NORMAL PROCESS REPRESENTATIVES

Vijay Gehlot *Carl Gunter* *

University of Pennsylvania
Department of Computer and Information Sciences
Philadelphia, PA 19104 U.S.A.

October 1989

Abstract

This paper discusses the relevance of a form of cut elimination theorem for linear logic tensor theories to the concept of a process on a Petri net. We base our discussion on two definitions of processes given by Best and Devillers. Their notions of process correspond to equivalence relations on linear logic proofs. It is noted that the cut reduced proofs form a process under the finer of these definitions. Using a strongly normalizing rewrite system and a weak Church-Rosser theorem, we show that each class of the coarser process definition contains exactly one of these finer classes which can therefore be viewed as a canonical or *normal* process representative. We also discuss the relevance of our rewrite rules to the categorical approach of Degano, Meseguer, and Montanari.

1 Introduction

It has often been useful to take ideas from proof theory and look at their computational significance. One very fruitful line of investigation has been the use of the Curry-Howard correspondence—the "propositions as types" idea—as a way of seeing proofs as programs and types as specifications. This correspondence reveals an analogy between cut elimination in systems of natural deduction and the reduction of lambda-terms, thus strongly connecting the study of a central proof-theoretic idea (with a history dating back at least to the 1930's) with a central computational concept in sequential functional programmming.

Another, more recent, line of investigation with a kinship to this sequential theory concerns the relationship between certain kinds of proofs and concepts in *concurrency.* A number of authors have discussed the idea of relating concurrent computations as represented by *Petri nets* to proofs in *linear logic* [7]. One line of research seeks to use the fact that nets give rise to a monoid structure and can therefore be used to model linear logic through the use of a phase semantics [6]. In this way a net can be viewed as a model of the linear connectives in which there is a correspondence between the *truth* of a linear sequent in the model and the *reachability* relation on the net. However, most of the research [8, 9, 1, 4] has focused on the idea that a net may be viewed as a *theory* in a fragment of linear logic (the tensor theory to be precise). In particular, when things are viewed in this way, there is a precise correspondence between *concurrent computations* on a Petri net and linear logic

*Research of both authors is supported by the Office of Naval Research. Electronic mail addresses for the authors are `gunter@cis.upenn.edu` and `gehlot@cis.upenn.edu`

proofs in its associated theory. This opens a way to investigate a transfer of ideas between proof theory and the theory of concurrent computation.

The purpose of this paper is to look at the significance of the cut elimination theorem of linear logic tensor theories in the context of computations on Petri nets. The role of cut in this theory is somewhat different from cut in the context of "propositions as types". A linear proof corresponds to a net computation and the elimination of a cut corresponds to a *transformation* of that computation, rather than an *enactment* of the computation. The authors of this paper have suggested before [8] that cut elimination corresponds a form of optimization in which a computation is transformed into a "more concurrent" computation. It is our goal in this paper to show how this view of the significance of the cut fits into a theory of Petri net processes.

In particular, we show that linear logic cut elimination provides a way of understanding the relationship between two definitions of the notion of a net process studied in Best and Devillers [3]. We will define a pair of relations $\mathcal{S}$ and $\mathcal{T}$ on linear logic proofs which corresponds to two of the concepts of net process discussed in [3]. The relation $\mathcal{T}$ is coarser than $\mathcal{S}$ and relates some computations which display different levels of dependency in their descriptions (i.e. one description permits two things to be done at the same time while the other description sequentializes them). What we wish to show is that there is a *unique* $\mathcal{S}$-equivalence class of processes in each $\mathcal{T}$-equivalence class τ which can be viewed as a "maximally concurrent" *representative* of τ. Moreover, the members of this $\mathcal{S}$-equivalence class will be exactly the set of cut reduced proofs in τ. Since we can demonstrate a strongly normalizing rewrite system for linear proofs which preserves $\mathcal{T}$-equivalence, we can therefore view the distinguished $\mathcal{S}$-equivalence class in τ as a *normal representative process* for it. In general, normal representatives of classes of $\mathcal{T}$ may be viewed as the "maximally concurrent" computations on a net.

Our first primary technical result is a strong normalization theorem (Theorem 2) for a set of rewrites which preserve $\mathcal{T}$-equivalence. Termination for the rewrites can be proved using a measure on cut formulas and the number of nodes in a proof tree. Our second primary technical result is a weak Church-Rosser theorem (Theorem 3) for the action of the rewrite system on equivalence classes of $\mathcal{S}$. Confluence of our rewrite system therefore follows from Newman's Lemma. This result is then used to show that there is a unique normal process representative in each equivalence class of $\mathcal{T}$(Theorem 4).

