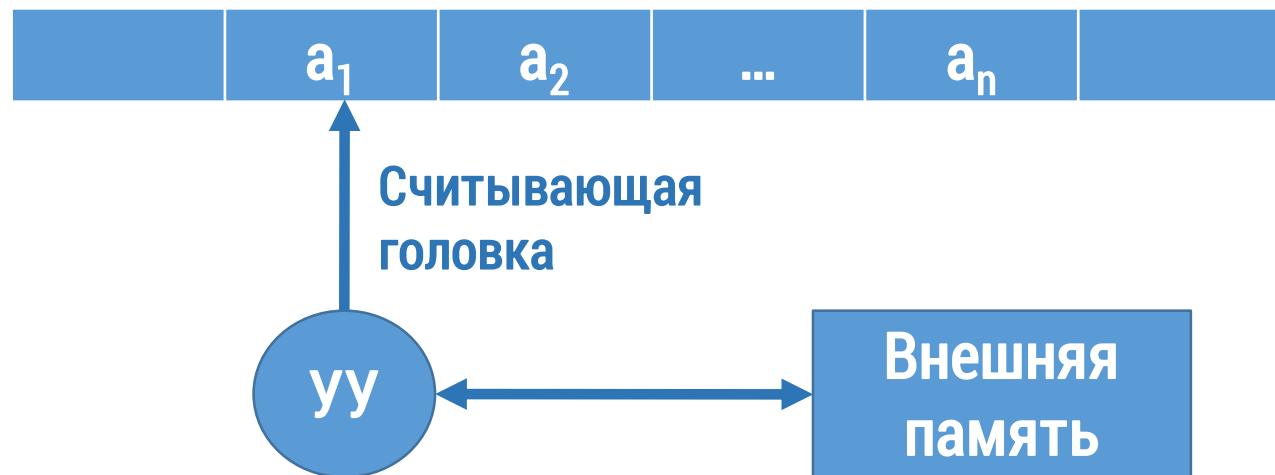


Распознаватели

Общая схема распознавателя

Распознаватель – специальный автомат, который позволяет определить принадлежность цепочки символов некоторому языку.



Элементарные операции

- Чтение очередного символа из входной цепочки
- Сдвиг входной цепочки на заданное количество символов
- Доступ к внешней памяти для чтения или записи информации
- Преобразование информации в памяти УУ, изменение состояния УУ

Конфигурация распознавателя определяется:

- содержимым входной цепочки символов и положением считывающей головки в ней
- состоянием УУ
- содержимым внешней памяти

Задача разбора

- *Задача разбора* заключается в построении распознавателя для языка на основе имеющейся грамматики этого языка
- Задача разбора в общем виде может быть решена не для всех языков
- Для языков программирования задача разбора разрешима и найдены формальные методы ее решения

Классификация распознавателей по структуре

- Односторонние
- Двусторонние

} по видам считывающего устройства

- Детерминированные
- Недетерминированные

} по видам управляющего устройства

- Без внешней памяти
- С ограниченной внешней памятью
- С неограниченной внешней памятью

} по видам внешней памяти

Классификация распознавателей по типам

Для языков типа 0: *машина Тьюринга* – недетерминированный двусторонний *автомат*, имеющий неограниченную внешнюю память

Для языков типа 1: двусторонние недетерминированные *автоматы* с линейно ограниченной внешней памятью

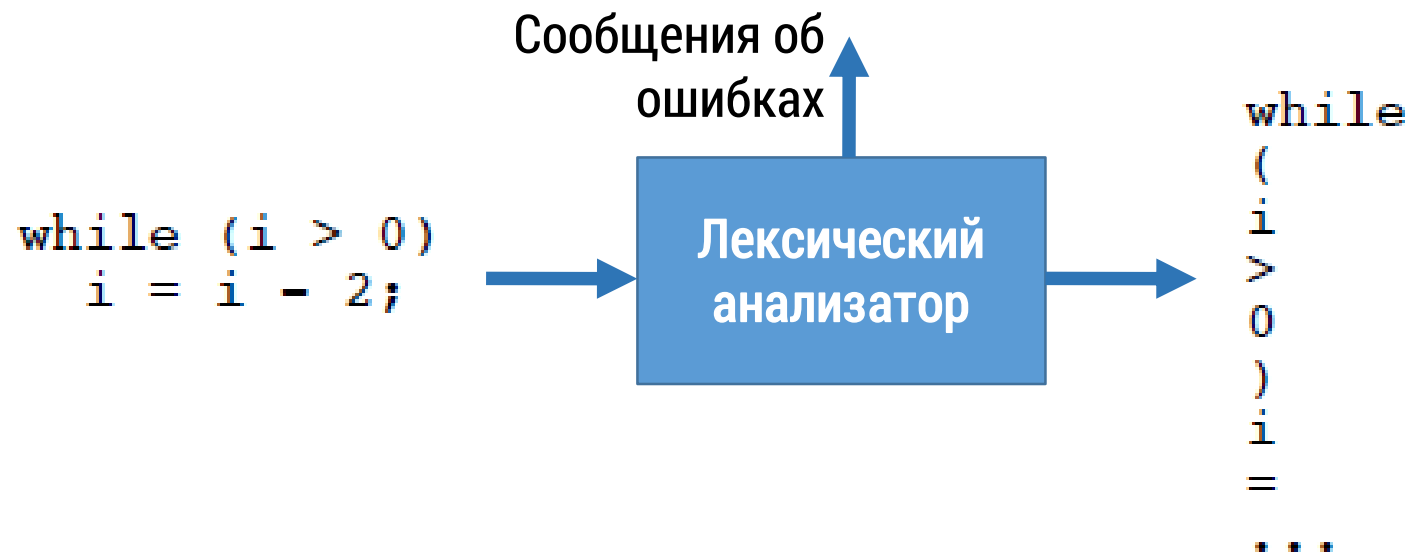
Для языков типа 2: односторонние недетерминированные *автоматы* с магазинной (стековой) внешней памятью (МП-автоматы)

