Wizualizacja wyników analizy składniowej zdań języka naturalnego

25 września 2010

Wstęp

Możliwość komputerowej analizy składniowej tekstu w języku polskim istnieje od powstania pierwszej gramatyki i pierwszego analizatora składniowego, napisanych w języku Prolog w latach 70-tych. Celem takiej analizy jest po pierwsze sprawdzenie, czy dany napis wejściowy jest poprawną reprezentacją zadanej jednostki składniowej tego języka, w szczególnym przypadku czy jest on całym wypowiedzeniem. W praktyce jest to osiągane przez sprawdzenie, czy możliwe jest zbudowanie struktury powierzchniowej dla danego tekstu, zgodnej z posiadaną gramatyką. Określenie struktury tekstu jest więc drugim celem analizatora. Oczywiście do wielu celów zadanie analizy tekstu nie kończy się na składni powierzchniowej, ale jest ona krokiem potrzebnym do dalszego przetwarzania. Otrzymywana struktura jest również obrazem działania analizatora i sposobem na sprawdzenie dlaczego decyzja w kwestii poprawności napisu jest konkretnie taka.

Również dla wielu innych języków istnieją mniej lub bardziej kompletne gramatyki i obsługujące je programy. Generalnie analizatory są pisane tak, aby mogły być adaptowane do gramatyk różnych języków lub obsługiwały je bez potrzeby adaptacji, jednak wysiłek związany ze stworzeniem gramatyki dla języka naturalnego i jej poziom skomplikowania powodują, że wymyślenie formalizmu opisu gramatyki i napisanie programu analizatora są niewielkimi zadaniami i bywają widziane jako detal. Wspólną cechą wielu formalizmów jest jednak opisywanie budowy jednostek jako sekwencje lub zbiory jednostek niższego poziomu, w efekcie oczekiwany końcowy opis składniowy tekstu ma formę drzewa.

Liśćmi drzewa i jednostkami najniższego poziomu w gramatyce są fragmenty wejściowego napisu z przypisanymi im własnościami morfologicznymi. Dla uproszczenia zadania analizy, jest ona dzielona na podzadanie analizy morfologicznej, w tym segmentacji tekstu na jednostki najniższego poziomu (leksemy), i właściwą analizę składniową za pomocą gramatyki. Zadania będą zwykle realizowane przez oddzielne programy.

Ani podział tekstu na segmenty, ani przypisywanie parametrów fleksyjnych, ani przypisanie struktury składniowej całemu tekstowi w ogólnym przypadku

nie gwarantują jednoznaczności wyniku. W szczególności języki naturalne są bogate w wieloznaczności na różnych poziomach. Ponieważ napis uważa się za należący do języka opisywanego przez gramatykę jeśli istnieje choć jedna interpretacja zgodna z gramatyką, to wspomniany wcześniej pierwszy analizator dla języka polskiego Szpakowicza przerywał analizę po znalezieniu pierwszego zgodnego drzewa [1]. W generalnym przypadku jednak można traktować analizator jako program na wejściu przyjmujący tekst do analizy, a na wyjściu podający opis całego zbioru struktur drzewiastych, czyli lasu, spełniających warunki gramatyki i opisujących zadany tekst. Poszczególne drzewa mają pewne cechy wspólne, bo opisują ten sam tekst.

Całościowa struktura lasu może być bardzo obszerna w zależności od wejścia, od języka i szczegółów implementacyjnych. Czytanie takiej struktury jest więc trudne dla człowieka, zarówno z powodu możliwej wysokości drzew składowych jak ilości drzew. Bez względu na cel przeglądania wyniku analizy, efektywność w wielkim stopniu będzie zależeć od formy przedstawienia tego wyniku, dlatego ważne jest optymalne dobranie tej formy do celu. Także prawdopodobnie przydatne byłoby istnienie jakiejś kanonicznej formy, którą czytelnik umie zrozumieć bez pełnego zaznajamiania się ze szczegółami opisywanego wyniku, tak jak to ma miejsce dla wielu innych zadań wykonywanych przez programy komputerowe i ich wyników. Do niektórych zadań przydatna może się okazać możliwość ręcznej edycji takiej struktury.

Wydaje się, że nie ma dziś w użyciu formalizmów ani narzędzi implementujących je, które spełniałyby takie warunki. Programy lingwistyczne wykorzystywane przy analizie składniowej, takie jak Linguistic User Interface z projektu DELPH-IN, edytor drzew TrED (z "Praskiego Treebanku") wykorzystywany przy anotacji korpusów i inne, operują na pojedynczych drzewach lub sekwencjach drzew wyświetlanych oddzielnie. Struktura lasu jest grafem, który generalnie nie jest szczególnym przypadkiem innych często wykorzystywanych struktur (tak, jak grafika 2D to przypadek szczególny grafiki 3D, w związku z czym można nadużywać pewnych narzędzi oryginalnie stworzonych dla grafiki 3D). Narzędzia do operacji na ogólnych grafach (yEd, graphviz, itp.) wyjątkowo źle oddają lingwistyczne własności drzew i lasów analizy składniowej i prezentowane przez nie wizualizacje są w efekcie nieczytelne. O braku takich narzędzi świadczy również brak przykładów wizualizacji poświęconych konkretnie wynikom analizy składniowej na stronach internetowych visualcomplexity.com i w Układzie Okresowym Metod Wizualizacji, choć mogą one być inspiracją.

Ta praca jest próbą analizy problemu, dostarczenia pewnych działających narzędzi do wizualizacji lasów i zasygnalizowania innych możliwych sposobów wizualizacji w nowych projektach, których cele mogą być bardzo różne od znanych teraz projektów. Wśród tych narzędzi znajduje się implementacja Arkuszy Syntaktycznych, a podane przykłady i programy przykładowe operują na wynikach działania analizatora składniowego dla (podzbioru) języka polskiego, Świgra.

