A logic for Petri nets
Une logique pour les réseaux de Petri

F. Girault, B. Pradin-Chézalviel†, R. Valette

LAAS-CNRS, 7, Av. du Colonel Roche
F-31077 Toulouse Cedex, France
Tél : (+33) 5 61 33 64 09
† also at Université Paul Sabatier, Toulouse, France
e-mail {girault, chezalviel, robert}@laas.fr

ABSTRACT: This paper presents a translation from Petri nets to linear logic with the objective of enhancing the analysis techniques of Petri nets. The net markings are denoted by formulas generated by the connective ⊗; the transitions are denoted by implicatives formulas between marking formulas and the firing operation is expressed by a sequent (i.e. a reasoning scheme). Starting from this translation, a logical calculus on transition sequences is defined, which allows to compose transitions in a sequential way as well as in a concurrent one.

RESUME: Cet article propose une traduction des réseaux de Petri en logique linéaire, l’objectif étant d’améliorer les techniques d’analyse existantes de ces réseaux. Les marques d’un réseau sont traduites par des formules engendrées sur l’alphabet des places par le connecteur ⊗; les transitions sont traduites par des formules implicatives entre des formules de marques, et l’opération de franchissement s’exprime par un séquent (i.e. un schéma de raisonnement). Partant de cette traduction, on définit un calcul logique des séquences de transitions qui permet de composer aussi bien séquentiellement que parallèlement les transitions.

KEY WORDS: linear logic, Petri nets, sequence characterization

MOTS-CLES: logique linéaire, réseaux de Petri, caractérisation de séquences de transitions
1. Introduction

The Petri net theory supplies powerful analysis techniques - invariants, general properties like liveness, etc ... - in order to prove system correctness; these techniques are specially used for discrete event systems. When Petri net models are also used for diagnostic purpose, it is necessary to develop a logical reasoning scheme about their behavior. Classical logic is of no help for this purpose because it only deals with eternal truth: actually, when a classical logic proposition is proved, it remains true along the whole reasoning scheme. Indeed, we assign propositions to states of a system; for systems evolving from one state to another one it is necessarily inconsistent because, by definition of the state concept, these propositions are not eternal truths.

Some authors, [MZ 88, PM 89] tried to tackle this problem using classical logic, but they have considered non cyclic and not bounded Petri nets. However, physical systems have bounded variables and we have to consider their cyclic evolutions.

Linear logic, proposed by J.Y. Girard [Gi 87, La 88, Gi 90, Gi 95], implicitly deals with the state concept: propositions are ressources used during the reasoning scheme. Actually, when these propositions are hypotheses, they are consumed by the derivation process and they are produced when they are conclusions. The set of available propositions at a particular step of the reasoning scheme formalizes the reached state and expresses the possible state changes. These changes are stated as causality links between hypotheses and conclusions.

In this paper, we are going to see how linear logic significantly enriches analysis methods for Petri nets. More precisely, it allows, using an appropriate calculus, to generalize to transition sequences the pre and post functions usually defined for isolated transitions. The firing order of the different transitions involved in these sequences is taken into account. This approach does not require enumerating the marking graph and it can be done independently from the initial marking.

Some authors already tackled the relation between Linear logic and Petri nets, but using different approaches. So, [EW 90] associates a proposition A to the set of previous markings of this particular one where place A gets a token. This approach uses the "phase semantics" defined by Girard. A similar method is proposed in [Li 92] for high level Petri nets. Instead of this interpretation, that deals with causalities between all the reachable markings of a net, we prefer the second approach proposed in [Br 89, GG 89, MM 91]. It directly operates on the net structure, transition by transition, independently from the initial marking of this net.

Section 2 quickly explains linear logic notions used in this paper while section 3 sets out our translation from Petri net structure and marking to linear logic.
Section 4 shows how we can logically characterize transition sequences and sets of sequences. Using some examples, section 5, deals with sequential and concurrent transitions composition.

2. Linear logic principles

2.1 Linear logic connectives

In order to consider propositions as resources, Girard changed the whole connective set of classical logic by proposing a very new system [Gi 95]. Actually, the \textit{and} connective of classical logic cannot express the conjunction of two resources because of its monotonicity: formula $A \land A$ is equivalent to $A$. This property of the \textit{and} connective is well-founded when considering eternal truths, for example if $A$ means \textit{all men must die}. But it is no longer valid when considering propositions as availability of resources: if $A$ means \textit{I get 5 dollars}, formula $A \land A$ has to express \textit{I get 10 dollars}. Girard defined the $\otimes$ connective whose properties are the same as the classical \textit{and} (commutativity, associativity) except monotonicity. Along this paper $A \otimes A$ will be denoted as $2A$.

