Undecidability of consequence relation in Full Non-associative Lambek Calculus

Karel Chvalovský

Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 271/2, 182 07 Prague 8, Czech Republic

e-mail: karel@chvalovsky.cz

May 16, 2014

Abstract

We prove that the consequence relation in the Full Non-associative Lambek Calculus is undecidable. An encoding of the halting problem for 2-tag systems using finitely many sequents in the language $\{\cdot, \vee\}$ is presented. Therefore already the consequence relation in this fragment is undecidable. Moreover, the construction works even when the structural rules of exchange and contraction are added.

1 Introduction

There are many motivations for studying substructural logics, which can be seen as logics lacking some structural rules when presented in the form of sequent calculi. It is of no surprise that some of the motivations come from linguistics. Lambek [13] introduced a calculus, which is now called after him (**L**), with exactly these motivations in mind. All the normal structural rules, i.e. exchange, contraction, and weakening, are missing in **L**. The standard language of **L** contains product ($\cdot$) and two implications ($/$ and $\backslash$) as connectives. Nevertheless, it is common in substructural logics to consider also additive join ($\vee$) and meet ($\wedge$). By adding rules for all these connectives we obtain the Full Lambek Calculus (**FL**).

With linguistics in mind it is natural that Lambek [14] also introduced a non-associative variant of his calculus, we call it **NL**. The main difference can be demonstrated by structures used to represent sequents. In **L** the basic structure is a sequence of formulae, but non-associativity in **NL** requires a tree with leaves that are labelled by formulae. Again by adding rules for join and meet we obtain the Full Non-associative Lambek Calculus (**FNL**), for details see e.g. [4, 9].

A natural question to ask is whether or not the provability in these calculi is decidable. It is known [4] that provability in **FNL** is decidable in polynomial space. The problem we deal with in this paper is whether provability in **FNL** is decidable if finitely many non-logical axioms are added, i.e. we study the decidability of the finitary consequence relation in **FNL**. Although this problem was shown [3] to be decidable in polynomial time for **NL**, we prove that it is undecidable for **FNL** by encoding the halting problem for 2-tag systems in the language $\{\cdot, \vee\}$, i.e. we show that the provability from finitely many non-logical axioms is undecidable in this fragment. Moreover, it is undecidable even if the structural rules of exchange and contraction are added. It is worth noting that we are in non-associative setting, because associativity usually plays an important role in similar results, cf. the undecidability of consequence relation in **L** [2, 3]. However, we show that the distributivity of product over join is sufficient. It is worth noting that the consequence relation in the distributive **FNL**, the distributive laws hold for $\{\wedge, \vee\}$, is decidable, see [4, 7, 11].

Although this paper deals solely with logic, there are some algebraic consequences. **FNL** is complete with respect to lattice-ordered residuated groupoids. Therefore we prove that their word problem is undecidable. This solves negatively Problem 7.1 in [11]. Moreover, the encoding of the halting problem for 2-tag systems can be used directly to obtain a similar result for their $\{\cdot, \vee\}$-reduct—join-semilattices expanded by a groupoid operation (product) where product distributes over join. Our construction shows that the word problem for such structures is generally undecidable. This is true even if $x \cdot y = y \cdot x$ and $x \leq x \cdot x$ hold. Moreover, we do not need the idempotency and commutativity of join in full generality.

The paper is organized as follows. The next section contains some basic definitions. As for our purposes it is sufficient to deal with $\{\cdot, \vee\}$ we concentrate solely on this language. Nevertheless, we present an entire sequent calculus for **FNL** first. Then we discuss a sequent calculus for the $\{\cdot, \vee\}$-fragment and some equivalences on formulae. In the last part of this section tag systems are introduced. In Section 3 we describe how a tag system can be encoded in the language $\{\cdot, \vee\}$. The most important part describes non-logical axioms which express the behaviour of a given tag system. Then we prove, in Sections 4 and 5, the correctness and completeness of our encoding. In Section 6 we discuss some possible modifications—addition of the structural rules and the one-variable fragment. Section 7 contains some algebraic consequences of our construction. A formulation of our results in terms of term rewriting systems is briefly discussed in Section 8.

2 Preliminaries

Formulae are formed out in the standard way from a denumerable set of propositional variables (atoms) using connectives and parentheses. All the connectives are binary: product ($\cdot$), two implications ($\backslash$ and $/$), join ($\vee$), and meet ($\wedge$). We denote formulae by small Greek letters and do not write parentheses if no confusion can arise.

As we are in non-associative setting we have to impose some other notions, following e.g. [3, 4, 9], which simplify the formulation of rules in our sequent calculus and handling of substitutions later on. We say that all formulae are (atomic) *structures*, the empty structure ε is a structure, and if $\mathbb{S}$ and $\mathbb{T}$ are structures then $(\mathbb{S}, \mathbb{T})$ is a structure, nothing else is a structure. Moreover, $(\varepsilon, \mathbb{S}) = (\mathbb{S}, \varepsilon) = \mathbb{S}$ and we always assume that a structure does not contain unnecessary empty structures. A *context* is a structure $\mathbb{S}[\circ]$ which has a single occurrence of the special structure symbol $\circ$, a place for substitution. Then $\mathbb{S}[\mathbb{T}]$ denotes the result of substitution of $\mathbb{T}$ for $\circ$ in $\mathbb{S}[\circ]$. We use the blackboard bold font for structures and contexts are distinguished by using square brackets for substitutions.

2.1 Full Non-associative Lambek Calculus

We present a sequent calculus for **FNL**, cf. [3, 4, 9]. For any structure $\mathbb{S}$ and formula φ we say that $\mathbb{S} \Rightarrow \varphi$ is a *sequent*. The definition of provability in a sequent calculus is standard—a proof is a tree labelled by sequents, where only axioms can occur as leaves and every other vertex (sequent) is obtained by an inference rule from its children.

Definition 2.1. Let φ, ψ, χ, φ_1, and φ_2 be arbitrary formulae and $\mathbb{S}$, $\mathbb{T}$ be arbitrary structures. The sequent calculus for **FNL** has the following axioms and inference rules:

$$(\mathrm{Id})\ \varphi \Rightarrow \varphi \qquad\qquad (\mathrm{Cut})\ \frac{\mathbb{S}[\varphi] \Rightarrow \psi \qquad \mathbb{T} \Rightarrow \varphi}{\mathbb{S}[\mathbb{T}] \Rightarrow \psi}$$

$$(\cdot\mathrm{L})\ \frac{\mathbb{S}[(\varphi,\psi)] \Rightarrow \chi}{\mathbb{S}[\varphi\cdot\psi] \Rightarrow \chi} \qquad (\cdot\mathrm{R})\ \frac{\mathbb{S}\Rightarrow\varphi \qquad \mathbb{T}\Rightarrow\psi}{(\mathbb{S},\mathbb{T})\Rightarrow\varphi\cdot\psi}$$

$$(\backslash\mathrm{L})\ \frac{\mathbb{S}[\varphi]\Rightarrow\psi \qquad \mathbb{T}\Rightarrow\chi}{\mathbb{S}[(\mathbb{T},\chi\backslash\varphi)]\Rightarrow\psi} \qquad (\backslash\mathrm{R})\ \frac{(\varphi,\mathbb{S})\Rightarrow\psi}{\mathbb{S}\Rightarrow\varphi\backslash\psi}$$

$$(/\mathrm{L})\ \frac{\mathbb{S}[\varphi]\Rightarrow\psi \qquad \mathbb{T}\Rightarrow\chi}{\mathbb{S}[(\varphi/\chi,\mathbb{T})]\Rightarrow\psi} \qquad (/\mathrm{R})\ \frac{(\mathbb{S},\varphi)\Rightarrow\psi}{\mathbb{S}\Rightarrow\psi/\varphi}$$

$$(\vee\mathrm{L})\ \frac{\mathbb{S}[\varphi]\Rightarrow\chi \qquad \mathbb{S}[\psi]\Rightarrow\chi}{\mathbb{S}[\varphi\vee\psi]\Rightarrow\chi} \qquad (\vee\mathrm{R})\ \frac{\mathbb{S}\Rightarrow\varphi_i}{\mathbb{S}\Rightarrow\varphi_1\vee\varphi_2}\ \text{for } i=1,2$$

$$(\wedge\mathrm{L})\ \frac{\mathbb{S}[\varphi_i]\Rightarrow\psi}{\mathbb{S}[\varphi_1\wedge\varphi_2]\Rightarrow\psi}\ \text{for } i=1,2 \qquad (\wedge\mathrm{R})\ \frac{\mathbb{S}\Rightarrow\varphi \qquad \mathbb{S}\Rightarrow\psi}{\mathbb{S}\Rightarrow\varphi\wedge\psi}$$

Our aim is to discuss calculi with non-logical axioms. Therefore we extend the previous definition to handle them. They are sequents $\mathbb{S}\Rightarrow\alpha$, where $\mathbb{S}$ and α are a structure and a formula, respectively. However, let us remark that non-logical axioms are not closed under substitutions and therefore propositional variables in them are treated as constants. We use Φ for a *set of non-logical axioms* and **FNL**(Φ) for the sequent calculus containing axioms and rules for **FNL** and non-logical axioms from Φ.

As the reader probably anticipated the intended meaning of comma in structures is product. Two natural extremes arise: the case when there are as many as possible commas instead of products and vice versa. For this purposes we define translations σ and ρ.

