

УДК 681.142.2

## ТЕХНОЛОГИЧЕСКИ ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ И ПРОГРАММ

*В.Я. Жихарев, д-р техн наук*

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ»*

Рассмотрены теоретические аспекты автоматизированного проектирования алгоритмов и программ, базовой моделью которых являются рекурсивные автоматы с программируемой логикой. В качестве инструментальных средств проектирования используются алгоритмические алгебры.

\* \* \*

Розглянуто теоретичні аспекти автоматизованого проектування алгоритмів і програм, базовою моделлю яких є рекурсивні автомати з програмованою логікою. Як інструментальні засоби проектування використовуються алгоритмічні алгебри.

\* \* \*

A new approach to designing algorithms and programs using its structural properties represented in recursive form is presented. As tool of designing algorithmic algebra's are used.

### Введение

Методы проектирования автоматизированных систем могут быть успешно реализованы при использовании высокоуровневых языков описания алгоритмов в соответствующей форме. Впервые один из таких языков – язык регулярных схем алгоритмов (РСА) – предложил в 1965 г. В.М. Глушков, заложив тем самым формальные основы структурного программирования. РСА являются конструкциями систем алгоритмических алгебр (САА), разработанных В.М. Глушковым, исходя из представления абстрактной модели ЭВМ посредством дискретного преобразователя (ДП)-композиции управляющего и операционного автомата [1].

### Формулирование целей статьи

В настоящей работе предложена принципиально новая теория алгоритмических алгебр (АА), которая по сравнению с другими способами задания алгоритмов отличается следующим:

- наглядностью описания и точным соответствием принципам структурного программирования;
- возможностью многокритериальной оптимизации по разным критериям последовательных и параллельных алгоритмов;

- возможностью представления в рекурсивной форме языков высокого уровня из иерархии Хомского в целях их последующей реализации стандартным образом;
- обеспечением естественного перехода от выражений в АА к программе или автомату, реализующим заданные выражения;
- унификацией способов обработки информации средствами АА.

### Решение проблемы

Система автоматизированного проектирования состоит из подсистемы технологически ориентированного проектирования формальных языков и подсистемы технологически ориентированного проектирования алгоритмов и программ (рис.1, где R – ДП - рекурсивный дискретный преобразователь, ЛА – лексический анализатор, СА - синтаксический анализатор) [1].

Подсистема технологически ориентированного проектирования формальных языков состоит из блока грамматик, блока алгоритмических алгебр, блока анализа, блока погружения алгебры рекурсивных событий (АРЕКС) в рекурсивные системы интерпретированных алгоритмических алгебр (Р-СИАА) и блока настройки на требуемый язык [2].