Для языков типа 3: односторонние недетерминированные *автоматы* без внешней памяти – конечные автоматы (КА)

Лексический анализатор

Лексический анализ

Лексический анализ = чтение исходной программы в виде потока символов и группировка их в токены



- Выделение лексем: идентификаторов, констант, служебных слов, знаков операций
- Исключение из текста программы комментариев и незначащих символов

Токены, шаблоны, лексемы

- *Токен* представляет собой пару, состоящую из имени токена и необязательного атрибута
- *Шаблон* – это описание вида, который может принимать лексема токена
- *Лексема* представляет собой последовательность символов исходной программы, которая соответствует шаблону токена и идентифицируется лексическим анализатором как экземпляр токена

```
printf("Total = %d\n", score);
```

Токен	Шаблон	Лексема
id	последовательность символов, начинающаяся с буквы или '_', за которыми могут следовать буквы и цифры	printf, score
separator	скобки, запятые, двоеточия, точки с запятой и т.п.	() , ;
literal	символы, заключенные в ""	Total = %d\n

Атрибуты токенов

➤ *Атрибут* задает значение, описывающее лексему

$$E = M * C / 2$$

<Токен, Атрибут>

<id, Указатель на запись в таблице идентификаторов для E>

<assign>

<id, Указатель на запись в таблице идентификаторов для M>

<mult>

<id, Указатель на запись в таблице идентификаторов для C>

<div>

<number, 2>

Виды ошибок

- Недопустимые или нераспознанные символы
- Недопустимые символы в строковой константе
- Незакрытые комментарии

```
int a double } switch b[2] =;
```

```
fi(a == f(x))
```

Проблема построения лексического анализатора

➤ Определение границ лексем

```
k = i+++++j;
```

Правильный вариант разбора:

```
k = i++ + ++j;
```

Возможный вариант разбора:

```
k = i++ ++ + j;
```

```
$ gcc 1.c (gcc версии 7.2.1)
```

```
1.c: In function 'main':
```

```
1.c:5:12: error: lvalue required as increment
```

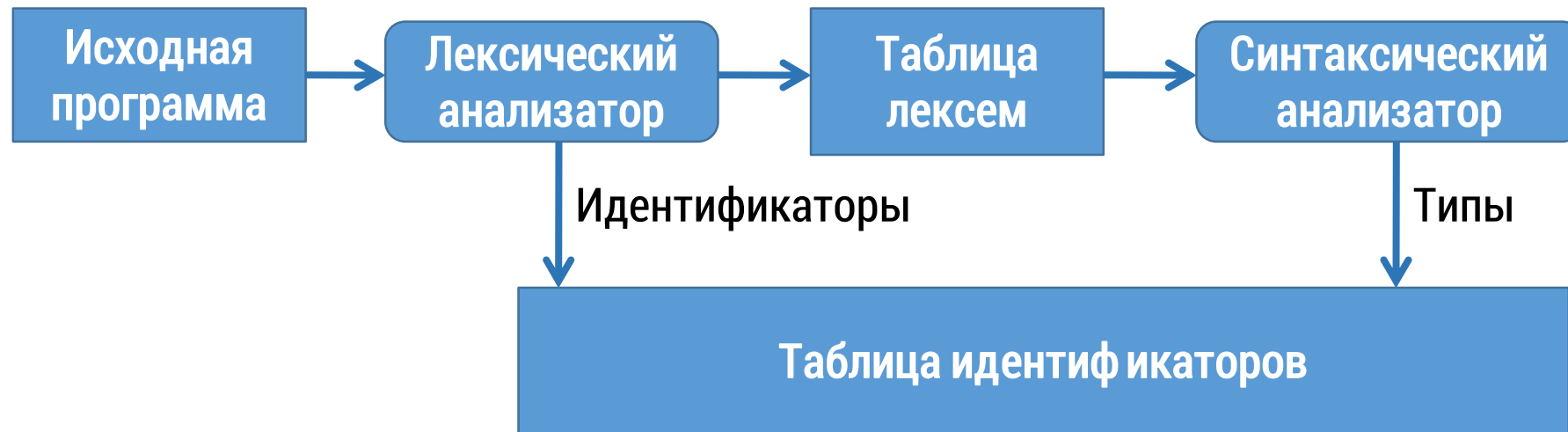
```
operand
```

```
    k = i+++++j;
```

```
        ^~
```

