

Доказательство. *Необходимость.* Пусть $E(x, y)$ – линеен. Тогда из первого свойства (x_0, ε) – продолжения получим, что оно не зависит от выбора $x_0 \in \Omega$ и $\varepsilon \neq 0$ и является линейным, поскольку имеет вид (7).

Достаточность. Обратное утверждение является следствием замечания 2. Действительно, если (x_0, ε) – продолжение не зависит (x_0, ε) и удовлетворяет условиям теоремы, то оно представляет собой линейный предикат на всем пространстве $\langle L, R \rangle$. Тогда предикат $E(x, y)$ просто обязан быть линейным, так как в противном случае его (x_0, ε) – продолжение зависело бы от выбора x_0, ε и, главное, не являлось бы линейным предикатом. Теорема 2 доказана.

Полученный результат обобщает ситуации, рассмотренные ранее. Его относительная простота и в формулировке, и в доказательстве является существенным преимуществом перед достаточно громоздкими в формулировке и трудоемкими в доказательстве теоремами, доказанными в работах [2–5]. В этом плодотворность идеи продолжения. Однако в отличие от ранее доказанных теорем нет непосредственно сформулированных свойств исходного предиката. На практике это означает необходимость предварительного пересчета входных сигналов, что не всегда удобно. Поэтому с точки зрения экспериментальной проверки преимущество имеют ранее полученные результаты [3].

Список литературы: 1. Ахиезер Н.И., Глазман И.М. Теория линейных операторов в гильбертовом пространстве. М.: Наука, 1966. 543 с. 2. Шабанов-Кушнаренко Ю.П. Теория интеллекта: Проблемы и перспективы. Т.3. Харьков, Выща шк., 1987. 158 с. 3. Воскобойник О.Н., Иващенко В.В. Компараторная идентификация абстрактных линейных операторов // АСУ и приборы автоматики, 2000. Вып.113. С.35–41. 4. Герасин С.Н. Математические модели процессов сенсорной факторизации и применение для идентификации линейных систем методом сравнения // Дисс. канд. техн. наук. Харьков, 1989. 134 с. 5. Шляхов В.В. Математические модели линейных процессов рецепции и их технические приложения // Дисс. канд. техн. наук. Харьков, 1984. 138 с.

Поступила в редколлегию

Иващенко Валерий Владимирович, соискатель ХТУРЭ. Научные интересы: математические методы анализа сложных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40–93–72.

УДК 519.673

В. Б. КЛИМУШЕВ

МОДЕЛИРОВАНИЕ КОРРЕЛЯЦИОННОГО АЛГОРИТМА РАСПОЗНАВАНИЯ ОБРАЗОВ НА БАЗЕ ФОРМАЛЬНОЙ ГРАММАТИКИ

Рассматриваются вопросы моделирования конкретного корреляционного алгоритма распознавания образов на базе формальных грамматик с разными системами продукции, но одинаковыми алфавитами и начальной аксиомой. Проведенный сравнительный анализ выводов представлений корреляционного алгоритма позволяет предложить алгоритм синтеза множественных продукции, который дает возможность сконструировать формальную грамматику, оптимально порождающую вывод представления корреляционного алгоритма.

Статья использует и развивает результаты, изложенные в [1].

Среди алгоритмов распознавания образов важное место занимает корреляционный алгоритм, что обусловлено широким его использованием в контурах управления телевизионных следящих систем.

Рассмотрим конкретный корреляционный алгоритм распознавания образов.

Пусть в памяти компьютера значения яркостей поля экрана записаны в виде одномерного массива элементов яркости экрана:

$$A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}.$$

Считаем, что на некоторой части площади экрана сформирован эталон, который сохраняется в памяти компьютера в виде массива элементов яркости эталона Э:

$$E = \{э_1, э_2, э_3, э_4\}.$$

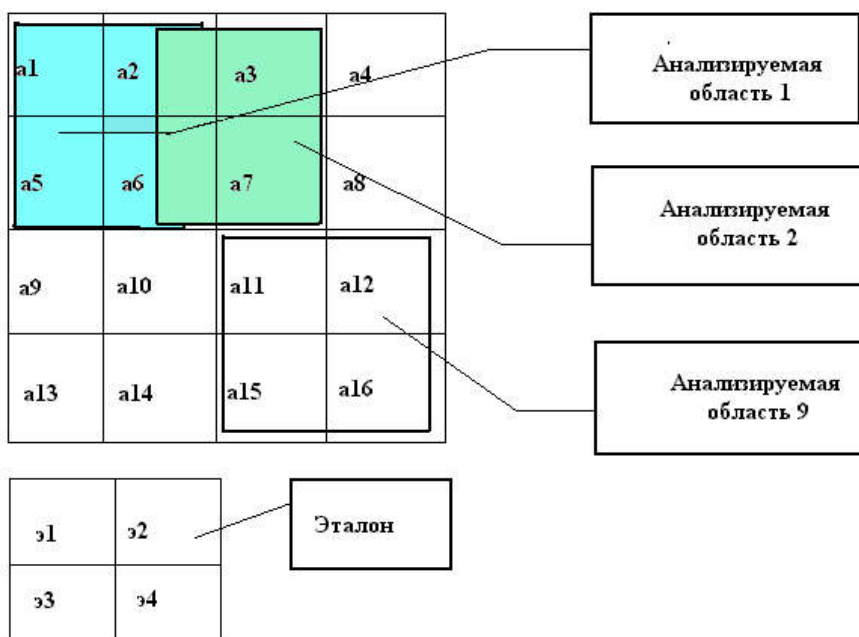
Графическая интерпретация введенных обозначений показана на рисунке.

