REPORTS ON MATHEMATICAL LOGIC
43 (2008), 5–24

Wojciech BUSZKOWSKI

ON THE COMPLEXITY OF SOME SUBSTRUCTURAL LOGICS

A b s t r a c t. We use a syntactic interpretation of MALL in BCI with $\wedge$, defined in [5], to prove the undecidability of the consequence relations for BCI with $\wedge$ and BCI with $\vee$, and the NP-completeness of BCI. Similar results are obtained for a variant of the Lambek calculus.

1. Introduction

Propositional Linear Logic (PLL) is undecidable [13]. In this paper we use the proof of the undecidability of PLL from Kanovich [10] to give a proof of the undecidability of the consequence relation for MALL (Multiplicative-Additive Linear Logic, i.e. PLL without exponentials). Then, we employ a syntactic interpretation of MALL in BCI with $\wedge$, stemming from [5], to infer the undecidability of the consequence relation for BCI with $\wedge$ and BCI with $\vee$. For BCI an analogous problem remains open, like the problem of the decidability of MELL [12]. Further, from the NP-completeness of MLL

Received 1 March 2007

(MALL without additives) [9, 14], we infer the NP-completeness of BCI and its single-variable fragment. At the end, we announce similar results for the $(\to)-$fragment of the Lambek calculus L* with the cyclic rule for the designated constant 0.

Interestingly, non-associative variants of these systems are essentially less complex: the consequence relation for the non-associative Lambek calculus (both noncommutative and commutative) is P-TIME [6].

PLL is a propositional logic in the language $(\otimes, \oplus, ^{\perp}, 0, 1, \wedge, \vee, !, ?)$, where $\otimes, \oplus, \wedge, \vee$ are binary connectives, $^{\perp}$ is a unary connective, $0, 1$ are constants, and $!, ?$ are unary modalities, called *exponentials*. We do not use the original notation of Girard [7], but a notation more popular in the community of substructural logics. $\otimes$, $\oplus$ and $^{\perp}$ are *multiplicatives*, called *product*, *par* and *linear negation*, respectively. 1 and 0 are multiplicative constants (interpreted as neutral elements for $\otimes$ and $\oplus$). $\wedge$ and $\vee$ are *additives*, called *meet* and *join*, respectively (or: additive conjunction and disjunction). ! and ? are called *ofcourse* and *whynot*, respectively. We avoid additive constants $\perp$ and $\top$.

We present a Gentzen-style sequent system for PLL (in a two-sided form). Sequents are expressions $\Gamma \Rightarrow \Delta$ such that Γ, Δ are (finite) multisets of formulas. A, B, C denote formulas, and Greek capitals denote multisets of formulas.

$$\text{(Id)}\ A \Rightarrow A$$

$$\text{(0L)}\ 0 \Rightarrow \quad \text{(0R)}\ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, 0}$$

$$\text{(1L)}\ \Rightarrow 1\ \text{(1R)}\ \frac{\Gamma \Rightarrow \Delta}{\Gamma, 1 \Rightarrow \Delta}$$

$$(^{\perp}\text{L})\ \frac{\Gamma \Rightarrow \Delta, A}{\Gamma, A^{\perp} \Rightarrow \Delta}\ (^{\perp}\text{R})\ \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A^{\perp}}$$

$$(\otimes\text{L})\ \frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \otimes B \Rightarrow \Delta}\ (\otimes\text{R})\ \frac{\Gamma_1 \Rightarrow \Delta_1, A;\ \Gamma_2 \Rightarrow \Delta_2, B}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2, A \otimes B}$$

$$(\oplus\text{L})\ \frac{\Gamma_1, A \Rightarrow \Delta_1;\ \Gamma_2, B \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2, A \oplus B \Rightarrow \Delta_1, \Delta_2}\ (\oplus\text{R }) \ \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \oplus B}$$

$$(\wedge\text{L})\ \frac{\Gamma, A_i \Rightarrow \Delta}{\Gamma, A_1 \wedge A_2 \Rightarrow \Delta}\ (\wedge\text{R})\ \frac{\Gamma \Rightarrow \Delta, A;\ \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}$$

$$(\vee\text{L})\ \frac{\Gamma, A \Rightarrow \Delta;\ \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta}\ (\vee\text{R})\ \frac{\Gamma \Rightarrow \Delta, A_i}{\Gamma \Rightarrow \Delta, A_1 \vee A_2}$$

$$(!\mathrm{L})\ \frac{\Gamma, A \Rightarrow \Delta}{\Gamma, !A \Rightarrow \Delta}\ (!\mathrm{R})\ \frac{!\Gamma \Rightarrow ?\Delta, A}{!\Gamma \Rightarrow ?\Delta, !A}$$

$$(?\mathrm{L})\ \frac{!\Gamma, A \Rightarrow ?\Delta}{!\Gamma, ?A \Rightarrow ?\Delta}\ (?\mathrm{R})\ \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, ?A}$$

By $!\Gamma$ (resp. $?\Delta$) one denotes a multiset whose all elements are of the form $!B$ (resp. $?B$). PLL also admits structural rules for ! and ?: Weakening (WEA) and Contraction (CON).

$$(!\mathrm{WEA})\ \frac{\Gamma \Rightarrow \Delta}{\Gamma, !A \Rightarrow \Delta}\ (!\mathrm{CON})\ \frac{\Gamma, !A, !A \Rightarrow \Delta}{\Gamma, !A \Rightarrow \Delta}$$

$$(?\mathrm{WEA})\ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, ?A}\ (?\mathrm{CON})\ \frac{\Gamma \Rightarrow \Delta, ?A, ?A}{\Gamma \Rightarrow \Delta, ?A}$$

One defines *linear implication*: $A \to B = A^{\perp} \oplus B$. It is a new multiplicative. The following rules are derivable in PLL.

$$(\to\mathrm{L})\ \frac{\Gamma, B \Rightarrow \Delta;\ \Phi \Rightarrow \Psi, A}{\Gamma, \Phi, A \to B \Rightarrow \Delta, \Psi}\ (\to\mathrm{R})\ \frac{\Gamma, A \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \to B}$$

We can add $\to$ to the basic language and affix rules ($\to$L), ($\to$R) to PLL. Then, $A \to B \Rightarrow A^{\perp} \oplus B$ and $A^{\perp} \oplus B \Rightarrow A \to B$ are provable (we will write $\Leftrightarrow$ for 'both $\Rightarrow$ and $\Leftarrow$'). One also proves $A^{\perp} \Leftrightarrow A \to 0$, $A^{\perp\perp} \Leftrightarrow A$.

Girard [7] proves the cut-elimination theorem for PLL (even for first-order Linear Logic). So, the cut rule:

$$(\mathrm{CUT})\ \frac{\Gamma_1 \Rightarrow \Delta, A;\ \Gamma_2, A \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2}$$

is admissible in PLL, and this also holds for PLL with $\to$.

