УДК 004.912

*А.В. Пруцков*

# ОБРАБОТКА ЧИСЛИТЕЛЬНЫХ ЕСТЕСТВЕННЫХ ЯЗЫКОВ С ПОМОЩЬЮ ФОРМАЛЬНЫХ ГРАММАТИК И НОРМАЛЬНЫХ АЛГОРИТМОВ МАРКОВА

*Разработан метод генерации и определения числительных, включающий трехуровневую обобщенную модель числительного и алгоритмы взаимодействия между моделью и числительными, позволяющие переводить числительные в числа и обратно, с одного естественного языка на другой. При разработке метода использовался математический аппарат формальных грамматик и нормальных алгоритмов Маркова. Предложено обобщение нормальных алгоритмов Маркова, позволяющее разрабатывать более эффективные алгоритмы.*

***Ключевые слова**: автоматическая обработка числительных, нормальные алгоритмы Маркова.*

**Определение задач генерации и определения числительных.** При преобразованиях «речь-текст» необходимо преобразовывать числа в числительные и обратно. Такие преобразования используются в следующих информационных системах:

– диалоговые системы «машина-человек»;
– системы синтеза и анализа речи;
– автоматизированные обучающие системы;
– системы машинного перевода.

Числительное – это часть речи, объединяющая группу слов со значением количества или слова со значением порядка предметов при счете. В данной работе рассматриваются только количественные числительные – части речи, объединяющие слова, служащие для наименования чисел или количества.

Определим задачи генерации и определения числительных. Генерация числительных заключается в переводе цифровой записи числительного в символьную запись (рисунок 1).

ЧИСЛО $\xrightarrow{\text{ГЕНЕРАЦИЯ}}$ ЧИСЛИТЕЛЬНОЕ

**Рисунок 1 – Задача генерации числительных**

Задача определения числительных состоит в преобразовании словесной формы записи числительных в цифровую форму записи (рисунок 2).

ЧИСЛИТЕЛЬНОЕ $\xrightarrow{\text{ОПРЕДЕЛЕНИЕ}}$ ЧИСЛО

**Рисунок 2 – Задача определения числительных**

**Грамматика Г. Хардегри и ее недостатки.** Для описания порядка образования числительных используется аппарат теории формальных грамматик.

Гари Хардегри предложил грамматику NG3 [1], позволяющую описывать порядок числительных и преобразование чисел в числительные английского языка. Однако данная грамматика обладает следующими недостатками: предназначена только для генерации числительных; предназначена только для английского языка; описывает правила образования только целой части числительных; не учитывается склонение числительных по падежам, так как в английском языке не существует грамматической категории падежа.

**Трехуровневая обобщенная модель числительного.** Устраним перечисленные недостатки с помощью предлагаемой трехуровневой обобщенной модели числительного.

Введем следующие обозначения. Обозначения в числах:

– цифры: $Z = \{Z_0|0, Z_1|1, Z_2|2, \ldots, Z_9|9\}$;
– знак минуса: $S_-$;
– разделитель целой и дробной частей: J.

Обозначения в числительных:

– названия цифр (простые непроизводные числительные): $C = \{C_0, C_1, C_2, \ldots, C_9\}$;
– названия десятков и сотен: $D = \{D_1, D_2\}$, где $D_1$ – обозначение десятков; $D_2$ – обозначение сотен;
– названия порядков триад: тысячи, миллионы, миллиарды и т. д.: $M = \{M_0, M_1, M_2, M_3, \ldots\}$, где $M_i$ – название i + 1-го порядка триады числительного;
– названия знаков числа: $P = \{P_-, P_+, P_0\}$, где

$P_-$ – название знака отрицательных чисел; $P_+$ – название знака положительных чисел, $P_0$ – название знака нуля;

– название разделителя целой и дробной частей: E;

– название окончания дробной части: B.

Пример 1. В предложенных обозначениях число

$1\,000\,400\,973 = Z_1\ Z_0Z_0Z_0\ Z_4Z_0Z_0\ Z_9Z_7Z_3$

будет иметь следующий вид:

$P_+C_1M_3C_4D_2M_1C_9D_2C_7D_1C_3M_0EB$. □

Введенные обозначения необходимы для обобщения модели и ее независимости от языка.

