The undecidability of second order multiplicative linear logic

Yves Lafont*

Laboratoire de Mathématiques Discrètes, UPR 9016 du CNRS,
163 Avenue de Luminy, Case 930, 13288 Marseille Cedex 9, France

AND

Andre Scedrov†

Department of Mathematics, University of Pennsylvania,
Philadelphia, PA 19104-6395, USA

February 20, 1996

Abstract

The multiplicative fragment of second order propositional linear logic is shown to be undecidable.

Introduction

Decision problems for propositional (quantifier-free) linear logic were first studied by Lincoln *et al.* [LMSS]. In referring to linear logic fragments, let **M** stand for multiplicatives, **A** for additives, **E** for exponentials (or modalities), **1** for first order quantifiers, **2** for second order propositional quantifiers, and **I** for the "intuitionistic" version.

In [LMSS] it was shown that full propositional linear logic is undecidable and that **MALL** is PSPACE-complete. The main problems left open in [LMSS] were the NP-completeness of **MLL**, the decidability of **MELL**, and the decidability of various fragments of propositional linear logic without exponentials but extended with second order propositional quantifiers. The decision problem for **MELL** is still open, but almost all the other problems have been solved:

- The NP-completeness of **MLL** has been obtained by Kanovich [K1]. Moreover, Lincoln and Winkler [LW] have established that **MLL** without proper atoms is already NP-complete.
- **MALL1** with function symbols is NEXPTIME-complete: the hardness has been obtained by Lincoln and Scedrov [LSv] and the membership, and hence completeness, by Lincoln and Shankar [LSr].
- The undecidability of **MLL12** and **MALL12** with function symbols has been proved by Amiot [A]. Lincoln *et al.* [LSS] have recently showed the undecidability of **IMLL2** and **IMALL2**, and the undecidability of **MLL12** and **MALL12** without function symbols. Lafont [Lt] then proved the undecidability of **MALL2**.

Here we use a refinement of the methods from [LSS] and [Lt] to establish the undecidability of **MLL2**.

*E-mail address: `lafont@lmd.univ-mrs.fr`

†E-mail address: `scedrov@cis.upenn.edu`. WWW: `http://www.cis.upenn.edu/~andre`. Partially supported by NSF Grant CCR-94-00907, by ONR Grant N00014-92-J-1916, and by CNRS. Scedrov is an American Mathematical Society Centennial Research Fellow.

In [LSS] a simple encoding of second order propositional intuitionistic logic in **IMLL2** is described, which uses the formula $\forall\alpha.\,(\alpha \multimap \alpha\otimes\alpha)$ to encode the structural rule of contraction. This encoding does not yield the undecidability in the classical case. Instead, a weaker formula $\forall\alpha.\,(\alpha \,\&\, \mathbf{1} \multimap \alpha\otimes\alpha)$ is used in [Lt]. The additive formula $\alpha \,\&\, \mathbf{1}$ serves as an approximation of the exponential $!\alpha$ in an encoding of register machines in propositional linear logic, which comes from [K2] and essentially goes back to [LMSS]. In [Lt] the faithfulness of the encoding is obtained as an application of the soundness theorem for the phase semantics of linear logic [G1, G2]. The present argument follows a similar pattern. However, the formula $\alpha \,\&\, \mathbf{1}$ is replaced by a multiplicative approximation $\alpha\otimes(\alpha\multimap\mathbf{1})$ and a new encoding of register machines is presented in order to avoid the use of additives in the encoding of the zero-test instruction.

The first two sections of this paper recall the syntax and semantics of second order multiplicative linear logic. For more information, we refer to [G1, G2, Sv1, Sv2].

1 Syntax of IMLL2 and MLL2

The connectives of **IMLL2** are the multiplicatives $\otimes$, $\mathbf{1}$, $\multimap$ and the second order propositional quantifiers $\forall$, $\exists$. The sequents are of the form $\Gamma\vdash A$ where Γ is a multiset of formulas and A is a formula. The rules are:

$$\frac{}{A\vdash A}\qquad \frac{\Gamma\vdash A\quad \Delta, A\vdash C}{\Gamma,\Delta\vdash C}$$

$$\frac{\Gamma\vdash A\quad \Delta\vdash B}{\Gamma,\Delta\vdash A\otimes B}\qquad \frac{\Gamma, A, B\vdash C}{\Gamma, A\otimes B\vdash C}\qquad \frac{}{\vdash\mathbf{1}}\qquad \frac{\Gamma\vdash C}{\Gamma,\mathbf{1}\vdash C}\qquad \frac{\Gamma, A\vdash B}{\Gamma\vdash A\multimap B}\qquad \frac{\Gamma\vdash A\quad \Delta, B\vdash C}{\Gamma,\Delta, A\multimap B\vdash C}$$