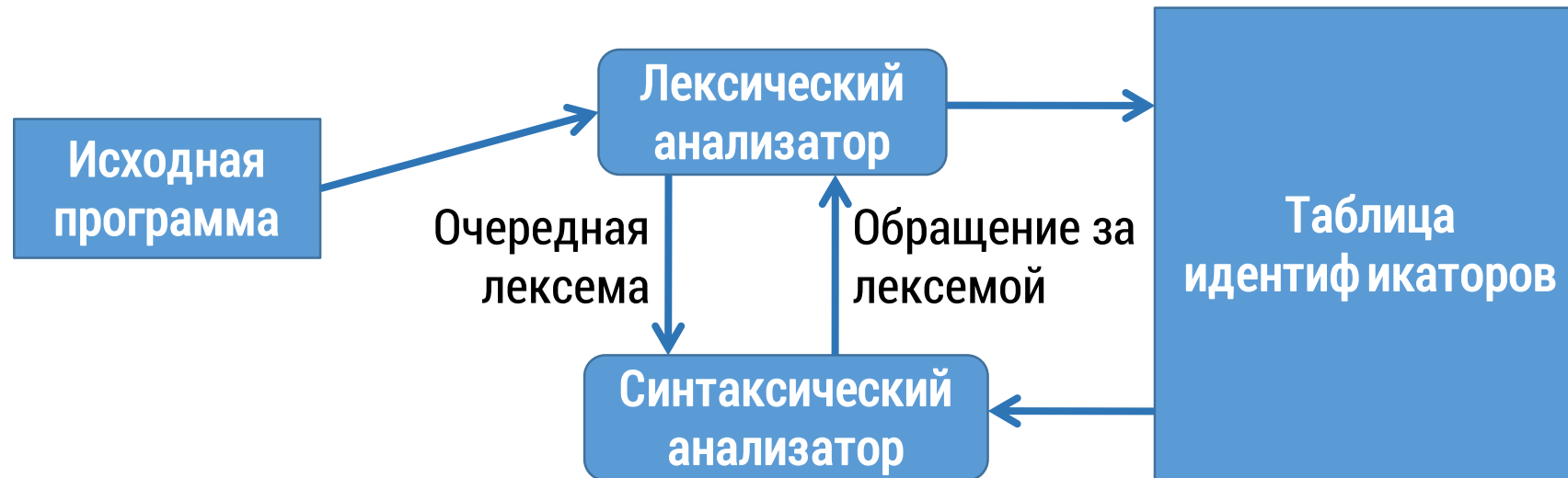
Методы взаимодействия лексического и синтаксического анализатора

1. Последовательный



Методы взаимодействия лексического и синтаксического анализатора

2. Параллельный



Конечные автоматы

1. Конечный автомат – распознаватель для регулярных языков
2. Два класса конечных автоматов:
 - а) Недетерминированные конечные автоматы
 - б) Детерминированные конечные автоматы

