



Нормальный алгоритм Маркова

Один из способов формального
определения понятия «алгоритм»

Андре́й Андре́евич Ма́рков (1903 - 1979)

- советский математик
- сын известного русского математика Андрея Андреевича Маркова (1856-1922)
- основоположник советской школы конструктивной математики
- автор понятия нормального алгоритма



Нормальный алгоритм (алгоритм)

- задает метод преобразования строк с помощью упорядоченной системы подстановок. Каждая подстановка состоит из слова-образца (\mathbf{A}_i) и слова-замены (\mathbf{B}_i), соединенных между собой \rightarrow или $\rightarrow \cdot$.

$\mathbf{A}_i \rightarrow \mathbf{B}_i$ – формула подстановки

$\mathbf{A}_i \rightarrow \cdot \mathbf{B}_i$ – завершающая формула подстановки

Работа нормального алгоритма


На каждом шаге работы алгоритма:

- все подстановки просматриваются по порядку сверху вниз, ищется первая из них, которая является применимой, т. е. в обрабатываемом слове содержится слово-образец (A_i) из этой подстановки
- применяется найденная подстановка $A_i \rightarrow B_i$ (A_i заменяется на B_i)


Работа нормального алгоритма

заканчивается в двух случаях:

- все подстановки оказались не применимыми
- была применена завершающая подстановка $A_i \rightarrow \cdot B_i$



Нормальный алгоритм является **неприменимым** к данному входному слову, если в процессе выполнения алгоритма бесконечное число раз выполняются не завершающие $(\mathbf{A}_i \rightarrow \mathbf{B}_i)$ подстановки



В левых и правых частях формул подстановок могут содержаться пустые слова.

Пустое слово содержится слева и справа от каждой буквы в преобразуемом слове.

Первое пустое слово расположено перед первой буквой слова

Примеры нормальных алгоритмов

1. $\rightarrow \mathbf{a}$

Бесконечно приписывает слева к входному слову букву **a**.

Этот алгоритм неприменимый ни к одному слову.

3. $\rightarrow \cdot \mathbf{a}$

Приписывает слева к входному слову букву **a**.

Этот алгоритм применимый к любому слову.

3. $\mathbf{a} \rightarrow$

$\mathbf{b} \rightarrow$

$\mathbf{c} \rightarrow$

Стирает во входном слове буквы **a, b, c**.

Пример 1

Закодировать слово из алфавита {**a**, **b**, **c**}
с помощью **0** и **1**

Нормальный алгоритм:

a → **00**

b → **01**

c → **10**

Входное слово: **caab**

Пример 2

Приписать к слову из алфавита $\{a, b, c\}$ справа букву **a**

Идея алгоритма:

Чтобы найти конец слова, припишем в начало слова какой-нибудь специальный символ, например, *, а затем переместим его вдоль всего слова, до конца, и заменим * на букву a.

Пример 2

Нормальный алгоритм:

приписывание * в начало

Пример 2

Нормальный алгоритм:

→ * приписывание * в начало

Пример 2

Нормальный алгоритм:

→ * приписывание * в начало

перемещение * вдоль слова

Пример 2

Нормальный алгоритм:

$\rightarrow *$ приписывание $*$ в начало

$*a \rightarrow a*$

$*b \rightarrow b*$

$*c \rightarrow c*$

перемещение $*$ вдоль слова

Пример 2

Нормальный алгоритм:

$\rightarrow *$ приписывание $*$ в начало

$*a \rightarrow a*$

$*b \rightarrow b*$ перемещение $*$ вдоль слова

$*c \rightarrow c*$

замена $*$ в конце слова на a

Пример 2

Нормальный алгоритм:

$\rightarrow *$ приписывание $*$ в начало

$*a \rightarrow a*$

$*b \rightarrow b*$ перемещение $*$ вдоль слова

$*c \rightarrow c*$

$* \rightarrow a$ замена $*$ в конце слова на a

В чем ошибка?

