Aplikace matematiky

Karel Čulík

Конструкция автоматного отображения

Aplikace matematiky, Vol. 10 (1965), No. 6, 459–468

Persistent URL: http://dml.cz/dmlcz/102987

КОНСТРУКЦИЯ АВТОМАТНОГО ОТОБРАЖЕНИЯ

КАРЕЛ ЧУЛИК II. (Karel Čulík II.)

(Поступило в редакцию 24 апреля 1964 г.)

Путём добавки слева „пустых" букв к выходным словам (временная задержка) можно свести любое однозначное отображение к автоматному отображению. Указывается минимальное необходимое число „пустых" букв для двух случаев:

1. Число „пустых" букв одинаково для всех выходных слов.

2. Для каждого выходного слова в отдельности определяется минимальное необходимое число „пустых" букв.

Введение. Отображение[1]) слов, индуцируемое абстрактным автоматом, называется (см. [1]) автоматным отображением. В [1] стр. 51 показано, что всякое автоматное отображение удовлетворяет следующим четырем условиям:

1. Автоматное отображение осуществляет однозначное отображение (вообще говоря, частичное) множества слов в некотором алфавите $\mathscr{X}$ (называемом входным алфавитом отображения) в множество слов в некотором конечном алфавите $\mathscr{Y}$ (называемом выходным).

2. Область определения автоматного отображения удовлетворяет условию полноты, то есть, иначе говоря, вместе с любым содержащимся в ней словом содержит также все начальные отрезки этого слова. Пустое слово всегда входит в область определения отображения.

3. Автоматное отображение φ сохраняет длину слова: любое слово входного алфавита, на котором отображение φ определено, имеет ту же длину, что и его образ $\varphi(p)$. В частности, пустое слово переводится отображением φ в пустое слово.

4. Автоматное отображение φ переводит любой начальный отрезок слова p, на котором оно определено, в соответствующий (имеющий ту же длину) начальный отрезок слова $\varphi(p)$.

[1]) Если не оговорено противное, то под отображением в общем случае понимаем многозначное отображение.

Автоматные отображения представляют собой только небольшой подкласс всех однозначных алфавитных отображений. Возможно, однако, превратить всякое однозначное алфавитное отображение в автоматное. Это делается с помощью двух операций: „операции выравнивания длин слов“ и „операции пополнения отображения“, которые описаны в [1] стр. 55–56 следующим способом:

Пусть φ – произвольное (вообще говоря, – частичное) отображение множества слов в (конечном) алфавите $\mathscr{X}$ в множество слов в (конечном) алфавите $\mathscr{Y}$. Обозначим через P область определения этого отображения. Будем применять к отображению φ две операции. Первая из них, которую мы назовем операцией выравнивания длин слов (или, кратко, операцией выравнивания), заключается в следующем. Во входной и выходной алфавиты отображения φ (то есть в $\mathscr{X}$ и $\mathscr{Y}$) добавляется по одной новой букве, которые мы будем называть пустыми и обозначать через α и β (иногда в качестве таких букв могут быть выбраны буквы алфавитов $\mathscr{X}$ и $\mathscr{Y}$; не исключено также совпадение букв α и β); к любому слову p из P приписывается справа m_p экземпляров букв α, а к его образу $q = \varphi(p)$ приписывается слева n_q экземпляров букв β так, чтобы длины полученных в результате приписывания новых букв слов p_1 и q_1 совпали.

Далее строится новое отображение φ_1 некоторого множества P_1 слов в алфавите $\mathscr{X} \cup (\alpha)$ в множество слов в алфавите $\mathscr{Y} \cup (\beta)$, которое переводит друг в друга слова p_1 и q_1, полученные в результате выравнивания длин слов p и q соответственно: $\varphi_1(p_1) = q_1$ (p пробегает при этом все множество P).

Примечание. Операцией выравнивания длин слов, описанной в [1], будем пользоваться в немного обобщенном виде, будем приписывать к выходным словам слева n_q и справа $\bar{n}_q$ экземпляров буквы β.