На основе проведенного анализа правил построения числительных естественных языков предлагается трехуровневая обобщенная модель числительного (далее просто модель числительного), состоящая из следующих уровней.

Уровень 1. Знак числа, целая и дробная части. Разделителями частей являются слова-связки «*целых*», «*запятая*».

Уровень 2. Трехразрядные составляющие (триады). Каждая часть разделяется на трехразрядные составляющие, начиная от разделителя целой и дробной частей. Разделителями трехразрядных составляющих являются слова-связки «*тысяч*», «*миллионов*» и т. д.

Уровень 3. Элементы трехразрядных составляющих. Разделителями являются слова-связки «*десятки*», «*сотни*».

Пример 2. Структура предлагаемой трехуровневой обобщенной модели числительных на примере числа 34 567,89 представлена на рисунке 3. □

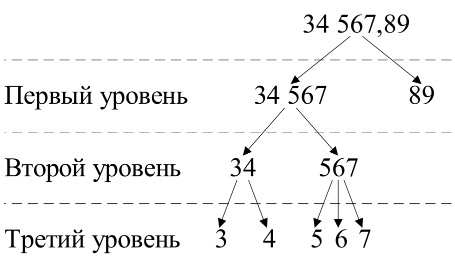


**Рисунок 3 – Пример разбиения числа 34 567,89 на три уровня**

Порядок образования числительного можно описать с помощью аппарата формальных грамматик.

Первый уровень модели

$K = P + N_1 + E + \{\varnothing|N_2\} + B$

$P = P_-|P_+|P_0$

Второй уровень модели

$N_1 = C_0|N_{10}|N_{11}|N_{12}|...|N_{1i}$

$N_2 = (\{T_1|C_0\} + N_2)|T_1|C_0$

$N_{10} = T + M_0$

$N_{11} = T + M_1 + \{\varnothing|N_{10}\}$

$N_{12} = T + M_2 + \{\varnothing|N_{10}|N_{11}\}$

...

$N_{1i} = T + M_i + \{\varnothing|N_{11}|N_{12}|...|N_{1(i-1)}\}$

Третий уровень модели

$T = T_1|T_2|T_3$

$T_1 = C_1|C_2|C_3|C_4|C_5|C_6|C_7|C_8|C_9$

$T_2 = T_1 + D_1 + \{\varnothing|T_1\}$

$T_3 = T_1 + D_2 + \{\varnothing|T_1|T_2\}$

Предложенная модель числительного – это обобщенная запись числительного. Модель лежит в основе метода генерации и определения числительных. Помимо модели, метод также включает алгоритмы взаимодействия между числами, числительными и моделью.

**Алгоритмы взаимодействия между числами, числительными и моделью.** Для записи алгоритмов генерации и определения числительных и чисел воспользуемся математическим аппаратом нормальных алгоритмов Маркова [2, 3].

Введем дополнительные обозначения в подстановках, уменьшающие количество подстановок нормальных схем и повышающие их наглядность:

$\to +\gamma$ – установка символа $\gamma$ в конец строки;

$\to \gamma+$ – установка символа $\gamma$ в начало строки;

$[Z_0]_k$ – последовательность k нулей;

_ (подчеркивание) – символ пробела.

Приведем основные алгоритмы взаимодействия.

Алгоритм генерации целой части числительных преобразует целую часть числа в обозначения модели.

