# Vector Addition Tree Automata

Philippe de Groote, Bruno Guillaume and Sylvain Salvati
LORIA UMR n° 7503 – INRIA
Campus Scientifique, B.P. 239
54506 Vandœuvre lès Nancy Cedex – France

## Abstract

*We introduce a new class of automata, which we call vector addition tree automata. These automata are a natural generalization of vector addition systems with states, which are themselves equivalent to Petri nets. Then, we prove that the decidability of provability in multiplicative exponential linear logic (which is an open problem) is equivalent to the decidability of the reachability relation for vector addition tree automata. This result generalizes the well-known connection existing between Petri nets and the !-Horn fragment of multiplicative exponential linear logic.*

## 1 Introduction

Petri nets (PN), and equivalent systems such as vector addition systems (VAS) [11] or vector addition systems with states (VASS) [7], have been extensively studied as models of parallelism and resource sensitive systems. Consequently, when Girard introduced linear logic [5] (which is a resource sensitive logic that allows for some kind of parallelism), several authors started to investigate the connections between this new logic and Petri nets, both on the syntactic and semantic sides [6, 1, 13, 2, 4].

Along this line of research, M. Kanovich established several equivalence results between different notions of Petri nets and some Horn-like fragments of linear logic [8, 9, 10]. In particular, he derived the decidability of the !-Horn fragment of multiplicative exponential linear logic (**MELL**) [9] from the decidability of the reachability relation in Petri nets [12, 14].

Unfortunately, Kanovich's decidability result cannot be easily generalized to the decidability of **MELL**, which is still open. Indeed, there is no clear correspondence between **MELL** provability and Petri net reachability. Therefore, a possible way of tackling the **MELL** decidability problem is to first answer the following question: is there a natural generalization of Petri nets whose reachability relation would be equivalent to provability in **MELL**?

In this paper, we propose an answer to this question. We introduce a notion of *vector addition tree automaton* (VATA) that generalizes both vector addition systems with states and tree automata (TA) [3]. In fact, our vector addition tree automata are a generalization of the usual tree automata in exactly the same way that vector addition systems with states are a generalization of finite state automata (FSA). From an orthogonal viewpoint, they generalize vector addition systems with states in exactly the same way that tree automata generalize finite state automata. The picture is thus the following:

$$\begin{array}{ccccc} \text{FSA} & \text{———} & \text{VASS} \equiv \text{PN} & \text{———} & \text{!-Horn} \\ | & & \vdots & & | \\ \text{TA} & \text{- - - - - -} & \text{VATA} & \text{- - - - -} & \text{MELL} \end{array}$$

The paper is organized as follows. The next section reminds the reader of some prerequisites, and fixes the notations. Section 3 introduces the notion of vector addition tree automata. We give the definition, define a notion of normal form, and establish an equivalence result between a VATA and its normal form. Section 4 is devoted to **MELL**: we define **IMELL**, the intuitionistic fragment of **MELL** and we recall the equivalence of **MELL** and **IMELL**. Then, we reduce provability in **IMELL** to provability in a very constrained fragment of **IMELL**, which we call $\mathbf{s\text{–}IMELL}_{\mathbf{0}}^{\multimap}$. Finally, in section 5, we establish a correspondance between provability in $\mathbf{s\text{–}IMELL}_{\mathbf{0}}^{\multimap}$ and reachability for VATA under normal form.

## 2 Preliminaries

This section introduces the necessary mathematical background and fixes the notations that we use in the sequel. For the sake of completeness, we first remind the reader of the usual notions of ranked alphabet, term, context, and linear tree-homomorphism.

**Definition 1.** *A ranked alphabet $\mathcal{F} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ is an indexed family of disjoint finite sets, such that $\bigcup_{n\in\mathbb{N}} \mathcal{F}_n$ is finite.*

By abuse of language, we sometimes confuse $\mathcal{F}$ with $\bigcup_{n\in\mathbb{N}} \mathcal{F}_n$, and we speak about the elements of $\mathcal{F}$ when we mean the elements of $\bigcup_{n\in\mathbb{N}} \mathcal{F}_n$. These elements will be called the symbols of the alphabet. When such a symbol belongs to $\mathcal{F}_n$, one says that its arity is $n$. In particular, the symbols of arity 0 are called constants.

**Definition 2.** *Given a ranked alphabet $\mathcal{F} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ and a possibly infinite set $\mathcal{X}$ disjoint from $\mathcal{F}$, the set of terms $\mathscr{T}(\mathcal{F},\mathcal{X})$ built over $\mathcal{F}$ and $\mathcal{X}$ is inductively defined as follows:*

1. *if $x \in \mathcal{X}$, then $x \in \mathscr{T}(\mathcal{F},\mathcal{X})$;*
2. *if $c \in \mathcal{F}_0$, then $c \in \mathscr{T}(\mathcal{F},\mathcal{X})$;*
3. *if $f \in \mathcal{F}_n$, and $t_0,\ldots,t_{n-1} \in \mathscr{T}(\mathcal{F},\mathcal{X})$, then $f(t_0,\ldots,t_{n-1}) \in \mathscr{T}(\mathcal{F},\mathcal{X})$, for $n > 0$.*

The usual case for $\mathcal{X}$ is to be a set of variables. Nevertheless, we will consider other cases. If $\mathcal{X}$ is empty, the set $\mathscr{T}(\mathcal{F},\varnothing)$ is called the set of ground terms and is written $\mathscr{T}(\mathcal{F})$.

**Definition 3.** *Let $\mathcal{F}$ be a ranked alphabet and $\mathcal{X}_n = \{x_i \mid i \in n\}$ be a set of $n$ variables, disjoint from $\mathcal{F}$. The set of $n$-contexts $\mathscr{C}_n(\mathcal{F})$ is the set of terms $C \in \mathscr{T}(\mathcal{F},\mathcal{X}_n)$ that contain exactly one occurrence of each variable $x_i \in \mathcal{X}_n$.*

Given $C \in \mathscr{C}_n(\mathcal{F})$, and $t_0,\ldots,t_{n-1} \in \mathscr{T}(\mathcal{F})$, one writes $C[t_0,\ldots,t_{n-1}]$ for the ground term obtained by replacing in $C$ the occurrence of $x_i$ by $t_i$, for each $i \in n$.

The notion of context allows the notion of linear tree-homomorphism to be defined.

**Definition 4.** *Let $\mathcal{F}$ and $\mathcal{G}$ be two ranked alphabets, and let $(\theta_n)_{n\in\mathbb{N}}$ be a family of functions that associate to each symbol $f \in \mathcal{F}_n$ a $n$-context $C_f \in \mathscr{C}_n(\mathcal{G})$. The linear tree-homomorphism $\theta : \mathscr{T}(\mathcal{F}) \to \mathscr{T}(\mathcal{G})$, generated by the family $(\theta_n)_{n\in\mathbb{N}}$, is inductively defined as follows: $\theta(f(t_0,\ldots,t_{n-1}) = C_f[\theta(t_0),\ldots,\theta(t_{n-1})] \in \mathscr{T}(\mathcal{G})$, for each $f \in \mathcal{F}_n$.*

Remark that in Definition 3 we require that each variable $x_i$ occurs exactly once. Consequently, our notion of a linear tree-homomorphism corresponds to what is usually called in the literature a *non-erasing* linear tree-homomorphism.

The next notions we introduce are proper to the kind of automata that we will define in the next section. These automata use vectors of natural numbers, i.e., elements of $\mathbb{N}^k$, for some $k \in \mathbb{N}$. We let boldface lowercase Roman letters, $\mathbf{x},\mathbf{y},\mathbf{z},\ldots$, range over such vectors. We use $\mathbf{0}$ to denote the null vector $(0,0,\ldots,0)$, and $\mathbf{e}_j$ ($j \in k$) to denote the standard base vectors, i.e, $\mathbf{e}_0 = (1,0,\ldots,0)$, $\mathbf{e}_1 = (0,1,\ldots,0)$, $\ldots$, $\mathbf{e}_{k-1} = (0,0,\ldots,1)$. For a vector $\mathbf{x} = (x_0,\ldots,x_{n-1})$ we write $\|\mathbf{x}\| = \sum_{i\in n} x_i$.

