

Статья. Библиографические данные:

Житников А.П.  
Базовая структурная семантика параллельных алгоритмов.  
/ Принятие решений в условиях неопределенности: Межвуз. научн. сб.  
– Уфа: УГАТУ, 2008. – С. 41-50.

## **БАЗОВАЯ СТРУКТУРНАЯ СЕМАНТИКА ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ**

Представлен электронный вариант оригинала печатной статьи.  
Статья приводится с технической доработкой:  
гиперссылки, оглавление, цветные элементы и т.п.

### **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ.....	2
1 ИСХОДНЫЕ ПОЛОЖЕНИЯ.....	3
Пример 1.1. Формулы потока данных и потока управления алгоритма.....	3
Примечание 1.1. Структурные формулы и логические схемы алгоритмов ....	4
Примечание 2.2. Структуры алгоритмов базового структурного класса .....	4
2 СТРУКТУРНАЯ СЕМАНТИКА СТРУКТУРНЫХ ФОРМУЛ И СХЕМ.....	6
2.1 Семантика структурных формул алгоритмов .....	6
2.2 Последовательные алгоритмы .....	6
Пример 2.1. Линейный алгоритм.....	6
Пример 2.2. Алгоритм с условным ветвлением типа "если-то" .....	8
Пример 2.3. Алгоритм с условным ветвлением типа "если-то-иначе" .....	10
2.3 Параллельные алгоритмы.....	12
Роль операторов распараллеливания (дивергенции) и соединения (конвергенции).....	13
Синтаксические формы записи структурной формулы.....	15
Уточнение двухмерной структурной формулы алгоритма .....	15
3 ФУНКЦИОНАЛЬНАЯ СЕМАНТИКА ПАРАЛЛЕЛЬНЫХ СТРУКТУР .....	18
Явная функциональная семантика.....	18
Функции структурных элементов.....	18
ЛИТЕРАТУРА.....	19

## ВВЕДЕНИЕ

Параллельный алгоритм – это предписание выполнить определенный комплекс действий в определенном порядке, допускающем совмещение (параллелизм) выполнения действий во времени.

Существует тенденция полиморфного представления алгоритмических предписаний – совместное использование разных согласованных знаковых форм разного назначения, в частности [1]:

- структурные и функциональные формулы алгоритмов – компактное аналитическое описание алгоритмов в разных синтаксических классах (инфиксные, префиксные, постфиксные, комбинированные формы);
- структурные и функциональные схемы алгоритмов разных видов (блок-схемы, штрих-схемы, графы-схемы и т.п.), временные и позиционные диаграммы, сетевые диаграммы, мнемосхемы и т.д.;
- псевдокоды алгоритмов (в разных синтаксических классах и лексических группах) и т.п.

Ключевую роль играет аналитическая форма алгоритмов, представленная структурными и функциональными формулами алгоритмов. Однако ее применение пока достаточно ограничено, что связано с определенными причинами (подлежащими выявлению).

# 1 ИСХОДНЫЕ ПОЛОЖЕНИЯ

## Функциональные и структурные формулы алгоритмов

Любая вычислительная формула с наличием многоместных функций предопределяет (обширную в общем случае) группу разных допустимых структур алгоритмов последовательного и параллельного типа.

Такие структуры алгоритмов могут быть представлены некоторыми структурными и функциональными (структурно-функциональными) формулами и системами таких формул.

### Пример 1.1. Формулы потока данных и потока управления алгоритма

Задана (функциональная) формула:

$$y = f_3(f_1(x_1), f_2(x_2)).$$

Можно добавить функцию  $(x_1, x_2) = f_0(x)$ , например, подготовки исходных данных (компонент  $x_1, x_2$  исходного объекта данных  $x$ ). Образуется некоторая общая функция  $y = f(x)$  с системой составляющих функций  $f_0, f_1, f_2, f_3$  (и промежуточных переменных):

$(x_1, x_2) = f_0(x)$	$y_1 = f_1(x_1)$	$y_2 = f_2(x_2)$	$y = f_3(y_1, y_2)$
-----------------------	------------------	------------------	---------------------

При этом система четырех составляющих функций предопределяет две сопряженные структуры алгоритма исполнения (в частности, вычисления) функции:

1) **Поток данных** (в некотором **канале данных**) – простая разветвленная схема цепей обработки данных, представленная общей (сводной) структурной формулой алгоритма (СФА, Табл. 1.1). В данном случае это структурно-функциональная формула (СФФА, СФА = СФФА). Причем представлена схемоподобная **двухмерная форма** (ДМФ) **структурной формулы** (СФА/ДМФ).

Табл. 1.1. Структурная (структурно-функциональная) формула потока данных алгоритма

ЛТА: Литерный текст алгоритма		//ЛЗФ: Литерная знаковая форма	
КД: Канал (поток) данных			
СФА: Структурная формула алгоритма / ДМФ: Двухмерная форма (СФА/ДМФ)			
СФФА: Структурно-функциональная формула алгоритма (СФА = СФФА)			
Courier: Шрифт типа Courier – запись с постоянным шагом // Проблемы воспроизведения спецзнаков (→) в полужирном тоне			
$x \rightarrow f_0 \mid \rightarrow x_1 \rightarrow f_1 \rightarrow y_1 \mid \mid f_3 \rightarrow y$ $\mid \rightarrow x_2 \rightarrow f_2 \rightarrow y_2 \rightarrow \mid$		$\mathbf{x \rightarrow f_0 \mid \rightarrow x_1 \rightarrow f_1 \rightarrow y_1 \mid \mid f_3 \rightarrow y}$ $\mid \rightarrow x_2 \rightarrow f_2 \rightarrow y_2 \rightarrow \mid$	
Times: шрифт типа Times – запись с переменным шагом Спецзнак → (стрела) – шрифт типа Symbol // Проблемы с позиционированием символов разных строк			
$x \rightarrow f_0 \mid \rightarrow x_1 \rightarrow f_1 \rightarrow y_1 \rightarrow \mid \mid f_3 \rightarrow y$ $\mid \rightarrow x_2 \rightarrow f_2 \rightarrow y_2 \rightarrow \mid$		$\mathbf{x \rightarrow f_0 \mid \rightarrow x_1 \rightarrow f_1 \rightarrow y_1 \rightarrow \mid \mid f_3 \rightarrow y}$ $\mid \rightarrow x_2 \rightarrow f_2 \rightarrow y_2 \rightarrow \mid$	

