2.3. Uproszczona gramatyka Szpakowicza

Przygotowana w latach 1976–1977 praca doktorska Szpakowicza [Szpakowicz 1978] — mająca w gruncie rzeczy charakter teoretyczny — została zilustrowana programem napisanym w języku Prolog i działającym na komputerze CDC Cyber 72 Środowiskowego Centrum Obliczeniowego obsługującego między innymi Uniwersytet Warszawski. Program ten był w istocie uproszczoną wersją gramatyki przedstawionej w pracy, której poświęcony jest punkt następny.

Na potrzeby pracy doktorskiej za pomocą programu przetworzono zaledwie 38 przykładów, co jednak w ówczesnych warunkach było dużym osiągnięciem — jak można przeczytać w artykule [Kluźniak i Szpakowicz 1984, s. 65–66], samo wczytanie reguł gramatycznych zajęło kilka miesięcy, bo z powodu dużych, jak na ówczesne czasy wymagań pamięciowych programów prologowych, wykonujące się 1 minutę zadanie czekało na swoją kolej około 10 godzin, a dłuższe zadania były wykonywane tylko w weekendy. Program ten wykorzystywał oryginalną implementację języka Prolog, czyli Prolog marsylijski, i gramatyki metamorficzne w ich oryginalnej postaci.

Program ten jest nazywany pierwszą wersją parsera Szpakowicza, ale jest to określenie umowne z kilku powodów.

Po pierwsze, program nie był tylko parserem, czyli analizatorem, ale pozwalał również generować zdania lub ich fragmenty — takie podwójne funkcje programów są łatwe do osiągnięcia w Prologu i innych językach programowania w logice.

Po drugie, dysertację ilustrowały w gruncie rzeczy dwa programy. W celu przyspieszenia przetwarzania przykładów tylko jeden z nich został zanalizowany za pomocą wersji prawidłowo obsługującej rekurencyjne reguły gramatyki — do sprawy wrócimy w punkcie 3.3.2 na s. 51. Ponieważ wiadomo było, że takie reguły nie są potrzebne do analizy pozostałych przykładów, były one przetworzone za pomocą uproszczonej wersji programu.

Słownik był ograniczony do kilkudziesięciu słów (tj. konkretnych form fleksyjnych), z powodów technicznych analizowane dane musiały być pisane dużymi literami, a polskie litery były reprezentowane za pomocą nieco zmodyfikowanej konwencji przedstawionej w artykule [Bień 1969b], na przykład `KT4ORA KSI4A4RKA JEST MA4LA?` [Szpakowicz 1999a, s. 127]. Ze względu na swój demonstracyjny charakter program zatrzymywał się po znalezieniu pierwszej interpretacji wejściowego napisu. Z tego też powodu stosował najprostszą — bo standardowo dostępną w Prologu — zstępującą (*top-down*) strategię analizy.

Drzewa rozbioru gramatycznego były prezentowane w sposób zilustrowany tutaj zaczerpniętym z pracy [Szpakowicz 1999a, s. 127] przykładem A.1 na s. 94.

Korzeń drzewa znajduje się w lewym górnym rogu. Węzły podrzędne są wypisywane pionowo pod węzłem nadrzędnym, w kolumnie przesuniętej o dwie spacje w stosunku do węzła nadrzędnego; kolejność z góry na dół odpowiada kolejności

od lewej do prawej przy tradycyjnym zapisie. Jeśli węzeł nadrzędny reprezentuje symbol nieterminalny z parametrami, to ich wartości są wypisane w osobnym wierszu.

Program Szpakowicza zachował się w używalnej postaci do dnia dzisiejszego, a to dzięki temu, że był on wykorzystywany jako program demonstracyjny w implementacjach języka Prolog opracowywanych w Instytucie Informatyki UW. Prolog został między innymi zaimplementowany na komputerach ODRA 1300[10].