1 Forma wyniku analizy składniowej

Las będący wynikiem analizy składniowej zakończonej sukcesem – czyli kiedy tekst wejściowy został uznany za poprawny – może być widziany jako zbiór drzew, jednak wygodniej jest uwzględnić w definicji cechy które łączą wszystkie drzewa. Zakładam, że wszystkie drzewa reprezentują struktury dla całości analizowanego tekstu w pełnej długości (zob. podsekcję o zdaniach niepoprawnych). W związku z tym suma segmentów reprezentowanych przez każdy liść w każdym drzewie daje cały tekst wejściowy. Wspólny jest również korzeń wszystkich drzew, dla uproszczenia można założyć, że analiza była zadaniem znalezienia interpretacji wejściowego tekstu jako tylko jednego rodzaju jednostki składniowej. Poza korzeniem, dowolna liczba innych węzłów może występować w tej samej formie we wszystkich lub części drzew.

Definicja lasu może więc wyglądać następująco: jest to zbiór wierzchołków, podzielonych na wewnętrzne oraz liście, z jednym wierzchołkiem wewnętrznym specjalnym – oznaczonym jako korzeń lasu. Każdy węzeł wewnętrzny posiada niepusty zbiór niepustych list swoich potomków. Każdy liść reprezentuje pewien odcinek tekstu wejściowego (ma pozycje początkową i końcową). W każdej pojedynczej liście potomków, węzły są uporządkowane według miejsca występowania fragmentu tekstu przez nie reprezentowanego w wejściowym tekście (czyli według pozycji początkowej), i reprezentowane przez nie odcinki nie mają części wspólnych a ich suma jest również odcinkiem (pozycja końcowa poprzedniego potomka jest pozycją początkową następnego). Dla węzła wewnętrznego odcinki będące sumami odcinków na poszczególnych listach potomków są sobie równe i są równe odcinkowi reprezentowanemu przez rodzica.

Z tej definicji wynika, że węzeł może mieć wielu rodziców oraz może występować w kilku różnych interpretacjach jednego rodzica. Każdy wierzchołek wewnętrzny definiuje pod-las, który jest właściwym lasem dla fragmentu tekstu.

Przyjmując terminologię analizy składniowej, węzły wewnętrzne lasu reprezentują symbole nieterminalne gramatyki z parametrami, których niektóre wartości mogą być ustalone, a liście odpowiadają symbolom nieterminalnym również z parametrami i jednocześnie odpowiadają leksemom tekstu. Korzeń odpowiada symbolowi początkowemu analizy, zwykle *wypowiedzenie*. Na tożsamość wierzchołka składa się jego pozycja początkowa i końcowa, symbol gramatyki i parametry, i w lesie nie powinny wystąpić dwa takie same wierzchołki, co wynika również z budowy gramatyki.

Pewne drzewo należy do lasu kiedy da się je otrzymać wybierając dla każdego węzła lasu jedną listę potomków i następnie usuwając ze zbioru węzłów te, do których nie można już dotrzeć z korzenia.

1.1 Możliwe wariacje lasu

1.1.1 Analiza tekstu niepoprawnego

Niektóre typy analizatorów dla gramatyk bezkontekstowych można zmodyfikować tak, aby nawet dla niezaakceptowanego napisu podały na wyjściu pewien zbiór

drzew ([3]), które zostały wygenerowane w czasie analizy przed stwierdzeniem porażki. Ten zbiór nie spełnia powyższej definicji bo drzewa te nie będą mieć wspólnego korzenia, a więc definicja i struktury danych wymagałyby pewnej modyfikacji. Mogłaby ona polegać na dodaniu sztucznego węzła korzenia łączącego korzenie wygenerowanych pod-lasów jako alternatywne listy potomków (dodając do tych list węzły terminali, których brakuje, aby korzeń reprezentował całą długość napisu wejściowego). Taka informacja jednak nie wiele mówi czytelnikowi, nawet autorowi gramatyki, o powodzie odrzucenia wejścia, dlatego nie zajmuję się tym przypadkiem w tej pracy. Zbiór poddrzew który otrzymuje się w ten sposób jest bardzo zależny od metody parsowania i innych szczegółów implementacyjnych. Zamiast tego analizatory wstępujące zwykle podają na wyjściu listę możliwych symboli oczekiwanych zamiast symbolu po którego wczytaniu nastąpiło stwierdzenie niezaakceptowania wejścia.[1]

1.1.2 Medium liniowe i nieliniowe

Tekst pisany oraz mowa są mediami liniowym, tzn. występuje w nich pełny porządek i dwa symbole nigdy nie mogą wystąpić równocześnie. W gramatykach dla języka polskiego, takich jak wspominana gramatyka Szpakowicza i GFJP w prawdzie wykorzystane zostały specjalne notacje do stwierdzenia, że dane symbole (pod-frazy) mogą wystąpić w realizacji symbolu wyższego poziomu (frazie) w dowolnej kolejności. Jednak informacja o tej kolejności jest zachowywana w wyniku analizy, dzięki czemu człowiek lub program dokonujący analizy semantycznej będzie mógł stwierdzić czy kolejność symboli w jakimś stopniu wpływa na znaczenie całego wypowiedzenia. Zwykle nawet jeżeli nie wpływa ona na strukturę składniową zdania to uważa się, że zdanie o innej kolejności symboli nie jest mu równoważne pod jakimś innym względem.