2) **Поток (команд) управления** (Табл. 1.2) порядком выполнения действий (в котором **канале управления**) – 3 варианта допустимых порядков выполнения команд Z0..Z3:

Табл. 1.2. Структурно-функциональные формулы потока управления алгоритма

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
КУ: Канал (поток) управления	
СФА: Структурная формула алгоритма / ОМФ: Одномерная форма (СФА/ОМФ)	
СФФА: Структурно-функциональная формула алгоритма (СФА = СФФА)	
A1 = [Z0 :: (x1, x2) = f0(x)] → [Z1 :: y1 = f1(x1)] → [Z2 :: y2 = f2(x2)] → [Z3 :: y = f3(x1,x2)]	
A2 = [Z0 :: (x1, x2) = f0(x)] → [Z2 :: y2 = f2(x2)] → [Z1 :: y1 = f1(x1)] → [Z3 :: y = f3(x1,x2)]	
A3 = [Z0 :: (x1, x2) = f0(x)] → {[Z1 :: y1 = f1(x1)]    [Z2 :: y2 = f2(x2)]} → [Z3 :: y = f3(x1,x2)]	

В абстрагировании от конкретных функций потока данных получают обобщенные **структурные формулы потока управления** (Табл. 1.3):

Табл. 1.3. Структурно-функциональные формулы потока управления алгоритма

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
КУ: Канал (поток) управления	
СФА: Структурная формула алгоритма / ОМФ: Двухмерная форма (СФА/ОМФ)	
A1 = Z <sub>0</sub> Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> = Z <sub>0</sub> → Z <sub>1</sub> → Z <sub>2</sub> → Z <sub>3</sub>	– (линейный) последовательный алгоритм
A2 = Z <sub>0</sub> Z <sub>2</sub> Z <sub>1</sub> Z <sub>3</sub> = Z <sub>0</sub> → Z <sub>2</sub> → Z <sub>1</sub> → Z <sub>3</sub>	– (линейный) последовательный алгоритм
A3 = Z <sub>0</sub> (Z <sub>1</sub>    Z <sub>2</sub> )Z <sub>3</sub> = Z <sub>0</sub> → (Z <sub>1</sub>    Z <sub>2</sub> ) → Z <sub>3</sub>	– параллельный алгоритм

Алгоритм A3 представляет первичную параллельную структуру управления с совмещением (параллелизмом) выполнения (Z<sub>1</sub> || Z<sub>2</sub>) двух команд во времени: Z<sub>1</sub> :: y<sub>1</sub> = f<sub>1</sub>(x<sub>1</sub>) и Z<sub>2</sub> :: y<sub>2</sub> = f<sub>2</sub>(x<sub>2</sub>).

### Примечание 1.1. Структурные формулы и логические схемы алгоритмов

1) Для линейных алгоритмов уплощенные (бесстрелочные) структурные формулы алгоритма типа Z<sub>0</sub>Z<sub>1</sub>Z<sub>2</sub>Z<sub>3</sub> и Z<sub>0</sub>Z<sub>2</sub>Z<sub>1</sub>Z<sub>3</sub> совпадают с так называемыми логическими схемами алгоритмов (ЛСА) [2] для таких линейных алгоритмов. На самом деле:

- это особые формулы алгоритмов, которые лежат в исходной основе (в базе) излагаемого языка (сопряженных) структурных формул и схем;
- они сами являются частным случаем таких структурных формул алгоритмов.

2) Для параллельного алгоритма упрощенная (бесстрелочная) структурная формула алгоритма типа Z<sub>0</sub>(Z<sub>1</sub> || Z<sub>2</sub>)Z<sub>3</sub> непосредственно не включается в язык ЛСА [2], но выводится из ЛСА параллельного типа (ПЛСА) – это будет показано далее.

### Примечание 2.2. Структуры алгоритмов базового структурного класса

1) Приведенные выше структурные формулы (потока управления) алгоритмов являются простейшими представителями алгоритмов базового структурного класса:

двухполюсные, постоянные (не переменные – без альтернативных условных переключений и без оперативного порождения компонент), ациклические, обыкновенные (не конвейерного типа) структуры.

2) В частности:

- линейные структуры типа  $Z_0Z_1Z_2Z_3 = Z_0 \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3$  представляют класс базовых структур последовательных алгоритмов;
- нелинейные структуры типа  $Z_0(Z_1 \parallel Z_2)Z_3 = Z_0 \rightarrow (Z_1 \parallel Z_2) \rightarrow Z_3$  представляют класс базовых структур параллельных алгоритмов.

3) Это простейшие структуры параллельных и последовательных алгоритмов ключевого опорного значения:

- это достаточно широко распространенные классы структур алгоритмов, особенно в технических и организационно-технических приложениях;
- на их основе поэтапно определяются любые другие структуры – методом расширения языка структурного описания алгоритмов;
- на этой простейшей структурной основе выявляется и обрабатывает обширная многоаспектная алгоритмическая проблематики базового значения, общая и переносимая на все прочие производные структурные классы.