$$\frac{\Gamma\vdash A[B]}{\Gamma\vdash\exists\alpha.\,A[\alpha]}\qquad \frac{\Gamma, A[\beta]\vdash C}{\Gamma,\exists\alpha.\,A[\alpha]\vdash C}\qquad \frac{\Gamma\vdash A[\beta]}{\Gamma\vdash\forall\alpha.\,A[\alpha]}\qquad \frac{\Gamma, A[B]\vdash C}{\Gamma,\forall\alpha.\,A[\alpha]\vdash C}$$

In the $\exists$-left and $\forall$-right rules, β is a fresh propositional variable. In fact, $\otimes$, $\mathbf{1}$, and $\exists$ are definable in terms of $\multimap$ and $\forall$ by

$$A\otimes B\equiv\forall\alpha.\,((A\multimap B\multimap\alpha)\multimap\alpha),\qquad \mathbf{1}\equiv\forall\alpha.\,(\alpha\multimap\alpha),\qquad \exists\alpha.\,A[\alpha]\equiv\forall\beta.\,((\forall\alpha.\,(A[\alpha]\multimap\beta))\multimap\beta),$$

where β is a fresh propositional variable. The connectives of **MLL2** are the multiplicatives $\otimes$, $⅋$, $\mathbf{1}$, $\bot$ and the second order propositional quantifiers $\forall$, $\exists$. Linear negation $A^\bot$ is given for propositional variables or atoms. It is extended to all formulas by $\alpha^{\bot\bot}=\alpha$ and by

$$(A\otimes B)^\bot = A^\bot ⅋ B^\bot,\qquad (A ⅋ B)^\bot = A^\bot\otimes B^\bot,\qquad \mathbf{1}^\bot=\bot,\qquad \bot^\bot=\mathbf{1},$$

$$(\exists\alpha.\,A[\alpha])^\bot=\forall\alpha.\,A[\alpha]^\bot,\qquad (\forall\alpha.\,A[\alpha])^\bot=\exists\alpha.\,A[\alpha]^\bot.$$

The sequents are of the form $\vdash\Gamma$ where Γ is a multiset of formulas, and the rules are:

$$\frac{}{\vdash A, A^\bot}\qquad \frac{\vdash A,\Gamma\quad \vdash A^\bot,\Delta}{\vdash\Gamma,\Delta}$$

$$\frac{\vdash A,\Gamma\quad \vdash B,\Delta}{\vdash A\otimes B,\Gamma,\Delta}\qquad \frac{\vdash A, B,\Gamma}{\vdash A ⅋ B,\Gamma}\qquad \frac{}{\vdash\mathbf{1}}\qquad \frac{\vdash\Gamma}{\vdash\bot,\Gamma}$$

$$\frac{\vdash A[\beta],\Gamma}{\vdash\forall\alpha.\,A[\alpha],\Gamma}\qquad \frac{\vdash A[B],\Gamma}{\vdash\exists\alpha.\,A[\alpha],\Gamma}$$

In the $\forall$ rule, β is a fresh propositional variable. Linear implication is defined by $A\multimap B = A^\bot ⅋ B$. An intuitionistic sequent $A_1,\ldots,A_n\vdash C$ is interpreted as $\vdash A_1^\bot,\ldots,A_n^\bot, C$, so that **MLL2** can be seen as a (non-conservative) extension of **IMLL2**.

2 Phase semantics of MLL2

A *phase space* [G1, G2] is a commutative monoid M endowed with a subset $\bot$ of M. If $X, Y \subset M$ define

$$XY = \{xy;\, x \in X \text{ and } y \in Y\}, \qquad X \multimap Y = \{z \in M;\, xz \in Y \text{ for all } x \in X\}, \qquad X^\bot = X \multimap \bot.$$

Also, if $x \in M$ and $Y \subset M$, it is convenient to write xY for $\{x\}Y$ and $x \multimap Y$ for $\{x\} \multimap Y$. An $X \subset M$ is *closed* iff $X^{\bot\bot} = X$. It is easy to see that $X \subset X^{\bot\bot}$ and $X^\bot = X^{\bot\bot\bot}$ for any $X \subset M$, so that $X^\bot$ is always closed. In particular $\bot = \{1\}^\bot$ is closed. Also, $X \multimap Y$ is closed whenever Y is closed, and any intersection of closed sets is closed. In particular, $X^{\bot\bot}$ is the intersection of all closed sets containing X, that is, the smallest closed set containing X. Define $\mathbf{1} = \bot^\bot = \{1\}^{\bot\bot}$. For any closed X, Y let

$$X \otimes Y = (XY)^{\bot\bot}, \qquad X \parr Y = (X^\bot \otimes Y^\bot)^\bot = (X^\bot Y^\bot)^\bot.$$