1) $Z_k\gamma_1^j \to \gamma_2^j C_k M_j$ ;j = 0, 1, 2, …; k = 1, 2, …, 9

2) $Z_k\gamma_2^j \to \gamma_3^j C_k D_1$ ;j = 0, 1, 2, …; k = 1, 2, …, 9

3) $Z_k\gamma_3^j \to \gamma_1^{j+1} C_k D_2$ ;j = 0, 1, 2, …; k = 1, 2, …, 9

4) $Z_0Z_0Z_0\gamma_1^j \to \gamma_1^{j+1}$ ;j = 0, 1, 2, …

5) $Z_0\gamma_1^j \to \gamma_2^j M_j$ ;j = 0, 1, 2, …

6) $Z_0\gamma_2^j \to \gamma_3^j$ ;j = 0, 1, 2, …

7) $Z_0\gamma_3^j \to \gamma_1^{j+1}$ ;j = 0, 1, 2, …

8) $\gamma_2^0 M_0 E \to P_0 C_0 EB \bullet$

9) $S_-\gamma_i^j C_k \to P_- C_k \bullet$ ;i = 1, 2, 3; j = 0, 1, 2, …; k = 0, 2, …, 9

10) $\gamma_i^j C_k \to P_+ C_k \bullet$ ;i = 1, 2, 3; j = 0, 1, 2, …; k = 0, 2, …, 9

11) $J \to \gamma_1^0 E$

12) $\to +\gamma_1^0 EB$

Алгоритм определения целой части числительных предназначен для преобразования записи целой части числительного в число.

1) $M_j\gamma_i^L \to \gamma_1^j [Z_0]_{(4-i)+3\times(j-L-1)}$ ;i = 1, 2, 3; L = 0, 1, 2, …; j = L+1, L+2, …

2) $M_j \gamma_1^j \to \gamma_1^j$ ;j = 0, 1, 2, …

3) $C_k \gamma_1^j \to \gamma_2^j Z_k$ ;j = 0, 1, 2, …; k = 1, 2, …, 9

4) $C_k D_1 \gamma_i^j \to \gamma_3^j Z_k [Z_0]_{2-i}$ ;i = 1, 2; j = 0, 1, 2, …; k = 1, 2, …, 9

5) $C_k D_2 \gamma_i^j \to \gamma_1^{j+1} Z_k [Z_0]_{3-i}$ ;i = 1, 2, 3; j = 0, 1, 2, …; k = 1, 2, …, 9

6) $P_+ \gamma_i^j Z_k \to Z_k \bullet$ ;i = 1, 2, 3; j = 0, 1, 2, …; k = 1, 2, …, 9

7) $P_- \gamma_i^j Z_k \to S_- Z_k \bullet$ ;i = 1, 2, 3; j = 0, 1, 2, …; k = 1, 2, …, 9

8) $P_+ \gamma_i^j Z_0 \to Z_1 Z_0 \bullet$ ;i = 2, 3; j = 0, 1, 2, …

9) $P_- \gamma_i^j Z_0 \to S_- Z_1 Z_0 \bullet$ ;i = 2, 3; j = 0, 1, 2, …

10) $P_0 C_0 EB \to Z_0 \bullet$

11) $EB \to \gamma_1^0$

12) $E \to \gamma_1^0 J$

В основе алгоритма определения числительных лежит предлагаемая модель числительного. Это дает возможность алгоритму не только распознавать числительные, но и проверять их запись на наличие ошибок.

Выполнение подстановок 6-9 является правильным завершением алгоритма. Невыполнение ни одной из подстановок означает, что числительное содержит ошибку.

Алгоритм генерации дробной части числительных предназначен для преобразования дробной части числа в обозначения модели числительных.

1) $\lambda Z_k \to C_k \lambda$ ;k = 0, 1, …, 9

2) $\lambda \to B \bullet$

3) $E \to E\lambda$

Алгоритм определения дробной части числительных предназначен для преобразования дробной части в обозначениях модели числительных в обозначения числа.

1) $\lambda C_k \to Z_k \lambda$ ;k = 0, 1, …, 9

2) $\lambda C_k B \to Z_k \bullet$ ;k = 0, 1, …, 9

3) $EB \to EB \bullet$

4) $E \to E\lambda$

**Характеристики образования числительных естественных языков.** Числительные в разных языках образуются по разным правилам. В таблице 1 представлено сравнение числительных пяти естественных языков различных семейств и групп по разным критериям.