4) В данной статье отражается, в основном, базовая структурная семантика формул и схем алгоритмов, которая относится к указанному базовому структурному классу алгоритмов. С этими аспектами и связано название статьи:

"Базовая структурная семантика параллельных алгоритмов".

5) По ряду причин попутно кратко затрагивается структурная семантика простейших переключаемых структур типа "если-то", "если-то-иначе".

### Технические параллельные алгоритмы управления

Для вычислительных задач представленные выше алгоритмы задаются (интерпретируются) системой численных функций.

Однако такие структуры могут быть интерпретированы как алгоритмы управления, заданные системой функций (исполнения) любой другой природы. Это может быть, в частности, управление движением объектов, обработкой потоков информационного или материального типа и т.п. Например (Рис. 1.1):

технологическая операция параллельной обработки детали (потока деталей), представленная тремя технологическими переходами [3]:

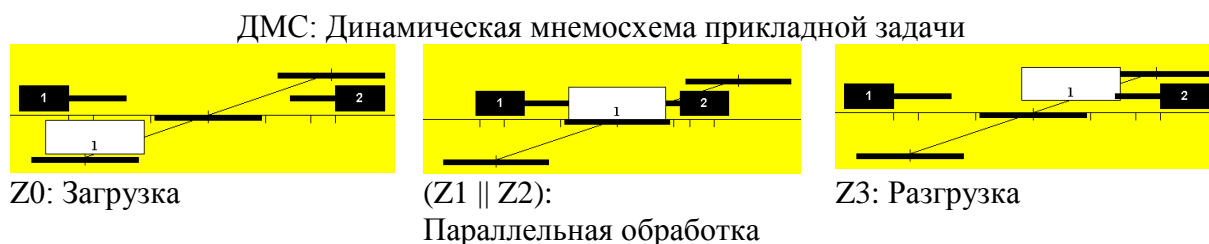


Рис. 1.1. Двухсторонняя параллельная обработка корпусной детали

В данном случае схема материального потока (материальных) данных и структурные формулы вариантов алгоритмов последовательной и параллельной обработки (материальных) данных (рабочих областей детали) также совпадают с численным примером.

## 2 СТРУКТУРНАЯ СЕМАНТИКА СТРУКТУРНЫХ ФОРМУЛ И СХЕМ

### 2.1 Семантика структурных формул алгоритмов

Семантика (смысл) записи последовательных алгоритмов типа  $A1 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3$  очевидна:

одно действие выполняется за другим (одна команда выполняется за другой).

Для параллельного алгоритма  $A3 = Z0 \rightarrow (Z1 \parallel Z2) \rightarrow Z3$  условная запись параллелизма (с наглядным геометрическим знаком параллелизма  $\parallel$ ) требует уточнения порядка его выполнения.

### 2.2 Последовательные алгоритмы

#### Структурные формулы и схемы последовательных алгоритмов

Первые две представленные ранее структурные формулы алгоритмов – это *логические схемы алгоритмов (ЛСА)* [2] для последовательных алгоритмов линейного типа. По своему существу ЛСА – это особые структурные формулы, которые хорошо ассоциируются с графическими структурными схемами.

Далее более подробно рассматривается структура простых последовательных алгоритмов линейного и нелинейного типа (с условными ветвлениями – переключательные структуры алгоритмов).

#### Пример 2.1. Линейный алгоритм

Далее (Табл. 2.1) приводится следующая структурная формула потока управления. Представлены разные варианты записи (элементы прагматики – для сопоставления и принятия разных решений).

Табл. 2.1. Структурная формула линейного алгоритма

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
ЛСА: Логическая схема алгоритма	// Особая формула
$A1 = Z_0Z_1Z_2Z_3$	
СФА: Структурная формула алгоритма (СФА = ЛСА)	
Times: Компактный шрифт с переменным шагом типа Times	
$A1 = Z_0Z_1Z_2Z_3 = Z0Z1Z2Z3 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3 = Z0-Z1-Z2-Z3$	
Courier: Шрифт с постоянным шагом типа Courier	
$A1 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3 = Z0-Z1-Z2-Z3 =$ $= -Z0-Z1-Z2-Z3 \rightarrow$	
<b><math>A1 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3 = Z0 \rightarrow Z1 \rightarrow Z2 \rightarrow Z3 = Z0-Z1-Z2-Z3 =</math></b> <b><math>= -Z0-Z1-Z2-Z3 \rightarrow</math></b>	

Форма записи типа  $A1 = Z_0Z_1Z_2Z_3$  – это ЛСА линейного алгоритма, представляющая собой структурную формулу с неявной *последовательной связкой*  $\rightarrow = -\rightarrow = -$  (которая представляет структурную операцию *секвенции* – в более поздней терминологии).

Этой формуле прямо соответствует структурная схема алгоритма (Рис. 2.1): используются мозаичные стили набора схем (псевдографика).

ЛТА: Литерный текст алгоритма		//ЛЗФ: Литерная знаковая форма	
ССА: Структурная схема алгоритма / ШСА: Штрих-схема алгоритма (СФА = ШСА)			
ПГ: Псевдографика / ЛПГ: Литерная псевдографика (ПГ = ЛПГ) / Courier			
КПГ: Клавиатурная псевдографика		СПГ: Специальная псевдографика	
A1: z0 z1 z2 z3 -----+---+---+---+--->		A1: z0 z1 z2 z3 _____         _____>     // - +	
<b>A1: z0 z1 z2 z3</b> -----+---+---+---+--->		<b>A1: z0 z1 z2 z3</b> _____         _____>	

Рис. 2.1. Схемная псевдографика линейного алгоритма

### Пример 2.2. Алгоритм с условным ветвлением типа "если-то"

Далее (Табл. 2.2) показана ЛСА простого нелинейного последовательного алгоритма, которая интерпретируется как структурная формула алгоритма (СФА = ЛСА), где  $\uparrow^1, \downarrow^1$  – вертикальные (условно поперечные) связи нелинейной передачи управления (порядком выполнения действий).