Пример 2

Нормальный алгоритм:

a** → **a

b** → **b

c** → **c

перемещение * вдоль слова

***** → **a**

замена * в конце слова на a

→ *****

приписывание * в начало

В чем ошибка?

Пример 2

Нормальный алгоритм:

$*a \rightarrow a*$

$*b \rightarrow b*$

$*c \rightarrow c*$

перемещение $*$ вдоль слова

$* \rightarrow \cdot a$

замена $*$ в конце слова на a

$\rightarrow *$

приписывание $*$ в начало

Пример 3

В слове из алфавита $\{a, b\}$ последнее вхождение символа a заменить на aa .

Идея алгоритма:

С помощью спецзнака $*$ найти конец слова. Заменить $*$ на спецзнак $\#$. С помощью спецзнака $\#$ пройти от конца слова до первого с конца (последнего вхождения в слово) символа a (или начала слова, если a в слове нет). Если a есть, заменить на aa . Остановиться.

Пример 3

Нормальный алгоритм:

a** → **a

b** → **b

***** → **#**

b# → **#b**

a# → **· aa**

→ **·**

→ *****

Пример 4

В слове из алфавита $\{a, b\}$ его первый символ перенести в конец слова.

Идея алгоритма:

1. Помечаем первый символ слова спецзнаком *
2. Заменяем * и этот первый символ на новый: a на A, b на B. (A и B – спецзнаки, помогающие отличить первый знак от остальных таких же знаков в слове)

Пример 4

Идея алгоритма:

3. Перегоняем новый символ A или B в конец слова.
4. Заменяем A или B в конце слова на прежний символ (A на a , B на b) и останавливаем алгоритм

Пример 4

Нормальный алгоритм:

***a** → **A**

***b** → **B**

Aa → **aA**

Bb → **bB**

Ab → **bA**

Ba → **aB**

A → · **a**

B → · **b**


***** → ·

→ *****

Пример 1

$A = \{0, 1, 2, 3\}$. Пусть P – непустое слово. Трактую его как запись неотрицательного целого числа в четверичной системе счисления, требуется получить запись этого же числа, но в двоичной системе.

Например: $0123 \rightarrow 00011011$



Для перевода числа из четверичной системы в двоичную надо следует каждую четверичную цифру заменить на пару соответствующих ей двоичных цифр: $0 \rightarrow 00$, $1 \rightarrow 01$, $2 \rightarrow 10$, $3 \rightarrow 11$.


Такая замена при первом рассмотрении реализуется следующим НАМ:

1. $0 \rightarrow 00$

2. $1 \rightarrow 01$

3. $2 \rightarrow 10$

4. $3 \rightarrow 11$



**Но этот алгоритм неправильный, в чём можно
убедиться на входном слове *0123*:**

0123 → 00123 → 000123 → ...

Ошибка:

после замены четверичной цифры на пару двоичных цифр уже никак нельзя отличить двоичные цифры от четверичных, поэтому НАМ начинает заменять и двоичные цифры.

Проблема: отделить ту часть числа, в которой уже была произведена замена, от той части, где замены ещё не было.

Идея: пометить слева спецзнаком * ту четверичную цифру, которая сейчас должна быть заменена на пару соответствующих двоичных цифр, а после того как такая замена будет выполнена, спецзнак нужно поместить перед следующей четверичной цифрой:

$0123 \rightarrow *0123 \rightarrow 00*123 \rightarrow 0001*23 \rightarrow 000110*3 \rightarrow 00011011*$

Алгоритм:

1. $*0 \rightarrow 00*$

2. $*1 \rightarrow 01*$

3. $*2 \rightarrow 10*$

4. $*3 \rightarrow 11*$

5. $* \mid \rightarrow$

6. $\rightarrow *$

Пример 2

$A = \{a, b\}$. Удвоить слово P , т.е. приписать к P (слева или справа) его копию.

Например: $abb \rightarrow abbabb$

Идея

1. Приписываем к концу слова P символ $=$, справа от которого будем строить копию P .
2. Просматриваем по очереди все символы слова P и, не уничтожая их, переносим копию каждого символа в конец.
3. Удаляем символ $=$, который отделял слово P от его копии, и останавливаем алгоритм.