Definition 2.2. Let $\mathbb{S}$ be any structure. The functions σ and ρ on structures are given by

$$\sigma(\mathbb{S}) = \begin{cases} (\sigma(\mathbb{T}),\sigma(\mathbb{U})) & \text{if } \mathbb{S}=(\mathbb{T},\mathbb{U}),\\ (\sigma(\varphi),\sigma(\psi)) & \text{if } \mathbb{S}=\varphi\cdot\psi,\\ \mathbb{S} & \text{otherwise.}\end{cases}$$

$$\rho(\mathbb{S}) = \begin{cases} \rho(\mathbb{T})\cdot\rho(\mathbb{U}) & \text{if } \mathbb{S}=(\mathbb{T},\mathbb{U}),\\ \mathbb{S} & \text{otherwise.}\end{cases}$$

It is clear that $\rho(\mathbb{S})=\rho(\sigma(\mathbb{S}))$, $\sigma(\mathbb{S})=\sigma(\rho(\mathbb{S}))$, and $\rho(\mathbb{S})$ is a formula, for non-empty $\mathbb{S}$.

Example 2.1. It holds that $\sigma(p\cdot(q\cdot r))=(p,(q,r))$, but $\sigma(p\vee(q\cdot r))=p\vee(q\cdot r)$, where p, q, and r are atoms.

Lemma 2.1. *For any set of non-logical axioms Φ, structure $\mathbb{S}$, and formula φ it holds that $\mathbb{S}\Rightarrow\varphi$ is provable in* **FNL**(Φ) *iff $\sigma(\mathbb{S})\Rightarrow\varphi$ is provable in* **FNL**(Φ) *iff $\rho(\mathbb{S})\Rightarrow\varphi$ is provable in* **FNL**(Φ).

Proof. It is sufficient to use ($\cdot$R) and (Cut), or ($\cdot$L). □

As our aim is to produce an encoding of an undecidable problem, we will have to show that such an encoding is correct and complete. In order to prove completeness we will study all the possible proofs of some sequents in our calculus. For this reason we want to produce a simpler calculus which proves the very same sequents.

A natural restriction is to allow only very specific cuts. We call a sequent $\mathbb{S}\Rightarrow\varphi$ *regular* if the only formulae occurring in the structure $\mathbb{S}$ are propositional variables, i.e. it contains no connectives. A set of non-logical axioms Φ is regular if all its members are regular. Let us assume the notation in Definition 2.1 for (Cut). We call φ there the *cut formula*. We say that a cut is *principal* if $\mathbb{T} \Rightarrow \varphi$ is a regular sequent from non-logical axioms, cf. [17, p. 174]. A proof containing only principal cuts is called *standard*.

We need to define some more notions. The *grade of a cut* is the length of the cut formula φ. The *size of a cut* is the number of sequents appearing in the proof of $\mathbb{S}[\varphi] \Rightarrow \psi$ and $\mathbb{T} \Rightarrow \varphi$. However, the size of all principal cuts is defined to be 0. Every application of a rule for connectives creates a formula and we call it the *main formula*. Other formulae appearing in this sequent are called *side formulae*.

Theorem 2.2. *For any regular set of non-logical axioms Φ, structure $\mathbb{S}$, and formula φ it holds that $\mathbb{S} \Rightarrow \varphi$ has a proof in* $\mathbf{FNL}(\Phi)$ *iff* $\mathbb{S} \Rightarrow \varphi$ *has a standard proof in* $\mathbf{FNL}(\Phi)$.

Proof. This is proved by standard cut-elimination techniques, cf. [17, 8, 6, 15]. The proof is by double induction on the grade and size of cuts. We take a top most (closest to leaves) non-principal cut and transform it into another cut and the sum of grades of all cuts in the proof either decreases, or remains equal and then the sum of their sizes decreases. As this process is well-founded it suffices to show that we are able to transform every possible non-principal cut.

Let us assume we have a top most non-principal cut. If the cut formula is a side formula in any of two input sequents then we can apply the rule for a sequent where the cut formula is a side formula after the cut. Therefore we obtain a cut with the same grade but smaller size.

If both the main formulae in the input sequents are the same as the cut formula then both were obtained by rules for the same connective. Let us assume that it was e.g. \. Then we can transform

$$(\mathrm{Cut})\,\dfrac{(\backslash\mathrm{L})\,\dfrac{\overset{\vdots}{\mathbb{T}[\xi] \Rightarrow \psi} \qquad \overset{\vdots}{\mathbb{U} \Rightarrow \chi}}{\mathbb{T}[(\mathbb{U}, \chi\backslash\xi)] \Rightarrow \psi} \qquad (\backslash\mathrm{R})\,\dfrac{\overset{\vdots}{(\chi, \mathbb{V}) \Rightarrow \xi}}{\mathbb{V} \Rightarrow \chi\backslash\xi}}{\mathbb{T}[(\mathbb{U}, \mathbb{V})] \Rightarrow \psi}$$

into

$$(\mathrm{Cut})\,\dfrac{(\mathrm{Cut})\,\dfrac{\overset{\vdots}{\mathbb{T}[\xi] \Rightarrow \psi} \qquad \overset{\vdots}{(\chi, \mathbb{V}) \Rightarrow \xi}}{\mathbb{T}[(\chi, \mathbb{V})] \Rightarrow \psi} \qquad \overset{\vdots}{\mathbb{U} \Rightarrow \chi}}{\mathbb{T}[(\mathbb{U}, \mathbb{V})] \Rightarrow \psi}$$

and we obtain two cuts with the sum of grades smaller then in the original cut. Similarly for other connectives.

As we have a set Φ we must handle the cases where a cut is applied on other principal cut(s). In such a case we can always transform

$$(\mathrm{Cut})\,\dfrac{\overset{\vdots}{\mathbb{T}[\xi] \Rightarrow \psi} \qquad (\mathrm{Cut})\,\dfrac{\overset{\vdots}{\mathbb{U}[\chi] \Rightarrow \xi} \qquad \mathbb{V} \Rightarrow \chi}{\mathbb{U}[\mathbb{V}] \Rightarrow \xi}}{\mathbb{T}[\mathbb{U}[\mathbb{V}]] \Rightarrow \psi}$$

into

$$(\mathrm{Cut})\,\dfrac{(\mathrm{Cut})\,\dfrac{\overset{\vdots}{\mathbb{T}[\xi] \Rightarrow \psi} \qquad \overset{\vdots}{\mathbb{U}[\chi] \Rightarrow \xi}}{\mathbb{T}[\mathbb{U}[\chi]] \Rightarrow \psi} \qquad \mathbb{V} \Rightarrow \chi}{\mathbb{T}[\mathbb{U}[\mathbb{V}]] \Rightarrow \psi}$$

where we can assume $\mathbb{V} \Rightarrow \chi$ is a member of Φ. Therefore we obtain cuts with smaller size and zero size while the sum of grades remains the same.

The final non-trivial case is if we want to use cut on $\mathbb{T}[\xi] \Rightarrow \psi$, which is a member of Φ, and $\mathbb{U}[\chi] \Rightarrow \xi$. Then ξ is an atom, e.g. $\xi = p$, because Φ is regular. Hence p cannot be the main formula in $\mathbb{U}[\chi] \Rightarrow p$. Therefore the technique for side formulae or the above mentioned cut transformation can be used on it. $\square$

2.2 Non-associative Lambek Calculus with only product and join

The variant of cut-elimination for **FNL**(Φ) proved in Theorem 2.2 has many consequences. Let Φ be a regular set of non-logical axioms and $\mathbb{S} \Rightarrow \varphi$ a sequent such that they contain only connectives from $\{\cdot, \vee\}$. Then $\mathbb{S} \Rightarrow \varphi$ is provable in **FNL**(Φ) iff it is provable using only logical axioms (Id), non-logical axioms from Φ, ($\cdot$L), ($\cdot$R), ($\vee$L), ($\vee$R), and (Cut). We will denote such a calculus $\mathbf{NL}^{\vee}(\Phi)$ to emphasize that the full language of **FNL**(Φ) is not needed. Moreover, $\mathbb{S} \Rightarrow \varphi$ has a proof in $\mathbf{NL}^{\vee}(\Phi)$ iff it has a standard proof in $\mathbf{NL}^{\vee}(\Phi)$.

The choice of $\{\cdot, \vee\}$ is not arbitrary. Our Φ, which will demonstrate the undecidability of the consequence relation, will be in this particular language. Therefore we restrict ourselves to this language from now on. We can also define a natural equivalence relation and normal forms on formulae in this language. Let the following equivalences hold for any formulae φ, ψ, and χ.

$$\varphi \sim \varphi \vee \varphi \qquad \text{(Idempotency)}$$
$$\varphi \vee \psi \sim \psi \vee \varphi \qquad \text{(Commutativity)}$$
$$\varphi \vee (\psi \vee \chi) \sim (\varphi \vee \psi) \vee \chi \qquad \text{(Associativity)}$$
$$\varphi \cdot (\psi \vee \chi) \sim (\varphi \cdot \psi) \vee (\varphi \cdot \chi) \qquad \text{(Left-Distributivity)}$$
$$(\varphi \vee \psi) \cdot \chi \sim (\varphi \cdot \chi) \vee (\psi \cdot \chi) \qquad \text{(Right-Distributivity)}$$

Let $\sim^*$ be the reflexive, symmetric, and transitive closure of $\sim$. If we read $\sim$ in the previous equivalences as $\Rightarrow$ and $\Leftarrow$ it is easy to verify that the corresponding sequents are provable in $\mathbf{NL}^{\vee}(\emptyset)$. Hence it is clear that if we prove $\mathbb{S}[\varphi] \Rightarrow \psi$ and $\varphi \sim^* \chi$ then we can also prove $\mathbb{S}[\chi] \Rightarrow \psi$ using (Cut).

Definition 2.3. A formula φ is *simple* if the only connective occurring in φ is product ($\cdot$).

