# VŠB TECHNICKÁ

# VSB TECHNICAL



#### www.vsb.cz

# Adjustment of goal-driven resolution for natural language processing in TIL

#### Marie Duží, Michal Fait, Marek Menšík

VSB – Technical University of Ostrava marie.duzi@vsb.cz, michal.fait@vsb.cz, marek.mensik@vsb.cz

December 7, 2019

VSB TECHNICAL | FACULTY OF ELECTRICAL UNIVERSITY | ENGINEERING AND COMPUTER OF OSTRAVA | SCIENCE





- 2 Reasoning with property modifiers
- 3 Reasoning with factive propositional attitudes

#### 4 Conclusions



- 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.
- We must deal with semantic rules concerning attitudes, property modifiers, anaphoric references, modalities, different grammatical tenses etc., and also with special technical rules of TIL like methods to operate *in* a hyperintensional context, special functions that operate on constructions, namely  $Sub/(*_n *_n *_n *_n)$  and  $Tr/(*_n \alpha)$ together with Double Execution, properties of propositions like *True*, *False* etc.



- 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.
- We must deal with semantic rules concerning attitudes, property modifiers, anaphoric references, modalities, different grammatical tenses etc., and also with special technical rules of TIL like methods to operate in a hyperintensional context, special functions that operate on constructions, namely  $Sub/(*_n *_n *_n *_n)$  and  $Tr/(*_n \alpha)$  together with Double Execution, properties of propositions like *True*, *False* etc.



- In previous works we assumed that it would be possible to *pre-process* TIL constructions (with respect to the above mentioned special rules) prior to the process of applying the algorithm of transformation into the Skolem clausal form and goal-driven resolution.
- 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.



- In previous works we assumed that it would be possible to *pre-process* TIL constructions (with respect to the above mentioned special rules) prior to the process of applying the algorithm of transformation into the Skolem clausal form and goal-driven resolution.
- 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.





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?

Types: 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;  $Amazing, Man/(o\iota)_{\tau\omega}$ ;  $Partner/(\iota\iota)_{\tau\omega}$ ;  $Believe/(o\iota*_n)_{\tau\omega}$ ;  $w \to \omega$ ;  $t \to \tau$ ;  $x, y \to \iota$ .

Premises:

$$\begin{split} \lambda w \lambda t \, [[{}^{0}Married^{m} \, {}^{0}Man]_{wt} \, {}^{0}John] \\ \lambda w \lambda t \, [[{}^{0}Partner_{wt} \, {}^{0}John] = \, {}^{0}Eve] \\ \lambda w \lambda t \, \forall x \, [[{}^{0}Married_{wt} \, x] \supset [{}^{0}Believe_{wt} \, x \, [{}^{0}Sub \, [{}^{0}Tr \, [{}^{0}Partner_{wt} \, x]] \, {}^{0}y \\ & {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} \, y]]]]] \end{split}$$

Conclusion/question:  $\lambda w \lambda t [^{0}Believe_{wt} \,^{0}John \,^{0}[\lambda w \lambda t \, [^{0}Amazing_{wt} \,^{0}Eve]]]$ 

IμI

**Types:** 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;  $Amazing, Man/(o\iota)_{\tau\omega}$ ;  $Partner/(\iota\iota)_{\tau\omega}$ ;  $Believe/(o\iota*_n)_{\tau\omega}$ ;  $w \to \omega$ ;  $t \to \tau$ ;  $x, y \to \iota$ .

#### Premises:

$$\begin{split} \lambda w \lambda t \, [[{}^{0}Married^{m} \, {}^{0}Man]_{wt} \, {}^{0}John] \\ \lambda w \lambda t \, [[{}^{0}Partner_{wt} \, {}^{0}John] = \, {}^{0}Eve] \\ \lambda w \lambda t \, \forall x \, [[{}^{0}Married_{wt} \, x] \supset [{}^{0}Believe_{wt} \, x \, [{}^{0}Sub \, [{}^{0}Tr \, [{}^{0}Partner_{wt} \, x]] \, {}^{0}y \\ & {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} \, y]]]]] \end{split}$$

Conclusion/question:  $\lambda w \lambda t [^{0}Believe_{wt} \, ^{0}John \, ^{0}[\lambda w \lambda t \, [^{0}Amazing_{wt} \, ^{0}Eve]]]$ 

կլլ

**Types:** 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;  $Amazing, Man/(o\iota)_{\tau\omega}$ ;  $Partner/(\iota\iota)_{\tau\omega}$ ;  $Believe/(o\iota*_n)_{\tau\omega}$ ;  $w \to \omega$ ;  $t \to \tau$ ;  $x, y \to \iota$ .

