Restructuring Świdziński's grammar of Polish

Maciej Ogrodniczuk

Chair of Formal Linguistics
Faculty of Modern Languages
Warsaw University

Browarna 8/10, 00-311 Warsaw, Poland
maciej.ogrodniczuk@gmail.com

Abstract

The article presents initial results of two changes applied to Świdziński's formal grammar of Polish: restructurization of the grammar to eliminate redundant cycles and its extension with nominal group definition (as defined by Szpakowicz and Świdziński). A short summary on the results of parsing the suite of test sentences with the original and modified variants of the grammar is also included.

1. Introduction

The grammar discussed here is Świdziński's formal grammar of Polish (known as Gramatyka Formalna Języka Polskiego; further referenced here as GFJP) expressed in a formalism inspired by metamorphosis grammars of Colmerauer (Colmerauer, 1978). GFJP has been formulated by Marek Świdziński in his habilitation thesis, later published (with minor modifications) as (Świdziński, 1992) and since then regarded as the largest[1] and most precise formal description of general grammar of Polish.

GFJP-based parsing of Polish has a long history with three major milestones resulting in working prototypes of parsers prepared under supervision of Janusz S. Bień: *AMOS-95*[2], *AS*[3] and — the most recent and the only one really usable — *Świgra*, implemented by Marcin Woliński within his doctoral dissertation (Woliński, 2004) and presented in these proceedings (see article *An efficient implementation of a large grammar of Polish*).

None of the previous parsing approaches did intend to modify GFJP beyond making corrections necessary to improve parsing and thus they all constituted close computer representation of the original grammar. The idea put forward in Woliński's dissertation and developed in this paper goes in opposite direction: use the most current version of Świgra parsing engine as a testing environment for modifications of the grammar that might result in better understanding of its underlying constructs.

First of such modifications is the attempt to improve the grammar by merging non-terminals which, as Woliński suggests, are distributionally equivalent.

The second modification extends GFJP definition of nominal phrase with formal description of co-ordinate nominal group adapted from (Szpakowicz and Świdziński, 1983) to make possible parsing of sentences with complex nominal expressions.

2. Elimination of redundant cycles

2.1. Recurrence in GFJP

GFJP capability of representing potentially infinite levels of substructures in parsed sentences is expressed in a peculiar way, which should be regarded as a consequence of Świdziński's conviction[4]:

> (...) to maintain recurrence it is necessary to allow for a lowest-level (simplest) syntax unit to be expressed as a highest-level syntax unit.

As a result[5], GFJP contains five cycles, each defined as a loop of clauses containing single non-terminal in clause head and clause body.

For example, the rule

$$\begin{aligned} &zsz(Wf, A, C, T, Rl, O, Neg, I, Z) \\ &\rightarrow s(s1), \\ &\quad zj(Wf, A, C, T, Rl, O, Neg, I, Z, Oz), \\ &\quad rozne(Oz, lub). \end{aligned}$$

is one of the elements of the cycle

$$zr \rightarrow zsz \rightarrow zj \rightarrow zp \rightarrow ze \rightarrow zr$$

Please note that different elements of the cycle might have different conditions attached to the rules (in the example above the condition states that `lub` is not allowed as the value of `Oz`).

Other cycles, all of them identified in Woliński's dissertation, are:

$$fno \rightarrow knodop \rightarrow knopm \rightarrow knoatr \rightarrow knoink \rightarrow knom \rightarrow fno$$

[1] It contains 463 rules as compared to 780 rules in Alvey grammar — the English grammar developed within the Alvey Natural Language Tools (Grover et al., 1993), probably the largest existing natural language grammar in Prolog.

[2] See (Bień, 1996a) and (Bień, 1996b).

[3] See (Bień et al., 1999) and (Bień, 2000).

[4] See (Świdziński, 1992), p. 59.

[5] However this idea seems unnatural, it can be justified by the spirit of GFJP which means for me the attempt to apply similar mechanisms to language constructs at different — sentence and phrase — levels.

