Лабораторна робота № 10

«Стековий парсер»

Теоретичні відомості:

Аналізатори строкових виразів (парсери) займають важливе місце серед алгоритмів сучасного програмування. Розробка синтаксичних та лексичних аналізаторів вважається окремою дисципліною - комп'ютерною лінгвістикою та стрімко розвивається.

Програма, яку ми пишемо мовою Сі, також обробляється досить складними парсерами для верифікації синтаксису та перетворення у зручний для компіляції вигляд.

Основним завданням парсера є як правило перетворення тексту в текст за певними правилами. Процедура розбору спирається на два механізми - лексичний та синтаксичний аналізатори. Завданням лексичного аналізатоа є розбиття строки символів на послідовності, які мають синтаксичне значення. Такі послідовності символів називають "токенами" або "лексемами" або просто словами. Послідовність токенів сприймає синтаксичний аналізатор, який спирається на наперед визначену логіку мови і визначає, коли ряд токенів закінчується і прочитане речення потребує певної реакції - виконання якої-небудь дії, або запису певного тексту у вихідний потік.

Синтаксичний аналізатор приймає рішення про завершення синтаксичної конструкції (речення) на підставі останнього токена (термінатора), але з урахуванням усіх попередніх. Для зберігання попредніх токенів до приходу термінального використовується стек.

Стек - це така послідовна структура даних, у яку можна вміщувати елементи та виймати їх тільки з одного боку. Стек підтримує дві операції - push(x) - вміщує елемент x у вершину стеку, та pop() - повертає черговий елемент з вершини стеку та видаляє його зі стеку.

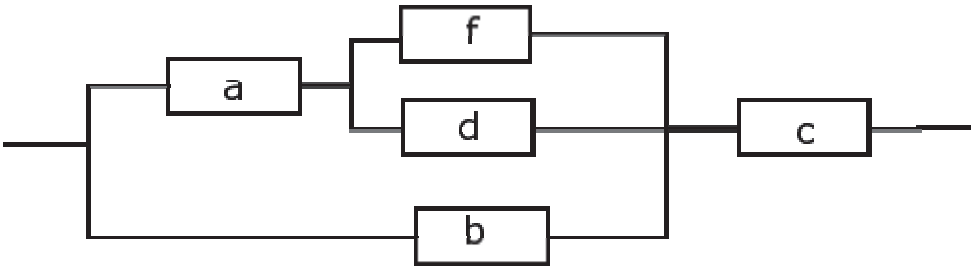
В даній роботі орзглядається функціонал лексичного та синтаксичного аналізу на базі аналізатора виразів. Вирази, наприклад арифметичні, можуть бути подані у інфіксному "a+b", префіксному "+ab" або постфіксному "ab+" записі. Останні дві форми називають «польською» та «оберненою польською нотацією» на честь логіка Яна Лукасевича, який у 1920 році запропонував таку форму запису логічних виразів без дужок. Обернений запис зручніший для синтаксичнго аналізу і найчастіше використовується у парсерах.

При аналізі виразу у оберненій нотації на прикладі наведеного вище, токени, які не є термінаторами, заносяться до стеку аж доки не буде отриманий перший термінатор. У нашому прикладі термінатором є токен "плюс" і він поєднує у сиснатксичній конструкції два попередні.

Парсер, отримавши термінатор, виймає зі стеку два попередні токени, проводить із ними потрібну арифметичну операцію і записує результат назад до стеку. Отриманий результат може бути використаний іншими синтаксичними конструкціями, які обгортають дану. Результат виконання всього виразу може бути вийнятий зі стеку після обробки останнього токена. Якщо на той момент у стеку залишилося більше одного елемента - вираз був невалідний.

Завдання:

Резистори у електронних схемах можуть бути поєднані послідовно (тоді $R_\Sigma = R_1 + R_2$) та паралельно ($R_\Sigma = R_1 R_2 / (R_1 + R_2)$). Таким же чином можна об'єднувати підсхеми, які в свою чергу мають всередині комбінації паралельних та послідовних з'єднань. Існує форма строкового запису, коли паралельне об'єднання записується у квадратних дужках, а послідовне - у круглих. Приклад схеми та її запису показаний на рис. 8.1.



([(a [f d]) b] c)

Рис. 1

Такий запис є оберненим, має 5 видів токенів - "відкрита кругла дужка", "закрита кругла", "відкрита квадратна", "закрита квадратна", та "число" і дві синтаксичні конструкції - "послідовне з'єднання" та "параллельне з'єднання". Причому обидві відкриті дужки є надлишковими з точки зору синтаксичного аналізатора і можуть бути проігноровані, за винятком ситуації коли вони використовуються для перевірки синтаксичної валідності запису. Термінаторами обох синтаксичних конструкцій є закриті дужки.

Завданням роботи є:

- Написати парсер, який приймає, наприклад з клавіатури, строку опису схеми і друкує результат обрахунку сумарного опору, або повідомлення про синтаксичні помилки.
- Числа вважати реальними додатніми у форматі IEEE754, тобто такими, що містять символи [0-9+-.].
- Розділювачами вважати пробіли або табуляції. Розділювачів може бути кілька поспіль.

- Решту символівв вважати синтаксичними помилками.

Структурно програма має містити основну функцію (не main!), яка приймає на вхід строку, а повертає результат, або друкує помилку, функцію лексичного аналізатора, яка працює у режимі "GET NEXT", тобто за кожним викликом повертає наступну лексему строки аж доки строка не закінчиться, та функцію синтаксичного аналізатора, яка оперує стеком.