Табл. 2.2. Структурная формула алгоритма с условным ветвлением типа "если-то"

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
ЛСА: Логическая схема алгоритма / Times	
$A4 = Z_0 p_1 \uparrow^1 Z_1 \downarrow^1 Z_2 Z_3$	
СФА: Структурная формула алгоритма (СФА = ЛСА) / Times	
$A4 = Z_0 p_1 \uparrow^1 Z_1 \downarrow^1 Z_2 Z_3 =$ $= Z_0 \rightarrow p_1 \uparrow^1 \rightarrow Z_1 \rightarrow \downarrow^1 \rightarrow Z_2 \rightarrow Z_3 = Z_0 \rightarrow p_1 \uparrow^1 \rightarrow Z_1 \rightarrow \downarrow^1 \rightarrow Z_2 \rightarrow Z_3$	
СФА: Структурная формула алгоритма (СФА = ЛСА) / Courier	
// Вводятся элементы псевдографики – связь нелинейной передачи управления	
$A4 = Z_0 \rightarrow p_1 \uparrow^1 \rightarrow Z_1 \rightarrow \downarrow^1 \rightarrow Z_2 \rightarrow Z_3 = -Z_0 - p_1 \overline{\uparrow^1 - Z_1 - \downarrow^1} - Z_2 - Z_3 \rightarrow$	
<b><math>A4 = Z_0 \rightarrow p_1 \uparrow^1 \rightarrow Z_1 \rightarrow \downarrow^1 \rightarrow Z_2 \rightarrow Z_3 = -Z_0 - p_1 \overline{\uparrow^1 - Z_1 - \downarrow^1} - Z_2 - Z_3 \rightarrow</math></b>	

Работа алгоритма поясняется далее. Предварительно приводится соответствующая структурная схема алгоритма в псевдографике (Рис. 2.2) – также в двух разных формах псевдографики.

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
ССА: Структурная схема алгоритма / ШСА: Штрих-схема алгоритма (СФА = ШСА)	
ПГ: Псевдографика / ЛПГ: Литерная псевдографика (ПГ = ЛПГ) / Courier	
КПГ: Клавиатурная псевдографика	СПГ: Специальная псевдографика
<pre> +-----+ A4: Z0 ↑1 Z1 ↓1 Z2 Z3 -----+---x---+---o---+---+---&gt;           p1 </pre>	<pre>            -----             -----  A4: Z0 ↑1 Z1 ↓1 Z2 Z3 -----+---x---+---o---+---+---&gt;           p1 </pre>
<pre> +-----+ <b>A4: Z0 ↑1 Z1 ↓1 Z2 Z3</b> -----+---<b>x</b>---+---o---+---+---&gt;           <b>p1</b> </pre>	<pre>            -----             -----  <b>A4: Z0 ↑1 Z1 ↓1 Z2 Z3</b> -----+---<b>x</b>---+---o---+---+---&gt;           <b>p1</b> </pre>

Рис. 2.2. Схемная псевдографика алгоритма с условным ветвлением типа "если-то"

Формула и схема задают структуру последовательного алгоритма **с условным переходом** по логическому условию ЛУ (предикату)  $p_1 = 0, 1$ :

- если ЛУ  $p_1 = 1$  (ЛУ является истинным), то после команды  $Z_0$  выполняется следующая после переключателя  $p_1$  команда  $Z_1$ , затем выполняется  $Z_2$  и затем  $Z_3$ :  
 $A4 = Z_0 Z_1 Z_2 Z_3$ .
- если ЛУ  $p_1 = 0$  (ЛУ является ложным), то после  $Z_0$  выполняется переход по стрелке  $\uparrow^1 \dots \downarrow^1$  (в обход следующей команды  $Z_1$ ) к  $Z_2$ , и затем выполняется  $Z_3$ :  
 $A4 = Z_0 Z_2 Z_3$



Задается простейшая **переключательная структура алгоритма**, которая соответствует синтаксической конструкции программирования типа "**если-то**" (**if-then**).

В данном случае получается достаточно приемлемое наглядное соответствие структурной формулы и схемы алгоритма (с переключательной "физикой") и их интерпретации синтаксической структурой типа "если-то". При этом:

- последнее выражение структурной формулы наглядно напоминает некоторую структурную схему в литерной псевдографике:

это простейшая **схемоподобная структурная формула** потока управления нелинейного последовательного алгоритма;

- данный мелкий факт имеет далеко идущие концептуальные последствия:

эта структурная формула интерпретируется как простейшая **двухмерная форма** (ДМФ) структурной формулы алгоритма (СФА/ДМФ) – его потока управления.

**Пример 2.3. Алгоритм с условным ветвлением типа "если-то-иначе"**

Далее (Табл. 2.3) ЛСА и структурная формула алгоритма с условным ветвлением типа "если-то-иначе",

где  $\omega \equiv 1$  – тождественно ложное ЛУ, превращающее условный переход по стрелке-метке 2 в безусловный переход (типа goto).

Если ЛУ  $p1 = 1$ , то после команды Z0 выполняется команда Z1 и затем команда Z3:

$$A5 = Z_0 Z_1 Z_3.$$

Если ЛУ  $p1 = 0$ , то после Z0 выполняется Z2 и затем Z3:

$$A5 = Z_0 Z_2 Z_3.$$

Это одномерная форма структурной формулы (СФА/ОМФ). При этом в последнем случае введены элементы двумерной (клавиатурной) псевдографики, что улучшает отображение нелинейных связей передачи управления.

