

КОНСТРУКЦИЯ БЕСКОНЕЧНОЙ КОНЕЧНО ОПРЕДЕЛЕННОЙ НИЛЬПОЛУГРУППЫ

И. А. Иванов-Погодаев, А. Я. Канель-Белов

АННОТАЦИЯ. Работа посвящена решению проблемы Л. Н. Шеврина и М. В. Сапира. Строится конечно определенная бесконечная нильполугруппа, удовлетворяющая тождеству $x^9 = 0$. Доказательство основано на изучении свойств аperiodических мозаик, аналогов теоремы Гудмана-Штраусса на равномерно-эллиптическом пространстве. Пространство называется *равномерно-эллиптическим*, если любые две точки A и B на расстоянии D соединяются системой геодезических, образующих диск ширины $\lambda \cdot D$ для некоторой глобальной константы $\lambda > 0$.

Исследование поддержано грантом РФФИ N 14-01-00548

СОДЕРЖАНИЕ

1. Введение	4
1.1. Проблемы бернсайдовского типа	4
1.2. Проблемы конечной определенности	5
2. Конечная определенность, связь с информатикой, геометрические методы	6
2.1. Построение конечно определенных объектов и системы конечных автоматов	6
2.2. Схема доказательства. Мозаики	7
2.3. Аperiodические мозаики. “Демо-версия” доказательства	8
2.4. Язык контактов vs язык соотношений. Подклейки.	10
2.5. Последовательная канонизация. Завершение доказательства.	11
2.6. Специфика геометрических методов в полугрупповом случае	11
2.7. Аperiodические замощения в нашей конструкции	12
2.8. Конечно определенные полугруппы	14
2.9. Схема зависимости материала и состав глав	15
3. Геометрическая структура комплекса	16
3.1. Формальное задание полугруппы	16
3.2. Иерархическое построение	17
3.3. Пути на комплексе	21
4. Строение полугруппы путей	29
5. Кодировка вершин и путей на комплексе	30
5.1. Параметр “тип” для вершин	31
5.2. Типы ребер и параметр “уровень” для вершин	32
5.3. Параметр “информация” для узлов	37
5.4. Ядро подклейки и флаг макроплитки	37
5.5. Кодировка путей	38
6. Функции на узлах и структура цепей	39
6.1. Свойства цепей	39
6.2. Указатели	54
6.3. Дополнительные функции	54
7. Оценка количества букв	56

8.	Разбор случаев расположения путей	59
8.1.	Обзор перебора случаев: сочетания типов сторон макроплитки	59
8.2.	Локальное преобразование 1	61
8.3.	Локальное преобразование 2	61
8.4.	Локальное преобразование 3	65
8.5.	Локальное преобразование 4	68
8.6.	Локальное преобразование 5	72
8.7.	Локальное преобразование 6	75
8.8.	Случай цепи $C1$; преобразования 8 и 9	79
8.9.	Случай цепи $C1$; преобразования 7 и 10	82
8.10.	Случай цепи $C2$; преобразования 8 и 9	85
8.11.	Случай цепи $C2$; преобразования 7 и 10	86
8.12.	Случай цепи $C3$; преобразования 7, 8, 9, 10	87
8.13.	Случай цепи $B1$; преобразования 8 и 9	87
8.14.	Случай цепи $B1$; преобразования 7 и 10	88
8.15.	Случай цепи $B2$; преобразования 8 и 9	88
8.16.	Случай цепи $B2$; преобразования 7 и 10	89
8.17.	Случай цепи $A0$; преобразования 8 и 9	90
8.18.	Случай цепи $A0$; преобразования 7 и 10	90
8.19.	Случай цепи $A1$; преобразования 8 и 9	91
8.20.	Случай цепи $A1$; преобразования 7 и 10	92
8.21.	Случай цепи $A2$; преобразования 8 и 9	92
8.22.	Случай цепи $A2$; преобразования 7 и 10	93
8.23.	Случай цепи $UL1$; преобразования 8 и 9	94
8.24.	Случай цепи $UL1$; преобразования 7 и 10	95
8.25.	Случай цепи $UL2$; преобразования 8 и 9	95
8.26.	Случай цепи $UL2$; преобразования 7 и 10	96
8.27.	Случай цепи $UL3$; преобразования 7, 8, 9, 10	96
8.28.	Случай цепи $UR1$; преобразования 8 и 9	97
8.29.	Случай цепи $UR1$; преобразования 7 и 10	98
8.30.	Случай цепи $UR2$; преобразования 8 и 9	99
8.31.	Случай цепи $UR2$; преобразования 7 и 10	100
8.32.	Случай цепи $UR3$; преобразования 7, 8, 9, 10	100
8.33.	Случай цепи $DR1$; преобразования 8 и 9	101
8.34.	Случай цепи $DR1$; преобразования 7 и 10	102
8.35.	Случай цепи $DR2$; преобразования 8 и 9	103
8.36.	Случай цепи $DR2$; преобразования 7 и 10	104
8.37.	Случай цепи $DR3$; преобразования 7, 8, 9, 10	104
8.38.	Случай цепи $DL1$; преобразования 8 и 9	104
8.39.	Случай цепи $DL1$; преобразования 7 и 10	105
8.40.	Случай цепи $DL2$; преобразования 8 и 9	106
8.41.	Случай цепи $DL2$; преобразования 7 и 10	107
8.42.	Случай цепи $DL3$; преобразования 7, 8, 9, 10	108
8.43.	Случай цепи $D1$; преобразования 8 и 9	108
8.44.	Случай цепи $D1$; преобразования 7 и 10	108
8.45.	Случай цепи $D2$; преобразования 8 и 9	109
8.46.	Случай цепи $D2$; преобразования 7 и 10	109
8.47.	Случай цепи $U1$; преобразования 8 и 9	109
8.48.	Случай цепи $U1$; преобразования 7 и 10	110

8.49.	Случай цепи U_2 ; преобразования 8 и 9	110
8.50.	Случай цепи U_2 ; преобразования 7 и 10	110
8.51.	Случай цепи U_3 ; преобразования 7, 8, 9, 10	110
8.52.	Случай цепи L_1 ; преобразования 8 и 9	111
8.53.	Случай цепи L_1 ; преобразования 7 и 10	111
8.54.	Случай цепи L_2 ; преобразования 8 и 9	112
8.55.	Случай цепи L_2 ; преобразования 7 и 10	112
8.56.	Случай цепи L_3 ; преобразования 7, 8, 9, 10	113
8.57.	Случай цепи R_1 ; преобразования 8 и 9	113
8.58.	Случай цепи R_1 ; преобразования 7 и 10	113
8.59.	Случай цепи R_2 ; преобразования 8 и 9	113
8.60.	Случай цепи R_2 ; преобразования 7 и 10	113
8.61.	Случай цепи R_3 ; преобразования 7, 8, 9, 10	114
8.62.	Случай цепи CUL_0 ; преобразования 7, 8, 9, 10	114
8.63.	Случай цепи CUL_1 ; преобразования 7, 8, 9, 10	115
8.64.	Случай цепи CUR_1 ; преобразования 7, 8, 9, 10	115
8.65.	Случай цепи CUR_2 ; преобразования 7, 8, 9, 10	116
8.66.	Случай цепи CDL_1 ; преобразования 7, 8, 9, 10	116
8.67.	Случай цепи CDL_2 ; преобразования 8 и 9	116
8.68.	Случай цепи CDL_2 ; преобразования 7 и 10	117
8.69.	Случай цепи CDL_3 ; преобразования 7 и 10	118
8.70.	Случай цепи CDR_1 ; преобразования 8 и 9	118
8.71.	Случай цепи CDR_1 ; преобразования 7 и 10	118
8.72.	Случай цепи CDR_2 ; преобразования 8 и 9	118
8.73.	Случай цепи CDR_2 ; преобразования 7 и 10	119
8.74.	Подсчет введенных соотношений	119
9.	Локальные преобразования при выходе в подклееную макроплитку	120
9.1.	Обзор случаев	123
9.2.	P1: Макроплитка с окружением (left, top, right, bottom)	124
9.3.	P2: Макроплитка с окружением (left, top, 1A, 3A)	126
9.4.	P3: Макроплитка с окружением (7A, x, 3B, 4A); $x = \text{left, top}$	128
9.5.	P4: Макроплитка с окружением (7A, x, 1A, 3A); $x = \text{left, top}$	129
9.6.	P5: Макроплитка с окружением (top, right, 6A, 2A)	131
9.7.	P6: Макроплитка с окружением (top, right, 1A, 3A)	132
9.8.	P7: Макроплитка с окружением (x, 3B, 6A, 2A); $x = \text{left, top}$	133
9.9.	P8: Макроплитка с окружением (x, 3B, 1A, 3A); $x = \text{left, top}$	135
9.10.	P9: Макроплитка с окружением (x, 1A, 6A, 2A); $x = \text{left, top}$	136
9.11.	P10: Макроплитка с окружением (x, 1A, 1A, 3A); $x = \text{left, top}$	137
9.12.	Оценка числа введенных соотношений	138
10.	Приведение к канонической форме	139
10.1.	Свойства путей, лежащих на одном ребре	140
11.	Приложение. Подсчет числа окружений	152
11.1.	Варианты расположения макроплиток	152
12.	Дальнейшие вопросы и соображения	156
	Список литературы	157

1. ВВЕДЕНИЕ

Работа посвящена построению конечно определенных нильполугрупп. Доказана следующая

Теорема. *Существует конечно определенная бесконечная нильполугруппа, удовлетворяющая тождеству $x^9 = 0$.*

Теории полугрупп посвящено ряд обзоров (см. например, [28, 58]). Построение таких полугрупп затрагивает, с одной стороны, проблемы Бернсайдовского типа, а с другой – проблемы построения конечно определенных объектов. Поэтому следует, хотя бы кратко, упомянуть историю вопроса в этой связи.

1.1. Проблемы бернсайдовского типа. Проблемы Бернсайдовского типа оказали огромное значение в развитие современной алгебры. Эта проблематика охватила большой круг вопросов, как в теории групп, так и в смежных областях, стимулировала алгебраические исследования. Проблемам бернсайдовского типа посвящена обзорная статья Е. И. Зельманова [60].

Важнейшее значение имеет теория групп. Ясно, что группы, удовлетворяющие тождеству $x^2 = 1$ коммутативны. Для групп с тождеством $x^3 = 1$ доказать локальную конечность несколько труднее (и она была доказана самим Бернсайдом [40]). Вопрос о локальной конечности групп с тождеством $x^4 = 1$ стоял открытым чуть меньше 40 лет [24], а с тождеством $x^6 = 1$ – свыше 50 [48].

Получению контрпримеров предшествовало построение бесквадратных слов над трехбуквенным алфавитом, а также бескубных слов над алфавитом их двух букв, т.е. построению бесконечных ниль-полугрупп с тождествами $x^2 = 0$ и $x^3 = 0$. Вопрос о локальной конечности групп с тождеством $x^n = 1$ был решен отрицательно в знаменитых работах П. С. Новикова и С. И. Адяна [19]: было доказано существование для любого нечетного $n > 4381$ бесконечной группы с $m > 1$ образующими, удовлетворяющей тождеству $X^n = 1$. Результаты Новикова-Адяна А. И. Мальцев рассматривал как основное событие алгебры 20 века. Эта оценка была улучшена до $n > 665$ С. И. Адяном [1]. Совсем недавно С. И. Адяном был совершен неожиданный прорыв: эта оценка была улучшена до $n \geq 101$ (см. [2]).

Работы П. С. Новикова и С. И. Адяна оказали огромное влияние на творчество И. А. Рипса, который в дальнейшем разработал метод канонической формы и также построил примеры бесконечных периодических групп. Позднее А. Ю. Ольшанский предложил геометрически наглядный вариант доказательства для нечетных $n > 10^{10}$ [20]. Для всех достаточно больших четных n примеры бесконечных 2-порожденных групп периода n были построены независимо И. Лысенком и С. Ивановым. Подробная библиография и история вопроса – см. [1].

В PI -случае вопросы локальной конечности алгебраических алгебр решаются положительно. В ассоциативном случае соответствующий результат был получен И. Капланским и Д. Левицким. Чисто комбинаторное доказательство для ассоциативного случая получается из теоремы Ширшова о высоте [29], [30]. Теореме Ширшова о высоте посвящена обширная библиография (см., например, работы [33, 50, 5, 6]). Для PI -алгебр Ли соответствующий результат был получен А. И. Кострикиным (для нулевой характеристики, кроме того в характеристике p им вместе с Е. И. Зельмановым показана локальная нильпотентность алгебр Ли с тождеством $y \circ \text{Ad}(x)^n = 0$) и в общем случае Е. И. Зельмановым. Это позволило решить так называемую “ослабленную проблему Бернсайда”, то есть доказать наличие максимальной конечной группы среди k -порожденных групп, удовлетворяющих тождеству $x^n = 1$. Подробная библиография по этому вопросу изложена в монографии

[16]. Первый контрпример к неограниченной проблеме был найден Е. С. Голодом в 1964 году на основе универсальной конструкции Е. С. Голода – И. Р. Шафаревича [8, 9]. Эта конструкция позволила также построить не локально конечные периодические группы неограниченной экспоненты.

1.2. Проблемы конечной определенности. Все имеющиеся примеры бесконечных периодических групп бесконечно порождены. Чрезвычайно глубоким и вдохновляющим является следующий открытый вопрос (входящий список основных алгебраических проблем в теории групп):

Вопрос. *Существует ли конечно определенная бесконечная периодическая группа?*

Известна классическая теорема Хигмана о вложении рекурсивно-определенных групп в конечно определенные.

В работе А. Ю. Ольшанского и М. В. Сапира [55] была построена конечно определенная группа являющаяся расширением конечно порожденной бесконечной периодической группы с помощью циклической.

На проблематику, связанную с построением разного рода экзотических объектов с помощью конечного числа определяющих соотношений обратил внимание В. Н. Латышев. Им же была поставлена проблема существования конечно определенного ниль-кольца [10].

Вопрос (В. Н. Латышев). *Существует ли конечно определенное бесконечномерное нилькольцо?*

В качестве продвижения в решении этого вопроса можно рассматривать результаты Г. П. Кукина, В. Я. Беляева о вложениях рекурсивно определенных объектов в конечно определенные [52, 37]. В. А. Уфнарским был построен пример конечно определенной алгебры промежуточного роста [27]. В работе В. В. Щиголева [31] была изучена связь между понятиями ниль и нильпотентности конечно определённых алгебр в зависимости от количества определяющих соотношений и порождающих. Также построен пример алгебр с малым количеством определяющих соотношений, у которых все слова длины два нильпотентны.

Фундаментальную проблему существования конечно определенной нильполугруппы поставили Л. Н. Шеврин и М. В. Сапир в Свердловской Тетради (3.616) [25], а также вопрос 3.8 в [54].

Вопрос (Л. Н. Шеврин, М. В. Сапир). *Существует ли конечно определенная бесконечная нильполугруппа?*

Авторам представляется, что полугрупповой вопрос предшествует кольцевому, а кольцевой – групповому. Этот вопрос привлекал внимание авторов в течение многих лет. Ряд результатов, таких как конструкция конечно-определенной полугруппы с нецелой размерностью Гельфанда–Кириллова, построение алгебр с конечным базисом Гребнера но неразрешимой проблемой делителей нуля возникли из работы над этой проблемой. В своей кандидатской диссертации [12] один из авторов построил пример конечно определенной полугруппы, содержащий ненильпотентный ниль-идеал.

Теорема 1.1 ([12]). *Существует конечно-определенная полугруппа H , с множеством слов G от образующих, удовлетворяющая следующим свойствам:*

- *Существует ненильпотентный идеал $I = LH$, где L – буква в H ;*
- *Если слово $A \in G$ представляется в виде $A = XYUZ$, где $X, Y, Z \in G$, тогда $LA = 0$.*

Основной результат данной работы заключается в следующем:

Теорема 1.2. *Существует конечно определенная бесконечная нильполугруппа, удовлетворяющая тождеству $x^9 = 0$.*

Благодарности. Авторы признательны руководителям семинара «Теория колец» на кафедре Высшей Алгебры механико-математического факультета МГУ В. Н. Латышеву и А. В. Михалеву за полезные обсуждения и постоянное, в течение ряда лет, внимание к работе. Мы также благодарны И. А. Рипсу, Л. Н. Шеврину, А. Х. Шеню за полезные обсуждения, Ф. Дюранду, Ц. Селле, Л. А. Бокутю, Ю. Чэну, Т. Ферни за поддержку в участии на конференциях, А. С. Малистову за помощь в оформлении статьи.

