Компьютерная дингвистика
Конспекты по предмету

Милов Данила

Оглавление

Глава 1

Конечные преобразователи

1.1 Конечные преобразователи

Определение 1.1.1: Конечный преобразователь

Шестёрка $S = (Q, \Sigma, \Delta, q_0, \delta, F)$, где:
Q — множество состояний,
Σ — входной алфавит,
$\delta \subset Q \times \Sigma^* \times \delta \times Q$ — конечное множество переходов(программа),
q_0 — начальное состояние,
$F \subseteq Q$ — множество закл. состояний

Определение 1.1.2: Конфигурация

Тройка (q, u, v), где $q \in Q, u \in \Sigma^*, v \in \Delta^*$

Определение 1.1.3: Переход за один шаг($\vdash$)

$(q, u, v) \vdash_S (p, x, y)$, если существует $w \in \Sigma^*, t \in \Delta^*$:
$u = wx, y = vt$ и $q, w \to p, v \in \delta$

Определение 1.1.4: Отношение, вычисляемое S

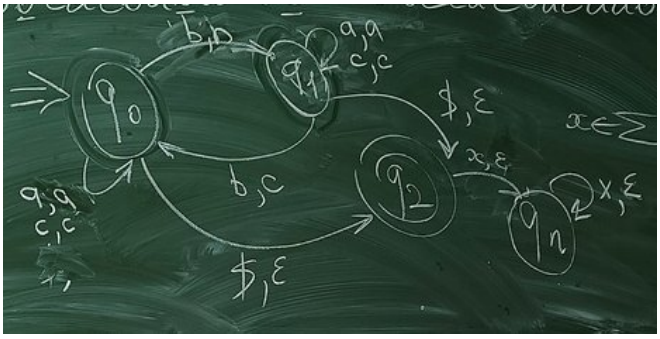
$R(S) = \{(x, y) : (q_0, x, \varepsilon) \vdash_S^* (q_f, \varepsilon, y)$ для некоторого $q_f \in F\}$

Определение 1.1.5: Детерминированный КП

Шестёрка $S = (Q, \Sigma, \Delta, \delta, q_0, F)$, где $\delta : Q \times \Sigma \to Q \times \Delta^*$

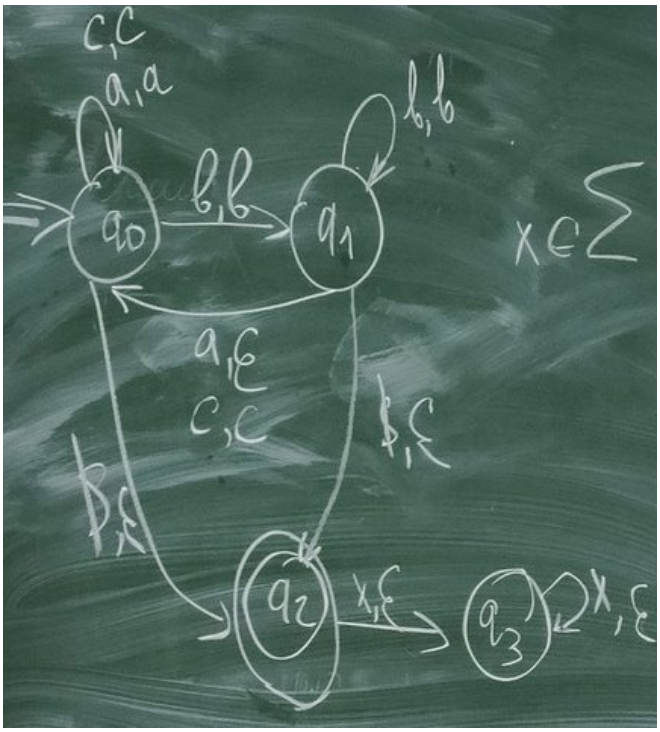
Вопрос 1: Построить КП

$\Sigma = \{a, b, c, \$\}$
Заменить все чётные символы b на c(нумерация с 1).



