System BV is NP-complete

Ozan Kahramanoğulları [1,2]

Computer Science Institute, University of Leipzig
International Center for Computational Logic, TU Dresden

Abstract

System BV is an extension of multiplicative linear logic (MLL) with the rules *mix*, *nullary mix*, and a self-dual, non-commutative logical operator, called *seq*. While the rules *mix* and *nullary mix* extend the deductive system, the operator seq extends the language of MLL. Due to the operator seq, system BV extends the applications of MLL to those where sequential composition is crucial, e.g., concurrency theory. System FBV is an extension of MLL with the rules *mix* and *nullary mix*. In this paper, by relying on the fact that system BV is a conservative extension of system FBV, I show that system BV is NP-complete by encoding the 3-Partition problem in FBV. I provide a simple completeness proof of this encoding by resorting to a novel proof theoretical method for reducing the nondeterminism in proof search, which is also of independent interest.

1 Introduction

Since its emergence, the multiplicative fragment of linear logic [5] remained in focus of researchers due to its resource conscious features that capture properties of concurrent computation (see, e.g., [1]). Max Kanovich showed in [8,9] that multiplicative linear logic (MLL) is NP-complete. In [10], Lincoln and Winkler show that constant-only fragment of MLL is also NP-complete.

However, from the point of view of applications, multiplicative linear logic lacks a natural notion of sequentiality, which is crucial for expressing many computational phenomena, e.g., sequential composition of processes in concurrency theory. In [6], Guglielmi introduced a system, called BV, which is an extension of MLL with the rules mix, nullary mix (mix0), and a self-dual, non-commutative logical operator, called *seq*. While the rules mix and mix0 extend

[1] This work has been supported by the DFG Graduierten-kolleg 446, and accomplished during my stay at the International Center for Computational Logic as a visiting researcher. I would like to thank Alessio Guglielmi, Lutz Straßburger, and Alwen Tiu for valuable remarks and improvements.

[2] Email: ozan@informatik.uni-leipzig.de

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

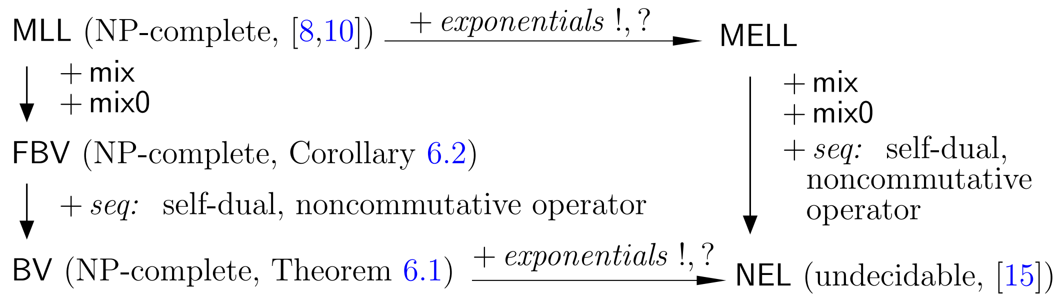

Fig. 1. The relationship between MLL, FBV, BV, MELL and NEL

the deductive system, the operator seq extends the language of MLL. This logic captures sequential and parallel composition of process algebra naturally by means of logical operators. In particular, Bruscoli showed, in [3], that there is a strict correspondence between a fragment of the process algebra CCS [11] and system BV.

System BV can not be designed in any standard sequent calculus, as it was shown by Tiu in [16]: in the sequent calculus, during bottom-up proof search, inference rules are applied at the main connective; however, in order to get all the provable formula of system BV by means of a deductive system, a notion of deep rewriting is necessary. System BV is designed in the proof theoretical framework, the calculus of structures [6], which allows for such deep rewriting. In the calculus of structures, the notion of main connective disappears and the notions of formula and sequent of the sequent calculus are replaced with the notion of structure. The inference rules can be applied deep inside structures, resulting in one of the distinguishing features of this formalism, that is, *deep inference*. In several other related work (see, e.g., [2,14]), deep inference gives rise to many interesting proof theoretical properties of other logics, that are not observable within the sequent calculus presentation of these logics.