We define a variant of normal forms for formulae in $\{\cdot, \vee\}$, because any such a formula can be equivalently expressed as a join of simple formulae. It is sufficient to apply left- and right-distributivity from left to right as many times as needed. Moreover, this representation can be considered unique (assuming the idempotency, commutativity, and associativity of join) if we impose an order, e.g. lexicographic, on simple formulae. However, the actual order is not important for us. Therefore we assume that it is fixed in our paper.

Definition 2.4. A formula ψ is a *simple representation* of a formula φ if $\psi = \bigvee_{i=1}^{n} \chi_i$, all χ_i are simple, and $\varphi \sim^* \psi$. We write $\psi = [\varphi]_s$ and also use notation that $\chi_i \in [\varphi]_s$ if this representation is unique, i.e. n is the minimal possible and simple formulae are ordered.

In order to simplify the notation we will use this representation of formulae. Moreover, we omit products and most parentheses in formulae as we implicitly assume that product is the default connective and parentheses tight to right, e.g. pqr is strictly speaking $(p \cdot (q \cdot r))$.

2.3 Tag systems

Tag systems were proposed by Post [16] and they operate on finite words, i.e. finite sequences of letters. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be a finite alphabet of letters with no special halting letter. A set of production rules is given by a function $\pi\colon [1,n] \to \mathcal{A}^*$, where $\mathcal{A}^*$ is the set of finite words over $\mathcal{A}$. The computation of the 2-tag system given by $\mathcal{A}$ and π on a word $w \in \mathcal{A}^*$ is defined as follows.

If $|w| < 2$ we terminate. Otherwise, we examine the first letter in w, which is some a_i. Then we delete the first two letters in w and append $\pi(i)$ to the rest of the word after the last letter and obtain a new word.

We repeat this process until it is possible, i.e. we can run forever or at some point we terminate—we obtain a word with less than two letters. We use $\downarrow$ for such a word. In the later case we say that the 2-tag system terminates on w and write $w \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$. It is well-known, see [5], that the halting problem for a 2-tag system, i.e. whether it terminates on a given w, is generally undecidable.

Example 2.2. Let $\mathcal{A} = \{a_1, a_2\}$, $\pi(1) = a_2$, and $\pi(2) = a_1$ describe a 2-tag system. Then our very elementary system, starting on $w = a_1a_2a_2$, computes as follows:

$$\begin{aligned} w &= \underline{a_1}a_2a_2 \\ & \cancel{a_1a_2}\underline{a_2}a_2 \\ & \cancel{a_1a_2a_2a_2}a_1 = \downarrow \end{aligned}$$

3 Encoding

In this section we present how to encode a computation of a 2-tag system in the language $\{\cdot, \vee\}$. We have an arbitrary but fixed 2-tag system given by $\mathcal{A} = \{a_1, \dots, a_n\}$ and $\pi\colon [1,n] \to \mathcal{A}^*$. Assume that $w \in \mathcal{A}^*$ is a word. We recall our simple representation of formulae—a join of formulae containing only products. For simplicity we also do not write products and most parentheses in formulae as we implicitly assume that product is the default connective and parentheses tight to right.

The encoding is based on a join of simple formulae which read from the left side contain some prefix, letters from $\mathcal{A}$, and end with a symbol (a capital letter) describing their meaning.

Moreover, to simplify the formulation of the encoding the letters from $\mathcal{A}$ are represented in *pairs* starting from the left side. It is convenient as 2-tag systems delete pairs of letters. Hence a pair of letters a_ia_j is represented by c_i^j. When the length of a word is odd then its last letter a_i is simply represented by c_i.

Definition 3.1. Let $\mathcal{A} = \{a_1, \dots, a_n\}$ and $\pi\colon [1,n] \to \mathcal{A}^*$ describe a 2-tag system. The set $\mathcal{C}(\mathcal{A})$ of finite words in the *pair notation* over $\mathcal{A}$ is given by $\mathcal{C}(\mathcal{A}) = \mathcal{C}(\mathcal{A})_p^* \cup \mathcal{C}(\mathcal{A})_p^* \times \mathcal{C}(\mathcal{A})_s$, where $\mathcal{C}(\mathcal{A})_p = \{\, c_i^j \mid 1 \le i,j \le n \,\}$ and $\mathcal{C}(\mathcal{A})_s = \{c_1, \dots, c_n\}$. The translation function $\delta\colon \mathcal{A}^* \to \mathcal{C}(\mathcal{A})$ is defined by

$$\delta(w) = \begin{cases} \epsilon & \text{if } w \text{ is the empty word,} \\ c_i & \text{if } w = a_i, \\ c_i^j\,\delta(v) & \text{if } w = a_ia_jv, \end{cases}$$

where ϵ represents the empty string. We will also use the reverse function $\delta^{-1}\colon \mathcal{C}(\mathcal{A}) \to \mathcal{A}^*$.

Example 3.1. It holds that $\delta(a_1a_2a_1a_2) = c_1^2c_1^2$ and $\delta(a_1a_2a_1) = c_1^2c_1$.

Although using our conventions on notation simple formulae may look like sequences, they are trees. Hence the cut rule only enables us to substitute a tree for a subtree in them. As in our representation parentheses tight to right it is easy to process the end of a formula, because it is a subtree. However, when we append or delete letters we have to transfer pieces of information between the beginning and end of words (formulae). To get around this problem, we will use join.

First, we present some basic ideas in Section 3.1. Second, the needed non-logical axioms are presented in Section 3.2. However, all the properties of the encoding will be apparent from Sections 4 and 5, where the correctness and completeness of this encoding are proved.

3.1 The basic ideas behind

As the encoding, given by non-logical axioms, is relatively complex we present the behavior of our encoding first. The rules should be then easier to understand.

3.1.1 State formulae

First, we describe how a state of a 2-tag system is represented by a formula in our encoding. Later on we will call these formulae state formulae. For this purposes the alphabet contains the following symbols:

- $c_1^1, \ldots, c_n^n$ represent the pairs of letters,
- $c_1, \ldots, c_n$ represent the letters,
- e and e' represent the deleted pairs of letters,
- X and X' represent the end of words.

A state of a computation of a 2-tag system is fully described by the word it processes. We check whether its length is at least two, delete the first two letters, and append letters according to π and the first deleted letter. In our encoding we switch deleting and appending. However, it does not matter, because we will append letters only if the length of the processed word is at least two.

We start by representing the word w as $e\,\delta(w)X$, where the only connective is product and all parentheses tight to right.

Let us look at the 2-tag system from Example 2.2, i.e. $w = a_1a_2a_2$, $\pi(1) = a_2$, and $\pi(2) = a_1$. Then the initial formula is $ec_1^2c_2X$. The computation continues in such a way that we add $\pi(1)$ and change X to X'. Hence we obtain $ec_1^2c_2^2X'$, where c_2^2 represents a_2a_2, see our pair notation.

The change of the primality of X (and other symbols) will enable us to recognize whether we only added letters or also deleted the corresponding pair of letters. This subtle detail will be essential to the completeness proof. It will ensure that the appending and deleting steps must alternate. Therefore there will be a straightforward translation between state formulae and words occurring in a computation of a 2-tag system.

In $ec_1^2c_2^2X'$ we delete c_1^2, which represents a_1a_2, by changing it to e'. Hence we have $ee'c_2^2X'$. This represents the next state of the computation as both X and the last e are primed. The computation can continue to $ee'c_2^2c_1X$ and by changing c_2^2 to e we get $ee'ec_1X$. However, then the computation cannot proceed as only one letter remains.

From the previous example we can see that during the computation we add letters to the end and change letters at the beginning to e or e'. Therefore correct formulae start with alternating e and e'. If X and the last e are both primed or both not primed then such a formula directly represents a state of a 2-tag system (a word). Otherwise, the letters were added according to π, but the corresponding first two letters, represented by c_i^j, had not been deleted.

If there are less than two letters from $\mathcal{A}$ and the closest e and X are both primed or both not primed we are in a termination state. As we want an unique representation of termination states we first delete the only letter (if any) from $\mathcal{A}$ and then also e and e' one by one. Therefore all the termination states can be represented by the formula eX.

In our example, we get the following sequence $ee'ec_1X$, $ee'eX$, $ee'X'$, and finally we get eX.

3.1.2 Auxiliary formulae

Nevertheless, it is clear that we must check whether we appended and deleted (changed to e or e') the right letters. Moreover, we must be very particular about the correct alternation of appending and deleting steps. Therefore we have to use some auxiliary formulae, which ensure that the system is simulated faithfully. Hence whenever we execute any such a step we obtain a join of a state formula and an auxiliary formula. This auxiliary formula certifies that the step was used correctly. For this purposes some more symbols are needed:

- A represents the end of general auxiliary formulae,
- C_i and C'_i, for $1 \leq i \leq n$, represent the end of auxiliary formulae for adding letters,
- D and D' represent the end of auxiliary formulae for deleting letters,
- d and d' represent the position where a pair of letters was deleted.

The idea is that whenever we obtain an auxiliary formula we must transform it into the general auxiliary formula eA. This ensures that the step was used correctly.

Let us again assume we have $ec_1^2c_2X$. We want to append $\pi(1)$. However, this is possible only if the word represented by this formula begins with a_1 and contains at least one more letter. Hence some c_1^j has to be after the last e. We will be able to prove

$$ec_1^2c_2X \Rightarrow ec_1^2(c_2^2X' \vee c_2C_1),$$

from our non-logical axioms and $ec_1^2(c_2^2X' \vee c_2C_1) \sim^* ec_1^2c_2^2X' \vee ec_1^2c_2C_1$. Here C_1 in $ec_1^2c_2C_1$ says that we appended $\pi(1)$ and hence we want to check whether the word represented by $ec_1^2c_2C_1$ starts with some c_1^j. We can show that by deleting all the letters but the first pair. Therefore from $ec_1^2c_2C_1$ we obtain $ec_1^2C_1$. Then we get eA, because C_1 matches the first letter in c_1^2, which represents a_1a_2, and the primality of C_1 and e is the same. Moreover, c_1^2 certifies that the original formula contained at least two letters from $\mathcal{A}$ before $\pi(1)$ was appended. It works similarly for C'_i, e', and X'.

