# Undecidability of Some Substructural Logics

## Nerozhodnutelnost některých substrukturálních logik

Karel Chvalovský

Theses of the Doctoral Dissertation

Teze disertační práce

Charles University in Prague
Faculty of Arts
Department of Logic

Univerzita Karlova v Praze
Filozofická fakulta
Katedra logiky

Subject of study / Studijní obor: Logic / Logika

Supervisor / Školitelka: Marta Bílková

2015

The thesis deals with propositional substructural logics and shows some undecidability (unsolvability) results for them. Formally speaking, it consists of an introduction[1] and two appended papers:

1. Karel Chvalovský. 'Undecidability of Consequence Relation in Full Non-associative Lambek Calculus'. *Journal of Symbolic Logic* (to appear). (Appendix A)

2. Karel Chvalovský and Rostislav Horčík. 'Full Lambek Calculus with Contraction is Undecidable'. *Journal of Symbolic Logic* (to appear). (Appendix B)

Although both appendices are completely independent they share some techniques; the main method used in Appendix A is crucial also in Appendix B.

Our results, we believe, are interesting mainly for the following reason. They show undecidable problems, seemingly simple, which had an unclear status—there were, as far as we know, attempts to prove their decidability. In particular, the problem studied in Appendix B was open for almost twenty years, see [SO96].

In Appendix A we show that the finite consequence relation for some basic non-associative substructural logics is undecidable. It is worth noticing that such results usually depend heavily on associativity. Algebraically speaking, we prove that some word problems are undecidable.

In Appendix B we show that the basic substructural logic with the rule of contraction and lattice connectives in its language has an undecidable set of theorems. Such results are rather rare, the relevance logic **R** [Urq84] being a prominent example. Algebraically speaking, we prove that the class of square-increasing residuated lattices has an undecidable equational theory.

Our main results are summarized in Table 1.1, where some other relevant results are also included.

# 1 Introduction

Substructural logics, a rapidly emerging field, have various motivations that come from linguistics, computer science, philosophy etc. Their common feature regardless of motivations is that we abandon the classical propositional Boolean logic for some reasons, for details see e.g. [Res00; Pao02; Gal+07; Bus10]. In particular, the properties of standard logical connectives can be inadequate in various situations.

[1] Note that we use parts of this introduction in this text without further notice.

Say we would like to reason about words in a formal language. Moreover, for our purposes the expressive power of propositional logic seems sufficient given that we have a reasonable encoding of words. One way to deal with this is to interpret letters as atoms and define the concatenation operation as a logical connective. It is quite clear that standard Boolean connectives are not suitable for this purpose.

However, we can define a new logic with connectives that have the desired properties. It does not necessarily mean that the problem is not expressible in the classical Boolean setting, but rather that there is a more suitable way how to deal with it. Usually we try to avoid a complicated encoding or unnecessarily expressive formal systems, e.g. first-order logic. Therefore we study only propositional logics in this thesis.

It turns out, quite interestingly, that completely different motivations often lead to the very same logic. Moreover, there is a uniform way how to obtain many of these logics using simple modifications of Gentzen's sequent calculus **LJ** for intuitionistic propositional logic [Gen35a; Gen35b], which is a single-conclusion variant of the calculus **LK** for classical propositional logic.

In this thesis substructural logics are described by sequent calculi obtained from **LJ** by dropping some (possibly all) structural rules. In particular, we deal only with rules affecting antecedents, namely *exchange* (e), *contraction* (c), and *left-weakening*[2] or *integrality* (i) that are given by

$$\text{(e)}\ \frac{\Gamma, \varphi, \psi, \Delta \Rightarrow \chi}{\Gamma, \psi, \varphi, \Delta \Rightarrow \chi} \qquad \text{(c)}\ \frac{\Gamma, \varphi, \varphi, \Delta \Rightarrow \psi}{\Gamma, \varphi, \Delta \Rightarrow \psi} \qquad \text{(i)}\ \frac{\Gamma, \Delta \Rightarrow \psi}{\Gamma, \varphi, \Delta \Rightarrow \psi}$$