Extending multiplicative linear logic with a self-dual, noncommutative operator was also considered in Retoré's pomset logic [12]. In [13], Retoré gives proof nets for the pomset logic, but so far there is no sequent calculus system for pomset logic with the cut-elimination property. In fact, Guglielmi conjectured, in [6], that pomset logic and system BV are equivalent.

In [7], Guglielmi and Straßburger introduced a system, called NEL, which extends system BV with the exponentials of linear logic. In other words, system NEL is an extension of multiplicative exponential linear logic (MELL) with the rules mix, mix0, and the self-dual, noncommutative logical operator seq. Although it is unknown if multiplicative exponential linear logic is decidable or not, in [15], Straßburger showed that system NEL is undecidable. However, the complexity of the decision problem in system BV remained an open problem.

In this paper, by encoding the 3-Partition problem [4] in multiplicative linear logic extended by the rules mix and mix0, i.e., system FBV, I show the NP-hardness of this logic. This result implies the NP-hardness of system BV, since system BV is a conservative extension of system FBV (MLL+mix+mix0):

every provable BV structure, which does not contain any seq structure, is also provable in FBV. Figure 1 summarizes the relationship between MLL, FBV, BV, MELL and NEL, and the contribution of this paper.

Although the encoding in the sequent calculus, which was used in [8] for showing the NP-hardness of MLL, can be used to show the NP-hardness of MLL + mix + mix0, in this paper, I provide a simpler encoding and an easier proof, within the calculus of structures, by means of an analysis of the proof theory of this logic: in contrast to sequent calculus, while applying the inference rules in bottom-up proof search, deep applicability of the inference rules in the calculus of structures introduces a greater nondeterminism. I introduce a novel technique for controlling the nondeterminism in proof search, which is also of independent interest from the point of view of applications: despite the combinatoric explosion in the applicability of the inference rules in the calculus of structures, my method reduces the nondeterminism in proof search without damaging the completeness of the system. This way, it becomes possible to separate the redundant nondeterminism, in my encoding, from the concise nondeterminism, and prove the completeness of the encoding without going into incomprehensible and complicated case analysis.

The rest of the paper is organized as follows: after introducing the calculus of structures and system BV in the next section, I present a method for controlling the nondeterminism in proof search in multiplicative linear logic extended by the rules mix and mix0, i.e., system FBV. I then present an encoding of the 3-partition problem in FBV, which is an NP-complete problem. Following this, by showing that the length of a proof in BV is bounded by a polynomial in the size of the structure being proved, I show that system BV is NP-complete.

2 The Calculus of Structures and System BV

This section re-collects some notions and definitions of the calculus of structures and system BV, following [6].

In the language of BV atoms are denoted by $a, b, c, \ldots$ Structures are denoted by $R, S, T, \ldots$ and generated by

$$S ::= \circ \mid a \mid \langle \underbrace{S; \ldots ; S}_{>0} \rangle \mid [\underbrace{S, \ldots , S}_{>0}] \mid (\underbrace{S, \ldots , S}_{>0}) \mid \overline{S} \quad ,$$

where $\circ$, the *unit*, is not an atom. $\langle S; \ldots ; S\rangle$ is called a *seq structure*, $[S, \ldots , S]$ is called a *par structure*, and $(S, \ldots , S)$ is called a *copar structure*, $\overline{S}$ is the *negation* of the structure S. Structures are considered equivalent modulo the relation $\approx$, which is the smallest congruence relation induced by the equalities shown in Figure 2. There $\vec{R}$, $\vec{T}$ and $\vec{U}$ stand for finite, non-empty sequence of structures. A *structure context*, denoted as in $S\{\ \}$, is a structure with a hole that does not appear in the scope of negation. The structure R is a *substructure* of $S\{R\}$ and $S\{\ \}$ is its *context*. Context braces are omitted if no

Associativity

$$\langle \vec{R}; \langle \vec{T} \rangle; \vec{U} \rangle \approx \langle \vec{R}; \vec{T}; \vec{U} \rangle$$
$$[\vec{R}, [\vec{T}]] \approx [\vec{R}, \vec{T}]$$
$$(\vec{R}, (\vec{T})) \approx (\vec{R}, \vec{T})$$

Context Closure

$$\text{if } R = T \text{ then } \quad S\{R\} = S\{T\} \quad \text{and } \bar{R} = \bar{T}$$