Implementacja ta, opisana w książce [Kluźniak i Szpakowicz 1983], była rozpowszechniana na taśmach magnetycznych. W latach 1980–81 zakupiło ją kilka uczelni polskich, otrzymując wraz z nią między innymi praktycznie niezmieniony program Szpakowicza. Kiedy pozwoliły na to możliwości techniczne, pliki z takiej taśmy dystrybucyjnej zostały przeniesione na dyskietki. W 1998 r. dwóch uczestników seminarium *Narzędzia i metody przetwarzania tekstów* (prowadzonego przeze mnie wspólnie z Krzysztofem Szafranem) — Piotr Stępień i Jacek Jóźwiak — zaadaptowali ODRA-Prolog do systemu Linux, dzięki czemu zainteresowane osoby mogą eksperymentować z programem na współczesnych komputerach. Adaptacja ta jest dostępna pod adresem (`ftp://ftp.mimuw.edu.pl/pub/users/jsbien/ODRA-Prolog/`).

Pierwsza wersja parsera Szpakowicza była bardzo niewygodna w użytkowaniu. Próbą nadania programowi bardziej użytkowego charakteru była opracowana przeze mnie — z niewielkim udziałem jego autora — w latach 1984–85 tzw. druga wersja parsera Szpakowicza, wspomniana m.in. w referacie [Bień i Nalbach 1984].

Wprowadzone zmiany były dwojakiego rodzaju. Pierwsze z nich zmierzały do zwiększenia szybkości działania programu i dotyczyły jego części gramatycznej. Polegały one przede wszystkim na przystosowaniu reguł gramatyki do stosowanej strategii analizy, a mianowicie na faktoryzacji reguł, czyli komasacji tych reguł, których prawe strony zaczynały się w ten sam sposób. Druga zastosowana technika to rozszerzenie niektórych reguł o podgląd kolejnego symbolu terminalnego, co pozwala niekiedy z góry stwierdzić, że dana reguła nie ma szansy powodzenia[11]. Do sprawy tej wrócimy w punkcie 3.3.2 na s. 53.

Zmiany drugiego rodzaju dotyczyły obsługi słownika programu. Nie tylko ułatwiono dodawanie nowych pozycji, ale także umożliwiono wyszukiwanie w słowniku pozycji o określonych własnościach gramatycznych. Umożliwiało to demonstrowanie możliwości programu na dowolnym zdaniu języka polskiego według następującego scenariusza: dla każdego nierozpoznanego słowa wyszukiwano

[10] Prace finansował program resortowy RI 14, którym w latach 1978–1986 kierował prof. Mieczysław Bazewicz. Implementacja języka Prolog została włączona do programu zapewne z inicjatywy Stanisława Waligórskiego.

[11] Podobne podejście, choć może bardziej eleganckie, 10 lat później zaproponował niezależnie Zygmunt Vetulani w notatce [Vetulani 1994] na potrzeby systemu POLINT [Vetulani 2004].

w słowniku słowo o identycznych własnościach gramatycznych, i zastępowano to nierozpoznane słowo (przeważnie okazywało się również niezbędne uproszczenie struktury zdania).

Prace nad tym programem były prowadzone na komputerze IBM 360 z wykorzystaniem implementacji Prologu przeznaczonych dla komputerów RIAD[12] (patrz [Szafran i Szpakowicz 1984]). Również i w tym wypadku odpowiednie pliki zostały skopiowane (z taśmy magnetycznej lub kart perforowanych) na dyskietki i zachowały się do dzisiaj w używalnej postaci.

Dzięki powstaniu tej wersji stało się możliwe przeprowadzenie przez Mirosława Bańkę testów adekwatności lingwistycznej parsera Szpakowicza. Było to przedmiotem zrealizowanej pod moim kierunkiem pracy magisterskiej [Bańko 1985]. Jej trwałym wynikiem jest sformułowanie metod pomiaru adekwatności gramatyk, przedstawionych później w artykule [Bańko 1990], do których wrócimy w rozdziale 3 na s. 37.

