

Petri Nets as Models of Linear Logic

Uffe Engberg

Glynn Winskel

Computer Science Department
Aarhus University
Ny Munkegade
DK-8000 Aarhus C, Denmark

Abstract

The chief purpose of this paper is to appraise the feasibility of Girard’s linear logic as a specification language for parallel processes. To this end we propose an interpretation of linear logic in Petri nets, with respect to which we investigate the expressive power of the logic.

1 Introduction

Girard’s linear logic has sparked off a great deal of interest in how it might be useful in the theory of parallelism, not least because of Girard’s initial claims for it [Gir87]. Linear logic has been described as a “resource conscious” logic by Martí-Oliet and Meseguer [MOM89]; in its proofs occurrences of propositions cannot be used more than once or disappear unless they are explicitly created or used up by the rules of inference. People were not long in spotting a relationship with Petri nets where there are similar ideas. Places in a Petri net hold to certain nonnegative multiplicities forming a multiset of places, traditionally called a marking; as transitions occur, multiplicities of places are consumed and produced in accord with a dynamic behaviour of nets, formalised in the so-called “token game”. Independently, Gunter and Gehlot [GG89], Asperti [Asp87] and Brown [Bro89a] showed that places are like atomic propositions in linear logic and transitions like proof rules. In [GG89] and [Asp87], the fine grain of linear logic proofs for the \otimes -fragment of linear logic is related to the token game in Petri nets. Essentially it is shown how a proof of $A \vdash B$ in linear logic, where A and B are built from place names just with \otimes , corresponds to a play of the token game taking the marking corresponding to A to that corresponding to B .

Recent work of Martí-Oliet and Meseguer [MOM89] extends that of [GG89] and [Asp87]. It is known that certain kinds of “linear” categories are models for linear logic. The work [MOM89] essentially proceeds by letting Petri nets freely generate a linear category and then interpreting linear logic in that setting. They restrict attention to linear classical logic. One problem with their approach is its consequence that if a net satisfies an proposition of linear logic then so does any augmentation of the net, obtained by adding transitions arbitrarily. This considerably weakens the case for regarding linear logic as a specification logic with respect to *their* notion of satisfaction¹. Nor at this exploratory stage, when we are trying to understand what use linear logic might be on Petri nets, is it so clear, that a linear category with lots of proof terms as morphisms has any advantage over a more accessible partial order semantics, of the kind presented here.

¹No one would take seriously a program logic with the property that if a program satisfied an assertion then so did the program with GOTO statements inserted arbitrarily.

had arisen before in the form of quantales. Indeed Girard’s phase semantics in [Gir87] for linear logic uses free quantales. Abramsky and Vickers [AV88] approached quantales from a computer science viewpoint, the hope being that it would lead to a “linear process logic”. Yetter [Yet] and Rosenthal [Ros] looked at quantales and linear logic more from the perspective of pure mathematics—how to represent them and their relationship with other bits of mathematics.

This note points out a straightforward way in which a Petri net induces a quantale and so becomes a model for linear intuitionistic logic. The model is for all of linear intuitionistic logic. A prime feature is its accessibility; this is important with the new logic, where working out what you can and cannot say is tricky. That it generalises the work of [GG89] and [Asp87] is clear. The paper provides evidence that linear intuitionistic logic, with the right notion of satisfaction, can be a reasonably expressive specification logic for parallel processes. It also throws some light on the use of linear classical logic; via a construction, observed by Abramsky and Vickers [AV88], generalising that of Girard’s for phase semantics, a quantale can easily be turned into a model for linear classical logic, although unfortunately when taken to nets, the resulting semantics of linear classical logic is often trivial.

2 Linear Intuitionistic Logic

In the Gentzen sequent calculus for intuitionistic logic a sequent, $A_1, \dots, A_n \vdash A$, is written to mean that the formula A is deducible from the assumption formulas A_1, \dots, A_n (we shall use Γ as an abbreviation for a sequence of assumption formulas). The calculus has the two structural rules

$$(1) \quad \frac{\Gamma \vdash B}{\Gamma, A \vdash B} \text{ (thinning)} \quad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B} \text{ (contraction)}$$

for adding an assumption and removal of a duplicate. In the presence of these rules the following two right introduction rules for conjunction

$$(2) \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \quad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

become interderivable in the sense that the first rule can be derived from the second by thinning, and the second from the first by contraction.

Now, as remarked in [AV88], if a proof is to be seen as representing the *process* of proving, then these structural rules become far from self-evident. Indeed, in linear intuitionistic logic these rules are dropped and the rules of (2) are no longer interderivable. Without them, propositions cannot be introduced arbitrarily into a list of assumptions and nor can a duplication in the assumptions be removed. It is in this sense that linear logic is a “resource conscious” logic. As a consequence of dropping (1) the two rules, (2), become two separate right introduction rules for two fundamentally different forms of conjunction; they are written as \otimes and $\&$ respectively. These new connectives are part of Girard’s linear intuitionistic logic, which we shall now present.

2.1 Proof Rules for Linear Intuitionistic Logic

The connectives of linear intuitionistic logic are:

- \otimes tensor, with unit **1**, called one,
- $\&$ conjunction, with unit **T**, called true,
- \oplus disjunction, with unit **F**, called false.

