ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ
КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»
ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

Автор курса:
Абрамов И.В.

Решение избранных задач по курсу «Теория Автоматов, Языков и Вычислений»

Авторы решений:
Алексеев М.А.
Алексеев С.А.
Моргунов С.А.

Москва 2007
21.01

1. Приведите примеры не менее чем пяти языков в алфавите
$$\Sigma = \{a, b, c, d, e\}.$$

Теория.

Определение. Языком в алфавите Σ называется множество цепочек в Σ.

Решение.

$$L_1 = \{ab, bc, cd, de\}$$
$$L_2 = \{a^i b^i c^i d^i e^i \mid i > 0\}$$
$$L_3 = \{\omega\omega^R \mid \omega \in \{a, b, c, d, e\}^*\}$$
$$L_4 = \{(ae)^i (bdc)^j \mid j - 1 = i \geqslant 0\}$$
$$L_5 = \{dead, bad\}$$

2. Напишите регулярные выражения для следующих языков:

(а) множество цепочек над алфавитом $\{a, b, c\}$, содержащих хотя бы один символ a и хотя бы один символ b;

(б) множество цепочек из нулей и единиц, в которых десятый от правого края символ равен 1;

(в) множество цепочек из нулей и единиц, содержащих не более одной пары последовательных единиц.

Теория.

Определение. Регулярные выражения в алфавите Σ и регулярные множества, которые они *обозначают*, определяются рекурсивно следующим образом:

(1) $\varnothing$ — регулярное выражение, обозначающее регулярное множество $\varnothing$,

(2) ε — регулярное выражение, обозначающее регулярное множество $\{\varepsilon\}$,

(3) если $a \in \Sigma$, то a — регулярное выражение, обозначающее регулярное множество $\{a\}$,

(4) если p и q — регулярные выражение, обозначающие регулярные множества P и Q соответсвенно, то

  (a) $(p + q)$ — регулярное выражение, обозначающее $P \cup Q$;

  (b) (pq) — регулярное выражение, обозначающее PQ;

  (c) $(p)^*$ — регулярное выражение, обозначающее P^*;

(5) ничто другое не является регулярным выражением.

Решение.

(а) $(a + b + c)^*((a^+(a + c)^*b^+) + (b^+(b + c)^*a^+))(a + b + c)^*$

(б) $(0 + 1)^*1(0 + 1)^9$

(в) $(0 + 10)^*(11)?(0 + 01)^*$

3. Постройте праволинейные грамматики для языков, состоящих из

(а) идентификаторов произвольной длины, состоящих из букв и цифр, но начинающихся с буквы (например, как в языке Си);

(б) идентификаторов, которые должны содержать от одного до шести символов и начинаться с буквы I, J, K, L, M или N;

(в) целочисленные константы языка Си (напомним, что целочисленные константы в Си бывают восьмеричные, десятичные и шестнадцатеричные, а в конце может следовать суффикс, задающий тип и знаковость константы);

(г) вещественных констант языка Си;

(д) всех цепочек из нулей и единиц, имеющих чётное число нулей и чётное число единиц.

Теория.

Грамматика G называется *праволинейной*, если каждое правило из P имеет вид $A \to xB$ или $A \to x$, где $A, B \in N, x \in \Sigma^*$.

Решение.

(а) $S \to aA' \mid \ldots \mid zA' \mid AA' \mid \ldots \mid ZA'$
$A' \to aA' \mid \ldots \mid ZA' \mid 0A' \mid \ldots \mid 9A' \mid \varepsilon$

(б) $S \to IA' \mid \ldots \mid NA'$
$A' \to aB' \mid \ldots \mid ZB' \mid 0B' \mid \ldots \mid 9B' \mid \varepsilon$
$B' \to \ldots$
$C' \to \ldots$
$D' \to \ldots$
$E' \to a \mid \ldots \mid Z \mid 0 \mid \ldots \mid 9 \mid \varepsilon$

(в) $S \to 0A' \mid 0D' \mid \ldots \mid 9D'$
$A' \to xB' \mid 0C' \mid \ldots \mid 7C'$
$B' \to 0B' \mid 1B' \mid \ldots \mid 9B' \mid aB' \mid \ldots \mid fB' \mid AB' \mid \ldots \mid FB' \mid$
$0z' \mid 1z' \mid \ldots \mid 9z' \mid az' \mid \ldots \mid fz' \mid Az' \mid \ldots \mid Fz'$
$C' \to OC' \mid \ldots \mid 7C' \mid z'$
$D' \to OD' \mid \ldots \mid 9D' \mid z'$
$z' \to u \mid l \mid ul \mid lu \mid U \mid L \mid UL \mid LU \mid \varepsilon$

(г) $S \to 0S \mid \ldots \mid 9S \mid .A \mid 0A \mid \ldots \mid 9A \mid 0D \mid \ldots \mid 9D$
$A \to 0A \mid \ldots \mid 9A \mid eD \mid e-D \mid e+D \mid 0D \mid \ldots \mid 9D$
$D \to 0D \mid \ldots \mid 9D \mid z'$
$z' \to l \mid L \mid \varepsilon$

(д) $A \to 0C \mid 1B \mid \varepsilon$
$B \to 0D \mid 1A$
$C \to 0A \mid 1D$
$D \to 0B \mid 1C$

4. Постройте КС-грамматики, порождающие

(а) все цепочки из нулей и единиц с одинаковым числом тех и других;

(б) $\{a_1a_2 \ldots a_na_n \ldots a_2a_1 \mid a_i \in \{0,1\}, 1 \leqslant i \leqslant n\}$;

(в) $\{0^i1^j \mid i \neq j$ и $i,j \geqslant 0\}$;

(г) всевозможные последовательности правильно расставленных скобок.

Теория.

Грамматика G называется *контекстно-свободной*, если каждое правило из P имеет вид $A \to \alpha$, где $A \in N, \alpha \in (N \cup \Sigma)^*$.

Решение.

(а) $S \to 0AS \mid A0S \mid \varepsilon$
$A \to 0AA \mid AA0 \mid 1$

(б) $S \to 0S0 \mid 1S1 \mid \varepsilon$

(в) $S \to 0S1 \mid A$
$A \to 1B \mid 0C \mid 1 \mid 0$
$B \to 1B \mid 1$
$C \to 0C \mid 0$

(г) $S \to (S) \mid ()S \mid \varepsilon$

5. Следующая грамматика порождает язык регулярного выражения $0^*1(0+1)^*$.

$$S \to A1B$$
$$A \to 0A \mid \varepsilon$$
$$B \to 0B \mid 1B \mid \varepsilon$$

Запишите левый и правый вывод, а также нарисуйте деревья вывода следующих цепочек:

(а) 00101;

(б) 1001;

(в) 00011.

Теория.

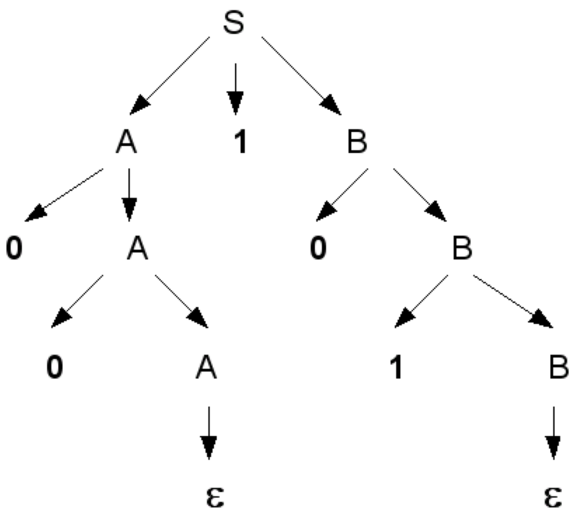
Определение. Вывод, при котором каждый раз правило применяется к самому левому нетерминалу, называется *левым выводом*. Аналогично определяется *правый вывод*.

