УВАЖАЕМЫЕ СТУДЕНТЫ!

ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться сзаданием и выполнить задания ответить на вопросы.

2. Предоставит отчет конспекта лекции прислать в виде скриншото в течении трех дней.

3. Отправить преподавателю на почту v.vika2014@mail.ru и указать свою Ф.И.О, группу, и название дисциплины тел 072-17-44-9-22

Практическое занятие № 3

Тема: «Составление нормальных алгоритмов Маркова»

Цель: получить практические навыки в записи алгоритмов с использованием нормальных алгоритмов Маркова.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Фиксируем некоторый алфавит A. Пусть символы «стрелка» и «точка» не принадлежат алфавиту A: $\rightarrow \notin A$, $\cdot \notin A$. α, β, γ – некоторые, возможно пустые, слова в алфавите A.

Марковская подстановка (МП) – это операция над словами α, β, γ, заключающаяся в следующем:

1) в исходном слове γ ищется самое левое вхождение слова α, если оно существует, α заменяем на β в слове γ;

2) полученное слово γ' является результатом применения марковской подстановки, к слову, γ;

3) если слово α не входит в слово γ, то говорят, что данная Марковская подстановка неприменима к слову γ.

Обычно МП записывается как $\alpha \rightarrow \beta$.

$|\alpha| \neq |\beta|$: в общем случае длины слов α, β могут не совпадать.

$|\gamma| \neq |\gamma'|$: длины слов γ, γ' в общем случае также могут не совпадать.

Если $|\alpha| = 0$ или $|\beta| = 0$, то соответствующее слово является пустым. В МП пустое слово никак не обозначается и не занимает никакого места.

В любом слове α имеется несколько вхождений пустого слова: перед первой буквой; после последней буквы; между любой парой букв внутри слова.

Частные случаи МП: $\alpha \to \beta$

1. $\to \beta$, α – пусто: слово β приписывается перед γ : $\beta\gamma$.

2. $\alpha \to$, β – пусто: слово α исключается из γ.

Нормальным алгоритмом Маркова (НАМ) в алфавите A называется упорядоченная конечная последовательность (столбец) Марковских подстановок типа: $\alpha \to \beta$ или (и) $\alpha \to .\beta$, где α, β – слова в алфавите A, а символы $\to \notin A$ и $\to \cdot \notin A$.

$\alpha \to \cdot \beta$ – заключительная подстановка.

Запись НАМ – столбец подстановок вида $\alpha \to \beta; \alpha \to .\beta$.

Функционирование НАМ

1. $N := 1$, N – номер подстановки в схеме НАМ.

2. Выбирается N-тая МП, пусть она имеет вид $\alpha \to \beta$

3. Левая часть подстановки α ищется в преобразуемом слове γ.

4. Если α найдено, то переход к пункту 7, иначе к пункту 5.

5. $N := N + 1$

6. Если N не превышает общего числа подстановок, то переход к пункту 2, иначе – алгоритм заканчивает работу, останавливается.

7. Выполняется замена α на β в преобразуемом слове γ (крайнее левое вхождение α в γ).

8. Если эта подстановка является заключительной, т.е. имеет вид $\alpha \to .\beta$, алгоритм останавливается, иначе переход к пункту 1.

После применения подстановки осуществляется заново просмотр столбца подстановок, а не продолжается просмотр γ.

Процесс заканчивается, если:

– не найдена применяемая подстановка;

– выполнена заключительная подстановка.

Алфавит B называется **расширением** алфавита A, если $A \subseteq B$.

Нормальный алгоритм над алфавитом A – это нормальный алгоритм в алфавите B, который слова в A, если он к ним применим, перерабатывает в слова в A. Нормальный алгоритм <u>в</u> A может использовать только буквы алфавита A.

Нормальный алгоритм <u>над</u> A может использовать вспомогательные символы. НАМ над A более мощные, чем НАМ в A.

Одноместная частичная словарная функция $f(\alpha)$, заданная в алфавите A, называется **нормально вычислимой**, если существует НАМ над алфавитом A, перерабатывающий слово α в слово $f(\alpha)$.

Соответствие между нормальными алгоритмами и алгоритмами в интуитивном смысле выражает **принцип нормализации** – аналог тезисов Черча и Тьюринга: каков бы ни был алгоритм, для которого допустимыми исходными данными и результатом являются слова в некотором алфавите, существует эквивалентный ему НАМ в этом алфавите.