MALL is PLL without exponentials. MLL is MALL without additives. Since cut elimination entails the subformula property, then each fragment of PLL, defined by a restriction of language, is a conservative fragment of PLL (and all richer fragments). MELL is PLL without additives.

BCI is a propositional logic in the language ($\to$) whose Gentzen-style system admits intuitionistic sequents $\Gamma \Rightarrow A$ only. The axioms are (Id) and the inference rules are:

$$(\to\mathrm{IL})\ \frac{\Gamma, B \Rightarrow C;\ \Phi \Rightarrow A}{\Gamma, \Phi, A \to B \Rightarrow C}\ (\to\mathrm{IR})\ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \to B}$$

It is a well-known substructural logic connected with lambda-calculus [15, 4]. Clearly, ($\to$IL) and ($\to$IR) are special instances of ($\to$L) and ($\to$R), respectively. Thus, BCI is a subsystem of MLL. It is a conservative subsystem of MLL, since one easily shows that any intuitionistic $(\to)-$sequent is provable in BCI, if it is provable in MLL (first, observe that no $(\to)-$sequent of the form $\Gamma \Rightarrow$ is provable in MLL; then, use induction on derivations in the $(\to)-$fragment of MLL).

BCI with $\wedge$ will be denoted BCI$\wedge$. The system admits all intuitionistic $(\to, \wedge)-$sequents, axioms (Id) and rules ($\to$IL), ($\to$IR) and:

$$(\wedge\text{IL})\ \frac{\Gamma, A_i \Rightarrow B}{\Gamma, A_1 \wedge A_2 \Rightarrow B}\ (\wedge\text{IR})\ \frac{\Gamma \Rightarrow A;\ \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}$$

Again, BCI$\wedge$ is a conservative subsystem of MALL. The same holds for the system BCI$\vee$, which will be defined in section 3.

In section 2, we show that MALL (resp. MLL) is syntactically interpretable in BCI$\wedge$ (resp. BCI). This result was obtained in [5], using the completeness of MALL with respect to phase-space models and the completeness of BCI$\wedge$ with respect to powerset models over commutative monoids. (It was used to prove FMP of MALL as a consequence of FMP of BCI$\wedge$.) Here we go the same way, but we need strong completeness, since the interpretation must be faithful for consequence relations, not only for pure logics. Since the interpretation map is P-TIME computable, we immediately infer the NP-completeness of BCI from the NP-completeness of MLL.

In section 3, we sketch the proof of the undecidability of PLL from [10]. Then, we show that the proof yields the undecidability of the consequence relation for MALL, and, by the interpretation from section 2, the undecidability of the consequence relations for BCI$\wedge$ and BCI$\vee$.

In section 4, we briefly discuss similar results for the $(\to)-$fragment of the Lambek calculus with 0 and the cyclic rule.

2. Interpretation

BCI$\wedge$ is strongly complete with respect to powerset models over commutative monoids; the weak completeness is proved in [4]. Although the proof of strong completeness is almost the same as for weak completeness and similar to the proofs of strong completeness for the $(\otimes)-$free Lambek calculus [2, 3], we sketch it here, and then a proof of the strong completeness of MALL with respect to phase-space models. The proofs show a close relationship between both systems and help to understand the correctness of our interpretation.

Let $\mathcal{M} = (M, \cdot, 1)$ be a commutative monoid. For sets $X, Y \subseteq M$, we define operations:

$$X \cdot Y = \{ab \in M : a \in X,\, b \in Y\}$$
$$X \to Y = \{a \in M : X \cdot \{a\} \subseteq Y\}$$

The structure $(P(M), \cdot, \to, \{1\}, \subseteq)$ is a commutative residuated monoid, this means, $(P(M), \cdot, \{1\})$ is a commutative monoid, and the residuation law holds:

$$\text{(CRES) } X \cdot Y \subseteq Z \text{ iff } Y \subseteq X \to Z \text{ iff } X \subseteq Y \to Z$$

This model will be denoted by $P(\mathcal{M})$.

Let μ be an assignment of variables in $P(M)$. It is naturally extended to a homomorphism from the formula algebra to the powerset model; we set $\mu(A \wedge B) = \mu(A) \cap \mu(B)$. We also define μ on multisets of formulas ($\emptyset$ denotes the empty multiset):

$$\mu(\emptyset) = \{1\};\ \mu(\Gamma, A) = \mu(\Gamma) \cdot \mu(A)$$

We say that $\Gamma \Rightarrow A$ is *true* for μ, if $\mu(\Gamma) \subseteq \mu(A)$. A sequent is *valid* in $P(\mathcal{M})$ if it is true for all assignments μ in this model.

Let $\mathcal{A}$ be a set of sequents. BCI$\wedge(\mathcal{A})$ denotes the system BCI$\wedge$ enriched with all sequents from $\mathcal{A}$ as new axioms; the intuitionistic cut rule:

$$\text{(ICUT) } \frac{\Phi \Rightarrow A;\ \Gamma, A \Rightarrow B}{\Gamma, \Phi \Rightarrow B}$$

must be added to the system.

It is easy to show that every sequent provable in BCI$\wedge(\mathcal{A})$ is true for any pair $(P(\mathcal{M}), \mu)$ such that all sequents from $\mathcal{A}$ are true for μ (soundness). To prove the converse (completeness) one defines a special model. Let M be the set of all (finite) multisets of formulas. Let $\cdot$ denote the operation of multiset addition. Then, $(M, \cdot, \emptyset)$ is a commutative monoid. One defines μ on variables:

$$(\mu 1)\ \mu(p) = \{\Gamma \in M : \Gamma \Rightarrow p \text{ is provable in BCI} \wedge (\mathcal{A})\}$$

By induction on A, one proves:

$$(\mu 2)\ \mu(A) = \{\Gamma \in M : \Gamma \Rightarrow A \text{ is provable in BCI} \wedge (\mathcal{A})\}$$

for any formula A. For $A = B \rightarrow C$, one argues as follows. Assume $\Gamma \in \mu(A)$. Since $B \Rightarrow B$ is provable, then $B \in \mu(B)$, by the induction hypothesis. Hence $(\Gamma, B) \in \mu(C)$, and consequently, $\Gamma, B \Rightarrow C$ is provable, again by the induction hypothesis. By ($\rightarrow$IR), $\Gamma \Rightarrow A$ is provable. Assume that $\Gamma \Rightarrow A$ is provable. Then, $\Gamma, B \Rightarrow C$ is provable, by the provability of $B \rightarrow C, B \Rightarrow C$ and (ICUT). Let $\Delta \in \mu(B)$. Then, $\Delta \Rightarrow B$ is provable, by the induction hypothesis, so $\Gamma, \Delta \Rightarrow C$ is provable, by (ICUT). By the induction hypothesis again, $(\Gamma, \Delta) \in \mu(C)$. This yields $\Gamma \in \mu(A)$. The case $A = B \wedge C$ is left to the reader.

