UNDECIDABILITY OF LINEAR LOGIC WITHOUT WEAKENING

JUN SUZUKI AND KATSUHIKO SANO

1. Introduction

In this abstract, we introduce a linear logic system **CLLR** in which a weakening rule is *completely omitted* in the sense that not only the weakening rule for an exponential modality is dropped but also the units $\mathbf{1}$ and $\bot$ are dropped from the syntax. The goal of this abstract is to establish that it is undecidable whether a sequent is provable in **CLLR**.

Our weakening-free linear logic has the following syntax $\mathcal{L}$:

$$\mathcal{L} \ni A ::= p \mid \top \mid \mathbf{0} \mid {\sim}A \mid A \otimes A \mid A \mathbin{\&} A \mid A \parr A \mid A \oplus A \mid A \multimap A \mid {!}A \mid {?}A.$$

Table 1 shows its *sequent calculus* system **CLLR**, in which a *sequent* is a pair of finite multisets of formulas enriched with the symbol " $\Rightarrow$ ". Classical propositional logic **CLL** (Girard [1]) is obtained by adding to **CLLR** the units $\mathbf{1}$, $\bot$ and the following rules for them, as well as weakening rules for the exponentials ! and ?.

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \bot}\,[\bot r]\ ,\quad \frac{}{\bot \Rightarrow}\,[\bot l]\ ,\quad \frac{}{\Rightarrow \mathbf{1}}\,[\mathbf{1}r]\ ,\quad \frac{\Gamma \Rightarrow \Delta}{\mathbf{1}, \Gamma \Rightarrow \Delta}\,[\mathbf{1}l]$$

It is known that **CLL** is undecidable [6, Theorem 3.7]. However, it is not obvious whether **CLL** without structural rules or units is decidable. For example, **CLL** without exponentials, **MALL**, which has neither weakening nor contraction, is decidable (Lincoln et al. [6, Theorem 2.2]). However, non-commutative classical propositional linear logic, **NCCLL**, which we can regard as **CLL** without exchange, is undecidable (Lincoln et al. [6, Theorem 4.8]). Furthermore, **NCCLL** without weakening is also undecidable (Kanovich et al. [4, Corollary 14]). This system is still undecidable if one omits the units $\mathbf{1}$ and $\bot$.

In this abstract, we establish the undecidability of **CLLR** by showing that the system can simulate any two-counter machine proposed by Minsky [7]. To show this, we use Lafont's method [5] with phase semantics. This method was originally introduced to establish the undecidability of second-order version of **MALL**. Using semantics allows us to avoid a combinatorial argument of translating proofs into computations with lots of case distinctions, as seen in Lincoln et al [6].

In **CLLR**, a contraction rule is restricted only to formulas in the antecedent prefixed with "!" (and dually in the succedent prefixed with "?"), while a weakening rule is not allowed at all. This system is capable of representing resources that can be freely copied but not discarded.

2. Phase Semantics

Let us introduce *phase semantics*. We adopt the definition of Girard [2, Section 2.1.2]. A *phase space* is a pair $(\mathcal{M}, \bot)$ where $\mathcal{M} = (|\mathcal{M}|, \cdot, 1)$ is a commutative

TABLE 1. Sequent Calculus of **CLLR**

$$\frac{}{A \Rightarrow A}\,\mathbf{id} \qquad \frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma' \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\,Cut$$

$$\frac{A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \sim A}\,[\sim r] \qquad \frac{\Gamma \Rightarrow \Delta, A}{\sim A, \Gamma \Rightarrow \Delta}\,[\sim l] \qquad \frac{}{\Gamma \Rightarrow \Delta, \top}\,[\top r] \qquad \frac{}{\mathbf{0}, \Gamma \Rightarrow \Delta}\,[\mathbf{0}l]$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma' \Rightarrow \Delta', B}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta', A \otimes B}\,[\otimes r] \qquad \frac{A, B, \Gamma \Rightarrow \Delta}{A \otimes B, \Gamma \Rightarrow \Delta}\,[\otimes l]$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \,\&\, B}\,[\& r] \qquad \frac{A_i, \Gamma \Rightarrow \Delta}{A_0 \,\&\, A_1, \Gamma \Rightarrow \Delta}\,[\& l_i]\ (i = 0, 1)$$

$$\frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \parr B}\,[\parr r] \qquad \frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma' \Rightarrow \Delta'}{A \parr B, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\,[\parr l]$$