However, we cannot continue with another appending step, because the primality of X' is different from the primality of e in $ec_1^2c_2^2X'$. Hence we would not be able to transform the auxiliary formula into eA. Therefore we now have to delete the first two letters, c_1^2 in our case. We will be able to prove

$$ec_1^2c_2^2X' \Rightarrow e(e' \vee d')c_2^2X'$$

and $e(e' \vee d')c_2^2X' \sim^* ee'c_2^2X' \vee ed'c_2^2X'$. Here $ed'c_2^2X'$ is not a state formula, because it contains d'. Hence it is an auxiliary formula and we want to obtain eA from it. First, we get $ed'c_2^2D'$. Here D' says that we deleted the first two letters, which are represented by d'. The procedure is then similar to the previous case, we delete all the letters between d' and D'. From $ed'c_2^2D'$ we obtain $ed'D'$ and then eA. This last step is possible for the following two reasons. First, D' and d' are primed and e is not primed. Second, there is no further symbol between e and d'. It works similarly for X, but we use D, d and get e in the state formula.

Note that we could now continue by appending $\pi(2)$, because e' and X' in $ee'c_2^2X'$ are both primed, but not by deleting c_2^2, because we would not be able to transform the obtained auxiliary formula into eA.

3.2 Actual representation

From the previous text it should be clear that the question whether a 2-tag system given by $\mathcal{A}$ and π terminates on a $w \in \mathcal{A}^*$ will be translated into the question whether

$$e\,\delta(w)X \Rightarrow eX \vee eA$$

is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$, where $\Phi[\mathcal{A},\pi]$ is a finite set of non-logical axioms given in the following definition.

Definition 3.2. Let $\mathcal{A}=\{a_1,\dots,a_n\}$ and $\pi\colon[1,n]\to\mathcal{A}^*$ describe a 2-tag system. The set of non-logical axioms $\Phi[\mathcal{A},\pi]$ contains

$$e'eX \Rightarrow e'X' \quad (1) \qquad ee'X' \Rightarrow eX \quad (4)$$
$$ec_iX \Rightarrow eX \quad (2) \qquad e'c_iX' \Rightarrow e'X' \quad (5)$$
$$ee'A \Rightarrow eA \quad (3) \qquad e'eA \Rightarrow e'A \quad (6)$$

$$ec_i^jC_i \Rightarrow eA \quad (7) \qquad e'c_i^jC_i' \Rightarrow e'A \quad (12)$$
$$c_j^kc_lC_i \Rightarrow c_j^kC_i \quad (8) \qquad c_j^kc_lC_i' \Rightarrow c_j^kC_i' \quad (13)$$
$$c_j^kc_l^mC_i \Rightarrow c_j^kC_i \quad (9) \qquad c_j^kc_l^mC_i' \Rightarrow c_j^kC_i' \quad (14)$$
$$c_jX \Rightarrow \delta(a_j\,\pi(i))X' \vee c_jC_i \quad (10) \qquad c_jX' \Rightarrow \delta(a_j\,\pi(i))X \vee c_jC_i' \quad (15)$$
$$c_j^kX \Rightarrow c_j^k\,\delta(\pi(i))X' \vee c_j^kC_i \quad (11) \qquad c_j^kX' \Rightarrow c_j^k\,\delta(\pi(i))X \vee c_j^kC_i' \quad (16)$$

$$ed'D' \Rightarrow eA \quad (17) \qquad e'dD \Rightarrow e'A \quad (24)$$
$$d'c_iD' \Rightarrow d'D' \quad (18) \qquad dc_iD \Rightarrow dD \quad (25)$$
$$d'c_i^jD' \Rightarrow d'D' \quad (19) \qquad dc_i^jD \Rightarrow dD \quad (26)$$
$$c_i^jc_kD' \Rightarrow c_i^jD' \quad (20) \qquad c_i^jc_kD \Rightarrow c_i^jD \quad (27)$$
$$c_i^jc_k^lD' \Rightarrow c_i^jD' \quad (21) \qquad c_i^jc_k^lD \Rightarrow c_i^jD \quad (28)$$
$$X' \Rightarrow D' \quad (22) \qquad X \Rightarrow D \quad (29)$$
$$c_i^j \Rightarrow e' \vee d' \quad (23) \qquad c_i^j \Rightarrow e \vee d \quad (30)$$

for every $i,j,k,l,m\in[1,n]$.

Remark. The set $\Phi[\mathcal{A},\pi]$ is not regular, but we can replace all the products on the left side of sequents by commas and obtain regular sequents, i.e. we use σ. Lemma 2.1 says that this does not change provability. Therefore we can always assume that $\Phi[\mathcal{A},\pi]$ is regular.

The intuitive and simplified meaning of these non-logical axioms is following. By (1,4) and (3,6) we say that e and e' have to alternate. The axioms (1,4) and (2,5) say that a terminating word can contain no or one letter from $\mathcal{A}$ and that X and the last e are simultaneously primed or not primed.

It is important to notice that some rules have two variants, because we use the pair notation and words can have odd or even length.

The axioms (7–9) (or 12–14) describe how formulae containing C_i (or C_i') have to look like, i.e. c_i^j is right after e (or e'). Likewise, the axioms (17–21) (or 24–28) describe formulae containing D' (or D), i.e. d' (or d) is right after e (or e').

The meaning of (10,11) (or 15,16) is more complicated. If we apply them to a simple formula we obtain a join of two simple formulae, using the simple representation. The first of them contains appended $\pi(i)$ and changed primality of X. The second formula contains C_i (or C_i') and we want to obtain eA from this formula. This is possible if c_i^j is right after e (or e') and it certifies that the rule was used correctly.

If the axioms (23) (or 30) are applied to a simple formula we again obtain a join of two simple formulae, using the simple representation. The first one contains e' (or e) instead of c_i^j. The second one contains d' (or d) instead of it. We want to eliminate, obtain eA, the second formula.

The axioms (22) (or 29) allow us to rewrite such a formula to one ending with D' (or D). We can eliminate it, obtain eA, if d' (or d) is right after e (or e').

4 Correctness of encoding

We presented the set of non-logical axioms $\Phi[\mathcal{A}, \pi]$ for a given 2-tag system described by $\mathcal{A} = \{a_1, \dots, a_n\}$ and $\pi\colon [1, n] \to \mathcal{A}^*$. From now on, this 2-tag system will be fixed.

In this section we show that the representation is correct, i.e. what is computed by the 2-tag system can be also proved using $\Phi[\mathcal{A}, \pi]$ and our encoding. As we already indicated we divide formulae into two groups: formulae representing a state of the 2-tag system and auxiliary formulae.

In the following text we use $\langle ee'\rangle^m$ to describe repeating e and e'. Let $\langle ee'\rangle^0$ be the empty sequence and $\langle ee'\rangle^{m+1} = ee'\langle ee'\rangle^m$. Moreover, to simplify our presentation we handle simple formulae as sequences. For our purposes here, we can do it, but we have to be aware of their real representation, i.e. they are trees containing products and parentheses which tight to right. Therefore if we assume that $evX = ec_1^2c_2X$ then the subsequence $v = c_1^2c_2$ is not here even a (sub)formula, because $ec_1^2c_2X$ is strictly speaking $(e \cdot (c_1^2 \cdot (c_2 \cdot X)))$. However, we can say that $v \in \mathcal{C}(\mathcal{A})$ and hence even $\delta^{-1}(v) \in \mathcal{A}^*$, because it represents the word $a_1a_2a_2$.

Definition 4.1. A formula φ is an *auxiliary formula* in $\mathcal{A}$ iff it is one of the following

1. $\langle ee'\rangle^m eA$,
2. $\langle ee'\rangle^m ec_i^j vC_i$,
3. $\langle ee'\rangle^m ed'vD'$,
4. $\langle ee'\rangle^m ed'vX'$,
5. $\langle ee'\rangle^m ee'A$,
6. $\langle ee'\rangle^m ee'c_i^j vC_i'$,
7. $\langle ee'\rangle^m ee'dvD$,
8. $\langle ee'\rangle^m ee'dvX$,

for $v \in \mathcal{C}(\mathcal{A})$, $m \geq 0$, and $1 \leq i, j \leq n$. We define the set of all auxiliary formulae in $\mathcal{A}$ by $\Gamma_\mathcal{A} = \{\, \psi \mid \psi \text{ is an auxiliary formula in } \mathcal{A} \,\}$.

In the following lemmata we simplify proofs using Lemma 2.1. If we want to use (Cut) with the cut formula φ, then there has to be a structure $\mathbb{S}[\varphi]$. However, sometimes we want to "substitute" for a subformula φ in a formula χ. This is possible if $\chi = \rho(\mathbb{S}[\varphi])$, because $\mathbb{S}[\varphi] \Rightarrow \psi$ is provable iff $\chi \Rightarrow \psi$ is provable by Lemma 2.1. Then using (Cut) on $\mathbb{S}[\varphi] \Rightarrow \psi$ and $\mathbb{T} \Rightarrow \varphi$ we obtain $\mathbb{S}[\mathbb{T}] \Rightarrow \psi$. Finally, using again Lemma 2.1 we get $\rho(\mathbb{S}[\mathbb{T}]) \Rightarrow \psi$. We denote such an application of (Cut) by (Cut*). Similarly, we can use ($\vee$L) and denote it ($\vee$L*).

