Adjustment of Goal-driven Resolution for Natural Language Processing in TIL

Marie Duží, Michal Fait, and Marek Menšík

VSB-Technical University of Ostrava, Department of Computer Science FEI,
17. listopadu 15, 708 33 Ostrava, Czech Republic
marie.duzi@vsb.cz, michal.fait@vsb.cz, marek.mensik@vsb.cz

Abstract. The paper deals with natural language reasoning and question answering. Having a fine-grained analysis of natural language sentences in the form of TIL (Transparent Intensional Logic) constructions, we apply the General Resolution Method (GRM) with its goal-driven strategy to answer the question (goal) raised on the natural language data. Not only that, we want to answer in an 'intelligent' way, so that to provide logical consequences entailed by the data. From this point of view, GRM appears to be one of the most plausible proof techniques. There are two main new results presented here. First, we found out that it is not always possible to apply all the necessary adjustments of the input constructions first, and then to go on in a standard way by applying the algorithm of the transformation of propositional constructions into the Skolem clausal form followed by the GRM goal-driven resolution techniques. There are plenty of features special for the rich natural language semantics that are dealt with by TIL technical rules and these rules must be integrated with the process of the goal-driven resolution technique rather than separated from it. Second, the strategy of generating resolvents from a given knowledge base cannot be strictly goal-driven. Though we start with a given goal/question, it may happen that there is a point at which we have to make a step aside. We have to apply those special TIL technical rules on another clause first, and only then it is possible to go on with the process of resolving clauses with a given goal. Otherwise our inference machine would be heavily under-inferring, which is not desirable, of course. We demonstrate these new results by two simple examples. The first one deals with property modifiers and anaphoric references. Anaphoric references are dealt with by our substitution method, and the second example demonstrates reasoning with factive verbs like 'knowing' together with definite descriptions and anaphoric references again. Since the definite description occurs *de re* here, we substitute a pointer to the individual referred to for the respective anaphoric pronoun.

Keywords: natural language reasoning, Transparent Intensional Logic, TIL, General Resolution Method, goal-driven strategy, property modifiers, factive verbs, anaphoric references

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2019, pp. 71–81, 2019. © Tribun EU 2019

1 Introduction

Natural language processing, computational linguistics and logic are the disciplines that have more in common than it might seem at first sight. The hard work of many linguists supported by computers produces large corpora of analyzed text data where a lot of information is contained. Logicians contribute by their rational arguments to logically organize and analyze the data so that we might "teach the computers to be intelligent". Artificial intelligence is flourishing. Or not? Actually, it turns out that there is an information overload. People are not able to know their way around the information labyrinth; internet is infested with fake news; artificial "intelligence" is not intelligent.

At the effort of improving the situation, we started the project on question answering system over natural language texts [5]. In this project linguists and logicians work hand-in-hand. After linguistic preprocessing of the texts a corpus of fine-grained logically structured semantic data has been produced. The project goal is this. Given a question the system should be able to answer the question in a more intelligent way than by just providing explicitly recorded text data sought by keywords. To this end we are building up an inference machine that operates on those logical structures so that not only to provide explicit textual knowledge but also to compute inferable logical knowledge [7] such that rational human agents would produce, if only this were not beyond their time and space capacity.

Our background logic is the well-known system of Transparent Intensional Logic (TIL) with its *procedural* rather than set-theoretic semantics. Hence, meaning of a sentence is an abstract procedure encoded by the sentence that can be viewed as an instruction how, in any possible world and time, to evaluate the truth-value of the sentence, if any. These procedures are known as TIL *constructions*. There are six kinds of such constructions defined, namely *variables*, *Trivialization*, *Composition*, *(λ)-Closure*, *Execution* and *Double Execution*. While the first two are atomic constructions, the latter four are molecular. Atomic constructions supply objects on which molecular constructions operate; where X is an object what so ever of TIL ontology, Trivialization ^0X produces X. *Composition* $[FA_1 \ldots A_m]$ is the procedure of applying a function produced by F to its arguments produced by $A_1, \ldots, A_m$ to obtain the value of the function; dually, Closure $[\lambda x_1 \ldots x_m\, C]$ is the procedure of declaring or constructing a function by abstracting over the values of λ-bound variables. As obvious, TIL is a typed λ-calculus that operates on functions (intensional level) and their values (extensional level), as ordinary λ-calculi do; in addition to this dichotomy, there is however the highest *hyperintensional* level of procedures producing lower-level objects. And since these procedures can serve as objects on which other higher-order procedures operate, we can execute procedures twice over. To this end there is the construction called *Double Execution*. To avoid vicious circle problem and keep track in the rich hierarchy of logical strata, TIL ontology is organized into *ramified hierarchy of types* built over a base. For the purpose of natural language processing we use the *epistemic base* consisting of for atomic types, namely o (the set of truth-values), ι (individuals), τ (times or real numbers) and ω (possible worlds). The type of constructions is $*_n$ where n is the order of a construction.