These rules describe structures occurring in antecedents and dropping all of them equals to assuming that we deal with sequences of formulae there. This also affects rules introducing logical connectives, because there are often two natural ways how to define them, which are equivalent in **LJ** but for weaker logics this need not be true. In particular, we usually obtain two conjunctions, where one corresponds to commas in antecedents, called product ($\cdot$) or fusion, and the other one more resembles "standard" conjunction in **LJ**, called meet ($\wedge$). These connectives are defined by

$$(\cdot\text{L})\ \frac{\Gamma, \varphi, \psi, \Delta \Rightarrow \chi}{\Gamma, \varphi \cdot \psi, \Delta \Rightarrow \chi} \qquad (\cdot\text{R})\ \frac{\Gamma \Rightarrow \varphi \qquad \Delta \Rightarrow \psi}{\Gamma, \Delta \Rightarrow \varphi \cdot \psi}$$

[2]We ignore the rule of right-weakening which is of no importance in our thesis.

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

Recall that product in a logic without the previously mentioned structural rules has the properties we desire to model the concatenation operator. Roughly speaking, the logic obtained from **LJ** by dropping all the previously mentioned rules is called the *Full Lambek calculus* (**FL**); its language contains product ($\cdot$), left ($\backslash$) and right ($/$) implications[3], meet ($\wedge$), and join ($\vee$).[4] A presence of structural rules is usually expressed in a subscript appended to the name of the original logic, e.g. $\mathbf{FL_c}$ is **FL** with the rule of contraction.

In fact, we can obtain even weaker logics by assuming that antecedents are non-associative, i.e. we have trees of formulae, instead of sequences, in them. This way we obtain the *Full Non-associative Lambek calculus* (**FNL**) from **FL**. A non-associative calculus was first introduced by Lambek [Lam61], who considered it more suitable for linguistic applications than his original associative calculus [Lam58].

## Simple undecidable problems

A standard way how to prove an undecidability result is to encode a simple problem which is already known to have that property. Many such problems have been used in logic, e.g. variants of tilling, the Post correspondence problem. Our results use two well-known abstract machines, namely counter machines and tag systems, which have an undecidable halting problem.

Counter machines were formalized by Minsky [Min61] and for that reason they are sometimes called after him Minsky machines. Roughly speaking, they are Turing machines with counters instead of tapes. Hence instructions are defined accordingly, we can only increment and decrement counters. However, the later instruction contains a test. If we attempt to decrement a counter which is equal to zero then it remains equal to zero but the computation continues in different state. It is well-known [Min61] that counter machines with two counters can simulate Turing machines and hence they have an undecidable halting problem. These machines are particularly suitable for our purposes, since they are easy to represent in

[3]Two implications occur in logics without the exchange rule. It can be easily seen algebraically, since implications are defined by residuation laws

$$a \cdot b \leq c \quad \text{iff} \quad b \leq a \setminus c \quad \text{iff} \quad a \leq c \,/\, b$$

where we can read $\leq$ as $\Rightarrow$. In algebra they are, for obvious reasons, called divisions.

[4]Usually, constants 0 and 1 are also included, but they do not play an important role in our thesis and hence we mostly ignore them.

terms of string rewriting systems. Moreover, strings can be easily handled in some substructural logics, e.g. product corresponds to the concatenation operation in **FL**.

Tag systems were proposed by Post [Pos43] and process finite words over a finite alphabet $\mathcal{A}$. Fix $n > 0$. An $n$-tag system is given by a set of production rules (a program) which is a function $\pi\colon \mathcal{A} \to \mathcal{A}^*$, where $\mathcal{A}^*$ is the set of finite words over $\mathcal{A}$. The computation of the $n$-tag system given by $\mathcal{A}$ and $\pi$ on a word $w \in \mathcal{A}^*$ is defined as follows. If $|w| < n$ we terminate. Otherwise, we examine the first letter in $w$, which is some $a \in \mathcal{A}$. Then we delete the first $n$ letters in $w$ and append $\pi(a)$ 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 $n$ letters. It is known [CM64; Wan63] that 2-tag systems have an undecidable halting problem.

# A Undecidability of Consequence Relation in Full Non-associative Lambek Calculus

In Appendix A we prove that the (finite) consequence relation, usually called the deducibility problem, in **FNL**, which is a non-associative logic, is undecidable. We provide a construction how to encode 2-tag systems in the language of **FNL** using only sequents with products and joins. Therefore already this fragment of **FNL** is undecidable. Note that it is known that the deducibility problem for **FNL** without join is decidable [Far08]. We also show, in the introduction, that the deducibility problem for the fragment of **FNL** only with an implication and join is also undecidable.

