УДК 519.768:17

И. А. КОРОЛЬ, И. В. СОВПЕЛЬ

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПЕРЕВОДА КЛЮЧЕВЫХ СЛОВ (СПЕРКС-1). II

На основании утверждений 1—3, приведенных в первой части работы [1], нами разработан алгоритм построения ядра L^* в банке равномощных языков L по заданному множеству словарей $\bar{L}$. Предварительно введем некоторые определения.

Пусть неориентированный граф $G=(X, A)$ является лесом и $T_i=(X_i, A_i)$, $T_j=(X_j, A_j)$ — две его произвольные компоненты. С к л е й к о й деревьев T_i и T_j назовем дерево $T_{ij}=(X_{ij}, A_{ij})$, где $X_{ij}=X_i \cup X_j$, $A_{ij}=A_i \cup A_j \cup a_{i_k j_n}$, i_k, j_n — любые из номеров вершин, принадлежащих соответственно множествам X_i и X_j. Ребро $a_{i_k j_n}$ назовем ф и к т и в н ы м ребром графа $G=(X, A)$. Если $|X_i|=n_i$, $|X_j|=n_j$, то, очевидно, может быть получено $n_i \cdot n_j$ различных склеек деревьев T_i и T_j. Обозначим $G=(X, A)=G^{(0)}$. Будем считать, что процедура построения слейки T_{ij} осуществляется о п е р а т о р о м с к л е й к и Δ. В результате выполнения оператора Δ для любых двух компонент графа $G^{(0)}$ образуется новый граф $G^{(1)}=\Delta(G^{(0)})$. Если граф $G^{(1)}$ является лесом, то на нем также можно выполнить оператор Δ и т. д. Процедуру последовательного выполнения оператора Δ на исходном графе $G^{(0)}$ назовем с к л е и в а н и е м леса и обозначим ее рекуррентной формулой $G^{(s)}=\Delta(G^{(s-1)})$, где $G^{(s-1)}$ — лес, $s=1, 2, \ldots, s^*$, $G^{(s^*)}$ — дерево. Дерево $G^{(s^*)}$ назовем с к л е й к о й л е с а $G^{(0)}$.

Процедура построения ядра L^* осуществляется в соответствии со следующим общим алгоритмом.

Шаг 1. Начало.

Шаг 2. По данному множеству $\bar{L}$ построить базовый граф $G=(X, A)$. Если $A=\varnothing$, то перейти к следующему шагу, иначе — к шагу 6.

Шаг 3. На вершинах графа $G=(X, A)$ построить полный неориентированный граф K_m.

Шаг 4. Построить множество T всех остовных деревьев графа K_m.

Шаг 5. Выбрать из множества T наиболее «подходящее» остовное дерево T^*. Все ребра дерева T^* пометить и перейти к шагу 11.

Шаг 6. Если $G=(X, A)$ — связный граф, то перейти к следующему шагу, иначе — к шагу 9.

Шаг 7. Построить множество T всех остовных деревьев графа $G=(X, A)$.

Шаг 8. Выбрать из множества T «наиболее подходящее» остовное дерево T^* и перейти к шагу 12.

Шаг 9. Для каждой компоненты T_i графа $G=(X, A)$ построить множество всех остовных деревьев и выбрать в каждом из полученных множеств наиболее «подходящее» остовное дерево T_i^*.

Шаг 10. Построить склейку T^* леса, компонентами которого являются деревья T^{i_*}. Все фиктивные ребра дерева T^* пометить.

Шаг 11. Составить список T' помеченных ребер дерева T^*.

Шаг 12. Конец.

В настоящей работе подробности реализации описанного алгоритма не приводятся, однако необходимо отметить следующее. Выбор наиболее «подходящего» остовного дерева и склейки предполагает возможность наличия на входе алгоритма списка «нежелательных» словарей, что равносильно заданию весов для ребер графа. Задача выбора в этом случае сводится к нахождению кратчайшего остова [2]. В тех ситуациях, когда такой список отсутствует, предпочтение, в силу характера решаемой задачи, отдается остовному дереву, содержащему вершину с наибольшей степенью, а среди деревьев с одинаковой наибольшей степенью их вершин выбирается дерево с наибольшей степенью оставшихся вершин и т. д. При склеивании леса каждый раз выбирается фиктивное ребро $a_{i_k j_n}$ такое, что степени вершин x_{i_k} и x_{j_n} являются максимальными из всех степеней вершин соответствующих деревьев (предполагается, что ребро $a_{i_k j_n}$ не соответствует «нежелательному» словарю). Все это позволяет выделить несколько основных (базовых) языков, что значительно облегчает подготовку исходной информации для ИБ системы.