We begin by discussing some relevant work and definitions in section 2 where we also define the two equivalence relations. In section 3 we give a set of rewrite rules on proofs which are shown to be strongly normalizing. Section 4 contains our second main result where we prove the Church-Rosser property for the induced rewrite rules on $\mathcal{S}$-equivalent classes. We then define an equivalence based on this notion of reduction and show that this equivalence coincides with $\mathcal{T}$-equivalence giving us the desired theorem about unique process representatives. Finally, in section 5 we discuss relationship between reduction on proofs and rewrite rules for arrows which arise by interpreting proofs as arrows.

2 Two theories of processes.

In this section we introduce the background on processes and linear logic needed to understand the central results of the paper. First, let's start with an example. Consider the net pictured in Figure 1. It has six *places,* drawn as circles and marked A, B, C, A', B', C' and it has five *transitions,* written as rectangles and marked r, s, r', s', t. The two closed circles on the places A and A' are tokens which indicate the availability of the "resources" A and A'. In the configuration in the picture, the transitions r and r' are *enabled* by the fulfillment of their preconditions A and A'

Figure 1: Net N with two processes of type $A \otimes A' \to C \otimes C'$.

respectively. Dynamically, computations proceed by the *firing* of transitions. If transition r fires, for example, then the token is removed from A and placed on its *postcondition* B; the transition r is now disabled since its precondition A is no longer filled. We may also speak of the *concurrent* firing of r and r' in the starting configuration of Figure 1 since there is no dependency between their pre-conditions.

For formal definitions we refer the reader to recent publications in **LICS** [10, 5]. For this paper we will take it as a working definition that a net (or, to be more precise, a place/transition net) is a set of pairs of multisets over a set S of places. This is really a special case of the definition of a net in [10, 5] where distinct transitions with the same pre and post conditions are permitted, but the restriction simplifies our notation since it avoids the need to label the linear sequents to preserve a precise correspondence between nets and linear tensor theories. For this preliminary discussion, it will be convenient to utilize their categorical treatment of nets and write transitions as arrows in a category with a binary operator $\otimes$ on its objects. In this notation, the transitions in the figure may be viewed as arrows:

$$\begin{array}{lc} r : A \to B & s : B \to C \\ & t : B \otimes B' \to C \otimes C' \\ r' : A \to B & s' : B \to C \end{array}$$

There are two operations on arrows. If $f : X \to Y$ and $g : Y \to Z$, then $f;g : X \to Z$ is the composition of f and g. If $f : X \to X'$ and $g : Y \to Y'$, then $f \otimes g : X \otimes Y \to Y \otimes Y'$ is the *tensor product* of f and g. Starting with the basic transitions, these operations generate a language of computations on the net. Intuitively we read $f;g$ as the *sequentialization* of f and g: "first do f and then do g". We read $f \otimes g$ as the *concurrent* performance of operations f and g: "do f and g at the same time".

Looking again at Figure 1, here are four sample computations of type $A \otimes A' \to C \otimes C'$ on the net N:

$$\begin{array}{lcl} f & = & (r \otimes A'); (s \otimes A'); (C \otimes r'); (C \otimes s') \\ f' & = & (r \otimes r'); (s \otimes s') \\ g & = & (r \otimes A'); (B \otimes r'); t \\ g' & = & (r \otimes r'); t \end{array}$$

where the idle transition (identity map) on a place X is written simply as X. Much of the research on nets (and, indeed, concurrency as a whole) has focused on the question of when two computations such as the ones above are "essentially the same". In the case of the computations above, one may well expect to distinguish between processes f and g, for example, since one of these computations

Structural Rules

$$\frac{}{A \vdash A}\ \text{(Identity)} \qquad \frac{\Gamma \vdash A \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\ \text{(Cut)}$$

Logical Rules

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ (\otimes\text{R}) \qquad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}\ (\otimes\text{L})$$

Figure 2: Structural and logical rules for the tensor fragment of linear logic.

performs transitions s and s' but not t, while the other performs t but neither s nor s'. On the other hand, it is debatable whether f and f' or g and g' should be identified. What this comes down to is a question of whether $\otimes$ is a *functor* or not, *i.e.* is it the case that the equation

$$(u \otimes v); (u' \otimes v') = (u; u') \otimes (v; v')$$

holds for arbitrary computations u, u', v, v' which make the equation type correct? Arguing one way, this is a pleasing equational property which is supported by a computational intuition that the only real difference between the left and right hand sides of this equality is the order in which things are done. Arguing against functoriality, it seems that computations f' and g' are somehow "better" than f and g, respectively, since they allow more independent computations to be performed concurrently. For example, f does not perform the transition r' until after r is complete, but this is unnecessary since r and r' can be done at the same time as in f'.