From these definitions one deduces easily for any $X, Y \subset M$,

$$X^{\bot\bot} \otimes Y^{\bot\bot} = (XY)^{\bot\bot}, \qquad X^\bot \parr Y^{\bot\bot} = X^{\bot\bot} \multimap Y^{\bot\bot} = X \multimap Y^{\bot\bot} = Y^\bot \multimap X^\bot.$$

In particular, $X^\bot \parr Y = X \multimap Y$ whenever Y is closed, which justifies the notation a posteriori. Furthermore, for any $X_1, \ldots, X_n, Y \subset M$, one gets

$$X_1^{\bot\bot} \otimes \cdots \otimes X_n^{\bot\bot} = (X_1 \cdots X_n)^{\bot\bot}, \qquad X_1^{\bot\bot} \otimes \cdots \otimes X_n^{\bot\bot} \multimap Y^{\bot\bot} = X_1 \cdots X_n \multimap Y^{\bot\bot}.$$

If a closed set $\alpha^\bullet$ is given for each atom α occurring in a formula A of **MLL**, one defines a closed set $A^\bullet$ by applying the above definitions inductively, and this interpretation is extended to **MLL2** by

$$(\forall \alpha.\, A[\alpha])^\bullet = \bigcap_{X \subset M} A^\bullet_X$$

where $A^\bullet_X$ is the interpretation defined by $A^\bullet_X = X^{\bot\bot}$ for $A[\alpha] = \alpha$. One says that M *satisfies* a formula A when $1 \in A^\bullet$.

Theorem 1 *(Girard [G1]) If A is provable in* **MLL2***, then M satisfies A.*

We shall not need the converse, which does not hold for this restricted notion of second order model. An easy consequence of Theorem 1 is that $A_1^\bullet \cdots A_n^\bullet \subset B^\bullet$ whenever $A_1, \ldots, A_n \vdash B$ is provable in **MLL2**. This holds more generally whenever $\Gamma, A_1, \ldots, A_n \vdash B$ is provable and M satisfies all formulas in Γ.

3 Encoding a restricted kind of contraction

Let A^n denote the tensor product $A \otimes \cdots \otimes A$ of n occurrences of A. Let $\#A$ denote $A \otimes (A \multimap \mathbf{1})$. If Γ is a multiset $A_1, \ldots, A_n$ let $\#\Gamma$ denote $\#A_1, \ldots, \#A_n$. Note that the sequent $\#A \vdash \mathbf{1}$ is always provable. Furthermore, $A \vdash \#A$ is provable whenever $A \vdash \mathbf{1}$ is provable. In particular, $\#\mathbf{1}$ is equivalent to $\mathbf{1}$. The following formula encodes a restricted kind of contraction:

$$\rho = \forall \alpha.\, \forall \beta.\, (\#\alpha \otimes \#\beta \multimap \alpha^2 \otimes \#\beta \otimes \beta).$$

Lemma 1 *The following rules are derivable in* **IMLL2***:*

$$\frac{\Gamma \vdash C}{\#A, \Gamma \vdash C} \qquad \frac{\Gamma \vdash C}{\rho, \Gamma \vdash C} \qquad \frac{\rho^2, \#A, A, \Gamma \vdash C}{\rho^2, \#A, \Gamma \vdash C}$$

Proof. Since $\#A \vdash \mathbf{1}$ is provable, the first rule is derivable. The sequents $\rho \vdash \#\mathbf{1} \otimes \#\mathbf{1} \multimap \mathbf{1}^2 \otimes \#\mathbf{1} \otimes \mathbf{1}$ and $\rho \vdash \#\rho \otimes \#A \multimap \rho^2 \otimes \#A \otimes A$ are provable by definition of ρ. But $\#\mathbf{1} \otimes \#\mathbf{1} \multimap \mathbf{1}^2 \otimes \#\mathbf{1} \otimes \mathbf{1}$ is clearly equivalent to $\mathbf{1}$, so that $\rho \vdash \mathbf{1}$ is provable, and also $\rho \vdash \#\rho$ by the above remark. It follows that the last two rules are derivable. **Q.e.d.**

Lemma 2 *ρ is satisfied by any phase space M satisfying the following conditions:*

1. *The unit 1 is the only element of the closed set* $\mathbf{1}$*,*
2. *The unit 1 is the only invertible element in M.*