Идея

Приписать некоторый символ к концу слова: надо сначала приписать слева к слову какой-то спецзнак (*), затем перегнать его направо через все символы слова и в конце, когда за * не окажется никакого символа, заменить на символ =

$$abbb \rightarrow *abbb \rightarrow a*bbb \rightarrow ab*b \rightarrow abbb* \rightarrow abbb=$$

Идея

если надо скопировать символ a , то порождаем за ним новый символ A (заменяем a на aA), после чего этот символ A переставляем с каждым последующим символом (в том числе и с символом $=$), перенося тем самым A в конец слова, где и заменяем на a :

$$abb= \rightarrow aAbb= \rightarrow abAb= \rightarrow abbA= \rightarrow abb=A \rightarrow$$
$$abb=a$$

Как узнать, какой именно символ исходного слова мы только что скопировали и какой символ надо копировать следующим?

Идея: будем помечать новым спецзнаком # тот символ, который должен копироваться следующим (вначале это первый символ входного слова):

$\#abb= \rightarrow a\#Abb= \rightarrow a\#bAb= \rightarrow a\#bbA= \rightarrow$
 $a\#bb=A \rightarrow a\#bb=a$


Как только копия очередного символа окажется в конце, спецзнак # должен «запустить» процесс копирования следующего символа:

$a\#bb=a \rightarrow ab\#Bb=a \rightarrow ab\#bB=a \rightarrow ab\#b=Ba$
 $\rightarrow ab\#b=aB \rightarrow ab\#b=ab \rightarrow$
 $\rightarrow abb\#B=ab \rightarrow abb\#=Bab \rightarrow abb\#=aBb \rightarrow$
 $abb\#=abB \rightarrow abb\#=abb$

Проблема: два спецзнака * и #, первый из которых нужен для приписывания символа = справа к входному слову, а второй – для указания, какой символ слова должен копироваться следующим.

Проблема: использовать для этого две формулы \rightarrow^* и $\rightarrow\#$ нельзя, т.к. первая из них будет блокировать доступ ко второй. Оба этих спецзнака надо вводить сразу одной формулой $\rightarrow\#^*$. При этом надо учитывать, что формулы с * должны применяться самыми первыми, поэтому они должны располагаться в начале НАМ.

Формулы же с #, A и B должны располагаться ниже, чтобы они работали только после того, как исчезнет * и появится символ =.


$$* a \rightarrow a *$$

$$* b \rightarrow b *$$

$$* \rightarrow =$$

$$Aa \rightarrow aA$$

$$Ab \rightarrow bA$$

$$A = \rightarrow =A$$

$$A \rightarrow a$$


$$Ba \rightarrow aB$$

$$Bb \rightarrow bB$$

$$B = \rightarrow =B$$

$$B \rightarrow b$$

$$\# a \rightarrow a \# \quad \# A$$



$*a \rightarrow aA*$

$*b \rightarrow bB*$

$* \rightarrow \#$

$Aa \rightarrow aA$

$Ab \rightarrow bA$

$Ba \rightarrow aB$

$Bb \rightarrow bB$

$A\# \rightarrow \#a$

$B\# \rightarrow \#b$

$\# \mid \rightarrow$

$\rightarrow *$


Пример 3

Составить НАМ вычисления функции $f(x)=x + 1$.

Для примера рассмотрим на троичной системе счисления, т. е. в алфавите $\{0, 1, 2\}$.

1. Входное слово – число 1023

2. Входное слово – число 223



1) $*0 \rightarrow 0*$

2) $*1 \rightarrow 1*$

3) $*2 \rightarrow 2*$

4) $* \rightarrow \#$

5) $0\# 1$

6) $1\# 2$

7) $2\# \rightarrow \#0$

8) $\# 1$

9) $\rightarrow *$

Задание

1. Действительное число из 8-ричной системы счисления перевести в двоичную систему счисления.
Например, из числа **73,1** получить число **111011,001**
2. В слове из алфавита {**a, b, c**} удалить его первый символ