One defines a formula $\Gamma \rightarrow A$ as follows:

$$\emptyset \rightarrow A = A;\ (B, \Gamma) \rightarrow A = B \rightarrow (\Gamma \rightarrow A)$$

$\Gamma \Rightarrow A$ is provable in BCI$\wedge(\mathcal{A})$ iff $\Rightarrow \Gamma \rightarrow A$ is provable; we identify the latter sequent with the formula in its consequent. Similarly, $\Gamma \Rightarrow A$ is true for μ iff $\Gamma \rightarrow A$ is true for μ.

Accordingly, it suffices to prove the completeness for formulas. Assume that A is not provable in BCI$\wedge(\mathcal{A})$. Then $\emptyset \notin \mu(A)$, whence A is not true for μ. Let $\mathcal{A}$ consist of sequents $\Gamma_i \Rightarrow A_i$, $i = 1, \ldots, n$. All formulas $\Gamma_i \rightarrow A_i$ are provable in BCI$\wedge(\mathcal{A})$, whence $\emptyset \in \mu(\Gamma_i \rightarrow A_i)$, for $i = 1, \ldots, n$, and consequently, all sequents from $\mathcal{A}$ are true for μ. We are done.

Phase-space models are defined as follows. Let $\mathcal{M} = (M, \cdot, 1)$ be a commutative monoid. One fixes a set $\mathbf{0} \subseteq M$. For $X \subseteq M$, one defines $X^{\perp} = X \rightarrow \mathbf{0}$. The operation $C(X) = X^{\perp\perp}$ is a closure operation on $P(M)$, this means, it satisfies:(i) $X \subseteq C(X)$, (ii) $X \subseteq Y$ entails $C(X) \subseteq C(Y)$, (iii) $C(C(X)) = C(X)$, (iv) $C(X) \cdot C(Y) \subseteq C(X \cdot Y)$, for all $X, Y \subseteq M$. The closed sets $X = C(X)$ are called *facts*. X is a fact iff $X = Y^{\perp}$, for some Y. Let $F(\mathcal{M}, \mathbf{0})$ denote the set of all facts. Since $X^{\perp\perp\perp} = X$, for any $X \subseteq M$, then $X^{\perp\perp} = X$, for any fact X. One defines other operations on $F(\mathcal{M}, \mathbf{0})$:

$$X \otimes Y = (X \cdot Y)^{\perp\perp};\ X \oplus Y = (X^{\perp} \cdot Y^{\perp})^{\perp}$$

$$X \wedge Y = X \cap Y;\ X \vee Y = (X^{\perp} \cap Y^{\perp})^{\perp}$$

$F(\mathcal{M}, 0)$ is closed under $\rightarrow$. $\mathbf{0}$ is a fact, since $\mathbf{0} = \{1\}^{\perp}$.. One defines $\mathbf{1} = \mathbf{0}^{\perp} = \{1\}^{\perp\perp}$. Notice that $X \otimes Y = C(X \cdot Y)$, $X \vee Y = C(X \cup Y)$ (use $(X \cup Y) \rightarrow Z = (X \rightarrow Z) \cap (Y \rightarrow Z)$, which is true for all $X, Y, Z \subseteq M$). So, $\otimes, \wedge, \vee$ are defined as in intuitionistic phase-space models [1].

The structure $(F(\mathcal{M}, \mathbf{0}), \otimes, \oplus, ^{\perp}, \rightarrow, \wedge, \vee, \mathbf{0}, \mathbf{1}, \subseteq)$ is a model of MALL. Let μ be an assignment in this structure. We say that A is true for μ, if $1 \in \mu(A)$, which is equivalent to $\mathbf{1} \subseteq \mu(A)$. $\Gamma \Rightarrow \Delta$ is true for μ, if $\mu(\otimes(\Gamma)) \subseteq \mu(\oplus(\Delta))$, where $\otimes(\Gamma)$ arises from Γ by replacing all commas with $\otimes$, and $\oplus(\Delta)$ arises from Δ by replacing all commas with $\oplus$; we set $\otimes(\emptyset) = 1$, $\oplus(\emptyset) = 0$. Clearly, $\Gamma \Rightarrow \Delta$ is true for μ iff $\otimes(\Gamma) \rightarrow \oplus(\Delta)$ is true for μ.

An analogous equivalence holds for 'provability in MALL($\mathcal{A}$)', which is defined like 'provability in BCI$\wedge$($\mathcal{A}$)'. (CUT) must be added to MALL($\mathcal{A}$).

Girard [7] proves the weak completeness of MALL with respect to phase-space models. Actually, this proof yields strong completeness, and it imitates the above proof for BCI$\wedge$. Soundness follows from algebraic properties of $F(\mathcal{M}, \mathbf{0})$. For completeness, one constructs a special model.

This is the same construction as for BCI$\wedge$ except that we regard all formulas of MALL, replace BCI$\wedge$ by MALL, and extend the equation (μ1) for $p = 0$ and $p = 1$. We define $\mathbf{0} = \mu(0)$. Then, $\mu(p) = \{p\}^{\perp\perp}$, whence $\mu(p)$ is a fact (also for $p = 1$). To facilitate the inductive proof of (μ2) one can use the provable equivalence of any formula A to a $(\rightarrow, \wedge, 0)-$formula, which is presented below. By soundness, it suffices to prove (μ2) for formulas A of the latter form, which is the same reasoning as for BCI$\wedge$.

Each MALL-formula A can be translated into a $(\rightarrow, \wedge, 0)-$formula $T(A)$, defined as follows:

$$T(p) = p;\ T(0) = 0;\ T(1) = 0 \rightarrow 0$$

$$T(A^{\perp}) = T(A) \rightarrow 0;\ T(A \rightarrow B) = T(A) \rightarrow T(B)$$

$$T(A \otimes B) = [T(A) \rightarrow (T(B) \rightarrow 0)] \rightarrow 0;\ T(A \oplus B) = (T(A) \rightarrow 0) \rightarrow T(B)$$

$$T(A \wedge B) = T(A) \wedge T(B);\ T(A \vee B) = [(T(A) \rightarrow 0) \wedge (T(B) \rightarrow 0)] \rightarrow 0$$

By induction on A, we show that $A \Leftrightarrow T(A)$ is provable in MALL. Let $A = B \otimes C$. We assume $B \Leftrightarrow T(B)$, $C \Leftrightarrow T(C)$. By ($\rightarrow$L), we get $B, C, T(B) \rightarrow (T(C) \rightarrow 0) \Rightarrow 0$, whence $B, C \Rightarrow T(A)$, by ($\rightarrow$R), and consequently, $A \Rightarrow T(A)$, by ($\otimes$L). We prove the converse. We have $T(B) \Rightarrow B$,

$T(C) \Rightarrow C, 0$, by the assumption and (0R), whence $T(B), T(C) \Rightarrow B \otimes C, 0$, by ($\otimes$R). This yields $\Rightarrow B \otimes C, T(B) \rightarrow (T(C) \rightarrow 0)$, by ($\rightarrow$R). Using (0L) and ($\rightarrow$L), we get $T(A) \Rightarrow A$. The remaining cases are left to the reader. The proof is simpler, if one uses (CUT) and provable equivalences $A^{\perp} \Leftrightarrow A \rightarrow 0$, $A^{\perp\perp} \Leftrightarrow A$. In particular, the right-hand side of the equation for $T(A \oplus B)$ is provably equivalent to $(T(A) \rightarrow 0) \rightarrow [(T(B) \rightarrow 0) \rightarrow 0]$.