Empirical sentences and terms denote (PWS-)*intensions*, objects of types $((\alpha\tau)\omega)$, or $\alpha_{\tau\omega}$, for short. Where variables w, t range over possible worlds $(w \to \omega)$ and times $(t \to \tau)$, respectively, constructions of intensions are usually Closures of the form $\lambda w \lambda t\, [\ldots w \ldots t \ldots]$. For a simple example, where *Surgeon* is a property of individuals of type $(o\iota)_{\tau\omega}$ and *John* is an individual of type ι, the sentence "John is a surgeon" encodes as its meaning the hyper-proposition

$$\lambda w \lambda t\, [[[{}^0 Surgeon\, w]\, t]\, {}^0 John], \text{ or } \lambda w \lambda t\, [{}^0 Surgeon_{wt}\, {}^0 John], \text{ for short.}$$

Hence, the input information base on which our inference machine operates, is a large collection of such hyper-propositions over which we ask queries.

For instance, given questions like "Is there a surgeon?", "Who is it?" the system derives answers logically entailed by the base like "yes", "he is John". This is very simple, of course. A given question is a *goal* the answer to which the system derives from the knowledge base. Thus a proof-calculus with a *goal-driven strategy* and backward chaining proof method [11, Ch.9] such as the well-known *General Resolution Method* (GRM) seems to be a natural choice here and the technique of *Resolution Theorem Proving* is broadly applied in artificial intelligence.

In [6] we briefly demonstrated application of GRM by the 'Sport Club' example. Since the FOL (first-order predicate logic) general resolution method operates on formulas in their clausal form, we specified the algorithm of transferring hyper-propositions, i.e. closed constructions that are typed to construct propositions, into the clausal form. Yet natural language is semantically much richer than the language of FOL. There are numerous semantic features of natural language that do not appear in a formal, unambiguous logical language. Apart from the problem of high ambiguity, processing natural language must deal with propositional and notional *attitudes*, *property modifiers*, *anaphoric references*, *modalities*, different grammatical *tenses* and *time references*, *definite descriptions* and presuppositions connected with them, other *presupposition triggers* like topic-focus articulation within a sentence, etc etc.

TIL is the system where such semantically salient special features are logically tractable. We have got special technical rules and methods to operate *in* a hyperintensional context, to deal with *de dicto* vs. *de re* attitudes, presupposition triggers, rules for factive attitudes like *knowing* or *regretting*, *partiality* (sentences with truth-value gaps), grammatical tenses and reference times, and so like. These technicalities include, inter alia, the *substitution method*, i.e. application of the special functions that operate on constructions, namely $Sub/(*_n\, *_n\, *_n *_n)$ and $Tr/(*\, \alpha)$ together with Double Execution, properties of propositions like *True*, *False* and *Undef*, transition from a property modifier to a property (pseudo-detachment, i.e. the *left subsectivity* rule [10]), and many others. Compared to this semantic richness, the semantics of FOL formulas in their clausal form is much simpler. Hence, there is a problem how to apply a formal FOL method such as general resolution without losing semantic information encoded in TIL constructions. In previous works ([1], [6]) we assumed that it would be possible to pre-process TIL constructions into the form plausible for the application of the algorithm of transformation into the Skolem clausal form first, and then apply the algorithm so that goal-driven resolution can start.

