Provability in TBLL: A Decision Procedure *

Jawahar Chirimar
CIS
University of Pennsylvania
and
James Lipton
Dept. of Mathematics
Wesleyan University

Abstract

We prove the decidability of the Tensor-Bang fragment of linear logic and establish upper (doubly exponential) and lower (NP-hard) bounds.

1 Introduction

Since Lincoln et al. [9] discovered in 1990 that propositional linear logic is undecidable, there has been a great deal of interest in determining the complexity of different fragments. Gehlot and Gunter [2, 5] established in 1991 that provability in the tensor fragment of linear logic with proper axioms (PrTLL) was equivalent to the reachability problem for Petri nets, whose decidability was established in 1982 in a difficult paper [8] by Kosaraju. It follows easily that the multiplicative-bang ($\otimes$,$\multimap$, !) fragment, MBLL (also called MELL) can encode Petri Net reachability. On the other hand it is known that MALL (the exponent-free or multiplicative-additive fragment of linear logic) is decidable (and PSPACE-complete) and that MLL is NP-complete [7] (see also below). It is thus of great interest to understand the "boundary" region (TBLL, PrTLL, MBLL) both to pin down as much as possible where undecidability happens as possibly to shed more light on the complexity of Petri Net reachability from a linear logic perspective.

Here we investigate the first such fragment, using a direct analysis of deduction somewhat in the spirit of formal language theory. We begin with the ground rules. The tensor-bang fragment of linear logic (TBLL) is comprised of the following symbols and rules of inference. The **language** consists of a (linearly ordered) set Σ of *propositional letters* $\{A_1, \ldots, A_n\}$. The *logical constants* are the binary connective *tensor* ($\otimes$) and the unary (prefix) connective *bang*[1] (!).

*An abbreviated version appeared in the proceedings of *Computer Science Logic (CSL)* '91, LNCS 626, Springer, 1992.

[1]also called "of course"

Rules of Inference: Sequents $\Gamma \vdash \theta$ consist of finite sequences Γ of propositions (separated by commas) and single propositions θ. In the rule called "axiom", only propositional letters occur.

$$A \vdash A \qquad \text{"axiom"}$$

$$\frac{\Gamma, \varphi, \psi, \Lambda \vdash \varphi}{\Gamma, \varphi \otimes \psi, \Lambda \vdash \varphi} \quad \otimes \text{ intro-left} (\otimes L) \qquad\qquad \frac{\Gamma \vdash \varphi \qquad \Lambda \vdash \psi}{\Gamma, \Lambda \vdash \varphi \otimes \psi} \quad \otimes \text{ intro-right} (\otimes R)$$

$$\frac{\Gamma, \theta \vdash \varphi}{\Gamma, !\theta \vdash \varphi} \quad \text{Dereliction} (\mathbf{D}) \qquad\qquad \frac{\Gamma, !\theta, !\theta \vdash \varphi}{\Gamma, !\theta \vdash \varphi} \quad \text{Contraction} (\mathbf{C})$$

$$\frac{\Gamma \vdash \varphi}{\Gamma, !\theta \vdash \varphi} \quad \text{Weakening} (\mathbf{W}) \qquad\qquad \frac{!\Gamma \vdash \varphi}{!\Gamma \vdash !\varphi} \quad !\text{-introduction } (!I)$$

The only structural rules are *exchange* $(\times)$:

$$\frac{\Gamma, \varphi, \psi, \Delta \vdash \theta}{\Gamma, \psi, \varphi, \Delta \vdash \theta}$$

and **cut**

$$\frac{\Gamma \vdash \theta \qquad \Delta, \theta \vdash \varphi}{\Gamma, \Delta \vdash \varphi}.$$

The following result will simplify our work below.

Theorem 1.1 *TBLL admits cut-elimination.*

The proof is almost identical to that for the full propositional logic. See e.g. [4].

2 A decision algorithm for deducibility in TBLL

To begin with we will need a series of translations and technical lemmas to define our main algorithm.

Definition 2.1 *A formula α is in* **TLL** *(tensor-linear logic) if it is built up from propositional letters using only the connective $\otimes$.*

Definition 2.2 *A formula in TBLL is called banged if it is of the form $!\alpha$ where α is any TBLL-formula, or if it is of the form $\alpha \otimes \beta$ where α and β are banged*[2]*. For a multiset Γ, $\otimes\Gamma$ is the tensor product of all the formulas in Γ. A multiset Γ is banged if $\otimes\Gamma$ is.*

The following properties of banged formulas will be useful.

Lemma 2.3 *The weakening and contraction rules of inference remain valid for arbitrary banged formulas β in place of $!\theta$ above. We can also replace the premiss in the rule of !-introduction by any banged formula.*

[2]This agrees with Girard's recent definition of *positive polarity* of a formula

The proof is immediate by structural induction.

We define a canonical form which will be repeatedly used in the sequel.

Lemma 2.4 (decomposition lemma) *Every TBLL formula α is either*

1. *in TLL or*
2. *banged or*
3. *equivalent (with respect to TBLL-deducibility) to a formula of the form $\alpha_0 \otimes \alpha_1$ where α_0 is in TLL and α_1 is banged.*

proof: If (1) and (2) do not hold, then α is of the form $\gamma \otimes \delta$ where not both γ and δ are in TLL. Inducting on length of formulas, and applying the tensoring rules of inference, there is a formula $\gamma_0 \otimes \gamma_1 \otimes \delta_0 \otimes \delta_1$, with γ_0 and δ_0 in TLL, γ_1 and δ_1 banged, which is equivalent to α (the cases γ in TLL, δ in TLL are left to the reader). Therefore $\alpha \equiv (\gamma_0 \otimes \delta_0) \otimes (\gamma_1 \otimes \delta_1)$. ∎

We will call case (3) in the statement of the previous lemma a **decomposition** of a TBLL formula α. The first formula will be called the *unbanged* or *pure tensor part* of the decomposition. Formulas in case (3) are said to be nontrivial, or to have a nontrivial decomposition.