Определение конечного автомата

$M(Q, V, \delta, q_0, F)$

Q – конечное множество состояний автомата

V – конечное множество допустимых входных символов

δ – функция переходов вида $\delta(q, a) = R, a \in V, q \in Q, R \subseteq Q$

q_0 – начальное состояние автомата, $q_0 \in Q$

F – непустое множество конечных состояний автомата, $F \subseteq Q$

Автоматные грамматики

Автоматные грамматики – подкласс регулярных грамматик

- Ливолинейные автоматные грамматики

$A \rightarrow Bt$ или $A \rightarrow t$, где $A, B \in VN$; $t \in VT$

- Праволинейные автоматные грамматики

$A \rightarrow tB$ или $A \rightarrow t$, где $A, B \in VN$; $t \in VT$

Преобразование регулярной грамматики к автоматному виду

$G(\{a, \{, \}, *, (,)\}, \{S, C, K\}, P, S)$

$P: S \rightarrow C^* \mid K\}$

$C \rightarrow (* \mid Ca \mid C\{ \mid C\} \mid C(\mid C^* \mid C)$

$K \rightarrow \{ \mid Ka \mid K(\mid K^* \mid K) \mid K\{$

1. $VN' = \{S, C, K\}$

2. $S \rightarrow C^* \rightarrow S \rightarrow S_1$ и $S_1 \rightarrow C^*$

$C \rightarrow (* \rightarrow C \rightarrow C_1^*$ и $C_1 \rightarrow ($

$G'(\{a, \{, \}, *, (,)\}, \{S, S_1, C, K\}, P', S)$

$P': S \rightarrow S_1 \mid K\}$

$S_1 \rightarrow C^*$

$C \rightarrow C_1^* \mid Ca \mid C\{ \mid C\} \mid C(\mid C^* \mid C)$

$C_1 \rightarrow ($

$K \rightarrow \{ \mid Ka \mid K(\mid K^* \mid K) \mid K\{$

Язык, заданный КА

- *КА принимает цепочку символов*, если получив на вход эту цепочку, он из начального состояния может перейти в одно из конечных состояний
- *Язык, заданный КА* – это множество всех цепочек символов, которые принимаются этим автоматом
- Два КА *эквивалентны*, если они задают один и тот же язык

Способы задания КА

- Функция переходов
- Таблица переходов
- Граф переходов

Функция переходов

$G'(\{a, \{, \}, *, (,)\}, \{S, S_1, C, C_1, K\}, P', S)$

$P': S \rightarrow S_1 \mid K\}$

$S_1 \rightarrow C^*$

$C \rightarrow C_1^* \mid Ca \mid C\{ \mid C\} \mid C(\mid C^* \mid C)$

$C_1 \rightarrow ($

$K \rightarrow \{ \mid Ka \mid K(\mid K^* \mid K) \mid K\}$

$M(\{H, C_1, K, C, S_1, S\}, \{a, \{, \}, *, (,)\}, \delta, H, \{S\})$

$\delta: \delta(H, () = C_1 \quad \delta(K,)) = K \quad \delta(K, \{) = K$

$\delta(H, \{) = K \quad \delta(C, a) = C \quad \delta(K, \}) = S$

$\delta(K, a) = K \quad \delta(C, \{) = C \quad \delta(S_1,)) = S$

$\delta(K, () = K \quad \delta(C, \}) = C \quad \delta(C, *) = \{C, S_1\}$

$\delta(K, *) = K \quad \delta(C, () = C \quad \delta(C_1, *) = C$

Таблица переходов

$G'(\{a, \{, \}, *, (,)\}, \{S, S_1, C, C_1, K\}, P', S)$

$P': S \rightarrow S_1 \mid K$

$S_1 \rightarrow C^*$

$C \rightarrow C_1^* \mid Ca \mid C\{ \mid C\} \mid C(\mid C^* \mid C)$

$C_1 \rightarrow ($

$K \rightarrow \{ \mid Ka \mid K(\mid K^* \mid K) \mid K\}$

Состояние	a	{	}	*	()
H	∅	K	∅	∅	C ₁	∅
C ₁	∅	∅	∅	C	∅	∅
K	K	K	S	K	K	K
C	C	C	C	S ₁ , C	C	C
S ₁	∅	∅	∅	∅	∅	S
S	∅	∅	∅	∅	∅	∅