[GL87, Laf88]:

Structural rules

$$\frac{}{A \vdash A} \text{ (identity)} \quad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \text{ (cut)} \quad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C} \text{ (exchange)}$$

Logical rules

$$\begin{array}{l} \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} (\vdash \otimes) \quad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} (\otimes \vdash) \quad \frac{\Gamma \vdash A}{\Gamma, \mathbf{1} \vdash A} \quad \frac{}{\vdash \mathbf{1}} \\ \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} (\vdash \&) \quad \frac{\Gamma, A \vdash C}{\Gamma, A \& B \vdash C} (l\&\vdash) \quad \frac{\Gamma, B \vdash C}{\Gamma, A \& B \vdash C} (r\&\vdash) \quad \frac{}{\Gamma \vdash \overline{\mathbf{1}}} \\ \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} (\vdash \oplus l) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} (\vdash \oplus r) \quad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C} (\oplus \vdash) \quad \frac{}{\Gamma, \overline{\mathbf{1}} \vdash A} \\ \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} (\vdash \multimap) \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C} (\multimap \vdash) \end{array}$$

As an example of a proof, we can derive the rule $\frac{\Gamma \vdash A \multimap B}{\Gamma, A \vdash B}$ from $(\multimap \vdash)$ by

$$\frac{\frac{\frac{}{A \vdash A} \quad \frac{}{B \vdash B}}{\Gamma \vdash A \multimap B} \quad \frac{}{A, A \multimap B \vdash B}}{\Gamma, A \vdash B}.$$

The absence of the rules for thinning and contraction is compensated, to some extent, by the addition of the logical operator “of course”. In [GL87, Laf88] this operator is presented in the alternative, but equivalent, formalism of categorical combinatory logic and reads as follows in the setting of the sequent calculus:

“Of course” rules

$$(3) \quad \frac{}{!A \vdash A} \quad \frac{}{!A \vdash \mathbf{1}} \quad \frac{}{!A \vdash !A \otimes !A}$$

and

$$(4) \quad \frac{B \vdash A \quad B \vdash \mathbf{1} \quad B \vdash B \otimes B}{B \vdash !A}$$

Given a proposition A , the assertion of $!A$ has the possibility of being instantiated by the proposition A , the unit $\mathbf{1}$ or $!A \otimes !A$, and thus of arbitrarily many assertions of $!A$.

How this operator compensates for the absence of the two structural rules can be seen from the (weaker) derived rules Girard originally presented in [Gir87]:

$$(5) \quad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} \quad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} \quad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}$$

and

$$(6) \quad \frac{! \Gamma \vdash B}{! \Gamma \vdash !B}$$

where $! \Gamma$ is a shorthand for $!A_1, \dots, !A_n$ where $\Gamma = A_1, \dots, A_n$.

Once a logical constant, $-$, denoting linear-absurdity is fixed on, linear negation (not to be mistaken for intuitionistic negation) is derivable:

$$A^\perp = A \multimap -$$

We have just seen the proof rules of linear intuitionistic logic. What are its models? As recognised by several people [AV88, Yet, Ros, Sam], quantales² provide an algebraic semantics for linear intuitionistic logic. Quantales are to linear intuitionistic logic as complete Heyting algebras are to intuitionistic logic. A quantale is a commutative monoid on a complete join semilattice. Spelled out:

Definition 3.1 A *quantale* \mathcal{Q} is a complete join semilattice (i.e. a partial order with an operation forming joins of arbitrary sets) together with an associative, commutative, binary operation \otimes and constant $\mathbf{1}$ such that

$$\begin{aligned} q \otimes \mathbf{1} &= q \\ q \otimes \bigvee P &= \bigvee \{q \otimes p \mid p \in P\} \end{aligned}$$

□

Entailment is interpreted as the order relation, \leq , on the underlying lattice of a quantale. The logical operation, \otimes , is interpreted by the corresponding binary operation in the quantale and the logical constant $\mathbf{1}$ is interpreted as $\mathbf{1}$ in the quantale. The disjunction, \oplus , of linear logic is understood as binary join and the conjunction, $\&$, as binary meet. The logical constants \top and \bot are interpreted as the top and bottom element respectively of the complete lattice. Linear implication is a derived operation:

$$p \multimap q \stackrel{\text{def}}{=} \bigvee \{r \mid r \otimes p \leq q\}$$

with respect to a quantale. The definition is analogous to that of implication on a complete Heyting algebra, but this time w.r.t. \otimes in place of \wedge . The definition of linear implication ensures the adjunction:

$$r \otimes p \leq q \quad \text{iff} \quad r \leq p \multimap q$$

With respect to a quantale, and interpretations of the atomic propositions as elements of a quantale, we can inductively associate a proposition A in linear logic with its denotation as a quantale element $\llbracket A \rrbracket$. An entailment

$$A_1, \dots, A_n \vdash A$$

holds in the quantale iff

$$\llbracket A_1 \rrbracket \otimes \dots \otimes \llbracket A_n \rrbracket \leq \llbracket A \rrbracket$$

The special case where $n = 0$ is allowed, in which case the situation amounts to

$$\vdash A \quad \text{iff} \quad \mathbf{1} \leq \llbracket A \rrbracket$$

It is a routine matter to check that each rule is sound with respect to this interpretation. For example the right and left introduction rules for disjunction, \oplus , and conjunction, $\&$, express that they are the join and meet with respect to entailment while the rules for linear implication express the adjunction which characterise it (with the help of the little proof given as an example in section 3). In this way it can be seen that, with respect to a quantale,

$$\Gamma \vdash A \quad \text{implies} \quad \Gamma \models A \quad \text{and} \quad \vdash A \quad \text{implies} \quad \models A.$$

We have so far ignored the treatment of $!A$. The rules of (3) for $!A$ are interderivable with the following single rule:

$$\frac{}{!A \vdash \mathbf{1} \& A \& (!A \otimes !A)}$$

²As originally defined, quantales need not be commutative and should satisfy the idempotency law $q \otimes q = q$. We shall take quantales to be commutative and relax the idempotency law.

$$(7) \quad x \leq 1 \ \& \ q \ \& \ (x \otimes x).$$

This will not in general characterise a unique value of the quantale; for instance taking x to be the bottom element of the lattice will always do. However from (4) it follows that any x satisfying (7) should be below $!q$, and hence $!q$ should be the greatest postfix point, and so fixed point, of

$$x \mapsto 1 \ \& \ q \ \& \ (x \otimes x)$$

in the complete lattice given together with the quantale. Such a solution ensures the soundness of the proof rules extended by those for $!A$.