Commutativity

$$[\vec{R}, \vec{T}] \approx [\vec{T}, \vec{R}]$$
$$(\vec{R}, \vec{T}) \approx (\vec{T}, \vec{R})$$

Units

$$\langle \circ; \vec{R} \rangle \approx \langle \vec{R}; \circ \rangle \approx \langle \vec{R} \rangle$$
$$[\circ, \vec{R}] \approx [\vec{R}]$$
$$(\circ, \vec{R}) \approx (\vec{R})$$

Negation

$$\overline{\circ} \approx \circ$$
$$\overline{\langle R; T \rangle} \approx \langle \overline{R}; \overline{T} \rangle$$
$$\overline{[R, T]} \approx (\overline{R}, \overline{T})$$
$$\overline{(R, T)} \approx [\overline{R}, \overline{T}]$$
$$\overline{\overline{R}} \approx R$$

Singleton

$$\langle R \rangle \approx [R] \approx (R) \approx R$$

Fig. 2. Equivalence relations underlying BV.

ambiguity is possible: for instance $S[R,T]$ stands for $S\{[R,T]\}$. A structure, or a structure context, is in *normal form* when the only negated structures appearing in it are atoms, no unit $\circ$ appears in it.

We will call the BV structures, which do not involve seq structures, FBV structures. There is a straightforward correspondence between FBV structures and formulae of multiplicative linear logic (MLL), which do not contain the units 1 and $\bot$. For example $[(a,b), \bar{c}, \bar{d}]$ corresponds to $(\,(a \otimes b) \parr c^{\bot} \parr d^{\bot})$, and vice versa. Units 1 and $\bot$ are mapped into $\circ$, since $1 \equiv \bot$, when the rules mix and mix0 are added to MLL.

$$\mathsf{mix}\,\frac{\vdash \Phi \quad \vdash \Psi}{\vdash \Phi, \Psi} \qquad\qquad \mathsf{mix0}\,\frac{}{\vdash}$$

For a more detailed discussion on the proof theory of BV and the precise relation between BV and MLL, the reader is referred to [6].

In the calculus of structures, an *inference rule* is a scheme of the kind $\rho \dfrac{T}{R}$, where ρ is the *name* of the rule, T is its *premise* and R is its *conclusion*. A typical (deep) inference rule has the shape $\rho \dfrac{S\{T\}}{S\{R\}}$ and specifies the implication $T \Rightarrow R$ inside a generic context $S\{\quad\}$, which is the implication being modeled in the system [3]. When premise and conclusion in an instance of an inference rule are equivalent, that instance is *trivial*, otherwise it is *non-trivial*. An inference rule is called an *axiom* if its premise is empty. Rules with empty contexts correspond to the case of the sequent calculus.

[3] Due to duality between $T \Rightarrow R$ and $\bar{R} \Rightarrow \bar{T}$, rules come in pairs of dual rules: a down-version and an up-version. For instance, the dual of the $\mathsf{ai}\downarrow$ rule in Figure 3 is the cut rule. In this paper, only the down rules, which provide a sound and complete system are considered.

$$\circ\!\downarrow \frac{}{\circ} \qquad \mathsf{ai}\!\downarrow \frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{s}\,\frac{S([R,T],U)}{S[(R,U),T]} \qquad \mathsf{q}\!\downarrow \frac{S\langle [R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

Fig. 3. System BV

A (formal) *system* $\mathscr{S}$ is a set of inference rules. A derivation Δ in a certain formal system is a finite chain of instances of inference rules in the system. A derivation can consist of just one structure. The topmost structure in a derivation, if present, is called the *premise* of the derivation, and the bottommost structure is called its *conclusion*. A derivation Δ whose premise is T, conclusion is R, and inference rules are in $\mathscr{S}$ will be written as $\begin{array}{c} T \\ \Delta \| \mathscr{S} \\ R \end{array}$.

Similarly, $\begin{array}{c} \Pi \| \mathscr{S} \\ R \end{array}$ will denote a *proof* Π, which is a finite derivation, whose topmost inference rule is an axiom. The *length* of a derivation (proof) is the number of instances of inference rules appearing in it.

Two systems $\mathscr{S}$ and $\mathscr{S}'$ are *equivalent* if for every proof of a structure T in system $\mathscr{S}$, there exists a proof of T in system $\mathscr{S}'$, and vice versa.

