

Конечные распознаватели и регулярные грамматики

Любой регулярный язык порождается регулярной грамматикой и может быть распознан конечным распознавателем, следовательно, для каждого конечного распознавателя существует регулярная грамматика, порождающая тот язык, который допускается распознавателем, и, наоборот, для регулярной грамматики существует конечный распознаватель, который допускает только то множество цепочек, которое порождает регулярная грамматика. Далее рассмотрим алгоритмы получения грамматики, порождающих множество цепочек, допускаемых заданным конечным распознавателем и алгоритмы построения конечного распознавателя, допускающего множество цепочек, порождаемое заданной регулярной грамматикой.

Получение грамматики из конечного распознавателя

Пусть задан конечный распознаватель.

Поставим в соответствие множеству входных символов распознавателя множество терминалов, а множеству состояний – множество нетерминалов.

Дадим нетерминалу S_i , соответствующему состоянию s_i такую интерпретацию:

$\langle \text{цепочки, допускаемые состоянием } s_i \rangle$

Если в распознавателе существует переход из состояния s_i в s_j под действием входного символа x ,

то $\langle \text{цепочки, допускаемые состоянием } s_i \rangle$ могут начинаться символом (терминалом) x , за которым следуют $\langle \text{цепочки, допускаемые состоянием } s_j \rangle$, т.е. получаем правило: $S_i \rightarrow x S_j$.

Если же в распознавателе есть ϵ -переход из состояния s_i в s_j ,

то $\langle \text{цепочки, допускаемые состоянием } s_i \rangle$ представляют собой $\langle \text{цепочки, допускаемые состоянием } s_j \rangle$,

т.е. получаем правило $S_i \rightarrow S_j$.

Если в распознавателе состояние s_i допускающее,

то $\langle \text{цепочка, допускаемая состоянием } s_i \rangle$ может быть пустой,

т.е. получаем правило: $S_i \rightarrow \epsilon$.

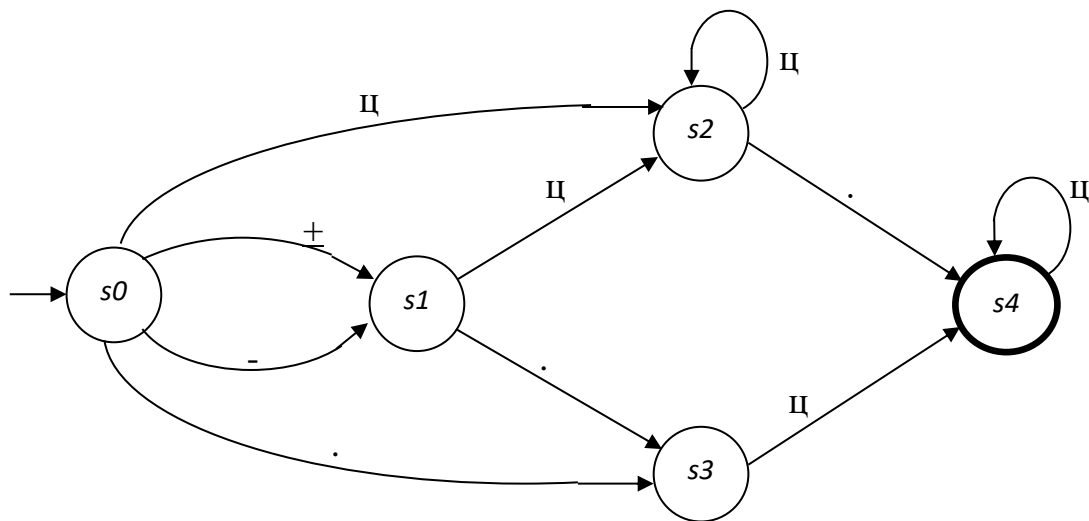
Если каждому переходу и каждому допускающему состоянию распознавателя сопоставить полученные выше правила, то получим автоматную правостороннюю грамматику (правостороннюю грамматику специального вида), возможно, с цепными правилами.

Алгоритм получения автоматной правосторонней грамматики из конечного распознавателя.

1. Множеством терминалов грамматики сделать множество входных символов распознавателя.
2. Множеством нетерминалов грамматики сделать множество состояний распознавателя, а начальным нетерминалом – его начальное состояние.
3. Если в распознавателе есть переход из состояния s_i в s_j под действием входного символа x , то в грамматику ввести правило: $S_i \rightarrow x S_j$.
4. Если в распознавателе есть ϵ -переход из состояния s_i в s_j , то в грамматику нужно ввести правило $S_i \rightarrow S_j$.
5. Если в распознавателе состояние s_i допускающее, то в грамматику ввести правило: $S_i \rightarrow \epsilon$.

Пример.

Граф конечного распознавателя.