При реализации алгоритма некоторые шаги в ряде случаев были опущены или объединены.

Итак, в результате выполнения описанного алгоритма построено дерево T^*, ребрам которого соответствует множество словарей, определяющее ядро L^*. На следующем этапе в соответствии с T' формируется само ядро L^*. При этом просматривается список T' и, если он не пуст, то по каждому фиктивному ребру $a_{i_k j_n}$ из T' специалистом-переводчиком устанавливается соответствие типа (1) между элементами языков L_{i_k} и L_{j_n}, т. е. строится словарь (L_{i_k}, L_{j_n}) либо (L_{j_n}, L_{i_k}). Далее процедура формирования ИБ системы СПЕРКС-1 осуществляется по следующей схеме.

1. Специальной программой, реализующей алгоритм, основанный на полном обходе дерева T^* и использовании процедуры P, формируются так называемые смысловые гнезда информационной базы, каждое отдельное смысловое гнездо содержит условно-эквивалентные по смыслу КС всех языков из банка языков L (говоря «условно-эквивалентные», имеем в виду, что процедура P использует свойства симметричности и транзитивности соответствия типа (1), которые могут, хоть и крайне редко, быть нарушены).

2. Все смысловые гнезда выводятся на печать, подвергаются анализу с целью внесения в необходимых случаях требующихся изменений в смысловые гнезда, устанавливающих абсолютную смысловую эквивалентность всех КС для каждого гнезда (изменения вносятся программно).

3. Всем КС каждого смыслового гнезда присваивается его номер, называемый концептуальным номером (КН).

4. Программно формируется в виде совокупности базовых массивов информационная база системы СПЕРКС-1; отдельный базовый массив $B_i (i=\overline{1, m})$ из ИБ соответствует языку L_i и содержит все КС языка L_i (по одному из каждого смыслового гнезда) с соответствующими им КН.

Обработка массива входных документов в системе СПЕРКС-1 осуществляется по следующему алгоритму.

Шаг 1. Чтение массива входных документов, выбор из поля j^* каждого документа всех КС и запись их в отдельный массив КС с присвоением каждому КС номера документа, которому оно соответствует, и порядкового номера данного КС в поле j^*.

Шаг 2. Выбор базового массива B_i, соответствующего языку L_i, указанному в поле i^*.

Шаг 3. Поиск КС из массива КС в массиве B_i с целью получения для каждого КС соответствующего ему КН.

Шаг 4. Выбор по полученным на шаге 3 КН в базовом массиве B_j, соответствующем языку L_j, указанному пользователем, выходных эквивалентов и запись их в выходной массив КС.

Шаг 5. Упорядочение выходного массива КС по номеру документа и порядковому номеру КС в поле j^*.

Шаг 6. Формирование поля j^* каждого входного документа на языке L_j.

Отметим, что в системе предусмотрена обработка «неопознанных» на шаге 3 КС. Программное обеспечение системы СПЕРКС-1 построено по модульному принципу. Оно включает модули, реализующие описанный алгоритм, а также многие сервисные функции по всей работе с ИБ системы, о которых говорилось ранее. Система работает под управлением ОС/ЕС, программирование выполнено на алгоритмическом языке ПЛ/1.

В заключение укажем, что поскольку нами достигнуты определенные результаты по автоматическому анализу и синтезу ЕЯ (см., например,

[3]), в дальнейшем планируется разработка второго варианта системы СПЕРКС-1 для обработки таких входных документов, в поле j^* которых задана неформализованная естественно-языковая информация.

ЛИТЕРАТУРА

1. Король И. А., Совпель И. В.— Вестн. Белорусского ун-та. Сер. 1, физ., мат. и мех., 1984, № 1, с. 30.
2. Кристофидес Н. Теория графов.— М., 1978.
3. Гончаренко В. В., Король И. А., Котельникова Н. М., Музалевская В. М., Совпель И. В.— Тез. докл. Всесоюз. конф.: Переработка текста методами инженерной лингвистики. Минск, 1982, с. 109.