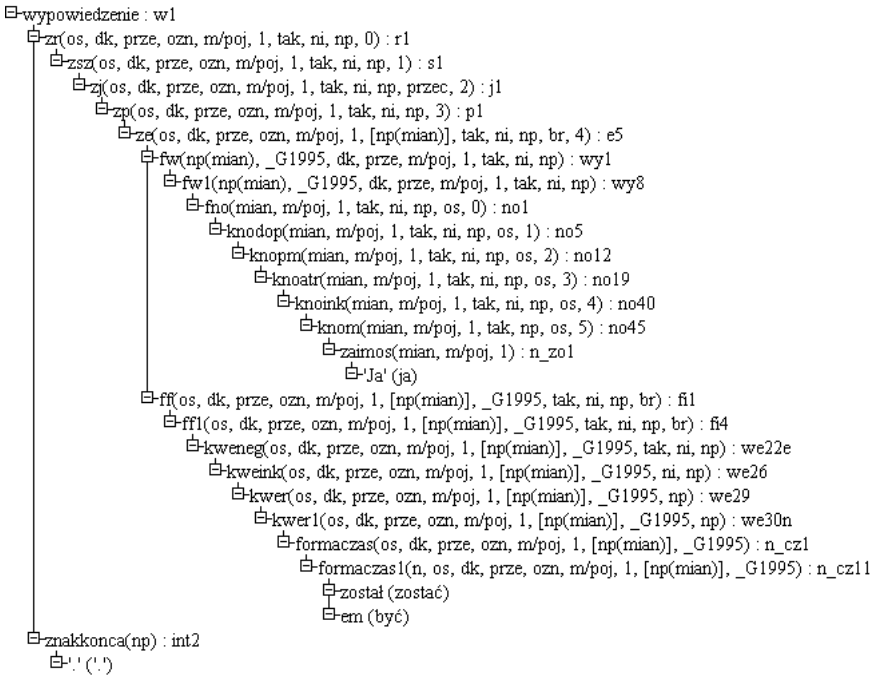

Figure 1. A result parsing tree generated with the original grammar

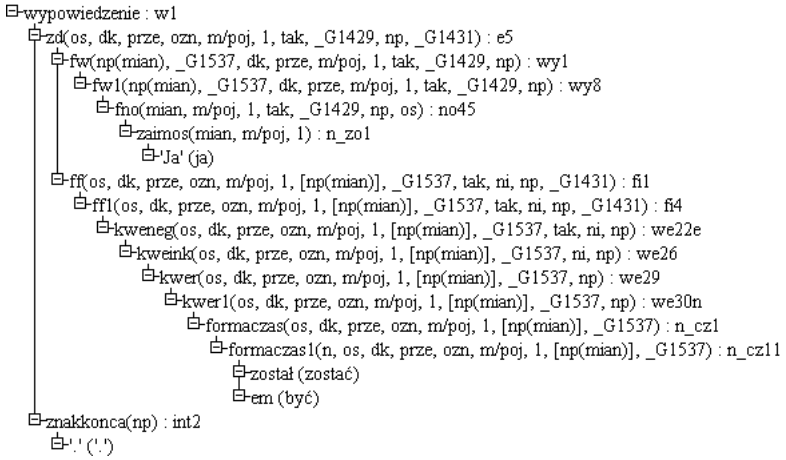

Figure 2. Flattened parsing tree: the grammar without cycles

$$fps \rightarrow kpspm \rightarrow kpsps \rightarrow kpsink \rightarrow kprzysl \rightarrow fps$$

$$fpt \rightarrow kptno \rightarrow kptpm \rightarrow kptps \rightarrow kptink \rightarrow kprzym \rightarrow fpt$$

$$fzd \rightarrow fzdsz \rightarrow fzdj \rightarrow fzdkor \rightarrow fzd$$

Such approach causes substantial problems for implementators of GFJP parser because presence of single non-terminal on both sides of the cycled clauses creates idle loops which result in infinite number of unproductive analyses (for a sentence analysed with a clause taking part in the cycle) containing repeated cycle path.

