Bilingual Logical Analysis of Natural Language Sentences

Marek Medveď, Aleš Horák, Vojtěch Kovář

Natural Language Processing Centre Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic

December 2, 2016

Parsing

- Synt and SET parsers are rule-based systems but they use different approach in providing syntactic analysis, which also influences the differences of their syntactic trees.

Synt

- meta-grammar concept based on traditional linguistic rule systems
- context-free grammar with contextual actions and performs a stochastic agenda-based head-driven chart analysis
- result in form of millions of possible syntactic trees
- implements TIL-based logical analysis of a sentence as a first implementation of the Normal Translation Algorithm

Synt

a)

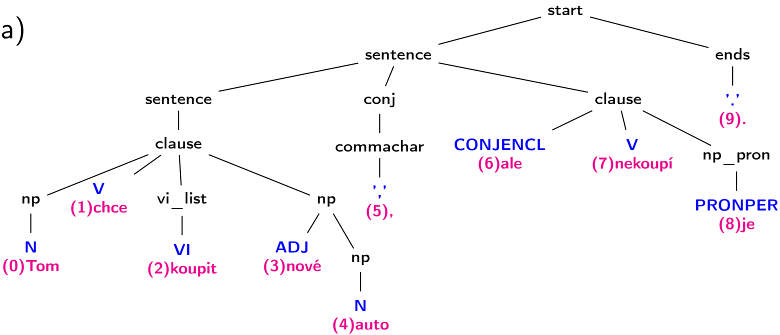

Figure: a) the syntactic tree for sentence "Tom chce koupit nové auto, ale nekoupí je." (Tom wants to buy a new car, but he will not buy it.)

Synt

$$\text{b)} \quad \lambda w_1 \lambda t_2 \Big((\exists x_3)(\exists i_4)(\exists i_5) \Big(\big[\mathbf{Does}_{w_1 t_2}, i_5, [\mathbf{Imp}_{w_1}, x_3]\big] \ \wedge$$
$$\big[[\mathbf{nový}, \mathbf{auto}]_{w_1 t_2}, i_4\big] \ \wedge \ x_3 = [\mathbf{chtít}, i_4]_{w_1} \ \wedge \ [\mathbf{Tom}_{w_1 t_2}, i_5]\Big) \ \wedge$$
$$\Big[\mathbf{Not}, \Big[\mathbf{True}_{w_1 t_2}, \lambda w_6 \lambda t_7 (\exists x_8)(\exists i_9)\Big(\Big[$$
$$\mathbf{Does}_{w_6 t_7}, On, [\mathbf{Perf}_{w_6}, x_8]\Big] \ \wedge \ [\mathbf{on}_{w_6 t_7}, i_9] \ \wedge \ x_8 =$$
$$[\mathbf{koupit}, i_9]_{w_6} \Big)\Big]\Big]\Big) \ldots \pi$$

Figure: b) the logical construction for sentence "Tom chce koupit nové auto, ale nekoupí je." (Tom wants to buy a new car, but he will not buy it.)

SET

- rule-based parser based on pattern-matching dependency rules
- includes grammars for Czech, Slovak and English
- input format for the parser is a morphologically annotated sentence in the vertical format
- output options include:
    - hybrid trees
    - dependency trees
    - phrase structure trees
    - "bush"
- 100% coverage

SET rule

```
TMPL: \$ATTR \$...* noun  AGREE 0 2 cgn   MARK 0  DEP 2   PROB 6000
    LABEL modifier
    LABEL rule_sch ( $\$ \$@ "[#1,#2]" )
```

Figure: SET rule for adjective-noun modifier (e.g. "black dog") supplemented with a TIL schema. The `$ATTR` and `noun` aliases are defined in other parts of the grammar. Actions require agreement in case, gender and number (cgn), create an *adjective* → *noun* dependency and set up heavy weight (PROB) for this rule. The TIL schema says that the attribute (#1) is applied as a function to the noun (#2) – these indexes refer directly to the resulting structured tree.

SET hybrid tree

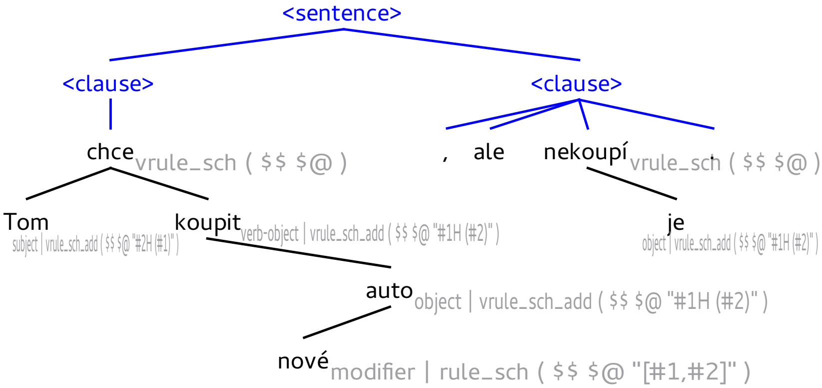

Figure: Hybrid SET trees for the sentence "Tom chce koupit nové auto, ale nekoupí je." (Tom wants to buy a new car but he will not buy it.)

