Report on the doctoral thesis of

Undecidability of some substructural logics,
by Karel Chvalovský

prepared by Nick Galatos.

The thesis is very strong and contains difficult and important results, and therefore my evaluation is extremely positive.

The candidate proves two new main results about the undecidability of specific logics. The first is about the undecidability of the consequence relation of the non-associative full Lambek calculus (**NFL**) and the second for the theorem-hood of contractive full Lambek calculus ($\mathbf{FL_c}$). The second problem has been open for quite some time and has resisted previous attempts by other researchers.

Both results are proved by interpreting each a known undecidable problem. This is a fairly simple matter for, say, the consequence relation of full Lambek calculus, as one can interpret there string rewrite systems (using the connective of multiplication and considering inequalities or implication) and therefore, for example, simulate the computation of a two-counter machine, a problem that is undecidable. However the absence of associativity as well as the presence of contraction impose problems with that encoding, and of course the demand that we focus on theorem-hood in the second case presents an even bigger hurdle. The candidate manages to overcome these obstacles and present sound and complete encodings.

A key idea underpinning both solutions is that of making use of the connective of join to simulate parallel computation, with the demand that all branches of the computation terminate appropriately. In particular, since computation cannot be performed as in the case of **FL** one branch of the parallel computation corresponds to the main computation (more-or-less corresponding to the single computation in **FL**, but inadvertently much more liberal) and a parallel stream is performed as a checking mechanism to make sure that the main computation is not too liberal and is applied as intended. Essentially, at the point of branching the minor branch serves as specifying *obligations* that need to be later satisfied for the whole process to terminate. This idea has precursors in the literature but its use to this extent and in this setting are novel and very impressive. Both papers, as well as the introduction are very well written and motivated, and they are presented in a clear, organized and also accessible way.

The results are not obtained by a direct translation and certain expertise is required on various computational models; the candidate has mastered them and can apply them in creative ways. In particular such computational models that feature in the thesis are the full Lambek calculus itself, counter machines and string rewrite systems, such as general ones, conditional atomic ones, context-free grammars, and tag systems.

In the first paper, 2-tag systems are used. However, NFL cannot interpret them directly even if they are presented as operating on non-associative words (namely binary labeled trees) on variables fully associated to the right, because such 2-tag systems operate on both sides of the word (to read and erase on the left, and to write on the right side of the word). If the word is fully associated to the right, such as $a(bc)$ then final parts of the word correspond to subtrees, for example bc, but initial ones, such as ab, do not. One could of course conceive of 2-tag systems that operate on (initial, in our case) parts of a tree that do not form a subtree, but **NFL** cannot simulate such behavior, as its rules (in particular cut) can only operate on subtrees.

It is natural to then consider *right-tag* systems, namely ones that operate only on final parts of words, but unfortunately, they cannot fully implement the behavior of regular tag systems. Here, if $s \rightsquigarrow t$ is a rule, the right-tag system is allowed to rewrite only as $us \to ut$, but not as the usual $usv \to utv$. Actually, given a 2-tag system one can write rules that allow for accepting in right-tag systems all the words accepted by the 2-tag system, but inadvertently the right-tag system ends up accepting more words, so the implementation is not faithful.

One would like to have a way of controlling the acceptance of these words and restrict only to the ones that are accepted by the initial 2-tag system. The candidate does exactly that, by using the connective of join and the idea of imposing an *obligation* that needs to be met in the minor track(s) of the computation, while the main computation is guaranteed to work as more words are accepted by the right-tag system. In effect this results in designing conditional right-tag systems that fully implement 2-tag systems, even though this is not explicitly stated in this way in the thesis. These specific conditional right-tag systems are then implementable in **NFL**.

In the second paper a plethora of nice ideas are implemented in the proof. Since the results are presented in a very modular way, this leads to facts that are of independent interest on their own right.

One of the obstacles is the presence of contraction which can affect the canonical presentation of words by contracting repeated subwords. The solution to this problem is to work only with square-free words, following ideas in the proof of the weaker result about the undecidability of the consequence relation of $\mathbf{FL_c}$. There an appropriate undecidable string rewriting system (SRS) is interpreted in the consequence relation (the language of all words that lead to a given fixed word is undecidable and also closed under the contraction rule). An interesting way to encode the rules of the SRS in a single formula and the insight to insert that formula between all letters of a word to be processed seem to achieve the encoding of the very same SRS in the equational theory, but technical difficulties come up in the cases where the rules of the SRS produce words that do not consist of a single letter (they are not atomic). This is in accordance with the expectation that the problem is more challenging as the goal is undecidability of theorem-hood.

The solution to this is similar to that of considering right-tag systems that are conditional, in the first paper. Here atomic SRSs, but which are conditional, are considered and shown to be able to interpret any usual SRS. Here again, the computation splits into the main one and minor computations, which at the branches impose *obligations* that later need to be met for the whole computation to succeed. The join connective does the job again here. The minor computations in the first paper were handled in an explicit ad hoc way, but here the work of controlling the minor computation is done by regular languages which restrict the context of the computed words.

