ТЕМА: РАЗРАБОТКА ЛЕКСИЧЕСКОГО АНАЛИЗАТОРА

Цель курсовой работы: разработать диаграмму состояний и лексический анализатор регулярной грамматики, исследовать работу лексического анализатора.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

В основе лексических анализаторов лежат регулярные грамматики, поэтому рассмотрим грамматики этого класса более подробно.

Под регулярной грамматикой будем понимать леволинейную грамматику.

Напомним, что грамматика G = (V, W, R, I) называется леволинейной, если каждое правило из R имеет вид $A \to Bt$ либо $A \to t$, где $A \in W$, $B \in W$, $t \in V$.

Предположим, что анализируемая цепочка заканчивается специальным символом $\perp$ - признаком конца цепочки.

Для грамматик этого типа существует алгоритм определения того, принадлежит ли анализируемая цепочка языку, порождаемому этой грамматикой (алгоритм разбора):

(1) первый символ исходной цепочки $a_1a_2...a_n \perp$ заменяем нетерминалом A, для которого в грамматике есть правило вывода $A \to a_1$ (другими словами, производим "свертку" терминала a_1 к нетерминалу A)

(2) затем многократно (до тех пор, пока не считаем признак конца цепочки) выполняем следующие шаги: полученный на предыдущем шаге нетерминал A и расположенный непосредственно справа от него очередной терминал a_i исходной цепочки заменяем нетерминалом B, для которого в грамматике есть правило вывода $B \to Aa_i$ (i = 2, 3,.., n);

Это эквивалентно построению дерева разбора методом "снизу-вверх": на каждом шаге алгоритма строим один из уровней в дереве разбора, "поднимаясь" от листьев к корню.

При работе этого алгоритма возможны следующие ситуации:

(1) прочитана вся цепочка; на каждом шаге находилась единственная нужная "свертка"; на последнем шаге свертка произошла к символу I. Это означает, что исходная цепочка $a_1a_2...a_n\perp \in L(G)$.

(2) прочитана вся цепочка; на каждом шаге находилась единственная нужная "свертка"; на последнем шаге свертка произошла к символу, отличному от I. Это означает, что исходная цепочка $a_1a_2...a_n\perp \notin L(G)$.

(3) на некотором шаге не нашлось нужной свертки, т.е. для полученного на предыдущем шаге нетерминала A и расположенного непосредственно справа от него очередного терминала a_i исходной цепочки не нашлось нетерминала B, для которого в грамматике было

бы правило вывода $B \to Aa_i$. Это означает, что исходная цепочка $a_1a_2...a_n\perp \notin L(G)$.

(4) на некотором шаге работы алгоритма оказалось, что есть более одной подходящей свертки, т.е. в грамматике разные нетерминалы имеют правила вывода с одинаковыми правыми частями, и поэтому непонятно, к какому из них производить свертку. Это говорит о недетерминированности разбора. Анализ этой ситуации будет дан ниже.

Допустим, что разбор на каждом шаге детерминированный.

Для того, чтобы быстрее находить правило с подходящей правой частью, зафиксируем все возможные свертки (это определяется только грамматикой и не зависит от вида анализируемой цепочки).

Это можно сделать в виде таблицы, строки которой помечены нетерминальными символами грамматики, столбцы - терминальными. Значение каждого элемента таблицы - это нетерминальный символ, к которому можно свернуть пару "нетерминал-терминал", которыми помечены соответствующие строка и столбец.

Например, для грамматики G = ({a, b, ⊥}, {I, A, B, C}, R, I), такая таблица будет выглядеть следующим образом:

R: $S \to C\perp$
$C \to Ab \mid Ba$
$A \to a \mid Ca$
$B \to b \mid Cb$

	a	b	⊥
C	A	B	S
A	-	C	-
B	C	-	-
I	-	-	-

Знак "-" ставится в том случае, если для пары "терминал-нетерминал" свертки нет.

Но чаще информацию о возможных свертках представляют в виде диаграммы состояний (ДС) - неупорядоченного ориентированного помеченного графа, который строится следующим образом:

(1) строят вершины графа, помеченные нетерминалами грамматики (для каждого нетерминала - одну вершину), и еще одну вершину, помеченную символом, отличным от нетерминальных (например, H). Эти вершины будем называть *состояниями***.** H - начальное состояние.

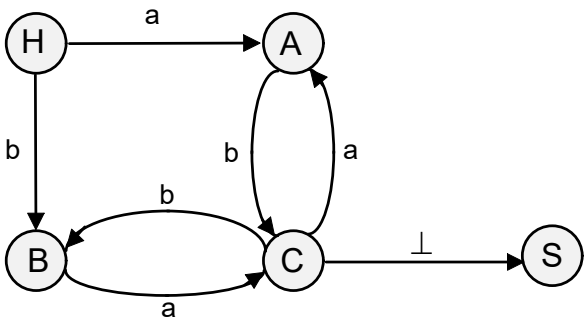
(2) соединяем эти состояния дугами по следующим правилам:

a) для каждого правила грамматики вида $W \to t$ соединяем дугой состояния H и W (от H к W) и помечаем дугу символом t;

b) для каждого правила $W \to Vt$ соединяем дугой состояния V и W (от V к W) и помечаем дугу символом t;

Диаграмма состояний для грамматики G (см. пример выше):

Замечание: на диаграмме состояний конечное состояние помечено вершиной S.