Now we define a syntactic interpretation I, which to any formula of MALL assigns a formula of BCI$\wedge$. 0 is added to the language of BCI$\wedge$ as a new constant. Let σ be a substitution such that $p\sigma = p \rightarrow 0$, for any variable p. We define $I(A) = T(A)\sigma$.

By induction on A, one shows $T(A)\sigma = T(A\sigma)$, for any formula A, and consequently, the equations defining T (except $T(p) = p$) remain true, after one has replaced T by I. The following lemmas have been proved in [5]. Here we give shorter proofs, using the provability of $A \Leftrightarrow T(A)$.

Lemma 1. *Let $\mathcal{M} = (M, \cdot, 1)$ be a commutative monoid. Let μ be an assignment in $P(\mathcal{M})$ (also defined on 0). We consider the phase-space model $F(\mathcal{M}, \mathbf{0})$, where $\mathbf{0} = \mu(0)$, and an assignment μ' (in the latter model), defined by $\mu'(p) = \mu(p\sigma)$. Then, $\mu'(A) = \mu(I(A))$, for any formula A of MALL.*

Proof. Clearly, $\mu'(p)$ is a fact, for any variable p. Since $A \Leftrightarrow T(A)$ is provable in MALL, then $\mu'(A) = \mu'(T(A))$, for any formula A. $T(A)$ is a $(\rightarrow, \wedge, 0)-$formula, whence $\mu'(T(A)) = \mu(T(A)\sigma) = \mu(I(A))$. □

Lemma 2. *Let $F(\mathcal{M}, \mathbf{0})$ be a phase-space model. Let ν be an assignment in this model. We consider the powerset model $P(\mathcal{M})$ and the assignment μ in $P(\mathcal{M})$, defined by: $\mu(p) = \nu(p\sigma)$, $\mu(0) = \mathbf{0}$. Then, $\mu(I(A)) = \nu(A)$, for any formula A of MALL.*

Proof. We define an assignment μ' in $F(\mathcal{M}, \mathbf{0})$ as in lemma 1. So, $\mu'(A) = \mu(I(A))$, by this lemma. Since $\nu(p)$ is a fact, we have

$$\mu'(p) = \mu(p \rightarrow 0) = (\nu(p) \rightarrow \mathbf{0}) \rightarrow \mathbf{0} = \nu(p)$$

for any variable p, whence $\mu' = \nu$. □

Let $\mathcal{A}$ be a set of formulas of MALL (each formula A is identified with the sequent $\Rightarrow A$). We prove the (strong) faithfulness of our interpretation.

Theorem 1. *For any formula A, A is provable in MALL($\mathcal{A}$) if, and only if, $I(A)$ is provable in BCI$\wedge$($I(\mathcal{A})$).*

Proof. Assume that $I(A)$ is not provable in BCI$\wedge$($I(\mathcal{A})$). By the strong completeness of BCI$\wedge$ with respect to powerset models, there exist a commutative monoid $\mathcal{M}$ and an assignment μ in $P(\mathcal{M})$ such that all formulas from $I(\mathcal{A})$ are true for μ, but $I(A)$ is not, this means, $1 \in \mu(I(B))$, for all $B \in \mathcal{A}$, but $1 \notin \mu(I(A))$. We construct an assignment μ' as in lemma 1. In $F(\mathcal{M}, \mathbf{0})$, we have $1 \in \mu'(B)$, for all $B \in \mathcal{A}$, but $1 \notin \mu'(A)$. By the strong soundness of MALL with respect to phase-space models, A is not provable in MALL($\mathcal{A}$).

Assume that A is not provable in MALL($\mathcal{A}$). By the strong completeness of MALL with respect to phase-space models, there exist a model $F(\mathcal{M}, \mathbf{0})$ and an assignment ν in this model such that $1 \in \nu(B)$, for all $B \in \mathcal{A}$, but $1 \notin \nu(A)$. We construct an assignment μ in $P(\mathcal{M})$ as in lemma 2. We have $1 \in \mu(I(B))$, for all $B \in \mathcal{A}$, but $1 \notin \mu(I(A))$. By the strong soundness of BCI$\wedge$ with respect to powerset models, $I(A)$ is not provable in BCI$\wedge$($I(\mathcal{A})$). □

Lemma 3. *The mapping I is computable in P-TIME.*

Proof. $T(A)\sigma$ can be computed from $T(A)$ in linear time on the length of $T(A)$ (as a string of symbols). $T(A)$ can be computed from A in time $O(n^2)$, where n is the length of A. The algorithm can be described as follows. We assume that formulas are written in Polish notation. The algorithm reads the left-most symbol. If the symbol is a variable or 0, it copies the symbol to the output tape; if the symbol is 1, it prints $\rightarrow 00$ on the output tape. If the symbol is $^{\perp}$, the algorithm determines the argument B (this can be done in n steps), prints $\rightarrow$ on the output tape, runs on B, and prints 0. If the symbol is $\rightarrow$ or $\wedge$, it determines the arguments B, C (this can be done in n steps), copies the symbol to the output tape, runs on B, and next on C. If the symbol is $\otimes$, then the output must be $\rightarrow\rightarrow T(B) \rightarrow T(C)00$. The algorithm determines B, C as above. Then, it prints $\rightarrow\rightarrow$ on the output tape, runs on B, prints $\rightarrow$, runs on C, and prints 00. The other cases are similar; we only note that, for the case of $\oplus$, the output is $\rightarrow\rightarrow T(B)0T(C)$, and for the case of $\vee$, the output is $\rightarrow \wedge \rightarrow T(B)0 \rightarrow T(C)00$. It is easy to prove that the running time is not greater than cn^2, where c is a constant depending on the computation model. More precisely, let k be the maximal constant time required to execute the non-recursive actions of each case, e.g. scanning the first symbol, marking the end of the first argument, printing 00 and so on. Then, for $A = B \circ C$, where $\circ$ is a binary connective,B of length m, C of length n, so A of length $m+n+1$, the algorithm needs the time $t(A) = m+n+k+t(B)+t(C)$. By the induction hypothesis $t(A) \leq m+n+k+cm^2+cn^2$. For $c = k$, this number is not greater than $c(m+n+1)^2$. □

Corollary 1. *The provability in BCI is NP-complete, and the same holds for the single-variable fragment of BCI.*

Proof. Kanovich [9] proves that MLL is NP-complete. Lincoln and Winkler [14] prove that even the fragment of MLL restricted to variable-free formulas is NP-complete. Our interpretation I translates MLL-formulas into formulas of BCI, and variable-free MLL-formulas into formulas of BCI, which are formed of 0 and $\rightarrow$ only. By theorem 1, A is provable in MLL iff $I(A)$ is provable in BCI. By lemma 3, this yields a P-TIME reduction of the provabilty problem for MLL to the provability problem for BCI, whence the latter is NP-hard. It is NP, since BCI is a conservative fragment of MLL. □

3. Consequence relations

We will prove that the consequence relation for BCI$\wedge$ is undecidable. First, we recall the proof of the undecidability of PLL, given in [10].

