Multilinguality Adaptations of Natural Language Logical Analyzer

Marek Medveď, Terézia Šulganová, and Aleš Horák

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech republic
{xmedved1,hales}@fi.muni.cz
445246@mail.muni.cz

Abstract. The AST (automated semantic analysis) system serves as a final pipeline component in translating natural language sentences to formulae of higher-order logic formalism, the Transparent Intensional Logic (TIL). TIL was designed as a full expressive tool capable of representing complex meaning relations of natural language expressions. AST was designed as a language independent tool, which was originally developed for the Czech language.
In this paper, we summarize the latest development of AST aiming at easy transfer of the underlying lexicons and rules to other languages. The changes are test with the English language selected as a representative of a different language family than Czech having general multilingual applicability of the process in mind.

Key words: Transparent Intentional Logic; TIL; logical analysis; natural language semantics

1 Introduction

Capturing full logical representation of natural language expressions is not a trivial task. In our work, we lean on a high-ordered logical formalism the Transparent Intensional Logic (TIL) [1, 2] that was originally designed to cover all logical phenomena present in natural language.

From the theoretical point of view, a semantic representation of one expression through multiple languages should be (structurally) very similar. In the following text, we are presenting the details of new developments of a system for TIL semantic analysis called AST (automated semantic analysis). AST was designed as language independent tool that from a syntactic tree sentence representation can create its logical representation.

To be able to prepare input to the AST processor from standard plain text sentences, we use the SET [3] parser that is also designed for multilingual processing. SET is based on a grammar of pattern-matching dependency rules that can be adapted for any new language.

English Pen TreeBank tags			*Czech attributive tags*		
word	*lemma*	*tag*	*word*	*lemma*	*tag*
Some	some	DT	Some	some	k3
agents	agent	NNS	agents	agent	k1gInP
are	be	VBP	are	be	k5mInP
mobile	mobile	JJ	mobile	mobile	k2gId1
,	,	,	,	,	kIx,
other	other	JJ	other	other	k2gId1
agents	agent	NNS	agents	agent	k1gInP
are	be	VBP	are	be	k5mInP
static	static	JJ	static	static	k2gId1
.	.	SENT	.	.	kIx.

Fig. 1. Tag translation example

In the following text, we describe the SET and AST modifications that allow flexible multilingual setup of the whole pipeline. The resulting translation of natural language expressions to logical formulae are currently tested with English, the changes are however general enough for a transfer to another languages.

2 Tagset Translation

The original implementation of TIL analysis [4] leans on morphological aspects of phrase agreement rules based on the Czech attributive tagset [5] that carry lot of information about the grammatical case, number, gender, person etc. The AST tool is based on the same principles of grammatical agreement test with the exploitation of similar set of attributes that allow to drive the analysis decisions to provide correct TIL constructions.

For example when the system is building a logical construction of a single clause and no acceptable subject was found among the sentence constituents, the system supplies an inexplicit subject formed with a personal pronoun. The actual pronoun specification then follows the subject-predicate agreement rules and identifies the pronoun number, gender and person from the form of the main verb.

Within the multilingual setup of AST, we have decided to keep the attributive morphological specification as a form of a "general pivot" which allows us to process another language (English) in the same way as Czech in many rules. We have supplemented the system with a tag translation module, in the test case from the English Penn TreeBank tagset [6] to the Czech equivalent that also contains additional information about gender, case etc.

Each original Penn TreeBank tag is translated into its Czech equivalent according to multiple rules (for an example tag translation see Figure 1). The additional information is based on definitions related to selected part-of-speech

```
TMPL: verbfin ... comma $CONJ ... verbfin ... rbound
   MARK 2 5 7 <clause>  HEAD 3  DEP 0  PROB 50
   LABEL vrule_sch()
   LABELDEP vrule_sch_add("#1H (#2)")
   $CONJ(tag): k3.*yR k3.*xQ k8.*xS
```

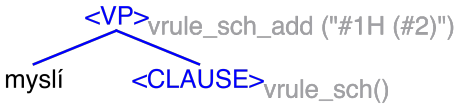


Fig. 2. LABELDEP in a relative clause

categories and in several cases to particular words by using the following list of rules:

- pronouns:
  - masculine gender (`gM`) for pronouns: *he*, *his*, *himself* (similarly for feminine and neuter genders),
  - personal pronoun (`xP`) for words with tag `PP`,
  - possessive pronoun (`xO`) for word with `CDZ`, `PPZ`;
- conjunctions:
  - coordinative type for *and*, *but*, *for*, *nor*, *or*, *so*, *yet*,
  - subordinate type otherwise.

3 SET Modifications

AST processes the input syntactic trees based on rule labels specified with a set of grammar rule labels. In case of phrasal trees, the labels can refer to specific constituents within one rule which can relate several right-hand-side terms in one action. In case of dependency parser, we need to introduce several new technical modifications to address complex constructions like inserted clauses and coordinations.

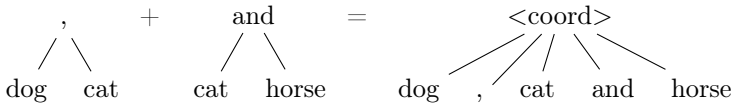
3.1 Action LABELDEP

Within the hybrid tree approach, selected rules in SET can create several dependency edges in one step – e.g. one rule creates a coordination node with 3 children, and at the same time it attaches this coordination node to a governor node. In this situation, AST needs two different label schemata for these two types of attachments to create a correct construction. Therefore, apart from the LABEL action that contains the TIL schemata relevant to the rule [7], we have added a new action called LABELDEP, where the TIL schema for the higher-level dependency is stored. Using this new construction, all nodes in the tree can be assigned a correct schema.

The same solution is applied in case of rules for relative clauses (see Figure 2) – they often recognize a relative sentence and attach it to its governor at the same time. LABELDEP action is used here as well.

3.2 Structured Clauses and Coordinations

The native SET algorithm joins and flattens structures where their inner structure is lacking or debatable – e.g. coordinations connecting 3 and more members (like "*dog, cat and horse*") are succesively built as 2 or more simple coordinations (e.g. *"dog, cat"* and "*cat and horse*)" and the parsing algorithm joins them into one "flat" coordination – all the tokens are appended under a single coordination node:



This is not suitable for the AST algorithm, because each TIL schema needs to know how many parameters are coming and the number is usually not variable. Therefore, we have decided not to flatten the coordinations for AST and rather nest them under each other – in the above example, the necessary form is thus *coord(dog , coord(cat and horse))*. This way, the schemata can process only 2 items at a time, and combine them together at the original parent arching node as depicted in Figure 3.

3.3 Action LABELTOP

SET constructs the top level of the tree automatically, without a reflection among the grammar rules – all the nodes (e.g. *clauses* in a complex sentence) that were not attached by rules anywhere, are attached to the root "sentence" node at the end of parsing. However, since AST needs to define clause coordinations at this level, the top level label has to be specified separately. A new keyword LABELTOP has thus been introduced, which specifies the rule schema of the root node (see Figure 4).

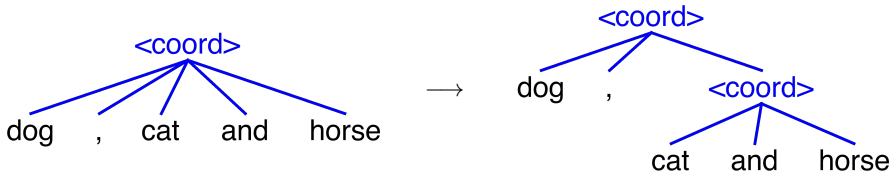


Fig. 3. Coordination nodes splitting in SET for AST processing.

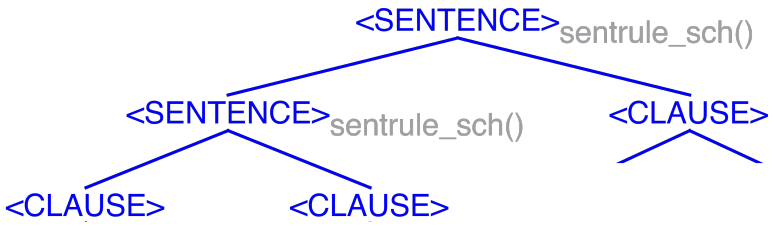


Fig. 4. "LABELTOP sentrule_sch()" label on the sentence level.

3.4 Coordination Nodes Morphology

AST works intensively with morphological tags at the tree nodes. Tree nodes at the SET output are assigned correct morphological tags as they were propagated from the bottom of the tree. However, in case of coordinations of constituents in singular number, the resulting coordination can express both a singular or a plural number according to the context:

$[A\ skier\ or\ a\ climber]_{\text{plural}}$ *are risking their lives.*

Each club member is $[a\ skier\ or\ a\ climber.]_{\text{singular}}$

Not taking this into account introduced errors in cases where this information was used to check morphological agreements in AST, e.g. between subject and predicate in Czech.

AST can now handle the dual number situation with a procedure that assigns both the correct tags to the coordination nodes (see Figure 5). In Czech, a correct gender of the plural case is also handled in accordance with standard grammar rules (masculine animate has precedence before other genders). This enables AST to build correct constructions for sentences with heterogeneous coordinations.

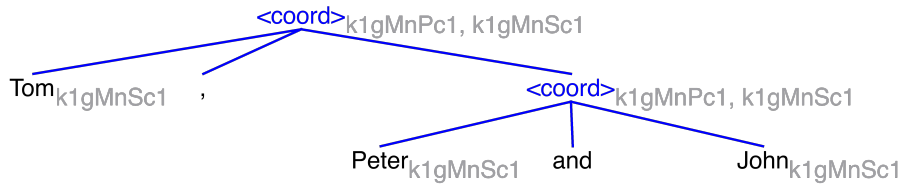


Fig. 5. Multiple grammatical numbers in coordination nodes.

4 AST Modifications

AST as language independent tool can be modified to analyze new language by supplying several language dependent lexicons: lexical item types, verb valency lexicon, prepositional types and sentence schema lexicon. These files were presented in previous publications [7], we are offering only examples of settings for other than the primary language here. Examples of the resulting constructions are presented in Figure 6.

4.1 TIL Types of Lexical Items

TIL lexicon of the bottom level types consists of a list of lexical item specifications with TIL types for each word in the input tree. An example of the types for the verb "*stop*" is the following:

```
stop
/k5/otriv ((o(oo_τω)(oo_τω))ω)
/k5/otriv (((o(oo_τω)(oo_τω))ω)ι)
```

This specification states that "*stop*" is an episodic verb [8, 9] with no object (when someone just stops) and with one object (when someone stops somebody else).

4.2 Verb Valency Lexicon

For each clause the system identifies the sentence valency frame and triggers a process that according schema matching the extracted valency frame specifies how the sub-constructions are put together. An example for verb "*stop*" is:

```
stop
hPTc4 :exists:V(v):V(v):and:V(v)=[[#0,try(#1)],V(w)]
hPTc2r{at} :exists:V(v):V(v):and:V(v) subset [#0,V(w)] and [#1,V(v)]
```

The two schemata correspond to the situations when *somebody* or *something stops* somebody or something else, or when the subject *stops at something*, in which case the second constituent provides a modifier of the verbal object.

4.3 Prepositional Type Lexicon

If the AST tool decides how a prepositional phrase participates on the analyzed valency frame, the actual preposition is the central distinctive feature. An example of a schema for transformation of a prepositional phrase with the preposition "*in*" is:

```
in
0 hL hW
```

The preposition "*in*" can introduce a locational prepositional phrase `hL` (*where*), or a temporal *when/until what time* specification `hW`. The first number can denote a grammatical case of the encompassed noun phrase, however, for English this is left unused.

$$
\begin{array}{l}
\lambda w_1 \lambda t_2 [\mathsf{P}_{t_2}, \\
\quad [\mathsf{Onc}_{w_1}, \\
\quad\quad \lambda w_3 \lambda t_4 (\exists x_5)(\exists i_6)(\exists i_7)(\\
\quad\quad\quad [\mathsf{Does}_{w_3 t_4}, \\
\quad\quad\quad\quad i_7, \\
\quad\quad\quad\quad [\mathsf{Perf}_{w_3}, x_5] \\
\quad\quad\quad] \\
\quad\quad\quad \wedge\ [\mathsf{Ostravsko}_{w_3 t_4}, i_6] \\
\quad\quad\quad \wedge\ x_5 = [\mathsf{navštívit}, i_6]_{w_3} \\
\quad\quad\quad \wedge\ [\\
\quad\quad\quad\quad [\mathsf{Of}, \\
\quad\quad\quad\quad\quad \mathsf{ministr}, \\
\quad\quad\quad\quad\quad \lambda w_8 \lambda t_9 \lambda x_{10}(\\
\quad\quad\quad\quad\quad\quad [\mathsf{doprava}_{w_8 t_9}, x_{10}] \\
\quad\quad\quad\quad\quad\quad\quad \wedge\ [\mathsf{ČR}_{w_8 t_9}, x_{10}] \\
\quad\quad\quad\quad\quad) \\
\quad\quad\quad\quad]_{w_3 t_4}, \\
\quad\quad\quad\quad i_7 \\
\quad\quad\quad] \\
\quad\quad) \\
\quad], \mathsf{Anytime} \\
] \ldots \pi
\end{array}
$$

$$
\begin{array}{l}
\lambda w_1 \lambda t_2 [\mathsf{P}_{t_2}, \\
\quad [\mathsf{Onc}_{w_1}, \\
\quad\quad \lambda w_3 \lambda t_4 (\exists x_5)(\exists i_6)(\exists i_7)(\\
\quad\quad\quad [\mathsf{Does}_{w_3 t_4}, \\
\quad\quad\quad\quad i_7, \\
\quad\quad\quad\quad [\mathsf{Perf}_{w_3}, x_5] \\
\quad\quad\quad] \\
\quad\quad\quad \wedge\ [\\
\quad\quad\quad\quad [\mathsf{the}, \\
\quad\quad\quad\quad\quad [\mathsf{Of}, \\
\quad\quad\quad\quad\quad\quad \mathsf{region}, \\
\quad\quad\quad\quad\quad\quad \mathsf{Ostrava} \\
\quad\quad\quad\quad\quad] \\
\quad\quad\quad\quad]_{w_3 t_4}, \\
\quad\quad\quad\quad i_6 \\
\quad\quad\quad] \\
\quad\quad\quad \wedge\ x_5 = [\mathsf{visit}, i_6]_{w_3} \\
\quad\quad\quad \wedge\ [\\
\quad\quad\quad\quad [\mathsf{the}, \\
\quad\quad\quad\quad\quad [\mathsf{Of}, \\
\quad\quad\quad\quad\quad\quad \mathsf{minister}, \\
\quad\quad\quad\quad\quad\quad [\mathsf{Of}, \\
\quad\quad\quad\quad\quad\quad\quad \mathsf{transport}, \\
\quad\quad\quad\quad\quad\quad\quad \mathsf{CR} \\
\quad\quad\quad\quad\quad\quad] \\
\quad\quad\quad\quad\quad] \\
\quad\quad\quad\quad]_{w_3 t_4}, \\
\quad\quad\quad\quad i_7 \\
\quad\quad\quad] \\
\quad\quad) \\
\quad], \mathsf{Anytime} \\
] \ldots \pi
\end{array}
$$

Fig. 6. Example constructions translated from the same sentence in Czech and English: *Ministr dopravy ČR navštívil Ostravsko* and *The minister of transport of CR visited the region of Ostrava.*

4.4 Sentence Schema Lexicon

The sentence schema lexicon drives the creation of the top level sentence logical construction. The schema takes as argument the sub-constructions of the two clauses that are in a subordination or a coordination structure. An example for the subordinate conjunction "when" is:

```
("when";","):  "lwt(tense_temp(awt(#2),awt(#1)))"
```

This schema specifies that the two sentences have to be combined by applying the subordinate temporal clause as a time interval specification of the main clause.

5 Conclusion

In this paper, we have detailed the latest developments of the pipeline used for logical analysis of natural language sentences by means of the Transparent Intensional Logic. The presented changes aimed at promoting multilingual setup of the system with Czech as a representative of a morphologically rich language providing the attributive basis for phrase agreement specifications and English used as the first tested transfer language.

In the future work, we plan to test the setup with more languages and offer wider scale comparisons of logical structure sharing between different language environments.

Acknowledgements This work has been partly supported by the Czech Science Foundation under the project GA15-13277S.

References

1. Tichy, P.: The foundations of Frege's logic. Walter de Gruyter (1988)
2. Duží, M., Jespersen, B., Materna, P.: Procedural semantics for hyperintensional logic: Foundations and applications of transparent intensional logic. Volume 17. Springer Science & Business Media (2010)
3. Kovář, V., Horák, A., Jakubíček, M.: Syntactic analysis using finite patterns: A new parsing system for czech. In: Human Language Technology. Challenges for Computer Science and Linguistics, Berlin/Heidelberg (2011) 161–171
4. Horák, A.: Computer Processing of Czech Syntax and Semantics. Tribun EU (2008)
5. Jakubíček, M., Kovář, V., Šmerk, P.: Czech Morphological Tagset Revisited. Proceedings of Recent Advances in Slavonic Natural Language Processing 2011 (2011) 29–42
6. Santorini, B.: Part-of-speech tagging guidelines for the penn treebank project (3rd revision). (1990)
7. Medveď, M., Horák, A., Kovář, V.: Bilingual logical analysis of natural language sentences. RASLAN 2016, Recent Advances in Slavonic Natural Language Processing (2016) 69–78
8. Tichý, P.: The semantics of episodic verbs. Theoretical Linguistics **7**(1-3) (1980) 263–296
9. Horák, A.: The Normal Translation Algorithm in Transparent Intensional Logic for Czech. PhD thesis (2002)