

5. *Cong J., Fang J., Xie M. and Zhang Y.* MARS—A multilevel full-chip gridless routing system // IEEE Trans Comput.-Aided Design Integr. Syst. – 2005. – Vol. 24, № 3. – P. 382-394.
6. *Лебедев Б.К., Лебедев В.Б.* Поисквые процедуры канальной трассировки, базирующиеся на моделировании адаптивного поведения роя частиц в пространстве решений с упорядоченным лингвистическим шкалированием // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 15-22.
7. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
8. *Walter Banks, Gordon Hayward.* Fuzzy logic in embedded microcomputers and control systems. Published by Byte Craft Limited, Waterloo Ontario Canada, 2001.
9. *Курейчик В.М.* Особенности построения систем поддержки принятия решений // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 92-98.
10. *Чернышев Ю.О., Венцов Н.Н., Крупенин А.В., Щербаков А.В., Мухтаров С.А.* Адаптивная оптимизация запросов в современных системах управления базами данных. Монография. Краснодар: ФВАС, 2011.
11. *Курейчик В.М., Кажаров А.А.* Использование роевого интеллекта в решении NP-трудных задач // Известия ЮФУ. Технические науки. – 2010. – № 7 (120). – С. 30-36.

Статью рекомендовал к опубликованию д.т.н., профессор Б.К. Лебедев.

Чернышев Юрий Олегович – Донской государственный технический университет; e-mail: myvnn@list.ru; 344000, г. Ростов-на-Дону, пл. Гагарина, 1; тел.: 88632738582, кафедра автоматизации производственных процессов; профессор.

Венцов Николай Николаевич – e-mail: vencov@list.ru; кафедра информационных технологий; доцент.

Мухтаров Сергей Артурович – Филиал Военной академии связи (г. Краснодар); e-mail: samucht@list.ru; 350035, г. Краснодар, ул. Красина, 4; соискатель.

Chernyshev Yury Olegovich – Don State Technical University; e-mail: myvnn@list.ru; 1, Gagarin Square, Rostov-on-Don, 344000, Russia; phone: +78632738582; the department of automation of productions; professor.

Vencov Nikolay Nikolaevich – e-mail: vencov@list.ru; the department of information technologies; associate professor.

Mukhtarov Sergey Arturovich – Branch of Military academy of communication (Krasnodar); e-mail: samucht@list.ru; 4, Krasina street, Krasnodar, 350035, Russia; competitor.

УДК 004.912

Р.Ю. Вишняков, Ю.М. Вишняков

«ВЫЧИСЛИТЕЛЬНОЕ» ПРЕДСТАВЛЕНИЕ СМЫСЛА ТЕКСТОВЫХ ФРАГМЕНТОВ НА ОСНОВЕ ОБРАТНОЙ ПОЛЬСКОЙ ЗАПИСИ

Предлагаемая работа развивает интерпретационную модель представления смыслов предложений текстов научно-технического стиля и посвящена конструированию «вычислительного» представления функционала смысловыразительности текстовых фрагментов на основе идей обратной польской записи. В работе разрабатывается процедура представления функционала смысловыразительности текстового фрагмента в нотации обратной польской записи, формулируются правила ее получения и вычисления. Определяются понятия операции контекстного уточнения смысла, контекстной связки, дерева контекстной связки и конструируются их представление в нотации обратной польской записи. Рассматриваются свойства и особенности операции контекстного уточнения смысла и исследуются особенности контекстной связки. Все рассмотренные понятия и соотношения разбираются и иллюстрируются на конкретных примерах.

Контекст; контекстная связка; операция контекстного уточнения смысла; дерево контекстной связки; обратная польская запись.

R.Yu. Vishnyakov, Yu.M. Vishnyakov

“CALCULATING” REPRESENTATION OF THE TEXT SELECTION MEANING BASED ON THE REVERSE POLISH NOTATION

The paper shows build up of meanings representation interpretive model of the scientific and technical texts and concentrates on constructing text fragments meaning representation tools development.