A (non-deterministic) Minsky machine stores integers in registers. A program employs a finite number of registers $R_1, \ldots, R_n$, and the integer currently stored in R_i is denoted x_i. A program is a finite set of labeled instructions; $L_0, L_1, L_2, \ldots$ are used for labels. There are five types of instructions:

(1) $L_i :\ x_m := x_m + 1;$ **goto** L_j,

(2) $L_i :\ x_m := x_m - 1;$ **goto** L_j,

(3) $L_i :$ **if** $x_m > 0$ **then goto** L_j,

(4) $L_i :$ **if** $x_m = 0$ **then goto** L_j,

(5) L_0 : **halt**.

L_0 is the label of the only instruction (5). Instructions of type (1)-(4) have labels L_i, for $i > 0$. A configuration is a $n+1-$tuple $(L_i, x_1, \ldots, x_n)$; L_i is the label of an instruction to be executed, and x_i is currently stored in R_i. The program starts in a start configuration $(L_1, k_1, \ldots, k_n)$, this means, k_i is stored in R_i, $i = 1, \ldots, n$, and the first executed instruction has label L_1. The sense of instructions should be clear. A computation is a sequence of configurations such that the first configuration C_1 is a start configuration and, for any further configuration C_k, C_k results from C_{k-1}, after one has executed an instruction whose label is contained in C_{k-1}. For instance, let $C_{k-1} = (L_1, 0, 1)$, and let $L_1 :\ x_1 := x_1 + 1$; **goto** L_3 be an instruction of the program. Then, $C_k = (L_3, 1, 1)$. The computation is successful, if it is finite and the end configuration is $(L_0, 0, 0, \ldots, 0)$.

One label (except for L_0) can be assigned to several instructions. Consequently, many divergent computations can begin in one start configuration. Without this nondeterminism, this model would not have enough strength. If (2) is to be executed, but $x_m = 0$, or (3), (4) are to be executed, but the predicate is false, then the computation ends without success. Yet, this is not essential, in general. One could employ deterministic Minsky machines as well for the purposes of this section.

A program P computes a relation R_P, consisting of all $n-$tuples $(k_1, \ldots, k_n)$ such that there is a successful computation of P from the start configuration $(L_1, k_1, \ldots, k_n)$. The relations computable by programs of the above form are precisely the recursively enumerable relations [10]. Actually, this can be shown by an easy translation of deterministic programs to programs of the above form. Consequently, the halting problem for the latter programs is undecidable. The halting problem is the following: given P and $(k_1, \ldots, k_n)$, decide whether $(k_1, \ldots, k_n)$ belongs to R_P.

Instructions (1)-(4) are represented by MALL-formulas. For the given program P, these formulas employ variables $r_1, \ldots, r_n$, $d_1, \ldots, d_n$ and $l_0, l_1, l_2, \ldots$. The variables r_i, d_i are related to the register R_i, and l_j to the label L_j, appearing in P. The idea is to represent the configuration $(L_i, k_1, \ldots, k_n)$ by the multiset $L_i, r_1^{k_1}, \ldots, r_n^{k_n}$. The formulas corresponding to (1)-(4) are:

(F1) $l_i \rightarrow (l_j \otimes r_m)$,

(F2) $(l_i \otimes r_m) \rightarrow l_j$, (F3) $(l_i \otimes r_m) \rightarrow (l_j \otimes r_m)$,

(F4) $l_i \rightarrow (l_j \vee d_m)$.

D_m is the set of the following formulas: $(d_m \otimes r_j) \rightarrow d_m$, for all $j \neq m$, $j = 1, \ldots, n$, and $d_m \rightarrow l_0$. So, the role of d_m is to delete all variables r_j, $j \neq m$. $\Gamma(P)$ is defined as the join of the set of all formulas corresponding to instructions from P and sets D_m, $m = 1, \ldots, n$. Of course, a set can be treated as a multiset without multiple occurrences of elements. $!\Gamma(P)$ denotes the multiset arising from $\Gamma(P)$, after one has replaced each element A by $!A$.

Kanovich [10] proves the following equivalence:

$$(k_1, \ldots, k_n) \in R_P \text{ iff } l_1, r_1^{k_1}, \ldots, r_n^{k_n}, !\Gamma(P) \Rightarrow l_0 \text{ is provable in PLL.}$$

This yields the undecidability of PLL. The sequent on the right-hand side will be denoted by $S(1, k_1, \ldots, k_n)$, and by $S(i, k_1, \ldots, k_n)$, if l_1 is replaced by l_i.

Kanovich's proof is indirect; he uses tree-like Horn programs as an intermediate tool. This equivalence can also be proved directly: the 'only if' part by induction on the length of a successful computation and the 'if' part essentially by induction on cut-free derivations in PLL. We do not present this full proof here, but we consider one case for instructions of type (4) to explain the role of $\vee$ in (F4).

For the 'only if' part, one proves a more general claim: if there exists a successful computation of P from $(L_i, k_1, \ldots, k_n)$, then $S(i, k_1, \ldots, k_n)$ is provable in PLL. We proceed by induction on the length of a successful computation. Assume that on the first step one executes the instruction (4). Then, $k_m = 0$, and the second configuration is $(L_j, k_1, \ldots, k_n)$ (no change in registers). By the induction hypothesis, $S(j, k_1, \ldots, k_n)$ is provable in PLL. Starting from $l_0 \Rightarrow l_0$ and using ($\rightarrow$L) several times, one proves a sequent:

$$d_m, r_1^{k_1}, \ldots, r_n^{k_n}, \Delta \Rightarrow l_0$$

where Δ is a multiset of formulas from D_m. Using (!L), (!CON), (!WEA), one proves:

$$d_m, r_1^{k_1}, \ldots, r_n^{k_n}, !\Gamma(P) \Rightarrow l_0$$

By ($\vee$L), from $S(j, k_1, \ldots, k_n)$ and the latter sequent, one gets a similar sequent in which $l_j \vee d_m$ replaces d_m. Then, by ($\to$L), one obtains:

$$l_i, l_i \to (l_j \vee d_m), r_1^{k_1}, \ldots, r_n^{k_n}, !\Gamma(P) \Rightarrow l_0$$

Finally, one proves $S(i, k_1, \ldots, k_n)$, by (!L), (!CON).

