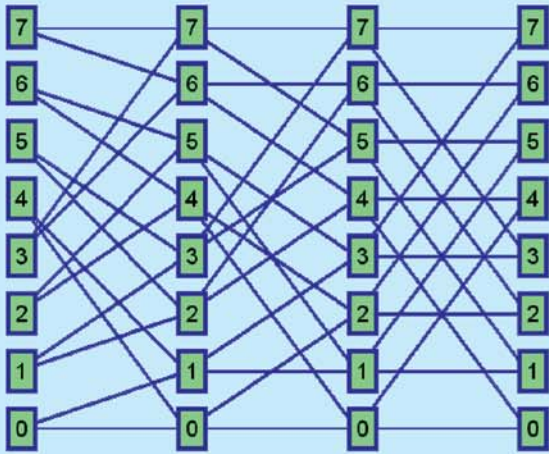
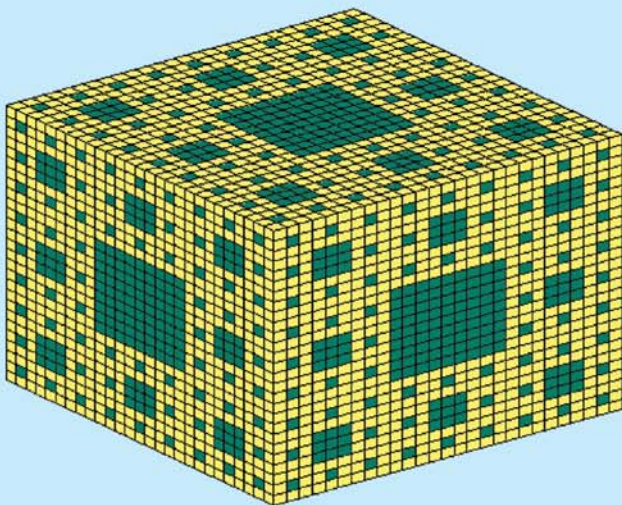
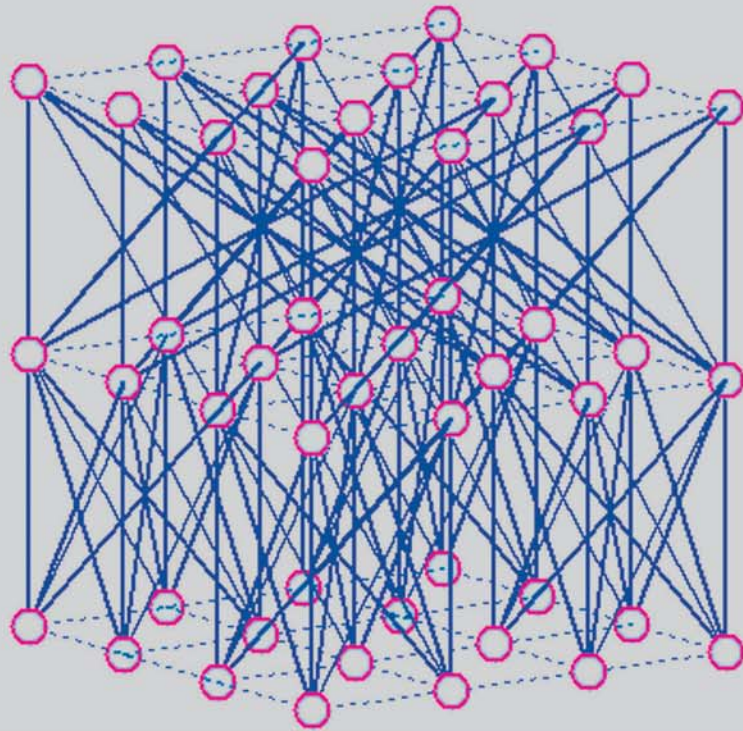


А. Ю. ДОРОГОВ



# Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей



Электронный аналог печатного издания: Дорогов, А. Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб. : «Политехника», 2014. 328 с.

УДК 681.518.54:572.983

ББК 3 818-02

Д71

**Дорогов А. Ю.**

Д71 Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб.: «Политехника», 2014. 328 с.

ISBN 978-5-7325-1036-2

В монографии дается современное изложение теории проектирования алгоритмов быстрых преобразований и модульных нейронных сетей. С системных позиций рассматриваются быстрые преобразования и многослойные нейронные сети с прореженным набором связей и модульной организацией. Предложены алгоритмы структурного проектирования и обучения перестраиваемых быстрых преобразований и нейронных сетей. Исследованы потенциально достижимые свойства быстрых преобразований по пластичности, разделяющей мощности и обобщающей способности. Все рассмотренные алгоритмы сопровождаются примерами и программными реализациями на языке MatLab. Изложение ориентировано на практическое использование в задачах сжатия сигналов высокой размерности, классификации и распознавания образов, спектрального анализа и фрактальной фильтрации. Кроме специалистов соответствующего профиля книга может быть полезна для аспирантов и студентов магистерского уровня подготовки. Тексты программ размещены на сайте <http://www.dorogov.su>.

УДК 681.518.54:572.983

ББК 3 818-02

Рецензенты: д-р техн. наук, проф. кафедры МО ЭВМ Г. А. Кухарев (СПбГЭТУ «ЛЭТИ»); д-р физ.-мат. наук, гл. науч. сотр. Главной астрономической обсерватории РАН Н. Г. Макаренко.

ISBN 978-5-7325-1036-2

© Дорогов А. Ю., 2014

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
<b>Часть I. ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ БЫСТРЫХ ПРЕОБРАЗОВАНИЙ</b> .....	11
<b>1. СТРУКТУРНОЕ И ТОПОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БЫСТРЫХ ПРЕОБРАЗОВАНИЙ</b> .....	12
1.1. Структурный анализ алгоритмов БПФ .....	12
1.2. Стратификация модельных представлений быстрых преобразований .....	18
1.3. Лингвистические модели алгоритмов быстрых преобразований .....	19
1.3.1. Формальный язык регулярных сетей .....	20
1.3.2. Семантическая интерпретация канонических предложений .....	23
1.4. Нейросетевая интерпретация быстрых перестраиваемых преобразований .....	26
1.5. Регулярные топологии быстрых алгоритмов .....	28
1.6. Графические образы регулярных топологий .....	30
1.7. Построение регулярных топологий .....	34
1.7.1. Порождающие схемы .....	34
1.7.2. Программная реализация генератора регулярных топологий ...	38
1.7.3. Топологическая траектория быстрого преобразования .....	43
1.7.4. Аналитическая форма топологических матриц .....	44
1.8. Построение топологических реализаций общего вида .....	46
1.9. Алгоритм реализации быстрого перестраиваемого преобразования ...	47
1.10. Вычислительная эффективность быстрых перестраиваемых преобразований .....	49
<b>2. ПЕРЕСТРАИВАЕМЫЕ ЛИНЕЙНЫЕ ПРЕОБРАЗОВАНИЯ В ЗАДАЧАХ ФУНКЦИОНАЛЬНОЙ ДИАГНОСТИКИ И КЛАССИФИКАЦИИ СИГНАЛОВ</b> .....	51
2.1. Метрические методы диагностирования и классификации сигналов .....	52
2.2. Оптимальные линейные преобразования .....	53
2.3. Методы построения оптимальных ортогональных преобразований ...	57
2.4. Обучение быстрых перестраиваемых преобразований для задач распознавания образов и классификации сигналов .....	60
<b>3. НАСТРОЙКА ПЕРЕСТРАИВАЕМЫХ ПРЕОБРАЗОВАНИЙ К ТИПОВЫМ СИСТЕМАМ ОРТОГОНАЛЬНЫХ ФУНКЦИЙ</b> .....	68
3.1. Мультипликативное разложение элементов матрицы быстрого преобразования .....	68
3.2. Настройка на базис Адамара .....	69