Разобьем поле экрана на анализируемые области $A_r \in A$, $r = \{1, 2, \dots, 9\}$, по площади равные площади эталона. Проверим все анализируемые области по степени соответствия эталону, для чего вычислим корреляционную функцию

$$\rho_r = \sum_{j=1}^4 |a_{rj} - э_j|,$$

где a_{rj} – элемент яркости экрана, входящий в анализируемую область, а $э_j$ – элемент яркости эталона, подлежащий сравнению с элементом a_{rj} .

Опишем рассмотренный корреляционный алгоритм в терминах формальной грамматики, причем под буквой алфавита, принадлежащего грамматике, будем понимать последовательность символов, не разделенных пробелом или знаком подчеркивания.



Графическая интерпретация корреляционного алгоритма

Реализации корреляционного алгоритма можно поставить в соответствие полный вывод цепочки в языке, порожденном грамматикой:

$$G_1 = (U_{T1}, V_{HT1}, \sigma_1, P_1),$$

где $U_{T1} = \{k\}$ – алфавит терминальных букв:

$V_{HT1} = \{\text{sigm}, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}, э_1, э_2, э_3, э_4, 1m_1, 1m_2, 1m_3, 1m_4, 2m_1, 2m_2, 2m_3, 2m_4, 3m_1, 3m_2, 3m_3, 3m_4, 4m_1, 4m_2, 4m_3, 4m_4, 5m_1, 5m_2, 5m_3, 5m_4, 6m_1, 6m_2, 6m_3, 6m_4, 7m_1, 7m_2, 7m_3, 7m_4, 8m_1, 8m_2, 8m_3, 8m_4, 9m_1, 9m_2, 9m_3, 9m_4, 1s_1, 1s_2, 2s_1, 2s_2, 3s_1, 3s_2, 4s_1, 4s_2, 5s_1, 5s_2, 6s_1, 6s_2, 7s_1, 7s_2, 8s_1, 8s_2, 9s_1, 9s_2, 1s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s, 1c, 2c, 3c, 4c, 5c, 6c, 7c, c\}$ – алфавит нетерминальных букв;

$\sigma_1 = \{\text{sigm}\}$ – начальная аксиома;