We first tackle the problem of deciding when one can deduce sequents of the form $\Gamma \vdash \alpha$ with Γ in TBLL and α in TLL. The number of occurrences of each letter of Σ in α will help determine the possible replications (contractions) of banged formulas in the antecedent required in the associated deduction we are trying to reconstruct. In order to make this precise we define the notion of an **instance** of a TBLL formula, namely a certain expression obtained by "decorating" the ! symbols with natural numbers.

Definition 2.5 (instances) *Let γ be a TBLL formula.*

1. *If γ is an atom A, then A is the sole* **instance** *of γ.*
2. *$\frac{!}{\mathbf{0}}(\delta')$ is an instance of $!\delta$ if and only if (δ') is an instance of δ and every ! in the subexpression (δ') is instantiated to $\mathbf{0}$.*
3. *$\frac{!}{\mathbf{m}}(\delta')$ $(m > 0)$ is an* **instance** *of $!\delta$ if δ' is an instance of δ and m is a natural number.*
4. *$\delta' \otimes \gamma'$ is an* **instance** *of $\delta \otimes \gamma$ if δ' (resp. γ') is an instance of δ (resp. γ).*

It will also be convenient to define a symbolic version of an instance.

Definition 2.6 *A* **symbolic instance** *of a TBLL formula is an indexing of every ! with a fresh variable.*

Note that if we pick a standard list of variables, and demand that consecutive variables be used as we proceed along the formula from left to right, a symbolic instance of a formula is unique. We now define a translation from TBLL formulas to a "polynomial expression" in the propositional letters of the language.

Definition 2.7 **Σ-polynomial expressions** *are (ordered) expressions of the form*

$$A_1^{e_1} A_2^{e_2} \cdots A_n^{e_n}$$

where $A_i \in \Sigma$, $A_i < A_{i+1}$ in the Σ-order, and each e_i is a sum $\sum_{ij} e_{ij}$ of terms of the form k_{ij} or $k_{ij}x_{ij}$ where k_{ij} is a natural number and x_{ij} a variable. The product of two such expressions $A_1^{i_1} A_2^{i_2} \cdots A_n^{i_n}$ and $A_1^{j_1} A_2^{j_2} \cdots A_n^{j_n}$ is $A_1^{i_1+j_1} A_2^{i_2+j_2} \cdots A_n^{i_n+j_n}$.

Let θ' be a symbolic instance of a proposition of TBLL. We define the **polynomial form** *of θ' to be the following Σ-polynomial expression $p(\theta')$, by induction on the structure of θ':*

1. $p(A) \stackrel{\text{def}}{\equiv} A$ *for* $A \in \Sigma$.
2. $p(\beta' \otimes \gamma') \stackrel{\text{def}}{\equiv} p(\beta')p(\gamma')$
3. $p(\underset{\mathbf{x}}{!}\gamma') \stackrel{\text{def}}{\equiv} p(\gamma')^{(x)}$ *where we define the* **exponent** $(\theta')^{(x)}$ *as follows:*
   - $A^{(x)} \stackrel{\text{def}}{\equiv} A^x$
   - $(uv)^{(x)} \stackrel{\text{def}}{\equiv} u^{(x)} v^{(x)}$
   - $(u^{(y)})^{(x)} \stackrel{\text{def}}{\equiv} u^{(y)}$

We will also call such an expression a polynomial form of the original TBLL-formula θ itself.

Note in the preceding definition that p and the exponent are defined via a series of rewrite rules. Strictly speaking $p(\theta)$ is the polynomial obtained by applying the above rewritings until a polynomial expression results (which is unique, once Σ is ordered, up to the names of the variables). Also note that the translation proceeds somewhat like an instantiation of regular expressions (with ! taking the place of $*$) except for the unusual iterated exponentiation "law" $(u^{(y)})^{(x)} \stackrel{\text{def}}{\equiv} u^{(y)}$ Note also that for α in TLL, $p(\alpha)$ has no variables, and is essentially a lexicographical re-ordering of α. When instantiated with natural numbers, the exponents of the polynomials just defined will correspond to certain choices of instantiations (copies) of banged formulas in the proof theory.

We are now in a position to define a *string test* which will be repeatedly used in our decision procedure, and which is, as we shall see below, sufficient to decide the special case $\Gamma \vdash \alpha$ when α is in TLL.

Definition 2.8 *Let Γ and α be propositions in TBLL and TLL respectively. Then the pair $\langle \Gamma, \alpha \rangle$ is said to satisfy the* **string test** *(in symbols $\Gamma \models_s \alpha$) if*

1. *There are natural number values for the variables in the polynomials $p(\otimes(\Gamma))$, $p(\alpha)$ such that the following equation has a solution:*

$$p(\otimes(\Gamma)) = p(\alpha) \tag{1}$$

and

2. *These numerical values for the variables in $p(\otimes(\Gamma))$ correspond to a legal instance (in the sense of definition 2.5) of the original formula (multiset) Γ when substituted for the same variables in the corresponding symbolic instance of θ. In particular any solution to 1 that results in the assignment of a nonzero decoration to a bang within the scope of a $\frac{!}{\mathbf{0}}$ does not qualify.*

We remark that certain variables from the symbolic instance of θ may not occur in $p(\theta)$ because of the collapse of exponents. This is always the case when there is a subformula of θ of the form $!\alpha$ where α is banged. In this case, our notion of "checking the legality" of a solution to equation (1) requires comment. We first replace those variables which occur in the polynomial equation with their numerical values. Then all uninstantiated variables in the scope of a $\frac{!}{\mathbf{0}}$ are set equal to 0. All the remaining ones are set equal to 1.

We will call an instantiation of the variables in $p(\theta)$ a *polynomial instance* for θ. If α is in TLL, Γ is in TBLL, and $\Gamma \models_s \alpha$, we call the corresponding instance of Γ the one "induced by the string test" $\Gamma \models_s \alpha$.

In the next definition and the lemma that follows it, it will be convenient to introduce the TLL-*unit* $\mathbf{1}$. For the purposes of the lemma we add the axiom $\vdash \mathbf{1}$ as a legal proof rule to the rules of TBLL. We call the resulting fragments T(B)LL$\mathbf{1}$.

