Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №1.**

**«Модели Машины Тьюринга и Алгоритмов Маркова»**

Выполнил студент гр. РИС-24-3б

Волков Денис Станиславович

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Машина Тьюринга**

1. Машина Тьюринга состоит из 3-х частей:
2. Устройство управления.
3. Голова Машины.
4. Исполняющая лента.
5. Элементарный шаг машины Тьюринга:
6. Голова считывает символ с ленты под ней.
7. Символ, который считан, и состояние головы обеспечивают переход машины Тьюринга в новое состояние  
   q1a1> q2 a2 d  
   Где:   
   a1 – символ, который считывается  
   q2 – новое состояние головы  
   a2 – новый записываемый символ  
   d - перемещение головы

Задача 1.

1. Дано число, состоящее из 0 и 1. Заменить все 0 на 1 и 1 на 0.
2. Голова находится на последней цифре числа. Будем заменять 0 на 1 и 0 на 1 передвигая влево.

Состояние Q1:

a) Заменяем 0 на 1 и передвигаем голову в лево, состояние не меняем

b) Заменяем 1 на 0 и передвигаем голову в лево, состояние не меняем

c) Если пустота, то завершаем алгоритм

3.

Расположение головы:

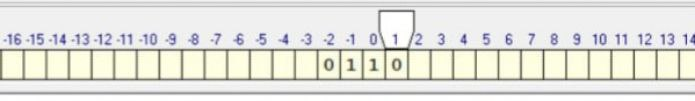
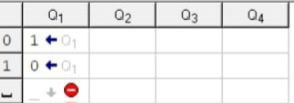
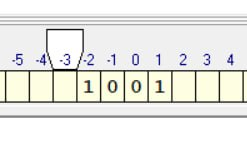


Таблица команд:



Результат:



Задача 2.

1. К случайному числу прибавить 4.
2. Голова находится на последней цифре числа. Прибавляем 4 к числам меньше 6 (от 0 до 5) и голову не двигаем. Если число больше 5 (от 6 до 9), то заменяем на разряд единиц полученный в сумме и переходим к следующему состоянию.

Состояние Q1:

1. Меняем 0 на 4 и останавливаем программу.
2. Меняем 1 на 5 и останавливаем программу.
3. Меняем 2 на 6 и останавливаем программу.
4. Меняем 3 на 7 и останавливаем программу.
5. Меняем 4 на 8 и останавливаем программу.
6. Меняем 5 на 9 и останавливаем программу.
7. Меняем 6 на 0 и переходим к следующему состоянию Q2.
8. Меняем 7 на 1 и переходим к следующему состоянию Q2.
9. Меняем 8 на 2 и переходим к следующему состоянию Q2.
10. Меняем 9 на 3 и переходим к следующему состоянию Q2.
11. Пробел не меняем.

Состояние Q2:

1. Меняем 0 на 1 и останавливаем программу.
2. Меняем 1 на 2 и останавливаем программу.
3. Меняем 2 на 3 и останавливаем программу.
4. Меняем 3 на 4 и останавливаем программу.
5. Меняем 4 на 5 и останавливаем программу.
6. Меняем 5 на 6 и останавливаем программу.
7. Меняем 6 на 7 и останавливаем программу.
8. Меняем 7 на 8 и останавливаем программу.
9. Меняем 8 на 9 и останавливаем программу.
10. Меняем 9 на 0 и двигаем голову влево не меняя состояния.
11. Меняем пробел на 1 и останавливаем программу.

3.

Расположение головы:

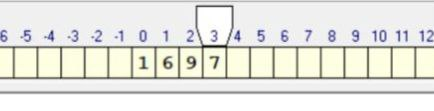
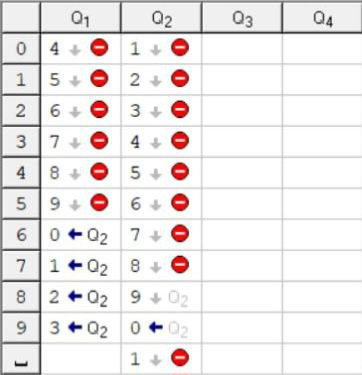
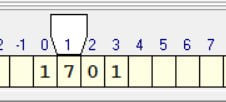


Таблица команд:



Результат:



Задача 3.

1. Дано десятичное число, если число четное, то меняем все цифры на 0, иначе на 1.
2. Голова находится на последней цифре числа. Меняем четные числа (0,2,4,6,8) на 0 и переходим к состоянию Q2.

Меняем нечетные числа (1,3,5,7,9) на 1 и переходим к состоянию Q3.

Состояние Q2:

Меняем все цифры на 0 и не меняем состояние. Если пустая ячейка, то завершаем программу.

Состояние Q3:

Меняем все цифры на 1 и не меняем состояние. Если пустая ячейка, то завершаем программу.

3.

