
Лекция 03.
Проблема грамматического разбора.
Распознаватели.



Разбор цепочек. Задача разбора

- **Задача разбора в общем виде:** на основе имеющейся грамматики некоторого языка **построить распознаватель** для этого языка
- Цепочка принадлежит языку, порождаемому грамматикой, только в том случае, если существует ее **вывод** из аксиомы этой грамматики.
- Процесс построения такого вывода (а, следовательно, и определения принадлежности цепочки языку) называется **разбором**.



Выводы в грамматике

□ **Определение:**

вывод цепочки $b \in (\mathbf{VT})^*$ из $S \in \mathbf{VN}$ в КС-грамматике $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$, называется *левым (левосторонним)*, если в этом выводе каждая очередная сентенциальная форма получается из предыдущей заменой самого левого нетерминала.

□ **Определение:**

вывод цепочки $b \in (\mathbf{VT})^*$ из $S \in \mathbf{VN}$ в КС-грамматике $G = (\mathbf{VT}, \mathbf{VN}, \mathbf{P}, S)$, называется *правым (правосторонним)*, если в этом выводе каждая очередная сентенциальная форма получается из предыдущей заменой самого правого нетерминала.



Пример. Выводы

- Для цепочки $a+b+a$ в грамматике $G = (\{a,b\}, \{S,T\}, \{S \rightarrow T \mid T+S; T \rightarrow a|b\}, S)$ можно построить выводы:
- Левый
 $S \rightarrow T+S \rightarrow T+T+S \rightarrow T+T+T \rightarrow a+T+T \rightarrow a+b+T \rightarrow a+b+a$
- Правый
 $S \rightarrow T+S \rightarrow a+S \rightarrow a+T+S \rightarrow a+b+S \rightarrow a+b+T \rightarrow a+b+a$
- Смешанный - не левый, не правый
 $S \rightarrow T+S \rightarrow T+T+S \rightarrow T+T+T \rightarrow T+T+a \rightarrow T+b+a \rightarrow a+b+a$

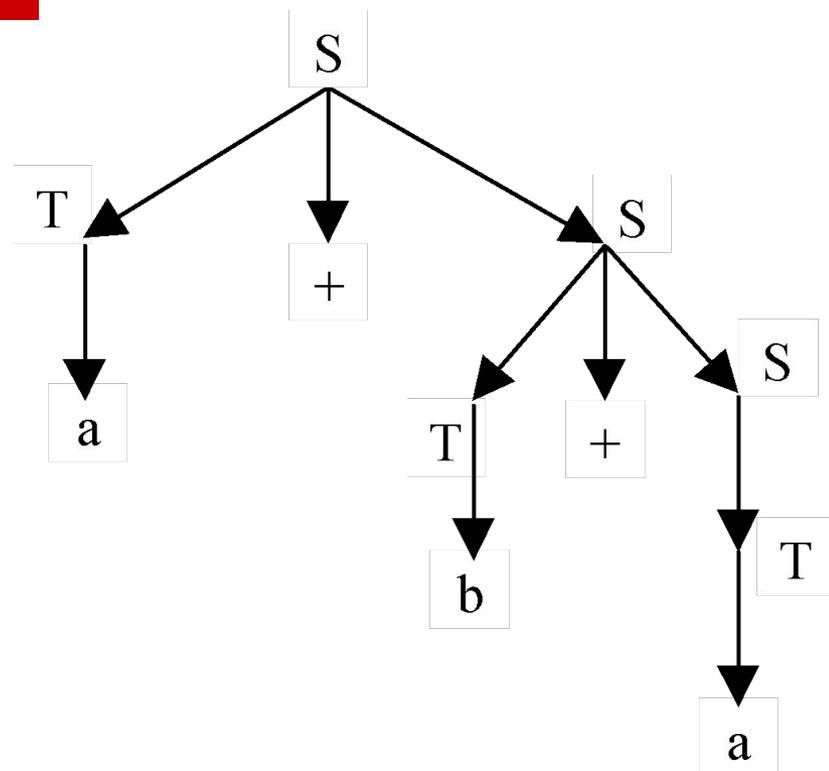


Дерево вывода для цепочки $a+b+a$

Определение:

КС-грамматика G называется *неоднозначной*, если существует хотя бы одна цепочка $a \in L(G)$, для которой может быть построено два или более различных деревьев вывода.