Proof. By definition of the interpretation, one has

$$\rho^\bullet = \bigcap_{X,Y\subset M} X(X \multimap \mathbf{1})Y(Y \multimap \mathbf{1}) \multimap (X^2Y^2(Y \multimap \mathbf{1}))^{\perp\perp}.$$

Let $X, Y \subset M$. If $x \in X$ and $z \in X \multimap \mathbf{1}$, then $xz \in \mathbf{1}$, so that $xz = 1$ by the first condition and $x = z = 1$ by the second one. The same holds for Y, so that

$$X(X \multimap \mathbf{1})Y(Y \multimap \mathbf{1}) \subset X^2Y^2(Y \multimap \mathbf{1}) \subset (X^2Y^2(Y \multimap \mathbf{1}))^{\perp\perp},$$

which means that ρ is satisfied. **Q.e.d.**

Note that the second condition is automatically satisfied if M is a free commutative monoid.

4 Two-counter machines

If $i \le j$ are integers, we write $[i, j]$ for the set $\{i, i{+}1, \dots, j\}$. A *deterministic two-counter machine* is given by an integer $m > 0$ and a map

$$\tau : [1, m] \to \{+\} \times \{1, 2\} \times [0, m] \,\cup\, \{-\} \times \{1, 2\} \times [0, m] \times [0, m].$$

If, at a given time, the machine is in the *configuration* $(i, p, q) \in [0, m] \times \mathbb{N} \times \mathbb{N}$, its next configuration is:

- $(j, p{+}1, q)$ if $\tau(i) = (+, 1, j)$ (increment first counter),
- $(j, p{-}1, q)$ if $\tau(i) = (-, 1, j, k)$ and $p > 0$ (decrement first counter),
- (k, p, q) if $\tau(i) = (-, 1, j, k)$ and $p = 0$ (test that first counter is zero),
- $(j, p, q{+}1)$ if $\tau(i) = (+, 2, j)$ (increment second counter),
- $(j, p, q{-}1)$ if $\tau(i) = (-, 2, j, k)$ and $q > 0$ (decrement second counter),
- (k, p, q) if $\tau(i) = (-, 2, j, k)$ and $q = 0$ (test that second counter is zero).

Since τ is only defined for $i \in [1, m]$, the machine stops when $i = 0$. One says that a configuration (i, p, q) is *accepted* by the machine if, starting from (i, p, q), the machine eventually stops on $(0, 0, 0)$.

Theorem 2 *(Lambek [Lk], Minsky [M]) There is a deterministic two-counter machine for which the set of accepted configurations is not recursive.*

In the following sections, such a machine is used to show that both **IMLL2** and **MLL2** are undecidable.

5 Encoding two-counter machines

Let $a, c_0, c_1, \dots, c_m$ be atoms, and let $\varphi[\gamma]$ and $\psi[\gamma]$ be two formulas with one free propositional variable γ. If $i \in [0, m]$ and $p, q \in \mathbb{N}$, the configuration (i, p, q) will be represented by the formula $c_i \otimes \varphi[a^p] \otimes \psi[a^q]$. In particular, the configuration $(0, 0, 0)$ will be represented by $\omega = c_0 \otimes \varphi[\mathbf{1}] \otimes \psi[\mathbf{1}]$. Let Θ be the multiset consisting of the formulas

$$\forall\gamma.\,(c_i \otimes \varphi[\gamma] \multimap c_j \otimes \varphi[a \otimes \gamma]) \qquad \text{for } i \in [1, m] \text{ such that } \tau(i) = (+, 1, j),$$

$$\left\{\begin{array}{l} \forall\gamma.\,(c_i \otimes \varphi[a \otimes \gamma] \multimap c_j \otimes \varphi[\gamma]) \\ c_i \otimes \varphi[\mathbf{1}] \multimap c_k \otimes \varphi[\mathbf{1}] \end{array}\right. \qquad \text{for } i \in [1, m] \text{ such that } \tau(i) = (-, 1, j, k),$$

and similarly for the second counter, using ψ instead of φ.

Theorem 3 *There are formulas $\varphi[\gamma]$ and $\psi[\gamma]$ such that, for any $(i,p,q) \in [0,m] \times \mathbb{N} \times \mathbb{N}$, the following assertions are equivalent:*

1. *The configuration (i,p,q) is accepted by the machine,*
2. *The sequent $\rho^2, \#\Theta, c_i, \varphi[a^p], \psi[a^q] \vdash \omega$ is provable in* **IMLL2**,
3. *The latter sequent is provable in* **MLL2**.