Definition 2.9 *Every instance θ has an associated* **TLL-normal form**, *namely the TLL$\mathbf{1}$ formula obtained as follows.*

1. $T(A) \stackrel{\text{def}}{\equiv} A$ *for* $A \in \Sigma$.
2. $T(\beta \otimes \gamma) \stackrel{\text{def}}{\equiv} T(\beta) \otimes T(\gamma)$
3. $T(\frac{!}{\mathbf{0}}\gamma) \stackrel{\text{def}}{\equiv} \mathbf{1}$
4. $T(\frac{!}{\mathbf{m}}\gamma)$ $(0 < m) \stackrel{\text{def}}{\equiv} \underbrace{\alpha \otimes \cdots \otimes \alpha}_{\text{m times}} \otimes T(\beta)$ *where $\alpha \otimes \beta$ is the canonical decomposition of γ (i.e. α in TLL and β banged).*

We lift the definition of instances to multisets of formulas in the obvious way.

Lemma 2.10 *Let γ be in TBLL. Then for any instance γ' of γ. the following sequent is derivable in TBLL$\mathbf{1}$: $\gamma \vdash T(\gamma')$.*

proof: The proof is by induction on the structure of γ

1. If γ is an atom A then its instance is A. The conclusion is immediate.

2. Suppose γ is $\phi \otimes \delta$. By definition we have that $T(\phi' \otimes \delta') = T(\phi') \otimes T(\delta')$. The result is thus obtained by induction hypothesis and $\otimes - R$.

3. Suppose γ is $!\phi$. By definition we have that $T(\underset{\mathbf{0}}{!}\phi') = \mathbf{1}$. But then we have the deduction

$$\frac{\vdash \mathbf{1}}{!\phi \vdash \mathbf{1}}\mathrm{W}$$

If $m > 0$ we have

$$T(\underset{\mathbf{m}}{!}\phi') = \underbrace{\alpha \otimes \cdots \otimes \alpha}_{\text{m times}} \otimes T(\beta')$$

where $\alpha \otimes \beta$ is a canonical decomposition of ϕ, and β' is an instance of β. We take m to be 2 for ease of notation below. The induction hypothesis $\beta \vdash T(\beta')$ along with uses of $\otimes - R$ and the derived rule of weakening for banged formulas completes the proof shown below.

$$\cfrac{\cfrac{\cfrac{\alpha, \beta, \alpha, \beta \vdash \alpha \otimes \alpha \otimes T(\beta')}{\phi, \phi \vdash \alpha \otimes \alpha \otimes T(\beta')}(\otimes{-}L)}{!\phi, !\phi \vdash \alpha \otimes \alpha \otimes T(\beta')}(D)}{!\phi \vdash \alpha \otimes \alpha \otimes T(\beta')}(C)$$

∎

We give an example:

$$p(A \otimes !(B \otimes !A) \otimes (!(A \otimes !B \otimes B))) = p(A \otimes \underset{\mathbf{x}}{!}(B \otimes \underset{\mathbf{y}}{!}A) \otimes (\underset{\mathbf{z}}{!}(A \otimes \underset{\mathbf{u}}{!}B \otimes B))) = A^{1+y+z}B^{x+u+z}. \quad (2)$$

If we take, e.g., $x = 2$, $y = 2$, $z = 0$ and $u = 0$ we obtain the *instance*

$$A \otimes \underset{\mathbf{2}}{!}(B \otimes \underset{\mathbf{2}}{!}A) \otimes (\underset{\mathbf{0}}{!}(A \otimes \underset{\mathbf{0}}{!}B \otimes B))$$

The corresponding polynomial instance is A^3B^2 and the associated tensor normal form is

$$A \otimes A \otimes A \otimes B \otimes B.$$

We consider a join operation on instances.

Definition 2.11 (join) *Let C', C'' be instances of the TBLL-formula C. We define the* **join** *$C' \bowtie C''$ as follows:*

- *If C is an atom A, $C \bowtie C = C$*

- *If C is of the form $!\theta$, with $C' = \underset{\mathbf{m}}{!}\theta'$ and $C'' = \underset{\mathbf{n}}{!}\theta''$ then*

$$\underset{\mathbf{m}}{!}\theta' \bowtie \underset{\mathbf{n}}{!}\theta'' = \underset{\mathbf{m+n}}{!}\theta' \bowtie \theta''$$

- *If C is of the form $\alpha \otimes \beta$, and $C' = \alpha' \otimes \beta'$ and $C'' = \alpha'' \otimes \beta''$ then*

$$(\alpha' \otimes \beta') \bowtie (\alpha'' \otimes \beta'') = (\alpha' \bowtie \alpha'') \otimes (\beta' \bowtie \beta'')$$

Observe that if α is in TLL then $\alpha \bowtie \alpha$ is α.

Lemma 2.12 (TLL-join lemma) *Let $\gamma'$and γ'' be instances of the same* **banged** *formula γ*

$$T(\gamma' \bowtie \gamma'') \vdash T(\gamma') \otimes T(\gamma'')$$

proof: Suppose $\gamma = !\varphi$. Then γ', γ'' are of the form $\underset{\mathbf{m}}{!}\varphi'$ and $\underset{\mathbf{n}}{!}\varphi''$ for some m and n. Then we have

$$T(\gamma' \bowtie \gamma'') = T(\underset{\mathbf{m}}{!}\varphi' \bowtie \underset{\mathbf{n}}{!}\varphi'') = T(\underset{\mathbf{n+m}}{!}\varphi' \bowtie \varphi'') \tag{3}$$

Let $\alpha \otimes \beta$ be the canonical decomposition of φ. Then $\varphi' \bowtie \varphi''$ must be equivalent to $\alpha \otimes (\beta' \bowtie \beta'')$ where β', β'' are the instances of β occurring in the instances φ', φ''. By definition of T, the expressions in (3) are equivalent to

$$\underbrace{\alpha \otimes \cdots \otimes \alpha}_{m+n \text{ times}} \otimes T(\beta' \bowtie \beta'').$$

This is easily shown equivalent to $T(\underset{\mathbf{m}}{!}\varphi') \otimes T(\underset{\mathbf{n}}{!}\varphi'')$ once we apply the definition of T and the induction hypothesis to β' and β''. The case $\theta = \delta \otimes \gamma$ is a straightforward induction and rearrangement of tensorands. The reader can easily check that in the $\underset{\mathbf{0}}{!}$ cases the appropriate TBLL1 sequent is provable. In particular if the left hand side is $\mathbf{1}$ or a tensor of $\mathbf{1}$'s, the right hand side must also be of this form. ∎

Theorem 2.13 *Let α be a proposition in TLL logic, and Γ a formula in TBLL. Then*

$$\Gamma \vdash \alpha \qquad \text{iff} \qquad \Gamma \models_s \alpha.$$

proof: We establish "soundness" ($\Gamma \vdash \alpha \Rightarrow \Gamma \models_s \alpha$) by induction on the length of proofs.