Our encoding works due to an interplay between product and join. In particular, the fact that product distributes over join is essential there. Roughly speaking, product enables us to represent words and join ensures that we can rewrite them correctly using a form of conditional rewriting. In particular, we use join to exchange pieces of information needed to correctly perform a computation of a 2-tag system.

The whole construction is rather robust. Hence the same undecidability results follow easily for $\mathbf{FNL_c}$, $\mathbf{FNL_e}$, and $\mathbf{FNL_{ec}}$[5]. Similarly, the deducibility problem for the one-variable fragment of **FNL** (and $\mathbf{FNL_e}$) is also shown to be undecidable.

[5]Note that $\mathbf{FL_{ec}}$, the associative variant of $\mathbf{FNL_{ec}}$, has a decidable deducibility problem.

| | Theorems | Deducibility problem |
|---|---|---|
| **FL** | decidable | undecidable |
| $\mathbf{FL_e}$ | decidable | undecidable |
| $\mathbf{FL_c}$ | undecidable (see Appendix B) | undecidable |
| $\mathbf{FL_{ec}}$ | decidable | decidable |
| $\mathbf{FL_i}$ | decidable | decidable |
| **FNL** | decidable | undecidable (see Appendix A) |
| $\mathbf{FNL_e}$ | decidable | undecidable (see Appendix A) |
| $\mathbf{FNL_c}$ | | undecidable (see Appendix A) |
| $\mathbf{FNL_{ec}}$ | | undecidable (see Appendix A) |
| $\mathbf{FNL_i}$ | decidable | decidable |

Table 1.1: Some substructural logics related to this thesis and their decidability.

It is clear that word problems for the corresponding algebraic structures, which are based on join-semilattice groupoids, are undecidable. In fact, even a little bit stronger results than those following immediately from the algebraic completeness of substructural logics can be obtained. Similarly, it follows that some reachability problems for term rewriting systems are also undecidable.

# B Full Lambek Calculus with Contraction is Undecidable

In Appendix B we deal with an associative logic, but it turns out that we face again a form of non-associativity there and hence techniques from Appendix A can be used again. A common approach how to prove that a substructural logic has a decidable set of theorems uses the cut elimination, since then there are usually only finitely many possible proofs to be checked. If the logic in question contains the contraction rule, as in our case, then the situation can be entirely different. However, it is still possible to prove decidability for some logics, e.g. $\mathbf{FL_{ec}}$ [KO91], using Kripke's combinatorial idea [Kri59] which was further extended by Meyer [Mey66]. However, in $\mathbf{FL_c}$ this is not possible, since we prove that this logic is undecidable and hence there is no bound on the proof-search in it.

An outline of our proof is as follows. First, we start with the string rewriting system [Hor15] developed to prove that the deducibility problem for $\mathbf{FL_c}$ is undecidable. It uses a representation of counter machines by

square-free words, they do not contain $uu$ as a subword, and therefore the rule of contraction cannot influence the encoding. Second, we show how to interpret such string rewriting systems in what we call atomic conditional string rewriting systems, where their conditionality is closely connected to the main technique used in Appendix A. Finally, such rewriting systems can be easily encoded in $\mathbf{FL_c}$ as formulae. It follows that the set of theorems in $\mathbf{FL_c}$ is undecidable. In fact, we prove that already the positive fragment of $\mathbf{FL_c}$ is undecidable, since the whole construction requires only an implication, join, and meet.

Clearly, it also follows that the corresponding algebraic structure, square-increasing residuated lattices, has an undecidable equational theory. In fact, the real order used in Appendix B is the opposite, we use algebraic methods to prove the correctness and completeness of our encoding there. It is not essential, but such an approach seems to be more convenient than a proof-theoretical one, which is used in Appendix A. However, all the arguments would remain the same regardless of the method used.

We can formulate an interesting corollary. As the set of theorems in $\mathbf{FL_c}$ is recursively enumerable but not recursive, it is possible to get the following variant of the deduction theorem. The way we obtain it is purely algorithmic and hence it differs significantly from how "standard" deduction theorems look like.

