The Normal Translation Algorithm in Transparent Intensional Logic for Czech

Aleš Horák

Faculty of Informatics, Masaryk University

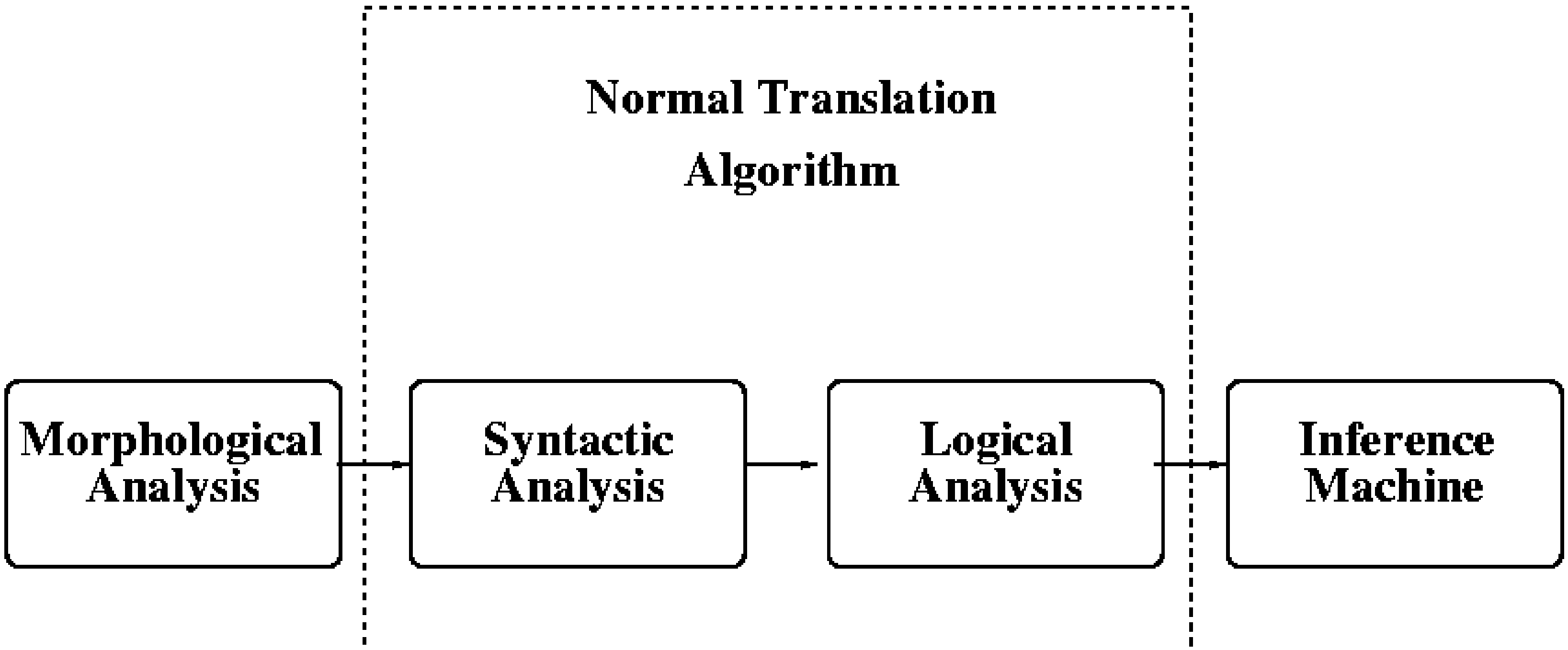
Botanická 68a, CZ-602 00 Brno, Czech Republic

E-mail: hales@fi.muni.cz

Outline

- motivations for NTA
- syntactic analysis
- logical analysis
- results & examples
- conclusions

Sentence Analysis (with TIL)