Classical implication is modified in a same way because it does not expresses causality in a satisfactory manner. The linear implication $\multimap$ takes the place of the classical one and allows to express \textit{consuming} and \textit{producing} concepts. The formula $A \multimap B$ means that proposition $B$ may be produced by consuming the proposition $A$; then, the proposition $A$ will no longer be available.

Formally, linear logic is obtained from classical logic by deleting, in the sequent calculus, contraction and weakening rules since they are responsible for monotonicity. This leads to the splitting of classical \textit{and} connective into two new ones and so is for the \textit{or} one. The generated set of connectives can be decomposed into three families:

- The "multiplicative" ones:
  - $\otimes$ (called \textit{times}) that expresses conjunction of resources,
  - $\multimap$ (called \textit{implies}) that expresses causality between resource consuming and producing,
  - $\mathcal{N}$ (called \textit{par}) that is the dual connective of $\otimes$.

- The "additive" ones:
  - $\&$ (called \textit{with}) that expresses the external choice (i.e. controlled) between resources,
  - $\oplus$ (called \textit{plus}), the dual one of $\&$, that expresses an internal choice (i.e. not controlled) between resources.

- The "exponential" ones: ! (called \textit{of course}) and ? (called \textit{why not}).
These two unary connectives allow to introduce contraction and weakening capabilities to deal with non consumable resources but in a very controlled way.

The keystone of the linear logic system is linear negation, denoted $(\cdot)^\perp$ (called $\text{nil}$).

2.2 Linear negation

The nil connective needs some detailed explanations. Since a proposition may be used several times, it is impossible to interpret its linear negation as the fact that this proposition is not available; actually, what would mean the fact that it is not twice available? Linear negation does not express a duality truth/falsehood, like in classical logic, but rather a dual point of view producing/consuming. When speaking about the derivation procedure, this duality states that we are reverting the reasoning scheme direction: working with “positives” propositions (such as $A$) allows to look at their future evolutions (i.e the different ways of consuming them) while working with “negative” propositions (such as $A^\perp$) allows to look at their past evolutions (i.e. the different ways of producing them). Beyond involutivity $(A^\perp)^\perp \equiv A)$, syntactic properties of linear negation express some duality aspects between connectives with a form very similar to De Morgan laws. For example, connectives $\otimes$, $\exists$ and $\rightarrow$ are tied up by laws very close to those of classical logic:

\[(A \otimes B)^\perp \equiv A^\perp \exists B^\perp \quad \quad (\exists \forall B)^\perp \equiv A^\perp \otimes B^\perp \]
\[A \rightarrow B \equiv B^\perp \rightarrow A^\perp \quad \quad A \rightarrow B \equiv A^\perp \exists B\]

2.3 Linear logic sequents

Due to the length of this paper and the complexity of formulas, we do not present here the sequent calculus rules that formally define linear logic. The interested reader can find a presentation in Girard papers [Gi 87, Gi 95] and also in a lot of more applicative papers such as [Br 89, GG 89, MM 91, PV 93]. Nevertheless, we will use the sequent calculus formalism to state some linear logic derivations. Let us remind that a sequent is expressed by $\Gamma \vdash \Delta$ where $\Gamma$ and $\Delta$ are formulas lists (i.e. $\Gamma = A_1, A_2, \ldots$ and $\Delta = B_1, B_2, \ldots$). It expresses the fact that $\Delta$ is provable from $\Gamma$. A provable sequent, using the sequent calculus rules, corresponds to a valid derivation. All the sequents we will use are provable (but we will omit demonstrations) and we will only use sequents where $\Delta$ is reduced to a unique formula.
3. The Petri net Translation

3.1 Net description

The translation principle is quite direct: an atomic proposition is associated with each place of the net and it states the presence of one token in this place. If several tokens are present at the same time, the atom is factorized by the number of tokens in this place, as we previously proposed when presenting the connective $\otimes$. Then, any marking is denoted as a conjunctive formula (using the connective $\otimes$) involving the marked places, each one with the number of present tokens.