Lemma 4.1. *If $\varphi \in \Gamma_\mathcal{A}$ then $\varphi \Rightarrow eA$ is provable in* $\mathbf{NL}^\vee(\Phi[\mathcal{A}, \pi])$.

Proof. We check all the cases. If $\varphi = \langle ee'\rangle^m eA$, for $m > 0$, then we use (6) and (3) from $\Phi[\mathcal{A}, \pi]$

$$\cfrac{\cfrac{ee'A \Rightarrow eA \qquad e'eA \Rightarrow e'A}{\langle ee'\rangle^1 eA \Rightarrow eA}\,(\mathrm{Cut}^\star) \qquad ee'A \Rightarrow eA}{\langle ee'\rangle^2 A \Rightarrow eA}\,(\mathrm{Cut}^\star)$$

$$\vdots$$

$$\cfrac{\langle ee'\rangle^m A \Rightarrow eA \qquad e'eA \Rightarrow e'A}{\langle ee'\rangle^m eA \Rightarrow eA}\,(\mathrm{Cut}^\star)$$

We get $\langle ee'\rangle^m ee'A \Rightarrow eA$ by one more application of (3).

If $\varphi = \langle ee'\rangle^m ec_i^j vC_i$ we first obtain

$$(\mathrm{Cut}^\star)\ \dfrac{\begin{matrix}\vdots & \\ \langle ee'\rangle^m eA \Rightarrow eA & ec_i^jC_i \Rightarrow eA\end{matrix}}{\langle ee'\rangle^m ec_i^jC_i \Rightarrow eA}$$

using (7) and then use (9) as many times as needed, i.e. $\lfloor|\,\delta^{\text{-}1}(v)|/2\rfloor$-times. If $\delta^{\text{-}1}(v)$ has odd length then we have to use also (8). The situation with $\langle ee'\rangle^m ee'c_i^jvC_i'$ is completely analogous.

If we want to prove $\langle ee'\rangle^m ed'vD' \Rightarrow eA$ (or $\langle ee'\rangle^m ee'dvD \Rightarrow eA$) we start from $\langle ee'\rangle^m eA \Rightarrow eA$ (or $\langle ee'\rangle^m ee'A \Rightarrow eA$) and use (17–21) (or 24–28) in a very similar fashion. Finally, using (22) (or 29) on $\langle ee'\rangle^m ed'vD' \Rightarrow eA$ (or $\langle ee'\rangle^m ee'dvD \Rightarrow eA$) we immediately get $\langle ee'\rangle^m ed'vX' \Rightarrow eA$ (or $\langle ee'\rangle^m ee'dvX \Rightarrow eA$). □

Corollary 4.2. *If $\varphi \in \Gamma_{\mathcal{A}}$ then $\varphi \Rightarrow eX \vee eA$ is provable in $\mathbf{NL}^\vee(\Phi[\mathcal{A},\pi])$.*

Definition 4.2. A formula φ is a *state formula* in $\mathcal{A}$ iff it is one of the following

1. $\langle ee'\rangle^m evX$,
2. $\langle ee'\rangle^m ee'c_i^jvX$,
3. $\langle ee'\rangle^m ee'vX'$,
4. $\langle ee'\rangle^m ec_i^jvX'$,

for $v \in \mathcal{C}(\mathcal{A})$, $m \geq 0$, and $1 \leq i,j \leq n$. We define the set of all state formulae in $\mathcal{A}$ by $\Lambda_{\mathcal{A}} = \{\,\psi \mid \psi \text{ is a state formula in } \mathcal{A}\,\}$.

The function τ which translates a state formula φ into a word in $\mathcal{A}^*$ is defined by

$$\tau(\varphi) = \delta^{\text{-}1}(v).$$

Example 4.1. It holds that $\tau(ec_1^2c_2X) = a_1a_2a_2$, $\tau(ec_1^2c_2^2X') = a_2a_2$, and $\tau(ee'c_1X') = a_1 = \downarrow$.

Lemma 4.3. *If $w \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$ then $\langle ee'\rangle^m e\,\delta(w)X \Rightarrow eX \vee eA$ and $\langle ee'\rangle^m ee'\,\delta(w)X' \Rightarrow eX \vee eA$ are provable in $\mathbf{NL}^\vee(\Phi[\mathcal{A},\pi])$ for any $m \geq 0$.*

Proof. We prove this by induction on the length of the 2-tag derivation. If w is in a termination state it means w is the empty sequence or $w = a_i$, for some $i \in [1,n]$. We use (4) and (1) m-times to obtain $\langle ee'\rangle^m eX \Rightarrow eX$. Using (2) we get $\langle ee'\rangle^m ec_iX \Rightarrow eX$, for $1 \leq i \leq n$. One more application of (4) on $\langle ee'\rangle^m eX \Rightarrow eX$ leads to $\langle ee'\rangle^m ee'X' \Rightarrow eX$. Then using (5) we get $\langle ee'\rangle^m ee'c_iX' \Rightarrow eX$, for $1 \leq i \leq n$. We complete all these cases by ($\vee$R).

Let us assume that one step computation of our 2-tag system leads from w to v and $\langle ee'\rangle^m ee'\,\delta(v)X' \Rightarrow eX \vee eA$ and $\langle ee'\rangle^m ee'e\,\delta(v)X \Rightarrow eX \vee eA$ are provable in $\mathbf{NL}^\vee(\Phi[\mathcal{A},\pi])$. It means that there are indexes i and j such that $w = a_ia_ju$ and $v = u\,\pi(i)$. Using Corollary 4.2 and then (23) we obtain

$$(\mathrm{Cut}^\star)\ \dfrac{(\vee\mathrm{L}^\star)\ \dfrac{\begin{matrix}\vdots & \vdots \\ \langle ee'\rangle^m ee'\,\delta(u\,\pi(i))X' \Rightarrow eX \vee eA & \langle ee'\rangle^m ed'\,\delta(u\,\pi(i))X' \Rightarrow eX \vee eA\end{matrix}}{\langle ee'\rangle^m e(e' \vee d')\,\delta(u\,\pi(i))X' \Rightarrow eX \vee eA} \qquad c_i^j \Rightarrow e' \vee d'}{\langle ee'\rangle^m ec_i^j\,\delta(u\,\pi(i))X' \Rightarrow eX \vee eA}$$

If u has odd length, $u = ta_k$, then $\delta(ta_k\,\pi(i)) = \delta(t)\,\delta(a_k\,\pi(i))$, and using Corollary 4.2 and then (10) we get

$$(\mathrm{Cut}^\star)\ \dfrac{(\vee\mathrm{L}^\star)\ \dfrac{\begin{matrix}\vdots & \vdots \\ \langle ee'\rangle^m ec_i^j\,\delta(t)\,\delta(a_k\,\pi(i))X' \Rightarrow eX \vee eA & \langle ee'\rangle^m ec_i^j\,\delta(t)c_kC_i \Rightarrow eX \vee eA\end{matrix}}{\langle ee'\rangle^m ec_i^j\,\delta(t)(\delta(a_k\,\pi(i))X' \vee c_kC_i) \Rightarrow eX \vee eA} \qquad c_kX \Rightarrow \delta(a_k\,\pi(i))X' \vee c_kC_i}{\langle ee'\rangle^m ec_i^j\,\delta(t)c_kX \Rightarrow eX \vee eA}$$

where $c_i^j\,\delta(t)c_k = \delta(a_ia_ju) = \delta(w)$. If u has even length we use (11).

The proof of $\langle ee'\rangle^m ee'c_i^j\,\delta(u)X' \Rightarrow eX \vee eA$ using the provability of $\langle ee'\rangle^m ee'e\,\delta(v)X \Rightarrow eX \vee eA$ is completely analogous. $\Box$

Corollary 4.4. *If $w \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$ then $e\,\delta(w)X \Rightarrow eX \vee eA$ is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$.*

5 Completeness of encoding

We have proved that our encoding can simulate any terminating computation of our fixed 2-tag system given by $\mathcal{A} = \{a_1,\ldots,a_n\}$ and $\pi\colon [1,n] \to \mathcal{A}^*$. It remains to prove the other direction—any proof of a sequent expressing that the system terminates can be translated into a terminating computation of the given 2-tag system.

Recall the definition of our simple representation, see Definition 2.4, the translation ρ, which replaces all commas in structures by products, and the reverse translation σ, see Definition 2.2.

As was already mentioned we can assume that all the members of $\Phi[\mathcal{A},\pi]$ are regular, i.e. any $\alpha \Rightarrow \beta$ from $\Phi[\mathcal{A},\pi]$ is treated as $\sigma(\alpha) \Rightarrow \beta$, see Lemma 2.1. We know from Theorem 2.2 that $\mathbb{S} \Rightarrow \varphi$ is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$ iff it has a standard proof in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$. Therefore the following three lemmata are proved by induction on the height of the proof, which is the length of the longest branch in its tree representation, with only principal cuts.

We want to prove that if $\mathbb{S} \Rightarrow eX \vee eA$ is provable then the simple representation of $\rho(\mathbb{S})$ contains only auxiliary formulae or state formulae that represent words with terminating computations.

The main reason, why we use the equivalence $\sim^*$ on formulae and simple representation, is the rule ($\vee$L). This rule enables us to join two structures, e.g. from $ec_1X \Rightarrow eX \vee eA$ and $ec_2X \Rightarrow eX \vee eA$ we can easily prove $e(c_1 \vee c_2)X \Rightarrow eX \vee eA$ and $(e \vee e)c_1X \Rightarrow eX \vee eA$. Our simple representation is a way how to handle similar obstacles.

