Лабораторная работа № 3
ПОСТРОЕНИЕ РЕШАЮЩИХ ТАБЛИЦ

Цель работы: Получить навыки в разработке нисходящего анализатора в классе LL(k)-грамматик.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Имея грамматику входного языка, необходимо выполнить ряд формальных преобразований над этой грамматикой, облегчающих построение распознавателя. После этого надо проверить относится ли полученная грамматика к одному из известных классов КС-языков, для которых существуют линейные распознаватели. Часто одна и та же КСГ может быть отнесена не к одному, а сразу к нескольким классам, допускающим построение линейных распознавателей. Тогда необходимо решить какой из нескольких возможных распознавателей выбрать для практической реализации. Желание использовать более простой класс грамматик для построения распознавателя может потребовать каких-то манипуляций с заданной грамматикой, необходимых для её преобразования к требуемому классу. Всё усложняется тем, что проблема преобразования произвольной КСГ к требуемому классу алгоритмически неразрешима, но это не значит, что задача преобразования к нужному классу не решается в каждом конкретной случае. Зачастую удаётся найти такие преобразования, но начиная преобразования гарантировать результат невозможно.

Наша задача в ходе этой лабораторной работы – построение грамматики, на базе которой можно реализовать нисходящий синтаксический анализатор.

Если грамматика является LL(k)-грамматикой, то просмотр вперед от текущего элемента не более k символов входной строки позволяет однозначно определить направление перехода и избежать возвратов. Любая LL(k)-грамматика однозначна.

Леворекурсивная грамматика не принадлежит классу LL(k) ни для какого k. Иногда удаётся преобразовать КСГ в эквивалентную ей LL(k)-грамматику с помощью устранения левой рекурсии и факторизации. Однако проблема существования эквивалентной LL(k)-грамматики для произвольной КСГ неразрешима.

Правило называется леворекурсивным, если оно имеет вид $U \rightarrow Ux$, где $U \in N$, а $x \in (N \cup T)^{+}$. От левой рекурсии всегда можно избавиться, заменив ее на правую рекурсию.

Пусть A – леворекурсивный элемент, a_i – цепочки, которые следую за леворекурсивным элементом A, и ни одна из цепочек b_i не начинается с. A. Тогда, выполним следующую замену:

$$A \rightarrow Aa_1|Aa_2|\dots|Aa_n|b_1|\dots|b_m \quad \Rightarrow \quad \begin{cases} A \rightarrow b_1|\dots|b_m|b_1U|\dots|b_mU \\ U \rightarrow a_1|\dots|a_n|a_1U|\dots|a_nU \end{cases},$$

введя в эквивалентную грамматику вместо правил, выводимых из элемента A, на котором есть левая рекурсия, правила в которых левая рекурсия заменена на правую.

Левая факторизация

Основная идея левой факторизации в том, что, когда неясно, какую из двух альтернатив надо использовать при выводе цепочки из нетерминала A, нужно переделать A-правила так, чтобы отложить решение до тех пор, пока не будет достаточно информации, для принятия правильного решения.

Среди всех правил, выводимых из нетерминала A ищем самый длинный префикс α, общий для двух или боле его альтернатив. Если нетривиальный общий префикс существует, заменяем все A-правила:

$$A \rightarrow \alpha b_1|\alpha b_2|\dots|\alpha b_n|\delta \quad \Rightarrow \quad \begin{cases} A \rightarrow \alpha U|\delta \\ U \rightarrow b_1|b_2|\dots|b_n \end{cases}$$

где U – новый нетерминал, а δ – все альтернативы, не начинающиеся с префикса α.

Повторно применяем это преобразование, пока никакие две альтернативы не будут иметь общего префикса.

Нисходящий анализатор

Алгоритм нисходящего разбора строит синтаксическое дерево, начиная с корня, постепенно спускаясь до уровня предложения. Реализация этого алгоритма усложняется главным образом из-за вспомогательных операций, которые необходимы для того, чтобы выполнять возвраты с твердой уверенностью, что все возможные попытки построения дерева были предприняты.

Алгоритм нисходящего разбора можно улучшить, если найти метод, по которому на каждом шаге алгоритма можно было бы однозначно выбрать одну из всего множества альтернатив. В этом случае возвраты бы не требовались и невозможность выбора рассматривалась как ошибка. Наиболее очевидным методом выбора является выбор альтернативы на основе анализа не просмотренной части входной цепочки.

Введем два множества FIRST и FOLLOW.

$$FIRST_k(\alpha) = \left\{ \begin{matrix} x \mid x \in T^*, \alpha \overset{*}{\Rightarrow}_l x\beta, & |x| = k, & \beta \in (N \cup T)^* \\ x \mid x \in T^*, \alpha \overset{*}{\Rightarrow}_l x, & |x| < k & \end{matrix} \right\}$$

где $\alpha \in (T \cup N)^*$, $A \in N$, $k > 0$.

Это множество терминальных цепочек, выводимых из α, укороченных до k символов.

$$FOLLOW_k(A) = \left\{ x \;\middle|\; x \in T^*, S \overset{*}{\Rightarrow} yA\beta, x \in FIRST_k(\beta), y \in T^* \right\}$$

где $\alpha \in (T \cup N)^*$, $A \in N$, $k > 0$.