If $\Gamma \vdash \alpha$ is a one step proof by "axiom": $A \vdash A$ then the polynomials are identical.

Now suppose that $\Gamma \vdash \alpha$ is a proof with $n + 1$ steps and inductively assume soundness holds for all proofs of shorter length. We consider the possible cases for the last step of the proof.

case $\otimes R$ The last step in the proof is an inference of the form

$$\frac{\Delta \vdash \beta \qquad \Lambda \vdash \gamma}{\Delta, \Lambda \vdash \beta \otimes \gamma}.$$

By the induction hypothesis and the definition of p the result is immediate.

case $\otimes - L$ Soundness for tensor left is built into the definition of the string-test.

case D The last step is an inference using *dereliction*:

$$\frac{\Delta, \theta \vdash \alpha}{\Delta, !\theta \vdash \alpha}.$$

we can obtain a solution to $p(\Delta \otimes !\theta) = p(\alpha)$ by using the instantiations obtained by inductive hypothesis, extended to the new premiss by taking the variable corresponding to the introduced ! to be 1 (assuming it is not eliminated due to collapse of exponents).

case C Suppose the last step in the proof is an instance of the rule of *contraction*:

$$\frac{\Gamma, !\theta, !\theta \vdash \alpha}{\Gamma, !\theta \vdash \alpha}$$

By induction there is a polynomial instance of the left hand side of the sequent $\Gamma, !\theta, !\theta \vdash \alpha$ agreeing with $p(\alpha)$ and satisfying the legality criterion of the string test. Note that the polynomial instances u and v corresponding to the first and second indicated occurrences of $!\theta$ may be different. Let $i_1 \ldots, i_n$ and $j_1 \ldots, j_n$ be the corresponding instantiations of exponents in u and v. Observe that the sequence of natural numbers $i_1 + j_1, \ldots, i_n + j_n$ instantiating the polynomial associated with the indicated occurrence of $!\theta$ in the sequent $\Gamma, !\theta \vdash \alpha$, satisfies

$$p(\Gamma, !\theta) = p(\alpha).$$

The instance is legal, since, if some $\frac{!}{i_r + j_r}$ occurs on the scope of a $\frac{!}{i_s + j_s}$ where $i_s + j_s = 0$ then $i_s = j_s = 0$ so by legality of the original solutions, $i_r = j_r = 0$.

case W Suppose the last rule used was *weakening*:

$$\frac{\Gamma, \vdash \alpha}{\Gamma, !\theta \vdash \alpha}$$

by inductive hypothesis we have a solution to

$$p(\Gamma) = p(\alpha)$$

By instantiating all the variables associated with the indicated $!\theta$ to 0 we get a solution for

$$p(\Gamma \otimes !\theta) = p(\alpha)$$

case $\times$ This is true by induction hypothesis and the fact that the polynomials are unaffected by exchange.

Now we prove the other direction: "completeness" ($\Gamma \models_s \alpha \;\Rightarrow\; \Gamma \vdash \alpha$), i.e., if there is a solution to $p(\Gamma) = p(\alpha)$ then $\Gamma \vdash \alpha$. Lemma 2.10 gives us that $\Gamma \vdash T(\Gamma')$, for any instance Γ' of Γ. It is easy to show that the instance obtained by the String Test is some permutation of α, thus Γ proves $T(\Gamma')$ and $T(\Gamma')$ proves α. Hence by Cut we are done. ▮

The string test algorithm

The string test problem, and hence the decidability of the TBLL-TLL -fragment is NP-Complete.

Lemma 2.14 (NP-hardness) *Provability in the TBLL/TLL-fragment is NP-hard.*

proof: The *set cover* decision problem is a well known NP-complete problem (see e.g. [1, 6]). We give a brief description here. Let $F = \{S_j\}$ be a finite family of finite sets. Let T be a subset of F. We say T is a **cover** of F if

$$\bigcup_{S_i \in T} S_i = \bigcup_{S_i \in F} S_i.$$

Let k be a natural number. The set cover decision problem is, given inputs $\langle F, k\rangle$, to determine if there is a cover of F containing k sets.

We now show that this problem can be reduced to provability in TBLL/TLL. Suppose $F = \{S_j : 1 \le j \le n\}$ and

$$S_j = \{s_{j1}, \ldots, s_{jt_j}\} \qquad (1 \le j \le n)$$

and define $\Sigma = \bigcup_1^n S_j$. Suppose $\Sigma = \{a_1, \ldots, a_l\}$. Introduce propositional letters $A_1, \ldots, A_l$ (one for each of the elements a_i), $B_1, \ldots, B_n$ (one for each of the members of F) and an additional letter Z. Then we claim that the set-cover problem $\langle F, k\rangle$ has a positive solution if and only if the following sequent is derivable in TBLL:

$$\begin{array}{l}(!B_1 \otimes \cdots \otimes !B_n) \otimes \\ \quad ![(!s_{11} \otimes \cdots \otimes !s_{1t_1}) \otimes Z \otimes B_1] \otimes \cdots \otimes ![(!s_{n1} \otimes \cdots \otimes !s_{nt_n}) \otimes Z \otimes B_n] \\ \qquad \vdash \\ \qquad\quad (A_1 \otimes A_2 \otimes \cdots \otimes A_l) \otimes Z^{\mathbf{k}} \otimes (B_1 \otimes \cdots \otimes B_n).\end{array} \tag{4}$$

