Constant-only multiplicative linear logic is NP-complete

Patrick Lincoln*
Computer Science Laboratory, SRI International, Menlo Park, CA 94025, USA

Timothy Winkler
1080 Noel Dr., Apt 1, Menlo Park, CA 94025, USA

Abstract

Linear logic is a resource-aware logic that is based on an analysis of the classical proof rules of contraction (copying) and weakening (throwing away). In this paper we study the decision problem for the multiplicative fragment of linear logic without quantifiers or propositions: the constant-only case. We show that this fragment is NP-complete. Earlier work by Kanovich showed that propositional multiplicative linear logic is NP-complete. With Natarajan Shankar, the first author developed a simplified proof for the propositional case. The structure of this simplified proof is utilized here with a new encoding which uses only constants. The end product is the somewhat surprising result that simply evaluating expressions in true, false, and, or in multiplicative linear logic (1, ⊥, ⊗, and ⊕) is NP-complete. By conversativity results not proven here, the NP-hardness of larger fragments of linear logic follows.

1. Introduction

When Girard introduced linear logic [7], he brought to light the expressive power which can be gained by restricting the structural rules of contraction (copying) and weakening (throwing away) for propositions. Without contraction or weakening, propositions may be thought of as resources, which must be carefully accounted for. Since linear logic treats propositions as resources natively, it has been called "resource-conscious" [5]. When propositions are treated as resources, as they are in linear logic, one is naturally led to consider two different forms of conjunction and
disjunction. Girard named the two kinds of connectives “additive” and “multiplicative”, and focused his attention on the multiplicative fragment by giving proof nets (a version of natural deduction tailored for linear logic) for this fragment. Since then much of the interest in linear logic has revolved around this fragment and small extensions to this fragment.

In order to explain the proof-theoretic difference between additive and multiplicative connectives, consider the conjunctive goal $\Sigma \vdash A \land B$. In all sequent calculi, one must prove $\Delta \vdash A$ and one must also prove $\Gamma \vdash B$, for some $\Delta$ and $\Gamma$ in order to prove this goal using the $\land$ rule. Various sequent calculi place different requirements on the relationship between $\Sigma, \Delta,$ and $\Gamma$. For example, in classical logic the latter two could be required to be subsets of the first ($\Delta \subseteq \Sigma$ and $\Gamma \subseteq \Sigma$). This may be seen as implicitly allowing copying of some propositions, (those which appear in all three contexts), and throwing away others (those which appear in the conclusion $\Sigma$, but not in either hypothesis). The multiplicative conjunction $\otimes$ of linear logic requires that the context $\Sigma$ be divided between its hypotheses (the multiset union of $\Delta$ and $\Gamma$ is $\Sigma$). The additive conjunction $\&$, on the other hand, requires that the context be duplicated in both hypotheses ($\Delta \equiv \Gamma \equiv \Sigma$). This critical difference is also reflected in the two forms of disjunction, which are the DeMorgan duals of the two forms of conjunction.

Girard also added “exponential” unary connectives (! and ?) to linear logic, increasing the expressive power of the logic greatly. In fact, propositional linear logic with exponentials is undecidable [15]. Without exponentials, multiplicative–additive linear logic is PSPACE-complete [15].

In this paper we focus on the smaller fragment with only the multiplicative connectives and constants, Constant-Only Multiplicative Linear Logic, or COMLL. In an earlier paper, the first author showed that the decision problem for Multiplicative Linear Logic (with propositions) MLL is in NP, by giving (a sketch of) an NP algorithm [15]. However, the NP-hardness of this problem was left open. Kanovich resolved this question showing that MLL is NP-complete [10-12].

An argument for the NP-hardness of this fragment was sketched by Kanovich in electronic mail [10], thus demonstrating that this decision problem is NP-complete. The authors along with Natarajan Shankar found a simplification of Kanovich’s argument which is presented later in this paper. Kanovich later updated his argument to show that the “Horn fragment” of the multiplicatives is also NP-complete [11].

Here we show that not only is MLL NP-complete, but COMLL is NP-complete as well. Note that this fragment contains no quantifiers or propositions, and thus one may view this decision problem as simply evaluating expressions in true, false, and, and or in multiplicative linear logic ($1, \perp, \otimes, \text{and} \lor$). In COMLL there are many logical values; in fact, if one identifies values by equivalence classes of expressions under provable equivalence, then most (in some sense) expressions are distinct logical values. We will not, in fact, present an algorithm for evaluating COMLL expressions, but will use the fact that the value of any provable COMLL expression must be 1.
1.1. Propositional linear logic

The formal framework we will work with throughout this paper is a Gentzen-style sequent calculus. We discuss three independent logics here: LL (full propositional linear logic), MLL (LL restricted to multiplicative connectives and constants), and COMLL (the constant-only fragment of MLL). We begin with a definition of LL.

