Заключение. Разработанные классификатор, словари и корпусы текстов могут быть использованы в различных задачах автоматической обработки ЕЯ: корректировки орфографии, машинного перевода, автоматического реферирования, информационного поиска и др., на начальном этапе лингвистического анализа текста (запроса пользователя) с целью аннотирования текста лексико-грамматическими кодами.

Кроме того, лексико-грамматические коды позволяют оптимизировать разработку так называемых патернов при выполнении синтаксического анализа текстов, поскольку обеспечивают их обобщение на уровне ЛГК, основанное преимущественно на морфологических признаках словоформ и правилах их комбинации.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Mcenery T., Wilson A. Corpus Linguistics. – Edinburgh: Edinburgh University Press, 1996. – 132 p.
2. Карпов В.А. Язык как система. – Мн.: Вышэйшая школа, 1992. – 302 с.
3. Лаптева О.А. Дискретность в устном монологическом тексте// Русский язык: Текст как целое и компонента текста (Виноградовские чтения XI). – М.: Наука, 1982. – С.77.
4. Богуславский И.М., Григорьев Н.В., Григорьева С.А. и др. Аннотированный корпус русских текстов: концепция, инструменты разметки, типы информации // Труды Международного семинара Диалог'2000 по компьютерной лингвистике и ее приложениям. – Протвино, 2000. – Том 2. – С. 41–47.
5. Методы автоматического анализа и синтеза текста / Пиотровский Р.Г., Билан В.Н., Боркун М.Н., Бобков А.К. – Мн.: Вышэйшая школа, 1985. – 222 с.
6. Leech G., Garside R., Bryant M. CLAWS4: The tagging of the British National Corpus // Proceedings of the 15th International Conference on Computational Linguistics (COLING 94) Kyoto, Japan, 1994. – P.622-628.
7. Совпель И.В. Инженерно-лингвистические принципы, методы и алгоритмы автоматической переработки текста. – Мн.: Вышэйшая школа, 1991. – 118 с.
8. Богуславский И.М., Григорьев Н.В., Григорьева С.А. и др. Аннотированный корпус русских текстов: концепция, инструменты разметки, типы информации // Труды Международного семинара Диалог'2000 по компьютерной лингвистике и ее приложениям. – Протвино, 2000. – Том 2. – С. 41–47.
9. Vintar Š. A Lexical Analysis of the IJS-ELAN Slovene-English Parallel Corpus // Vintar, Š. (ed.) Proceedings of the workshop Language Technologies –Multilingual Aspects. Ljubljana: Faculty of Arts., 1999.
10. Курс беларускай мовы: Падручник / Л.І. Сямешка, І.Р. Шкраба, З.І. Бадзевіч. – Мн.: Універсітэцкае, 1996. – 654 с.
11. Русский язык. Часть I. Изд. 3-е, испр. и доп. / А.М. Бордович, Н.И. Гурский, Е.С.Хмелевская, Э.К. Бирилло. – Мн.: Вышэйшая школа, 1977. – 416 с.
12. Григорян В.М. Соотношение категорий субстантивности и залоговости // Русский язык: Проблемы грамматической семантики и оценочные факторы в языке (Виноградовские чтения XIX–XX). – М.: Наука, 1991. – С.3–11.

Материал поступил в редакцию 22.01.08

RUBASHKO N.K. Development of lexical and grammatical qualifiers of Russian and Byelorussian languages and their application

The purpose of the present job is the description of principles of development of lexical and grammatical qualifiers for Russian and Byelorussian languages and their use at creation of the dictionaries and cases of the texts.

УДК 004.89:004.4

Постаногов Д.Ю.

РАСПОЗНАЮЩИЕ ШАБЛОНЫ НА ОСНОВЕ РАСШИРЕННЫХ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В ЗАДАЧАХ АВТОМАТИЧЕСКОГО АНАЛИЗА ТЕКСТА

Введение. Эффективность применения интеллектуальных информационных систем (ИИС) главным образом определяется качеством и количеством знаний, содержащихся в используемых ими базах знаний (БЗ) [1] [2] [3]. Возможность разработки действенной БЗ достаточно большого объема, в свою очередь, обуславливается способами организации взаимодействия разработчиков ИИС, как правило – инженеров по знаниям, экспертов и программистов [4].

