

ОГЛЯД ЛЕКСИЧНОГО ТА СИНТАКСИЧНОГО АНАЛІЗУ МОВ ПРОГРАМУВАННЯ ДЛЯ ЗАСТОСУВАННЯ У КОМПІЛЯТОРАХ

Вінницький національний технічний університет

Анотація

У даній роботі проаналізовано процеси лексичного та синтаксичного аналізу для застосування у компіляторах.

Ключові слова: Лексичний аналіз, синтаксичний аналіз, лексема, токенизація, компілятор.

Abstract

In this work processes of lexical and syntactic analysis for usage in compilers have been analyzed.

Keywords: Lexical analysis, syntactic analysis, lexeme, tokenization, compiler.

Вступ

Значення слова лексичний в лексичному аналізі походить від слова "лексема". Лексема - це абстрактна одиниця морфологічного аналізу у лінгвістиці. Лексичний аналізатор [1] використовується у різних програмах, наприклад у редакторах тексту та компіляторах. Компілятор перекладає та компілює програму, написану відповідною мовою програмування, на цільову мову, зазвичай машинну, через декілька фаз. Лексичний аналіз це найперша фаза у процесі компіляції при якій відбувається обробка вхідного коду для розбиття на токени (лексеми) [2]. Наступною фазою є синтаксичний аналіз при якому отримані токени перевіряються на валідність та формуються у синтаксичні дерева. Однією з існуючих проблем, які виникають при лексичному та синтаксичному аналізі є їх недостатня формалізація.

Мета роботи полягає у формалізації та аналізі процесів лексичного та синтаксичного аналізу для застосування у компіляторах.

Результати дослідження

Компілятор - це системне програмне забезпечення, яке перетворює програму мови високого рівня програмування в еквівалентну програму на низькорівневій (машинній) мові.

Розглянемо лексичний аналіз:

Лексичний аналіз - це перший етап процесу компіляції. Лексичний аналізатор посимвольно обробляє вхідну програму перетворюючи її у послідовність токенів. Цей процес ще називають токенизацією. Токен – це об'єкт що представляє лексему. Під час токенизації відбувається розбиття коду на токени відповідно до правил вхідної мови, які можуть включати визначення змінних, зарезервованих слів, виразів, операторів, та ін., а також очищення вхідного коду від коментарів та непотрібних символів, обробка директив препроцесора. На цьому етапі обробка помилок відносно проста у реалізації.

Розглянемо синтаксичний аналіз:

Синтаксичний аналіз – це другий етап процесу компіляції. На основі отриманих токенів формується синтаксичне дерево, у якому перевіряється відповідність послідовності токенів до правил вхідної мови. Формуються таблиці літералів (констант вказаних безпосередньо у коді програми) та таблиці символів, що містять ідентифікатори (назви змінних, функцій та ін.). Синтаксичне дерево методом семантичного аналізу перетворюється у анотоване синтаксичне дерево яке додатково містить

значення атрибутів вузлів. Синтаксичне дерево формує проміжне представлення структури коду, та містить посилання на таблицю символів. З анованого дерева генерується проміжний код, який може бути використаний у віртуальній машині, чи псевдоасемблері. Під час цього етапу може бути застосований набір оптимізацій, наприклад складання констант, чи виразів [1]. Етапи лексичного та синтаксичного аналізу представлені графічно на рисунку 1 [3].

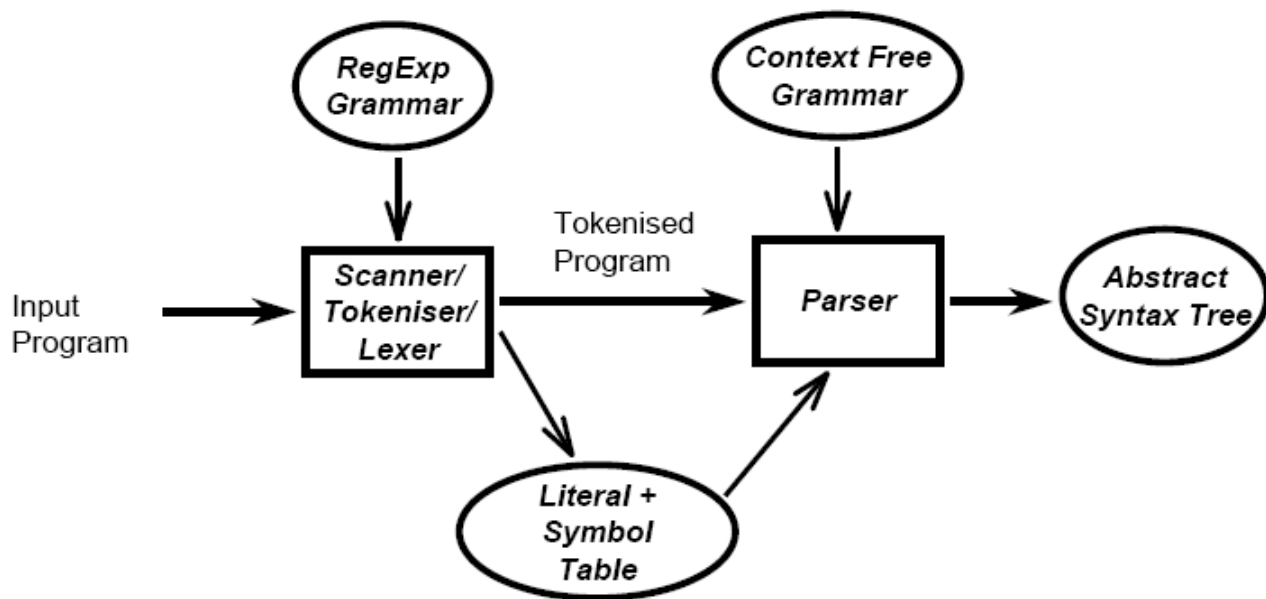


Рисунок 1 – Лексичний та синтаксичний аналіз

Під час останнього етапу з проміжного коду генерується цільовий код, який може бути безпосередньо машинним кодом або асемблерним кодом. Асемблерний код далі має передаватися відповідному асемблеру. До цільового коду, в свою чергу, може бути застосована подальша оптимізація.

Висновки

У роботі формалізовано та проаналізовано процеси лексичного та синтаксичного аналізу, які застосовуються для перетворення вхідного коду у проміжний код.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Compilers, principles, techniques, and tools / Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman - Pearson Education, 2007. – 993 p.
2. An Exploration on Lexical Analysis / Farhanaaz, Sanju. V // Proceedings on International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). – Chennai, India, 2016. – DOI: 10.1109/ICEEOT.2016.7755127. – P. 253-258.
3. Implementation of Lexical Analysis / Rupeder Chuhan, Vishendera Singh, Kapil Makhan, Mukesh Kumar // International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2017. – ISSN: 2321-9653. – P. 614-617.

Кулик Максим Сергійович — студент групи ІСІ-166, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м. Вінниця, e-mail: kulik.maksim.sergeevich@gmail.com

Севастьянов Володимир Миколайович — канд. техн. наук, доцент кафедри метрології та промислової автоматики, Вінницький національний технічний університет, м. Вінниця, e-mail: sevastyanov.vladimir@vntu.edu.ua

Бевз Олександр Миколайович — канд. техн. наук, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, e-mail: rebrest@i.ua

Науковий керівник: **Севастьянов Володимир Миколайович**— канд. техн. наук, доцент кафедри метрології та промислової автоматики, Вінницький національний технічний університет, м. Вінниця.

Kulyk Maksim S. — student, Faculty of Computer Control Systems and Automatics, Vinnytsia National Technical University, Vinnytsia, e-mail: kulik.maksim.sergeevich@gmail.com

Sevastyanov Volodimir M. — Cand. Sc. (Eng), Associate Professor, Faculty of Metrology and Industrial Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: sevastyanov.vladimir@vntu.edu.ua

Bevz Oleksandr M. — Cand. Sc. (Eng), Associate Professor, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: rebrest@i.ua

Supervisor: **Sevastyanov Volodimir M.** — Cand. Sc. (Eng), Associate Professor, Faculty of Metrology and Industrial Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: sevastyanov.vladimir@vntu.edu.ua