However, as it turns out, this way is under-inferring. It can be the case that we might derive the respective answer entailed by the knowledge base if only we could harmonically integrate those special TIL rules with the goal-driven resolution process. The goal of this paper and its novel contribution is a proposal of such a method. Using examples, we demonstrate how to deal with property modifiers, anaphoric references, factive propositional attitudes and definite descriptions.

The rest of the paper is organized as follows. Reasoning with property modifiers and anaphoric references is demonstrated by an example involving a 'married man' in Section 2. Section 3 deals with factive propositional attitudes, definite descriptions and anaphoric references. Concluding remarks can be found in Section 4.

2 Reasoning with property modifiers

Scenario. John is a married man. John's partner is Eve. Everybody who is married believes, that his/her partner is amazing.
Question. Does John believe that Eve is amazing?

Formalization starts with assigning *types* to the objects that receive mention in our scenario:
$John, Eve/\iota$; $Married^m/((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$ a property modifier; $Married/(o\iota)_{\tau\omega}$ the property of being married; $Amazing/(o\iota)_{\tau\omega}$; $Partner/(\iota\iota)_{\tau\omega}$; $Believe/(o\iota*)_{\tau\omega}$; $w \rightarrow \omega$; $t \rightarrow \tau$; $x, y \rightarrow \iota$.

The analysis of the three premises and the question comes down to these constructions:

A: $\lambda w \lambda t\, [[{}^0Married^m\, {}^0Man]_{wt}\, {}^0John]$
B: $\lambda w \lambda t\, [[{}^0Partner_{wt}\, {}^0John] = {}^0Eve]$
C: $\lambda w \lambda t\, \forall x\, [[{}^0Married_{wt}\, x] \supset [{}^0Believe_{wt}\, x\, [{}^0Sub\, [{}^0Tr\, [{}^0Partner_{wt}\, x]]\, {}^0y\, {}^0[\lambda w \lambda t\, [{}^0Amazing_{wt}\, y]]]]]]$
Q: $\lambda w \lambda t\, [{}^0Believe_{wt}\, {}^0John\, {}^0[\lambda w \lambda t\, [{}^0Amazing_{wt}\, {}^0Eve]]]$

Note that at this point we cannot adjust the premise (C), i.e. to evaluate the substitution, because this is a general rule. In other words, we do not know as yet which individuals should be substituted for x, and thus also for y.

The algorithm of transferring constructions into their Skolem clausal form [6] starts with the elimination of the left-most $\lambda w \lambda t$, and negation of the question (Q), thus obtaining the first goal (G):

1. Elimination of the left-most $\lambda w \lambda t$, and obtaining the first goal G by negating the question Q.
   A: $[[^0Married^m\ ^0Man]_{wt}\ ^0John]$
   B: $[[^0Partner_{wt}\ ^0John] = {}^0Eve]$
   C: $\forall x\ [[^0Married_{wt}\ x] \supset [^0Believe_{wt}\ x\ [^0Sub\ [^0Tr\ [^0Partner_{wt}\ x]]\ ^0y\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ y]]]]]$
   G: $\neg[^0Believe_{wt}\ ^0John\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ ^0Eve]]]$

2. Elimination of $\supset$
   A: $[[^0Married^m\ ^0Man]_{wr}\ ^0John]$
   B: $[[^0Partner_{wt}\ ^0John] = {}^0Eve]$
   C: $\forall x\ [\neg[^0Married_{wt}\ x] \vee [^0Believe_{wt}\ x\ [^0Sub\ [^0Tr\ [^0Partner_{wt}\ x]]\ ^0y\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ y]]]]]$
   G: $\neg[^0Believe_{wt}\ ^0John\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ ^0Eve]]]$

3. Elimination of $\forall$
   A: $[[^0Married^m\ ^0Man]_{wt}\ ^0John]$
   B: $[[^0Partner_{wt}\ ^0John] = {}^0Eve]$
   C: $\neg[^0Married_{wt}\ x] \vee [^0Believe_{wt}\ x\ [^0Sub\ [^0Tr\ [^0Partner_{wt}\ x]]\ ^0y\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ y]]]]$
   G: $\neg[^0Believe_{wt}\ ^0John\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ ^0Eve]]]$