$P_1 = \{p_{1j} \mid j = 1, 2, \dots, 92\}$ – система продукций, причем

$p_{11} : \text{sigm} \rightarrow a1_sigm;$ $p_{12} : \text{sigm} \rightarrow a2_sigm;$ $p_{13} : \text{sigm} \rightarrow a3_sigm;$
 $p_{14} : \text{sigm} \rightarrow a4_sigm;$ $p_{15} : \text{sigm} \rightarrow a5_sigm;$ $p_{16} : \text{sigm} \rightarrow a6_sigm;$
 $p_{17} : \text{sigm} \rightarrow a7_sigm;$ $p_{18} : \text{sigm} \rightarrow a8_sigm;$ $p_{19} : \text{sigm} \rightarrow a9_sigm;$
 $p_{110} : \text{sigm} \rightarrow a10_sigm;$ $p_{111} : \text{sigm} \rightarrow a11_sigm;$ $p_{112} : \text{sigm} \rightarrow a12_sigm;$
 $p_{113} : \text{sigm} \rightarrow a13_sigm;$ $p_{114} : \text{sigm} \rightarrow a14_sigm;$ $p_{115} : \text{sigm} \rightarrow a15_sigm;$
 $p_{116} : \text{sigm} \rightarrow a16_sigm;$ $p_{117} : \text{sigm} \rightarrow \varepsilon1_sigm;$ $p_{118} : \text{sigm} \rightarrow \varepsilon2_sigm;$
 $p_{119} : \text{sigm} \rightarrow \varepsilon3_sigm;$ $p_{120} : \text{sigm} \rightarrow \varepsilon4;$ $p_{121} : a1_ \varepsilon1 \rightarrow 1M1;$
 $p_{122} : a2_ \varepsilon2 \rightarrow 1M2;$ $p_{123} : a5_ \varepsilon3 \rightarrow 1M3;$ $p_{124} : a6_ \varepsilon4 \rightarrow 1M4;$
 $p_{125} : 1M1_ 1M2 \rightarrow 1s1;$ $p_{126} : 1M3_ 1M4 \rightarrow 1s2;$ $p_{127} : 1s1_ 1s2 \rightarrow 1s_sigm;$
 $p_{128} : a2_ \varepsilon1 \rightarrow 2M1;$ $p_{129} : a3_ \varepsilon2 \rightarrow 2M2;$ $p_{130} : a6_ \varepsilon3 \rightarrow 2M3;$
 $p_{131} : a7_ \varepsilon4 \rightarrow 2M4;$ $p_{132} : 2M1_ 2M2 \rightarrow 2s1;$ $p_{133} : 2M3_ 2M4 \rightarrow 2s2;$
 $p_{134} : 2s1_ 2s2 \rightarrow 2s_sigm;$ $p_{135} : a3_ \varepsilon1 \rightarrow 3M1;$ $p_{136} : a4_ \varepsilon2 \rightarrow 3M2;$
 $p_{137} : a7_ \varepsilon3 \rightarrow 3M3;$ $p_{138} : a8_ \varepsilon4 \rightarrow 3M4;$ $p_{139} : 3M1_ 3M2 \rightarrow 3s1;$
 $p_{140} : 3M3_ 3M4 \rightarrow 3s2;$ $p_{141} : 3s1_ 3s2 \rightarrow 3s_sigm;$ $p_{142} : a5_ \varepsilon1 \rightarrow 4M1;$
 $p_{143} : a6_ \varepsilon2 \rightarrow 4M2;$ $p_{144} : a9_ \varepsilon3 \rightarrow 4M3;$ $p_{145} : a10_ \varepsilon4 \rightarrow 4M4;$
 $p_{146} : 4M1_ 4M2 \rightarrow 4s1;$ $p_{147} : 4M3_ 4M4 \rightarrow 4s2;$ $p_{148} : 4s1_ 4s2 \rightarrow 4s_sigm;$
 $p_{149} : a6_ \varepsilon1 \rightarrow 5M1;$ $p_{150} : a7_ \varepsilon2 \rightarrow 5M2;$ $p_{151} : a10_ \varepsilon3 \rightarrow 5M3;$
 $p_{152} : a11_ \varepsilon4 \rightarrow 5M4;$ $p_{153} : 5M1_ 5M2 \rightarrow 5s1;$ $p_{154} : 5M3_ 5M4 \rightarrow 5s2;$
 $p_{155} : 5s1_ 5s2 \rightarrow 5s_sigm;$ $p_{156} : a7_ \varepsilon1 \rightarrow 6M1;$ $p_{157} : a8_ \varepsilon2 \rightarrow 6M2;$
 $p_{158} : a11_ \varepsilon3 \rightarrow 6M3;$ $p_{159} : a12_ \varepsilon4 \rightarrow 6M4;$ $p_{160} : 6M1_ 6M2 \rightarrow 6s1;$
 $p_{161} : 6M3_ 6M4 \rightarrow 6s2;$ $p_{162} : 6s1_ 6s2 \rightarrow 6s_sigm;$ $p_{163} : a9_ \varepsilon1 \rightarrow 7M1;$
 $p_{164} : a10_ \varepsilon2 \rightarrow 7M2;$ $p_{165} : a11_ \varepsilon3 \rightarrow 7M3;$ $p_{166} : a12_ \varepsilon4 \rightarrow 7M4;$
 $p_{167} : 7M1_ 7M2 \rightarrow 7s1;$ $p_{168} : 7M3_ 7M4 \rightarrow 7s2;$ $p_{169} : 7s1_ 7s2 \rightarrow 7s_sigm;$
 $p_{170} : a10_ \varepsilon1 \rightarrow 8M1;$ $p_{171} : a11_ \varepsilon2 \rightarrow 8M2;$ $p_{172} : a14_ \varepsilon3 \rightarrow 8M3;$
 $p_{173} : a15_ \varepsilon4 \rightarrow 8M4;$ $p_{174} : 8M1_ 8M2 \rightarrow 8s1;$ $p_{175} : 8M3_ 8M4 \rightarrow 8s2;$
 $p_{176} : 8s1_ 8s2 \rightarrow 8s_sigm;$ $p_{177} : a11_ \varepsilon1 \rightarrow 9M1;$ $p_{178} : a12_ \varepsilon2 \rightarrow 9M2;$
 $p_{179} : a15_ \varepsilon3 \rightarrow 9M3;$ $p_{180} : a16_ \varepsilon4 \rightarrow 9M4;$ $p_{181} : 9M1_ 9M2 \rightarrow 9s1;$
 $p_{182} : 9M3_ 9M4 \rightarrow 9s2;$ $p_{183} : 9s1_ 9s2 \rightarrow 9s;$ $p_{184} : 1s_ 2s \rightarrow 1c;$
 $p_{185} : 1c_ 3s \rightarrow 2c;$ $p_{186} : 2c_ 4s \rightarrow 3c;$ $p_{187} : 3c_ 5s \rightarrow 4c;$
 $p_{188} : 4c_ 6s \rightarrow 5c;$ $p_{189} : 5c_ 7s \rightarrow 6c;$ $p_{190} : 6c_ 8s \rightarrow 7c;$
 $p_{191} : 7c_ 9s \rightarrow c;$ $p_{192} : c \rightarrow k.$

Опишем математический смысл введенных обозначений: sigm – начальная аксиома. Буквы, кодирующие исходные данные, таковы: $a1, a2, \dots, a16$ – буквы нетерминального алфавита, соответствующие элементам яркости экрана; $\varepsilon1, \dots, \varepsilon4$ – буквы нетерминального алфавита, соответствующие элементам яркости эталона.

Оставшиеся буквы нетерминального алфавита являются производными от букв, кодирующих исходные данные; они необходимы для синтеза системы подстановок и имеют следующий смысл:

$1M1 = |a1-\varepsilon1|;$ $1M2 = |a2-\varepsilon2|;$
 $1M3 = |a5-\varepsilon3|;$ $1M4 = |a6-\varepsilon4|;$
 $2M1 = |a2-\varepsilon1|;$ $2M2 = |a3-\varepsilon2|;$
 $2M3 = |a6-\varepsilon3|;$ $2M4 = |a7-\varepsilon4|;$
 $3M1 = |a3-\varepsilon1|;$ $3M2 = |a4-\varepsilon2|;$
 $3M3 = |a7-\varepsilon3|;$ $3M4 = |a8-\varepsilon4|;$
 $4M1 = |a5-\varepsilon1|;$ $4M2 = |a6-\varepsilon2|;$
 $4M3 = |a9-\varepsilon3|;$ $4M4 = |a10-\varepsilon4|;$
 $5M1 = |a6-\varepsilon1|;$ $5M2 = |a7-\varepsilon2|;$
 $5M3 = |a10-\varepsilon3|;$ $5M4 = |a11-\varepsilon4|;$
 $6M1 = |a7-\varepsilon1|;$ $6M2 = |a8-\varepsilon2|;$
 $6M3 = |a11-\varepsilon3|;$ $6M4 = |a12-\varepsilon4|;$
 $7M1 = |a9-\varepsilon1|;$ $7M2 = |a10-\varepsilon2|;$
 $7M3 = |a13-\varepsilon3|;$ $7M4 = |a14-\varepsilon4|;$
 $8M1 = |a10-\varepsilon1|;$ $8M2 = |a11-\varepsilon2|;$

$$\begin{array}{ll}
8m3 = |a14-э3|; & 8m4 = |a15-э4|; \\
9m1 = |a11-э1|; & 9m2 = |a12-э2|; \\
9m3 = |a15-э3|; & 9m4 = |a16-э4|; \\
1m1+1m2 = 1s1; & 1m3+1m4 = 1s2; \\
2m1+2m2 = 2s1; & 2m3+2m4 = 2s2; \\
3m1+3m2 = 3s1; & 3m3+3m4 = 3s2; \\
4m1+4m2 = 4s1; & 4m3+4m4 = 4s2; \\
5m1+5m2 = 5s1; & 5m3+5m4 = 5s2; \\
6m1+6m2 = 6s1; & 6m3+6m4 = 6s2; \\
7m1+7m2 = 7s1; & 7m3+7m4 = 7s2; \\
8m1+8m2 = 8s1; & 8m3+8m4 = 8s2; \\
9m1+9m2 = 9s1; & 9m3+9m4 = 9s2; \\
1s1+1s2 = 1s; & 2s1+2s2 = 2s; \\
3s1+3s2 = 3s; & 4s1+4s2 = 4s; \\
5s1+5s2 = 5s; & 6s1+6s2 = 6s; \\
7s1+7s2 = 7s; & 8s1+8s2 = 8s; \\
9s1+9s2 = 9s; & 1s \quad 2s = 1c; \\
1c \quad 3s = 2c; & 2c \quad 4s = 3c; \\
3c \quad 5s = 4c; & 4c \quad 6s = 5c; \\
5c \quad 7s = 6c; & 6c \quad 8s = 7c; \\
7c \quad 9s = c.
\end{array}$$

И далее в рассмотренных ранее выражениях:

символ "m" входит в буквы нетерминального алфавита, кодирующие модули разности соответствующих элементов экрана;

символ "s" входит в буквы нетерминального алфавита, кодирующие промежуточные, и окончательные суммы модулей анализируемых областей;

символ "c" входит в буквы нетерминального алфавита, кодирующие промежуточные и окончательный результат сравнения по абсолютной величине сумм модулей всех анализируемых областей;

k – буква терминального алфавита, кодирующая расположения объекта. По значению координаты объекта компьютер формирует управляющее воздействие, подаваемое на вход сервомеханизма, который управляет положением телекамеры; – обозначение бинарной операции сравнения. Опишем корреляционный алгоритм, поставив ему в соответствие полный вывод терминальной цепочки $k \in L_1(G_1)$:

sigm, a1_sigm, a1_э1_sigm, 1m1_sigm, 1m1_a2_sigm, 1m1_a2_э2_sigm, 1m1_1m2_sigm, 1m1_1m2_a5_sigm, 1m1_1m2_a5_э3_sigm, 1m1_1m2_1m3_sigm, 1m1_1m2_1m3_a6_sigm, 1m1_1m2_1m3_a6_э4_sigm, 1m1_1m2_1m3_1m4_sigm, 1s1_1m3_1m4_sigm, 1s1_1s2_sigm, 1s_sigm,... и далее аналогично для анализируемых областей с номерами 2, 3, ..., 9. Таким образом, впоследствии имеем

..., 1s_2s_3s_4s_5s_6s_7s_8s_9s, 1c_3s_4s_5s_6s_7s_8s_9s, 2c_4s_5s_6s_7s_8s_9s, 3c_5s_6s_7s_8s_9s, 4c_6s_7s_8s_9s, 5c_7s_8s_9s, 6c_8s_9s, 7c_9s, c, k.

Проанализируем представление корреляционного алгоритма. В отличие от известных методов описания алгоритмов, подобное представление позволяет описывать процесс выделения и подготовки данных для реализации алгоритма, что выгодно отличает предложенный метод описания алгоритма от других известных методов.

Действительно, правила подстановки $p_{11} \div p_{120}$ позволяют моделировать процесс выделения и подготовки данных для работы соответствующих продукций путем вывода необходимых цепочек. Так, для работы продукции p_{121} необходимо вывести в G_1 цепочку a1_э1, что достигается путем применения продукций p_{11} и p_{117} .

Изменим правила подстановки, оставив без изменения алфавиты грамматики G_1 . Таким образом, мы получим новую порождающую грамматику G_2 .

