

Описание алгоритма:

D - массив состояний, D[j][X] - состояние D<sub>j</sub>, когда следующий элемент за точкой X, где X - либо терминал, либо нетерминал, либо \$ - конец слова

E - массив символов алфавита

N - массив нетерминалов

P - словарь, в котором для каждой левой части правила хранится список правых частей

from\_net\_symb\_to\_int - словарь, где каждому нетерминалу сопоставляется его номер из N

start\_symb - стартовый символ грамматики

State - класс состояние, хранит номер нетерминала левой части, i - сколько символов уже выводимо от начала

слова(все написано примерно как рассказывал Павел

Ахтямов на лекции 9 и 3/4), left и right - массивы то что до

точки и то что после точки, каждый элемент массива -

пара вида ['N', число] или ['E', число] - в зависимости от

того, что находится слева и справа от точки - для

терминала - N для нетерминала - E, число - номер в

соответствующем массиве

Функция Fit - принимает введенные данные(грамматику) и строит массивы E, N, P

Scan - 2 вложенных цикла

Complete - 2 вложенных цикла

Predict - 4 вложенных цикла, но ассимптотика кубическая,

тк 1 из этих циклов по нетерминалам грамматики, еще 1

цикл по правилам в правой части из грамматики для этого нетерминала

Основная часть в этих функциях - разбор случаев -

следующий символ терминал или нет и проверка, а не

попали ли мы туда, уже где были, то есть лежит ли уже

очередная найденная ситуация в массиве ситуаций

predict\_result - принимает слово и собирает все функции,

описанные выше, вместе, сначала строя массив состояний для  $D[0]$ , а затем индукционно для каждого  $D[i]$  где  $i$  до длины слова, таким образом будет не более куба операций, тк каждая из функций Predict, Complete, Scan работает не более квадрата раз.

В конце функции - проверка, что правило, возвращающее нас в вершину дерева разбора действительно есть в массиве правил

В файле с тестами написано много тестов, в итоге coverage составил 94%, не просмотренными остались лишь строки с continue в циклах, когда я разбирал разные случаи

Конце файла main.py закомментировал для тестирования, но его можно раскомментировать и проверить работоспособность программы.