In order to discuss our claim it will be convenient to selectively instantiate certain of the ! symbols in (4). Consider

$$\begin{array}{l}(!B_1 \otimes \cdots \otimes !B_n) \otimes \\ \quad \frac{!}{\xi_1}[(!s_{11} \otimes \cdots \otimes !s_{1t_1}) \otimes Z \otimes B_1] \otimes \cdots \otimes \frac{!}{\xi_n}[(!s_{n1} \otimes \cdots \otimes !s_{nt_n}) \otimes Z \otimes B_n] \\ \qquad \vdash \\ \qquad\quad (A_1 \otimes A_2 \otimes \cdots \otimes A_l) \otimes Z^{\mathbf{k}} \otimes (B_1 \otimes \cdots \otimes B_n).\end{array} \tag{5}$$

Observe that if the sequent is provable, only instantiations of 0 or 1 for the ξ_i could have occurred in the string test. Otherwise one of the B_i would occur more than once on the left and only once on the right. Observe that the presence of $(!B_1 \otimes \cdots \otimes !B_n)$ in the top row of the sequent is to supply one copy of B_i if the instantiation of ξ_i induced by the string test is 0. An instantiation of ξ_i to 1 corresponds to the inclusion of the ith set in the cover. We also note that the letters $A_1, \ldots, A_l$ each occur only once on the right hand side of the sequent, but may occur more than once (as one of the s_{ij}) on the left. This is the reason each s_{ij} occurs with a ! in each of the tensorands $[(!s_{i1} \otimes \cdots !s_{it_i}) \otimes Z \otimes B_i]$, so that it may be instantiated to 0 if the letter has already occurred as s_{uv} for some $u < i$. We illustrate with an example and leave the details to the reader. Let

$$F = \left\{ \begin{array}{l} \{1,2,4,7\} \\ \{2,6,7,8\} \\ \{1,3,6\} \\ \{1,2,3,4\} \\ \{2,6,8\} \end{array} \right\}$$

and let $\mathbf{k}= 3$. The set-cover problem for $\langle F, 3\rangle$ has the positive solution $\{1,2,4,7\}\{2,6,7,8\}\{1,3,6\}$, the union of which is the set $\{1,2,3,4,6,7,8\}$. The test-sequent for this problem is produced as follows. We introduce letters $A_1, A_2, A_3, A_4, A_6, A_7, A_8$ corresponding to the members of $\bigcup F$, as well as $B_1, \ldots, B_5$ and Z as described above. The associated sequent, then, is

$$\begin{array}{l} \quad\quad (!B_1 \otimes \cdots \otimes !B_5) \otimes \\ ![(!A_1 \otimes !A_2 \otimes !A_4 \otimes !A_7) \otimes Z \otimes B_1] \otimes ![(!A_2 \otimes !A_6 \otimes !A_7 \otimes !A_8) \otimes Z \otimes B_2] \otimes \\ ![(!A_1 \otimes !A_3 \otimes !A_6 \otimes) \otimes Z \otimes B_3] \otimes \\ [(!A_1 \otimes !A_2 \otimes !A_3 \otimes !A_4) \otimes Z \otimes B_4] \otimes ![(!A_2 \otimes !A_6 \otimes !A_8 \otimes) \otimes Z \otimes B_5] \end{array} \tag{6}$$

$$\vdash$$

$$(A_1 \otimes A_2 \otimes A_3 \otimes A_4 \otimes A_6 \otimes A_7 \otimes A_8) \otimes Z^3 \otimes (B_1 \otimes \cdots \otimes B_5). \tag{7}$$

It is easy to check that the following instantiations, which reflect the set-cover solution given above satisfy the string test:

$$\begin{array}{l} \quad (\frac{!}{\mathbf{0}}B_1 \otimes \frac{!}{\mathbf{0}}B_2 \otimes \frac{!}{\mathbf{0}}B_3 \otimes \frac{!}{\mathbf{1}}B_4 \otimes \frac{!}{\mathbf{1}}B_5) \otimes \\ \frac{!}{\mathbf{1}}\left[(\frac{!}{\mathbf{1}}A_1 \otimes \frac{!}{\mathbf{1}}A_2 \otimes \frac{!}{\mathbf{1}}A_4 \otimes \frac{!}{\mathbf{1}}A_7) \otimes Z \otimes B_1\right] \otimes \frac{!}{\mathbf{1}}\left[(\frac{!}{\mathbf{0}}A_2 \otimes \frac{!}{\mathbf{1}}A_6 \otimes \frac{!}{\mathbf{0}}A_7 \otimes \frac{!}{\mathbf{1}}A_8) \otimes Z \otimes B_2\right] \otimes \\ \frac{!}{\mathbf{1}}\left[(\frac{!}{\mathbf{0}}A_1 \otimes \frac{!}{\mathbf{1}}A_3 \otimes \frac{!}{\mathbf{0}}A_6 \otimes) \otimes Z \otimes B_3\right] \otimes \\ \frac{!}{\mathbf{0}}[(!A_1 \otimes !A_2 \otimes !A_3 \otimes !A_4) \otimes Z \otimes B_4] \otimes \frac{!}{\mathbf{0}}[(!A_2 \otimes !A_6 \otimes !A_8 \otimes) \otimes Z \otimes B_5] \end{array} \tag{8}$$

$$\vdash$$

$$(A_1 \otimes A_2 \otimes A_3 \otimes A_4 \otimes A_6 \otimes A_7 \otimes A_8) \otimes Z^3 \otimes (B_1 \otimes \cdots \otimes B_5). \tag{9}$$

∎

Lemma 2.15 (NP-completeness) *Provability in the TBLL/TLL-fragment is NP-complete.*

proof: This is almost immediate from the discussion of the deterministic string-algorithm in the next paragraph. To show a test sequent $\Gamma \vdash \alpha$ derivable one has to guess the value (in the equations (10) below) of as many variables as there are ! symbols in the sequent. Each value is bounded by the number of occurrences of letters in α hence both the number of guesses and the values are bounded by the length of the test sequent. ∎

We can describe a deterministic string test algorithm informally as follows. The input is an ordered pair $\langle \Gamma, \alpha \rangle$ with Γ in TBLL and α in TLL. We then compute the *polynomial forms* $p(\alpha)$ and $p(\otimes\Gamma)$ associated with these formulas (see definition 2.7). For α in TLL, $p(\alpha)$ is an expression of the form

$$A_1^{f_1} A_2^{f_2} \cdots A_n^{f_n}$$

and $p(\Gamma)$ is of the form

$$B_1^{e_1} B_2^{e_2} \cdots B_m^{e_m}$$

where the $f_i > 0$ are natural numbers and the e_j are sums of terms of the form k_{ij} or $k_{ij}x_{ij}$.