Each transition is translated by an implicational formula ($\rightarrow$ connective) since it expresses causality between two marking formulas. The left side of $\rightarrow$ states the minimal marking to fire this transition while the right side represents the reached marking after the firing from this minimal marking.

So, the structure of the Petri net presented in figure 1 is described by the following formulas:

$$t_1 : (2A \rightarrow B \otimes a) , \quad t_2 : (B \otimes a \rightarrow C).$$

Let us suppose that the initial marking $M_0$ of this net consists of one token in place $A$ and another one in place $C$, the (consumable) formula is:

$$M_0 : A \otimes C$$

It is interesting to note that this description remains valid whatever are the other transitions of the net. Adding or removing a transition will not modify the formulas of other transitions. If some place is added or removed, only the concerned transition formulas (i.e. those whose input or output include these places) will be changed.
3.2 *Marking evolution*

Starting from the Petri net description, a transition firing is expressed by a valid sequent. For example, for the net in figure 1, this sequent can be proved:

\[ 2A, (2A \rightarrow B \otimes \alpha) \vdash B \otimes \alpha \]

It states the marking change: starting from the marking \( M(2A) \), the firing of transition \( t_1 \) leads to marking \( M'(B \otimes \alpha) \). Let us note that marking \( M \) may be greater than the minimal marking necessary to fire \( t_1 \). In such a situation, the exceeding tokens remain present in the reached marking. For example, this sequent is valid:

\[ 2A \otimes \alpha \otimes C, (2A \rightarrow B \otimes \alpha) \vdash B \otimes 2\alpha \otimes C \]

It expresses the firing of \( t_1 \) in a different context. As a result of the very good adequation between linear logic and Petri nets it has to be noted that this sequent is no more provable if we omit any element either in the right or left side of this sequent.

These two formulas express what happens when \( t_1 \) is fired exactly once. If we need to consider marking evolution when \( t_1 \) is fired twice, we can prove that this sequent is valid:

\[ 5A \otimes \alpha, (2A \rightarrow B \otimes \alpha), (2A \rightarrow B \otimes \alpha) \vdash A \otimes 2B \otimes 3\alpha \]

So, with linear logic we can easily and strictly precise how many times a transition is fired.

3.3 *Sequence characterization*

Let us consider again the net in figure 1. Starting from formulas of transitions \( t_1 \) and \( t_2 \), we derive the following sequent:

\[ (2A \rightarrow B \otimes \alpha), (B \otimes \alpha \rightarrow C) \vdash (2A \rightarrow C) \]

It states a new formula which characterizes the transition sequence \( s = t_1t_2 \). As we can see, this formula has exactly the same form than the one denoting a transition: the antecedent of linear implication describes the minimal marking necessary to fire \( s \) and the right side gives the reached marking after firing this sequence from the minimal marking. This formula gives a more global description than the two transitions ones.

Because of this identical characterization, a sequence firing is expressed by a sequent very similar to the one expressing a transition firing. For example, the firing of sequence \( s \) from marking \( 2A \otimes \alpha \) is described by:

\[ 2A \otimes \alpha, (2A \rightarrow C) \vdash C \otimes \alpha \]
Let us remark that it can be compared to:

$$2A \otimes \alpha, (2A \rightarrow B \otimes \alpha), (B \otimes \alpha \rightarrow C) \vdash C \otimes \alpha$$

This sequent also expresses the sequential firing of $t_1$ and $t_2$ but transitions formulas explicitly appear.

4. Sequence calculus

Because of the direct translation of a Petri net into linear logic formulas, we derive, using sequent calculus, very attractive logical formulas for transition sequences. Putting together with this deduction method a formal calculus on transitions alphabet, we are going to define a more precise characterization method than the usual algebraic one (based on the characteristic vector $\bar{s}$ for any sequence $s$) that uses incidence matrix $C$ of the net.

4.1 Some notes on Petri net theory

Petri net analysis methods (cf. [BR 83]) using linear algebra are based on the characteristic equation:

$$\text{If } M \xrightarrow{s} M', \text{ Then } M' = M + C\bar{s}$$

where $M$ and $M'$ are markings while $s$ is a transition sequence and $C$ the incidence matrix of the net. But, this equation is only valid if the sequence $s$ is fireable from the marking $M$. The $C\bar{s}$ vector is not sufficient to state it, because it only describes the changes resulting from the transition firing. In some situations this condition is not a sufficient one to fire $s$.