The system $\{\circ\!\downarrow, \mathsf{ai}\!\downarrow, \mathsf{s}, \mathsf{q}\!\downarrow\}$, shown in Figure 3, is denoted by BV, and called *basic system* V. The rules of the system are called *unit* ($\circ\!\downarrow$), *atomic interaction* ($\mathsf{ai}\!\downarrow$), *switch* (s) and *seq* ($\mathsf{q}\!\downarrow$). The multiplicative linear logic system extended by mix and $\mathsf{mix0}$, or system $\{\circ\!\downarrow, \mathsf{ai}\!\downarrow, \mathsf{s}\}$, is denoted by FBV.

3 Preliminaries

In a proof search episode, the inference rules can be applied to a structure, nondeterministically, in many different ways, but only few of these rule instances can provide a proof. While providing a rich combinatorial analysis of the logic being studied, applicability of the inference rules at any depth causes an even greater nondeterminism. However, the mutual dependencies between atoms, which are easily observable due to the notion of structure, provides ways of controlling the nondeterminism without breaking the proof theoretical properties. In this section, I present a system equivalent to system FBV, where the nondeterminism in proof search is reduced by taking these mutual dependencies between dual atoms into consideration.

Definition 3.1 Given a structure S, the notation $\mathsf{at}\,S$ indicates the set of all the atoms appearing in S. Let *interaction switch* be the rule

$$\mathsf{is}\,\frac{S([R,T],U)}{S[(R,U),T]} \quad ,$$

where $\mathsf{at}\,\overline{T} \cap \mathsf{at}\,R \neq \emptyset$. Let *system* FBV *with interaction switch*, or system

FBVi be the system $\{\circ\downarrow, \mathsf{ai}\downarrow, \mathsf{is}\}$.

Lemma 3.2 *For any* FBV *structures* R, U *and* T, *if* $[R, U]$ *has a proof in* FBVi, *then there is a derivation* $\begin{array}{c} T \\ \Delta \| \mathsf{FBVi} \\ [(R,T),U] \end{array}$.

Proof. If $\mathsf{at}\,\overline{R} \;\cap\; \mathsf{at}\,U \neq \emptyset$, then trivial. Otherwise R and U must have separate proofs. □

Theorem 3.3 *(Shallow Splitting for* FBVi*) For all structures* R, T *and* P, *if* $[(R,T),P]$ *is provable in* FBVi *then there exists* P_1, P_2 *and* $\begin{array}{c} [P_1, P_2] \\ \Delta \| \mathsf{FBVi} \\ P \end{array}$ *such that* $[R, P_1]$ *and* $[T, P_2]$ *are provable in* FBVi.

Proof. (Sketch) Proof by induction with Lemma 3.2. Take the induction measure (m, n) where $m = |\,[(R,T),P]\,|$ and n is the length of the proof Π of $[(R,T),P]$. Single out the bottom most rule application ρ in Π. Apply the induction argument to ρ, similar to the proof of the splitting theorem for BV in [6]. The following are the non-trivial cases for ρ:

- $\rho = \mathsf{is}$ such that $R = (R', R'')$, $T = (T', T'')$, $P = (P', P'')$ and

$$\mathsf{is}\,\frac{[([(R',T'),P'],R'',T''),P'']}{[(R',R'',T',T''),P',P'']} \quad .$$

- $\rho = \mathsf{is}$ such that $P = [(P', P''), U', U'']$ and

$$\mathsf{is}\,\frac{[([(R,T),P',U'],P''),U'']}{[(R,T),(P',P''),U',U'']} \quad .$$

□

Theorem 3.4 (Context Reduction for FBVi) *For all structures* R *and for all contexts* $S\{\ \}$ *such that* $S\{R\}$ *is provable in* FBVi, *there exists a structure* U *such that for all structures* X *there exist derivations:*

$$\begin{array}{c} [X, U] \\ \| \mathsf{FBVi} \\ S\{X\} \end{array} \qquad \textit{and} \qquad \begin{array}{c} \| \mathsf{FBVi} \\ [R, U] \end{array} \quad .$$