Табл. 2.3. Структурная формула и схема алгоритма с ветвлением типа "если-то-иначе"

ЛСА: Логическая схема алгоритма / Times	
$A5 = Z_0 p_1 \uparrow^1 Z_1 \omega \uparrow^2 \downarrow^1 Z_2 \downarrow^2 Z_3$	
СФА: Структурная формула алгоритма (СФА = ЛСА) / ОМФ: Одномерная форма Times: Шрифт с переменным шагом типа Times	
$A5 = Z_0 p_1 \uparrow^1 Z_1 \omega \uparrow^2 \downarrow^1 Z_2 \downarrow^2 Z_3 =$ $= Z_0 \rightarrow p_1 \uparrow^1 \rightarrow Z_1 \rightarrow \omega \uparrow^2 \rightarrow \downarrow^1 \rightarrow Z_2 \rightarrow \downarrow^2 \rightarrow Z_3 = -Z_0 - p_1 \uparrow^1 - Z_1 - \uparrow^2 \downarrow^1 - Z_2 - \downarrow^2 V - Z_3 \rightarrow$	
Courier: Шрифт с постоянным шагом типа Courier // Замена $\omega = \mathbf{w}$	
<pre>           +-----+         +----- ---+        A5 = Z0-p1↑1-Z1-w↑2 ↓1-Z2-V↓2-Z3     </pre>	
ССА: Структурная схема алгоритма ШСА: Штрих-схема алгоритма (СФА = ШСА) / ОМФ: Одномерная форма ПГ: Псевдографика / ЛПГ: Литерная псевдографика (ПГ = ЛПГ) / Courier	
КПГ: Клавиатурная псевдографика	СПГ: Специальная псевдографика
<pre>           +-----+         +----- ---+        A5: Z0 ↑1 Z1 ↑2 ↓1 Z2 ↓2 Z3 -----+--x--+--x  o--+--o--+--&gt;         p1      ω      V     </pre>	<pre>           +-----+         +----- ---+        A5: Z0 ↑1 Z1 ↑2 ↓1 Z2 ↓2 Z3 -----+--x--+--x  o--+--o--+--&gt;         p1      ω      V     </pre>

На этой основе строится (Табл. 2.4) двухмерная форма структурной формулы (СФА/ДМФ) и соответствующая структурная схема алгоритма – в двух формах псевдографики (и в разных вариантах исполнения).

**Табл. 2.4.** Двухмерная структурная формула и схема структуры типа "если-то-иначе"

ЛТА: Литерный текст алгоритма //ЛЗФ: Литерная знаковая форма	
СФА: Структурная формула алгоритма / ДМФ: Двухмерная форма	
$A5 = \begin{array}{c} +---Z2--+ \\ -Z0-p1\uparrow 1-Z1-\downarrow 2V-Z3-> \end{array}$	$A5 = \begin{array}{c} -Z0-p1-Z1-V--Z3-> \\ \downarrow 1 \quad \uparrow 2 \\ +---Z2--+ \end{array}$
ССА: Структурная схема алгоритма ШСА: Штрих-схема алгоритма (СФА = ШСА) / ДМФ: Одномерная форма ПГ: Псевдографика / ЛПГ: Литерная псевдографика (ПГ = ЛПГ) / Courier	
КПГ: Клавиатурная псевдографика	СПГ: Специальная псевдографика
$A5: \begin{array}{c} Z0 \ p1 \ Z1 \ V \ Z3 \\ - - - - + - - - x - - - + - - - o - - - + - - - > \\ \downarrow 1 \ Z2 \ \uparrow 2 \\ + - - - + - - - + \end{array}$	$A5: \begin{array}{c} Z0 \ p1 \ Z1 \ V \ Z3 \\ \text{--- ---x--- ---o--- --->} \\ \downarrow 1 \ Z2 \ \uparrow 2 \\ \text{└───┬───┘} \\   \end{array}$

Как уже указано, эта стрелочная структурная формула типа ЛСА задает переключающую (альтернативную) структуру алгоритма типа "если-то-иначе" (if-then-else).

Можно задавать структуры разных циклов:  
с обратными связями передачи управления, с предусловием или постусловием и т.п.

## 2.3 Параллельные алгоритмы

### Структурные формулы и схемы параллельных алгоритмов

Но основе последовательных ЛСА были предложены параллельные ЛСА (ПЛСА) [2]. Далее (Табл. 2.5) представлена ПЛСА для приведенного ранее примера структурной формулы для параллельного алгоритма  $A_3 = Z_0 - (Z_1 \parallel Z_2) - Z_3$  (модификация записи),

где  $R \uparrow^1 \uparrow^2$  ( $R \uparrow^1 \uparrow^2$ ) – **оператор распараллеливания** (в терминологии работы [2]); **оператор разделения (дивергенции) потоков** – в более поздней терминологии [];

$\downarrow^3 \downarrow^4 S$  ( $\downarrow^3 \downarrow^4 S$ ) – **оператор соединения** (в терминологии работы [2]); **оператор соединения (конвергенции) потоков** – в более поздней терминологии [].

Там же приводится соответствующая одномерная структурная формула алгоритма (СФА/ОМФ), а также одномерная структурная схема алгоритма типа штрих-схемы (ССА = ШСА/ОМФ) в клавиатурной псевдографике (ПГ = КПГ).