Пусть имеем $G_2 = (U_{T_2}, V_{HT_2}, \sigma_2, P_2)$, где

$U_{T_2} = U_{T_1}$;

$V_{HT_2} = V_{HT_1}$;

$\sigma_2 = \sigma_1 = \{ \text{sigm} \}$,

$P_2 : \{ p_{2j} \mid j=1,2,\dots,29 \}$, где

$p_{21} : \text{sigm} \rightarrow a1_э1_a2_э2_a5_э3_a6_э4_ \text{sigm}$;

$p_{22} : \text{sigm} \rightarrow a2_э1_a3_э2_a6_э3_a7_э4_ \text{sigm}$;

$p_{23} : \text{sigm} \rightarrow a3_э1_a4_э2_a7_э3_a8_э4_ \text{sigm}$;

$p_{24} : \text{sigm} \rightarrow a5_э1_a6_э2_a9_э3_a10_э4_ \text{sigm}$;

$p_{25} : \text{sigm} \rightarrow a6_э1_a7_э2_a10_э3_a11_э4_ \text{sigm}$;

$p_{26} : \text{sigm} \rightarrow a7_э1_a8_э2_a11_э3_a12_э4_ \text{sigm}$;

$p_{27} : \text{sigm} \rightarrow a9_э1_a10_э2_a13_э3_a14_э4_ \text{sigm}$;

$p_{28} : \text{sigm} \rightarrow a10_э1_a11_э2_a14_э3_a15_э4_ \text{sigm}$;

$p_{29} : \text{sigm} \rightarrow a11_э1_a12_э2_a15_э3_a16_э4$;

$p_{210} : a1_э1_a2_э2_a5_э3_a6_э4 \rightarrow 1m1_1m2_1m3_1m4$;

$p_{211} : a2_э1_a3_э2_a6_э3_a7_э4 \rightarrow 2m1_2m2_2m3_2m4$;

$p_{212} : a3_э1_a4_э2_a7_э3_a8_э4 \rightarrow 3m1_3m2_3m3_3m4$;

$p_{213} : a5_э1_a6_э2_a9_э3_a10_э4 \rightarrow 4m1_4m2_4m3_4m4$;

$p_{214} : a6_э1_a7_э2_a10_э3_a11_э4 \rightarrow 5m1_5m2_5m3_5m4$;

$p_{215} : a7_э1_a8_э2_a11_э3_a12_э4 \rightarrow 6m1_6m2_6m3_6m4$;

$p_{216} : a9_э1_a10_э2_a13_э3_a14_э4 \rightarrow 7m1_7m2_7m3_7m4$;

$p_{217} : a10_э1_a11_э2_a14_э3_a15_э4 \rightarrow 8m1_8m2_8m3_8m4$;

$p_{218} : a11_э1_a12_э2_a15_э3_a16_э4 \rightarrow 9m1_9m2_9m3_9m4$;

$p_{219} : 1m1_1m2_1m3_1m4 \rightarrow 1s$; $p_{220} : 2m1_2m2_2m3_2m4 \rightarrow 2s$;

$p_{221} : 3m1_3m2_3m3_3m4 \rightarrow 3s$; $p_{222} : 4m1_4m2_4m3_4m4 \rightarrow 4s$;

$p_{223} : 5m1_5m2_5m3_5m4 \rightarrow 5s$; $p_{224} : 6m1_6m2_6m3_6m4 \rightarrow 6s$;

$p_{225} : 7m1_7m2_7m3_7m4 \rightarrow 7s$; $p_{226} : 8m1_8m2_8m3_8m4 \rightarrow 8s$;

$p_{227} : 9m1_9m2_9m3_9m4 \rightarrow 9s$; $p_{228} : 1s_2s_3s_4s_5s_6s_7s_8s_9s \rightarrow c$;

$p_{229} : c \rightarrow k$.

Представим описание корреляционного алгоритма в грамматике G_2 , поставив ему в соответствие полный вывод терминальной цепочки k в G_2 :

$\text{sigm}, a1_э1_a2_э2_a5_э3_a6_э4_ \text{sigm}$. Последовательно применяя продукции $p_{229} \div p_{29}$, получаем цепочку

$a1_э1_a2_э2_a5_э3_a6_э4_a2_э1_a3_э2_a6_э3_a7_э4_a3_э1_a4_э2_a7_э3_a8_э4_a5_э1_a6_э2_a9_э3_a10_э4_a6_э1_a7_э2_a10_э3_a11_э4_a7_э1_a8_э2_a11_э3_a12_э4_a9_э1_a10_э2_a13_э3_a14_э4_a10_э1_a11_э2_a14_э3_a15_э4_a11_э1_a12_э2_a15_э3_a16_э4$.

Заметим, что таким образом удалось вывести цепочку, содержащую все исходные данные для работы корреляционного алгоритма. Применяя последовательно продукции $p_{210} \div p_{226}$, получаем цепочку и продолжаем вывод:

$1s_2s_3s_4s_5s_6s_7s_8s_9m1_9m2_9m3_9m4, 1s_2s_3s_4s_5s_6s_7s_8s_9s, c, k$.

Проведем сравнительный анализ представлений корреляционного алгоритма $\Pi_{kor}(L_1)$ и $\Pi_{kor}(L_2)$ в языках L_1 и L_2 , порожденных грамматиками G_1 и G_2 , соответственно.