To minimize interference with the grammar but eliminate the idle cycles, Woliński blocks them by introducing additional parameter for each of the cycled clauses which controls cycle length to avoid idle paths while parsing — without having to restructure the grammar.

The opposite idea is to test Woliński's assumption that all non-terminals involved in each individual cycle are distributionally equivalent and thus they can be replaced with (or merged into) one common non-terminal for each cycle.

2.2. Merging non-terminals

At first, one common non-terminal has been introduced for each cycle to replace all cycled non-terminals. For instance, one single non-terminal `zd` has been introduced to replace all sentence-level non-terminals (`zr` $\rightarrow \ldots \rightarrow$ `ze`) etc. After applying this change, cycled clauses could have been (in most cases) turned off since clause head and clause body carried the same non-terminal with equivalent parameters.

One of the most apparent results of this procedure is

flattening the parsing trees — e.g. all types of sentences (from complex sentences with co-ordinate clauses to elementary sentence) are now expressed with a single non-terminal `zd` which, if applicable, enables the parser to reach the phrase level in one step.

During the process of rewriting the grammar it appeared that for some clauses additional conditions (added for blocking certain parsing paths, e.g. to disallow occurrences of *lub* conjunction at the beginning of a sentence) make the translation of grammar rules far more than a trivial task. Consequently, an attempt has been made to apply all the conditions to the new grammar by adopting such techniques as shifting them to a higher-level clause, redistributing them over equivalent other clauses or applying additional conditions. In a few cases, mutually exclusive conditions appeared in two separate clauses which could then be merged into single one not carrying any conditions.

Nevertheless, some of the conditions turned out to be more sophisticated than the others, so the process of their transfer has not yet been completed and the changed version of the grammar is not completely equivalent to the original GFJP. However, the initial test results presented below allow to say that both versions already reached similar power of expression.

2.3. Test suite-based comparison

The test suite used in the comparison (known as GFJP-A) has been extracted from (Świdziński, 1992) in the framework of the project described in (Bień, 2000).

It contains 660 Polish sentences marked up with manually encoded general segmentation marks and annotated with symbols of GFJP clauses necessary to parse each sentence. Such precise annotation, especially as it has been prepared by the author of the grammar (Świdziński, 1992), makes the test suite ideal for checking effects of modifications of particular GFJP clauses.

Different segmentation variants (following different usage of grammar clauses in manual parsing) available for some of the examples result in limiting the total number of unique sentences to 636. An attempt to parse all of them with both grammars, original and modified, has been made. The example of the result parsing trees for one of the shortest sentences from the suite is shown in figures 1 and 2.

The following tables sum up the parsing results for the original grammar:

	Correct sentences	**Incorrect sentences**	**Debatably correct**
Parsed successfully	53.0%	3.6%	0.0%
Parsing failed	14.5%	17.3%	0.5%
Parsing takes $>$ 10 s	8.6%	1.4%	1.1%

and for the modified grammar:

	Correct sentences	**Incorrect sentences**	**Debatably correct**
Parsed successfully	53.7%	4.4%	0.0%
Parsing failed	13.8%	16.5%	0.5%
Parsing takes $>$ 10 s	8.6%	1.4%	1.1%

Relatively high figures for successfully parsed incorrect sentences (23% of the total number of incorrect sentences from the test set) result from particular treatment of „free phrases" — a GFJP notion which usually corresponds to an adverbial or adverbial complement that can be freely removed from the construct it constitutes (see Woliński's article for more detailed discussion).

3. Adding the nominal group definition to GFJP

Definition of nominal phrase in GFJP, however sophisticated, is not intended to cover co-ordinate relationship between nominal constructs, such as in *kot i pies* (two simple nominal phrases joined with co-ordinate conjunction). Therefore, sentences containing even such simple phrases cannot be parsed with original version of GFJP.