The procedure of the text fragments meaning representation tools based on the reverse polish notation is developed as well as rules of its calculation is stated in the project. Conceptions of context-sensitive meaning computation, context link, context link tree are defined. Their representation based on the reverse polish notation is designed. Features and properties of context-sensitive meaning computation are considered. Context link peculiarities are carried out. All the above terms are illustrated with specific examples.

Context; context link; context-sensitive meaning computation; context link tree; reverse polish notation.

Постановка задачи. Повышенный интерес к выделению семантической составляющей текстов сформирован сегодняшней практикой информационных технологий и связан с возросшими запросами пользователей к качеству обработки текстовой информации. Это связано с тем, что используемые на практике подходы основаны на частотной парадигме релевантности, наверное, уже достигли предела своих возможностей и не могут удовлетворить возрастающие запросы пользователей. Предлагаемая работа развивает интерпретационную модель представления смыслов предложений текстов научно-технического стиля [2, 3] и посвящена конструированию «вычислительного» представления функционала смысловыразительности текстовых фрагментов на основе идей обратной польской записи.

Метод решения. Обратная польская запись (нотация) арифметических выражений была предложена польским математиком Я. Лукасевичем и получила широкое распространение в методах трансляции языков программирования в связи со способом ее вычисления на основе автомата с магазинной памятью, предложенного голландским ученым Дейкстрой [1]. В нотации обратной польской записи (ОПЗ) символ операции всегда завершает список ее операндов. Поэтому, если в традиционном представлении задана некоторая k -нарная операция $\Phi(x_1, x_2, \dots, x_k)$, то нотация ОПЗ этой операции имеет вид:

$$\underbrace{x_1, x_2, \dots, x_k}_{k \text{ операндов}} \quad \underbrace{\Phi}_{\text{операция}} . \quad (1)$$

Для указания k -арности операции (границы действия операции в линейной записи выражения) перед символом операции явно проставляют счетчик числа операндов k , после чего ОПЗ выражения (1) принимает вид:

$$\underbrace{x_1, x_2, \dots, x_k}_{k \text{ операндов}} \quad \underbrace{k}_{\text{счетчик операндов}} \quad \underbrace{\Phi}_{\text{операция}} . \quad (2)$$

Отличительной особенностью ОПЗ является линейный порядок ее вычисления слева направо за один проход, а отсюда и отсутствие в нотации скобок. По этой причине применительно к ОПЗ часто добавляют слово «бесскобочная» запись.

Теперь обсудим следующие понятия [2–4]:

1. Функционал смысловыразительности некоторого осмысленного текстового фрагмента $\alpha = x_1 x_2 \dots x_n$ представляется следующим образом:

$$\begin{aligned} S(\alpha) &= \Phi(S(x_1), S(x_2), \dots, S(x_n)); \\ S(\alpha) &\subset S(x_i), \end{aligned} \quad (3)$$

где x_1, x_2, \dots, x_n слова текстового фрагмента α , представляющего собой некоторое осмысленное выражение (словосочетание, предложение), x_i – главное слово фрагмента текста α , $S(x_j)$ – множество смыслов (семантических значений) слова x_j , $S(\alpha)$ – смысл (семантическое значение) текстового фрагмента α , $\Phi(S(x_1), S(x_2), \dots, S(x_n))$ – функционал смысловыразительности.

Данный функционал вычленяет из множества смысловых значений слов их определенные семантические значения и на их множестве строит смысловое значение всего выражения α , которое в общем случае является частью смыслового значения главного слова выражения.