Так как $\Pi_{kor}(L_1) = \Pi_{kor}(L_2) = k$, то основную информацию о корреляционном алгоритме несут выводы представлений $W_1 = W(\Pi_{kor}(L_1))$ и $W_2 = W(\Pi_{kor}(L_2))$. Вследствие того, что $\Pi_{kor}(L_1)$ выводится за 144 шага, а $\Pi_{kor}(L_2)$ за 29 шагов, то длина вывода $l(W_1) = 144$, а длина вывода $l(W_2) = 29$, таким образом, $l(W_2) \ll l(W_1)$. $\Pi_{kor}(L_2)$ выводится быстрее за счет более удачной системы продукций $P_2 \in G_2$. Действительно, в состав P_2 входят многоместные продукции, с числом букв в правой или левой частях продукции, доходящих до 9. Например, в продукциях $p_{21} \div p_{28}$ число букв, входящих в правые части – 9, а в продукции p_{228} число букв, входящих в левую часть – 9. Все остальные продукции, кроме p_{229} , также

являются многоместными. Рассмотрим систему продукций $P_1 \in G_1$, в состав P_1 входят продукции с числом букв в правой и левой частях не более 2, а продукции p_{120} и p_{192} являются одностепенными. Таким образом, в отличие от P_2 , P_1 содержит только двуместные и одностепенные продукции; следовательно, грамматика G_1 представляет собой алгоритмическую систему, позволяющую описывать только последовательные алгоритмы.

Предложим алгоритм синтеза многоместных продукций (СМП) на основе имеющейся системы двуместных продукций. Алгоритм СМП пошагово будет выглядеть следующим образом:

Шаг 1. Проанализируем продукции побуквенно и присвоим буквам уровни. Для начальной аксиомы $sigm$ – 0-й уровень, а для всех букв, находящихся в правых частях продукций, которые содержат в левой части букву $sigm$ – 1-й уровень.

Шаг 2. Для всех букв, входящих в правые части продукций, содержащих в левой части хотя бы одну букву 1-го уровня, присвоим 2-й уровень.

Шаг 3. Продолжим процесс присваивания букве уровня, пока терминальная буква не получит свой уровень. Таким образом, если у буквы $k \in U_{T1}$ будет n -й уровень, то у буквы $s \in V_{HT1}$ будет $(n-1)$ -й уровень.

Шаг 4. Проанализируем продукцию, имеющую наибольший i -й номер (конечную продукцию). Если в левой части этой продукции имеются буквы с разными уровнями, то путем подстановки левых частей соответствующих продукций синтезируем новую продукцию, содержащую в своем составе буквы более низкого уровня. Если же в левой части i -й продукции нет букв с разным уровнем, то начинаем работать с продукцией, имеющей $(i-1)$ -й номер.

Шаг 5. На основе многоместной продукции, полученной первой, синтезируем новую продукцию, позволяющую вывести левую часть уже синтезированной многоместной продукции (в дальнейшем именуемой исходной продукцией). Для этого заменим правую букву левой части исходной продукции, используя соответствующую подстановку, и получим новую продукцию, в левой части которой будут буквы разных уровней. К этой продукции применим действия, описанные в "шаге 4".

Шаг 6. Продолжаем процесс синтеза новых продукций, пока не синтезируем продукцию, содержащую в своей левой части только одну букву $sigm$.

Шаг 7. В новую систему продукций включаем все синтезированные многоместные продукции и все продукции исходной системы, не использованные алгоритмом СМП.

Проиллюстрируем основные моменты предложенного алгоритма СМП, используя в качестве исходных данных систему продукций $P_1 \in G_1$.

После присваивания уровней буквы получают следующие уровни: "k" – 7-й уровень, "c" – 6-й уровень, "7c, 6c, 5c, 4c, 3c, 2c, 1c" – 5-й уровень, "9s, 8s, 7s, 6s, 5s, 4s, 3s, 2s, 1s" – 4-й уровень.

Первой продукцией, к которой применим шаг 4 алгоритма СМП, является продукция p_{191} . Последовательно понижая уровень букв левой части продукции p_{191} , получаем следующую последовательность продукций:

$$\begin{aligned}
 p_{191}: & 7c_9s \rightarrow c; \\
 & 6c_8s_9s \rightarrow c; \\
 & 5c_7s_8s_9s \rightarrow c; \\
 & 4c_6s_7s_8s_9s \rightarrow c; \\
 & 3c_5s_6s_7s_8s_9s \rightarrow c; \\
 & 2c_4s_5s_6s_7s_8s_9s \rightarrow c; \\
 & 1c_3s_4s_5s_6s_7s_8s_9s \rightarrow c; \\
 p_{34}: & 1s_2s_3s_4s_5s_6s_7s_8s_9s \rightarrow c
 \end{aligned}$$

Первой многоместной продукцией будет p_{34} , которая синтезирована в результате действия шага 4 алгоритма СМП. Применим к продукции p_{34} шаг 5 алгоритма

ма СМП (используя подстановку, обусловленную продукцией p_{183}), результатом действия которого будет следующая продукция:

$P_{исх}: 1s\ 2s\ 3s\ 4s\ 5s\ 6s\ 7s\ 8s\ 9s \rightarrow 1s\ 2s\ 3s\ 4s\ 5s\ 6s\ 7s\ 8s\ 9s;$

$P_{исх}$ будет исходной продукцией для дальнейшего действия алгоритма СМП.