В противном случае грамматика называется *однозначной*.



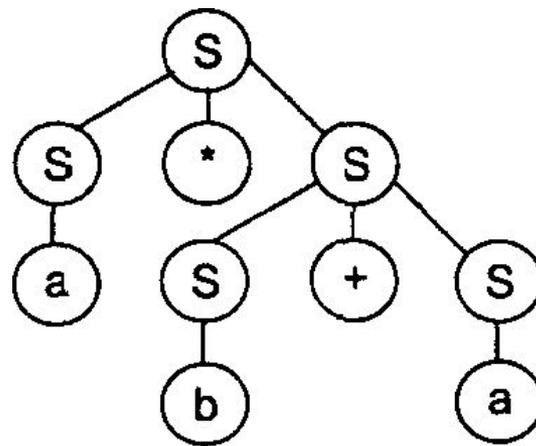
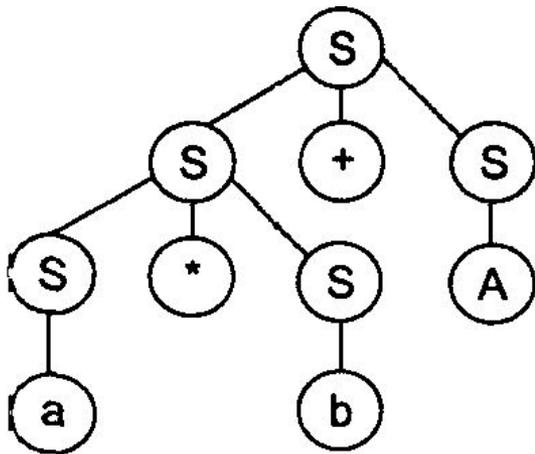
Пример неоднозначной грамматики

- $G = (\{if, then, else, a, b\}, \{S\}, \mathbf{P}, S)$,
где $\mathbf{P} = \{S \rightarrow if\ b\ then\ S\ else\ S \mid if\ b\ then\ S \mid a\}$.
- В этой грамматике для цепочки
if b then if b then a else a
можно построить два дерева вывода.
- Неоднозначность – свойство грамматики, а не языка
- $S \rightarrow if\ b\ then\ S \mid if\ b\ then\ S' \ else\ S \mid a$
 $S' \rightarrow if\ b\ then\ S' \ else\ S' \mid a$



Еще раз вернемся к неоднозначным грамматикам

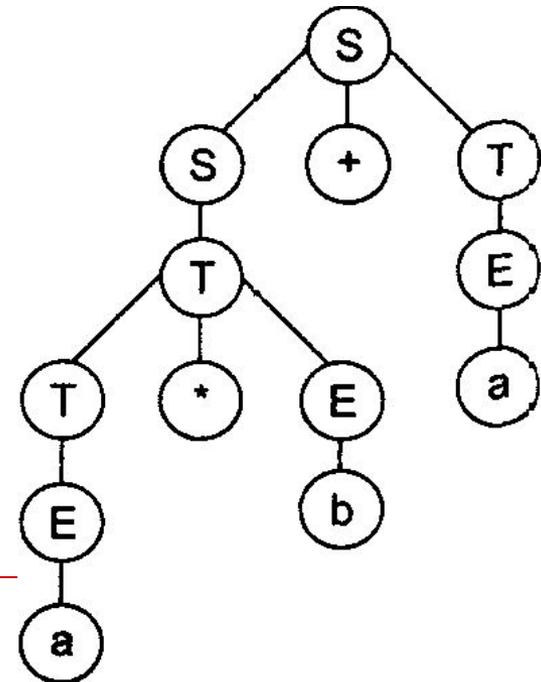
- $G(\{+, -, *, /, (,), a, b\}, \{S\}, \mathbf{P}, S)$:
 $\mathbf{P}: S \rightarrow S+S \mid S-S \mid S*S \mid S/S \mid (S) \mid a \mid b$
- Для цепочки $a*b+a$ возможны два левых вывода:
(1) $S \Rightarrow S+S \Rightarrow S*S+S \Rightarrow a*S+S \Rightarrow a*b+S \Rightarrow a*b+a$
(2) $S \Rightarrow S*S \Rightarrow a*S \Rightarrow a*S+S \Rightarrow a*b+S \Rightarrow a*b+a$