Одним из базовых компонентов ИИС и, прежде всего, различных систем автоматической обработки текстов является лингвистический процессор (ЛП), основу которого, в свою очередь, составляет лингвистическая база знаний (ЛБЗ). Важным аспектом в обеспечении эффективности их разработки и оптимального устройства является степень интегрированности лингвистических знаний о естественном языке (ЕЯ) в программный код системы. Данный вопрос касается как *декларативной* части ЛБЗ, к которой можно отнести словари языка и списки различного назначения, вероятностные модели и другие данные статического характера, так и наиболее значительной, *процедурной* части ЛБЗ, которую составляют правила различного рода в зависимости от соответствующего им этапа анализа текста и структурного уровня ЕЯ, например, правила лексико-грамматического, синтаксического и семантического анализа предложений, в совокупности и в сочетании с декларативной частью ЛБЗ представляющие собой определенную лингвистическую модель ЕЯ.

Очевидно, что программист, описывающий алгоритм работы ЛП на определенном языке программирования, чаще всего, не обладая глубокими специальными знаниями о структуре ЕЯ, не способен самостоятельно в сжатые сроки реализовать достаточное количество лингвистических правил, которые бы обеспечили обработку текста с высоким качеством. Более того, организация взаимодействия программиста с экспертом–лингвистом, который лишь передает ему свои знания о ЕЯ напрямую либо через посредничество инженера по знаниям, также не позволяет достичь высокой эффективности разработки ЛП в силу необходимости неоднократного внесения правок в программный код при выполнении трудоемких процедур проверки используемых лингвистических гипотез, тестирования и отладки системы. В этой связи наиболее перспективным подходом к разработке процедурной части ЛБЗ является явное обособление лингвистических правил от программного кода за счет использования проблемно-ориентированных языков [5] как средств описания знаний о ЕЯ самим экспертом-лингвистом. Данный подход обеспечивает открытость и расширяемость системы в смысле возможности явного разделения труда экспертов данной предметной области, т. е. лингвистов, которые описывают декларативные и процедурные знания о ЕЯ на таких языках, и программистов, разрабатывающих аппарат их интерпретации [6]. Основными задачами инженера по знаниям в этом случае являются: взаимодействие с экспертом-лингвистом с целью определения требований к описательным возможностям этих языков, проектирование соответствующих систем обозначений — *нотаций*, а также описание способов их интерпретации, в дальнейшем реализуемых программистом в виде соответствующих алгоритмов.

В данной статье рассматриваются вопросы разработки проблемно-ориентированных языков и аппарата их эффективной интерпретации для описания систем так называемых *шаблонов*, являющихся одним из формальных представлений лингвистических правил и имеющих чрезвычайно широкий спектр применения в теории и практике инженерно-лингвистического подхода [7] к обработке ЕЯ.

Применение регулярных выражений в шаблонах. В общем случае правила ЛБЗ реализуются в виде так называемых распознающих шаблонов, которые главным образом сводятся к форме

***Постаногов Денис Юрьевич**, начальник отдела разработки средств интеллектуализации информационных систем ИП «Инвеншион Машин».*

Таблица 1. *Примеры УСЛОВИЙ распознающих шаблонов, представленных в форме АГ*

Задача лингвистического анализа текста	Пример поля УСЛОВИЕ	Терминальные символы АГ
Разбиение текста на предложения посредством определения первого символа предложения	[a-z0-9)"](.!?]|\.\.\.)[ ]<[A-Z]>	одиночные символы текста, составляющие регулярное выражение
Поиск глаголов в инфинитивной форме	"in" "order" "to" < . > "a"|"the"	конкретные слова предложения (знак «.» соответствует любому слову)
Выделение именных групп	< (AT|JJ|NN)* NN >	ЛГК слов (AT – артикль, JJ – имя прилагательное, NN – имя существительное)

продукционных правил вида «если УСЛОВИЕ, то ДЕЙСТВИЕ», где поле УСЛОВИЕ обеспечивает распознавание тех конфигураций входных данных, для которых применение соответствующего ДЕЙСТВИЯ является наиболее корректным в целях достижения желаемого результата.