Определение. Дерево вывода в грамматике $G = (N, \Sigma, P, S)$ — это помеченное упорядоченное дерево, каждая вершина которого помечена символом из множества $N \cup \Sigma \cup \{\varepsilon\}$. Если внутренняя вершина помечена символом A, а ее прямые потомки — символами $X_1X_2 \dots X_n$, то $A \to X_1X_2 \dots X_n$ — правило этой грамматики.

Решение.

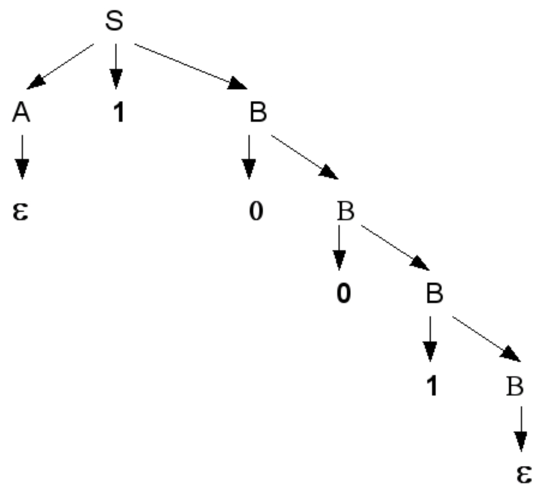
(а) левый вывод: $S \to A1B \to 0A1B \to 00A1B \to 001B \to 0010B \to 00101B \to 00101$

правый вывод: $S \to A1B \to A10B \to A101B \to A101 \to 0A101 \to 00A101 \to 00101$

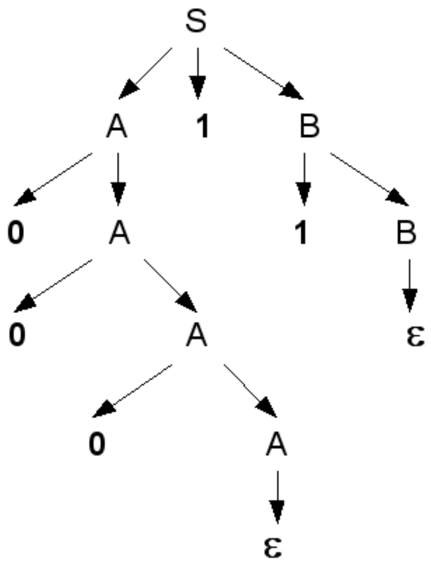


(б) левый вывод: $S \to A1B \to 1B \to 10B \to 100B \to 1001B \to 1001$

правый вывод: $S \to A1B \to A10B \to A100B \to A1001B \to A1001 \to 1001$



(в) левый вывод: $S \to A1B \to 0A1B \to 00A1B \to 000A1B \to 0001B \to 00011B \to 00011$
правый вывод: $S \to A1B \to A11B \to A11 \to 0A11 \to 00A11 \to 000A11 \to 00011$



6. Постройте КЗ-грамматики, порождающие

(а) $\{a^{n^2} \mid n \geqslant 1\}$;

(б) $\{\omega\omega \mid \omega \in \{a,b\}^+\}$;

(в) $\{\omega \mid \omega \in \{a,b,c\}^+$ и число букв a в цепочке ω равно числу букв b, равному числу букв $c\}$;

(г) $\{a^m b^n a^m b^n \mid m,n \geqslant 1\}$.

Теория.

Определение. Грамматика G называется *контекстно-зависимой*, если каждое правило из P имеет вид $\alpha \to \beta$, где $|\alpha| \leqslant |\beta|$.

Решение.

(а) В начале составим цепочку вида $B\underbrace{aaa\ldots aAa}_{n+1}FE$, где B, A, F и E нетерминалы определенного назначения. Сделаем это с помощью правил:

$$S \to BTaFE$$
$$T \to aT \mid A$$

Теперь будем A продвигать влево до B и при переходе через каждый символ a будем создавать новый нетерминал C. Наша цепочка примет вид $BAaCaC\ldots CaFE$. Тем самым количество C равно $n-1$ и означает, что нужно создать $n-1$ копию исходной цепочки из символов a, полученной на предыдущем шаге, чтобы получить n^2 самих символов a. (*Замечание.* $n-1$ по той причине, что одна копия строки из n символов a у нас имеется с самого начала). Для описанных действий в грамматику добавится правило:

$$aA \to AaC$$

Назначение символа A себя исчерпало, так что дойдя до B он нам уже не нужен. В этот момент заведём специальный символ Z, для которого опишем ε-правило и с его помощью избавимся от A. В результате наша цепочка примет вид $BaCaC\ldots aCaFE$, а в грамматику добавятся:

$$BA \to BZ$$
$$Z \to \varepsilon$$

Следующий шаг заключается в перемещении всех символов C к символу E. Для этого нам нужно организовать беспрепятственный переход C через a и F. В итоге мы получим цепочку $B\underbrace{aaa\ldots a}_{n}F\underbrace{CC\ldots C}_{n-1}E$. Правила, выполняющие это действие описаны ниже.

$$Ca \to aC$$
$$CF \to FC$$

Теперь из каждого нетерминала C, использую рядом стоящий специальный символ E, создадим новый нетерминал D, который в последвии примет участие в непосредственном копировании нашей исходной цепочки. Также заметим, что необходимо обеспечить свободный переход символов C, через появившиеся D, чтобы они могли переместиться к E и замениться на символ D. После замены всех символов C получим цепочку $B\underbrace{aaa\dots a}_{n}F\underbrace{DD\dots D}_{n-1}E$, а в грамматику запишем следующее:

$$CE \to DE$$
$$CD \to DC$$

Теперь будем непосредственно реализовывать копирование исходной цепочки. Для этого каждый символ D будем перетаскивать через всю цепочку к B и переходя через каждый символ a будем создавать новый нетерминал Y, который в последствии и превратится в терминалы a нашей цепочки. Всего символов D, у нас $n-1$, а это и есть то количество копий цепочки, которое необходимо создать. Поэтому переместив все D к B мы создадим столько нетерминалов Y, сколько нам не хватало, чтобы количество символов a стало равным n^2. Но не стоит забывать, что D может столкнуться с F и появившимися Y, и в этом случае необходимо реализовать свободный переход D через F и Y. В результате цепочка примет вид $B\underbrace{DDD\dots D}_{n-1}\underbrace{aY\dots Ya}_{n^2}FE$, а в грамматику допишем правила:

$$aD \to DaY$$
$$YD \to DY$$
$$FD \to DF$$

Заметим, что символ D нам больше не понадобится, поэтому от него можно избавиться. Наша цепочка упростится и примет вид $B\underbrace{aY\dots Ya}_{n^2}FE$. И для реализации этого шага грамматике потребуется одно новое правило

$$BD \to BZ$$

Далее следует все нетерминалы Y заменить символами a. Для этого нам нужно реализовать свободный проход Y через x и F, а потом при переходе через E заменять его символом a.

Т.е. сначала цепочка станет $B\underbrace{aa\ldots a}_{n}F\underbrace{YY\ldots Y}_{n(n-1)}E$, потом примет вид $B\underbrace{aa\ldots a}_{n}FE\ \underbrace{aa\ldots a}_{n(n-1)}$. Вышеописанные действия выполнят следующие правила грамматики:

$$Ya \to aY$$
$$YF \to FY$$
$$YE \to Ea$$

Теперь осталось только избавиться от нетерминальных символов B и E. Перенесем F к B посредством свободного перехода F через a. А потом уничтожим B и F, создав новый символ G. Наша цепочка преобразуется в $Gaa\ldots aEaa\ldots a$. И в грамматику добавятся правила:

$$aF \to Fa$$
$$BF \to GZ$$

Теперь осталось переместить G к E и и оба их удалить. Для этого нам понадобятся правила:

$$Ga \to aG$$
$$GE \to ZZ$$

В итоге грамматика запишется таким образом:

$$S \to BTaFE$$
$$T \to aT \mid A$$
$$aA \to AaC$$
$$BA \to BZ$$
$$Z \to \varepsilon$$
$$Ca \to aC$$
$$CF \to FC$$
$$CE \to DE$$
$$CD \to DC$$
$$aD \to DaY$$
$$YD \to DY$$
$$FD \to DF$$
$$BD \to BZ$$
$$Ya \to aY$$
$$YF \to FY$$
$$YE \to Ea$$
$$aF \to Fa$$
$$BF \to GZ$$
$$Ga \to aG$$
$$GE \to ZZ$$