2. Если в некотором фрагменте текста имеются такие слова x и y , что x – главное слово, y – зависимое слово, то операция контекстного уточнения смысла $\vec{\cap}$ определяется следующим образом:

$$\begin{aligned} S(\overline{x:\vec{y}}) &= S(x) \vec{\cap} S(y), \\ S(\overline{x:\vec{y}}) &\subset S(x), \end{aligned} \quad (1)$$

здесь: $S(x)$ и $S(y)$ смысловые значения слов x и y соответственно, $\vec{\cap}$ – операция контекстного уточнения смысла, $S(\overline{x:\vec{y}})$ – результат операции контекстного уточнения смысла.

3. Контекстная связка. Если в некотором фрагменте текста v есть некоторое слово и x_1, x_2, \dots, x_n прямо от него зависимые слова, то контекстной связкой для слова v назовем запись вида $v: \{x_1, x_2, \dots, x_n\}$. Смысловое значение данной контекстной связки определяется следующим образом:

$$S(v: \{x_1, x_2, \dots, x_n\}) = S(\overline{v, x_1}) \cap S(\overline{v, x_2}) \cap \dots \cap S(\overline{v, x_n}), \quad (5)$$

где $S(\overline{v, x_1}) = S(v) \vec{\cap} S(x_1)$; $S(\overline{v, x_2}) = S(v) \vec{\cap} S(x_2)$; ...; $S(\overline{v, x_n}) = S(v) \vec{\cap} S(x_n)$.

Поскольку операция контекстного уточнения смысла (4) не является симметричной [3], то при ее конструировании в нотации ОПЗ необходимо сохранить логику линейного следования операндов и их вычисления. С учетом данного обстоятельства в нотации обратной польской записи операция контекстного уточнения смысла запишется следующим образом:

$$S(y)S(v) \vec{\cap}. \quad (6)$$

Следует отметить, что направление стрелки над операцией контекстного уточнения смысла в нотации обратной польской записи меняет направление, что соответствует логике выполнения операции.

Нотация обратной польской записи контекстной связки $v: \{x_1, x_2, \dots, x_n\}$, определяемой выражением (5) с учетом (6), соответственно примет вид:

$$S(v: \{x_1, x_2, \dots, x_n\}) = S(\overline{v, x_1})S(\overline{v, x_2}) \dots S(\overline{v, x_n}) \cap, \quad (7)$$

где

$$S(\overline{v, x_1}) = S(x_1)S(v) \vec{\cap}; S(\overline{v, x_2}) = S(x_2)S(v) \vec{\cap}; \dots; S(\overline{v, x_n}) = S(x_n)S(v) \vec{\cap}. \quad (8)$$

Подставляя (8) в (7), получим следующее выражение в нотации обратной польской записи для вычисления смыслового значения контекстной связки:

$$S(v: \{x_1, x_2, \dots, x_n\}) = \underbrace{S(x_1)S(v) \vec{\cap}}_1 \underbrace{S(x_2)S(v) \vec{\cap}}_2 \dots \underbrace{S(x_n)S(v) \vec{\cap}}_n \underset{k=n}{\cap}. \quad (9)$$

В выражении (9) горизонтальными фигурными скобками выделены операции контекстного уточнения смысла, результаты которых представляют операнды операции пересечения \cap . Фигурной скобкой также отмечен счетчик числа операндов операции пересечения. Как это нетрудно заметить, в сконструированном выражении (9) основные свойства и особенности нотации обратной польской записи контекстной связки сохранены.

Сконструируем процедуры построения обратной польской записи контекстного смыслоуточнения для фрагментов текстов. Заметим, что построение обратной польской записи арифметических выражений осуществляется путем обхода дерева арифметического выражения слева направо [1], в результате чего формируется строка обратной польской записи. Однако данное правило обхода для нашего случая напрямую не применимо уже хотя бы потому, что дерево зависимостей фрагмента текста в виду наличия операции контекстного уточнения смысла не подобно дереву арифметических выражений. Для сохранения общей логики обхода дерева [1] модифицируем данное понятие применительно к вычислению функционала смысловыразительности фрагментов текстов. Для этого в дереве зависимостей фрагмента текста дуги пометим символами операций контекстного смыслоуточнения $\bar{\cap}$, а узлы – операциями пересечения \cap .