Proof. (Sketch) By induction on the size of $S\{\circ\}$ with Theorem 3.3 and Lemma 3.2, similar to the proof of context reduction for BV in [6]. □

Theorem 3.5 *System* FBV *and* FBVi *are equivalent.*

Proof. Observe that every proof in FBVi is also a proof in FBV. For the other direction, single out the upper-most instance of the switch rule in the FBV proof, which is not an instance of the interaction switch rule:

$$\mathsf{s}\,\frac{\overset{\|\mathsf{FBVi}}{S([R,U],T)}}{S[(R,T),U]}$$

From Theorem 3.4, we have

$$\begin{array}{c}[\{\ \},V]\\ \| \mathsf{FBVi}\\ S\{\ \}\end{array} \quad \text{such that} \quad \begin{array}{c}\|\mathsf{FBVi}\\ [([R,U],T),V]\end{array} .$$

Then, from Theorem 3.3, we have

$$\begin{array}{c}[K_1,K_2]\\ \| \mathsf{FBVi}\\ V\end{array} \quad ; \quad \begin{array}{c}\|\mathsf{FBVi}\\ [R,U,K_1]\end{array} \quad ; \quad \text{and} \quad \begin{array}{c}\Pi\|\mathsf{FBVi}\\ [K_2,T]\end{array} .$$

We can then construct the following proof

$$\begin{array}{c}\|\mathsf{FBVi}\\ [R,U,K_1]\\ \Delta\|\mathsf{FBVi}\\ [(R,T),U,K_1,K_2]\\ \|\mathsf{FBVi}\\ [(R,T),U,V]\\ \|\mathsf{FBVi}\\ S[(R,T),U]\end{array}$$

where Δ is the derivation delivered by Lemma 3.2 with the proof Π. Repeat the above procedure inductively till all the instances of the switch rule, which are not instances of interaction switch rule, are removed. □

Splitting technique was originally introduced, in [6], to prove cut elimination for system BV. Because of the splitting technique used in the completeness argument above, from which cut elimination immediately follows, system FBVi remains clean from a proof theoretic point of view.

Proposition 3.6 *System* BV *is a conservative extension of system* FBV*, that is, if a structure* R*, not containing any seq structures, is provable in* BV*, then it is also provable in* FBV.

Proof. Let R be a BV structure that does not contain any seq structures. By induction on the length of the proof Π of R in BV, construct the proof Π' of R in FBV. Since the only rule that involves seq structures is the rule $\mathsf{q}\!\downarrow$, it must be $\Pi = \Pi'$. □

4 BV is NP-hard

In this section, I present an encoding of the 3-Partition Problem in system FBV to show the NP-hardness of this logic, and system BV. This problem was also used by Lincoln and Winkler, in [10], to show the NP-hardness of the constant only fragment of MLL. By providing a similar encoding, and resorting to the proof theory of system FBV, in the lines of Section 3, I provide a very simple correctness proof without going into a complicated case analysis.

Problem 4.1 *[4] (3-Partition) Given a set of $A = \{a_1, a_2, \ldots, a_{3m}\}$ of elements, a bound $B \in \mathbb{Z}^+$, and a size $S(a) \in \mathbb{Z}^+$ for each $a \in A$ such that $\frac{1}{4}B < S(a) < \frac{1}{2}B$ and $\sum_{a \in A} S(a) = Bm$, does there exist a partition of A into m disjoint subsets A_i so that $\sum_{a \in A_i} S(a) = B$ for each A_i in the partition.*

The constraints on the $S(a)$ imply that such a partition must have exactly three elements in each of its sets. This problem 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 my encoding, since I represent the input problem by using atoms.

4.1 Encoding the 3-Partition Problem in FBV

Given an instance of 3-Partition equipped with a set $A = \{a_1, a_2, \ldots, a_{3m}\}$, a unary function S, and a natural number B, presented as a tuple $\langle A, m, B, S \rangle$, the encoding function θ is defined as $\theta(\,\langle A, m, B, S \rangle\,) =$

$$[(k, [\underbrace{c, \ldots, c}_{\times S(a_1)}]), \ldots, (k, [\underbrace{c, \ldots, c}_{\times S(a_{3m})}]), (\underbrace{[\bar{k}, \bar{k}, \bar{k}, (\underbrace{\bar{c}, \ldots, \bar{c}}_{\times B})], \ldots, [\bar{k}, \bar{k}, \bar{k}, (\underbrace{\bar{c}, \ldots, \bar{c}}_{\times B})]}_{\times m})]$$

