Full Lambek Calculus with contraction is undecidable

Karel Chvalovský

joint work with Rostislav Horčík

Institute of Computer Science,
Academy of Sciences of the Czech Republic, Prague

Introduction

- a rather technical proof; all details are in our paper
- several small things which fit together nicely
- it is not so common among substructural logics that a logic has an undecidable set of theorems, cf. some known examples
  - in relevance logics (Urquhart [1984]),
  - in linear logic (Lincoln et al. [1992]),
  - the equational theory of modular lattices (Freese [1980])

Gentzen sequent calculus for $\mathbf{FL_c}$

Assume we have a sequent

$$\Gamma \Rightarrow \chi$$

where Γ is a sequence of formulae separated by commas.

We need structural rules

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

exchange | contraction | left-weakening

(+right-weakening) $= \mathbf{LJ}$

Gentzen sequent calculus for $\mathbf{FL_c}$

Assume we have a sequent

$$\Gamma \Rightarrow \chi$$

where Γ is a sequence of formulae separated by commas.

Roughly speaking, if we have only

$$\text{(c)}\ \frac{\Gamma, \varphi, \varphi, \Delta \Rightarrow \chi}{\Gamma, \varphi, \Delta \Rightarrow \chi}$$

contraction

$=$ the Full Lambek Calculus with contraction $(\cdot, \backslash, /, \wedge, \vee, 1)$.

In fact, this is the positive fragment of $\mathbf{FL_c}$ denoted $\mathbf{FL_c^+}$ since we do not allow the empty succedent (0).

Algebraic counterpart

$\mathcal{RL}_c$-algebras

A *square-increasing residuated lattice* $\mathbf{A} = \langle A, \wedge, \vee, \cdot, \backslash, /, 1\rangle$ is an algebraic structure such that

- $\langle A, \cdot, 1\rangle$ is a monoid,
- $\langle A, \wedge, \vee\rangle$ is a lattice,
- the law of residuation holds—for all $a, b, c \in A$ hold

$$a \cdot b \leq c \qquad \text{iff} \qquad b \leq a \backslash c \qquad \text{iff} \qquad a \leq c \,/\, b,$$

- $x \leq x \cdot x$.

($\leq$ is induced by the lattice structure)

Fact

$\varphi \Rightarrow \psi$ *is provable in* $\mathbf{FL_c^+}$ *iff* $\varphi \leq \psi$ *holds in* $\mathcal{RL}_c$.

Our result

Theorem
The set of theorems provable in $\mathbf{FL_c^+}$ (and hence $\mathbf{FL_c}$) is undecidable.

Theorem
The equational theory of $\mathcal{RL}_c$ (and hence $\mathbf{FL_c}$-algebras) is undecidable.

Decidability using cut-elimination

The elimination of (Cut) usually immediately gives decidability, but here it is not the case since we have

$$\text{(c)}\ \frac{\Gamma, \varphi, \varphi, \Delta \Rightarrow \psi}{\Gamma, \varphi, \Delta \Rightarrow \psi}$$

Still we have decidability for $\mathbf{LJ}$, $\mathbf{FL_{ec}}$, and $\mathbf{L_c}$ (Bimbó [2014]).
Usually based on a combinatorial idea of Kripke.

Undecidability proofs (for consequence relations)

1. We pick a machine with an undecidable halting problem
   - e.g. counter machines
2. We express such a problem in terms of rewriting systems
   - states=strings,
   - QUESTION: is the accepting state A reachable from a state S, i.e. $S \to^* A$?

     (given rewriting rules describing the behaviour of machine)
3. We express such a reachability as a provability problem
   - $S \to^* A$ is equivalent to proving $S \Rightarrow A$ ($\cdot$ is concatenation)
   - QUESTION: is $S \Rightarrow A$ provable?

     (given a theory based on rewriting rules)

Undecidability proofs (for consequence relations)

1. We pick a machine with an undecidable halting problem
   - e.g. counter machines
2. We express such a problem in terms of rewriting systems
   - states=strings,
   - QUESTION: is the accepting state A reachable from a state S, i.e. $S \rightarrow^* A$?

     (given rewriting rules describing the behaviour of machine)
3. We express such a reachability as a provability problem
   - $S \rightarrow^* A$ is equivalent to proving $S \Rightarrow A$ ($\cdot$ is concatenation)
   - QUESTION: is $S \Rightarrow A$ provable?