3.3. Настройка на базис Виленкина–Крестенсона.....	73
3.4. Настройка на базис Фурье с естественным упорядочением функций по частотам следования .....	75
3.5. Настройка на быстрое вейвлет-преобразование.....	82
4. ФРАКТАЛЬНЫЕ СВОЙСТВА БЫСТРЫХ ПРЕОБРАЗОВАНИЙ.....	90
4.1. Инварианты самоподобия.....	90
4.2. Системные инварианты слабосвязанных сетей.....	91
4.3. Аналитическая форма регулярного фрактала.....	93
4.4. Дискретная аппроксимация фракталов и самоподобных объектов.....	95
4.4.1. Ковер Серпинского.....	95
4.4.2. Фрактал «Салфетка Серпинского» .....	99
4.4.3. Самоподобный объект Лебега.....	101
4.4.4. Ковер Лебега .....	103
4.5. Системы итерирующих функций для детерминированного квазифрактала .....	105
5. ОДНОМЕРНЫЕ БЫСТРЫЕ ПРИСПОСОБЛЕННЫЕ ПРЕОБРАЗОВАНИЯ .....	107
5.1. Фрактальная фильтрация сигналов.....	107
5.2. Программная реализация фрактальных фильтров .....	110
5.3. Приспособленное преобразование в арифметическом пространстве.....	112
5.3.1. Мультипликативная декомпозиция функций.....	112
5.3.2. Цензурирование нулей.....	113
5.3.3. Настройка приспособленного преобразования .....	115
5.4. Приспособленные преобразования в минимаксном пространстве....	121
5.5. Трансформирующее приспособленное преобразование .....	124
5.6. Спектральные приспособленные преобразования .....	129
5.6.1. Достаточные условия приспособленности быстрого ортогонального преобразования .....	130
5.6.2. Генерация приспособленных ортогональных ядер.....	131
5.6.3. Алгоритм построения приспособленного ортогонального преобразования .....	134
5.7. Быстрые ортогональные трансформирующие преобразования.....	137
5.8. Градиентное обучение трансформирующих перестраиваемых преобразований .....	139
5.9. Градиентное обучение спектральных перестраиваемых преобразований .....	145
5.10. Быстрые перестраиваемые преобразования в квантовых вычислениях .....	147

6. ДВУМЕРНЫЕ БЫСТРЫЕ ПРИСПОСОБЛЕННЫЕ ПРЕОБРАЗОВАНИЯ .....	152
6.1. Двумерные быстрые преобразования .....	153
6.2. Мультипликативная декомпозиция элементов матрицы быстрого преобразования .....	157
6.3. Фрактальная фильтрация изображений.....	158
6.4. Настройка приспособленных двумерных преобразований .....	163
6.5. Построение объемных фракталов .....	165
<b>ЧАСТЬ II. СЛАБОСВЯЗАННЫЕ НЕЙРОННЫЕ СЕТИ .....</b>	<b>169</b>
7. МНОГОСЛОЙНЫЕ ЯДЕРНЫЕ НЕЙРОННЫЕ СЕТИ .....	170
7.1. Цели и задачи нейротехнологии .....	170
7.2. Математические модели нейронных сетей.....	172
7.2.1. Модели одиночных нейронов .....	172
7.2.2. Архитектура нейронных сетей прямого распространения .....	174
7.2.3. Многослойные нейронные сети и особенности их применения ...	175
7.3. Многослойные ядерные нейронные сети.....	176
7.3.1. Ранговые матрицы ядерных нейронных сетей .....	178
7.3.2. Построение топологии ядерной сети.....	178
7.3.3. Операторная модель ядерной нейронной сети.....	182
7.3.4. Алгоритм обработки данных в нейронной сети с ядерной организацией .....	183
7.3.5. Обучение ядерных нейронных сетей.....	185
7.3.6. Каскадирование многослойных нейронных сетей.....	188
7.3.7. Программная реализация алгоритма последовательной адаптации ядерных нейронных сетей.....	189
7.3.8. Программная реализация алгоритма обучения с групповой стратегией.....	197
7.3.9. Двойственное функционирование нейронных сетей. Пространства модальных состояний.....	203
7.4. Регулярные ядерные нейронные сети с реконфигурируемой структурой и топологией .....	204
7.4.1. Объектно-ориентированная модель реконфигурируемой регулярной нейронной сети .....	204
7.4.2. Изменение размерности нейронных ядер .....	205
7.4.3. Изменение ранга межъядерных связей .....	206
7.4.4. Изменение числа нейронных слоев .....	209
7.4.5. Объединение и деление нейронных слоев.....	209
7.5. Программная реализация алгоритма проектирования топологий регулярных ядерных сетей .....	211
7.6. Оценка быстродействия ядерных нейронных сетей .....	214

8. МОДУЛЬНЫЕ СЛАБОСВЯЗАННЫЕ НЕЙРОННЫЕ СЕТИ.....	216
8.1. Концепция модульности нейронных сетей.....	216
8.2. Топологические модели модульных нейронных сетей .....	220
8.3. Обучение модульных нейронных сетей прямого распространения .....	224
8.4. Математические модели слабосвязанных нейронных сетей .....	228
8.4.1. Концепция слабой связанности.....	228
8.4.2. Морфология слабосвязанных сетей.....	228
8.4.3. Генезис слабосвязанных сетей.....	232
8.5. Алгоритм структурной генерации слабосвязанной сети.....	235
8.5.1. Наследование связей .....	235
8.5.2. Криэйторы градуированных нейронных сетей .....	236
8.5.3. Криэйторы ядерных нейронных сетей .....	237
8.5.4. Программная реализация алгоритма структурного синтеза ядерной слабосвязанной сети.....	239
8.5.5. Программная реализация алгоритма построения топологии ядерной нейронной сети .....	246
9. ПЛАСТИЧНОСТЬ НЕЙРОННЫХ СЕТЕЙ .....	249
9.1. Системные модели «общего положения» .....	250
9.2. Операторные многообразия нейронных модулей .....	250
9.3. Пластичность модульных нейронных сетей.....	253
9.4. Расчет модальных состояний в слабосвязанных нейронных сетях.....	254
9.5. Графический расчет пластичности слабосвязанных сетей .....	256
9.6. Пластичность регулярных нейронных сетей .....	260
9.7. Пластичность многослойной нейронной сети прямого распространения .....	262
9.8. Пластичность двухслойной модульной сети .....	264
9.9. Топологическая пластичность нейронных сетей .....	266
10. ОЦЕНКИ КАЧЕСТВА НЕЙРОННЫХ СЕТЕЙ.....	271
10.1. Построение оценок качества нейронных сетей.....	271
10.2. Оценка способности нейронной сети к аппроксимации отображений.....	280
10.3. Вычисление операторного ранга модульной нейронной сети.....	284
10.4. Измерение разделяющей мощности нейронных сетей.....	287
10.5. Измерение обобщающей способности нейронных сетей.....	290
СПИСОК ЛИТЕРАТУРЫ.....	294
Основная .....	294
Дополнительная .....	299
ПРИЛОЖЕНИЯ .....	301

# 1. СТРУКТУРНОЕ И ТОПОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БЫСТРЫХ ПРЕОБРАЗОВАНИЙ

Всякая вещь есть форма проявления беспредельного разнообразия.

