The undecidability of second order linear affine logic

Alexei P. Kopylov*
Department of Mathematics and Mechanics
Moscow State University
119899, Moscow, Russia
e-mail: alexei@lpcs.math.msu.ru

August, 1995

Abstract

The quantifier-free propositional linear affine logic (i.e. linear logic with the weakening) is decidable [Kop]. Recently, Lafont and Scedrov proved that multiplicative fragment of *second-order* linear logic is undecidable [LS]. In this paper we show that the second order linear affine logic is undecidable too. At the same time it turns out that even its multiplicative fragment is undecidable. Moreover, we obtain the whole class of undecidability second order logics which lie between Lambek calculus (LC) and linear affine logic. The proof is based on an encoding two-counter Minsky machines in second order linear affine logic. The faithfulness of the encoding is proved by means of the phase semantic.

1 Introduction and summary

Our notation. Linear logic has been introduced by Girard [Gir]. The inference rules of second order linear logic are represented on Table 1. Linear affine logic is linear logic with the weakening rule (see Table 2) [T]. Non-commutative linear logic is linear logic without the permutation rule [Abr91]. Note than non-commutative LL has two implications: right one and left one ($\multimap$ and $\circ\!-$). We shall abbreviate second order linear logic and linear affine logic as LL2 and LLW2 correspondingly. And non-commutative version of these logics as N-LL2 and N-LLW2. We shall use the abbreviations LL, LLW, N-LL, N-LLW for quantifier-free fragments of corresponding logics.

There are also intuitionistic version of all logics mentioned above. As usually an intuitionistic derivation is a derivation containing only sequents which has no more than one formulas in the consequent. The letter **I** stands for intuitionistic logics (e.g. ILL, ILLW and so on).

Connectives and constants of LL are divided into the three groups: the multiplicatives ($\otimes$, $\wp$, $\multimap$, 1 and $\perp$), the additives ($\oplus$, &, 0 and $\top$) and the exponentials (! and ?).

In referring to linear logic fragments,

M stands for multiplicative fragment (i.e. fragment containing only multiplicatives),

A stands for additive fragment (i.e. fragment containing only additives),

E stands for exponential fragment, (i.e. fragment containing only exponentials).

---

*The research described in this publication was made possible in part by Grant No. NFQ300 from the International Science Foundation