     (given a theory based on rewriting rules)

Undecidability proofs (for consequence relations)

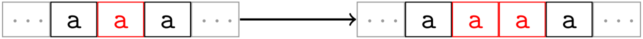
1. We pick a machine with an undecidable halting problem
   - e.g. counter machines
2. We express such a problem in terms of rewriting systems
   - states=strings,
   - QUESTION: is the accepting state A reachable from a state S, i.e. $S \to^* A$?

     (given rewriting rules describing the behaviour of machine)
3. We express such a reachability as a provability problem
   - $S \to^* A$ is equivalent to proving $S \Rightarrow A$ $\quad$ ($\cdot$ is concatenation)
   - QUESTION: is $S \Rightarrow A$ provable?

     (given a theory based on rewriting rules)

String rewriting systems, contraction, and counters

What is the problem with contraction?



since $a \Rightarrow aa$ is provable.

Square-free words (strings)

If we have a morphism over the alphabet $\{a, b, c\}$ defined by

$$h(a) = abc \qquad h(b) = ac \qquad h(c) = b$$

then $h^m(a)$ is square free for any m. Hence we can represent

$$a^m \qquad \text{by} \qquad h^m(a).$$

This way we obtain a suitable string rewriting system, i.e. no squares occur in accepting derivations, see (Horčík [2015]).

String rewriting systems, contraction, and counters

What is the problem with contraction?

$$\cdots \; \boxed{\mathtt{a}}\boxed{\mathtt{a}}\boxed{\mathtt{a}} \; \cdots \longrightarrow \cdots \; \boxed{\mathtt{a}}\boxed{\mathtt{a}}\boxed{\mathtt{a}}\boxed{\mathtt{a}} \; \cdots$$

since $a \Rightarrow aa$ is provable.

Square-free words (strings)

If we have a morphism over the alphabet $\{a, b, c\}$ defined by

$$h(a) = abc \qquad h(b) = ac \qquad h(c) = b$$

then $h^m(a)$ is square free for any m. Hence we can represent

$$a^m \quad \text{by} \quad h^m(a).$$

This way we obtain a suitable string rewriting system, i.e. no squares occur in accepting derivations, see (Horčík [2015]).

String rewriting systems inside $\mathbf{FL_c}$

The main problem is how to express a string rewriting system inside $\mathbf{FL_c}$ using (Id) as the only initial sequents.

We can simulate a rewriting rule

Assume we have a rule $s \Rightarrow t$. We can simulate it by $s \setminus t$ since $s(s \setminus t) \Rightarrow t$ is provable.

$$\underbrace{\boxed{\cdots}\boxed{\mathtt{s}}\boxed{\mathtt{s\backslash t}}\boxed{\cdots}}_{\varphi} \Longrightarrow \underbrace{\boxed{\cdots}\boxed{\mathtt{t}}\boxed{\cdots}}_{\psi}$$

meaning $\varphi \Rightarrow \psi$ is provable in $\mathbf{FL_c}$.

String rewriting systems inside $\mathbf{FL_c}$

The main problem is how to express a string rewriting system inside $\mathbf{FL_c}$ using (Id) as the only initial sequents.

We can simulate a rewriting rule

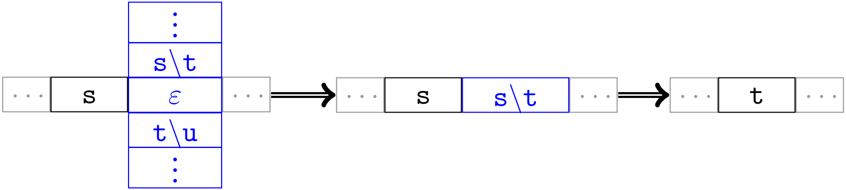
Assume we have a rule $s \Rightarrow t$. We can simulate it by $s \setminus t$ since $s(s \setminus t) \Rightarrow t$ is provable.

$$\underbrace{\boxed{\cdots}\boxed{\texttt{s}}\boxed{\texttt{s}\backslash\texttt{t}}\boxed{\cdots}}_{\varphi} \quad \Rightarrow \quad \underbrace{\boxed{\cdots}\boxed{\texttt{t}}\boxed{\cdots}}_{\psi}$$

meaning $\varphi \Rightarrow \psi$ is provable in $\mathbf{FL_c}$.