Another operation is that of linear negation. Its semantics with respect to a quantale is determined by a choice for the denotation of $-$. Then A^\perp is understood as $A \multimap -$. In this abstract set-up, for a general quantale we cannot hope to say much more. However, in a moment, when considering quantales got from a Petri net, a special element will be proposed as being useful. We can observe, following Abramsky and Vickers [AV88], that from a choice for $-$ in a quantale, we can obtain a model of linear classical logic. The key feature of the classical version is that linear negation is an involution. With respect to an element $-$ of a quantale \mathcal{Q} , write q^\perp for $q \multimap -$. We cannot expect $q = q^{\perp\perp}$ to hold for a general quantale. However, the image of the map $(-)^{\perp\perp} : \mathcal{Q} \rightarrow \mathcal{Q}$ is a subquantale of \mathcal{Q} on which $(-)^{\perp}$ is involutive, so forming a model of linear classical logic.

4 Petri Nets

Petri nets are a model of processes (or systems) in terms of types of resources, represented by *places* which can hold to arbitrary nonnegative multiplicity, and how those resources are consumed or produced by actions, represented by *transitions*. They are described using the notation of multisets.

A *multiset* over a set P is a function, $m : P \rightarrow \mathbb{N}$, and the set of multisets over P . With addition, $+$, of multisets defined by $(m + m')(a) = m(a) + m'(a)$ for all $a \in P$, multisets over P form a (free) commutative monoid with $\underline{0}$ ($\forall a \in P. \underline{0}(a) = 0$), the empty multiset, as neutral element. An element a of P can be turned into the singleton multiset \underline{a} ; we write \underline{a} for the multiset given by: $\underline{a}(a) = 1$ and $\underline{a}(b) = 0$ when $a \neq b$. Observe that multisets are partially ordered by $m \leq m'$ iff $\forall a \in P. m(a) \leq m'(a)$. For $n \in \mathbb{N}$ and m a multiset, the scalar multiplication nm is defined by $(nm)(a) = n \cdot m(a)$.

We take a Petri net N to consist of $(P, T, \cdot(-), (-)\cdot)$, where P , a set of *places*, and T , a set of *transitions*, are accompanied by maps $\cdot(-), (-)\cdot$ on transitions T which for each $t \in T$ give a multiset of P , called the pre- and post (multi)set of t respectively. There will be none of the usual restrictions on the net, such as absence of isolated elements, and in particular transitions with empty pre sets and/or post sets will be allowed. And we are actually considering place/transition nets with unconstrained capacity.

A Petri net possesses a notion of state, intuitively corresponding to a distribution of resources, formalised in the definition of a marking. A *marking* of N will simply be a multiset over P . We will generally use \mathcal{M} to denote the set of markings of a net, understood from the context. Sometimes nets are associated with an initial marking m_0 . The behaviour of a net is expressed by saying how markings change as transitions occur (or fire). For markings, m, m' , and a transition $t \in T$, $m [t] m'$ stands for t fires from m to m' ; that is

$$m [t] m' \quad \text{iff} \quad \exists m'' \in \mathcal{M}. m = m'' + \cdot t \text{ and } t + m'' = m'$$

reachability relation, the reflexive and transitive relation given by:³

$$m \rightarrow m' \quad \text{iff} \quad \exists t_1, \dots, t_n \in T, m_1, \dots, m_n \in \mathcal{M}, n \geq 0. m [t_1 \rangle m_1 \cdots [t_n \rangle m_n = m'$$

The markings forwards reachable from m form the set $\uparrow(m) = \{m' \in \mathcal{M} \mid m \rightarrow m'\}$. Similarly, we shall use $\downarrow(m)$ to denote the set of markings which can reach m . We will generally call this set the downwards closure of m . It is defined by $\downarrow(m) = \{m' \in \mathcal{M} \mid m' \rightarrow m\}$. We shall also use this notation, extended pointwise, on sets of markings M , taking e.g. $\downarrow(M) = \{m' \mid \exists m \in M. m' \rightarrow m\}$.

Petri nets can be presented by using the well-known graphical notation, which will be used throughout this paper. Places are represented by circles, transitions as squares, and arcs of appropriate multiplicities used to indicate the pre and post sets. The formal definitions can then be brought to life in the so called “token game” where markings are visualised as consisting of a distribution of tokens over places; the number of tokens residing on a place expresses the multiplicity to which it holds according to the marking. The tokens are consumed and produced as transitions occur. A basic reference for Petri nets is [Rei85].

Since we are dealing with place/transition nets without capacities we have the following proposition which is crucial for the net semantics of linear logic in the sequel.

Proposition 4.1 For any $m, m', m'' \in \mathcal{M}$ we have:

$$m \rightarrow m' \quad \text{implies} \quad m + m'' \rightarrow m' + m''$$

The notions and definitions in the following will be relative to a given Petri net, N , over a fixed set of places P , giving rise to a set of markings \mathcal{M} and reachability relation \rightarrow . Only when necessary will definitions be subscripted to indicate the relevant net.

5 Quantales from Nets

From a given net N , and associated markings and reachability relation \mathcal{M}, \rightarrow a quantale, \mathcal{Q} , can be constructed. Its elements are to be the downwards closed subsets of markings; a subset M of \mathcal{M} is downwards closed iff it satisfies

$$m' \rightarrow m \text{ and } m \in M \quad \text{implies} \quad m' \in M.$$

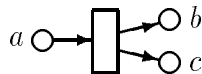
The set \mathcal{Q} is ordered by inclusion, \subseteq . With respect to inclusion, it is easy to see that union and intersection form the join and meet operations. The binary operation \otimes has as unit element: $\mathbf{1} = \downarrow(\underline{0})$. For $p, q \in \mathcal{Q}$ the binary operation \otimes is defined by:

$$p \otimes q = \downarrow(p + q), \text{ where } p + q = \{m_p + m_q \mid m_p \in p, m_q \in q\}$$

Thus

$$p \otimes q = \{m \mid \exists m_p \in p, m_q \in q. m \rightarrow m_p + m_q\}.$$

The downwards closure, \downarrow , in the definition of \otimes above is needed to make \otimes well-defined as can be seen from this net with $P = \{a, b, c\}$:



³Because we shall be chiefly concerned with the reachability relation between markings, we can get by with single steps in the firing relation, and ignore the variant in which transitions can fire concurrently.