The dividing line between a theory of processes which identifies f with f' and g with g' and a theory which does not make these identifications has been carefully examined in [3] and [5]. We will not introduce the theory in the form that they have done, but instead present it as an equational theory of *proofs*. The reader can check that our presentations of the relations $\mathcal{S}$ and $\mathcal{T}$ for proofs as given below correspond to the equational theories with these names presented by Degano, Meseguer, and Montanari [5] in the last **LICS** symposium.

A linear *tensor formula* is either a propositional atom or has the form $A \otimes B$ where A and B are tensor formulas. A linear logic tensor *sequent* is a pair $\Gamma \vdash A$ where A is a tensor formula and Γ is a multiset. The rules for deriving sequents of this fragment of linear logic are given in Figure 2. A tensor theory is a set of pairs $A \vdash B$ where A and B are tensor formulas. Given a net N with places S, the *associated* tensor theory is the set of all sequents $A \vdash B$ where A and B are tensor formulas formed over atoms from S such that the pair of multisets $(M(A), M(B))$ determined by A and B is an element of the net N. There is a precise correspondence between computations on N and proofs over the associated theory in which uses of axioms in a proof correspond to firings of transitions on the net and uses of the rules for tensor on the right and cut correspond to the tensor and composition of computations respectively. We omit a further discussion of this correspondence here since it has been exposited adequately elsewhere ([9] provides a rigorous treatment for example). The remainder of this paper will be concerned with computations as represented by proofs.

We now begin a more formal discussion of proofs and the cut rule. In the presence of proper axioms it is not possible to *eliminate* cuts. Our concept of *cut reduction* will be based on a definition of a normal form for proofs and a system of rewrite rules for proofs not in normal form. We say that an instance of the cut rule in a proof is *trivial* if at least one of the premisses is an axiom of the form $A \vdash A$. A linear formula A is said to be a *netformula* (of T) if it appears in one of the formulas of the theory T. An instance of a cut rule is said to be *essential* (in a proof in the theory T) if it is non-trivial and has the form

$$\dfrac{\Gamma \vdash A \qquad\qquad A \vdash B}{\Gamma \vdash B}\;\mathit{Cut}$$

where A is a netformula. A proof is said to be in *normal* form if all of its cuts are essential. We will discuss normalization of proofs in the next section.

Definition 1 Let N be a net. The equivalence relation $\mathcal{S}(N)$ on proofs is defined as the smallest equivalence relation satisfying the following equations between proof trees.

(1)
$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma, B, C \vdash D\end{array} \qquad \begin{array}{c}\Pi'\\ \Delta \vdash E\end{array}}{\Gamma, B, C, \Delta \vdash D \otimes E}\;\otimes\mathrm{R}}{\Gamma, B \otimes C, \Delta \vdash D \otimes E}\;\otimes\mathrm{L} \quad = \quad \dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma, B, C \vdash D\end{array}}{\Gamma, B \otimes C \vdash D}\;\otimes\mathrm{L} \qquad \begin{array}{c}\Pi'\\ \Delta \vdash E\end{array}}{\Gamma, B \otimes C, \Delta \vdash D \otimes E}\;\otimes\mathrm{R}$$

(2)
$$\dfrac{\begin{array}{c}\Pi\\ \Gamma \vdash A\end{array} \qquad \begin{array}{c}\Pi'\\ \Delta \vdash B\end{array}}{\Gamma, \Delta \vdash A \otimes B}\;\otimes\mathrm{R} \quad = \quad \dfrac{\begin{array}{c}\Pi'\\ \Delta \vdash B\end{array} \qquad \begin{array}{c}\Pi\\ \Gamma \vdash A\end{array}}{\Delta, \Gamma \vdash B \otimes A}\;\otimes\mathrm{R}$$

(3)
$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma, B, C, \Delta, D, E \vdash F\end{array}}{\Gamma, B \otimes C, \Delta, D, E \vdash F}\;\otimes\mathrm{L}}{\Gamma, B \otimes C, \Delta, D \otimes E \vdash F}\;\otimes\mathrm{L} \quad = \quad \dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma, B, C, \Delta, D, E \vdash F\end{array}}{\Gamma, B, C, \Delta, D \otimes E \vdash F}\;\otimes\mathrm{L}}{\Gamma, B \otimes C, \Delta, D \otimes E \vdash F}\;\otimes\mathrm{L}$$