(I)	$\dfrac{}{A\Rightarrow A}$	(CUT)	$\dfrac{\Gamma_1\Rightarrow\Delta_1,A \quad A,\Gamma_2\Rightarrow D_2}{\Gamma_1,\Gamma_2\Rightarrow\Delta_1,\Delta_2}$
$(L^{\perp})$	$\dfrac{\Gamma\Rightarrow A,\Delta}{A^{\perp},\Gamma\Rightarrow\Delta}$	$(R^{\perp})$	$\dfrac{A,\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow A^{\perp},\Delta}$
$(L\otimes)$	$\dfrac{A,B,\Gamma\Rightarrow\Delta}{A\otimes B,\Gamma\Rightarrow\Delta}$	$(R\otimes)$	$\dfrac{\Gamma_1\Rightarrow A,\Delta_1 \quad \Gamma_2\Rightarrow B,\Delta_2}{\Gamma_1,\Gamma_2\Rightarrow A\otimes B,\Delta_1,\Delta_2}$
$(L\wp)$	$\dfrac{A,\Gamma_1\Rightarrow\Delta_1 \quad B,\Gamma_2\Rightarrow\Delta_2}{A\wp B,\Gamma_1,\Gamma_2\Rightarrow\Delta_1,\Delta_2}$	$(R\wp)$	$\dfrac{\Gamma\Rightarrow A,B,\Delta}{\Gamma\Rightarrow\Delta,A\wp B}$
$(R\multimap)$	$\dfrac{A,\Gamma\Rightarrow B,\Delta}{\Gamma\Rightarrow A\multimap B,\Delta}$	$(L\multimap)$	$\dfrac{\Gamma_1\Rightarrow A,\Delta_1 \quad B,\Gamma_2\Rightarrow\Delta_2}{A\multimap B,\Gamma_1,\Gamma_2\Rightarrow\Delta_1,\Delta_2}$
$(L1)$	$\dfrac{\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow 1,\Delta}$	$(R1)$	$\dfrac{}{\Rightarrow 1}$
$(L\perp)$	$\dfrac{}{\perp\Rightarrow}$	$(R\perp)$	$\dfrac{\Gamma\Rightarrow\Delta}{\perp,\Gamma\Rightarrow\Delta}$
$(L\oplus)$	$\dfrac{A,\Gamma\Rightarrow\Delta \quad B,\Gamma\Rightarrow\Delta}{A\oplus B,\Gamma\Rightarrow\Delta}$	$(R\oplus)$	$\dfrac{\Gamma\Rightarrow A,\Delta}{\Gamma\Rightarrow\Delta,A\oplus B}$
$(L\&)$	$\dfrac{A,\Gamma\Rightarrow\Delta}{A\& B,\Gamma\Rightarrow\Delta}$	$(R\&)$	$\dfrac{\Gamma\Rightarrow A,\Delta \quad \Gamma\Rightarrow B,\Delta}{\Gamma\Rightarrow A\& B,\Delta}$
$(L0)$	$\dfrac{}{0,\Gamma\Rightarrow\Delta}$	$(R\top)$	$\dfrac{}{\Gamma\Rightarrow\top,\Delta}$
$(L!)$	$\dfrac{A,\Gamma\Rightarrow\Delta}{!A,\Gamma\Rightarrow\Delta}$	$(R!)$	$\dfrac{!\Gamma\Rightarrow A,?\Delta}{!\Gamma\Rightarrow !A,?\Delta}$
$(W!)$	$\dfrac{\Gamma\Rightarrow\Delta}{!A,\Gamma\Rightarrow\Delta}$	$(C!)$	$\dfrac{!A,!A,\Gamma\Rightarrow\Delta}{!A,\Gamma\Rightarrow\Delta}$
$(L?)$	$\dfrac{A,!\Gamma\Rightarrow ?\Delta}{?A,!\Gamma\Rightarrow ?\Delta}$	$(R?)$	$\dfrac{\Gamma\Rightarrow A,\Delta}{\Gamma\Rightarrow ?A,\Delta}$
$(W?)$	$\dfrac{\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow ?A,\Delta}$	$(C?)$	$\dfrac{\Gamma\Rightarrow ?A,?A,\Delta}{\Gamma\Rightarrow ?A,\Delta}$
$(L\forall)$	$\dfrac{\Gamma,A[B/\alpha]\Rightarrow\Delta}{\Gamma,\forall\alpha A\Rightarrow\Delta}$	$(R\forall)^1$	$\dfrac{\Gamma\Rightarrow A[\beta/\alpha],\Delta}{\Gamma\Rightarrow\forall\alpha A,\Delta}$
$(L\exists)^1$	$\dfrac{\Gamma,A[\beta/\alpha]\Rightarrow\Delta}{\Gamma,\exists\alpha A\Rightarrow\Delta}$	$(R\exists)$	$\dfrac{\Gamma\Rightarrow A[B/\alpha],\Delta}{\Gamma\Rightarrow\exists\alpha A,\Delta}$

[1]In the rules $(R\forall)$ and $(L\exists)$ β must be a fresh variable.

Table 1: The inference rules of LL.

(W)	$\dfrac{W,\Gamma_0\Rightarrow\Delta_0}{W,\Gamma\Rightarrow\Delta}$, where $\Gamma_0\subseteq\Gamma, \Delta_0\subseteq\Delta$

Table 2: The weakening rule for LLW.

(I)	$\dfrac{}{A\Rightarrow A}$	(CUT)	$\dfrac{\Gamma,A,\Delta\Rightarrow Z \quad \Pi\Rightarrow A}{\Gamma,\Pi,\Delta\Rightarrow Z}$
$(L\otimes)$	$\dfrac{\Gamma,A,B,\Delta\Rightarrow Z}{\Gamma,A\otimes B,\Delta\Rightarrow Z}$	$(R\otimes)$	$\dfrac{\Gamma\Rightarrow A \quad \Delta\Rightarrow B}{\Gamma,\Delta\Rightarrow A\otimes B}$
$(L\multimap)$	$\dfrac{\Gamma,B,\Delta\Rightarrow Z \quad \Pi\Rightarrow A}{\Gamma,\Pi,A\multimap B,\Delta\Rightarrow Z}$	$(R\multimap)^2$	$\dfrac{A,\Gamma\Rightarrow B}{\Gamma\Rightarrow A\multimap B}$
$(L\multimapinv)$	$\dfrac{\Gamma,B,\Delta\Rightarrow Z \quad \Pi\Rightarrow A}{\Gamma,B\multimapinv A,\Pi,\Delta\Rightarrow Z}$	$(R\multimapinv)^2$	$\dfrac{\Gamma,A\Rightarrow B}{\Gamma\Rightarrow B\multimapinv A}$

[2]In the rules $(R\multimap)$ and $(R\multimapinv)$ Γ is not empty.

Table 3: The inference rules of LC.

For example, MLL means the multiplicative fragment of LL, MALLW denotes multiplicative-additive fragment of LLW, and so on.