Построение эквивалентной однозначной грамматики

- $G'(\{+, -, *, ./, (,), a, b\}, \{S, T, E\}, P', S);$
 $P' = \{S \rightarrow S+T \mid S-T \mid T; T \rightarrow T^*E \mid T/E \mid E;$
 $E \rightarrow (S) \mid a \mid b\}$
- Для цепочки a^*b+a возможен только один левый вывод:

$S \Rightarrow S+T \Rightarrow T+T \Rightarrow T^*E+T \Rightarrow$
 $E^*E+T \Rightarrow a^*E+T \Rightarrow a^*b+T \Rightarrow$
 $a^*b+E \Rightarrow a^*b+a$

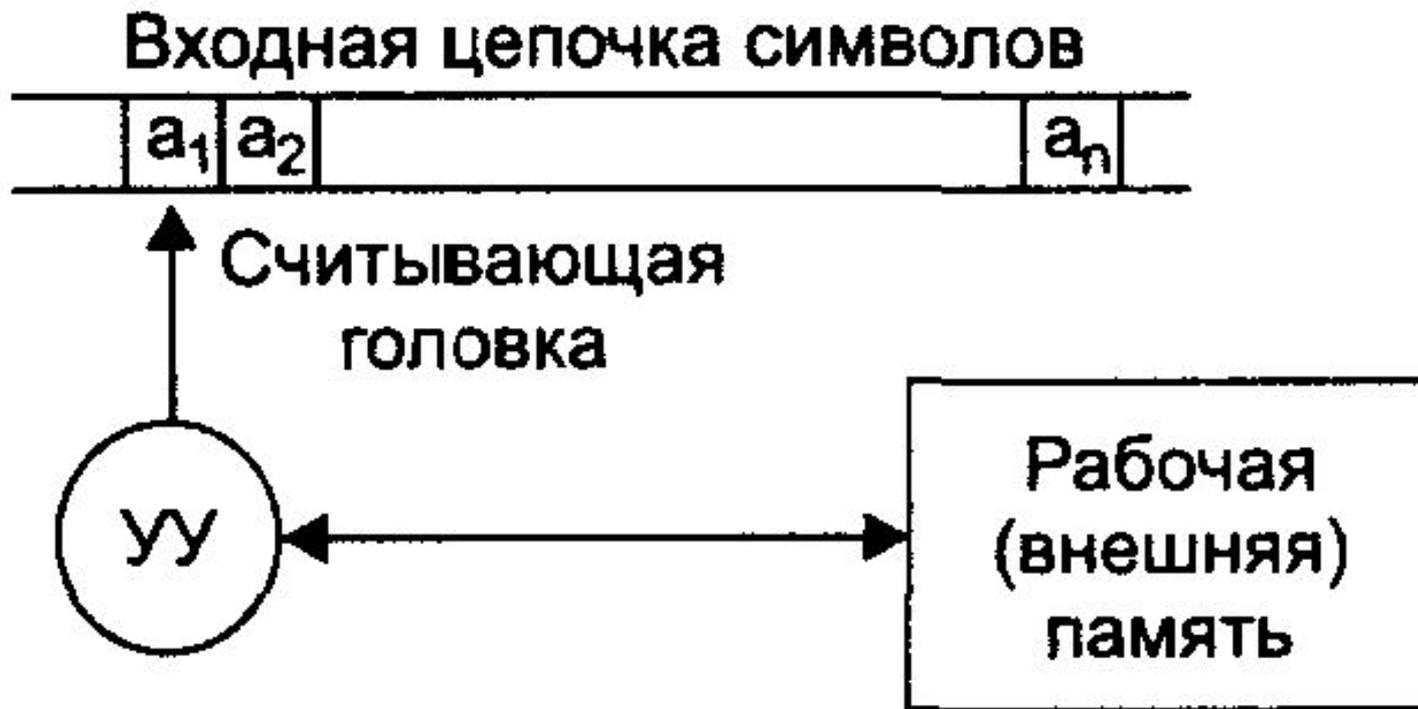


Распознаватели

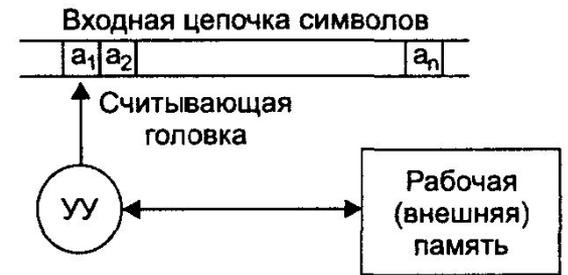


Распознаватели.

Условная схема распознавателя



Компоненты распознавателя



- **лента**, содержащая исходную цепочку входных символов, и **считывающая головки**, обзорающей очередной символ в этой цепочке;
- **устройство управления (УУ)**, которое координирует работу распознавателя, имеет некоторый набор состояний и конечную память (для хранения своего состояния и некоторой промежуточной информации);
- **внешняя (рабочая) память**, которая может хранить некоторую информацию в процессе работы распознавателя и в отличие от памяти УУ может иметь неограниченный объем
- Алфавит распознавателя – ленты + внутренний
- Операции распознавателя – чтение/запись



Работа распознавателя состоит из *последовательности тактов*

- В начале каждого такта состояние распознавателя определяется его **конфигурацией**.
- В процессе работы конфигурация меняется.
- *Конфигурация* определяется:
 - содержимым входной цепочки символов и положением считывающей головки в ней;
 - состоянием УУ;
 - содержимым внешней памяти.
- Всегда задается *начальная конфигурация*.
И несколько *конечных конфигураций*



Работа распознавателя.

Язык, определенный распознавателем

- В начальной конфигурации:
 - считывающая головка на первом символе входной цепочки,
 - УУ находится в заданном начальном состоянии,
 - внешняя память пуста.
- В конечной конфигурации
 - считывающая головка - за концом исходной цепочки