Доказано, что грамматика *G(N,T,P,S)* является LL(k) тогда и только тогда, когда для любых правил из P вида $A \to b$ и $A \to y$, где $a \neq b$ выполнено $FIRST_k(b\varpi) \cap FIRST_k(y\varpi) = \emptyset$, для всех цепочек.

Пример. Проверить, является ли приведённая грамматика LL(k), если нет, выполнить необходимые преобразования и проверить класс получившейся после преобразований грамматики.

Задана грамматика **G**($\{E, T\}$, $\{a, +\}$, P, E),

где $P = \begin{cases} E \rightarrow T|E+T \\ T \rightarrow a \end{cases}$

Грамматика **G** с приведенными правилами не является LL(k), так как содержит левую рекурсию в правиле E.

Устраним левую рекурсию:

$$E \rightarrow T|E+T \quad \Rightarrow \quad \begin{cases} E \rightarrow T|TU \\ U \rightarrow +T|+TU \end{cases}$$

Для каждой пары полученных правил выполним левую факторизацию:

$$E \rightarrow T|TU \quad \Rightarrow \quad \begin{cases} E \rightarrow TX \\ X \rightarrow \varepsilon|U \end{cases}$$

$$U \rightarrow +T|+TU \quad \Rightarrow \quad \begin{cases} U \rightarrow +TX \\ X \rightarrow \varepsilon|U \end{cases}$$

Так как правые части второго и третьего правил совпадают, для их обозначения в обоих случаях выбран одинаковый нетерминал X.

В результате получили грамматику, не содержащую леворекурсивных правил:

$$P` = \begin{cases} E \rightarrow TX \\ X \rightarrow \varepsilon|U \\ U \rightarrow +TX \\ T \rightarrow a \end{cases}$$

Докажем, что полученная после преобразований грамматика является LL(k): построим присоединённую грамматику, заполним таблицу правилами и вычислим первые терминалы, выводимые из каждого нетерминала для каждого правила грамматики, т.е. множество $\text{FIRST}_k(A)$, в случае если нужная длина цепочки получена и множество $\text{FIRST}_k(A) \cup \text{FOLLOW}_m(A)$, с таким значением m, чтобы полученная после конкатенации цепочка была не короче заданного значения k.

Правила грамматики	$FIRST_1(A) \cup FOLLOW_1(A)$
$E \to TX$	a
$T \to a$	a
$X \to \varepsilon$ $X \to U$	\$ +
$U \to +TX$	+

Так как выбрать нужное правило из двух альтернативных выводимых из X можно на основе анализа одного элемента входной строки, то грамматика является LL(1).

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Проверить принадлежность разработанной грамматики к классу LL(k).
2. Выполнить необходимые преобразования КСГ.
3. Построить решающую таблицу нисходящего анализатора
4. Доказать, что полученная после преобразований грамматика принадлежит классу LL(k) и определить k.

Содержание отчета

- Титульный лист
- Исходная грамматика языка
- Текст задания
- Описание принадлежности к классу LL(k)
- При необходимости, описание процесса проведения преобразований: устранение левой рекурсии и левой факторизации
- Полученная после преобразований грамматика
- Решающая таблица нисходящего анализатора
- Доказательство принадлежности полученной грамматики к классу LL(k)

Вопросы для самоконтроля

- Назначение синтаксического анализатора
- Принципы нисходящего анализа программы
- Условие принадлежности грамматики к классу LL(k)
- Типы грамматик по классификации Хомского
- Алгоритм устранения леворекурсивных правил
- Алгоритм устранения левой факторизации.
- Понятие множеств $FIRST_k$ и $FOLLOW_k$
- Назначение решающей таблицы нисходящего анализа

Список литературы

1. Шульга, Т. Э. Теория автоматов и формальных языков : учебное пособие / Т. Э. Шульга. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2015. — 104 c. — ISBN 987-5-7433-2968-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/76519.html (дата обращения: 15.04.2021). — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/76519 - https://www.iprbookshop.ru/76519.html

2. Алымова, Е. В. Конечные автоматы и формальные языки : учебник / Е. В. Алымова, В. М. Деундяк, А. М. Пеленицын. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2018. — 292 c. — ISBN 978-5-9275-2397-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/87427.html (дата обращения: 15.04.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/87427.html

3. Пентус, А. Е. Математическая теория формальных языков : учебное пособие / А. Е. Пентус, М. Р. Пентус. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 218 c. — ISBN 978-5-4497-0662-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/97548.html (дата обращения: 15.04.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/97548.html

4. Миронов, С. В. Формальные языки и грамматики : учебное пособие для студентов факультета компьютерных наук и информационных технологий / С. В. Миронов. — Саратов : Издательство Саратовского университета, 2019. — 80 c. — ISBN 978-5-292-04613-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/99047.html (дата обращения: 15.04.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/99047.html

5. Малявко, А. А. Формальные языки и компиляторы : учебник / А. А. Малявко. — Новосибирск : Новосибирский государственный технический университет, 2014. — 431 c. — ISBN 978-5-7782-2318-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/47725.html (дата обращения: 15.04.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/47725.html