The two main motives are:

1. the $\bar{s}$ vector does not take into account the order of the transitions in the sequence $s$,

2. elementary loops are not considered by the incidence matrix $C$.

Because of these two problems, some necessary conditions for the firing of sequence $s$ are not stated by $C\bar{s}$. Consider now the net in figure 2 and the sequence $s = t_1 t_2$. Marking changes for this sequence are $-1$ for place $A$ and $+1$ for place $C$. But the minimal marking to fire $s$ contains two more tokens: one in place $\beta$ and one in place $\gamma$. The bad characterization for $\gamma$ is not surprising because it belongs to an elementary loop. The bad characterization for $\beta$ is due to a rather different reason: $\beta$ does not belong to an elementary loop but to an internal one and $C\bar{s}$ is not able to deal with such situations.
Moreover it has to be noted that marking changes (using $C.\bar{s}$) are the same either for $s' = t_2t_1$ or for $s = t_1t_2$. But the actual minimal marking for firing $s'$ is: one token in places $A$, $B$, $\gamma$ and $\alpha$. Comparing this result with the actual result for sequence $s$, we see that the arc directions in an internal loop modify the firing condition. For the sequence $s'$, the place $\beta$ belongs to an internal loop but the considered token is first produced and then consumed (so it does not appear in the pre-conditions of the sequence firing) while it is the opposite for sequence $s$. The only way to state the necessary condition for firing a sequence is to expand pre-conditions functions $\text{Pre}(., t)$ of transitions $t$ involved in $s$ in a $\widehat{\text{Pre}}(., s)$ function. The efficiency of this function is poor because it is recursively defined and needs a lot of calculus steps:

- if $s = \lambda$ (empty seq.), then $\widehat{\text{Pre}}(., s) = 0$,
- else $s = s't$ and $\widehat{\text{Pre}}(., s) = \text{Max}\{\widehat{\text{Pre}}(., s'), \text{Pre}(., t) - C.\bar{s}\}$

4.2 Sequential composition of transitions

Let us now consider our logical approach. We have previously stated that a valid sequent was expressing the derivation of a sequence formula: on the left side are the formulas characterizing two transitions and on the right one the sequence formula. For example, the formula for sequence $s = t_1t_2$ in figure 2 is derived using this sequent:

$(A \otimes \gamma \otimes \beta \rightarrow B \otimes \gamma \otimes \alpha), (B \otimes \alpha \rightarrow C \otimes \beta) \vdash (A \otimes \gamma \otimes \beta \rightarrow C \otimes \gamma \otimes \beta)$

This derived sequence formula contains all conditions for firing $s$ when some are not expressed in $C.\bar{s}$. But linear logic is a commutative one: some exchange rules, included in the sequent calculus, specify that order inside premises and conclusions of the sequent does not matter. As a consequence, if sequent $A_1 \cdot A_2 \vdash B$ is valid, so is sequent $A_2 \cdot A_1 \vdash B$, and vice versa. While the derived sequence formula takes the firing order of transitions into account, this order does not explicitly appear in the sequent: we only see it when expliciting the
proof. But we need to easily distinguish the sequent formalizing the sequence $s = t_1 t_2$ from the one characterizing $s' = t_2 t_1$ because derived formulas are different. Here is the sequent for $s'$:

$$(A \otimes \gamma \otimes \beta \rightarrow B \otimes \gamma \otimes \alpha), \ (B \otimes \alpha \rightarrow C \otimes \beta) \vdash (B \otimes \alpha \otimes A \otimes \gamma \rightarrow B \otimes \alpha \otimes C \otimes \gamma)$$

And that’s why we define a formal calculus on the transition alphabet; this calculus explicitly states the firing order of a sequence that is only implicitly expressed by the sequent calculus.

First, let us consider the general rule to express the sequential composition of two transitions. It can be represented by this sequent:

$$(M_1 \rightarrow M_2 \otimes L), \ (M_3 \rightarrow K) \vdash (M_1 \otimes K \rightarrow M_2 \otimes L)$$

where meta-variables $M_1$, $M_2$, $M$, $K$ and $L$ stand for marking formulas. One more condition has to be verified: $K$ and $L$ may not have any common atomic proposition. In that way, if some pre-conditions of the second transition are produced by the first one, they will not appear in the pre-conditions of the sequence formula. So, $M$ is the set of all post-conditions of the first transition that also are pre-conditions of the second one.