   Basically, our constructions are in the Skolem clausal form now, though a little bit more complex form. The resolution process should start with the goal G and look for a clause where a positive constituent $[^0Believe_{wt}\ x\ ...]$ occurs. Obviously, it is the clause C. Resolution rule in FOL makes use of Robinson's *unification* algorithm. In principle, this algorithm substitutes terms for variables, which transforms in TIL into the substitution of constituents for variables. Yet, to unify constructions of the arguments of the function produced by $^0Believe_{wt}$ as they occur in the clauses G and C, it is not sufficient to substitute the constituent ^0John for the variable x. In addition, we have to exploit the clause B and substitute ^0Eve for $[^0Partner_{wt}\ ^0John]$. As a result, we obtain

4. Unification and Resolution

   $$[^0Believe_{wt}\ ^0John\ [^0Sub\ [^0Tr\ ^0Eve]\ ^0y\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ y]]]]$$

   Next, the application of the functions *Sub* and *Tr* must be evaluated by applying these transformations:

   $$[^0Tr\ ^0Eve] \implies {}^0Eve$$
   $$[^0Sub\ [^0Tr\ ^0Eve]\ ^0y\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ y]]] \implies {}^0[\lambda w \lambda t\ [^0Amazing_{wt}\ ^0Eve]]$$

   As a result, we obtain the adjusted clause

   C′: $\neg[^0Married_{wt}\ ^0John] \vee [^0Believe_{wt}\ ^0John\ ^0[\lambda w \lambda t\ [^0Amazing_{wt}\ ^0Eve]]]$

Only now can the clauses G and C′ be resolved so that the new goal is obtained:

R1: $\neg[{}^0Married_{wt}\ {}^0John]$ (G+C′)

However, this goal cannot be met, unless the rule of *left subsectivity* ([10], [9], [2]) that is universally valid for any kind of a property modifier is applied. Where $P^m \to ((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$ is a construction of a property modifier and $P \to (o\iota)_{\tau\omega}$ a construction of the property corresponding to the modifier, the rule is this:[1]

$$[[P^m Q]_{wt}\ x] \vdash [P_{wt}\ x]$$

In our case the rule results in $[[{}^0Married^m\ {}^0Man]_{wt}\ x] \vdash [{}^0Married_{wt}\ x]$. For the purpose of resolution method, we rewrite the rule into the implicative, hence clausal form, thus obtaining another clause

M: $\neg[[{}^0Married^m\ {}^0Man]_{wt}\ z] \vee [{}^0Married_{wt}\ z]$

Now the last goal R1 is easily met:

R2: $\neg[[{}^0Married^m\ {}^0Man]_{wt}\ {}^0John]$ (R1+M), ${}^0John/z$
R3: # (R2+A)

By applying an indirect proof we obtained the empty clause that cannot be satisfied, hence the answer to the question Q is YES.

By this simple example we demonstrated that it is not possible to evaluate constructions stemming from the special TIL techniques like the anaphoric substitution by means of the functions *Sub* and *Tr* in the phase of pre-processing constructions and their transformation into the Skolem clausal form. Rather, we have to integrate these techniques with the unification of clauses into the process of deriving resolvents. In addition, we also have to apply special rules that are rooted in the rich semantics of natural language like the rule of left subsectivity. We propose to specify such rules in the form of additional semantic clauses that are recorded in agent's ontology.

3 Reasoning with factive propositional attitudes

In this section we are going to demonstrate by an example reasoning with factive propositional attitudes like 'knowing that' for which special rules rendering the fact that the truth of the known sub-proposition is a presupposition of the whole proposition. In other words, if a proposition *P* is not true, then *P* can be neither

[1] The rigorous definition of the property corresponding to the respective modifier can be found in [2]; roughly, *P* is defined as the property of *x* such that there is a property *q* with respect to which *x* is a $[P^m q]$. For instance, a skillful surgeon is skillful as a surgeon. Hence, there is a property with respect to which a skillful surgeon is skillful.

known nor not known. Thus, the rules that we have to apply in this example are specified as follows.

$$[^0Know_{wt}\ a\ C] \vdash [^0True_{wt}\ ^2C]$$
$$\neg[^0Know_{wt}\ a\ C] \vdash [^0True_{wt}\ ^2C]$$

Types. $Know \rightarrow\ (o\iota *_n)_{\tau\omega};\ a \rightarrow \iota;\ C \rightarrow *_n;\ ^2C \rightarrow o_{\tau\omega};\ True/(oo_{\tau\omega})_{\tau\omega}$.

