

Lab 4

Wprowadzenie

Zadanie jest kontynuacją poprzedniego zadania.

Zadanie polega na stworzeniu analizatora błędów semantycznych.

Analizator semantyczny powinien wykrywać m.in. następujące błędy semantyczne:

- inicjalizacja macierzy przy użyciu wektorów o różnych rozmiarach
- odwołania poza zakres macierzy (w przypadku indeksów będących stałymi)
- dla danej operacji binarnej użycie stałej, skłara, wektora lub macierzy o niekompatybilnych typach lub rozmiarze, np.
 - dodawanie skłara lub wektora do macierzy
 - operacje binarne na wektorach lub macierzach o niekompatybilnych wymiarach
- użycie funkcji inicjalizujących (funkcje `eye`, `zeros`, `ones`) z niepoprawnymi parametrami
- niepoprawne użycie instrukcji:
 - instrukcje `break` lub `continue` poza pętlą

Analiza błędów semantycznych nie powinna być przerywana po napotkaniu pierwszego błędu, lecz wykrywać jak największą liczbę błędów. Z każdym błędem powinna być skojarzona informacja o miejscu wystąpienia błędu (nr linii, ew. numer kolumny).

Do implementacji zadania należy skorzystać z wzorca *visitor*. W tym celu w klasie `TypeChecker` należy zdefiniować metody `visit_<classname>`, każdej klasie z AST będzie odpowiadała jedna metoda. Omówienie wzorca wizytor można znaleźć w *A. Allen, Modern Compiler Implementation in Java, 2nd ed., Chapt. 4*.

Do stworzenia analizatora semantycznego można wykorzystać pliki:

- [main.py](#)
- [SymbolTable.py](#)
- [TypeChecker.py](#), nie wymaga dodania metody `accept` w klasie `AST.Node`

Przykładowe błędne programy: [control_transfer.m](#), [init.m](#), [opers.m](#)