Наиболее распространенной в компьютерной лингвистике формой представления поля УСЛОВИЕ шаблонов являются нотации с использованием регулярных выражений [8] [9] и других автоматных грамматик (АГ) [10], терминальными символами которых являются конкретные слова ЕЯ [11] либо лексико-грамматические классы (ЛГК) слов [12]. При этом поле ДЕЙСТВИЕ чаще всего задается в программном коде систем, в которых применяются такие шаблоны. В таблице 1 приведены примеры поля УСЛОВИЕ шаблонов для решения различных задач лингвистического анализа текста (здесь и далее примеры будут даваться для английского языка).

Например, первый шаблон описывает одно из возможных правил разбиения входного текста на предложения за счет поиска их начальных (граничных) символов. Здесь в качестве ДЕЙСТВИЯ подразумевается выделение латинских букв верхнего регистра, соответствующих подвыражению шаблона, заключенному в треугольные скобки, и следующих за последовательностью символов: латинская буква нижнего регистра, цифра, закрывающая круглая скобка или кавычка, далее – знак препинания («.»,«!»,«?» или «…»), далее – пробел.

Большая распространенность данных форм представления распознающих шаблонов главным образом обуславливается достаточным уровнем разработанности формального математического аппарата интерпретации АГ — теории конечных автоматов, использование которых, как правило, позволяет свести алгоритмическую трудоемкость задачи анализа текста к $O(n)$, где n – количество анализируемых элементов входной цепочки.

Однако, как показали эксперименты, для разработки лингвистических моделей, соответствующих более высоким уровням ЕЯ, в частности, семантическому [13], применение в распознающих шаблонах исключительно нотаций АГ не является достаточным в силу следующих причин:

- невозможность одновременного использования терминальных символов АГ (одиночных символов текста, слов и их лексико-грамматических классов) при описании одного распознающего шаблона;
- отсутствие в нотациях операторов отрицания, исключения и пересечения выражений, что значительно затрудняет процесс описания сложных условий в распознающих шаблонах;
- необходимость задействования в шаблонах различных словарных списков, в том числе учитывающих назначенные словам лексико-грамматические классы;
- невозможность прямого задания в поле ДЕЙСТВИЕ различных лингвистических преобразований анализируемого контекста в зависимости от срабатывающих условий правил.

Указанные недостатки могут быть преодолены введением соответствующих расширений нотации регулярных выражений, сохраняющих при этом их основные преимущества: высокую описательную мощность нотации и низкую трудоемкость анализа входных цепочек.

Понятие расширенных регулярных выражений. Введем следующие обозначения: $L(P)$ – множество цепочек, порождаемых регулярным выражением P, иначе – язык, порождаемый P. $T(A)$ – язык, принимаемый детерминированным конечным автоматом A.

Определение: Пусть $\Re$ – конечное множество одноместных предикатов, определенных на множестве V – алфавите терминальных символов, такое что: $\forall \rho \in \Re,\ \forall a \in V \Rightarrow \rho(a) \in \{истина, ложь\}$. Тогда расширенным регулярным выражением, заданным одноместными предикатами из множества $\Re$, является:

1) λ – пустое выражение, порождающее язык, состоящий лишь из пустой цепочки терминальных символов: $L(\lambda) = \{\varnothing\}$;
2) ρ $(\rho \in \Re)$ – одноместный предикат, которому соответствует множество цепочек $L(\rho) = \{a_i\}$, состоящих из одного терминального символа $a_i \in V$: $\rho(a_i) = истина$;
3) PQ – конкатенация двух расширенных регулярных выражений P и Q, порождающая язык $L(PQ) = \{x_i y_j\}$: $x_i \in L(P)$, $y_j \in L(Q)$;
4) $P|Q$ – логическое ИЛИ расширенных регулярных выражений P и Q, порождающее язык $L(P|Q) = L(P) \cup L(Q)$;
5) P^* – итерация расширенного регулярного выражения P, порождающая язык $L(P^*) = \{\varnothing, x_i, x_i x_i, x_i x_i x_i, \ldots\}$, где цепочки $x_i \in L(P)$;
6) $!P$ – отрицание расширенного регулярного выражения P, порождающее язык $L(!P) = \{x_i\}$: $x_i = a_{i1}, a_{i2}, \ldots a_{in}$, $a_{ij} \in V$, $n \geq 0$: $x_i \notin L(P)$;
7) $P\&Q$ – пересечение расширенных регулярных выражений P и Q, порождающее язык $L(P\&Q) = L(P) \cap L(Q)$.