- Распознаватель *допускает входную цепочку a*, если,
 - находясь в начальной конфигурации и
 - получив на вход эту цепочку,
 - он может проделать последовательность шагов, заканчивающуюся одной из его конечных конфигураций.
- Язык, определяемый распознавателем, — это множество всех цепочек, которые допускает распознаватель.



Виды распознавателей

- По **видам считывающего устройства**
 - *двусторонние и односторонние*
- По **видам устройства управления**
 - *детерминированные и недетерминированные*
- По **видам внешней памяти:**
 - *распознаватели без внешней памяти;*
 - *распознаватели с ограниченной внешней памятью;*
 - *распознаватели с неограниченной внешней памятью.*



Классификация распознавателей по типам языков

- Для языков с фразовой структурой (тип 0) необходим распознаватель, равносильный машине Тьюринга
 - недетерминированный двусторонний автомат, с неограниченной внешней памятью. Практического применения не имеет
- Для контекстно-зависимых языков (тип 1)
 - двусторонние недетерминированные автоматы с линейно ограниченной внешней памятью. Экспоненциальная сложность
- Для контекстно-свободных языков (тип 2)
 - односторонние недетерминированные автоматы с магазинной внешней памятью — МП-автоматы. Полиномиальная сложность
 - Детерминированные КС-языки – ДМП-автоматы
- Для регулярных языков (тип 3)
 - детерминированные автоматы без внешней памяти — конечные автоматы (КА). Линейная сложность



Задача разбора

- Задача разбора в общем виде:
 - на основе имеющейся грамматики некоторого языка построить распознаватель для этого языка

- Работа распознавателей в составе компиляторов
 - сводится к построению дерева разбора входной цепочки. Затем уже это дерево разбора используется компилятором для синтеза результирующего кода
 - не только установить факт присутствия ошибки во входной программе, но и по возможности определить тип ошибки и то место в цепочке символов, где она встречается



Следующая тема:
«Регулярные выражения»