The above derivation would be impossible, if r_m occurred in $S(j, k_1, \ldots, k_m)$. This could happen, if (4) were executed incorrectly, with a non-zero content of R_m. Accordingly, $\vee$ in (F4) and the formulas from D_m are responsible for 'zero check'.

The proof of the 'if' part is more involved, since derivations in MALL need not directly follow computations of the machine. One proves a series of lemmas which show that every derivation of $S(i, k_1, \ldots, k_n)$ can be reduced to a normal form which simulates a successful computation. We omit details; a full proof, using tree-like Horn programs, can be found in [10].

We fix a program P. For any formula $A \in \Gamma(P)$, we choose a new variable p_A (it is different from all variables r_m, d_m, l_i). Let $\mathcal{A}$ consist of the following formulas of MALL:

$$\text{(A1) } p_A \to A, \text{ (A2) } p_A \to 1, \text{ (A3) } p_A \to p_A \otimes p_A$$

for any formula A from $\Gamma(P)$. Let $\Phi(P)$ denote the set of all atoms p_A, for $A \in \Gamma(P)$. By $\varphi(i, k_1, \ldots, k_n)$ we denote the sequent:

$$l_i, r_1^{k_1}, \ldots, r_n^{k_n}, \Phi(P) \Rightarrow l_0$$

Notice that $S(i, k_1, \ldots, k_n)$ is a $(\to, \vee, \otimes, !)-$sequent in which every occurrence of a formula $!A$ is negative. Recall that each formula positively occurs in itself. If a formula occurs positively (resp. negatively) in B, then it occurs positively (resp. negatively) in $!B$, $B \circ C$, $C \circ B$, for $\circ \in \{\otimes, \oplus, \wedge, \vee\}$, and in $C \to B$, and it occurs negatively (resp. positively) in $B^\perp$ and $B \to C$. For a sequent $\Gamma \Rightarrow \Delta$, formulas occurring positively (resp. negatively) in formulas from Γ occur negatively (resp. positively) in the sequent, and formulas occurring positively (resp. negatively) in formulas from Δ occur positively (resp. negatively) in the sequent.

In a sequent system like PLL, left-introduction rules introduce negative occurrences of formulas, and right-introduction rules introduce positive occurrences of formulas. No inference rule changes the polarity of any subformula. Accordingly, if $S(i, k_1, \ldots, k_n)$ is provable in PLL, then it is provable without any application of (!R).

Lemma 4. *Let S be a sequent of PLL in which any subformula of the form $!A$ occurs negatively and belongs to $!\Gamma(P)$; S contains no new variable p_A. Let S' arises from S by replacing each subformula $!A$ by p_A. Then, S is provable in PLL if, and only if, S' is provable in MALL($\mathcal{A}$).*

Proof. We prove the 'if' part. Assume that S' is provable in MALL($\mathcal{A}$). Fix a derivation. In this derivation, substitute $!A$ for every occurrence of p_A, for any $A \in \Gamma(P)$. New axioms (A1), (A2), (A3) are transformed into some formulas provable in PLL, and S' is transformed into S. Consequently, S is provable in PLL.

We prove the 'only if' part. Assume that S is provable in PLL. We proceed by induction on cut-free derivations in PLL. If S is an axiom (Id), (0L) or (1R), then it contains no occurrence of ! (for the case of (Id), if ! occurred in S, then there would be both positive and negative occurrences of ! in S), and consequently $S' = S$. If S arises by any rule, not introducing !, then we apply the induction hypothesis to the premises and the same rule in MALL($\mathcal{A}$). Assume that S arises by (!L), (!WEA) or (!CON). Observe that the following sequents are provable in MALL($\mathcal{A}$):

$$\text{(S1) } p_A \Rightarrow A, \text{ (S2) } p_A \Rightarrow 1, \text{ (S3) } p_A \Rightarrow p_A \otimes p_A$$

by (Id), ($\rightarrow$L), (A1)-(A3), and (CUT). Let S arise by (!L). So, S equals $\Gamma, !A \Rightarrow \Delta$, and the premise is $\Gamma, A \Rightarrow \Delta$. We apply the induction hypothesis to the premise, then (S1) and (CUT), which proves S' in MALL($\mathcal{A}$). Let S arise by (!WEA). So, S is as above, and the premise is $\Gamma \Rightarrow \Delta$. We apply the induction hypothesis to the premise, then (1L), (S2) and (CUT), which proves S' in MALL($\mathcal{A}$). Let S arise by (!CON). Then, S is as above, and the premise is $\Gamma, !A, !A \Rightarrow \Delta$. We apply the induction hypothesis to the premise, then ($\otimes$L), (S3) and (CUT), which proves S' in MALL(**A**). □

Corollary 2. *$S(i, k_1, \ldots, k_n)$ is provable in PLL if, and only if, $\varphi(i, k_1, \ldots, k_n)$ is provable in MALL($\mathcal{A}$).*

Corollary 3. *The consequence relation for MALL is undecidable, and it also holds for the ($\rightarrow, \otimes, \vee$)−fragment of MALL.*

Proof. This follows from corollary 2 and the result of Kanovich. In the proof of lemma 4, if S is a $(\rightarrow, \otimes, \vee, !)-$sequent, then the resulting proof of S' uses the restricted fragment of MALL($\mathcal{A}$) only. □

The undecidability of the consequence relation for MALL is known, (in [16] it is stated without proof), but we need the above argument, especially in the proof of theorem 3.

There exists a program P such that R_P is not recursive. Then, there exists a finite set $\mathcal{A}$ such that the provability problem for MALL($\mathcal{A}$) is undecidable. As a consequence of theorem 1 and corollary 3, we obtain:

Theorem 2. *The consequence relation for BCI$\wedge$ is undecidable.*

There exists a finite set $\mathcal{A}$, of formulas of MALL, such that the provability problem for BCI$\wedge$($I(\mathcal{A})$) is undecidable. By the above results, $\mathcal{A}$ can be the set of formulas (A1), (A2) ,(A3), for $A \in \Gamma(P)$, where P is a program. The provability problem in MALL($\mathcal{A}$) is undecidable, even for sequents of the form $\varphi(i, k_1, \ldots, k_n)$. If S is a sequent of this form, then $I(S)$ is a $(\rightarrow)-$sequent; here we set $I(\Gamma \Rightarrow A) = I(\Gamma) \Rightarrow I(A)$, where $I(\Gamma)$ is the image of the multiset Γ under I, which agrees with the translation of $\Gamma \Rightarrow A$ into the formula $\Gamma \rightarrow A$. Accordingly, the relation '$\Gamma \Rightarrow A$ is provable in BCI$\wedge$($I(\mathcal{A})$)' is undecidable even for $(\rightarrow)-$sequents $\Gamma \Rightarrow A$. On the other hand, some formulas in $I(\mathcal{A})$ contain $\wedge$, namely the formulas $I(B)$, for B of the form (A1), where A is of the form (F4). We have:

$$I((p, p') \rightarrow (q \vee r)) = (I(p), I(p')) \rightarrow \{[(I(q) \rightarrow 0) \wedge (I(r) \rightarrow 0)] \rightarrow 0\}$$

(According to our notation, $(p, p') \rightarrow A$ denotes $p \rightarrow (p' \rightarrow A)$.) So, the consequence relation for BCI$\wedge$ is undecidable, even for assumptions in $(\rightarrow, \wedge)$, not containing positive occurrences of $\wedge$, and conclusions in $(\rightarrow)$.