Автоматная правосторонняя грамматика.

$$S_0 \rightarrow + S_1$$

$$S_0 \rightarrow - S_1$$

$$S_0 \rightarrow \text{ц} S_2$$

$$S_0 \rightarrow \cdot S_3$$

$$S_1 \rightarrow \text{ц} S_2$$

$$S_1 \rightarrow \cdot S_3$$

$$S_2 \rightarrow \text{ц} S_2$$

$$S_2 \rightarrow \cdot S_4$$

$$S_3 \rightarrow \text{ц} S_4$$

$$S_4 \rightarrow \text{ц} S_4$$

$$S_4 \rightarrow \epsilon$$

Нетерминалу S_i , соответствующему состоянию s_i , можно дать другую интерпретацию:

<цепочки, связывающие начальное состояние с состоянием s_i >

Если в распознавателе существует переход из состояния s_j в s_i под действием входного символа x ,

то <цепочки, связывающие начальное состояние с состоянием s_i > могут представлять собой <цепочки, связывающие начальное состояние с состоянием s_j >, к которым приписан символ (терминал) x ,

т.е. получаем правило: $S_i \rightarrow S_j x$.

Если в распознавателе существует ε -переход из состояния s_j в s_i ,

то <цепочки, связывающие начальное состояние с состоянием s_i > представляют собой <цепочки, связывающие начальное состояние с состоянием s_j >,

т.е. получаем правило: $S_i \rightarrow S_j$.

Если в распознавателе состояние s_i начальное,

то <цепочка, связывающая начальное состояние с состоянием s_i > представляет собой пустую цепочку,

т.е. получаем правило: $S_i \rightarrow \varepsilon$.

Если в распознавателе состояние s_i допускающее,

то <цепочки, связывающие начальное состояние с состоянием s_i > представляют собой цепочки, допускаемые распознавателем,

поэтому S_i — начальный нетерминал.

Если же в распознавателе $n > 1$ допускающих состояний, то можно ввести новое допускающее состояние и из каждого допускающего состояния, кроме введённого, провести ε -дугу в новое состояние.

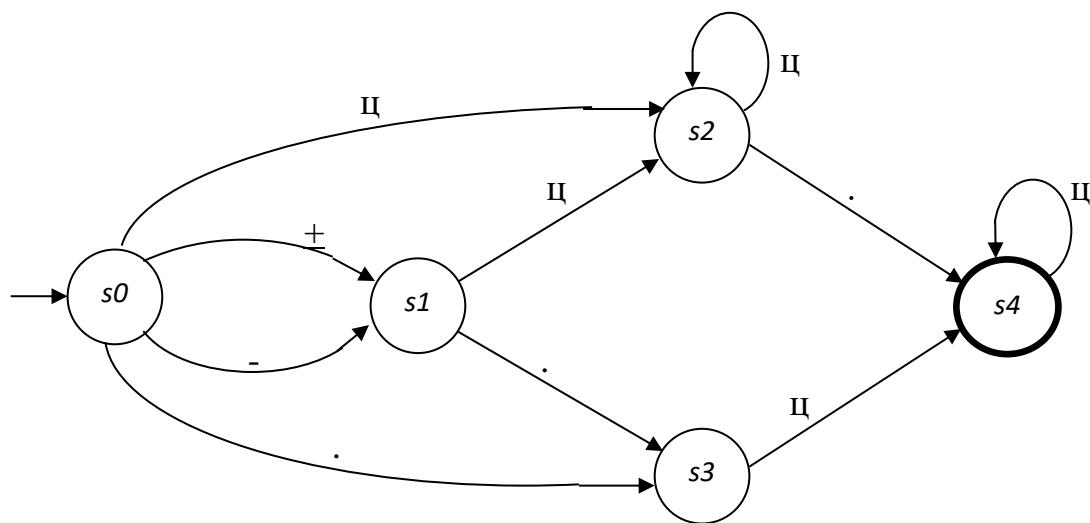
Сопоставив каждому переходу, начальному и допускающему состоянию распознавателя полученные выше правила, получим автоматную левостороннюю грамматику, возможно, с цепными правилами.

Алгоритм получения автоматной левосторонней грамматики из конечного распознавателя с одним допускающим состоянием.

1. Множеством терминалов грамматики сделать множество входных символов распознавателя.
2. Множеством нетерминалов грамматики сделать множество состояний распознавателя.
3. Если в распознавателе состояние s_i допускающее, то S_i — начальный нетерминал.
4. Если в распознавателе состояние s_i начальное, то в грамматику ввести правило: $S_i \rightarrow \varepsilon$.
5. Если в распознавателе есть переход в состояние s_i из s_j под действием входного символа x , то в грамматику ввести правило: $S_i \rightarrow S_j x$.
6. Если в распознавателе есть ε -переход в состояние s_i из s_j , то в грамматику ввести правило: $S_i \rightarrow S_j$.