Pomimo wprowadzenia narzędzi wspomagających tworzenie słownika, zadanie to okazało się bardzo uciążliwe. Dodatkową konsekwencją eksperymentu Bańki była więc decyzja o zawieszeniu prac nad analizą składniową do czasu stworzenia analizatora morfologicznego, co nastąpiło dopiero prawie 10 lat później — patrz [Szafran 1993].

Druga wersja parsera Szpakowicza została użyta ponownie — już na komputerze PC — na potrzeby wspomnianej wcześniej pracy magisterskiej Adama Wachowskiego [Wachowski 2000]. Aktualnie ma ona jednak wyłącznie charakter historycznej ciekawostki.

2.4. Gramatyka Szpakowicza

Jak było wspomniane, program komputerowy stanowiący element pracy doktorskiej Szpakowicza [Szpakowicz 1978] pełnił w niej tylko rolę ilustracji, jej zasadniczą treść stanowiła bowiem gramatyka formalna pewnego podzbioru polszczyzny. Praca ta w sposób nie zawsze jawny proponowała też pewną metodologię formalnego opisu języka.

Bezpośrednim impulsem do podjęcia tych prac była wizyta Alaina Colmerauera — twórcy języka programowania Polog — w Instytucie Informatyki UW w czerwcu 1977 roku, który w kuluarowych rozmowach przekazał mi szereg cennych sugestii na temat wykorzystania gramatyk metamorficznych do opisu języków fleksyjnych — do sprawy tej wrócimy w punkcie 5.3 na s. 77. Sugestie te trafiły na podatny grunt, ponieważ właśnie ukazała się książka Saloniego [1976a] stanowiąca dobry punkt wyjścia do formalnego opisu składni. Ponieważ sam byłem wówczas zaabsorbowany problematyką świeżo powstałej *Cognitive Science*,

[12] Były to produkowane w krajach RWPG kopie komputerów IBM 360.

podsunąłem koledze Stanisławowi Szpakowiczowi pomysł stworzenia niewielkiej gramatyki formalnej dla języka polskiego, który on przyjął (por. [Szpakowicz 1978, s. viii]) i zrealizował w nadzwyczaj udany sposób.

Zasadnicza część pracy doktorskiej była przygotowana tradycyjnie na maszynie do pisania, ale zawierała ona również wydruki pełnego tekstu programów, danych testowych i ich wyników. Z czasem całość pracy została wprowadzona do komputera i uzupełniona o dodatkowe informacje przez uczestników mojego seminarium *Lingwistyka informatyczna*, co umożliwiło mi sporządzenie wydania elektronicznego dostosowanego do aktualnych standardów wydawniczych[13]. Wprowadziłem również pewne dodatkowe konwencje omówione szczegółowo w posłowiu redaktora [Bień 1999]. W końcowej fazie przygotowanie wydania było wspierane przez grant KBN nr 8 T11C 002 13 *Zestaw testów do weryfikacji i oceny analizatorów składniowych* realizowany w latach 1997–1999 w Instytucie Informatyki UW pod moim kierunkiem. Od 1999 r. roku praca jest dostępna w Internecie razem z innymi wynikami tego grantu, do których wrócimy w punkcie 2.5.3 na s. 33.

W środowisku informatyków, do którego należał Szpakowicz, było oczywiste, że opisem języka powinni w pierwszej kolejności zajmować się lingwiści. Dlatego tekst pracy został przeredagowany tak, aby jego odbiorcami mogli być poloniści, i opublikowany w formie książkowej [Szpakowicz 1983]; po wyczerpaniu nakładu w r. 1986 ukazało się niezmienione wydanie drugie [Szpakowicz 1986] (uzupełnione tylko o dodatkową przedmowę).