Табл. 2.5. Структурная формула и схема алгоритма с ветвлением типа "если-то-иначе"

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
ЛСА: Логическая схема алгоритма / Times +Courier	
$A_3 = Z_0 R \uparrow^1 \uparrow^2 \downarrow^1 Z_1 \uparrow^3 \downarrow^2 Z_1 \uparrow^4 \downarrow^3 \downarrow^4 S Z_3$ $A_3 = Z_0 R \uparrow^1 \uparrow^2 \downarrow^1 Z_1 \uparrow^3 \downarrow^2 Z_1 \uparrow^4 \downarrow^3 \downarrow^4 S Z_3$	
СФА: Структурная формула алгоритма (СФА = ЛСА) / ОМФ: Одномерная форма Times: Шрифт с переменным шагом типа Times	
$A_3 = Z_0 R \uparrow^1 \uparrow^2 \downarrow^1 Z_1 \uparrow^3 \downarrow^2 Z_1 \uparrow^4 \downarrow^3 \downarrow^4 S Z_3 = Z_0 \rightarrow R \uparrow^1 \uparrow^2 \downarrow^1 Z_1 \uparrow^3 \downarrow^2 Z_1 \uparrow^4 \downarrow^3 \downarrow^4 S \rightarrow Z_3$ $A_3 = Z_0 R \uparrow^1 \uparrow^2 \downarrow^1 Z_1 \uparrow^3 \downarrow^2 Z_1 \uparrow^4 \downarrow^3 \downarrow^4 S Z_3 =$	
Courier: Шрифт с постоянным шагом типа Courier	
<pre> +-----+ +- ---+   +--- ----- ---+   A3 = Z0-R↑1↑2 ↓1-Z1-↑3 ↓2-Z2-↑4 ↓3↓4S-Z3 </pre>	

На этой основе получается (Рис. 2.3) эквивалентная структурная схема типа штрих-схемы со спрямлением цепей потока управления.

ЛТА: Литерный текст алгоритма	//ЛЗФ: Литерная знаковая форма
ССА: Структурная схема алгоритма / ШСА: Штрих-схема алгоритма (СФА = ШСА)	
ДМФ: Двухмерная форма	
ПГ: Псевдографика / ЛПГ: Литерная псевдографика (ПГ = ЛПГ) / Courier	
КПГ: Клавиатурная псевдографика	СПГ: Специальная псевдографика
<p>A3: Z0 R Z1 S Z3          -----+---o---+---o---+---&gt;                            ↓2 Z2 ↑4                            o---+---o</p>	<p>A3: Z0 R Z1 S Z3          ----- o--- o--- o---&gt;                            ↓2 Z2 ↑4                            o--- o</p>
<p>A3: Z0 R Z1 S Z3          -----+---+---+---+---+---&gt;                               Z2                               +---+---+</p>	<p>A3: Z0 R Z1 S Z3          ----- ----- ----- -----&gt;                                                                                          Z2                                       +-----+-----+</p>

Рис. 2.3. Схемная псевдографика параллельного алгоритма

### Роль операторов распараллеливания (дивергенции) и соединения (конвергенции)

В работе [2] операторы R и S называются фиктивными: в используемой концепции ЛСА они только *символизируют переход* к выполнению параллельных ветвей и конец выполнения параллельных ветвей, но *не задают функций обработки данных*.

Действительно, это вспомогательные структурные элементы потока управления, которые не задают функции обработки данных в потоке данных.

Однако, всякие структурные элементы выполняют некоторые функции. В данном случае это структурные элементы в составе потока управления, которые выполняют некоторые функции управления в некоторой логике алгоритмических механизмов управления. При этом:

- в первом приближении такая логика представлена некоторой *логикой здравого смысла* в отношении параллельного во времени выполнения (двух в данном случае) потоков управления:

она интуитивно, так или иначе, усваивается по мере практической работы с параллельными алгоритмами и их реализациями (программной, аппаратной, организационной реализацией), но это не гарантирует точного понимания выполнения параллельных процессов и взаимопонимания представителей разных категорий субъектов;

- появляется задача существенного концептуального, теоретического и практического значения:

выявление природы указанных служебных операторов R, S как структурных операторов потока управления (порядком выполнения действий).

### Уточнения смысла операторов распараллеливания и соединения

В ряде работ, например [4]: *Система алгоритмических алгебр* (ССА) для подобных параллельных структур используются бесстрелочные формулы алгоритмов типа:

$$A3 = Z_0(Z_1 \& Z_2)Z_3.$$

Знак конъюнкции & означает условие завершения выполнения параллельных ветвей Z1, Z2 алгоритма по логической функции И (And) следующего типа:

выполняется команда Z3, если выполнена команда Z1 и (&) выполнена команда Z2.

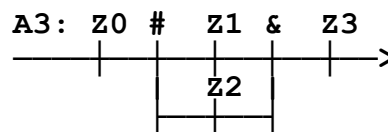
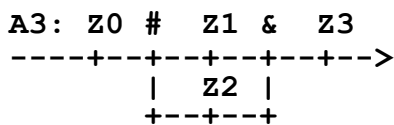
Это проясняется работу оператора соединения параллельных ветвей, который в ПЛСА ассоциируется с операцией конъюнкцией  $S = \&$ :

Однако пока еще не ясно, как общая идея оператора соединения  $S$  реализуется конкретно – это действительно функция конъюнкции или некоторая алгоритмическая условность (некоторая метафора)?

Наличие оператора распараллеливания  $R$  в формулах САА явно не отражается.

В различных системах графического структурного описания параллельных процессов (например в стандарте IDEF3 и т.п.) и их привязке к сетям Петри присутствуют парные элементы типа И-разветвление и И-слияние, визуально различимые, с интуитивно понятной логикой их работы и табличной формой описания процессов, но также без аналитического функционального представления (посредством формул).