Inaczej jest na przykład w przypadku języków migowych, z których te będące w najszerszym użyciu są językami naturalnymi i wynikiem ich analizy składniowej są również zbiory drzew (według Noama Chomsky'ego jest to bezpośrednia konsekwencja występowania rekurencji w językach naturalnych). Z powodu nieliniowego medium, którym jest tzw. scena migania, kilka symboli wewnątrz wypowiedzenia może występować jednocześnie. Polski język migowy (pjm) i spokrewnione z nim języki (np. francuski i amerykański, a w przeciwieństwie do brytyjskiego) są bogate w znaki jednoręczne, które mogą być artykułowane równocześnie z innym znakiem, i w pewnych notacjach są również zapisywane jeden nad drugim. W takich wypadkach wydaje się, że definicja lasu powinna być odpowiednio zmodyfikowana aby dopuszczać częściowy porządek symboli. Natomiast aby uniknąć sytuacji, w której dwa identyczne zdania będą wyświetlane jako różniące się od siebie, można wprowadzić pewien dowolny, ale ustalony porządek w zbiorze znaków, i wyświetlać je posortowane według tego porządku.

[1] Jest to informacja na której można polegać, ale skądinąd programistom wiadomo, że często jest mało pomocna w ustalaniu powodu niezaakceptowania wejścia. Jeśli wejście było poprawne z wyjątkiem jakiegoś jego fragmentu, to człowiek analizujący napis zwykle wskaże na zupełnie inny fragment niż parser wstępujący.

W tej pracy zakładam jednak, że kolejność symboli jest ustalona, a dostosowanie przedstawionych rozwiązań do języków migowych może wymagać o wiele większych zmian.

1.2 Złożoność algorytmów na lasach

Podana definicja jest definicją lepiej pasującą do "upakowanego lasu" wg. [2]. Wierzchołków w lesie może być znacznie mniej niż wierzchołków we wszystkich indywidualnych drzewach oraz niż samych drzew. Parser Birnam wykorzystywany w analizatorze Świgra pokazuje, że taki upakowany las dla gramatyki bezkontekstowej można wygenerować w czasie o wiele krótszym niż O(liczba drzew wynikowych), mimo, że jest on znacznie dłuższy niż O(liczba wierzchołków w lesie). Wygenerowanie wszystkich drzew siłą rzeczy musiałoby zająć co najmniej O(liczba wierzchołków we wszystkich drzewach lasu), i co najmniej taką samą złożoność wymusiłoby na każdym programie do wizualizacji wyników. Dowolna wizualizacja która następnie pokazywałaby te wszystkie drzewa bez zmniejszenia ilości powtarzających się informacji wymagałaby również odpowiednio długiego czasu na przeczytanie przez użytkownika, odpowiednio dużej objętości pliku do zapisania jako prosty dokument, lub odpowiednio dużej ilości papieru do wydrukowania.

Dlatego praktyczne wizualizacje i inne programy przetwarzające wynik analizy prawdopodobnie muszą unikać pełnego "rozpakowania" lasu i wyświetlać tylko tę jego część, która interesuje czytelnika. Przykładowo dla wiersza z rozdziału [link], Świgra wygenerowała las zawierający aż 640076800 drzew a jedynie 1305 węzłów, i, jak dalej postaram się pokazać, nie jest to wcale wynik bezużyteczny lub trudny do dalszego przetwarzania.

O ile w pojedynczym drzewie przeszukiwanie zarówno wgłąb i wszerz trwa O(liczba wierzchołków), to w upakowanym lesie algorytm oparty na pełnym przeszukiwaniu wgłąb naraża się na złożoność rzędu (liczba wierzchołków) ^ (liczba wierzchołków). Aby uzyskać złożoność O(liczba wierzchołków), obliczenia wykonywane na całym lesie mogą zapamiętywać wierzchołki raz odwiedzone i przy przeszukiwaniu wgłąb nie przetwarzać po raz kolejny ich pod-lasów, o ile na potrzeby danego obliczenia da się wykorzystać wartość obliczoną przy pierwszym przeszukiwaniu. Podobnie jest możliwe przeszukiwanie wszerz.

Przykładowo liczbę indywidualnych drzew w których pojawia się dany węzeł lasu, spośród wszystkich możliwych drzew, można obliczyć dla wszystkich węzłów stosując przeszukiwanie wszerz. Natomiast przy ustaleniu pewnego porządku dla indywidualnych drzew (których nie generujemy – chyba, że użytkownik tego zażąda), każdemu węzłowi można przyporządkować numery pierwszych n drzew, w których on występuje, w czasie o pesymistycznym oszacowaniu O(n * liczba wierzchołków) stosując przeszukiwanie wgłąb[2].

[2]Być może można to zrobić szybciej, ale problem wydaje się skomplikowany.

2 Wizualizacja

Sposób przedstawienia informacji ma ogromny wpływ na łatwość i czas potrzebny na ich zrozumienie. Szczególnie kiedy zbiór danych jest obszerny nieczytelna jego wizualizacja może uniemożliwić znalezienie szukanej odpowiedzi. Las w taki sposób jak go zdefiniowaliśmy jest szczególnym przypadkiem grafu, może być również interpretowany jako zbiór drzew. Zarówno metody przedstawiania ogólnie grafów, jak i konkretnie drzew, jako struktur danych są tematem wielu badań, w związku z czym można by oczekiwać, że znalezienie sposobu na czytelną wizualizację lasu nie będzie problemem.

Struktura drzewa jasno sugeruje sposób wizualizacji, zarówno przez rodzaj danych, które reprezentuje (w naszym przypadku pewną konkretną interpretację składniowej struktury zdania), jak i przez odwołanie do słowa drzewo w języku ogólnym, a także przez ogromną ilość istniejących przedstawień w pracach z prawie wszystkich dziedzin nauki. Wizualizacje takie zwykle opierają się na tych samych zasadach co wizualizacje innych grafów: przy przedstawieniu na płaszczyźnie (ekranie komputera lub druku na powierzchni) umieszcza się węzły w postaci jakichś form geometrycznych tak aby nie pokrywały się, i łączy je krzywymi obrazującymi gałęzie drzewa, tak, aby krzywe nie przecinały się ani nie przecinały kształtów węzłów. Te warunki dają się spełnić dla każdego drzewa, mówi się więc, że drzewo jest płaskie lub jest tworem dwu-wymiarowym. Najłatwiej jest to zrobić, gdy węzły w relacji rodzica i potomka leżą blisko siebie.