*Козьма Прутков. Мысли и афоризмы*

## 1.1. Структурный анализ алгоритмов БПФ

Набор алгоритмов, называемых алгоритмами быстрого преобразования Фурье, вошел в практику спектрального анализа, начиная с работ Кули–Тьюки в 60-х гг. XX в. Существует множество различных схем построения алгоритмов БПФ, основанных, как правило, на способе разбиения входного вектора на подвекторы. Следуя [8], рассмотрим в качестве примера схему «с прореживанием по частоте». Для вектора  $X = \{x(u)\}$ ,  $0 \leq u \leq N-1$ , прямое дискретное преобразование Фурье (ДПФ) определяется выражением

$$y(v) = \sum_{u=0}^{N-1} x(u) e^{-j \frac{2\pi}{N} uv},$$

где  $v = 0, 1, 2, \dots, N-1$  – номер гармоники. Величину  $\omega = e^{-j \frac{2\pi}{N}}$  обычно называют поворачивающим множителем. Число комплексных операций умножения при выполнении ДПФ в общем случае равно  $N^2$ . Для алгоритмов БПФ, как правило, выбирают значение  $N = 2^n$ , где  $n$  – целое число. В схеме с прореживанием по частоте вектор  $X = \{x(u)\}$  разбивается на две части, так что

$$y(v) = \sum_{u=0}^{N/2-1} x(u) \omega_N^{uv} + \sum_{u=0}^{N/2-1} x(u + N/2) \omega_N^{(u+N/2)v},$$

где для сокращения записи используется обозначение  $\omega_N^{uv} = e^{-j \frac{2\pi}{N} uv}$ .

Поскольку  $\omega_N^{(u+N/2)v} = e^{-j \frac{2\pi}{N} (u+N/2)v} = e^{-j \frac{2\pi}{N} uv} e^{-j\pi v} = (-1)^v e^{-j \frac{2\pi}{N} uv}$ , то

$$y(v) = \sum_{u=0}^{N/2-1} x(u) \omega_N^{uv} + (-1)^v \sum_{u=0}^{N/2-1} x(u + N/2) \omega_N^{uv}. \quad (1.1)$$

Рассмотрим выражения для четных и нечетных координат вектора  $Y$  в (1.1):

$$y(2v) = \sum_{i=0}^{N/2-1} (x(i) + x(i + N/2)) \omega_{N/2}^{iv}, \quad (1.2)$$

$$y(2v+1) = \sum_{i=0}^{N/2-1} (x(i) - x(i + N/2)) \omega_N^u \omega_{N/2}^{iv}.$$

Этот прием позволяет свести  $N$ -точечное преобразование к двум  $N/2$ -точечным преобразованиям. За счет этого снижается количество вычислительных операций. На рис. 1.1 в графической форме показан пример перехода от восьмиточечного ДПФ к двум четырехточечным ДПФ для размерности  $N = 8$ . Дальнейшее снижение вычислительных затрат достигается аналогичным сведением  $N/2$ -точечного преобразования к двум  $N/4$ -точечным и т. д. вплоть до минимальных двухточечных преобразований. Результат последовательного разложения представляется операторным графом (рис. 1.2 для размерности  $N = 8$ ).

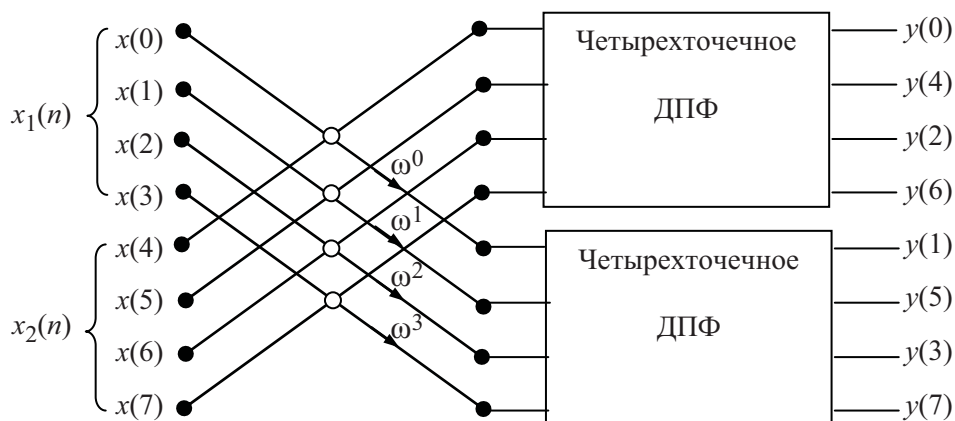


Рис. 1.1. Переход от восьмиточечного ДПФ к двум четырехточечным

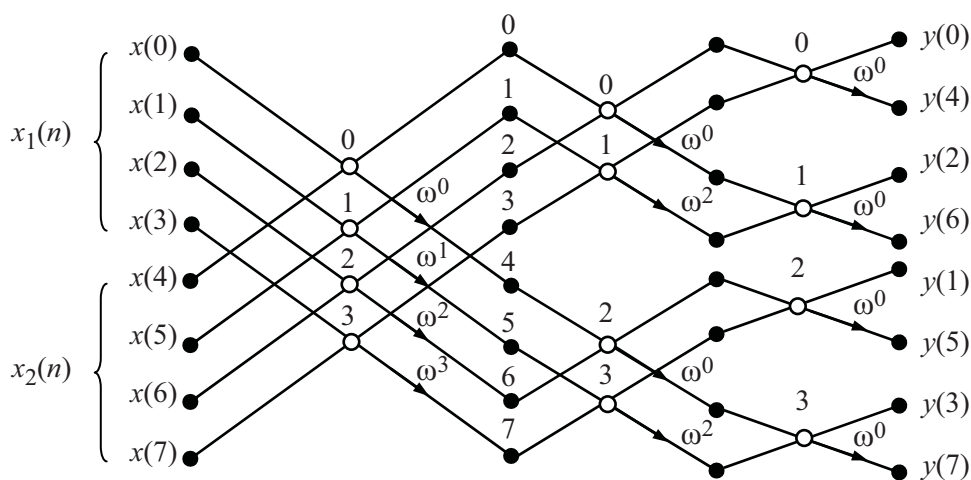


Рис. 1.2. Полный граф восьмиточечного БПФ с «прореживанием по частоте»

Для каждого слоя графа выделены и пронумерованы базовые операции, стрелкой отмечены ветви, в которых знак операнда меняется на противоположный. На рис. 1.3 приведена базовая операция «бабочка» для алгоритма БПФ «с прореживанием по частоте». Базовая операция реализует умножение двухкоординатного вектора-строки на унитарную матрицу, элементы которой зависят от порядковых номеров отсчетов.