(б) Изначально создадим либо цепочку $aCAD$, либо $bCBD$. Затем используя нетерминал C придадим цепочке необходимую длинну, причем с появлением терминала будет появляться и его нетерминальный аналог, дабы в результате получить цепочку из двух одинаковых половинок – $\omega\omega$. Для описанных действий в грамматику добавятся правила:

$$S \to aCAD \mid bCBD$$
$$C \to aCA \mid bCB \mid \varepsilon$$

В результате получим цепочку вида $\omega\Omega^R D$, где $\omega \in \{a, b\}, \Omega \in \{A, B\}$. Теперь нам нужно заменить A и B на соответсвующие им терминалы a и b. Для этого воспользуемся маркером D. Будем постепенно перемещать все нетерминалы к D, менять их на соответсвующие терминалы, которые в свою очередь будут перемещаться к центру, чтобы в итоге мы получили цепочку $\omega\omega$. Правила грамматики, осуществляющие соответсвующую замену и перемещение описаны ниже:

$$AD \to aD$$
$$BD \to bD$$
$$Aa \to aA$$
$$Ab \to bA$$
$$Ba \to aB$$
$$Bb \to bB$$
$$D \to \varepsilon$$

В итоге грамматика состоит из следующих правил:

$$S \to aCAD \mid bCBD$$
$$C \to aCA \mid bCB \mid \varepsilon$$
$$AD \to aD$$
$$BD \to bD$$
$$Aa \to aA$$
$$Ab \to bA$$
$$Ba \to aB$$
$$Bb \to bB$$
$$D \to \varepsilon$$

(в) На начальном этапе создадим нужное количество символов a в цепочке. И на концах цепочки поставим специальные нетерминальные символы. В итоге, используя нижеперечисленные правила, получим цепочку вида $GaAaA \ldots aAE$.

$$S \to GaAFE$$
$$F \to aAF \mid \varepsilon$$

Теперь переместим все A к E, используя свободный переход через терминал a. В результате получим цепочку $G\underbrace{aa\ldots a}_{n}\underbrace{AA\ldots A}_{n}E$. А затем E используем для превращения каждого A в пару BC. Для этого в грамматику запишутся правила:

$$Aa \to aA$$
$$AE \to EBC$$

Наша цепочка теперь представляет собой $G\underbrace{aa\ldots a}_{n}E\underbrace{BCBC\ldots BC}_{2n}$. Далее нужно избавиться от символа E. Для этого переведем G к E и используя ε-правило уничтожим их. Для это необходимы следующие правила:

$$Ga \to aG$$
$$GE \to ZZ$$
$$Z \to \varepsilon$$

Ну и наконец осталось только выстовить символы в нужной последовательности и заменить нетерминалы на терминалы. При добавлении соответсвующих правил, грамматика примет окончательный вид:

$$S \to GaAFE$$
$$F \to aAF \mid \varepsilon\ Aa \to aA$$
$$AE \to EA$$
$$Ga \to aG$$
$$GE \to ZZ$$
$$Z \to \varepsilon$$
$$BC \to CB$$
$$CB \to BC$$
$$aB \to Ba$$
$$Ba \to aB$$
$$aC \to Ca$$
$$Ca \to aC$$
$$B \to b$$
$$C \to C$$

(г) Решение сводится к решению задачи (б) с той лишь разницей, что в нашем случае цепочка ω имеет особый вид $a^m b^n$. Для реализации такого вида цепочки потребуются следубщие правила:

$$S \to aCAD$$
$$C \to aCA \mid bEB$$
$$E \to bEB \mid \varepsilon$$

Остальная часть грамматики идентична грамматике из задания (б):

$$
\begin{aligned}
&S \to aCAD \\
&C \to aCA \mid bEB \\
&E \to bEB \mid \varepsilon \\
&AD \to aD \\
&BD \to bD \\
&Aa \to aA \\
&Ab \to bA \\
&Ba \to aB \\
&Bb \to bB \\
&D \to \varepsilon
\end{aligned}
$$

7. Решите систему уравнений с регулярными коэффициентами

$$
\begin{aligned}
&A_1 = (01^* + 1)A_1 + A_2 \\
&A_2 = 11 + 1A_1 + 00A_3 \\
&A_3 = \varepsilon + A_1 + A_2
\end{aligned}
$$

Теория.

Определение. Уравнения, коэффициенты которых — регулярные выражения, будем называть уравнениями с *регулярными коффициентами.*

Замечание. Решение уравнения с регулярными коэффициентами $X = aX + b$ имеет вид $X = a^*b$

Решение.

Исходя из замечания найдём вид решения первого уравнение и подставим его соответсвенно во второе и третье. Получим

$A_1 = (01^* + 1)^*A_2$

$A_2 = 11 + 1(01^* + 1)^*A_2 + 00A_3$

$A_3 = (01^* + 1)^*A_2 + A_2 + \varepsilon$

Теперь найдём решение второго уравнения, которое будет явно зависеть от A_3.

$A_1 = (01^* + 1)^*A_2$

$A_2 = (1(01^* + 1)^*)^*(00A_3 + 11)$

$A_3 = ((01^* + 1)^* + \varepsilon)A_2 + \varepsilon$

Расскроем скобки во втором уравнении и подсавим значение A_2 в третье уравнение.

$A_1 = (01^* + 1)^*A_2$

$A_2 = (1(01^* + 1)^*)^*00A_3 + (1(01^* + 1)^*)^*11$

$A_3 = ((01^*+1)^*+\varepsilon)(1(01^*+1)^*)^*00A_3+(((01^*+1)^*+\varepsilon)(1(01^*+1)^*)^*+\varepsilon)$

Третье уравнение содержит только одну независимую переменную A_3, значение которой находится с использованием всё того же замечания.
$A_3 = (((01^*+1)^*+\varepsilon)(1(01^*+1)^*)^*00)^*(((01^*+1)^*+\varepsilon)(1(01^*+1)^*)^*+\varepsilon)$
$A_2 = \dots$
$A_1 = \dots$

8. Пусть L — регулярное множество. Докажите, что L^R — регулярное множество.

Решение.
Для начала заметим, что по определнию любое регулярное множество описывается соответсвующим регулярным выражением. Доказательство проведём по индукции.
База индукции:
1) $L_1 = L_1^R = \varnothing$ — регулярное выражение, обозначающее регулярное (пустое) множество;
2) $L_1 = L_1^R = \varepsilon$ — регулярное выражение, обозначающее регулярное множество $\{\varepsilon\}$;
3) $L_1 = L_1^R = a$ — регулярное выражение, обозначающее регулярное множество $\{a\}$;

Предположение индукции:
Если L_i — регулярное выражение, то и L_i^R также является регулярным выражением.
Шаг индукции:
Рассмотрим выражение L_{i+1}. Из определения регулярного выражение следует, что это выражение может быть получено тремя способами:
1) $L_{i+1} = L_i+L_1$, а значит выражение $L_{i+1}^R = (L_i+L_1)^R = L_i^R+L_1^R$ также является регулярным по определению;
2) $L_{i+1} = L_iL_1$, тогда выражение $L_{i+1}^R = (L_iL_1)^R = L_1^RL_i^R$ также является регулярным;
3) $L_{i+1} = L_i^*$, а следовательно выражение $L_{i+1}^R = (L_i^*)^R = (L_i^R)^*$ – регулярное, исходя из определения.

Таким образом получаем, что через n шагов индукции мы получим выражения $L = L_n$ и $L^R = L_n^R$, которые оба будут регулярными и будут описывать два исходных регулярных множества L и L^R соответсвенно.

9. Докажите нерегулярность следующих множеств:

(а) $\{0^n1^n \mid n \geqslant 1\}$. Этот язык, который состоит из всех цепочек, состоящих из нулей, за которыми следует такое же количество единиц.