String rewriting systems inside $\mathbf{FL_c}$

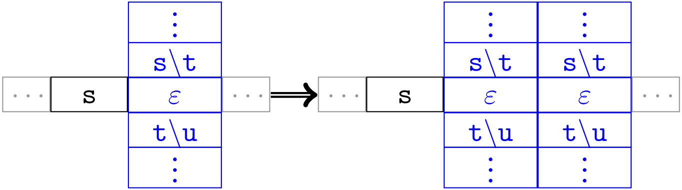
We can simulate finitely many rules using meet



since $\cdots \wedge (s \setminus t) \wedge \ldots \Rightarrow s \setminus t$ is provable.

String rewriting systems inside $\mathbf{FL_c}$

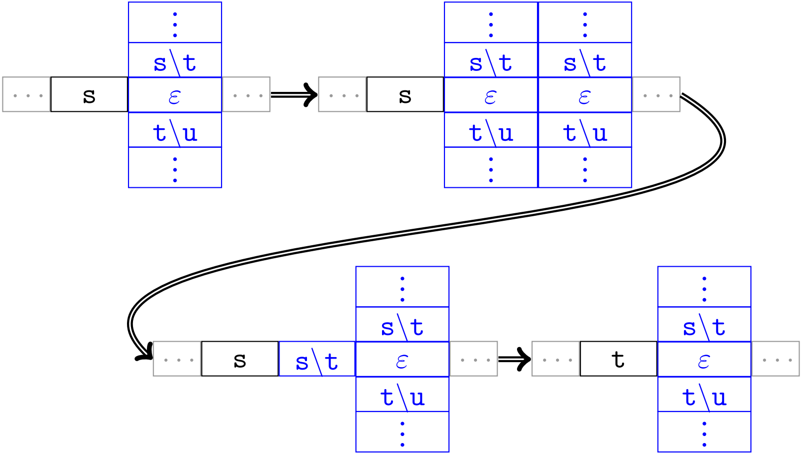
We can reuse rules thanks to contraction



Hence we can simulate rewriting by placing those rules next to every atomic symbol (letter).

String rewriting systems inside $\mathbf{FL_c}$

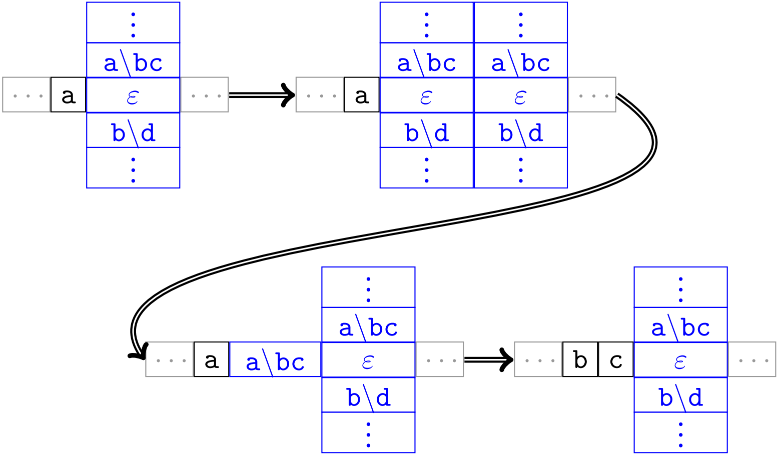
We can reuse rules thanks to contraction



Hence we can simulate rewriting by placing those rules next to every atomic symbol (letter).

String rewriting systems inside $\mathbf{FL_c}$

It is a bit more complicated. . .



It would work if the right sides were only atomic (=atomic rules).

Simulation of a non-atomic rule by atomic rules

We produce a modification of the string rewriting system we started with to fulfill the previous atomicity condition.

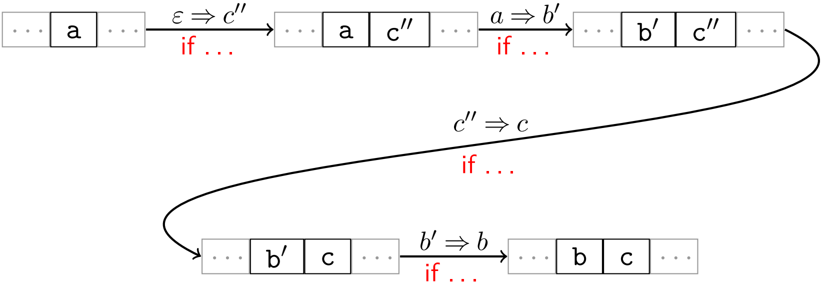
Rule $a \Rightarrow bc$ is simulated by atomic rules used in a right order

How do we represent the conditional part?