(4)
$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma \vdash A\end{array} \qquad \begin{array}{c}\Pi'\\ \Delta \vdash C\end{array}}{\Gamma, \Delta \vdash A \otimes B}\;\otimes\mathrm{R} \qquad \begin{array}{c}\Pi''\\ \Lambda \vdash C\end{array}}{\Gamma, \Delta, \Lambda \vdash A \otimes B \otimes C}\;\otimes\mathrm{R} \quad = \quad \dfrac{\begin{array}{c}\Pi\\ \Gamma \vdash A\end{array} \qquad \dfrac{\begin{array}{c}\Pi'\\ \Delta \vdash C\end{array} \qquad \begin{array}{c}\Pi''\\ \Lambda \vdash C\end{array}}{\Delta, \Lambda \vdash B \otimes C}\;\otimes\mathrm{R}}{\Gamma, \Delta, \Lambda \vdash A \otimes B \otimes C}\;\otimes\mathrm{R}$$

(5)
$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma \vdash A\end{array} \qquad \begin{array}{c}\Pi'\\ \Delta, A \vdash B\end{array}}{\Gamma, \Delta \vdash B}\;\mathit{Cut} \qquad \begin{array}{c}\Pi''\\ \Lambda, B \vdash C\end{array}}{\Gamma, \Delta, \Lambda \vdash C}\;\mathit{Cut} \quad = \quad \dfrac{\begin{array}{c}\Pi\\ \Gamma \vdash A\end{array} \qquad \dfrac{\begin{array}{c}\Pi'\\ \Delta, A \vdash B\end{array} \qquad \begin{array}{c}\Pi''\\ \Lambda, B \vdash C\end{array}}{\Delta, A, \Lambda \vdash C}\;\mathit{Cut}}{\Gamma, \Delta, \Lambda \vdash C}\;\mathit{Cut}$$

(6)
$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma, B, C \vdash D\end{array} \qquad \begin{array}{c}\Pi'\\ D \vdash E\end{array}}{\Gamma, B, C \vdash E}\;\text{e-cut}}{\Gamma, B \otimes C \vdash E}\;\otimes\mathrm{L} \quad = \quad \dfrac{\dfrac{\begin{array}{c}\Pi\\ \Gamma, B, C \vdash D\end{array}}{\Gamma, B \otimes C \vdash D}\;\otimes\mathrm{L} \qquad \begin{array}{c}\Pi'\\ D \vdash E\end{array}}{\Gamma, B \otimes C \vdash E}\;\text{e-cut}$$

In equation (6) above, 'e-cut' means that the corresponding cut is an essential one. The equations above may be read as defining associativity and commutativity of proofs in presence of proper axioms. We now define another equivalence relation $\mathcal{T}(N)$ on proofs to be the least equivalence relation generated by the equations defining $\mathcal{S}$ plus the following equation.

$$(7)\quad \dfrac{\dfrac{\overset{\Pi}{\Gamma\vdash A}\qquad \overset{\Pi'}{\Delta\vdash B}}{\Gamma,\Delta\vdash A\otimes B}\otimes\mathrm{R}\qquad \dfrac{\dfrac{\overset{\Pi''}{A\vdash C}\qquad \overset{\Pi'''}{B\vdash D}}{A,B\vdash C\otimes D}\otimes\mathrm{R}}{A\otimes B\vdash C\otimes D}\otimes\mathrm{L}}{\Gamma,\Delta\vdash C\otimes D}\mathit{Cut} \;=\; \dfrac{\dfrac{\overset{\Pi}{\Gamma\vdash A}\qquad \overset{\Pi''}{A\vdash C}}{\Gamma\vdash C}\mathit{Cut}\qquad \dfrac{\overset{\Pi'}{\Delta\vdash B}\qquad \overset{\Pi'''}{B\vdash D}}{\Gamma\vdash D}\mathit{Cut}}{\Gamma,\Delta\vdash C\otimes D}\otimes\mathrm{R}$$

This last rule corresponds to the functoriality of the tensor operation. Via translation, the relations $\mathcal{S}(N)$ and $\mathcal{T}(N)$ as we have just defined them correspond to the theories $\mathcal{S}[N]$ and $\mathcal{T}[N]$ as given in [5]. Their results there demonstrate the correspondence between these equivalence classes of proofs and the processes in [3]. We will therefore refer to equivalence classes of proofs in $\mathcal{S}(N)$ and $\mathcal{T}(N)$ as $\mathcal{S}$-processes and $\mathcal{T}$-processes respectively (dropping the N when it is understood as fixed).

