

6. Распознавание КС-языков методом динамического программирования, нормальная форма Хомского.

Нормальная форма Хомского (ХНФ)

Нормальная форма Хомского — это канонический вид контекстно-свободной грамматики, в котором правила вывода максимально стандартизированы.

- **Определение:** КС-грамматика считается находящейся в нормальной форме Хомского, если она является **λ -свободной** (не содержит аннулирующих правил, кроме, возможно, правила для аксиомы), а все её неаннулирующие правила вывода имеют строго один из двух видов:
 1. $A \rightarrow BC$ — нетерминал порождает ровно два нетерминала;
 2. $A \rightarrow a$ — нетерминал порождает ровно один терминал.
- **Эквивалентность:** любая КС-грамматика может быть преобразована в эквивалентную ей грамматику в нормальной форме Хомского.

Алгоритм Кока-Янгера-Касами

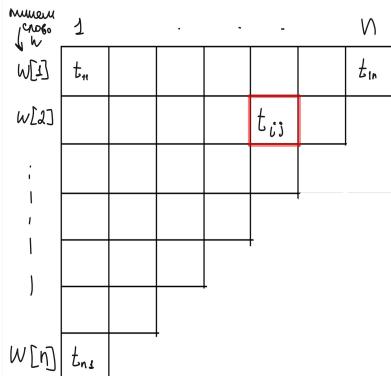
Вход: L — КСЯ, G — в ХНФ, $L(G) = L$ и самое главное w — строка длины n .

Выход: $w \in L$?

(Можно ли данной грамматике получить слово w ?)

Алгоритм

Строим верхнетреугольную матрицу размера $n \times n$



Клетка t_{ij} — это множество нетерминалов из которых выводится $w[i, i + j - 1]$ — подслово начиная с i позиции длины j .

Если t_{1n} (правая верхняя) содержит аксиому грамматики, то $w \in L$

В ХНФ правила имеют вид:

1. $A \rightarrow BC$
2. $A \rightarrow a$

Значит $t_{11}, t_{12}, \dots, t_{n1}$ заполняется непосредственно по правилам типа (2). То есть в клетку записываем нетерминалы из которого сразу выводится терминал $w[i]$.

Для заполнения t_{ij} нужно размотреть все возможные разбиения $w[i, i + j - 1]$ на два не пустых подслова.

Если в t_{ik} есть нетерминал B в $t_{i+k, j-k}$ есть нетерминал C ($A \rightarrow BC \in P$), то в t_{ij} поместить A $k \in \{1, \dots, j-1\}$

Если сложно то, вот ответ от нейронки

2. Алгоритм Кока-Янгера-Касами

- **Предпосылка:** Грамматика G дана в НФХ. Стока $w = a_1 a_2 \dots a_n$ длины n .

- **Идея:** Построить таблицу D размером $n \times n$, где элемент $D[i][j]$ — **множество нетерминалов**, из которых выводится подстрока w длиной j , начинающаяся с позиции i (индексация может немного варьироваться).
- **Алгоритм (по шагам):**
 - Инициализация (база ДП, подстроки длины 1):**
Для каждой позиции i от 1 до n :
 $D[i][1] = \{ A \mid (A \rightarrow a_i) \in P \}$ (все нетерминалы, порождающие терминал a_i).
 - Заполнение таблицы (переход ДП, подстроки длины $L > 1$):**
Для длины L от 2 до n :
Для начала i от 1 до $n - L + 1$:
 - $j = i + L - 1$ (конец подстроки).
 - Множество $D[i][L]$ изначально пусто.
 - Для всех точек разбиения k от i до $j-1$:
 - Рассматриваем разбиение подстроки на две части: $(i..k)$ и $(k+1..j)$.
 - Пусть $X \in D[i][k-i+1]$, $Y \in D[k+1][j-k]$.
 - Для **всех** правил $A \rightarrow XY$ в грамматике добавляем A в $D[i][L]$.
 - Завершение:** Стока w принадлежит $L(G)$ **тогда и только тогда, когда** аксиома S содержится в множестве $D[1][n]$.
- **Сложность:** $O(n^3 * |G|)$, где $|G|$ — размер грамматики. Это полиномиально, в отличие от экспоненциального перебора.

Пример(с пары)

Слово $w = aacbc$

x — это пустая клетка/пустое множество.

Рассмотрим грамматику

$$S \rightarrow A'A|BB'|SS$$

$$A \rightarrow A'A|A'D|c$$

$$D \rightarrow CB'$$

$$B \rightarrow BB'|A'D|c$$

$$C \rightarrow A'D|c$$

$$A' \rightarrow a$$

$$B' \rightarrow b$$

и цепочку $aacbc$.

Разбор(шаги):

1. Заполняем первый столбец по грамматике (если символ выводится сразу за 1 шаг, добавляем левую часть правила в ячейку)

	1	2	3	4	5	6
1	A'					
2	A'					
3	ABC					
4	B'					
5	ABC					
6	B'					

2. Смотрим есть ли рядом стоящие $A'A'$ в правых частях? Нет -> пустое мно-во.

	1	2	3	4	5	6
1	A'	-				
2	A'					
3	ABC					
4	B'					
5	ABC					
6	B'					

3. Смотрим есть ли $A'A$ или $A'B$ или $A'C$ в правых частях? Да, значит левые части (нетерминалы) правил вывода добавляем в ячейку.

	1	2	3	4	5	6
1	A'	-				
2	A'	SA				
3	ABC					
4	B'					
5	ABC					
6	B'					

4. И так далее до конца второго столбца.

	1	2	3	4	5	6
1	A'	-				
2	A'	SA				
3	ABC	SDB				
4	B'	-				
5	ABC	SDB				
6	B'					

5. 3-ий столбец, заполняем по похожему правилу (Обрати внимание какие ячейки мы сейчас берём)

Рассмотрим грамматику

$$S \rightarrow A'A|BB'|SS$$

$$A \rightarrow A'A|A'D|c$$

$$D \rightarrow CB'$$

$$B \rightarrow BB'|A'D|c$$

$$C \rightarrow A'D|c$$

$$A' \rightarrow a$$

$$B' \rightarrow b$$

	1	2	3	4	5	6
1	A'	-	SA			
2	A'	SA				
3	ABC	SDB				
4	B'	-				
5	ABC	SDB				
6	B'					

Если в паре нетерминалов при переборе, встретилось – значит можно скипнуть и перебирать дальше

	1	2	3	4	5	6
1	A'	-	SA			
2	A'	SA				
3	ABC	SDB				
4	B'	-				
5	ABC	SDB				
6	B'					

6. Шагаем вот так (цифры это какие ячейки будут участвовать на текущей итерации):

	1	2	3	4	5	6
1	A'	-	SA	SA		
2	A'	SA	ABC	-		
3	$\overset{1}{ABC}$	$\overset{2}{SDB}$	$\overset{3}{-}$	S		
4	B'	-	-	$\overset{1}{-}$		
5	ABC	$\overset{2}{SDB}$				
6	$\overset{3}{B'}$					

шаги для
этой клетки

7. Итог аксиома выводится из w.

	1	2	3	4	5	6
1	A'	-	SA	SA	-	S
2	A'	SA	ABC	-	-	
3	ABC	SDB	-	S		
4	B'	-	-			
5	ABC	$\overset{2}{SDB}$				
6	B'					