Вторая операция применяется только к выравненным алфавитным отображениям φ, то есть к таким отображениям, у которых длины входных и соответствующих им выходных слов равны между собой. Сущность этой операции, которую мы будем называть операцией пополнения отображения φ, заключается в распространении отображения φ на начальные отрезки слов. Операция пополнения имеет следующий вид: если s – произвольный начальный отрезок любого слова p из области определения отображения φ, то мы полагаем $\varphi(s)$ равным начальному отрезку слова $\varphi(p)$, имеющему равную с начальным отрезком s длину.

Операция пополнения точно определена, но в операции выравнивания нужно в каждом частном случае определить числа m_p, n_q, $\bar{n}_q$ так, чтобы выравненное отображение осталось однозначным после пополнения.

В [1] стр. 124 сказано, что это можно сделать или путем так называемой стандартной операции выравнивания, которая, однако, ведет к лишнему количеству экземпляров пустых букв, или произведением ряда последовательных попыток. Наша цель – показать, как определить минимальное необходимое число пустых букв.

Покажем решение в двух случаях:

1. Число слева присоединенных пустых букв одинаково для всех выходных слов.

2. Необходимое число пустых букв минимально для каждого выходного слова в отдельности.

Число n_q пустых букв, присоединенных слева к выходному слову q, можем понимать как замедление, с которым отображено слово p в слово q.

1. Пусть φ — однозначное алфавитное отображение слов с областью определения P. Элементы P представляют собой конечные последовательности букв из алфавита $\mathscr{X}$, значит, P или конечна или счетна. Из этого следует, что элементы P можно нумерировать, скажем $P \equiv \{p_i\}$, где $i \in M$ и M — подмножество множества натуральных чисел.

Пусть p_i, p_j — пары входных слов $(i, j \in M)$. Обозначим $k_{i,j}$ длину общего начального отрезка этих слов. Если их изображения $q_i = \varphi(p_i)$, $q_j = \varphi(p_j)$ различны, то обозначим $m_{i,j}$ длину общего начального отрезка слов q_i, q_j; если $q_i = q_j$, то положим $m_{i,j} = k_{i,j}$. Далее положим

$$r_{i,j} = \begin{cases} 0\,, & \text{если} \quad m_{i,j} \geqq k_{i,j}\,, \\ k_{i,j} - m_{i,j}\,, & \text{если} \quad k_{i,j} > m_{i,j}\,. \end{cases} \tag{1}$$

Замечание 1. Если не будет ясным, которому отображению $r_{i,j}$ принадлежит, приписываем обозначение отображения как верхний индекс $r_{i,j}$. Например, $r^{\varphi}_{i,j}$ (подобно $k^{\varphi}_{i,j}$).

Замечание 2. Для любых i, j, s имеем

$$k_{i,j} \geqq \min\left(k_{i,s}, k_{j,s}\right), \tag{2}$$

для $k_{i,s} \neq k_{j,s}$ даже $k_{i,j} = \min\left(k_{i,s}, k_{j,s}\right)$.

Замечание 3. $r_{i,j}$ и $k_{i,j}$ симетричны по отношению к своим индексам $k_{i,j} = k_{j,i}$, $r_{i,j} = r_{j,i}$.

Пусть φ — произвольное онднозначное алфавитное отображение. Если в некоторой паре, образованной входным словом p и ему соответствующим выходным словом q, слова p и q различной длины (длина слова p не равна длине $\varphi(p)$), то более короткое слово в паре дополним буквами α или β до длины более длинного. Новые слова обозначим $\bar{p}_i$, $\bar{q}_i$ (одно из них осталось без изменения). Через $\bar{\varphi}$ обозначим отображение, для которого $\bar{\varphi}(\bar{p}) = \bar{q}$ и которое пополнено на начальные отрезки входных слов.

Лемма 1. *Пусть отображение φ — произвольное однозначное алфавитное отображение. В таком случае отображение $\bar{\varphi}$ однозначно тогда и только тогда, когда для любой пары входных слов p_i, p_j*

$$r^{\varphi}_{i,j} = 0\,.$$

Доказательство. 1. Пусть существуют $i, j \in M$, для которых $r_{i,j} > 0$. Тогда для общего начального отрезка длины $k_{i,j}$ слов этой пары выходное слово неопределено однозначно.

2. Пусть $r_{i,j} = k_{i,j} - m_{i,j} = 0$ для всех $i, j \in M$. Тогда из (1) следует $k_{i,j} \leqq m_{i,j}$, а, значит, для всякой пары одинаковых начальных отрезков входных слов их образы одинаковы.

Пусть φ — однозначное алфавитное отображение с областью определения $P \equiv \{p_i\}$. для всех $i \in M$. Пусть n_i (для всех $i \in M$) — целые неотрицательные числа. Припишем к каждому выходному слову q_i слева n_i экземпляров буквы β и полученное таким образом слово обозначим через q'_i. Через φ' обозначим отображение, для которого $\varphi'(p_i) = q'_i$. Наконец, отображение $\overline{(\varphi')}$ обозначим через $\tilde{\varphi}$.

Лемма 2. *Для любого отображения φ с областью определения $P \equiv \{p_i\}$ (для всех $i \in M$), отображение $\tilde{\varphi}$ является однозначным тогда и только тогда, когда n_i (для всех $i \in M$) удовлетворяют для всякой пары $i, j \in M$ $(i \neq j)$ следующим условиям:*

$$\min(n_i, n_j) \geqq k_{i,j} \quad \text{для} \quad n_i \neq n_j\,, \tag{3a}$$

$$n_i \geqq r_{i,j} \quad \text{для} \quad n_i = n_j\,. \tag{3b}$$

Доказательство. Из леммы 1 и (1) следует, что отображение $\tilde{\varphi}$ однозначно тогда и только тогда, когда для всякой пары $i, j \in M$ выполнено

$$m^{\tilde{\varphi}}_{i,j} \geqq k^{\tilde{\varphi}}_{i,j}\,. \tag{4}$$

Для $n_i \neq n_j$

$$k^{\tilde{\varphi}}_{i,j} = k^{\varphi}_{i,j} \quad \text{и} \quad m^{\tilde{\varphi}}_{i,j} = \min(n_i, n_j)\,;$$

из этого следует, что (4) эквивалентно (3a).

Для $n_i = n_j$, $\varphi(p_i) \neq \varphi(p_j)$

$$k^{\tilde{\varphi}}_{i,j} = k^{\varphi}_{i,j} \quad \text{и} \quad m^{\tilde{\varphi}}_{i,j} = n_i + m^{\varphi}_{i,j}\,;$$

из этого следует, что (4) эквивалентно (3b).

Для $n_i = n_j$, $\varphi(p_i) = \varphi(p_j)$

$$k^{\tilde{\varphi}}_{i,j} = m^{\tilde{\varphi}}_{i,j} = k^{\varphi}_{i,j} = ш^{\varphi}_{i,j},\ r_{i,j} = 0\,;$$

из этого следует, что (4) и (3b) всегда выполнены.

Теорема 1. *Пусть φ — однозначное алфавитное отображение с областью определения $P \equiv \{p_i\}$ (для всех $i \in M$). Пусть $n_i = C(C \geqq 0$, целое$)$ для всех $i \in M$. В таком случае выше определенное отображение $\tilde{\varphi}$ является автоматным отображением тогда и только тогда, когда*

$$\max_{\substack{i,j \in M \\ i \neq j}} r_{i,j} \leqq C\,. \tag{5}$$

Доказательство. Условия 2, 3, 4 автоматности отображения выполнены для любого C. Из леммы 2 вытекает, что условие 1 автоматности выполнено тогда и только тогда, когда C удовлетворяет неравенству (5).

2. Теорема 1 решает проблему минимального необходимого замедления, с которым возможно отображение φ осуществлять при помощи абстрактного автомата тогда, когда требуем, чтобы замедление было одинаково для всех входных слов. Далее будем искать минимальные замедления, не требуя, чтобы они были равными. Решение дает следующая теорема, в которой тоже уточнено, в каком смысле замедления минимальны.

Теорема 2. *Пусть φ — однозначное алфавитное отображение с областью определения $P \equiv \{p_i\}$, для всех $i \in M$. Пусть целые неотрицательные числа n_i определены следующим образом:*

$$n_i = \sup_{j \in M} \min\left(k_{i,j}, w_j\right),$$

где

$$w_j = \sup_{\substack{s \in M \\ j \neq s}} r_{j,s},$$

$$w_i = k_{i,i},$$

$k_{i,j}$ для $i = j$, $r_{i,j}$ и $\tilde{\varphi}$ обозначают то же самое, что и выше.

A. *Если n_i, для всех $i \in M$, конечны, то*

1. *отображение $\tilde{\varphi}$ является автоматным,*

2. *если для некоторых других n_i (для всех $i \in M$) соответствующее отображение $\tilde{\varphi}$ однозначно (и, следовательно, автоматно), то для всех $i \in M$ $n'_i \geqq n_i$.*

B. *Если по крайней мере одно n_i бесконечно, то не существуют конечные числа n_i, $n_i \leqq C$ (для всех $i \in M$) такие, чтобы соответствующее отображение $\tilde{\varphi}$ было автоматным.*

Замечание. Если некоторое w_i бесконечно, то подразумеваем

$$\min\left(k_{i,j}, \infty\right) = k_{i,j}.$$

Доказательство: ***A*** 1. Пусть отображение $\tilde{\varphi}$ неоднозначно; тогда из леммы 2 следует, что существуют $i, j \in M$ такие, что (3) не выполняется. Так как $n_i \geqq w_i$ и

$$w_i = \sup_{\substack{j \in M \\ i \neq j}} r_{i,j},$$

то для всех $i \in M$

$$n_i \geqq \sup_{\substack{j \in M \\ i \neq j}} r_{i,j}.$$

Итак, неравенство (3b) не может не выполняться и, следовательно, существуют $i_0, j_0 \in M$ $(i_0 \neq j_0)$ такие, что невыполнено (3a). Предположим, что

$$n_{i_0} > n_{j_0}, \tag{6}$$

(в случае $n_{j_0} > n_{i_0}$ изменим обозначение индексов) и, следовательно,

(7) $$k_{i_0,j_0} > n_{j_0}\,.$$

Потому что n_i (для всех $i \in M$) конечны, можем писать

(8) $$n_{i_0} = \max_{s \in M} \min\left(k_{i_0}, s, w_s\right).$$

Обозначим через s_0 то s в выражении (8), для которого достигается максимума. Итак,

(9) $$n_{i_0} = \min\left(k_{i_0,s_0}, w_{s_0}\right),$$

и (6) можно переписать в виде

(10) $$\min\left(k_{i_0,s_0}, w_{s_0}\right) > n_{j_0}\,.$$

Из (7) и (10) следует, что

$$\min\left(k_{i_0,j_0}, k_{i_0,s_0}, w_{s_0}\right) > n_{j_0}\,,$$

и далее из (2) и в силу определения n_{j_0} получаем

$$\min\left(k_{j_0,s_0}, w_{s_0}\right) > \sup_{s \in M} \min\left(k_{j_0,s}, w_s\right),$$

что есть искомое противоречие.

A 2. Пусть существуют целые неотрицательные числа n'_i (для всех $i \in M$) такие, что соответствующее отображение $\tilde{\varphi}$ однозначно, и пусть для некоторого i_0 имеем $n'_{i_0} < n_{i_0}$. Обозначим опять через s_0 то s в выражении (8), для которого достигается максимума. Итак, опять верно выражение (9). Из (3) и в силу определения w_i получаем

$$w_{s_0} \leqq n'_{s_0}\,.$$

Предположим, во-первых, что $n'_{i_0} = n'_{s_0}$; тогда из (9) следует

$$n'_{i_0} = n'_{s_0} \geqq w_{s_0} \geqq n_{i_0}\,,$$

что противоречит предположению. Итак, остается возможность $n'_{i_0} \neq n'_{s_0}$. В этом случае из (3а) и (10) получаем

$$n'_{i_0} \geqq k_{i_0,s_0} \geqq n_{i_0}\,,$$

что опять противоречит предположению.

B. Если для некоторого s

$$n_s = \sup_{j \in M} \min\left(k_{s,j}, w_j\right) = \infty\,,$$

то и

(11) $$\sup_{j \in M} w_j = \sup_{\substack{j,s \in M \\ j \neq s}} r_{j,s} = \infty\,.$$

Пусть для некоторых n_j (для всех $j \in M$) соответствующее отображение $\tilde{\varphi}$ однозначно. Так как $k_{j,s} \geqq r_{j,s}$, то из леммы 2 следует

$$n'_j \geqq r_{j,s} \quad \text{для всех} \quad j, s \in M \, (j \neq s)$$

и из (11) следует, что не существует C такое, чтобы

$$n'_j \leqq C \quad \text{для всех} \quad j \in M \, .$$

3. Пример. Превратим в автоматное следующее алфавитное отображение:

Входной алфавит: a, b, c, d, e, ,,пустая буква“ α.

Выходной алфавит: 0, 1, ,,пустая буква“ β.

Соответствующие пары слов:

очередный номер	входное слово	выходное слово
1	*a b c d*	0 0 1
2	*a b c e e*	0 0 0 1 1
3	*b a d a b c*	1 0 1 0 1
4	*b a d a c*	1 0 0 0 1
5	*b a d b*	1 0 1 1 1
6	*b e a a c*	1 1 1 0
7	*b e a a b*	1 0 1 0 1 1
8	*c d*	0 1 0
9	*d e e a*	0 0 1
10	*e d a*	0 1 1 1

Таблица $k_{i,j}$

	2	3	4	5	6	7	8	9	10
1	3	0	0	0	0	0	0	0	0
2		0	0	0	0	0	0	0	0
3			4	3	2	2	0	0	0
4				3	2	2	0	0	0
5					2	2	0	0	0
6						4	0	0	0
7							0	0	0
8								0	0
9									0

Таблица $r_{i,j}$

	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0
2		0	0	0	0	0	0	0	0
3			2	0	1	0	0	0	0
4				1	1	0	0	0	0
5					1	0	0	0	0
6						3	0	0	0
7							0	0	0
8								0	0
9									0

Таблица w_i

$w_1 = 1$	$w_6 = 3$
$w_2 = 0$	$w_7 = 3$
$w_3 = 2$	$w_8 = 0$
$w_4 = 1$	$w_9 = 0$
$w_5 = 1$	$w_{10} = 0$

Искомое n_i

$n_1 = 1$	$n_6 = 3$
$n_2 = 1$	$n_7 = 3$
$n_3 = 2$	$n_8 = 0$
$n_4 = 2$	$n_9 = 0$
$n_5 = 2$	$n_{10} = 0$

4. Запишем, далее, алгорифм нахождения минимальных n_i (минимальных в отдельности для каждого входного слова), для случая отображения заданного конечным числом пар слов, как процедуру на языке Algol. Пусть отображение слов (входные и выходные слова дополнены пустыми буквами до равной длины) записано на ленте данных так, что

1. различным буквам соответствуют различные целые числа;
2. отдельные буквы слова (целые числа) следуют в очереди: первая, вторая, ...;
3. отдельные слова следуют в очереди: первое входное слово, первое выходное слово, второе входное слово, ... $(p_1, q_1, p_2, q_2, p_3, \ldots)$.

Предполагаем, что процедура read (s) читает одно значение из ленты данных, присваивает его переменной s и подвигает ленту на одно место.

```
procedure минимальное запаздивание (n) число слов: (d) длина одного слова: (g)
  value d, g; integer d, g; integer array n;
  begin integer array w, n [1 : d], r, k [1 : d, 1 : d], p, q [1 : d, 1 : g];
    integer i, j, e, s;
    integer procedure min (a, b); integer a, b;
      min := if a < b then a else b;
    for i := 1 step 1 until d do
      begin for j := 1 step 1 until g do read (p[i, j]);
        for j := 1 step 1 until g do read (q[i, j]);
      end;
```

```
for i := 1 step 1 until d do
  for j: = 1 step 1 until i − 1 do
    begin s: = 0;
      for e: = 1 step 1 until g do
        if p[i, e] = p[j, e] then s:= s + 1 else go to L 1;
      L 1: k[i, j] := s;
          s:= 0;
      for e:= 1 step 1 until g do
        if q[i, e] = q[j, e] then s:= s + 1 else go to L 2;
      r[i, j]:= 0;
      go to L 3;
    L 2: r[i, j]:= if s ≧ k[i, j] then 0 else k[i, j] − s;
    L 3: end вычисления r[i, j] и k[i, j], следует вычисление w[i] и n[i];

for i:= 1 step 1 until d do
  begin s:= 0;
    for j:= 1 step 1 until i − 1 do
      if r[i, j] > s then s:= r[i, j];
    for j:= i + 1 step 1 until d do
      if r[j, i] > s then s:= r[j, i];
    w[i]:= k[i, i] := s;
  end;

for i:= 1 step 1 until d do
  begin s:= 0;
    for j:= 1 step 1 until i − 1 do
      if min (k[i, j], w[j]) > s then s:= min (k[i, j], w[j]);
    for j:= i step 1 until d do
      if min (k[j, i], w[j]) > s then s:= min (k[j, i], w[j] min (k[i, j,], w[j]);
    n[i]:= s
  end вычисления n[i];

end минимальное запаздивание
```

Эта программа была проведена для выше рассмотренного примера на машине National Elliott 803 B.

Литература

[1] *Глушков В. М.*: Синтез цифровых автоматов. Физматгиз 1962.

Souhrn

KONSTRUKCE AUTOMATOVÉHO ZOBRAZENÍ

KAREL ČULÍK II.

Jsou uvedeny podmínky, při jejichž splnění je zobrazení slov automatové (viz [1]). Každé jednoznačné zobrazení může být převedeno na automatové zobrazení přidáním určitého počtu „prázdných" symbolů zprava ke každému vstupnímu slovu a zleva ke každému výstupnímu slovu. Původní zobrazení je pak realizováno s určitým „zpožděním".

V práci je nalezen minimální nutný počet „prázdných" symbolů pro dva případy:

1. Počet „prázdných" symbolů je stejný pro všechna výstupní slova (konstantní „zpoždění").
2. Počet „prázdných" symbolů je minimální pro každé výstupní slovo zvlášť.

Algoritmus pro nalezení minimálního počtu „prázdných" symbolů (pro každé slovo zvlášť) je popsán v jazyce ALGOL 60.

Summary

CONSTRUCTION OF THE AUTOMATON MAPPING

KAREL ČULÍK II.

A constitutes an automaton mapping if it fulfills the condition stated in [1]. Every mapping of words can be transformed into an automaton mapping by concatenating a certain number of empty symbols to the end of every imput word and a number of empty symbols to the beginning of every output word; by this process, additional delays are introduced.

The minimum number of empty symbols in both the following cases is given:

1. The number of empty symbols is the same for all output words.
2. The number of empty symbols is minimal for every individual output word.

The algorithm for finding the minimum number of empty symbols (in the second case) is described in ALGOL 60.

Адрес автора: Karel Čulík II., Matematická laboratoř ČVUT, Horská 4, Praha 2.