#### Premises:

$$\begin{split} \lambda w \lambda t \, [[{}^{0}Married^{m} \, {}^{0}Man]_{wt} \, {}^{0}John] \\ \lambda w \lambda t \, [[{}^{0}Partner_{wt} \, {}^{0}John] = \, {}^{0}Eve] \\ \lambda w \lambda t \, \forall x \, [[{}^{0}Married_{wt} \, x] \supset [{}^{0}Believe_{wt} \, x \, [{}^{0}Sub \, [{}^{0}Tr \, [{}^{0}Partner_{wt} \, x]] \, {}^{0}y \\ & {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} \, y]]]]] \end{split}$$

#### Conclusion/question: $\lambda w \lambda t [^{0}Believe_{wt} \ ^{0}John \ ^{0}[\lambda w \lambda t \ [^{0}Amazing_{wt} \ ^{0}Eve]]]$

կլլ

**Types:** 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;  $Amazing, Man/(o\iota)_{\tau\omega}$ ;  $Partner/(\iota\iota)_{\tau\omega}$ ;  $Believe/(o\iota*_n)_{\tau\omega}$ ;  $w \to \omega$ ;  $t \to \tau$ ;  $x, y \to \iota$ .

#### Premises:

$$\begin{split} \lambda w \lambda t \, [[{}^{0}Married^{m} \, {}^{0}Man]_{wt} \, {}^{0}John] \\ \lambda w \lambda t \, [[{}^{0}Partner_{wt} \, {}^{0}John] = \, {}^{0}Eve] \\ \lambda w \lambda t \, \forall x \, [[{}^{0}Married_{wt} \, x] \supset [{}^{0}Believe_{wt} \, x \, [{}^{0}Sub \, [{}^{0}Tr \, [{}^{0}Partner_{wt} \, x]] \, {}^{0}y \\ & {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} \, y]]]]] \end{split}$$

Conclusion/question:  $\lambda w \lambda t [^{0}Believe_{wt} \,^{0}John \,^{0}[\lambda w \lambda t \, [^{0}Amazing_{wt} \,^{0}Eve]]]$ 

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

# G: $\neg [^{0}Believe_{wt} \, ^{0}John \, ^{0}[\lambda w\lambda t \, [^{0}Amazing_{wt} \, ^{0}Eve]]]$

The resolution process should start with the goal G and look for a clause where a positive constituent  $[{}^{0}Believe_{wt} x \dots]$  occurs. Obviously, it is the clause C.

In TIL, *unification* consists in substituting constituents for variables. Yet, to unify constructions of the arguments of the function produced by  ${}^{0}Believe_{wt}$  as they occur in the clauses G and C, it is not sufficient to substitute the constituent  ${}^{0}John$  for the variable x.

 $\begin{array}{l} \mathsf{A:} \ [[{}^{0}Married^{m} {}^{0}Man]_{wt} {}^{0}John] \\ \mathsf{B:} \ [[{}^{0}Partner_{wt} {}^{0}John] = {}^{0}Eve] \\ \mathsf{C:} \ \neg [{}^{0}Married_{wt} x] \lor [{}^{0}Believe_{wt} x \, [{}^{0}Sub \, [{}^{0}Tr \, [{}^{0}Partner_{wt} x]] \, {}^{0}y \\ {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} y]]]] \\ \mathsf{G:} \ \neg [{}^{0}Believe_{wt} \, {}^{0}John \, {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} \, {}^{0}Eve]]] \end{array}$ 

The resolution process should start with the goal G and look for a clause where a positive constituent  $[{}^0\!Believe_{wt}\,x\,\ldots]$  occurs. Obviously, it is the clause C.

In TIL, *unification* consists in substituting constituents for variables. Yet, to unify constructions of the arguments of the function produced by  ${}^{0}Believe_{wt}$  as they occur in the clauses G and C, it is not sufficient to substitute the constituent  ${}^{0}John$  for the variable x.

 $\begin{array}{l} \mathsf{A:} \ [[{}^{0}Married^{m} {}^{0}Man]_{wt} {}^{0}John] \\ \mathsf{B:} \ [[{}^{0}Partner_{wt} {}^{0}John] = {}^{0}Eve] \\ \mathsf{C:} \ \neg [{}^{0}Married_{wt} x] \lor [{}^{0}Believe_{wt} x \, [{}^{0}Sub \, [{}^{0}Tr \, [{}^{0}Partner_{wt} x]] \, {}^{0}y \\ {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} y]]]] \\ \mathsf{G:} \ \neg [{}^{0}Believe_{wt} \, {}^{0}John \, {}^{0}[\lambda w \lambda t \, [{}^{0}Amazing_{wt} \, {}^{0}Eve]]] \end{array}$ 

The resolution process should start with the goal G and look for a clause where a positive constituent  $[{}^{0}Believe_{wt} x \dots]$  occurs. Obviously, it is the clause C.

In TIL, *unification* consists in substituting constituents for variables. Yet, to unify constructions of the arguments of the function produced by  ${}^{0}Believe_{wt}$  as they occur in the clauses G and C, it is not sufficient to substitute the constituent  ${}^{0}John$  for the variable x.

կլլ

In addition, we have to exploit the clause B and substitute  ${}^0\!Eve$  for  $[{}^0\!Partner_{wt}\; {}^0\!John].$  As a result, we obtain

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

Next, the application of the functions *Sub* and *Tr* must be evaluated. As a result, we obtain the adjusted clause:

 $\mathsf{C}':\neg[{}^{0}Married_{wt} \, {}^{0}John] \lor [{}^{0}Believe_{wt} \, {}^{0}John \, {}^{0}[\lambda w\lambda t \, [{}^{0}Amazing_{wt} \, {}^{0}Eve]]]$ 

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

R1:  $\neg$ [<sup>0</sup>*Married*<sub>wt</sub> <sup>0</sup>*John*]

կլլ

In addition, we have to exploit the clause B and substitute  ${}^0\!Eve$  for  $[{}^0\!Partner_{wt} \, {}^0\!John]$ . As a result, we obtain

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

Next, the application of the functions Sub and Tr must be evaluated. As a result, we obtain the adjusted clause:

 $\mathsf{C'}:\neg[{}^{0}\!Married_{wt}\,{}^{0}\!John] \vee [{}^{0}\!Believe_{wt}\,{}^{0}\!John\,{}^{0}[\lambda w\lambda t\,[{}^{0}\!Amazing_{wt}\,{}^{0}\!Eve]]]$ 

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

R1:  $\neg$ [<sup>0</sup>*Married<sub>wt</sub>* <sup>0</sup>*John*]

կլլ

In addition, we have to exploit the clause B and substitute  ${}^0\!Eve$  for  $[{}^0\!Partner_{wt} \, {}^0\!John]$ . As a result, we obtain

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

Next, the application of the functions Sub and Tr must be evaluated. As a result, we obtain the adjusted clause:

 $\mathsf{C'}:\neg[{}^{0}\!Married_{wt}\,{}^{0}\!John] \vee [{}^{0}\!Believe_{wt}\,{}^{0}\!John\,{}^{0}[\lambda w\lambda t\,[{}^{0}\!Amazing_{wt}\,{}^{0}\!Eve]]]$ 

Only now can the clauses G and  $C^{\prime}$  be resolved so that the new goal is obtained:

R1:  $\neg [^{0}Married_{wt} ^{0}John]$ 

(G+C')

However, this goal *cannot* be met, unless the rule of *left subsectivity* that is universally valid for any kind of a property modifier is applied. In our case the rule results in  $[[^{0}Married^{m} {}^{0}Man]_{wt} x] \vdash [^{0}Married_{wt} x]$ . For the purpose of resolution method, we rewrite the rule into the implicative, hence clausal form, thus obtaining another clause

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

Now the last goal R1 is easily met:

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

However, this goal *cannot* be met, unless the rule of *left subsectivity* that is universally valid for any kind of a property modifier is applied. In our case the rule results in  $[[^{0}Married^{m \ 0}Man]_{wt} x] \vdash [^{0}Married_{wt} x]$ . For the purpose of resolution method, we rewrite the rule into the implicative, hence clausal form, thus obtaining another clause

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

Now the last goal R1 is easily met:

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

 R3: #
 (R2+A)

However, this goal *cannot* be met, unless the rule of *left subsectivity* that is universally valid for any kind of a property modifier is applied. In our case the rule results in  $[[^{0}Married^{m} \, ^{0}Man]_{wt} \, x] \vdash [^{0}Married_{wt} \, x]$ . For the purpose of resolution method, we rewrite the rule into the implicative, hence clausal form, thus obtaining another clause

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

Now the last goal R1 is easily met:

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

 R3: #
 (R2+A)

However, this goal *cannot* be met, unless the rule of *left subsectivity* that is universally valid for any kind of a property modifier is applied. In our case the rule results in  $[[^{0}Married^{m} {}^{0}Man]_{wt} x] \vdash [^{0}Married_{wt} x]$ . For the purpose of resolution method, we rewrite the rule into the implicative, hence clausal form, thus obtaining another clause

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

Now the last goal R1 is easily met:

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

Factive propositional attitudes

The truth of the known sub-proposition is a *presupposition* of the whole proposition.

$$\begin{split} [{}^{0}Know_{wt} \, a \, c] &\vdash [{}^{0}True_{wt} \, {}^{2}c] \\ \neg [{}^{0}Know_{wt} \, a \, c] \vdash [{}^{0}True_{wt} \, {}^{2}c] \end{split}$$
Types.  $Know/(ol*_{n})_{\tau\omega}; \, a \to l; \, c \to *_{n}; \, {}^{2}c \to o_{\tau\omega}; \, True/(oo_{\tau\omega})_{\tau\omega}. \end{split}$ 

# Additional special rules



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

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

In the resolution process we also make technical adjustments, in particular by applying the rule of  $^{20}\mathchar`-conversion$ :

$$^{20}C = C$$

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?

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

 $\begin{array}{l} \lambda w \lambda t \, [{}^{0}\!K now_{wt} \, {}^{0}\!M ayor_{wt} \, {}^{0}\![\lambda w \lambda t \, \neg [{}^{0}\!K now_{wt} \, {}^{0}\!President_{wt} \\ \\ [{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}\![\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]]] \\ \lambda w \lambda t \, [{}^{0}\!President_{wt} = \, {}^{0}\!Snasel] \end{array}$ 

Conclusion/question:  $\lambda w \lambda t \ [{}^{0}Go_{wt} \, {}^{0}Snasel \, {}^{0}Brussels]$ 

In addition to these premises and conclusion, we have the above rules in their implicative form so that we obtain three additional clauses:

```
\begin{split} [{}^{0}Know_{wt} \, x \, c] &\supset [{}^{0}True_{wt} \, {}^{2}c] \\ \neg [{}^{0}Know_{wt} \, x \, c] &\supset [{}^{0}True_{wt} \, {}^{2}c] \\ [{}^{0}True_{wt} \, p] &\supset p_{wt} \end{split}
```

կլլ

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:

$$\begin{split} \lambda w \lambda t \, [{}^{0}\!Know_{wt} \, {}^{0}\!Mayor_{wt} \, {}^{0}\![\lambda w \lambda t \, \neg [{}^{0}\!Know_{wt} \, {}^{0}\!President_{wt} \\ [{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}\![\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]]]] \\ \lambda w \lambda t \, [{}^{0}\!President_{wt} = \, {}^{0}\!Snasel] \end{split}$$

Conclusion/question:  $\lambda w \lambda t \ [^0Go_{wt}\ ^0Snasel\ ^0Brussels]$ 

In addition to these premises and conclusion, we have the above rules in their implicative form so that we obtain three additional clauses:

```
\begin{split} [{}^{0}Know_{wt} \, x \, c] &\supset [{}^{0}True_{wt} \, {}^{2}c] \\ \neg [{}^{0}Know_{wt} \, x \, c] &\supset [{}^{0}True_{wt} \, {}^{2}c] \\ [{}^{0}True_{wt} \, p] &\supset p_{wt} \end{split}
```