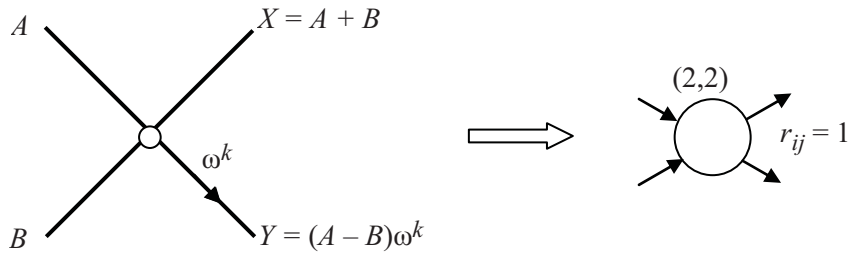


Рис. 1.3. Базовая операция «бабочка»

**Параметрическая модель.** Приведенный на рис. 1.2 граф полностью определяет алгоритм БПФ «с прореживанием по частоте». Однако способ его построения не вполне формализован, скорее он основан на «здоровом смысле» и подтвержден достоверностью получаемых результатов. Для того чтобы показать существующую проблему, дополним этап конструирования алгоритма дополнительным формальным построением. Выражения (1.2) принято трактовать как действие совокупности базовых операций. Обозначим

$$\begin{aligned} y^0(u) &= x^0(u) + x^0(u + N/2), \\ y^0(u + N/2) &= (x^0(u) - x^0(u + N/2))\omega_N^u, \end{aligned} \quad (1.3)$$

где вместо обозначения  $x$  использовано обозначение  $x^0$ . Эти выражения определяют действие базовых операций нулевого слоя на входной вектор  $X$ . Верхний индекс здесь и далее обозначает номер слоя. Для алгоритмов быстрых преобразований принято использовать нумерацию слоев, начиная с нулевого значения. Индекс  $u$  в этих выражениях определяет упорядочение координат векторов и номер базовой операции. Правая часть выражений не вызывает сомнений, поскольку она определяется уравнениями (1.2), но в левой части упорядочение по координатам задано с произволом «здорового смысла» и не является очевидным, хотя типично принимается в алгоритме БПФ. Для того чтобы формализовать правило упорядочения операндов, перейдем к двоичным представлениям индекса. Будем использовать для представления номера координаты двоичную систему счисления [15], определив ее для входного вектора  $X$  и выходного  $Y$  в двух различных формах:

$$\begin{aligned} u &= \langle u_0 u_1 \dots u_{n-2} u_{n-1} \rangle = u_0 2^{n-1} + u_1 2^{n-2} + \dots + u_{n-2} 2 + u_{n-1} - \text{для } X; \\ v &= \langle v_{n-1} v_{n-2} \dots v_1 v_0 \rangle = v_{n-1} 2^{n-1} + v_{n-2} 2^{n-2} + \dots + v_1 2 + v_0 - \text{для } Y. \end{aligned} \quad (1.4)$$

Выражения (1.3) теперь можно переписать в виде одного уравнения:

$$y^0 \langle v_0 u_1 u_2 \dots u_{n-1} \rangle = \sum_{u_0} x^0 \langle u_0 u_1 u_2 \dots u_{n-1} \rangle w_{i^0}^0(u_0, v_0), \quad (1.5)$$

где  $w_{i^0}^0(u_0, v_0)$  – элементы матрицы базовой операции с номером  $i^0$ . В поразрядном представлении номер базовой операции выражается следующим образом:

$$i^0 = \langle u_1 u_2 \dots u_{n-1} \rangle,$$

а сама базовая операция определяется матрицей:

$$W_{i^0}^0 = \begin{bmatrix} 1 & \omega_N^{i^0} \\ 1 & -\omega_N^{i^0} \end{bmatrix}.$$

Отметим, что поразрядное представление позволило зафиксировать правило выбора упорядочения координат правой части выражения (1.5) за счет использования управляющего разряда  $v_0$ . Выход нулевого слоя совпадает с входом первого слоя, поэтому для межслойной передачи можно записать:

$$x^1 \langle v_0 u_1 u_2 \dots u_{n-1} \rangle = y^0 \langle v_0 u_1 u_2 \dots u_{n-1} \rangle.$$

Суммы в правых частях выражений (1.2) представляют собой преобразования Фурье над последовательностями длиной  $N/2$ , а сами последовательности являются выходами нулевого слоя. Выполняя разбиение входных последовательностей слоя 1 и выделяя четную и нечетную составляющую выходных компонент, приходим к следующему выражению для совокупности операций в слое 1:

$$y^1 \langle v_0 v_1 u_2 u_3 \dots u_{n-1} \rangle = \sum_{u_1} x^1 \langle v_0 u_1 u_2 \dots u_{n-1} \rangle w_{i^1}^1(u_1, v_1), \quad (1.6)$$

где номер составляющей базовой операции определяется выражением

$$i^1 = \langle v_0 u_2 \dots u_{n-1} \rangle,$$

а сама базовая операция определяется матрицей

$$W_{i^1}^1 = \begin{bmatrix} 1 & \omega_{N/2}^{i^1} \\ 1 & -\omega_{N/2}^{i^1} \end{bmatrix}.$$

Повторяя разложение подобным образом, для слоя с номером  $\lambda$  получим:

$$y^\lambda \langle v_0 v_1 \dots v_\lambda u_{\lambda+1} u_{\lambda+2} \dots u_{n-1} \rangle = \sum_{u_\lambda} x^\lambda \langle v_0 v_1 \dots v_\lambda u_\lambda u_{\lambda+1} u_{\lambda+2} \dots u_{n-1} \rangle w_{i^\lambda}^\lambda(u_\lambda, v_\lambda), \quad (1.7)$$

$$i^\lambda = \langle v_0 v_1 \dots v_{\lambda-1} u_{\lambda+1} u_{\lambda+2} \dots u_{n-1} \rangle. \quad (1.8)$$

Для последнего слоя имеем:

$$y^{n-1} \langle v_0 v_1 \dots v_{n-1} \rangle = \sum_{u_{n-1}} x^{n-1} \langle v_0 v_1 \dots v_{n-1} u_{n-1} \rangle w_{i^{n-1}}^{n-1}(u_{n-1}, v_{n-1}), \quad (1.9)$$

$$i^{n-1} = \langle v_0 v_1 \dots v_{n-1} \rangle.$$

Отметим, что порядок нумерации координат выходного вектора двоично-инверсен первоначально заданному, поэтому требуется дополнительная пере-

становка для получения естественного порядка следования частот. Совокупность выражений (1.7) (для  $\lambda = 0, \dots, n-1$ ) определяет параметрическую модель БПФ.