That \mathcal{Q} is indeed a commutative quantale follows from the commutativity of $+$ and the fact that for $p \in \mathcal{Q}$, and $Q \subseteq \mathcal{Q}$:

$$p \otimes \bigcup Q = \downarrow(p + \bigcup Q) = \downarrow(\bigcup\{p + q \mid q \in Q\}) = \bigcup\{\downarrow(p + q) \mid q \in Q\} = \bigcup\{p \otimes q \mid q \in Q\}$$

Defining $p \multimap q = \bigcup\{r \mid r \otimes p \subseteq q\}$ we can simplify it to

$$p \multimap q = \{m \mid \forall m_p \in p. m + m_p \in q\}.$$

This is because

$$\begin{aligned} & m \in p \multimap q \\ \text{iff} & \exists r. m \in r, r \otimes p \subseteq q \\ \text{iff} & \downarrow(m) \otimes p \subseteq q, \quad \text{as } \downarrow(m) \in \mathcal{Q} \text{ and } m \in r \Rightarrow \downarrow(m) \subseteq r \\ \text{iff} & \forall m' \in \downarrow(m), m_p \in p. m' + m_p \in q, \quad \text{as } q \text{ is downwards closed} \\ \text{iff} & \forall m_p \in p. m + m_p \in q \end{aligned}$$

where the “if” of the last equivalence follows from the downwards closure of q and $m' \in \downarrow(m) \Rightarrow m' \rightarrow m \Rightarrow m' + m_p \rightarrow m + m_p \Rightarrow m' + m_p \in \downarrow(m + m_p)$.

6 Interpretation

In order to fix an interpretation of linear logic in a net we just need an interpretation of the atomic propositions as elements of the associated quantale. For simplicity we consider a linear logic language where the atomic propositions correspond to the places of the net. I.e. formulas are given by:

$$\begin{aligned} A ::= & \mathbf{T} \mid \mathbf{F} \mid \mathbf{1} && \text{constants} \\ & \mid a && \text{propositional letters where } a \text{ is a place} \\ & \mid A \otimes A \mid A \multimap A && \text{multiplicative connectives} \\ & \mid A \& A \mid A \oplus A && \text{additive connectives} \end{aligned}$$

We make the choice of interpreting an atomic proposition a as the downwards closure of the associated marking \underline{a} . This choice is consistent with the following intuitive understanding: the denotation of an assertion is to be thought of as the set of requirements sufficient to establish it. This reading will be discussed further shortly, after presenting the semantics.

With respect to a net N , linear logic formulas are interpreted as follows:

$$\begin{aligned} \llbracket \mathbf{T} \rrbracket_N &= \mathcal{M} \\ \llbracket \mathbf{F} \rrbracket_N &= \emptyset \\ \llbracket \mathbf{1} \rrbracket_N &= \{m \mid m \rightarrow \underline{\mathbf{0}}\} \\ \llbracket a \rrbracket_N &= \{m \mid m \rightarrow \underline{a}\} \\ \llbracket A \otimes B \rrbracket_N &= \{m \mid \exists m_A \in \llbracket A \rrbracket_N, m_B \in \llbracket B \rrbracket_N. m \rightarrow m_A + m_B\} \\ \llbracket A \multimap B \rrbracket_N &= \{m \mid \forall m_A \in \llbracket A \rrbracket_N. m + m_A \in \llbracket B \rrbracket_N\} \\ \llbracket A \& B \rrbracket_N &= \llbracket A \rrbracket_N \cap \llbracket B \rrbracket_N \\ \llbracket A \oplus B \rrbracket_N &= \llbracket A \rrbracket_N \cup \llbracket B \rrbracket_N \end{aligned}$$

Because of the interpretation of $\mathbf{1}$ validity of an assertion A for the given net, N , can be expressed by:

$$\models_N A \quad \text{iff} \quad \underline{\mathbf{0}} \in \llbracket A \rrbracket_N$$

$$A \models_N B \quad \text{iff} \quad \llbracket A \rrbracket_N \subseteq \llbracket B \rrbracket_N$$

Because of the interpretation of linear implication, this is equivalent to

$$\models_N A \multimap B.$$

Usually we shall leave out the subscript of N when the net in consideration is clear from the context.

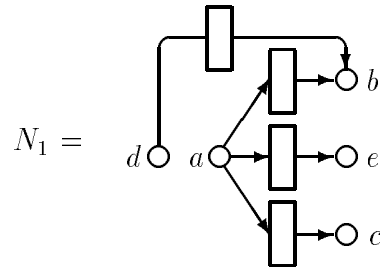
So we see that with respect to a Petri net, an assertion A is denoted by a set of markings $\llbracket A \rrbracket$. As we have discussed, a marking of net can be viewed as a distribution of resources. When $m \in \llbracket A \rrbracket$ we can think of the marking m as a distribution of resources sufficient to establish A according to the net; in this sense the marking m is one of the (in general many) requirements sufficient to establish A . The meaning of an assertion A is specified by saying what requirements are sufficient to establish it—this is the content of the denotation $\llbracket A \rrbracket$. Accordingly, a net satisfies an assertion A when $\underline{0} \in \llbracket A \rrbracket$, expressing that A can be established with no resources.

This reading squares with the fact that assertions denote subsets of markings which are downwards closed with respect to the reachability relation of the net; if $m \in \llbracket A \rrbracket$, so m is a requirement sufficient to establish A , and $m' \rightarrow m$ so we can obtain m for m' , then so also is m' a sufficient requirement of A . Casting an eye over the definition of the semantics of assertions we can read, for example, the definition of $\llbracket a \rrbracket$, for an atom a , as expressing that a sufficient requirement of a is any marking from which the marking \underline{a} can be reached according to the net. Similarly, the sufficient requirements of $A \& B$ are precisely those which are sufficient requirements of both A and of B . An element of $\llbracket A \multimap B \rrbracket$ can be seen as what is required, in addition to any requirement of A , in order to establish B . There are similar restatements of the semantics for the other connectives as well.