կլլ

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:

$$\begin{split} \lambda w \lambda t \, [{}^{0}\!Know_{wt} \, {}^{0}\!Mayor_{wt} \, {}^{0}\![\lambda w \lambda t \, \neg [{}^{0}\!Know_{wt} \, {}^{0}\!President_{wt} \\ [{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}\![\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]]]] \\ \lambda w \lambda t \, [{}^{0}\!President_{wt} = \, {}^{0}\!Snasel] \end{split}$$

#### Conclusion/question: $\lambda w \lambda t \ [{}^{0}Go_{wt} \ {}^{0}Snasel \ {}^{0}Brussels]$

In addition to these premises and conclusion, we have the above rules in their implicative form so that we obtain three additional clauses:

$$\begin{split} [{}^{0}Know_{wt} \, x \, c] &\supset [{}^{0}True_{wt} \, {}^{2}c] \\ \neg [{}^{0}Know_{wt} \, x \, c] &\supset [{}^{0}True_{wt} \, {}^{2}c] \\ [{}^{0}True_{wt} \, p] &\supset p_{wt} \end{split}$$

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

 $\begin{array}{l} \text{Premises:} \\ \lambda w \lambda t \, [{}^{0}\!K now_{wt} \, {}^{0}\!M ayor_{wt} \, {}^{0}\![\lambda w \lambda t \, \neg [{}^{0}\!K now_{wt} \, {}^{0}\!President_{wt} \\ \\ [{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}\![\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]]] \\ \lambda w \lambda t \, [{}^{0}\!President_{wt} = \, {}^{0}\!Snasel] \end{array}$ 

Conclusion/question:  $\lambda w \lambda t \ [{}^{0}Go_{wt} \, {}^{0}Snasel \, {}^{0}Brussels]$ 

In addition to these premises and conclusion, we have the above rules in their implicative form so that we obtain three additional clauses:

$$\begin{split} & [{}^{0}\!Know_{wt}\,x\,c] \supset [{}^{0}\!True_{wt}\,{}^{2}\!c] \\ & \neg [{}^{0}\!Know_{wt}\,x\,c] \supset [{}^{0}\!True_{wt}\,{}^{2}\!c] \\ & [{}^{0}\!True_{wt}\,p] \supset p_{wt} \end{split}$$

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

Since the strategy is goal-driven, we aim at choosing a clause with a positive constituent  $[{}^{0}Go_{wt}...]$ . The only candidate is the clause A. However, there is a problem here. The constituent  ${}^{0}Go$  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.

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

Since the strategy is goal-driven, we aim at choosing a clause with a positive constituent  $[{}^{0}Go_{wt}...]$ . The only candidate is the clause A. However, there is a problem here. The constituent  ${}^{0}Go$  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.

կլլ

Before resolving the goal G with any other clause, we must resolve A, M1, M2 and T until the constituent  ${}^{0}Go$  gets down to the extensional level. Here is how.

կլլ

Before resolving the goal G with any other clause, we must resolve A, M1, M2 and T until the constituent  ${}^{0}Go$  gets down to the extensional level. Here is how.

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

կլլ

Before resolving the goal G with any other clause, we must resolve A, M1, M2 and T until the constituent  ${}^{0}Go$  gets down to the extensional level. Here is how.

 $\begin{array}{c} \mathsf{R3:} [{}^{0}\!True_{wt} \; {}^{2}[{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}[\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]] \\ & (\mathsf{R2+M2}) \\ {}^{20}\text{-conversion,} \\ restricted \, \beta\text{-conversion,} \\ {}^{0}\!President_{wt}/y, \\ [{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}[\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]/d \end{array}$ 

In order to resolve R2 a M2 to get R3, we had to apply  $^{20}$ - and restricted  $\beta$ -conversion in addition to the substitution of constructions for variables.