Поступила в редакцию 24.05.82.

Кафедра МО АСУ

УДК 519.926

А. А. ЛЕВАКОВ

РЕГУЛИРУЕМОСТЬ НЕЛИНЕЙНЫХ СИСТЕМ

В работе [1] введено понятие управляемости системы динамическим регулятором и показано, что линейная стационарная система управляема стационарным линейным регулятором тогда и только тогда, когда система управляема, а регулятор наблюдаем [2]. Исследование управляемости различных линейных систем с помощью регулятора проведено в [1, 3]. В предлагаемой заметке эти результаты распространены на нелинейные системы.

Рассмотрим объект, описываемый уравнениями:

$\dot{x}=f(t, x, y, u)$, $x(t_0)=x_0$ (1), $u=g(t, x, y)$, (2), $\dot{y}=\omega(t, x, y)$, $y(t_0)=y_0$, (3)

где $x\in R^n$, $y\in R^k$, $t\in R_+=[t_0, +\infty[$, $u\in R^m$, $f:R_+\times R^n\times R^k\times R^m\to R^n$, $g:R_+\times R^n\times R^k\to R^m$, $\omega:R_+\times R^n\times R^k\to R^k$ — j-кратно непрерывно дифференцируемые функции. Обозначим через

$$\begin{cases} x=x(t, t_0, x_0, y_0) \\ y=y(t, t_0, x_0, y_0)' \end{cases}$$

решение системы (1)—(3) и через $\Gamma(\beta, t, x_0)$, $\Gamma(t, x_0)$ — множества

$$\Gamma(\beta, t, x_0)=\{x\in R^n \mid x=x(t, t_0, x_0, y_0'),\ \|y_0'-y_0\|\leqslant\beta\},$$

$$\Gamma(t, x_0)=\{x\in R^n \mid x=x(t, t_0, x_0, y_0),\ y_0\in R^k\}.$$

Определения: 1. Систему (1)—(3) назовем локально регулируемой вдоль $x(t, t_0, x_0, y_0)$, если для всех $\beta>0$ и для всех t из некоторого промежутка $]t_0, t_1]$ точка $x(t, t_0, x_0, y_0)$ является внутренней для $\Gamma(\beta, t, x_0)$.

2. Систему (1)—(3) будем называть регулируемой, если $\Gamma(t, x_0)=R^n$ при всех t из некоторого промежутка $]t_0, t_1]$ и при всех $x_0\in R^n$.

Введем оператор U, действующий на функции $(t_0, x_0, y_0)\to\xi(t_0, x_0, y_0)\in R^n$ по правилу $U(\xi)=\frac{\partial\xi}{\partial t_0}+\frac{\partial\xi}{\partial y_0}\omega(t_0, x_0, y_0)+\frac{\partial\xi}{\partial x_0}f(t_0, x_0, y_0, g(t_0, x_0, y_0))$. Степени оператора U определим равенствами $U^0(\xi)=\xi$, $U^n(\xi)=U(U^{n-1}(\xi))$, $n\geqslant 1$. Обозначим через R_j, W_j следующие матрицы:

$$R_j=\left(\frac{\partial U^1(x_0)}{\partial y_0}, \ldots, \frac{\partial U^j(x_0)}{\partial y_0}\right),\quad W_j=\left(\left(\frac{\partial U^1(x_0)}{\partial y_0}\right)^{\mathrm{т}}, \ldots, \left(\frac{\partial U^j(x_0)}{\partial y_0}\right)^{\mathrm{т}}\right)$$

(т — знак транспонирования).

Отметим, что матрицы R_j, W_j строятся непосредственно по параметрам системы (1)—(3).

Теорема 1. Если существует натуральное число j такое, что

$$\operatorname{rank} R_j=n,\ \operatorname{rank} W_j\geqslant n, \qquad (4)$$

то система (1)—(3) является локально регулируемой вдоль $x(t, t_0, x_0, y_0)$.

Доказательство легко вытекает из теоремы о неявной функции, так как ранг матрицы Якоби отображения $z(t, y_0')=x(t, t_0, x_0, y_0')-x(t, t_0, x_0, y_0)$ при $y_0'=y_0$ и всех t из некоторого промежутка $]t_0, t_1]$ равен n.