Параллельное создание славянских грамматических ресурсов

Parallel construction of Slavic grammatical resources

Avgustinova Tania (avgustinova@coli.uni-saarland.de)
DFKI GmbH & Saarland University

Представлена идея параллельного создания компьютерных грамматик для славянских языков на основе формализма HPSG с использованием общеславянского модуля и совместимых с ним расширений для отдельных языков. Важным требованием к проекту является динамичная связь между создаваемыми грамматиками и синтаксически размеченными корпусами.

1. Introduction

Our long-term goal is to develop grammatical resources for Slavic languages and to make them freely available for the purposes of research, teaching and natural language applications. We build upon our previous research in language-family oriented grammar design and systematic specification of shared and non-shared grammar for modeling Slavic morphosyntax. An imperative objective in this context concerns what we see as methodical corpus-based grammar elaboration. To this effect, we envisage exploiting freely available linguistically interpreted corpora at all stages of the project and especially in discovering structured knowledge to be reflected in our grammars. Interfacing a morphological analyzer is a crucial prerequisite for any grammar development activity involving Slavic languages. For research purposes, such systems are by and large freely available nowadays, and the grammar engineering environment we plan to use provides the required interface for integrating a morphological preprocessor. An important desideratum for the individual resource grammars is to eventually couple them with syntactically interpreted text corpora (treebanks) which either pre-exist or will be constructed in parallel. The development of Slavic resource grammars is part of an on-going international collaborative effort which became popular under the name DELPH-IN.[1] It is based on a shared commitment to re-usable, multi-purpose resources and active exchange. Our project utilizes DELPH-IN software (linguistic knowledge builder[2] with integrated evaluation and benchmarking

[1] Deep Linguistic Processing with HPSG Initiative, URL: http://www.delph-in.net/

[2] The LKB (Linguistic Knowledge Builder) system is a grammar and lexicon development environment for use with unification-based linguistic formalisms. While not restricted to HPSG,

tools[3]) as a grammar development platform, and has strong affinity to the LinGO[4] Grammar Matrix. We envision a core Slavic grammar whose components can be commonly shared among the set of languages, and facilitate individual resource grammar development.

The Slavic core grammar is intended to encode mutually interoperable analyses of a wide variety of linguistic phenomena, taking into account eminent typological commonalities and systematic differences. Determining what counts as a worthwhile linguistic phenomenon is a challenge in its own right. for a corresponding operational notion, however, it would suffice to conclusively reflect the fact that what grammatical representations have in common, independently of their theoretical origin or purpose, is that they (i) identify linguistic items of different motivation and complexity, (ii) encode their properties, and (iii) specify explicit or implicit relationships between them. As the interconnectedness of grammatical phenomena is at the heart of research in theoretical syntax, one of our objectives is to contribute a language-family oriented perspective to the data-driven cross-linguistic exploration of that interconnection. Our concept of Slavic core grammar will shape up and crystallize through rigorous testing in parallel grammar engineering for a closed set of languages for which a variety of linguistic resources is already available. All individual grammars will be designed to support an innovative implementation of a Slavic core module that consolidates strategies for constructing a cross-linguistic resource.

2. Background

Rule-based precision grammars are linguistic resources designed to model human languages as accurately as possible. Unlike statistical grammars, they are handbuilt and take into account the respective grammarian's theory and analysis of how to best represent various syntactic and semantic phenomena in the language of interest. A side effect of this is that such grammars tend to substantially differ from each other, with no established best practices or common representations.[5] As implementations evolved for several languages within the formalism of Head-driven Phrase Structure Grammar (Pollard and Sag 1994), it became clear that homogeneity among

---

the LKB implements the DELPH-IN reference formalism of typed feature structures (jointly with other DELPH-IN software using the same formalism).
URL: http://wiki.delph-in.net/moin/LkbTop