Для узла распараллеливания принимаем интерпретацию множественного повторителя с обозначением  $R = \#$ . Соответствующая САА / ШСА:



В словесном выражении работа структуры определяется так:

- после завершения выполнения команды  $Z0$  сигнал передачи управления подается на общий вход оператора распараллеливания  $\#$ , размножается на нем и одновременно запускается выполнение команд  $Z1, Z2$ ;
- далее команды  $Z1, Z2$  выполняются независимо, имеют в общем случае разную длительность во времени и заканчиваются в разное время;
- после того, как обе команды  $Z1$  и ( $\&$ )  $Z2$ , завершатся, сигнал передачи управления выдается с общего выхода узла соединения (по функции  $\&$ ) и далее запускает команду  $Z3$ .

Такая словесная формулировка далее используется для точного описания подобного базового механизма параллелизма. При этом:

в целом структурная конструкция параллелизма  $\parallel$  представляет **парную структурную операцию**  $\parallel = \#\&$ , которая именуется далее как замкнутая (на входе) **параллельная конъюнкция**.

Такая интерпретация существенного продвигается задачу анализа простейших, но базовых по значению механизмов параллелизма. Однако при этом появляется другая ситуация:

если есть параллельная конъюнкция, то теоретически и практически напрашивается некоторая **параллельная дизъюнкция**  $\parallel = \#\vee$ :

это проблемный, но актуальный вопрос, требующий особого анализа [3] – он принимается к сведению на последующие этапы анализа.

### Точные структурные формулы параллелизма (примеры) исходного примера

Анализ вопроса выявляет следующую типовую картину с минимальным параллелизмом – на две параллельные ветви (на примере A3):

1) Используются парная структурная операция  $RS$  параллельного соединения двух ветвей алгоритма (между крайними последовательными участками – для более ясной общей картины).

2) Составляющие операции:

$R = \#$ : оператор распараллеливания или узел дивергенции (разделения) потоков команд передачи управления – множественный повторитель;

$S$ : оператор соединения или узел конвергенции (соединения) потоков.

Возможны варианты реализации (с последующими уточнениями):

а)  $S = \&$ : оператор реализует аналог операции конъюнкции:

структура в целом реализует (замкнутую) параллельную конъюнкцию  $\parallel = \#\&$ ;

$$A31 = Z_0(Z_1 \parallel Z_2)Z_3 = Z_0 \rightarrow (Z_1 \parallel Z_2) \rightarrow Z_3 = Z_0 \rightarrow (Z_1 \# \& Z_2) \rightarrow Z_3$$

б)  $S = V$ : оператор реализует аналоги операции дизъюнкции: структура в целом реализует (замкнутую) параллельную дизъюнкцию  $\parallel = \#V$ .

$$A32 = Z_0(Z_1 \parallel Z_2)Z_3 = Z_0 \rightarrow (Z_1 \parallel Z_2) \rightarrow Z_3 = Z_0 \rightarrow (Z_1 \#V | Z_2) \rightarrow Z_3$$

### Синтаксические формы записи структурной формулы

Вводятся разные синтаксические формы записи двухместных (и многоместных) операций (относительного их операндов)

1) Общий вид – две формы представления парной операции RS:

а) ИнФ: **Инфиксная форма** записи всех операций

$$A3 = Z_0 - (Z_1 RS Z_2) - Z_3 =$$

б) ИнПрПоФ: **Инфиксно-префиксно-постфиксная** форма – комбинированная синтаксическая форма:

$$A3 = Z_0 - (Z_1 RS Z_2) - Z_3 = Z_0 - R(Z_1, Z_3)S - Z_3$$

2) Соответственно этому определяются синтаксические формы для частных видов реализации структурных операций:

а) (замкнутая на входе) параллельная конъюнкция:

$$A31 = Z_0 - (Z_1 \# \& Z_2) - Z_3 = Z_0 - \#(Z_1, Z_2) \& - Z_3$$

б) (замкнутая на входе) параллельная дизъюнкция:

$$A32 = Z_0 - (Z_1 \#V Z_2) - Z_3 = Z_0 - \#(Z_1, Z_2)V - Z_3$$

Указанный принцип структурных уточнений позволяет решать существенно более точные задачи структурного анализа, структурных построений параллельных алгоритмов в классе базовых структур [1, 3]:

точное отображение структурных формул на структурные схемы алгоритмов;

построение псевдокодов алгоритмов, ориентированных на разные синтаксические классы и лексические группы языков программирования и т.п.

Комбинированная форма (ИнПрПоФ) СФА со знаками операций за скобками визуально хорошо согласуются со ССА (Рис. 2.4).

### Уточнение двумерной структурной формулы алгоритма

На основе исходной (одномерной) комбинированной синтаксической формы структурной формулы строится двумерная структурная формула алгоритма (СФА/ДМФ):

$$A3 = Z_0 - R(Z_1, Z_2)S - Z_3 = Z_0 - R(Z_1)S - Z_3 = -Z_0 - R | -Z_1 - | S - Z_3 - > \\ \quad \quad \quad \lrcorner \quad \quad \quad (Z_2) \quad \quad \quad | -Z_2 - |$$

Очевидным является повышение степени соответствия структурной формулы и структурной схемы алгоритма.

Подобные структурные формулы параллельных алгоритмов являются шаблонами для построения структурных схем алгоритмов (Рис. 2.4):

- для мозаичной техники исполнения – непосредственно;
- для линейной техники исполнения (векторной графики), формируемой на основе мозаичной техники.

## Мозаичная и векторная структурная графика

Далее (Рис. 2.4) представлены ССА в разной технике исполнения, построенные на основе СФА как шаблонов разными (учебными) средствами программной поддержки.