(б) $\{0^n10^n \mid n \geqslant 1\}$

(в) $\{0^n1^m2^n \mid n$ и m — произвольные целые числа$\}$;

(г) $\{0^n1^m \mid n \leqslant m\}$;

(д) $\{0^n1^2n \mid n \geqslant 1\}$.

Теория.

Лемма (о разрастании для регулярных множеств). Пусть L – регулярное множество. Существует такая константа p, что если $\omega \in L$ и $|\omega| \geqslant p$, то цепочку ω можно представить в виде xyz, где $0 < |y| < p$ и $xy^iz \in L$ для всех $i \geqslant 0$.

Для доказательсва воспользуемся отрицанием изложенной леммы и наглядно покажем её не выполнение.

Решение.

(а) 1) если $y = 0^+$, т.е. $0\underbrace{0\ldots0}_{y}01\ldots1$, то $xy^0z \notin L$

2) если $y = 1^+$, т.е. $0\ldots01\underbrace{1\ldots1}_{y}1$, то $xy^0z \notin L$

3) если $y = 0^+1^+$, т.е. $0\underbrace{0\ldots01\ldots1}_{y}1$, то $xyyz \notin L$

(б) 1) если $y = 0^*10^*$, т.е. $0\underbrace{0\ldots010\ldots0}_{y}0$, то $xyyz \notin L$

2) если $y = 0^+$, т.е. $0\underbrace{0\ldots0}_{y}010\ldots0$, то $xy^0z \notin L$

(в) При $m = 0$ нерегулятность множества аналогична нерегулярности множества $\{0^n1^n \mid n \geqslant 1\}$

(г) 1) если $y = 0^+1^+$, т.е. $0\underbrace{0\ldots01\ldots1}_{y}1$, то $xyyz \notin L$

2) если $y = 0^+$, то $xy^iz \notin L$, где $i > \frac{m-n+|y|}{|y|}$

3) при $m = n$ и $y = 1^+$, т.е. $0\ldots01\underbrace{1\ldots1}_{y}1$, то $xy^0z \notin L$

(д) 1) если $y = 0^+ \mid 1^+$, то $xy^0z \notin L$

2) если $y = 0^+1^+$, то $xyyz \notin L$

10. Постройте детерминированные конечные автоматы с минимальным числом состояний для следующих регулярных выражений.

(а) $(1 \mid 0)^*0(1 \mid 0)$

(б) $(1 \mid 0)^*0(1 \mid 0)(1 \mid 0)$

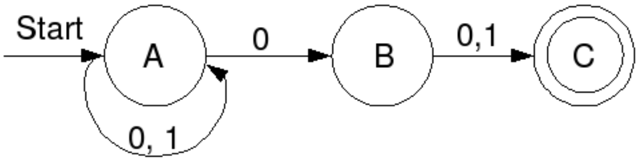
(в) $(1)^*(01)^*1001(1^*0)^*$

Теория.

Определение. Пусть $M = (Q, \Sigma, \delta, q_0, F)$ — недетерминированный конечный автомат. Назовем автомат M *детерминированным*, если множество $\delta(q, a)$ содержит не более одного состояния для любых $q \in Q$ и $a \in \Sigma$.

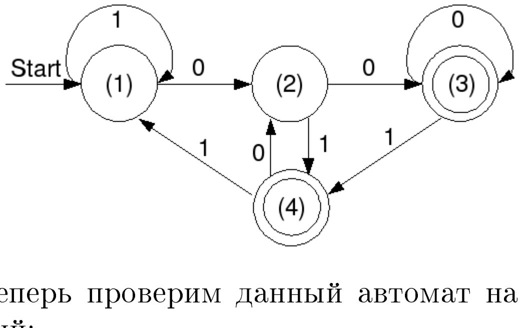
Решение.

(а) Построим сначала недетерминированный конечный автомат.



Опишем состояния детерминированного автомата и построим его.

	0	1
$(1) = \{A\}$	(2)	(1)
$(2) = \{A, B\}$	(3)	(4)
$(3) = \{A, B, C\}$	(3)	(4)
$(4) = \{A, C\}$	(2)	(1)

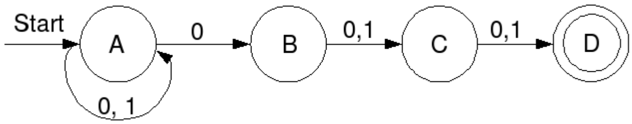


Теперь проверим данный автомат на минимальность состояний:

$$\equiv^0 \{1, 2\}\{3, 4\}$$
$$\equiv^1 \{1\}\{2\}\{3\}\{4\}$$

Следовательно данный автомат содержит минимальное количество состояний, необходимых для описания исходного выражения.

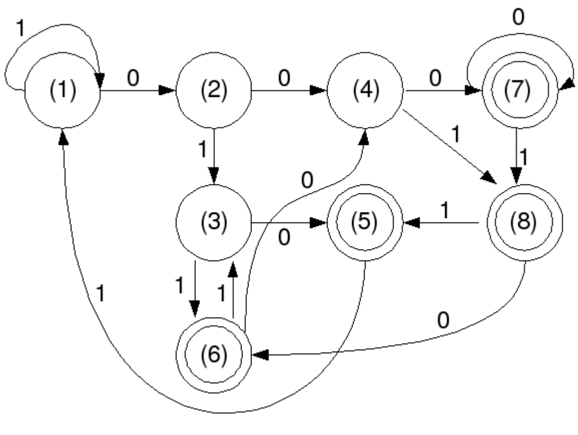
(б) Недетерминированный автомат выглядет следующим образом



Вычислим таблицу состояний и переходов для детерминированного автомата.

	0	1
$(1) = \{A\}$	(2)	(1)
$(2) = \{A, B\}$	(4)	(3)
$(3) = \{A, C\}$	(5)	(6)
$(4) = \{A, B, C\}$	(7)	(8)
$(5) = \{A, D\}$	(2)	(1)
$(6) = \{A, B, D\}$	(4)	(3)
$(7) = \{A, B, C, D\}$	(7)	(8)
$(8) = \{A, C, D\}$	(6)	(5)

В результате получили



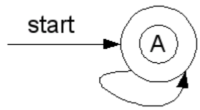
Проверим автомат на минимальность:

$$\equiv^0 \{1,2,3,4\}\{5,6,7,8\}$$
$$\equiv^1 \{1,2\}\{3,4\}\{5,6\}\{7,8\}$$
$$\equiv^2 \{1\}\{2\}\{3\}\{4\}\{5\}\{6\}\{7\}\{8\}$$

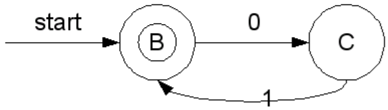
Делаем вывод, что данный автомат содержит минимальное количество состояний, необходимых для описания исходного выражения.

(в) Недетерминированный автомат будем строить по частям. Для начала построим автомат для выражений (1)* и (01)*.

(1)*

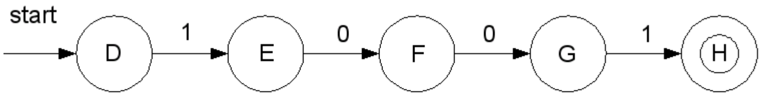


(01)*

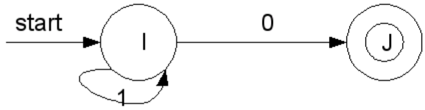


Теперь постороим автомат для выражений 1001 и 1*0.