3 Strongly normalizing cut reduction.

In this section we give a set of reduction rules for proofs and show that they are strongly normalizing. This will provide the desired algorithm for finding the normal representative of a $\mathcal{T}$-process.

Assume that a proof $\mathcal{P}$ ends with an inessential cut and has the following form:

$$\mathcal{P}:\quad \dfrac{\Gamma\vdash A\qquad A,\Delta\vdash B}{\Gamma,\Delta\vdash B}\mathit{Cut{:}A}$$

We will refer to the left and right sub-proofs as $\mathcal{P}'$ and $\mathcal{P}''$, respectively. We will divide these reduction rules in three classes—axiom, permutation, and logical—and give an illustrative transformation in each class.

1. *Axioms.* This case is applicable when at least one of the sub-proofs is an axiom. When $\mathcal{P}'$ is an axiom, we have the following transformation:

$$\dfrac{A\vdash A\qquad A,\Delta\vdash B}{A,\Delta\vdash B}\mathit{Cut{:}A}\qquad\Rightarrow\qquad A,\Delta\vdash B$$

2. *Permutation.* This rule is applied when at least one of the sub-proofs $\mathcal{P}'$ and $\mathcal{P}''$ terminates with a logical rule with the main formula being different from the cut formula A or with an essential cut. For the case when the last rule of $\mathcal{P}''$ is a $\otimes$R and cut formula A is in upper left sequent of the last rule of $\mathcal{P}''$, we have the following rewrite:

$$\dfrac{\Gamma\vdash A\qquad \dfrac{\Delta',A\vdash B\qquad \Delta''\vdash C}{\Delta',\Delta'',A\vdash B\otimes C}\otimes\mathrm{R}}{\Gamma,\Delta',\Delta''\vdash B\otimes C}\mathit{Cut{:}A}\quad\Rightarrow\quad \dfrac{\dfrac{\Gamma\vdash A\qquad \Delta',A\vdash B}{\Gamma,\Delta'\vdash B}\mathit{Cut{:}A}\qquad \Delta''\vdash C}{\Gamma,\Delta',\Delta''\vdash B\otimes C}\otimes\mathrm{R}$$

3. *Logical.* This is the case where the cut formula is the main formula of a logical rule in both $\mathcal{P}'$ and $\mathcal{P}''$ and is introduced only by this instance of the rule. The transformation in this case

depends on the outermost logical symbol of the cut formula and since we only have one logical connective, there is only one case to consider here.

$$\dfrac{\dfrac{\Gamma' \vdash A_1 \qquad \Gamma'' \vdash A_2}{\Gamma', \Gamma'' \vdash A_1 \otimes A_2}\otimes\mathrm{R} \qquad \dfrac{A_1, A_2, \Delta \vdash B}{A_1 \otimes A_2, \Delta \vdash B}\otimes\mathrm{L}}{\Gamma', \Gamma'', \Delta \vdash B}\ \mathit{Cut}{:}A_1 \otimes A_2$$

$$\Rightarrow \qquad \dfrac{\Gamma'' \vdash A_2 \qquad \dfrac{\Gamma' \vdash A_1 \qquad A_1, A_2, \Delta \vdash B}{\Gamma', A_2, \Delta \vdash B}\ \mathit{Cut}{:}A_1}{\Gamma'', \Gamma', \Delta \vdash B}\ \mathit{Cut}{:}A_2$$

The following property of the rewrite rules is not difficult to check:

Proposition 1 (Soundness of Rewrite Rules) *The above rewrite rules preserve the $\mathcal{T}$-equivalence of proofs.* ▮

We now show that the these reduction rules are strongly normalizing. We will need the following definition in the proof of the strong normalization theorem.

Definition 2 The *grade* g of a formula A is the number of occurrences of $\otimes$ contained in A. The grade of an inessential cut is the grade of its cut formula.

Thus, by the definition above, grade of an essential cut is 0.

Theorem 2 (Strong Normalization) *There is no infinite reduction sequence beginning with any proof $\mathcal{P}$.*

Proof: Let the *complexity* of a proof be a pair (a, b), where

- $a =$ sum of the grade g of cut formulas of all inessential cuts in the proof.
- $b =$ sum of the nodes above all inessential cuts (including the premisses and conclusion of the cut).