This understanding should be born in mind when considering the examples that follow, where we shall make use of the fact that \otimes , $\&$ and \oplus are associative and assume the precedence: $\multimap < \&$, $\oplus < \otimes$.

Examples

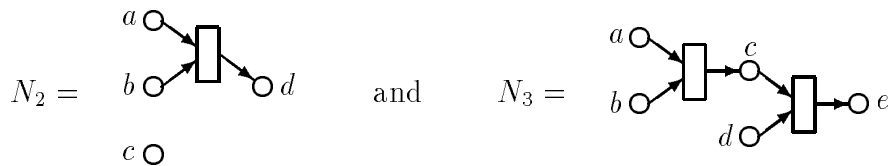
Consider the net:



Here we have $\llbracket b \rrbracket = \{\underline{d}, \underline{a}, \underline{b}\}, \dots, \llbracket c \rrbracket = \{\underline{a}, \underline{c}\}$ and $\llbracket b \otimes c \rrbracket = \{\underline{d} + \underline{a}, \underline{d} + \underline{c}, \underline{a} + \underline{a}, \underline{a} + \underline{c}, \underline{b} + \underline{a}, \underline{b} + \underline{c}\}$, so consequently

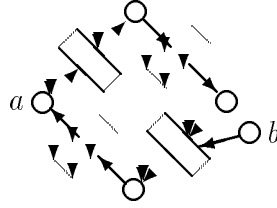
$$\begin{aligned} \llbracket d \otimes a \rrbracket &\subseteq \llbracket b \otimes c \rrbracket \text{ or equivalently } \models d \otimes a \multimap b \otimes c \\ &\models a \otimes a \multimap b \otimes c \\ &\models a \multimap b \oplus c, \models d \oplus a \multimap b, \models d \multimap b \oplus c \\ &\models b \& c \multimap e, \models a \multimap b \& c \end{aligned}$$

The most difficult connective to comprehend is probably linear implication so we give a few more examples here. For the nets:



$$\begin{aligned} \llbracket c \multimap d \rrbracket_{N_2} &= \emptyset \text{ and } \models_{N_2} (c \multimap d) \multimap \mathbf{F} \\ \llbracket b \multimap d \rrbracket_{N_2} &= \{\underline{a}\} \text{ and } \models_{N_2} (b \multimap d) \multimap a \\ \llbracket a \multimap e \rrbracket_{N_3} &= \{\underline{b} + \underline{d}\} \end{aligned}$$

A more peculiar example of linear implication is given by the following net:



In this net $\llbracket a \multimap a \rrbracket = \{n\underline{b} \mid n \in \mathbb{N}\}$.

7 Expressing Properties of Nets

In the following we shall use the abbreviation $a^n = \overbrace{a \otimes \cdots \otimes a}^n$, for $n \geq 0$, with the special case $a^0 = \mathbf{1}$. Identifying a nonempty marking, m , with the formula:

$$\bigotimes_{a \in P, m(a) \neq 0} a^{m(a)}$$

and the empty marking, $\underline{0}$, with the formula $\mathbf{1}$ we can express that one marking, m' , is reachable from another m :

Proposition 7.1 For any markings m and m' :

$$m \rightarrow m' \quad \text{iff} \quad \models m \multimap m'$$

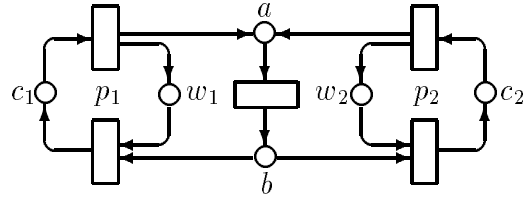
Proof Simply note: $m \rightarrow m'$ iff $\downarrow(m) \subseteq \downarrow(m')$ iff $\llbracket m \rrbracket \subseteq \llbracket m' \rrbracket$ iff $\models m \multimap m'$. □

Before reading the list of sample properties below observe: $\llbracket m \otimes \mathbf{T} \rrbracket = \downarrow\{m' \mid m' \geq m\}$.

Sample Properties

- From the initial marking, m_0 , it is possible to reach a marking where a is marked: $\models m_0 \multimap a \otimes \mathbf{T}$
- From the initial marking it is possible to reach a marking where a place, a , of $S \subseteq P$ is marked: $\models m_0 \multimap (\oplus_{a \in S} a) \otimes \mathbf{T}$
- Once a is marked it is possible to reach a marking where b is marked: $\models a \otimes \mathbf{T} \multimap b \otimes \mathbf{T}$
- Once a is marked either b or c can become marked (but we do not know which): $\models a \otimes \mathbf{T} \multimap (b \oplus c) \otimes \mathbf{T}$
- Once a is marked both b and c can become marked (but not necessarily simultaneously): $\models a \otimes \mathbf{T} \multimap (b \& c) \otimes \mathbf{T}$

Consider the net:



where the marking of the place w_1 indicates that the first process, p_1 , is working outside its critical region, c_1 , and similarly for the other process, p_2 . The resource corresponding to b is used to ensure mutual exclusion of the critical regions and after a process has been in its critical region it returns a resource, a , which then is prepared (transformed into b) for the next turn. The initial marking, m_0 , will be $m_0 = \underline{b} + \underline{w_1} + \underline{w_2}$. We can now express that e.g. p_1 can enter its critical region (from the initial marking) by: $\models m_0 \multimap c_1 \otimes \top$. However this does not ensure that no undesired tokens are present, so it is better to express it: $\models m_0 \multimap c_1 \otimes w_2$. If the system is in a “working state” then both processes have the possibility of entering their critical section: $\models w_1 \otimes (a \oplus b) \otimes w_2 \multimap c_1 \otimes w_2 \& w_1 \otimes c_2$. The property, that when p_1 is in its critical section and p_2 is working it is possible that p_2 can later come into its critical section with p_1 working, is expressed by: $\models c_1 \otimes w_2 \multimap w_1 \otimes c_2$. Similar other “positive” properties can be expressed. Shortly we shall see how to express the “negative” property that both processes cannot be in their critical regions at the same time.