**Структурная модель.** Структурная модель быстрого алгоритма представляет собой граф, вершинами которого являются базовые операции, а дуги определяют связи между ними. Полученные поразрядные представления номеров базовых операций в слоях позволяют достаточно просто описать и построить структурную модель. Покажем это на примере преобразования для размерности  $N = 8$ . Введем поразрядную нумерацию входного и выходного векторов преобразования Фурье, следуя выражениям (1.4):

$$u = \langle u_0 u_1 u_2 \rangle, \quad v = \langle v_2 v_1 v_0 \rangle.$$

Тогда номера вершин по слоям будут иметь разрядные представления:

$$i^0 = \langle u_1 u_2 \rangle, \quad i^1 = \langle v_0 u_2 \rangle, \quad i^2 = \langle v_0 v_1 \rangle.$$

В табл. 1.1 приведены численные значения для данных выражений.

Таблица 1.1

Поразрядная нумерация базовых операций

Номер вершины	Слой 0	Слой 1	Слой 2
	$\langle u_1 u_2 \rangle$	$\langle v_0 u_2 \rangle$	$\langle v_0 v_1 \rangle$
0	$\langle 00 \rangle$	$\langle 00 \rangle$	$\langle 00 \rangle$
1	$\langle 01 \rangle$	$\langle 01 \rangle$	$\langle 01 \rangle$
2	$\langle 10 \rangle$	$\langle 10 \rangle$	$\langle 10 \rangle$
3	$\langle 11 \rangle$	$\langle 11 \rangle$	$\langle 11 \rangle$

Объединяя между собой вершины смежных слоев, для которых одноименные разрядные переменные в таблице имеют совпадающие значения, получим граф, показанный на рис. 1.4. Вес вершин представлен парой чисел, определяющих размерность базовой операции по входу и выходу. Базовые операции смежных слоев связаны между собой только по одной из координат, поэтому операторный ранг ( $r_{ij}$ ) всех связей равен единице.

Структура межслойных связей может быть описана ранговыми матрицами. Элемент ранговой матрицы равен значению ранга оператора, связывающего базовые операции смежных слоев. Например, для структурной модели, показанной на рис. 1.4, ранговые матрицы будут иметь вид:

$$R_0 = \begin{array}{c|cccc} i^0/i^1 & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 \end{array}, \quad R_1 = \begin{array}{c|cccc} i^1/i^2 & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 1 \\ 3 & 0 & 0 & 1 & 1 \end{array}.$$

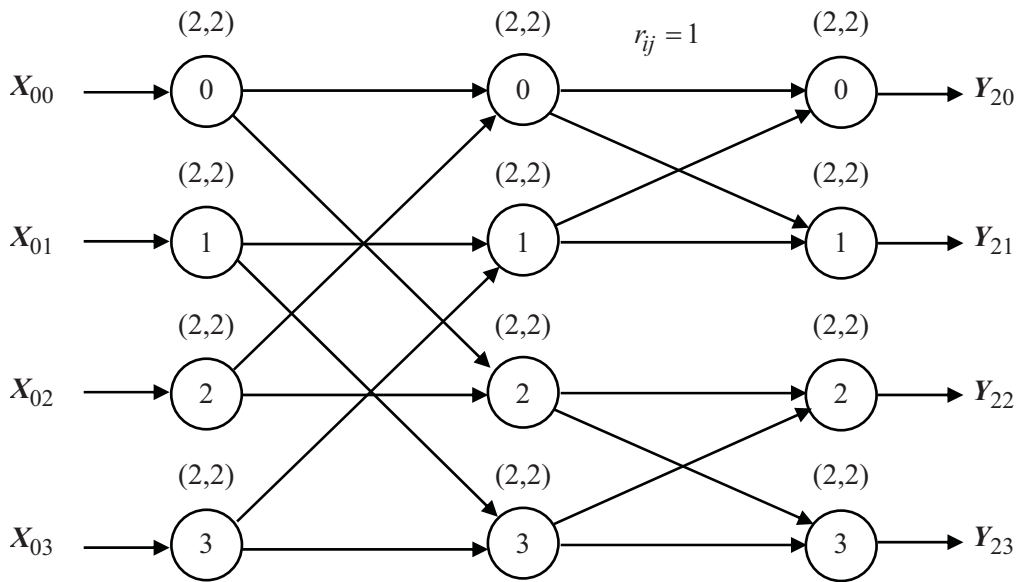


Рис. 1.4. Структурная модель восьмиточечного БПФ

**Топологическая модель.** Входы и выходы базовых операций структурной модели показаны как двухкоординатные векторы без привязки к номеру отсчета. Структурная модель непосредственно не переключается на алгоритм, ее назначение – исследовать общие свойства быстрого алгоритма. Более детальное представление об алгоритме можно получить от топологической модели. В топологической модели элементами рассмотрения являются контакты базовых операций и связи между ними. Базовая операция имеет 2 входных и 2 выходных контакта. Контакты имеют локальную нумерацию, определенную разрядной переменной в пределах базовой нумерации, и глобальную нумерацию в пределах слоя. На рис. 1.2 показана выбранная глобальная нумерация для выходных контактов базовых операций нулевого слоя. Точно такая же нумерация выбрана для выходных контактов других слоев и для контактов входного слоя преобразования. Топологическая модель показывает, как привязаны базовые операции к контактному полю слоя. Для построения топологической модели воспользуемся поразрядным представлением номеров координат векторов  $X$  и  $D_m$  в выражениях (1.5), (1.6), (1.9). В табл. 1.2 приведена поразрядная нумерация контактных полей для БПФ размерности  $N = 8$ .

К базовой операции слоя относятся контакты, которые в смежных столбцах таблицы имеют 2 одноименных разряда с попарно совпадающими значениями. Например, для слоя 0 базовая операция  $i^0 = \langle u_1 u_2 \rangle = \langle 00 \rangle$  имеет контакты по входу и выходу с глобальными номерами 0 и 4. Для слоя 1 базовая операция  $i^1 = \langle v_0 u_2 \rangle = \langle 01 \rangle$  имеет контакты по входу и выходу с глобальными номерами 1 и 3. Нетрудно проверить, что полный граф БПФ «с прореживанием по частоте», показанный на рис. 1.2 полностью отвечает данной схеме построения.

Двоичная нумерация контактных полей

Глобальный номер контакта	$x^0$	$x^1 = y^0$	$x^2 = y^1$	$y^2$
	$\langle u_0 u_1 u_2 \rangle$	$\langle v_0 u_1 u_2 \rangle$	$\langle v_0 v_1 u_2 \rangle$	$\langle v_0 v_1 v_2 \rangle$
0	$\langle 000 \rangle$	$\langle 000 \rangle$	$\langle 000 \rangle$	$\langle 000 \rangle$
1	$\langle 001 \rangle$	$\langle 001 \rangle$	$\langle 001 \rangle$	$\langle 001 \rangle$
2	$\langle 010 \rangle$	$\langle 010 \rangle$	$\langle 010 \rangle$	$\langle 010 \rangle$
3	$\langle 011 \rangle$	$\langle 011 \rangle$	$\langle 011 \rangle$	$\langle 011 \rangle$
4	$\langle 100 \rangle$	$\langle 100 \rangle$	$\langle 100 \rangle$	$\langle 100 \rangle$
5	$\langle 101 \rangle$	$\langle 101 \rangle$	$\langle 101 \rangle$	$\langle 101 \rangle$
6	$\langle 110 \rangle$	$\langle 110 \rangle$	$\langle 110 \rangle$	$\langle 110 \rangle$
7	$\langle 111 \rangle$	$\langle 111 \rangle$	$\langle 111 \rangle$	$\langle 111 \rangle$

Из приведенного выше анализа можно сделать следующие выводы:

- На структурном уровне быстрый алгоритм представляется многослойной модульной сетью, где каждому модулю отвечает базовая операция.
- Размерность быстрого преобразования выражается составным числом, где множителями являются размерности базовых операций.
- Базовые операции определяются квадратными матрицами, которые в пределах слоя имеют одинаковый порядок.
- Ранги связей между базовыми операциями равны единице.
- Топология алгоритма определяется принятым правилом упорядочения выходных операндов в базовых операциях.