Clearly, a cut-reduced proof has complexity (0,0). We now show that each step of reduction on a proof reduces its complexity. Consider the three classes of the transformations above. It is easy to see that application of these transformations in each case to a proof reduces its complexity.

Axiom: Both a and b are reduced.

Permutation: b is reduced keeping a the same.

Logical: a is reduced.

Thus, all reduction sequences terminate. ▮

In the following section we show that the induced reduction relation on the equivalence classes modulo the relation $\mathcal{S}(N)$ on proofs enjoys the Church-Rosser property. We will then show that every $\mathcal{T}$- equivalence class has a unique normal process representative by showing that the equivalence defined by the reduction relation on $\mathcal{S}$-equivalence classes coincides with the relation $\mathcal{T}$. The $\mathcal{S}$-equivalence class of normal forms will then be the unique process representative of a $\mathcal{T}$-equivalence class.

4 Normal process representatives.

Let $\Rightarrow$ be the reduction relation on proofs and let $\Rightarrow_{\mathcal{S}}$ be the induced reduction relation on the equivalence classes of proofs modulo the equivalence relation $\mathcal{S}$. Our aim is to show that $\Rightarrow_{\mathcal{S}}$ is weakly Church-Rosser.

Theorem 3 (Weak Church-Rosser) *The relation $\Rightarrow_{\mathcal{S}}$ satisfies the weak diamond property, that is*

$$\begin{array}{ccccc} & & \Pi & & \\ & \swarrow_{\Rightarrow_{\mathcal{S}}} & & \searrow^{\Rightarrow_{\mathcal{S}}} & \\ \Pi' & & & & \Pi'' \\ & \searrow_{\Rightarrow^*_{\mathcal{S}}} & & \swarrow^{\Rightarrow^*_{\mathcal{S}}} & \\ & & \Sigma & & \end{array}$$

Proof: (Sketch) The result is proved by induction on the structure of the proof tree by analyzing the various cases that can arise. The case where the last rule used is a $\otimes$L or a $\otimes$R follows from the induction hypothesis. The same holds when the last rule is an essential cut or when the last rule is an inessential cut and a reduction cannot be applied to this last rule. The case where latter does not hold is the only interesting case. In this case, a permutation rule may be applied in two different ways. Analyzing different possibilities, we show the existence of Σ above by applying further reduction steps to Π' and Π''. For example, let Π be of the form

$$\Pi = \dfrac{\dfrac{\overset{\Pi_1}{\Gamma, B, C \vdash A}}{\Gamma, B \otimes C \vdash A}\otimes\mathrm{L} \qquad \dfrac{\overset{\Pi_2}{\Delta', A \vdash D} \qquad \overset{\Pi_3}{\Delta'' \vdash E}}{\Delta', \Delta'', A \vdash D \otimes E}\otimes\mathrm{R}}{\Gamma, B \otimes C, \Delta', \Delta'' \vdash D \otimes E}\,\mathit{Cut}$$

then let Π' and Π'' be obtained by application of the permutation rule for $\otimes$L and $\otimes$R, respectively. That is,

$$\Pi' = \dfrac{\dfrac{\overset{\Pi_1}{\Gamma, B, C \vdash A} \qquad \dfrac{\overset{\Pi_2}{\Delta', A \vdash D} \qquad \overset{\Pi_3}{\Delta'' \vdash E}}{\Delta', \Delta'', A \vdash D \otimes E}\otimes\mathrm{R}}{\Gamma, B, C, \Delta', \Delta'' \vdash D \otimes E}\,\mathit{Cut}}{\Gamma, B \otimes C, \Delta', \Delta'' \vdash D \otimes E}\otimes\mathrm{L}$$

and

$$\Pi'' = \dfrac{\dfrac{\dfrac{\overset{\Pi_1}{\Gamma, B, C \vdash A}}{\Gamma, B \otimes C \vdash A}\otimes\mathrm{L} \qquad \overset{\Pi_2}{\Delta', A \vdash D}}{\Gamma, B \otimes C, \Delta' \vdash D}\,\mathit{Cut} \qquad \overset{\Pi_3}{\Delta'' \vdash E}}{\Gamma, B \otimes C, \Delta', \Delta'' \vdash D \otimes E}\otimes\mathrm{R}$$