Пример.

Граф конечного распознавателя.



Автоматная левосторонняя грамматика, полученная из распознавателя:

| | | |
|---------------------------------|---------------------------------|-------------------------------|
| $S_4 \rightarrow S_4 \text{ Ц}$ | $S_3 \rightarrow S_0 \cdot$ | $S_1 \rightarrow S_0 +$ |
| $S_4 \rightarrow S_2 \cdot$ | $S_2 \rightarrow S_0 \text{ Ц}$ | $S_1 \rightarrow S_0 -$ |
| $S_4 \rightarrow S_3 \text{ Ц}$ | $S_2 \rightarrow S_1 \text{ Ц}$ | $S_0 \rightarrow \varepsilon$ |
| $S_3 \rightarrow S_1 \cdot$ | $S_2 \rightarrow S_2 \text{ Ц}$ | |

Построение конечных распознавателей по регулярным грамматикам

К регулярным грамматикам относятся право- и левосторонние грамматики, которые можно преобразовать в автоматные право- и левосторонние грамматики. По автоматным грамматикам можно построить конечные распознаватели, “обратив” алгоритмы получения автоматных грамматик из конечных распознавателей.

Алгоритм построения конечного распознавателя по автоматной правосторонней грамматике.

1. Множеством входных символов распознавателя сделать множество терминалов грамматики.
2. Множеством состояний распознавателя сделать множество нетерминалов грамматики, а начальным состоянием – состояние, соответствующее начальному нетерминалу.
3. Если в грамматике есть правило $S_i \rightarrow x S_j$, то в распознавателе ввести переход из состояния s_i в состояние s_j под действием входного символа x .
4. Если в грамматике есть правило $S_i \rightarrow \varepsilon$, то состояние s_i сделать допускающим.

В результате выполнения алгоритма может быть получен детерминированный или недетерминированный распознаватель без ε -переходов с одним начальным состоянием.

Алгоритм построения конечного распознавателя по автоматной левосторонней грамматике.

1. Множеством входных символов распознавателя сделать множество терминалов грамматики.
2. Множеством состояний распознавателя сделать множество нетерминалов грамматики, а допускающим состоянием – состояние, соответствующее начальному нетерминалу.
3. Если в грамматике есть правило $S_i \rightarrow S_j x$, то в распознавателе ввести переход в состояние s_i из состояния s_j под действием входного символа x .
4. Если в грамматике есть правило $S_i \rightarrow \varepsilon$, то состояние s_i сделать начальным.

В результате выполнения алгоритма может быть получен детерминированный или недетерминированный распознаватель без ε -переходов с одним допускающим состоянием.

Ещё один способ взаимного преобразования право- и левосторонних грамматик

Алгоритмы построения конечных распознавателей по право- и левосторонним грамматикам и получения право- и левосторонних грамматик из конечных распознавателей можно использовать для взаимного преобразования право- и левосторонних грамматик.

Алгоритм преобразования правосторонней грамматики в левостороннюю.

1. Правостороннюю грамматику преобразовать в автоматную правостороннюю.
2. По автоматной правосторонней грамматике построить конечный распознаватель.
3. Получить левостороннюю грамматику из конечного распознавателя.

Алгоритм преобразования левосторонней грамматики в правостороннюю.

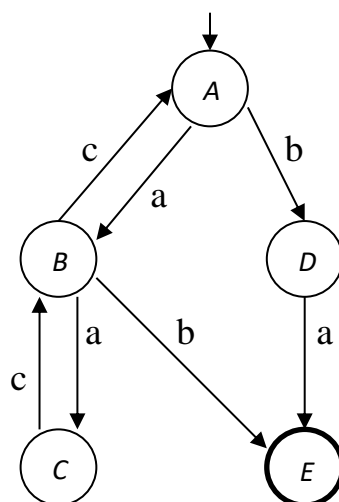
1. Левостороннюю грамматику преобразовать в автоматную левостороннюю.
2. По автоматной левосторонней грамматике построить конечный распознаватель.
3. Получить правостороннюю грамматику из конечного распознавателя.

Пример.

Автоматная
правосторонняя
грамматика

1. $A \rightarrow aB$
2. $A \rightarrow bD$
3. $B \rightarrow aC$
4. $B \rightarrow cA$
5. $B \rightarrow bE$
6. $C \rightarrow cB$
7. $D \rightarrow aE$
8. $E \rightarrow \varepsilon$

Конечный
распознаватель



Автоматная
левосторонняя
грамматика

1. $E \rightarrow Da$
2. $E \rightarrow Bb$
3. $A \rightarrow Bc$
4. $A \rightarrow \varepsilon$
5. $B \rightarrow Aa$
6. $B \rightarrow Cc$
7. $C \rightarrow Ba$
8. $D \rightarrow Ab$