**Таблица 1**

| Критерии | Русский | Английский | Немецкий | Испанский | Финский |
|---|---|---|---|---|---|
| Последовательность в триаде «сотни-десятки-единицы» | + | + | – | + | + |
| Вспомогательные слова-связки между десятками и единицами | – | – | + | + | – |
| Числительное без пробелов | – | – | + | – | + |
| Перечисление названий цифр дробной части числа | – | + | + | + | + |
| Склонение числительных по падежам | + | – | + | – | + |

Пример 3. Приведем запись числа 34 567,89 на перечисленных в таблице 1 языках.

Русский: «тридцать четыре тысячи пятьсот шестьдесят семь целых восемьдесят девять сотых».

Английский: «thirty four thousand five hundred sixty seven point eight nine».

Немецкий: «vierunddreißigtausendfünfhundertsiebenundsechzig punkt acht neun».

Испанский: «treinta y cuatro mil quinientos sesenta y siete punto ocho nueve».

Финский: «kolmekymmentäneljätuhattaviisisataakuusikymmentäseitsemän pistu kahdeksan yhdeksän». □

Для каждого языка описаны обозначения языка и слова-исключения в основном падеже языка (именительном падеже) и разработаны алгоритмы, связывающие числительные языка и модель числительного, если они необходимы.

Из-за большого объема эти данные в статье не приводятся.

**Порядок взаимодействия алгоритмов, модели и числительных.** Обозначения модели носят общий характер. В каждом языке обозначения модели имеют свой уникальный вид. Обозначения модели в данном языке назовем обозначениями языка. Например, в русском языке $C_0$ = «*ноль*», $C_1$ = «*один*» и т. д.

В некоторых языках числительные образуются по правилам, отличающимся от правил, заложенных в модель числительного. В этом случае потребуется корректировка числительного с помощью алгоритмов. Необходимо наличие двух алгоритмов:

1) алгоритма корректировки числительного

модели, преобразующего числительное в правилах модели в числительное в правилах языка;

2) алгоритма корректировки числительного языка, преобразующего числительное в правилах языка в числительное в правилах модели.

Данные алгоритмы совершают обратные друг другу действия.

Чтобы перевести число в числительное, необходимо выполнить следующие действия:

1) вызвать алгоритм генерации целой части;

2) вызвать алгоритм генерации дробной части;

3) вызвать алгоритм корректировки числительного модели, если он существует;

4) заменить обозначения модели обозначениями языка в нужном падеже.

Чтобы перевести числительное в число, необходимо выполнить следующие действия:

1) заменить обозначения языка обозначениями модели;

2) вызвать алгоритм корректировки числительного языка, если он существует;

3) вызвать алгоритм определения дробной части;

4) вызвать алгоритм определения целой части.

С помощью предлагаемой модели числительных можно переводить числительные с одного языка на другой. Чтобы перевести числительное с языка L1 на язык L2, необходимо выполнить следующие действия:

1) заменить обозначения языка L1 обозначениями модели;

2) вызвать алгоритм корректировки числительного языка L1, если он существует;

3) вызвать алгоритм корректировки числительного модели к языку L2, если он существует;

4) заменить обозначения модели обозначениями языка L2 в нужном падеже.

Помимо перевода с помощью модели числительных можно склонять числительные по падежам. Чтобы изменить падеж числительного с падежа P1 на падеж P2, необходимо выполнить следующие действия:

1) заменить обозначения языка L1 в падеже P1 обозначениями модели;

2) заменить обозначения модели обозначениями языка в падеже P2.

Порядок замены обозначений модели обозначениями языка и обратно следующий: обозначения меняются, начиная с самых длинных обозначений модели.

Кроме склонения числительных, можно определить падеж числительного. Для этого необходимо сравнить слова в числительном с обозначениями языка во всех падежах. Числительное имеет падеж P, если все слова в числительном записаны в падеже P. Если не все слова в числительном записаны в одном падеже, значит, числительное записано с ошибками.

**Перевод числительных с помощью трехуровневой обобщенной модели числительных.** При машинном переводе возникает необходимость в переводе числительных с одного языка на другой. Пословный перевод неприменим, так как правила образования числительных в разных языках различны (таблица 1).