Lemma 5.1. *If $\mathbb{S} \Rightarrow eA$ is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$ then for all $\psi \in [\rho(\mathbb{S})]_s$ hold $\psi \in \Gamma_{\mathcal{A}}$.*

Proof. The proof is by induction on the height of the standard proof. If the height is zero then the only possibilities are (Id) or members of $\Phi[\mathcal{A},\pi]$. Hence $[\rho(\mathbb{S})]_s$ can contain only eA, $ee'A$, $ed'D'$, or $ec_i^jC_i$, for any $i,j \in [1,n]$. All these formulae are elements of $\Gamma_{\mathcal{A}}$.

We assume that this lemma holds for all standard proofs with height at most n. Fix an arbitrary proof of $\mathbb{S} \Rightarrow eA$ with height $n+1$. We have to check all the possible last steps.

It is impossible that ($\vee$R) is the last step and if it is ($\cdot$L) then we get $\mathbb{S} \Rightarrow eA$ from some $\mathbb{S}' \Rightarrow eA$, where $\rho(\mathbb{S}) = \rho(\mathbb{S}')$.

If ($\vee$L) is the last step then it means that there are $\mathbb{W}$, χ, and ξ such that $\mathbb{S} = \mathbb{W}[\chi \vee \xi]$, $\mathbb{W}[\chi] \Rightarrow eA$, and $\mathbb{W}[\xi] \Rightarrow eA$. Hence $\psi \in [\rho(\mathbb{S})]_s$ iff $\psi \in [\rho(\mathbb{W}[\chi])]_s$ or $\psi \in [\rho(\mathbb{W}[\xi])]_s$.

If ($\cdot$R) is the last step then there exist $\mathbb{U}$ and $\mathbb{V}$ such that $\mathbb{S} = (\mathbb{U},\mathbb{V})$, $\mathbb{U} \Rightarrow e$, and $\mathbb{V} \Rightarrow A$. It is easy to prove by induction that $e = [\mathbb{U}]_s$ and $A = [\mathbb{V}]_s$. Hence $eA = [(\mathbb{U},\mathbb{V})]_s$.

The last case is when we use a principal cut. Let us assume that from $\mathbb{S}' \Rightarrow eA$ and a member of $\Phi[\mathcal{A},\pi]$ we obtain $\mathbb{S} \Rightarrow eA$.

If an axiom $\sigma(\alpha) \Rightarrow \beta$ with no join is used then it is sufficient to show for any $\rho(\mathbb{T}[\beta]) \in [\rho(\mathbb{S}')]_s$, i.e. $\rho(\mathbb{T}[\beta]) \in \Gamma_{\mathcal{A}}$, that $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$. We check all the possible axioms and assume $\mathbb{T}[\beta]$ is an arbitrary but fixed.

If we use one of (3), (6–9), (12–14), (17–21), or (24–28) then from $\mathbb{T}[\beta]$, where $\rho(\mathbb{T}[\beta]) \in \Gamma_{\mathcal{A}}$, we obtain $\mathbb{T}[\sigma(\alpha)]$ such that $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$.

If we use (22) or (29) then $\rho(\mathbb{T}[\beta])$, which is in $\Gamma_{\mathcal{A}}$, is equal to some $\langle ee'\rangle^m ed'vD'$ or $\langle ee'\rangle^m ee'dvD$. Therefore the formula $\rho(\mathbb{T}[\alpha])$ is also in $\Gamma_{\mathcal{A}}$.

The other axioms from $\Phi[\mathcal{A},\pi]$ cannot be applied, because the induction hypothesis would be violated as we will show in the rest of the proof.

If (1), (2), (4), or (5) were used it would mean that $\mathbb{T}[\beta]$ contains eX or $e'X'$. Hence $\rho(\mathbb{T}[\beta]) \notin \Gamma_{\mathcal{A}}$.

We have to also check axioms $\sigma(\alpha) \Rightarrow \beta \vee \gamma$ containing join. It suffices to show for any two formulae $\rho(\mathbb{T}[\beta]) \in [\rho(\mathbb{S}')]_s$ and $\rho(\mathbb{T}[\gamma]) \in [\rho(\mathbb{S}')]_s$, i.e. $\rho(\mathbb{T}[\beta]), \rho(\mathbb{T}[\gamma]) \in \Gamma_{\mathcal{A}}$, that $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$, because $(\rho(\mathbb{T}[\beta]) \vee \rho(\mathbb{T}[\gamma])) \sim^* \rho(\mathbb{T}[\beta \vee \gamma])$. Let $\mathbb{T}[\beta]$ and $\mathbb{T}[\gamma]$ be arbitrary but fixed.

If (10) or (11) were used then we could clearly assume $\rho(\mathbb{T}[\beta])$ ends with X' and $\rho(\mathbb{T}[\gamma])$ with C_i. Moreover, they are members of $\Gamma_{\mathcal{A}}$. Therefore $\rho(\mathbb{T}[\gamma])$ has to be equal to some $\langle ee'\rangle^m ec_i^j vC_i$ and hence $\rho(\mathbb{T}[\beta]) = \langle ee'\rangle^m ec_i^j\, \delta(\delta^{-1}(v)\, \pi(i))X'$, but then $\rho(\mathbb{T}[\beta]) \notin \Gamma_{\mathcal{A}}$. It follows that (10) and (11) are not applicable in this case. Similarly for (15) and (16).

If (23) were used then we could clearly assume $\rho(\mathbb{T}[\gamma])$ contains d'. It is impossible that $\rho(\mathbb{T}[\gamma]) = \langle ee'\rangle^m ed'vD'$ and $\rho(\mathbb{T}[\beta]) = \langle ee'\rangle^m ee'vD'$ as $\rho(\mathbb{T}[\beta]) \notin \Gamma_{\mathcal{A}}$. Therefore $\rho(\mathbb{T}[\gamma]) = \langle ee'\rangle^m ed'vX'$ and $\rho(\mathbb{T}[\beta]) = \langle ee'\rangle^m ee'vX'$, but then also $\rho(\mathbb{T}[\beta]) \notin \Gamma_{\mathcal{A}}$. Similarly for (30). □

Lemma 5.2. *If $\mathbb{S} \Rightarrow eX$ is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$ then for all $\psi \in [\rho(\mathbb{S})]_s$ hold $\psi \in \Lambda_{\mathcal{A}}$ and $\tau(\psi) = \downarrow$.*

Proof. The proof is by induction on the height of the standard proof. If the height is zero then the only possibilities are (Id) or members of $\Phi[\mathcal{A},\pi]$. Hence $[\rho(\mathbb{S})]_s$ can contain only eX, $ee'X'$, or ec_iX, for any $i \in [1,n]$. All these formulae are trivially equal to $\downarrow$ under τ.

We assume that this lemma holds for all standard proofs with height at most n. Fix an arbitrary proof of $\mathbb{S} \Rightarrow eA$ with height $n+1$. We need to consider all the possible last steps.

It is impossible that ($\vee$R) is the last step and if it is ($\cdot$L) or ($\vee$L) then we can use arguments from the previous lemma.

If ($\cdot$R) is the last step then $\mathbb{S} = (\mathbb{U},\mathbb{V})$, $\mathbb{U} \Rightarrow e$, and $\mathbb{V} \Rightarrow X$. It is easy to prove by induction that $e = [\mathbb{U}]_s$ and $X = [\mathbb{V}]_s$. Hence $eX = [(\mathbb{U},\mathbb{V})]_s$.

The last case is when we use a principal cut. Let us assume that from $\mathbb{S}' \Rightarrow eA$ and a member of $\Phi[\mathcal{A},\pi]$ we obtain $\mathbb{S} \Rightarrow eA$. It is clear that only the axioms (1), (2), (4), and (5) from $\Phi[\mathcal{A},\pi]$ are applicable. The other axioms contain on the right side a symbol, which does not occur in $\Lambda_{\mathcal{A}}$. This violates the induction hypothesis.

As an axiom $\sigma(\alpha) \Rightarrow \beta$ with no join is used, it is sufficient to show for any $\rho(\mathbb{T}[\beta]) \in [\rho(\mathbb{S}')]_s$, i.e. $\tau(\rho(\mathbb{T}[\beta])) = \downarrow$, that $\tau(\rho(\mathbb{T}[\alpha])) = \downarrow$. Fix $\mathbb{T}[\beta]$ and we know that the axiom (1), (2), (4), or (5) is used. Hence $\mathbb{T}[\beta]$ contains eX or $e'X'$. We conclude from $\rho(\mathbb{T}[\beta]) \in \Lambda_{\mathcal{A}}$ that $\tau(\rho(\mathbb{T}[\alpha])) = \downarrow$. □

Lemma 5.3. *If $\mathbb{S} \Rightarrow eX \vee eA$ is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$ then for all $\psi \in [\rho(\mathbb{S})]_s$ hold*

1. *$\psi \in \Gamma_{\mathcal{A}}$, or*
2. *$\psi \in \Lambda_{\mathcal{A}}$ and $\tau(\psi) \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$.*

Proof. The proof is by induction on the height of the standard proof. If the height is zero then the only possibility is (Id). Hence $[\rho(\mathbb{S})]_s$ contains eX and eA. We know that $\tau(eX) = \downarrow$ and $eA \in \Gamma_{\mathcal{A}}$. Note that τ is defined only for state formulae and therefore the application of τ on any formula implies that this formula is in $\Lambda_{\mathcal{A}}$.

We assume that this lemma holds for all standard proofs with height at most n. Fix an arbitrary proof of $\mathbb{S} \Rightarrow eX \vee eA$ with height $n+1$. We need to consider all the possible last steps.

It is impossible that ($\cdot$R) is the last step and if it is ($\cdot$L) or ($\vee$L) then we can use arguments from Lemma 5.1.