**Theorem.** *Let $\Gamma \cup \{\varphi\}$ be a finite set of formulae. There is an explicit algorithm that produces a formula $\psi$ (given an input $\varphi$ and $\Gamma$) such that $\psi$ is provable in $\mathbf{FL_c}$ iff $\varphi$ is provable in $\mathbf{FL_c}$ from $\Gamma$.*

# Bibliography


[Bus10] Wojciech Buszkowski. 'Lambek Calculus and Substructural Logics'. *Linguistic Analysis* 36 (2010), pp. 15–49. URL: http://www.linguisticanalysis.com/volumes/36issue1-4.

[CM64] John Cocke and Marvin Lee Minsky. 'Universality of Tag Systems with P=2'. *Journal of the ACM* 11.1 (1964), pp. 15–20. DOI: 10.1145/321203.321206.

[Far08] Maciej Farulewski. 'Finite embeddability property for residuated groupoids'. *Reports on Mathematical Logic* 43 (2008), pp. 25–42. URL: http://rml.tcs.uj.edu.pl/rml-43/02-farulewski.pdf.

[Gal+07] Nikolaos Galatos, Peter Jipsen, Tomasz Kowalski and Hiroakira Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Vol. 151. Studies in Logic and the Foundations of Mathematics. Amsterdam: Elsevier, Apr. 2007, p. 532.

[Gen35a] Gerhard Gentzen. 'Untersuchungen über das logische Schließen I'. *Mathematische Zeitschrift* 39.1 (1935), pp. 176–210. DOI: 10.1007/BF01201353.

[Gen35b] Gerhard Gentzen. 'Untersuchungen über das logische Schließen II'. *Mathematische Zeitschrift* 39.1 (1935), pp. 405–431. DOI: 10.1007/BF01201363.

[Hor15] Rostislav Horčík. 'Word Problem for Knotted Residuated Lattices'. *Journal of Pure and Applied Algebra* 219.5 (May 2015), pp. 1548–1563. DOI: 10.1016/j.jpaa.2014.06.015.

[KO91] Eiji Kiriyama and Hiroakira Ono. 'The contraction rule and decision problems for logics without structural rules'. *Studia Logica* 50.2 (1991), pp. 299–319. ISSN: 0039-3215. DOI: 10.1007/BF00370189.

[Kri59] Saul Aaron Kripke. 'The problem of entailment'. *Journal of Symbolic Logic* 24 (1959). abstract, p. 324. URL: http://www.jstor.org/stable/2963903.

[Lam58] Joachim Lambek. 'The Mathematics of Sentence Structure'. *American Mathematical Monthly* 65.3 (1958), pp. 154–170. URL: http://www.jstor.org/stable/2310058.

[Lam61] Joachim Lambek. 'On the calculus of syntactic types'. In: *Structure of Language and Its Mathematical Aspects*. Ed. by Roman Jakobson. Providence, Rhode Island: American Mathematical Society, 1961, pp. 166–178. DOI: 10.1090/psapm/012.

[Mey66] Robert K. Meyer. 'Topics in modal and many-valued logic'. PhD thesis. University of Pittsburgh, 1966.

[Min61] Marvin Lee Minsky. 'Recursive Unsolvability of Post's Problem of "Tag" and other Topics in Theory of Turing Machines'. *Annals of Mathematics* 74.3 (1961), pp. 437–455. DOI: 10.2307/1970290.

[Pao02] Francesco Paoli. *Substructural Logics: A Primer*. Trends in Logic. Dordrecht: Springer, 2002.

[Pos43] Emil Leon Post. 'Formal Reductions of the General Combinatorial Decision Problem'. *American Journal of Mathematics* 65.2 (1943), pp. 197–215. URL: http://www.jstor.org/stable/2371809.

[Res00] Greg Restall. *An Introduction to Substructural Logics*. London: Routledge, 2000.

[SO96] Bayu Surarso and Hiroakira Ono. 'Cut elimination in noncommutative substructural logics'. *Reports on Mathematical Logic* 30 (1996), pp. 13–29.

[Urq84] Alasdair Urquhart. 'The Undecidability of Entailment and Relevant Implication'. *Journal of Symbolic Logic* 49.4 (1984), pp. 1059–1073. DOI: 10.2307/2274261.

[Wan63] Hao Wang. 'Tag systems and lag systems'. *Mathematische Annalen* 152.1 (1963), pp. 65–74. DOI: 10.1007/BF01343730.