In addition to these rules, we often need to apply the rule of *True elimination*:

$$[^0True_{wt}\ p] \vdash p_{wt}$$

Similarly as in the previous example, the above rules will be specified in their implicative form so that we obtain three additional clauses. In the resolution process we also make technical adjustments, in particular by applying the rule of ^{20}-*conversion*:

$$^{20}C = C$$

for any closed construction C that is typed to v-construct a non-procedural object of a type of order 1.

Scenario. The Mayor of Ostrava knows that the President of TUO does not know (yet) that he (the President) will go to Brussels. The President of TUO is prof. Snasel.
Question. Will prof. Snasel go to Brussels?

Formalization.
As always, first types: *Snasel*, *Brussels*$/\iota$; *Know*$/(o\iota *_n)_{\tau\omega}$; *President*(-of TUO), *Mayor*(-of Ostrava)$/\iota_{\tau\omega}$; *Go*$/(o\iota\iota)_{\tau\omega}$.

Premises:
A: $\lambda w\lambda t\ [^0Know_{wt}\ ^0Mayor_{wt}\ ^0[\lambda w\lambda t\ \neg[^0Know_{wt}\ ^0President_{wt}$
$[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w\lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]]]]$
B: $\lambda w\lambda t\ [^0President_{wt} =\ ^0Snasel]$
Conclusion/question:
Q: $\lambda w\lambda t\ [^0Go_{wt}\ ^0Snasel\ ^0Brussels]$

In addition to these premises and conclusion, we have the above rules which result into three clauses M1, M2 and T:

M1: $[^0Know_{wt}\ x\ c] \supset [^0True_{wt}\ ^2c]$
M2: $\neg[^0Know_{wt}\ x\ c] \supset [^0True_{wt}\ ^2c]$
T: $[^0True_{wt}\ p] \supset p_{wt}$

Additional types. $c \rightarrow *_n;\ ^2c \rightarrow o_{\tau\omega};\ p \rightarrow o_{\tau\omega}$

The algorithm of transferring these constructions into their clausal form proceeds as follows:

1. Elimination of the left-most $\lambda w \lambda t$, renaming variables, negation of the question Q that results in the goal G.
   A: $[{}^0Know_{wt}\, {}^0Mayor_{wt}\, {}^0[\lambda w \lambda t\, \neg[{}^0Know_{wt}\, {}^0President_{wt}$
   $[{}^0Sub\, [{}^0Tr\, {}^0President_{wt}]\, {}^0he\, {}^0[\lambda w \lambda t\, [{}^0Go_{wt}\, he\, {}^0Brussels]]]]]]$
   B: $[{}^0President_{wt} = {}^0Snasel]$
   M1: $[{}^0Know_{wt}\, x\, c] \supset [{}^0True_{wt}\, {}^2c]$
   M2: $\neg[{}^0Know_{wt}\, y\, d] \supset [{}^0True_{wt}\, {}^2d]$
   T: $[{}^0True_{wt}\, p] \supset p_{wt}$
   G: $\neg[{}^0Go_{wt}\, {}^0Snasel\, {}^0Brussels]$

2. Elimination of $\supset$
   A: $[{}^0Know_{wt}\, {}^0Mayor_{wt}\, {}^0[\lambda w \lambda t\, \neg[{}^0Know_{wt}\, {}^0President_{wt}$
   $[{}^0Sub\, [{}^0Tr\, {}^0President_{wt}]\, {}^0he\, {}^0[\lambda w \lambda t\, [{}^0Go_{wt}\, he\, {}^0Brussels]]]]]]$
   B: $[{}^0President_{wt} = {}^0Snasel]$
   M1: $\neg[{}^0Know_{wt}\, x\, c] \vee [{}^0True_{wt}\, {}^2c]$
   M2: $[{}^0Know_{wt}\, y\, d] \vee [{}^0True_{wt}\, {}^2d]$
   T: $\neg[{}^0True_{wt}\, p] \vee p_{wt}$
   G: $\neg[{}^0Go_{wt}\, {}^0Snasel\, {}^0Brussels]$

