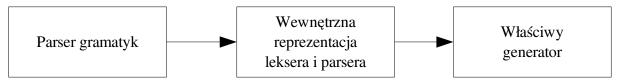
yatta (Yet Another Tool for Text Analysis) parser generator by Maciej Piróg

IMPLEMENTACJA

STRUKTURA

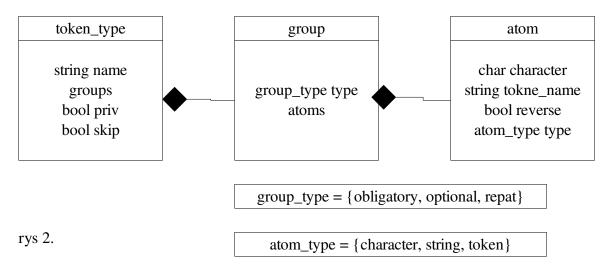
Yatta Parser Generator składa się z 2 części: parsera gramatyk i części odpowiedzialnej za właściwą generację parsera i leksera. Pomiędzy nimi znajdują się struktury przeznaczone do przechowywania wewnętrznej reprezentacji definicji leksera i parsera (rys 1.).



rys 1. Ogólna struktura programu

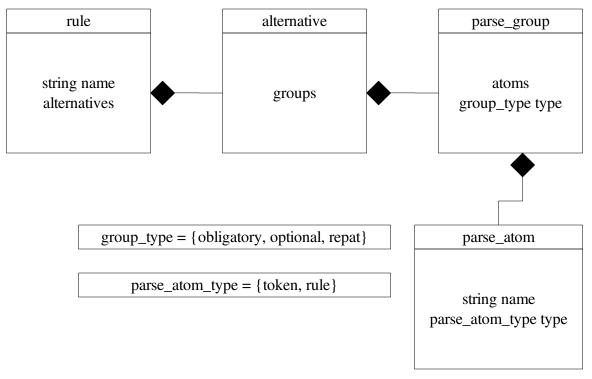
WEWNĘTRZNA REPREZENTACJA DEFINICJI LEKSERA

Definicja każdej jednostki leksykalnej przechowywana jest w obiekcie klasy token_type, która zawiera w sobie kolekcję kolejnych grup, z konkatenacji których składają się tokeny. Grupy zawierają kolekcję symboli podstawowych, takich jak znak, napis, odwołanie do innego tokenu (rys 2.).



WEWNĘTRZNA REPREZENTACJA DEFINICJI PARSERA

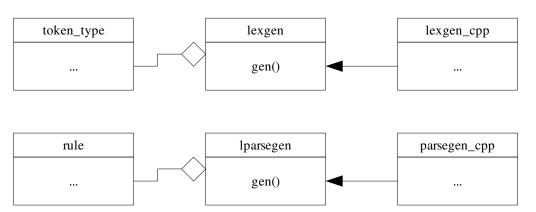
Definicja parsera jest prechowywana w strukturach analogicznych do struktury leksera. Każda produkcja składa się z alternatyw, które są konkatenacją grup. Grupy to konkatenacje tokenów (rys. 3).



rys 3. Wewnętrzna reprezentacja definicji parsera

WŁAŚCIWY GENERATOR PARSERA I LEKSERA

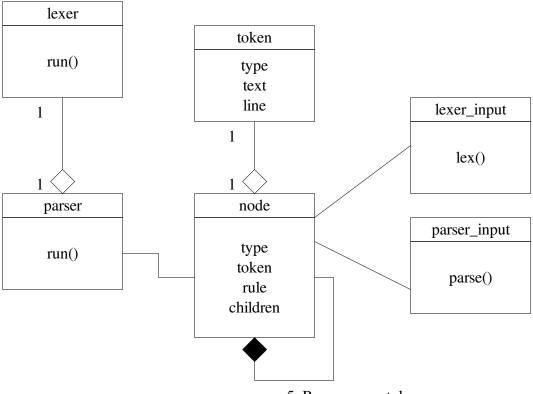
Do generowania leksera i parsra służą osobne klasy. Są to pochodne klas lexgen i parsegen. Takie rozwiązanie ma ułatwić wprowadzenie w przyszłości generowanie parserów także w innych językach niż C++ bez konieczności ingerowania w inne części całego programu (rys 4.).



rys 4. Za generowanie odpowiedzialne są klasy potomne

PARSER GRAMATYK

Parser gramatyk został wygenerowany przy użyciu programu Yatta (gramatyka pliku zawierającego gramatykę została wpisana "ręcznie" w pliku ygparsegram.cpp, który nie jest potrzebny przy kolejnej kompilacji). Dlatego struktura parsera gramatyk jest taka sama jak struktura każdego parsera wygenerowanego przy pomocy programu Yatta (rys 5.).



rys 5. Parser gramatyk

Klasy lexer_input i parser_input odpowiedzialne są za tłumaczenie drzewa rozbioru pliku z gramatyką do wewnętrznej reprezentacji definicji parsera i leksera.