APPLICATION PERSPECTIVES OF SYNCHRONOUS MATRIX GRAMMARS

Petr Horáček
Doctoral Degree Programme (4), FIT BUT
E-mail: xhorac06@stud.fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

Abstract: This paper illustrates the application perspectives of synchronous matrix grammars in formal description of natural languages and their translations. First, it recalls the necessary definitions. Then, using examples from translation between Czech (a free-word-order language with rich inflection) and English, it demonstrates some of the main advantages of synchronous matrix grammars, such as their power and flexibility.

Keywords: synchronous grammar, matrix grammar, regulated rewriting, natural language processing, translation

1 INTRODUCTION

Machine translation is one of the major tasks in natural language processing (NLP). With increasing availability of large corpora, corpus-based systems became favoured over rule-based, using statistical methods and machine-learning techniques. They mostly rely on formal models that represent local information only (for example n-gram models). However, recently, there have been attempts to improve results by incorporating syntactic information (global) into such systems (see [7], [11], [2]).

To do this, we need formal models that can describe syntactic structures and their transformations. In our work, we study well-known models from formal language theory, and extend them for application in NLP, particularly in translation. Based on the principles of synchronous grammars (see [3]), we have previously introduced synchronous versions of some regulated grammars, and we have studied their theoretical propeties (see [6]). In this paper, we further demonstrate the linguistic applications perspectives of synchronous matrix grammars.

2 PRELIMINARIES

We assume that the reader is familiar with the basic aspects of modern FLT (see [10], [8]) and NLP (see [9], [1]). For further information about matrix grammars, see [4].

Definition 1 (Context-free grammar)**.** *A* context-free grammar *(CFG) G is a quadruple $G = (N,T,P,S)$, where N is a finite set of nonterminals, T is a finite set of terminals, $N \cap T = \emptyset$, $P \subset N \times (N \cup T)^*$ is a finite set of rules, $(u,v) \in P$ is written as $u \to v$, and $S \in N$ is the start symbol. Further, let $u,v \in (N \cup T)^*$ and $p = A \to x \in P$. Then, we say that uAv* directly derives *uxv according to p in G, written as $uAv \Rightarrow_G uXv[p]$ or simply $uAv \Rightarrow uxv$. We further define $\Rightarrow^+$ as the transitive closure and $\Rightarrow^*$ as the transitive and reflexive closure of $\Rightarrow$. The* language generated by *G, denoted by $L(G)$, is defined as $L(G) = \{w : w \in T^*, S \Rightarrow^* w\}$.*

Definition 2 (Matrix grammar)**.** *A* matrix grammar *(MAT) H is a pair $H = (G,M)$, where $G = (N,T,P,S)$ is a CFG and M is a finite language over P ($M \subset P^*$) – a sentence of this language is called a* matrix. *Further, for $u,v \in (N \cup T)^*$, $m \in M$ we define $u \Rightarrow v[m]$ in H, if there are strings*

$x_0, \ldots, x_n$ *such that* $u = x_0$, $v = x_n$ *and* $x_0 \Rightarrow x_1\,[p_1] \Rightarrow x_2\,[p_2] \Rightarrow \ldots \Rightarrow x_n\,[p_n]$ *in* G, *and* $m = p_1 \ldots p_n$. *The* language generated by H, *denoted by* $L(H)$, *is defined as* $L(H) = \{w : w \in T^*, S \Rightarrow^* w\}$.

3 SYNCHRONOUS MATRIX GRAMMARS

This section recalls basic definitions from [6].

Definition 3 (Synchronous matrix grammar)**.** *A* synchronous matrix grammar *(SMAT)* H *is a 7-tuple* $H = (G_I, M_I, G_O, M_O, \Psi, \varphi_I, \varphi_O)$, *where* (G_I, M_I) *and* (G_O, M_O) *are matrix grammars,* Ψ *is a set of* matrix labels*, and* φ_I *is a function from* Ψ *to* M_I *and* φ_O *is a function from* Ψ *to* M_O. *Further, the* translation defined by H, *denoted by* $T(H)$, *is the set of pairs of sentences, which is defined as* $T(H) = \{(w_I, w_O) : w_I \in T_I^*, w_O \in T_O^*, S_I \Rightarrow^*_{(G_I, M_I)} w_I\,[\alpha], S_O \Rightarrow^*_{(G_O, M_O)} w_O\,[\alpha], \alpha \in \Psi^*\}$.