ηд

 $\begin{array}{c} \mathsf{R3:} [{}^{0}True_{wt} \; {}^{2}[{}^{0}Sub \, [{}^{0}Tr \, {}^{0}President_{wt}] \, {}^{0}he \, {}^{0}[\lambda w \lambda t \, [{}^{0}Go_{wt} \, he \, {}^{0}Brussels]]]] \\ & (\mathsf{R2+M2}) \\ {}^{20}\text{-conversion,} \\ & \text{restricted} \; \beta\text{-conversion,} \\ {}^{0}President_{wt} / y, \\ [{}^{0}Sub \, [{}^{0}Tr \, {}^{0}President_{wt}] \, {}^{0}he \, {}^{0}[\lambda w \lambda t \, [{}^{0}Go_{wt} \, he \, {}^{0}Brussels]]]/d \end{array}$ 

In order to resolve R2 a M2 to get R3, we had to apply  $^{20}\text{-}$  and restricted  $\beta\text{-}conversion}$  in addition to the substitution of constructions for variables.

ηд

 $\begin{array}{c} \mathsf{R3:} [{}^{0}\!True_{wt} \; {}^{2}[{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}[\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]] \\ & (\mathsf{R2+M2}) \\ {}^{20}\text{-conversion,} \\ restricted \, \beta\text{-conversion,} \\ {}^{0}\!President_{wt}/y, \\ [{}^{0}\!Sub \, [{}^{0}\!Tr \, {}^{0}\!President_{wt}] \, {}^{0}\!he \, {}^{0}[\lambda w \lambda t \, [{}^{0}\!Go_{wt} \, he \, {}^{0}\!Brussels]]]/d \end{array}$ 

In order to resolve R2 a M2 to get R3, we had to apply  $^{20}\text{-}$  and restricted  $\beta\text{-conversion}$  in addition to the substitution of constructions for variables.

ηд

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  ${}^{0}Snasel$  for  ${}^{0}President_{wt}$ . Thus, we have

As a result, we obtained an adjusted clause R4':  $[{}^{0}Go_{wt} {}^{0}Snasel {}^{0}Brussels]$ R5: # Hence, the answer to the question Q is YES.

(R4' + G)

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 <sup>0</sup>Snasel for <sup>0</sup>President<sub>wt</sub>. Thus, we have

As a result, we obtained an adjusted clause R4': [ ${}^{0}Go_{wt} {}^{0}Snasel {}^{0}Brussels$ ] R5: # Hence, the answer to the guestion Q is YES. 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  ${}^{0}Snasel$  for  ${}^{0}President_{mt}$ . Thus, we have

As a result, we obtained an adjusted clause R4':  $[{}^{0}Go_{wt} {}^{0}Snasel {}^{0}Brussels]$ R5: # Hence, the answer to the question Q is YES.

(R4'+G)

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  ${}^{0}Snasel$  for  ${}^{0}President_{wt}$ . Thus, we have

As a result, we obtained an adjusted clause R4':  $[{}^{0}Go_{wt} {}^{0}Snasel {}^{0}Brussels]$ R5: # (R4'+G) Hence, the answer to the question Q is YES.

#### Notes



#### • We cannot proceed strictly from a goal to meet another goal.

- At the beginning of the resolution process, 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.
- An adjustment of the general resolution method for natural language processing in TIL *cannot be strictly goal-driven*.



- We cannot proceed strictly from a goal to meet another goal.
- At the beginning of the resolution process, 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.
- An adjustment of the general resolution method for natural language processing in TIL *cannot be strictly goal-driven*.



- We cannot proceed strictly from a goal to meet another goal.
- At the beginning of the resolution process, 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.
- An adjustment of the general resolution method for natural language processing in TIL *cannot be strictly goal-driven*.

# Concluding remarks

- 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.
- 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.

- 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.
- 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.

- 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.
- 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.

- 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.
- 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.

#### Thank you for your attention

#### Marie Duží, Michal Fait, Marek Menšík

VSB – Technical University of Ostrava marie.duzi@vsb.cz, michal.fait@vsb.cz, marek.mensik@vsb.cz

December 7, 2019

VSB TECHNICAL | FACULTY OF ELECTRICAL UNIVERSITY | ENGINEERING AND COMPUTER OF OSTRAVA | SCIENCE