A linear logic sequent is a \( \vdash \) followed by a multiset of linear logic formulas. Note that in standard presentations of sequent calculi, sequents are often built from sets of formulas, where here we use multisets; this difference is crucial. We assume given a set of propositional symbols \( p_i \), along with their associated negations, \( p_i^\perp \). Below we give the inference rules for the linear sequent calculus, along with the definition of negation and implication. The reader should note that negation is a defined concept, not an operator.

The following notational conventions are followed in this paper:

- \( p_i \): positive propositional symbol
- \( p_i^\perp \): negative propositional symbol
- \( A, B, C \): arbitrary formulas
- \( \Sigma, \Gamma, \Lambda \): arbitrary multisets of formulas

Thus the identity rule (I below) is restricted to atomic formulas, although in fact the identity rule for arbitrary formulas (\( \vdash A, A^\perp \)) is derivable in this system. For notational convenience, it is usually assumed that \( \rightarrow \) and \( \otimes \) associate to the right, and that \( \otimes \) has higher precedence than \( \rightarrow \). The notation \( ?\Sigma \) is used to denote a multiset of formulas which all begin with ?.

The names for the rules given below are shown on the right. Note that there is no rule for the 0 constant.

\[
\begin{align*}
\text{I} & \quad \vdash p_i, p_i^\perp \\
\text{Cut} & \quad \vdash \Sigma, A \quad \vdash \Gamma, A^\perp \\
& \quad \vdash \Sigma, \Gamma \\
\otimes & \quad \vdash \Sigma, A \quad \vdash B, \Gamma \\
& \quad \vdash \Sigma, (A \otimes B), \Gamma \\
\Rightarrow & \quad \vdash \Sigma, A, B \\
& \quad \vdash \Sigma, (A \Rightarrow B) \\
\oplus & \quad \vdash \Sigma, A \\
& \quad \vdash \Sigma, (A \oplus B) \\
& \quad \vdash \Sigma, B \\
\& & \quad \vdash \Sigma, (A \& B) \\
\text{W} & \quad \vdash \Sigma, ?A \\
\text{C} & \quad \vdash \Sigma, ?A, ?A \\
& \quad \vdash \Sigma, ?A \\
\text{D} & \quad \vdash \Sigma, ?A \\
\end{align*}
\]

identity \( \quad \) MLL

\( \vdash \) \( \quad \) cut \( \quad \) MLL \( \quad \) COMLL

\( \vdash \) \( \quad \) tensor \( \quad \) MLL \( \quad \) COMLL

\( \vdash \) \( \quad \) par \( \quad \) MLL \( \quad \) COMLL

\( \vdash \) \( \quad \) plus \( \quad \) with

\( \vdash \) \( \quad \) weakening

\( \vdash \) \( \quad \) contraction

\( \vdash \) \( \quad \) dereliction
Linear negation is defined as follows:

\[

d(p_i)^\perp \triangleq p_i^\perp \\
(p_i^\perp)^\perp \triangleq p_i \\
(A \otimes B)^\perp \triangleq B^\perp \otimes A^\perp \\
(A \boxdot B)^\perp \triangleq B^\perp \otimes A^\perp \\
(A \oplus B)^\perp \triangleq A^\perp \& B^\perp \\
(A \& B)^\perp \triangleq A^\perp \oplus B^\perp \\
(!A)^\perp \triangleq ?A^\perp \\
(?A)^\perp \triangleq !A^\perp \\
(1)^\perp \triangleq \perp \\
(\perp)^\perp \triangleq 1 \\
(0)^\perp \triangleq \top \\
(T)^\perp \triangleq 0
\]

Linear implication, \(\rightarrow\), is defined as follows:

\[
A \rightarrow B \triangleq A^\perp \boxdot B.
\]

Note that \(A^\perp\) should be thought of as standing for the translation defined above.

1.2. Multiplicative linear logic

The multiplicative fragment of linear logic MLL is defined as follows. The formulas of MLL are those of linear logic except that \(\oplus, \&, !, ?, 0,\) and \(\top\) are not allowed. The sequent rules for MLL are the same as those for LL except that the rules for the additive connectives, additive constants, and exponentials are omitted: \(\oplus, \&, ?W, ?C, ?D, !S,\) and \(\top\). This leaves only the rules \(\textbf{I}, \textbf{Cut}, \otimes, \boxdot, \perp, \top,\) and \(\textbf{1}\). These rules are marked on the right with MLL in the listing of rules above.