Also we shall consider Lambek calculus (LC). In contrast to the traditionally notations for connectives of LC: $\backslash$, $/$, $\cdot$, we shall use the following notations: $\multimap$ and $\circ\!-$ stands for the left and right implications, and $\otimes$ stands for the tensor product. (See Table 3 for the inference rules). It is clear that LC $\subseteq$ N-ILL $\subseteq$ ILL $\subseteq$ LL $\subseteq$LLW.

Main results. Lincoln, Scedrov, and Shankar showed the undecidability of IMLL2 and IMALL2 by embedding of LJ2 [LSS]. Lafont has proved the undecidability of MALL2 [Laf1]. Then Lafont and Scedrov proved that MLL2 is undecidable too [LS]. Emms showed embedding of LJ2 into N-IMLL2. Kanovich demonstrated the undecidability of N-MLL2, cyclic LL and second order Lambek Calculus (LC2) [Kan2]. On the other hand, quantifier-free linear affine logic is decidable [Kop]. The decidability problem for second order linear affine logic remained open.

In the current paper we prove the undecidability of LLW2. Also we prove that for any logic L if LC2 $\subseteq L \subseteq$ LLW2, then L is undecidable. In particular, all second-order logics mentioned above are undecidable as well as MLLW2, MALLW2, LLW2, IMLLW2, N-MLLW2, etc. The main ideas of the proof is similar to the ideas in [LS]. Namely, we encode two counter machines (Minsky machines) in LC2 and LLW2. This encoding is like as the encodings in [Kan1, Laf1, LS]. In order to obtain the faithfulness of the encoding we use (as in [Laf1, LS]) the phase semantic, but here we need the phase semantic for linear affine logic.

2 Phase semantic

Let us remind some definitions about phase semantic [Gir, Laf2]. Phase space is the triple $(M, \perp, K)$, where M is a commutative monoid, $\perp \subseteq M$ and K is a submonoid of the submonoid $J(M) = \{x \in \perp^{\perp} \mid x \in \{x^2\}^{\perp\perp}\}$. For instance, K may be $\{1\}$.

Let $X, Y \subseteq M$, then, by definition,

$$XY = \{xy \mid x \in X, y \in Y\}, \quad X \multimap Y = \{z \in M \mid \forall x \in X \; xz \in Y\}, \quad X^{\perp} = X \multimap \perp.$$

We say that X is a fact, when there is Y such that $X = Y^{\perp}$. It is easy to see, that X is a fact if and only if $X = X^{\perp\perp}$. If X is any subset of M, then $X^{\perp\perp}$ is the smallest fact containing X. By definition,

$$\begin{gathered} X \otimes Y = (XY)^{\perp\perp} \quad X \wp Y = (X^{\perp} \otimes Y^{\perp})^{\perp} \\ X \& Y = X \cap Y \quad X \oplus Y = (X^{\perp} \& Y^{\perp})^{\perp} \\ 1 = \perp^{\perp} \quad \top = M \quad 0 = \top^{\perp} \end{gathered}$$

If we have an interpretation of all atoms p by facts $p^{\bullet}$, then for any formula A from LL2 we can naturally define a fact $A^{\bullet}$. Namely, $^{\bullet}$ is commutated with all connectives, and $(\forall\alpha\, A[\alpha])^{\bullet}$ is defined as

$$\bigcap_{X \text{ is a fact}} A[X]^{\bullet},$$

where $A[X]^{\bullet}$ is an interpretation of A, when $\alpha^{\bullet} = X$. By definition, a formula A is satisfied, if $1 \in A^{\bullet}$. A sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_k$ is satisfied, if the formula $(A_1 \otimes \ldots \otimes A_n) \multimap (B_1 \wp \ldots \wp B_k)$ is satisfied. It is equal to that $(A_1 \otimes \ldots \otimes A_n)^{\bullet} \subseteq (B_1 \wp \ldots \wp B_k)^{\bullet}$.

Definition ([Laf2]) We say that the phase space is an affine phase space if $\perp$ is an ideal, i.e. $M \cdot \perp \subseteq \perp$.

Theorem 1 ([Laf2]) *If a sequent Φ is derivable in LLW2, then any affine phase space $(M, \perp, K)$ satisfies Φ.*

Remark *For quantifier-free linear affine logic without modalities the converse theorem holds too.*

If $(M, \perp)$ is an affine phase space, then $M \cdot \perp = \perp$, and $\{1\}^{\perp\perp} = \perp^{\perp} = M$. Hence, for any fact X if $1 \in X$ then $X = M$. Moreover, $\perp = M^{\perp}$ is the least fact. Now let us prove one useful lemma. Let we have a homomorphism $\gamma : M \to N$. (N is a monoid of natural numbers with the operation $+$). Define $S_n = \gamma^{-1}(\{n\})$ is a prototype of $\{n\}$, and $M_n = \bigcup_{k \geq n} S_k$. We assume that $M_{-n} = M$, where $n \geq 0$.