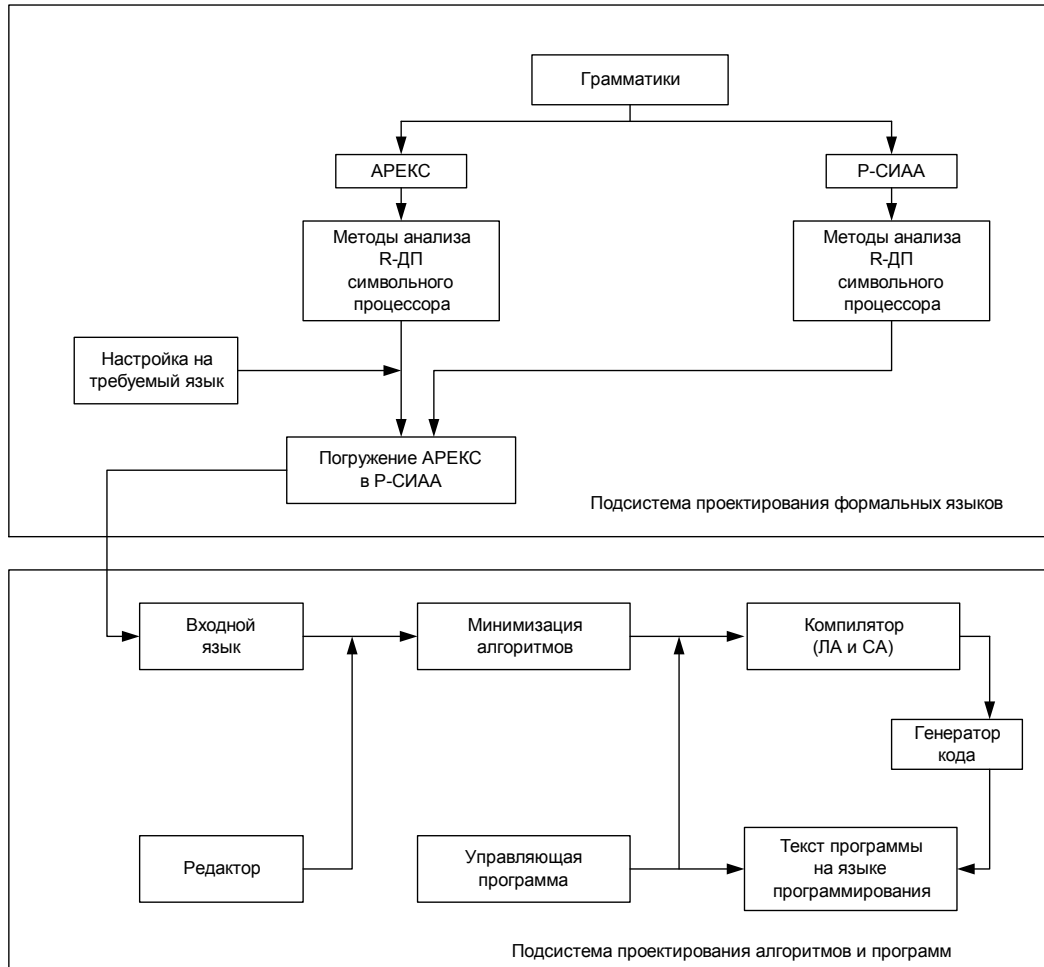


Рис.1. Система технологически ориентированного проектирования формальных языков, алгоритмов и программ

Входной информацией для проектирования формальных языков являются формальные грамматики, порождающие множество языков из классификации Хомского (автоматная грамматика, праволинейная и леволинейная грамматики, контекстно - свободная грамматика (КС-грамматика), контекстно - зависящая грамматика (КЗ-грамматика) и грамматика общего вида)[ 3,4].

В общем случае непосредственное построение алгоритмов по заданной грамматике невозможно в связи с ее описанием с помощью сложных правил вывода по древовидной схеме.

Для решения задачи технологического проектирования формальных языков применяют алгебраические средства АРЕКС и Р-СИИА, обладающие высоким уровнем формализации. Эти алгоритмиче-

ские алгебры позволяют представить в алгебраической форме формальные языки различного уровня посредством их представления в виде рекурсивных событий ( $r$ -событий) и рекурсивных формул ( $r$ -формул) алгоритмов [2].

Для анализа математической модели проектирования формальных языков разработаны матричный, реляционный и алгебраические методы [5].

Эти методы позволяют анализировать и синтезировать операционную структуру символического процессора, разработанного в виде рекурсивного автомата с программируемой логикой (РАПЛ) [6].

Для анализа и синтеза управляющей структуры РАПЛ применяют реляционный и комплексный методы.

Разработка символьного процессора в виде РАПЛ проводится в соответствии с этапами анализа, оптимизации, абстрактного, структурного и логического синтеза.

На этапе анализа входные грамматики погружаются в АРЕКС для определения  $r$ -событий с последующей их оптимизацией.

Для анализа символьного процессора разработаны матричный, реляционный алгебраический методы, предусматривающие построение алгоритмов представимости автоматных, КС-языков, КЗ-языков и языков общего вида посредством  $r$ -событий. Цель представимости – построение рекурсивного процессора данных, являющегося основой информационной базы символьного процессора, т.е. параметрически настраиваемой частью операционной системы.

Необходимость разработки  $r$ -событий вызвана тем, что непосредственное нахождение алгоритмов реализации формальных языков высшего уровня по заданной грамматике практически невозможно в связи с трудностями сопоставления терминалов с логическими условиями. Таким образом,  $r$ -событие – это промежуточная структура, необходимая для автоматизированного проектирования алгоритмов в Р-СИИА.

Разработанные методы ориентируются на программную реализацию.