The string test algorithm checks that every A_i is among the B_j. If A_i matches B_j then we get an equation $f_i = e_j$ else it fails as $f_i = 0$ has no solution. If all letters in A_i's are found among the B_j's then we get a set of equations of the following kind.

$$\begin{aligned} f_1 &= e_{i_1} && (10)\\ &\vdots \\ f_n &= e_{i_n} \\ &\vdots \\ 0 &= e_{i_m} && (11) \end{aligned}$$

Then it tries to solve these equations using bounded search, the bound being given by f_j for the variables in e_{i_j}. Moreover the algorithm checks that the solution corresponds to a legal instantiation of Γ (in the sense of definition 2.5).

The number of variables in e_i is bounded by b_Γ, the number of bangs in Γ and f_i are bounded by l_α, the length of α. Thus the search is bounded by $O(l_\alpha^{b_\Gamma})$.

An Example

Consider the candidate sequent "$A \otimes !(B \otimes !A) \otimes (!(A \otimes !B \otimes B)) \vdash A \otimes (B \otimes A)$." Applying the string test reduces to finding if there are $x, y, z, u \in \mathcal{N}$ for which

$$\begin{aligned} 2 &= 1 + y + z \\ 1 &= x + u + z \end{aligned}$$

$x = 1, y = 1, u = z = 0$ is a solution, and a legal instance, so the test **succeeds**, showing that the sequent is in fact derivable. Of course, the test can always be performed by a bounded search. Each equation generated is of the form

$$k = a_1 x_1 + a_2 x_2 + \cdots + a_m x_m$$

where each a_i on the right is a natural number, and each x_i on the right satisfies $0 \leq x_i \leq k$.

The following example suggests that there is no immediate generalization of the string test to deal with the general case of TBLL-sequents. Consider $\Gamma :=!(A \otimes A \otimes A)\otimes!(A \otimes A)$ and $\theta := A \otimes A\otimes!A$.

The set of polynomial instances of Γ is $\{A^{3x+2y} : x, y \in \mathbf{N}\}$ which is precisely the same set $\{A^{x+2z} : z \in \mathbf{N}\}$ of instances associated with θ (Using a little algebra: the set $\{3x + 2y : x, y \in \mathbf{N}\}$ is the (semi)- ideal of $\mathbf{N}$ generated by gcd(3,2) intersected with the set of numbers greater than (3-1)(2-1) = 2). However $\Gamma \not\vdash \theta$ and $\theta \not\vdash \Gamma$, as can be checked using the algorithm described next.

Now we are ready to deal with general TBLL-sequents.

Definition 2.16 (reductions) *Let θ' be an instance of a TBLL-formula φ (i.e. a decoration of every ! in the original formula with a numerical subscript, as in definition 2.5). We abuse language and define a* **decomposition** *of the instance θ' to be the "decorated" decomposition of the original formula φ in which we retain the subscripts of the instance θ'. Let the* **reduced form** *$r(\theta')$ be the following associated TBLL-formula defined by induction on the structure of θ'. Primed formulas denote instances, and are dropped for TLL-formulas, (whose sole instances are themselves).*

1. $r(a) = a$ *(a an atom).*
2. $r(\varphi' \otimes \psi') = r(\varphi') \otimes r(\psi')$
3. $r(\underset{\mathbf{0}}{!}\varphi') =!\varphi$.
4. $r(\underset{\mathbf{k}}{!}\varphi')$ $(k > 0) =!\varphi \otimes \alpha^k \otimes r(\beta')$, *where $\alpha \otimes \beta'$ is a decomposition of φ'.*

We establish a few properties of the reduction r.

Lemma 2.17 *Let φ be any TBLL-formula, with instance φ'. Then*

$$\varphi \vdash r(\varphi')$$

proof: By structural induction. The atomic case is axiom. In the case that φ' is a tensor, the result is immediate from the induction hypothesis. Suppose φ' is $\underset{\mathbf{m}}{!}\psi'$. Then use the induction hypothesis on ψ' together and the derived rules of weakening and contraction on banged formulas (2.3) to obtain the proof. ∎

Lemma 2.18 *Let $\alpha \otimes \beta$ be a decomposition of a nontrivial TBLL-formula. Then*

$$\Gamma \vdash \alpha \otimes \beta \qquad \Rightarrow \qquad \Gamma \models_{s} \alpha.$$

proof: It is clear that $\alpha \otimes \beta \vdash \alpha$, by weakening on β which is banged (2.3). By the cut rule $\Gamma \vdash \alpha$. Now by the Soundness of the String Test we get the result. ▮

Lemma 2.19 *Suppose α is in TBLL and it is banged. If $\Gamma \vdash \alpha$ then Γ is banged.*

proof: The lemma is shown by an easy induction on proofs. The only rule to check is that of $\otimes - R$, as in all other rules if the context Γ is banged prior to the application of the rule then it remains so after the application.

$$\frac{\Gamma_0 \vdash \beta \qquad \Gamma_1 \vdash \delta}{\Gamma_0, \Gamma_1 \vdash \beta \otimes \delta}.$$

then Γ_0 and Γ_1 are banged by induction hypothesis, hence so is $\Gamma \equiv \Gamma_0, \Gamma_1$. ▮

Definition 2.20 *Let γ' be an instance of the TBLL formula γ. The* **!-closure** $\widehat{\gamma'}$ *of γ' is a banged formula (or the empty multiset) defined inductively as follows.*

- *If a is an atom $\widehat{a}$ is the empty multiset.*
- $\widehat{\underset{\mathbf{0}}{!}\theta'}$ *is* $!\theta$.
- $\widehat{\underset{\mathbf{m}}{!}\theta'}$ *is* $!\theta \otimes \widehat{\theta'} \qquad (m > 0)$.
- $\theta' \widehat{\otimes} \gamma'$ *is* $\widehat{\theta'} \otimes \widehat{\gamma'}$

Lemma 2.21 (TBLL-join lemma) *Let $\gamma'$and γ'' be instances of the same* **banged** *formula γ*

$$\widehat{\gamma' \bowtie \gamma''} \vdash \widehat{\gamma'} \otimes \widehat{\gamma''}$$

The proof of this is a straight forward induction on the structure of the formula, and is quite similar to the proof of TLL-join lemma.