Przy niektórych typach drzew wprowadza się pojęcie aktualnego węzła i całe drzewo rysuje w układzie dysku Eulera [termin?], tak, że aktualny węzeł jest blisko jego środka, a dalsze węzły są od niego mniejsze i odległości między nimi zbiegają do zera przy odległości od aktualnego węzła w drzewie rosnącej do nieskończoności. Pozwala to uzyskać efekt dzięki któremu wybrany węzeł jest najwyraźniej widoczny a rozgałęzienia w drzewie nie powodują, że każdy kolejny poziom zajmuje eksponencjalnie więcej miejsca w diagramie. Ten zabieg nie wydaje się jednak przydatny przy drzewach struktury składniowej tekstu.

Dodatkowo w drzewie obrazującym hierarchię oczywiste jest umieszczenie korzenia na jednym krańcu diagramu, a węzły potomków przesunięte względem węzła rodzica w ustalonym kierunku. Dla drzewa analizy łatwo uzyskać dodatkową czytelność przez zasygnalizowanie którą część tekstu opisuje dany węzeł. To może być zrealizowane przez nadanie rodzicowi rozmiaru sumy rozmiarów jego potomków. W ten sposób uzyskujemy to, że węzły na tym samym poziomie drzewa leżą w jednym "wierszu", a pokrywają "kolumny" odpowiadające fragmentom tekstu przez nie reprezentowanym. Można je wtedy potraktować jak *tabelę*, i dla zwiększonej czytelności pokolorować tła wierszy lub kolumn dwoma (lub więcej) kolorami naprzemiennie – efekt często stosowany dla tabel. Wiersze te nie muszą być prostymi, mogą być przykładowo okręgami współśrodkowymi o rosnącym promieniu względem odległości od korzenia, a kolumny będą wtedy wycinkami koła.

Oprócz położenia węzłów i kształtów gałęzi znaczenie ma również dobór kolorów (o tym nie piszę w tej pracy ponieważ informacje na ten temat należą do prac z dziedzin takich jak psychologia i mogą być skomplikowane, chociaż

podstawowa intuicja będzie często wystarczająca), ilość szczegółów zapisanych w kształcie węzła takich jak nazwa jednostki z parametrami i sposób ich zapisu (stosowanie skrótów i kodów występujących w danym formalizmie lub stosowanie pełniejszych nazw, czy też symboli graficznych). W interaktywnym programie można wyświetlać tylko tę najważniejszą część opisu węzła domyślnie, ale wyświetlać pełniejsze informacje lub rozwijać skróty w reakcji na działania użytkownika.

Nie ma tak dobrze ustalonych zasad dla wizualizacji grafów w przypadku ogólnym. Graf jest uważany za twór trój-wymiarowy i dla wielu grafów bez względu na umiejscowienie węzłów na płaszczyźnie nie da się znaleźć krzywych dla wszystkich krawędzi tak, aby nie przecinały się[3]. Chcąc przedstawić graf na płaszczyźnie dopuszcza się więc krzyżujące się krawędzie, ale minimalizując liczbę ich przecięć lub korzystając z innych kryteriów. Nie istnieje jednak efektywny algorytm gwarantujący minimalną liczbę przecięć i większość z używanych algorytmów próbuje iteracyjnie poprawiać wygląd grafu z ograniczeniem na liczbę iteracji. Algorytmy te są tematem co najmniej jednej corocznej konferencji o rysowaniu grafów (ang. graph drawing). Istnieje kilka znanych algorytmów, wiele z nich zaimplementowanych w istniejących programach i ich wyniki nie są zadowalające dla lasów analizy składniowej, które również są tworami trójwymiarowymi w ogólnym przypadku.

Jednak nawet tam gdzie stosuje się trójwymiarowe reprezentacje grafów (np. w genetyce), czy to za pomocą projekcji na płaszczyznę z uwzględnieniem perspektywy, czy też za pomocą dwóch ekranów, czy fizycznego modelu, używane algorytmy rozmieszczania węzłów w przestrzeni są najczęściej iteracyjne i nie odporne na lokalne minima. Dla lasów analizy składniowej można sobie wyobrazić wizualizację polegającą na wyrysowaniu części lasu bez rozgałęzień stosującą reguły podane wcześniej dla pojedynczych drzew w pewnej płaszczyźnie, natomiast tam gdzie węzeł ma więcej niż jedną możliwą listę potomków, stosując przesunięcie węzłów na kolejnych listach prostopadle do płaszczyzny o stałą odległość. Szerokość drzewa w kierunku prostopadłym do płaszczyzny mogłaby być eksponencjalnie duża w stosunku do głębokości drzewa. Inna oczywista możliwość to narysowanie wszystkich możliwych pełnych drzew analizy w oddzielnych płaszczyznach, otrzymując szerokość lasu rzędu n^n. Obydwie możliwości wydają się niepraktyczne.

W tej pracy nie podaję rozwiązania problemu narysowania grafu na płaszczyźnie ani problemu umiejscowienia węzłów lasu w przestrzeni trójwymiarowej.

Sposób wizualizacji lasu z pewnością musi być dostosowany do konkretnego zastosowania o ile jest ono znane. Problem jest również spokrewniony z problemem ręcznej edycji takich struktur, ponieważ najpopularniejsze dziś edytory do wszelkich rodzajów struktur są typu WYSIWYG, a więc muszą znać sposób wizualizacji aby umożliwić edycję.

[3]Można to udowodnić np. dla kliki o rozmiarze 5.