[3] [incr tsdb()] — URL: http://www.delph-in.net/itsdb/

[4] The Linguistic Grammars Online (LinGO) team is committed to the development of linguistically precise grammars based on the HPSG framework, and general-purpose tools for use in grammar engineering, profiling, parsing and generation. URL: http://lingo.stanford.edu/

[5] Exceptions do exist, of course: ParGram (Parallel Grammar) project is one example of multiple grammars developed using a common standard. It aims at producing wide coverage grammars for a wide variety of languages. These are written collaboratively within the linguistic framework of Lexical Functional Grammar (LFG) and with a commonly-agreed-upon set of grammatical features. URL: http://www2.parc.com/isl/groups/nltt/pargram/

existing grammars could be increased and development cost for new grammars greatly reduced by compiling an inventory of cross-linguistically valid (or at least useful) types and constructions. Hence the LinGO Grammar Matrix has been set up as a multi-lingual grammar engineering project (Bender et al. 2002) in an attempt to distil the wisdom of already existing broad coverage grammars and document it in a form that can be used as the basis for new grammars. The generalizations observed across linguistic objects and across languages result in a cross-linguistic type hierarchy[6] coming with a collection of phenomenon-specific libraries, which would optimally represent salient dimensions of cross-linguistic variation.

The original Grammar Matrix consisted of types defining the basic feature geometry (Copestake et al. 2001), types for lexical and syntactic rules encoding the ways that heads combine with arguments and adjuncts, and configuration files for the LKB grammar development environment (Copestake 2002) and the PET system (Callmeier 2000). Subsequent releases have refined the original types and developed a lexical hierarchy, including linking types for relating syntactic to semantic arguments, and the constraints required to compositionally build up semantic representations in the format of Minimal Recursion Semantics (Copestake et al. 2005; Flickinger and Bender 2003; Flickinger et al. 2003). These constraints are intended to be language-independent and monotonically extensible in any given grammar. In its recent development, the Grammar Matrix project aims at employing typologically motivated, customizable extensions to a language-independent core grammar (Bender and Flickinger 2005) to handle cross-linguistically variable but still recurring patterns. A web-based configuration tool eliciting typological information from users-linguists through a questionnaire is currently under active construction. While users specify phenomena relevant to their particular language, the resulting selections are compiled from libraries of available analyses into starter grammars which can be immediately loaded into the LKB environment in order to parse sentences using the rules and constraints defined therein. The regression testing facilities of [incr tsdb()] allow for rapid experimentation with alternative analyses as new phenomena are brought into the grammars (Oepen et al. 2002). The ultimate ambition is thus to allow the linguist to revise decisions in the face of new information or improved linguistic analyses. Apart from the shared 'core' in the Grammar Matrix the customization script treats the individual languages as separate instances, which is rather insufficient for our purposes. Because it is driven purely by the specific phenomena in the target language, this strategy is consistent with "bottom-up" data driven investigation of linguistic universals and constraints on cross-linguistic variation. Obviously, the fact that we have to do with a group of systematically related languages cannot be taken into account in the original setting. With grammars being created individually, the treatment of shared phenomena would work to the degree that satisfies but does not guarantee cross-linguistic compatibility. It is a legitimate expectation,

[6] In a lexicalized constraint-based framework, the grammars are expressed as a collection of typed feature structures which are arranged into a hierarchy such that information shared across multiple lexical entries or construction types is represented only on a single super-type.

though, that the constraint definitions supplied to grammar developers can be extended to also capture generalizations holding only for subsets of languages. It is essential therefore to augment the approach with a "top-down" perspective introducing intermediate levels of typological variation.