SET phrasal tree

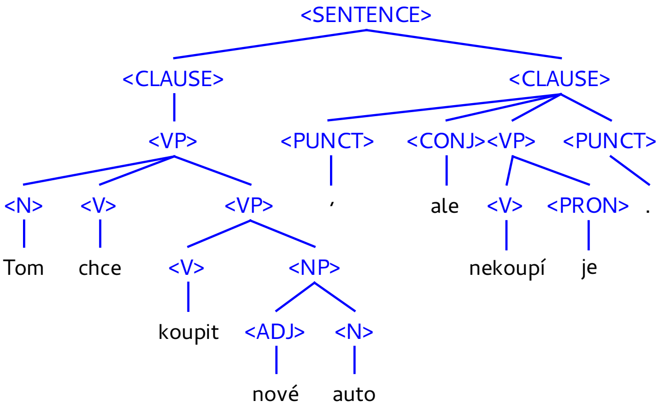

Figure: Phrasal SET trees for the sentence "Tom chce koupit nové auto, ale nekoupí je." (Tom wants to buy a new car but he will not buy it.)

SET new “structured” tree

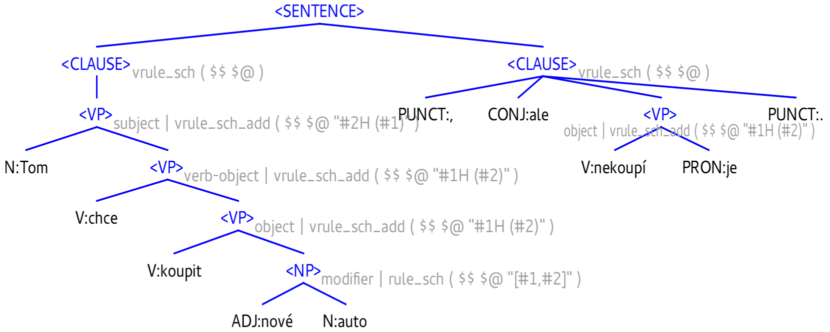

Figure: New “structured” SET trees for the sentence “Tom chce koupit nové auto, ale nekoupí je.” (Tom wants to buy a new car but he will not buy it.)

SET modifications for TIL analysis

- phrase structure output of SET was the most suitable for the TIL analysis → conversion from the hybrid tree
- TIL analysis algorithm processes the tree level by level:
  - TIL schema is asigned to leach level
  - we developed a new type of SET phrase structure output where each level of the tree corresponds exactly to one rule in the SET grammar

AST

- AST is a language independent tool for semantic analysis of natural language sentences
- emerged from the `synt` parser
- easy adaptations for new languages and syntactic analyser
- follows the principles of the NTA algorithm and uses the Transparent Intentional Logic (TIL)
- modular system that consists of seven main parts

Input processor

- reads the input with a syntactic tree in textual form:

Example

id	*nterm:word*	*lemma*	*tag*	*pid*	*til schema*
0	N:Tom	Tom	k1gMnSc1;ca14	p	
1	V:chce	chtít	k5eAaImIp3nS	15	
2	V:koupit	koupit	k5eAaPmF	16	
3	ADJ:nové	nový	k2eAgNnSc4d1	17	
4	N:auto	auto	k1gNnSc4	17	
5	PUNCT:,	,	kIx	10	
6	CONJ:ale	ale	k8xC	10	
7	V:nekoupí	koupit	k5eNaPmIp3nS	13	
8	PRON:je	on	k3xPp3gNnSc4	13	
9	PUNCT:.	.	kIx.	10	
10	<CLAUSE>		k5eNaPmIp3nS	12	vrule_sch ( )
11	<CLAUSE>		k5eAaImIp3nS	12	vrule_sch ( )
12	<SENTENCE>			-1	
13	<VP>	koupit	k5eNaPmIp3nS	10	vrule_sch_add ("#1H (#2)")
14	<VP>	chtít	k5eAaImIp3nS	11	vrule_sch_add ("#2H (#1)")
15	<VP>	chtít	k5eAaImIp3nS	14	vrule_sch_add ("#1H (#2)")
16	<VP>	koupit	k5eAaPmF	15	vrule_sch_add ("#1H (#2)")
17	<NP>	auto	k1gNnSc4	16	rule_sch ("[#1,#2]")

Grammar processor

- reads the list of grammar rules and schema actions to obtain a schema for each node inside the tree (only for `synt` parser).