1.3. Constant-only multiplicative linear logic

In this paper, our focus is on the constant-only multiplicative fragment of linear logic COMLL. The formulas of COMLL are built from just \(1, \perp, \otimes,\) and \(\boxdot\). The sequent rules for COMLL are those of MLL except \(\textbf{I}\). These rules are marked on the right with COMLL in the listing of rules above.
1.4. Multiplicative linear logic is NP-complete

In this section we summarize results about the decision problem for propositional multiplicative linear logic. An argument for the NP-hardness of this fragment was first sketched by Kanovich in electronic mail [10]. Together with the earlier result [15] that the multiplicatives are in NP, Kanovich's result showed that this decision problem is NP-complete. Kanovich later updated his argument to show that the "Horn fragment" of the multiplicatives is also NP-complete [11, 12], using a novel computational interpretation of this fragment of linear logic. This paper continues this trend by providing a proof that evaluating expressions in true, false, and, and or in multiplicative linear logic is NP-complete. That is, even without propositions, multiplicative linear logic is NP-complete.

MLL and COMLL are in NP. Informally, the argument showing membership in NP is simply that every connective in a multiplicative linear logic formula is analyzed exactly once in any cut-free proof. Thus an entire proof, if one exists, can be guessed and checked in nondeterministic polynomial time.

Formally, we first state a fundamental theorem originally due to Girard [7], but proven in complete detail in [15].

**Theorem 1.1 (Cut elimination).** If a sequent is provable in MLL, then it is provable in MLL without using the Cut rule.

The above references actually prove this theorem for full linear logic, but the result for the fragments in question here follows immediately.

Without cut, multiplicative proofs are quite concise.

**Theorem 1.2 (Small-proofs).** Every connective is analyzed exactly once in any cut-free MLL or COMLL proof.

From Theorem 1.1 and Theorem 1.2, we know that given a MLL or COMLL sequent of size \( n \), if there is any proof of this sequent, then there is a proof with exactly \( n \) total applications of inference rules. Since each application of an inference rule may be represented in space linear in \( n \), we may simply guess and check an entire \( n^2 \) representation of a proof tree in nondeterministic polynomial time.

The following is one of a large family of permutabilities of inferences. Propositional classical logic allows all possible permutabilities (that is, it never matters which formula one chooses to break first in a classical proof), while intuitionistic and linear logic exhibit some impermutabilities [13]. The following permutability of (multiplicative) disjunction holds in linear logic.

**Lemma 1.3 (Permutability of \( \& \)).** If there is a proof of \( \vdash \Gamma, (A \& B) \), then there is a proof of \( \vdash \Gamma, A, B \).
This lemma essentially states that comma is the same as $\otimes$ in sequents.

The corresponding permutability for $\otimes$ does not hold, as demonstrated by the following example $\vdash (1 \otimes 1), (\bot \otimes \bot)$.

2. COMLL is NP-complete

Some time ago, Girard [8] developed a necessary condition for the provability of COMLL expressions:

Lemma 2.1 (Girard [8]). Define a function $M$ from constant multiplicative linear expressions to be integers as follows:

\[
\begin{align*}
M(1) &= 1 \\
M(\bot) &= 0 \\
M(A \otimes B) &= M(A) + M(B) \\
M(A \otimes B) &= M(A) + M(B) - 1.
\end{align*}
\]

If a formula $A$ is provable in multiplicative linear logic and contains no propositions, then $M(A) = 1$.

In other words, the number of tensors is one less than the number of ones in any provable constant-only MLL (COMLL) formula. Avron (and others) have studied generalizations of this "semantic" measure to include propositions (where a proposition $p$ is given value 1, and $p^\perp$ is given value 0) yielding necessary conditions for MLL provability [3]. One may go even further, achieving a necessary condition for provability in larger fragments of linear logic, using min for $\&$ and max for $\oplus$, and plus and minus infinity for the additive constants. One may also generalize these conditions somewhat, replacing all instances of the numeric value 1 with an arbitrary constant $c$, and allowing propositions to have different (although fixed) values, where $p$ has value $v_p$, and $p^\perp$ has value $c - v_p$ [3]. Other related work is given in [19] and [4].

By defining $M(\vdash A_1, A_2, \ldots, A_n) = \sum_{1 \leq i \leq n} M(A_i)$ the function $M$ can be extended to sequents. A sequent will be provable only if its measure is 1.

Since the above is only a necessary condition, there has been a question as to whether some form of simple "truth table" or numerical evaluation function like the above could yield a necessary and sufficient condition for provability of COMLL expressions. The main result of this paper shows that even this multiplicative constant expression evaluation or circuit evaluation problem is NP-complete.

We will encode an NP-complete problem, 32-Partition, in COMLL, and show that our encoding is sound and complete. The main idea is that the small-proof property of COMLL allows us to encode "resource distribution" problems naturally. Note that our encoding remains sound and complete in larger fragments of linear logic that are conservative over COMLL. However, the complexity of most larger
fragments of linear logic have already been completely characterized [15]. The 432 Partition problem is introduced to separate the argument into properties of NP-complete problems in general and properties specific to linear logic.