Lemma 4.2 *Let $S(a_1)$, $S(a_2)$ and $S(a_3)$ be natural numbers such that, for some natural number B, it holds that $\frac{1}{4}B < S(a_1), S(a_2), S(a_3) < \frac{1}{2}B$. If $S(a_1) + S(a_2) + S(a_3) = B$, then*

$$\begin{array}{c}[R, Q] \\ \Delta \Big\Vert \mathsf{FBV} \\ [R, (k, [\underbrace{c, \ldots, c}_{\times S(a_1)}]), (k, [\underbrace{c, \ldots, c}_{\times S(a_2)}]), (k, [\underbrace{c, \ldots, c}_{\times S(a_3)}]), (Q, [\bar{k}, \bar{k}, \bar{k}, (\underbrace{\bar{c}, \ldots, \bar{c}}_{\times B})])]\end{array} .$$

Proof. Take the following derivation where the redex in the conclusion of the applied rule is highlighted.

$$
\begin{array}{l}
\mathsf{i}\downarrow \dfrac{[R,Q]}{[R,(Q,[\underbrace{c,\ldots,c}_{\times S(a_1)},\underbrace{c,\ldots,c}_{\times S(a_2)},\underbrace{c,\ldots,c}_{\times S(a_3)},(\underbrace{\bar{c},\ldots,\bar{c}}_{\times B})])]} \\
\mathsf{ai}\downarrow \dfrac{}{\vdots} \\
\mathsf{s}\dfrac{}{[R,(k,[c,\ldots,c]),(k,[c,\ldots,c]),(Q,[\bar{k},\bar{k},c,\ldots,c,(\bar{c},\ldots,\bar{c})])]} \\
\mathsf{ai}\downarrow \dfrac{}{[R,(k,[c,\ldots,c]),(k,[c,\ldots,c]),(Q,[([k,\bar{k}],[c,\ldots,c]),\bar{k},\bar{k},(\bar{c},\ldots,\bar{c})])]} \\
\mathsf{s}\dfrac{}{[R,(k,[c,\ldots,c]),(k,[c,\ldots,c]),(Q,[(k,[c,\ldots,c]),\bar{k},\bar{k},\bar{k},(\bar{c},\ldots,\bar{c})])]} \\
\mathsf{s}\dfrac{}{[R,(k,[\underbrace{c,\ldots,c}_{\times S(a_1)}]),(k,[\underbrace{c,\ldots,c}_{\times S(a_2)}]),(k,[\underbrace{c,\ldots,c}_{\times S(a_3)}]),(Q,[\bar{k},\bar{k},\bar{k},(\underbrace{\bar{c},\ldots,\bar{c}}_{\times B})])]}
\end{array}
$$

□

Theorem 4.3 *If a 3-Partition problem $\langle A,m,B,S\rangle$ is solvable, then there is a proof of $\theta(\langle A,m,B,S\rangle)$ in* FBV.

Proof. By induction on m, using Lemma 4.2. □

4.2 Completeness of the Encoding

Theorem 4.4 *For A, m, B, and S satisfying the constraints of 3-Partition, if there is a proof of $\theta(\langle A,m,B,S\rangle)$ in* FBV*, then the 3-Partition problem $\langle A,m,B,S\rangle$ is solvable.*

Proof. By induction on m: the case for $m=0$ corresponds to empty problem. Let $\langle A,m+1,B,S\rangle$ be such that $A=\{a_1,a_2,\ldots,a_{3m},a_{3m+1},a_{3m+2},a_{3m+3}\}$. Assuming that we have a proof of $\theta(\langle A,m+1,B,S\rangle)$, we show that $\langle A,m+1,B,S\rangle$ is solvable. Let

$$R=[(k,[\underbrace{c,\ldots,c}_{\times S(a_1)}]),(k,[\underbrace{c,\ldots,c}_{\times S(a_2)}]),\ldots,(k,[\underbrace{c,\ldots,c}_{\times S(a_{3m+2})}]),(k,[\underbrace{c,\ldots,c}_{\times S(a_{3m+3})}])]$$

$$\text{and}\qquad Q=\underbrace{([\bar{k},\bar{k},\bar{k},(\underbrace{\bar{c},\ldots,\bar{c}}_{\times B})],\ldots,[\bar{k},\bar{k},\bar{k},(\underbrace{\bar{c},\ldots,\bar{c}}_{\times B})])}_{\times m}$$

such that

$$\theta(\langle A,m+1,B,S\rangle)=[R,(Q,[\bar{k},\bar{k},\bar{k},(\underbrace{\bar{c},\ldots,\bar{c}}_{\times B})])] \quad .$$