Граф переходов

$G'(\{a, \{, \}, *, (,)\}, \{S, S_1, C, C_1, K\}, P', S)$

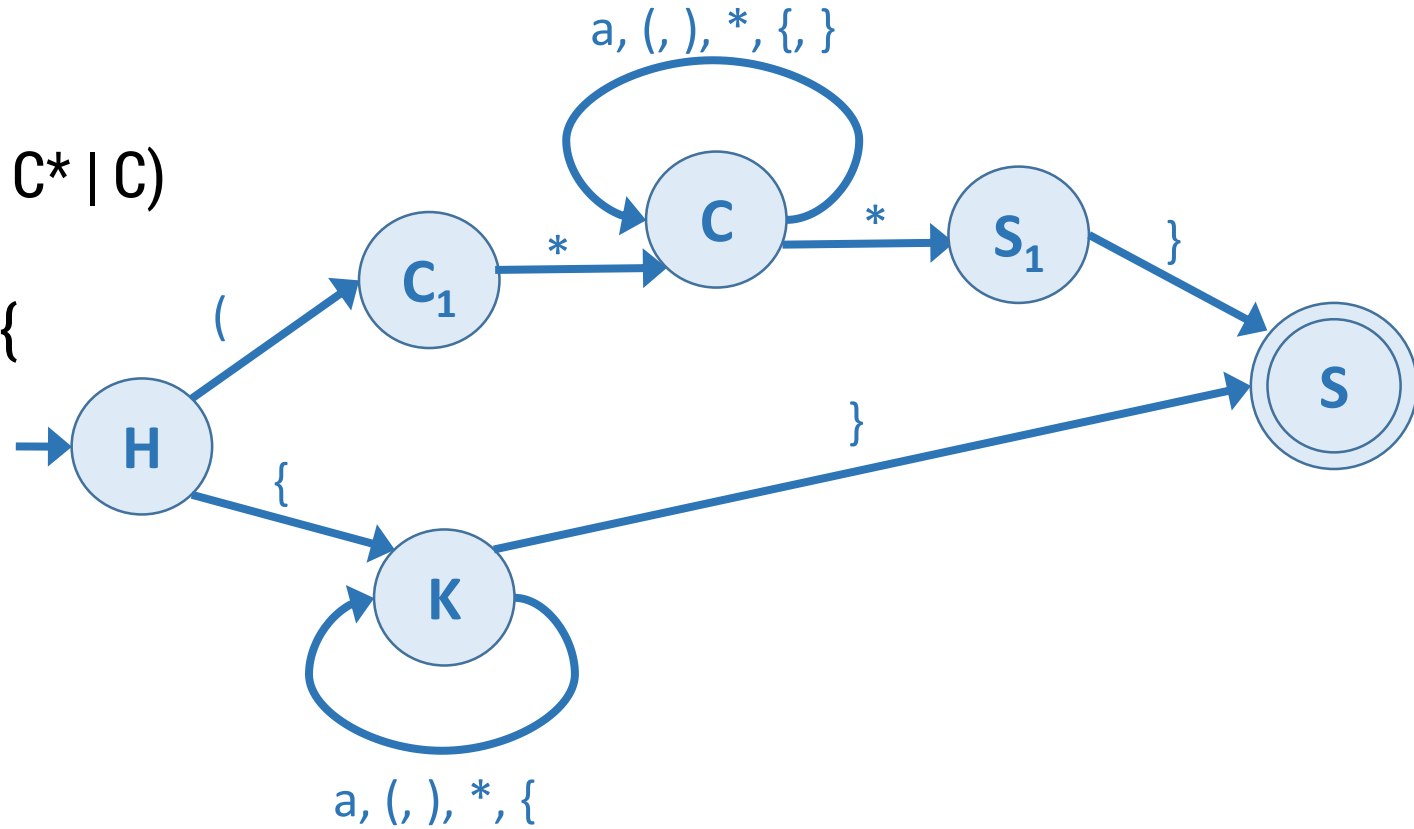
$P': S \rightarrow S_1 \mid K\}$

$S_1 \rightarrow C^*$

$C \rightarrow C_1^* \mid Ca \mid C\{ \mid C\} \mid C(\mid C^* \mid C)$

$C_1 \rightarrow ($

$K \rightarrow \{ \mid Ka \mid K(\mid K^* \mid K) \mid K\{$



Преобразование праволинейной автоматной грамматики к левوليнейному виду

$G (\{S, A, B, C, D\}, \{a, b, c, d, \#, \perp\}, P, S)$,

P :