Рассмотрим два подхода к переводу числительных:

1) через число – преобразование числительного с одного языка к числу, а затем обратное преобразование числа к числительному на другом языке (рисунок 4, а);

2) через модель – перевод числительных с использованием модели числительных в качестве промежуточного этапа (рисунок 4, б).

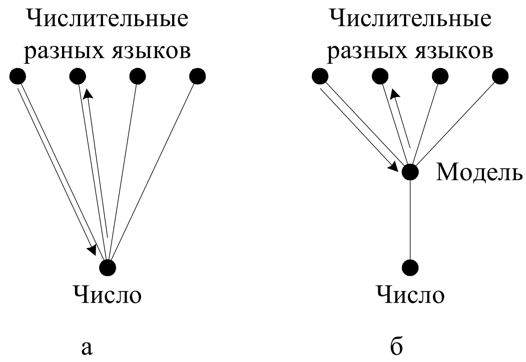


**Рисунок 4 – Схема преобразования числительных и чисел в существующих и предлагаемом методах**

Данные подходы отличаются друг от друга трансфером – промежуточным этапом, к которому сводятся числительные. В первом случае – это число, а во втором – модель числительного.

Сравним трудоемкость двух этих процессов.

Пример 4. Дано число 34 567,89 или числительное в обозначениях модели $P_+C_3D_1C_4M_1C_5D_2C_6D_1C_7M_0EC_8C_9B$. Определим характеристики данного числа, необходимые для вычисления трудоемкости:

– количество цифр в целой части числа $dc = 5$;

– количество нулевых триад в целой части числа $mc = 0$;

– количество названий цифр в целой части числительного $wc = 5$;

– количество триад в целой части числительного $mw = 2$;

– количество десятков и единиц в целой части числительного $wde = 2$;

– количество названий цифр в дробной части числительного $wd = 2$;

– количество триад в дробной части числительного md = 1;

– количество цифр в дробной части числа dd = 2. □

Пример 5. Переведем числительное, соответствующее числу 34 567,89, с русского языка на немецкий язык двумя способами: через число и через модель.

При переводе через число необходимо выполнить следующие этапы.

1. Заменить слова русского языка цифрами. Целая и дробная части числительного русского языка образуются по одинаковому принципу, например «*триста пятьдесят две целых триста пятьдесят две тысячных*». Этапу соответствует вызов алгоритма определения целой части числительного. Для дробной части необходимо использовать алгоритм определения дробной части числительного русского языка, состоящий из следующих подстановок.

1) $M_j \gamma_i^L \rightarrow \gamma_1^j [Z_0]_{(4-i)+3\times(j-L+1)}$ ;i = 1, 2, 3; L = 0, 1, 2, …; j = L+1, L+2, …

2) $M_j \gamma_1^j \rightarrow \gamma_1^j$ ;j = 0, 1, 2, …

3) $C_k \gamma_1^j \rightarrow \gamma_2^j Z_k$ ;j = 0, 1, 2, …; k = 1, 2, …, 9

4) $C_k D_1 \gamma_i^j \rightarrow \gamma_3^j Z_k [Z_0]_{2-i}$ ;i = 1, 2; j = 0, 1, 2, …; k = 1, 2, …, 9

5) $C_k D_2 \gamma_i^j \rightarrow \gamma_1^{j+1} Z_k [Z_0]_{3-i}$ ;i = 1, 2, 3; j = 0, 1, 2, … k = 1, 2, …, 9

6) $E \gamma_i^j \rightarrow E [Z_0]_{t-((i-1)+3\times j)} \bullet$ ;i = 1, 2, 3; j = 0, 1, 2, …; t = 1, 2, …

7) $M_0 B_t \rightarrow \gamma_1^0 B$ ;t = 1, 2, …

8) $EB \rightarrow EB \bullet$

2. Заменить цифры словами немецкого языка. Этапу соответствует вызов алгоритмов генерации целой и дробной частей числительного.

3. Скорректировать числительное на немецком языке. Этапу соответствует вызов алгоритма корректировки числительного немецкого языка.