Informally, SMAT is a system of two MATs with linked matrices. The linking is done by shared labels. Then, in every derivation step in SMAT, we make a derivation step in each MAT, and the matrices applied in both MATs must have the same label. In other words, the input and output sentence must have the same parse, which is the sequence of matrices (denoted by their labels) applied in order to generate the sentences.

4 LINGUISTIC APPLICATION PERSPECTIVES

To illustrate some of the main advantages of synchronous matrix grammars, we will consider translation between Czech and English. Czech is a relatively challenging language in terms of NLP. It is a free-word-order language with rich inflection (see [5]).

For example, consider the Czech sentence *dva růžoví sloni přišli na přednášku* (*two pink elephants came to the lecture*). All of the following permutations of words also make for a valid sentence:

dva růžoví sloni přišli na přednášku	*dva růžoví sloni na přednášku přišli*
růžoví sloni přišli na přednášku dva	*růžoví sloni na přednášku přišli dva*
dva sloni přišli na přednášku růžoví	*dva sloni na přednášku přišli růžoví*
sloni přišli na přednášku dva růžoví	*sloni na přednášku přišli dva růžoví*

There may be differences in meaning or emphasis, but the syntactic structure remains the same. Why is this problematic? Compare the syntax trees in Figure 1. Because of the crossing branches, the second tree cannot be produced by any CFG. Of course, it is still possible to construct a CFG that generates the sentence *růžoví sloni přišli na přednášku dva* if we consider a different syntax tree, for example such as in Figure 2. However, this tree no longer captures the relation between the noun *sloni* and its modifying numeral *dva* (represented by the dotted line). We need to know this relation for instance to ensure agreement between the words (person, number, gender…), so that we can choose their appropriate forms.

In a purely context-free framework, this is complicated. The necessary information has to be propagated through the tree, even if the structure is not actually affected. This can result in a high number of rules. With MAT, we can instead represent the relations using matrices.

Here, we present an example of SMAT $H = (G_{cz}, M_{cz}, G_{en}, M_{en}, \Psi, \varphi_{cz}, \varphi_{en})$ that describes the translations between the English sentence *two pink elephants came to the lecture* and any of the above Czech sentences, correctly distinguishing between male and female gender (to demonstrate female gender, we also include *opice* in Czech, *monkeys* in English). H is designed to allow for easy extension to include other grammatical categories (person…) and also different syntactic structures.

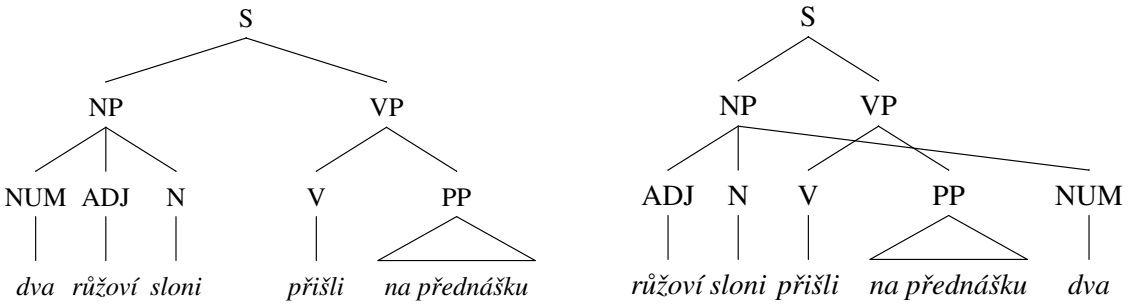

Figure 1: Syntax trees for example sentences

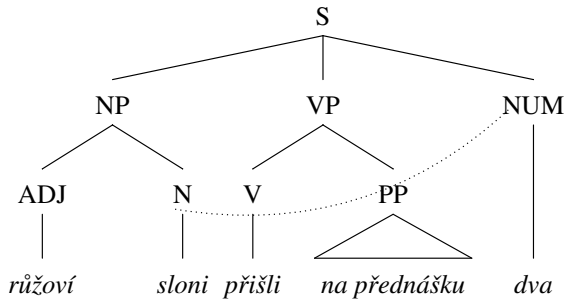

Figure 2: Modified syntax tree

Context-free rules in G_{cz} (Czech), nonterminals are in capitals, S is the start symbol:

s :	S	→	NP VP NUM ADJS ,	np :	NP	→	NUM ADJS N ,
vp :	VP	→	ADVS V ADVS ,	num_ε :	NUM	→	ε ,
adjs :	ADJS	→	ADJ ADJS ,	adjs_ε :	ADJS	→	ε ,
advs :	ADVS	→	ADV ADVS ,	advs_ε :	ADVS	→	ε ,
n_m :	N	→	N_m ,	n_f :	N	→	N_f ,
n_{mm} :	N_m	→	N_m ,	n_{ff} :	N_f	→	N_f ,
v_m :	V	→	V_m ,	v_f :	V	→	V_f ,
adj_m :	ADJ	→	ADJ_m ,	adj_f :	ADJ	→	ADJ_f ,
adv :	ADV	→	PP ,	num_m :	NUM	→	NUM_m ,
num_f :	NUM	→	NUM_f ,	dict_1 :	N_m	→	*sloni* ,
dict_2 :	N_f	→	*opice* ,	dict_{3m} :	V_m	→	*přišli* ,
dict_{3f} :	V_f	→	*přišly* ,	dict_{4m} :	ADJ_m	→	*růžoví* ,
dict_{4f} :	ADJ_f	→	*růžové* ,	dict_{5m} :	NUM_m	→	*dva* ,
dict_{5f} :	NUM_f	→	*dvě* ,	dict_6 :	PP	→	*na přednášku*

Context-free rules in G_{en} (English), nonterminals are in capitals, S is the start symbol:

s :	S	→	NP VP ,	np :	NP	→	NUM ADJS N ,
vp :	VP	→	V ADVS ,	num_ε :	NUM	→	ε ,
adjs :	ADJS	→	ADJ ADJS ,	adjs_ε :	ADJS	→	ε ,
advs :	ADVS	→	ADV ADVS ,	advs_ε :	ADVS	→	ε ,
adv :	ADV	→	PP ,	dict_1 :	N	→	*elephants* ,
dict_2 :	N	→	*monkeys* ,	dict_3 :	V	→	*came* ,
dict_4 :	ADJ	→	*pink* ,	dict_5 :	NUM	→	*two* ,
dict_6 :	PP	→	*to the lecture*				

Matrices:

	M_{cz}	M_{en}		M_{cz}	M_{en}		M_{cz}	M_{en}
s :	s	s	np :	np	np	vp :	vp	vp
num :	num_ε	ε	$num_{\varepsilon\varepsilon}$:	num_ε num_ε	num_ε	$adjs$:	adjs	adjs
$adjs_\varepsilon$:	adjs_ε adjs_ε	adjs_ε	$advs$:	adjs	adjs	$advs_\varepsilon$:	advs_ε advs_ε	advs_ε
n_m :	n_m	ε	n_f :	n_f	ε	v_m :	v_m n_{mm}	ε
v_f :	v_f n_{ff}	ε	adj_m :	adj_m n_{mm}	ε	adj_f :	adj_f n_{ff}	ε
adv :	adv	adv	num_m :	num_m n_{mm}	ε	num_f :	num_f n_{ff}	ε
$dict_1$:	dict_1	dict_1	$dict_2$:	dict_2	dict_2	$dict_{3m}$:	dict_{3m}	dict_3
$dict_{3f}$:	dict_{3f}	dict_3	$dict_{4m}$:	dict_{4m}	dict_4	$dict_{4f}$:	dict_{4f}	dict_4
$dict_{5m}$:	dict_{5m}	dict_5	$dict_{5f}$:	dict_{5f}	dict_5	$dict_6$:	dict_6	dict_6

Note for example the matrix adj_f in M_{cz}, which ensures agreement between noun and adjective (both must be in female gender). Also note that the linked matrices (sharing the same label) in M_{cz} and M_{en} may contain completely different rules and in some cases one can even be empty (ε).