Now Π' can be reduced to

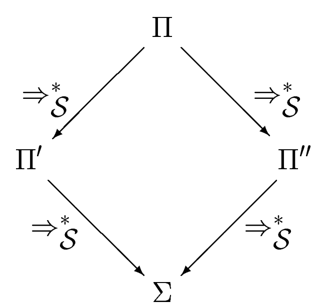
$$\Sigma' = \dfrac{\dfrac{\dfrac{\overset{\Pi_1}{\Gamma, B, C \vdash A} \qquad \overset{\Pi_2}{\Delta', A \vdash D}}{\Gamma, B, C, \Delta' \vdash D}\,\mathit{Cut} \qquad \overset{\Pi_3}{\Delta'' \vdash E}}{\Gamma, B, C, \Delta', \Delta'' \vdash D \otimes E}\,\otimes\mathrm{R}}{\Gamma, B \otimes C, \Delta', \Delta'' \vdash D \otimes E}\,\otimes\mathrm{L}$$

by another application of a permutation rule and similarly Π'' can be reduced to

$$\Sigma'' = \dfrac{\dfrac{\dfrac{\overset{\Pi_1}{\Gamma, B, C \vdash A} \qquad \overset{\Pi_2}{\Delta', A \vdash D}}{\Gamma, B, C, \Delta' \vdash D}\,\mathit{Cut}}{\Gamma, B \otimes C, \Delta' \vdash D}\,\otimes\mathrm{L} \qquad \overset{\Pi_3}{\Delta'' \vdash E}}{\Gamma, B \otimes C, \Delta', \Delta'' \vdash D \otimes E}\,\otimes\mathrm{R}$$

It is easy to see that $\Sigma' \mathcal{S} \Sigma''$ and thus the required existence of Σ has been shown. ▮

Since the reduction rules are strongly normalizing by Theorem 2, we use the Newman's Lemma (see [2] on page 58) which says that WCR and SN implies CR to conclude that $\Rightarrow^*_{\mathcal{S}}$ satisfies the following diamond property which will be used in the proof of Theorem 4 below.



Definition 3 A *normal process representative* is an $\mathcal{S}$-equivalence class of normal forms.

Theorem 4 (Unique Process Representative) *Let N be a net. In every $\mathcal{T}$-equivalence class, there is a unique normal process representative.*

Proof: Let Π and Π' be two $\mathcal{S}$-equivalent classes. Define $\Pi \Downarrow \Pi'$ if they both reduce to same normal form modulo the equivalence $\mathcal{S}$. To prove the theorem, we only have to show that $\Pi \Downarrow \Pi'$ iff $\Pi\ \mathcal{T}\Pi'$,*i.e.* the two equivalences coincide. Since the only if part follows from the soundness of the rewrite rules, we are only left with the if part. To prove the if part, we show that if two proofs are equivalent by virtue of equation (7) in section 2, then there is a sequence of reduction $\Rightarrow^*_{\mathcal{S}}$ from one to another. We thus rewrite the left-hand side of the equation (7) to a form which is $\mathcal{S}$-equivalent to the right-hand side of the equation.

$$\dfrac{\dfrac{\overset{\Pi}{\Gamma \vdash A} \qquad \overset{\Pi'}{\Delta \vdash B}}{\Gamma, \Delta \vdash A \otimes B}\,\otimes\mathrm{R} \qquad \dfrac{\dfrac{\overset{\Pi''}{A \vdash C} \qquad \overset{\Pi'''}{B \vdash D}}{A, B \vdash C \otimes D}\,\otimes\mathrm{R}}{A \otimes B \vdash C \otimes D}\,\otimes\mathrm{L}}{\Gamma, \Delta \vdash C \otimes D}\,\mathit{Cut}$$

$$\Rightarrow_{\mathcal{S}} \quad \dfrac{\overset{\Pi'}{\Delta \vdash B} \qquad \dfrac{\overset{\Pi}{\Gamma \vdash A} \qquad \dfrac{\overset{\Pi''}{A \vdash C} \qquad \overset{\Pi'''}{B \vdash D}}{A, B \vdash C \otimes D}\,\otimes\mathrm{R}}{\Gamma, B \vdash C \otimes D}\,\mathit{Cut}}{\Gamma, \Delta \vdash C \otimes D}\,\mathit{Cut}$$

$$\Rightarrow_{\mathcal{S}} \quad \dfrac{\overset{\Pi'}{\Delta \vdash B} \qquad \dfrac{\dfrac{\overset{\Pi}{\Gamma \vdash A} \qquad \overset{\Pi''}{A \vdash C}}{\Gamma \vdash C}\,\mathit{Cut} \qquad \overset{\Pi'''}{B \vdash D}}{\Gamma, B \vdash C \otimes D}\,\otimes\mathrm{R}}{\Gamma, \Delta \vdash C \otimes D}\,\mathit{Cut}$$

$$\Rightarrow_{\mathcal{S}} \quad \dfrac{\dfrac{\overset{\Pi}{\Gamma \vdash A} \qquad \overset{\Pi''}{A \vdash C}}{\Gamma \vdash C}\,\mathit{Cut} \qquad \dfrac{\overset{\Pi'}{\Delta \vdash B} \qquad \overset{\Pi'''}{B \vdash D}}{\Gamma \vdash D}\,\mathit{Cut}}{\Gamma, \Delta \vdash C \otimes D}\,\otimes\mathrm{R}$$