**Definition 5.** *Let $Q$ be a finite set, the elements of which will be called states, and let $k \in \mathbb{N}$. A $k$-configuration over $Q$ is a pair $(q,\mathbf{x}) \in Q \times \mathbb{N}^k$.*

We now define the notion of description, which is the data structure on which our automata operate.

**Definition 6.** *Let $\mathcal{F}$ be a ranked alphabet, let $Q$ be a finite set of states, and let $k \in \mathbb{N}$. The set of $k$-descriptions over $\mathcal{F}$ and $Q$, $k$-$\mathscr{D}(\mathcal{F},Q)$, is defined to be the set of terms $\mathscr{T}(\mathcal{F}, Q \times \mathbb{N}^k)$.*

Adapting the notion of context to the case of description is straightforward.

**Definition 7.** *Let $\mathcal{F}$ be a ranked alphabet, $x$ be a symbol that does not belong to $\mathcal{F}$, $Q$ be a finite set of states, and let $k \in \mathbb{N}$. The set of $k$-description contexts, $k$-$\mathscr{C}(\mathcal{F},Q)$, is the set of terms $C \in \mathscr{T}(\mathcal{F},(Q\times\mathbb{N}^k)\cup\{x\})$ that contain exactly one occurrence of $x$.*

Given a $k$-description $t \in k$-$\mathscr{D}(\mathcal{F},Q)$, one writes $C[t]$ for the $k$-description obtained by replacing the occurrence of $x$ by $t$ in $C$.

# 3 Vector Addition Tree Automata

## 3.1 Definition

We are now in the position of giving the main definition of this paper, i.e., the definition of vector addition tree automata. Vector addition systems with states may be seen as finite state automata where transitions are labelled with integer vectors. Our notion of vector addition tree automaton generalizes this to the case of (bottom-up) finite tree-automata.

**Definition 8.** *A vector addition tree automaton of dimension $k$ ($k$-VATA, for short) is a quadruple $\langle \mathcal{F}, Q, C_f, \Delta\rangle$ where:*

1. *$\mathcal{F}$ is a ranked alphabet;*
2. *$Q$ is a finite set of states;*

3. $C_f$ *is a finite set of* $k$*-configurations over* $Q$*, called accepting configurations;*

4. $\Delta \subset \bigcup_{n\in\mathbb{N}}(\mathcal{F}_n \times (Q \times \mathbb{N}^k)^n \times Q \times \mathbb{N}^k)$ *is a finite set of transition rules.*

*A* $k$*-VATA is strongly deterministic if there is at most one transition in* $\Delta$ *for each symbol in* $\mathcal{F}$*.*

Let $\langle \mathcal{F}, Q, C_f, \Delta\rangle$ be a $k$-VATA, and consider some transition rule:

$$(f, (q_0, \mathbf{z}_0), \ldots, (q_{n-1}, \mathbf{z}_{n-1}), q, \mathbf{z}) \in \Delta.$$

Such a transition rule will be written as:

$$f((q_0, \mathbf{x}_0), \ldots, (q_{n-1}, \mathbf{x}_{n-1})) \longrightarrow \Big(q, \sum_{i\in n}(\mathbf{x}_i - \mathbf{z}_i) + \mathbf{z}\Big)$$

where $\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}$ are formal parameters ranging over $\mathbb{N}^k$. Under this form, $\Delta$ appears to be a rewriting system acting on $k$-descriptions. This motivates the definition of the move relation.

**Definition 9.** *Let* $\mathcal{A} = \langle \mathcal{F}, Q, C_f, \Delta\rangle$ *be a* $k$*-VATA. The move relation* $\longrightarrow_{\mathcal{A}}$ *is defined as follows. Let* $t, u \in k\text{-}\mathscr{D}(\mathcal{F}, Q)$*,*

$$t \longrightarrow_{\mathcal{A}} u$$

*if and only if there exists a description context* $C \in k\text{-}\mathscr{C}(\mathcal{F}, Q)$ *and a production rule*

$$f((q_0, \mathbf{x}_0), \ldots, (q_{n-1}, \mathbf{x}_{n-1})) \longrightarrow \Big(q, \sum_{i\in n}(\mathbf{x}_i - \mathbf{z}_i) + \mathbf{z}\Big)$$

*in* $\Delta$ *such that*

1. $t = C[f((q_0, \mathbf{y}_0), \ldots, (q_{n-1}, \mathbf{y}_{n-1}))]$*;*

2. $u = C[(q, \sum_{i\in n}(\mathbf{y}_i - \mathbf{z}_i) + \mathbf{z})]$*;*

3. $\forall i \in n.\ \mathbf{y}_i - \mathbf{z}_i \in \mathbb{N}^k$*.*

As usual, $\longrightarrow^*_{\mathcal{A}}$ denotes the reflexive transitive closure of the move relation $\longrightarrow_{\mathcal{A}}$.

In the above definition, Condition 3 is central. It corresponds to the positivity condition of vector addition systems. Consider, for instance, the following 2-VATA:

$$\begin{array}{rcl}
\mathcal{F} & = & \{\, a, b, g(), f(,) \,\} \\
Q & = & \{\, q_0, q_1, q_2 \,\} \\
C_f & = & \{\, (q_0, (0,0)) \,\} \\
\Delta & = & \{\ a \longrightarrow (q_2, (1,0)), \\
 & & \ \ b \longrightarrow (q_2, (0,1)), \\
 & & \ \ f((q_2, \mathbf{x}), (q_2, \mathbf{y})) \longrightarrow (q_1, \mathbf{x} + \mathbf{y}), \\
 & & \ \ g((q_1, \mathbf{x})) \longrightarrow (q_0, \mathbf{x} - (1,1)) \ \}
\end{array}$$

Then we have:

$$g(f(a,b)) \longrightarrow^*_{\mathcal{A}} (q_0, (0,0)),\ g(f(b,a)) \longrightarrow^*_{\mathcal{A}} (q_0, (0,0))$$

On the other hand, we have:

$$g(f(a,a)) \longrightarrow^*_{\mathcal{A}} g((q_1, (2,0)))$$

but then we are stuck because of Condition 3.

As one expects, a ground term is recognized by a $k$-VATA if it can be rewritten into an accepting configuration.

**Definition 10.** *Let* $\mathcal{A} = \langle \mathcal{F}, Q, C_f, \Delta\rangle$ *be a* $k$*-VATA. The tree language* $\mathcal{L}_{\mathcal{A}}$ *recognized by* $\mathcal{A}$ *is defined as follows:*

$$\mathcal{L}_{\mathcal{A}} = \{t \in \mathscr{T}(\mathcal{F}) \mid t \longrightarrow^*_{\mathcal{A}} (q, \mathbf{z}) \text{ for some } (q, \mathbf{z}) \in C_f\}$$

With different motivations (modelisation of cryptographic protocols), other authors have defined a formalism equivalent to VATA (they have called it BVASS for Branching VASS) [17].

### 3.2 Normal form

In order to give a simple translation of a VATA in linear logic, we define a normal form for VATAs. The main point is that checking emptiness for a given VATA is equivalent to checking emptiness for another VATA in normal form.

**Definition 11.** *A production rule is in normal form if it has one of the following forms:*

$$\begin{array}{l}
c \longrightarrow (q, \mathbf{e}_i) \text{ for some } i \in k, \\
f((q_0, \mathbf{x}_0)) \longrightarrow (q, \mathbf{x}_0 - \mathbf{e}_i) \text{ for some } i \in k, \\
f((q_0, \mathbf{x}_0), (q_1, \mathbf{x}_1)) \longrightarrow (q, \mathbf{x}_0 + \mathbf{x}_1).
\end{array}$$

*A* $k$*-VATA* $\mathcal{A} = \langle \mathcal{F}, Q, C_f, \Delta\rangle$ *is in normal form iff*

1. $\forall n > 2.\ \mathcal{F}_n = \varnothing$*;*

2. $C_f = \{\, (q_f, \mathbf{0}) \,\}$ *for some* $q_f \in Q$*;*

3. *the production rules in* $\Delta$ *are in normal form.*

The construction of an automaton in normal form proceeds in three steps (the three following lemmas). First, we construct an automaton with only one final state; then, we make it strongly deterministic; and, then, we modify the set of productions to have only normal forms.