ССА: Структурная схема алгоритм / КУ: Канал (поток) управления	
Параллельная конъюнкция: #&	Параллельная дизъюнкция: #V
ШСА: Штрих-схема алгоритма (ССА = ШСА)	
ПГ: Псевдографика / ГПГ: Геометрическая псевдографика (ПГ = ГПГ) // Мозаичный набор	
Явная сетка наборного поля	Неявная сетка наборного поля
ВГ: Векторная графика – линейная техника схемных построений	
БСА: Блок-схема алгоритма	
ШСА: Штрих-схема алгоритма	

Рис. 2.4. Виды и стили исполнения структурной схемы алгоритма

На приведенных примерах демонстрируется возможность полиморфного описания параллельных структур алгоритмов в двух сопряженных формах – структурные формулы и схемы на уровне:

- их изоморфизма (взаимного структурного соответствия);
- полного пространственного соответствия в одномерной и двухмерной форме записи (так называемое прямое структурное соответствие – совпадения с точностью до замены обозначений).

### Значение одномерной и двухмерной формы структурных формул

Исходные одномерные формы структурных формул параллельных (и последовательных) алгоритмов в своей точной структурной форме:

- точно отражают первичную структурную логику параллелизма;
- обеспечивают построение структурных схем, представленных в одномерной и двухмерной форме записи.

Кроме того, вводятся двухмерные структурные формулы параллельных (и последовательных) алгоритмов, которые:

- строго выводятся из одномерных форм структурных формул;
- могут представлять собой шаблоны для построения структурных схем:



обеспечивают задачи строгой формализации и автоматизации графических структурных построений по заданным структурным формулам.

При этом предполагается двухмерный язык структурных формул с двухмерными синтаксисом, который полностью сводится к первичному синтаксису одномерных форм структурных формул [1].

### 3 ФУНКЦИОНАЛЬНАЯ СЕМАНТИКА ПРАЛЛЕЛЬНЫХ СТРУКТУР

#### Явная функциональная семантика

Идея полиморфного синтаксиса двухполюсных структур параллельных алгоритмов очевидна из представленных примеров с обычными обобщениями. Формальный полиморфный синтаксис двухполюсных структур параллельных алгоритмов представлен в работе [1].

Предшествующие выше выкладки отражают разные элементы многоаспектной семантики (различных смыслов) записи параллельных алгоритмов и, в частности, общую идею функциональной семантики.

Уточнение функциональной семантики выполняется в рамках концепции виртуальной аппаратной реализации алгоритмов (Рис. 3.1):

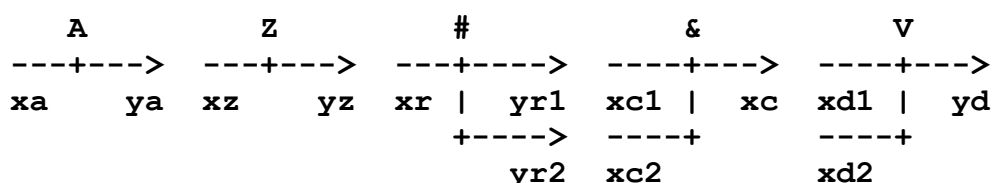


Рис. 3.1. Структурные элементы аппаратной реализации параллелизма

#### Функции структурных элементов

Функции выходов блоков алгоритмов A и команд Z:

$$ya = A(xa) = Axa = D^{ma}xa \quad yz = Z(xz) = Zxz = D^{mz}xz$$

где  $D^{ma}$ ,  $D^{mz}$  – операции задержки сигналов передачи управления;  
 $ma$ ,  $mz$  – длительности задержки.

Оператор распараллеливания  $R = \#$  – множественный повторитель:

$$yr1 = yr2 = xr$$

Узел сборки  $S = \&$  или  $S = V$ :

- для потенциальных цепей передачи управления – это обычный конъюнктор  $yc = xc1 \& xc2$  или дизъюнктор  $yd = xd1 \vee xd2$ ;
- для импульсных цепей передачи управления – это более сложные элементы, но также на основе конъюнктора и дизъюнктора.

Таким образом, простые конъюнктивные и дизъюнктивные представления пригодны только для потенциальных моделей, которые:

- имеют ограниченное практическое применение;
- обеспечивают простые, но строгие базисные математические модели.

Для реальных алгоритмов более адекватными являются импульсные модели, более сложные в математическом описании, но с концептуальной опорой на первичные потенциальные модели.

Указанные первичные средства функциональной интерпретации обеспечивают возможности:

- строго моделирования поведения параллельных алгоритмических структур с выводом временных диаграмм исполнения алгоритмов [1,3];
- построения первичной алгебры структурных преобразований;
- логико-временной интерпретации в метрической логике времени;
- являются базой обобщений на все прочие виды структур алгоритмов.

Применение аппарата квантификации обеспечивает возможности формирования более развитой и выразительной алгебры и логико-временной интерпретации в топологической логике времени.

## ЛИТЕРАТУРА

1. Житников А.П., Житников В.П., Шерыхалина Н.М. [Программно-методический комплекс "Параллельные алгоритмы и программы"](#) / Материалы научно-технического совещания "Высокопроизводительные вычислительные ресурсы России: состояние и перспективы развития". – Уфа: УГАТУ, 2003. С. 151 - 161.
2. Лазарев В. Г., Пийль Е. И. Синтез управляющих автоматов. – М.: Энергоатомиздат, 1989. 328 с.
3. Житников А.П. Параллельные алгоритмы технологических мехатронных систем. / 2-я Всероссийская науч.-тех. конф. "Мехатроника, автоматизация, управление: Сб. трудов. Том 2. – Уфа: УГАТУ, 2005. С. 155 – 160.  
Ссылка на статью будет представлена позднее.
4. Ющенко Е.Л., Цейтлин Г.Е., Грицай В.П., Терзян Т. К. Многоуровневое структурное проектирование программ: Теоретические основы, инструментарий. – М.: Финансы и статистика, 1989. 208 с.