2.1. 3-Partition

3-Partition is the basis of our proof of the NP-completeness of 432-Partition. 3-Partition is described in Garey and Johnson [6, Page 224].

Instance: Set $A$ of $3m$ elements, a bound $B \in \mathbb{Z}^+$, and a size $s(a) \in \mathbb{Z}^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$.

Question: Can $A$ be partitioned into $m$ disjoint sets $A_1, A_2, \ldots, A_m$ such that, for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B$ (note that each $A_i$ must therefore contain exactly 3 elements from $A$)?

Reference: [Garey and Johnson [6], 1975].

Comment: NP-complete in the strong sense.

Note that 3-Partition is NP-complete in the strong sense, which implies that even when the input is represented in unary, the problem is NP-hard. This property of 3-Partition is essential for our application, since we represent the input problem in unary by multiplicities of linear formulas.

2.2. 432-Partition

We introduce a new NP-complete problem, a variant of 3-Partition, that we call 432-Partition:

Instance: Set $A$ of $3m$ elements, a bound $B \in \mathbb{Z}^+$ with $B > 8$, and a size $s(a) \in \mathbb{Z}^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$.

Question: Can $A$ be partitioned into $m$ disjoint sets $A_1, A_2, \ldots, A_m$ such that, for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B + |A_i| - 3$?

Comment: NP-complete in the strong sense.

We will show that solutions of 432-Partition correspond to solutions of 3-Partition for the same problem instance when $B > 8$. The restriction that $B > 8$ is added merely to simplify the statement of certain results. Informally, 432-Partition is in NP since one can nondeterministically guess and check a solution in polynomial time. 432-Partition is NP-hard since one can polynomially reduce 4-Partition (which is also known to be NP-complete in the strong sense [6]) to 3-partition such that the bound $B$ is strictly greater than 8, and, for common problem instances, 432-Partition and 3-Partition are equivalent. In Garey and Johnson [6] the standard transformation from 4-Partition to 3-Partition showing 3-Partition is NP-complete results in bound $B$ strictly greater than 67.

In fact, there is a very strong equivalence between these two problems when $B > 8$: the instances are the same, and all common instances are solvable in one case exactly
when they are solvable in the other. Furthermore, solutions in one case directly correspond to solutions in the other case. For common instances, it is clear that solutions to 3-Partition are solutions for the same instance of 432-Partition.

Suppose we are given a solution to an instance of 432-Partition. First we will show that the groups of elements in the partition must have 4, 3, or 2 elements. For an arbitrary \( A_i \), let \( A_i \) consist of \( x_1, \ldots, x_n \) from \( A \), and let \( X_1 = s(x_1), \ldots, X_n = s(x_n) \). Clearly we have \( B/4 < X_i < B/2 \).

If \( n = 0 \), we have \( 0 = B - 3 \), which is false by our assumption that \( B > 8 \). If \( n = 1 \), we have \( X_1 = B - 2 \), but the sizes are bounded above by \( B/2 \), and with the assumption that \( B > 8 \), there is a contradiction. Since \( B/4 < X_i \) and \( \sum_{1 \leq i \leq n} X_i = n + B - 3 \) we have \( nB/4 < n + B - 3 \). From this, via \( nB < 4B + 4n - 12 \), we get \( 12 - 4B < (4 - B)n \), and then \( (B - 4)n < 4B - 12 \). Since \( 8 < B \) implies that \( 0 < (B - 4)n \), we have \( n < (4B - 12)/(B - 4) \) and hence \( n < 4 + 4/(B - 4) \). Given that \( 8 < B \) and the right-hand expression is decreasing as a function of \( B \), we conclude that \( n < 5 \). This leaves the \( n = 2, n = 3, \) and \( n = 4 \) cases.

Thus we have a partition each element of which consists of either two, three, or four elements.

We will now describe how to construct a partition that is a solution to the 3-partition instance from the 432-Partition solution. For any partition group \( A_i \) consisting of \( x_1, \ldots, x_n \), as before, in the case that \( n = 3 \), we have \( \sum_{1 \leq i \leq 3} X_i = B \), and thus this set identifies a group which directly satisfies the requirement for 3-Partition, that is, the sum is equal to \( B \).

Note that if \( n = 2 \), we have by above constraint that \( X_1 + X_2 = B - 1 \), and if \( n = 4 \), then \( X_1 + X_2 + X_3 + X_4 = B + 1 \). Since \( \sum_{a \in A} s(a) = mB \), and the sums of the \( m \) groups of the partition must also add up to \( mB \), there are exactly the same number of groups with four elements as there are groups with two elements.

Let \( C = \text{floor}(B/4) \), then \( 4C \leq B \leq 4C + 3 \). There also exists an \( 0 \leq a \leq 3 \), such that \( B = 4C + a \). Since the size of each element must be \( > B/4 \), the smallest element size will be \( C + 1 \). Since each element size is \( < B/2 \), the largest element size is \( 2C + 1 \). If there are any groups of four, then the problem constraint \( X_1 + X_2 + X_3 + X_4 = B + 1 \) implies, using the lower bound on the element size, that \( 4(C + 1) \leq B + 1 \), that is, \( 4C + 3 \leq B \). But this can only be true when \( B = 4C + 3 \). Furthermore, the elements in a group of four must all have this minimal size \( C + 1 \), otherwise the constraint cannot be satisfied. If there are any groups of two, then \( X_1 + X_2 = B - 1 \), and there must be a group of four, so \( B = 4I + 3 \) as above and \( X_1 + X_2 = 4C + 2 \). This forces any group of two to consist of elements with the maximal size \( 2C + 1 \). Noting that there are exactly as many groups of two as groups of four, we may rearrange the elements of a group of four and a group of two into two groups of three by taking two elements from the group of four and one element from the group of two to form each group of three. Both resulting groups of three have total size \( 4C + 3 \), which happily is equal to \( B \). This "reshuffling" will result in a solution to the 3-Partition problem for the same problem instance. Therefore 3-Partition and 432-Partition are equivalent problems.
Note that, by the reduction above, 432-Partition is NP-complete in the strong sense. Thus 432-Partition is NP-complete even in unary notation. This is important, since we utilize a unary representation of instances in our linear logic encoding.

2.3. Encoding with propositions

We use the notation, for \( x \) proposition and \( Y \) number,

\[
x^Y = x \otimes x \otimes \cdots \otimes x \otimes x.
\]

Given an instance of 3-Partition equipped with a set \( A = \{a_1, \ldots, a_{3m}\} \), an integer \( B \), and a unary function \( s \), presented as a tuple \( \langle A, m, B, s \rangle \), we define the encoding function \( \Theta \) as

\[
\Theta(\langle A, m, B, s \rangle) = [(k \leftarrow c^{S_1}) \otimes \cdots \otimes (k \leftarrow c^{S_{3m}})] \leadsto [(k \leftarrow c^B)^m].
\]

We write \( S_i \) for \( s(a_i) \) to improve the readability here and in the following.

This encoding is based on an encoding developed by the first author and Natarajan Shankar. It has been shown that this formula is provable in the multiplicative fragment of linear logic if and only if the 3-Partition problem is solvable [16].

The encoding using only constants can be generated from this one by replacing all occurrences of \( k \) and \( c \) by \( \bot \).

2.4. Constant-only encoding

We will now describe how 432-Partition instances (which are at the same time 3-Partition instances) can be encoded in COMLL.

We will use the following notation:

\[
x^Y = x \otimes x \otimes \cdots \otimes x \otimes x,
\]

as before and

\[
x^{\langle Y \rangle} = x \otimes x \otimes \cdots \otimes x \otimes x.
\]

Note that \((x^Y)^\perp = (x^Y)^\perp\) and \((x^{\langle Y \rangle})^\perp = (x^{\langle Y \rangle})^\perp\). We also take \( x^0 = 1 \) and \( x^{(0)} = \bot \).

Given an instance of 432-Partition equipped with a set \( A = \{a_1, \ldots, a_{3m}\} \), an integer \( B \), and a unary function \( s \), presented as a tuple \( \langle A, m, B, s \rangle \), we define the encoding function \( \Theta \) as

\[
\Theta(\langle A, m, B, s \rangle) = [(\bot \leftarrow \bot^{S_1}) \otimes \cdots \otimes (\bot \leftarrow \bot^{S_{3m}})] \leadsto [(\bot \leftarrow B)^m].
\]

(Recall that \( S_i \) abbreviates \( s(a_i) \).)

Using the contrapositive \((A \leftarrow B \equiv B^\perp \leftarrow A^\perp)\), we can develop a "1 only" encoding.

\[
[(1^{\langle S_1 \rangle} \leftarrow 1) \otimes (1^{\langle S_2 \rangle} \leftarrow 1) \otimes \cdots \otimes (1^{\langle S_{3m} \rangle} \leftarrow 1)] \leadsto [(1^{\langle B \rangle} \leftarrow 1^{\langle 1 \rangle})^m].
\]
Eliminating the linear implication in favour of $\otimes$ these formulas both become:

$$\Theta((A, m, B, s)) = (1^{(S_1)} \otimes \bot) \otimes (1^{(S_2)} \otimes \bot) \otimes \cdots \otimes (1^{(S_{3m})} \otimes \bot) \otimes (1^{(3)})^{m}.$$  

We will use the last form of this formula, since it contains no implicit negations (linear implication). One may see that this formula satisfies Girard’s measure condition, Lemma 2.1, if there are $3m$ elements, and the sum of the sizes equals $mB$, side conditions on the statement of 432-Partition (and 3-Partition). Here are some useful basic results for Girard's measure function: $M(\bot^Y) = -Y + 1$ when $1 < Y$, $M(1^{(Y)}) = Y$, $M(x^Y) = YM(x) - Y + 1$, and $M(x^{<Y}) = YM(x)$.

The claim is that (any one of) the above encoding formulas are provable in the multiplicative fragment of linear logic if and only if the 432-Partition problem is solvable.

2.5. Soundness

Lemma 2.2 (Soundness). If a 3-Partition problem $(A, m, B, S)$ is solvable, then we are able to find a proof of the COMLL formula $\Theta((A, m, B, s))$.

Proof. The proof is straightforward. For each group of 4, 3, or 2 elements in the assumed solution to the 432-Partition problem, one forms the following subproof, assuming that two of the elements of the group are numbered $x$ and $y$. $\Gamma$ is a schematic element which will be discussed below.

$$
\vdash 1^{\langle S_x \rangle}, 1^{\langle S_y \rangle}, \Gamma, 1, \bot^B
\quad \vdash 1, \bot
\quad \vdash 1, \\
\vdash (1^{\langle S_x \rangle} \otimes \bot), 1^{\langle S_y \rangle}, \Gamma, 1, 1, \bot^B
\quad \vdash (1^{\langle S_x \rangle} \otimes \bot), 1^{\langle S_y \rangle}, \Gamma, 1, 1, 1, \bot^B
\quad \vdash (1^{\langle S_x \rangle} \otimes \bot), 1^{\langle S_y \rangle}, \Gamma, 1, 1, \bot^B
\quad \vdash (1^{\langle S_x \rangle} \otimes \bot), 1^{\langle S_y \rangle}, \Gamma, 1, 1, \bot^B
\quad \vdash (1^{\langle S_x \rangle} \otimes \bot), 1^{\langle S_y \rangle}, \Gamma, (1^{(2)})^B
\quad \vdash (1^{\langle S_x \rangle} \otimes \bot), 1^{\langle S_y \rangle}, \Gamma, (1^{(3)})^B
$$

Depending on whether the group has 4, 3, or 2 elements $\Gamma$ is, respectively, “$1^{\langle S_z \rangle} \otimes 1$”, “$1^{\langle S_w \rangle} \otimes 1$”, or empty, where $z$ and $w$ are also numbers of elements of the group. The elided proof of $\vdash 1^{\langle S_x \rangle}, 1^{\langle S_y \rangle}, 1, 1, \bot^B$ is guaranteed to exist by the conditions on the solution to 432-Partition, as we can see by considering the different cases.

As demonstrated in the above proof schemata, since $\vdash 1, \bot$ is provable, a requirement to prove $\vdash (A \otimes \bot), 1, \Sigma$ can be directly reduced to the requirement to prove $\vdash A, \Sigma$ by a use of the $\otimes$ rule. By induction we get that $\vdash 1^{\langle N \rangle}, \bot^N$, and in this the
occurrences of 1 may be arbitrarily grouped by using the $\otimes$ rule. (This means that Girard's condition, Lemma 2.1, is necessary and sufficient for sequents of this very simple form.)

In the case that the group has just 2 elements, it is necessary to prove $\vdash 1^{\langle Sx \rangle}, 1^{\langle Sy \rangle}, 1, \bot^B$, but this is easy since conditions on the instance require that $Sx + Sy = B - 1$ and so $Sx + Sy + 1 = B$.

In the case that the group has 3 elements, it is necessary to prove $\vdash 1^{\langle Sx \rangle}, 1^{\langle Sy \rangle}, (1^{\langle Sz \rangle} \otimes \bot), 1, \bot^B$, which reduces by a use of the $\otimes$ rule to $\vdash 1^{\langle Sx \rangle}, 1^{\langle Sy \rangle}, 1^{\langle Sz \rangle}, \bot^B$. This last is easily seen to be provable since the conditions on the instance require that $Sx + Sy + Sz = B$.

In the case that the group has 4 elements, it is necessary to prove $\vdash 1^{\langle Sx \rangle}, 1^{\langle Sy \rangle}, (1^{\langle Sz \rangle} \otimes \bot), (1^{\langle Sw \rangle} \otimes \bot), 1, \bot^B$. First apply the $\otimes$ rule once to $(1^{\langle Sw \rangle} \otimes \bot)$ and the 1 that is available. Then apply the $\otimes$ rule once more on $(1^{\langle Sz \rangle} \otimes \bot)$. The resulting open proof goal is $1^{\langle Sx - 1 \rangle}, 1^{\langle Sy \rangle}, 1^{\langle Sz \rangle}, 1^{\langle Sw \rangle}, \bot^B$, which is easily seen to be provable, since the conditions on the instance require that $Sx + Sy + Sz + Sw = B + 1$, and so $(Sx - 1) + Sy + Sz + Sw = B$.

Given the $m$ proofs constructed as above from each of the $m$ groups of elements, one combines them with $\otimes$ into a proof of

$$\vdash (1^{\langle S1 \rangle} \otimes \bot), \ldots, (1^{\langle Sm \rangle} \otimes \bot), (1^3 \otimes \bot)^m.$$

The proof can then be completed with $3m$ applications of the $\otimes$ rule. □

2.6. Completeness

Lemma 2.3 (Completeness). For $A$, $m$, $B$, and $s$ satisfying the constraints of 432-Partition, if there is a proof of the COMLL formula $\Theta(\langle A, m, B, s \rangle)$, then the 432-Partition problem $\langle A, m, B, s \rangle$ is solvable.

Proof. The following makes heavy use of Lemma 2.1.

Assuming we have a proof of

$$\vdash (1^{\langle S1 \rangle} \otimes \bot) \otimes (1^{\langle S2 \rangle} \otimes \bot) \otimes \ldots \otimes (1^{\langle Sm \rangle} \otimes \bot) \otimes (\bot^B \otimes 1^{\langle S \rangle})^m,$$

we show that the corresponding 432-Partition problem is solvable.

If there is a proof of this sequent, then there is a cut-free proof, by the cut elimination theorem (Theorem 1.1). By repeated applications of Lemma 1.3, if there is a proof of this sequent, then there is a proof of $\vdash (1^{\langle X1 \rangle} \otimes \bot), (1^{\langle X2 \rangle} \otimes \bot), \ldots, (1^{\langle Xn \rangle} \otimes \bot), (\bot^B \otimes 1^{\langle S \rangle})^m$.

We then perform complete induction over a generalization of $m$. Consider, in general, sequents of the form $\vdash (1^{\langle X1 \rangle} \otimes \bot), (1^{\langle X2 \rangle} \otimes \bot), \ldots, (1^{\langle Xn \rangle} \otimes \bot), (\bot^B \otimes 1^{\langle S \rangle})^k$, where $k \leq m$, and the $Xj$ are a subset of the $Si$ and so satisfy $B/4 < Xj < B/2$. If $k > 1$, the
proof of this sequent must end in a use of the $\otimes$ rule, since all formulas have main
connective $\otimes$. We next show that the principal formula of that rule application must
be $(\bot^B \otimes 1^{(3)})^k$.

First, we note that each formula $(1^{(X_j)} \otimes \bot)$ has measure $X_j - 1$. Since $B > 8$, the
condition, $X_j > B/4$, ensures that for all $j$, $X_j > 2$, and therefore $X_j - 1 > 1$. There is only
one formula, $(\bot^B \otimes 1^{(3)})^k$, with negative measure.

If we assume that one of the $(1^{(X_j)} \otimes \bot)$ formulas is principal in an application
of $\otimes$, by Lemma 2.1, each hypothesis sequent must have measure one. In this case we
have the following supposed proof for some $\Sigma$ and $\Delta$ with the multiset union
$\Sigma \cup \Delta \cup (1^{(X_j)} \otimes \bot)$ being equal to the conclusion:

$$\vdash \Sigma, \bot \quad \vdash \Delta, 1^{(X_j)}$$

$$\vdash (1^{(X_j)} \otimes \bot), (1^{(X_2)} \otimes \bot), \ldots, (1^{(X_n)} \otimes \bot), (\bot^B \otimes 1^{(3)})^k$$

But $1^{(X_j)}$ which occurs in one hypothesis of the rule has measure $> 2$. Therefore, the
formula with negative measure, $(\bot^B \otimes 1^{(3)})^k$, must occur in $\Delta$. Now consider the other
hypothesis, which must contain $\bot$, and other formulas $\Sigma$ from the conclusion sequent.
If any formulas of the form $(1^{(X_j)} \otimes \bot)$ are included in $\Sigma$, the measure of that
hypothesis is greater than 1. If no such formulas are included, then the sequent has
measure 0. In either case, by Lemma 2.1, that sequent is not provable. Thus the
assumption that one of the $(1^{(X_j)} \otimes \bot)$ formulas is principal must be in error, and
$(\bot^B \otimes 1^{(3)})^k$ must be principal.

Thus if $k > 1$, the only possible next proof step is $\otimes$, with principal formula
$(\bot^B \otimes 1^{(3)})^k$. Any such application of the $\otimes$ rule partitions the elements of the sequent
into two groups and splits $(\bot^B \otimes 1^{(3)})^k$ into two fragments of the same form. Using this
result, we see that the proof under consideration must repeatedly refine a partition of
the elements $(1^{(S_j)} \otimes 1)$ by splitting a group into two smaller pieces under control of
an element of the form $(\bot^B \otimes 1^{(3)})^k$, where initially $k = m$. It is easy to see that these
controlling elements limit the splitting process so that at most $m$ groups can be
created. If all groups with $k > 1$ have been eliminated, then there will be exactly
$m$ groups in the partition.

