Processing Polish
with Metamorphosis Grammar*

Janusz S. Bień
Institute of Informatics, Warsaw University
jsbien@mimuw.edu.pl

September 19th, 1996

The Prolog programming language has been created by Alain Colmerauer especially for text processing. He designed also a parsing technique, which was already in use in the spring of 1974, when David H.D. Warren and myself has visited Colmerauer in Marseille. The technique has been used for Polish as soon as Prolog was installed in Warsaw, and the first program using this technique ([4]) was ready in 1975.

The technique required some “syntactic sugar” to be used comfortably. An extension of Prolog syntax was incorporated into the new version of Marseille Prolog [9] (later called Prolog I). At the same time Colmerauer described the theoretical foundation of the technique, which he started to call metamorphosis grammar (in the source code of the interpreter another name was used — regles super-Q), in the paper [2]. The paper was immediately submitted for publication, but through the editor and the publisher fault it appeared as late as in 1978. The delay substantially diminished the impact of the paper; on the other hand, the paper was extremely difficult to read and rather confusing for a casual reader.

In the meantime David Warren proposed another syntactic form for a limited variant of the technique, calling it Definite Clause Grammar, and incorporated it into a new implementation of Prolog for the DECSystem-10 computer [8]; as this was a very popular computer, this implementation soon became widely used. Moreover, together with Fernando Pereira he wrote an excellent paper [7] advocating the superiority of Definite Clause Grammar over Augmented Transition Network. In consequence, the original Metamorphosis Grammar has been overshadowed by Definite Clause Grammar and remained little known; as far as I know, it has been adequately covered only in [13] and [14].

Metamorphosis Grammar was used in a simple question-answering system described in [15] and [16], and in a quite sophisticated Polish railway expert system ([5] and [6]). However, the most important work consisted in using Metamorphosis Grammar as a formalism for linguistic description of Polish syntax developed by Stanisław Szpakowicz ([10], [11], [12]). This formalism has been adapted by other researches working on Polish syntax, in particular by Marek Świdziński.

*This is the handout of a talk presented at International Center “East-West” Workshop on Computational Linguistics at the University of Tübingen, 16–27 September 1996.
Szpakowicz has implemented a subset of his grammar as a parser, which later was developed by myself into so called 2nd version of Szpakowicz’s parser. In improving its efficiency I have used a look-ahead technique suggested by Małgorzata Nalbach [1]. Recently a new parser, based on Świdziński’s grammar, has been created as a part of the AMOS system. Below I summarize the most important extensions to the original Metamorphosis Grammar, used in the AMOS system.

- We use the syntax of Definite Clause Grammar to allow syntax checking on any implementation of Prolog.
- Because the grammar does not operate on strings, but on the output of the morphological analyser, the terminal symbols are replaced by pseudo-nonterminal symbols _uja_ (called _universal elementary unit_) and _inter_ (for interpunction marks).
- To account for the terminal symbols on the left hand side of the rule, allowed with some limitations in Metamorphosis Grammar but illegal in Definite Clause Grammar, the pseudo-nonterminal symbol _next_ was introduced, which accepts as its argument _uja_ or _inter_ symbols.
- The _next_ symbol is used also to look ahead for the next element in various stages of the parsing process. We consider this a purely pragmatic device, changing the parsing strategy but not affecting the grammar semantics.
- We have introduced also the _previous_ pseudo-nonterminal, analogical to the _next_ one. It appeared useful for the _ad hoc_ solution of some problems, but it seems that its use can be completely eliminated.
- The parse tree is constructed automatically, and various features are provided to facilitate debugging.

More information about the AMOS system can be found on WWW at the address

References