1001



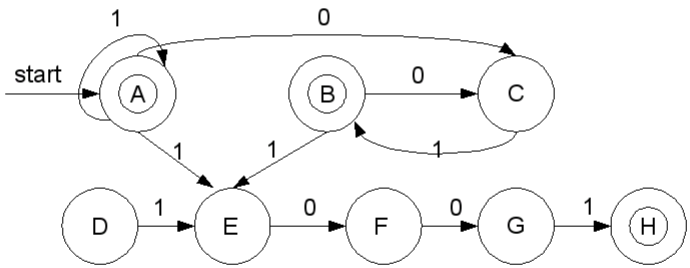
1*0



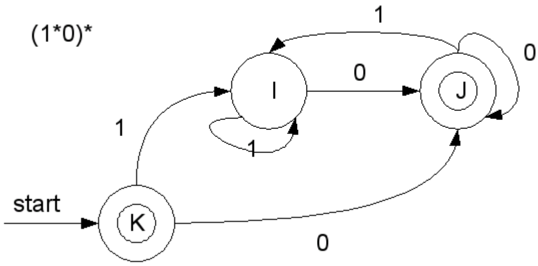
Соединяем части в выражение (1)*(01)*1001 по следующему замечанию.

*Замечание.*Из каждого конечного состояния проводим стрелки, которые идут из каждого начального.

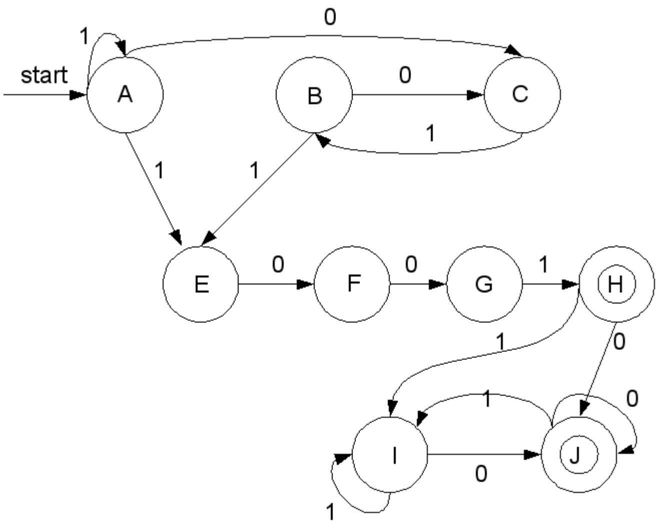
(1)*(01)*



Теперь создадим последнюю часть данного выражения – (1*0)*.



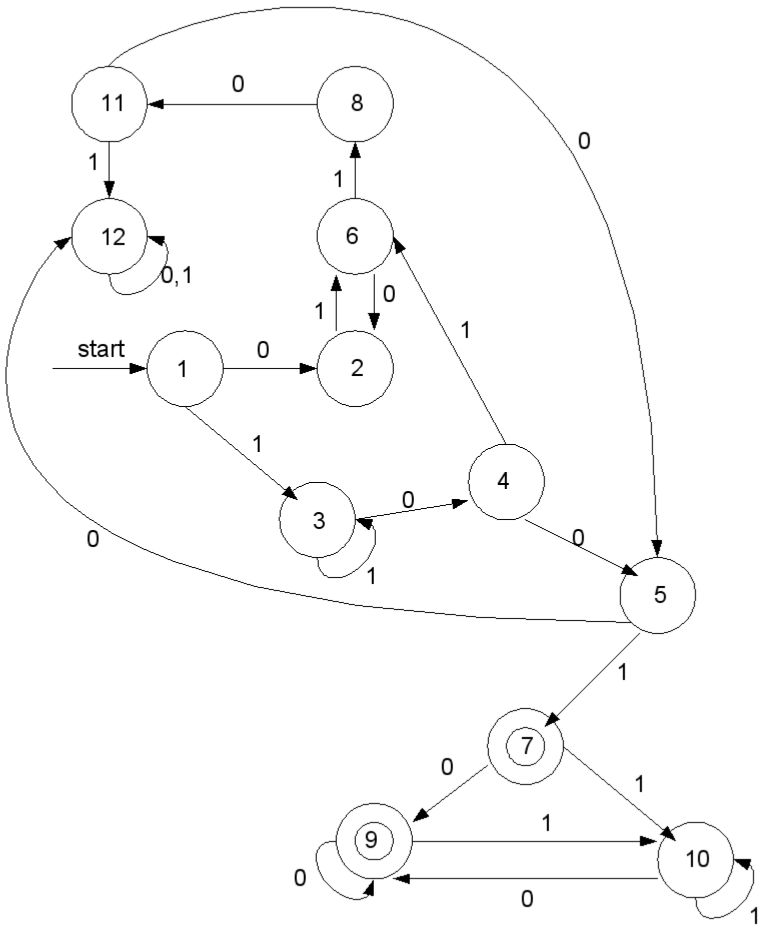
Исспользуя вышеупомянутое замечание постоим недетерминированный автомат для всего выражения.



Вычислим таблицу состояний и переходов для детерминированного автомата.

	0	1
$(1) = \{A\}$	(2)	(3)
$(2) = \{C\}$	$-$	(6)
$(3) = \{A, E\}$	(4)	(3)
$(4) = \{C, F\}$	(5)	(6)
$(5) = \{G\}$	$-$	(7)
$(6) = \{B\}$	(2)	(8)
$(7) = \{H\}$	(9)	(10)
$(8) = \{E\}$	(11)	$-$
$(9) = \{J\}$	(9)	(10)
$(10) = \{I\}$	(9)	(10)
$(11) = \{F\}$	(5)	$-$

По таблице строим детерминированный автомат.



Проведём оценку автомата на свойство минимальности состояний:

$\equiv^0 \{1,2,3,4,5,6,8,10,11,12\}\{7,9\}$

$\equiv^1 \{1,2,3,4,6,8,11,12\}\{5\}\{7\}\{9\}\{10\}$

$\equiv^2 \{4,11\}\{1,2,3,6,8,12\}\{5\}\{7\}\{9\}\{10\}$

$\equiv^3 \{4,11\}\{3,8\}\{1,2,6,12\}\{5\}\{7\}\{9\}\{10\}$

$\equiv^4 \{4,11\}\{3\}\{8\}\{1,6\}\{2,12\}\{5\}\{7\}\{9\}\{10\}$

$\equiv^4 \{4\}\{11\}\{3\}\{8\}\{1\}\{6\}\{2\}\{12\}\{5\}\{7\}\{9\}\{10\}$

Получили, что данный автомат удовлетворяет свойству минимальности.

11. Постройте МП-автоматы, допускающие следующие языков:

(а) $\{a^n b^n a^n \mid n \geqslant 1\}$;
(б) $\{\omega\omega^R \mid \omega \in \{a,b\}^*\}$;
(в) $\{a^m b^n a^m b^n \mid m,n \geqslant 1\}$
(г) $\{\omega\omega \mid \omega \in \{a,b\}^*\}$

Теория.

Определение. Автомат с магазинной памятью — это семерка

$$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

1. Q — конечное множество символов состояний, представляющих всевозможные состояния управляющего устройства;
2. Σ — конечный *входной алфавит*;
3. Γ — конечный алфавит *магазинных символов*;
4. δ — отображение множества $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ в множество конечных подмножеств множества $Q \times \Gamma^*$;
5. $q_0 \in Q$ — *начальное состояние* управляющего устройства;
6. $Z_0 \in \Gamma$ — символ, находящийся в магазине в начальный момент (*начальный символ*);
7. $F \subseteq Q$ — множество *заключительных состояний.*

Лемма (о разрастании для КС-языков). Пусть L — КС-язык. Тогда существует такая константа k, что если $|z| \geqslant k$ и $z \in L$, то цепочку z можно представить в виде $z = uvwxy$, где $vx \neq \varepsilon$, $|vwx| \leqslant k$ и $uv^iwx^iy \in L$ для всех $i \geqslant 0$.

Постоить МП-автоматы для данных языков, кроме (б), невозможно. Для доказательсва данного факта воспользуемся отрицанием изложенной леммы и наглядно покажем её не выполнение.

Решение.

(а) 1) если $v = a^+$ и $x = a^+$, т.е. $a\underbrace{a\ldots a}_{v}ab\ldots ba\underbrace{a\ldots a}_{x}a$, то $uv^0wx^0y \notin L$

2) если $v = b^+$ и $x = b^+$, т.е. $a\ldots ab\underbrace{b\ldots b}_{v}b\underbrace{b\ldots b}_{x}ba\ldots a$, то $uv^0wx^0y \notin L$

3) если $v = a^+b^+$ или $x = b^+a^+$, т.е. $a\ldots a\underbrace{a\ldots ab\ldots b}_{v}b\underbrace{b\ldots ba\ldots a}_{x}a$, то $uv^2wx^2y \notin L$

(б) Язык является КС, поэтому предъявим МП-автомат, допускающий все цепочки данного языка.
$P = (\{q_0, q_1, q_2\}, \{a, b\}, \{Z, a, b\}, \delta, q_0, Z, \{q_2\})$, где

$$\begin{aligned}
&\delta(q_0, \varepsilon, Z) = \{(q_2, \varepsilon)\}\\
&\delta(q_0, a, Z) = \{(q_0, aZ)\}\\
&\delta(q_0, b, Z) = \{(q_0, bZ)\}\\
&\delta(q_0, a, b) = \{(q_0, ab)\}\\
&\delta(q_0, a, a) = \{(q_0, aa), (q_1, \varepsilon)\}\\
&\delta(q_0, b, b) = \{(q_0, bb), (q_1, \varepsilon)\}\\
&\delta(q_0, b, a) = \{(q_0, ba)\}\\
&\delta(q_1, b, b) = \{(q_1, \varepsilon)\}\\
&\delta(q_1, a, a) = \{(q_1, \varepsilon)\}\\
&\delta(q_1, \varepsilon, Z) = \{(q_2, \varepsilon)\}
\end{aligned}$$

(в) 1) если $v = a^+b^+$ или $v = b^+a^+$, то $uv^2wx^2y \notin L$

2) если $v = b^+$ и $x = b^+$, т.е. $a \dots ab\underbrace{b \dots b}_{v}ba \dots a\underbrace{b \dots b}_{x}ba \dots a$, то взяв в условии леммы цепочку с $m = k$ нарушится условие $|vwx| \leqslant k$, а значит $uvwxy \notin L$

3) аналогично предыдущему случаю, если $v = a^+$ и $x = a^+$.

(г) В уловиях леммы для k будем брать такую цепочку $z = \omega\omega$, что $|\omega| = k$. Тогда для выполнения условия $uv^iwx^iy \in L$ необходимо, чтобы $|v + w| = |\omega|$, но в этом случае нарушается уловие $|vwx| = |\omega| + |x| = k + |x| \leqslant k$ и значит лемма не выполняется, а следовательно язык не КС.

12. Наидите грамматику порождающую $L(P)$, где

$$P = (\{q_0, q_1, q_2\}, \{a, b\}, \{Z_0, A\}, \delta, q_0, Z_0, \{q_2\})$$

и δ задается равенствами:

$$\begin{aligned}
&\delta(q_0, a, Z_0) = (q_1, AZ_0)\\
&\delta(q_0, a, A) = (q_1, AA)\\
&\delta(q_1, a, A) = (q_1, AA)\\
&\delta(q_1, a, A) = (q_1, A)\\
&\delta(q_2, a, A) = (q_1, \varepsilon)
\end{aligned}$$

13. Рассмотрим два следующих языка:
$L_1 = \{a^nb^{2n}c^m \mid n, m \geqslant 0\}$
$L_2 = \{a^nb^mc^{2m} \mid n, m \geqslant 0\}$

(а) покажите, что каждый из них являеся контекстно-свободным, построив для них КС-грамматики;

(б) укажите, является ли $L_1 \cap L_2$ КС-языком. Ответ обоснуйте.

Теория.

Грамматика G называется *контекстно-свободной*, если каждое правило из P имеет вид $A \to \alpha$, где $A \in N, \alpha \in (N \cup \Sigma)^*$.

Лемма (о разрастании для КС-языков). Пусть L — КС-язык. Тогда существует такая константа k, что если $|z| \geqslant k$ и $z \in L$, то цепочку z можно представить в виде $z = uvwxy$, где $vx \neq \varepsilon, |vwx| \leqslant k$ и $uv^iwx^iy \in L$ для всех $i \geqslant 0$.

Решение.

(а) Построим грамматику для языка L_1.

$$S \to XY$$
$$X \to aXbb \mid \varepsilon$$
$$Y \to cY \mid \varepsilon$$

Для L_2 грамматика будет выглядеть следующим образом.

$$S \to XY$$
$$X \to aX \mid \varepsilon$$
$$Y \to bYcc \mid \varepsilon$$

(б) $L_1 \cap L_2 = \{a^n b^{2n} c^{4n} \mid n \geqslant 0\}$

Данный язык не является КС, для доказательства воспользуемся леммой о разрастании.

1) если $v = a^+$, $v = b^+$ или $v = c^+$, то для uv^0wx^0y нарушается соотношение степеней и $uv^0wx^0y \notin L$
2) если $v = a^+b^+$, $v = a^+b^{2n}a^+$ или $v = b^+a^+$, то для uv^2wx^2y нарушется порядок следования символов и $uv^2wx^2y \notin L$

14. Найдте грамматику, не содержащую бесполезных символов и эквивалентную следующей грамматике:

$$S \to AB \mid CA$$
$$A \to a$$
$$B \to BC \mid AB$$
$$C \to aB \mid b$$

Терия.

Алгоритм "Устранение бесполезных символов".

1. Применив к G алгоритм "Непуст ли язык?", получить N_e. Положить $G_1 = (N \cap N_e, \Sigma, P_1, S)$, где P_1 состоит из правил множества P, содержащих только символы из $N_e \cup \Sigma$.

2. Применив к G_1 алгоритм "Устранение недостижимых символов", получить $G' = (N', \Sigma', P', S')$.

(1) **Алгоритм** "Непуст ли язык $L(G)$?"
 1. $N_0 = \varnothing$ и $i = 1$.
 2. $N_i = \{A \mid A \to \alpha \in P$ и $\alpha \in (N_{i-1} \cup \Sigma)^*\} \cup N_{i-1}$.
 3. Если $N_i \neq N_{i-1}$, то $i = i + 1$ и перейти к шагу 2, иначе $N_e = N_i$.
 4. Если $S \in N_e$: "ДА" иначе "НЕТ".

(2) **Алгоритм** "Устранение недостижимых символов"
 1. $V_0 = \{S\}$ и $i = 1$.
 2. $V_i = \{X \mid$ в P есть $A \to \alpha X \beta$ и $A \in V_{i-1}\} \cup V_{i-1}$.
 3. Если $V_i \neq V_{i-1}$, то $i = i + 1$ и перейти к шагу 2, в противном случае пусть $N' = V_i \cap N, \Sigma = V_i \cap \Sigma, P'$ состоит из множества правил P, содержащих только символы из $V_i, G' = (N', \Sigma', P', S)$.

Решение.
Выполняя алгоритм "Непуст ли язык?" получаем:

$$
\begin{aligned}
&N_0 = \varnothing \\
&N_1 = \{A, C\} \\
&N_2 = \{S, A, C\} \\
&N_3 = N_2 = \{S, A, C\} = N_e
\end{aligned}
$$

$$
\begin{aligned}
&S \to CA \\
&A \to a \\
&B \to b
\end{aligned}
$$

Теперь выполним алгорит "Устранение недостижимых символов":

$$
\begin{aligned}
&V_0 = \{S\} \\
&V_1 = \{C, A, S\} \\
&V_2 = \{b, a, C, A, S\} \\
&V_3 = V_2
\end{aligned}
$$

В итоге грамматика примет вид:

$$
\begin{aligned}
&S \to CA \\
&A \to a \\
&B \to b
\end{aligned}
$$

15. Рассмотрите следующую грамматику:

$$\begin{aligned}&S \to ASB \mid \varepsilon\\&A \to aAS \mid a\\&B \to SbS \mid A \mid bb\end{aligned}$$

(а) есть ли в грамматике бесполезные символы? Если да, то удалите их;
(б) удалите ε-правила;
(в) удалите цепные правила;

(г) приведите грамматику к нормальной форме Хомского.

Теория.

Определение. Назовем КС-грамматику $G = (N, \Sigma, P, S)$ *грамматикой без ε-правил*, если либо

1) P не содержит ε-правил, либо
2) есть точно одно ε-правило $S \to \varepsilon$ и S не встречается в правых частях правил из P.