Depending on transitions, two cases have to be considered:

1. $M$ is not equal to $I$, the neutral element of the connective $\otimes$: in this case, the compounded transitions, according to the sequence order, admit at least one intermediate place; they are related by some causality relation;

2. $M$ is equal to $I$: there is no causality relation according to the firing order.

Whatever the value of $M$, this sequent expresses how to derive a sequence formula, taking the firing order into account; the causality relation included in this ordered sequence is defined by $M$. With the aim of easily expliciting the firing order, transition formulas are prefixed by their name (i.e. $t$ or $t'$) and the conclusion formula, expressing the sequence, is prefixed by the concatenation of the two transition names.

The calculus rule is then defined as:

$$t : (M_1 \rightarrow M_2 \otimes L) \quad t' : (M_3 \rightarrow K) \quad t t' : (M_1 \otimes K \rightarrow M_2 \otimes L)$$

In order to use this rule, we only need to identify $M$, $K$ and $L$ markings of the two considered transitions.
For example, sequences $s = t_1 t_2$ and $s' = t_2 t_1$ of the net in figure 2 are:

$$
\begin{align*}
t_1 : (A \otimes \gamma \otimes \beta \to B \otimes \alpha \otimes \gamma) \\
t_2 : (B \otimes \alpha \to C \otimes \beta)
\end{align*}
$$

$$
\begin{align*}
t_1 t_2 : (A \otimes \gamma \otimes \beta \to C \otimes \beta \otimes \gamma) \\
t_2 t_1 : (B \otimes \alpha \otimes A \otimes \gamma \to B \otimes \alpha \otimes \gamma \otimes C)
\end{align*}
$$

Indeed, this calculus rule is powerful because sequence formulas are identical to transitions ones. It can be generalized to manage the composition of two sequences. In such a case, premises of the rule are labeled by words of the free monoid on the transitions alphabet. And so is the conclusion of this rule. So is stated an internal composition rule on $T^*$; it is written:

$$
\begin{align*}
s : (M_1 \multimap M \otimes L) \\
s' : (M \otimes K \multimap M_2)
\end{align*}
$$

$$
ss' : (M_1 \otimes K \multimap M_2 \otimes L)
$$

Conditions concerning meta-variables are identical to those for transitions. This rule can be iteratively applied to define sequences of any length. A sequence can so, be defined by composition of sub-sequences. Comparing this method to the matricial one for generalized functions $Pre$ and $Post$, we see that it is much more efficient because it only deals with places and transitions which are effectively involved in this sequence $s$. Moreover, and we have seen it is essential, the derived formula is much more precise than results obtained using $C$.$\ddagger$. As a consequence, formulas obtained with this rule can be used for deriving marking formulas using this sequent scheme:

$$
\begin{align*}
P \otimes Q, (Q \multimap R) & \vdash P \otimes R \\
M & \vdash s \\
M' & \vdash
\end{align*}
$$

To use it, we first have to compute the sequence $s$ and then identify $P, Q$ and $R$ markings. If preconditions for firing are satisfied by marking $M$ (i.e. $Q$ exists) marking $M'$ is automatically derived.

### 4.3 Transitions interleaving

Interleaving concerns transitions simultaneously fireable. First, let us deal with such two transitions $t$ and $t'$. We are going to identify the two different sequences $tt'$ and $t't$ using the sequential composition rule, previously defined.

$$
\begin{align*}
t : (M \otimes K_1 \multimap M' \otimes L_1) \\
t' : (M' \otimes K_2 \multimap M \otimes L_2)
\end{align*}
$$

$$
\begin{align*}
tt' : (M \otimes K_1 \otimes K_2 \multimap M \otimes L_1 \otimes L_2)
\end{align*}
$$
\[
\begin{align*}
t' : (M' \otimes K_2 \rightarrow M \otimes L_2) \quad t : (M \otimes K_1 \rightarrow M' \otimes L_1) \\
t't : (M' \otimes K_1 \otimes K_2 \rightarrow M' \otimes L_1 \otimes L_2)
\end{align*}
\]

Meta-variables \( M, M', K_1, L_1, K_2 \) and \( L_2 \) stand for input and output marking formulas of transitions \( t \) and \( t' \). Each sequence is identified by a different formula. These two sequences will have a unique characterization if \( M \) and \( M' \) are identical. This condition expresses the fact that the set of intermediate places between \( t \) and \( t' (M') \) must be strictly identical to \( M \), the set of intermediate places between \( t' \) and \( t \). In addition, as we use the sequential composition rule, usual conditions must be verified: \( K_1 \) and \( L_2 \) do not have any common atomic atom, and so are \( K_2 \) and \( L_1 \).