Подсистема технологически ориентированного проектирования алгоритмов и программ состоит из компилятора (лексического и синтаксического анализаторов), блока минимизации структур алгоритмов на основе алгоритмической алгебры, редактора входного языка.

В систему минимизации алгоритмов на основе Р-СИИА входят:

- минимизация структур по числу операторов и логических условий с универсальным распределением сдвигов;

- минимизация структур по числу операторов и логических условий с заданным распределением сдвигов;
- минимизация структур по числу операторов и логических условий с учетом неиспользуемых наборов логических переменных;
- изоморфизм структур алгоритмов;
- объединения структур алгоритмов.

Например, по исходному выражению

$$\lg \frac{\bar{q}(\Phi)}{q(0)} = a \cdot \lg \bar{q}_1 + b \cdot \lg \bar{q}_2 + c \lg \Phi + d$$

запишем грамматику, порождающую язык этого выражения:

$$E \rightarrow E + T$$

$$E \rightarrow T$$

$$T \rightarrow F \times T$$

$$T \rightarrow F$$

$$F \rightarrow G/S$$

$$F \rightarrow G$$

$$G \rightarrow G = E$$

$$G \rightarrow K$$

$$K \rightarrow \lg(E)$$

$$K \rightarrow H$$

$$H \rightarrow \bar{q}(E)$$

$$H \rightarrow N$$

$$N \rightarrow q(0)$$

$$N \rightarrow a | b | c | d | \bar{q}_1 | \bar{q}_2 | \Phi | \bar{q} | q$$

Полученная грамматика является КС-грамматикой вида  $\{a^n b^n c^n\}$ . Преобразуем полученную КС-грамматику в  $r$ -событие с использованием алгоритма

$$P = \{\lg\}^+ \vee \{\{\}^+ \vee \{a \vee b \vee c \vee \bar{q} \vee q \vee d \vee \bar{q}_1 \vee \bar{q}_2 \vee \Phi \Pi_2(+ \vee \times \vee \backslash \vee =)\}^+ \vee \Pi_2\} \#.$$

Проведем разметку состояний  $r$ -события

$$P = \left| \begin{array}{cccccccc} \{\lg\}^+ & \vee & \{\{\}^+ & \vee & \{a & \vee & b & \vee & c & \vee & \bar{q} & \vee \\ 1 & & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \vee & | & q & \vee & d & \vee & \bar{q}_1 & \vee & \bar{q}_2 & \vee & \Phi & | \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ \vee & | & \backslash & \vee & | & = & | \}^+ & \vee & \Pi_2 & | & \{ \} & | \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 2 & 3 \end{array} \right| \#.$$

Граф автомата, соответствующий  $r$ -событию  $P$ , имеет вид, показанный на рис.2.

Для полученного  $r$ -события запишем систему уравнений в АРЕКСе

$$P_1 = \lg P_1 \vee (P_1 \vee (a \vee b \vee c \vee \bar{q} \vee \vee q \vee d \vee \bar{q}_1 \vee \bar{q}_2 \vee \Phi)P_2.$$

$$P_2 = (+ \vee \times \vee \backslash \vee =)P_1 \vee P_2 \vee \# P_3.$$

Погрузим эту систему в Р-СИАА, т.е найдем ДФ алгоритма путем кодирования исходных данных:

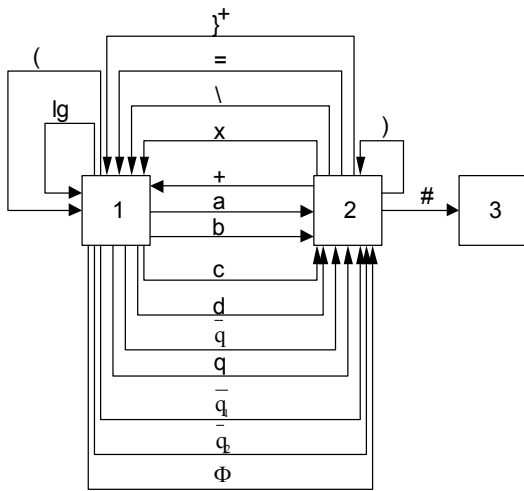


Рис. 2. Граф автомата, соответствующий  $r$ -событию  $P$

$$P_1 = (S_1 P_1 \vee (S_2 P_1 \vee (S_3 P_2 \vee (S_4 P_2 (S_5 P_2 \vee \vee (S_6 P_2 \vee (S_7 P_2 \vee (S_8 P_2 \vee (S_9 P_2 \vee (S_{10} P_2 \vee \vee (S_{11} P_2 \vee N))))))))))$$