Natomiast aby udowodnić, że każdy graf można przedstawić w przestrzeni trójwymiarowej spełniając wcześniej podane warunki dla węzłów i krawędzi, wystarczy umieścić węzły na prostej w dowolnej kolejności i wytyczyć krawędzie jako łuki łączące węzeł początkowy i końcowy w płaszczyźnie zawierającej daną prostą ale innej dla każdego łuku.

Oczywiście nie można dziś przewidzieć listy wszystkich zastosowań wizualizacji wyników analizy składniowej tekstu, ale pod uwagę biorę te, z którymi spotkałem się do dziś:

- poprawianie, analiza lub zobrazowanie działania gramatyki języka.
- poprawianie, analiza lub zobrazowanie działania programu analizatora.
- wybranie podzbioru właściwych drzew (według oceny człowieka) spośród wyników działania gramatyki generującej nadmiarowe interpretacje – problem, który można nazwać dezambiguacją.

2.1 Istniejące podejścia

2.1.1 GraphML i XSLT

GraphML to schemat XML dla opisu grafów stworzony przez grupę [xxx]. Dostępne są dla niego definicje schematów w formatach DTD i XSD oraz formalna definicja. Zwykle używana końcówka nazwy plików to .graphml, takie pliki są rozpoznawane przez kilka programów. Popularny program graficzny yEd do prezentacji grafów używa tego formatu jako swojego domyślnego formatu, zapisując w nim dodatkowe informacje nie zdefiniowane ale dopuszczane przez schemat.

Wykorzystanie prostego schematu XML jest ciekawe o tyle, że istnieje powszechnie znany i ustandaryzowany język przetwarzania struktury pliku XML pod nazwą XSLT. W pliku XSLT zapisany jest sposób tłumaczenia jednego schematu XML na inny. Biorąc jako schemat wejściowy GraphML a jako wyjściowy SVG – XML-owy format grafiki wektorowej będący standardem W3C – można w XSLT zapisać algorytm wizualizacji grafu. Procesory XSLT niestety nie są przystosowane do interpretowania skomplikowanych algorytmów, a raczej do prostych przekształceń drzewa dokumentu, niemniej możliwe jest zapisanie dowolnego algorytmu iteracyjnego i procesor XSLT "saxon" poprawnie je interpretuje, mimo słabej efektywności.

Oto przykładowy dokument GraphML opisujący proste drzewo ze zdefiniowanymi dwoma parametrami dla węzłów i jednym dla krawędzi:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphml
  xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
   "http://graphml.graphdrawing.org/xmlns">
 <key attr.name="terminal" attr.type="string"
   for="node" id="t" />
 <key attr.name="nonterminal" attr.type="string"
   for="node" id="nt" />
 <key attr.name="rule" attr.type="string"
   for="edge" id="r" />
 <graph edgedefault="directed" id="tree">
```