## 1.2. Стратификация модельных представлений быстрых преобразований

Математические модели служат средством изучения и проектирования систем. От формальной модели требуется, чтобы она была простой, но достаточной. Основной проблемой математического моделирования является нахождение приемлемого компромисса между детализацией и простотой описания. Один из путей решения этой проблемы заключается в формировании иерархически вложенных семейств моделей, где каждый уровень иерархии соответствует уровню разумного абстрагирования свойств системы, что ведет к упрощению каждой частной модели. Такое многоуровневое представление модели принято называть *стратификацией* [16], а каждый уровень модельного представления – стратой. Стратификация позволяет исследовать систему на разных стадиях познания и описывать каждый уровень адекватными средствами. Стратифицирование можно рассматривать и как средство последовательного углубления представления о системе: при спуске по иерархии страт вниз система раскрывается в деталях; при подъеме на более абстрактные уровни яснее становится смысл и значение всей

системы. В математической формулировке стратификация связана с выделением на каждом уровне иерархии эквивалентных отношений и переходом к факторным моделям, описывающим следующий уровень.

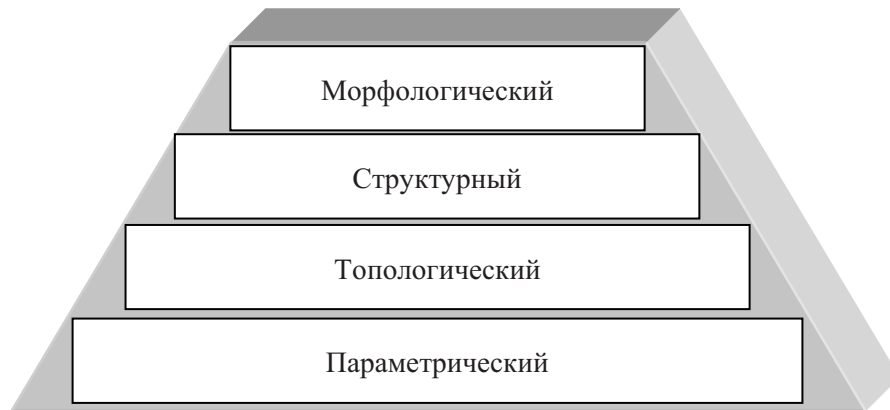


Рис. 1.5. Иерархия уровней стратифицированной модели алгоритма быстрого преобразования

При анализе алгоритма БПФ были построены модели структурного и топологического уровней. В *структурной модели* отсутствует информация об индексах координат вектора данных. Эта модель предназначена для оценки качественных показателей быстрого перестраиваемого преобразования, таких, как быстродействие и пластичность. Привязка векторов к структурной модели может иметь множество вариантов и порождает множество топологических реализаций быстрых преобразований. *Топологическая модель* позволяет выбрать форму алгоритма, удобную для практической реализации. Топологическая модель, дополненная значениями коэффициентов базовых операций, образует модель *параметрического уровня*. На параметрическом уровне реализуются методы обучения и настройки быстрых алгоритмов по заданным показателям качества. Если в структурной модели абстрагироваться от весов вершин и дуг, то приходим к модели *морфологического уровня*. На морфологическом уровне (как это будет показано далее) могут быть описаны системные инварианты быстрых преобразований. Страты модельных представлений упорядочены по степени абстракции (рис. 1.5). Высший уровень абстракции соответствует морфологическому представлению, а низший – параметрическому. Стратификация моделей позволяет разделить исследование алгоритма быстрого преобразования на относительно независимые этапы и использовать для каждого этапа специфичный метод исследования.

### 1.3. Лингвистические модели алгоритмов быстрых преобразований

Подводя итог предварительному анализу можно отметить, что алгоритм БПФ имеет регулярную структуру и регулярную топологию. Под этим понимается, что в пределах слоя веса вершин структурной модели одинаковы, ранги всех связей между вершинами равны единице, организация межслойных связей

структурной и топологической моделей имеет закономерный характер. Подобные сетевые графы можно получить и для других видов спектральных преобразований, общим свойством для них является условие регулярности.

В контексте синтеза быстрых перестраиваемых преобразований задача состоит в том, чтобы построить математические модели, описывающие регулярные структуры и топологии, и найти закономерности их построения. В предварительном анализе было показано, что подходящим языком описания моделей могут служить символические представления чисел в позиционной системе счисления. Поэтому представляется целесообразным использовать лингвистические модели для описания структурного и топологического уровней быстрых преобразований. Лингвистические модели основаны на концепции формального языка. В контексте формального языка структура сети рассматривается как семантические интерпретации допустимых предложений.

### 1.3.1. Формальный язык регулярных сетей

Любой формальный язык является средством для выражения инвариантных свойств предметной области знаний, в этом качестве его удобно использовать для описания инвариантов алгоритмов быстрых преобразований. Для построения языка вводится алфавит, определяется грамматика и устанавливается семантика слов и предложений.

**Алфавит языка и грамматика слов.** Алфавитом языка будем считать кортеж неповторяющихся символов  $A = (z_{i_0} z_{i_1} \dots z_{i_k})$ . Символ языка – это буква с нижним индексом. Букву будем называть родовым именем символа. Отношение линейного порядка на алфавитном множестве назовем *канон* языка и обозначим через  $K$ . Канон языка может быть задан таблицей

$$K = \begin{pmatrix} z_{i_0} & z_{i_1} & \dots & z_{i_k} \\ 0 & 1 & \dots & k \end{pmatrix}.$$

Верхняя строка определяет символ алфавита, а нижняя – приписанный ему порядковый номер. Для определенности будем полагать, что собственные позиции алфавита всегда упорядочены по возрастанию слева направо, начиная с нулевой. Таким образом, кортеж  $\langle z_{i_0} z_{i_1} \dots z_{i_k} \rangle$  задает одновременно алфавит и его канон. В сокращенной форме алфавит, состоящий из букв с общим родовым именем, далее записывается в виде:  $A = \langle z \rangle_K$ .

Пусть алфавит  $A$  состоит из  $k$  символов. Допустимым словом языка будем считать любой кортеж длиной  $n \leq k$ , состоящий из неповторяющихся символов алфавита. Для сокращенного обозначения кортежа слова, все элементы которого имеют общее родовое имя (в данном случае –  $z$ ), будем использовать форму:

$\langle z \rangle_J$ , где  $J$  – отношение линейного порядка на множестве символов слова. Если порядок следования символов в слове соответствует канону, то слово будем называть каноническим. Например, для алфавита  $\langle z_0 z_1 z_2 z_3 z_4 \rangle$  слово  $\langle z_1 z_3 z_4 \rangle$  является каноническим, а слово  $\langle z_1 z_4 z_3 \rangle$  таковым не является, поскольку порядок следования символов в кортеже слова не соответствует порядку следования символов в алфавите.