Thus we have shown the existence of a unique normal $\mathcal{S}$-class for each $\mathcal{T}$-class. ▮

In the next section we briefly sketch how rewrites on proofs can be viewed as (typed) rewrite on arrows in a suitable category. A detailed analysis of this relationship will be discussed elsewhere.

5 A note on arrows vs. proofs.

As we mentioned before, some authors have found it convenient to work with arrows (in strictly symmetric strict monoidal categories to be exact) rather than proofs as we have done in this paper. To some extent this is a matter of taste, but it can be illuminating to see things in both ways. For example, the rewrite rules for proofs in section 3 are translated respectively as the following rewrite on arrows.

1. $$i^{A\to A}\,;f^{A\to B} \Rightarrow f^{A\to B}$$

2. $$(f^{G\to A} \otimes i^{D\otimes E\to D\otimes E})\,;(g^{D\otimes A\to B} \otimes h^{E\to C}) \Rightarrow ((f^{G\to A} \otimes i^{D\to D})\,;g^{D\otimes A\to B}) \otimes h^{E\to C}$$

3. $$((f^{G\to A} \otimes g^{E\to C}) \otimes i^{D\to D})\,;h^{A\otimes C\otimes D\to B} \Rightarrow (g^{E\to C} \otimes i^{G\otimes D\to G\otimes D})\,;((f^{G\to A} \otimes i^{E\otimes D\to E\otimes D})\,;h^{A\otimes C\otimes D\to B})$$

These may at first sight seem rather unwieldy, but our proof-theoretic results show that they will work. Moreover, we found that working with proofs helped us to get the right definition of normal form. Concerning the rewrite system, the proof of Theorem 4 suggests that if one is given associativity and commutativity of arrows for free, the following rewrite will work whenever the right-hand side is defined.

$$(f_1 \otimes f_2)\,;(g_1 \otimes g_2) \Rightarrow (f_1\,;g_1) \otimes (f_2\,;g_2)$$

In other words, this rewrite rule will give a unique "normal" $\mathcal{S}[N]$ arrow for each $\mathcal{T}[N]$ equivalent class of arrows of [5]. In the rewrite above, the left-hand side is always defined whenever the right-hand side is defined but not vice-versa. In particular, subject reduction fails drastically, so the rewrite system must maintain the types of the terms.

References

[1] A. Asperti, G. L. Ferrari, and R. Gorrieri. Implicative formulae in the "proofs as computations" analogy. In P. Hudak, editor, *Principles of Programming Languages*, pages ??–??, ACM, 1990.

[2] H. Barendregt. *The Lambda Calculus: Its syntax and Semantics.* Volume 103 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, revised edition, 1984.

[3] E. Best and R. Devillers. Sequential and concurrent behaviour in Petri net theory. *Theoretical Computer Science*, 55(1):87–136, November 1987.

[4] C. Brown. *Relating Petri nets to formulae of linear logic.* Technical Report ECS-LFCS-89-87, University of Edinburgh, 1989.

[5] P. Degano, J. Meseguer, and U. Montanari. Axiomatizing net computations and processes. In R. Parikh, editor, *Logic in Computer Science*, pages 175–185, IEEE, IEEE Computer Society, June 1989.

[6] U. Engberg and G. Winskel. On linear logic and Petri nets. Unpublished manuscript.

[7] J. Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[8] C. A. Gunter and V. Gehlot. Nets as tensor theories. (preliminary report). In G. De Michelis, editor, *Applications of Petri Nets*, pages 174–191, 1989. Also University of Pennsylvania, Logic and Computation Report Number 17.

[9] N. Martí-Oliet and J. Meseguer. *From Petri nets to linear logic.* Research Report SRI-CSL-89-4, SRI International, Menlo Park, March 1989.

[10] J. Meseguer and U. Montanari. *Petri Nets Are Monoids.* Research Report SRI-CSL-88-3, SRI International, Menlo Park, January 1988.