**Методический материал для лабораторной работы № 5 . Устранить из КС-грамматики цепные правила и устранить левую рекурсию**

**Алгоритм для удаления длинных правил.**

С каждым длинным правилом *A*→*a*1*a*2…*ak* , *k*>2, *ai* ∈ Σ∪*N*

проделаем следующее:

* Добавим в грамматику *k*−2 новых нетерминалов *B*1,*B*2,…*Bk*−2.
* Добавим в грамматику *k*−1 новое правило:

*A*→*a*1*B*1

*B*1→*a*2*B*2

*B*2→*a*3*B*3

…

*Bk*−2→*ak*−1*ak*

  Удалим из грамматики правило *A*→*a*1*a*2…*ak*.

**Рассмотрим, как описанный алгоритм будет работать на следующей грамматике:**

*S*→*AB*

*A*→*aBcB*

*B*→*def*

Для правила ***A*→*aBcB***

вводим 2 новых нетерминала *A*1, *A*2 и 3 новых правила:

*A*→*aA*1

*A*1→*BA*2

*A*2→*cB*

Для правила ***B*→*def***

вводим 1 новый нетерминал *B*1 и 2 новых правила:

*B*→*dB*1

*B*1→*ef*

В итоге полученная грамматика G′ будет иметь вид:

*S*→*AB*

*A*→*aA*1

*A*1→*BA*2

*A*2→*cB*

*B*→*dB*1

*B*1→*ef*

**Алгоритм удаление цепных правил из грамматики**

**Цепное правило** (*unit rule*) — правило вида *A*→*B*, где *A* и *B* — нетерминалы.

Пусть G — КС- грамматика, содержащая цепные правила. Требуется построить эквивалентную грамматику G′, не содержащую цепных правил. Задача удаления цепных правил из грамматики возникает при попытке её приведения к нормальной.

**Цепная пара** (*unit pair*) — упорядоченная пара (*A*,*B*), в которой *A*⇒∗*B*, используя только цепные правила.

1. Найти все цепные пары в грамматике G
2. Для каждой цепной пары (*A*,*B*) добавить в грамматику G′ все правила вида *A*→*α*, где *B*→*α* — нецепное правило из G
3. Удалить все цепные правила

**Примечание: как искать цепные правила**

Найти все цепные пары можно по индукции:

**Базис.** (*A*, *A*) — цепная пара для любого нетерминала, так как *A*⇒∗*A* за ноль шагов.

**Индукция.** Если пара (*A*,*B*) - цепная, и есть правило *B*→*C*, то (*A*,*C*)

— цепная пара.

**Рассмотрим, как описанный алгоритм будет работать на следующей грамматике:**

Рассмотрим грамматику:

*A*→*B*|*a*

*B*→*C*|*b*

*C*→*DD*|*c*

, в которой есть два цепных правила *A*→*B* и *B*→*C*

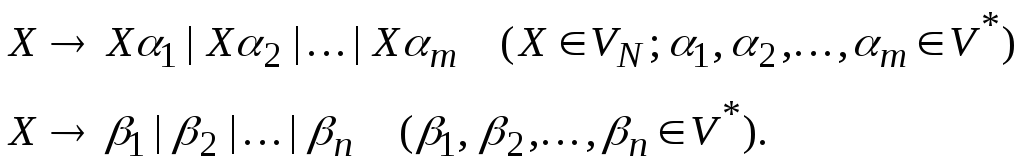
* Для каждого нетерминала создадим цепную пару. Теперь множество цепных пар будет состоять из (*A*,*A*), (*B*,*B*), (*C*,*C*) и (*D*,*D*)
* Рассмотрим цепное правило *A*→*B*. Так как существует цепная пара (*A*,*A*), второй элемент которой совпадает с левым нетерминалом из правила,  
  добавим в множество пару (*A*,*B*) у которой первый элемент такой же как у найденной, а второй равен правому нетерминалу из текущего правила.
* Повторим второй пункт для правила *B*→*C* и пары (*B*,*B*). Теперь множество цепных пар будет состоять из (*A*,*A*), (*B*,*B*), (*C*,*C*), (*D*,*D*), (*A*,*B*) и (*B*,*C*)
* Повторим второй пункт для правила *B*→*C* и пары (*A*,*B*), и получим множество {(*A*,*A*),(*B*,*B*),(*C*,*C*),(*D*,*D*),(*A*,*B*),(*B*,*C*),(*A*,*C*)}
* д ля каждой пары добавим в G′ новые правила:
* *A*→*b* для (*A*,*B*)
* *A*→*c* и *A*→*DD* для (*A*,*C*)
* *B*→*c* и *B*→*DD* для (*B*,*C*)

 Оставшиеся цепные пары новых правил не добавят.

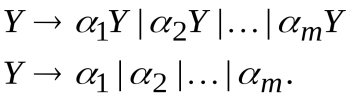
## Алгоритм Устранение прямой левой рекурсии в КС-грамматике.

Выход: Эквивалентная КС-грамматика G1 = (VT ,  V1N , P1  ,S1 ) без прямой левой рекурсии, т.е. без правил видаhttps://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-rvwowq.png

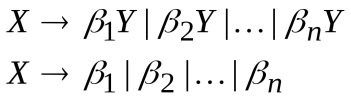
**Шаг 1.** Вывести из грамматики все правила для рекурсивного нетерминала https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-kX3CAZ.png:

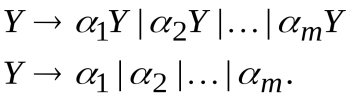


**Шаг 2.** Внести новый нетерминал https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-pV_pup.pngтак, чтобы он описывал любой «хвост» строки, порождаемой рекурсивным нетерминаломhttps://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-Ib04JC.png:



**Шаг 3.** Заменить в рекурсивном правиле для https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-iRtiul.pngправую часть, используя новый нетерминал и все нерекурсивные правила дляhttps://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-S3mpaJ.pngтак, чтобы генерируемый язык не изменился:





**Шаг 4.** Пополнить множество нетерминалов грамматики новым нетерминалом https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-7buhdm.png. Пополнить множество правил грамматики правилами, полученными на шаге 3.

**Шаг 5.** Повторить действия шагов 1-4 для всех рекурсивных нетерминалов грамматики, после чего полученные множества нетерминалов и правил принять в качестве V1N и P1

**Пример.** Данаграмматика https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-9IqaCt.pngс правиламиhttps://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-6pMvJY.pnghttps://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-qArPGt.png.

Шаг алгоритма. Действия и результат.

1. https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-sllutX.png ( применяем шаг 1 )

2. https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-fsaeZ7.png ( применяем шаг 2 )

3. https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-fsyioS.pnghttps://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-VRHjJU.png ( применяем шаг 3)

4. *VN =*{*S,A,B,C,Z*}, P1=

https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-S1T5mp.png (применяем шаг 4)

После устранения прямой левой рекурсии получим эквивалентную грамматику

G1 = ( {a, b, c, d, z}, {S, A, B, C, Z}, P1 , S) с правилами P1 :

https://studfile.net/html/2706/601/html_PojTlE4jy2.GOVa/img-VV34tG.png