

Autorzy: Patryk Duda, Paweł Wandzel

Wymagania: Python 3.6, PLY 3.11

Uruchomienie: python3 main.py

Przykładowe wyrażenia: examples.txt

Sposób wykonania:

Preprocessing:

- sprowadzenie liczebników do formy podstawowej (przy pomocy słownika z numerals.txt)
- algorytm przekształcający liczbę w zapisie słownym na ciąg cyfr (z obsługą liczb z kropką dziesiętną)

Skaner (tokenizer.py)

- Po preprocessingu rozpoznajemy kluczowe wyrażenia za pomocą prostych wyrażeń regularnych i zamieniamy je na tokeny,

Parser (grammar.py)

- W tym pliku zawarliśmy całą gramatykę. Podczas parsowania sprawdzana jest poprawność wyrażenia (niepoprawne wyrażenia nie sparsują się), a tokeny są zamieniane na odpowiednie znaki możliwe do zinterpretowania przez funkcję eval()

Gramatyka obsługuje:

- liczby z opcjonalną kropką dziesiętną
- liczby: pi i e
- dodawanie/odejmowanie/mnożenie/dzielenie
- wyrażenia w nawiasach
- operację modulo
- potęgowanie
- funkcje: sinus, cosinus, pierwiastek kwadratowy, silnia (+ możliwość łatwego dodania kolejnych)

Ciekawostka:

Podczas rozszerzania gramatyki o funkcje trygonometryczne napotkaliśmy następujący problem:

Reguły funkcji trygonometrycznych w stylu SINUS expression generowały konflikty shift/reduce. Oznacza to, że nie wiadomo czy przykładowo kolejne składniki sumy są argumentem funkcji sinus (suma jako argument, konflikt rozwiązany jako shift), czy jednak argumentem funkcji jest liczba a poza sinusem dodajemy składnik. Po dogłębnym analizowaniu okazało się, że ten problem występuje również w języku naturalnym. Proszę rozważyć następujący fragment:

"sinus 45 plus 90"

Tak sformułowane wyrażenie choć poprawne jest nieprecyzyjne. Zazwyczaj gdy dyktujemy to dostarczamy dodatkowych informacji przykładowo:

"sinus z 45 plus 90 zamknij nawias" czyli sin(45+90)

"sinus z 45 plus 90 poza sinusem" czyli sin(45)+90

"sinus, a w nim 45 plus 90" czyli sin(45+90)

Po prostu czytając dodajemy dodatkowe informacje (o tym czy ma być shift czy reduce) dla słuchającego, by mógł odtworzyć wyrażenie.

W programie rozwiązaliśmy to w taki sposób, że po funkcji spodziewamy się wyrażenia w nawiasie, nawet jeśli to jest prosta liczba, czyli u nas poprawnym będzie:

"sinus otwórz nawias 45 zamknij nawias"