Theorem 2.22 (bang-closure theorem) *Let $\alpha \otimes \beta$ be a decomposition of a TBLL-formula, and suppose*

$$\Lambda \vdash \alpha \otimes \beta.$$

Then there is an instance Λ' of Λ such that

$$T(\Lambda') \vdash \alpha$$

and

$$\widehat{\Lambda'} \vdash \beta.$$

Conversely, suppose α is in TLL, β is a banged formula, and $\Lambda \models_s \alpha$, inducing an instance Λ'. Then

$$\widehat{\Lambda'} \vdash \beta \Rightarrow \Lambda \vdash \alpha \otimes \beta$$

proof: ($\Rightarrow$) Suppose $\Lambda \vdash \alpha \otimes \beta$. By the preceding lemma $\Lambda \models_s \alpha$, i.e., we have solution to $p(\Lambda) = p(\alpha)$ which yields an instance Λ'. Now by lemma 2.10 we have that $T(\Lambda') \vdash \alpha$.

Now we show $\widehat{\Lambda'} \vdash \beta$. By the cut elimination theorem, there is a proof of $\Lambda \vdash \alpha \otimes \beta$ whose last right-introduction is $(\otimes - R)$ resulting in the formation of $\alpha \otimes \beta$, i.e., there is a proof of the form:

$$\cfrac{\cfrac{\cfrac{\Delta_0 \vdash \alpha \qquad \Delta_1 \vdash \beta}{\Delta_0, \Delta_1 \vdash \alpha \otimes \beta}}{\left.\vdots \qquad\right\} (*)}}{\Lambda \vdash \alpha \otimes \beta} \tag{12}$$

where the final steps are left introductions. Our proof that $\widehat{\Lambda'} \vdash \beta$ will be by induction on the length of the displayed proof of $\Lambda \vdash \alpha \otimes \beta$ starting from $\Delta_0, \Delta_1 \vdash \alpha \otimes \beta$, in particular on the length n of the "mid-section" (*). Our induction hypothesis is that if the proof above has length n then $\Lambda \models_s \alpha$ and for the instance Λ' induced by the string test, $\widehat{\Lambda'} \vdash \beta$.

Base case: If $n = 0$ then Λ is Δ_0, Δ_1 and we must show that $\widehat{\Delta_0'}, \widehat{\Delta_1'} \vdash \beta$. Since the bang-closure of a formula is banged, it suffices to notice that for banged formulas φ, $\widehat{\varphi'} \vdash \varphi$ (almost immediate from the definition). As Λ_1 is banged as β is, then we have $\widehat{\Delta_1'} \vdash \Delta_1$ and $\Delta_1 \vdash \beta$, which, by the cut rule gives $\widehat{\Delta_1'} \vdash \beta$, whence by the derived rule of weakening for banged formulas (see 2.3), we obtain $\widehat{\Delta_0'}, \widehat{\Delta_1'} \vdash \beta$.

Inductive case: suppose the induction hypothesis true for all shorter derivations than the one displayed above in (12). We consider all possible rules used in the last step.

<u>D:</u>: Suppose that the last step of (12) is a dereliction:

$$\cfrac{\cfrac{\vdots}{\Gamma, C \vdash \alpha \otimes \beta}}{\Gamma, !C \vdash \alpha \otimes \beta}.$$

By our induction hypothesis, $\Gamma, C \models_s \alpha$ and the instance Γ', C' induced by the string test satisfies $\widehat{\Gamma \otimes C'} \vdash \beta$. It is easy to see that $\Gamma, !C \models_s \alpha$ with the following instantiation

$$\Gamma', \tfrac{!}{1}C'$$

By decorating the outermost ! of $!C$ with a 1, we preserve precisely the same instance we had before. Now by the definition of !-closure

$$\widehat{\Gamma', \underset{1}{!}C'} = \widehat{\Gamma'}, \widehat{\underset{1}{!}C'} = \widehat{\Gamma'}, !C, \widehat{C'}.$$

But we have $\widehat{\Gamma'}, \widehat{C'} \vdash \beta$ by the induction hypothesis, hence, by weakening $\widehat{\Gamma'}, !C, \widehat{C'} \vdash \beta$.

<u>$\otimes$-L</u>: This is immediate: the string test and the definition of instance and !-closure remain unchanged when a subformula of a premiss C, D is rewritten as $C \otimes D$.

<u>W</u>: Suppose the last step of the proof (12) is weakening:

$$\frac{\dfrac{\vdots}{\Gamma \vdash \alpha \otimes \beta}}{\Gamma, !C \vdash \alpha \otimes \beta}.$$

We take the **zero instance** $\underset{0}{!}C$ for $!C$, and use the same instance of Γ. Then note that $\widehat{\underset{0}{!}C'} = !C$ By induction hypothesis $\widehat{\Gamma'} \vdash \beta$ and so $\widehat{\Gamma'}, !C \vdash \beta$ as well.

<u>C</u>: We are left with the only slightly delicate case, namely where the last step in the proof (12) is contraction:

$$\frac{\dfrac{\vdots}{\Gamma, !C, !C \vdash \alpha \otimes \beta}}{\Gamma, !C \vdash \alpha \otimes \beta}.$$

and the induction hypothesis that there are instances $\Gamma', !C', !C''$ induced by the string-test, for which $\widehat{\Gamma'}, \widehat{!C'}, \widehat{!C''} \vdash \beta$. Now $!C' \bowtie !C''$ provides us an instantiation such that the instantiation of the conclusion is the same as that of the premiss. Now use TBLL-join lemma along with the Induction hypothesis to get $\widehat{\Gamma'}, \widehat{!C' \bowtie !C''} \vdash \beta$.