Трудоемкость данных этапов приведена в таблице 2.

Трудоемкость этапов перевода через модель числительного приведена в таблице 3. Замена обозначений модели обозначениями языка не рассматриваются, так как встречаются при любом из рассматриваемых подходов.

**Таблица 2**

| Название этапа | Трудоемкость |
|---|---|
| 1. Замена слов русского языка цифрами:<br>– алгоритм определения целой части числительного;<br>– алгоритм определения дробной части числительного русского языка; | $T_{ОЦ}$ = wc + mw + 2 = 5 + 2 + 2 = 9<br>$T_{ОДРУ}$ = wd + (md – 1) + 2 =<br>= 2 + (1 – 1) + 2 = 4 |
| 2. Замена цифр словами немецкого языка:<br>– алгоритм генерации целой части числительного немецкого языка;<br>– алгоритм генерации дробной части числительного немецкого языка | $T_{ГЦ}$ = dc – 2·mc + 2 = 5 – 2·0 + 2 = 7<br>$T_{ГД}$ = dd + 2 = 2 + 2 = 4 |
| 3. Корректировка числительного немецкого языка | $T_{КМНЕ}$ = wde + wd + 2 = 2 + 2 + 2 = 6 |
| Всего | 30 |

**Таблица 3**

| Название этапа | Трудоемкость |
|---|---|
| 1. Алгоритм корректировки числительного русского языка | $T_{КЯРУ}$ = 2·wd – 2·mdo + 3 =<br>= 2·2 – 2·0 + 3 = 7 |
| 2. Алгоритм корректировки числительного модели для немецкого языка | $T_{КМНЕ}$ = wde + wd + 2 =<br>= 2 + 2 + 2 = 6 |
| Всего | 13 |

Приведенные расчеты трудоемкости показывают, что перевод числительных с использованием в качестве трансфера предложенной модели более эффективен, чем перевод через число. Объяснение этому очевидно. При переводе с помощью модели не выполняются такие действия, как вставка и удаление названий порядков триад, обработка нулевых триад и т. п. □

Очевидно, что алгоритм перевода числительных конкретной пары языков будет иметь меньшую трудоемкость. Однако в этом случае потребуется разработка алгоритмов для каждой пары языков, что снижает эффективность данного подхода.

Написав алгоритмы приведения к модели и обратно, разработчик получает возможность переводить числительные на все языки, приведенные к модели ранее.

Данный подход к переводу числительных обладает следующими преимуществами:

1) для перевода не нужно составлять грамматику для числительных языка; достаточно описать отличия правил образования числительных языка и модели в виде алгоритмов коррек-

тировки;

2) алгоритм прямого перевода числительных языков L1 и L2 будет эффективней по трудоемкости; однако для n языков понадобится написать $n(n-1)$ различных алгоритмов перевода и с ростом n преимущество такого подхода нивелируется.

**Обобщение теории нормальных алгоритмов Маркова.** Для описания алгоритмов использовались нормальные алгоритмы Маркова. Пусть в некотором языке числительное записывается в обратном порядке: слева находятся единицы, а справа десятки, сотни и т. д. Чтобы привести такое числительное к модели и обратно, надо его обратить, то есть записать в обратном порядке, например преобразовать число 123 в число 321.

Введем обозначения: n – количество цифр Z, которые могут использоваться для записи строки; m – длина обращаемой строки ($m > 0$).

Решим задачу обращения с помощью нормальной схемы алгоритма Маркова. Схема нормального алгоритма обращения строк включает следующие подстановки.

1) $\lambda Z_i Z_j \rightarrow Z_j \lambda Z_i$ ;i, j = 1, 2, …, n

2) $\lambda Z_i \varphi \rightarrow \varphi Z_i$ ;i = 1, 2, …, n

3) $\lambda Z_i \rightarrow \varphi Z_i$ ;i = 1, 2, …, n

4) $\lambda\varphi \rightarrow \varnothing$ •

5) $\rightarrow \lambda +$

Трудоемкость алгоритма обращения $T_{обр}$ определяется следующим выражением:

$T_{обр} = m(m + 3)/2 + 2$,

то есть алгоритм является полиномиальным.

