

Lab 2

Wprowadzenie

Zadanie polega na stworzeniu parsera języka do operacji macierzowych. Parser powinien rozpoznać (akceptować) kod źródłowy w formie tokenów, bądź zgłaszać błąd parsingu w przypadku nieprawidłowego wejścia. Parser powinien rozpoznawać następujące konstrukcje:

- wyrażenia binarne, w tym operacje macierzowe 'element po elemencie'
- wyrażenia relacyjne,
- negację unarną,
- transpozycję macierzy,
- inicjalizację macierzy konkretnymi wartościami,
- macierzowe funkcje specjalne,
- instrukcję przypisania, w tym różne operatory przypisania
- instrukcję warunkową if-else,
- pętle: while and for,
- instrukcje break, continue oraz return,
- instrukcję print,
- instrukcje złożone,
- tablice oraz ich zakresy.

Przykładowo, parser powinien akceptować następujący kod:

```
A = zeros(5); # create 5x5 matrix filled with zeros
D = A.+B' ;    # add element-wise A with transpose of B

for j = 1:10
    print j;
```

Przykładowe poprawne wejścia: [example1.m](#), [example2.m](#), [example3.m](#)

- Do rozwiązania zadania należy użyć generatora parserów PLY.
- Rozpoznawany język powinien być spójny z przykładami z plików `examplen.m`. Wystąpienia białych znaków oraz sposób formatowania tekstu nie powinny wpływać na poprawność kodu.
- Parser powinien rozpoznawać niepoprawny syntaktycznie kod wejściowy. W takim przypadku należy wypisać numer niepoprawnej linii oraz informację o wystąpieniu błędu.
- Przy tworzeniu parsera należy wykorzystać stworzony skaner (`scanner.py`) na poprzednich zajęciach.
- Można wykorzystać poniższe szablony [Mparser.py](#) oraz [main.py](#).