Помимо указанных операций конкатенации, ИЛИ, итерации, отрицания и пересечения расширенных регулярных выражений можно также ввести дополнительные операции:

8) $P+$ – непустая итерация: $P+ = PP^*$;
9) $P?$ либо $[P]$ – опциональное вхождение: $P? = [P] = P|\lambda$;
10) $P–Q$ – вычитание выражений: $P–Q = P\&(!Q)$.

По сути, расширенные регулярные выражения являются обобщением традиционных регулярных выражений за счет того, что в них в качестве терминальных символов используются предикаты. Таким образом, условия на элементы входной цепочки налагаются посредством обращения к свойствам элементов множества V. Еще одним существенным их отличием от регулярных выражений является введение операций отрицания и пересечения выражений.

Решение задачи эффективного анализа текста по распознающим шаблонам, основанным на расширенных регулярных выражениях, требует доказательства того, что, по крайней мере, существует возможность применения детерминированного конечного автомата для разбора цепочек по расширенным регулярным выражениям.

Теорема: Для любого расширенного регулярного выражения R, заданного предикатами из $\Re$ в алфавите V, существуют сюръективное отображение алфавита $\gamma_\Re : V \rightarrow V'$ и детерминированный конечный автомат A с алфавитом V', такой что: $T(A) = \gamma_\Re(L(R))$.

Доказательство. Пусть $|\Re| = m$ и $\rho_j \in \Re$, $1 \leq j \leq m$. Введем разбиение множества V на 2^m непересекающихся подмножеств $C_k \subseteq V$, $1 \leq k \leq 2^m$:

$C_1 = \{a_i\}$, $a_i \in V$: $\rho_1(a_i) = ложь$, $\rho_2(a_i) = ложь$, …, $\rho_m(a_i) = ложь$.

$C_2 = \{a_i\}$, $a_i \in V$: $\rho_1(a_i)$ = *истина*, $\rho_2(a_i)$ = *ложь*, …, $\rho_m(a_i)$ = *ложь*.

…

$C_2^m = \{a_i\}$, $a_i \in V$: $\rho_1(a_i)$ = *истина*, $\rho_2(a_i)$ = *истина*, …, $\rho_m(a_i)$ = *истина*.

Пусть множество $V' = \{k\}$, $1 \le k \le 2^m$. Тогда положим $\gamma_{\Re}$ – сюръективное отображение каждого элемента алфавита $a \in V$ в номер множества, которому принадлежит данный элемент: $\gamma_{\Re}(a) = k : a \in C_k, k \in V'$.

Обозначим $\gamma_{\Re}(x)$, где $x = a_1, a_2, \ldots a_n$, $n \ge 0$ – отображение цепочки символов алфавита $a_i \in V$: $\gamma_{\Re}(x) = \gamma_{\Re}(a_1, a_2, \ldots a_n) = \gamma_{\Re}(a_1), \gamma_{\Re}(a_2), \ldots \gamma_{\Re}(a_n)$. Тогда $\gamma_{\Re}(\varnothing) = \varnothing$ и $\gamma_{\Re}(\lambda) = \lambda$. Введем также отображение множества цепочек: $\gamma_{\Re}(L) = \gamma_{\Re}(\{x_i\}) = \{\gamma_{\Re}(x_i)\}$.

Далее доказательство производится индуктивно.

Допустим, что R представлено пустым расширенным регулярным выражением λ. Очевидно, что существует детерминированный конечный автомат A_λ с единственным состоянием, являющимся конечным, и пустым множеством переходов, такой что: $T(A) = \{\lambda\} = \{\gamma_{\Re}(\lambda)\} = \gamma_{\Re}(L(\lambda))$.

Пусть R представлено предикатом $\rho_j \in \Re$, $1 \le j \le m$. Обозначим $V_j' \subseteq V'$ –множество всех номеров $\{k_{jl}\}$, соответствующих C_{kjl}, элементы которых $a \in C_{kl} \subseteq V : \rho_j(a)$ = *истина* ($|V_j'| = m / 2$). Тогда, согласно теореме Клини [8], существует детерминированным конечный автомат $A_{\rho j}$, принимающий язык регулярного выражения $(k_{j1} \mid k_{j2} \mid \ldots \mid k_{jl})$, где $k_{jl} \in V_j'$. Рассмотрим цепочку, состоящую из одного терминального символа $a \in V$. Если $\rho_j(a)$ = *истина*, то $\gamma_{\Re}(a) = k \in V_j' \Rightarrow \gamma_{\Re}(a) \in T(A_{\rho j})$. Аналогично, если $\rho_j(a)$ = *ложь*, то $\gamma_{\Re}(a) = k \notin V_j' \Rightarrow \gamma_{\Re}(a) \notin T(A_{\rho j})$. Следовательно, $\gamma_{\Re}(L(\rho_j)) = T(A_{\rho j})$.

Пусть R = PQ. Тогда существует детерминированный конечный автомат A_{PQ}, полученный, согласно теореме [14, с. 121], детерминизацией из недетерминированного автомата, являющегося последовательным объединением автомата A_P (соответствующего выражению P) и A_Q (соответствующего выражению Q) с добавлением ε-перехода от конечных состояний A_P к начальным состояниям A_Q. При этом, очевидно, что при $x_i \in L(P)$, $y_j \in L(Q) \Rightarrow \gamma_{\Re}(L(PQ)) = \{\gamma_{\Re}(x_i y_j)\} = \{\gamma_{\Re}(x_i)\ \gamma_{\Re}(y_j)\} = T(A_{PQ})$. Аналогично доказывается существование автоматов $A_{P|Q}$ и A_{P^*}.

Автомат $A_{!P}$, соответствующий расширенному регулярному выражению !P, может быть получен из автомата A_P путем детерминизации его дополнения, то есть переименования неконечных состояний в конечные и наоборот, а также добавления нового конечного состояния f_0 и переходов к нему из $\forall s$ по $\forall k$, если данного перехода нет в исходном автомате A_P.

Автомат $A_{P\&Q}$ для выражения P&Q строится путем детерминизации недетерминированного автомата с начальным состоянием s_0 и ε-переходами от него к начальным состояниям автоматов A_P и A_Q. В этом случае конечными состояниями детерминированного автомата $A_{P\&Q}$ будут являться состояния, которые эквивалентны одновременно конечным состояниям из A_P и A_Q. *Теорема доказана.*

Распознающие шаблоны на основе расширенных регулярных выражений WRE.

Определение: Расширенным регулярным выражением WRE (Word-based regular expression) будем называть расширенное регулярное выражение на алфавите, элементами которого являются пары вида (*Word, Tag*) – слова (последовательности символов Word) с назначенными им ЛГК (Tag), заданное предикатами следующих типов:

- предикаты проверки слова, имеющие запись вида "word" и принимающие значение *истина* на элементах алфавита, слово которых совпадает с *word*;
- предикаты проверки ЛГК, имеющие запись вида Tag и принимающие значение *истина* на элементах алфавита, назначенный ЛГК которых совпадает с *Tag*;
- предикаты проверки слова по регулярному выражению, имеющие запись вида 'regexp' и принимающие значение *истина* на элементах, последовательность символов слова которых принадлежит языку, порождаемому регулярным выражением *regexp*;
- предикаты проверки слова по лексико-грамматическому словарю, имеющие запись вида _Tag и принимающие значение *истина* на элементах, для слов которых в лексико-грамматическом словаре одним из возможных ЛГК допускается ЛГК *Tag*;
- предикаты проверки слова по специальному словарю семантических классов слов, имеющие запись вида @Class и принимающие значение *истина* на элементах, слова которых входят в класс *Class* согласно данному словарю.

Введены также следующие дополнения: допускаются логические комбинации предикатов в условиях на анализируемое слово с использования знака «|» (логическое ИЛИ) либо «/» (логическое И между различными типами предикатов), а также задание нескольких предикатов одного типа в одном условии без разделителей (логическое И); знак «.» (точка) обозначает логическое ИЛИ всех элементов алфавита, то есть соответствует любому слову; знак «!», находящийся непосредственно перед предикатом, является обозначением отрицания одиночного предиката.

Выражение WRE допускает непустые цепочки слов, за исключением цепочек, в которых отсутствует хотя бы одно слово с ЛКГ VBN (причастия прошедшего времени в пассивном залоге), а также цепочек, в которых перед словами с ЛКГ VBN контактно расположены любые словоформы вспомогательного глагола «have», после которых может идти отрицательная частица. То есть, допускаются, например, следующие цепочки слов алфавита WRE: «has_HVZ been_BEN evaporated_VBN», «was_BEZ calculated_VBN», «will_MD be_BE turned_VBN off_RP», при этом не допускаются следующие цепочки: «finish_VB», «had_HVD n't_XNOT warned_VBN».

Как следует из доказанной выше теоремы, для любого расширенного регулярного выражения WRE можно построить эквивалентный по принимаемому языку детерминированный конечный автомат с соответствующим преобразованием исходного алфавита терминальных символов к алфавиту мощности 2^m, где m – число рассматриваемых в выражении предикатов. Очевидно, что большая часть из 2^m получаемых комбинаций предикатов тождественно-ложны в силу их определения и могут быть исключены из рассмотрения (например, комбинации предикатов, такие как VBN∧XNOT и "have"∧"had", задают невозможные условия на наличие двух различных ЛКГ или двух вариантов написания, назначенных одному и тому же слову в предложении), поэтому объемы данных, необходимых для осуществления разбора цепочек по WRE, на практике оказываются достаточно малыми при наличии соответствующей проверки выполнимости комбинаций предикатов. Способ построения и применения автомата и алфавитного преобразования вытекает из доказательства теоремы.

Таким образом, WRE выступает в качестве достаточно простой и удобной формы записи поля УСЛОВИЕ шаблонов. При этом, во-первых, преодолеваются по крайней мере три первых недостатка использования регулярных выражений, указанных ранее, и, во-вторых, за счет применения детерминированных конечных автоматов обеспечивается эффективная интерпретация системы шаблонов с использованием WRE за время $O(n)$, где n – количество слов анализируемого контекста.

Пример выражения WRE для задачи распознавания в предложении глагольной группы в пассивном залоге:

```
( .+ ) - ( ( !VBN+ ) |
           ( .* ('have|\'v'/HV | "has"/HVZ | "had"|"'d"/HVD|HVN |
                 "having"/HVG ) ["not"|"n't"/XNOT] VBN .* ) )
```

Пример шаблона с лингвистическим преобразованием текстового фрагмента:

```
УСЛОВИЕ:        .* <0 (NN|NNS)+ > "can"|"may"|"must"|"will" "be" <1 VBN > .*
ДЕЙСТВИЕ:       "to_TO" + 1->VB + "the_ATI" + 0
```

Кроме того, выражения WRE могут быть использованы не только для определения срабатывания условия на цепочку слов, но и в целях ее разбора на составляющие компоненты либо определения по ней набора срабатывающих условий из заданного списка. Для этого в нотацию WRE дополнительно введены извлекающие нумерованные скобки вида <$_N$ … >. Извлечение соответствующих скобкам контекстов анализируемой фразы, а также определение набора сработавших условий происходит за счет построения автоматов специального вида: Мура или Мили [15] в зависимости от задачи. В таком случае, реализация шаблонов общего вида «если УСЛОВИЕ, то ДЕЙСТВИЕ» для анализа и преобразования непрерывных текстовых фрагментов, в том числе представленных в виде последовательности слов с назначенными им ЛГК, может осуществляться с использованием соответствующего проблемно-ориентированного языка, описывающего набор правил, со следующими полями:

- УСЛОВИЕ, которое задается с помощью выражений WRE с извлекающими нумерованными скобками;
- ДЕЙСТВИЕ, которое задается с помощью определенной нотации для описания лингвистических преобразований фрагментов входа, извлеченных нумерованными скобками.

Шаблон с лингвистическим преобразованием текстового фрагмента задает поиск подлежащего, состоящего из последовательности имен существительных (NN – имя существительное в единственном, NNS – во множественном числе) и следующего за ним сказуемого в пассивной форме с использованием определенных модальных глаголов. В случае обнаружения таких конструкций подлежащее извлекается в скобку с номером 0, главный глагол сказуемого – в скобку с номером 1. Далее, в результате указанного действия анализируемая фраза преобразуется в конструкцию с активным залогом. Например, аннотированное ЛГК предложение: «The_ATI computer_NN monitors_NNS may_NN be_BE manufactured_VBN using_VBG LCD_NP technology_NN ._.» в результате применения данного шаблона преобразуется во фразу «to_TO manufacture_VB the_AT computer_NN monitors_NNS».

Заключение. Описанный в данной статье аппарат расширенных регулярных выражений WRE обладает рядом существенных преимуществ по сравнению с регулярными выражениями на алфавите одиночных символов в применении к разработке распознающих лингвистических моделей анализа текста. Нотации, использующие WRE, обладают более высокой описательной мощностью и могут применяться для эффективной реализации шаблонов на различных уровнях ЕЯ. При этом, за счет доказанной возможности задействования аппарата конечных детерминированных автоматов для анализа входных цепочек длины n по выражениям WRE с трудоемкостью $O(n)$, достигается достаточно высокая скорость обработки текста.