Расположение головы:

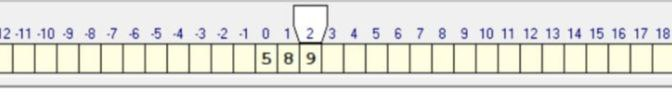
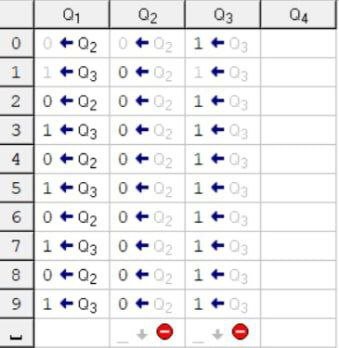
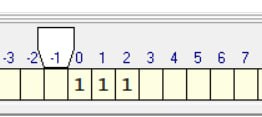


Таблица команд:



Результат:



**Нормальные алгоритмы Маркова**

1. Модель алгоритмов Маркова преобразует исходные слова с помощью заданных правил, которые могут заменять часть исходной строки.
2. Слова – это строки, состоящие из символов.

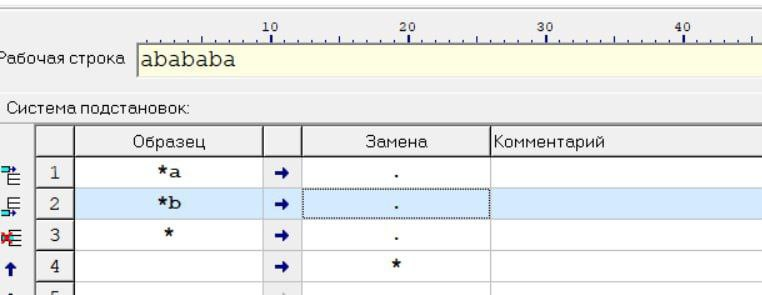
Также в алгоритмах Маркова используются приоритеты применения этих правил:  
1. Правило находит первое своё вхождение.  
2. Правила выполняются по порядку их номеров.  
3. Если правило не применимо к данной строке, то алгоритм переходит к выполнению следующего по порядку правила.  
4. Если правило выполнено, то алгоритм возвращается к проверке правил, снова начиная с начала списка.  
5. Алгоритм завершается, если не одно правило не применимо.

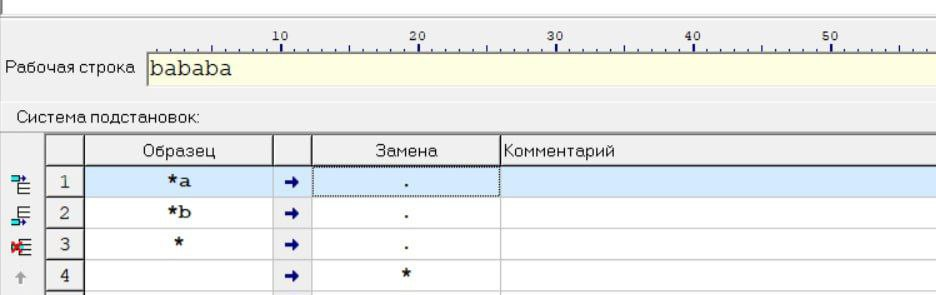
Виды подстановок:  
1. Терминальная – это подстановка, после которой выполнение алгоритма заканчивается.   
2. Нетерминальные – это подстановка, которая предполагает дальнейшее применение правил.

Задача 1.

1. Удалить из слова первую букву.
2. Алгоритм:
3. \*a меняем на точку и заканчиваем.
4. \*b меняем на точку и заканчиваем.
5. \* меняем на пустую строку и заканчиваем (этот шаг нужен чтобы в пустой строке при появлении \* эта звёздочка удалялась и алгоритм завершался).
6. Пустую строку меняем на \*.

Ход решения: сначала выполняется 4 шаг потом 1 или 2 (в зависимости от того какой первый символ строки).





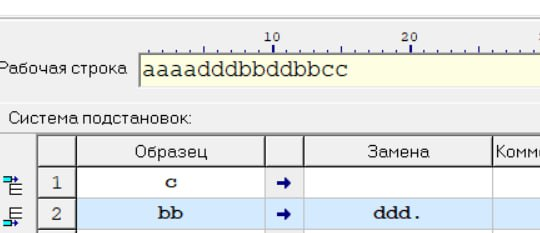
Задача 2.

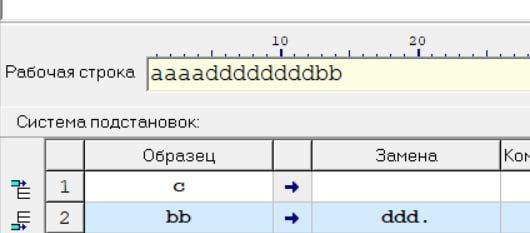
1. В слове удалить все вхождения c, а также заменить первое вхождение bb на ddd.
2. Алгоритм:
3. Меняем c на пустую строку.
4. Затем bb меняем на ddd и останавливаем строку.

Ход решения:

Сначала выполняется первый шаг алгоритма, как только все вхождения c удалены переходит на второй шаг алгоритма после первой замены bb на ddd алгоритм останавливается.







Задача 3.

1. Требуется приписать символ a к концу слова.
2. Алгоритм:
3. \*a заменяем на a\*.
4. \*b заменяем на b\*.
5. \* на a и заканчиваем.
6. Пустую строку заменяем на \*.