Lemma 2.1 *Let we have two sets $X = X' \cup X^*$ and $Y = Y' \cup M_{k+1}$, where $\emptyset \neq X' \subseteq S_n$; $X^* \subseteq M_{n+1}$ and $Y' \subseteq S_k$. Then*

$$X \multimap Y = (X' \multimap Y') \cup M_{k-n+1}.$$

Proof Really, $M_{k-n+1}X \subseteq M_{k+1} \subseteq Y$. So, $M_{k-n+1} \subseteq X \multimap Y$. Moreover,

$$(X' \multimap Y')X = (X' \multimap Y')X' \cup (X' \multimap Y')X^* \subseteq Y' \cup M_{k-n}M_{n+1} = Y.$$

So, $X' \multimap Y' \subseteq X \multimap Y$.

Conversely, if $z \notin M_{k-n}$, then $zX \ni zx$, where $x \in X'$. But $\gamma(zx) < k$ and $zx \notin Y$. Hence, $z \notin X \multimap Y$. At last, if $\gamma(z) = k - n$, and $z \in X \multimap Y$, then $zX' \subseteq Y$. But $zX' \subseteq S_k$. Therefore, $zX' \subseteq Y'$, and $z \in X' \multimap Y'$. ■

Corollary 2.2 *If $\perp = \perp' \cup M_{k+1}$, where $\perp' \subseteq S_k$, then any fact is a set of the kind: $X' \cup M_{n+1}$, where $X' \subseteq S_n$.*

3 Two counter Minsky machines

A configuration of a two counter Minsky machine is determined by three numbers (i, p, q). Here $p, q \in N$ are the states of the machine's counters, and $i \in [0, m]$ is a label. A program of the machine is given by a map $\tau : [1, m] \to \{+\} \times \{1, 2\} \times [0, m] \cup \{-\} \times \{1, 2\} \times [0, m] \times [0, m]$. If the current configuration of the machine is (i, p, q), then the 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 for zero first counter),
- $(j, p, q+1)$ 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 for zero second counter).

The machine stops when $i = 0$. A configuration (i, p, q) is accepted by the machine if, starting from (i, p, q), it eventually stops on $(0, 0, 0)$. This transitions are denoted by the following way:

- $(i, p, q) \to (j, p+1, q)$,
- $(i, p, q) \to (j, p-1, q)$ if $p > 0$,
- $(i, 0, q) \to (k, 0, q)$,
- $(i, p, q) \to (j, p, q+1)$,
- $(i, p, q) \to (j, p, q-1)$ if $q > 0$,
- $(i, p, 0) \to (k, p, 0)$.

4 Encoding two counter Minsky machines

We can encode Minsky machines by the following way. Let us consider two formulas:

$$\varphi[\alpha] = (\alpha \multimap f) \multimap h,$$
$$\psi[\alpha] = (\alpha \multimap g) \multimap e.$$

We construct the following infinite sets of formulas:

$$\begin{array}{lll} \varphi_{-1} = a, & \varphi_n = \varphi[\varphi_{n-1}], & n \in N \\ \psi_{-1} = b, & \psi_n = \psi[\psi_{n-1}], & n \in N \end{array}$$

Any machine configuration (i, p, q) is encoded by the following formula:

$$c_i \otimes \varphi_p \otimes \psi_q.$$

Here a, b, c_i, e, f, g, h are literals. An increment transition $(i, p, q) \to (j, p+1, q)$ is encoded by the formula

$$\forall \alpha, \beta (c_i \otimes \varphi[\alpha] \otimes \beta \multimap c_j \otimes \varphi[\varphi[\alpha]] \otimes \beta).$$

A decrement transition $(i, p, q) \to (j, p-1, q)$ if $p > 0$ is encoded by the formula

$$\forall \alpha, \beta (c_i \otimes \varphi[\varphi[\alpha]] \otimes \beta \multimap c_j \otimes \varphi[\alpha] \otimes \beta).$$

And a test-for-zero transition $(i, 0, q) \to (k, 0, q)$ is encoded by the formula

$$\forall \beta (c_i \otimes \varphi[a] \otimes \beta \multimap c_k \otimes \varphi[a] \otimes \beta).$$