Simulation of a non-atomic rule by atomic rules

We produce a modification of the string rewriting system we started with to fulfill the previous atomicity condition.

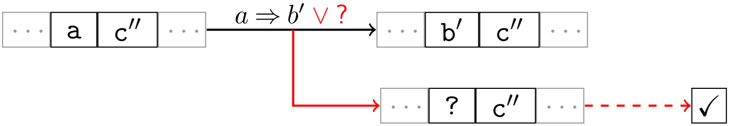
Rule $a \Rightarrow bc$ is simulated by atomic rules used in a right order



How do we represent the conditional part?

Conditional rewriting system in $\mathbf{FL_c}$

Rule $a \Rightarrow b'$ simulated correctly



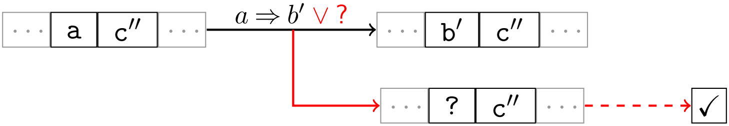
cf. (Lincoln et al. [1992],. . . ,Chvalovský [2015]).

Tests require another level of rewriting

where Pac-Man is a string rewriting system with only atomic rules. In fact, it is a finite automaton (=recognizes a regular language).

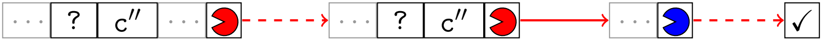
Conditional rewriting system in $\mathbf{FL_c}$

Rule $a \Rightarrow b'$ simulated correctly



cf. (Lincoln et al. [1992],...,Chvalovský [2015]).

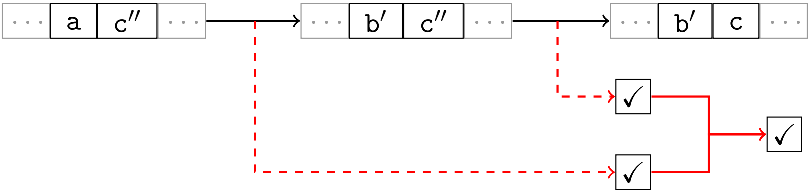
Tests require another level of rewriting



where Pac-Man is a string rewriting system with only atomic rules. In fact, it is a finite automaton (=recognizes a regular language).

Conditional rewriting system in $\mathbf{FL_c}$

Join is idempotent



If we put all those little things together we obtain the whole construction:

a string rewriting system $\rightsquigarrow$
$\rightsquigarrow$ **an atomic conditional variant of it** $\rightsquigarrow$
$\rightsquigarrow$ **an encoding of atomic conditional systems in $\mathbf{FL_c}$.**

Final remarks

- the completeness of construction is proved algebraically
- it is enough to have an implication, join, and meet
- an "algorithmic" deduction theorem follows from our result

Thank you!

Bibliography

Katalin Bimbó. *Proof Theory: Sequent Calculi and Related Formalisms*. Discrete Mathematics and Its Applications. CRC Press, London, 2014.

Karel Chvalovský. Undecidability of consequence relation in full non-associative Lambek calculus. *Journal of Symbolic Logic*, 80(2):567–586, 2015. doi: 10.1017/jsl.2014.39.

Ralph Freese. Free modular lattices. *Transactions of the AMS*, 261(1):81–91, 1980. doi: 10.1090/S0002-9947-1980-0576864-X.

Rostislav Horčík. Word problem for knotted residuated lattices. *Journal of Pure and Applied Algebra*, 219(5):1548–1563, May 2015. doi: 10.1016/j.jpaa.2014.06.015.

Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56(1–3):239–311, 1992. doi: 10.1016/0168-0072(92)90075-B.

Alasdair Urquhart. The undecidability of entailment and relevant implication. *Journal of Symbolic Logic*, 49(4):1059–1073, 1984. doi: 10.2307/2274261.