However, a formal description of such constructs is contained in (Szpakowicz and Świdziński, 1983), where the hierarchy of co-ordinate nominal groups is expressed using similar formalism as in (Świdziński, 1992) which facilitates their direct inclusion in GFJP:

`rgn`	co-ordinate nominal group
`sgn`	sequential nominal group
`jgn`	homogeneous nominal group
`pgn`	single nominal group

To test this definition, another GFJP variant has been prepared along with the general test suite for nominal groups. Two differences have been made to the description from Szpakowicz and Świdziński's article: the ellipsis parameter (present in the description „for future extension of the definition") has been omitted and the values of class parameter (introduced to group together nominal constructs representing the same grammatical class) have been limited to negative pronoun[6]. The definition has been embedded in GFJP by replacing all occurrences of nominal phrase (`fno`) in GFJP clause bodies with newly defined general nominal group (`gno`), which in turn included definition of the highest-level unit — co-ordinate nominal group (in only clause).

Single nominal group, left in the article for future specification, has been defined as GFJP nominal phrase (although it should be noted that its description is significantly simpler than authors' intentions for single nominal group). To make both descriptions compatible, existing

[6]The original description contains five other classes; four of them had been left for future development by article authors while the remaining one — genitive numeral, referenced explicitly, had to be removed from the grammar since current version of GFJP lacks formal description of Polish numerals.

nominal phrase definition had to be extended with additional class and selective negation parameters.

GFJP extended in this way can be now used to parse real-life nominal expressions such as quite common complex subjects.

3.1. The position parameter

One of the most interesting components of the nominal group description is by far the position parameter. It is defined as the place of nominative nominal group with respect to verbal group in the sentence and can be expressed as `post`, `pre` or `undefined`.

According to relationships between position, gender and number the sentence containing such nominal group is regarded correct by the authors:

1. when gender and number of the verbal group agree with the parameters of the nominal group as a whole[7] — in such case the nominal group can occur before or after the verbal group:

   masculine and plural, `undefined` position
   (Nie przyszli) ani dziecko, ani ojciec.
   Ani dziecko, ani ojciec (nie przyszli).

2. when gender and number of the verbal group agree with the parameters of the first component of the nominal group — in such case the nominal group can occur only after the verbal group:

   neuter and singular, `post` position
   (Nie przyszło) ani dziecko, ani ojciec.

3. when gender and number of the verbal group agree with the parameters of the last component of the nominal group — in such case the nominal group can occur only before the verbal group:

   masculine and singular, `pre` position
   Ani dziecko, ani ojciec (nie przyszedł).

Nevertheless, certain sentences formed according to these rules seem contrary to linguistic intuition (*Powiedziała mu ona i on.*)[8]. Basing on this ambiguity, unorthodox decision has been made to make use of the parameter outside the grammar. The underlying idea is to mark the result parsing trees containing wrong position as incorrect without changing the grammar significantly (since the article deals only with the calculation of position parameter, not with how it further intervenes with the verbal group).

Elimination of the parsing trees with incompatible position, gender and number has been achieved by changing the way result trees are post-processed in the parser. The tree fragments containing calculated position of the verbal and nominal groups inconsistent with the position parameter resulting from nominal group definition are removed from the result set and are therefore blocked to be used as valid results.

[7] This implies plural number; gender category is calculated using separate rules, the main of which is domination of masculine gender, see (Kallas, 1976; Świdziński, 1978) for details.

[8] An ordinary „language user" could say that the more distant the verb in the sentence is from the subject, the more likely the sentence seems correct, but it definitely require more studies.

3.2. Testing the definition

No test suite similar to the one used in previous task was available for the grammar with nominal groups — however, part 4 of the article (Szpakowicz and Świdziński, 1983) could be used as a source of test samples.

As a consequence, 216 test sentences (135 correct, 79 incorrect and 2 debatably correct) were extracted from the article and saved in the format compatible with GFJP-A and B suites (each sample containing number of the section it derives from, symbol of the clause it can be parsed with, serial number and correct/incorrect/questionable mark of the sample):

```
[R2; GNO 4.2.3; 1]
Zarówno chłopiec, jak i dziewczyna
(przyszli).
```

The test suite has been limited to sentences with groups not containing class parameters other than the currently available one (negative pronoun), which reduced the number of processed sentences to 177.

