Системы программирования.

лекции 28.03.2010

Регулярные грамматики.
Разбор по регулярным грамматикам.
Лексический анализ.

Классификация формальных грамматик (повт.)

G1:
S → 0A | A1 | 1
A → 0A

G2:
S → 0A1 | 1B
00A → 0A
B → ε

G3:
S → 0A | 1A |1
A → 0A

G4:
S → 0A1 | 1
0A → 00A1

Этапы трансляции

- лексический анализ
- синтаксический анализ
- семантический анализ
- генерация внутреннего представления программы
- оптимизация
- генерация объектной программы

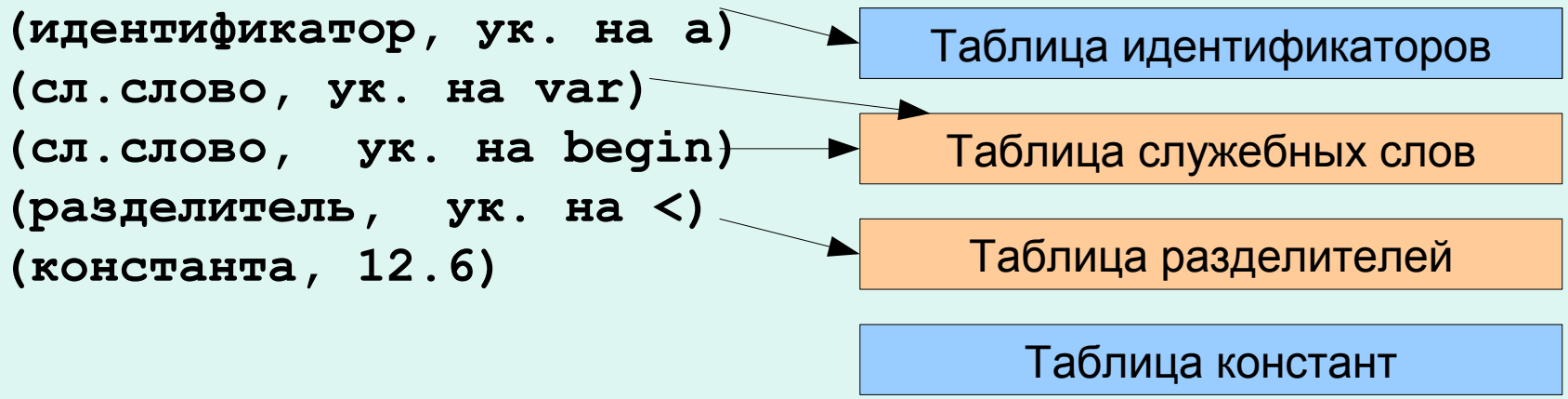
Лексический анализ

последовательность литер → последовательность лексем

Пример : лексемы языка программирования Паскаль

```
a, var, begin, <, 12.6
```

Типы лексем: идентификаторы,
служебные слова,
константы,
разделители (ограничители)



Задачи, решаемые лексическим анализатором

1. Выделить в программе лексемы (это сделает представление программы более удобным для дальнейшей обработки)

2. Удалить из программы комментарии и несущественные пробелы

Если будет изменена кодировка в исходном представлении программы, то это отразится только на лексическом анализаторе

Какие ошибки может обнаружить лексический анализатор?

Лексемы описываются с помощью регулярных грамматик (автоматных, без ε)

Пример: целое без знака

N → N0 | N1 |N2 |N3 |N4 |N5 |N6 |N7 |N8 |N9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Разбор по регулярным грамматикам

Задача разбора*:*

Даны грамматика ***G (***в общем случае, КС***)***
и цепочка ***x.***
Вопрос*: x∈L(G) ?*
Если да, то построить дерево разбора для ***x***
(или левый вывод для ***x,*** или правый вывод для ***x).***

Соглашение: анализируемая цепочка оканчивается символом ⊥

Пример

Грамматика:

S → A⊥

A → A0 | A1 | 0 | 1

Цепочки: **011⊥** **12⊥**

Детерминированный конечный автомат

это пятерка $\langle K, T, \delta, H, S \rangle$, где

K — конечное множество состояний;

T — конечное множество допустимых входных символов;

δ — отображение множества $K \times T$ в K,
определяющее поведение автомата (функция переходов);

$H \in K$ — начальное состояние;

$S \in K$ — заключительное состояние

Заключительных состояний в ДКА может быть более одного, но для любого регулярного языка, все цепочки которого заканчиваются $\perp$, существует ДКА с единственным заключительным состоянием

ДКА *допускает цепочку* $a1a2...an$, если

$\delta(H, a_1) = A_1; \delta(A_1, a_2) = A_2; \ldots; \delta(A_{n-2}, a_{n-1}) = A_{n-1}; \delta(A_{n-1}, a_n) = S,$

где $a_i \in T, A_j \in K, j = 1, 2, ..., n-1; i = 1, 2, ..., n$; H - начальное состояние,

S — заключительное состояние.

Множество цепочек, допускаемых ДКА, составляет определяемый им *язык*

Недетерминированный разбор

Пример

грамматика	цепочка
S → A ⊥	bbb ⊥
A → a | Bb	
B → b | Bb	

Недетерминированный конечный автомат (НКА) —
это пятерка $\langle K, T, \delta, H, S \rangle$, где
K — конечное множество состояний;
T — конечное множество допустимых входных символов;
δ — отображение множества $K \times T$ в множество подмножеств K;
$H \subset K$ — конечное множество начальных состояний;
$S \subset K$ — конечное множество заключительных состояний.

$\delta(A, t) = \{B1, B2, ..., Bn\}$ означает, что из состояния A
по входному символу t можно осуществить переход
в любое из состояний Bi, $i = 1, 2, ..., n$.

Недетерминированный разбор (прод.)

Успешный путь — путь из начальной вершины в заключительную

Язык, допускаемый НКА, - это множество пометок успешных путей

Класс языков, определяемых НКА, совпадает с классом языков, определяемых с помощью ДКА.

Алгоритм построения ДКА по НКА

Недетерминированный разбор (прод.)

Успешный путь — путь из начальной вершины в заключительную

Язык, допускаемый НКА, - это множество пометок успешных путей

Класс языков, определяемых НКА, совпадает с классом языков, определяемых с помощью ДКА.

Алгоритм построения ДКА по НКА

Лексический анализатор для модельного языка

Описание языка

P → program D1; B⊥
D1 → var D {,D}
D → I {,I}: [int | bool]
B → begin S {;S} end
S → I := E | if E then S else S | while E do S | B | read (I) | write (E)
E → E1 [= | < | > | <= | >= | !=] E1 | E1
E1 → T {[+ | - | *or*] T}
T → F {[* | / | *and*] F}
F → I | N | L | *not* F | (E)
L → true | false
I → a | b| ...| z | Ia | Ib |...| Iz | I0 | I1 |...| I9
N → 0 | 1 |...| 9 | N0 | N1 |...| N9

Диаграмма состояний

Свойство регулярных языков

Лемма о разрастании

Задача

Доказать, что язык $L=\{1^n0^n \mid n>0\}$ не является регулярным