By analogous, for the second counter the transitions $(i, p, q) \to (j, p, q+1)$, $(i, p, q) \to (j, p, q-1)$ if $q > 0$, and $(i, p, 0) \to (k, p, 0)$ are encoded by the following formulas

$$\forall \alpha, \beta (c_i \otimes \alpha \otimes \psi[\beta] \multimap c_j \otimes \alpha \otimes \psi[\psi[\beta]]),$$
$$\forall \alpha, \beta (c_i \otimes \alpha \otimes \psi[\psi[\beta]] \multimap c_j \otimes \alpha \otimes \psi[\beta]),$$
$$\forall \alpha (c_i \otimes \alpha \otimes \psi[b] \multimap c_k \otimes \alpha \otimes \psi[b]).$$

Let $T_1, \ldots, T_n$ be formulas of the above kind which encode all transitions of our machine. Let $T = T_1 \otimes \ldots \otimes T_n$. We may assume without lost of generalization, that $n \geq 1$. Consider the following formulas:

$$W = \forall x, y, z (x \otimes y \otimes z \multimap y),$$
$$W' = \forall y, z (y \otimes z \multimap y),$$
$$C = \forall x (x^4 \multimap x^6).$$

The two first formulas encode weakening, and the last formula encodes contraction. Here α^n means $\alpha \otimes \alpha \otimes \ldots \otimes a$ (n times).

Theorem 2 *The following four assertions are equivalent:*

(i) *The configuration (i, p, q) is accepted by the machine.*

(ii) *The following sequent is derivable in LC2:*

$$c_i \otimes \varphi_p \otimes \psi_q, (T \otimes W)^4, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0$$

(iii) *The above sequent is derivable in any logic L, such that LC2$\subseteq$ L $\subseteq$LLW2.*

(iv) *This sequent is derivable in LLW2.*

Lambek [Lk] and Minsky [M] have demonstrated that there is a Minsky machine for which the set of accepted configuration is not recursive. Hence, we have

Corollary 4.1 *Any logic L, such that LC2$\subseteq$ L $\subseteq$LLW2 is undecidable.*

Proof of the theorem 2. The implications (ii)$\to$(iii) and (iii)$\to$(iv) are trivial. Let us prove the implication (i)$\to$(ii). We begin with

Lemma 4.2 *The following rules are derivable in LC2:*

$$\frac{\Pi, A^6, C^5, \Gamma \vdash Z}{\Pi, A^4, C^5, \Gamma \vdash Z}(\tilde{C}) \qquad \frac{\Pi, A, \Gamma \vdash Z}{\Pi, S_1 \otimes A \otimes S_2, W, \Gamma \vdash Z}(\tilde{W}) \qquad \frac{\Pi, A, \Gamma \vdash Z}{\Pi, A, \Sigma, W', \Gamma \vdash Z}(\tilde{W}')$$

In the last rule Σ is not empty.

Proof The derivation of the rule $(\tilde{C})$ is following:

$$\frac{\overline{C^4, C \vdash C^6} \qquad \dfrac{\overline{A^4, C \vdash A^6} \quad \Pi, A^6, C^5, \Gamma \vdash Z}{\Pi, A^4, C^6, \Gamma \vdash Z}(CUT)}{\Pi, A^4, C^5, \Gamma \vdash Z}(CUT)$$

And here the derivations of the rules $(\tilde{W})$ and $(\tilde{W}')$:

$$\frac{\overline{S_1 \otimes A \otimes S_2, W \vdash A} \qquad \Pi, A, \Gamma \vdash Z}{\Pi, S_1 \otimes A \otimes S_2, W, \Gamma \vdash Z}(CUT) \qquad \frac{\overline{A, \Sigma, W' \vdash A} \qquad \Pi, A, \Gamma \vdash Z}{\Pi, A, \Sigma, W', \Gamma \vdash Z}(CUT)$$

■

Lemma 4.3 *For any transition $(i,p,q) \to (i',p',q')$ if T_k is a formula encoding this transition then the sequent $c_i \otimes \varphi_p \otimes \psi_q, T_k \vdash c_{i'} \otimes \varphi_{p'} \otimes \psi_{q'}$ is derivable in LC2.*

Proof It is easy to see that the following sequents are derivable in LC2:

$$\begin{array}{l}
c_i \otimes \varphi_p \otimes \varphi_q,\ \forall\alpha,\beta(c_i \otimes \varphi[\alpha] \otimes \beta \multimap c_j \otimes \varphi[\varphi[\alpha]] \otimes \beta) \ \vdash \ c_j \otimes \varphi_{p+1} \otimes \varphi_q, \\
c_i \otimes \varphi_p \otimes \varphi_q,\ \forall\alpha,\beta(c_i \otimes \varphi[\varphi[\alpha]] \otimes \beta \multimap c_j \otimes \varphi[\alpha] \otimes \beta) \ \vdash \ c_j \otimes \varphi_{p-1} \otimes \varphi_q, \\
c_i \otimes \varphi_0 \otimes \varphi_q,\ \forall\beta(c_i \otimes \varphi[a] \otimes \beta \multimap c_k \otimes \varphi[a] \otimes \beta) \ \vdash \ c_k \otimes \varphi_0 \otimes \varphi_q, \\
c_i \otimes \psi_p \otimes \psi_q,\ \forall\alpha,\beta(c_i \otimes \alpha \otimes \psi[\beta] \multimap c_j \otimes \alpha \otimes \psi[\psi[\beta]]) \ \vdash \ c_j \otimes \psi_p \otimes \psi_{q+1}, \\
c_i \otimes \psi_p \otimes \psi_q,\ \forall\alpha,\beta(c_i \otimes \alpha \otimes \psi[\psi[\beta]] \multimap c_j \otimes \alpha \otimes \psi[\beta]) \ \vdash \ c_j \otimes \psi_p \otimes \psi_{q-1}, \\
c_i \otimes \psi_p \otimes \psi_0,\ \forall\alpha(c_i \otimes \alpha \otimes \psi[b] \multimap c_k \otimes \alpha \otimes \psi[b]) \ \vdash \ c_k \otimes \psi_p \otimes \psi_0.
\end{array}$$

We prove the implication (i)$\to$(ii) by induction on the length of the computation. The base of induction is holds because of the rule $(\tilde{W}')$:

$$\frac{c_0 \otimes \varphi_0 \otimes \psi_0 \vdash c_0 \otimes \varphi_0 \otimes \psi_0}{c_0 \otimes \varphi_0 \otimes \psi_0, (T \otimes W)^4, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0}(\tilde{W}')$$

Now let us verify the induction step. If the computation has the following form: $(i,p,q) \to (i',p',q') \to \ldots \to (0,0,0)$, then we have the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Lemma 4.3}}{c_i \otimes \varphi_p \otimes \psi_q, T_k \vdash c_{i'} \otimes \varphi_{p'} \otimes \psi_{q'}} \quad \dfrac{\text{induction hypothesis}}{c_{i'} \otimes \varphi_{p'} \otimes \psi_{q'}, (T\otimes W)^4, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0}}{c_i \otimes \varphi_p \otimes \psi_q, T_k, (T\otimes W)^4, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0}(CUT)}{c_i \otimes \varphi_p \otimes \psi_q, T\otimes W \otimes T, W, (T\otimes W)^4, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0}(\tilde{W})}{c_i \otimes \varphi_p \otimes \psi_q, (T\otimes W)^6, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0}(L\otimes)}{c_i \otimes \varphi_p \otimes \psi_q, (T\otimes W)^4, C^5, W' \vdash c_0 \otimes \varphi_0 \otimes \psi_0}(\tilde{C})$$

If $k=1$, then we may use the rule $(\tilde{W})$ because of $T\otimes W\otimes T = (T\otimes W)\otimes T_k \otimes (T')$. Otherwise, we also may use the rule $(\tilde{W})$ because of $T\otimes W\otimes T = (T')\otimes T_k \otimes (T''\otimes W\otimes T)$ or $T\otimes W\otimes T = (T')\otimes T_k\otimes(W\otimes T)$. ■

5 Faithfulness

Now let us check the implication (iv)→(i) using the phase space. Consider the free commutative monoid M generated by infinite set of atoms $\overline{a}$, $\overline{b}$, (c_i), $\overline{f}$, (f_n), $\overline{h}$, (h_n), $\overline{g}$, (g_n), $\overline{e}$, (e_n), where $i = 0,\ldots,m$; $n \geq -1$. Consider the homomorphism $\gamma : M \to N$, which is defined by the following equations:

$$\gamma(c_i) = \gamma(\overline{f}) = \gamma(f_n) = \psi(\overline{h}) = \gamma(h_n) = \gamma(\overline{g}) = \gamma(g_n) = \gamma(\overline{e}) = \gamma(e_n) = 1, \quad \gamma(\overline{a}) = \gamma(\overline{b}) = 2,$$

and $\gamma(xy) = \gamma(x) + \gamma(y)$. Let M_n, as in the lemma 2.1, be a prototype of the set $\{k | k \geq n\}$.