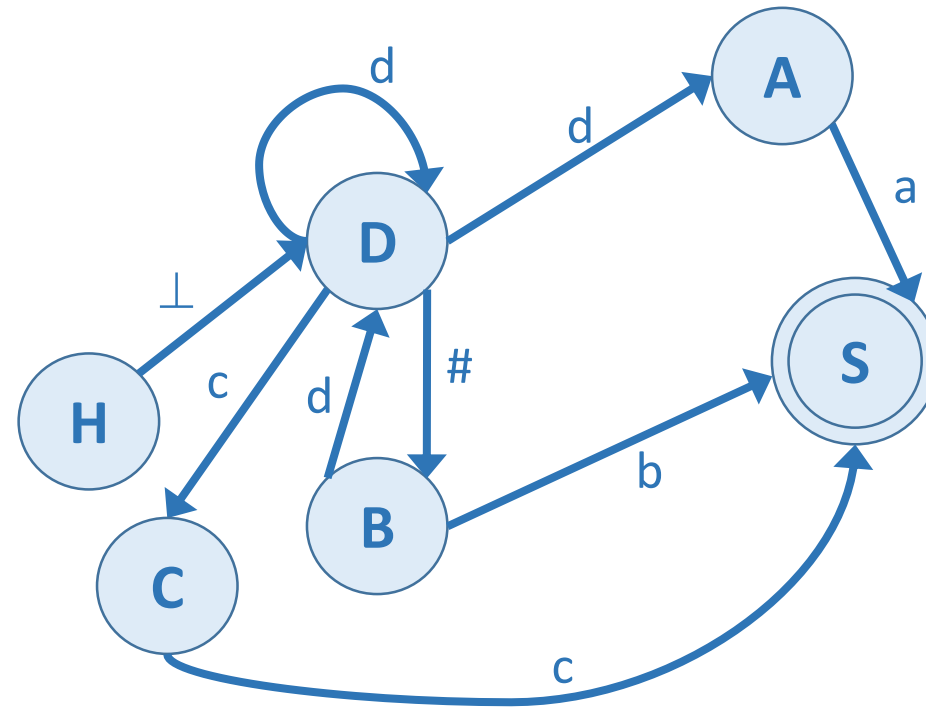
$S \rightarrow aA \mid bB \mid cC$

$A \rightarrow dD$

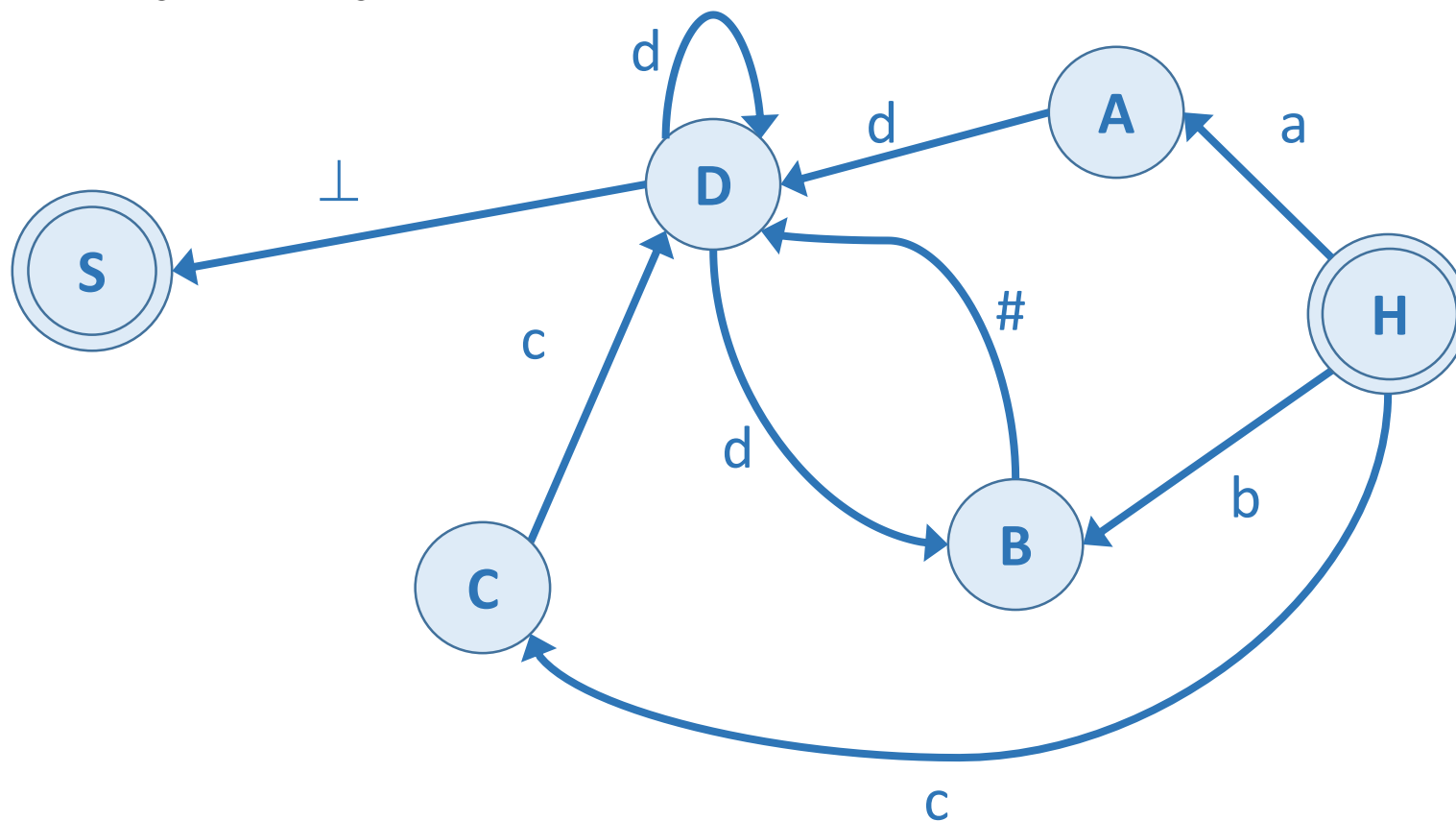
$B \rightarrow \#D$

$D \rightarrow dD \mid dB \mid \perp$

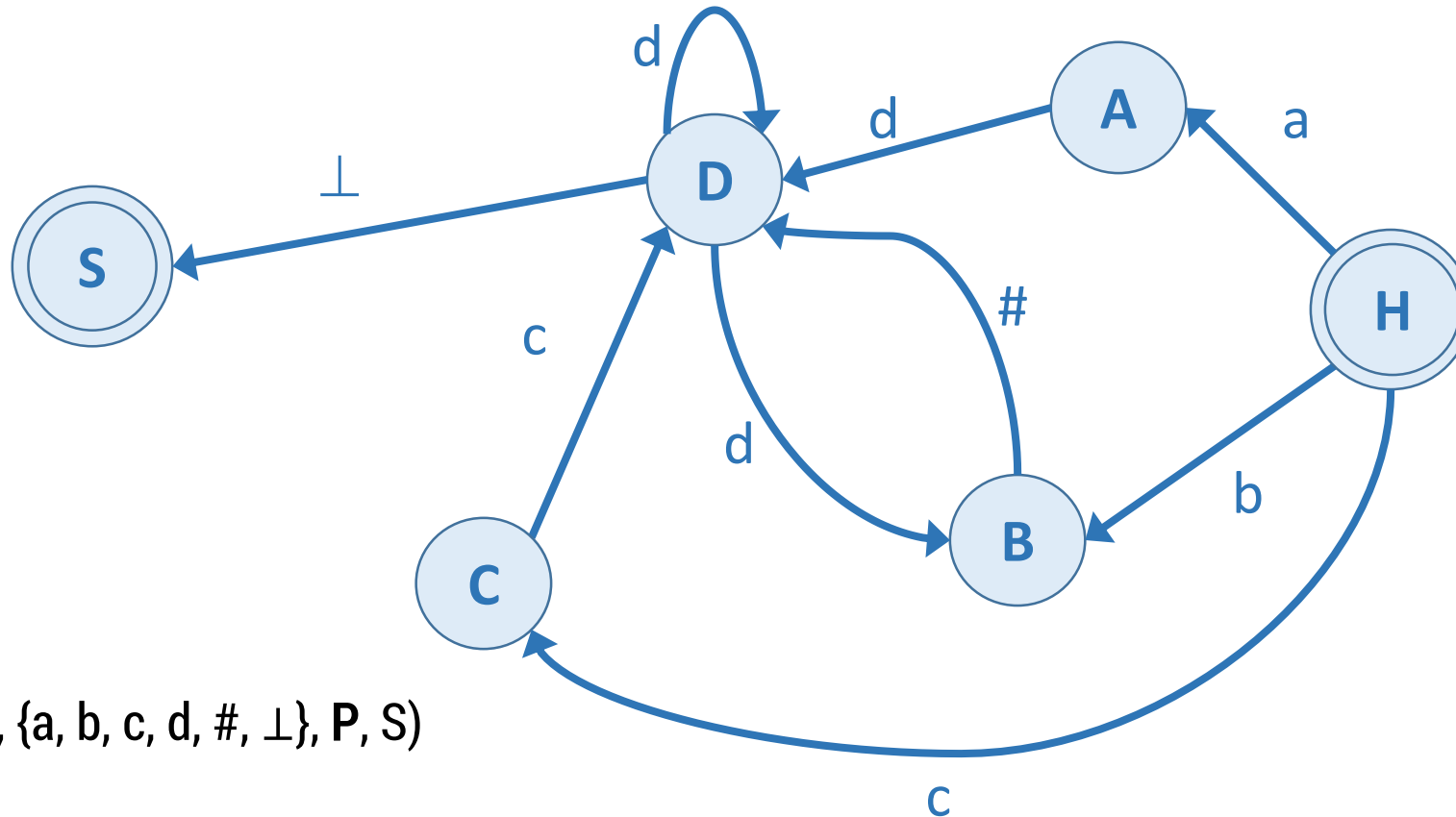
$C \rightarrow cD$



Преобразование праволинейной автоматной грамматики к левوليнейному виду



Преобразование праволинейной автоматной грамматики к левوليнейному виду



$G (\{S, A, B, C, D\}, \{a, b, c, d, \#, \perp\}, P, S)$

P:

$S \rightarrow D\perp$

$D \rightarrow Dd \mid Ad \mid Cc \mid B\#$