If \( M \) and \( M' \) are identical, it is possible to characterize the both sequences by the same formula. So, this formula specifies a set of sequences. This rule is:

\[
\begin{align*}
t : (M \otimes K_1 \rightarrow M \otimes L_1) \\
t' : (M \otimes K_2 \rightarrow M \otimes L_2)
\end{align*}
\]

\[\{tt', tt'\} : (M \otimes K_1 \otimes K_2 \rightarrow M \otimes L_1 \otimes L_2)\]

As when considering sequential composition, two situations can occur:

1. \( M \) is equivalent to \( 1 \), the neutral element of \( \otimes \), i.e. transitions \( t \) and \( t' \) have no common place and, so, are independent ones;

2. \( M \) is not equal to \( 1 \): interleaving is then dealing with transitions with some causality links; this causality is expressed by \( M \) that remains present in the resulting formula on the both sides of linear implication. Let us note that, in this case, these causality links are symmetric (\( M \) appears in the both sides of the two transitions formulas) and interleaving scheme can be applied. If \( M \) and \( M' \) are not identical sets, \( t \) and \( t' \) are tied up by non-symmetric causality links and the interleaving rule cannot be applied.

An important point has to be noticed: the derived formula has exactly the same form as the sequence one. So, it is possible to derive markings as previously done for sequences or transitions. But interleaving scheme is much more complex than sequential composition and the interleaving rule cannot be extended to sequences without some more conditions. This point needs a more complex rule and is not presented in this paper. Here, we only deal with transitions interleaving.

### 4.4 Concurrent composition of transitions

As in case of sequential composition, it is possible to consider concurrent composition of transitions and then to extend these results to sequences. The
derived sequent characterizes the concurrent firings of two transitions (i.e. the independent firings). We want to derive a unique formula for specifying a set of sequences: either firings are simultaneous, or not. In order to get a clear notation, we adopt this one:

\[
\begin{align*}
    t &: (M_1 \rightarrow M_2) \\
    t' &: (M_3 \rightarrow M_4) \\
    t \parallel t' &: (M_1 \otimes M_3 \rightarrow M_2 \oplus M_4)
\end{align*}
\]

where meta-variables \(M_1, M_2, M_3\) and \(M_4\) stand for input and output markings of transitions \(t\) and \(t'\).

When comparing this rule to the interleaving one, two cases have to be explicated:

1. Either transitions \(t\) and \(t'\) are truely independent (\(M\) is equal to 1 when applying the interleaving rule to \(t\) and \(t')\) and the both rules derive the same result formula. In such a case \(t \parallel t' = \{tt', t't\}\) and the result formula of concurrent composition characterizes the set \(\{tt', t't, t \parallel t'\}\).

2. Or transitions \(t\) and \(t'\) are not independent ones: applying the concurrent composition rule is always possible (concurrent firing of two transitions) even in cases where interleaving one cannot be applied. We are going to see an example later.

As for sequential composition, this rule is attractive if it supports an internal composition rule. The formula characterizing the set of sequences denoted \(t \parallel t'\) having exactly the same form as a transition formula, it is possible to extend it to premises that are sequences and then sets of sequences. Each premise formula is prefixed by elements of \(E = \mathcal{P}(T^*)\), the set of \(T^*\) parts.

So, we can define the \(\pi\) rule:

\[
\begin{align*}
    \epsilon_1 &: (M_1 \rightarrow M_2) \\
    \epsilon_2 &: (M_3 \rightarrow M_4) \\
    \epsilon_1 \parallel \epsilon_2 &: (M_1 \otimes M_3 \rightarrow M_2 \oplus M_4)
\end{align*}
\]

where \(\epsilon_1 = \{s_1, s_2, \ldots\}\) and \(\epsilon_2 = \{s'_1, s'_2, \ldots\}\) are elements of \(E\).