**Семантика слов.** Практическая ценность формального языка определяется его семантической интерпретацией, поэтому построение языка регулярных сетей целесообразно начать с его семантики. Рассмотрим вначале семантику слов.

Пусть  $\langle z_{n-1} z_{n-2} \dots z_0 \rangle$  – некоторое допустимое слово формального языка, состоящее из набора символов. Поставим в соответствие каждому символу  $z_i$  целое положительное число  $p_i$  и будем рассматривать символ как обозначение переменной, принимающей целочисленные значения из интервала  $Z_i = [0, 1, \dots, p_i - 1]$ . Слово языка будем интерпретировать как целочисленную функцию, заданную правилом:

$$\langle z_{n-1} z_{n-2} \dots z_0 \rangle = z_{n-1} p_{n-2} p_{n-3} \dots p_0 + z_{n-2} p_{n-3} p_{n-4} \dots p_0 + \dots + z_1 p_0 + z_0. \quad (1.10)$$

Нетрудно видеть, что функция (1.10) представляет собой правило перехода от позиционного представления числа в многоосновной системе счисления [15] к его количественному значению. В этом контексте переменную  $z_i$  уместно называть разрядом, а константу  $p_i$  – основанием разряда.

**Операция перестановки.** Пусть  $Q$  есть полная группа перестановок (см. приложение, п. 1) на кортеже слова  $\langle z \rangle_K$  и  $q \in Q$  – некоторая перестановка. Операцию перестановки символов для слова  $\langle z \rangle_K$  будем записывать в виде:  $f_0, f_1, \dots, f_{k-1}$ , например:

$$\langle z_1 z_3 z_0 z_6 \rangle * \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix} = \langle z_6 z_0 z_3 z_1 \rangle.$$

Значение числа в таблице подстановки указывает номер позиции символа в слове. Позиции символа нумеруются, начиная с 1, и отсчитываются от крайнего левого символа. На множестве слов языка определим отношение эквивалентности  $\pi$  следующим образом. Два слова  $\langle z \rangle_{J_1}$  и  $\langle z \rangle_{J_2}$  одинаковой длины будем считать эквивалентными, если существует такая перестановка  $q$ , что

$$\langle z \rangle_{J_1} * q = \langle z \rangle_{J_2}.$$

Отношение эквивалентности разбивает множество слов языка на классы. Понятно, что если  $\langle z \rangle_{J_1}$  и  $\langle z \rangle_{J_2}$  – 2 различных канонических слова, то они всегда при-



надлежат разным классам, т. е. канонические слова являются представителями классов эквивалентности. Каждый класс может быть образован всевозможными перестановками символов канонического слова.

**Непосредственно вложенные классы.** Пусть задан алфавит длиной  $k$ . Класс эквивалентности с длиной слов равной  $n$  будем называть непосредственно вложенным в класс с длиной слов  $n + 1$ , если они отличаются только одним образующим символом. Поскольку в допустимых словах символы не повторяются, непосредственно вложенные классы образуют упорядоченные цепочки по подмножествам входящих символов. Пустое слово можно считать началом каждой цепочки. С учетом пустого слова длина каждой цепочки классов не превышает  $k + 1$ . Цепочки классов длиной  $k + 1$  назовем *максимальными*.

**Непосредственно связанные классы.** Далее будем использовать алфавит, состоящий из символов с двумя различными буквами. Каждая буква (например,  $i$  или  $j$ ) определяет родовое имя символа. Определим алфавит  $A$  формального языка как линейно упорядоченный набор символов с двумя видами родовых имен  $\langle i \rangle$  и  $\langle j \rangle$ :

$$A = \langle \langle i \rangle_I \oplus \langle j \rangle_J \rangle_K.$$

Число символов каждого вида равно  $k - 1$ . Допустимым будем считать слово длиной  $k - 1$ , образованное конкатенацией слов-представителей двух максимальных цепочек классов с разными родовыми именами.

Рассмотрим пример. Пусть алфавит задан кортежем  $A = \langle i_1 i_2 i_3 j_1 j_2 j_3 \rangle$ . Следующие цепочки классов являются максимальными в подклассах родовых имен  $\langle i \rangle$  и  $\langle j \rangle$ :

$$[\emptyset \quad i_1 \quad i_1 i_2 \quad i_1 i_2 i_3] \quad [\emptyset \quad j_1 \quad j_1 j_2 \quad j_1 j_2 j_3].$$

Допустимые слова в двухбуквенном алфавите имеют длину, равную трем, и формируют следующее множество:

$$[j_1 j_2 j_3 \quad i_1 j_1 j_2 \quad i_1 i_2 j_1 \quad i_1 i_2 i_3].$$

Как и прежде, будем полагать, что слова являются представителями классов эквивалентности, образованных действием всевозможных перестановок символов в пределах слова. Два класса допустимых слов назовем непосредственно связанными, если образующие их подклассы родовых имен являются непосредственно вложенными. Непосредственно связанные классы образуют дважды упорядоченные цепочки по составу символов. Максимальная длина цепочек равна  $k + 1$ .

**Грамматика предложений.** Допустимым предложением будем считать упорядоченную последовательность слов, образующих максимальную по длине цепочку непосредственно связанных классов. В допустимом предложении представ-

лены все связанные классы цепочки и каждый класс представлен только одним словом. Длина предложения всегда равна  $k$ . Порядок слов в предложении естественным образом устанавливается по одному из двух возможных вариантов упорядочения непосредственно связанных классов. Зафиксируем для определенности следующее правило построения предложения из множества допустимых слов:

- слова в предложении линейно упорядочены по числу символов с родовым именем  $j$ ;
- первое слово предложения выбирается из класса  $(i)_I$  и имеет поэтому нулевое число вхождений символов с родовым именем  $j$ .

Каждая цепочка непосредственно связанных классов порождает множество однотипных предложений. Среди этого множества существует единственное предложение, составленное из канонических слов, которое назовем *каноническим*. Все остальные предложения для данной цепочки могут быть получены произвольной перестановкой символов в словах канонического предложения. Поэтому достаточно рассматривать порождающую грамматику только для канонических предложений. Пусть числа  $\lambda$  и  $\lambda + 1$  определяют порядковые номера двух непосредственно связанных классов в цепочке. Обозначим через  $I_\lambda$  и  $J_\lambda$  канонически упорядоченные подмножества символов каждого вида для класса  $\lambda$ , а через  $i^\lambda = \langle \langle i \rangle_{I_\lambda} \oplus \langle j \rangle_{J_\lambda} \rangle$  – его каноническое слово. Тогда правило вывода для канонических предложений можно сформулировать следующим образом:

$$I_\lambda \supset I_{\lambda+1}, \quad J_\lambda \subset J_{\lambda+1}, \\ I_0 = I, \quad I_{k-1} = \emptyset, \quad J_{k-1} = J, \quad J_0 = \emptyset.$$

Построенное правило не связано ни с размерностью сети, ни с ее топологией, ни со структурными характеристиками вершин – это правило является инвариантом морфологического уровня, который в аксиоматической форме раскрывает внутреннюю сущность быстрых алгоритмов. Нетрудно проверить, что выражение (1.8) для номеров вершин структурной модели БПФ удовлетворяет построенному инварианту.