```
  <node id="n0">
   <data key="nt"><![CDATA[formaczas]]></data>
  </node>
  <node id="n1">
   <data key="t"><![CDATA[śpiewać]]></data>
  </node>
  <edge source="n0" target="n1">
   <data key="r"><![CDATA[n_cz4]]></data>
  </edge>
 </graph>
</graphml>
```

A to fragment przykładowego dokumentu XSLT autorstwa [xxx] z [5] implementującego algorytm sprężynowy [check] umiejscawiania węzłów w grafach:

Po wykonaniu komendy "saxon ..." gdzie podstawione zostały odpowiednie nazwy plików otrzymamy obrazek taki jak widoczny na [Figure xxx]. Podobny dokument możnaby napisać dla dowolnego algorytmu przedstawiającego lasy lub pojedyncze drzewa wynikowe analizy składniowej.

2.2 Dendrarium

Projekt Dendrarium powstaje pod kierownictwem dr Marcina Wolińskiego w Instytucie Podstaw Informatyki PAN głównie na potrzeby projektu treebanku dla części Narodowego Korpusu Języka Polskiego (NKJP). Jest to aplikacja składająca się z części serwerowej i jej klienta. Aplikacja pozwala na efektywne zarządzanie zbiorem wyników analizy składniowej w postaciach oryginalnych wyprodukowanych przez parser i po ujednoznacznieniu przez człowieka. Jej główne funkcje to magazynowanie tych danych i ujednoznacznianie lasów wynikowych wraz ze wszystkimi funkcjami administracyjnymi wspomagającymi ten proces, czyli logowanie i przydzielanie ról użytkownikom, przydzielanie im zadań z kolejki, rozstrzyganie konfliktów pomiędzy wyborami różnych użytkowników. Główne role to *dendrolog* (osoba zajmująca się ujednoznacznianiem lasów, odrzucaniem lasów niezawierających poprawnych interpretacji i sygnalizowaniem problemów potencjalnie pochodzących od błędów w gramatyce), *superdendrolog* (osoba rozstrzygająca końcową ocenę wyniku analizy zdania w przypadku konfliktu pomiędzy wyborami dendrologów) i *gramatyk* (ta rola pozwala użytkownikowi przeglądać lasy zgłoszone do niego przez innych użytkowników).

Lista lasów nieprzetworzonych jest trzymana w relacyjnej bazie danych na serwerze i na żądanie dendrologa, przydzielanych jest mu kilka zdań z tej listy, dla których lasy analizy należy ujednoznacznić lub ocenić decyzję podjętą przez system automatycznie. Lista wybranych zdań pojawia się w jego panelu, który jest podstroną strony internetowej serwisu. Dla wymienionych ról (dendrologa, superdendrologa i gramatyka) cała obsługa aplikacji jest wykonywana właśnie przez przeglądarkę internetową za pośrednictwem tej strony. Dendrolog może wybrać jedno ze zdań przechodząc do panelu konkretnego zadania do wykonania. Istotny jest dla nas tylko przypadek, kiedy dla zdania parser wygenerował

niejednoznaczny las analizy. Na górze panelu widoczne jest zdanie o którym mowa w kontekście fragmentu go poprzedzającego i następującego po nim w obrabianym tekście. To dlatego, że w systemie wybierane są interpretacje składniowe poprawne nie tylko w sensie składniowym ale również spójne z semantyką zdania w danym kontekście, nie zaś wszystkie poprawne składniowo interpretacje. Trzeba zaznaczyć, że ostatecznie wybierana jest tylko jedna interpretacja nawet jeżeli kilka z nich mogłoby być poprawne na wszystkich poziomach. Proces wyboru tego pojedynczego drzewa realizowany jest jako sekwencja pytań do użytkownika, w której przy każdym wyborze proszony jest on o wybór jednej z interpretacji dla kolejnych niejednoznacznych węzłów w kolejności od tych najbliższych korzeniowi. Niektóre wybory dokonywane są automatycznie na podstawie różnych kryteriów. Użytkownik dokonuje wyboru na podstawie informacji widocznych na ekranie, a nie na podstawie całego obrazu lasu, dzięki czemu potencjalnie unika bycia zarzuconym dużą ilością informacji nie pomocnych w podjęciu wyboru. Definicja lasu w Dendrarium rozszerzona jest o informację o nazwach reguł gramatyki użytych do wyprowadzenia każdego symbolu oraz o *centrum*, lub *podfrazie głównej* frazy. Ta druga informacja pochodzi z gramatyki, która jest zmodyfikowana (w stosunku do wspomnianej już GFJP) tak, aby prawe strony reguł wskazywały jeden symbol jako centrum frazy.

Poniżej kontekstu zdania wyświetlane są trzy pola. W pierwszym z nich widoczna jest część lasu już ujednoznaczniona, czyli drzewo, w którym korzeniem jest korzeń lasu a liśćmi węzły które nadal mają więcej niż jedną możliwą listę potomków, lub też liście lasu. W drzewie, ani w żadnej innej części Dendrarium nie są wyświetlane pełne drzewa analizy, włączone są do nich tylko węzły, w których występuje rozgałęzienie lub są bezpośrednimi potomkami takich węzłów. Jest to więc postać skrócona tak jak opisano w [6]. Węzły te posiadałyby tylko symbole gramatyki (zwykle nie odpowiadające żadnej konstrukcji składniowej znanej z tradycyjnej gramatyki szkolnej języka polskiego) i wartości parametrów wynikające tylko z wartości parametrów ich potomków lub przodków. Ukryte są również nazwy reguł użytych do wyprowadzenia tych ukrytych symboli. Wydaje się, że nazwy reguł są informacją zbędną dla dendrologa jak i do oceny poprawności danej interpretacji generalnie. Są one jednak wyświetlane dla wszystkich widocznych węzłów wewnętrznych.

Struktura drzewa ma tradycyjny układ z korzeniem u góry i wypozycjonowanym po środku kolumny fragmentu tekstu, który reprezentuje. Każdy węzeł zawiera tylko symbol gramatyki (będący pewnym skrótem nazwy konstrukcji składniowej, którą dany symbol implementuje). Kolor tła węzła mówi użytkownikowi o podstawie wyboru danej interpretacji symbolu nieterminalnego. Różne kolory odpowiadają więc węzłom z tylko jedną możliwa interpretacją, węzłom, dla których wyboru interpretacji dokonał użytkownik w którymś z poprzednich kroków, węzłowi rozstrzyganemu w danym kroku, oraz węzłom rozstrzygniętym w sposób automatyczny. Bezpośrednio pod węzłem jest nazwa reguły gramatyki. Dopiero po przesunięciu kursora myszki nad dany węzeł, pojawia się okienko zawierające listę parametrów danego symbolu z ich wartościami. Okienko przykrywa część widoku drzewa. Parametry są zawsze wyświetlane w takiej samej kolejności. Po najechaniu kursorem na nazwę parametru, wartość

