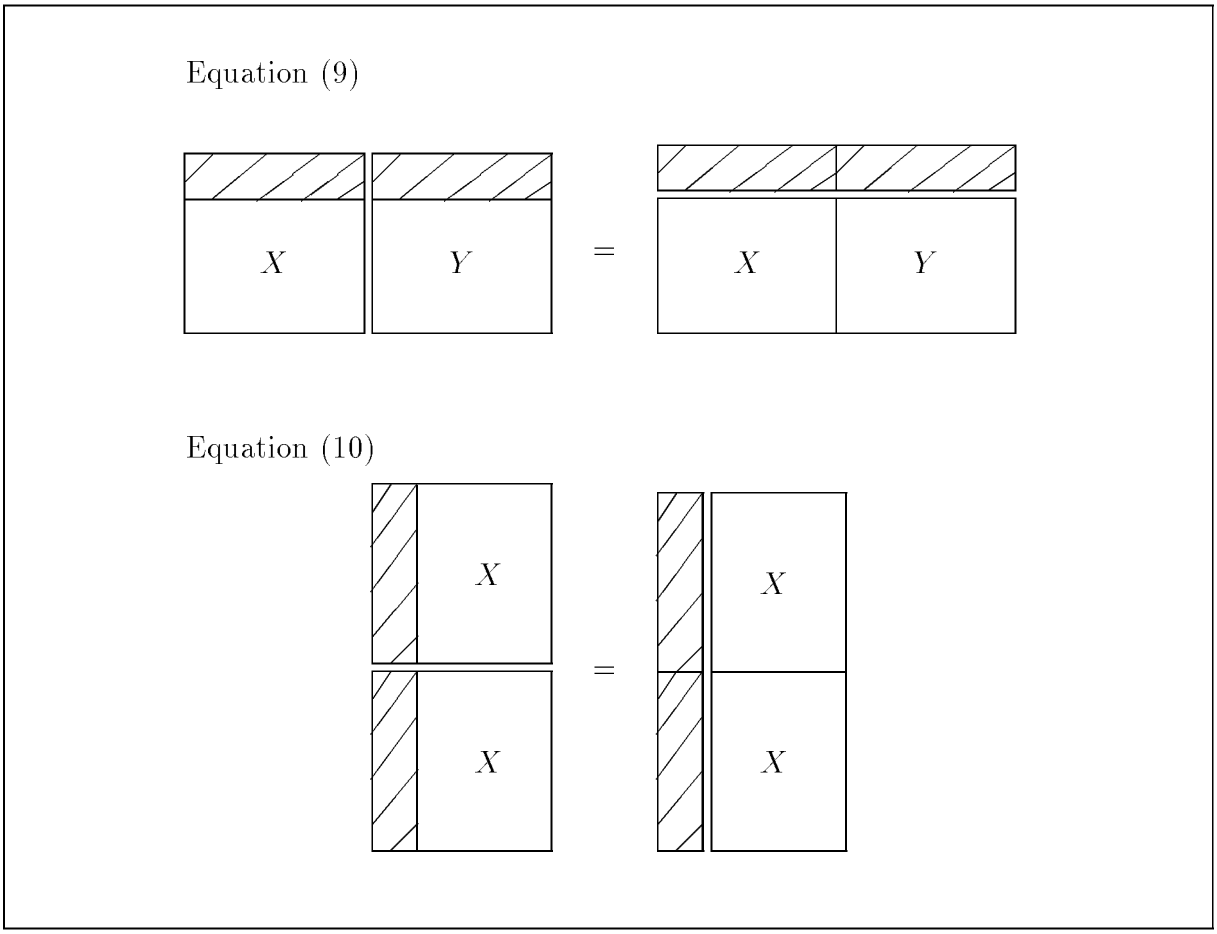
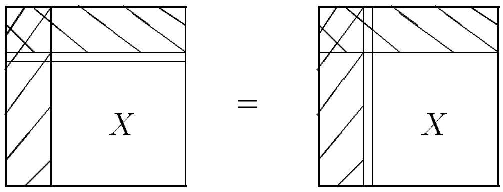


Figure 3: Permuting conversions (9), (10)

Equation(11)



Equation (12)

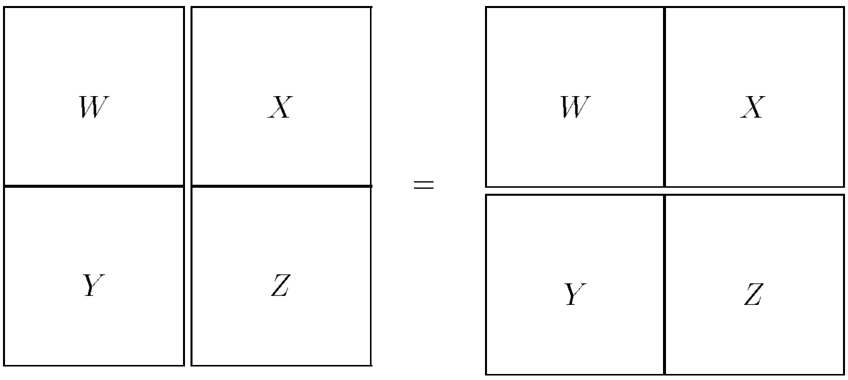


Figure 4: Permuting conversions (11), (12)

There are two reductions that are relevant to this setting, corresponding to the cases of (5) and (6) where $I = \emptyset$. Given terms $f\colon X \vdash Y$ and $g\colon Y \vdash Z$ we have the following reductions. (To aid the reader, we indicate the typing, though it may be deduced easily from the terms.)