$$P_2 = (S_{12} P_1 \vee (S_{13} P_1 \vee (S_{14} P_1 \vee (S_{15} P_1 \vee \vee (S_{16} P_2 \vee (S_{17} \vee N))))))$$

Решение системы уравнений проводим реляционным методом:

$\bar{P}$	$P_1$		$P_2$		
	2: $P_1, P_2$		2: $P_1, P_2$		
$\alpha_1$	$S_1$	$P_1$	$\alpha_{12}$	$S_{12}$	$P_1$
$\alpha_2$	$S_2$	$P_1$	$\alpha_{13}$	$S_{13}$	$P_1$

$\alpha_3$	$S_3$	$P_2$	$\alpha_{14}$	$S_{14}$	$P_1$
$\alpha_4$	$S_4$	$P_2$	$\alpha_{15}$	$S_{15}$	$P_1$
$\alpha_5$	$S_5$	$P_2$	$\alpha_{16}$	$S_{16}$	$P_2$
$\alpha_6$	$S_6$	$P_2$	$\alpha_{17}$	$S_{17}$	-
$\alpha_7$	$S_7$	$P_2$	-	-	-
$\alpha_8$	$S_8$	$P_2$	-	-	-
$\alpha_9$	$S_9$	$P_2$	-	-	-
$\alpha_{10}$	$S_{10}$	$P_2$	-	-	-
$\alpha_{11}$	$S_{11}$	$P_2$	-	-	-

Определяем  $\bar{P}$  ранги нетерминалов, применяем к атрибуту  $P_2$  операцию  $\vee_\alpha$ -проекции, проводим свертку циклов для атрибутов  $P_1$  и  $P_2$ . В результате получаем таблицу:

$\bar{P}$	$P_1$	$P_2$
	1: $P_2$	1: $P_1$
	$\{S_1\}^+ \vee \{S_2\}^+,$ $(S_3 \vee (S_4 \vee \vee (S_5 \vee (S_6 \vee \vee (S_7 \vee (S_8 \vee \vee (S_9 \vee (S_{10} \vee \vee (S_{11} \vee N))))))))))$	$(S_{12} \vee (S_{13} \vee \vee (S_{14} \vee (S_{15} \vee \vee N))))),$ $\{S_{16}\}^+,$ $\alpha_{12} \cdot \alpha_{13} \cdot \alpha_{14} \times \alpha_{15} \cdot \alpha_{16} \times \alpha_{17} \cdot S_{17}.$

Подставим значение атрибута  $P_2$  в  $P_1$  вместо нетерминала  $P_2$  и, проведя свертку циклов, получим искомую  $r$ -формулу

$$P_1 = \{S_1\}^+ \vee \{S_2\}^+ \vee \{y (S_{12} \vee (S_{13} \vee (S_{14} \vee \vee (S_{15} \vee N))))\}^+ \vee y \cdot \alpha_{12} \times \alpha_{13} \cdot \alpha_{14} \cdot \alpha_{15} \cdot \alpha_{16} \cdot \alpha_{17} \cdot S_{17},$$

где  $y = (S_3 \vee (S_4 \vee (S_5 \vee (S_6 \vee (S_7 \vee \vee (S_8 \vee (S_9 \vee (S_{10} \vee (S_{11} \vee N))))))))))$ .

Еще одной формой представления является стандартный квазирегулярный полином [5]

$$P_1' = \{S_1\}_{\alpha_1}^+ \vee \{S_2\}_{\alpha_1\alpha_2}^+ \vee \{y \cdot \underline{\alpha}_{12} \cdot S_{12}\}^+ \vee \\ \vee \{y \cdot \underline{\alpha}_{12} \cdot \underline{\alpha}_{13} \cdot S_{13}\}^+ \vee \{y \cdot \underline{\alpha}_{12} \cdot \underline{\alpha}_{13} \cdot \underline{\alpha}_{14} \cdot S_{14}\}^+ \vee \\ \vee \{y \cdot \underline{\alpha}_{12} \cdot \underline{\alpha}_{13} \cdot \underline{\alpha}_{14} \cdot \underline{\alpha}_{15} \cdot S_{15}\}^+ \vee \\ \vee y \cdot \underline{\alpha}_{12} \cdot \underline{\alpha}_{13} \cdot \underline{\alpha}_{14} \cdot \underline{\alpha}_{15} \cdot \underline{\alpha}_{16} \cdot S_{16}^+ \vee y \cdot \overline{\alpha}_{12} \cdot \overline{\alpha}_{13} \cdot \overline{\alpha}_{14} \cdot \overline{\alpha}_{15} \times \\ \times \overline{\alpha}_{16} \cdot \underline{\alpha}_{17} \cdot S_{17}.$$

С учетом показателей переходов получим

$$P_1'' = \{S_1\}_{\alpha_1}^+ \vee \{S_2\}_{\alpha_1\alpha_2}^+ \vee \{y \Pi_2 \cdot (S_{12} \vee \\ \vee (S_{13} \vee (S_{14} \vee (S_{15} \vee N))))\}_{\alpha_{13}}^+ \vee \\ \vee \Pi_2 \cdot \{S_{16}\}_{\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{15}\alpha_{16}}^+ \vee \Pi_2^+ \vee \\ \vee \alpha_{12} \cdot \overline{\alpha}_{13} \cdot \overline{\alpha}_{14} \cdot \overline{\alpha}_{15} \cdot \overline{\alpha}_{16} \cdot \underline{\alpha}_{17} \cdot S_{17}.$$

## Заключение

Таким образом, методика технологически ориентированного проектирования алгоритмов и программ заключается в следующем:

1. Получить у заказчика (разработчика) модель процесса или объекта в виде спецификации.
2. По модели или спецификации создать базы данных.
3. Представить модель или спецификацию процесса в виде грамматики. Разработать синтаксические диаграммы или бэкусовскую нормальную форму, если пользователь применяет специальный входной язык.
4. Преобразовать грамматику или синтаксические диаграммы в  $r$ -событие на языке АРЕКС, для чего:
  - записать систему уравнений;
  - применить соответствующее правило вывода;
  - используя тождественные преобразования, записать  $r$ -событие.
5. Преобразовать  $r$ -событие в рекурсивную форму методом погружения  $r$ -события в Р-СИИА, для чего:

- сделать разметку состояний  $r$ -события по соответствующим правилам;
- записать систему уравнений;
- закодировать терминалы полученных уравнений набором значений логических условий;
- привести уравнения к фильтрам;
- решить полученные уравнения, применив соответствующие правила вывода, и записать полученное решение, т.е.  $r$ -формулу в Р-СИИА.

6. Провести минимизацию полученной  $r$ -формулы.
7. Провести лексический и синтаксический анализы  $r$ -формулы алгоритма.
8. Сгенерировать код и получить требуемую программу на языке программирования.

## Литература

1. Жихарев В.Я., Илюшко В.М., Чумаченко И.В. Проектирование электронных компиляторов. – Х.: Факт. – 1999. – 88 с.
2. Жихарев В.Я., Илюшко В.М., Чумаченко И.В. Математические основы проектирования рекурсивных автоматов с программируемой логикой. – Х.: Факт. – 1999. – 144 с.
3. Хомский Н. Три модели описания языка // Кибернетический сборник. – 1961. – Вып. 2. – С.237-266.
4. Методы проектирования символьных процессоров / В.Я. Жихарев, В.М. Илюшко, Н.В. Нечипорук, И.В. Чумаченко. – Х.: Факт. – 2000. – 184 с.
5. Жихарев В.Я. Обобщенная модель символьного процессора // Открытые информационные и компьютерные интегрированные технологии. – Х.: Харьк. авиац. ин-т. – 1998. – Вып. 1. – С. 138-144.

Поступила в редакцию: 13.02.03

**Рецензент:** д-р техн. наук, профессор Харченко В.С., Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», г. Харьков.