Алгоритм "Преобразование в грамматику без ε-правил"

1) Необходимо построить $N_\varepsilon = \{A \mid A \in N \text{ и } A \Rightarrow^+_G \varepsilon\}$.
   (a) $N_0 = \varnothing$ и $i = 1$.
   (b) $N_i = \{A \mid A \to \varepsilon \text{ или } A \to \alpha, \alpha \in N^*_{i-1}\} \cup N_{i-1}$.
   (c) Если $N_i \neq N_{i-1}$, то $i = i + 1$ и перейти к шагу (b), иначе $N_\varepsilon = N_i$

2) Построить P' так:
   (a) Если $A \to \alpha_0 B_1 \alpha_1 B_2 \ldots \alpha_{k-1} B_k \alpha_k$ пренадлежит $P, k \geqslant 0$ и $B_i \in N_\varepsilon$ для $1 \leqslant i \leqslant k$, но ни один символ в цепочках $\alpha_j (0 \leqslant j \leqslant k)$ не пренадлежит N_ε, то включить в P' все правила вида
   $$A \to \alpha_0 X_1 \alpha_1 X_2 \ldots \alpha_{k-1} X_k \alpha_k,$$
   где X_i – либо B_i, либо ε, но не включать правило $A \to \varepsilon$.
   (б) Если $S \in N_\varepsilon$, включить в P' правила
   $$S' \to \varepsilon \mid S,$$
   где S' – новый символ, положить $N' = N \cup \{S'\}$. В противном случае положить $N' = N$ и $S' = S$.
3) Положить $G' = (N', \Sigma', P', S')$.

Определение. Цепные правила — правила вида $A \to B$.
Алгоритм "Устранение цепных правил"

1) Для каждого $A \in N$ построить $N_A = \{B \mid A \Rightarrow^* B\}$ следующим образом:

(a) $N_0 = \{A\}$ и $i = 1$.
(b) $N_i = \{C \mid B \to C$ пренадлежит P и $B \in N_{i-1}\} \cup N_{i-1}$.
(c) Если $N_i \neq N_{i-1}$, то $i = i + 1$ и повторить к шагу (b). В противном случе положить $N_A = N_i$

2) Построить P' так: если $B \to \alpha$ пренадлежит P и не является цепным правилом, включить в P' правило $A \to \alpha$ для всех таких A, что $B_i \in N_A$.
3) Положить $G' = (N', \Sigma', P', S)$.

Определение. КС-грамматика $G = (N, \Sigma, P, S)$ называется грамматикой в *нормальной форме Хомского*, если каждое правило из P имеет один из следующих видов:
- $A \to BC$, где A, B и C принадлежат N,
- $A \to a$, где $a \in \Sigma$,
- $A \to \varepsilon$, если $\varepsilon \in L(G)$, причем S не встречается в правых частях правил.

Алгоритм "Преобразование к нормльной форме Хомского."

1) Включить в P' каждое правило из P вида $A \to a$.
2) Включить в P' каждое правило из P вида $A \to BC$.
3) Включить в P' правило $S \to \varepsilon$, если оно было в P.
4) Для каждого правила из P вида $A \to X_1 \dots X_k$, где $k > 2$, включить в P' правила
$$\begin{array}{l} A \to X_1'[X_2 \dots X_k] \\ [X_2 \dots X_k] \to X_2'[X_3 \dots X_k] \\ . \\ . \\ . \\ [X_{k-2}X_{k-1}X_k] \to X_{k-2}'[X_{k-1}X_k] \\ [X_{k-1}X_k] \to X_{k-1}'X_k' \end{array}$$
где $X_i' = X_i$, если $X_i \in N$; X_i' — новый нетерминал, если $X_i \in \Sigma$; $[X_i \dots X_k]$ — новый нетерминал.
5) Для каждого правила из P вида $A \to X_1X_2$, где хотябы один из символов X_1 и X_2 принадлежит Σ, включить в P' правило $A \to X_1'X_2'$.
6) Для каждого нетерминала вида a', введённого на шагах (4) и (5), включить в P' правило $a' \to a$. Наконец, путь N' — это N вместе со всеми новыми нетерминалами, введёнными при построении P'.

Решение.

(a) Аналогично предыдущему примеру выполним алгоритм "Устранение бесполезных символов".

(1)

$$
\begin{aligned}
&N_0 = \varnothing \\
&N_1 = \{S, A, B\} \\
&N_2 = N_1 = \{S, A, B\} = N_e
\end{aligned}
$$

В результате грамматика примет вид:

$$
\begin{aligned}
&S \to ASB \mid \varepsilon \\
&A \to aAS \mid a \\
&B \to SbS \mid A \mid bb
\end{aligned}
$$

(2)

$$
\begin{aligned}
&V_0 = \{S\} \\
&V_1 = \{S, A, B\} \\
&V_2 = V_1
\end{aligned}
$$

В результате выполнения алгоритма грамматика примет вид:

$$
\begin{aligned}
&S \to ASB \mid \varepsilon \\
&A \to aAS \mid a \\
&B \to SbS \mid A \mid bb
\end{aligned}
$$

(б) Выполняя алгоритм “Преобразование в грамматику без ε-правил” получаем:

$$
\begin{aligned}
&N_0 = \varnothing \\
&N_1 = \{S\} \\
&N_2 = N_1 = \{S\} = N_\varepsilon
\end{aligned}
$$

грамматика примет вид:

$$
\begin{aligned}
&S' \to S \mid \varepsilon \\
&S \to ASB \mid AB \\
&A \to aAS \mid aA \mid a \\
&B \to SbS \mid bS \mid Sb \mid b \mid A \mid bb
\end{aligned}
$$

(в) Выполним алгорим “Устранение цепных правил”:

$$
S'\colon\quad
\begin{aligned}
&N_0 = \{S'\} \\
&N_1 = \{S, S'\} \\
&N_2 = N_1 = \{S, S'\} = N_{S'}
\end{aligned}
$$

$$
S\colon\quad
\begin{aligned}
&N_0 = \{S\} \\
&N_1 = N_0 = \{S\} = N_S
\end{aligned}
$$

$$
A\colon\quad
\begin{aligned}
&N_0 = \{A\} \\
&N_1 = N_2 = \{A\} = N_A
\end{aligned}
$$

$$
\begin{aligned}
&N_0 = \{B\}\\
B\colon\ &N_1 = \{A, B\}\\
&N_2 = N_1 = \{A, B\} = N_B
\end{aligned}
$$

В итоге грамматика примет вид:

$$
\begin{aligned}
&S' \to ASB \mid AB \mid \varepsilon\\
&S \to ASB \mid AB\\
&A \to aAS \mid aA \mid a\\
&B \to SbS \mid bS \mid Sb \mid b \mid aAS \mid aA \mid a \mid bb
\end{aligned}
$$

(г) Используя алгоритм "Преобразование к нормльной форме Хомского." грамматика изменится так:

$$
\begin{aligned}
&S' \to A\langle SB\rangle \mid AB \mid \varepsilon\\
&S \to A\langle SB\rangle \mid AB\\
&A \to a'\langle AS\rangle \mid a'A \mid a\\
&B \to S\langle bS\rangle \mid b'S \mid Sb' \mid b \mid a'\langle AS\rangle \mid a'A \mid a \mid b'b'\\
&\langle SB\rangle \to SB\\
&\langle AB\rangle \to AB\\
&\langle AS\rangle \to AS\\
&\langle bS\rangle \to b'S\\
&\langle Sb\rangle \to Sb'\\
&a' \to a\\
&b' \to b
\end{aligned}
$$

16. Устраните левую рекурсию в следующей грамматике:

$$
\begin{aligned}
S &\to AA \mid 0\\
A &\to SS \mid 1
\end{aligned}
$$

Теория. *Определение.* Нетерминал A КС-грамматики $G = (N, \Sigma, P, S)$ называется *рекурсивным*, если $A \Rightarrow^+ \alpha A \beta$ для некоторых α и β. Если $\alpha = \varepsilon$, то A называется *леворекурсивным*. Аналогично, если $\beta = \varepsilon$, то A называется *праворекурсивным*. *Определение.* Грамматика, имеющая хотя бы один леворекурсивный нетерминал, называется *леворекурсивной*. Аналогично, грамматика, имеющая хотя бы один праворекурсивный нетерминал, называтеся *праворекурсивной*. **Алгоритм.** "Устранение левой рекурсии".