2. КОНЕЧНАЯ ОПРЕДЕЛЕННОСТЬ, СВЯЗЬ С ИНФОРМАТИКОЙ, ГЕОМЕТРИЧЕСКИЕ МЕТОДЫ

2.1. Построение конечно определенных объектов и системы конечных автоматов. При построении разного рода объектов трудности вызывает контроль над следствиями из вводимых соотношений, в особенности тогда, когда следует доказать, что некое соотношение не является следствием заданных. Зачастую используются три метода контроля над соотношениями:

- (1) Базис Гребнера и бриллиантовая лемма;
- (2) Теория малых сокращений;
- (3) Реализация машины Тьюринга или машины Минского.

В конечно определенном случае вопросы, связанные с построением объектов, обладающих заданными свойствами, сильно усложняются и наибольшее значение приобретает третий метод. При этом буква интерпретируется как состояние конечного автомата, а слово – как цепочка взаимодействующих конечных автоматов. Если число соотношений конечно, то это взаимодействие *локально* и мы получаем связь с задачами самоорганизации, типа задачи Майхилла о стрелках и т.д. На этом пути были решены задача о построении конечно определенных полугрупп с рекурсивной размерностью Гельфанда-Кириллова. Отметим, что подобная техника довольно громоздка для построений, требующих малого роста. Например, не удалось пока построить конечно определенную полугруппу с размерностью Гельфанда-Кириллова равной 2.5.

На этом же пути был получен ответ на известный открытый вопрос, поставленный В. Н. Латышевым – была построена алгебра с неразрешимой проблемой делителей нуля и конечным базисом Гребнера [12]. Отметим, что для автоматных мономиальных алгебр (в частности, конечно определенных) а также *алгебр с ограниченной переработкой* аналогичный вопрос (также поставленный В. Н. Латышевым) решается положительно [34, 4, 14].

Задача о построении конечно определенной бесконечной нильполугруппы имеет интерпретацию в этих терминах. Рассмотрим цепочку локально взаимодействующих конечных автоматов. У них есть цвета корпусов. Если автомат объявляет себя нулем (совершает самоубийство), то вся цепочка погибает. Можно ли добиться того, чтобы преобразования были обратимы (если $u = v$, то $v = u$), при этом существовали сколь угодно длинные живые цепочки, и чтобы любая цепочка, у которой несколько раз подряд повторились цвета корпусов, погибала. Мы задаем локальный закон взаимодействия, а наш враг – задает внутренние состояния автоматов. Требуется достичь адекватного поведения цепочки автоматов.

Хотя решение проблемы построения бесконечной нильполугруппы было достигнуто геометрическими методами, данная интерпретация демонстрирует связь с самоорганизующимися системами и могут быть интересными с точки зрения получения результата в этой области.

2.2. Схема доказательства. Мозаики. Пусть W – бесквадратное слово над конечным алфавитом из трех букв. Если каждое его неподслово (т.е. антислово) объявить нулем, то естественно возникающая полугруппа слов обладает тождеством $x^2 = 0$. Однако естественная конструкция, связанная с заданием множества нулевых слов как подслов слов из некоторого семейства в конечно определенном случае работает плохо.

Невозможно задать конечно определенную ненильпотентную нильполугруппу только *мономимальными* соотношениями, т.е. соотношениями типа $v = 0$. Ибо если есть конечный список запретов и бесконечное слово без запрещенных подслов, то есть и бесконечное *периодическое* слово также без запрещенных подслов.

Что если от одномерных расположений (слов) перейти к двумерным? Известно, что существуют конечные наборы многоугольников (плиток) которыми можно замостить плоскость лишь непериодически. Впервые такой набор был построен Робертом Бергером [38]. В дальнейшем были построены более простые примеры, например, Рафаэлем Робинсоном [56]. Широко известна также мозаика Пенроуза. Итак, имеются контактные правила, для которых:

- Существует замощение всей плоскости, удовлетворяющее запретам.
- Однако таких периодических замощений не существует.

Поэтому если бы можно было бы умножать слева-справа-сверху-снизу, то такого рода объекты можно было бы построить.

Но как придать всему этому смысл? Будем интерпретировать элементы полугруппы как пути на мозаике (дальнейший анализ показывает, что удобно иметь дело с кратчайшими путями – иначе можно много раз проходить один и тот же цикл). Буквы кодируют плитки и переходы между ними. Если локальный не порядок (два символа плитки без символа перехода между ними и т.д.) – то произведение ноль. Кроме того, если локальный участок не располагается на мозаике, или не располагается как участок кратчайшего пути – то он также нулевой.

Если же любую пару узлов, соединяемую кратчайшим путем с кодом s_1 можно соединить кратчайшим путем с кодом s_2 и наоборот, то $s_1 = s_2$.

Итак, пусть есть периодическое слово $U = W^n$. Начинаем добавлять клетки к слову U , локально перебрасывая пути. Оно окружается мозаикой. Поскольку U периодично, то не может быть расположено на нашей мозаике. Поэтому в какой то момент вставлять клетки не получится и мы доберемся до локального участка, несовместимого с мозаикой. Тем самым устанавливается равенство слова $U = W^n$ нулю.

Таким образом, возникает мозаика со своей глобальной структурой, которая и обеспечивает аперриодичность. Локальные правила задают эту самую структуру, и путь, “перекидывание” которого задает область на мозаике. Возникают три языка: геометрический язык, описывающий глобальное поведение комплекса, комбинаторно геометрический язык контактов (локальных правил) и полугрупповой язык соотношений – переброски путей. Для решения задачи слеует научиться переводить с одного языка на другой и, главное, обеспечить саму эту возможность.

Возможность перевода с языка контактов на глобальный язык связана с утверждениями типа теоремы Х. Гудмана-Штраусса [46]. Теорема Х. Гудмана-Штрауса утверждает, что любую плоскую мозаику, связанную с подстановочной системой (плоской DOLL-системой) можно задать локальными правилами. Наш подход, использующий реберную структуру, ближе к подходу Тома Ферни и Николая Бедериде [32].

2.3. Аперриодические мозаики. “Демо-версия” доказательства.

2.3.1. *Аперриодические замощения.* Наличие аперриодических мозаик связано со следующим обстоятельством. Рассмотрим работу машины Тьюринга на клетчатой ленте. Движение головки и обмен сигналами можно перевести на язык укладки плиток на фазовой диаграмме “пространство-время”. Из этого выводится алгоритмическая неразрешимость проблемы дополнения заданного набора плиток до замощения плоскости.

Алгоритмическая неразрешимость проблемы замощения, когда изначально ничего не положено, доказывається технически сложнее. Здесь “пространство” и “время” перемешиваются. Строятся замощения, в некоторых местах которых вынуждается один шаг машины Тьюринга, в других – два шага и т.д. Линии, отвечающие работе машины, образуют некую структуру. Дальнейшее совершенствование и упрощение конструкций приводит к аперриодическим мозаикам, ставшими классическими (плитки Аммана, мозаики Пенроуза и др.).

В 1961 году Хао Вангом [59] были рассмотрены квадратные плитки с разноцветными сторонами. Разные плитки можно прикладывать друг к другу сторонами одного цвета. Был поставлен вопрос, существуют ли конечные наборы таких плиток, с помощью которых могут быть получены только неперриодические замощения плоскости. Первым такой набор был построен Робертом Бергером [38], основная идея состояла в том, что замощения моделировали работу машины Тьюринга. При этом использовалось несколько тысяч плиток. Позднее были придуманы наборы, содержащие небольшое количество плиток. Например, интересна конструкция Рафаэля Робинсона [56]. В этой связи можно также упомянуть мозаики Пенроуза и Аммана, состоящие из многоугольников, которыми, при заданных условиях, можно замостить плоскость лишь неперриодически. Различные примеры аперриодических мозаик есть также в работе [13].

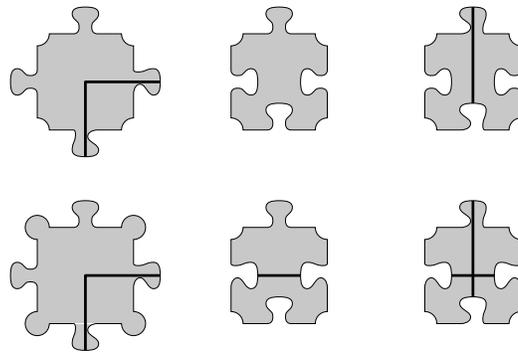


Рис. 1. Набор плиток Робинсона.

2.3.2. *Иерархия и аперриодичность.* Рассмотрим другой способ получения неперриодических замощений. Пусть имеется конечное число типов плиток и мы задаем правила, по которым из нескольких маленьких плиток можно составлять большие макроплитки тех же типов.

Пример. Плитки могут быть квадратами A и B , при этом, чтобы составить квадрат A второго уровня, нужно взять четыре квадрата A , A , A , B . А чтобы составить квадрат B , нужно взять четыре квадрата B A B A .

Таким образом, получается иерархическая система. Каждую плитку можно разбить на требуемое число уровней иерархии. Можно показать, что получаемое замощение будет

непериодично. Аналогичный способ построения используется в подстановочных системах, например, с помощью подстановок $1 \rightarrow 10, 0 \rightarrow 01$ получается бескубное слово

$$1001011001101001 \dots$$

Оказывается, язык граничных условий и язык иерархий схожи. А именно, любую иерархическую систему можно задать конечным числом граничных условий.

Пусть имеется конечное число типов плиток, причем заданы правила иерархии, по которым плитка уровня n составляется из нескольких плиток уровня $n - 1$. Тогда для начального набора плиток первого уровня можно задать конечную систему граничных условий так, чтобы задавалась мозаика, получаемая при иерархическом способе задания.

Иерархичность системы плиток гарантирует непериодичность замощения. Вследствие этого можно задавать с помощью граничных условий мозаики, которые будут с гарантией непериодическими.

2.3.3. *Демо-версия доказательства.* Рассмотрим одну из классических мозаик – знаменитую *мозаику Пенроуза* (см. рисунок 2).

Конструкция мозаики Пенроуза. Используются плитки двух видов – толстый и тонкий ромбы. Есть граничные условия: стороны каждого ромба раскрашены в две пары цветов. Соприкастись два ромба могут только сторонами двух цветов, образующих пару. На рисунке 2 цвета обозначены внешними и внутренними насечками двух типов.

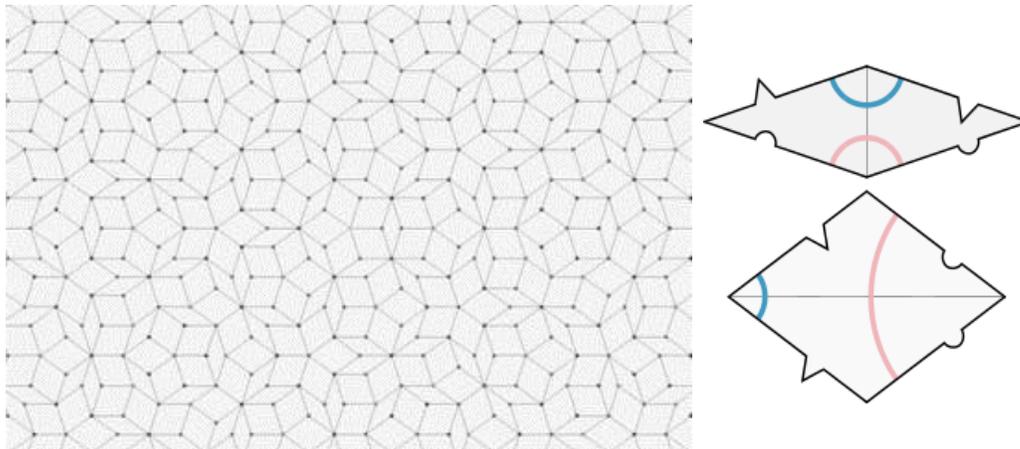


Рис. 2. Мозаика Пенроуза.

Конструкция полугруппы мозаики Пенроуза. Легко видеть, возможно конечное количество типов узлов-вершин где сходятся несколько плиток. Кроме того, ребра в мозаике могут иметь десять возможных направлений. Можно выписать все возможные типы узлов и обозначить их буквами алфавита. Теперь последовательность букв (слово) будет кодировать последовательность узлов, которые мы проходим вдоль пути.

Для каждого узла введем также два параметра: по какому ребру мы в него вошли и по какому ребру мы вышли (включая информацию о цветах ребер). Теперь можно расширить алфавит, добавив буквы для всевозможных сочетаний параметров для разных типов узлов. Некоторые последовательности букв с гарантией не смогут представлять путь на мозаике (например, если ребро, по которому мы пришли в узел не соответствует по цвету ребру, по которому мы вышли из предыдущего узла). Такие последовательности мы будем заносить в список запрещенных.

Некоторые пути на мозаике можно объявить *эквивалентными*: например путь по двум соседним ребрам ромба эквивалентен пути по другой паре сторон. Можно выписать все такие эквивалентности для разных вариантов получающихся узлов и получить полный конечный список. Тогда, с помощью локальных замен, можно переводить одни пути в другие.

Таким образом, с мозаикой Пенроуза можно связать конечно определенную полугруппу, где словам соответствуют пути. Запрещенные пути – это нулевые слова. Можно также запретить пути, которые не являются кратчайшими между двумя узлами. Для этого нужно сначала внести в список запрещенных короткие такие пути. После этого можно показать что с помощью локальных замен любой не кратчайший путь приводится к виду содержащему запрещенный короткий подпуть.

В случае, если бы можно было бы показать, что любой путь, не вкладывающийся в мозаику, приводится к нулю с помощью указанных локальных правил, получающаяся полугруппа была бы нильполугруппой, так как на мозаике не лежит периодических путей.

Почему нужна другая мозаика. Проблема заключается в том, что на мозаике Пенроуза есть пути, которые недостаточно сильно меняются локальными заменами, то есть, недостаточно “извиваются”. Это приводит к тому, что можно сконструировать путь, каждый локальный кусок которого может быть вложен в мозаику, но весь путь не может быть вложен. Локальные замены меняют его незначительно и преобразовать его в достаточной мере, чтобы диагностировать несоответствие мозаике, не получается.

В целом, каждый путь можно трактовать как массив информации об его окрестности. Когда мы производим локальные замены, происходит перенос информации вдоль пути. В случае, если пути можно шевелить незначительно, канал переноса информации будет ограничен, что не позволит выявить ситуацию, когда длинный кусок пути не может являться частью мозаики. (Например, когда путь – это степень разрешенного слова.)

2.4. Язык контактов vs язык соотношений. Подклейки. Итак, если осуществлять нашу программу на базе классических мозаик, то возникают трудности, связанные с тем, что в некоторых направлениях геодезические пути не поддаются изгибу и вокруг них ничего из переброски не наращивается. А именно достаточно протяженный в двух измерениях кусок мозаики обеспечивает вычислительный процесс.

Перевод с языка соотношений (т.е. движения геодезического пути) на язык контактов вещь более сложная и обеспечить его возможность не так просто. Чтобы локальное шевеление геодезической заполнило достаточную область, пространство должно быть *равномерно-эллиптическим*. Пространство называется *равномерно-эллиптическим*, если любые две точки A и B на расстоянии D соединяются системой геодезических, образующих диск ширины $\lambda \cdot D$ для некоторой глобальной константы $\lambda > 0$. (Априори не очевидно, что такое пространство существует.) Такое пространство следует собрать из мозаики и установить аналог теоремы Гудмана-Штраусса для этой сборки.

Равномерно-эллиптическое пространство строится из подстановочной системы указанной на рис. 3 при этом следует озаботиться тем, чтобы степени узлов были бы ограничены, что усложняет конструкцию, вынуждая сделать композицию разных подстановок 5.

Далее, когда путь проходит через узел и по обе стороны от него есть пара плиток более низкой ступени иерархии, сходящиеся в этом узле, необходимы *подклейки*. Бесконечное число иерархических уровней приводит к тому, что к ребру может примыкать неограниченное число подклеек. При этом необходимо избежать следующей ситуации: мы вышли в подклейку и вернулись. Далее продеформировали участок пути в подклейку так, что его “голова” снова коснулась исходного ребра, а вход и выход в “подклейку” не исчезли. Далее

возникший участок по ребру может уйти в иную подклейку там снова изогнуться и коснуться исходного ребра и т.д. Конструкция специально построена таким образом, чтобы этого избежать. То есть *все организовано так, что путь может вернуться из подклейки ТОЛЬКО в исходную точку*. Далее размеры подклеек к плиткам, подклеек к подклейкам и т.д. экспоненциально убывают достаточно быстро. Тем самым, помимо всего прочего, обеспечивается гомотопическая тривиальность комплекса, образованного “подклейками”.