Теперь сконструируем обход дерева контекстной связки для построения ее обратной польской записи.

Алгоритм 1.

1. Обход выполняем слева направо, начиная с самого левого контекстного узла;
2. Для каждого контекстного узла строим обратную польскую запись операции контекстного уточнения смысла (6), которая определяется дугой зависимости и главным словом.
3. После обхода всех контекстных узлов в строку-результат записываем операцию пересечения со значением счетчика операндов, равным числу контекстных узлов.
4. Результатом обхода дерева контекстной связки является ее обратная польская запись.

Например, модифицированное дерево контекстной связки $v: \{x, y, z\}$ и его обход показаны на рис. 1.

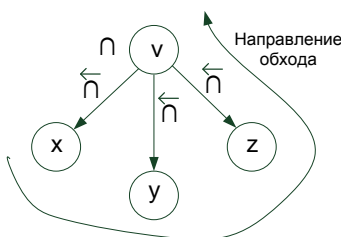


Рис. 1. Модифицированное дерево контекстной связки $v: \{x, y, z\}$

В результате обхода дерева (см. рис. 1) будет сформирована обратная польская запись контекстной связки $v: \{x, y, z\}$ вида:

$$\text{ОПЗ: } S(v: \{x, y, z\}) = \underbrace{S(x)S(v)}_1 \bar{\cap} \underbrace{S(y)S(v)}_2 \bar{\cap} \underbrace{S(z)S(v)}_3 \bar{\cap} \underset{k=3}{3} \cap. \quad (10)$$

Для построения обратной польской записи смыслового значения $S(\alpha)$ всего фрагмента текста α сконструируем следующий алгоритм:

Алгоритм:

1. Выполнить слева направо последовательный обход узлов дерева фрагмента α , начиная с самого левого нижнего узла.
2. Для каждого рассматриваемого узла построить ОПЗ его контекстной связки, после чего перейти к следующему узлу.

Справедливость сконструированного алгоритма рассмотрим на следующем примере фрагмента текста α :

« международное признание образовательных программ российских вузов »
 $\underbrace{\hspace{1.5cm}}_a \quad \underbrace{\hspace{1.5cm}}_b \quad \underbrace{\hspace{1.5cm}}_c \quad \underbrace{\hspace{1.5cm}}_d \quad \underbrace{\hspace{1.5cm}}_e \quad \underbrace{\hspace{1.5cm}}_f$

Закодируем латинскими буквами слова данного фрагмента текста. С учетом кодировки дерево зависимостей фрагмента текста показано на рис. 2

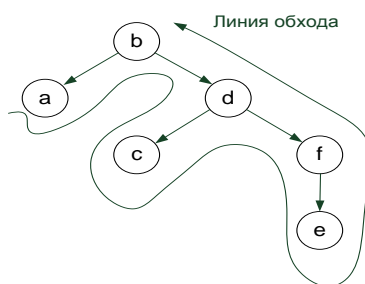


Рис. 2. Дерево зависимостей фрагмента текста

В результате сконструированного обхода дерева (см. рис. 2) получим ОПЗ функционала смысловыразительности $S(\alpha)$ фрагмента текста α вида:

$$S(\alpha) = \underbrace{S(a)S(b)}_1 \bar{\cap} \underbrace{S(c)S(d)}_2 \bar{\cap} \underbrace{S(e)S(f)}_3 \bar{\cap} \underbrace{S(d)}_4 \bar{\cap} \underbrace{\cap}_{k=2} \underbrace{S(b)}_{k=2} \bar{\cap} \underbrace{\cap}_{k=2}, \quad (11)$$

$\underbrace{\hspace{10em}}_5$
 $\underbrace{\hspace{10em}}_6$
 $\underbrace{\hspace{10em}}_7$

в которой горизонтальными фигурными скобками выделены конкретные операции уточнения смысла и пересечения с их операндами, выполняемые на каждом шаге вычисления, а номер шага вычисления указан непосредственно под фигурной скобкой.