В сочетании с использованием определенных нотаций описания преобразующих действий, данный подход может применяться при разработке распознающих шаблонов экспертами-лингвистами без участия инженера по знаниям и программиста. Эффективность использования таких проблемно-ориентированных языков подтверждается также нашим опытом их практического применения в качестве основы инструментария экспертов-лингвистов при разработке правил процедурной части ЛБЗ ЛП известной системы Goldfire™ [16] на различных уровнях естественного языка.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Искусственный интеллект: В 3 кн. Кн. 1. Системы общения и экспертные системы: Справочник / Под ред. Э. В. Попова. — М.: Радио и связь, 1990. — 464 с.: ил.
2. Городецкий В. И. Современное состояние технологии извлечения знаний из баз и хранилищ данных (часть 1) / В. И. Городецкий, В. В. Самойлов, А. О. Малов // Новости искусственного интеллекта. — 2002. — № 3 (50). — С. 3-12.
3. Orchard R. A. Knowledge Processing: A Semantics For The Klir Hierarchy of General Systems // Selection of papers from the International Workshop (EUROCAST'89). — Las Palmas, Spain, 1989.
4. Гаврилова Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. — СПб.: Питер, 2000. — 384 с.: ил.
5. Heering J. Domain-specific languages / J. Heering, M. Mernik, A. M. Sloane // Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03). — Big Island, USA, 2003. — P. 323-323.
6. Kolovos D. S. Requirements for Domain-Specific Languages / D. S. Kolovos, R. F. Paige, T. Kelly, F. A. C. Polack // Proceedings of the 1st ECOOP Workshop on Domain-Specific Program Development (DSPD 2006). — Nantes, France, 2006.
7. Пиотровский Р. Г. Инженерная лингвистика и теория языка. — Л.: Наука, 1979. — 111 с.
8. Kleene S. C. Representation of events in nerve nets and finite automata // Automata studies, Ann. of Math. Studies. — Princeton: Princeton Univ. Press, 1956. — no. 34. — P. 3-41.
9. Jackson P. Natural language processing for online applications: text retrieval, extraction and categorization / P. Jackson, I. Moulinier. — Amsterdam: John Benjamins Publishing Company, 2002. — 227 p.
10. Partee B.H. Studies in linguistics and philosophy: Vol. 30. Mathematical Methods in Linguistics / B. H. Partee, A. ter Meulen, R. E. Wall. — Dordrecht: Kluwer Academic, 1990. — 692 p.
11. Ö. Dahl. The Growth and Maintenance of Linguistic Complexity. — Amsterdam: John Benjamins Publishing Company, 2004. — 337 p.
12. Voutilainen A. Helsinki Taggers and Parsers for English // J.M.Kirk (ed.). Corpora galore: analyses and techniques in describing English : papers from the 19th International Conference on English Language Research on Computerised Corpora (ICAME 1998). — no. 30. — Amsterdam: Rodopi, 2000.
13. Hazez S. B. Modeling Textual Content in Linguistic Pattern Matching // A. Gelbukh (ed.). Computational Linguistics and Intelligent Text Processing: Second International Conference (CICLing'2001). — Mexico, 2001. — P. 92-95.
14. Rabin M. O. Finite automata and their decision problems / M. O. Rabin, D. Scott // IMB Journal of Research and Development. — 1959. — no. 2. — P. 114-125.
15. Брауэр В. Введение в теорию конечных автоматов. — М.: Радио и связь, 1987. — 392 с.: ил. — (В пер.).
16. Goldfire. — http://goldfire.com.

Материал поступил в редакцию 22.01.08

POSTANOGOV D.Y. Recognizing patterns on the basis of the extended regular expressions in tasks of the automatic analysis of the

The notation of the extended regular expressions is offered as a basis of the problem-oriented languages used by the experts by development of recognizing linguistic models in structure of base of knowledge of the linguistic processor. Are given the proof of existence of the final determined automatic device equivalent on accepted language to any extended regular expression, both way of his construction and uses for effective interpretation of the given notation ensuring high speed of processing of the entrance text.