8 The “Of Course” Operator

From the discussion in section 3 it immediately follows how to interpret an “of course” formula:

Definition 8.1 Given a formula A , the interpretation of “of course” A , is

$$\llbracket !A \rrbracket = \bigcup \{q \in \mathcal{Q} \mid q \text{ is a postfix point of } f_A\}$$

where $f_A : \mathcal{Q} \rightarrow \mathcal{Q}$ is the function given by:

$$(8) \quad x \mapsto \mathbf{1} \& \llbracket A \rrbracket \& (x \otimes x)$$

□

In order to gain a better understanding of the “of course” operator we give the following characterisation of a postfix point.

Proposition 8.2 $q \in \mathcal{Q}$ is a post fixed point of (8) iff there exists an $M \subseteq \mathcal{M}$ such that

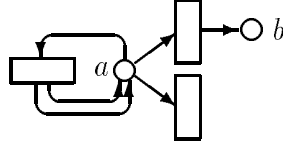
- i) $q = \downarrow(M)$
- ii) $M \subseteq \llbracket A \rrbracket$
- iii) $\forall m \in M. m \rightarrow \underline{0}$
- iv) $\forall m \in M \exists m', m'' \in M. m \rightarrow m' + m''$

if: $\downarrow(M) = q$ a postfix point of (8) means $q \subseteq \mathbf{1} \cap \llbracket A \rrbracket \cap (q \otimes q)$. From iii) follows $q = \downarrow(M) \subseteq \underline{\mathbf{0}} = \mathbf{1}$ and from ii) $q = \downarrow(M) \subseteq \downarrow \llbracket A \rrbracket = \llbracket A \rrbracket$ as $\llbracket A \rrbracket$ is downwards closed. To see $q \subseteq q \otimes q$ assume $m \in q$. Then $\exists m''' \in M. m \rightarrow m'''$ and so by iv) $\exists m', m'' \in M. m''' \rightarrow m' + m''$. Transitivity of \rightarrow then gives $m \in \downarrow(M + M) = M \otimes M \subseteq \downarrow(M) \otimes \downarrow(M) = q \otimes q$.

only if: Take $M = q$ and we have $\downarrow(M) = M$. Properties i) – iii) follows readily as does iv) by $q \subseteq q \otimes q$ and the downwards closure of M . \square

In the light of this proposition, we can see a requirement of $!A$ as any requirement of A which can both vanish and duplicate into any number of additional requirements of $!A$.

Example 8.3 For the net



we get: $\llbracket b \rrbracket = \{b\} \cup \{n\underline{a} \mid n > 0\}$ which properly contains $\llbracket !b \rrbracket = \{n\underline{a} \mid n > 0\}$.

9 Linear Negation

We have discovered that in a net one useful interpretation of the extra logical constant $-$, for linear-absurdity, is:

$$\llbracket - \rrbracket = \{m \mid \underline{\mathbf{0}} \not\rightarrow m\}$$

That is $-$ denotes exactly the set of markings which cannot be reached from the empty marking. For $\llbracket - \rrbracket$ to be well-defined, we need that it is downwards-closed: To see this, suppose $m' \rightarrow m$ for $m \in \llbracket - \rrbracket$. Assume $m' \notin \llbracket - \rrbracket$. Then $\underline{\mathbf{0}} \rightarrow m'$, whence from $m' \rightarrow m$ and transitivity of \rightarrow we conclude $\underline{\mathbf{0}} \rightarrow m$ —contradicting $m \in \llbracket - \rrbracket$.

Since linear negation can be expressed in terms of $-$ and \multimap (by $A^\perp = A \multimap -$), we get:

$$\llbracket A^\perp \rrbracket = \{m \mid \forall m_A \in \llbracket A \rrbracket. \underline{\mathbf{0}} \not\rightarrow m_A + m\}$$

The interesting consequence of this particular choice for $-$ is, that whatever property we could state before in terms of validity of a formula A can now be stated negatively as $\models A^\perp$. Formally:

Proposition 9.1 $\models A^\perp$ iff $\not\models A$

Proof Immediate from: $\not\models A^\perp$ iff $\not\models A \multimap -$ iff $\llbracket A \rrbracket \not\subseteq \llbracket - \rrbracket$ iff $\llbracket A \rrbracket \not\subseteq \{m \mid \underline{\mathbf{0}} \not\rightarrow m\}$ iff $\exists m_A \in \llbracket A \rrbracket. \underline{\mathbf{0}} \rightarrow m_A$ iff $\underline{\mathbf{0}} \in \llbracket A \rrbracket$ iff $\models A$, where the second last equivalence follows from the downwards closure of $\llbracket A \rrbracket$. \square

Combining this proposition with proposition 7.1 we can express that a marking m' cannot be reached from another m :

Corollary 9.2 For markings m and m' : $m \not\rightarrow m'$ iff $\models (m \multimap m')^\perp$

We can now express that the processes, p_1 and p_2 , of example 7.2, cannot get into their critical regions at the same time. We might try $\models (m_0 \multimap c_1 \otimes c_2)^\perp$. This is not quite right however, since $\models (m_0 \multimap c_1 \otimes c_2)^\perp$ merely states that the two processes cannot be in their critical regions at the same time when no other tokens are present; the correct statement is $\models (m_0 \multimap c_1 \otimes c_2 \otimes \mathbf{T})^\perp$.

- For any marking obtained from m the load on the place a cannot exceed $n \in \mathbb{N}$, i.e. with $m = m_0$ this means that a is n -safe:
 $\models (m \multimap a^{n+1} \otimes \mathbb{T})^\perp$
 Suppose on the contrary $\exists m'. m \rightarrow m'$ and $m'(a) > n$; that is $m' \geq (n+1)\underline{a}$. Then m' belongs to $\llbracket a^{n+1} \otimes \mathbb{T} \rrbracket$ which is downwards closed and so $m \in \llbracket a^{n+1} \otimes \mathbb{T} \rrbracket$ wherefore $\models m \multimap a^{n+1} \otimes \mathbb{T}$ —contradicting $\models (m \multimap a^{n+1} \otimes \mathbb{T})^\perp$.
- safeness, i.e. $\forall m \in \uparrow(m_0) \forall a \in P. m(a) \leq 1$ (N is 1-safe):
 $\models (m_0 \multimap (\oplus_{a \in Pa} \otimes a) \otimes \mathbb{T})^\perp$
 Again suppose on the contrary $\exists m' \exists a. m_0 \rightarrow m'$ and $m'(a) > 1$. Then $m' \in \llbracket (\oplus_{a \in Pa} \otimes a) \otimes \mathbb{T} \rrbracket$ and so $\models m_0 \multimap (\oplus_{a \in Pa} \otimes a) \otimes \mathbb{T}$ —a contradiction.

It follows from the remarks of section 3 that by denoting a proposition A by $\llbracket A \rrbracket^{\perp\perp}$ we obtain a Petri-net semantics of linear classical logic. In general it is hard to get an impression of $\llbracket A \rrbracket^{\perp\perp}$, but by slightly narrowing the class of nets we consider to those whose transitions always have a nonempty preset, i.e. $\underline{0} \neq \cdot t$ for all transitions t , we get:

$$(9) \quad \llbracket A \rrbracket^{\perp\perp} = \begin{cases} \llbracket \mathbb{T} \rrbracket & \text{if } \models A \\ \llbracket \mathbb{F} \rrbracket & \text{otherwise} \end{cases}$$

To see this notice first for each net we have $\llbracket - \rrbracket = \{m \mid m \neq \underline{0}\} = \mathcal{M} \setminus \{\underline{0}\}$ and so

$$\begin{aligned} q^\perp &= \{m \mid \forall m_q \in q. m + m_q \neq \underline{0}\} = \mathcal{M} \setminus \{m \mid \exists m_q \in q. m + m_q = \underline{0}\} \\ &= \begin{cases} \mathcal{M} \setminus \{\underline{0}\} & \text{if } \underline{0} \in q \\ \mathcal{M} & \text{otherwise} \end{cases} \end{aligned}$$

from which (9) follows. It also follows that the image of the map $(-)^{\perp\perp}$ forms a 2-element quantale where the logical operations act as the usual operations of propositional logic, and \otimes acts like $\&$. In this quantale all the propositional letters would have the same interpretation as \mathbb{F} and $-$. Such a classical interpretation seems of limited interest.

10 Recursion and Quantification

Though we now have the possibility to use linear negation to express properties of a net it is hard to express properties such as:

From the initial marking it is possible to reach a marking where just a is marked (to arbitrary multiplicity)

One way to increase the power of the logic language is to extend it with the possibility of writing recursive formulas using least (μ) and greatest (ν) fixpoint operators on assertions.

With their aid we can define propositions a^+ and a^* . Abbreviating $\mu x. a \oplus (a \otimes x)$ by a^+ and $a^0 \oplus a^+ = \mathbf{1} \oplus a^+$ by a^* we have:

$$\begin{aligned} m \in \llbracket a^+ \rrbracket & \quad \text{iff} \quad \exists n > 0. m \rightarrow n\underline{a} \\ m \in \llbracket a^* \rrbracket & \quad \text{iff} \quad \exists n \geq 0. m \rightarrow n\underline{a} \end{aligned}$$

Clearly the “of course” operator can be expressed as a maximum fixed point: $!A = \nu x. A \& \mathbf{1} \& x \otimes x$.

- From the initial marking it is possible to reach a marking where just a is marked: $\models m_0 \multimap a^+$
- From the marking m it is possible to reach a marking with a marked and b unmarked: $\models m \multimap a^+ \otimes (\otimes_{c \in P \setminus \{a,b\}} c^*)$
- The places of $S \subseteq P$ can always be emptied of tokens: $\models \top \multimap \otimes_{a \in P \setminus S} a^*$

Another possible extension is to add second order quantifiers with variables ranging over elements of the quantale, or quantifiers with variables over markings, for example; they can be understood as special kinds of meet and join operations. Unfortunately we don't have ready examples of their use.

11 Equivalences on Nets

Linear logic induces an equivalence relation between nets having the same set of places. With respect to a set of places, we build up the propositions of a linear logic with the places as atoms. If A is such a proposition whose atoms are amongst the places of a net N we say N satisfies A , written $\models_N A$, iff A is valid with respect to the semantics of section 6. We say two nets N_0, N_1 with the same places are equivalent if they make the same propositions valid, i.e. we define

$$N_0 \sim N_1 \quad \text{iff} \quad (\models_{N_0} A \text{ iff } \models_{N_1} A, \text{ for all propositions } A).$$

A less refined equivalence can be got by taking two nets to be equivalent iff they give rise to the isomorphic quantales. The latter notion has the advantage of not depending on the nets having the same set of places. We shall concentrate on the former equivalence as it implies the latter and suffices to point out some of the idiosyncracies of linear logic when interpreted on nets. The equivalence has a direct bearing on the suitability of linear logic as a specification language for Petri nets. If it is wished to distinguish between two nets which are equivalent, there is no hope of specifying one and not the other within pure linear logic.

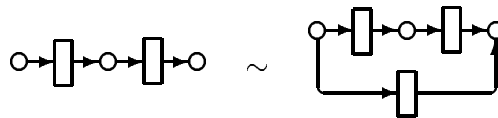
Observe that because our semantics of linear logic is built on the relation \multimap , any two nets, over a common set of places, which induce the same relation \multimap are bound to be equivalent. This means that whenever $m \multimap m'$, for a net N , we can adjoin a new transition t with the pre places m and post places m' , and still get a net equivalent to N . So, given any net N , we can add “identity” transitions at any multiset of places, “compositions” and “synchronizations” of transitions as new transitions and obtain a net equivalent to N .

Examples

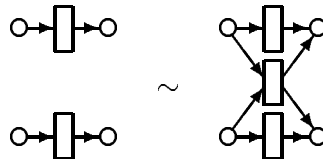
Adding “identity” transitions:



Adding “composition” transitions:



Adding “synchronization” transitions:



Also note that these two nets are equivalent:



However, we can distinguish between different directions of looping if another place is present to act like a loop counter:

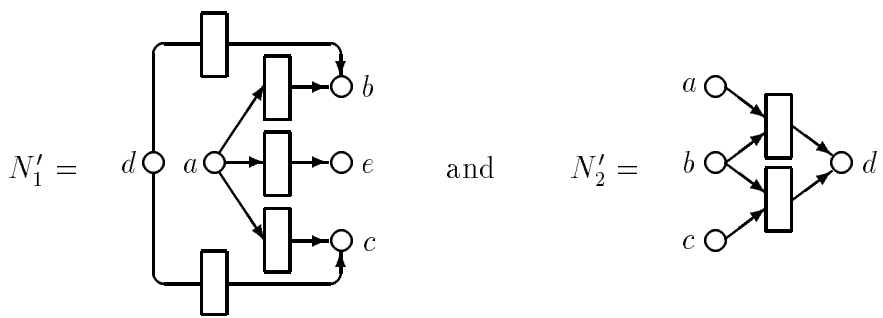


These nets are not equivalent.

Notice that whether or not the equivalence induced on nets is reasonable, if two nets are equivalent in our sense then so are they too with respect to the satisfaction relation of [MOM89]. This is because their construction of a linear category from a net factors through the construction of what they call a Petri-category. The Petri-category made from a net has objects the net's markings and has a morphism $f : m \rightarrow m'$ iff $m \rightarrow m'$ in our sense. A difference with the approach of [MOM89] can be seen by considering a preorder induced by the logic rather than just equivalence. For nets with the same set of places, define

$$N_0 \preceq N_1 \quad \text{iff} \quad (\models_{N_0} A \text{ implies } \models_{N_1} A, \text{ for all propositions } A).$$

The preorder induced by our definition of satisfaction will differ from theirs. In [MOM89], if a net N satisfies a formula A , in their sense, then any net obtained from N by adding an arbitrary number of new transitions will also satisfy A . This is not the case for our definition of satisfaction, as can be seen by adding transitions to the nets N_1 and N_2 in section 6: Take



For these nets we have $\not\models_{N'_1} b \& c \multimap e$ and $\not\models_{N'_2} (b \multimap d) \multimap a$ whereas $\models_{N_1} b \& c \multimap e$ and $\models_{N_2} (b \multimap d) \multimap a$.

12 Conclusion

From the examples we have given, we conclude that Girard's linear intuitionistic logic is expressive enough to be considered seriously as a specification logic for parallel processes. This is most compelling when linear-absurdity is present in the language of assertions to facilitate the expression of negative properties, and perhaps further extensions to the language, like recursion and quantification, are used. On the other hand, it must be confessed that linear logic is not an easy language to understand and express properties in.

Of course, our judgements are very much bound up with our choice of how to interpret the logic on Petri nets, and indeed with the implicit assumption that Petri nets are a good general model of parallel processes. To justify the latter we can cite the wealth of literature using Petri nets in the formal and informal analysis of parallel computation. As for our particular way of interpreting the logic on nets, it's harder to be definitive. Certainly some alternatives like [MOM89] suffer much greater problems from the point of view of saying interesting things about nets in the terms of

ours; for example, in [Bro89b] Carolyn Brown proposes using the upwards-closed sets with respect to reachability as the elements of the quantale (our reading of the elements of a denotation as “sufficient requirements” has then to be replaced by something else). In her thesis work she is following up the consequences of this twist and other relationships between nets and linear logic.

At present we lack a completeness theorem for linear logic interpreted in Petri nets; certainly additional rules, like the distributivity of $\&$ and \oplus , will be necessary. This problem and that of obtaining a compositional proof system for linear logic on Petri nets (with suitable combinators) are the subject of present study.

References

- [Asp87] Andrea Asperti. A Logic for Concurrency. manuscript, November 1987.
- [AV88] Samson Abramsky and Steve Vickers. Linear Process Logic. Notes by Steve Vickers, 1988.
- [Bro89a] Carolyn Brown. Relating Petri Nets to Formulae of Linear Logic. Technical Report ECS LFCS 89-87, University of Edinburgh, 1989.
- [Bro89b] Carolyn Brown. Petri Nets as Quantales. Technical Report ECS LFCS 89-96, University of Edinburgh, 1989.
- [GG89] Carl Gunter and Vijay Gehlot. A Proof-Theoretic Operational Semantics for True Concurrency. Preliminary Report, 1989.
- [Gir87] Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50(1):1–102, 1987.
- [GL87] Jean-Yves Girard and Yves Lafont. Linear Logic and Lazy Computation. In *Proc. TAPSOFT 87 (Pisa), vol. 2*, pages 52–66. Springer-Verlag (LNCS 250), 1987.
- [Laf88] Yves Lafont. The Linear Abstract Machine. *Theoretical Computer Science*, 59:157–180, 1988.
- [MOM89] Narciso Martí-Oliet and José Meseguer. From Petri Nets to Linear Logic. In *Category Theory and Computer Science, Manchester, UK*. Springer-Verlag (LNCS 389), 1989.
- [Rei85] Wolfgang Reisig. *Petri Nets, An Introduction*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.
- [Ros] Kimmo I. Rosenthal. A Note on Girard Quantales. To appear in: Cah. de Top. et G. D.
- [Sam] G. Sambin. Manuscript reported to us by Per Martin-Löf.
- [Yet] D. Yetter. Quantales and Non-Commutative Linear Logic. (preprint).