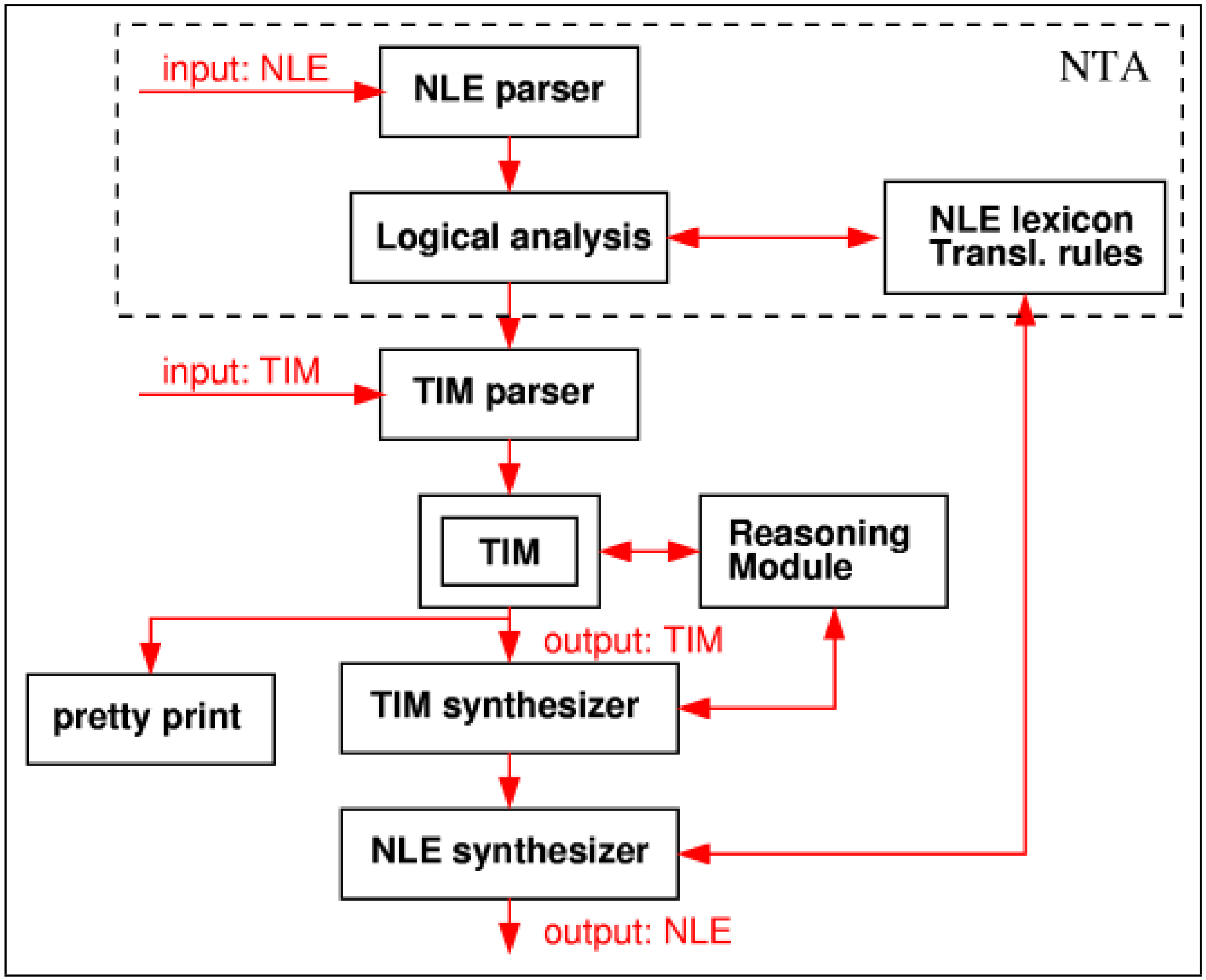


Knowledge of language is modular.

COLING'2000: Angela Friederici, *Language Processing in the Human Brain*, Max Planck Institute of Cognitive Neuroscience, Leipzig

The CAT System Outline

Communication and Artificial Reasoning with TIM



Syntactic Parser (NTA$_1$)

- team work — with Pavel Smrž and Vladimír Kadlec
- metagrammar concept
- head-driven chart parser
- packed shared forest + packed dependency graph
- output:
  - derivation trees
  - dependency trees

Parsing System Design

- efficiency and portability of the parser – C/C++ code implementation
- procedural approach vs. rule based (simplicity of rules)
- grammar maintenance by linguists $\longrightarrow$ declarativeness
- connection to the morphological analyser
- massive syntactic ambiguity

metagrammar formalism:

- CF backbone + functional constraints
- translation of functional constraints to CF rules
- Czech — free word order + very rich morphology (3000 tags)
- searching the optimal parsing strategy for Czech

Forms of Grammar

Metagrammar (G1)

- rules with combinatoric constructs + global order constraints
- actions (= grammatical tests + contextual actions)
- Czech linguistics tradition — dependency structures, agreement checks, word order rules: topic–focus (thema–rhema), strict rules for enclitics

Generated Grammar (G2)

- CF rules
- tests (functional constraints) + actions

Expanded Grammar (G3)

- CF rules (tests translated to rules)

Meta-grammar

= global order constraints + special flags

The main combinatoric constructs in the meta-grammar are `order()`, `rhs()` and `first()` which are used for generating variants of assortments of given terminals and nonterminals.

`order()` generates all possible permutations of its components.
`first()` argument cannot be preceded by any other construct
`rhs()` gives all possible RHS of its argument

/ budu se ptát */*

```
clause ===> order(VBU,R,VRI)
```

/ který ... */*

```
relclause ===> first(relprongr) rhs(clause)
```

Meta-grammar (cont.)

`->` ordinary CFG transcription

`-->` intersegments between each couple of listed elements

`==>` + checking of correct enclitics order

`===>` intersegments in the beginning and the end of RHS, conjunctions, . . .

```
ss -> conj clause
```

/ budu muset číst */*

```
futmod --> VBU VOI VI
```

/ byl bych býval */*

```
cpredcondgr ==> VBL VBK VBLL
```

/ musím se ptát */*

```
clause ===> VO R VRI
```

Meta-grammar (cont.)

Global order constraints inhibit some combinations of terminals in rules

`%enclitic` – which terminals should be regarded as enclitics

`%order` guarantees the pre-defined order

/ jsem, bych, se */*

```
%enclitic = (VB12, VBK, R)
```

/ byl — četl, ptal, musel */*

```
%order VBL = {VL, VRL, VOL}
```

/ býval — četl, ptal, musel */*

```
%order VBLL = {VL, VRL, VOL}
```

Grammatical tests

- grammatical case test for particular words and noun groups

```
noun-genitive-group  ->  noun-group noun-group
          test_genitive($2)
          propagate_all($1)
```

- agreement test of case in prepositional construction
- agreement test of number and gender for relative pronouns
- agreement test of case, number and gender for noun groups

```
prepositional-group  ->  PREPOSITION noun-group
          agree_case_and_propagate($1,$2)
          add_prep_ngroup($1)
```

- test of agreement between subject and predicate
- test of the verb valencies

```
clause  ->  subj-part verb-part
          agree_subj_pred($1,$2)
          test_valency_of($2)
```