elementu lub fragment tej wartości (pod-ciąg znaków dla atomu wewnątrz ciągu znaków reprezentującego cały term), podświetlane są wszystkie węzły w aktualnym widoku drzewa oraz innych drzewach wyświetlanych aktualnie w przeglądarce, których nazwa symbolu lub parametr zawiera daną wartość. Dzięki temu szybko można zobaczyć jaka część drzewa ma daną wspólną cechę z danym węzłem, często wskazuje to na terminal z którego pochodzi dana wartość parametru. Pod elementem aktualnie rozstrzyganym wyświetlany jest kształt symbolizujący pewne poddrzewo (można go uznać za formę elipsy) i znak zapytania. Pod węzłem wypisany jest fragment zdania, którego strukturę opisuje. Elementy główne frazy pokazywane są w dwojaki sposób. Dla węzłów widocznych w wyświetlanym drzewie ich ścieżka łącząca z węzłem rodzica jest podkreślona grubszą linią koloru szarego. Natomiast dla węzła aktualnie rozstrzyganego podfraza, która należy do centrum frazy we wszystkich interpretacjach frazy, jest podświetlana jaśniejszym kolorem. Wewnątrz tej podfrazy jej podfraza główna jest również podświetlana, tak, że wyraz "najbardziej centralny" podświetlany jest najjaśniejszym tłem, jeśli taki istnieje. Ułatwia to dendrologom wybór poprawnej interpretacji.

Jedną z interpretacji dendrolog wybiera z pośród listy możliwych poddrzew pokazanych poniżej ujednoznacznionej części drzewa. Dla każdego poddrzewa pokazany jest tylko jeden poziom – nie jest powielany korzeń poddrzewa, którym jest zawsze węzeł rozstrzygany. Poddrzewa do wyboru pogrupowane są według podziału frazy rozstrzyganej na podfrazy. Przykładowo jeśli fraza *«Ala ma kota»* może być zinterpretowana na dwa sposoby jako jednostka nadrzędna podfraz *«Ala»* i *«ma kota»*, oraz na dwa sposoby jako podfrazy *«Ala ma»* i *«kota»*, to użytkownik zobaczy cztery poddrzewa w dwóch grupach o różnych kolorach tła. Dla każdego węzła potomnego rozstrzyganej frazy pokazany jest zarówno symbol gramatyki i lista jego parametrów. Wewnątrz każdej grupy Dendrarium wyróżnia innym kolorem parametry, których wartości różnią się między poddrzewa. Pozwala to użytkownikowi natychmiast zauważyć różnice między poddrzewami. Z jednej strony przyspiesza to znacznie proces wyboru najodpowiedniejszej interpretacji, z drugiej jednak może zmniejszać uwagę poświęcaną przez użytkownika weryfikacji poprawności pozostałych parametrów.

Do dyspozycji jest też możliwość podejrzenia całego pod-lasu dla każdego wariantu do wyboru i możliwość podejrzenia całego lasu. Przeglądanie lasu jest zrealizowane przez wyświetlenie pojedynczego drzewa, ale z możliwością przełączania pomiędzy różnymi realizacjami każdego symbolu w reakcji na działanie użytkownika. Pod każdym węzłem, w którym istnieje niejednoznaczność wyboru listy potomków, są wyświetlane dwa przyciski z symbolami strzałek w lewo i w prawo. Użycie przycisku lub kółka myszy pokazuje przełączenie na poprzednie lub następne poddrzewo / pod-las. Jeśli wyświetlana jest pierwszy lub ostatni spośród wariantów to jeden z przycisków jest ukrywany. Również w widoku ujednoznacznionej części lasu jest możliwość poruszania się po odrzuconych wariantach lasu w ten sam sposób.

Ostatnim polem na dole strony jest miejsce na wpisanie komentarza dendrologa i przyciski powodujące stwierdzenie braku poprawnego drzewa w lesie.

Większość elementów tu opisanych jest elementami aktualnej wersji Den-

drarium, a nie założeniami projektu, więc może ulec zmianie.

2.2.1 Ocena efektywności

Według opinii osób zaangażowanych w projekt, ten rodzaj wizualizacji oraz cały interfejs użytkownika sprawdza się dobrze w aktualnym projekcie. Interfejs jest łatwy w użyciu i z pewnością nie wymaga od użytkownika długiego okresu przyzwyczajania się do niego.

Nie istnieją niestety dokładne dane na temat wzorców zachowań użytkowników na różnym poziomie przyzwyczajenia do interfejsu aplikacji. Badania takie są często robione dla programów komputerowych i dzięki nim można ustalić efektywność interfejsu oraz skrócić ścieżkę do wybieranej opcji, zarówno mierzoną w czasie jak i ilości akcji wykonywanych przez użytkownika (przyciśnięć klawiszy, kliknięć i przesunięć myszki, ilość dotknięć ekranu w przypadku ekranu dotykowego, ilość przewinięć zawartości okna za pomocą pasków przewijania itd).

Średni czas pracy dendrologa poświęcany jednemu zdaniu wynosi około trzech minut. Dendrolodzy rzadko przerywają normalny tok dokonywania wyborów aby skorzystać z możliwości podejrzenia większej części lasu, są to sytuacje "awaryjne". Co ciekawe prawdopodobieństwo kolizji pomiędzy wyborami dwóch dendrologów dla jednego zdania, dla zdań korpusu i aktualnie używanej gramatyki wynosi aż 0,26.

W panelu superdendrologa występuje konieczność obejrzenia dwóch drzew jednocześnie w celu porównania ich struktur. Aktualnie interfejs ogranicza się do widoku dwóch drzew obok siebie. Być może możliwe jest poprawienie efektywności pracy przez ułatwienie dostrzeżenia różnic między dwoma drzewami. (Jest to problem odrębny od zadania porównania dwóch lasów.)

2.3 Arkusze syntaktyczne

Arkusze syntaktyczne są formą prezentacji całego lasu wynikowego analizy na płaszczyźnie i formą bardzo dobrze dostosowaną do druku na papierze. Jest to formalizm pierwszy raz opisany przez prof. Janusza Bienia w 2007. Ponieważ ich struktura jest wyczerpująco opisana w [4] i [6], nie przytaczam tu pełnej definicji oraz wszystkich zaproponowanych tam wariantów arkuszy. Warianty te pozwalają naświetlić pewne konkretne elementy struktury lasu oraz oszczędzić zajmowaną przestrzeń ukrywając nieciekawe lub mało relewantne fragmenty tej struktury.