Example of a derivation in Czech follows:

S $\Rightarrow$ NP VP NUM ADJS $[s]$ $\Rightarrow$ NUM ADJS N VP NUM ADJS $[np]$ $\Rightarrow$ NUM ADJS N ADVS V ADVS NUM ADJS $[vp]$ $\Rightarrow$ ADJS N ADVS V ADVS NUM ADJS $[num]$ $\Rightarrow$ ADJ ADJS N ADVS V ADVS NUM ADJS $[adjs]$ $\Rightarrow$ ADJ N ADVS V ADVS NUM $[adjs_\varepsilon]$ $\Rightarrow$ ADJ N ADVS V ADV ADVS NUM $[advs]$ $\Rightarrow$ ADJ N V ADV NUM $[advs_\varepsilon]$ $\Rightarrow$ ADJ N_m V ADV NUM $[n_m]$ $\Rightarrow$ ADJ N_m V_m ADV NUM $[v_m]$ $\Rightarrow$ ADJ_m N_m V_m ADV NUM $[adj_m]$ $\Rightarrow$ ADJ_m N_m V_m PP NUM $[adv]$ $\Rightarrow$ ADJ_m N_m V_m PP NUM_m $[num_m]$ $\Rightarrow$ ADJ_m *sloni* V_m PP NUM_m $[dict_1]$ $\Rightarrow$ ADJ_m *sloni přišli* PP NUM_m $[dict_{3m}]$ $\Rightarrow$ *růžoví sloni přišli* PP NUM_m $[dict_{4m}]$ $\Rightarrow$ *růžoví sloni přišli na přednášku* NUM_m $[dict_5]$ $\Rightarrow$ *růžoví sloni přišli na přednášku dva* $[dict_{6m}]$

The corresponding derivation in English may look like this:

S $\Rightarrow$ NP VP $[s]$ $\Rightarrow$ NUM ADJS N VP $[np]$ $\Rightarrow$ NUM ADJS N V ADVS $[vp]$ $\Rightarrow$ NUM ADJS N V ADVS $[num]$ $\Rightarrow$ NUM ADJ ADJS N V ADVS $[adjs]$ $\Rightarrow$ NUM ADJ N V ADVS $[adjs_\varepsilon]$ $\Rightarrow$ NUM ADJ N V ADV ADVS $[advs]$ $\Rightarrow$ NUM ADJ N V ADV $[advs_\varepsilon]$ $\Rightarrow$ NUM ADJ N V ADV $[n_m]$ $\Rightarrow$ NUM ADJ N V ADV $[v_m]$ $\Rightarrow$ NUM ADJ N V ADV $[adj_m]$ $\Rightarrow$ NUM ADJ N V PP $[adv]$ $\Rightarrow$ NUM ADJ N V PP $[num_m]$ $\Rightarrow$ NUM ADJ *elephants* V_m PP $[dict_1]$ $\Rightarrow$ NUM ADJ *elephants came* PP $[dict_{3m}]$ $\Rightarrow$ NUM *pink elephants came* PP $[dict_{4m}]$ $\Rightarrow$ NUM *pink elephants came to the lecture* $[dict_5]$ $\Rightarrow$ *two pink elephants came to the lecture* $[dict_{6m}]$

The entire derivation tree for the Czech sentence is shown in Figure 3. The dotted lines represent relations described by matrices. The triangle from N_m to N_m is an abstraction which in this particular case essentially means that this step is repeated until all agreement issues are resolved.

5 CONCLUSION

We have illustrated the main advantages of synchronous matrix grammars with regard to their application perspectives in natural language translation. Their power and flexibility allow us to capture even some of the more problematic features of natural languages (such as free word order and rich inflection in Czech) efficiently. For future work, parsing algorithms for (synchronous) matrix grammars still remain an open problem.

ACKNOWLEDGEMENT

This work was partially supported by the BUT FIT grant FIT-S-11-2 and the research plan MSM 0021630528.