($\Leftarrow$) Suppose that $\Lambda \models_s \alpha$. Let Λ' be the *instance of* Λ *induced by the string test* as defined in definition (2.5). Now by (lemma 2.17) we get that

$$\Lambda \vdash r(\Lambda')$$

Notice that $r(\Lambda')$ is (equivalent to) $\widehat{\Lambda'}, T(\Lambda')$. This gives us the result. ▮

We are now ready to describe a decision procedure for TBLL.

The Algorithm: "decide_TBLL"

Input: An ordered pair $\langle\Gamma, \theta\rangle$ where Γ is a multiset of TBLL formulas, and θ is a formula in TBLL.

begin

1. if $\theta \in$ TLL then execute string-test for $\langle\Gamma, \theta\rangle$, else
2. if θ is *banged* then check Γ is *banged* and

   case 1: θ is $!\varphi$. Then apply `decide_TBLL` to $\langle\Gamma, \varphi\rangle$

   case 2: θ is $\sigma \otimes \tau$, (where σ and τ are *banged*): then apply `decide_TBLL` to $\langle\Gamma, \sigma\rangle$ and $\langle\Gamma, \tau\rangle$.

3. else let $\alpha \otimes \varphi$ be a nontrivial decomposition of θ (α in TLL, φ banged). Then do:

   3.1 apply string-test to $\langle\Gamma, \alpha\rangle$. If it fails then **fail**. If it succeeds, let $\Gamma' :=$ an instance induced by the string test not already used up. If all are used up then **fail**.

   3.2 let $\widehat{\Gamma}'$ be the *bang-closure* of Γ'

   3.3 apply `decide_TBLL` to $\langle\widehat{\Gamma}', \varphi\rangle$. If it fails, then call this instance Γ' used up, and **goto 3.1**.

end

Remark on the algorithm Note that in step 3. we backtrack through all instantiations produced by the string test.

We briefly reconsider the example discussed following the string test algorithm, above.
We claimed that

$$\Gamma \not\vdash \theta \qquad \text{and} \qquad \theta \not\vdash \Gamma$$

where $\Gamma :=!(A \otimes A \otimes A)\otimes!(A \otimes A)$ and $\theta := A \otimes A\otimes!A$.

As our algorithm shows in two cycles, $\Gamma \vdash \theta$ is not a theorem of TBLL: $(A \otimes A)\otimes!A$ is a decomposition of θ, so we apply `decide_TBLL` to $\langle\Gamma, A \otimes A\rangle$ and $\langle\Gamma, !A\rangle$. The first input yields "yes" (by the string-test) but the second one yields the application of `decide_TBLL` to $\langle\Gamma, A\rangle$ which *fails* the string-test. The failure of $\theta \vdash \Gamma$ is immediate: θ is not banged.

2.1 Correctness and complexity of the algorithm

The correctness of the algorithm `decide_TBLL` is essentially the content of lemma 2.13 (which justifies step 1.), the decomposition lemma 2.4, lemma 2.4, lemma 2.19 on banged formulas, and the bang-closure theorem 2.22. The complexity is bounded by the complexity of the string test times number of calls to the string test on the pair $\langle \widehat{\Gamma}, \alpha \rangle$ where $\widehat{\Gamma}$ is the bang-closure of Γ. $\widehat{\Gamma}$ is bounded by the "full" bang-closure obtained by the taking the multiset of all banged subformulas of Γ whose length is bounded by $lth(\Gamma^2)$. The number of calls is bounded by the $2^{(\otimes\alpha)}$ where $(\otimes\alpha)$ = number of tensors in α, which is bounded by l. Letting $l = lth(\alpha) + lth(\Gamma), !!l = lth(\widehat{\Gamma}) \leq l^2$, This gives a bound of $(!!l)^{l^2} \cdot 2^l$ steps. We state this bound as a theorem and leave the details of formalizing this argument to the reader. We also recall that an NP lower bound for just the TBLL/TLL-fragment was established in lemma 2.14 above.

Theorem 2.23 *There is an $o(l^{2l^2})$ algorithm for deciding the derivability of sequents in TBLL. The problem is NP-hard.*

The authors would like to thank Andre Scedrov and Ramesh Subrahmanyam for comments and suggestions, and Dirk Roorda for pointing out errors in earlier drafts and for helpful suggestions.

References

[1] Garey, Michael R. and Johnson, Davis S.[1979] *Computers and intractability : a guide to the theory of NP-completeness* San Francisco : W. H. Freeman.

[2] Gehlot, V.[1991] *A proof-theoretic approach to semantics of concurrency*, Dissertation, University of Pennsylvania.

[3] Girard, Lafont, Taylor [1989] *Proofs and Types*, Cambridge University Press (Cambridge Tracts in Theoretical Computer Science 7), Cambridge.

[4] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[5] Gunter, C.A. and V. Gehlot [1989] "Nets as Tensor Theories" in *Proc. 10-th International Conference on Application and Theory of Petri Nets, Bonn*, G. De Michelis, ed.

[6] Horowitz, E. and Sahni S. [1978] *Fundamentals of Computer Algorithms*, Computer Science Press.

[7] Kanovich, M. [1992] "Horn Programming in Linear Logic is NP-Complete" in *Proc. 7-th Annual IEEE Symposium on Logic in Computer Science, Santa Cruz, California*, pp. 200-210, IEEE Computer Society Press, Los Alamitos, California.

[8] Kosaraju, S.R. [1982], "Decidability of Reachability in Vector Addition Systems", *Proc. 14-th ACM Symp. on Theory of Computing*, pp. 267-281.

[9] Lincoln, P. Mitchell, J., Scedrov, A. and Shankar, N. [1990] "Decision Problems for Propositional Linear Logic", proc. 31st IEEE symp. on Foundations of Computer Science.

[10] Stockmeyer, L. [1987] "Classifying the computational complexity of problems", *Journal of Symbolic Logic*, volume 52, pp. 1-43.

[11] Troelstra, A. S. [1991], *Lectures on Linear Logic* and *Lectures on Linear Logic: errata and Supplement*, lecture notes, Institute for Language, logic and information, Department of mathematics and Computer Science, University of Amsterdam. To appear as a book in the CSLI- Stanford series.