We may then focus on the case when $k = 1$ which will correspond to a small leaf
branch of the proof tree. We claim that each such branch in the proof corresponds to
one partition in the solution of the original 432-Partition problem. That is, we claim
that when $k = 1$, we must be left with a sequent of the form:

$$\vdash (1^{(X_1)} \otimes \bot), (1^{(X_2)} \otimes \bot), \ldots, (1^{(X_n)} \otimes \bot), (\bot^B \otimes 1^{(3)})$$

where the $X_j$ are a subset of the $S_i$, and no $X_j$ appears in more than one such sequent.
There are exactly $B + n - 1$ occurrences of $\otimes$ in this sequent, and $\sum_{1 \leq i \leq n} X_i + 3$ ones
in this sequent. By Lemma 2.1, $(\sum_{1 \leq i \leq n} X_i + 3) - (B + n - 1) = 1$, or equivalently
$\sum_{1 \leq i \leq n} X_i = B + n - 3$. This gives rise to the conditions for one group of a partition.
Constant-only multiplicative linear logic is NP-complete

that is a solution of the 432-Partition problem instance. (The conditions on the sizes $S_i$ are sufficient to ensure that $2 \leq n \leq 4$ as in Section 2.2.)

Thus, given any proof of $\Theta(\langle A, m, B, s \rangle)$, we first see that one may identify $m$ branches, each of which is of the form $\Gamma - (1^{X_1} \otimes 1), (1^{X_2} \otimes 1), \ldots, (1^{X_n} \otimes \bot), (\bot^B \otimes 1^{(3)})$. From these $m$ branches, we may identify $m$ groups of 4, 3, or 2 elements of the associated 432-Partition problem. In other words, from any proof of the given sequent, one may construct a solution to the 432-Partition problem. □

2.7. Main result

From the preceding, soundness and completeness, we immediately achieve our stated result.

**Theorem 2.4** (COMLL NP-complete). *The decision problem for constant-only multiplicative linear logic is NP-complete.*

Also, with an easy conservativity result, we find that this NP-hardness proof suffices for multiplicative linear logic as well.

**Theorem 2.5** (Conservativity). *Multiplicative linear logic is conservative over constant-only multiplicative linear logic.*

**Proof.** By induction on cut-free MLL proofs. □

3. Alternate encodings

Consider the earlier encodings of 3-Partition in full multiplicative linear logic:

\[
[(k \multimap c^{(a_1)}) \otimes \cdots \otimes (k \multimap c^{(a_m)}) \otimes (c^B \multimap j)^m] \multimap (k^B \multimap j)^m.
\]

Constant-only encodings can be generated by replacing $c$ by bottom, and $k$ by $1^{(C)}$ for some integer $C$. A value of $C$ that is particularly interesting is $C = \sum_{a \in A} s(a)$. Although they are still polynomial, such encodings tend to be larger than the one advocated above, and result in somewhat less complicated proofs of soundness. The case of $C = 1$ is an incorrect encoding, and one may consider the “bottom only” encoding proved sound and complete above to be generated from the case $C = 0$.

4. Conclusion

We have demonstrated that simply evaluating expressions in true, false, and, and or in multiplicative linear logic ($1$, $\bot$, $\otimes$, and $\forall$) is NP-complete. By conservativity results, the NP-hardness of larger fragments of linear logic follow, although some of
these results were known previously. These results constitute further dramatic evidence of the extreme expressive power of linear logic. Other results along these lines have previously shown that full propositional linear logic is undecidable, and that there are natural fragments which are PSPACE-complete, EXPTIME-complete, and NP-complete.

Complexity results for fragments of linear logic indicate the difficulty of constructing efficient decision procedures for large fragments of linear logic. It may have been hoped previously that some “semantic” measure condition could be used to immediately decide constant-only expressions in linear logic. When constructing theorem provers for linear logic, one must consider carefully the sources of exponential blowup identified by these suites of results (in this case, the splitting of contexts in applications of the \(\otimes\) rule). In constructing such theorem provers, conditions such as Lemma 2.1 may be useful in pruning proof search trees.

However, it is still essentially unknown how to harness the evident power of linear logic for useful purposes. Several interesting attempts have been made, including using linear logic as the basis for a logic programming language [9, 2, 17, 18], and as the basis for a functional programming language [1, 14]. The results given here have more direct impact on the logic programming approach, which is still in its infancy.

References

[10] M. Kanovich, The multiplicative fragment of linear logic is NP-complete, Electronic mail submission to the Linear Logic mailing list, 1991.
Constant-only multiplicative linear logic is NP-complete