If ($\vee$R) is the last step then there are two possibilities. If we use ($\vee$R) on $\mathbb{S} \Rightarrow eA$ then for all $\psi \in [\rho(\mathbb{S})]_s$ hold $\psi \in \Gamma_{\mathcal{A}}$ by Lemma 5.1. And if we use ($\vee$R) on $\mathbb{S} \Rightarrow eX$ then for all $\psi \in [\rho(\mathbb{S})]_s$ hold $\tau(\psi) = \downarrow$ by Lemma 5.2.

The last case is when we use a principal cut. Let us assume that from $\mathbb{S}' \Rightarrow eX \vee eA$ and a member of $\Phi[\mathcal{A},\pi]$ we obtain $\mathbb{S} \Rightarrow eX \vee eA$.

If an axiom $\sigma(\alpha) \Rightarrow \beta$ with no join is used then it is sufficient to show for any $\rho(\mathbb{T}[\beta]) \in [\rho(\mathbb{S}')]_s$, i.e. $\rho(\mathbb{T}[\beta]) \in \Gamma_{\mathcal{A}} \cup \Lambda_{\mathcal{A}}$, that $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$ or $\tau(\rho(\mathbb{T}[\alpha])) \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$. We have to check all the possible axioms and assume $\mathbb{T}[\beta]$ is an arbitrary but fixed.

If we use the axiom (1), (2), (4), or (5) it means that $\mathbb{T}[\beta]$ contains eX or $e'X'$. Hence $\rho(\mathbb{T}[\beta]) \in \Lambda_{\mathcal{A}}$ and $\tau(\rho(\mathbb{T}[\alpha])) = \downarrow$.

If we use one of (3), (6–9), (12–14), (17–21), or (24–28) then $\rho(\mathbb{T}[\beta]) \in \Gamma_{\mathcal{A}}$. Applying any of these axioms we obtain $\mathbb{T}[\sigma(\alpha)]$ such that $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$.

If we use (22) or (29) then $\rho(\mathbb{T}[\beta]) \in \Gamma_{\mathcal{A}}$, because $\rho(\mathbb{T}[\beta])$ is equal to some $\langle ee'\rangle^m ed'vD'$ or $\langle ee'\rangle^m ee'dvD$ and hence $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$.

We have to also check axioms $\sigma(\alpha) \Rightarrow \beta \vee \gamma$ containing join. In this case two formulae $\rho(\mathbb{T}[\beta]) \in [\rho(\mathbb{S}')]_s$ and $\rho(\mathbb{T}[\gamma]) \in [\rho(\mathbb{S}')]_s$, i.e. $\rho(\mathbb{T}[\beta]), \rho(\mathbb{T}[\gamma]) \in \Gamma_{\mathcal{A}} \cup \Lambda_{\mathcal{A}}$, are sufficient as $(\rho(\mathbb{T}[\beta]) \vee \rho(\mathbb{T}[\gamma])) \sim^* \rho(\mathbb{T}[\beta \vee \gamma])$. We have to show that $\rho(\mathbb{T}[\alpha]) \in \Gamma_{\mathcal{A}}$ or $\tau(\rho(\mathbb{T}[\alpha])) \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$. Let $\mathbb{T}[\beta]$ and $\mathbb{T}[\gamma]$ be arbitrary but fixed.

If we use (10) or (11) then we can clearly assume $\rho(\mathbb{T}[\beta])$ ends with X' and $\rho(\mathbb{T}[\gamma])$ with C_i. Therefore $\rho(\mathbb{T}[\gamma])$ is equal to some $\langle ee'\rangle^m ec_i^j vC_i$ and hence $\rho(\mathbb{T}[\beta]) = \langle ee'\rangle^m ec_i^j\, \delta(\delta^{-1}(v)\, \pi(i))X'$. Hence $\rho(\mathbb{T}[\alpha]) = \langle ee'\rangle^m ec_i^j vX$ and $\tau(\rho(\mathbb{T}[\alpha])) = a_i a_j\, \delta^{-1}(v)$. We know that $\tau(\rho(\mathbb{T}[\beta])) = \delta^{-1}(v)\, \pi(i) \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$ and hence $\tau(\rho(\mathbb{T}[\alpha])) \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$, because we get $\delta^{-1}(v)\, \pi(i)$ from $a_i a_j\, \delta^{-1}(v)$ by a single step of the 2-tag system. Similarly for (15) and (16).

If we use (23) then we can clearly assume $\rho(\mathbb{T}[\gamma])$ contains d'. It is impossible that $\rho(\mathbb{T}[\gamma]) = \langle ee'\rangle^m ed'vD'$ and $\rho(\mathbb{T}[\beta]) = \langle ee'\rangle^m ee'vD'$, because $\rho(\mathbb{T}[\beta]) \notin \Gamma_{\mathcal{A}} \cup \Lambda_{\mathcal{A}}$. Therefore $\rho(\mathbb{T}[\gamma]) = \langle ee'\rangle^m ed'vX'$ and $\rho(\mathbb{T}[\beta]) = \langle ee'\rangle^m ee'vX'$. Hence $\rho(\mathbb{T}[\alpha]) = \langle ee'\rangle^m ec_i^j vX'$ and $\tau(\rho(\mathbb{T}[\alpha])) = \delta^{-1}(v) = \tau(\rho(\mathbb{T}[\beta])) \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$. Similarly for (30). □

Corollary 5.4. *If $e\,\delta(w)X \Rightarrow eX \vee eA$ is provable in $\mathbf{NL}^{\vee}(\Phi[\mathcal{A},\pi])$ then $w \rightsquigarrow^*_{\mathcal{A},\pi} \downarrow$.*

As the halting problem for 2-tag systems is generally undecidable Corollaries 4.4 and 5.4 give us that the consequence relation in $\mathbf{NL}^{\vee}$, and therefore in **FNL**, is undecidable.

6 Some possible modifications

In this section we discuss some easy modifications of our result. Although they could influence the validity of the completeness of encoding proved in the previous section, we will show that they have no or negligible effect on the proofs of Lemmata 5.1–5.3.

6.1 Adding structural rules

We can add structural rules to our non-associative calculus, cf. [9]. The rules of *exchange* (e), *contraction* (c), and *left-weakening* or *integrality* (i) are defined as:

$$\text{(e)}\ \frac{\mathbb{S}[(\mathbb{T},\mathbb{U})] \Rightarrow \psi}{\mathbb{S}[(\mathbb{U},\mathbb{T})] \Rightarrow \psi} \qquad \text{(c)}\ \frac{\mathbb{S}[(\mathbb{T},\mathbb{T})] \Rightarrow \psi}{\mathbb{S}[\mathbb{T}] \Rightarrow \psi} \qquad \text{(i)}\ \frac{\mathbb{S}[\varepsilon] \Rightarrow \psi}{\mathbb{S}[\mathbb{T}] \Rightarrow \psi}$$

It is obvious that our construction fails with the rule of left-weakening. In fact then the consequence relation is decidable, see [1].

However, our construction works with exchange and contraction. It is easy to check that Theorem 2.2 is provable when the rule of exchange is added. If the rule of contraction is added then the only non-trivial case is when

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \mathbb{T}[(\mathbb{U}[\chi],\mathbb{U}[\chi])]\Rightarrow\psi\end{matrix}}{\mathbb{T}[\mathbb{U}[\chi]]\Rightarrow\psi}\,(\mathrm{c}) \qquad \begin{matrix}\vdots\\ \mathbb{V}\Rightarrow\chi\end{matrix}}{\mathbb{T}[\mathbb{U}[\mathbb{V}]]\Rightarrow\psi}\,(\mathrm{Cut})$$

Nevertheless, such a proof can be transformed into

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\\ \mathbb{T}[(\mathbb{U}[\chi],\mathbb{U}[\chi])]\Rightarrow\psi\end{matrix} \qquad \begin{matrix}\vdots\\ \mathbb{V}\Rightarrow\chi\end{matrix}}{\mathbb{T}[(\mathbb{U}[\mathbb{V}],\mathbb{U}[\chi])]\Rightarrow\psi}\,(\mathrm{Cut}) \qquad \begin{matrix}\vdots\\ \mathbb{V}\Rightarrow\chi\end{matrix}}{\mathbb{T}[(\mathbb{U}[\mathbb{V}],\mathbb{U}[\mathbb{V}])]\Rightarrow\psi}\,(\mathrm{Cut})}{\mathbb{T}[\mathbb{U}[\mathbb{V}]]\Rightarrow\psi}\,(\mathrm{c})$$

Thus we can prove a variant of Theorem 2.2 assuming a more general definition of (Cut) that makes possible to perform these two cuts at once, see [12], and a small change in the inductive argument.[1]

Therefore it is sufficient to show that Corollary 5.4 holds even with exchange and contraction.

6.1.1 Exchange

As all our simple formulae can be represented as sequences of propositional variables, where brackets tight to right, it is a simple matter to show that our construction works when the rule of exchange is added, because in our case we can define a normal form. For example it is clear that $(e((Xe)e'))$ is equivalent to $ee'eX$. We define the function l by

$$\mathrm{l}(\varphi\cdot\psi)=\begin{cases}\varphi\cdot\mathrm{l}(\psi) & \text{if } \psi \text{ is not a propositional variable,}\\ \psi\cdot\mathrm{l}(\varphi) & \text{if } \varphi \text{ is not a propositional variable,}\\ \varphi\cdot\psi & \text{if } \psi \text{ is a propositional variable which is a capital letter,}\\ \psi\cdot\varphi & \text{if } \varphi \text{ is a propositional variable which is a capital letter,}\end{cases}$$

and $\mathrm{l}(p)=p$ for any propositional variable p. We can change the definition of simple representation, see Definition 2.4, not to be a join of simple formulae χ_i, but a join of simple formulae $\mathrm{l}(\chi_i)$. Then it is easy to check that Lemmata 5.1–5.3 hold with this adapted definition.