Here are the parsing results for the constrained test suite:

	Correct sentences	**Incorrect sentences**	**Debatably correct**
Parsed successfully	31.1%	5.7%	1.1%
Parsing failed	30.5%	31.6%	0.0%

The parsing results still seem to leave plenty of room for improvement. Closer look at the results shows that the percentage of failures for correct sentences containing negative pronoun is significantly higher that for other sentences. Further studies will be dedicated to explain this issue.

4. Perspectives

The two relatively simple changes should only be considered as the training ground for further changes in GFJP. The next task on to-do list is integration of the description of numerals into the grammar. Thanks to the earlier works such as (Gruszczyński and Saloni, 1978) and (Derwojedowa et al., 2003), this is not expected to be difficult.

As for longer perspective, the more mature version of the modification will be used in automated processing of a small treebank of Polish expressions created and marked-up with the results of manual GFJP-based parsing in Świdziński's research grant (Świdziński, 1996). The comparison of computer-generated structures with the hand-made annotations will definitely allow to draw interesting conclusions.

5. References

Bień, Janusz S., 1996a. *Computer validation of a description of Polish syntax*. [In Polish]. TR 96-06 (227), Institute of Informatics, Warsaw University.

Bień, Janusz S., 1996b. *Computer validation of Świdziński's formal grammar*. [In Polish]. PTJ Bulletin, vol. LII, pp. 147–164.

Bień, Janusz S., 2000. *Test suite for verification and assessment of parsers of Polish. Final (slightly modified) report of KBN 8 T11C C 002 13 grant*. [In Polish]. Electronic version: `ftp://ftp.mimuw.edu.pl/pub/users/polszczyzna/tajp/`.

Bień, Janusz S., K. Szafran, and M. Woliński, 1999. *Experimental Parsers of Polish*. [in:] *3. Europäische Konferenz "Formale Beschreibung slavischer Sprachen, Leipzig 1999"*, vol. 75 of *Linguistische ArbeitsBerichte,* series, pp. 185–190, Institut für Linguistik, Universität Leipzig.

Colmerauer, A., 1978. *Metamorphosis grammar*. [in:] L. Bolc (ed.), *Natural Language Communication with Computers*, Lecture Notes in Computer Science 63, pp. 133–189, Springer-Verlag.

Derwojedowa, M., M. Rudolf, and M. Świdziński, 2003. *Two formal approaches to Polish numeral phrases implemented*. [in:] *Studia z gramatyki i leksykologii języka polskiego*, pp. 93–108, Uniwersytet Mikołaja Kopernika, Toruń.

Grover, C., J. Carroll, and E. Briscoe, 1993. *The Alvey Natural Language Tools grammar*. (4th release). Computer Laboratory, Cambridge University, UK, Technical Report 284. Electronic version: `ftp://ftp.cl.cam.ac.uk/nltools/reports/grammar.ps`.

Gruszczyński, W. and Z. Saloni, 1978. *Syntax of numeral phrases in modern Polish*. [in:] Studia gramatyczne, Vol. II. Wrocław, pp. 17–42.

Kallas, K., 1976. *On sentences such as* Pachniał wiatr i morze*,* Andrzej i Amelia milczeli. [In Polish]. Studia z filologii polskiej i słowiańskiej, vol. 4, Warsaw.

Szpakowicz, S. and M. Świdziński, 1983. *Formal definition of co-ordinate nominal group in present written Polish*. [In Polish]. Studia gramatyczne IX, ISBN 83-04-03303-8.

Świdziński, M., 1978. *Sample transformations in Polish*. [In Polish]. "Polonica" IV.

Świdziński, M., 1992. *Formal grammar of Polish*. [In Polish]. Warsaw University Dissertations, Warsaw.

Świdziński, M., 1996. *Syntax properties of Polish expressions*. [In Polish]. Institute of Polish Philology, Warsaw University, Warsaw.

Woliński, M., 2004. *Computer-aided verification of Świdziński's grammar*. [In Polish]. Ph.D. thesis. Institute of Computer Science, Polish Academy of Sciences, Warsaw.