The assertion that (1) implies (2) holds independently of the choice of $\varphi[\gamma]$ and $\psi[\gamma]$. This is proved by a straightforward induction on the length of the computation. The first two rules of Lemma 1 are used in the base case. The third rule of Lemma 1 is used in the induction step. For instance, an increment transition is handled as follows:

$$\dfrac{\dfrac{\dfrac{}{c_i \vdash c_i} \quad \dfrac{}{\varphi[a^p] \vdash \varphi[a^p]}}{c_i,\, \varphi[a^p] \vdash c_i \otimes \varphi[a^p]} \quad \dfrac{\begin{array}{c}\vdots\\ \rho^2,\, \#\Theta,\, c_j,\, \varphi[a^{p+1}],\, \psi[a^q] \vdash \omega\end{array}}{\rho^2,\, \#\Theta,\, c_j \otimes \varphi[a^{p+1}],\, \psi[a^q] \vdash \omega}}{\dfrac{\dfrac{\rho^2,\, \#\Theta,\, c_i \otimes \varphi[a^p] \multimap c_j \otimes \varphi[a^{p+1}],\, c_i,\, \varphi[a^p],\, \psi[a^q] \vdash \omega}{\rho^2,\, \#\Theta,\, \forall\gamma.\, (c_i \otimes \varphi[\gamma] \multimap c_j \otimes \varphi[a \otimes \gamma]),\, c_i,\, \varphi[a^p],\, \psi[a^q] \vdash \omega}}{\rho^2,\, \#\Theta,\, c_i,\, \varphi[a^p],\, \psi[a^q] \vdash \omega}}$$

Also, it is clear that (2) implies (3). The next section is devoted to the hard part of the proof, namely that (3) implies (1) for some appropriate $\varphi[\gamma]$ and $\psi[\gamma]$,

$$\varphi[\gamma] = \gamma^3 \otimes (\gamma \multimap d) \otimes (\gamma \multimap e) \multimap f, \qquad \psi[\gamma] = \gamma^3 \otimes (\gamma \multimap d) \otimes (\gamma \multimap e) \multimap g,$$

where d, e, f, g are new atoms.

6 Faithfulness of the encoding

In order to show that the above encoding is faithful, consider the free commutative monoid M generated by the atoms $a, c_0, c_1, \ldots, c_m, \overline{d}, \overline{e}, \overline{f}, \overline{g}$ and by the infinite families (d_n), (e_n), (f_n), (g_n), all indexed by $\mathbb{N}$. The structure of phase space is given by

$$\begin{aligned} \perp \quad = \quad & \{c_i f_p g_q;\ (i,p,q) \text{ is accepted by the machine}\} \cup \{\overline{d} d_n a^n;\ n \in \mathbb{N}\} \cup \{\overline{e} e_n a^n;\ n \in \mathbb{N}\} \cup \\ & \{\overline{f} f_n d_{n+p} e_{n+q} a^{3n+p+q};\ n,p,q \in \mathbb{N}\} \cup \{\overline{g} g_n d_{n+p} e_{n+q} a^{3n+p+q};\ n,p,q \in \mathbb{N}\}. \end{aligned}$$

The interpretation is defined by

$$a^\bullet = \{a\}^{\perp\perp}, \qquad c_i{}^\bullet = \{c_i\}^{\perp\perp}, \qquad d^\bullet = \{\overline{d}\}^\perp = \{d_n a^n;\ n \in \mathbb{N}\}, \qquad e^\bullet = \{\overline{e}\}^\perp = \{e_n a^n;\ n \in \mathbb{N}\},$$

$$f^\bullet = \{\overline{f}\}^\perp = \{f_n d_{n+p} e_{n+q} a^{3n+p+q};\ n,p,q \in \mathbb{N}\}, \qquad g^\bullet = \{\overline{g}\}^\perp = \{g_n d_{n+p} e_{n+q} a^{3n+p+q};\ n,p,q \in \mathbb{N}\}.$$

Note that the singleton $\{a^n\} = \{\overline{d} d_n\}^\perp$ is closed for any $n \in \mathbb{N}$. In particular, $\{1\} = \{a^0\}$ is closed and $\mathbf{1} = \perp^\perp = \{1\}$, so that ρ is satisfied by Lemma 2.