Podobnie jak w przypadku pracy doktorskiej, z pomocą studentów przygotowałem elektroniczną edycję tej książki — patrz [Szpakowicz 1999b].

Wynikami Szpakowicza i stosowanymi przez niego narzędziami zainteresował się polonista Marek Świdziński, co zaowocowało ich bliską współpracą. Niektóre ich wspólne publikacje mają obecnie tylko charakter ciekawostek historycznych (np. [Szpakowicz i Świdziński 1979, 1982]), ale jedna z nich — obszerny artykuł [Szpakowicz i Świdziński 1990] (wcześniej dostępny jako maszynopis powielony [Szpakowicz i Świdziński 1981]) — ma znaczenie również współcześnie, jak zobaczymy w punkcie 2.6 na s. 34. Wypracowana przez Szpakowicza metodologia wywarła też duży wpływ na stworzoną przez Świdzińskiego tzw. gramatykę GFJP, o której będzie mowa m.in. w punkcie następnym.

Prace Szpakowicza wywarły też wpływ na środowisko informatyczne — mniejsze lub większe fragmenty gramatyki były adaptowane do różnych celów w kilku ośrodkach zajmujących się przetwarzaniem tekstów. Znane nam adaptacje powstały w Warszawie ([Kubacka i Makowski 1991], [Dobryjanowicz 1992]), we Wrocławiu [Piasecki 1993] oraz w Poznaniu [Jassem 1997].

[13] Do tego celu zostały wykorzystane między innymi narzędzia opisane w przygotowanych pod moim kierunkiem pracach magisterskich [Pietrzak 1999], [Woliński 1998a] i publikacjach pochodnych [Woliński 1996, 1998b].

Koncepcja stworzenia analizatora będącego implementacją pełnej gramatyki Szpakowicza pojawiła się dopiero 20 lat po jej powstaniu (i 10 lat po omawianej dalej gramatyce Świdzińskiego). Przyczyną był fakt, że pierwsza próba implementacji gramatyki Świdzińskiego – analizator AMOS (patrz punkt 2.5.2 na s. 31) – znalazła się w impasie z powodu zaporowo[14] długiego czasu analizy nawet bardzo prostych i krótkich zdań. Powstało w związku z tym pytanie, czy istotnie prostsza gramatyka Szpakowicza pozwoli uzyskać akceptowalną efektywność analizatora bez zmieniania jego zasadniczej koncepcji.

Wynikiem była pierwsza wersja analizatora GraSz. Większość reguł gramatyki została wpisana w 1996 i 1997 roku przez uczestników seminariów *Lingwistyka informatyczna* oraz *Narzędzia i metody przetwarzania tekstów*, prowadzonych przeze mnie wspólnie z Krzysztofem Szafranem. Choć uczestnicy wprowadzali do komputera również oryginalne reguły wraz z tekstem pracy i książki Szpakowicza na potrzeby ich elektronicznej publikacji, to ich właściwe zadanie było bardziej skomplikowane. Po pierwsze, dokonywali oni transkrypcji „publikacyjnej" formy reguł gramatycznych na formę zgodną ze składnią używanej wersji Prologu. Po drugie, dokonywali oni od razu faktoryzacji reguł w sposób zachowujący informację, która z oryginalnych reguł została użyta do analizy – służyły do tego dwa predykaty `s` (jeden jednoargumentowy, drugi dwuargumentowy). Dla przykładu reguły (cytowane niżej w wersji publikacyjnej):

ZDANIEŻE
= **SPÓJŻE ZDANIEZŁOŻ** (zze1)
= **SPÓJŻE ZDANIEZŁOŻ SPÓJSZER ZDANIEŻE .** (zze2)

otrzymywały postać jednej reguły:

```
zdanieze --> s(zze(NrR)/1),
      spojze,
      zdaniezloz,
      (s(zze(NrR),zze(2)),
       spojszer,
       zdanieze
      ;
      s(zze(NrR),zze(1))).
```

Jak widać, wspólne początki obu reguł znajdują się przed alternatywą ich pozostałych części, a symbol reguły jest ustalony dopiero wtedy, kiedy zostanie wybrany konkretny człon alternatywy. Symbol ten miał odpowiednio postać `zze(1)/1` lub `zze(2)/1` – liczba po ukośniku to numer reguły po faktoryzacji.

