НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра Обчислювальної Техніки

Лабораторна робота №1

з Програмного забезпечення комп’ютерних систем

Виконав:
Студент 5 курсу, групи ІМ-42мп
БАРДІН Владислав Дмитрович

_____

_____

Перевірив:
Асистент кафедри обчислювальної техніки,
КОБИЛЮК Андрій Григорович

Київ — 2024 року

Мета роботи

Навчитися виконувати лексичний та синтаксичний аналіз заданого арифметичного виразу.

Вхідні дані

Арифметичний вираз з дужками або в бездужковому записі, елементами якого є константи, імена змінних, алгебраїчні операції та математичні функції (по узгодженню з викладачем).

Завдання роботи

Реалізувати лексичний та синтаксичний аналізатор арифметичного виразу з використанням будь-якої мови програмування. Необхідно, щоб аналізатор перевіряв такі типи помилок:

- помилки на початку арифметичного виразу (наприклад, вираз не може починатись із закритої дужки, алгебраїчних операцій * та /);
- помилки, пов’язані з неправильним написанням імен змінних, констант та при необхідності функцій;
- помилки у кінці виразу (наприклад, вираз не може закінчуватись будь-якою алгебраїчною операцією);
- помилки в середині виразу (подвійні операції, відсутність операцій перед або між дужками, операції* або / після відкритої дужки тощо);
- помилки, пов’язані з використанням дужок (нерівна кількість відкритих та закритих дужок, неправильний порядок дужок, пусті дужки).

Синтаксичний аналізатор потрібно реалізувати за допомогою кінцевого автомату. Результатом виконання даної лабораторної роботи є список помилок, виявлених під час синтаксичного аналізу.

Опис алгоритму виконання роботи

Синтаксичний аналіз виконується за допомогою кінцевого автомату. Назви функцій (sin, cos, sqrt) та змінних можуть складатися лише з a-zA-Z літер англійського алфавіту. Функції можуть приймати від 1 до N параметрів вказаних через кому.

Граф автомату

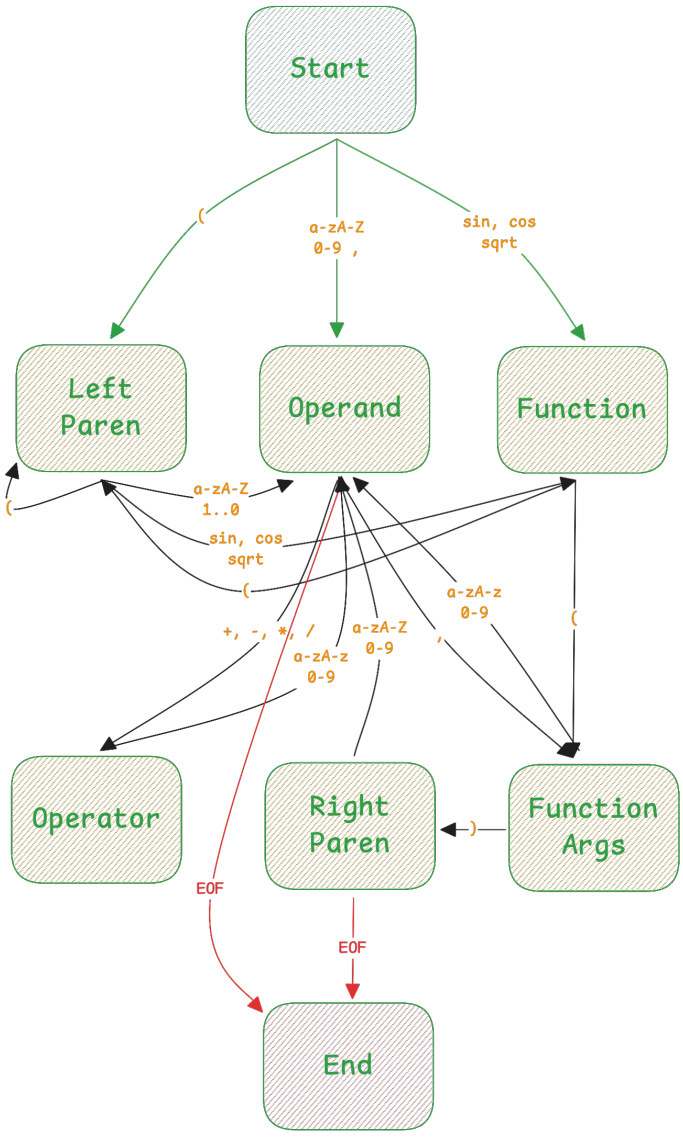

Рис. 1 — граф кінцевого автомату, що використовується для синтаксичного аналізу виразу