Metavariables \(M_1, M_2, M_3\) and \(M_4\) stand for marking formulas expressing pre and post-conditions of sequence firings belonging respectively to \(\epsilon_1\) and \(\epsilon_2\). The internal composition law \(\parallel\) on \(E\) is defined by \(\epsilon_1 \parallel \epsilon_2 = \{s \parallel s' | s \in \epsilon_1\) and \(s' \in \epsilon_2\)\). This law is commutative and associative and admits a singleton (reduced to the empty sequence \(\{\lambda\}\)) as neutral element.

4.5 Sequential composition of sets of sequences

As we have just introduced sets of sequences concept for interleaving and concurrent composition rules, it seems interesting to extend the sequential composition rule in order it can also manage sets of sequences as premise formulas.
The rule $\sigma$ is then defined by:

$$
e_1 : (\text{M}_1 \to \text{M}
\otimes \text{L}) \quad e_2 : (\text{M}
\otimes \text{K} \to \text{M}_2)
$$

where $e_1 = \{s_1, s_2, \ldots\}$ and $e_2 = \{s'_1, s'_2, \ldots\}$ are both elements of $\mathcal{E}$. Application conditions for $M$, $K$ and $L$ are identical to those previously defined for others sequential rules. The concatenation law, internal on $\mathcal{E}$, is defined by $e_1 e_2 = \{ss' \mid s \in e_1 \text{ and } s' \in e_2\}$. As for the previous one, it is associative and admits $\lambda$ as neutral element, but not surprisingly, it is non commutative: we previously saw that sequences characterizations were ordered ones.

5. Some examples

Let us consider, as a first example, the net in figure 3. Its linear logic translation is:

$$t_1 : (A \to B \otimes D), \quad t_2 : (B \to C), \quad t_3 : (D \to E), \quad t_4 : (C \otimes E \to F).$$

We are trying to compute sequences involving transitions $t_1$ to $t_4$. Rules $\pi$ and $\text{entr}$ can be applied to transitions $t_2$ and $t_3$. Then it is possible to complete these sequences using the $\sigma$ rule.

Let us first consider the structural concurrence concerning $t_2$ and $t_3$:

$$
t_2 : (B \to C) \quad t_3 : (D \to E)
$$

$$t_2 \parallel t_3 : (B \otimes D \to C \otimes E) \quad \pi$$

$$t_1 : (A \to B \otimes D) \quad t_2 \parallel t_3 : (B \otimes D \to C \otimes E)
$$

$$t_1(t_2 \parallel t_3) : (A \to C \otimes E) \quad \sigma$$

$$t_1(t_2 \parallel t_3) : (A \to C \otimes E) \quad t_4 : (C \otimes E \to F)
$$

$$t_1(t_2 \parallel t_3)t_4 : (A \to F) \quad \sigma$$
Let us now deal with interleaving:

\[
t_2 : (B \rightarrow C) \quad t_3 : (D \rightarrow E) \\
\{t_2t_3, t_3t_2\} : (B \otimes D \rightarrow C \otimes E)
\]

entr

We finally get:

\[
\{t_1t_2t_3t_4, t_1t_3t_2t_4\} : (A \rightarrow F)
\]

Not surprisingly, the two derived formulas \(t_1(t_2 \parallel t_3)t_4\) and \(\{t_1t_2t_3t_4, t_1t_3t_2t_4\}\) are identical because \(t_2\) and \(t_3\) are truely independent transitions.

Let us now consider the net in figure 4. In this net, only \(t_2\) and \(t_3\) differ from the previous one: they are assymetrically tied up by place \(\alpha\).

\[
t_2 : (B \rightarrow C \otimes \alpha), \quad t_3 : (D \otimes \alpha \rightarrow E).
\]

The potential simultaneous firing of \(t_2\) and \(t_3\), involved in rule \(\pi\), produces this formula:

\[
t_2 : (B \rightarrow C \otimes \alpha) \quad t_3 : (D \otimes \alpha \rightarrow E) \\
\frac{t_2 \parallel t_3 : (B \otimes D \otimes \alpha \rightarrow C \otimes E \otimes \alpha)}{\pi}
\]

Then, the complete sequence is derived using the \(\sigma\) rule:

\[
t_1(t_2 \parallel t_3)t_4 : (A \otimes \alpha \rightarrow F \otimes \alpha)
\]