An analogous theorem holds for BCI$\vee$, which is the $(\rightarrow, \vee)-$fragment of MALL. It can be axiomatized in the language $(\rightarrow, \vee)$ by axioms (Id), rules ($\rightarrow$IL), ($\rightarrow$IR) and:

$$(\vee\text{IL}) \ \frac{\Gamma, A \Rightarrow C; \ \Gamma, B \Rightarrow C}{\Gamma, A \vee B \Rightarrow C} \quad (\vee\text{IR}) \ \frac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_1 \vee A_2}$$

BCI($\wedge, \vee$) denotes the system in the language $(\rightarrow, \wedge, \vee)$ which admits all axioms and rules of BCI$\wedge$ and BCI$\vee$. It amounts to the $(0, 1, \otimes)-$free fragment of Full Lambek Calculus with Exchange (FL$_e$) [15, 1]. In BCI($\wedge, \vee$)

$(A \vee B) \to C \Leftrightarrow (A \to C) \wedge (B \to C)$, whence $I((p, p') \to (q \vee r))$ is provably equivalent to:

$$(\mathrm{I}\vee)\ (I(p), I(p')) \to \{[(I(q) \vee I(r)) \to 0] \to 0\}$$

Theorem 3. *The consequence relation for BCI∨ is undecidable.*

Proof. Let $\mathcal{A}$ be a set of formulas, corresponding to a program P. Let $\mathcal{B}$ arise from $I(\mathcal{A})$ by replacing each formula $I((p, p') \to (q \vee r))$ by the corresponding formula (I∨). Let $\Gamma \Rightarrow A$ be a $(\to)$−sequent. We prove: $\Gamma \Rightarrow A$ is provable in BCI$\wedge(I(\mathcal{A}))$ iff it is provable in BCI$\vee(\mathcal{B})$. The theorem follows from this equivalence, by the above results.

Obviously, in BCI$(\wedge, \vee)$, the sets $I(\mathcal{A})$ and $\mathcal{B}$ have the same consequences. So, it suffices to prove: (1) $\Gamma \Rightarrow A$ is a consequence of $I(\mathcal{A})$ in BCI∧ iff it is so in BCI$(\wedge, \vee)$, (2) $\Gamma \Rightarrow A$ is a consequence of $\mathcal{B}$ in BCI∨ iff it is so in BCI$(\wedge, \vee)$. The 'only if' parts of (1), (2) are obvious.

To prove the 'if' part of (1) assume that $\Gamma \Rightarrow A$ is not a consequence of $I(\mathcal{A})$ in BCI∧. By the completeness results of section 2, there exist a powerset model $P(\mathcal{M})$ and an assignment μ in $P(\mathcal{M})$ such that all formulas from $I(\mathcal{A})$ are true, but $\Gamma \Rightarrow A$ is not true for μ. $P(\mathcal{M})$ can be expanded to a model of BCI$(\wedge, \vee)$: one interprets $\vee$ as the join of sets. Then, $\Gamma \Rightarrow A$ is not a consequence of $I(\mathcal{A})$ in BCI$(\wedge, \vee)$.

The 'if' part of (2) can be proved in a similar way, although the methods of section 2 do not provide the strong completeness of BCI∨ with respect to powerset models in which $\vee$ is interpreted as $\cup$. One proves the strong completeness of BCI∨ with respect to intuitionistic phase-space models whose elements are closed subsets of a monoid, and one defines $X \vee Y = C(X \cup Y)$. In the special model, the monoid $\mathcal{M}$ is as above (see section 2), and $C(X)$ is the meet of all sets $[A]$, containing X, where $[A] = \{\Gamma \in M : \Gamma \Rightarrow A \text{ is provable}\}$ [1]. The closed sets are closed under $\cap$, whence these models can be expanded to models of BCI$(\wedge, \vee)$. □

4. Non-commutative logics

Similar results can be obtained for some non-commutative systems. We consider the Lambek calculus L*, a variant of the Lambek calculus L, introduced in [11]. L* is the $(\otimes, \to, \leftarrow)$−fragment of Full Lambek Calculus

(FL) [15, 1]. Now, Greek capitals denote finite sequences of formulas. Sequents are of the form $\Gamma \Rightarrow A$. The axioms are (Id), and the rules are:

$$(\otimes\text{nIL})\ \frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, A \otimes B, \Delta \Rightarrow C}\ (\otimes\text{nIR})\ \frac{\Gamma \Rightarrow A;\ \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \otimes B}$$

$$(\rightarrow\text{nIL})\ \frac{\Gamma, B, \Delta \Rightarrow C;\ \Phi \Rightarrow A}{\Gamma, \Phi, A \rightarrow B, \Delta \Rightarrow C}\ (\rightarrow\text{nIR})\ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B}$$

$$(\leftarrow\text{nIL})\ \frac{\Gamma, B, \Delta \Rightarrow C;\ \Phi \Rightarrow A}{\Gamma, B \leftarrow A, \Phi, \Delta \Rightarrow C}\ (\leftarrow\text{nIR})\ \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow B \leftarrow A}$$

In FL one adds rules for $\wedge$, $\vee$, 1 and (possibly) 0. L is restricted to sequents with nonempty antecedents. L* is strongly complete with respect to powerset models over monoids [3]. For its $\otimes-$free fragment, the proof is analogous to that for BCI$\wedge$. In powerset models, $\leftarrow$ is defined by:

$$Y \leftarrow X = \{a \in M : \{a\} \cdot X \subseteq Y\}$$

Cyclic MLL is classical system in the language $(\otimes, \oplus, ^{\perp}, 0, 1)$. $\rightarrow$ and $\leftarrow$ are definable: $A \rightarrow B = A^{\perp} \oplus B$, $B \leftarrow A = B \oplus A^{\perp}$. We omit axiomatization. Yetter [19] proves that the system is complete with respect to cyclic phase-space models, described below (he does it for Cyclic MALL).