3. Shared grammar

Successful multilingual natural language processing systems employ generic linguistic resources that are adaptable to specific language and application requirements. If parallel grammars for more than one language are needed for an application like machine translation or computer-assisted language learning, it pays off to define and implement shared grammars. The reuse of portions of grammars for the description of additional languages speeds up grammar development which is a demanding and time consuming task. A shared grammar approach not only facilitates the difficult task of maintaining consistency within and across the individual parallel grammars, but it also strongly supports for the area of natural language processing the prospects of what in programming language research is called modularity. In applied computational linguistics the need for employing operational notions of shared grammar stems from multilingual grammar engineering — cf. projects like (DiET 1997–1999; LinGO 2002; LS-GRAM 1994–1996; ParGram 1995–2002; TSNLP 1993–1995; XTAG 2002). Computational linguists engaged in multilingual grammar development have always tried to reduce their labour by importing existing grammar components in a simple copy-paste-modify fashion. But there were also a number of systematic attempts to create and describe shared grammars that are convincingly documented in publications. (Kameyama 1988) demonstrates the concept for a relatively restricted domain, the grammatical description of simple nominal expressions in five languages. (Bemová et al. 1988) were able to exploit the grammatical overlap of two Slavic languages, for the design of a lean transfer process in Russian to Czech machine translation. In multilingual application development within Microsoft research, grammar sharing has extensively been exploited (Gamon et al. 1997; Pinkham 1996). Current international collaborative efforts within the DELPH-IN partnership (Uszkoreit et al. 2001; Uszkoreit 2002a; b) exploit the notion of shared grammar both for the rapid development of grammars for new languages and for the systematic adaptation of grammars to variants of languages. The leading idea is to combine linguistic and statistical processing methods for getting at the meaning of texts and utterances. Based on contributions from several members and joint development over many years, an open-source repository of software and linguistic resources has been created that already enjoys wide usage in education, research, and application building.

The construction of shared-grammar fragments proposed in (Avgustinova 2007) presupposes a common core module which is abstract enough to be shared by all Slavic languages modulo the appropriate further specification. For a language family, this module is expected to be relatively large and to cover the major phenomena areas. Certainly, there are groups and sub-groups of languages exhibiting particular properties and phenomena which are not attested in other members of the family. Yet, these phenomena constitute natural extensions of the common core module. So, for instance, one could

distinguish a South-Slavic extension or an East-Slavic extension, and possibly extensions of any further granularity. Nevertheless, there are language-specific traits that identify specific languages and dialects. While the common core module is expected to be relatively large and to cover all major phenomena areas, it has still to be abstract enough in order to be shared by all Slavic languages modulo the appropriate further specification. Intuitively, the core incorporates what is interpretable as typical Slavic. The extensions can be of different granularity in order to encode properties and phenomena that are characteristic of respective subgroups, but need not be attested in other members of the family. Yet, all these phenomena have to be consistent with the common core module, constituting natural extensions. For example, a modular Bulgarian grammar in such a setting would include the common core, the South-Slavic extension, and the Bulgarian extension.

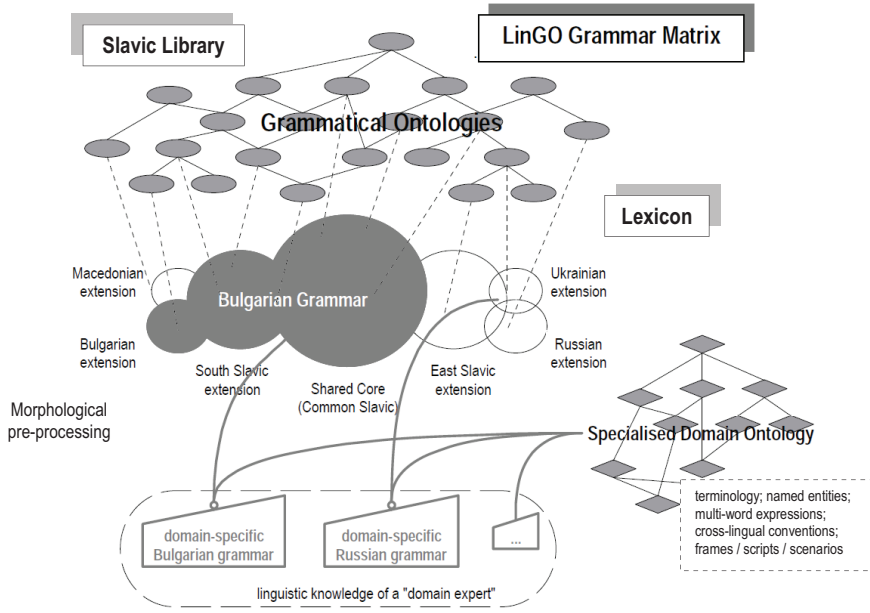

Figure 1. Resource architecture

Methodologically, the adopted shared-grammar perspective reveals two different aspects of structuring the grammatical knowledge. One is cross-linguistic and can by and large be viewed as consisting of an under-specified core and a competence-driven specification "switching on" various parameters. The other aspect of structuring the grammatical knowledge can be called intra-linguistic and concerns the interesting interaction with specialized domain ontologies that model the expert knowledge in particular subject domains like electrical engineering or microbiology. It is never the case that the entire power of a natural language grammar is employed in restricted specialized areas. A domain-specific grammar extraction relies, therefore, on specialized text corpora. A rather naive approach would be to assume

an over-specified full-fledged grammar of a given language in combination with a performance-driven relaxation "switching off" various parameters. More attractive, however, is the idea that specific domain ontologies interact with grammatical ontologies to derive restricted grammars of, e. g., Russian or Bulgarian as used in, e. g., electrical engineering or microbiology. A theoretically rewarding result is the straightforward model of the linguistic knowledge of a domain expert. For example, an electrical engineer or a microbiologist hardly needs to be fluent in all languages he uses in order to be effectively multilingual in his restricted subject domain.

4. Grammar resource development

Any approach to computational grammar design which maintains the notion of grammar sharing lends itself to a formal linguistic description of individual languages as well as groups of languages motivated by genetic origin or areal contact. The strategy we adopt is meant by design to be compatible with the current Grammar Matrix program: we use the customization system to quickly build small grammars for individual languages; shared analyses are put into a Slavic core; when the next language is added, the Slavic core helps to more efficiently build the new grammar, simultaneously receiving a cross-Slavic validation. Yet, a distinctive feature of our approach to Slavic grammatical resources is that grammar engineering for each individual language takes place in a common Slavic setting. This in particular means that if, for example, two possibilities are conceivable of how to model a particular phenomenon observed in a certain Slavic language we strongly prefer the option that would potentially be consistent with what is found in the other grammars. The reason is that related languages share a much wider range of linguistic information than typically assumed in standard multilingual grammar architectures. We can, as a result, directly and effectively work with what has traditionally been regarded as "prototypically Slavic".

Currently we focus on the Russian resource grammar as a showcase on how its development can be assisted by interfacing with existing corpora and processing tools for the language (Avgustinova and Zhang 2009a; b; c; d; e). Applying the innovative corpus-oriented grammar development approach proposed by (Miyao et al. 2005) to the syntactically annotated and manually disambiguated part of the Russian National Corpus (Boguslavsky et al. 2000; Boguslavsky et al. 2002) we can obtain a unique Russian HPSG-style treebank (Avgustinova and Zhang 2010). We have also been looking into data-driven approaches of dependency parsing with the SynTagRus treebank. Specifically, we have rebuilt the transition-based dependency parsing models and cross-compare results with those reported in (Nivre et al. 2008). Subsequently we shall concentrate on resource grammars for Bulgarian and Polish, thus including representatives of the three main subgroups in the Slavic language family: East Slavic (Russian), South Slavic (Bulgarian); West Slavic (Polish).

For an illustration let us consider the Slavic case system, because it provides abundant scope for complex categorisation with many cross-linguistic tendencies. From the morphological perspective, there exists a spectrum of case marking possibilities.

The most common and typical are the synthetic means like suffixation and inflexion, possibly in combination with supra-segmental distinctions. Apart from that, the case marking can involve analytical adpositional means like prepositions or postpositions. A further case marking possibility is the suppletion of forms (e. g., in pronominal paradigms), where the ultimate union of stem and case can be observed. Syntactically, there are two general ways in which case is acquired by the respective case-marked category: in concord (due to case-matching between a governor and a dependent) or under government (via non-congruent, non-agreeing case selection). This naturally results in distinguishing *concordial case* and *relational case*. Even though the term "case marking" traditionally refers to inflectional marking, it could successfully be extended to cover adpositions.

The majority of Slavic languages exhibits a rich inflectional case marking system and is traditionally classed among the *synthetic* languages. Characteristic of the synthetic language type is that prepositions, like verbs, *govern* cases. Moreover, relational cases can be expressed by the combination of a preposition and case inflection. Consider, for example, Russian prepositions such as *v* ('in'), *na* ('on'), *pod* ('under'), etc. Their combination with the locative/prepositional case inflection encodes location, while their combination with the accusative case inflection expresses direction. In the *analytic* language type, the situation is rather different. Adpositions bear the sole burden of marking the relations and, thus, of expressing relational cases. For example, Bulgarian prepositions *combine* with the oblique form of the substantive, whereby any suffix or inflection, if available, is redundant with respect to case marking. Also, recall in this context that the above-mentioned opposition direction vs. location is altogether lost in this Slavic language.

To respond to the need of morphosyntactic abstraction over regular case variation and language-specific constraints with respect to case marking, the notion of *functional case* is employed. A shared Slavic case taxonomy encoded as a multiple inheritance hierarchy is sketched below — for detailed motivation cf. (Avgustinova 2007) p. 25–34.

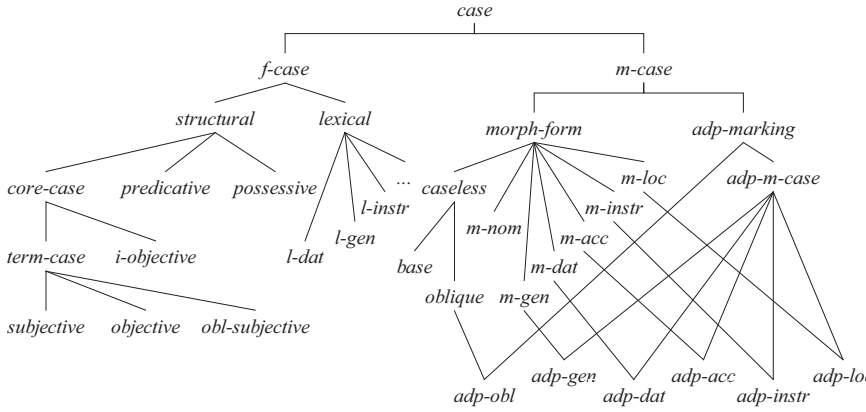

Figure 2. Slavic case system

The type *case* is classified along two dimensions: functional (*f-case*) and marking (*m-case*). Two types of relational (subcategorised) cases are assumed: *lexical* (inherent) and *structural* (syntactic, grammatical). These disjoint types partition the type *case* along the functional dimension, hence, are subtypes of *f-case* in the type hierarchy sketched below. This in particular means that in the lexical entry of a verb, the case value of subcategorised nominal categories is either specific, lexically predetermined or systematically under-specified, i. e. to be resolved by grammatical constraints / principles. A case value specified as *structural* is to be further instantiated to a particular (more specific) instance of *functional case*, namely, *subjective*, *objective* and *obl-subjective* (that is, the case specification of passivised subjects). These abstract case values will be expanded to their concrete instances on the basis of lexical and contextual constraints, taking into consideration the relevant (language-specific) morphological and adpositional case marking. An important aspect of such an approach is that the distinction between lexical and structural cases is applied not only to morphological, but also to adpositional case marking. In particular, the marking dimension of the case hierarchy is refined by allowing classification of *m-case* according to morphological form (*morph-form*) and adpositional marking (*adp-marking*). The type *morph-form* extends further either to concrete case inflection, i. e. to morphological nominative (*m-nom*), morphological genitive (*m-gen*), morphological dative (*m-dat*), and so on, or just to *caseless*, i. e. to *base* or *oblique* morphological form — in the case of Bulgarian nouns. The type *adp-marking* encodes the adpositional marking on the noun, i. e. the respective PP; in particular, it interacts with *morph-form* in specifying the types *adp-oblique* (for Bulgarian) and *adp-m-case* (for other Slavic languages).

5. Outlook

The formal specification of shared grammar is also extremely important for developing stringent models of language change. Historical linguistics and sociolinguistics need formal models of grammar in which possible and factual shared grammars can be specified. It is a justified expectation that a formal notion of shared grammar should also be useful for theoretical and applied work on second-language acquisition. The precise specification of shared grammar could explain preferences of the second-language learner as well as contamination and interference phenomena. Designing specialized methodologies for second language learning that take into account the properties of the learner's first language could likewise benefit from a good description of shared grammar.

References

1. *Avgustinova T.,* 2007. Language Family Oriented Perspective in Multilingual Grammar Design. Peter Lang — Europäischer Verlag der Wissenschaft, Frankfurt am Main, Berlin, Bern, Bruxelles, New York, Oxford, Wien.

2. *Avgustinova T., Zhang Y.,* 2009a. Parallel Grammar Engineering for Slavic Languages. In: Workshop on Grammar Engineering Across Frameworks at the ACL/IJCNLP 2009 Conference, Singapore.
3. *Avgustinova T., Zhang Y.,* 2009b. Developing a Russian HPSG based on the Russian National Corpus. In: DELPH-IN Summit, Barcelona.
4. *Avgustinova T., Zhang Y.,* 2009c. Exploiting the Russian National Copus in the Development of a Russian Resource Grammar. In: Workshop on Adaptation of Language Resources and Technology to New Domains at the RANLP 2009 Conference, Borovets, Bulgaria.
5. *Avgustinova T., Zhang Y.,* 2009d. Exploiting the Russian National Corpus in the Development of a Russian Resource Grammar. In: Proceedings of the RANLP-2009 Workshop on Adaptation of Language Resources and Technology to New Domains, Borovets, Bulgaria.
6. *Avgustinova T., Zhang Y.,* 2009e. Parallel Grammar Engineering for Slavic Languages. In: Workshop on Grammar Engineering Across Frameworks at the ACL/IJCNLP, Singapore.
7. *Avgustinova T., Zhang Y.,* 2010. Conversion of a Russian dependency treebank into HPSG derivations. In: Proceedings of the 9th International Workshop on Treebanks and Linguistic Theories (TLT'9), Tartu, Estonia.
8. *Bemová A., Oliva K., Panevová J., 1988. Some problems of machine translation between closely related languages. In: COLING'88, Budapest.*
9. *Bender E. M., Flickinger D., Oepen S.,* 2002. The Grammar Matrix: An Open-Source-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In: Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics, Carroll J., Oostdijk N., Sutcliffe R. (Eds.), Taipei, Taiwan, pp. 8–14.
10. *Bender E. M., Flickinger D.,* 2005. Rapid Prototyping of Scalable Grammars: Towards Modularity in Extensions to a Language-Independent Core. In: 2nd International Joint Conference on Natural Language Processing, Jeju, Korea.
11. *Boguslavsky I., Grigorjeva S., Grigorjev N., Kreidlin L., Frid N.,* 2000. Dependency treebank for Russian: Concept, tools, types of information. In: COLING, pp. 987–991.
12. *Boguslavsky I., Chardin I., Grigorjeva S., Grigoriev N., Iomdin L., Kreidlin L., Frid., N.,* 2002. Development of a dependency treebank for Russian and its possible applications in NLP. In: Third International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas, pp. 852–856.
13. *Callmeier U.,* 2000. PET — a platform for experimentation with efficient HPSG processing techniques. Natural Language Engineering 6 99–107
14. *Copestake A., Lascarides A., Flickinger D.,* 2001. An algebra for semantic construction in constraint-based grammars. In: The 39th Meeting of the Association for Computational Linguistics, Toulouse, France.
15. *Copestake A.,* 2002. Implementing Typed Feature Structure Grammars.
16. *Copestak, A., Flickinger D., Sag I. A., Pollard C.,* 2005. Minimal Recursion Semantics: An Introduction. Journal of Research on Language and Computation 3 (4), 281–332.

17. *DiET,* 1997–1999. Diagnostic and Evaluation Tools for Natural Languages Applications. http://diet.dfki.de/.
18. *Flickinger D., Bender E. M.,* 2003. Compositional Semantics in a Multilingual Grammar Resource. In: ESSLLI Workshop on Ideas and Strategies for Multilingual Grammar Development, pp. 33–42.
19. *Flickinger D., Bender E. M., Oepen S.,* 2003. MRS in the LinGO Grammar Matrix: A Practical User's Guide.
20. *Gamon M., Lozano C., Pinkham J., Reutter T.,* 1997. Practical experiencer with grammar sharing in multilingual NLP. Microsoft, Redmond.
21. *Kameyama M.,* 1988. Atomization in grammar sharing. In: 26th Annual Meeting of ACL, New York.
22. *LinGO,* 2002. Linguistic Grammars Online. http://lingo.stanford.edu/.
23. *LS-GRAM,* 1994–1996. Large-Scale Grammatical Resources. http://clwww.essex.ac.uk/group/projects/lsgram/.
24. *Miyao Y., Ninomiya T., Tsujii J.,* 2005. Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In: Natural Language Processing — IJCNLP 2004, LNAI3248, Hainan Island, China, Su, K.-Y., Tsujii, J.i., Lee, J.-H., Kwong, O. Y. (Eds.), Springer-Verlag, pp. 684–693.
25. *Nivre J., Boguslavsky I., Iomdin L.,* 2008. Parsing the SynTagRus Treebank. In: COLING, pp. 641–648.
26. *Oepen S., Toutanova K., Shieber S., Manning C., Flickinger D., Brants T.,* 2002. The LinGO Redwoods treebank. Motivation and preliminary applications. In: 19th International Conference on Computational Linguistics, Taipei, Taiwan.
27. *ParGram,* 1995–2002. Parallel Grammar Project. http://www.parc.xerox.com/istl/groups/nltt/pargram/.
28. *Pinkham J.,* 1996. Grammar sharing in French and English. In: IANLP'96.
29. *Pollard C., Sag I.,* 1994. Head-Driven Phrase Structure Grammar. University of Chicago Press, Chicago.
30. *TSNLP,* 1993–1995. Test Suites for Natural Language Processing. http://cl-www.dfki.uni-sb.de/tsnlp/.
31. *Uszkoreit H., Flickinger D., Oepen S.,* 2001. Proposal of Themes and Modalities for International Collaboration on Deep Linguistic Processing with HPSG. DFKI LT Lab and Saarland University, CSLI Stanford and YY Technologies.
32. *Uszkoreit H.,* 2002a. DELPHIN: Deep Linguistic Processing with HPSG — an International Collaboration.
33. *Uszkoreit H.,* 2002b. New Chances for Deep Linguistic Processing. In: The 19th International Conference on Computational Linguistics COLING'02, Taipei, Taiwan.
34. *XTAG,* 2002. Lexicalised Tree Adjoining Grammar. http://www.cis.upenn.edu/~xtag/.