$A \rightarrow a$

$C \rightarrow c$

$B \rightarrow b \mid Dd$

Недетерминированный КА

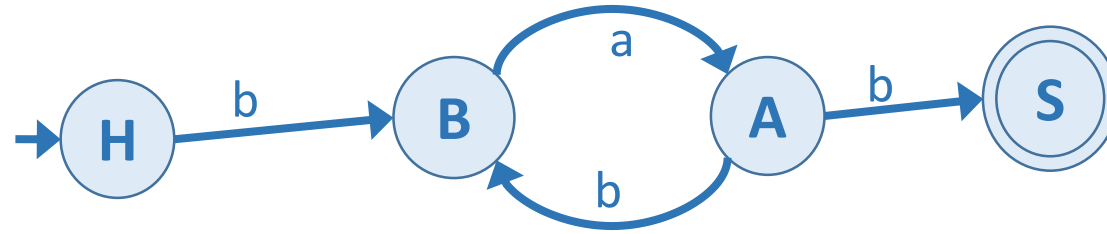
1. Один и тот же символ может помечать дуги, исходящие из одного состояния в несколько разных других
2. Дуга может быть помечена пустой цепочкой символов ε вместо символа входного алфавита (или вместе с ним)

Детерминированный КА (ДКА)

ДКА представляет собой частный случай НКА, в котором

- нет переходов для входа ε
- для каждого состояния q и входного символа a имеется ровно одна дуга, выходящая из q и помеченная a