Lemma 3 *Let $X \subset M$. If X is of the form $\{a^n\}$, then $\varphi_X^\bullet = \{f_n\}$ and $\psi_X^\bullet = \{g_n\}$, else $\varphi_X^\bullet = \psi_X^\bullet = M$.*

Proof. Let us discuss $\varphi_X^\bullet$; the case of $\psi_X^\bullet$ is similar. By definition of the interpretation, one has

$$\varphi_X^\bullet = X^3 (X \multimap d^\bullet)(X \multimap e^\bullet) \multimap f^\bullet.$$

If $x \in M$, the closed set $x \multimap d^\bullet$ is

$$\{d_{n+p} a^p;\ p \in \mathbb{N}\} \text{ if } x \text{ is of the form } a^n, \qquad \{a^n\} \text{ if } x \text{ is of the form } d_{n+p} a^p, \qquad \emptyset \text{ otherwise.}$$

Similarly, the closed set $x \multimap e^\bullet$ is

$$\{e_{n+p}a^p;\ p \in \mathbb{N}\} \text{ if } x \text{ is of the form } a^n, \qquad \{a^n\} \text{ if } x \text{ is of the form } e_{n+p}a^p, \qquad \emptyset \text{ otherwise.}$$

Note that $x \multimap d^\bullet$ or $x \multimap e^\bullet$ is empty if x is not of the form a^n. Since the $a^n \multimap d^\bullet$ are pairwise disjoint, it follows easily that for any $X \subset M$, the set $X^3(X \multimap d^\bullet)(X \multimap e^\bullet)$ is

$$\{d_{n+p}e_{n+q}a^{3n+p+q};\ p, q \in \mathbb{N}\} \text{ if } X \text{ is of the form } \{a^n\}, \qquad \emptyset \text{ otherwise.}$$

Assume that X is of the form $\{a^n\}$. Clearly, $\perp$ has been defined in such a way that $f_n \in \varphi^\bullet_X$. Conversely, if $z \in \varphi^\bullet_X$, then $yz \in f^\bullet$ for any $y \in X^3(X \multimap d^\bullet)(X \multimap e^\bullet)$, in particular for $y = d_n e_n a^{3n}$. This is only possible if z is of the form $f_r a^s$, in which case $yz = f_r d_n e_n a^{3n+s}$ and there are $p, q \in \mathbb{N}$ such that

$$n = r + p, \qquad n = r + q, \qquad 3n + s = 3r + p + q.$$

This gives $p = q = -s$, so that $p = q = s = 0$ and $r = n$, which means that $z = f_n$. Hence $\varphi^\bullet_X = \{f_n\}$. Finally, if X is not of the form $\{a^n\}$, one gets $\varphi^\bullet_X = \emptyset \multimap f^\bullet = M$. **Q.e.d.**

Here is an immediate consequence of this lemma: If $i \in [0, m]$ and X is of the form $\{a^p\}$, then one has

$$(c_i\varphi^\bullet_X)^\perp = \{c_i f_p\}^\perp = \{g_q;\ (i, p, q) \text{ is accepted}\},$$

else

$$(c_i\varphi^\bullet_X)^\perp = (c_i M)^\perp = \emptyset.$$

It follows that all the formulas of Θ are satisfied:

- If A is $\forall\gamma.\,(c_i \otimes \varphi[\gamma] \multimap c_j \otimes \varphi[a \otimes \gamma])$ where $\tau(i) = (+, 1, j)$, then

  $$A^\bullet = \bigcap_{X \subset M} c_i\varphi^\bullet_X \multimap (c_j\varphi^\bullet_{aX})^{\perp\perp} = \bigcap_{X \subset M} (c_j\varphi^\bullet_{aX})^\perp \multimap (c_i\varphi^\bullet_X)^\perp.$$

  But if X is of the form $\{a^p\}$, then $aX = \{a^{p+1}\}$ and

  $$(c_j\varphi^\bullet_{aX})^\perp = \{g_q;\ (j, p{+}1, q) \text{ is accepted}\} \subset \{g_q;\ (i, p, q) \text{ is accepted}\} = (c_i\varphi^\bullet_X)^\perp,$$

  else the inclusion holds trivially since $(c_j\varphi^\bullet_{aX})^\perp = \emptyset = (c_i\varphi^\bullet_X)^\perp$. So one has $1 \in (c_j\varphi^\bullet_{aX})^\perp \multimap (c_i\varphi^\bullet_X)^\perp$ for any X, which means that A is satisfied.

- If A is $\forall\gamma.\,(c_i \otimes \varphi[a \otimes \gamma] \multimap c_j \otimes \varphi[\gamma])$ where $\tau(i) = (-, 1, j, k)$, the argument is similar.

- If A is $c_i \otimes \varphi[\mathbf{1}] \multimap c_k \otimes \varphi[\mathbf{1}]$ where $\tau(i) = (-, 1, j, k)$, then

  $$A^\bullet = c_i\varphi^\bullet_{\mathbf{1}} \multimap (c_k\varphi^\bullet_{\mathbf{1}})^{\perp\perp} = (c_k\varphi^\bullet_{\mathbf{1}})^\perp \multimap (c_i\varphi^\bullet_{\mathbf{1}})^\perp,$$

  and

  $$(c_k\varphi^\bullet_{\mathbf{1}})^\perp = \{g_q;\ (k, 0, q) \text{ is accepted}\} \subset \{g_q;\ (i, 0, q) \text{ is accepted}\} = (c_i\varphi^\bullet_{\mathbf{1}})^\perp,$$

  which means that A is satisfied.

- The other three cases are similar.

In fact, a stronger property is needed to prove the theorem, namely that $\#A$, and not only A, is satisfied for every A in Θ. Let us say that a formula A, which encodes some transition of the machine, is *relevant* if this transition occurs in at least one accepting computation. It is easy to see that, if A is relevant, then $A^\bullet = \{1\} = \mathbf{1}$ so that $\#A$ is satisfied. Otherwise, $A^\bullet = M$ and $\#A$ is not satisfied. Nothing ensures that all the formulas of Θ are relevant, but this problem can be solved easily in either of the following ways:

1. One can remove the irrelevant formulas from Θ and still have the fact that $\rho^2, \#\Theta, c_i, \varphi[a^p], \psi[a^q] \vdash \omega$ is provable whenever the configuration (i, p, q) is accepted.

2. Without changing Θ, one can introduce extra generators f_*, g_*, and put all $c_i f_p g_*$ and all $c_i f_* g_q$ in $\bot$. It follows that
$$(c_i \varphi_X^\bullet)^\bot = \{g_q;\ (i,p,q) \text{ is accepted}\} \cup \{g_*\} \neq \emptyset$$
whenever X is of the form $\{a^p\}$. This property and its analogue for $\psi_X^\bullet$ ensure that $A^\bullet = \mathbf{1}$ (so that $\#A$ is satisfied by the new model) for every A in Θ.

In order to complete the proof of Theorem 3, assume that $\rho^2, \#\Theta, c_i, \varphi[a^p], \psi[a^q] \vdash \omega$ is provable in **MLL2**. Lemma 3 gives $f_p \in \varphi[a^p]^\bullet$, $g_q \in \psi[a^q]^\bullet$, and
$$\omega^\bullet = \{c_0 f_0 g_0\}^{\bot\bot} = \{1\}^\bot = \bot.$$
Since ρ and all formulas of $\#\Theta$ are satisfied, one gets
$$c_i f_p g_q \in c_i^\bullet \varphi[a^p]^\bullet \psi[a^q]^\bullet \subset \omega^\bullet = \bot,$$
which means that the configuration (i,p,q) is accepted.

Note that, in Theorem 3, **IMLL2** and **MLL2** can be replaced by **IMALL2** and **MALL2**. Since the phase semantics interprets the additives, the proof is exactly the same. Therefore one gets for free the undecidability of the four fragments, although the problem remained open only for **MLL2**.

7 Discussion

We do not claim that our encoding is the simplest possible one. In fact, the formulas $\varphi[\gamma]$ and $\psi[\gamma]$ have been constructed in such a way that Lemma 3 holds, but it may happen that Theorem 3 holds with simpler formulas. In any case, $\varphi[\gamma]$ and $\psi[\gamma]$ cannot be too simple. For instance, the trivial encoding given by $\varphi[\gamma] = \psi[\gamma] = \gamma$ does not work for an obvious reason: it mixes up the counters. Of course, one can try to encode (i,p,q) by $c_i \otimes a^p \otimes b^q$, using distinct atoms for the counters, but then the formula $c_i \otimes \varphi[\mathbf{1}] \multimap c_k \otimes \varphi[\mathbf{1}]$ is equivalent to $c_i \multimap c_k$, which entails $c_i \otimes a^p \otimes b^q \multimap c_k \otimes a^p \otimes b^q$ for any $p,q \in \mathbb{N}$. This means that, if $\tau(i) = (-,1,j,k)$, the transition $(i,p,q) \to (k,p,q)$ can be simulated even when $p \neq 0$. In other words, the encoding of the 0-test is not faithful. For that precise reason, additives were needed in [LMSS, K2, Lt].