[14] Tak informatycy tłumaczą na polski pożyteczne angielskie słowo *prohibitively*.

W 1998 roku treść reguł objął swoją opieką Adam Wachowski, rozpoczynając w ten sposób przygotowywanie wspomnianej wcześniej pracy magisterskiej [Wachowski 2000]. Połączył on w całość i poprawił wpisane reguły, był też autorem większości naniesionych od tamtego czasu poprawek. Było ich dość dużo; głównie z tego powodu GraSz tak naprawdę nigdy nie zaistniał w formie zamkniętej wersji — funkcjonowały jedynie tzw. wersje robocze analizatora. Wyniki działania analizatora były jednak równie niesatysfakcjonujące jak w przypadku analizatora AMOS. Na komputerze PC z procesorem Pentium II analiza nawet bardzo prostych zdań mogła wymagać nawet kilku dni, analizowanie bardziej złożonych przykładów było praktycznie niemożliwe.

Stało się zatem jasne, że problemem nie jest gramatyka, lecz strategia analizy. Zadania porównania ważniejszych strategii znanych z literatury przedmiotu podjął się w 1999 r. Bartłomiej Kruszyński w swojej pracy magisterskiej przygotowywanej pod kierunkiem Krzysztofa Szafrana [Kruszyński 2001]; strategie te miały być porównywane właśnie na materiale gramatyki Szpakowicza. W celu przystosowania jej do testów strategii wstępującej (*bottom-up*) było niezbędne zlikwidowanie faktoryzacji i przywrócenie regułom postaci zbliżonej do oryginalnych — zostało to wykonane za pomocą odpowiedniego programu. Ta postać gramatyki, wraz z programem wykorzystującym ją do analizy zgodnie z metodą *chart parsing*[15], stanowiła łącznie drugą wersję analizatora GraSz (por. [Gazdar i Mellish 1989]).

Już na wstępnych etapach przygotowania tej pracy magisterskiej wyższość nowej metody analizy okazała się niewątpliwa — duża szybkość działania pozwalała na znajdowanie wszystkich możliwych rozbiorów syntaktycznych zdania (jak już pisaliśmy, wcześniejsze analizatory zatrzymywały analizę po znalezieniu pierwszej interpretacji). Druga wersja analizatora GraSz nadawała się już do praktycznego użytku i w związku z tym — z pewnymi mało istotnymi zmianami — została uwzględniona w eksperymentach Wachowskiego. Jednak ze względu na to, że pewne rozwiązania techniczne miały w nim charakter prowizoryczny, wersja ta nie była dalej rozwijana.

Przy okazji pracy nad niniejszą książką została stworzona przeze mnie trzecia wersja analizatora GraSz. Korzysta ona z tego samego analizatora morfologicznego i tej samej strategii analizy, co omawiane dalej analizatory Świgra (patrz punkt 2.5.4 na s. 33) i GraŚwiSzpaO (patrz punkt 2.6 na s. 34). Dzięki temu łatwiej jest je porównywać, a porównania te są bardziej rzetelne.

[15] Za pierwszą prezentację tej metody uważa się wewnętrzny raport: Martin Kay (1980). Algorithm schemata and data structures in syntactic processing. Technical Report CSL-80-12, Xerox PARC, Palo Alto, CA. Metoda ta była potem często opisywana przez różnych autorów w wielu publikacjach. Woliński [2004, s. 44] *chart parsing* tłumaczy jako *analiza tablicowa*, dla recenzenta termin ten ma jednak szersze znaczenie.