$$\frac{\Gamma \Rightarrow \Delta, A_i}{\Gamma \Rightarrow \Delta, A_0 \oplus A_1}\,[\oplus r_i]\ (i = 0, 1) \qquad \frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \oplus B, \Gamma \Rightarrow \Delta}\,[\oplus l]$$

$$\frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \multimap B}\,[\multimap r] \qquad \frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma' \Rightarrow \Delta'}{A \multimap B, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\,[\multimap l]$$

$$\frac{!A, !A, \Gamma \Rightarrow \Delta}{!A, \Gamma \Rightarrow \Delta}\,[!\mathbf{C}] \qquad \frac{!\Gamma \Rightarrow ?\Delta, A}{!\Gamma \Rightarrow ?\Delta, !A}\,[!r] \qquad \frac{A, \Gamma \Rightarrow \Delta}{!A, \Gamma \Rightarrow \Delta}\,[!l]$$

$$\frac{\Gamma \Rightarrow \Delta, ?A, ?A}{\Gamma \Rightarrow \Delta, ?A}\,[?\mathbf{C}] \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, ?A}\,[?r] \qquad \frac{A, !\Gamma \Rightarrow ?\Delta}{?A, !\Gamma \Rightarrow ?\Delta}\,[?l]$$

monoid and $\bot$ be an arbitrary subset of the domain $|\mathcal{M}|$ of $\mathcal{M}$. A *phase model* $\mathcal{P}$ is a pair $((\mathcal{M}, \bot), v)$ of a phase space $(\mathcal{M}, \bot)$ and a function $v\colon \mathsf{Prop} \to \wp(|\mathcal{M}|)$ such that for all $p \in \mathsf{Prop}$, $v(p) = \sim\sim v(p)$, where for $X \subseteq |\mathcal{M}|$, $\sim X \subseteq |\mathcal{M}|$ is defined by

$$\sim X = \{y \in |\mathcal{M}| \mid (\forall x \in X)[x \cdot y \in \bot]\}.$$

For a phase model $\mathcal{P} = ((|\mathcal{M}|, \cdot, 1), \bot, v)$, we define an *interpretation* $[\cdot]_{\mathcal{P}}\colon \mathcal{L} \to \wp(|\mathcal{M}|)$ of formulas inductively as follows (if it is clear from the context which model is considered, the subscript can be omitted), although only those of $\otimes$, $\&$, $\oplus$ $\bot$, $\multimap$ and $!$ are used in this abstract:

- $[p] = v(p)$, $[\mathbf{1}] = \sim\sim\{1\}$, $[\bot] = \bot$, $[\top] = |\mathcal{M}|$, $[\mathbf{0}] = \sim\sim\emptyset$, $[\sim A] = \sim[A]$,
- $[A \otimes B] = \sim\sim([A][B])$, $\quad [A \,\&\, B] = [A] \cap [B]$,
- $[A \parr B] = \sim(\sim[A] \cdot \sim[B])$, $\quad [A \oplus B] = \sim\sim([A] \cup [B])$,
- $[A \multimap B] = \{z \in |\mathcal{M}| \mid (\forall x \in [A])[x \cdot z \in [B]]\}$,
- $[!A] = \sim\sim([A] \cap \mathcal{I})$, $\quad [?A] = \sim(\sim[A] \cap \mathcal{I})$,

where $\mathcal{I} = \{i \in \sim\sim\{1\} \mid i \cdot i = i\}$ and for $X, Y \subseteq |\mathcal{M}|$, $X \cdot Y \subseteq |\mathcal{M}|$ is defined as follows (the operator "$\cdot$" and parentheses may be omitted as $(X \cdot Y) \cdot Z = XYZ$):

$$X \cdot Y = \{x \cdot y \mid x \in X \text{ and } y \in Y\}.$$

We say that a formula $A \in \mathcal{L}$ is *true* in $\mathcal{P}$ if $1 \in [A]_{\mathcal{P}}$.

The following lemma is useful for proving the undecidability of **CLLR** and can be shown by using the soundness theorem.

Lemma 1 (Lafont [5, p.545])**.** Let $\mathcal{P}$ be a phase model. If a sequent $\Gamma, A_1, ..., A_n \Rightarrow C$ $(n \geq 1)$ of $\mathcal{L}$ is provable in **CLLR** and B is true in $\mathcal{P}$ for all $B \in \Gamma$, then $[A_1]_{\mathcal{P}} \cdots [A_n]_{\mathcal{P}} \subseteq [C]_{\mathcal{P}}$.