Example

```
np -> left_modif np
    rule_schema ( "[#1,#2]" )
```

Lexical items processor

- a list of lexical item specifications with TIL types for each leaf (word) in the tree structure

Example (lexical item for the verb "koupit" (buy))

koupit

`/k5/otriv` $(((o(oo_{\tau\omega})(oo_{\tau\omega}))\omega)\iota)$

Schema processor

- general module for operations over the logical construction schemata

Example (construction for the noun phrase "nové auto" (new car))

rule_schema: '[#1,#2]'
Processing schema with params:
#1: ^{0}nový...$((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$
#2: ^{0}auto...$(o\iota)_{\tau\omega}$
Resulting constructions:
[^{0}nový/$((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$, ^{0}auto/$(o\iota)_{\tau\omega}$]...$(o\iota)_{\tau\omega}$

Verb valency processor

- identifies the correct valency frame for the given sentence and triggers the schema parser on sub-constructions according to the schema coming with the valency

Example (verb "koupit" (buy))

```
koupit
hPTc4 :exists:V(v):V(v):and:V(v)=[[#0,try(#1)],V(w)]
```

Prepositional valency expression processor

- translates analysed prepositional phrases to possible valency expressions

Example (record for the preposition "k")

```
k
3 hA hH
```

The preposition "k" (to) can introduce a prepositional phrase of a *where-to* direction `hA`, or a modal *how/what* specification `hH`.

Sentence schema processor

- structure contains subordination or coordination clauses the sentence schema parser is triggered. The sentence schemata are classified by the conjunctions used between clauses

Example (conjunction “ale” (but))

```
("";"ale") : "lwt(awt(#1) and awt(#2))"
```

AST Adaptations for the SET Trees

- organization of the SET output in an adapted phrasal-dependency tree

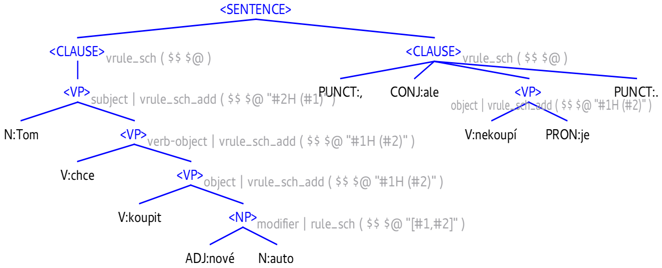

Figure: New "structured" SET trees for the sentence "Tom chce koupit nové auto, ale nekoupí je." (Tom wants to buy a new car but he will not buy it.)

AST Adaptations for the SET Trees

- splitting the corresponding verb rule schemata to binary additive actions denoted as `vrule_sch_add`
  - in the SET tree, the additive actions `vrule_sch_add` build the arguments needed by the clause-level `vrule_sch` action (synt: all constituent groups are reachable directly from the clause node)
- automatic triggering of function for a recursive combination of clauses (`rule_sch`)

AST input

Example

id	*word:nterm*	*lemma*	*tag*	*pid*	*til schema*
0	N:Tom	Tom	k1gMnSc1;ca14	p	
1	V:chce	chtít	k5eAaImIp3nS	15	
2	V:koupit	koupit	k5eAaPmF	16	
3	ADJ:nové	nový	k2eAgNnSc4d1	17	
4	N:auto	auto	k1gNnSc4	17	
5	PUNCT:,	,	kIx	10	
6	CONJ:ale	ale	k8xC	10	
7	V:nekoupí	koupit	k5eNaPmIp3nS	13	
8	PRON:je	on	k3xPp3gNnSc4	13	
9	PUNCT:.	.	kIx.	10	
10	<CLAUSE>		k5eNaPmIp3nS	12	vrule_sch ($$ $@)
11	<CLAUSE>		k5eAaImIp3nS	12	vrule_sch ($$ $@)
12	<SENTENCE>			-1	
13	<VP>	koupit	k5eNaPmIp3nS	10	vrule_sch_add ($$ $@ "#1H (#2)")
14	<VP>	chtít	k5eAaImIp3nS	11	vrule_sch_add ($$ $@ "#2H (#1)")
15	<VP>	chtít	k5eAaImIp3nS	14	vrule_sch_add ($$ $@ "#1H (#2)")
16	<VP>	koupit	k5eAaPmF	15	vrule_sch_add ($$ $@ "#1H (#2)")
17	<NP>	auto	k1gNnSc4	16	rule_sch ($$ $@ "[#1,#2]")

Bilingual Logical Analysis

- SET parser contains comparable grammars for the Czech and English → enables bilingual logical analysis
  - Czech grammar has 71 rules
  - English grammar has 45 rules.