Рассмотрим следующую вариацию задачи обращения. Пусть в k ячейках находится m пронумерованных шариков, при этом $k \geq 2m$. Необходимо разместить их в обратном порядке. Для решения задачи необходимо взять предпоследний шарик и положить его за последним шариком (рисунок 5). Затем взять следующий шарик слева и поместить его в первую пустую ячейку справа от последнего шарика и так для всех шариков.

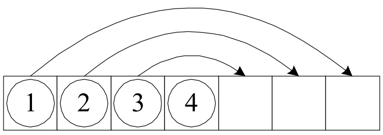


**Рисунок 5 – Метод решения задачи обращения**

Очевидно, что для решения задачи необходимо $m - 1$ операций поднятия и опускания шарика в ячейку (последний шарик остается на месте).

Алгоритм является линейным, однако предложенный нормальный алгоритм решения задачи полиномиальный по трудоемкости. Данное обстоятельство можно объяснить недостаточностью средств, предоставляемых теорией нормальных алгоритмов Маркова.

Предлагается обобщение теории нормальных алгоритмов Маркова, схема которого имеет следующий общий вид:

$P_1$: $P_1^{(1)} \rightarrow Q_1^{(1)}$; $P_2^{(1)} \rightarrow Q_2^{(1)}$; …; $P_{k_1}^{(1)} \rightarrow Q_{k_1}^{(1)}$ [•]

$P_2$: $P_1^{(2)} \rightarrow Q_1^{(2)}$; $P_2^{(2)} \rightarrow Q_2^{(2)}$; …; $P_{k_2}^{(2)} \rightarrow Q_{k_2}^{(2)}$ [•]

…

$P_d$: $P_1^{(d)} \rightarrow Q_1^{(d)}$; $P_2^{(d)} \rightarrow Q_2^{(d)}$; …; $P_{k_d}^{(d)} \rightarrow Q_{k_d}^{(d)}$ [•]

В схеме P только с нижним индексом и P с верхним и нижним индексами – подстроки.

Каждая строка схемы содержит метку P с нижним индексом и соответствующие этой метке подстановки.

Алгоритм над строкой R работает следующим образом. Схема просматривается сверху вниз, и выбирается та строка схемы, метка P которой содержится в строке R. Если метка P найдена, то выполняются последовательно слева направо подстановки, соответствующие метке P, в строке R. Подстановка состоит в замене первого вхождения слева подстроки из левой части подстановки на подстроку из правой части подстановки. После того, как все подстановки, соответствующие метке P, выполнены, схема просматривается вновь с первой подстановки.

Подстановки, помеченные пустым символом $\varnothing$, выполняются для любой строки R.

Алгоритм заканчивает выполнение в двух случаях:

1) ни одна метка P не встречается в строке R;

2) выполнена подстановка, заканчивающаяся символом «•».

Если алгоритм заканчивает свою работу (прекращает выполнение) со строкой R, то алгоритм считается применимым к строке R. Результатом работы алгоритма является строка R после применения всех подстановок.

Если алгоритм никогда не заканчивает свое выполнение (зацикливается) со строкой R, то алгоритм считается неприменимым к строке R и результат алгоритма неопределен.

Схему любого нормального алгоритма Маркова можно привести к схеме обобщенного алгоритма. Для этого достаточно добавить метки к каждой подстановке. Метка равна левой части подстановки. Каждой метке соответствует только одна подстановка.