3. Two-Counter Machine

Let us introduce a two-counter machine. We employ a formulation from Lafont [5]. A *two-counter machine* M consists of a set S of states, the terminal state $s_t \in S$ and a function

$$\tau \colon S \backslash \{s_t\} \to (\{+\} \times \{A, B\} \times S) \cup (\{-\} \times \{A, B\} \times S \times S).$$

An element of $S \times \mathbb{N} \times \mathbb{N}$ is said to be an *instantaneous description* (*ID*), which means a state and values of the two counters. For an ID (s_j, p, q), $\tau(s_j)$ represents a program that commands a transition from one ID to the next, which can take one of the following four forms: $(+, A, s_k), (-, A, s_k, s_l), (+, B, s_k), (-, B, s_k, s_l)$. Transitions of IDs by programs are as follows:

- if $\tau(s_j) = (+, A, s_k)$, then $(s_j, p, q) \rightsquigarrow (s_k, p + 1, q)$,
- if $\tau(s_j) = (-, A, s_k, s_l)$,
  - if $p > 0$, then $(s_j, p, q) \rightsquigarrow (s_k, p - 1, q)$, and
  - if $p = 0$, then $(s_j, p, q) \rightsquigarrow (s_l, p, q)$,
- if $\tau(s_j) = (+, B, s_k)$, then $(s_j, p, q) \rightsquigarrow (s_k, p, q + 1)$,
- if $\tau(s_j) = (-, B, s_k, s_l)$,
  - if $q > 0$, then $(s_j, p, q) \rightsquigarrow (s_k, p, q - 1)$, and
  - if $q = 0$, then $(s_j, p, q) \rightsquigarrow (s_l, p, q)$,

Provided an ID (s_i, p, q), M computes sequentially, starting with the program $\tau(s_i)$ corresponding to the state s_i. The computation terminates when it reaches the state s_t. We call an *accepted sequence* of (s_i, p, q) in M a finite sequence of IDs

$$(s_0, p_0, q_0), (s_1, p_1, q_1), \ldots, (s_n, p_n, q_n),$$

such that for all $s_k \in S\backslash\{s_t\}$, $\tau(s_{k-1})$ is a program that makes $(s_{k-1}, p_{k-1}, q_{k-1})$ transition to (s_k, p_k, q_k), and $(s_0, p_0, q_0) = (s_i, p, q)$, $(s_n, p_n, q_n) = (s_t, 0, 0)$. An ID (s_i, p, q) is *accepted* by M if there is an accepted sequence of (s_i, p, q) in M.

Lemma 2 (Minsky [7, Theorem Ia])**.** There exists a two-counter machine M such that the problem of whether an input is accepted by M is undecidable.

4. Undecidability of **CLLR**

We define the formula that is a translation of programs of a two-counter machine. Given a finite set $S = \{s_t, s_1, ..., s_n\}$ of states, we stipulate that $c_t, c_1, ..., c_n$ are propositional variables corresponding to $s_t, s_1, ..., s_n$, respectively.

Definition 3. Let $M = (S, s_t, \tau)$ be a two-counter machine. Fix propositional variables a, b, a', b'. We write θ_M for the formula obtained by connecting with & the set of the following formulas corresponding to the programs of M and further four others:

- for $\tau(s_j) = (+, A, s_k)$: $c_j \multimap c_k \otimes a$,
- for $\tau(s_j) = (-, A, s_k, s_l)$: $c_j \otimes a \multimap c_k$ and $c_j \multimap c_l \oplus (a' \mathbin{\&} c_t)$,
- for $\tau(s_j) = (+, B, s_k)$: $c_j \multimap c_k \otimes b$,
- for $\tau(s_j) = (-, B, s_k, s_l)$: $c_j \otimes b \multimap c_k$ and $c_j \multimap c_l \oplus (b' \mathbin{\&} c_t)$,
- $a' \multimap a' \mathbin{\&} c_t$, $(a' \mathbin{\&} c_t) \otimes b \multimap a' \mathbin{\&} c_t$, $b' \multimap b' \mathbin{\&} c_t$, $(b' \mathbin{\&} c_t) \otimes a \multimap b' \mathbin{\&} c_t$.

The propositional variables "a", "b" correspond to two counters, while a', b' are introduced to deal with conditional branches of decrement commands. The implication "$\multimap$" represents the transition of states and the incrementing or decrementing of the counters. The variable "a" or "b" to the right of the implication, say in $c_j \multimap c_k \otimes a$ or $c_j \multimap c_k \otimes b$, corresponds to incrementing of the first or second counter, while "a" or "b" to the left of the implication, say in $c_j \otimes a \multimap c_k$ or $c_j \otimes b \multimap c_k$, corresponds to decrementing of the first or second counter.