Введем условно массив временных переменных $R = \{r_1, r_2, \dots, r_n\}$, в которых будем сохранять промежуточные результаты вычисления операций. Тогда пошаговое вычисление обратной польской записи (11) функционала смысловыразительности с запоминанием промежуточных результатов во временных переменных представлено в табл. 1.

Таблица 1

Вычисление ОПЗ функционала смысловыразительности с использованием временных переменных

№ пп	Текущая конфигурация ОПЗ	Выполняемая операция
1	$\underbrace{S(a)S(b)}_1 \bar{\cap} S(c)S(d) \bar{\cap} S(e)S(f) \bar{\cap} S(d) \bar{\cap} \cap S(b) \bar{\cap} \cap$	$r_1 = S(a)S(b) \bar{\cap}$
2	$r_1 \underbrace{S(c)S(d)}_2 \bar{\cap} S(e)S(f) \bar{\cap} S(d) \bar{\cap} \cap S(b) \bar{\cap} \cap$	$r_2 = S(c)S(d) \bar{\cap}$
3	$r_1 r_2 \underbrace{S(e)S(f)}_3 \bar{\cap} S(d) \bar{\cap} \cap S(b) \bar{\cap} \cap$	$r_3 = S(e)S(f) \bar{\cap}$

Окончание табл. 1

№ пп	Текущая конфигурация ОПЗ	Выполняемая операция
4	$r_1 r_2 \underline{r_3 S(d) \bar{n} 2 \cap S(b) \bar{n} 2 \cap}$	$r_4 = r_3 S(d) \bar{n}$
5	$r_1 r_2 r_4 2 \cap S(b) \bar{n} 2 \cap$	$r_5 = r_2 r_4 2 \cap$
6	$r_1 r_5 S(b) \bar{n} 2 \cap$	$r_6 = r_5 S(b) \bar{n}$
7	$r_1 r_6 2 \cap$	$r_7 = r_1 r_6 2 \cap$
	Процесс завершен, результат представлен в r_7	

Здесь выполняемая на каждом шаге операция выделена горизонтальной фигурной скобкой. Из примера нетрудно увидеть, что все характерные особенности и свойства обратной польской записи (отсутствие скобок и линейность процесса вычисления) при ее вычислении сохраняются.

Заключение. В работе рассмотрено «вычислительное» представление функционала смысловыразительности текстовых фрагментов в обратной польской записи, для чего в данной нотации сконструированы операция контекстного уточнения смысла и процедура вычисления смысла контекстной связки. Для контекстной связки предложена конструкция модифицированного дерева зависимостей, сконструирована процедура его обхода, результатом которой является ОПЗ функционала смысловыразительности текстового фрагмента. Показано, что основные особенности обратной польской записи в нотации обратной польской записи функционала смысловыразительности (однопроходной линейный порядок вычисления и отсутствие скобок) полностью сохранены.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лебедев В.Н. Введение в системы программирования. – М.: Статистика, 1975. – 312 с.
2. Вишняков Ю.М., Вишняков Р.Ю. Проблемы семантического информационного поиска // Труды международных научно-технических конференций «Интеллектуальные системы» (AIS'06) и «Интеллектуальные САПР» (CAD-2006). Научное издание в 3-х томах. Т. 2. – М.: Физматлит, 2006. – С. 308-314.
3. Вишняков Р.Ю. Контекстное уточнение смысла слов в связанном текстовом фрагменте // Сборник трудов VI Всероссийской научной школы-семинар молодых ученых, аспирантов и студентов «Семантическая интерпретация и интеллектуальная обработка текстов, их приложения в информационном поиске, хранении и обработке документов в электронных архивах и библиотеках». – Таганрог: Изд-во ТТИ ЮФУ, 2012. – С. 112-116.
4. Вишняков Р.Ю. Использование нотации обратной польской записи для контекстного уточнения смыслов фрагментов текстов // Сборник трудов Всероссийской научной школы-семинар молодых ученых, аспирантов и студентов «Семантическая интерпретация и интеллектуальная обработка текстов, их приложения в информационном поиске, хранении и обработке документов в электронных архивах и библиотеках». – Таганрог: Изд-во ТТИ ЮФУ, 2012. – С. 106-112.