Представим в виде обобщенной схемы предложенный алгоритм обращения:

1) $\lambda Z_i Z_j$: $\lambda Z_i Z_j \rightarrow Z_j \lambda Z_i$ ;i, j = 1, 2, …, n

2) $\lambda Z_i\varphi$: $\lambda Z_i\varphi \rightarrow \varphi Z_i$ ;i = 1, 2, …, n

3) $\lambda Z_i$: $\lambda Z_i \rightarrow \varphi Z_i$ ;i = 1, 2, …, n

4) $\lambda\varphi$: $\lambda\varphi \rightarrow \varnothing$ •

5) $\varnothing$: $\rightarrow \lambda+$

Теперь запишем схему алгоритма обращения строки с использованием аппарата обобщенных нормальных алгоритмов Маркова. Схема состоит из следующих подстановок:

1) $\lambda Z_i$: $\lambda Z_i \rightarrow \lambda$; $\varphi \rightarrow \varphi Z_i$ ;i = 1, 2, …, n

2) $\lambda\varphi$: $\lambda\varphi \rightarrow \varnothing$ •

3) $\varnothing$: $\rightarrow \lambda+$; $\rightarrow +\varphi$

Трудоемкость обобщенного алгоритма обращения $T_{об}$ определяется по следующей формуле:

$T_{об} = 2m + 3$.

Сократим число шагов за счет увеличения количества подстановок алгоритма. Модифицированный обобщенный алгоритм обращения будет состоять из следующих подстановок:

1) $\lambda Z_i\varphi$: $\lambda Z_i\varphi \rightarrow Z_i$ • ;i = 1, 2, …, n

2) $\lambda Z_i$: $\lambda Z_i \rightarrow \lambda$; $\varphi \rightarrow \varphi Z_i$ ;i = 1, 2, …, n

3) $\varnothing$: $\rightarrow \lambda+$; $\rightarrow +\varphi$

Отличие данного алгоритма от предыдущего алгоритма заключается в изменении порядка подстановок и вида подстановки 1. Изменение подстановки 1 позволило не выполнять последний шаг и закончить алгоритм быстрее.

Трудоемкость модифицированного обобщенного алгоритма обращения $T_{мод}$ определяется по следующей формуле:

$T_{мод} = 2(m – 1) + 3 = 2m + 1$.

Данные о трудоемкости и о количестве подстановок предложенных алгоритмов обращения представлены в таблице 4.

**Таблица 4**

| Название алгоритма обращения | Трудоемкость | Количество подстановок алгоритма | Тип алгоритма |
|---|---|---|---|
| Алгоритм обращения | $m(m + 3)/2 + 2$ | $n^2 + 2n + 2$ | Полиномиальный |
| Обобщенный нормальный алгоритм обращения | $2m + 3$ | $2n + 3$ | Линейный |
| Модифицированный обобщенный нормальный алгоритм обращения | $2m + 1$ | $3n + 2$ | Линейный |

Обобщение теории нормальных алгоритмов Маркова позволяет сделать алгоритмы более эффективными по сравнению с нормальными алгоритмами Маркова, уменьшая как число шагов, необходимых для решения задачи, так и количество подстановок в схеме. Модифицированный обобщенный алгоритм позволяет сократить трудоемкость по сравнению с обобщенным алгоритмом, однако количество подстановок в алгоритме увеличивается почти на n.

Трудоемкость любого обобщенного алгоритма будет не хуже нормального алгоритма Маркова, так как любой нормальный алгоритм можно записать в виде обобщенного алгоритма.

**Заключение.** Результатами работы являются следующие.

Разработана трехуровневая обобщенная модель числительного – усредненное представление числительных разных языков, позволяющая производить следующие операции по обработке числительных:

– переводить число в числительное и обратно;

– переводить числительные с одного языка на другие языки; перевод числительных с помощью модели эффективнее, чем перевод через число в качестве промежуточного этапа;

– склонять числительные по падежам;

– находить ошибки записи числительных.

Предложено обобщение нормальных алгоритмов Маркова, позволяющее разрабатывать эффективные алгоритмы как по пространственной, так и по временной сложностям.

## *Библиографический список*


1. *Hardegree G.* Symbolic Logic. A First Course. Third Edition. – McGraw-Hill, 1999. – 420 pp.

2. *Могилев А.В., Пак Н.И., Хеннер Е.К.* Информатика: учеб. пособие / под ред. Е.К. Хеннера. – 2-е изд., стер. – М.: Издательский центр «Академия», 2003. – 816 с.

3. *Мощенский В.А.* Лекции по математической логике. – Минск: Изд-во Белорус. ун-та, 1973. – 160 с.