Пример 1. Нормальный алгоритм в алфавите $A = \{*, a, b\}$, перерабатывающий каждое слово вида $*\alpha$ в слово $*\alpha a$, где слово α – слово, состоящее из букв a, b.

$$
\begin{aligned}
&* a \to a * \\
&* b \to b * \\
&* \to a \\
&\to \cdot *
\end{aligned}
$$

Пусть $\alpha = abb$. Тогда, последовательность преобразований имеет вид:

$$\underline{*a}bb \Rightarrow a\underline{*b}b \Rightarrow ab\underline{*b} \Rightarrow abb\underline{*} \Rightarrow \underline{\ }abba \Rightarrow *abba.$$

Пример 2. Нормальный алгоритм над алфавитом $A = \{0,1,2\}$ стирающий все символы входного слова до первого символа 2 включительно.

$$
\begin{aligned}
&x0 \to 0x\\
&x1 \to 1x\\
&x2 \to y\\
&x \to y\\
&1y \to y\\
&0y \to y\\
&y \to .\\
&\to x
\end{aligned}
$$

Здесь вспомогательные символы x и y, таким образом, алфавит $B = \{0,1,2,x,y\}$. Буква x служит для того, чтобы найти букву 2 последовательным перебором слева направо. Буква y позволяет стереть буквы $0,1,2$ движением справа налево.

Пример функционирования НАМ по переработке слова 101201:

$$101201 \Rightarrow \underline{x1}01201 \Rightarrow 1\underline{x0}1201 \Rightarrow 10\underline{x1}201 \Rightarrow 101\underline{x2}01 \Rightarrow 101\underline{y}01 \Rightarrow 1\underline{0y}01 \Rightarrow$$

$$\Rightarrow \underline{1y}01 \Rightarrow \underline{y}01 \Rightarrow 01.$$

Пример 3. Нормальный алгоритм над алфавитом $A = \{0,1,è,ë\}$, который выдает “и”, если в исходном слове, состоящем из нулей и единиц, есть комбинация 1101, и “л”, в противном случае.

$$
\begin{aligned}
&1x \to x\\
&x1 \to x\\
&0x \to x\\
&x0 \to x\\
&x \to .è\\
&1101 \to x\\
&1 \to\\
&0 \to\\
&\to .ë
\end{aligned}
$$

Приведем два примера функционирования НАМ:

1) $010\underline{110}10 \Rightarrow 010\underline{x1}0 \Rightarrow 01\underline{0x}0 \Rightarrow 0\underline{1x}0 \Rightarrow \underline{0x}0 \Rightarrow \underline{x0} \Rightarrow x \Rightarrow è.$

2) $\underline{1}010111 \Rightarrow 0\underline{1}0111 \Rightarrow 00\underline{11}1 \Rightarrow 00\underline{1}1 \Rightarrow 00\underline{1} \Rightarrow \underline{0}0 \Rightarrow \underline{0} \Rightarrow ë.$

Пример 4. Пример НАМ, который работает бесконечно.

$$
\begin{aligned}
&a \to b\\
&b \to a
\end{aligned}
$$

Задание к практической работе:

1. Изучите теоретические сведения к практической работе.

2. С помощью эмулятора машины Поста, решите задачи представленные ниже.

3. Оформите отчет, включив в него: код программы, блок схему алгоритма, выводы, ответы на контрольные вопросы.

Задания к выполнению:

1. Реализовать алгоритм, выполняющий замену в слове α в алфавите $A=\{a,d,c\}$ каждого символа a на символ c.

2. Реализовать алгоритм, выполняющий перестановку в слове α в алфавите $A=\{0,1\}$ букв таким образом, чтобы сначала стояли все нули, а затем все единицы.

3. Реализовать алгоритм в алфавите $A=\{0,1\}$, меняющий местами первую и последнюю буквы слова.

4. Реализовать алгоритм над алфавитом $A=\{0,1\}$, меняющий местами первый ноль и последнюю единицу.

5. Реализовать алгоритм над алфавитом $\{a,b\}$, переставляющий буквы в обратном порядке.

Контрольные вопросы:

1. Что такое Марковская подстановка?

2. Что такое заключительная Марковская подстановка, как она обозначается?

3. В каком случае Марковская подстановка считается неприменимой к некоторому слову?

4. Как функционирует нормальный алгоритм Маркова?

5. В каких случаях НАМ заканчивает работу и останавливается?

6. Чем отличается НАМ в алфавите *A* от алгоритма над алфавитом *A*?

7. Привести пример бесконечно работающего нормального алгоритма Маркова.

8. Привести определение нормально вычислимой словарной функции.
9. Сформулировать принцип нормализации.