6.1.2 Contraction

The case when the rule of contraction is added is easy, because this rule is not applicable in the proofs of Lemmata 5.1–5.3. The reason is that no formula occurring in our construction has $\varphi\cdot\varphi$ as a subformula, cf. the definition of $\Phi[\mathcal{A},\pi]$, $\Gamma_{\mathcal{A}}$, and $\Lambda_{\mathcal{A}}$. Moreover, the rule of exchange has no effect on that.

6.2 One-variable fragment

It is easily seen that in our construction we can encode any finite sequence of variables by one variable using non-associativity. Let p be the only variable and we want to encode the sequence of variables $p_0,\ldots,p_m$. The variable p_i is uniquely determined by i and hence also by $2i+1$.[2]

[1] It is worth noting that we have the rule of contraction for structures and not only for formulae. The rule of contraction for formulae does not admit cut-elimination. However, as we have product in the language both rules are equivalent in the presence of (Cut), cf. [9].

[2] The reason why we use $2i+1$ instead of i is that this encoding works even with the rule of exchange, because 1 is always the last symbol in the binary representation.

We can represent p_i by the binary representation of $2i+1$, hence we use $k = \lceil \log_2(m+1) \rceil + 1$ bits, where we replace all zeros by p and ones by (pp) and all the parentheses in the resulting formula tight to right.

Example 6.1. If we have $p_0, \ldots, p_{15}$ then we need 5 bits. Therefore p_2 is represented by 00101 and hence by $(p(p((pp)(p(pp)))))$.

It is obvious that this encoding does not work with the rule of contraction. However, it is easy to obtain a function similar to l from Section 6.1.1 which works nicely with the rule of exchange.

7 Remarks on algebraic consequences

As **FNL** is complete with respect to lattice-ordered residuated groupoids, see e.g. [9], our paper proves that the word problem for them is undecidable. However, we can obtain a stronger result as we need only $\{\cdot, \vee\}$. Let $\mathcal{G}$ be a set with a groupoid operation (product) and join on it, where join is idempotent, commutative, associative, and product distributes over join, i.e. they satisfy equalities in Section 2.2. As $\langle \mathcal{G}, \vee \rangle$ is a join-semilattice we can define $x \leq y$ iff $y = x \vee y$. Our construction shows that the word problem for such a structure is generally undecidable. Therefore it does not have the finite embedability property (FEP).[3]

The translation is fairly straightforward. We know that $\mathbb{S} \Rightarrow \varphi$ is provable iff $\rho(\mathbb{S}) \Rightarrow \varphi$ is provable. We can translate that into $\rho(\mathbb{S}) \leq \varphi$, i.e. $\varphi = \rho(\mathbb{S}) \vee \varphi$. This way we can translate all the axioms from $\Phi[\mathcal{A}, \pi]$, see Definition 3.2, and use them as a theory. The 2-tag system given by $\mathcal{A}$ and π terminates on a word $w \in \mathcal{A}^*$ iff $eX \vee eA = e\,\delta(w)X \vee eX \vee eA$ is provable from this theory.

A direct proof that this works can be based on Sections 4 and 5. The correctness is proved as in Section 4. The proof of completeness by induction on the length of the derivation starting from $eX \vee eA$ is analogous to the proof in Section 5. Note that our simple representation is convenient for this purpose.

Let us also note that the structural rules of exchange and contraction given in Section 6.1 correspond to $x \cdot y = y \cdot x$ and $x \leq x \cdot x$, respectively. Therefore the word problem is undecidable even if these equalities hold.

On the other hand, it is easy to check that in the proof of correctness we do not need the idempotency and commutativity of join in the full generality. It suffices to add the following particular equality

$$eA \vee eX = eX \vee eA. \tag{31}$$

However, for our construction to work we need the associativity of join and the distributivity of product over join.

8 Remarks on term rewriting systems

A natural way how to think about our construction is that we produce a term rewriting system. For our purposes here, a term rewriting system is a collection of *rewriting rules* $\varphi \to \psi$, where φ and ψ are formulae. Such a rule says that in a formula we can replace any of its subformula φ by ψ.

[3]On the contrary, the FEP holds for distributive lattice-ordered residuated groupoids, see [4, 7, 11]. There is a quasi-identity in the language $\{\cdot, \vee\}$ that holds in the distributive case but does not hold in the non-distributive one. A join-semilattice L is *distributive* if $a \leq a_1 \vee a_2$, for $a, a_1, a_2 \in L$, implies $a = a_1' \vee a_2'$, for some $a_1', a_2' \in L$, where $a_1' \leq a_1$ and $a_2' \leq a_2$, see [10]. A particular example of such a quasi-identity is $x \leq x_1 \vee x_2$ and $y \leq y_1 \vee y_2$ implies $xy \leq xy_1 \vee x_1y_2 \vee x_2y$, which was kindly provided by Rostislav Horčík.

It is clear that we can read $\varphi \Rightarrow \psi$ as $\varphi \to \psi$. Therefore our rewriting system is given by $\Phi[\mathcal{A}, \pi]$ and the rules of the sequent calculus. Note that strictly speaking rewriting rules are stronger than (Cut), but in our case it is a simple matter to simulate them by (Cut).

An obvious question to ask is whether all rules given by the sequent calculus are necessary for the proof of correctness in Section 4. It is easily seen that we can replace the sequent calculus with the following rules. For any formulae φ, ψ, and χ we need

$$\varphi \cdot (\psi \vee \chi) \to (\varphi \cdot \psi) \vee (\varphi \cdot \chi), \tag{32}$$

$$(\varphi \vee \psi) \cdot \chi \to (\varphi \cdot \chi) \vee (\psi \cdot \chi). \tag{33}$$

Moreover, we do not need the full idempotency, commutativity, and associativity of join. It is sufficient to add the particular instance

$$(eX \vee eA) \vee eA \to eX \vee eA \tag{34}$$

to $\Phi[\mathcal{A}, \pi]$. It follows that the *accessibility problem* $\varphi \to^* \psi$, whether we can rewrite φ to ψ in finitely many steps, is generally undecidable even for rewriting systems satisfying only rules (32) and (33). A particular undecidable problem is $e\,\delta(w)X \vee eA \to^* eX \vee eA$.

It follows easily from the previous sections that the accessibility problem remains generally undecidable if we add any (general) rules which are subsumed by the commutativity and contractivity of product; the idempotency, commutativity, and associativity of join; the distributivity of product over join (in the opposite direction).

Acknowledgements

The author would like to thank to Rostislav Horčík who identified the $\{\cdot, \vee\}$-fragment as the important one. He was also very helpful and supportive during the author's work on the problem. The anonymous referee, Zuzana Haniková, and Marta Bílková had many useful comments on various versions of this paper. Félix Bou asked the question whether the construction works in the one-variable fragment. The work was supported by grant P202/10/1826 of the Czech Science Foundation and by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Willem J. Blok and Clint J. van Alten. On the finite embeddability property for residuated ordered groupoids. *Transactions of the AMS*, 357(10):4141–4157, 2004.

[2] Wojciech Buszkowski. Some decision problems in the theory of syntactic categories. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 28(33-38):539–548, 1982.

[3] Wojciech Buszkowski. Lambek calculus with nonlogical axioms. In C. Casadio, P. J. Scott, and R. A. G. Seely, editors, *Language and Grammar: Studies in Mathematical Linguistics and Natural Language*, number 168 in CSLI Lecture Notes, pages 77–93. CSLI, Stanford, 2005.

[4] Wojciech Buszkowski and Maciej Farulewski. Nonassociative lambek calculus with additives and context-free languages. In Orna Grumberg, Michael Kaminski, Shmuel Katz, and Shuly Wintner, editors, *Languages: From Formal to Natural*, number 5533 in Lecture Notes in Computer Science, pages 45–58. Springer, 2009.

[5] John Cocke and Marvin Lee Minsky. Universality of Tag systems with P=2. *Journal of the ACM*, 11(1):15–20, 1964.

[6] Kosta Došen. Sequent-systems and groupoid models. I. *Studia Logica*, 47(4):353–385, 1988.

[7] Maciej Farulewski. Finite embeddability property for residuated groupoids. *Reports on Mathematical Logic*, 43:25–42, 2008.

[8] Nikolaos Galatos, Peter Jipsen, Tomasz Kowalski, and Hiroakira Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*, volume 151 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, April 2007.

[9] Nikolaos Galatos and Hiroakira Ono. Cut elimination and strong separation for substructural logics: An algebraic approach. *Annals of Pure and Applied Logic*, 161(9):1097–1133, 2010.

[10] George Grätzer. *Lattice Theory: Foundation*. Springer, 2011.

[11] Zuzana Haniková and Rostislav Horčík. Finite embeddability property for residuated groupoids. *Algebra Universalis*, to appear.

[12] Ryuichi Hori, Hiroakira Ono, and Harold Schellinx. Extending intuitionistic linear logic with knotted structural rules. *Notre Dame Journal of Formal Logic*, 35(2):219–242, 03 1994.

[13] Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65(3):154–170, 1958.

[14] Joachim Lambek. On the calculus of syntactic types. In Roman Jakobson, editor, *Structure of Language and Its Mathematical Aspects*, pages 166–178, Providence, 1961. American Mathematical Society.

[15] Hiroakira Ono. Proof-theoretic methods in nonclassical logic: An introduction. In M. Takahashi, M. Okada, and M. Dezani-Ciancaglini, editors, *Theories of Types and Proofs*, volume 2 of *MSJ Memoirs*, pages 207–254. Mathematical Society of Japan, 1998.

[16] Emil Leon Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65(2):197–215, 1943.

[17] Anne Sjerp Troelstra. *Lectures on Linear Logic*. Number 29 in CSLI lecture notes. Stanford, CA : Center for the Study of Language and Information, 1992.