Алгоритм разбора по диаграмме состояний:

(1) объявляем текущим состояние H;

(2) затем многократно (до тех пор, пока не считаем признак конца цепочки) выполняем следующие шаги: считываем очередной символ исходной цепочки и переходим из текущего состояния в другое состояние по дуге, помеченной этим символом. Состояние, в которое мы при этом попадаем, становится текущим.

При работе этого алгоритма возможны следующие ситуации (аналогичные ситуациям, которые возникают при разборе непосредственно по регулярной грамматике):

(1) прочитана вся цепочка; на каждом шаге находилась единственная дуга, помеченная очередным символом анализируемой цепочки; в результате последнего перехода оказались в состоянии I. Это означает, что исходная цепочка принадлежит L(G).

(2) прочитана вся цепочка; на каждом шаге находилась единственная "нужная" дуга; в результате последнего шага оказались в состоянии, отличном от I. Это означает, что исходная цепочка не принадлежит L(G).

(3) на некотором шаге не нашлось дуги, выходящей из текущего состояния и помеченной очередным анализируемым символом. Это означает, что исходная цепочка не принадлежит L(G).

(4) на некотором шаге работы алгоритма оказалось, что есть несколько дуг, выходящих из текущего состояния, помеченных очередным анализируемым символом, но ведущих в разные состояния. Это говорит о недетерминированности разбора. Анализ этой ситуации будет приведен ниже.

Диаграмма состояний определяет конечный автомат, построенный по регулярной грамматике, который допускает множество цепочек, составляющих язык, определяемый этой грамматикой. Состояния и дуги ДС - это графическое изображение функции переходов конечного автомата из состояния в состояние при условии, что очередной анализируемый символ совпадает с символом-меткой дуги. Среди всех состояний выделяется начальное (считается, что в начальный момент своей работы автомат находится в этом состоянии) и конечное (если автомат завершает работу переходом в это состояние, то анализируемая цепочка им допускается).

Для более удобной работы с диаграммами состояний введем несколько соглашений:

a) если из одного состояния в другое выходит несколько дуг, помеченных разными символами, то будем изображать одну дугу, помеченную всеми этими символами;
b) непомеченная дуга будет соответствовать переходу при любом символе, кроме тех, которыми помечены другие дуги, выходящие из этого состояния.
c) введем состояние ошибки (ER); переход в это состояние будет означать, что исходная цепочка языку не принадлежит.

По диаграмме состояний легко написать анализатор для регулярной грамматики.

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

1. Разработать диаграмму состояний регулярной грамматики.
2. Разработать лексический анализатор регулярной грамматики.
3. Исследовать работу лексического анализатора.

Регулярная грамматика определяется по варианту, выданному преподавателю.

ВАРИАНТЫ ЗАДАНИЙ

1. $G = (\{a, b, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to C\perp$
$C \to Ab \mid Ba$
$A \to a \mid Ca$
$B \to b \mid Cb$
2. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Ab \mid C\perp$
$A \to b \mid Bc$
$B \to a \mid Aa$
$C \to c \mid Cb$
3. $G = (\{a, b, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Ba \mid C\perp$
$A \to a \mid Bb$
$B \to b \mid Ca$
$C \to Aa$
4. $G = (\{a, b, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $:I \to A\perp$
$A \to Ab \mid Ba$
$C \to a \mid Ca$
$B \to b \mid Cb$

5. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Ab \mid C\perp$
$A \to b \mid Bc \mid Ba$
$B \to a \mid Aa$
$C \to c \mid Cb$

6. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Ab \mid B\perp$
$A \to b \mid Ba$
$B \to a \mid Aa$
$C \to c \mid Cb$

7. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Bb \mid C\perp$
$A \to b \mid Ba \mid Bb$
$B \to a \mid Aa$
$C \to c \mid Cb$

8. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to A\perp$
$A \to c \mid Bc \mid Ac$
$B \to a \mid Aa$
$C \to b \mid Cb$

9. $G = (\{a, b, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Ba \mid B\perp$
$A \to a \mid Bb$
$B \to b \mid Ca$
$C \to Aa$

10. $G = (\{a, b, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to B\perp$
$A \to b \mid Ba$
$B \to a \mid Ca$
$C \to Ab$

11. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Bb \mid C\perp$
$A \to b \mid Bc$
$B \to a \mid Aa$
$C \to c \mid Cb$

12. $G = (\{a, b, c, \perp\}, \{I, A, B\}, R, I)$,
где R: $I \to AB\perp$
$A \to a \mid cA$
$B \to bA$

13. $G = (\{a, b, c, \perp\}, \{I, A, B, C\}, R, I)$,
где R: $I \to Ab \mid C\perp$
$A \to b \mid Bb$

$B \rightarrow a \mid Aa$
$C \rightarrow c \mid Ca$