Contextual actions

- `propagate_all` and `*_and_propagate`

  propagate relevant information upwards in derivative tree

- `head` and `depends`

  build dependency structure

- `rule_schema` and `verb_rule_schema`

  definitions for TIL logical analysis

Parser Actions

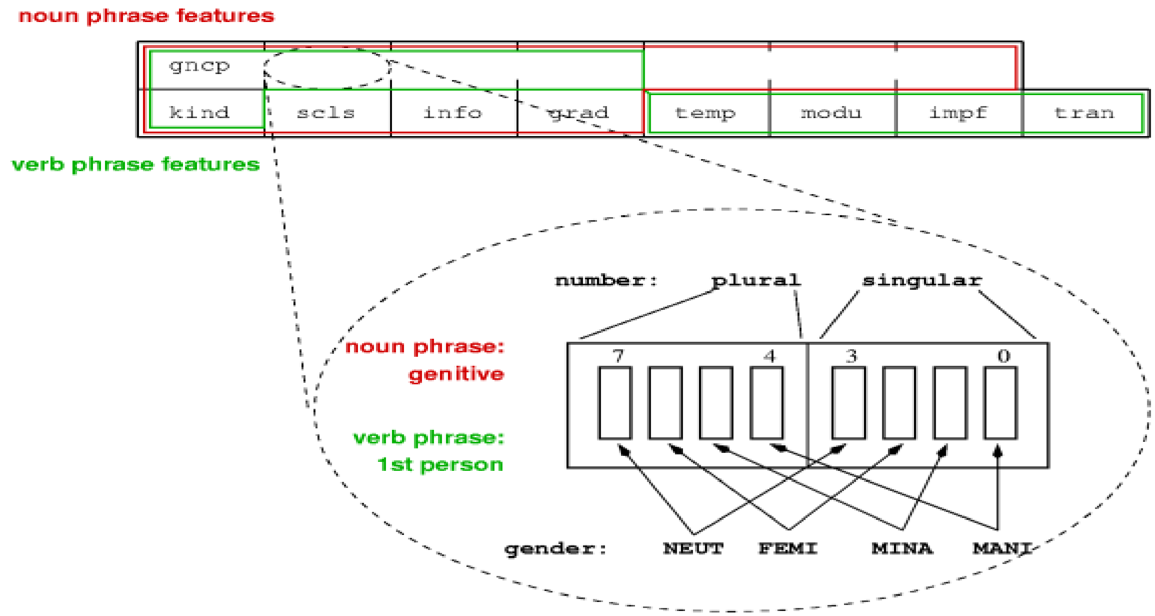
4 kinds of contextual actions, tests or functional constraints:

1. rule-tied actions
2. agreement fulfilment constraints
3. post-processing actions
4. actions based on derivation tree

Parser

- head-driven chart parser
- 6 hash tables for edges and rules
- resulting data structure — packed shared forest

data structure for constraint evaluation



language specific

feature merging —

COLING'2000

- motivations for NTA
- syntactic analysis

$\Rightarrow$ logical analysis

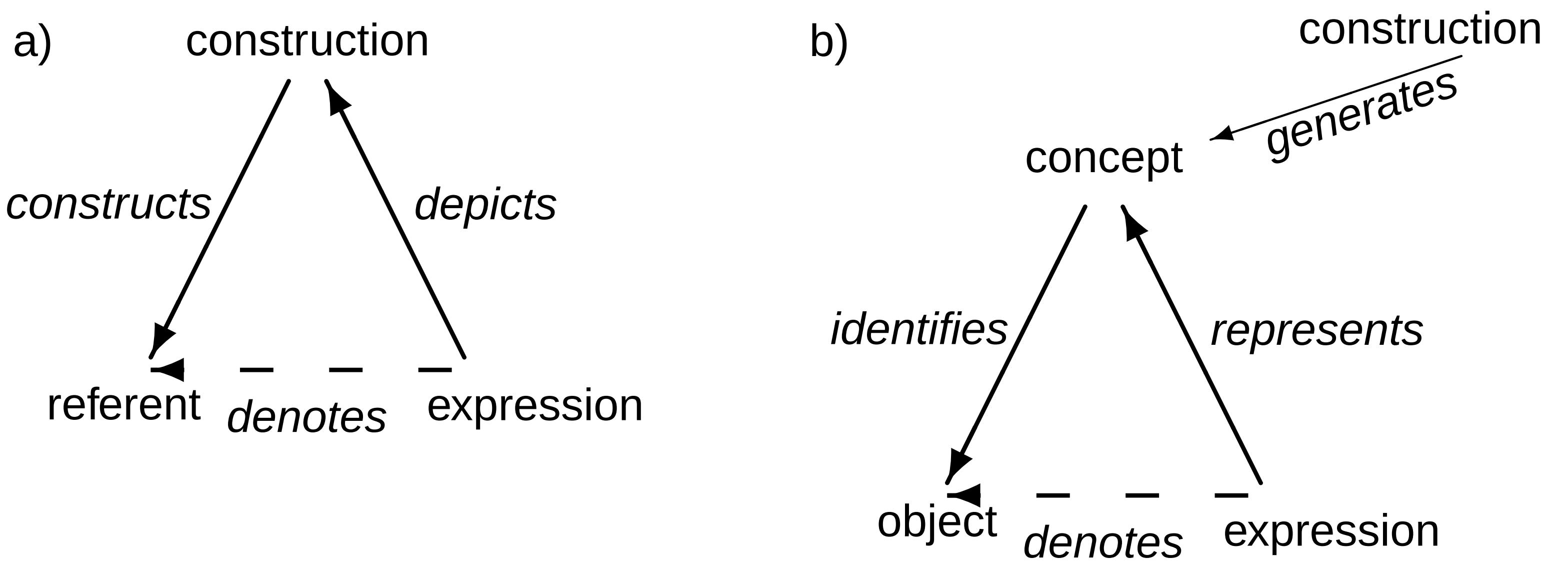
- results & examples
- conclusions

Logical Analysis in TIL (NTA$_2$)

- based on *compositionality principle*
- aim: prepare input for *TIL Inference Machine*
- description of *Knowledge Base Representation*
- in cooperation with Leo hadacz

Expression-Meaning Relationship

a) the expression-meaning relation in TIL and b) with Materna's conceptual approach.



enhancements:

- construction normal form
- new definition of concept

TIL — Transparent Intensional Logic

Tichý, P., *The Foundations of Frege's Logic*, de Gruyter, Berlin, New York, 1988.

- logical system suitable as a meaning surrogate (intensions, possible worlds, temporal and modal variability)
- parallel to Montague's logic, TIL has greater expressivity
- typed λ-calculus logic with particular epistemic framework
- *basic types* = $\{\iota, o, \tau, \omega\}$, (individuals, truth values, real numbers or time moments and possible worlds); *other types*: functions or higher rank types ($\iota_{\tau\omega}$ – individual role, $(o\iota)_{\tau\omega}$ – class of individuals or property, $(o\alpha\beta)_{\tau\omega}$ – intensional relation between object of types α and β, $*_n$ – class of constructions of order n,...)
- constructions – λ-calculus formulae with specific modes of constructions (trivialization).
- inference rules for TIL are well defined
- Normal Translation Algorithm (NTA)

Logical Analysis of NL Sentences

- Verb Phrase
- Noun Phrase
- Sentence Building
- Folding of Constituents
- Special Compound
- Questions and Imperatives

Verb Phrase

- Episodic Verb — events, episodes, verbal object, verb
- Verb Aspect
- Verb Tense
- Active and Passive Voice
- Adverbial Modification
- Auxiliary and Modal Verbs
- Infinitive
- Verb Valency

Noun Phrase

- Adjective Modifier
- Prepositional Noun Phrase
- Genitive Construction
- Pronoun and Proper Name (interrogative, indefinite and negative pronoun)
- Numeral
- Quantificational Phrase

Compound Constituents

Sentence Building

- subordinate clauses
- coordinate clauses

Folding of Constituents

- lists of constituents

Special Compound

- extensions (numbers, date, time, . . .)

Questions and Imperatives

match $x : C$

x ... object or variable, C construction

both construct (or are) one and the same object

kinds of attitudes to proposition:

Yes/No

Je Petr vyšší než Karel? (Is Peter taller than Charles?)

Wh-

Která hora je nevyšší na světě? (Which mountain is the highest in the world?)

Expl

Proč je Marie smutná? (Why is Mary sad?)

Imp

Petře, uvař oběd! (Peter, make lunch!)

- motivations for NTA
- syntactic analysis
- logical analysis

$\Rightarrow$ results & examples

- conclusions

Results

Grammar — number of rules

G1 meta-grammar – # rules	326
G2 generated grammar – # rules	2919
shift/reduce conflicts	48833
reduce/reduce conflicts	5067
G3 expanded grammar – # rules	10207

System coverage on 10000 sentences

	# of sent.	percentage
successful at level 0, corpus	5150	51.5 %
successful at level 99, corpus	3986	39.9 %
successful at level 0, text	304	3.0 %
successful at level 99, text	211	2.1 %
unsuccessful	349	3.5 %
overall successful	9651	**96.5 %**
sum	10000	100.0 %

Timing Results

average time for sentence	0.17 s
minimum —‖—	$<$0.01 s
maximum —‖—	32.47 s
median of —‖—	**0.09 s**
average number of words in sentence	15.4
minimum —‖—	1
maximum —‖—	73
median of —‖—	**14**
average number of trees	$890 \cdot 10^{12}$
minimum —‖—	1
maximum —‖—	$5.7 \cdot 10^{18}$
median of —‖—	**56**
average number of edges	6519.7
minimum —‖—	81
maximum —‖—	186329
median of —‖—	**4181**

Precision Estimates

correct analysis — passes the parsing process + (at least one) output tree reflects the required *context* relations in the input.

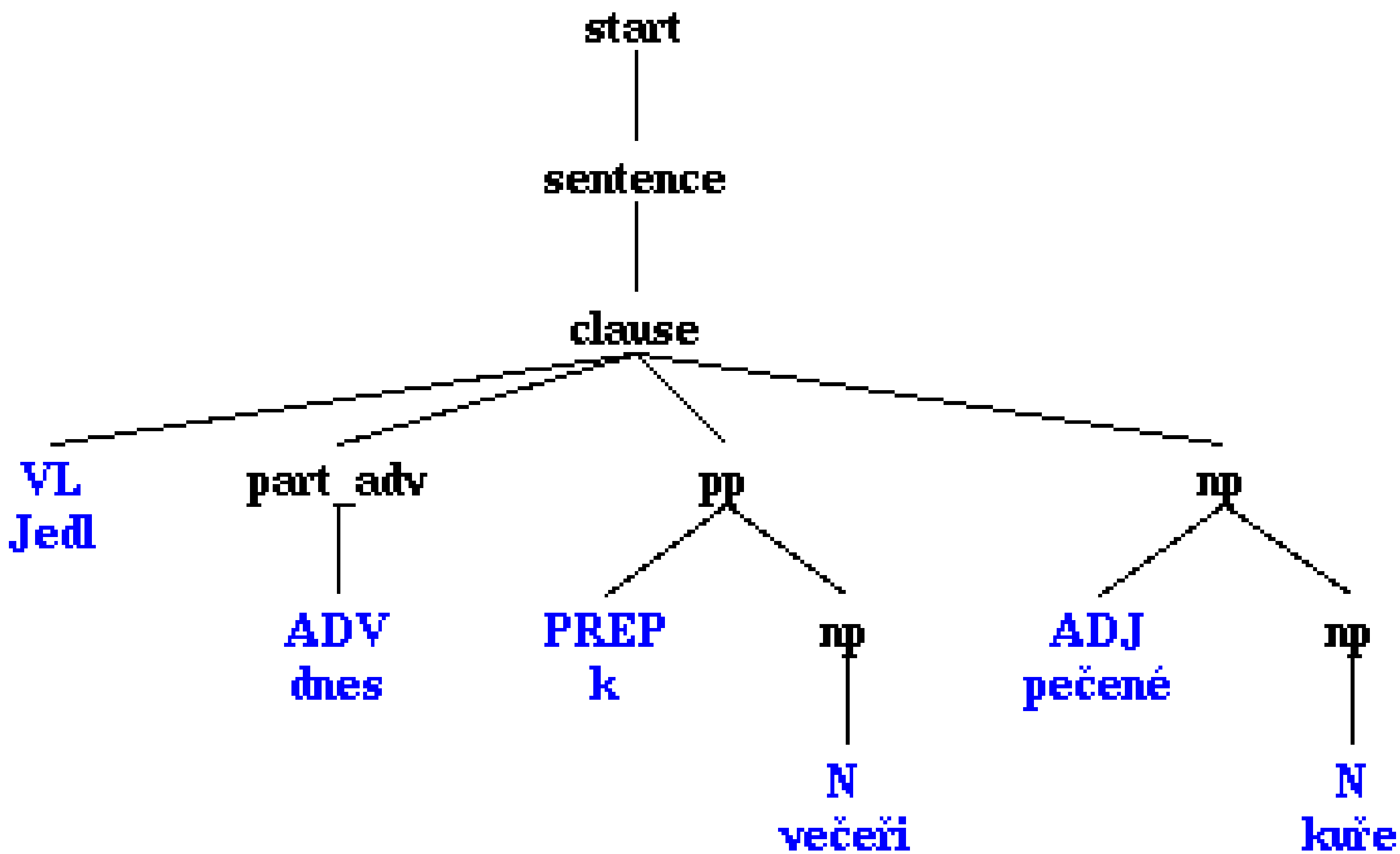
hit precision — percentage describing the portion of correct analyses.