Применяя далее алгоритм СМП, получаем систему продукций P_3 :

$P_3 = \{p_{3j} \mid j=1,2,\dots,5\}$, где

P_{31} : $sigm \rightarrow a1\ \bar{a}1\ a2\ \bar{a}2\ a5\ \bar{a}3\ a6\ \bar{a}4\ a2\ \bar{a}1\ a3\ \bar{a}2\ a6\ \bar{a}3\ a7\ \bar{a}4\ a3\ \bar{a}1\ a4\ \bar{a}2\ a7\ \bar{a}3\ a8\ \bar{a}4\ a5\ \bar{a}1\ a6\ \bar{a}2\ a9\ \bar{a}3\ a10\ \bar{a}4\ a6\ \bar{a}1\ a7\ \bar{a}2\ a10\ \bar{a}3\ a11\ \bar{a}4\ a7\ \bar{a}1\ a8\ \bar{a}2\ a11\ \bar{a}3\ a12\ \bar{a}4\ a9\ \bar{a}1\ a10\ \bar{a}2\ a13\ \bar{a}1\ a14\ \bar{a}4\ a10\ \bar{a}1\ a11\ \bar{a}2\ a14\ \bar{a}3\ a15\ \bar{a}4\ a11\ \bar{a}1\ a12\ \bar{a}2\ a15\ \bar{a}3\ a16\ \bar{a}4;$

P_{32} : $a1\ \bar{a}1\ a2\ \bar{a}2\ a5\ \bar{a}3\ a6\ \bar{a}4\ a2\ \bar{a}1\ a3\ \bar{a}2\ a6\ \bar{a}3\ a7\ \bar{a}4\ a3\ \bar{a}1\ a4\ \bar{a}2\ a7\ \bar{a}3\ a8\ \bar{a}4\ a5\ \bar{a}1\ a6\ \bar{a}2\ a9\ \bar{a}3\ a10\ \bar{a}4\ a6\ \bar{a}1\ a7\ \bar{a}2\ a10\ \bar{a}3\ a11\ \bar{a}4\ a7\ \bar{a}1\ a8\ \bar{a}2\ a11\ \bar{a}3\ a12\ \bar{a}4\ a9\ \bar{a}1\ a10\ \bar{a}2\ a13\ \bar{a}1\ a14\ \bar{a}4\ a10\ \bar{a}1\ a11\ \bar{a}2\ a14\ \bar{a}3\ a15\ \bar{a}4\ a11\ \bar{a}1\ a12\ \bar{a}2\ a15\ \bar{a}3\ a16\ \bar{a}4 \rightarrow 1m1\ 1m2\ 1m3\ 1m4\ 2m1\ 2m2\ 2m3\ 2m4\ 3m1\ 3m2\ 3m3\ 3m4\ 4m1\ 4m2\ 4m3\ 4m4\ 5m1\ 5m2\ 5m3\ 5m4\ 6m1\ 6m2\ 6m3\ 6m4\ 7m1\ 7m2\ 7m3\ 7m4\ 8m1\ 8m2\ 8m3\ 8m4\ 9m1\ 9m2\ 9m3\ 9m4;$

P_{33} : $1m1\ 1m2\ 1m3\ 1m4\ 2m1\ 2m2\ 2m3\ 2m4\ 3m1\ 3m2\ 3m3\ 3m4\ 4m1\ 4m2\ 4m3\ 4m4\ 5m1\ 5m2\ 5m3\ 5m4\ 6m1\ 6m2\ 6m3\ 6m4\ 7m1\ 7m2\ 7m3\ 7m4\ 8m1\ 8m2\ 8m3\ 8m4\ 9m1\ 9m2\ 9m3\ 9m4 \rightarrow 1s\ 2s\ 3s\ 4s\ 5s\ 6s\ 7s\ 8s\ 9s;$

P_{34} : $1s\ 2s\ 3s\ 4s\ 5s\ 6s\ 7s\ 8s\ 9s \rightarrow c;$

P_{35} : $c \rightarrow k.$

Используя P_3 , сформируем $G_3 = (U_{T3}, V_{HT3}, \sigma_3, P_3)$, причем $U_{T3} = U_{T1}$, $V_{HT3} = V_{HT1}$, $\sigma_3 = \sigma_1 = \{sigm\}$. Таким образом, имеем грамматику G_3 , отличающуюся от грамматик G_1 и G_2 только системой продукций.

Полный вывод цепочки k в грамматике G_3 будет иметь следующий вид:

$sigm, a1\ \bar{a}1\ a2\ \bar{a}2\ a5\ \bar{a}3\ a6\ \bar{a}4\ a2\ \bar{a}1\ a3\ \bar{a}2\ a6\ \bar{a}3\ a7\ \bar{a}4\ a3\ \bar{a}1\ a4\ \bar{a}2\ a7\ \bar{a}3\ a8\ \bar{a}4\ a5\ \bar{a}1\ a6\ \bar{a}2\ a9\ \bar{a}3\ a10\ \bar{a}4\ a6\ \bar{a}1\ a7\ \bar{a}2\ a10\ \bar{a}3\ a11\ \bar{a}4\ a7\ \bar{a}1\ a8\ \bar{a}2\ a11\ \bar{a}3\ a12\ \bar{a}4\ a9\ \bar{a}1\ a10\ \bar{a}2\ a13\ \bar{a}1\ a14\ \bar{a}4\ a10\ \bar{a}1\ a11\ \bar{a}2\ a14\ \bar{a}3\ a15\ \bar{a}4\ a11\ \bar{a}1\ a12\ \bar{a}2\ a15\ \bar{a}3\ a16\ \bar{a}4, 1m1\ 1m2\ 1m3\ 1m4\ 2m1\ 2m2\ 2m3\ 2m4\ 3m1\ 3m2\ 3m3\ 3m4\ 4m1\ 4m2\ 4m3\ 4m4\ 5m1\ 5m2\ 5m3\ 5m4\ 6m1\ 6m2\ 6m3\ 6m4\ 7m1\ 7m2\ 7m3\ 7m4\ 8m1\ 8m2\ 8m3\ 8m4\ 9m1\ 9m2\ 9m3\ 9m4, 1s\ 2s\ 3s\ 4s\ 5s\ 6s\ 7s\ 8s\ 9s, c, k.$