**Lemma 1.** *Let* $\mathcal{A} = \langle \mathcal{F}, Q, C_f, \Delta\rangle$ *a* $k$*-VATA. There is a* $k$*-VATA* $\mathcal{A}' = \langle \mathcal{F}', Q', \{(q_f, \mathbf{0})\}, \Delta'\rangle$ *and a linear tree-homomorphism* $\theta : \mathscr{T}(\mathcal{F}) \to \mathscr{T}(\mathcal{F}')$ *such that* $\mathcal{L}_{\mathcal{A}'} = \theta(\mathcal{L}_{\mathcal{A}})$*.*

*Proof.* We define $\mathcal{F}' = \mathcal{F} \cup \{g\}$ where $g \in \mathcal{F}'_1$ is a new symbol of arity 1; $Q' = (Q \times \{\uparrow, \downarrow\}) \cup \{q_f\}$ and, for each $q \in Q$, we write $q^\uparrow$ and $q^\downarrow$ the two copies of $q$ in $Q'$. $\Delta'$ is given by:

- for each production in $\delta \in \Delta$:

$$f((q_0, \mathbf{x}_0), \ldots, (q_{n-1}, \mathbf{x}_{n-1})) \longrightarrow \left(q, \sum_{i \in n} (\mathbf{x}_i - \mathbf{z}_i) + \mathbf{z}\right)$$

  the following production $\delta'$ is in $\Delta'$:

$$f((q_0^\uparrow, \mathbf{x}_0), \ldots, (q_{n-1}^\uparrow, \mathbf{x}_{n-1})) \longrightarrow \left(q^\downarrow, \sum_{i \in n} (\mathbf{x}_i - \mathbf{z}_i) + \mathbf{z}\right)$$

- for each state $q \in Q$, the production $\delta'$ is in $\Delta'$:

$$\delta' = g((q^\downarrow, \mathbf{x})) \longrightarrow (q^\uparrow, \mathbf{x})$$

- for $(q, \mathbf{z}) \in C_f$, the production $\delta_{(q,\mathbf{z})}$ is in $\Delta'$:

$$\delta_{(q,\mathbf{z})} = g((q^\downarrow, \mathbf{x})) \longrightarrow (q_f, \mathbf{x} - \mathbf{z})$$

The linear tree-homomorphism $\theta$ is defined by: $\theta(f) = g(f(x_0, \ldots, x_{n-1}))$ for each $f \in \mathcal{F}_n$.

We can now prove by induction the two facts:

1. if $t \longrightarrow^*_{\mathcal{A}} (q, \mathbf{z})$ then there is some $u \in \mathscr{T}(\mathcal{F}')$ such that $\theta(t) = g(u)$ and $u \longrightarrow^*_{\mathcal{A}'} (q^\downarrow, \mathbf{z})$.
2. if $u \longrightarrow^*_{\mathcal{A}'} (q^\downarrow, \mathbf{z})$ then there is some $t \in \mathscr{T}(\mathcal{F})$ such that $g(u) = \theta(t)$ and $t \longrightarrow^*_{\mathcal{A}} (q, \mathbf{z})$.

Finally, we prove that $\mathcal{L}_{\mathcal{A}'} = \theta(\mathcal{L}_{\mathcal{A}})$. Let $t \in \mathcal{L}_{\mathcal{A}}$, that is $t \longrightarrow^*_{\mathcal{A}} (q, \mathbf{z})$ with $(q, \mathbf{z})$ an accepting state. By fact 1, there is some $u$ such that $\theta(t) = g(u)$ and $u \longrightarrow^*_{\mathcal{A}'} (q^\downarrow, \mathbf{z})$. As $(q, \mathbf{z})$ is an accepting state, $\theta(t) = g(u) \longrightarrow^*_{\mathcal{A}'} (q_f, \mathbf{0})$, hence $\theta(t) \in \mathcal{L}_{\mathcal{A}'}$. Conversely, let $u \in \mathcal{L}_{\mathcal{A}'}$, i.e. $u \longrightarrow^*_{\mathcal{A}'} (q_f, \mathbf{0})$ then necessarily, $u = g(u')$ with $u' \longrightarrow^*_{\mathcal{A}'} (q^\downarrow, \mathbf{z})$ and $(q, \mathbf{z})$ is an accepting state of $\mathcal{A}$. By fact 2, there is some $t$ such that $g(u) = \theta(t)$ and $t \longrightarrow^*_{\mathcal{A}} (q, \mathbf{z})$, hence $t \in \mathcal{L}_{\mathcal{A}}$. □

**Lemma 2.** *Let $\mathcal{A} = \langle \mathcal{F}, Q, \{(q_f, \mathbf{0})\}, \Delta \rangle$ a $k$-VATA. There is a strongly deterministic $k$-VATA $\mathcal{A}' = \langle \mathcal{F}', Q, \{(q_f, \mathbf{0})\}, \Delta' \rangle$ and a linear tree-homomorphism $\theta : \mathscr{T}(\mathcal{F}') \to \mathscr{T}(\mathcal{F})$ such that $\mathcal{L}_{\mathcal{A}} = \theta(\mathcal{L}_{\mathcal{A}'})$.*

The idea of the proof is to split each symbol $f \in \mathcal{F}$ into some symbols $f_1, f_2, \ldots$ such that two different production rules always refer to different symbols. Then, the tree-homomorphism just maps each $f_i$ to $f$.

It is important to work with strongly deterministic automata rather than with deterministic ones. Indeed, the transformation $\Longrightarrow_1$, in the lemma 3, preserves strong determinism while it may not preserve determinism.

*Proof.* We write $\Delta_f$ the set of productions of $\Delta$ associated with the symbol $f$. For each $f \in \mathcal{F}$, let $j(f)$ be the cardinal of $\Delta_f$ and $\{\delta_{f_1}, \ldots, \delta_{f_{j(f)}}\}$ a fixed enumeration of $\Delta_f$. We define $\mathcal{F}'$:

$$\mathcal{F}' = \bigcup_{f \in \mathcal{F}} \{f_1, \ldots, f_{j(f)}\}$$

For each production $\delta_{f_i}$ of the form

$$f((q_0, \mathbf{x}_0), \ldots, (q_{n-1}, \mathbf{x}_{n-1})) \longrightarrow \left(q, \sum_{i \in n} (\mathbf{x}_i - \mathbf{z}_i) + \mathbf{z}\right)$$

we define $\delta'_{f_i}$:

$$f_i((q_0, \mathbf{x}_0), \ldots, (q_{n-1}, \mathbf{x}_{n-1})) \longrightarrow \left(q, \sum_{i \in n} (\mathbf{x}_i - \mathbf{z}_i) + \mathbf{z}\right)$$

and then

$$\Delta' = \bigcup_{f \in \mathcal{F}} \{\delta'_{f_1}, \ldots, \delta'_{f_{j(f)}}\}$$

Finally, the linear tree-homomorphism $\theta$ is defined by $\theta(f_i) = f(x_0, \ldots, x_{n-1})$ for each $f \in \mathcal{F}_n$.

With the previous definitions, the fact that $\mathcal{L}_{\mathcal{A}} = \theta(\mathcal{L}_{\mathcal{A}'})$ is straightforward. □

**Lemma 3.** *Let $\mathcal{A} = \langle \mathcal{F}, Q, \{(q_f, \mathbf{0})\}, \Delta \rangle$ be a strongly deterministic $k$-VATA. There is a $k$-VATA in normal form $\mathcal{A}' = \langle \mathcal{F}', Q', \{(q_f, \mathbf{0})\}, \Delta' \rangle$ and a linear tree-homomorphism $\theta : \mathscr{T}(\mathcal{F}) \to \mathscr{T}(\mathcal{F}')$ such that $\mathcal{L}_{\mathcal{A}'} = \theta(\mathcal{L}_{\mathcal{A}})$.*

*Proof.* We give a set of elementary automata transformations (see fig. 1) that are used to construct step by step a normal form. For each transformation, written $\Longrightarrow_i$, we only give as the left hand side the production $\delta_f$ that is replaced and the set of new productions added as the right hand side of $\Longrightarrow_i$; for the linear tree-homomorphism $\theta$, we give only its value on $f$, and suppose that $\theta$ is the identity function elsewhere.

The construction of the automaton in normal form proceeds as follows:

- Reaching $\forall n > 2.\ \mathcal{F}_n = \varnothing$. We define $\eta(\mathcal{A}) = \sum_{n>2} n|\mathcal{F}_n|$. By induction, if $\eta(\mathcal{A}) = 0$, there is nothing to do; else, with $\Longrightarrow_1$, we construct an automaton with a smaller $\eta$.
- Obtaining an automaton with productions of arity 2 in normal form. We replace each production of arity 2 which is not in normal form with the transformation $\Longrightarrow_2$.
- Obtaining an automaton with productions of arity 1 in normal form. Each production of arity 1 can be written

$$f(q_0, \mathbf{x}_0) \longrightarrow (q, (\mathbf{x}_0 - \mathbf{z}_0) + \mathbf{z})$$

$$f((q_0,\mathbf{x}_0),\ldots,(q_{n-1},\mathbf{x}_{n-1})) \longrightarrow (q, \textstyle\sum_{i\in n}(\mathbf{x}_i-\mathbf{z}_i)+\mathbf{z}) \Longrightarrow_1 \left\{\begin{array}{rcl} f_1((q_0,\mathbf{x}_0),(q_1,\mathbf{x}_1)) & \longrightarrow & (q',(\mathbf{x}_0-\mathbf{z}_0)+(\mathbf{x}_1-\mathbf{z}_1)) \\ f_2((q',\mathbf{x}'),(q_2,\mathbf{x}_2),\ldots,(q_{n-1},\mathbf{x}_{n-1})) & \longrightarrow & (q,\mathbf{x}'+\sum_{1<i<n}(\mathbf{x}_i-\mathbf{z}_i)+\mathbf{z}) \end{array}\right.$$
$$\theta(f(x_0,\ldots,x_{n-1})) = f_2(f_1(x_0,x_1),x_2,\ldots,x_{n-1})$$

$$f((q_0,\mathbf{x}_0),(q_1,\mathbf{x}_1)) \longrightarrow (q,(\mathbf{x}_0-\mathbf{z}_0)+(\mathbf{x}_1-\mathbf{z}_1)+\mathbf{z}) \Longrightarrow_2 \left\{\begin{array}{rcl} f_0(q_0,\mathbf{x}_0) & \longrightarrow & (q_0',\mathbf{x}_0-\mathbf{z}_0) \\ f_1(q_1,\mathbf{x}_1) & \longrightarrow & (q_1',\mathbf{x}_1-\mathbf{z}_1) \\ f_2((q_0',\mathbf{x}_0),(q_1',\mathbf{x}_1)) & \longrightarrow & (q',\mathbf{x}_0+\mathbf{x}_1) \\ f_3(q',\mathbf{x}) & \longrightarrow & (q,\mathbf{x}+\mathbf{z}) \end{array}\right.$$
$$\theta(f(x_0,x_1)) = f_3(f_2(f_0(x_0),f_1(x_1)))$$

$$f(q_0,\mathbf{x}_0) \longrightarrow (q,(\mathbf{x}_0-\mathbf{z}_0)+\mathbf{z}) \Longrightarrow_3 \left\{\begin{array}{rcl} f_0(q_0,\mathbf{x}_0) & \longrightarrow & (q',\mathbf{x}_0-\mathbf{z}_0) \\ f_1 & \longrightarrow & (q'',\mathbf{z}) \\ f_2((q',\mathbf{x}_0),(q'',\mathbf{x}_1)) & \longrightarrow & (q,\mathbf{x}_0+\mathbf{x}_1) \end{array}\right.$$
$$\theta(f(x)) = f_2(f_0(x),f_1)$$

$$f(q_0,\mathbf{x}_0) \longrightarrow (q,\mathbf{x}_0) \Longrightarrow_4 \left\{\begin{array}{rcl} f_0 & \longrightarrow & (q',\mathbf{0}) \\ f_1((q_0,\mathbf{x}_0),(q',\mathbf{x}_1)) & \longrightarrow & (q,\mathbf{x}_0+\mathbf{x}_1) \end{array}\right.$$
$$\theta(f(x)) = f_1(x,f_0)$$

$$f(q_0,\mathbf{x}_0) \longrightarrow (q,\mathbf{x}_0-(\mathbf{z}_0+\mathbf{e}_i))) \Longrightarrow_5 \left\{\begin{array}{rcl} f_0(q_0,\mathbf{x}_0) & \longrightarrow & (q',\mathbf{x}_0-\mathbf{e}_i) \\ f_1(q',\mathbf{x}_0) & \longrightarrow & (q,\mathbf{x}_0-\mathbf{z}_0) \end{array}\right.$$
$$\theta(f(x)) = f_1(f_0(x))$$

$$f \longrightarrow (q,\mathbf{0}) \Longrightarrow_6 \left\{\begin{array}{rcl} f_0 & \longrightarrow & (q',\mathbf{e}_0) \\ f_1(q',\mathbf{x}_0) & \longrightarrow & (q,\mathbf{x}_0-\mathbf{e}_0) \end{array}\right.$$
$$\theta(f) = f_1(f_0)$$

$$f \longrightarrow (q,\mathbf{z}+\mathbf{e}_i) \Longrightarrow_7 \left\{\begin{array}{rcl} f_0 & \longrightarrow & (q',\mathbf{z}) \\ f_1 & \longrightarrow & (q'',\mathbf{e}_i) \\ f_2((q',\mathbf{x}_0),(q'',\mathbf{x}_1)) & \longrightarrow & (q,\mathbf{x}_0+\mathbf{x}_1) \end{array}\right.$$
$$\theta(f) = f_2(f_0,f_1)$$

**Figure 1. Elementary automata transformations**

We proceed with the 3 following steps.

- For each production such that $\mathbf{z} \neq 0$, we apply transformation $\Longrightarrow_3$.
- Suppose $\mathbf{z} = \mathbf{0}$, for each production such that $\mathbf{z}_0 = \mathbf{0}$, we apply $\Longrightarrow_4$.
- Now, each production is such that $\mathbf{z} = \mathbf{0}$ and $\mathbf{z}_0 \neq \mathbf{0}$. If $\mathbf{z}_0 = \mathbf{e}_i$ then the production is in normal form. Else, we apply inductively $\Longrightarrow_5$ which decreases $\|\mathbf{z}_0\|$.

- Finally, for each production $f \longrightarrow (q,\mathbf{z})$:
  - if $\mathbf{z} = \mathbf{0}$, we apply $\Longrightarrow_6$,
  - if $\mathbf{z} \neq \mathbf{0}$, we apply inductively $\Longrightarrow_7$ which decreases $\|\mathbf{z}\|$. □

**Proposition 1.** *For any $k$-VATA $\mathcal{A}$ there exists a $k$-VATA $\mathcal{A}'$ in normal form such that $\mathcal{L}_{\mathcal{A}} = \emptyset$ iff $\mathcal{L}_{\mathcal{A}'} = \emptyset$.*

*Proof.* This is an easy consequence of the three previous lemmas and the fact that for any set of trees $\mathcal{L}$ and any linear tree-homomorphism $\theta$, $\theta(\mathcal{L}) = \varnothing$ if and only if $\mathcal{L} = \varnothing$. □

# 4 Multiplicative exponential linear logic

## 4.1 Definitions

In this section, we introduce four fragments of multiplicative exponential linear logic. We start with the intuitionistic fragment, which we call **IMELL**.

The formulas of **IMELL** are built upon a set of atomic formulas $\mathcal{A}$ according to the following syntax:

$$\mathcal{F} \ ::= \ \mathbf{1} \mid \mathcal{A} \mid \mathcal{F} \otimes \mathcal{F} \mid \mathcal{F} \multimap \mathcal{F} \mid\ !\mathcal{F}$$

The formulas of the form $A \otimes B$ and $A \multimap B$ are called multiplicative formulas, and the formulas of the form $!A$ are called exponential formulas.

Let Roman uppercase letters range over formulas, and Greek uppercase letters over multisets of formulas. The deduction relation of **IMELL** is specified by means of the following intuitionistic sequent calculus.

**Identity rules**

$$A \vdash A \quad \text{(ident)} \qquad\qquad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \ \text{(cut)}$$

**Logical rules**

$$\frac{\Gamma \vdash A}{\Gamma, \mathbf{1} \vdash A}\ (\mathbf{1}\text{-left}) \qquad \vdash \mathbf{1} \quad (\mathbf{1}\text{-right})$$

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}\ (\otimes\text{-left}) \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ (\otimes\text{-right})$$

$$\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}\ (\multimap\text{-left}) \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ (\multimap\text{-right})$$

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\ (!\text{-left}) \qquad \frac{!\Gamma \vdash A}{!\Gamma \vdash !A}\ (!\text{-right})$$

where, in Rule (!-right), $!\Gamma$ stands for a multiset of exponential formulas, i.e., formulas of the form $!F$.

**Structural rules**

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ (\mathrm{W}) \qquad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ (\mathrm{C})$$

Rules (!-left), (!-right), (W), and (C) are also called *dereliction, promotion, weakening*, and *contraction*, respectively. The cut rule is dispensable, and the cut-free proofs satisfy the subformula property.

At this point, we can say few words about classical **MELL**. The formulas of this system are built from positive atomic propositions $a, b, c, \ldots$, and corresponding negative atomic propositions $a^\perp, b^\perp, c^\perp, \ldots$ The constants and connectives include "$\mathbf{1}$", "$\otimes$", and "!", together with their duals "$\perp$", "$⅋$", and "?". Implication is defined as in classical logic, i.e., $A \multimap B = A^\perp ⅋ B$, where the negation, $(\cdot)^\perp$ is defined by means of de Morgan's laws: $\mathbf{1}^\perp = \perp$, $\perp^\perp = \mathbf{1}$, $(A \otimes B)^\perp = A^\perp ⅋ B^\perp$, etc. It is well known that the decidability problems of classical **MELL** and **IMELL** are equivalent (this may be established, for instance, by using a negative translation à la Kolmogorov [16]). Moreover, we have that the classical system is conservative over the intuitionistic one, in other words, any intuitionistic sequent that is classically provable is also intuitionistically provable. We will use this fact in the proof of Lemma 10.

The next fragments of interest are obtained by constraining the syntax of the formulas. We first define $\mathbf{IMELL_0}$ to be the fragment where exponential formulas are not allowed as strict subformulas. More formally, the formulas of $\mathbf{IMELL_0}$ obey the following syntax:

$$\begin{array}{lcl} \mathcal{F}_0 & ::= & \mathcal{M} \mid !\mathcal{M} \\ \mathcal{M} & ::= & \mathbf{1} \mid \mathcal{A} \mid \mathcal{M} \otimes \mathcal{M} \mid \mathcal{M} \multimap \mathcal{M} \end{array}$$

Moreover, exponential formulas are not allowed in the right-hand sides of the sequents (in fact, the only possibility of having an exponential formula in the right-hand side of a sequent would be in the conclusion of a derivation whose last rule is a promotion).

Then, we define $\mathbf{IMELL_0^{\multimap}}$ to be the implicative fragment of $\mathbf{IMELL_0}$, i.e., the fragment whose formulas obey the following syntax:

$$\begin{array}{lcl} \mathcal{F}_0^{\multimap} & ::= & \mathcal{M} \mid !\mathcal{M} \\ \mathcal{M} & ::= & \mathcal{A} \mid \mathcal{M} \multimap \mathcal{M} \end{array}$$

Finally, we define $\mathbf{s}$–$\mathbf{IMELL_0^{\multimap}}$ to be the fragment of $\mathbf{IMELL_0^{\multimap}}$ whose formulas are either atomic formulas or exponential formulas of one of the following forms:

$$!(a \multimap b) \qquad !(a \multimap (b \multimap c)) \qquad !((a \multimap b) \multimap c)$$

where $a$, $b$, and $c$ are atomic formulas.

The deduction relation of the three new fragments is the one of **IMELL**. Nevertheless, it is not difficult to prove that the following sequent calculus is correct and complete for $\mathbf{s}$–$\mathbf{IMELL_0^{\multimap}}$.

Let $!\Sigma$ range over multisets of exponential formulas of the specified form, and $\Gamma$ and $\Delta$ range over multisets of atomic formulas:

$$!\Sigma, a \vdash a \quad (\mathrm{T}_0)$$

$$\frac{!\Sigma, \Gamma \vdash a}{!\Sigma, \Gamma \vdash b}\ (\mathrm{T}_1) \text{ if } !(a \multimap b) \in !\Sigma$$

$$\frac{!\Sigma, \Gamma, a \vdash b}{!\Sigma, \Gamma \vdash c}\ (\mathrm{T}_2) \text{ if } !((a \multimap b) \multimap c) \in !\Sigma$$

$$\frac{!\Sigma, \Gamma \vdash a \quad !\Sigma, \Delta \vdash b}{!\Sigma, \Gamma, \Delta \vdash c}\ (\mathrm{T}_3) \text{ if } !(a \multimap (b \multimap c)) \in !\Sigma$$

### 4.2 From IMELL to $\mathbf{IMELL_0}$

In this section, we prove that **IMELL** is decidable if and only if $\mathbf{IMELL_0}$ is. The construction we give is inspired by a similar unpublished construction due to M. Kanovich.

Consider a given sequent of **IMELL**, $\Gamma \vdash A$. We intend to construct a set of formulas $\Gamma$ and a sequent $\Gamma^* \vdash A^*$ such that:

1. $\Gamma^* \vdash A^*$ is a purely multiplicative sequent;

2. $\Sigma$ is a sequence of formulas of the form $!S$, where $S$ is a purely multiplicative formula;

3. $\Sigma, \Gamma^* \vdash A^*$ is provable if and only if $\Gamma \vdash A$ is.

Let $\mathscr{M}$ be the set of formulas $F$ such that $!F$ is a subformula of the sequent $\Gamma \vdash A$. Then, to each formula in $F \in \mathscr{M}$, we associate a fresh atomic proposition $p_F$, and we define the following transformation over the subformulas of $\Gamma \vdash A$:

1. $\mathbf{1}^* = \mathbf{1}$;
2. $a^* = a$, for $a$ atomic;
3. $(F \otimes G)^* = F^* \otimes G^*$;
4. $(F \multimap G)^* = F^* \multimap G^*$;
5. $(!F)^* = p_F$.

In order to construct the sequence of formulas $\Sigma$, we associate to each formula $F \in \mathscr{M}$ three modal formulas:

$$\begin{array}{rcl} D_F & = & !(p_F \multimap F^*), \\ C_F & = & !(p_F \multimap p_F \otimes p_F), \\ W_F & = & !(p_F \multimap \mathbf{1}), \end{array}$$

and we define $\Sigma_0$ to be the following sequence of formulas:

$$\Sigma_0 = (D_F, C_F, W_F)_{F \in \mathscr{M}}.$$

The idea behind this definition is that dereliction, contraction and weakening will be simulated by the formulas $D_F$, $C_F$, and $W_F$, respectively. It therefore remains to allow promotion to be simulated. To this end, we will saturate the sequence of formulas $\Sigma_0$.

Let $\mathbf{P} = P_0, P_1, \ldots$ be an enumeration of all the formulas of the form

$$!(p_{F_1} \multimap \cdots \multimap p_{F_n} \multimap p_F)$$

such that

1. $F_1, \ldots, F_n, F \in \mathscr{M}$,
2. $F_i \neq F_j$, whenever $i \neq j$.

Notice that this enumeration is finite because of Condition 2. Then, define

$$\Sigma_{i+1} = \Sigma_i, P_k$$

where $P_k = !(p_{F_1} \multimap \cdots \multimap p_{F_n} \multimap p_F)$ is the first formula in $\mathbf{P}$ such that:

1. $P_k \notin \Sigma_i$,
2. $\Sigma_i, p_{F_1}, \ldots, p_{F_n} \vdash F^*$ is provable.

Finally, define $\Sigma$ to be the limit of the finite sequence $\Sigma_0, \Sigma_1, \ldots$.

Since $\mathscr{M}$ and $\mathbf{P}$ are finite, the only possibility of non-effectiveness in constructing $\Sigma$ is that one has to decide the provability of sequents (Condition 2 in the definition of $P_k$). Notice that these sequents belong to $\mathbf{IMELL_0}$. Consequently, we have the following lemma.

**Lemma 4.** *Let $\Gamma \vdash A$ be a sequent of* **IMELL**. *If* $\mathbf{IMELL_0}$ *is decidable then the construction of the associated* $\mathbf{IMELL_0}$ *sequent, $\Sigma, \Gamma^* \vdash A^*$, is effective.*

We now prove that $\Sigma, \Gamma^* \vdash A^*$ is provable if the original sequent $\Gamma \vdash A$ is.

**Lemma 5.** *Let $\Gamma \vdash A$ be a sequent of* **IMELL**, *and let $\Sigma, \Gamma^* \vdash A^*$ be the associated* $\mathbf{IMELL_0}$ *sequent. If $\Gamma \vdash A$ is provable, so is $\Sigma, \Gamma^* \vdash A^*$.*

*Proof.* By induction on the cut-free derivation of $\Gamma \vdash A$, the exponential formulas in $\Sigma$ allowing dereliction, contraction, weakening, and promotion to be simulated. Notice that, by the subformula property, $\Delta^*$ and $B^*$ are defined whenever $\Delta \vdash B$ is a sequent occurring in the cut-free derivation of $\Gamma \vdash A$. $\square$

In order to prove the converse of Lemma 5, we will use a semantic argument based on phase semantics. We will not give here a complete definition of the notion of phase space but refer the reader to [5] for the original definition, and to [15] for a definition tailor-made for intuitionistic linear logic.

**Lemma 6.** *Let $\Gamma \vdash A$ be a sequent of* **IMELL**, *and let $\Sigma, \Gamma^* \vdash A^*$ be the associated* $\mathbf{IMELL_0}$ *sequent. Let $\mathbb{P}$ be any phase space, and let $\eta$ be any valuation that interprets the atomic propositions as facts of $\mathbb{P}$. Then there exists a valuation $\eta'$ such that:*

$$[\![\Gamma \vdash A]\!]\eta = [\![\Sigma, \Gamma^* \vdash A^*]\!]\eta'$$

*Proof.* Let $\eta'$ be the valuation such that:

$$\eta'(a) = \begin{cases} [\![!F]\!]\eta & \text{if } a = p_F \text{ for some } F \in \mathscr{M} \\ \eta(a) & \text{otherwise} \end{cases}$$

By a straightforward induction, we have that $[\![\Gamma]\!]\eta = [\![\Gamma^*]\!]\eta'$ and $[\![A]\!]\eta = [\![A^*]\!]\eta'$. Moreover, under this valuation, we have that $[\![\Sigma]\!]\eta' = [\![\mathbf{1}]\!]$, which established the desired property. $\square$

We are now in the position of establishing the converse of Lemma 5.

**Lemma 7.** *Let $\Gamma \vdash A$ be a sequent of* **IMELL**, *and let $\Sigma, \Gamma^* \vdash A^*$ be the associated* $\mathbf{IMELL_0}$ *sequent. If $\Sigma, \Gamma^* \vdash A^*$ is provable, so is $\Gamma \vdash A$.*

*Proof.* Since $\Sigma, \Gamma^* \vdash A^*$ is provable, it is semantically valid in any phase space, under any interpretation. This implies, by Lemma 6, that $\Gamma \vdash A$ is also semantically valid in any phase space, under any interpretation. Consequently, by the phase semantics completeness theorem, we have that $\Gamma \vdash A$ is provable. $\square$

We obtain the main proposition of this section as an immediate consequence of Lemmas 4, 5, and 7.

**Proposition 2.** **IMELL** *is decidable if and only if* $\mathbf{IMELL_0}$ *is decidable.*

### 4.3 From $\mathbf{IMELL_0}$ to $\mathbf{IMELL_0^{\multimap}}$

In order to reduce provability in $\mathbf{IMELL_0}$ to provability in $\mathbf{IMELL_0^{\multimap}}$, we introduce the following positive and negative translations of the multiplicative formulas:

1. $A^+ = A^- \multimap b$, for any formula $A$;
2. $\mathbf{1}^- = b$;
3. $a^- = a \multimap b$;
4. $(A \otimes B)^- = A^+ \multimap (B^+ \multimap b)$
5. $(A \multimap B)^- = (A^+ \multimap B^+) \multimap b$

where $b$ is a fresh variable. The positive translation is then extended to the formulas of $\mathbf{IMELL_0}$ by defining $(!A)^+$ to be $!(A^+)$.

The above translation may be interpreted in classical multiplicative exponential linear logic as a translation by double negation. More formally, we have the following lemma.

**Lemma 8.** *Let $A$ be an* $\mathbf{IMELL_0}$ *formula, and let $A^+[b := \bot]$ denote the formula obtained by replacing, in the positive translation of $A$, each occurrence of the fresh variable $b$ by the constant $\bot$. Then, $A^+[b := \bot]$ is classically equivalent to $A$.*

*Proof.* By a straightforward induction on the structure of $A$, using de Morgan's laws and the following classical equivalences: $A \otimes \mathbf{1} = A = A \parr \bot$. $\square$

We now prove that any $\mathbf{IMELL_0}$ sequent is provable if and only if its positive translation is provable.

**Lemma 9.** *Let $\Gamma \vdash A$ be an* $\mathbf{IMELL_0}$ *sequent. If $\Gamma \vdash A$ is provable, so is $\Gamma^+ \vdash A^+$.*

*Proof.* By induction on the derivation of $\Gamma \vdash A$. $\square$

**Lemma 10.** *Let $\Gamma \vdash A$ be an* $\mathbf{IMELL_0}$ *sequent. If $\Gamma^+ \vdash A^+$ is provable, so is $\Gamma \vdash A$.*

*Proof.* If $\Gamma^+ \vdash A^+$ is intuitionistically provable, it is, a fortiori, classically provable. Then, since $b$ is a fresh variable, we also have that $\Gamma^+[b := \bot] \vdash A^+[b := \bot]$ is classically provable. From this, by Lemma 8, we have that $\Gamma \vdash A$ is classically provable, which allows us to conclude the proof because classical multiplicative exponential linear logic is conservative over intuitionistic multiplicative exponential linear logic. $\square$

We obtain the next proposition as an immediate consequence of Lemmas 9 and 10.

**Proposition 3.** $\mathbf{IMELL_0}$ *is decidable if and only if* $\mathbf{IMELL_0^{\multimap}}$ *is decidable.*

### 4.4 From $\mathbf{IMELL_0^{\multimap}}$ to s–$\mathbf{IMELL_0^{\multimap}}$

Finally, we show that $\mathbf{IMELL_0^{\multimap}}$ provability is equivalent to **s–**$\mathbf{IMELL_0^{\multimap}}$ provability. The reduction is based on the following lemma.

**Lemma 11.** *Let $\Gamma \vdash A$ and $\Gamma' \vdash A'$ be* **IMELL** *sequents, $p$ be an atomic proposition, and $F$ be a formula such that $\Gamma = \Gamma'[p := F]$ and $A = A'[p := F]$. Then, $\Gamma \vdash A$ is provable if and only if $!(p \multimap F), !(F \multimap p), \Gamma' \vdash A'$.*

*Proof.* If $\Gamma \vdash A$ is provable, the provability of $!(p \multimap F), !(F \multimap p), \Gamma' \vdash A'$ may be established by induction on the derivation of $\Gamma \vdash A$, the two exponential formulas allowing the negative and positive occurrences of $F$ to be replaced by $p$.

On the other hand, if $!(p \multimap F), !(F \multimap p), \Gamma' \vdash A'$ is provable, we have that $!(F \multimap F), !(F \multimap F), \Gamma'[p := F] \vdash A'[p := F]$ is provable. This is, by hypothesis, $!(F \multimap F), !(F \multimap F), \Gamma \vdash A$. Then, the result follows by a contraction followed by a cut. $\square$

Then the reduction proceeds as follows.

**Lemma 12.** *Let $!\Sigma, \Gamma \vdash A$ be an* $\mathbf{IMELL_0^{\multimap}}$ *sequent. Then there exists an* $\mathbf{IMELL_0^{\multimap}}$ *sequent $!\Sigma', \Gamma' \vdash a$ such that $\Gamma'$ contains only atomic formulas, and $!\Sigma, \Gamma \vdash A$ is provable if and only if $!\Sigma', \Gamma' \vdash a$ is provable.*

*Proof.* The proof proceeds by induction on the number of non-atomic formulas in $\Gamma, A$. Suppose there is such a non-atomic formulas $B$ in $\Gamma, A$, and let $\Gamma_1, A_1$ be such that $(\Gamma_1, A_1)[p := B] = (\Gamma, A)$, where $p$ is a fresh variable that occurs in $(\Gamma_1, A_1)$. Then, by Lemma 11, $!\Sigma, !(p \multimap B), !(B \multimap p), \Gamma_1 \vdash A_1$ is provable if and only if $!\Sigma, \Gamma \vdash A$ is, and the result follows by induction hypothesis. $\square$

**Lemma 13.** *Let* $!\Sigma, \Gamma \vdash a$ *be an* $\mathbf{IMELL_0^{\multimap}}$ *sequent such that* $\Gamma$ *contains only atomic formulas. Then there exists a sequent* $!\Sigma', \Gamma \vdash a$ *such that all the formulas belonging to* $\Sigma'$ *contain at most two occurrences of "*$\multimap$*".*

*Proof.* The proof proceeds by induction on the sum of the lengths of the formulas that contain more than two occurrences of "$\multimap$". Let $!\Sigma = !\Sigma_1, !A_1$, where $A_1$ contains more than two occurrences of "$\multimap$". Then there exist two atomic formulas $a$ and $b$ in $A_1$, and a formula $A_1'$ such that $A_1 = A_1'[p := (a \multimap b)]$, where $p$ is a fresh variable occurring in $A_1'$. Then, by Lemma 11, we have that $!\Sigma_1, !A_1', !(p \multimap (a \multimap b)), !((a \multimap b) \multimap p), \Gamma \vdash a$ is provable if and only if $!\Sigma, \Gamma \vdash a$ is, and the result follows by induction hypothesis. $\square$

**Proposition 4.** $\mathbf{IMELL_0^{\multimap}}$ *is decidable if and only if* $\mathbf{s\text{–}IMELL_0^{\multimap}}$ *is.*

*Proof.* Let $!\Sigma, \Gamma \vdash A$ be an $\mathbf{IMELL_0^{\multimap}}$ sequent. By Lemmas 12 and 13, there exist a sequent $!\Sigma', \Gamma' \vdash a$ such that all the formulas in $\Gamma'$ are atomic, all the formulas in $!\Sigma'$ contain at most two occurrences of "$\multimap$", and $!\Sigma', \Gamma' \vdash a$ is provable if and only if $!\Sigma, \Gamma \vdash A$ is. Then, the only reason why $!\Sigma', \Gamma' \vdash a$ would not be a $\mathbf{s\text{–}IMELL_0^{\multimap}}$ sequent is that $!\Sigma'$ would contain formulas of the form $!b$ (for $b$ atomic). These formulas may be replaced by $!((b \multimap b) \multimap b)$. $\square$

# 5 Relating VATA and MELL

Let $\{a_0, \dots a_{k-1}\}$ be a fixed enumeration of atomic formulas. For a multiset $\Gamma$ whose elements are in the enumeration, we write $|\Gamma|$ the corresponding vector of $\mathbb{N}^k$. Conversely, for $\mathbf{x} \in \mathbb{N}^k$, we write $\Gamma_{\mathbf{x}}$ the corresponding multiset.

## 5.1 From VATA to MELL

Let $\mathcal{A} = \langle \mathcal{F}, Q, \{(q_f, \mathbf{0})\}, \Delta\rangle$ be a $k$-VATA in normal form. We define the set of atomic formulas $A$ of to be $Q \cup \{a_0, \dots a_{k-1}\}$, where $a_0, \dots a_{k-1}$ are fresh symbols. Then we define the set of formulas $\Sigma$ to be $\Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ where:

$$\begin{aligned}
&\Sigma_0 = \{a_i \multimap q \mid f \longrightarrow (q, \mathbf{e}_i) \in \Delta\}\\
&\Sigma_1 = \{(a_i \multimap q_0) \multimap q \mid \\
&\qquad f((q_0, \mathbf{x}_0)) \longrightarrow (q, \mathbf{x}_0 - \mathbf{e}_i) \in \Delta\}\\
&\Sigma_2 = \{q_0 \multimap (q_1 \multimap q) \mid \\
&\qquad f((q_0, \mathbf{x}_0), (q_1, \mathbf{x}_1)) \longrightarrow (q, \mathbf{x}_0 + \mathbf{x}_1) \in \Delta\}
\end{aligned}$$

**Proposition 5.** $\mathcal{L}(\mathcal{A}) \neq \varnothing$ *iff* $!\Sigma \vdash q_f$

*Proof.* First, by induction on the structure of $t$, we prove that if $t \longrightarrow^*_{\mathcal{A}} (q, \mathbf{x})$ then $!\Sigma, \Gamma_{\mathbf{x}} \vdash q$. If the head symbol of $t$ is unary (resp. binary), the result is easily obtained with rule $(T_2)$ (resp. rule $(T_3)$) and with the induction hypothesis; the remaining case is: $f \longrightarrow_{\mathcal{A}} (q, \mathbf{e}_i)$, then with rule $(T_0)$, we have $!\Sigma, a_i \vdash a_i$ and then with rule $(T_1)$, we get $!\Sigma, a_i \vdash q$ because $a_i \multimap q \in \Sigma$.

Secondly, we can observe that each provable sequent $!\Sigma, \Gamma \vdash \alpha$ is either of the form $!\Sigma, a_i \vdash a_i$ or is such that all elements of $\Gamma$ are in $\{a_0, \dots a_{k-1}\}$ and $\alpha \in Q$ (easy induction on the derivation). We can prove by induction on the derivation that if $!\Sigma, \Gamma \vdash q$, then there is some $t$ such that $t \longrightarrow^*_{\mathcal{A}} (q, |\Gamma|)$. Actually, the previous remark ensures that $|\Gamma|$ is always well defined. $\square$

## 5.2 From MELL to VATA

Let $!\Sigma, \Gamma \vdash a_0$ be a sequent of $\mathbf{s\text{–}IMELL_0^{\multimap}}$ and $\{a_0, \dots, a_{k-1}\}$ a fixed enumeration of the atomic formulas of the sequent.

We can write $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ where $\Sigma_0$ (resp. $\Sigma_1$, resp. $\Sigma_2$) contains only formulas of the form $a_j \multimap a_l$ (resp. $(a_j \multimap a_l) \multimap a_m$, resp. $a_j \multimap (a_l \multimap a_m)$) and we fix an enumeration of these formulas : $\{f_0, \dots, f_{n_0-1}\}$ (resp. $\{g_0, \dots, g_{n_1-1}\}$, resp. $\{h_0, \dots, h_{n_2-1}\}$)

We define a $k$-VATA $\mathcal{A} = \langle \mathcal{F}, Q, C_f, \Delta\rangle$ as follows:

- $\mathcal{F} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ with

$$\begin{aligned}
&\mathcal{F}_0 = \{c_0, \dots, c_{k-1}\}\\
&\mathcal{F}_1 = \{f_0, \dots, f_{n_0-1}\} \cup \{g_0, \dots, g_{n_1-1}\}\\
&\mathcal{F}_2 = \{h_0, \dots, h_{n_2-1}\}\\
&\mathcal{F}_n = \varnothing \text{ if } n > 2;
\end{aligned}$$

- $Q = \{q_0, \dots, q_{k-1}\}$;
- there is only one final state: $C_f = \{(q_0, |\Gamma|)\}$
- there is exactly one production for each symbol of the alphabet:
  - $c_i \longrightarrow (q_i, \mathbf{e}_i)$;
  - if $f_i = a_j \multimap a_m$ then $f_i((q_j, \mathbf{x})) \longrightarrow (q_m, \mathbf{x})$;
  - if $g_i = (a_j \multimap a_l) \multimap a_m$ then
  $$g_i((q_l, \mathbf{x})) \longrightarrow (q_m, \mathbf{x} - \mathbf{e}_j)\,;$$
  - if $h_i = a_j \multimap (a_l \multimap a_m)$ then
  $$h_i((q_j, \mathbf{x}_0), (q_l, \mathbf{x}_1)) \longrightarrow (q_m, \mathbf{x}_0 + \mathbf{x}_1).$$

**Proposition 6.** *The sequent* $!\Sigma, \Gamma \vdash a_0$ *is provable in* **s–IMELL**$_{\mathbf{0}}^{-\circ}$ *if and only if* $\mathcal{L}(\mathcal{A}) \neq \emptyset$.

*Proof.* By induction on the derivation, we prove that: if the sequent $!\Sigma, \Gamma \vdash a_i$ is provable in **IMELL**$_{\mathbf{0}}^{-\circ}$ then there is a term $t$ such that $t \longrightarrow^*_{\mathcal{A}} (q_i, |\Gamma|)$.

$(T_0)$ $\dfrac{}{!\Sigma, a_i \vdash a_i}$ then $c_i \longrightarrow_{\mathcal{A}} (q_i, e_i)$.

$(T_1)$ $\dfrac{!\Sigma, \Gamma \vdash a_j}{!\Sigma, \Gamma \vdash a_i}$ with $f_n = a_j \multimap a_i \in \Sigma$ for some $n \in n_0$. By induction, there is a $t$ such that $t \longrightarrow^*_{\mathcal{A}} (q_j, |\Gamma|)$. Then $f_n(t) \longrightarrow^*_{\mathcal{A}} (q_i, |\Gamma|)$.

$(T_2)$ $\dfrac{!\Sigma, \Gamma, a_j \vdash a_l}{!\Sigma, \Gamma \vdash a_i}$ with $g_n = (a_j \multimap a_l) \multimap a_i \in \Sigma$ for some $n \in n_1$. By induction, there is a $t$ such that $t \longrightarrow^*_{\mathcal{A}} (q_l, |\Gamma| + e_j)$. Then $g_n(t) \longrightarrow^*_{\mathcal{A}} (q_i, |\Gamma|)$.

$(T_3)$ $\dfrac{!\Sigma, \Gamma, \vdash a_j \quad !\Sigma, \Gamma, \vdash a_l}{!\Sigma, \Gamma_1, \Gamma_2 \vdash a_i}$ with $h_n = a_j \multimap (a_l \multimap a_i) \in \Sigma$ for some $n \in n_2$. By induction, there are $t$ and $u$ such that $t \longrightarrow^*_{\mathcal{A}} (q_j, |\Gamma_1|)$ and $u \longrightarrow^*_{\mathcal{A}} (q_l, |\Gamma_2|)$. Then $h_n(t, u) \longrightarrow^*_{\mathcal{A}} (q_i, |\Gamma_1, \Gamma_2|)$ (because $|\Gamma_1, \Gamma_2| = |\Gamma_1| + |\Gamma_2|$)

Conversely, by induction on the structure of the term $t$: if there is term $t$ such that $t \longrightarrow^*_{\mathcal{A}} (q_i, \mathbf{x})$ then the sequent $!\Sigma, \Gamma_{\mathbf{x}} \vdash a_i$ is provable in **IMELL**$_{\mathbf{0}}^{-\circ}$.

- If $t = c_n \longrightarrow_{\mathcal{A}} (q_i, \mathbf{e}_i)$ then $n = i$ and, with rule $(T_0)$, the sequent $!\Sigma, \Gamma_{\mathbf{e}_i} \vdash a_i$ is provable.

- If $t = f_n(u) \longrightarrow^*_{\mathcal{A}} (q_i, \mathbf{x})$ with $f_n = a_j \multimap a_i \in \Sigma$ and $u \longrightarrow^*_{\mathcal{A}} (q_j, \mathbf{x})$ then, by induction hypothesis, $!\Sigma, \Gamma_{\mathbf{x}} \vdash a_j$ is provable and with rule $(T_1)$, the sequent $!\Sigma, \Gamma_{\mathbf{x}} \vdash a_i$ is provable.

- If $t = g_n(u) \longrightarrow^*_{\mathcal{A}} (q_i, \mathbf{x})$ with $g_n = (a_j \multimap a_l) \multimap a_i \in \Sigma$ and $u \longrightarrow^*_{\mathcal{A}} (q_l, \mathbf{x} + \mathbf{e}_j)$, then, by induction hypothesis, $!\Sigma, \Gamma_{\mathbf{x}}, a_j \vdash a_l$ is provable and then with rule $(T_2)$, the sequent $!\Sigma, \Gamma_{\mathbf{x}} \vdash a_i$ is provable.

- If $t = h_n(u, v) \longrightarrow^*_{\mathcal{A}} (q_i, \mathbf{x})$ with $h_n = a_j \multimap a_l \multimap a_i \in \Sigma$, $u \longrightarrow^*_{\mathcal{A}} (q_j, \mathbf{x}_0)$ and $v \longrightarrow^*_{\mathcal{A}} (q_l, \mathbf{x}_1)$ with $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1$, then, by induction hypothesis, $!\Sigma, \Gamma_{\mathbf{x}_0} \vdash a_j$ and $!\Sigma, \Gamma_{\mathbf{x}_1} \vdash a_l$ and then with rule $(T_3)$, the sequent $!\Sigma, \Gamma_{\mathbf{x}} \vdash a_i$ is provable. □

## References


[1] A. Asperti, G.-L. Ferrari, and R. Gorrieri. Implicative formulae in the "proofs as computations" analogy. In *Proceedings of the 17th ACM Symposium on Priciples of Programming Langages*, pages 59–71, San Francisco, 1990.

[2] C. Brown. *Linear Logic and Petri Nets: Categories, Algebra and Proof.* PhD thesis, University of Edinburgh, 1990.

[3] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 1997. release October, 1rst 2002.

[4] B. Farwer. A linear logic view of object petri nets. *Fundamenta Informaticae*, 37(3):225–246, 1999.

[5] J.-Y. Girard. Linear logic. *Theoritical Computer Science*, 50(1):1–102, 1987.

[6] C. Gunter and V. Gehlot. Nets as tensor theories. In *Proceedings of the 10th International Conference on Application and Theory of Petri Nets, 1989, Bonn, Germany*, pages 174–191, 1989.

[7] J. E. Hopcroft and J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.

[8] M. I. Kanovich. Linear Logic as a logic of computations. *Annals of Pure and Applied Logic*, 67:183–212, 1994.

[9] M. I. Kanovich. Petri nets, Horn programs, Linear Logic and vector games. *Annals of Pure and Applied Logic*, 75:107–135, 1995.

[10] M. I. Kanovich. Linear Logic automata. *Annals of Pure and Applied Logic*, 78:147–188, 1996.

[11] R. Karp and R. Miller. Parallel program schemata. *Journal of Computer and Systems Sciences*, 3(2):147–195, 1969.

[12] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proc. 14th Ann. Symp. on Theory of Computing*, pages 267–291, 1982.

[13] N. Martí-Oliet and J. Meseguer. From petri nets to linear logic. In *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, 1989.

[14] E. W. Mayr. An algorithm for the general petri net reachability problem. *Siam J. Comput.*, 13:441–460, 1984.

[15] M. Okada. Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic. *Theoretical Computer Science*, 227(1–2):333–396, 1999.

[16] A. Troelstra. *Lectures on Linear Logic*, volume 29 of *CSLI Lecture Notes*. Center for the Study of Language and Information, 1992.

[17] K. N. Verma and J. Goubault-Larrecq. Karp-miller trees for a branching extension of vass. Technical Report LSV-04-3, ENS Cachan, 2004.