### 1.3.2. Семантическая интерпретация канонических предложений

Поставим в соответствие каноническому предложению  $n$ -слойный сетевой граф. Каждое слово с порядковым номером  $\lambda$  будем интерпретировать как функциональное правило вычисления номера вершины в слое  $\lambda$ , определенное семантикой (1.10). Вершины двух смежных слоев будем считать связанными дугой, если в поразрядном представлении номеров вершин

$$i^\lambda = \langle \langle i \rangle_{I_\lambda} \oplus \langle j \rangle_{J_\lambda} \rangle, \quad i^{\lambda+1} = \langle \langle i \rangle_{I_{\lambda+1}} \oplus \langle j \rangle_{J_{\lambda+1}} \rangle$$

одноименные разрядные переменные имеют совпадающие значения. Рассмотрим пример построения сетевого графа. Пусть алфавит языка определен кортежем

$$A = \langle i_1 i_2 i_3 j_3 j_2 j_1 \rangle.$$

Нетрудно убедиться, что предложение

$$[\langle i_1 i_2 i_3 \rangle \langle i_1 i_2 j_1 \rangle \langle i_1 j_2 j_1 \rangle \langle j_3 j_2 j_1 \rangle] \quad (1.11)$$

является каноническим. Будем полагать, что основания разрядных переменных заданы таблицей

$$\begin{pmatrix} A \\ p \end{pmatrix} = \begin{pmatrix} i_1 & i_2 & i_3 & j_3 & j_2 & j_1 \\ 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}.$$

В соответствии с выражением (1.10) слова предложения интерпретируются как следующие алгебраические выражения:

$$i^0 = \langle i_1 i_2 i_3 \rangle = 2^2 i_1 + 2i_2 + i_3,$$

$$i^1 = \langle i_1 i_2 j_1 \rangle = 2^2 i_1 + 2i_2 + j_1,$$

$$i^2 = \langle i_1 j_2 j_1 \rangle = 2^2 i_1 + 2j_2 + j_1,$$

$$i^3 = \langle j_3 j_2 j_1 \rangle = 2^2 j_3 + 2j_2 + j_1.$$

Эти формулы используются для вычисления номеров вершин сетевого графа. На рис. 1.6 приведены поразрядные представления номеров вершин для каждого слоя графа и частично показаны принципы образования межслойных связей.

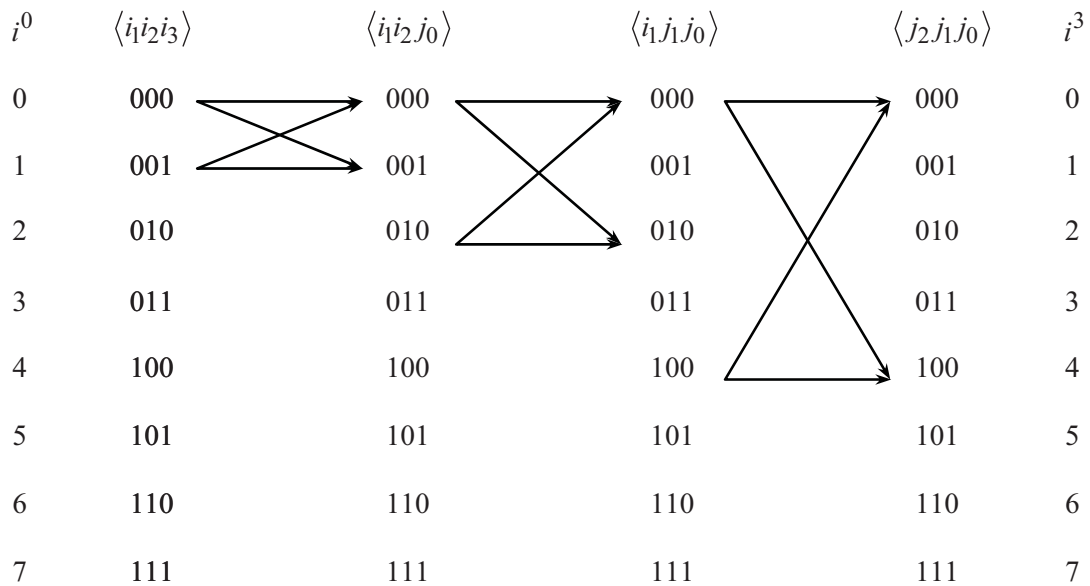


Рис. 1.6. Правило построения графической интерпретации предложения

Фрагмент 1.1 представляет программную реализацию алгоритма построения структурной модели на языке MatLab.

```

function R=Generator_StructureModel(P,G)
%Генератор структурной модели быстрого преобразования
%Функция создает набор межслойных ранговых матриц
% P,G – массивы оснований разрядных переменных с родовыми именами i и j.
% © А.Ю.Дорогов
N=prod(P); % размерность входа
M=prod(G); % размерность выхода
B=Get_sentencel_radix(P,G); %получить основания разрядных переменных
% для слов канонического предложения
D=length(B);% число слов в предложении
for q=1:D-1
    prv=prod(B{q}); %число вершин в предшествующем слое
    nxt=prod(B{q+1}); %число вершин в следующем слое
    R{q}=sparse(prv,nxt); %создание ранговой матрицы
end
for n=1:N %перечисление путей
    [i{1:length(P)}]=ind2sub(P,n); %i- разрядные числа
    for m=1:M %перечисление путей
        [j{1:length(G)}]=ind2sub(G,m); %j- разрядные числа
        Z=Get_Path1(i,j,B); %вершины пути
        for q=1:D-1
            R{q}(Z(q),Z(q+1))=1; %заполнение ранговых матриц
        end
    end
end
Show_Structure_model_graph(R); %графическое отображение структурной
модели
% -----
function Z=Get_Path1(i,j,B)
%Получить путь между вершинами начального и конечного слоев
%i{} – массив ячеек значений разрядных переменных с родовым именем i
%j{} – массив ячеек значений разрядных переменных с родовым именем j
% B – массив ячеек – основания разрядных переменных для слов канонического
% предложения
%Z – массив вершин пути
% © А.Ю.Дорогов
S={ [i(3) i(2) i(1)] [j(1) i(2) i(1)] [j(1) j(2) i(1)] [j(1) j(2) j(3)] };
D=length(S); % число слоев
for k=1:D
    Z(k)=sub2ind([B{k},1],S{k}{1:end}); % номер вершины
end
% -----
function B=Get_sentencel_radix(P,G)
%Получить основания разрядных переменных для слов канонического
предложения
% P,G – массивы оснований разрядных переменных с родовыми именами i и j.
% B – массив ячеек, каждая содержит массив разрядных переменных для
одного % % слова канонического предложения
% © А.Ю.Дорогов
B={ [P(3) P(2) P(1)] [G(1) P(2) P(1)] [G(1) G(2) P(1)] [G(1) G(2) G(3)] };

```

Фрагмент 1.1. Программная реализация генератора структурной модели