This translation is obtained by modifying those of Kanovich [3] and Lafont [5]. Furthermore, however Kanovich and Lafont used a finite multiset of formulas, we use a single formula θ_M made by &. This corresponds to the situation where only a necessary program is extracted. This becomes important when expressing a system where resources cannot be discarded.

Lemma 4. For any two-counter machine $M = (S, s_t, \tau)$ and any ID (s_i, p, q), if (s_i, p, q) is accepted by M, then the sequent $(!\theta_M)^{g(s_i)}, c_i, a^p, b^q \Rightarrow c_t$ is provable in **CLLR**, where $g(s_i) = 0$ if $s_i = s_t$, otherwise $g(s_i) = 1$.

To prove the converse of Lemma 4, we introduce a special kind of phase model, which is the same one used in Lafont [5]. In the following, we write a^2b for a multiset $\{a, a, b\}$.

Definition 5. Given a two-counter machine $M = (S, s_t, \tau)$, the phase model $\mathcal{P}_M = ((\mathcal{M}, \perp), v)$ derived from M is defined as follows:

- $|\mathcal{M}| = \{\Gamma \mid \Gamma$ is a finite multiset of $\mathcal{L}$ formulas$\}$. The unit $1 = \emptyset$. The monoid operator $\cdot = \cup$ (the union operation of multisets).
- $\perp$ is defined by

$$\perp = \{c_i a^p b^q \mid (s_i, p, q) \text{ is accepted by } M\} \\ \cup \{a' b^q \mid q \in \mathbb{N}\} \cup \{b' a^p \mid p \in \mathbb{N}\}.$$

- $v(p) = {\sim\sim}\{p\}$.
- It is clear by definition that $\mathcal{I} = \{1\}$ and that $v(c_t) = \perp$.

Lemma 6. For any two-counter machine $M = (S, s_t, \tau)$ and any ID (s_i, p, q), if the sequent $(!\theta_M)^{g(s_i)}, c_i, a^p, b^q \Rightarrow c_t$ is provable in **CLLR**, then (s_i, p, q) is accepted by M.

Proof. Suppose that the sequent $(!\theta_M)^{g(s_i)}, c_i, a^p, b^q \Rightarrow c_t$ is provable in **CLLR**. By Lemma 1, if $!\theta_M$ is true in the phase model $\mathcal{P}_M$, i.e., $1 = \emptyset \in [!\theta_M]$, then $[c_i][a]^p[b]^q \subseteq [c_t]$, which means that, by the definition of $\mathcal{P}_M$ and a property of "$\sim\sim$", $c_i a^p b^q \in v(c_t) = \perp$. If $c_i a^p b^q \in \perp$, by the definition of $\perp$, the ID (s_i, p, q) is accepted by M. So we show that $!\theta_M$ is true in $\mathcal{P}_M$. By the fact that $!\theta_M = {\sim\sim}(\theta_M \cap \mathcal{I})$ and $\mathcal{I} = \{1\}$, and by the definition of &, it suffices to show that all the formulas connected by & when defining θ_M are true in $\mathcal{P}_M$. □

Corollary 7. For any two-counter machine $M = (S, s_t, \tau)$ and any ID (s_i, p, q), the sequent $(!\theta_M)^{g(s_i)}, c_i, a^p, b^q \Rightarrow c_t$ is provable in **CLLR** iff (s_i, p, q) is accepted by M.

Combining this with Lemma 2, we get the undecidability.

Theorem 8. **CLLR** is undecidable.

References

[1] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[2] Jean-Yves Girard. Linear logic : its syntax and semantics. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, London Mathematical Society Lecture Note Series, pages 1–42. Cambridge University Press, 1995.

[3] Max Kanovich. The direct simulation of Minsky machines in linear logic. *Advances in Linear Logic*, pages 123–145, 1995.

[4] Max Kanovich, Stepan Kuznetsov, Vivek Nigam, and Andre Scedrov. Subexponentials in non-commutative linear logic. *Mathematical Structures in Computer Science*, 29(8):1217–1249, 2019.

[5] Yves Lafont. The undecidability of second order linear logic without exponentials. *The Journal of Symbolic Logic*, 61:541–548, 1996.

[6] Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56:239–311, 1992.

[7] Marvin L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961.