This extra control achieves the simpler form of the SRS rules and permits the above-mentioned encoding for the main computation in the theory of $\mathbf{FL_c}$. However, the minor computation(s) also need to be interpreted in $\mathbf{FL_c}$. The thesis explains a way of encoding regular languages, via their generating right-linear context-free grammars, in $\mathbf{FL_c}$, again taking the rewriting rules (which are the inverses of the generating rules of the grammar) and encoding them into a formula. Here no issues arise, as the corresponding SRS (by reading the generation rules backward) is atomic by its nature. Also, due to the form of the (inverse) generation rules, the formula need not be inserted in all positions (between letters of the processed word) and hence the computation is simpler than for general atomic (conditional) SRSs.

An ingenious part of the second paper is that the completeness of the encoding is established by semantical methods, bypassing a long syntactic analysis as the one done in the completeness proof of the first paper. Actually, because of its constructive nature, the corresponding semantical construction can be recasted, if desired, in a syntactic analysis, but the current presentation is short and elegant, but also illuminating about the connections between syntax and semantics in this setting.

In both papers further discussion is undertaken about extending the results to related logics, and in both cases it is noted that the result can be seen as establishing a restricted (to some fragment of the language) deduction theorem which allows the reduction of the consequence relation into theorem-hood, and which additionally is fully constructive.

In summary the results are impressive and the combination of the ideas and techniques is very clever, resulting in a very high quality doctoral thesis by the candidate. Having met personally the candidate for the degree, I can attest to his strong mathematical abilities and it is my evaluation that he is fully worthy of the degree.

I was recently informed that no corrections can be made to the current form of the submitted thesis. However, I mention some typos that I noticed both in the introduction and in the papers, as well as a couple of minor suggestions on the presentation. In no way do these comments indicate any shortcoming in the thesis or should be interpreted as taking away from its value. As the stage of acceptance of the two papers is not clear to me, and corrections could possibly be allowed at this stage, and also as some of the suggestions below may be useful in oral presentations of the results, I decided to list them.

Typos

1. Page 9, line -1. *instances*
2. Page 12, line -7. *have*
3. Page 19, Definition 1.5.1. Both notations $[1, m]$ and $[1 \dots m]$ seem to denote the same set of natural numbers.
4. Page 26, line 9. *decrement*
5. Page 28, line -6 of Section 1.6. *the previous two*
6. Page 29, line 10 (and other places). The word *tight* is an adverb and an adjective, but not a verb.
7. Page 29, line 4 of fourth paragraph. We need *to be able* to...
8. Page 29, line 10 of fourth paragraph. perform *the* previous...
9. Page 43, line 4. we want *a* unique.
10. Page 43, line 5 of Auxiliary formulae. any such step
11. Page 62, line 9. based on *the* variety

Suggestions

1. The $(Cut^\star)$ rule could be mentioned explicitly as

$$\frac{ut \Rightarrow w \quad s \Rightarrow t}{us \Rightarrow w}\,(Cut^\star)$$

If this is presented as

$$\frac{ut \Rightarrow w}{us \Rightarrow w}\ (Cut^\star : s \Rightarrow t)$$

then it is clear that the calculus allows rewriting in the form of $us \to ut$, given the rule $s \rightsquigarrow t$, and combined with regular (Cut), actually the transitivity part of it, as well as the identity axiom, we can string along (multiple or no) rewrites as usual; namely we obtain the usual transitive reflexive closure. Likewise the rule $(L^\star)$ allows for the derivations $u(s \vee t)v \to usv \vee utv$. In that respect one can identify the computational implementation power of **NFL**, and separate it from the 2-tag and right-tag systems, giving a more modular presentation of the results, similar to the presentation in the second paper. All these are of course alluded to and also given implicitly in the paper in various places, such as in Section A.8, although it would take a more careful presentation to separate these two parts in the completeness argument and identify the right-tag systems involved.

2. The reader of the second paper could be a bit confused by the fact that the rewrite arrow in the string rewrite systems considered is represented by left division (put differently it follows the increasing direction of the inequality), but the same rewrite arrow for right-linear context-free grammars is represented by left division by swapping the roles of numerator and denominator (and follows the decreasing direction of inequalities). This is of course simply because in SRSs the rewrite is pointing to the *terminating* direction while in context-free grammars it is pointing to the *generating* direction, which is opposite to the terminating direction for testing acceptance of a word in a language. Even though it would be probably unnatural (although maybe better in this context) to write grammars with arrows pointing to the opposite of the generating direction, the reader could be warned about this fact.

3. It is mentioned that there is a constructive way to obtain the formula in the deduction theorem. It is possible to describe this formula so that the reader does not need to revisit the construction in the proofs and it would be useful to present the deduction theorem in that, more detailed, manner.