



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2

Дисциплина Конструирование компиляторов

Тема Преобразования грамматик

Вариант №2

Студент Брянская Е.В.

Группа ИУ7-21М

Преподаватель Ступников А.А.

Москва.
2023 г.

Цель работы: приобретение практических навыков реализации наиболее важных (но не всех) видов преобразований грамматик, чтобы удовлетворить требованиям алгоритмов синтаксического разбора.

Задание

Постройте программу, которая в качестве входа принимает приведенную КС-грамматику $G = (N, \Sigma, P, S)$ и преобразует ее в эквивалентную КС-грамматику G' без левой рекурсии.

Вариант 2. Устранение бесполезных символов.

Назовем символ $X \in N \cup \Sigma$ бесполезным в КС-грамматике $G = (N, \Sigma, P, S)$, если в ней нет вывода вида $S \Rightarrow^* wXy \Rightarrow^* wxu$, где w, x, y принадлежат Σ^* .

Чтобы установить, бесполезен ли нетерминал A , надо построить сначала алгоритм, выясняющий, может ли нетерминал порождать какие-нибудь терминальные цепочки, т. е. решающий проблему пустоты множества $\{w \mid A \Rightarrow^* w, w \in \Sigma^*\}$.

Постройте программу, которая в качестве входа принимает произвольную КС-грамматику $G = (N, \Sigma, P, S)$ и преобразует ее в эквивалентную КС-грамматику $G' = (N', \Sigma', P', S')$, не содержащую бесполезных символов.

Контрольные вопросы

1. *Как может быть определён формальный язык?*
 - a. Простым перечислением слов, входящих в данный язык.
 - b. Словами, порождёнными некоторой формальной грамматикой
 - c. Словами, порождёнными регулярным выражением.
 - d. Словами, распознаваемыми некоторым конечным автоматом
2. *Какими характеристиками определяется грамматика?*
 - a. Σ – множество терминальных символов
 - b. N – множество нетерминальных символов
 - c. P – множество правил (слева – непустая последовательность терминалов/нетерминалов, содержащая хотя бы один нетерминал, справа – любая последовательность терминалов/нетерминалов)
 - d. S – начальный символ из множества нетерминалов
3. *Дайте описания грамматик по иерархии Хомского.*
 - a. Регулярные – контекстно-свободные, правила имеют вид $A \rightarrow B\gamma$ (то есть левая часть только один нетерминал, справа – цепочка терминалов, может быть в конце/начале только один нетерминал)
 - b. Контекстно-свободные – правая часть любая

- с. Контекстно-зависимые – в левой/правой часть могут быть цепочки, присутствующие в обеих частях, нужны для того, чтобы сработало правило
 - d. Неограниченные – всё, что угодно может находиться в левой/правой частях.
4. *Какие абстрактные устройства используются для разбора грамматик?*
- a. Распознающие грамматики – устройства (алгоритмы), которым на вход подается цепочка языка, а на выходе устройство печатает «Да», если цепочка принадлежит языку, и «Нет» — иначе.
 - b. Порождающие грамматики - вид устройств, использующийся для порождения цепочек языков по требованию.
5. *Оцените временную и емкостную сложность предложенного вам алгоритма*
- a. $O(N_{\text{нт}}^2 \cdot n_{\text{правил}})$ – временная сложность
 - b. $O(N_{\text{нт}} + n_{\text{т}})$ – ёмкостная сложность

Результаты

```

Controller -- Init grammar
Name: G0
Nonterminal symbols: S, A
Terminal symbols: a, b, c, d, +, *, (, )
Rules:
A -> Ac | Sd
S -> Aa | b
-----
Controller -- After deletion recursion
Name: G0
Nonterminal symbols: S, A
Terminal symbols: a, b, c, d, +, *, (, )
Rules:
A -> bdA'
S -> Aa | b
A' -> cA' | adA' | ε
-----
Controller -- ne: A, S, A'
Controller -- After update
Name: G0
Nonterminal symbols: S, A
Terminal symbols: a, b, c, d, +, *, (, )
Rules:
A -> bdA'
S -> Aa | b
A' -> cA' | adA' | ε
-----
Controller -- After deletion unreachable symbols
Name: G0
Nonterminal symbols: S, A
Terminal symbols: a, b, c, d
Rules:
A -> bdA'
S -> Aa | b
A' -> cA' | adA' | ε
-----

Init grammar
Name: G0
Start symbol: S
Nonterminal symbols: S, A, B
Terminal symbols: a, b, c, d, +, *, (, )
Rules:
A -> AB
B -> b
S -> a | A
-----
ne: B, S
After update
Name: G0
Start symbol: S
Nonterminal symbols: S, B
Terminal symbols: a, b, c, d, +, *, (, )
Rules:
B -> b
S -> a
-----
After deletion unreachable symbols
Name: G0
Start symbol: S
Nonterminal symbols: S
Terminal symbols: a
Rules:
S -> a

```

Выводы

Были изучены и реализованы такие алгоритмы как: алгоритм устранения левой рекурсии и бесполезных символов, включающий в себя проверку языка на непустоту и устранение недостижимых символов.

Список литературы

1. АХО А., УЛЬМАН Дж. Теория синтаксического анализа, перевода и компиляции: В 2-х томах. Т.1.: Синтаксический анализ. - М.: Мир, 1978.

2. АХО А.В, ЛАМ М.С., СЕТИ Р., УЛЬМАН Дж.Д. Компиляторы: принципы, технологии и инструменты. – М.: Вильямс, 2008