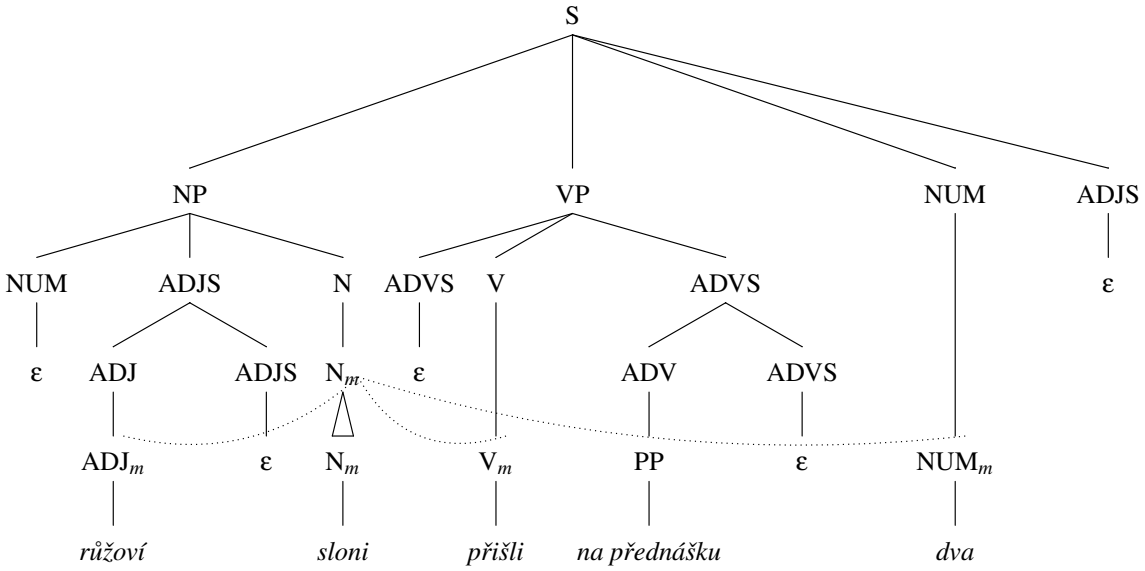

Figure 3: Derivation tree

REFERENCES

[1] Allen, J.: *Natural Language Understanding* (2nd ed). Benjamin-Cummings Publishing Company, Redwood City, CA, USA, 1995.

[2] Bojar, O., Čmejrek, M.: *Mathematical Model of Tree Transformations* [online]. EuroMatrix Deliverable, 2007, `http://ufal.mff.cuni.cz/euromatrix/`

[3] Chiang, D.: An Introduction to Synchronous Grammars [online]. Part of a tutorial given at *44th Annual Meeting of the Association for Computational Linguistics*, 2006, `http://www.isi.edu/~chiang/papers/synchtut.pdf`

[4] Dassow, J., Păun, Gh.: *Regulated Rewriting in Formal Language Theory*. Springer, 1989.

[5] Hajič, J.: *Disambiguation of Rich Inflection: Computational Morphology of Czech*. Wisconsin Center for Pushkin Studies, Karolinum, 2004.

[6] Horáček, P., Meduna, A.: Synchronous Versions of Regulated Grammars: Power and Linguistic Applications. In: *Theoretical and Applied Informatics*, Vol. 24, no. 3, 2012, p. 175–190

[7] Khalilov, M., Fonollosa, J.A.R.: N-gram-based Statistical Machine Translation versus Syntax Augmented Machine Translation: comparison and system combination. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2009, p. 424–432.

[8] Meduna, A.: *Automata and Languages: Theory and Applications*. Springer, 2005, ISBN 1-85233-074-0, 892 p.

[9] Mitkov, R. (ed.): *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2004, ISBN 978-0-19-927634-9.

[10] Rozenberg, G., Salomaa, A.: *Handbook of Formal Languages: Volume I*. Springer, 1997.

[11] Zollmann, A., Venugopal, A.: Syntax Augmented Machine Translation via Chart Parsing. In: *Proceedings of the Workshop on Statistical Machine Translation*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, p. 138–141.