

3. Первый способ - можем выполнить замену всех вхождений символа  $B$  в каждом правиле, заменив его  $2^k$  новыми правилами при условии что  $B$  входило  $k$  раз в исходное правило. После этого правило  $B \rightarrow D$  можно исключить, если в грамматике есть правила  $D \rightarrow D$ , исключить их. Если в списке правил есть одинаковые правила, исключить их. Также важно отметить, что нетерминал  $B$  не должен быть начальным нетерминалом. Если он таковым является, то в правиле  $B \rightarrow D$  можно выполнить замену края. Второй способ - замена края. В правиле  $B \rightarrow D$  будем заменять  $D$  на правые части правил, у которых в левой части находится  $D$ , а само правило  $B \rightarrow D$  удалим.

4. Наличие  $\varepsilon$ -правил и цепных правил может привести к ухудшению времени работы алгоритма (в худшем случае экспоненциально) и привести к циклам. Так например для следующей грамматики с цепными правилами:

1.  $A \rightarrow B$
2.  $B \rightarrow A$
3.  $B \rightarrow b$

После выполнения алгоритма придём к грамматике следующего вида:

1.  $A \rightarrow B$
2.  $B \rightarrow B$
3.  $B \rightarrow b$

1.  $A \rightarrow B$
2.  $B \rightarrow bB'$
3.  $B' \rightarrow B'$
4.  $B' \rightarrow \varepsilon$

В итоге после преобразований получили саморекурсивное правило 3. При наличии  $\varepsilon$ -правил также рискуем получить правило, подобное  $B \rightarrow B$ , которое приведёт к заикливанию или несоответствию грамматики условию, например:

1.  $A \rightarrow \varepsilon$
2.  $B \rightarrow AB$
3.  $B \rightarrow b$

1.  $A \rightarrow \varepsilon$
2.  $B \rightarrow B$
3.  $B \rightarrow b$

1.  $A \rightarrow \varepsilon$
2.  $B \rightarrow bB'$
3.  $B' \rightarrow B'$
4.  $B' \rightarrow \varepsilon$

Для грамматики  $G_1$  можно применить алгоритм, так как подобных случаев возникнуть не может, однако для гарантированной работы алгоритма необходимо избавляться от  $\varepsilon$ - и цепных правил.

5. Грамматика неэквивалентна, так как начального нетерминала  $T$  нет. Исправил.

6. Были допущены ошибки при выполнении 1 пункта, также была выполнена замена на неединичный нетерминал, там где требовался одиночный. Исправил.

7. Правила грамматики в НФГ содержат только правила вида  $A \rightarrow t\alpha$ , где  $t$  - терминал, а  $\alpha$  - цепочка нетерминалов, возможно пустая. Нет. Нет. Исправил.

8. Для правила вида  $A \rightarrow B\alpha$ , где  $B$  - нетерминал,  $\alpha$  - цепочка терминалов и нетерминалов, возможно пустая, выполним замену и левую факторизацию. Если хоть одно из получившихся правил содержит правую часть  $B\alpha$  - алгоритм заиклился. Иначе - продолжаем выполнять замену и левую факторизацию для правил и отслеживаем появление  $B\alpha$  в правой части новых правил.

9. Исправил.

10. Символ  $D$  в правилах 1, 2 занимает крайнюю правую позицию. В правиле 3 - нет. Символ  $b$  в правилах 3, 4, 5 занимает крайнюю правую позицию. В правиле 5 - нет. Исправил.