SET English tree

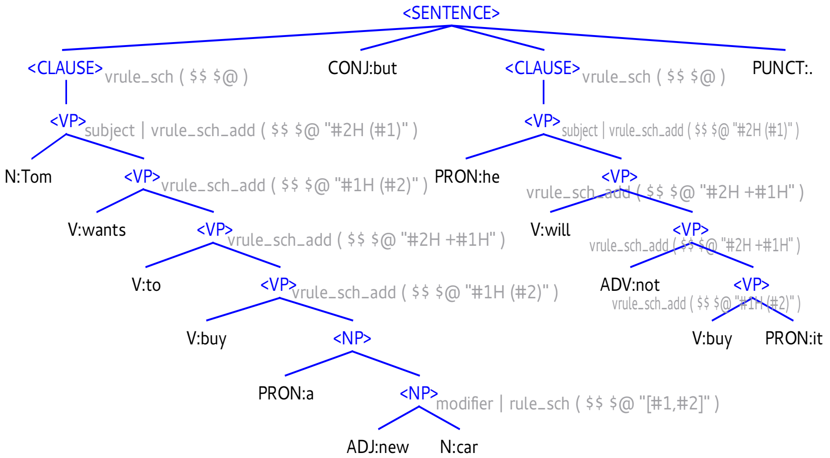

Figure: SET tree for English sentence "Tom wants to buy a new car but he will not buy it.", according to SET grammar for English, and annotated with TIL schemata.

TIL construction of Enlish sentence

$$a)\ \lambda w_1 \lambda t_2 \Big((\exists x_3)(\exists i_4)(\exists i_5) \Big(\big[\mathbf{Does}_{w_1 t_2}, i_5, [\mathbf{Imp}_{w_1}, x_3] \big] \ \wedge$$

$$\big[[\mathbf{new}, \mathbf{car}]_{w_1 t_2}, i_4 \big]$$

$$\wedge \ x_3 =$$

$$[\mathbf{to\_want}, i_4]_{w_1} \ \wedge \ [\mathbf{Tom}_{w_1 t_2}, i_5] \Big) \ \wedge \ \Big[\mathbf{Not}, \Big[\mathbf{True}_{w_1 t_2},$$

$$\lambda w_6 \lambda t_7 (\exists x_8)(\exists i_9) \Big(\big[\mathbf{Does}_{w_6 t_7}, He, [\mathbf{Perf}_{w_6}, x_8] \big] \ \wedge$$

$$[\mathbf{it}_{w_6 t_7}, i_9]$$

$$\wedge \ x_8 = [\mathbf{to\_buy}, i_9]_{w_6} \Big) \Big] \Big] \Big) \ldots \pi$$

Figure: the logical construction for sentence "Tom wants to buy a new car, but he will not buy it."

TIL construction of Czech sentence

b) $\lambda w_1 \lambda t_2 \Big((\exists x_3)(\exists i_4)(\exists i_5)\Big(\big[\mathbf{Does}_{w_1 t_2}, i_5, [\mathbf{Imp}_{w_1}, x_3]\big] \wedge \big[[\mathbf{nov\acute{y}}, \mathbf{auto}]_{w_1 t_2}, i_4\big] \wedge x_3 = [\mathbf{cht\acute{i}t}, i_4]_{w_1} \wedge [\mathbf{Tom}_{w_1 t_2}, i_5]\Big) \wedge \Big[\mathbf{Not}, \Big[\mathbf{True}_{w_1 t_2}, \lambda w_6 \lambda t_7 (\exists x_8)(\exists i_9)\Big(\big[\mathbf{Does}_{w_6 t_7}, \mathit{On}, [\mathbf{Perf}_{w_6}, x_8]\big] \wedge [\mathbf{on}_{w_6 t_7}, i_9] \wedge x_8 = [\mathbf{koupit}, i_9]_{w_6}\Big)\Big]\Big]\Big) \ldots \pi$

Figure: the logical construction for sentence "Tom chce koupit nové auto, ale nekoupí je." (Tom wants to buy a new car, but he will not buy it.)

Conclusions

- independent automated semantic tool AST
- semantic analysis for two languages – Czech and English

Future Directions

- connection of English TIL lexicons to large available resources such as VerbNet and WordNet
- evaluation of English and Czech semantic analysis on a representative set of bilingual sentence pairs

Acknowledgements

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

Thank you for your attention.