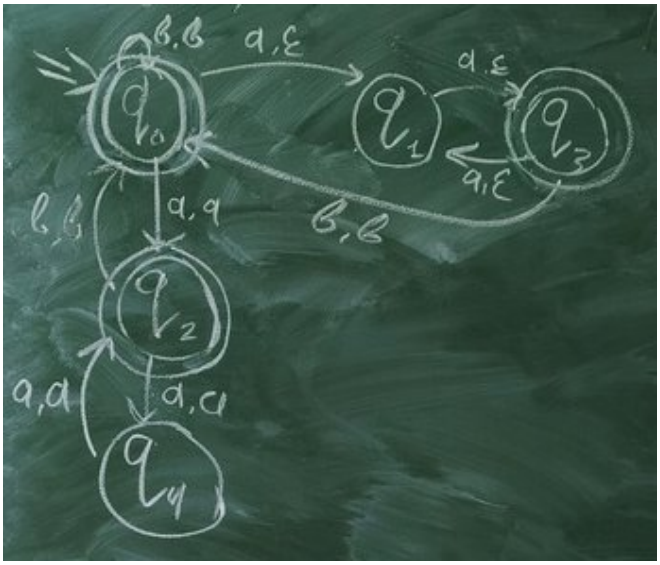
Вопрос 2: Построить КП

$\Sigma = \{a, b, c, \$\}$
Стереть все символы a, стоящие непосредственно после символов b.



Вопрос 3: Построить КП

$\Sigma = \{a, b\}$
Удалить из x все блоки символов a чётной длины.



Note:
Домашнее задание:
№134(в,г),133,137

1.2 Морфологический анализ

Задачи морфологического анализа:

1. Дано слово, поставить его в нужную форму.
2. Дано слово, найти его начальную форму.

Предложение 1.2.1 Рассматриваемые характеристики

- N — существительное(noun)
- V — глагол(verb)
- SG — единственное число(singular)
- PL — множественное число(plural)

Пример 1.2.1

cat+N+PL $\Rightarrow$ cats
cats $\Rightarrow$ cat+N+PL

Определение 1.2.1: Морфологические правила

Определяют, как образуются формы.
Пример: английский язык, множественное число = ед. число + s.

Определение 1.2.2: Орфографические правила

Определяют, как меняется написание.
Пример — англ. язык, мн. число:
Слово заканчивается на s $\longrightarrow$ вставить e.
Слово заканчивается на y $\longrightarrow$ заменить на ie.
Слово является исключением $\longrightarrow$ особый случай.

Определение 1.2.3: Основа

«Главная часть» слова.

Определение 1.2.4: Аффексы

Части слова, приписываемые к основе:

1. Префиксы: пишутся спереди.
2. Суффиксы: пишутся сзади.
3. Инфиксы: пишутся внутри.
4. Циркумфиксы: спереди и сзади.

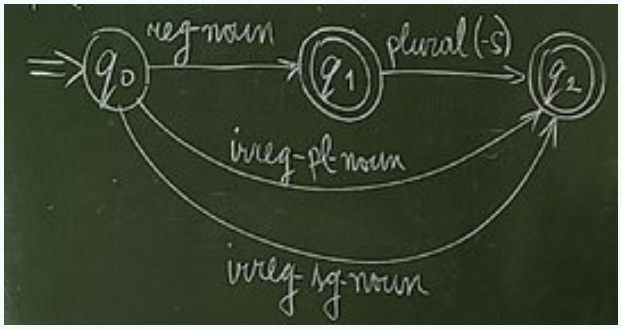
Note:

Для морфологического анализатора необходимы:

1. Словарь — список основ и аффексов.
2. Морфотактика — правила объединения морфем.
3. Орфографические правила.

Пример 1.2.2 (Автомат, проверяющий правильность определения множественного числа)

reg-noun — правильные существительные;
irreg-sg-noun — единственное число «неправильных» существительных;
irreg-pl-noun — множественное число «неправильных» существительных;
plural(-s) — s.



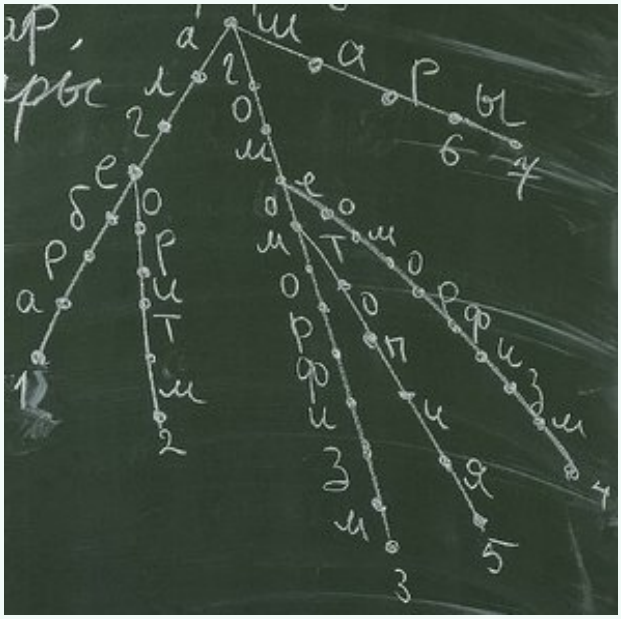
Определение 1.2.5: Префиксное дерево

Есть множество слов $S = \{S_1, \ldots, S_n\}$.

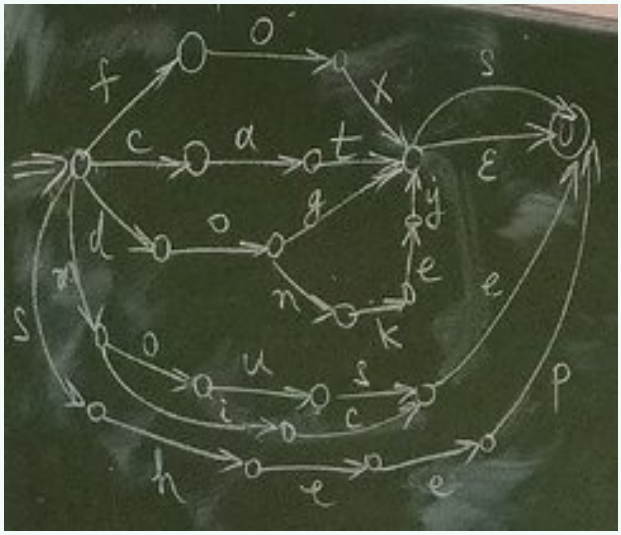
1. Рёбра дерева помечены символами.
2. Рёбра, выходящие из одной вершины, помечены разными символами.
3. На пути из корня в листья написаны слова из S.
4. Для каждого S_i существует вершина v такая, что на пути из корня в v написано S_i(v отмечены i).

Пример 1.2.3 (Префиксное дерево)

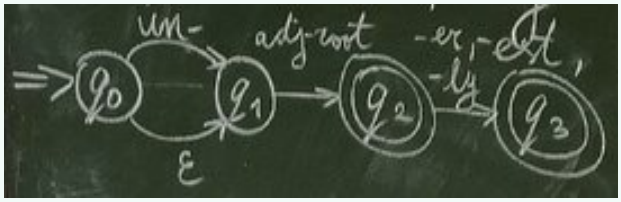
Префиксное дерево для слов: алгебра, алгоритм, гомоморфизм, гомотопия, гомеоморфизм, шар, шары.



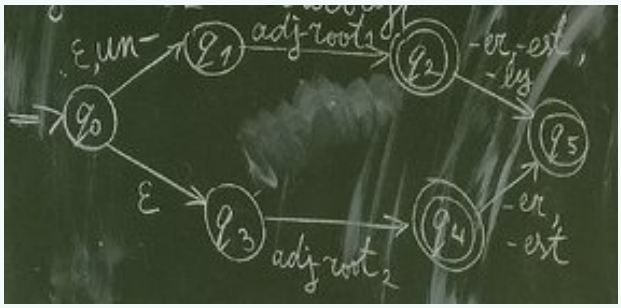
Пример 1.2.4 (Автомат, распознающий множественное число слов)



Пример 1.2.5 (Автомат для распознавания прилагательных(1))
adj-root — корень прилагательного
un-, -er, -est, -ly — аффексы слов.



Пример 1.2.6 (Автомат для распознавания прилагательных(2))
adj-root$_1$ — корень прилагательного, который может употребляться с приставками
adj-root$_2$ — корень прилагательного, который не может употребляться с приставками
un-, -er, -est, -ly — аффексы слов.
Это недетерминированный вариант автомата(угадываем переход по пустому слову в q_1 или q_3).

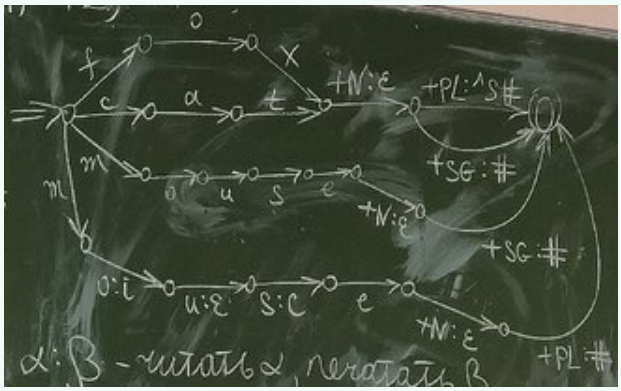


Предложение 1.2.2 Три уровня конечного преобразователя

1. Лексический: слово и его признаки.(f | o | x | +N | +PL)
2. Промежуточный: морфемы и доп.метки(f | o | x | ^ | s)
3. Поверхностный: слово(f | o | x | e | s)

Пример 1.2.7 (Преобразователь слов с лексического уровня в промежуточный)

Обозначения:
$\alpha : \beta$ — читаем α, пишем β,
α — читаем α и печатаем α.



Пример 1.2.8 (Примеры правил)

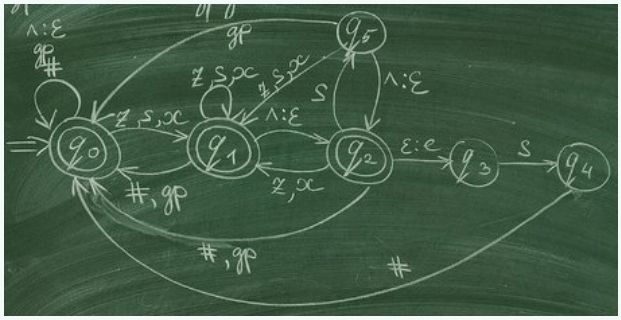
1. Замена Y. Замена -y на -ie перед -s, -i перед -ed.($try \Rightarrow tries$)
2. Удвоение согласной перед -ing/-ed.($bed \Rightarrow begging$)
3. Удаление -e перед -ing,-ed.(make⇒making)
4. Вставка -e после -s,-z,-x,-ch,-sh перед -s.(watch⇒watches)

Предложение 1.2.3 Запись правил

$a \Rightarrow b/c\_d$ — Заменить a на b между c и d.
a,b,c,d — строки.
$cad \Rightarrow cbd$

Пример 1.2.9 (КП для распознавания особых случаев образования множественного числа)

Символы: $s, z, x, \hat{}, \#$, др — любой другой символ.



Определение 1.2.6: Обратное отношение

$S = (Q, \Sigma, \Delta, \delta, q_0, F), R(S) \subseteq \Sigma^* \times \Delta$ — конечный преобразователь.
$S' = (Q, \Delta, \Sigma, \delta', q_0, F)$ — конечный преобразователь, для которого множества входных и выходных данных поменяны местами.
А программа имеет вид:
$q, x \to p, y \in \delta \Leftrightarrow q, y \to p, x \in \delta'$
Тогда:
$R(S') = (R(S))^{-1}$
$(x, y) \in R(S') \Leftrightarrow (y, x) \in R(S)$

Предложение 1.2.4 Морфологический анализ слов

Конечный преобразователь из примера выше может быть использован не только для того, чтобы проверить правильность образования формы множественного числа, но и для того, чтобы на его основе построить автомат S', обратный данному, восстанавливающий второй уровень слова по третьему.

Пример: выделим морфемы в словах cats и foxes. Для этого построим автомат, вычисляющий отношение, обратное $R(S)$.

c	a	t	s	#	
q_0	q_0	q_0	q_0	q_1	
c	a	t	$\hat{}$	s	$\#$

f	o	x	e		s	#	
q_0	q_0	q_0	q_1	q_2	q_3	q_4	q_0
f	o	x	$\hat{}$	ε	s	$\#$	

1.3 Алгоритм Портера

Note:
Алгоритм Портера в презентации Бориса Николаевича

Вопрос 4: Применить алгоритм

Слово: **compiling**

1. —
2. 2.а) compil
   2.б) —
3. —
4. —
5. —
6. —
7. —

Вопрос 5: Применить алгоритм

Слово: **comlipation**

1. —
2. —
3. —
4. compilate
5. —
6. compil
7. —

Note:

Д/з: прогнать алгоритм Портера для слов:
formalization, axiomatization, enumeration, fuzzieness, finger, singer