   Our constructions (hyperpropositions) are in the Skolem clausal form now, and the process of resolution together with unification can start. Since the strategy is goal-driven, we aim at choosing a clause with a positive constituent $[{}^0Go_{wt}...]$. The only candidate is the clause A. However, there is a problem here. The constituent ${}^0Go$ occurs in the goal G extensionally, while the same constituent occurs in A in the hyperintensional context, i.e. closed by Trivialization and thus not amenable to logical operations.[2] Yet, the 'magic trick' of this argument consists in the fact that *Knowing* is a factivum. In other words, by applying the rules M1, M2 and T we can decrease the context down to the extensional level. This in turn means that before exploiting the goal G we have to make a 'step aside'. Before resolving the goal G with any other clause, we must resolve A, M1, M2 and T until the constituent ${}^0Go$ gets down to the extensional level. Here is how.

3. Unification and resolution
   R1: $[{}^0True_{wt}\, {}^{20}[\lambda w \lambda t\, \neg[{}^0Know_{wt}\, {}^0President_{wt}$
   $[{}^0Sub\, [{}^0Tr\, {}^0President_{wt}]\, {}^0he\, {}^0[\lambda w \lambda t\, [{}^0Go_{wt}\, he\, {}^0Brussels]]]]]]$
   (A+M1)
   ${}^0Mayor_{wt}/x,$
   ${}^0[\lambda w \lambda t\, \neg[{}^0Know_{wt}\, {}^0President_{wt}\, [{}^0Sub\, [{}^0Tr\, {}^0President_{wt}]\, {}^0he$
   ${}^0[\lambda w \lambda t\, [{}^0Go_{wt}\, he\, {}^0Brussels]]]]]/c$

[2] For details on the three kinds of context in which a construction can occur within another construction, see [4], and the details on hyperintensionally closed constructions can be found in [3] and [8].

R2: $^{20}[\lambda w \lambda t\ \neg[^0Know_{wt}\ ^0President_{wt}$
$[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]]]_{wt}$

(R1+T)
$^{20}[\lambda w \lambda t\ \neg[^0Know_{wt}\ ^0President_{wt}$
$[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]]]/p$

R3: $[^0True_{wt}\ ^2[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]]$

(R2+M2)
^{20}-conversion,
restricted β-conversion,
$^0President_{wt}/y$,
$[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]/d$

R4: $^2[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]_{wt}$

(R3+T)
$^2[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t[^0Go_{wt}\ he\ ^0Brussels]]]/p$

To evaluate the substitution, i.e. to obtain the proposition v-constructed by R4, we make use of the clause B in order to substitute ^0Snasel for $^0President_{wt}$. Thus, we have

$^2[^0Sub\ [^0Tr\ ^0President_{wt}]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]_{wt} =$ R4, B
$^2[^0Sub\ [^0Tr\ ^0Snasel]\ ^0he\ ^0[\lambda w \lambda t\ [^0Go_{wt}\ he\ ^0Brussels]]]_{wt} =$ *Sub*, *Tr*
$^{20}[\lambda w \lambda t\ [^0Go_{wt}\ ^0Snasel\ ^0Brussels]]_{wt} =$ ^{20}-conversion, β
$[^0Go_{wt}\ ^0Snasel\ ^0Brussels]$

As a result, we obtained an adjusted clause
R4’: $[^0Go_{wt}\ ^0Snasel\ ^0Brussels]$
R5: # (R4’+G)

Hence, the answer to the question Q is YES.

By this example we demonstrated that though our strategy is goal-driven, we cannot proceed strictly from a goal to meet another goal. At the beginning of the resolution process, we used the goal G to determine the clause A as the one that might be suitable for meeting the goal G. However, to do so, we first had to resolve the clause A with other clauses (M1, M2, T), then adjust the result by means of the clause B and the special TIL technical rules, and only then could we resolve the result with our goal G to obtain an empty clause, and thus answer the question Q in positive.

Summarising, an adjustment of the general resolution method for natural language processing in TIL *cannot be strictly goal-driven*, which is another novel result of this paper.