In this case, the rule entr cannot be applied and we have to separately characterize the two sequences. The only applicable rule is the \(\sigma\) one.

\[
t_1t_2t_3t_4 : (A \rightarrow F) \\
t_1t_3t_2t_4 : (A \otimes \alpha \rightarrow F \otimes \alpha)
\]

As the entr rule cannot be applied, two different formulas are derived. This example illustrates a first case where the concurrent firing rule leads to results differing from formulas expressing the two sequential compositions.
At last, consider now the net in figure 5. Again, transitions $t_2$ and $t_3$ differ from nets in figures 3 and 4. In this case, $t_2$ and $t_3$ are tied up by symmetrical links (place $\alpha$).

$$t_2 : (B \otimes \alpha \rightarrow C \otimes \alpha), \quad t_3 : (D \otimes \alpha \rightarrow E \otimes \alpha).$$

Sequences $t_1 t_2 t_3 t_4$ and $t_1 t_3 t_2 t_4$ are then characterized by identical formulas, and the rule $\text{entr}$ can be applied to $t_2$ and $t_3$.

$$\begin{align*}
\{t_2 t_3, t_3 t_2\} & : (B \otimes \alpha \rightarrow C \otimes \alpha) \\
& \text{entr} \\
\end{align*}$$

In this case, the complete sequence is:

$$\{t_1 t_2 t_3 t_4, t_1 t_3 t_2 t_4\} : (A \otimes \alpha \rightarrow F \otimes \alpha)$$

Of course, it is possible to apply the rule $\pi$ to $t_2$ and $t_3$ (there is no particular condition to use it). The derived formula differs from the one obtained using the rule $\text{entr}$: simultaneous firing of $t_2$ and $t_3$ does not require the same preconditions than their interleaved firing.

$$\begin{align*}
\pi & : (B \otimes \alpha \rightarrow C \otimes \alpha) \quad (D \otimes \alpha \rightarrow E \otimes \alpha) \\
\{t_2, t_3\} & : (B \otimes D \otimes 2 \alpha \rightarrow C \otimes E \otimes 2 \alpha) \\
\pi & : t_1 (t_2 \parallel t_3) t_4 : (A \otimes 2 \alpha \rightarrow F \otimes 2 \alpha)
\end{align*}$$

This result illustrates a major advantage of linear logic since it clearly differentiates a concurrent firing from an interleaving one.

Finally, let us see how these examples can be differentiated when using the classical $C, \pi$ characterization: using this method there is no difference whatever the considered net (figures 3, 4 or 5) and whatever the considered sequence ($t_1 t_2 t_3 t_4$ or $t_1 t_3 t_2 t_4$). In all cases the result is the same: it expresses that the algebraic changes are $+1$ for place $A$ and $-1$ for place $F$. Others places are not involved in $C, \pi$ conditions.
6. Conclusion

The goal of this paper is to study how linear logic could enhance the Petri net theory.

At least, three points have to be pointed out:

First, linear logic permits a Petri net description which is as concise as the one offered by a formal grammar on the place alphabet. It is possible to construct reasoning schemes and rules according to usual concepts of the Petri net theory: net structure, markings and firings are distinctly formalized.

Second, by attaching valid reasoning schemes to transition and sequence rules, we get a sequence calculus much more powerful and simple than the one provided by usual algebraic techniques using matrix \( C \). In addition, this sequence calculus handles sets of sequences for sequential and concurrent compositions. However, we think that the usual algebraic methods cannot be dropped out because they are, for example, very powerful tools for computing invariants. We did not show it in this paper, but a lot of problems can be efficiently solved by combining these two complementary techniques. In our mind, they are not exclusive and some examples show how it is essential to simultaneously use both.

Third, some tools, like the calculus for firing pre-conditions, were already existing but linear logic permits to define them in a richer context.

Some points have not been discussed in this paper but it has to be noticed that some not clearly defined concepts can be better analyzed when using the whole set of linear logic connectives. For example, the additive ones permit to better explicit conflicts, while linear negation permits retrospective abductive reasoning schemes. All these points are future trends for our work.

Acknowledgements:
We are very grateful to Yves Lafont (Laboratoire de Mathématiques discrètes, Marseille, France) for his answers to our questions about linear logic.

7. References