From Theorem 3.5 we have that $\theta(\langle A,m+1,B,S\rangle)$ has a proof in FBV iff it has a proof in FBVi. It follows from Theorem 3.3 that

$$\begin{array}{c}[K_1, K_2]\\ \Delta \| \mathsf{FBVi}\\ R\end{array} \quad \text{such that} \quad \begin{array}{c}\Pi \| \mathsf{FBVi}\\ [K_1, Q]\end{array} \quad \text{and} \quad \begin{array}{c}\| \mathsf{FBVi}\\ [K_2, \bar{k}, \bar{k}, \bar{k}, (\bar{c}, \ldots, \bar{c})]\end{array} \quad .$$

Since there are only positive atoms in R, it follows that none of the rules $\mathsf{ai}\downarrow$ and is can be applied in Δ, hence the derivation Δ must be the structure R. This implies that $[K_1, K_2]$ are two partitions of R. Observe that in K_2 there must be exactly 3 occurrences of k, which implies that

$$K_2 = [(k, [\underbrace{c, \ldots, c}_{\times S(a_i)}]), (k, [\underbrace{c, \ldots, c}_{\times S(a_j)}]), (k, [\underbrace{c, \ldots, c}_{\times S(a_k)}])]$$

and $S(a_i) + S(a_j) + S(a_k) = B$, and Π is the proof delivered by the induction hypothesis. □

Corollary 4.5 *System* FBV *is NP-hard.*

Proof. Follows immediately from Theorem 4.3 and Theorem 4.4. □

Since system BV is a conservative extension of system FBV, this result implies the NP-hardness of system BV.

Corollary 4.6 *System* BV *is NP-hard.*

Proof. Follows immediately from Proposition 3.6 and Corollary 4.5. □

5 System BV is in NP

In this section, I show that the proof of a BV structure is bounded by a polynomial in the size of this structure.

Definition 5.1 [6] Given a structure S, we talk about *atom occurrences* when considering all the atoms appearing in S as distinct (for example, by indexing them so that two atoms, which are equal, get different indices). The notation $\mathsf{occ}\, S$ indicates the set of all the atom occurrences appearing in S. The *size* of S is the cardinality of the set $\mathsf{occ}\, S$. Given a structure S in normal form, the *structural relation* $\downarrow \subset (\mathsf{occ}\, S)^2$ and, for every $S'\{\ \}$, U and V and for every a in U and b in V, the following holds: if $S = S'[U, V]$ then $a \downarrow_S b$. To a structure that is not in normal form we associate the structural relation obtained from any of its normal forms, since they yield the same relation $\downarrow_S$.

Remark 5.2 Let $R = S[a, \bar{a}]$ and $R' = S\{\circ\}$ be BV structures with pairwise distinct atoms. If $\mathsf{ai}\downarrow \dfrac{R'}{R}$, then $\downarrow_{R'} = \downarrow_R \setminus \{ (a, \bar{a}), (\bar{a}, a) \}$.

Remark 5.3 Let $R = S[(P,T),U]$ and $R' = S([P,U],T)$ be BV structures with pairwise distinct atoms. If $\mathsf{s}\,\dfrac{R'}{R}$, then

$$\downarrow_{R'} = \downarrow_R \setminus (\{(x,y) \mid x \in \mathsf{occ}\,T \wedge y \in \mathsf{occ}\,U\} \cup \{(x,y) \mid x \in \mathsf{occ}\,U \wedge y \in \mathsf{occ}\,T\}).$$