Алгоритм обробки помилок базується на використанні графу переходів. Під час аналізу кожен токен виразу обробляється відповідно до поточного стану. Якщо для поточного стану парсера, існує допустимий перехід, то цей символ вважається коректним, і відбувається перехід до нового стану. Якщо ж для символу перехід не передбачений, він вважається неочікуваним, що вказує на наявність помилки у виразі. При помилці, Записується помилка, та розбір виразу зупиняється, через те що подальший пошук помилок не є доцільним через те, що неочікуваний токен може мати суттєвий вплив на подальший вираз. Окремо здійснюється перевірка коректності дужок у виразі (парна кількість, порядок).

Результати виконання

```
"/Users/vbardin/Documents/GitHub/KPI/KPI-Masters/Computer System Software/Source Code/Peat.Cli/bin/Debug/net9.0/Peat.Cli" analyze (A+B)+C/D+G+(K/L+M+N)

🔍 Analyzing expression: "(A+B)+C/D+G+(K/L+M+N)"
═══════════════════════════════════════

✅Result: Valid
⏱ Time: 1,66ms


Process finished with exit code 0.
```

```
"/Users/vbardin/Documents/GitHub/KPI/KPI-Masters/Computer System Software/Source Code/Peat.Cli/bin/Debug/net9.0/Peat.Cli" analyze (AB)+C/D+G+(K/L+M+N)

🔍 Analyzing expression: "(AB)+C/D+G+(K/L+M+N)"
═══════════════════════════════════════

✅Result: Valid
⏱ Time: 1,52ms


Process finished with exit code 0.
```

AB в даному випадку розпізнається як одна змінна, а дужки як такі, що не впливають на коректність виразу.

```
"/Users/vbardin/Documents/GitHub/KPI/KPI-Masters/Computer System Software/Source Code/Peat.Cli/bin/Debug/net9.0/Peat.Cli" analyze "(A -+* B)+C/D+G+(K/L+M+N)"

🔍 Analyzing expression: "(A -+* B)+C/D+G+(K/L+M+N)"
═══════════════════════════════════════

❌Analysis failed:

❌Error:
(A -+* B)+C/D+G+(K/L+M+N)
    ^ Unexpected token: Operator

Process finished with exit code 0.
```

```
"/Users/vbardin/Documents/GitHub/KPI/KPI-Masters/Computer System Software/Source Code/Peat.Cli/bin/Debug/net9.0/Peat.Cli" analyze "(A () B)+C/D+G+(K/L+M+N)"

🔍 Analyzing expression: "(A () B)+C/D+G+(K/L+M+N)"
═══════════════════════════════════════

❌Analysis failed:

❌Error:
(A () B)+C/D+G+(K/L+M+N)
^ Unmatched opening parenthesis

Process finished with exit code 0.
```

```
"/Users/vbardin/Documents/GitHub/KPI/KPI-Masters/Computer System Software/Source Code/Peat.Cli/bin/Debug/net9.0/Peat.Cli" analyze "sin(A+ B - C)+C * cos(1) /    D * 34 +G+(K/L+M+N)"

🔍 Analyzing expression: "sin(A+ B - C)+C * cos(1) /    D * 34 +G+(K/L+M+N)"
═══════════════════════════════════════

✅Result: Valid
⏱ Time: 1,82ms

Process finished with exit code 0.
```

```
"/Users/vbardin/Documents/GitHub/KPI/KPI-Masters/Computer System Software/Source Code/Peat.Cli/bin/Debug/net9.0/Peat.Cli" analyze "sin(A+ B - C)+C * cos(((1))) /    D * 34 +G+(K/L+M+N)"

🔍 Analyzing expression: "sin(A+ B - C)+C * cos(((1))) /    D * 34 +G+(K/L+M+N)"
═══════════════════════════════════════

✅Result: Valid
⏱ Time: 1,63ms

Process finished with exit code 0.
```

Лістинг програмного коду

Код лабораторної роботи розміщено на веб-ресурсі GitHub, і доступний для ознайомлення за посиланням: https://github.com/Bardin08/KPI-Masters/pull/5.