Вывод представления корреляционного алгоритма $W_3 = W(\Pi_{кор}(L_3))$ в языке $L_3 = L_3(G_3)$ реализуется за 5 шагов, следовательно, $l(W_3) = 5$ и $l(W_3) < l(W_2) < l(W_1)$, при этом выполняется $\Pi_{кор}(L_3) = \Pi_{кор}(L_2) = \Pi_{кор}(L_1) = k$.

$\Pi_{кор}(L_3)$ выводится наиболее быстро за счет самой удачной системы продукций $P_3 \in G_3$, полученной в результате действия алгоритма СМП на систему продукций P_1 .

Отличительной особенностью продукций, входящих в P_3 , является то, что они позволяют реализовать все рекуррентно независимые подстановки в одной продукции, а это в свою очередь соответствует параллельной алгоритмической реализации всех возможных рекуррентно независимых вычислений за один шаг. Таким образом, система продукций P_3 позволяет выводить $\Pi_{кор}(L_3)$ в темп с математической рекуррентностью представляемого алгоритма.

В результате проведенного моделирования можно сделать следующие выводы.

- Синтезированная грамматика $G_3 = (U_{T3}, V_{HT3}, \sigma_3, P_3)$ выводит $\Pi_{кор}(L_3)$ максимально быстро и, следовательно, согласно [1] является задачей для корреляционного алгоритма.

- Воздействуя алгоритмом СМП на последовательную систему продукций, можно синтезировать новую многоместную систему продукций, являющуюся основой для формирования новой грамматики, в которой представление алгоритма будет выводиться наиболее быстро в темп с математической рекуррентностью представляемого алгоритма.

- В результате действия алгоритма СМП на систему продукций P_1 получена система продукций P_3 , в состав которой входит продукция

$p_{34}: 1s\ 2s\ 3s\ 4s\ 5s\ 6s\ 7s\ 8s\ 9s \rightarrow c;$

Продукция p_{34} является многоместной и ей должна алгоритмически соответствовать многоместная математическая операция сравнения. Таким образом, показана возможность существования многоместной математической операции сравнения и возможность ее аппаратной реализации.

Алгоритм СМП синтезирует многоместные продукты не только представляющие многоместные математические операции, но и описывающие алгоритмы работы с данными, что обуславливает возможность легко сопоставить системе продуктов соответствующую структуру вычислительной системы.

Список литературы: 1. Путьгин Е.П., Климушев В.Б. Математическая модель понятия задачи// АСУ и приборы автоматики. 1983. Вып. 66. С. 9 –12 .

Поступила в редколлегию 29.04.01

Климушев Виктор Борисович, старший преподаватель кафедры информатики ХТУРЭ. Научные интересы: программное обеспечение и синтез структур параллельных вычислительных систем. Адрес: Украина, 61166, Харьков, ул. Космическая, 11, кв. 29, 40–94–19.

УДК 519.7

О.Н. ВОСКОБОЙНИК

ОБЩИЕ СВОЙСТВА МЕТРИЧЕСКИХ ПРЕДИКАТОВ

Рассматриваются свойства предикатов, аксиоматически определяющие на некотором множестве структуру линейного пространства и позволяющие задать метрику.

Метод компараторной идентификации, основанный на изучении свойств бинарных отношений, позволяет выявить структуру неизвестного преобразования с точностью до взаимно-однозначного отображения. Нами уже подчеркивалось [1], что с точки зрения построения математической модели "черного ящика" этого вполне достаточно, поскольку речь идет об изоморфизме моделей, а невозстанавливаемая взаимно-однозначная часть представляет собой некую шкалу, выбор которой во многих реальных задачах может быть произволен.

Вместе с тем, на практике часто знание шкалы необходимо. К подобному выводу несложно прийти, проанализировав круг задач, возникающих в теории измерений, экономике, психофизике и во многих других областях.

В рамках рассмотренных в данной работе моделей, когда речь шла в основном о преобразованиях линейного пространства в R_n , как правило, нелинейная взаимно-однозначная часть по сути дела искривляет n -мерное арифметическое пространство. Еще раз подчеркнем, что на базе бинарных отношений это искривление найти невозможно, более того его нельзя даже "уловить", т. е. установить сам факт его присутствия. Однако идеология компараторного метода может быть сохранена и распространена на случай n -арных отношений. При таком подходе поставленная выше задача может быть решена и это является целью настоящей статьи.

На примере психофизической системы "орган зрения человека" сформулируем некоторые факты, предшествующие формальной постановке задачи.

Рассмотрим процесс распознавания цветов. Многочисленные исследования в области психофизики в совокупности с методом компараторной идентификации позволили в настоящее время прийти к выводу о так называемой трехкомпонентной модели [2] распознавания цветов. Цвет может быть охарактеризован тремя числами: $\alpha_1, \alpha_2, \alpha_3$, а все множество цветов представляет собой некоторую область (цветовой треугольник) в R_3 . Сама процедура распознавания может быть описана в виде следующей математической модели [3]: