



On the Role of Restarting Automata in Grammar Checking of Czech

Vladislav Kuboň

Charles University in Prague

How did it all start?

Project LATESLAV – Language Technology for Slavic Languages
JRP PECO 2824 coordinated by Universität des Saarlandes in Saarbrücken

Partners: University of Barcelona, University of Sofia, Charles University and two companies, one Bulgarian and one Czech.

Task: To develop a grammar-based grammar checker for both Slavic languages involved.

Project start: 1993

Project duration: 3 years

Main challenge: Free word order (error patterns do not work)

How free is the word-order in Czech?

The system was not able to open the highlighted file.

Systemu se nepodařilo otevřít označený soubor.

Systemu se nepodařilo označený soubor otevřít.

Systemu se označený soubor nepodařilo otevřít.

Otevřít označený soubor se systému nepodařilo.

Otevřít se nepodařilo systému označený soubor.

Označený soubor se nepodařilo systému otevřít.

Označený soubor se systému otevřít nepodařilo.

Označený soubor se systému nepodařilo otevřít.

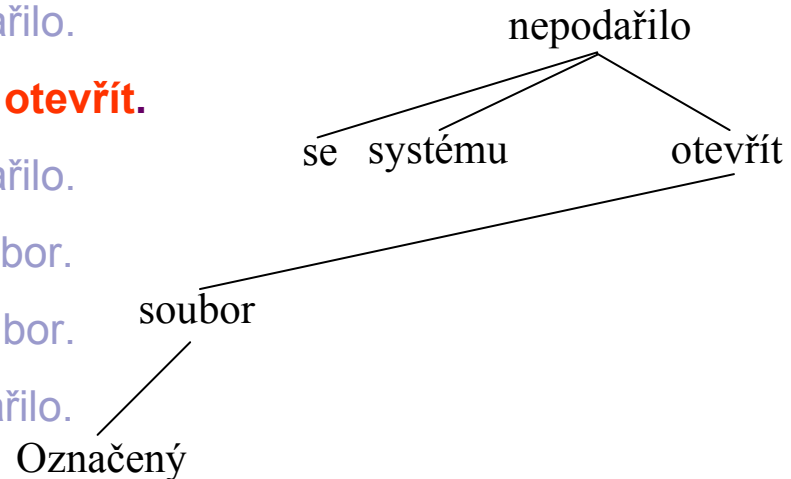
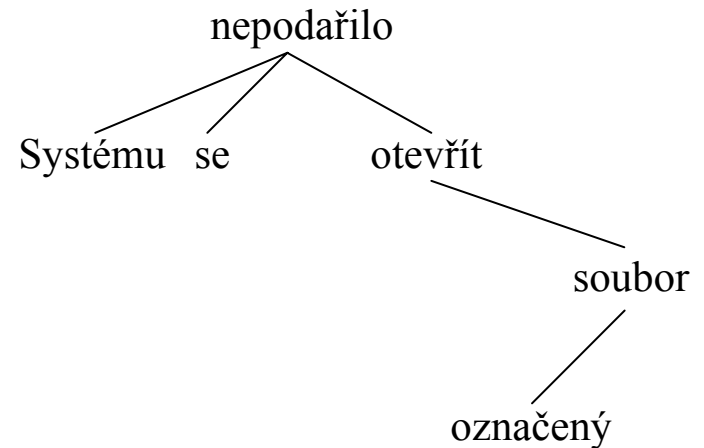
Otevřít se označený soubor systému nepodařilo.

Nepodařilo se systému otevřít označený soubor.

Označený se systému nepodařilo otevřít soubor.

Systemu se otevřít označený soubor nepodařilo.

etc



Very much, but there are limits:

The system was not able to open the highlighted file.

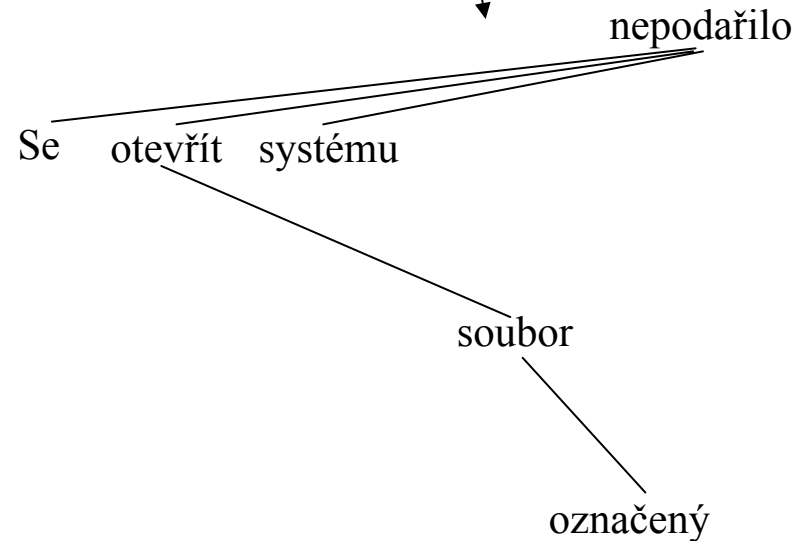
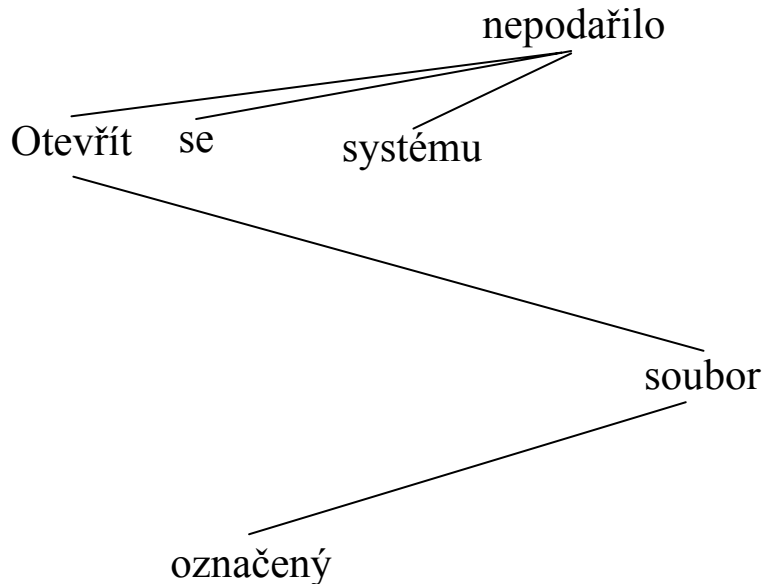
?Otevřít se označený systému nepodařilo soubor.

?Označený se systému soubor nepodařilo otevřít.

***Se otevřít systému soubor označený nepodařilo.**

*Soubor systému otevřít nepodařilo označený se.

etc.



Initial Idea – Restarting Automata

COLING 1994 – an article describing the idea (M.Plátek)

Two restarting automata – a positive and a negative one

Stepwise reduction of an input sentence preserving a basic invariant:

After each reduction by a positive automaton the number of errors should not decrease. Only negative automaton can remove an error (and report it to the user).

Example (from the COLING '94 paper):
The little **boys** I mentioned **runs** very quickly.
The **boys** I mentioned **runs** very quickly.
The **boys** I mentioned **runs** quickly.
The **boys runs** quickly
The **boys runs**.
boys runs.

Robust Free-Order Dependency Grammar

It turned out that it is more convenient to transform the automata into a grammar while preserving the initial idea – each metarule of a grammar tries to remove a single item from the input while preserving the correctness/incorrectness.

Robust Free-Order Dependency Grammar (RFODG) is a 4-tuple $(\mathbf{N}, \mathbf{T}, \mathbf{St}, \mathbf{P})$, where \mathbf{N} is the set of nonterminals, \mathbf{T} is the set of terminals and the union of \mathbf{N} and \mathbf{T} is denoted as \mathbf{V} , $\mathbf{St} \subset \mathbf{N}$ is the set of root symbols (starting symbols), and \mathbf{P} is the set of rewriting rules of two types of the form:

- a) $\mathbf{A} \rightarrow_X \mathbf{BC}$, where $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbf{V}$, \mathbf{X} is denoted as the subscript of the rule, $\mathbf{X} \in \{\mathbf{L}, \mathbf{R}, \mathbf{LP}, \mathbf{RP}\}$,
- b) $\mathbf{A} \rightarrow \mathbf{B}$, where $\mathbf{A}, \mathbf{B} \in \mathbf{V}$.

We suppose that $\mathbf{V} = \mathbf{Vp} \cup \mathbf{Vn}$, where \mathbf{Vp} is the set of positive (correct) symbols, and \mathbf{Vn} is the set of negative symbols (negative symbols mark syntactic inconsistencies contained in the tree representing the syntactic structure of a sentence).

Data Definition Language

```
dělali
LEXF: dělat
WCL: vb
SYNTCL: v
REFL: 0
FRAMESET: (
  [ ACTANT: act
    CASE: nom
    PREP: 0 ]

  [ ACTANT: pat
    CASE: acc
    PREP: 0 ] )

PERS: 3
NUM: pl
TENSE: past
GENDER: anim
END
```

```
květiny
LEXF: květina
WCL: noun
SYNTCL: noun
TANT: 0
GENDER: fem
?
  NUM: pl
  CASE: ? voc , acc , nom !
,
  NUM: sg
  CASE: gen
!
DEPPRN: yes
DEPNUM: yes
RIGHTGEN: yes
END
```

Grammar Description Language

```
METARULE PrepPhrase
;-----
; 6th rule - simple prepositional phrase
;
PROJECTIVE
  A.syntcl = prep
  B.syntcl = noun

  A.case ? B.case  Case_Dis_Prep_Noun

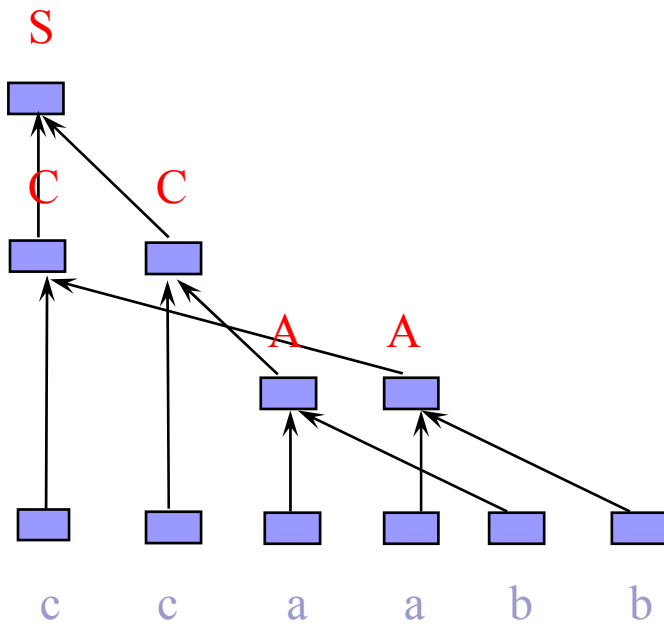
  X:= B
  X.syntcl := prephr

  X.prep := A.lexf
  ; The lexical value of the preposition is stored
  ; for the future use

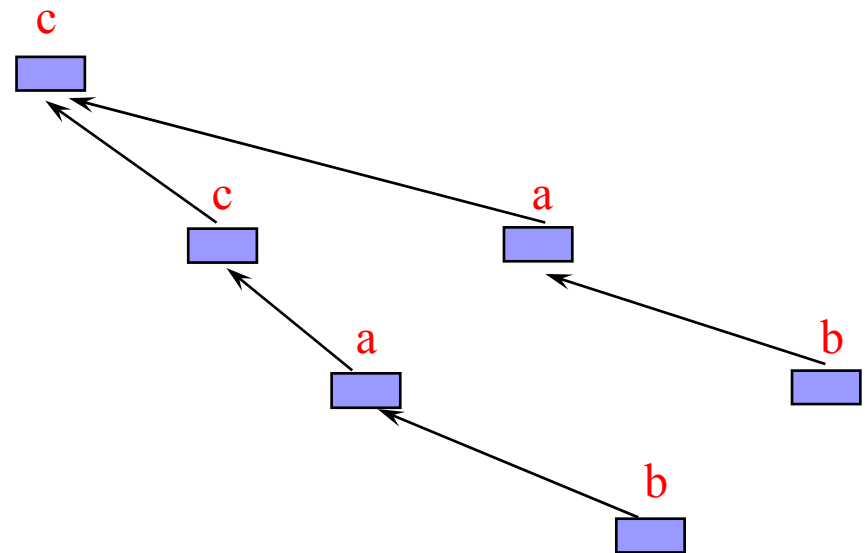
  OK
END_METARULE
```

Data types

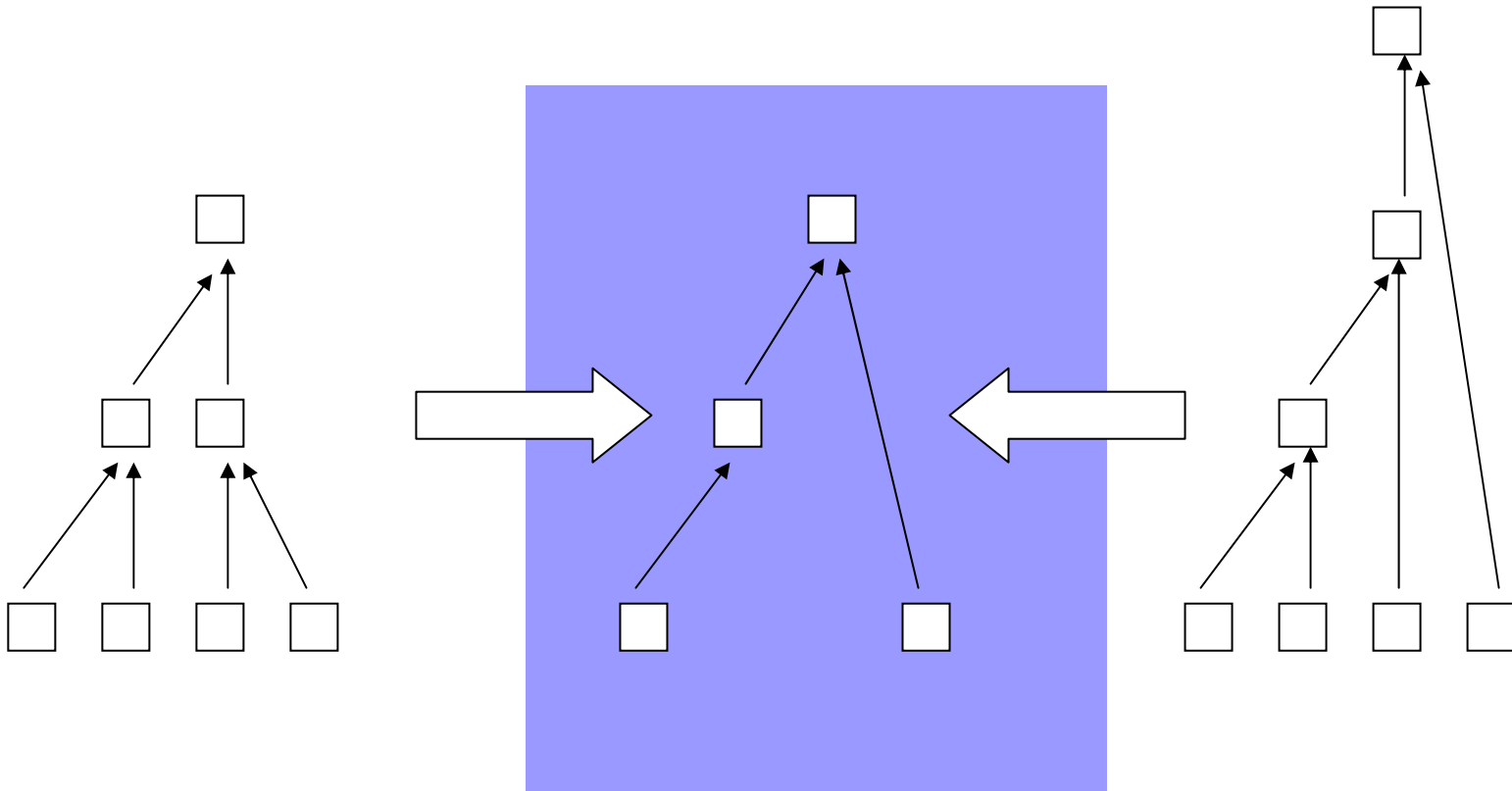
DR (delete-rewrite) trees



D (dependency) trees



Mutual Relationship of D-Trees and DR-Trees



The Grammar Checking According to RFODG

- one metarule of the grammar may at the same time describe a correct and incorrect construction, it is the local and global constraints which decide how it should be interpreted
- the calculation had originally 3 phases:
 - positive projective
 - negative projective or positive non-projective
 - negative non-projective

A newer implementation (Holan 2001) allows even more subtle parametrization of the interpretation (max. No. of non-projective constructions or errors).

The Results of the Project

- a pivot implementation of the grammar checker for Czech working as a macro in MS Word
- a solid basis of further theoretical research of syntactic properties of languages with high degree of word-order freedom
- experience gained in the project helped to better understand what actually is a grammatical error, how to locate and identify them, which errors can and which cannot be automatically found etc.
- follow-up research led to the definition of a reasonable robustness in NLP, to an exact measure of a degree of non-projectivity and to a hypothesis that there is no upper limit of this degree for Czech
- the implementation of the RFODG interpreter helped to formulate an estimation of the parsing complexity of non-projective constructions

Towards the Real Application

Although the pivot implementation could not be used due to the lack of grammar and lexicon coverage, the experiment attracted the attention of Microsoft.

After two failures of their vendor, Lingea, to deliver a reliable grammar checker for Czech in 2000 and 2002, Microsoft initiated the cooperation between some of the people from the original LATESLAV team and the Hungarian company Morphologic. After three years of development Microsoft adopted the result which has become a standard part of the Czech version of Microsoft Office in 2005.

The application was based upon the experience gained in LATESLAV and in other NLP projects as well (rule-based disambiguation of Czech morphology etc.).

LanGR

author of the formalism: P. Květoň 2003

authors of the grammar: V.Petkevič, K.Oliva

Properties:

- developed primarily for the disambiguation of the Czech morphology
- it works with positive and negative disambiguation rules
- the rules may have an unlimited context
- reduction – the aim is to preserve 100% accuracy
- the rules are hand-written, but on the basis of corpus evidence
- The rules are mutually independent, they are not ordered and they are used in cycles
- 4 parts: context, disambiguation part, report and action

cont₁ disamb₁ cont₂ disamb₂ ... cont_n disamb_n cont_{n+1} report action

LanGR

Example of a grammar rule:

```
/* Neither verb, nor preposition, nor conjunction can immediately follow the (in)definite
article */
rule ArtVerbPrepConj2 {
  possart = ITEM Possible Article;
  /* this is a disambiguation area: at least one of the interpretations of the word form possart
must be interpretable as an article */
  safeverbpreconj = ITEM IsSafe Verb or Preposition or Conjunction;
  /* a simple context specifying one corpus element as a verb or preposition or conjunction
only */

  REPORT (The article possart cannot immediately precede the form safeverbpreconj!);

  /* disambiguation actions: article interpretation (tag) in possart is discarded */
  DELETE Article FROM possart;
  /* or */
  LEAVE ONLY not Article IN possart;

}; // end of rule ArtVerbPrepConj2
```

The rule can be successfully applied e.g. to the following sentence:

(2) *The letter a(Article | Noun) from(Preposition) the given alphabet is represented in blue.*

Conclusions

Restarting automata played a key role in the development of the grammar checker of Czech. It provided an adequate mechanism allowing to handle different phenomena in an uniform manner.

It also provided a very good theoretical background for a wide range of topics related to the word order and to the ill-formed input.

The current trends in natural language processing are based on statistical methods, but the story of the Czech grammar checker clearly demonstrates that there are still application areas where an adequate formal tool is more important than terabytes of data. Such a tool may not only help to get deeper insight into a problem, but it may also help to build a real commercial application.