$$\begin{array}{rcll} \langle\rangle_Y \,;\, g & \Longrightarrow & \langle\rangle_Z & : 0 \vdash Z \qquad (5) \\ f \,;\, ()_Y & \Longrightarrow & ()_X & : X \vdash 1 \qquad (6) \end{array}$$

In addition there are five permuting conversions, corresponding to the cases of (9) and (10) when $I = \emptyset$, and to three variants of (12), corresponding to the cases when only $I = \emptyset$, only $J = \emptyset$, and both $I = J = \emptyset$.

$$\begin{array}{rcll} b_k(\langle\rangle_{X_k}) & \models\!\!\!=\!\!\!\dashv & \langle\rangle_{\Sigma_J X_j} & : 0 \vdash \sum_J X_j \qquad (9) \\ ()_{\Pi_J X_j} & \models\!\!\!=\!\!\!\dashv & p_k(()_{X_k}) & : \prod_J X_j \vdash 1 \qquad (10) \\ (\langle\rangle_{Y_j})_{j\in J} & \models\!\!\!=\!\!\!\dashv & \langle\rangle_{\Pi_J Y_j} & : 0 \vdash \prod_J Y_j \qquad (12) \\ ()_{\Sigma_I X_i} & \models\!\!\!=\!\!\!\dashv & \langle()_{X_i}\rangle_{i\in I} & : \sum_I X_i \vdash 1 \qquad (12) \\ ()_0 & \models\!\!\!=\!\!\!\dashv & \langle\rangle_1 & : 0 \vdash 1 \qquad (12) \end{array}$$

With the reductions above, these are all (of course) special cases of the axioms (which we do not need to explicitly add) $f = \langle\rangle_Y$ and $g = ()_X$ for $f\colon 0 \vdash Y$ and $g\colon X \vdash 1$. These follow from the reductions and conversions given here, in view of our "Whitman theorem".

References

[Blute *et al.* 1996] R.F. Blute, J.R.B. Cockett, R.A.G. Seely, and T.H. Trimble (1996) Natural deduction and coherence for weakly distributive categories. Journal of Pure and Applied Algebra **113** 229–296.

[Dershowitz–Manna 1976] N. Dershowitz, Z. Manna (1976) Proving termination with multiset orderings. Comm. of the ACM 22, No. 8 (465–475).

[Došen 1999] K. Došen (1999) *Cut Elimination in Categories.* Trends in Logic 6, Kluwer Academic Pub.

[Girard 1987] J.-Y. Girard (1987) Linear logic. Theoretical Computer Science **50** 1–102.

[Girard 1995] J.-Y. Girard (1995) Proof-nets: the parallel syntax or proof-theory. In *Logic and Algebra*, New York, Marcel Dekker.

[Huet 1980] G. Huet (1980) Confluent reductions: abstract properties and applications to term rewriting systems. JACM 27, No. 4 (797–821).

[Joyal 1995] A. Joyal (1995) Free bicompletion of enriched categories. Math. Reports XVII, Acad. Sci. Canada, 213–218.

[Lambek–Scott 1986] J. Lambek and P.J. Scott (1986) *Introduction to Higher-Order Categorical Logic.* Cambridge studies in advanced mathematics 7, Cambridge University Press.

[Santocanale 1999] L. Santocanale (1999) Sur les μ-treillis libres. Ph.D. Thesis, Université de Québec à Montréal.

[Whitman 1941] P.M. Whitman (1941) Free lattices. Ann. of Math. 42, 325–330.

Department of Computer Science, University of Calgary,
2500 University Drive,
Calgary, AL, T2N1N4, Canada.

Department of Mathematics, McGill University,
805 Sherbrooke St.,
Montréal, PQ, H3A 2K6, Canada.

Email: `robin@cpsc.ucalgary.ca` and `rags@math.mcgill.ca`

This article may be accessed via WWW at `http://www.tac.mta.ca/tac/` or by anonymous ftp at `ftp://ftp.tac.mta.ca/pub/tac/html/volumes/8/n5/n5.{dvi,ps}`

THEORY AND APPLICATIONS OF CATEGORIES (ISSN 1201-561X) will disseminate articles that significantly advance the study of categorical algebra or methods, or that make significant new contributions to mathematical science using categorical methods. The scope of the journal includes: all areas of pure category theory, including higher dimensional categories; applications of category theory to algebra, geometry and topology and other areas of mathematics; applications of category theory to computer science, physics and other mathematical sciences; contributions to scientific knowledge that make use of categorical methods.

Articles appearing in the journal have been carefully and critically refereed under the responsibility of members of the Editorial Board. Only papers judged to be both significant and excellent are accepted for publication.

The method of distribution of the journal is via the Internet tools `WWW/ftp`. The journal is archived electronically and in printed paper format.

Subscription information. Individual subscribers receive (by e-mail) abstracts of articles as they are published. Full text of published articles is available in .dvi, Postscript and PDF. Details will be e-mailed to new subscribers. To subscribe, send e-mail to `tac@mta.ca` including a full name and postal address. For institutional subscription, send enquiries to the Managing Editor, Robert Rosebrugh, `rrosebrugh@mta.ca`.

Information for authors. The typesetting language of the journal is TEX, and LATEX is the preferred flavour. TEX source of articles for publication should be submitted by e-mail directly to an appropriate Editor. They are listed below. Please obtain detailed information on submission format and style files from the journal's WWW server at `http://www.tac.mta.ca/tac/`. You may also write to `tac@mta.ca` to receive details by e-mail.