4 Conclusion

In the paper we introduced reasoning with fine-grained semantics of natural language in the question-answer system based on Transparent Intensional Logic. We examined application of the General Resolution Method with its goal-driven strategy that is also characterized as backward chaining, because a given goal determines which clauses are selected and used for generating resolvents. Backward chaining starts with a goal (question or hypothesis) and works backwards from the consequent to the antecedent to see if any clause supports any of these consequents. We solved two problems. First, how to integrate special rules rooted in the rich natural language semantics with the process of generating resolvents. Second, we found out that due to these special rules the process cannot be strictly goal-driven; it starts with a given goal/question, yet it may happen that we have to make a 'step aside' in order to adjust other clauses first, and only then we can resolve.

Future research will be oriented to forward-chaining inference, as it is applied in the Gentzen natural deduction system. In such a system there is not a problem of smoothly integrating other rules as needed; rather, we will have to solve the problem how to answer a given question and not to get lost in the huge labyrinth of input data. Last but not least, we would eventually like to make a comparison of such two different approaches to the design of an inference machine for natural language processing using TIL.

Acknowledgements. This research was funded by the Grant Agency of the Czech Republic (GACR) project GA18-23891S "Hyperintensional Reasoning over Natural Language Text" and by the internal grant agency of VSB-Technical University of Ostrava, project No. SP2019/40, "Application of Formal Methods in Knowledge Modelling and Software Engineering I". Michal Fait was also supported by the Moravian-Silesian regional program No. RRC/10/2017 "Support of science and research in Moravian-Silesian region 2017" and by the EU project "Science without borders" No. CZ.02.2.69/0.0/0.0/16_027/0008463.

References

1. Číhalová, M., Duží, M., Ciprich, N., Menšík, M. (2010), 'Agents' reasoning using TIL-Script and Prolog', *Frontiers in Artificial Intelligence and Applications* 206: 135-154.
2. Duží, M. (2017): Property modifiers and intensional essentialism. *Computación y Sistemas*, vol. 21, No. 4, 2017, pp. 601–613. DOI: 10.13053/CyS-21-4-2811.
3. Duží, M. (2019): If structured propositions are logical procedures then how are procedures individuated? *Synthese* special issue on the Unity of propositions, vol. 196, No. 4, pp. 1249-1283. DOI: 10.1007/s11229-017-1595-5
4. Duží, M., Fait, M., Menšík, M. (2017): Context Recognition for a Hyperintensional Inference Machine. In the AIP proceeding of *ICNAAM 2016*, International Conference of Numerical Analysis and Applied Mathematics, vol. 1863, Article No. 330004
5. Duží, M., Horák, A. (2015): TIL as hyperintensional logic for natural language analysis. In *RASLAN 2015*, A. Horák, P. Rychlý and A. Rambousek (eds.), pp. 113-124.

6. Duží, M., Horák, A. (2019): Hyperintensional Reasoning based on Natural Language Analysis. To appear in the *International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems (IJUFKS)*.
7. Duží, M., Menšík, M. (2017): Logic of Inferable Knowledge. In Jaakkola, H., Thalheim, B., Kiyoki, Y. and Yoshida, N., eds., Information Modelling and Knowledge Bases XXVIII, *Frontiers in Artificial Intelligence and Applications*, Amsterdam: IOS Press, vol. 292, 2017, pp. 405-425.
8. Fait, M., Duží, M. (2019): Substitution rules with respect to a context. In *Lecture Notes in Electrical Engineering* 554, I. Zelinka, P. Brandstetter, T.T. Dao, Vo. H. Duy, S.B. Kim eds., AETA 2018 – Recent Advances in Electrical Engineering and Related Sciences: Theory and Applications, Springer, pp. 55-66.
9. Jespersen, B. (2015): Structured lexical concepts, property modifiers, and Transparent Intensional Logic. *Philosophical Studies*, vol. 172, No. 2, pp. 321-345.
10. Jespersen, B. (2016): Left Subsectivity: How to Infer that a Round Peg is Round. *Dialectica*, vol.70, No. 4, pp. 531-547.
11. Russell, S., Norvig, P. (2010): *Artificial Intelligence: A Modern Approach*. Person, 3rd edition.