1. Пусть $N = \{A_1, \ldots, A_n\}$. Преобразуем G таким образом, чтобы в правиле $A_i \to \alpha$ цепочка α начиналась либо с терминала, с такого A_j, что $j > i$. С этой целью положим $i = 1$
2. Пусть множество A_i-правил — это $A_i \to A_i\alpha_1 \mid \ldots \mid A_i\alpha_m \mid \beta_1 \mid \ldots \mid \beta_p$, где ни одна из цепочек β_j не начинается с A_k, если $k \leqslant i$. Заменим A_i-правила правилами:

$$A \to \beta_1 \mid \ldots \mid \beta_p \mid \beta_1 A'_i \mid \ldots \mid \beta_p A'_i$$
$$A'_i \to \alpha_1 \mid \ldots \mid \alpha_m \mid \alpha_1 A'_i \mid \ldots \mid \alpha_m A'_i$$

где A'_i — новый нетерминал. Правые части всех A_i-правил начинаются теперь с терминала или с A_k для некоторого $k > j$.

3. Если $i = n$, полученую грамматику G' считать результатом и остановиться. В противном случае положить $i = i + 1$ и $j = 1$.
4. Заменить каждое правило вида $A_i \to A_j\alpha$ правилами $A_i \to \beta_1\alpha \mid \ldots \mid \beta_m\alpha$, где $A_j \to \beta_1 \mid \beta_m$ — все A_j-правила. Так как правая часть каждого A_j-правила начинается уже с терминала или с A_k для $k > j$, то и правая часть каждого A_i-правила будет теперь обладать этим свойством.
5. Если $j = i - 1$, нужно перейти к шагу (2). В противном случае положить $j = j + 1$ и перейти к шагу (4).

Решение. Составим множество N таким образом:

$$N = \{S, A\}$$

Выполняя описанный алгоритм сначала получим:

$$S \to AA \mid 0$$
$$A \to AAS \mid 0S \mid 1$$

И итоге грамматика примет вид:

$$S \to AA \mid 0$$
$$A \to 0S \mid 1 \mid 0SA' \mid 1A'$$
$$A' \to AS \mid ASA'$$

18. Постройте МП-автоматы, допускающие следующие языки. Можно использовать как допуск по заключительному состоянию, так и по пустому магазину.

(а) $\{0^n 1^n \mid n \geqslant 1\}$;

(б) множество всех цепочек из 0 и 1, в префиксах которых количество символов 1 не больше количества символов 0;

(в) множество всех цепочек из 0 и 1 с одинаковыми количествами символов 0 и 1.

Решение.

(а) Построим МПА, определяющий язык $\{0^n 1^n \mid n \geqslant 1\}$. Пусть $P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z, 0\}, \delta, q_0, Z, \{q_0\})$,

$$\delta(q_0, 0, Z) = \{(q_1, 0Z)\},$$
$$\delta(q_1, 0, 0) = \{(q_1, 00)\},$$
$$\delta(q_1, 1, 0) = \{(q_2, \varepsilon)\},$$
$$\delta(q_2, 1, 0) = \{(q_2, \varepsilon)\},$$
$$\delta(q_2, \varepsilon, Z) = \{(q_0, \varepsilon)\}.$$

Для входной цепочки 0011 автомат P проделает такую последовательность тактов:

$$\begin{aligned}(q_0, 0011, Z) &\vdash (q_1, 011, 0Z),\\ &\vdash (q_1, 11, 00Z),\\ &\vdash (q_2, 1, 0Z),\\ &\vdash (q_2, \varepsilon, Z),\\ &\vdash (q_0, \varepsilon, \varepsilon).\end{aligned}$$

Вообще можно показать, что

$$\begin{aligned}(q_0, 0, Z) &\vdash (q_1, \varepsilon, 0Z),\\ (q_1, 0^i, 0Z) &\vdash^i (q_1, \varepsilon, 0^{i+1}Z),\\ (q_1, 1, 0^{i+1}Z) &\vdash (q_2, \varepsilon, 0^iZ),\\ (q_2, 1^i, 0^iZ) &\vdash^i (q_2, \varepsilon, Z)\\ (q_2, \varepsilon, Z) &\vdash (q_0, \varepsilon, \varepsilon)\end{aligned}$$

Объединяя всё это, получаем для $n \geqslant 1$

$$\begin{aligned}(q_0, 0^n1^n, Z) &\vdash^{2n+1} (q_0, \varepsilon, \varepsilon),\\ (q_0, \varepsilon, Z) &\vdash^1 (q_0, \varepsilon, \varepsilon),\end{aligned}$$

(б) Исходя из условия, можно сделать вывод, что цепочка обязательно начинается с символа 0. Используя этот факт и обеспечивая постоянное превосходство в префиксах 0 над 1. В итоге получим следующий автомат:
$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z, 0\}, \delta, q_0, Z, \{q_2\})$,

$$\begin{aligned}\delta(q_0, 0, Z) &= \{(q_1, 0Z)\},\\ \delta(q_1, 0, 0) &= \{(q_1, 00)\},\\ \delta(q_1, 0, Z) &= \{(q_1, 0Z)\},\\ \delta(q_1, 1, 0) &= \{(q_1, \varepsilon)\},\\ \delta(q_1, \varepsilon, 0) &= \{(q_1, \varepsilon)\},\\ \delta(q_1, \varepsilon, Z) &= \{(q_2, \varepsilon)\},\end{aligned}$$

Для входной цепочки 00110100 автомат P проделает такую последовательность тактов:

$$\begin{aligned}(q_0, 00110100, Z) &\vdash (q_1, 0110100, 0Z),\\ &\vdash (q_1, 110100, 00Z),\\ &\vdash (q_1, 10100, 0Z),\\ &\vdash (q_1, 0100, Z),\\ &\vdash (q_1, 100, 0Z),\\ &\vdash (q_1, 00, Z),\\ &\vdash (q_1, 0, 0Z),\\ &\vdash (q_1, \varepsilon, 00Z),\\ &\vdash (q_1, \varepsilon, 0Z),\\ &\vdash (q_1, \varepsilon, Z),\\ &\vdash (q_2, \varepsilon),\end{aligned}$$

(в) Построим автомат, который проверяет равенство 0 и 1 во входной цепочке
$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z, 0, 1\}, \delta, q_0, Z, \{q_2\}),$

$$\delta(q_0, 0, Z) = \{(q_1, 0Z)\},$$
$$\delta(q_0, 1, Z) = \{(q_1, 1Z)\},$$
$$\delta(q_1, 0, Z) = \{(q_1, 0Z)\},$$
$$\delta(q_1, 1, Z) = \{(q_1, 1Z)\},$$
$$\delta(q_1, 0, 0) = \{(q_1, 00)\},$$
$$\delta(q_1, 0, 1) = \{(q_1, \varepsilon)\},$$
$$\delta(q_1, 1, 0) = \{(q_1, \varepsilon)\},$$
$$\delta(q_1, 1, 1) = \{(q_1, 11)\},$$
$$\delta(q_1, \varepsilon, Z) = \{(q_2, \varepsilon)\},$$

Для входной цепочки 101001 автомат P проделает такую последовательность тактов:

$$\begin{aligned}(q_0, 101001, Z) &\vdash (q_1, 01001, 1Z),\\ &\vdash (q_1, 1001, Z),\\ &\vdash (q_1, 001, 1Z),\\ &\vdash (q_1, 01, Z),\\ &\vdash (q_1, 1, 0Z),\\ &\vdash (q_1, \varepsilon, Z),\\ &\vdash (q_2, \varepsilon),\end{aligned}$$