Статью рекомендовал к опубликованию д.т.н., профессор В.П. Карелин.

Вишняков Ренат Юрьевич – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: rvishn.sfu.edu@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: +78634314485; кафедра системного анализа и телекоммуникаций; ассистент.

Вишняков Юрий Муссович – e-mail: vishn@tsure.ru; факультет автоматики и вычислительной техники; декан.

Vishnyakov Renat Yur'evich – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: rvishn.sfu.edu@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634314485; the department of system analysis and telecommunication; assistant.

Vishnyakov Yuriy Mussovich – e-mail: vishn@tsure.ru; the college of automation and computer engineering; dean.

УДК 004.272.43

А.Э. Саак

УГЛОВОЙ АЛГОРИТМ ДИСПЕТЧЕРИЗАЦИИ МАССИВАМИ ЗАЯВОК КРУГОВОГО ТИПА

Рассматривается круговой тип массива заявок пользователей на компьютерное обслуживание в Grid-системах, многопроцессорных вычислительных системах. Предлагается и исследуется угловой полиномиальный алгоритм назначения заявок кругового квадратичного типа. Проведено сравнение с оптимальным алгоритмом распределения вычислительных ресурсов на примере массивов натуральных ресурсных квадратов. Оценено качество алгоритма по эвристической мере на тех же массивах требований. Даются рекомендации о возможности использования углового полиномиального алгоритма в диспетчере как МВС, так и центра Grid-технологий.

Grid-система; многопроцессорная вычислительная система; диспетчирование; круговой квадратичный тип массива требований пользователей; угловой полиномиальный алгоритм.

A.E. Saak

AN ANGULAR ALGORITHM FOR SCHEDULING BY SETS OF CIRCULAR-TYPE USER TASKS

A circular-type task queue waiting for service in Grid systems or multiprocessor computer systems is considered. An angular polynomial algorithm for circular-type quadratic tasks assigning are proposed and considered. The results are compared with the optimal algorithm of computer resources scheduling in which sets of natural resource squares are taken as an example. Quality of the algorithm is estimated by an heuristic measure on the same sets of user tasks. Recommendations on possible use of the angular polynomial algorithm in a control system of a multiprocessor computer system or Grid system are given.

Grid system; multiprocessor computer system; scheduling; circular and quadratic type of set of user tasks; angular polynomial algorithm.

Введение. Классификация массивов заявок пользователей на круговой, гиперболический и параболический квадратичные типы [1] была дополнена работами с алгоритмами диспетчеризации, адаптированными под соответствующий вид требований. Так, в [2–5] рассматривались алгоритмы для кругового, в [2, 6, 7] – для гиперболического, в [2, 8] – для параболического типов. Качество алгоритмов оценивается эвристической мерой [6]. В настоящей статье предлагается и исследуется угловой алгоритм назначения на обслуживание заявок кругового квадратичного типа и приводится сравнение с оптимальным алгоритмом.

Полиномиальный угловой алгоритм. При представлении заявки пользователя для обслуживания диспетчером центра Grid-технологий или операционной системы МВС ресурсным прямоугольником $[(a(j), b(j))]$ горизонтальное и вертикальное измерения, соответственно, принимаются равными числу единиц ресурса времени $a(j)$ и процессоров $b(j)$, требуемому для обработки j -й заявки.