Arkusz syntaktyczny jest tabelą, w której, tak jak w opisywanej częstej metodzie rysowania drzew analizy, pozycja pozioma jest połączona z pozycją w tekście, a kolumny odpowiadają fragmentom tekstu. Szerokość tabeli jest więc ściśle powiązana z długością tekstu analizowanego, a wysokość jest proporcjonalna do liczby węzłów w lesie. Pozwala to stwierdzić, że arkusz taki efektywnie wykorzystuje przestrzeń i nawet lasy o gigantycznych liczbach drzew mogą zmieścić się w dokumentach drukowanych.

Arkusz syntaktyczny jest rodzajem wizualizacji wymagającej wyjaśnienia i taką, której czytanie w celu znalezienia konkretnej informacji może sprawiać wysiłek nawet znając zasady formalizmu. W związku z tym w zastosowaniach komputerowych lub tam gdzie istnieje możliwość interakcji z użytkownikiem nadal efektywniejszą wydaje się struktura drzewa z możliwością nawigacji po lesie przełączając interpretację poddrzewa.

Mimo, że możliwość istnienia arkuszy o różnej szczegółowości opisu lasu zwiększa elastyczność wizualizacji, powoduje też, że każdy arkusz musi być akompaniowany przez dodatkowy opis aby czytelnik wiedział jak ma dany arkusz interpretować.

Nieintuicyjnym jest również fakt, że położenie pionowe komórek nie niesie wiele informacji, w szczególności umieszczenie kilku komórek w jednym wierszy nie świadczy o jakimkolwiek związku między nimi. W świetle tego identyfikatory wierszy oraz naprzemienne kolorowanie teł wierszy nie niosą zysku dla efektywności lokalizowania informacji.

2.4 Biblioteka do wizualizacji lasu

W ramach pracy powstała biblioteka dla języka JavaScript zawierająca narzędzia do wizualizacji lasów analizy składniowej. Konkretnie implementuje ona kontrolki wyświetlające Arkusze Syntaktyczne i lasy w postaci takiej jak Dendrarium oraz zawiera podstawowe klasy i narzędzia dla lasów. Biblioteka korzysta z hierarchii DOM dla HTML i dostarczane przez nią kontrolki są również podobne do elementów DOM pod względem interfejsu programisty, w związku z czym jest łatwa do wykorzystania na stronach i w serwisach internetowych. Wygląd większości elementów może być łatwo dostosowywany do wyglądu strony za pomocą arkuszy stylów CSS, więc zmiana wyglądu nie wymaga zmian w kodzie i wygląd jest opisany łatwym, standardowym językiem, powszechnie znanym przez twórców stron internetowych.

Biblioteka powstała w pierwszej kolejności na potrzeby projektu Dendrarium (zob. [xxx]) i jest obecnie wykorzystywana w tym projekcie do budowania interfejsu użytkownika.

2.4.1 Program Przeglądarka Lasów

mt

2.4.2 Dokumentacja API

compl

2.4.3 Kierunki rozwoju

gwt

2.4.4 Analizator morfologiczny

Na potrzeby pracy powstał prosty analizator morfologiczny Gpleusz. Jego jedynym zadaniem było wypełnienie luki w możliwości analizy składniowej tekstu za pomocą komputerowej realizacji GFJP, Świgry z użyciem wolnego oprogramowania. Jakość jego wyników pozostawia aktualnie wiele do życzenia i z pewnością nie można go wykorzystać do celów automatycznej analizy poza eksperymentami. Głównym źródłem danych morfologicznych jest wydanie Słownika Polszczyzny Lat 60-ych pod nazwą Słownik Frekwencyjny, które jest anotowane tagsetem IPI PAN czyli tym, którego używa Świgra i Morfeusz. W uproszczeniu program dokonuje segmentacji tekstu wejściowego a następnie sprawdza możliwe interpretacje dla każdego segmentu i generuje wyjściowy *graf morfologiczny* jakiego potrzebuje Świgra. W najprostszym przypadku segment jest odnajdywany w indeksie segmentów ze słownika i w ten sposób otrzymuje gotową odpowiedź[4]. Gdy segment nie zostaje odnaleziony, program nie poddaje się jednak i dopasowuje segment o najdłuższym wspólnym sufiksie z indeksu. Dzięki temu udało się zanalizowanie Świgrą dwóch spośród wielu istniejących tłumaczeń na język polski pierwszej strofy wiersza Lewisa Carolla *«Jabberwocky»*:

Było smaszno, a jaszmije smukwijne

Świdrokrętnie na zegwniku wężały,

Peliczaple stały smutcholijne

I zbłąkinie rykoświstąkały.

– Dżabbersmok, Maciej Słomczyński

oraz:

Grozeszły się po mrokolicy

Smokropne strasznowiny:

Dziwolęk znowu smokolicy

Ponurzył się w grzęstwiny.

– Dziwolęki, Antoni Marianowicz i Hanna Bałtyn

W drugim tłumaczeniu dwukropek musiał zostać zastąpiony przecinkiem. Wygenerowany został las zawierający 640076800 drzew i w szczytowym momencie Świgra zajęła 380MB pamięci.

Takie działanie, mimo, że minimalizuje ilość poprawnych wyrazów nierozpoznawanych, powoduje wprowadzenie błędów dwóch typów: rozpoznawanie nieistniejących słów jako istniejące, oraz, jeśli dla znalezionego segmentu o najdłuższym wspólnym sufiksie z szukanym segmentem w słowniku, wystąpiły tylko niektóre z możliwych interpretacji, to również w wyniku Gpleusza dla szukanego segmentu zabraknie tych potencjalnie poprawnych interpretacji.

Jako ciekawostka zdanie *«Ala ma kota.»* otrzymuje dodatkowe interpretacje poza tymi udokumentowanymi w [7] z powodu występowania formy "ma" w

[4] Niewątpliwą zaletą takiej budowy jest możliwość obarczenia danych wejściowych winą za niepoprawne działanie programu.

Słowniku Frekwencyjnym jako realizacji słowa "być" w zdaniach zawierających bezosobowe "nie ma". Większość tych nadmiarowych interpretacji udaje się wyeliminować dodając w Świgrze kod eliminujący tą interpretację "ma" jeśli nie występuje bezpośrednio po "nie".[5]

Możliwych jest wiele sposobów poprawienia jakości wyników Gpleusza, które nie zostały wykorzystane tylko z powodu ograniczeń czasowych. Jednym z nich jest sprawdzanie niektórych wartości tagów dla niektórych części zdania w słowniku z sjp.pl (również na licencji GPL), jak też sprawdzenie za jego pomocą, czy odgadnięta przez Gpleusza forma słownikowa istnieje.

[5]Ponieważ Słownik Frekwencyjny jest w rzeczywistości korpusem, występuje w nim ogromna ilość słownictwa nie występującego w typowych słownikach, oraz nieoczekiwanych interpretacji morfosyntaktycznych. Generalnie jest to dobra cecha choć płaci się za nią większą objętością lasów wynikowych analizy.