$\perp$ is defined by the following way:

$$\begin{aligned} \perp' = {} & \{c_i f_p g_q | (i,p,q) \text{ is accepted by the machine}\} \cup \\ & \{\overline{f} f_{n-1} h_n | n \in N\} \cup \{\overline{h} h_n f_n | n \in N\} \cup \{\overline{a} f_{-1}\} \cup \\ & \{\overline{g} g_{n-1} e_n | n \in N\} \cup \{\overline{e} e_n g_n | n \in N\} \cup \{\overline{b} g_{-1}\}, \\ \perp = {} & \perp' \cup M_4. \end{aligned}$$

Note, that $\perp' \subseteq S_3$. So, we can use corollary 2.2. The interpretation of literals is following:

$$\begin{array}{ll} f^\bullet = \{\overline{f}\}^\perp = \{f_{n-1}h_n | n \in N\} \cup M_3, & h^\bullet = \{\overline{h}\}^\perp = \{h_n f_n | n \in N\} \cup M_3, \\ g^\bullet = \{\overline{g}\}^\perp = \{g_{n-1}e_n | n \in N\} \cup M_3, & e^\bullet = \{\overline{e}\}^\perp = \{e_n g_n | n \in N\} \cup M_3, \\ a^\bullet = \{\overline{a}\}^\perp = \{f_{-1}\} \cup M_2, & b^\bullet = \{\overline{b}\}^\perp = \{g_{-1}\} \cup M_2, \\ c_i^\bullet = \{c_i\}^{\perp\perp} & \end{array}$$

Lemma 5.1

$$\varphi_n^\bullet = \{f_n\} \cup M_2 \qquad \psi_n^\bullet = \{g_n\} \cup M_2$$

Lemma 5.2 *M satisfies all formulas T_k encoding our transitions.*

l	$\alpha = \alpha' \cup M_{l+1}$	$\alpha \multimap f^\bullet$	$\varphi[\alpha] = (\alpha \multimap f^\bullet) \multimap h^\bullet$	$\varphi[\varphi[\alpha]]$	$(c_i\varphi[\alpha])^\perp$	$(c_j\varphi[\varphi[\alpha]])^\perp$
0	M	$f^\bullet$	M_1	M_1	M_2	M_2
1	$\mathbf{\{f_{n-1}\} \cup M_2}$	$\mathbf{\{h_n\} \cup M_2}$	$\mathbf{\{f_n\} \cup M_2}$	$\mathbf{\{f_{n+1}\} \cup M_2}$	$\mathbf{\{g_q \mid q : (i, n, q)}$ **is accepted**$\mathbf{\} \cup M_2}$	$\mathbf{\{g_q \mid q : (j, n+1, q)}$ **is accepted**$\mathbf{\} \cup M_2}$
	$\{h_n\} \cup M_2, n \geq 1$	$\{f_{n-1}\} \cup M_2$	$\{h_{n-1}\} \cup M_2$	$\{h_{n-2}\} \cup M_2$	M_2	M_2
	$\{h_0\} \cup M_2$	$\{f_{-1}\} \cup M_2$	$\{h_{-1}\} \cup M_2$	M_1	M_2	M_2
	otherwise	M_2	M_1	M_1	M_2	M_2
2	$\alpha \subseteq f^\bullet$	M_0	$h^\bullet$	M_2	M_1	M_1
	otherwise	M_1	M_2	M_2	M_1	M_1
OTHERWISE		M_0	$h^\bullet$	M_2	M_1	M_1

Table 4: $\varphi[\alpha]$, $\varphi[\varphi[\alpha]]$, $(c_i\varphi[\alpha])^\perp$, $(c_j\varphi[\varphi[\alpha]])^\perp$ for $\alpha = \alpha' \cup M_{l+1}$.

Proof of 5.1 and 5.2 We can count $\varphi[\alpha]$, $\varphi[\varphi[\alpha]]$, $(c_i\varphi[\alpha])^\perp$, $(c_j\varphi[\varphi[\alpha]])^\perp$ for any set $\alpha \subseteq M$, where $\alpha = \alpha' \cup M_{l+1}$ and $\alpha' \subseteq M_l$, using the lemma 2.1 (see Table 4).

We prove that $\varphi^\bullet_n = \{f_n\} \cup M_2$ by induction on n. For $n = -1$ we have that $\varphi^\bullet_{-1} = a^\bullet = \{f_{-1}\} \cup M_2$ by definition. The induction step is satisfied because of $\varphi[\alpha] = \{f_n\} \cup M_2$, for $\alpha = \{f_{n-1}\} \cup M_2$. The second assertion of the lemma 5.1 is proved similarly.