Negative formulas $\varphi[\gamma] = \gamma \multimap f$ and $\psi[\gamma] = \gamma \multimap g$, where f and g are distinct atoms, lead into trouble as well. Assume indeed that the machine has the following transitions:
$$(i,p,q) \to (j,p{+}1,q) \text{ for any } p,q \in \mathbb{N}, \qquad (j,0,q) \to (k,0,q) \text{ for any } q \in \mathbb{N}.$$
These transitions are encoded by $A = \forall\gamma.\,(c_i \otimes (\gamma \multimap f) \multimap c_j \otimes (a \otimes \gamma \multimap f))$ and $B = c_j \otimes f \multimap c_k \otimes f$. In A, the variable γ stands for an arbitrary formula, not necessarily of the form a^n. For instance, A entails $c_i \otimes (f \multimap f) \multimap c_j \otimes (a \otimes f \multimap f)$. Using this, it is easy to prove $A, B \vdash c_i \otimes f \multimap c_k \otimes (a \multimap f)$, which means that the incorrect transition $(i,0,q) \to (k,1,q)$ can be simulated. In fact, there is another difficulty related to the use of $\#$ in the encoding. Assume indeed that the machine has the following transition:
$$(i,p{+}1,q) \to (i,p,q) \text{ for any } p,q \in \mathbb{N}.$$
This transition is encoded by $A = \forall\gamma.\,(c_i \otimes (a \otimes \gamma \multimap f) \multimap c_i \otimes (\gamma \multimap f))$. Since γ occurs negatively in $\varphi[\gamma]$, one proves $a \vdash A$ and then $\#A \vdash A \otimes (a \multimap \mathbf{1})$. Using this, it is possible to simulate an incorrect transition.

Acknowledgement

Andre Scedrov thanks Jean-Yves Girard and members of the *Equipe de Logique de la Programmation* at LMD Marseille for their hospitality during his visit in January–March 1995, when this research was done.

References

[A] Amiot, G. (1994), Unification et logique du second ordre, Thèse de Doctorat, Université de Paris 7 (Denis Diderot).

[G1] Girard, J.-Y. (1987), Linear logic, *Theoretical Computer Science* **50**, 1–102.

[G2] Girard, J.-Y. (1995), Linear logic: its syntax and semantics, *in* "Advances in Linear Logic" (J.-Y. Girard, Y. Lafont & L. Regnier, Eds.), 1–42, London Mathematical Society Lecture Note Series 222, Cambridge University Press.

[K1] Kanovich, M. (1994), The complexity of Horn fragments of linear logic, *Annals of Pure and Applied Logic* **69**, 195–241.

[K2] Kanovich, M. (1995), The Direct Simulation of Minsky machines in Linear logic, *in* "Advances in Linear Logic" (J.-Y. Girard, Y. Lafont & L. Regnier, Eds.), 123–145, London Mathematical Society Lecture Note Series 222, Cambridge University Press.

[Lt] Lafont, Y. (1995), The undecidability of second order linear logic without exponentials, to appear in *Journal of Symbolic Logic.*

[Lk] Lambek, J. (1961), How to program an infinite abacus, *Canadian Math. Bulletin* **4**, 295–302.

[LMSS] Lincoln, P., Mitchell, J., Scedrov, A., and Shankar, N. (1992), Decision Problems for Propositional Linear Logic, *Annals of Pure and Applied Logic* **56**, 239–311.

[LSv] Lincoln, P., and Scedrov, A. (1994), First Order Linear Logic Without Modalities is NEXPTIME-Hard, *Theoretical Computer Science* **135**, 139–154.

[LSS] Lincoln, P., Scedrov, A. and Shankar, N. (1995), Decision Problems for Second Order Linear Logic, *in* "Proc. 10-th Annual IEEE Symposium on Logic in Computer Science, San Diego, California," 476–485.

[LSr] Lincoln, P., and Shankar, N. (1994), Proof Search in First-Order Linear Logic and Other Cut-Free Sequent Calculi, *in* "Proc. 9-th Annual IEEE Symposium on Logic in Computer Science, Paris," 282–291.

[LW] Lincoln, P., and Winkler, T. (1994), Constant-Only Multiplicative Linear Logic is NP-Complete, *Theoretical Computer Science* **135**, 155–169.

[M] Minsky, M. (1961), Recursive unsolvability of Post's problem of 'tag' and other topics in the theory of Turing machines, *Annals of Mathematics* **74**, 437–455.

[Sv1] Scedrov, A. (1993), A Brief Guide to Linear Logic, *in* "Current Trends in Theoretical Computer Science" (G. Rozenberg and A. Salomaa, Eds.), 377–394, World Scientific Publishing Co.

[Sv2] Scedrov, A. (1995), Linear Logic and Computation: A Survey, *in* "Proof and Computation, Proceedings Marktoberdorf Summer School 1993" (H. Schwichtenberg, Ed.), 379–395, NATO Advanced Science Institutes, Series F, Springer-Verlag.