Remark 5.4 Let $R = S[\langle P;T\rangle,\langle U;V\rangle]$ and $R' = S\langle [P,U];[T,V]\rangle$ be BV structures with pairwise distinct atoms. If $\mathsf{q}{\downarrow}\,\dfrac{R'}{R}$, then

$$\downarrow_{R'} = \downarrow_R \setminus (\{(x,y) \mid x \in \mathsf{occ}\,P \wedge y \in \mathsf{occ}\,V\} \cup \{(x,y) \mid x \in \mathsf{occ}\,V \wedge y \in \mathsf{occ}\,P\} \cup \{(x,y) \mid x \in \mathsf{occ}\,U \wedge y \in \mathsf{occ}\,T\} \cup \{(x,y) \mid x \in \mathsf{occ}\,T \wedge y \in \mathsf{occ}\,U\}).$$

Proposition 5.5 *The length of a proof of a* BV *structure* R *is bounded by* $\mathcal{O}(|\mathsf{occ}\,R|^2)$.

Proof. With Remark 5.2, 5.3, and 5.4; observe that $\downarrow_R \subset (\mathsf{occ}R)^2$, hence $|\downarrow_R| < |\mathsf{occ}R|^2$. For each (non-trivial) application of an inference rule such that $\rho\,\dfrac{R'}{R}$, we have that $|\downarrow_{R'}| < |\downarrow_R|$. □

6 Main Result

The main result of the paper follows from the results in Sections 4 and 5:

Theorem 6.1 *System* BV *is NP-complete.*

Proof. Follows immediately from Corollary 4.6 and Proposition 5.5. □

Corollary 6.2 *Multiplicative linear logic extended by the rules* mix *and* mix0*, or System* FBV*, is NP-complete.*

Proof. Follows immediately from Corollary 4.5 and Proposition 5.5. □

References

[1] G. Bellin. Subnets of proof-nets in multiplicative linear logic with MIX. In *Mathematical Structures in Computer Science*, volume 7, pages 663–699. Cambridge University Press, 1997.

[2] Kai Brünnler. *Deep Inference and Symmetry in Classical Proofs.* PhD thesis, Technische Universität Dresden, 2003.

[3] Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference,*

volume 2401 of *Lecture Notes in Computer Science*, pages 302–316. Springer-Verlag, 2002.

[4] M. R. Garey and D. S. Johnson. *Computer and Intractability: A Guide to Theory of NP-Completeness.* Freeman, San Francisco, 1979.

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

[6] Alessio Guglielmi. A system of interaction and structure. Technical Report WV-02-10, TU Dresden, 2002. To app. in ACM Transactions on Computational Logic.

[7] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *Lecture Notes in Artificial Intelligence*, pages 231–246. Springer-Verlag, 2002.

[8] Max Kanovich. The multiplicative fragment of linear logic is NP-complete. Technical Report X-91-13, Institute for Language, Logic, and Information, 1991.

[9] Max Kanovich. Horn programming in linear logic is NP-complete. In *Proccedings of the 7th Annual IEEE Symposium on Logic in Computer Science, Santa Cruz*, pages 200–210, Los Alamitos, California, 1992. IEEE Computer Society Press.

[10] Patrick Lincoln and Timothy C. Winkler. Constant-only multiplicative linear logic is NP-complete. In *Theoretical Computer Science*, volume 135(1), pages 155–169. 1994.

[11] Robin Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.

[12] Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Ph. de Groote and J. R. Hindley, editors, *Typed Lambda Calculus and Applications, TLCA'97*, volume 1210 of *Lecture Notes in Computer Science*, pages 300–318. Springer-Verlag, 1997.

[13] Christian Retoré. Pomset logic as a calculus of directed cographs. In V. M. Abrusci and C. Casadio, editors, *Dynamic Perspectives in Logic and Linguistics*, pages 221–247. Bulzoni, Rome, 1999. Also available as INRIA Rapport de Recherche RR-3714.

[14] Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures.* PhD thesis, TU Dresden, 2003.

[15] Lutz Straßburger. System NEL is undecidable. In Ruy De Queiroz, Elaine Pimentel, and Lucília Figueiredo, editors, *10th Workshop on Logic, Language, Information and Computation (WoLLIC)*, volume 84 of *Electronic Notes in Theoretical Computer Science*, 2003.

[16] Alwen Fernanto Tiu. Properties of a logical system in the calculus of structures. Technical Report WV-01-06, Technische Universität Dresden, 2001.