Let $\mathcal{M} = (M, \cdot, 1)$ be a monoid. A set $X \subseteq M$ is said to be *cyclic*, if it satisfies the condition: if $ab \in X$, then $ba \in X$, for all $a, b \in M$. A cyclic phase-space model is determined by a pair $(\mathcal{M}, \mathbf{0})$ such that $\mathcal{M}$ is a monoid and $\mathbf{0}$ is a cyclic subset of M. For sets $X, Y \subseteq M$, one defines $X \cdot Y$, $X \rightarrow Y$ and $Y \leftarrow X$, as above. One also defines $X^{\perp} = X \rightarrow \mathbf{0}$. Since $\mathbf{0}$ is cyclic, then $X^{\perp} = \mathbf{0} \leftarrow X$. The operation $C(X) = X^{\perp\perp}$ is a closure operation on $P(M)$. The closed sets are called facts. X is a fact iff $X = Y^{\perp}$, for some set Y. (One uses $Y^{\perp\perp\perp} = Y^{\perp}$, for any set Y.) One defines the operations $\otimes, \oplus$ on facts:

$$X \otimes Y = (X \cdot Y)^{\perp\perp};\ X \oplus Y = (Y^{\perp} \cdot X^{\perp})^{\perp}$$

Facts are closed under $\rightarrow, ^{\perp}$. $\mathbf{0}$ is a fact. One defines $\mathbf{1} = \mathbf{0}^{\perp}$. For an assignment μ (of facts to variables), A is true for μ, if $1 \in \mu(A)$.

As above, the completeness proof yields, actually, the strong completeness of Cyclic MLL with respect to cyclic phase-space models. We will define an interpretation of Cyclic MLL in L*C, i.e. L* with a designated constant 0 and the cyclic rule:

$$\text{(C)}\ \frac{\Gamma, \Delta \Rightarrow 0}{\Delta, \Gamma \Rightarrow 0}$$

The rule (ICUT) is admissible in L* and L*C. Our interpretation I, defined in [5], maps formulas of Cyclic MLL into $(\rightarrow, 0)-$formulas of L*C. Then, we interpret Cyclic MLL in the $(\rightarrow, 0)-$fragment of L*C. First, $T(A)$ is defined as for MALL with the only modification:

$$T(A \otimes B) = [T(B) \rightarrow (T(A) \rightarrow 0)] \rightarrow 0$$

(in noncommutative models $(X \cdot Y) \rightarrow Z = Y \rightarrow (X \rightarrow Z)$). As above, $I(A) = T(A)\sigma$.

Theorem 4. *A is a consequence of $\mathcal{A}$ in Cyclic MLL if, and only if, $I(A)$ is a consequence of $I(\mathcal{A})$ in the $(\rightarrow, 0)-$fragment of L*C.*

The proof uses the analogues of lemmas 1 and 2. First, one proves that L*C is strongly complete with respect to powerset models over monoids in which 0 is interpreted as a cyclic set.

Pentus [17] proves the NP-completeness of L, L* and Cyclic MLL. Since I is P-TIME computable, the $(\rightarrow, 0)-$fragment of L*C is NP-complete. (It is NP, as a conservative subsystem of Cyclic MLL; this can also be shown directly.)

The consequence relations for L* and Cyclic MLL are undecidable. This follows from the fact that the word problem for groups is reducible to each of these relations. (Notice that groups are models for these systems: both $\otimes$ and $\oplus$ are interpreted as the group operation $\cdot$, $^\perp$ as $^{-1}$, $a \rightarrow b = a^{-1}b$, $b \leftarrow a = ba^{-1}$, $0 = 1$, and $\Rightarrow$ as $=$.) One can also reduce some word problem for semigroups, by some finer considerations; see [2, 6], or use the undecidability of the consequence relation for the $(\rightarrow)-$fragment of L* (in [2] it was shown for L, but the proof also worked for L*) and the fact that this fragment is conservative in L*C, also for consequence relations. By theorem 4, we obtain the undecidability of the consequence relation for the $(\rightarrow, 0)-$fragment of L*C.

In a similar way, Cyclic MALL can be interpreted in L*C$\wedge$. For systems with additives, the provability relation is P-SPACE (an upper bound). The complexity classes of the $\otimes-$free fragments of L* are not known.

Acknowledgements. This paper was written during the author's stay at the Rovira i Virgili University, Research Group on Mathematical Linguistics, in Tarragona, 2007.

References

[1] F. Belardinelli, P. Jipsen and H. Ono, *Algebraic Aspects of Cut Elimination*, Studia Logica **77** (2004), pp. 209–240.

[2] W. Buszkowski, *Some decision problems in the theory of syntactic categories*, Zeitschrift f. mathematische Logik u. Grundlagen d. Math. **28** (1982), pp. 539–548.

[3] W. Buszkowski, *Completeness results for Lambek Syntactic Calculus*, Zeitschrift f. math. Logik u. Grundlagen d. Math., **32** (1986), pp. 13–28.

[4] W. Buszkowski, *The Finite Model Property for BCI and Related Systems*, Studia Logica **57** (1996), pp. 303–323.

[5] W. Buszkowski, *Finite Models of Some Substructural Logics*, Mathematical Logic Quarterly **48** (2002), pp. 63–72.

[6] W. Buszkowski, *Lambek Calculus with Nonlogical Axioms*, in: C. Casadio, P.J. Scott, R. Seely (eds.), Language and Grammar. Studies in Mathematical Linguistics and Natural Language, CSLI Lecture Notes 168, Stanford, 2005, pp. 77–93.

[7] J-Y. Girard, *Linear logic*, Theoretical Computer Science **50**, (1987), pp. 1–102.

[8] J-Y. Girard, Y. Lafont, L. Regnier (eds.), Advances in Linear Logic, Cambridge University Press, 1995.

[9] M.I. Kanovich, *The multiplicative fragment of linear logic is NP-complete*, Technical Report X-91-13, Institute for Logic, Language and Information, Amsterdam, 1991.

[10] M.I. Kanovich, *The Direct Simulation of Minsky Machines in Linear Logic*, in: [8], pp. 123–145.

[11] J. Lambek, *The mathematics of sentence structure*, American Mathematical Monthly **65** (1958), pp. 154–170.

[12] P. Lincoln, *Deciding Provability of Linear Logic Formulas*, in: [8], pp. 109–122.

[13] P. Lincoln, J.Mitchell, A. Scedrov, N. Shankar, *Decision problems for propositional linear logic*, Annals of Pure and Applied Logic **56** (1992), pp. 239–311.

[14] P. Lincoln, T. Winkler, *Constant-Only Multiplicative Linear Logic is NP-Complete*, Theoretical Computer Science **135** (1994), pp. 155–169.

[15] H. Ono, *Semantics of Substructural Logics*, in: [18], pp. 259–291.

[16] H. Ono, *Decidabilty and finite model property of substructural logics*, in: J. Ginzburg et al., eds., The Tbilisi Symposium on Logic, Language and Computation: Selected Papers, CSLI Publications, 1998, pp. 263–274.

[17] M. Pentus, *Lambek calculus is NP-complete*, Theoretical Computer Science **357** (2006), pp. 186–201.

[18] P. Schroeder-Heister, K. Dosen (eds.), Substructural Logics, Clarendon Press, Oxford, 1993.

[19] D.N. Yetter, *Quantales and (Non-Commutative) Linear Logic*, Journal of Symbolic Logic **55** (1990), pp. 41–64

Adam Mickiewicz University in Poznań
Rovira i Virgili University in Tarragona
buszko@amu.edu.pl