One can see from Table 4 that $(c_i\varphi[\alpha])^\perp = (c_j\varphi[\varphi[\alpha]])^\perp$ for any $\alpha \neq \{f_{n-1}\} \cup M_2$. Moreover, if the increment transition $(i, p, q) \to (j, p+1, q)$ is accepted, then $(c_i\varphi[\alpha])^\perp \supseteq (c_j\varphi[\varphi[\alpha]])^\perp$ for $\alpha = \{f_{n-1}\} \cup M_2$. Hence, the formula

$$\forall \alpha, \beta (c_i \otimes \varphi[\alpha] \otimes \beta \multimap c_j \otimes \varphi[\varphi[\alpha]] \otimes \beta)$$

is satisfied. By analogous, if the decrement transition $(i, p, q) \to (j, p-1, q)$ is accepted, then $(c_j\varphi[\alpha])^\perp \subseteq (c_i\varphi[\varphi[\alpha]])^\perp$ for $\alpha = \{f_{n-1}\} \cup M_2$. So, the formula

$$\forall \alpha, \beta (c_i \otimes \varphi[\varphi[\alpha]] \otimes \beta \multimap c_j \otimes \varphi[\alpha] \otimes \beta)$$

is satisfied. Finally, if the test-for-zero transition $(i, 0, q) \to (k, 0, q)$ is accepted, then $(c_k\varphi[a^\bullet])^\perp \subseteq (c_i\varphi[a^\bullet])^\perp$. So the formula

$$\forall \beta (c_i \otimes \varphi[a] \otimes \beta \multimap c_k \otimes \varphi[a] \otimes \beta)$$

is satisfied. Clearly, all formulas encoded transition with second counter are satisfied too. ∎

Lemma 5.3 *M satisfies the formula C.*

Proof Let X be a fact. If $1 \in X$, then $X = M$ and $X^4 = X^6$. Otherwise, $XXXX \subseteq M_4 \subseteq \perp$, and $X^4 = (XXXX)^{\perp\perp} = \perp$, because $\perp$ is the least fact. By the same way $X^6 = \perp$. In the both cases we have $X^4 = X^6$. Therefore, C is satisfied. ∎

Lemma 5.4 *M satisfies the formulas W and W'.*

Proof In LLW2 the formulas W and W' are derivable. Hence, they are satisfied. ■

Now let us prove (iv)→(i). It is following from (iv) and the lemmas 5.2, 5.3 and 5.4 that M satisfies the implication

$$c_i \otimes \varphi_p \otimes \psi_q \multimap c_0 \otimes \varphi_0 \otimes \psi_0.$$

By definition $c_0 f_0 g_0 \in \bot$. Hence, $(c_0 \otimes \varphi_0 \otimes \psi_0)^\bullet = \{c_0 f_0 g_0\}^{\bot\bot} = \bot$, because $\bot$ is the least fact. Therefore, we have

$$c_i f_p g_q \in (c_i \otimes \varphi_p \otimes \psi_q)^\bullet \subseteq (c_0 \otimes \varphi_0 \otimes \psi_0)^\bullet = \bot.$$

So, (i) holds. ■

Acknowledgments

I am grateful to Sergei Artemov for his supervision. I am also indebted to Max Kanovich for his inspiring introduction to the problem and discussions on it. I would like to thank Yves Lafont for his helpful remarks.

References

[Abr91] V.M. Abrusci. Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic. Jornal of Symbolic logic 56, 1403–1451. 1991.

[Gir] J.-Y. Girard. Linear Logic. *Theoretical Computer Science,* 50, 1–102, 1987.

[Kan1] M.I. Kanovich. The Direct Simulation of Minsky machine in Linear logic. To appear in *Advances in Linear Logic,* edited by J.-Y. Girard, Y. Lafont & L. Regnier, London Mathematical Society Lecture Note Series, Cambridge University Press. 1995.

[Kan2] M.I. Kanovich. Second order Lambek is undecidable. Message to LL list. 3 July 1995.

[Kop] A.P. Kopylov. Decidability of Linear Affine Logic. 10-th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, IEEE Computer Society Press. 1995.

[Laf1] Y. Lafont. The Undecidability of Second Order Linear Logic without Exponentials. To appear in Jornal of Symbolic Logic. Available by anonymous ftp from `lmd.univ-mrs.fr` as `pub/lafont/undecid.[dvi,ps].Z`. 1995.

[Laf2] Y. Lafont. The Finite Model Property for Various Fragments of Linear Logic. 1995. Available by anonymous ftp from `lmd.univ-mrs.fr` as `pub/lafont/model.[dvi,ps].Z`. 1995.

[LS] Y. Lafont, A. Scedrov. The Undecidability of Second Order Multiplicative Linear Logic. Available by anonymous ftp from `lmd.univ-mrs.fr` as `pub/lafont/mll2.[dvi,ps].Z`. 1995.

[LSS] P. Lincoln, A. Scedrov, N. Shankar. Decision Problem for Second Order Linear Logic. 10-th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, IEEE Computer Society Press. 1995.

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

[M] M. Minsky. Recursive unsolvability of Post's problem of 'tag' and other topics in the theory of Turing machines. *Annals of Mathematics,* 74:3:437-455, 1961.

[T] A.S. Troelstra. Lectures on Linear Logic. *CSLI lecture notes; no.29*, 1992