Statistical data describing the analysis of 100 sentences and their hit precision:

	# of sent.	percentage
hit precision of sentences of 1-10 words	32	100.0 %
hit precision of sentences of 11-20 words	37	80.4 %
hit precision of sentences of more than 20 words	8	57.1 %
overall hit precision	77	**83.7 %**
number of sentences with mistakes in input	8	8.0 %
number of sentences	100	100.0 %

Example — derivation tree

An example of resulting derivation tree for sentence '`Jedl dnes k večeři pečené kuře.`' (*He ate a roast chicken for dinner today.*)



Example — logical analysis

evaluation of `rule_schema` for `np` '`pečené kuře`'

```
4, 6, -npnl -> .{ left_modif } np .: k1gNnSc145
agree_case_number_gender_and_propagate OK
rule_schema: 2 nterms, 'lwtx(awtx(#1) and awtx(#2))'
And constrs, Abstr and Exi vars are just gathered
1 (1x1) constructions:
```

$$\lambda w_2 \lambda t_3 \lambda x_4 ([\textbf{pečený}_{w_2 t_3}, x_4] \ \wedge \ [\textbf{kuře}_{w_2 t_3}, x_4]) \ldots (o\iota)_{\tau\omega}$$

```
And constrs: none added
Exi vars: none added
```

Example — logical analysis (cont.)

evaluation of `verb_rule_schema` for the whole clause

```
verb_rule_schema:  3 groups
no acceptable subject found:  supplying an inexplicit one
```

`inexplicit subject:  k3xPgMnSc1,k3xPgInSc1:` $On \ldots \iota$

```
Clause valency list:    jíst <v>#1:(1)hA-#2:(2)hPTc1,    ...
Verb valency list:      jíst <v>#2:hH-#1:hPTc4ti
Matched valency list:   jíst <v>#2:(1)hH-#1:(2)hPTc4ti
```

`time span:` $\lambda t_{12}\textbf{dnes}_{t\,t_{12}} \ldots (o\tau)$

`frequency:` $\textbf{Onc} \ldots ((o(o\tau))\pi)_{\omega}$

`verbal object:` $x_{15} \ldots (o(o\pi)(o\pi))$

`present tense clause:`

$\lambda w_{17}\lambda t_{18}(\exists i_{10})(\exists x_{15})(\exists i_{16})([\textbf{Does}_{w_{17}t_{18}}, On, [\textbf{Imp}_{w_{17}}, x_{15}]] \wedge [\textbf{večeře}_{w_{17}t_{18}}, i_{10}] \wedge [\textbf{pečený}_{w_{17}t_{18}}, i_{16}] \wedge [\textbf{kuře}_{w_{17}t_{18}}, i_{16}] \wedge x_{15} = [\textbf{jíst}, i_{16}]_{w_{17}} \wedge [[\textbf{k}_{w_{17}t_{18}}, i_{10}]_{w_{17}}, x_{15}]) \ldots \pi$

`clause:`

$\lambda w_{19}\lambda t_{20}[\textbf{P}_{t_{20}}, [\textbf{Onc}_{w_{19}}, \lambda w_{17}\lambda t_{18}(\exists i_{10})(\exists x_{15})(\exists i_{16})([\textbf{Does}_{w_{17}t_{18}}, On, [\textbf{Imp}_{w_{17}}, x_{15}]] \wedge [\textbf{večeře}_{w_{17}t_{18}}, i_{10}] \wedge [\textbf{pečený}_{w_{17}t_{18}}, i_{16}] \wedge [\textbf{kuře}_{w_{17}t_{18}}, i_{16}] \wedge x_{15} = [\textbf{jíst}, i_{16}]_{w_{17}} \wedge [[\textbf{k}_{w_{17}t_{18}}, i_{10}]_{w_{17}}, x_{15}])], \lambda t_{12}\textbf{dnes}_{t\,t_{12}}] \ldots \pi$

Conclusions

- the mettagrammar formalism for syntactic analysis
- translation of functional constraints to CF rules is feasible
- implementation of a fully competitive parser for Czech
- comparison of TIL to other semantic representations
- new definition of concept
- Normal Translation Algorithm
  - first exact algorithm of such extent
  - new analysis of most phenoma in Czech