Преобразование КА к детерминированному виду

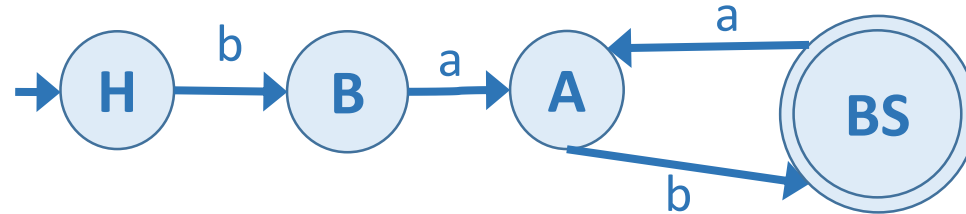


$$\delta(H, b) = \{B\}$$

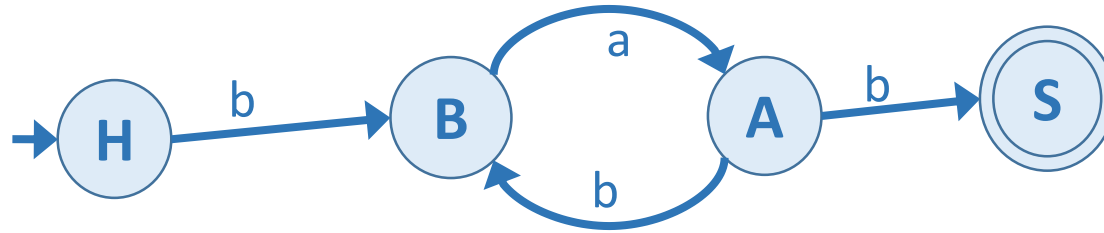
$$\delta(B, a) = \{A\}$$

$$\delta(A, b) = \{B, S\} \text{ – совмещенное состояние BS}$$

$$\delta(BS, a) = \{A\}$$



Преобразование КА к детерминированному виду



1. $Q' = \{H, B, A, S, HB, HA, HS, BA, BS, AS, HBA, HBS, BAS, HAS, HBAS\}$

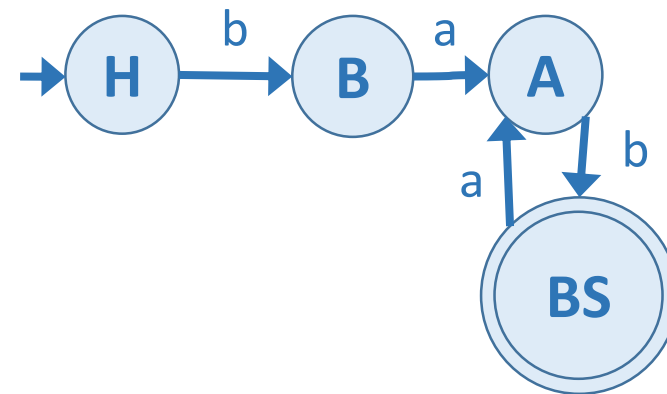
2. $\delta'(H, b) = B$	$\delta'(B, a) = A$	$\delta'(A, b) = BS$
$\delta'(HA, b) = BS$	$\delta'(HB, a) = A$	$\delta'(HB, b) = B$
$\delta'(HS, b) = B$	$\delta'(BA, a) = A$	$\delta'(BA, b) = BS$
$\delta'(AS, b) = BS$	$\delta'(BS, a) = A$	$\delta'(HBA, a) = A$
$\delta'(HBA, b) = BS$	$\delta'(HAS, b) = BS$	$\delta'(HBS, b) = B$
$\delta'(HBS, a) = A$	$\delta'(BAS, b) = BS$	$\delta'(BAS, a) = A$
$\delta'(HBAS, a) = A$	$\delta'(HBAS, b) = BS$	

3. $q'_0 = H$

4. $F' = \{S, HS, AS, BS, HAS, HBS, BAS, HBAS\}$

5. $M'(\{H, B, A, BS\}, \{a, b\}, H, \{BS\})$

$\delta'(H, b) = B$	$\delta'(B, a) = A$	$\delta'(A, b) = BS$	$\delta'(BS, a) = A$
---------------------	---------------------	----------------------	----------------------



Задания для самостоятельной работы

1. Построить по заданной регулярной грамматике детерминированный конечный автомат:

$G = (\{X, Y, Z, W, V\}, \{0, 1, \sim, \#, \&\}, P, X)$, где P :

$X \rightarrow 0Y \mid 1Z \mid 1 \quad Y \rightarrow 0Z \mid \sim W \mid \#$

$Z \rightarrow 1Y \mid 1W \mid 0V \quad W \rightarrow 0W \mid 1W \mid \#$

$V \rightarrow \&Z$

2. Построить по заданной регулярной грамматике детерминированный конечный автомат:

$G = (\{K, L, M, N, P\}, \{0, 1, \&, \%, a, b\}, C, K)$, где C :

$K \rightarrow 1M \mid 1 \quad M \rightarrow 0L \mid \&N \mid \&P$

$L \rightarrow 1L \mid 0L \mid \%P \quad N \rightarrow aN \mid bN \mid \%P$

$P \rightarrow 1P \mid aP \mid 0$