2.5. Последовательная канонизация. Завершение доказательства. Доказательство завершается так. Рассмотрим слово. Надо его либо привести к пути на комплексе (тем самым оно неперiodично), либо к нулю. Оно последовательно приводится к k -каноническому виду и при этом k растет, т.е. оно состоит из участков границ плиток k -го уровня иерархии, кроме начала и конца, целиком содержащихся в плитке уровня $k - 1$ (возможно в подклейке $(k - 1)$ -го уровня) и при этом являющихся $(k - 1)$ -каноническими.

Далее с помощью соотношений (в том числе отвечающих выходу в подклейку) проверяется согласование соседних участков пути, располагаемых на границах плиток k -го уровня, их нахождение в $(k + 1)$ -ом уровне или возможность находиться в двух соседних плитках $(k + 1)$ -го уровня. Отсутствие согласования означает локальную невозможность располагаться преобразованного пути на комплексе – т.е. вывод равенства нулю. Вначале рассматривается плоский процесс. Если имеются команды входа в подклейку и выхода из нее – то проверяется ситуация между ними и производится сокращение.

Согласование означает возможность преобразования к $(k + 1)$ -каноническому виду, после чего процесс повторяется. Он заканчивается канонической формой слова (в нашем случае каноничность не означает однозначности, поскольку можно выбирать разные стороны плиток). Наверное, существует такой выбор – но он более сложен и ситуация похожа на теорию канонической формы элементов в гиперболических (или более обще, в группах с неположительной кривизной) развитую И. А. Рипсом, когда вначале строится предканоническая форма (для данного уровня иерархии), потом осуществляется выбор.

2.6. Специфика геометрических методов в полугрупповом случае. Интерпретация соотношений в полугруппах через “rewriting diagram” похожа на применение диаграмм Ван-Кампена в теории групп. Рассмотрение мозаик также родственно работе с диаграммами Ван-Кампена в теории групп. Использование путей по ребрам в плиточных замощениях и групп связанных с ними, встречается в работах Дж. Конвея [42]. С помощью групповой техники он показал отсутствие различных разбиений. Вот модельный

Пример. Рассмотрим шахматную доску, из которой вырезаны уголки. Тогда ее нельзя разбить на доминошки 1×2 .

Предположим, что такое разбиение существует. Рассмотрим доминошки как клетки в диаграмме Ван-Кампена. Букве a сопоставим вертикальную стрелку, букве b – горизонтальную, обратным буквам – обратные стрелки. Вертикальная доминошка отвечает соотношению $a^2ba^{-2}b^{-1} = e$, а горизонтальная – $b^2ab^{-2}a^{-1} = 1$. Границе квадрата с вырезанными углами отвечает путь $a^7bab^7a^{-7}b^{-7}a^{-1}$. Если сопоставить элементу a транспозицию элементов 1 и 2, а элементу b – транспозицию 2 и 3, то $a^2 = b^2 = e$ и соотношения выполнены, в то же время как $a^7bab^7a^{-7}b^{-7}a^{-1} = (ab)^4 = ab \neq e$ есть трехчленный цикл.

Тем же методом решаются многие другие олимпиадные задачи на раскраску, но в то же время и такие, которые раскраской не делаются (подробности см. [7]).

Тем не менее мозаики и диаграммы Ван-Кампена существенно различаются. Например, свойства величин углов имеют далеко не полное отражение. Аналогия между мозаиками

и группами довольно глубокая, но не вполне формализована. Имеется классический результат об алгоритмической разрешимости проблемы равенства слов в группе с одним соотношением. Имеется и аналогичный вопрос об алгоритмической разрешимости проблемы разбиения плоскости транслятами одной фигуры. Для связанных фигур такая алгоритмическая разрешимость доказана, а в общем случае вопрос остается открытым.

Далее, при разбиении квадрата на домино возникают подквадратики 2×2 , разбитые на пары доминошек. Такую пару можно повернуть на 90° и сделать *флип*. Такими флипами можно от одного разбиения перейти к любому другому. Аналогичный факт верен и разбиения на k -миношки и для многих других разбиениях. Данный факт родственен конечной порождаемости группы π_2 комплекса (с учетом действия фундаментальной группы π_1). Хорошо бы прояснить аналогию.

В полугруппах геометрические идеи работают не совсем так, как в группах, и в нашем случае эффекты “углов” лучше представлены. По всей видимости, полугрупповая теория может пролить свет на мозаики. Более того, есть некая близость теории полугрупп к кольцам как к “квантовой взвеси”, так что и кольцевая теория, если будет построена, даст дополнительное понимание. Возможно, наш подход окажется полезным и для других построений в полугруппах и кольцах.

2.7. Аперриодические замощения в нашей конструкции. Как уже говорилось, существуют конечные наборы многоугольников (плиток) которыми можно замостить плоскость лишь непериодически. Методы построения таких мозаик, как правило, опираются на иерархические правила построения: задаются универсальные правила построения плиток уровня $n + 1$ из плиток уровня n , для нескольких типов плиток A_1, \dots, A_k . Х. Гудман-Штраусс [46] показал, что иерархические мозаики можно получить, задав конечное число локальных правил. Таким образом, локальные условия (конечность набора) могут приводить к глобальному эффекту (непериодичности замощения).

Итак, основной задачей является конструирование мозаики, в которой указанные выше трудности были решены. Для этого она должна обладать несколькими свойствами:

1. **Локальная конечность:** Речь идет конечности возможных типов узлов, конечности изначально задаваемых пар эквивалентных путей, а также конечности списка запрещенных путей.
2. **Возможность “шевеления” любого пути:** Любой достаточно длинный путь, соединяющий узлы A и B может быть переведен локальными заменами в другой путь, отличающийся достаточно сильно от начального. Например, мерой отличия путей может являться максимальное расстояние между соответствующими точками путей.
3. **Аперриодичность:** На мозаике не должно быть путей, отвечающих периодическим словам.

Как уже говорилось, мы используем геометрическую интерпретацию для алгебраических построений. Запрет для двух (или более) плиток находиться рядом друг с другом схож с запретом для двух букв стоять рядом в разрешенном слове. Возникает интерпретация слова как последовательности плиток на выложенной мозаике. Аперриодичность мозаики приводит к непериодичному характеру таких “плиточных слов”. В свою очередь, непериодических замощений можно добиться, если применять иерархический способ построения.

В связи с этим используются языки плиточных примыканий и иерархий. Эти языки во многом схожи, например, заданные правила иерархии, когда плитки A_1, \dots, A_k уровня

$n + 1$ составляются из плиток A_1, \dots, A_k уровня n , можно выразить с помощью конечного множества локальных правил для набора A_1, \dots, A_k . Эти локальные правила будут порождать те же непериодические мозаики что и исходные правила иерархии.

Дальнейшее развитие этих языков приводит к появлению более универсального языка путей на графе. То есть плиточная мозаика рассматривается как граф, где вершины это узлы мозаики, а ребра – границы плиток. В этом смысле понятие плитки можно обобщить, рассматривая их уже как локальные подграфы из которых, с помощью локальных правил, можно составлять граф, покрывающий плоскость.

Аналогом буквы будет тип вершины графа, аналогом слова – путь, проходящий через несколько вершин. Аналогом соотношения будет эквивалентность между путями с общими концами: например, в простом 4-цикле $ABCD$ выполнено соотношение $ABC = ADC$. Помимо таких, есть также мономиальные соотношения, выражающие идею о невозможности существования какого-то пути на мозаике. Также, для обеспечения необходимого контроля над множеством ненулевых слов, вводятся мономиальные соотношения, обнуляющие слова, соответствующие некрайчайшим путям. Оказывается, что при этом можно обойтись конечным числом соотношений. Немонмиальные соотношения, при этом, не меняют длины слова.

Язык путей на мозаике-графе позволяет выразить те же концепции и определить те же мозаики, что и языки иерархических плиток или граничных условий. Таким образом возникает связанная с мозаикой конечно определенная полугруппа с набором свойств, индуцированных мозаикой.

Для построения нильполугруппы используется мозаика сгенерированная с помощью следующего иерархического правила разбиения (рисунок 3).

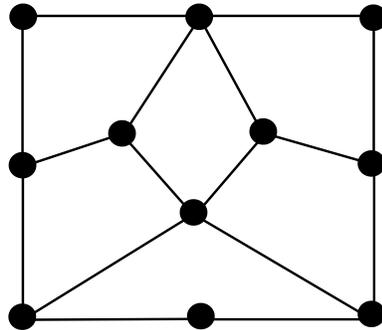


Рис. 3. Правило разбиения.

От мозаики нужно потребовать ряд дополнительных свойств. В частности, любой длинный путь должен допускать возможность “шевеления” то есть, локального преобразований над ним, позволяющих в достаточной мере менять его. Для достижения этого свойства к плоской мозаике производятся “подклейки”, представляющие собой небольшие плоские подграфы, не лежащие в исходной плоскости, и позволяющие обходить “узкие места” на исходном плоском графе. Структура подклеек также имеет иерархическую природу и так же может быть задана на языке преобразований путей конечным образом.

В итоге мы получаем граф, обладающий набором важных для нас свойств, в котором любой путь, соединяющий произвольные точки A и B , является членом семейства геодезических эквивалентных друг другу путей, соединяющих эти точки. Причем эквивалентность двух путей из этого семейства может быть получена путем цепочки локальных преобразований переводящих один путь в другой. “Шевеление” пути играет роль передачи

информации. Фактически, определяющие соотношения задают правила передачи информации по пути. Если задан произвольный длинный путь, мы можем начать работать над ним, совершая локальные преобразования. При этом возможны две альтернативы:

- (1) В результате этой работы можно получить внутри некрайчайший подпуть, либо подпуть указанный в числе запрещенных. В этом случае наш путь представляет нулевое слово.
- (2) В результате этой работы мы восстанавливаем некоторый кусок мозаики, внутри которого лежит пучок геодезических путей, эквивалентных нашему.

Мозаика не может содержать в себе путей, выражаемых периодическим словом. То есть, все периодические слова могут быть приведены к нулю локальными преобразованиями. При этом геодезические пути, лежащие на мозаике не приводятся к нулю, и могут иметь любую длину. Таким образом полугруппа, соответствующая построенному графу-мозаике, будет конечно определенной нильполугруппой.

2.8. Конечно определенные полугруппы. Рассмотрим алфавит Ω , состоящий из конечного числа букв. Слова, составленные из букв образуют полугруппу относительно операции приписывания одного слова к другому. Кроме того, есть специальная буква 0 (ноль), такая что $W0 = 0W = 0$ для любого слова W из полугруппы. Определяющим соотношением в полугруппе считается равенство вида $W_1 = W_2$, где W_1 и W_2 – некоторые слова, причем одно из них может быть нулем (или нулевым словом). В этом случае, соотношение называется *мономиальным*. Для натурального n степенью слова W^n называется слово $WW \cdots W$, где W выписано n раз подряд. Элемент полугруппы (слово) называется *ниль-элементом*, если существует натуральное n , такое что $W^n = 0$. Если все элементы полугруппы являются нильэлементами, то вся полугруппа называется нильполугруппой. Разумеется, наша полугруппа содержит *ноль*.

Мы пользуемся геометрической интерпретацией: *буквам* отвечают вершины специального графа, а *словам* – пути в этом графе. И если слово не может быть представлено путем на графе, оно всегда будет приводиться к нулю.

2.8.1. Плитки и непериодические замощения. Начнем с чистой геометрии. Мы задаем конечный набор граничных условий (как можно прикладывать плитки друг к другу), и через задание локальных условий достигается глобальный эффект. Задание граничных условий схоже с заданием определяющих соотношений в полугруппе, этим объясняется интерес к плиткам и замощениям.

2.8.2. Узлы на мозаике и пути по границам плиток. Помимо языков иерархических систем и граничных условий есть еще один подход. Можно рассматривать мозаики с точки зрения путей по границам плиток. Это естественный шаг для перехода к полугруппе, так как пути по мозаике логично отождествляются со словами в подходящем алфавите.

Назовем *узлами* вершины мозаики, где сходится несколько плиток. Ясно, что в плоской мозаике с конечным набором типов плиток возможно конечное число видов узлов. Обозначим их буквами конечного алфавита. Последовательность букв (слово) соответствует пути на мозаике. Некоторые слова могут вообще не встречаться на мозаике, а другие встречаться в разных местах.

Третий подход к заданию мозаик состоит в определении конечного списка невозможных путей, а также задание конечного числа пар эквивалентных путей. С помощью такого языка можно задавать мозаики также, как и с использованием иерархических систем плиток или граничных условий на плитки.

Мы будем строить такую мозаику в два приема. На первом этапе построим ее *плоскую часть*, которую будем считать базовой. Она будет строиться в виде иерархического бесконечного графа. На втором этапе мы покажем, как к построенному графу производятся *подклейки* – плоские и конечные подграфы, лежащие в другой плоскости, но имеющие с базовым графом несколько общих ребер. В результате получится трехмерный комплекс.

Подклейки нужны для того, чтобы появился маршрут, альтернативный проходу по ребру, к которому производится подклейка. В этом случае, становится возможным перенос информации вдоль пути и мы можем выявить ситуации с невозможным на мозаике путем.

В дальнейших главах мы построим мозаику, где выполнены указанные выше свойства.

2.9. Схема зависимости материала и состав глав. Доказательство основного результата достаточно объемное. В этом параграфе мы кратко опишем состав глав, составляющих работу.

Во **введении** обсуждаются проблемы бернсайдовского типа и конечной определенности и связанные с этим вопросы.

В главе 2 Обсуждается общая стратегия доказательства а также смежные вопросы.

В главе 3 (“Геометрическая структура комплекса”) обсуждаются чисто геометрические свойства комплекса, лежащего в основе построения.

В главе 4 (“Строение полугруппы путей”) и главе 5 (“Кодировка вершин и путей на комплексе”) вводится параметризация, благодаря которой структура построенного геометрического комплекса оказывается связанной со структурой полугруппы. В частности, вводится алфавит букв, кодирующих вершины и ребра на комплексе.

В главе 6 (“Функции на узлах и структура цепей”) обсуждаются особенности комплекса, благодаря которым можно вычислять параметры вершин, по известным другим вершинам, в случае если вершины занимают друг относительно друга определенное положение. В этой главе мы готовим возможности вычисления близлежащих вершин комплекса.

В главе 7 (“Оценка количества букв”) проводится оценка, сколько букв используется в алфавите. Если читателю этот вопрос не так интересен, эту главу можно пропустить без ущерба для дальнейшего изложения.

Глава 8 (“Разбор случаев расположения путей”) является самой объемной в работе. В ней доказывається, что по параметрам трех вершин, лежащих в вершинах некоторой плитки на комплексе, всегда можно вычислить параметры четвертой вершины. Это основное свойство комплекса, на котором основано построение полугруппы. Объем обусловлен большим количеством случаев расположения вершин. При первоначальном прочтении можно ознакомиться с несколькими из них. При рассмотрении локальных преобразований 7, 8, 9, 10 более подробно рассмотрен случай S_1 -цепи, в частности, в явной форме выписана таблица характеристики локального преобразования для S_1 -цепи по виду кода.

В главе 9 (“Локальные преобразования при выходе в подклееную макроплитку”) также разбираются случаи расположения путей, но в этот раз в ситуации, когда происходит выход в подклееную часть (предыдущая глава посвящена плоской ситуации). Аналогично, при первоначальном прочтении можно ознакомиться с несколькими случаями.

Глава 10 (“Приведение к стандартной форме”) завершает доказательство. С помощью свойств комплекса, рассмотренных в предыдущих главах, показывается как можно преобразовать слово. В частности, показывается, что слово, содержащее подслово в девятой степени, приводится к нулю.

Глава 11 (“Приложение. Подсчет числа окружений”) носит технический характер. Выделяются возможные значения четверок-комбинаций типов ребер.

В главе 12 (“Вопросы и перспективы”) обсуждаются смежные вопросы, представляющие интерес в связи с полугрупповой конструкцией.

3. ГЕОМЕТРИЧЕСКАЯ СТРУКТУРА КОМПЛЕКСА

3.1. Формальное задание полугруппы. В этой главе мы обсудим формальную сторону записи полугруппы, то есть какие алфавиты используются для записи слов. Структура алфавита полугруппы сильно зависит от геометрии мозаики (главы 5 и 6).

Алфавит, отвечающий образующим полугруппы (в дальнейшем, просто *алфавит*) будет состоять из букв, кодирующих вершины и букв, кодирующих входящие и выходящие ребра в вершины. Слово из букв кодирует путь – последовательность $X_1e_1e_2X_2e_3e_4X_3\dots$, где X_i – буквы, кодирующие вершины, а e_i – буквы, кодирующие входящие и выходящие ребра.

Для вершины определяется несколько параметров: тип, уровень, окружение, информация, флаг подклейки. В целом эти параметры характеризуют геометрическое положение вершины внутри построенного комплекса. Для каждого параметра мы задаем конечное число значений. Буква, кодирующая вершину, представляет собой сочетание значений всех параметров. В некоторых случаях мы будем рассматривать множество букв, имеющих фиксированное значение некоторых параметров. В частности, определяющие соотношения вводятся для пар слов заданной формы, где мы фиксируем часть параметров для некоторых букв.

Для входящих и выходящих ребер используется свой алфавит, содержащий конечное число букв. Удобно ввести свой набор таких букв для вершины каждого типа, это будет сделано ниже.

В следующих параграфах мы последовательно определим каждый из параметров вершин, вместе с объяснением геометрического смысла. Основная задача состоит в реализации одной из следующих возможностей:

1. По выданному слову установить, что данное слово не кодирует никакой путь (или кодирует запрещенный) и в этом случае привести слово к нулю, применяя определяющие соотношения;

2. По выданному слову установить, что данное слово кодирует некоторый путь на построенном геометрическом комплексе. При этом применение определяющих соотношений будет приводить к преобразованию пути в эквивалентный ему.

Смысл задания параметров для вершин (тип, уровень, окружение, информация, флаг подклейки) состоит в том, чтобы знание набора значений этих параметров для простого пути P_1 из трех вершин и двух ребер между ними позволяло восстановить набор значений этих параметров для другого пути P_2 из трех вершин и двух ребер, имеющего начало и конец такие же как у P_1 . Введение определяющих соотношений представляет собой перебор различных случаев расположения таких пар путей и приравнивание их кодов.

Другая часть определяющих соотношений состоит в обнулении слов, которые либо не представляют никакой путь, либо кодируют путь, который мы хотим сделать нулевым.

После введения таких определяющих соотношений появляется возможность преобразования пути через операцию с его кодом. То есть геометрическому преобразованию пути будет соответствовать преобразование слова в полугруппе. После этого можно будет показать, что слова, содержащие девятую степень любого слова, можно привести к нулю. Кроме того, можно будет убедиться, что на комплексе будут существовать сколь угодно длинные пути (и, соответственно, слова), не приводимые к нулю.

Определим *алфавит типов вершин*, состоящий из символов

$\{A, B, C, U, R, D, L, UL, LU, UR, RU, DL, LD, DR, RD\}$.

Обозначим его как $\{\text{nodes}\}$.

Пусть $\{\text{edges}\}$ – алфавит из 20 символов:

$\{\text{left, top, right, bottom, 1A, 1B, 2A, 2B, \dots, 8A, 8B}\}$.

Будем называть его *алфавит типов ребер*.

Рассмотрим множество упорядоченных четверок (x_1, x_2, x_3, x_4) , где $x_i \in \{\text{edges}\}$. Всего в нем 20^4 элементов. Мы будем использовать только 210 из них. Назовем эти 210 четверок *алфавитом окружений макроплиток* и будем обозначать его как $\{\text{fours}\}$.

Список используемых четверок получается из комбинаторных свойств конструируемой мозаики. Подробно про это написано в Приложении, глава 11, предложения 11.2 и 11.3.

Остальные параметры мы определим в связи с соответствующей геометрической структурой в последующих параграфах.

3.2. Иерархическое построение. Построим *комплекс уровня n* с помощью итерационного процесса. На каждом шаге мы будем иметь дело со графом, состоящим из вершин и ребер, причем простые циклические пути из четырех ребер будем называть *плитками*.

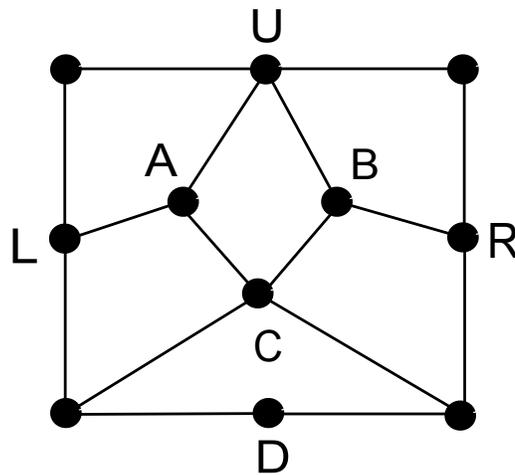


Рис. 4. Комплекс второго уровня.

Комплекс уровня 1 представляет собой простой цикл из четырех вершин, *Комплекс уровня 2* это граф на рисунке 4. Вершины A, B, C будем называть *внутренними* или *черными*, а вершины U, L, R, D – *боковыми*. То есть, на втором этапе у нас имеется шесть плиток. Будем называть их, согласно их положению, левой верхней, левой нижней, правой нижней, правой верхней, средней, нижней.

Будем считать, что вершины A, B, C, U, L, R, D имеют *нулевой уровень*.

С комплексом будут производиться итерации двух типов: *разбиение* и *подклейка*.

Определение 3.1. При операции *Разбиения* каждая плитка комплекса разбивается на шесть плиток согласно правилу на рисунке 5. При этом эта старая плитка перестает быть просто плиткой и становится макроплиткой второго уровня, создаются новые четыре боковые и три внутренние вершины. Одна из четырех возможных ориентаций определяется согласно положению (одному из шести) разбиваемой плитки в ее родительской макроплитке второго уровня (рисунок 6).

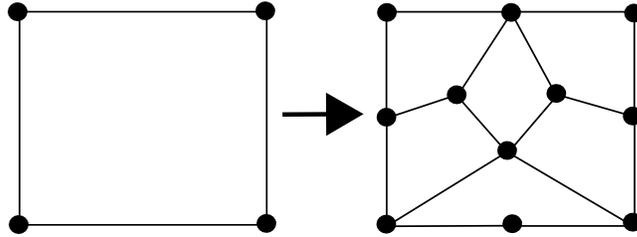


Рис. 5. Разбиение.

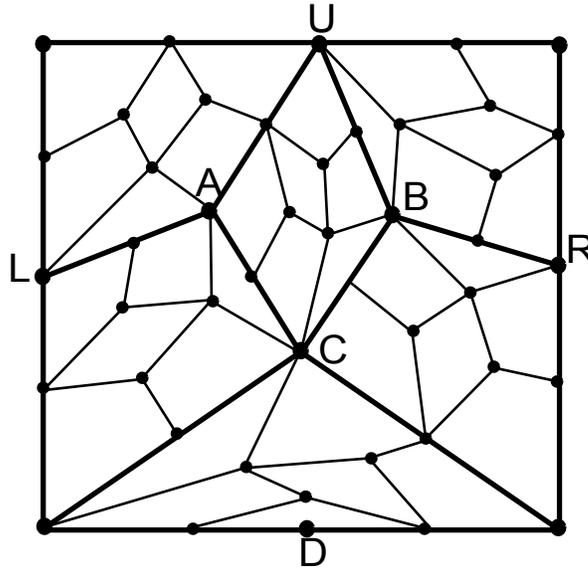


Рис. 6. Макроплитка третьего уровня.

Определение 3.2. *Макроплиткой* уровня n назовем: для $n = 1$ – обычную плитку (простой 4-цикл), для $n > 1$ – результат применения операции Разбиения к макроплитке уровня $n - 1$.

Все создаваемые при разбиении вершины получают уровень на 1 больший, чем максимальный уровень до разбиения, в частности, если создается вершина в середине ребра, она получает уровень на 1 больший, чем максимальный уровень двух концов ребра.

Определение 3.3. При *подклейке* мы рассматриваем все пути $X_1X_2YZ_1$ из пяти вершин (и четырех ребер), такие что:

- 1) Вершины X_1, Y, Z_1 **не** являются тремя углами из четырех никакой макроплитки;
- 2) Вершины X_1, Z_1 являются боковыми вершинами уровня $k - 1$, где k – максимальный уровень вершины в комплексе;
- 3) Вершина X_2 является серединой X_1Y , а Z_2 – серединой Z_1Y , то есть, X_2 и Z_2 являются боковыми вершинами уровня k , созданными в комплексе самыми последними, при последней операции *Разбиения*.
- 4) Вершина Y имеет уровень $k - 2$.

Далее, для каждого такого пути создаются шесть новых вершин $T_1, T_2, T_3, T_A, T_B, T_C$, не лежащих в плоскости X_1YZ_1 , а также проводятся новые ребра $X_1T_2, X_2T_A, X_2T_B, T_2T_B, T_CT_B, T_CT_A, T_2T_1, T_3T_1, T_3Z_1, T_CZ_1, T_AZ_2$ и появляется новая *подклеенная* макроплитка $X_1YZ_1T_1$ (рисунок 7).

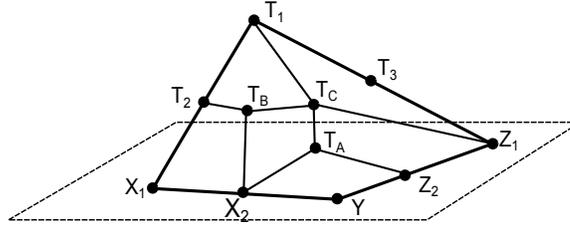


Рис. 7. Подклейка.

Созданные вершины будем называть *подклеенными*, причем T_1 присваивается уровень $k - 1$, а вершинам T_2, T_3, T_A, T_B, T_C – уровень k . Вершину T_1 будем считать угловой, вершины T_2, T_3 – боковыми, а вершины T_A, T_B, T_C – внутренними.

Примечание 1. Подклеенная макроплитка выглядит так же, как если бы к плитке $X_1Y Z_1T_1$ применили Разбиение, считая сторону X_1Y верхней. Также можно считать, что в этой макроплитке T_2 середина стороны T_1X_1 , а T_3 – середина стороны T_1Z_1 .

Примечание 2. Вершину Y будем называть *ядром подклейки*.

Определение 3.4. Комплексом $n + 1$ уровня становится комплекс n уровня, к которому применены *Разбиение* и затем *Подклейка*.

Замечание 1. После разбиения, примененного к комплексу первого уровня, образуется граф на рисунке 6, на котором нет пяти вершин, где можно было бы выполнить подклейку. То есть, подклейки начинают проводиться после второго разбиения.

Замечание 2. Начиная с комплекса второго уровня, подклейки начинают применяться и к подклееным плиткам. Таким образом, возникают подклейки к частям подклеенных когда-то макроплиток и т.д. Кроме того, начиная с третьего уровня комплекса путь $X_1Y Z_1$ из определения операции подклейки может частично лежать в базовой плоскости, а частично в подклеенной части (если Z_1 появилась при предыдущей подклейке в качестве вершины T_2 или T_3).

Предложение 3.1 (Об ограниченности роста степени вершины). *Для каждой вершины X существует такое натуральное N , что начиная с уровня комплекса N , степень (число входящих ребер) вершины X не меняется, то есть она одинакова для комплексов уровня N и $N + k$ для любого натурального k .*

Доказательство. Сначала разберемся с операцией разбиения. Рассмотрим некоторую плитку T . У нее четыре угла X, Y, P, Q (рисунок 8).

При разбиении некоторые углы разбиваются ребрами, а некоторые нет. Будем называть *простыми* те углы, которые не разбиваются ребрами (такими будут углы X и Y). Из правил разбиения плиток следует, что простые углы уже никогда не будут разбиты ребрами при следующих разбиениях.

Назовем угол *ограниченным*, если либо при следующем разбиении он будет разбит на простые углы, либо на простые и ограниченные углы. Например, угол P (правый нижний в плитке T) – ограниченный угол, так как на следующем разбиении он будет подразбит на левые верхние углы новообразованных двух плиток, и дальше уже подразбиваться не будет.

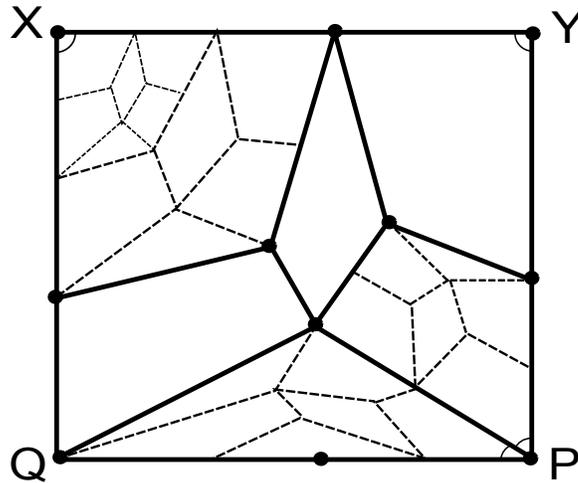


Рис. 8. Простые углы отмечены дугами.

Можно проверить, что каждая из создаваемых боковых и внутренних вершин создает вокруг себя только простые и ограниченные углы, и если проводить только разбиения (без подклеек) то через два этапа после создания вершины, в нее не будут добавляться входящие ребра и ее степень перестает меняться.

Заметим, что степень вершины может возрасти в двух случаях: при разбиении плитки, в которой она участвует и при подклейке плитки, где она выступает в качестве вершины X_1 , X_2 , Z_1 , Z_2 из определения подклейки. Разбиение плиток, начиная с некоторого момента, перестает менять степень выбранной вершины. Участвовать же в подклеенных плитках можно только на трех этапах (следующих после того, когда вершина создается) так как в определении подклеек участвуют вершины от $k - 2$ уровня до k уровня.

Таким образом, начиная со второй итерации после появления, степень вершины может возрасти только при *Разбиении*, а после четвертого разбиения она уже не поменяется.

Таким образом, каждая вершина в комплексе имеет конечную степень, зависящую от уровня комплекса. При этом, число ребер не ведущих в подклеенные плитки, ограничено в совокупности для всех вершин.

Плоские части комплекса представляют собой макроплитки некоторого уровня. Каждая такая макроплитка либо является начальной, базовой плоскостью, либо в какой-то момент была подклеена, а потом несколько раз разбита. Макроплитки состоят из плиток (простых циклов длины 4).

Предложение 3.2 (О боковой вершине). 1) Каждая боковая вершина лежит на середине стороны в какой-либо макроплитке или в двух макроплитках одного уровня, лежащих в одной плоскости;

2) Если боковая вершина не находится на границе исходного комплекса первого уровня, то она лежит на одном из восьми внутренних ребер (рисунок 9) в некоторой макроплитке, либо лежит на границе подклеенной макроплитки.

Доказательство. В комплексе первого уровня свойство выполняется. На втором уровне также все верно для всех созданных боковых вершин.

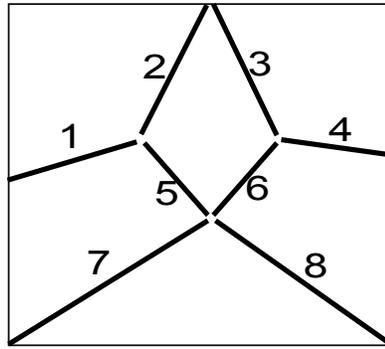


Рис. 9. Восемь типов внутренних ребер.

Пусть для уровня k комплекса все эти свойства выполнены. Для уже существующих боковых вершин оба свойства будут сохраняться при дальнейших разбиениях и подклейках. Если новая боковая вершина возникла при подклейке, то это она использовалась в качестве T_2 или T_3 из определения подклейки, и в качестве макроплитки можно взять как раз подклееную в тот момент макроплитку.

Созданная при разбиении вершина, очевидно, лежит на середине стороны только что разбитой плитки. Если это подклееное ребро или граница начального комплекса первого уровня, то такая плитка одна (и тогда это граница подклеенной макроплитки, либо всего комплекса), в остальных случаях таких макроплиток две.

Для уровней 1 и 2 комплекса заметим, что все стороны плиток, не лежащие на границе, являются частью какого-то большего ребра, классифицируемого, как одного из восьми типов из рисунка 9. При дальнейших подклейках и разбиениях эти ребра не могут сменить тип. При этом создаваемые при разбиении внутренние ребра, очевидно, получаются одного из этих восьми типов.

Если ребро, при разбиении которого образовалась вершина не является границей для всего комплекса или подклеенной макроплитки, то оно принадлежит одному из восьми типов из рисунка 9 для какой-то макроплитки. Значит и вершина тоже принадлежит такому ребру.

Примечание. Для каждой боковой вершины X можно выделить пару вершин (Y, Z) , являющихся концами ребра, на котором лежит X . Эту пару вершин будем называть “начальниками” для X .

3.3. Пути на комплексе. В этом параграфе мы установим несколько свойств путей, проходящих по комплексу.

Путь, идущий по макроплитке, может быть локально преобразован:

Предложение 3.3 (О переброске пути). Пусть $XYZT$ – некоторая макроплитка. Рассмотрим путь XYZ (состоящий из двух соседних макроребер). Тогда, если разрешается менять подпуть из двух соседних ребер любой плитки на подпуть из двух других ребер (с общими началом и концом у этих подпутей), то путь XYZ может быть преобразован в XTZ – путь по другим двум макроребрам.

Доказательство по индукции. Для случая, когда макроплитка – это плитка, преобразование можно сделать сразу. Шаг индукции можно совершить, выполняя локальные преобразования по правилам, показанным на рисунке 10.

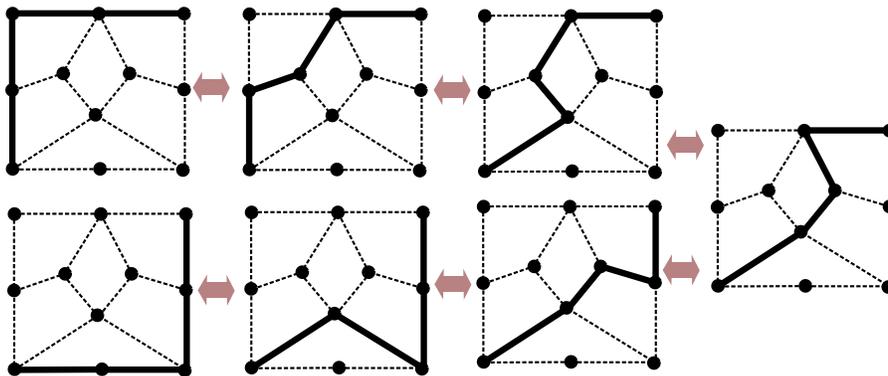


Рис. 10. Переброска пути на другую сторону макроплитки.

Примечание. Применяя аналогичные рассуждения, можно доказать то, что также перебросить можно и путь, начало и конец которого лежат в серединах противоположных ребер макроплитки.

Определение 3.5. Замены подпутей из пары соседних ребер плитки на подпуть из другой пары будем называть *локальными заменами*. А возможность с помощью таких замен достичь некоторого состояния, будем называть *локальным преобразованием*.

Определение 3.6. Будем считать форму пути *нулевой*, если она содержит подпуть длины 2 по ребру некоторой минимальной плитки (туда-обратно).

Предложение 3.4 (О “выносе” пути на границу). Пусть начало и конец пути P , проходящего по макроплитке T , лежат на границе T . Тогда можно реализовать одну из двух возможностей:

- 1) P может быть локально преобразован в нулевую форму.
- 2) P может быть локально преобразован в форму P' так, что P' полностью лежит на границе T .

Доказательство по индукции. Любой путь по минимальной плитке и так лежит на ее границе. Пусть макроплитка уровня n . Рассмотрим ее разбиение на 6 (макро)плиток уровня $n-1$. Путь P разбивается на несколько участков, каждый из которых имеет начало и конец на границе какой-то из этих подплиток. По предположению индукции, эти участки можно локально преобразовать в форму P_1 так, чтобы весь путь P_1 проходил только по границам подплиток.

Обозначим буквами L, U, R, D середины соответственно левой, верхней, правой, нижней сторон T , и внутренние вершины как A, B, C . Допустим, путь P_1 содержит дважды некоторую вершину нашей макроплитки T . Тогда он должен полностью обходить некоторую подмакроплитку уровня $n-1$, примыкающую к этой вершине. В этом случае можно применить лемму о переброске пути и получить форму, где выполнено 1).

Пусть P_1 содержит A один раз. Войти и выйти P_1 может только по трем ребрам (направленным в сторону U, L и C). Если вход и выход – по одному ребру, то условие 1) выполнено.

Пусть во входе и выходе задействованы ребра в сторону U и L . Если, например, выйти в сторону L , то сойти с ребра мы не можем (так как P_1 проходит только по границам макроплиток $n - 1$ уровня) и вернуться назад тоже (так как тогда будет выполнено 1). Заметим, что тогда P_1 содержит подпуть LAU или UAL , в обоих случаях этот подпуть, согласно лемме о переброске пути, можно локально преобразовать чтобы он проходил по границе макроплитки T и, таким образом, не будет содержать A .

Если во входе и выходе задействованы ребра в сторону L и C , то P_1 содержит подпуть LAC или CAL и он может быть преобразован в форму соответственно LXC или CXL , где X – левый нижний угол, и в этом случае опять форма не будет содержать A .

Если во входе и выходе задействованы ребра в сторону U и C , то P_1 содержит подпуть UAC или CAU и он может быть преобразован в форму соответственно UBC или CBL , и тоже форма не будет содержать A .

Проделав аналогичные рассуждения для симметричного случая вершины B , мы можем заключить, что путь можно преобразовать в форму P_2 , такую что она либо вообще не содержит A и B , либо содержит кусок UBC или CBU (который может быть преобразован в UAC или CAU соответственно).

Если P_2 не содержит A и B и содержит C , то войти и выйти в C можно только по ребрам, ведущим в левый нижний и правый нижний углы. В этом случае можно применить лемму о переброске пути для участка, содержащего эти углы и C и получить форму, не содержащую вершин A, B, C .

Пусть P_2 содержит кусок UBC или CBU . Посмотрим, как P_2 проходит через C : один вход (или выход) в сторону B , другой не может идти в сторону A , так как P_2 не содержит A , значит он идет в сторону нижнего левого или нижнего правого угла. Если это нижний правый угол Y то P_2 содержит кусок $UBCY$ или $YCBU$. Применяя лемму о переброске путей, $UBCY$ переводится в $UBRY$ и далее в $UZRY$, где Z правый верхний угол. Аналогично, $YCBU$ переводится в $YRZU$. Если это нижний правый угол X , то P_2 содержит кусок $UBCX$ или $XCBU$. Аналогично применяя лемму 3.3 о переброске пути, переводим $UBCX$ в $UACX$ и далее в $UALX$ и потом в $UWLX$, где W – левый верхний угол. Аналогично $XCBU \Rightarrow XCAU \Rightarrow XLAU \Rightarrow XLWU$.

Таким образом, мы получим форму не содержащую вершин A, B, C . В этом случае, если нельзя выполнить условие 1), то весь путь проходит по границе T , и выполнено условие 2).

Предложение 3.5 (О “шевелении” пути). Пусть P – путь, состоящий из макроребра XU некоторой макроплитки T уровня не менее 3. Тогда P может быть локально преобразован в форму P' , состоящей из трех частей: первая и третья это участки по $1/4$ пути P , примыкающие к X и Y соответственно, а вторая представляет собой путь по двум соседним ребрам подклеенной макроплитки, два других ребра которой являются частью P .

Для доказательства достаточно проследить разбиение T (на три уровня в глубину). Обозначим середину XU как H , середины XH и HU как G_1 и G_2 соответственно, и середины G_1H и HG_2 как G_3 и G_4 . Так как G_3 и G_4 будут иметь уровень на 2 больше, чем H , значит, была проведена подклейка для пути $G_1G_3HG_4G_2$. Пусть F – созданная при этой подклейке угловая вершина. По предыдущей лемме, G_1HG_2 можно локально преобразовать в G_1FG_2 .

Предложение 3.6 (О выделении локального участка). Если путь P , проходящий по макроплитке T уровня n , имеет длину не менее $5 \cdot 2^{n-2}$, он может быть локально преобразован в нулевую форму P' .

Если путь P , проходящий по макроплитке T уровня n , имеет длину не менее $2^n + 1$, и начинается в углу T , либо в середине стороны T , он может быть локально преобразован в нулевую форму P' .

Доказательство. Согласно лемме о выносе на границу, можно считать, что весь путь P проходит по границе T . Пусть при этом не образовалось нулевой формы, то есть путь покрывает какую-то часть границы T , совершая движение либо по часовой стрелке (либо против). Отметим на границе T восемь точек – углы и середины сторон. В обоих случаях 1) и 2) наш путь включает в себя не менее 5 из отмеченных точек, иначе он был бы короче, чем $5 \cdot 2^{n-2}$ или $2^n + 1$ соответственно. Тогда найдутся две отмеченные точки, диаметрально противоположные относительно T , и такие, что путь P полностью покрывает половину границы между ними. Заметим при этом, что хотя бы одна из этих отмеченных точек не является концевой для P . Пользуясь леммой о переброске (с примечанием) можно перевести подпуть из половины границы на другую половину. Ясно, что получившаяся форма будет нулевой.

Предложение 3.7 (О непродолжаемом пути). Пусть путь P лежит в некоторой макроплитке T , причем начало и конец P лежат в серединах противоположных сторон T . Тогда любые плоские пути вида WP или PW , где длина W более длины P , могут быть приведены к нулевой форме.

Доказательство. Случаи WP и PW аналогичны, рассмотрим PW . Пусть a – это та сторона T , в середине X которой заканчивается P и начинается W . Если выход из X идет внутрь P , либо по стороне a , тогда рассматривая путь P плюс одна вершина, мы получаем путь длины $2^n + 1$, лежащий в макроплитке P и можем применить лемму о выделении локального участка. Значит, выйти из X может только в макроплитку T' , соседствующую с T по стороне a . Допустим, существует вершина Y из пути W , лежащая на границе T' , отличная от X . В этом случае, кусок пути W от X до Y может быть локально преобразован так, что он будет проходить по границе T' . Но тогда он выход этого куска из X будет по стороне a , и в этом случае опять получаем путь длины $2^n + 1$, лежащий в макроплитке P . Если такой вершины Y не найдется, то все остальные вершины W лежат внутри T' , то есть получаем опять путь длины $2^n + 1$, на этот раз лежащий на T' .

Определение 3.7. Паттерном пути будем называть последовательность его вершин, из которой выброшены все боковые вершины, вход и выход в которые проходят по ребрам, лежащим на границе между одной и той же парой макроплиток (главные ребра для боковой вершины). То есть в паттерн пути входят все черные (внутренние) вершины, и все боковые вершины, где происходит вход(выход) в (из) макроплитку. Фактически паттерн пути – это его карта, показывающая маршрут на ориентировочных точках.

Определение 3.8. Мертвым будем называть такой паттерн P , что любой достаточно большой путь W , собственным подпутем которого является путь с паттерном P , может быть локально преобразован к нулевой форме.

Предложение 3.8 (О мертвых паттернах). Рассмотрим некоторую макроплитку T и обозначим в ней внутренние вершины A, B, C и боковые U, R, D, L (аналогично обозначениям при разбиениях). Тогда паттерны AUB, ACB, CXD (где X – любая вершина) являются мертвыми.

Доказательство. Пути с паттернами AUB и ACB локально преобразуются друг в друга, так что достаточно рассмотреть один из этих случаев. Пусть W – достаточно большой путь, содержащий подпуть с паттерном ACB . Тогда можно считать, что W содержит

вершины на границе T , лежащие до и после нашего подпути. Рассмотрим разбиение макроплитки T на 6 дочерних подплиток (в соответствии с операцией разбиения) и будем локально преобразовывать W так, чтобы он проходил по границам этих дочерних подплиток. Тогда из вершины B путь должен идти в по ребру к R , а в A путь должен входить по ребру из L , в остальных случаях образуется обход подплитки по трем ребрам (и этот кусок преобразуется к нулевой форме). Таким образом, в W можно выделить подпуть, проходящий по макроплитке T , начало которого в L и конец в R . Можно воспользоваться леммой о непродолжаемом пути и получить требуемое.

Для паттерна CXD , пусть T' – макроплитка, соседняя с T по нижней стороне. Заметим, что выход из вершины D обязательно должен быть внутрь макроплитки T' , иначе можно применить лемму о выделении локального участка для нижней дочерней подплитки T . Проследим путь W до ближайшей точки X на границе T' (после D) и преобразуем кусок DX чтобы он проходил по периметру T' . Но тогда из D путь W продолжится по граничной между T и T' стороне. То есть теперь, все таки, можно применить лемму о выделении локального участка для нижней дочерней подплитки T и получить требуемую нулевую форму.

Примечание. Ясно, что и паттерны BUA , BVA , DXC тоже мертвые, доказательство полностью аналогично.

Предложение 3.9 (О мертвых путях в нижней подплитке). *Рассмотрим некоторую макроплитку T . Пусть путь XYZ лежит в T , причем X лежит на внутреннем ребре, идущем из левого нижнего угла T во внутреннюю вершину C , Y – нижний угол T , а Z лежит на нижней стороне T . Тогда существует такое натуральное N , что для любых путей W_1, W_2 , длины которых более N , путь W_1XYZW_2 можно локально преобразовать к нулевой форме.*

Доказательство. Пусть Y – левый нижний угол. Обозначим середину нижней стороны T как D . Пусть T' нижняя подплитка T . Преобразуем путь W_1XYZW_2 так, чтобы он проходил по границам макроплиток уровня T' (не заходя в более мелкие плитки). Ясно, что W_1 должен проходить через C , а W_2 через D , иначе возникает очевидный кусок с нулевой формой. Таким образом, после преобразования, путь будет содержать подпуть с паттерном CXD , который является мертвым, по лемме о мертвых паттернах. Следовательно, для достаточно больших W_1 и W_2 наш путь можно преобразовать к нулевой форме.

Замечание. Доказательство аналогично переносится на случай, когда Y – правый нижний угол.

Определение 3.9. *Некорректным участком* пути будем называть такой подпуть XYZ , что вершина Y лежит на границе некоторой макроплитки T , а вершины X и Z лежат внутри T (не на границе).

Предложение 3.10 (О некорректных участках). *Пусть есть некорректный участок XYZ . Тогда существует константа N , что для любых путей W_1, W_2 , длины более N , путь W_1XYZW_2 , может быть локально преобразован к нулевой форме.*

Доказательство. Пусть T – минимальная макроплитка, внутри которой проходит путь XYZ . Из минимальности следует, что вершина Y лежит в углу T , либо в середине стороны. Из иерархии разбиений следует, что из всех углов кроме нижнего левого и из середины нижней стороны выходит не более одного ребра внутрь T , так что Y может лежать только на середине левой, правой или верхней стороны или в левом нижнем углу.

1. Пусть Y – левый нижний угол. Из этого угла внутрь выходит два ребра, оба к внутренней C -вершине, одна в макроплитке T , другая в нижней дочерней подплитке T . То есть, тогда путь XYZ попадает под условие леммы о нижней подплитке и все доказано.

2. Пусть Y – середина левой стороны. Из этой вершины выходит три ребра и поэтому есть три варианта расположения пути XYZ . В двух случаях можно применить лемму о нижней подплитке. Оставшийся случай изображен на рисунке 11.

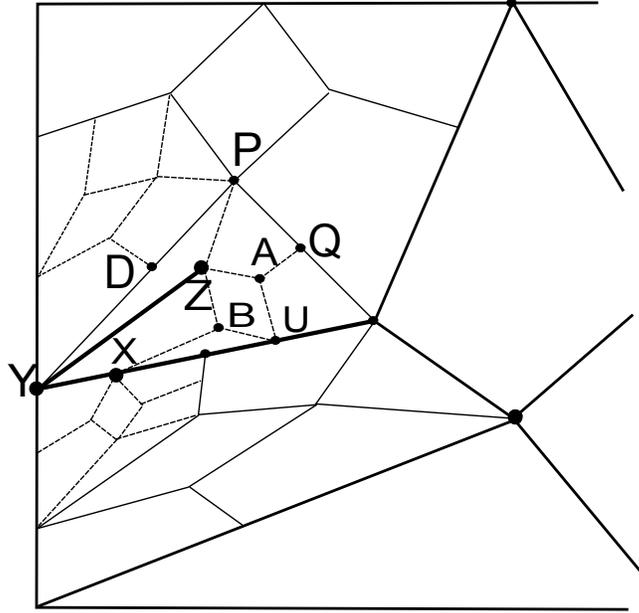


Рис. 11. Вершина Y – середина левой стороны.

Из вершины Z дальше путь может пойти:

i) в вершину Z_1 . В этом случае кусок YZZ_1 преобразуется в YDZ_1 и для участка XYD можно применить лемму о нижней подплитке.

ii) в вершину A . Посмотрим, куда путь может пойти дальше. Если это вершина Z_2 , то участок $YZAZ_2$ преобразуется в YZZ_1Z_2 и далее в YDZ_1Z_2 , после чего опять можно применить лемму о нижней подплитке. Во втором случае, если путь идет в вершину U , то путь $XYZAU$ сразу можно привести к нулевой форме: $XYZAU \rightarrow XYZBU \rightarrow XYXBU$.

3. Пусть Y – середина правой стороны. Этот случай симметричен второму, только вместо левой верхней подплитки мы имеем дело с правой нижней. Все рассуждения полностью аналогичны второму случаю.

4. Пусть Y – середина верхней стороны. Из этой вершины исходят четыре ребра (рисунок 12).

В случае, когда X и Z не лежат на левом ребре (идущем из Y к внутренней вершине A макроплитки T), то получается ситуация, полностью аналогичная второму и третьему случаям выше, на этот раз вместо левой верхней подплитки мы имеем дело с правой верхней. Итак, пусть верхняя правая подплитка T это T' и, ради определенности, X лежит на ребре, уходящем из Y к внутренней вершине A макроплитки T , а Z – на одном из трех ребер, попадающих в T' (Отмечены пунктиром на рисунке 12).

Проведем локальные преобразования пути W_1 так, чтобы он проходил по границам макроплиток уровня T' (не заходя внутрь них). Последняя вершина W_1 это X . Таким образом, после этого преобразования либо последний участок будет YX (и тогда весь наш

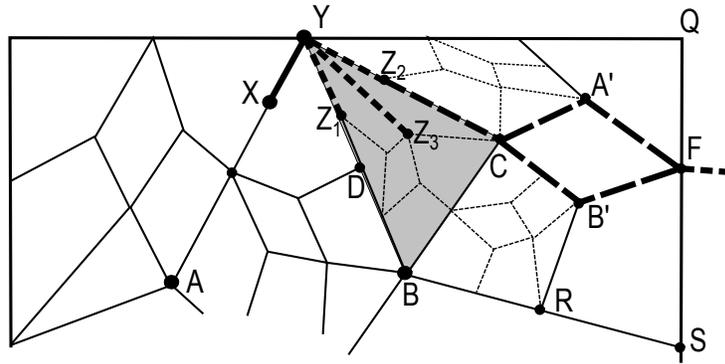


Рис. 12. Вершина Y – середина верхней стороны.

путь получит нулевую форму) либо преобразованный W_1 будет проходить через внутреннюю вершину A макроплитки T . Теперь рассмотрим путь W_2 . Наша цель – преобразовать его к виду, чтобы он проходил через внутреннюю вершину B макроплитки T .

Пусть T'' это нижняя подплитка T' (отмечена цветом на рисунке). Проведем локальные преобразования W_2 так, чтобы он проходил по границам макроплиток уровня T'' (не заходя внутрь них). Ясно, что ближайшая к Z часть W_2 пройдет по одному из участков $YCA'F$ или $YCB'F$, указанных длинным пунктиром на рисунке 12 (либо W_2 пройдет через B). Таким образом, W_2 теперь проходит через вершину F . Теперь, если из F путь W_2 идет к вершине Q , то оба возможных куска $YCA'FQ$ и $YCB'FQ$ приводятся к нулевой форме. Если путь из F идет в S , то кусок $YCA'FS$ (или $YCBFS$) приводится к $YDBRS$. Остается возможность, когда W_2 покидает макроплитку T , уходя в соседнюю макроплитку по правой стороне (короткий пунктир). В этом случае весь кусок W_2 после F мы преобразуем так, чтобы он проходил по границам макроплиток уровня T . Тогда преобразованный участок будет уходить из F только в Q или S , а эти случаи мы разобрали.

Итак W_1 можно привести к форме, проходящей через внутреннюю вершину A макроплитки T , а W_2 к форме, проходящей через внутреннюю вершину B . Теперь получившийся путь имеет подпуть с мертвым паттерном AUB , вследствие чего, для достаточно больших путей W_1 и W_2 мы можем привести путь W_1XYZW_2 к нулевой форме.

Примечание. Рассуждения не меняются, если рассмотренная макроплитка T имеет меньший уровень. В этом случае вершина Z_1 совпадает с D , а Z_3 с C .

Предложение 3.11 (О корректности путей). Пусть путь P представляет собой проход по двум соседним ребрам некоторой макроплитки. Тогда любые локальные преобразования не могут привести P к форме, содержащей некорректный участок или к нулевой форме.

Доказательство. Для двух точек комплекса определено расстояние как наименьшая из длин соединяющих их путей. Путь, проходящий по двум соседним ребрам макроплитки является кратчайшим для вершин в его концах, что легко доказывается по индукции по уровню макроплитки. Локальные преобразования не меняют длину пути. В случае, если после очередного преобразования появляется нулевая форма, путь не может быть кратчайшим.

Теперь, допустим, что наш путь можно привести к форме, содержащей некорректный участок. Заметим, что путь после любого числа преобразований имеет форму W_1UW_2 , где W_1, W_2 полностью лежат на границах макроплитки, а у подпути U концы лежат на

границе, а все остальные вершины лежат внутри. Для макроплиток первых уровней это проверяется непосредственно, а для остальных уровней можно проверить по индукции. В случае, если путь содержит некорректный участок, он не может иметь такую форму.

Предложение 3.12 (О расстоянии от края макроплитки до выхода в подклейку). Пусть вершина X лежит на краю некоторой макроплитки T , вершина Y принадлежит T , но не находится на ее границе, а выход из Y идет в подклееную макроплитку уровня $n \geq 2$. Тогда расстояние от X до Y (длина кратчайшего пути по ребрам) не менее 2^{n-1} .

Доказательство проводится индукцией по уровню макроплитки T . Для уровней 2,3 подклеек внутри T не проводится. Для уровня 4 ближайшая вершина с подклееным ребром лежит в середине любого ребра, выходящего на середину любого из граничных ребер. Расстояние от нее до границы T будет равно 2, при этом подклеенная макроплитка будет иметь уровень 2.

Пусть уровень T равен $k > 4$. Возьмем кратчайший путь XWY . Можно считать, что W не содержит вершин на границе T . Кроме того, Y не может лежать внутри какой-то из подплиток T , иначе X можно взять на границе этой подплитки. Значит, Y обязательно лежит на одном из ребер, которые разделяют макроплитку T на 6 макроплиток при разбиении. Опять используя минимальность пути, замечаем, что Y лежит в углу подклеенной макроплитки. Из определения подклейки получаем, что уровень Y был на 1 меньше максимального в тот момент, когда эта подклейка производилась. Если после этого разбиений не было, то расстояние XW не менее 2, а также макроплитка при Y – второго уровня. Если после этого провели $l \geq 1$ разбиений, расстояние XW увеличилось в 2^l раз, а уровень макроплитки при Y возрос на 1.

Следствие 3.1. Расстояние между двумя выходами в подклеенные области не менее 2^{n-1} , где n уровень макроплитки, куда идет второй выход. То есть если путь P имеет форму XP_1YZ , где P_1 некоторый путь, выходы из вершин X и Y осуществляются по подклееным ребрам, а все остальные входы во все вершины, и выходы из них плоские. Уровень подклеенной макроплитки, куда идет выход из Y равен n . Тогда расстояние от X до Y не менее 2^{n-1} .

Таким образом, если в пути встречаются два ребра выходящие из вершин по подклееным ребрам, то они разделены, как минимум, одним плоским ребром.

Каждая подклеенная макроплитка имеет ограниченные размеры, соответственно, путь, не имеющий нулевой формы и не возвращающийся из подклеенной макроплитки будет иметь ограничения на длину.

Предложение 3.13 (Об ограниченности пути, уходящем в подклеенную часть два раза подряд). Пусть путь P имеет форму XP_1YP_2 , где P_1Y – плоский участок (лежащий в какой-то макроплитке) и выходящие из вершин X и Y ребра идут в подклеенные плитки, и больше выходов в подклеенную область или входов в вершину из подклеенной области нет. Пусть также P не может быть приведен к нулевой форме. Тогда длина YP_2 менее, чем $\frac{5}{2}$ длины XP_1Y .

Доказательство. Пусть уровень макроплитки, куда уходит ребро из Y_2 , равен n . Тогда длина участка YP_2 меньше чем $5 \cdot 2^{n-2}$ по лемме о выделении локального участка. Кроме того, длина XP_1Y не менее 2^{n-1} по следствию леммы о расстоянии от границы макроплитки до подклеенного ребра. Следовательно, длина YP_2 меньше, чем $\frac{5}{2}$ длины XP_1Y .

4. СТРОЕНИЕ ПОЛУГРУППЫ ПУТЕЙ

В предыдущих разделах мы описали геометрическую структуру комплекса. В следующей главе 5 будет показано, как связать с ним специальную кодировку. Мы покажем, что с помощью конечного числа букв можно закодировать узел. Кроме того, конечным числом букв можно закодировать входящие и выходящие ребра в каждый узел, как плоские, так и выходящие в подклейку.

В этом разделе мы покажем, для чего нужна такая кодировка и как необходимая нам полугруппа строится с помощью кодировки путей на комплексе.

Пусть X_i – буквы, кодирующие входящие ребра, Y_i – буквы, кодирующие узлы (их типы, окружения и информации), Z_i – буквы, кодирующие выходящие ребра. (Детали кодирования будут показаны в последующих разделах.)

Будем говорить, что слово W имеет *CODE-форму*, если в нем справа от любой (не последней в слове) буквы семейства X обязательно стоит буква семейства Y , справа от любой (не последней в слове) буквы семейства Y стоит буква семейства Z , а справа от любой (не последней в слове) буквы семейства Z стоит буква семейства X .

Часть таких CODE-слов соответствует путям на комплексе. Рассмотрим полугруппу с нулем S с порождающими $\{X_i, Y_i, Z_i\}$.

Предложение 4.1 (О стандартной форме слова). *В полугруппе с нулем S с порождающими $\{X_i, Y_i, Z_i\}$ можно ввести конечное число определяющих мономиальных соотношений так, чтобы все не CODE-слова, можно было привести к нулевому слову.*

Доказательство. Действительно, введем соотношения $X_i Z_j = 0$, $X_i X_j = 0$, $Y_i X_j = 0$, $Z_i Y_j = 0$, $Y_i Y_j = 0$, $Z_i Z_j = 0$, для всевозможных пар (i, j) . В получившейся полугруппе, любое ненулевое слово будет иметь CODE-форму: за буквой X_i может следовать только Y_j , за Y_j – только Z_k , за Z_k – только X_m .

Заметим также, что можно выписать все CODE-слова длины не более 4, которые кодируют какой-нибудь путь на комплексе. Пусть F_1, \dots, F_N – все оставшиеся CODE-слова длины не более 4. Их мы будем считать *запрещенными* и введем в полугруппе S соотношения $F_i = 0$, для всех $i = 1, \dots, N$.

Таким образом, любое слово W в полугруппе S либо приводится к нулю, либо имеет CODE-форму, где каждое подслово W длины не более 4 отвечает некоторому пути на комплексе.

В разделах 8 и 9 будет показано, что для двух эквивалентных путей, не содержащих мертвых паттернов и нулевых форм, по коду одного из них можно восстановить код другого. Это дает возможность ввести конечное множество соотношений вида $W_i = W_j$, где W_i, W_j – коды двух эквивалентных путей.

Таким образом, появляется возможность преобразовывать слова, меняя подпути на им эквивалентные. При этом если до преобразования слово являлось кодировкой некоторого пути на комплексе, то и после преобразования слово оно будет кодировкой эквивалентного ему пути. Допустим, после нескольких таких замен образуется запрещенное подслово. Это значит, что соответствующий этой кодировке путь не может быть реализован на комплексе. Но тогда и эквивалентный ему путь не мог быть реализован. Таким образом, если, преобразовывая слова-кодировки по установленным правилам, мы получаем запрещенное слово, значит изначальное слово не соответствует пути на комплексе.

Нулевая форма. Если слово W представляет собой кодировку пути, имеющего *нулевую форму*, мы вводим в полугруппе соотношение $W = 0$. Рассмотрим слово-кодировку

пути, концы которого лежат на периметре некоторой макроплитки T , и хотя бы один конец попадает в угол T . Заметим, что если путь не кратчайший (то есть существует путь короче по длине с теми же концами), то согласно предложению 3.6, путь может быть преобразован в нулевую форму. Это означает, что соответствующее слово-кодировка приводится к нулю.

Фактически, обнуление нулевых форм позволяет приводить к нулю все слова, соответствующие не кратчайшим путям.

Мертвые паттерны. Мертвые паттерны представляют собой некоторые типы слов-кодировок, которые никогда не могут встретиться в достаточно больших ненулевых словах. Нас будут интересовать только мертвые паттерны AUB , ACB , CXD , BUA , BCA , DXC , которые обсуждаются в предложении 3.8. Выпишем все слова, длины не более 4, содержащие хотя бы один из перечисленных мертвых паттернов. Для каждого такого слова W , введем соотношение $W = 0$.

Заметим, что все достаточно большие ненулевые слова не могут содержать мертвый паттерн (и не могут быть приведены к форме, его содержащей).

Будем называть слово *регулярным*, если оно является кодировкой некоторого пути на построенном геометрическом комплексе и при этом не может быть преобразовано в нулевую форму или форму содержащую мертвый паттерн.

Заметим, что замена некоторого подслова регулярного слова на эквивалентное ему, не нарушает регулярность. То есть, если слово можно преобразовать так, что оно будет содержать запрещенное подслово, нулевую форму или мертвый паттерн, то значит оно приводится к нулю, и не может быть регулярным.

Нашей основной целью будет показать, что ненулевыми словами в полугруппе являются только регулярные слова, то есть, что любое нерегулярное слово может быть приведено к нулю. Фактически, это будет означать, что ненулевыми элементами полугруппы являются только слова, кодирующие кратчайшие пути, не содержащие мертвые паттерны.

Примечание о использовании мертвых паттернов для конструкции. Ценой некоторого усложнения кодировки можно добиться того, чтобы пути, содержащие мертвые паттерны, допускали такие же локальные преобразования, как и обычные слова. В этом случае, можно не вводить мономиальные соотношения для слов, их содержащих. И тогда ненулевыми элементами полугруппы будут просто кратчайшие пути на построенном комплексе. Но в целях упрощения конструкции, представляется разумным ввести такие соотношения.

5. КОДИРОВКА ВЕРШИН И ПУТЕЙ НА КОМПЛЕКСЕ

Рассмотрим путь на комплексе. Для каждой вершины из этого пути, кроме первой и последней, есть ребро входа и ребро выхода. Ребра входа и выхода могут вести в подклеенные области, а могут принадлежать той же базовой плоскости, что и сама вершина.

Каждая вершина принадлежит некоторой макроплитке. Кроме того из нее могут выходить ребра ведущие в подклеенные плитки/макроплитки. То есть, каждая вершина лежит на своей базовой плоскости и еще участвует в нескольких подклеенных плоскостях. Фактически, вершина играет свою роль (занимает определенное положение) в своей макроплитке на базовой плоскости, и кроме этого выполняет другие какие-то роли в своих подклеенных макроплитках. При этом в подклеенных макроплитках эта вершина лежит всегда на краю.

Мы хотим определить систему кодирования путей на комплексе. Каждый путь это конечная последовательность вершин, причем любые две соседние соединены ребром плитки

минимального уровня (не макроплитки). Для кодирования путей нужно сначала закодировать все вершины, которые могут встретиться на этом пути, конечным числом букв.

Каждая вершина может принимать участие во множестве подклеенных макроплиток, то есть входящие в нее ребра либо относятся к плоскости вершины либо классифицируются по принадлежности к различным подклееным макроплиткам. В общем случае, путь приходит в вершину из одной подклеенной макроплитки и выходит в другую. Общая кодировка вершины будет состоять из трех *плоских* кодов: первый представляет собой плоский код вершины (без учета подклеенных плиток) второй- кодирует положение этой же вершины в плоскости подклеенной макроплитки, откуда приходит входящее ребро, и третий кодирует положение этой же вершины в плоскости подклеенной макроплитки, куда уходит выходящее ребро.

Если рассматривать вершину только в рамках одной макроплитки, в которую она входит, имеет смысл говорить о *плоском* коде. Например, если весь путь лежит в одной плоскости и не выходит в подклеенные области, то код такого пути будет плоским. Плоский код одной вершины будет состоять из следующих частей:

- 1) тип вершины;
- 2) уровень вершины;
- 3) окружение;
- 4) информация.

Мы последовательно определим, что значит каждая из этих частей.

5.1. Параметр “тип” для вершин. Все вершины, встречающиеся на комплексе, мы разделим на следующие категории:

1) *Угловые* (лежащие в углах подклеенных макроплиток или всего комплекса). *Тип угловой вершины* определим как один из четырех вариантов углов, в котором она может находиться: CUL, CUR, CDR, CDL. (Corner Up-Left и так далее.)

2) *Краевые* (лежащие на стороне подклеенной макроплитки). Каждая такая вершина лежит в середине стороны некоторой макроплитки, прилегающей к краю. *Тип краевой вершины* определим в соответствии с тем, серединой какой стороны в этой макроплитке она является: L, R, D, U.

3) *Регулярные* (лежащие внутри некоторой макроплитки, либо на границе между двумя макроплитками). В этой категории определим три типа *внутренних* вершин: A, B, C, отвечающих внутренним, черным узлам макроплиток. А также определим восемь типов *боковых* вершин, лежащих в середине сторон двух соседних макроплиток. Учитывая ориентацию этих макроплиток, типы будут следующие: DR, RD, DL, LD, UR, RU, UL, LU.

В дальнейшем, вершину типа A будем, для простоты, называть A-вершиной или A-узлом. Аналогично для других типов. Вообще, вершины на графе мы также будем называть узлами. В левой части рисунка 13 представлены внутренние и боковые вершины. Следующее предложение показывает, что возможны только указанные сочетания сторон.

Предложение 5.1 (О комбинациях сторон на границе макроплиток). *Пусть вершина X лежит в середине сторон двух макроплиток, лежащих в одной плоскости. Тогда возможны только следующие сочетания ориентаций сторон: правая и нижняя, верхняя и правая, левая и нижняя, либо верхняя и левая.*

Доказательство. По второй части леммы о боковой вершине, X лежит на некотором ребре YZ, которое относится к одному из восьми типов. (Правая часть рисунка 13). Рассмотрим вершину F, лежащую в середине этого ребра. F тоже боковая вершина и сочетания ориентаций сторон у нее могут быть только право и низ, верх и право, лево

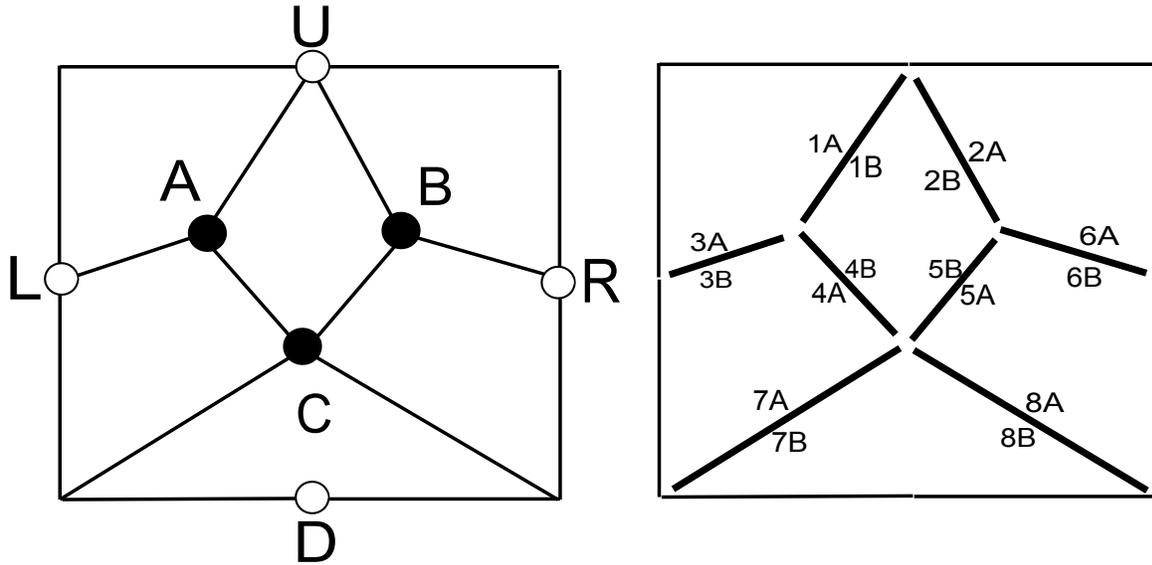


Рис. 13. Черные (внутренние) и белые (боковые) вершины; типы ребер

и низ, верх и лево, то есть как в условии леммы. Если F совпадает с X , то все доказано. Иначе рассмотрим разбиение наших соприкасающихся макроплиток. Легко проверить, что, согласно правилам разбиения, середины ребер YF и FZ тоже могут иметь сочетание ориентаций только право и низ, верх и право, лево и низ, верх и лево (RD , UR , LD , UL и транспозиции). Например, если макроплитки соприкасаются нижней и правой сторонами, то F имеет тип DR или RD , и тогда YF и FZ будут иметь типы RU и UL (или наоборот). Повторяя операцию разбиения, можно получить, что все боковые вершины на ребре YZ будут относиться к одному из указанных четырех типов, то есть являются серединами правой и нижней, либо верхней и правой, либо левой и нижней, либо верхней и левой сторон.

5.2. Типы ребер и параметр “уровень” для вершин. Будем считать, что вершина *принадлежит* макроплитке, если она образуется при первом разбиении этой макроплитки (когда она из плитки становится макроплиткой второго уровня). То есть, белые вершины принадлежат тем макроплиткам, в середине сторон которых они лежат, а черные – макроплиткам минимального уровня, внутри которых они находятся.

Типы ребер. Всего существует 16 внутренних ребер (с учетом стороны, рисунок 13), это ребра $1A, 1B, \dots, 8B$ и 4 краевых ребра: *left, top, right, bottom*.

Уровни вершин. Для черных, внутренних вершин уровень определять не будем (то есть, можно считать, что у всех один уровень).

Для белых, боковых вершин уровня будет три: первый у вершин, принадлежащих макроплиткам второго уровня, второй – у вершин, принадлежащих макроплиткам третьего уровня, третий – у всех остальных белых вершин.

Рассмотрим черную вершину X . Если X имеет тип A и B , то она находится на стыке трех макроплиток (разбивающих макроплитку, которой X принадлежит), для C таких макроплиток четыре. Назовем ребра, исходящие из X и при этом лежащие на границе каких-то двух из этих макроплиток, *главными ребрами* для X .

Пусть X – белая вершина, лежащая на середине сторон двух макроплиток T_1 и T_2 . Из X выходит два ребра, лежащих на границе между T_1 и T_2 , их будем называть *главными*.

Если боковая вершина лежит на краю только одной макроплитки, то главные ребра – это лежащие на ее границе. В зависимости от типа X , из нее выходят еще несколько неглавных ребер. Мы можем разобрать все белые вершины третьего уровня и убедиться, что все прилежащие к X углы будут простые и при дальнейших разбиениях новых исходящих ребер не добавляется.

Ребра входа и выхода. Обозначения. Первым главным ребром для вершин типа \mathbb{A} и \mathbb{B} будем называть то главное ребро, которое уходит к середине верхней стороны той макроплитки, которой принадлежит данная вершина. Для \mathbb{C} первым главным ребром будем считать ребро, идущее в \mathbb{A} -узел.

Для боковых вершин оба главных ребра являются двумя частями некоторого внутреннего ребра. Первым из них будем считать то ребро, относительно которого \mathbb{A} -сторона (из определения типа внутреннего ребра) остается по правую сторону. Второе и третье главные ребра определяются по часовой стрелке после первого. Все главные ребра будем обозначать цифрами (от 1 до 4).

Для краевых вершин первым главным ребром будем считать то ребро, которое соответствует обходу макроплитки по часовой стрелке. То есть, макроплитка остается по правую сторону от направления этого ребра. Ребро, идущее против часовой стрелке, будет вторым.

У вершин типа \mathbb{A} может быть два неглавных ребра, одно уходит в левую верхнюю подплитку, другое в левую нижнюю. Обозначим их, соответственно, lu и ld . У вершин типа \mathbb{B} может быть три неглавных ребра, одно уходит в правую верхнюю подплитку, второе в правую нижнюю, третье в среднюю. Обозначим их, соответственно, ru , rd , mid . У вершин типа \mathbb{C} может быть шесть неглавных ребра, два уходят в левую нижнюю подплитку, два в среднюю, и два в нижнюю. В каждой паре одно из ребер иерархически старше (появляется при разбиении раньше). Обозначим ребра, соответственно, ld_1 , ld_2 , mid_1 , mid_2 , d_1 , d_2 . Более старшему ребру даем первый номер, другому – второй.

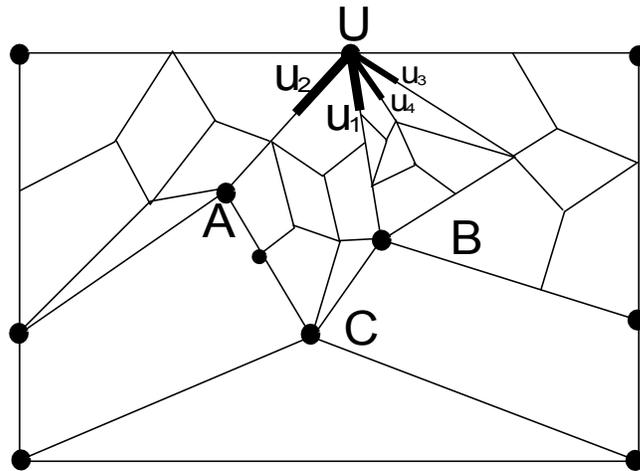


Рис. 14. Неглавные ребра выходящие из бокового узла

Для боковых и краевых вершин будем давать неглавным ребрам имена в соответствии с типом вершины и стороны, откуда они уходят:

\mathbb{U} -узел или U -часть \mathbb{UL} -узла или \mathbb{UR} -узла – это u_1, u_2, u_3, u_4 ;

\mathbb{R} -узел или R -часть \mathbb{UR} -узла или \mathbb{DR} -узла – r, r_2, r_3 ;

\mathbb{L} -узел или L -часть \mathbb{UL} -узла или \mathbb{LD} -узла – l, l_2, l_3 .

В \mathbb{D} -узел неглавные ребра не приходят.

Ребра нумеруются по иерархическому старшинству, при равном старшинстве – по часовой стрелке. Например, есть четыре неглавных ребра из узла типа \mathbb{LU} , входящих внутрь плитки T , примыкающей верхней стороной (рисунок 14). u_1 будет ребром в сторону \mathbb{B} -узла T , u_2 – ребром в сторону \mathbb{A} -узла T . u_3 и u_4 , будут входить в верхнюю правую подплитку T .

Типы ребер, ведущих в подклеенные области. Рассмотрим вершину X . В подклеенной макроплитке она может иметь один из следующих типов: \mathbb{L} , \mathbb{U} , \mathbb{CUR} , \mathbb{CUL} , \mathbb{CDL} .

Из вершины \mathbb{L} могут выходить ребра l_1, l_2, l_3 (как и в плоском случае). Чтобы показать, что имеются в виду ребра в подклеенную область, будем записывать их как $\hat{l}_1, \hat{l}_2, \hat{l}_3$ (рисунок 15).

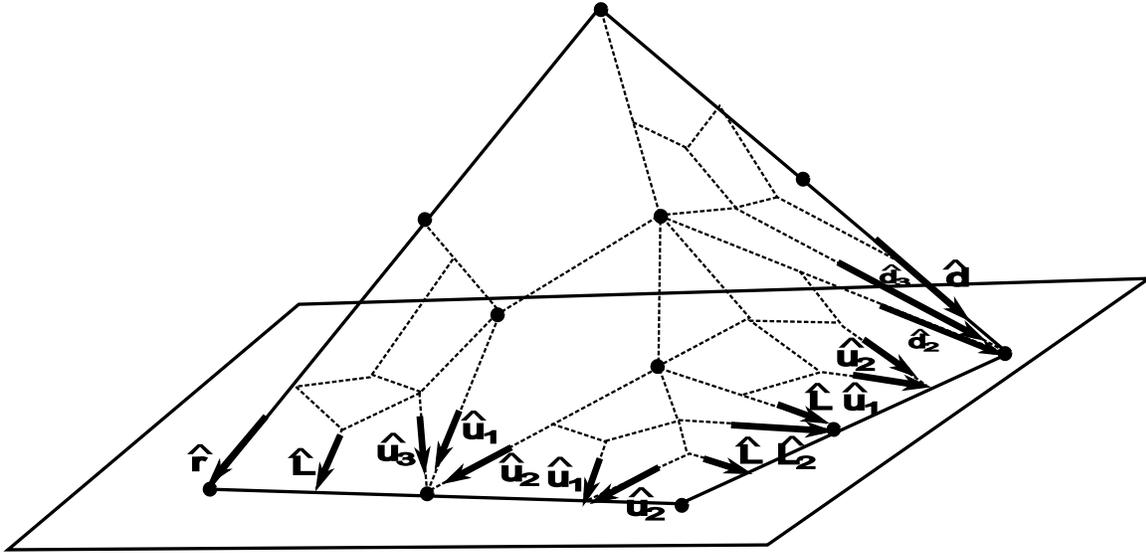


Рис. 15. Ребра в подклеенные области

Аналогично, из вершины \mathbb{U} могут выходить ребра $\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4$.

Из вершины \mathbb{CUR} может выходить только одно ребро в подклеенную область – это ребро по правой стороне подклеенной макроплитки. Будем обозначать его как \hat{r} .

Из вершины \mathbb{CDL} может выходить два ребра в подклеенную область – по нижней стороне подклеенной макроплитки, по ребру 7 подклеенной макроплитки, а также по 8 ребру макроплитки, образующейся при разбиении нижней подплитки подклеенной макроплитки. Будем обозначать их как $\hat{d}_1, \hat{d}_2, \hat{d}_3$ соответственно.

Из вершины \mathbb{CUL} (она же – ядро подклеенной макроплитки) ребер в подклеенные области не выходит.

Цепи. Выберем некоторую вершину X . Обозначим как $T(X)$ множество всех макроплиток одного размера, для которых X является левым верхним углом. ($T(X)$ может быть выбрано несколькими способами, для разных размеров макроплиток.)

Определение 5.1. Совокупность ребер 1 и 3 типов в макроплитках из $T(X)$ и боковых узлов в их концах будем называть *цепью* для X . Будем называть вершину X *центром* цепи.

Из определения следует, что каждый узел может входить только в одну цепь, причем центр для этой цепи определяется однозначно. Под *макроплитками из цепи* будем понимать макроплитки из соответствующего множества $T(X)$. Заметим, что между ребрами,

выходящими из X и вершина цепи, а то есть, количество узлов в любой цепи ограничено сверху (Напомним, что в цепь могут входить только боковые вершины).

Для вершин в одной цепи можно задать упорядоченность: первой будем считать тот узел, который лежит на первом главном ребре, выходящем из X , а остальные перечисляются в соответствии с выходящими из X ребрами по часовой стрелке. Номер вершины в ее цепи будем называть *указателем*.

Для выбранной вершины X можно построить несколько цепей с центром X (в каждой будут макроплитки одного размера). Каждой цепи можно присвоить уровень: для самой крупной цепи – нулевой (0-цепь), далее первый и так далее. Если узел X принадлежит некоторой макроплитке T , то ребра цепи уровня k принадлежат макроплитке, получающейся после $k + 1$ разбиений T .

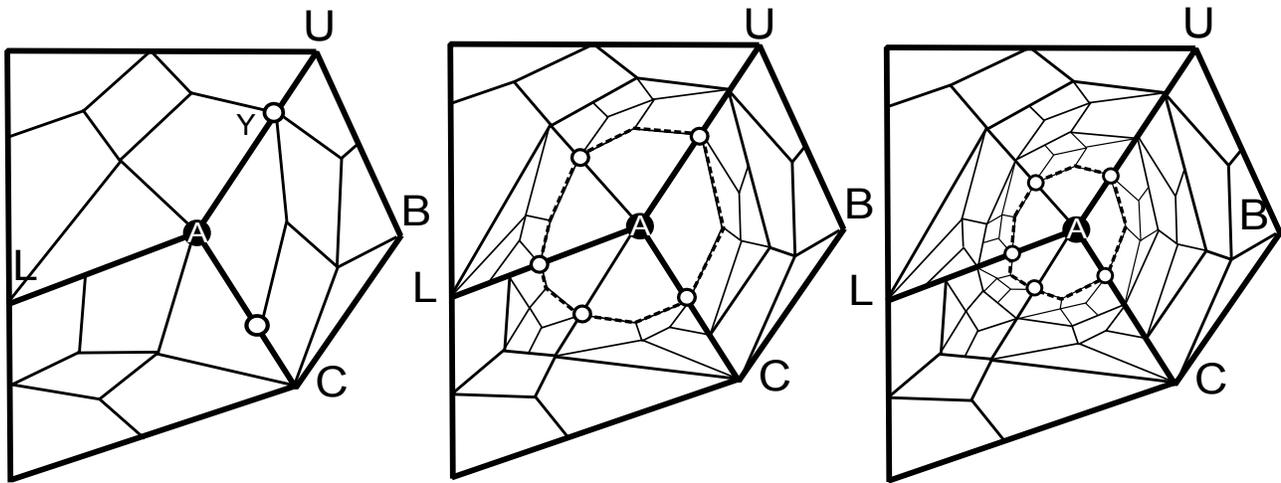


Рис. 16. Цепи узла А.

На рисунке показаны цепи нулевого, первого и второго уровней с центром типа А.

Параметр “окружение” для узлов.

Определение 5.2. Для каждой макроплитки можно записать упорядоченную четверку типов ребер, на котором лежат ее стороны: левая, верхняя, правая, нижняя (строго в данном порядке). Такую четверку будем называть *окружением* данной макроплитки.

Рассмотрим некоторую вершину X . Она может входить в одну или несколько подклеенных областей. Рассмотрим одну из таких подклеенных областей P .

Определение 5.3. *Меткой вершины относительно P* будем называть:

1. Для регулярных внутренних вершин (типы А, В, С) – окружение макроплитки, которой принадлежит данная вершина.

Пример: У узла типа А внутри средней подплитки метка $[1B, 2B, 4B, 5B]$.

2. Для регулярных боковых вершин (типы UL/LU, UR/RU, LD/DL, RD/DR) – упорядоченную пару окружений двух макроплиток, в середине сторон которых лежит данная вершина.

Пример: У узла Y на левой части рисунка 16 метка $[x, y, 1A, 3A] - [1B, 2B, 4B, 5B]$, где x, y – типы левой и верхней сторон всей макроплитки рисунка.

3. Для угловых вершин – метка будет (left,top,right,bottom), то есть как окружение подклеенной макроплитки.

4. Для краевых вершин – метка соответствует окружению макроплитки, в середине стороны которой лежит вершина.

Замечание. У вершины может быть несколько меток – по числу подклеенных областей, в которые она входит. Ясно, что только в своей базовой плоскости вершина может быть регулярной. В подклеенных областях она будет обязательно угловой или краевой.

Определение 5.4. *Окружением цепи* будем называть упорядоченный набор меток всех вершин цепи.

Пример. У цепи в средней части рисунка 16 окружение

$$\begin{aligned} & [1A, 8A, 5A, 6B] : [2B, 4B, 1A, 3A] \times [2B, 4B, 1A, 3A] : [8B, 4A, 4A, 7B] \times \\ & \times [8B, 4A, 4A, 7B] : [3B, 8A, 5A, 6B] \times [3B, 8A, 5A, 6B] : [8B, 3A, 3A, 7B] \times \\ & \times [8B, 3A, 3A, 7B] : [1A, 8A, 5A, 6B] \end{aligned}$$

Ниже будет показано, что тип узла в центре цепи уровня выше нулевого полностью определяет окружение этой цепи.

Определение 5.5. *Окружением вершины* относительно плоскости P будем называть:

1. Для регулярных боковых вершин типов UL/LU , UR/RU , LD/DL , а также для краевых вершин типов U и L – упорядоченную пару (x, y) , где x – окружение цепи, в которую входит данная вершина, y – указатель, номер данной вершины в ее цепи.

2. Для регулярных боковых вершин типов RD/DR – тип ребра, на которой находится данная вершина (это ребро 2, 3, 5 или 6).

3. Для всех остальных вершин (угловые, краевые, регулярные типов A , B , C) – метку этой вершины относительно плоскости P .

Таким образом, если вершина входит в цепь, то зная окружение вершины, мы знаем окружения всех вершин в этой цепи. Окружение вершины X относительно плоскости P будем обозначать как $\text{SurrP}(X)$. Если из контекста ясно, о какой плоскости идет речь, будем использовать обозначение $\text{Surr}(X)$.

Пусть X – узел, входящий в некоторую цепь. Если Y – центр этой цепи, то X соответствует одному из выходящих из Y ребер. Номер этого ребра, (начиная с первого выходящего из X) мы называем *указателем* для вершины X . Очевидно, что по указателю и окружению цепи можно установить окружение самого узла.

Определение 5.6. И наконец, определим *расширенное окружение* относительно области P . Это упорядоченная пара окружения относительно P и окружения относительно базовой плоскости вершины.

Таким образом, расширенное окружение несет информацию не только об окрестности вершины в подклейке, но и об окрестности в базовой плоскости.

5.3. Параметр “информация” для узлов. Начальники.

Определим понятие *начальников* для регулярных вершин. Пусть вершина X – регулярная (то есть не угловая, не боковая). Тогда можно выбрать такую минимальную макроплитку T , что X находится внутри (не на границе) T . В этом случае, X либо является внутренней вершиной (A, B или C), либо лежит на одном из восьми внутренних ребер T .

Определение 5.7. *Первым начальником* вершины X будем называть узел в середине верхней стороны макроплитки T .

Для боковых вершин, лежащих на ребрах 2, 5, 6 макроплитки T , *вторым начальником* будем называть узел в правом нижнем углу макроплитки T .

Для боковых вершин, лежащих на ребрах 7 и 8, а также для для вершин типа C определим *второго* и *третьего начальников* как узлы в левом нижнем и правом нижнем углах T соответственно.

У краевых вершин начальников не будет. Из угловых вершин определим начальника только для CDR-узла, и им будет CDL-узел в той же макроплитке.

Теперь дадим определение информации. Оно будет зависеть от расположения вершины внутри макроплитки.

Определение 5.8. Для регулярных боковых вершин, лежащих на внутренних ребрах 1, 3, 4, а также для A-узлов *информацией вершины* будем называть упорядоченную тройку состоящую из типа, уровня и расширенного окружения ее единственного начальника.

Для регулярных боковых вершин, лежащих на внутренних ребрах 2, 5, 6, а также для B-узлов *информацией вершины* будем называть упорядоченную четверку состоящую из типа, уровня и расширенного окружения ее первого начальника, а также типа ее второго начальника.

Для регулярных боковых вершин, лежащих на внутренних ребрах 7, 8, а также для C-узлов *информацией вершины* будем называть упорядоченную девятку состоящую из типов, уровней и расширенных окружения всех трех ее начальников.

У краевых вершин информации не будет.

Из угловых вершин информацию определим только для CDR-узла, и она будет равна упорядоченной тройке состоящей из типа, уровня и расширенного окружения ее единственного начальника, CDL-узла в той же макроплитке.

Обозначения. Информацию о первом начальнике узла X будем обозначать как $FBoss(X)$. Иногда это обозначение будет употребляться для типа или окружения начальника, тогда это будет понятно по контексту. Аналогичны обозначения для второго и третьего начальников: $SBoss(X)$, $TBoss(X)$.

Таким образом, информация – это данные о начальниках, а для боковых вершин на ребрах 2, 5, 6 – это еще знание, какой тип у узла в правом нижнем углу.

5.4. Ядро подклейки и флаг макроплитки.

Рассмотрим некоторую макроплитку T , подклееную к ребрам e_1 и e_2 некоторой вершины X . Вершину X будем называть *ядром подклейки*. Она лежит в левом верхнем углу T . Сочетание значений параметров типа, базового окружения, информации X , а также упорядоченную пару типов выходящих ребер e_1 и e_2 будем называть *флагом подклейки* для макроплитки T .

Для каждой вершины Y , лежащей внутри подклеенной макроплитки T (не на ее краю) определим параметр *флаг подклейки*, со значением, равным значению этого параметра для макроплитки. Таким образом, для всех таких вершин Y , лежащих внутри подклеенной макроплитки значение этого параметра будет одинаковым.

5.5. Кодировка путей. Рассмотрим путь $X_1e_1X_2e_2X_3e_3 \dots e_{n-1}X_n$, где X_i – вершины, e_i – ребра между соседними вершинами в пути. Обозначим $\text{In}(e_i)$ – тип e_i как выходящего ребра из вершины X_i . Также пусть $\text{Out}(e_i)$ – тип e_i как входящего ребра в вершину X_{i+1} .

Рассмотрим некоторую вершину X_i . Если и входящее в X_i ребро e_{i-1} , и выходящее ребро e_i являются плоскими, то кодом X_i будем считать упорядоченную четверку, состоящую из типа X_i , ее уровня, ее окружения и ее информации.

Если одно из ребер, например входящее ребро e_{i-1} приходит из подклеенной области P_{i-1} , а выходящее ребро e_i – плоское, то кодом X_i будем считать упорядоченный набор из четырех элементов:

- (1) *первый элемент* – упорядоченная пара из типа X_i в области P_{i-1} и типа X_i в базовой плоскости;
- (2) *второй элемент* – упорядоченная пара из уровня X_i в области P_{i-1} и уровня X_i в базовой плоскости;
- (3) *третий элемент* – упорядоченная пара из окружения X_i в области P_{i-1} и окружения X_i в базовой плоскости;
- (4) *четвертый элемент* – информация у X_i .

Таким образом кодировка учитывает и положение вершины в макроплитке, откуда пришло ребро и в макроплитке базовой плоскости. Для случая, когда входящее ребро – плоское, а выходящее ведет в подклейку, определение аналогично, только в упорядоченной паре (типов, уровней, окружений) сначала будет идти базовая плоскость, а потом подклеенная.

Если оба ребра, и входящее e_{i-1} , и выходящее e_i , ведут в подклеенные области P_{i-1} и P_i , то кодом X_i будем считать упорядоченный набор из четырех элементов:

- (1) *первый элемент* – упорядоченная тройка – тип X_i в области P_{i-1} , тип X_i в базовой плоскости, тип X_i в области P_i ;
- (2) *второй элемент* – упорядоченная тройка – уровень X_i в области P_{i-1} , уровень X_i в базовой плоскости, уровень X_i в области P_i ;
- (3) *третий элемент* – упорядоченная тройка – окружение X_i в области P_{i-1} , окружение X_i в базовой плоскости, окружение X_i в области P_i ;
- (4) *четвертый элемент* – информация у X_i .

Замечание. Как видно, информация не подчиняется общей логике. Это происходит по причине того, что любая вершина в своих подклеенных макроплитках всегда занимает место на верхней или левой сторонах, либо в лежит в любом углу, кроме правого нижнего. Во всех этих случаях информация вершины в рамках такой подклеенной макроплитки, по определению, пустая.

Код вершины X будем обозначать как $\text{Code}(X)$. Теперь определим код всего пути.

Определение 5.9. Кодом пути $X_1e_1X_2e_2X_3e_3 \dots e_{n-1}X_n$ будем называть упорядоченный набор

$$\text{Code}(X_1), \text{Out}(e_1), \text{In}(e_1), \text{Code}(X_2), \text{Out}(e_2), \dots, \text{In}(e_{n-1}), \text{Code}(X_n).$$

Здесь $\text{Code}(X_i)$ – код вершины X_i , $\text{Out}(e_i)$ – тип e_i как выходящего ребра из вершины X_i , $\text{In}(e_i)$ – тип e_i как входящего ребра в вершину X_{i+1} .

Нашей основной целью будет определить локальные преобразования кодов путей так, чтобы переход от одного кода к другому соответствовал переходу к эквивалентному пути на комплексе.

При этом элементарным преобразованиям путей будут соответствовать замены одних слов-кодов путей на другие. То есть элементарные преобразования путей являются аналогами определяющих соотношений в полугруппе слов-кодов.

Для того, чтобы задать такие определяющие соотношения, нужно рассмотреть все локальные преобразования путей и для каждого из них указать, какой паре равных кодов соответствует данное локальное преобразование.

Часто знание типа, окружения или информации некоторой вершины помогает нам понять устройство ближайшей окрестности. Для удобства, мы зададим набор функций на узлах, аргументом которых являются типы, окружения или информации некоторых специально расположенных узлов, а значениями – окружения или типы узлов в окрестности узла-аргумента. В следующем параграфе мы определим набор таких функций.

6. ФУНКЦИИ НА УЗЛАХ И СТРУКТУРА ЦЕПЕЙ

Ниже приведены функции, которые облегчают нам работу по восстановлению параметров вершин.

- (1) `TopFromCorner` Значение: окружение узла в середине верхней стороны T .
- (2) `RightCorner` Значение: окружение узла в правом нижнем углу T .
- (3) `TopRightType` Значение: тип узла в правом верхнем углу T , а также тип ребра для боковых узлов.
- (4) `BottomLeftType` Значение: тип узла в левом нижнем углу T , а также тип ребра для боковых узлов.
- (5) `LevelPlus` Значение: окружение цепи с уровнем на один больше, чем у X , с тем же центром и указателем.
- (6) `BottomRightTypeFromRight` Значение: тип узла в правом нижнем углу T , а также тип ребра для боковых узлов.
- (7) `TopFromRight` Значение: окружение узла в середине верхней стороны T .

Функции 1 и 2 содержат два аргумента: первый – это окружение узла X , являющимся левым нижним углом в макроплитке T , второй – тип ребра выхода из X , соответствующего нижней стороне T .

Функции 3, 4, 5 содержат один аргумент: окружение узла X , являющегося серединой верхней стороны в макроплитке T

Функции 6 и 7 содержат два аргумента: окружение **и информацию** узла X , являющимся серединой правой стороны в макроплитке T .

Еще раз отметим, что под окружением узла в цепи мы понимаем окружение всей этой цепи вместе с указателем – типом ребра входа-выхода, на котором лежит узел из цепи.

6.1. Свойства цепей. В этом параграфе мы опишем свойства цепей для различных типов узлов. В частности, мы установим, что зная тип центра, можно узнать окружение цепи, и наоборот. Помимо этого, мы используем свойства цепей для доказательства того, что по аргументу функции можно узнать ее значение.

Мы разберем цепи с различными центрами и в каждом случае будем проверять, что если мы знаем аргумент функции, то можем узнать ее значение.

Цепи с центром в узле типа \mathbb{A} . На рисунке 17 белыми точками отмечены цепи узла \mathbb{A} нулевого, первого и второго уровней. Заметим, что на втором уровне все пять макроплиток, левый верхний угол которых попадает в узел \mathbb{A} , занимают левое верхнее положение в своих родительских макроплитках. Значит, если взять следующий уровень подразделения, типы ребер-границ макроплиток не изменятся. То есть окружения вершин

в цепях будут те же, то есть, окружение цепи третьего и последующих уровней совпадает с окружением цепи второго уровня.

Таким образом, существует только три возможных окружения цепи с центром в узле типа \mathbb{A} . Заметим, что по окружению цепи, мы можем установить, является ли центр цепи узлом типа \mathbb{A} , и какого уровня цепь. Действительно, одна из макроплиток в цепи с центром в \mathbb{A} обязательно имеет окружение $(4\mathbb{B}, 1\mathbb{B}, 2\mathbb{B}, 5\mathbb{B})$ или $(4\mathbb{B}, 1\mathbb{B}, 1\mathbb{A}, 3\mathbb{A})$, причем, макроплитки с такими окружениями не могут входить в цепи с другими центрами. Цепь нулевого уровня содержит макроплитку с окружением $(4\mathbb{B}, 1\mathbb{B}, 2\mathbb{B}, 5\mathbb{B})$ (остальные – нет). Для цепи второго уровня все четверки типов ребер имеют вид $(x, y, 1\mathbb{A}, 3\mathbb{A})$, а для первого уровня это не так.

Забегая вперед, можно сказать, что возможность установить тип центра по окружению цепи будет и для центров других типов (не только \mathbb{A}).

Заметим также, что мы можем выписать полностью все окружение 1-цепи и 2-цепи с центром в узле типа \mathbb{A} , а зная окружение узла \mathbb{A} , можем выписать и окружение 0-цепи.

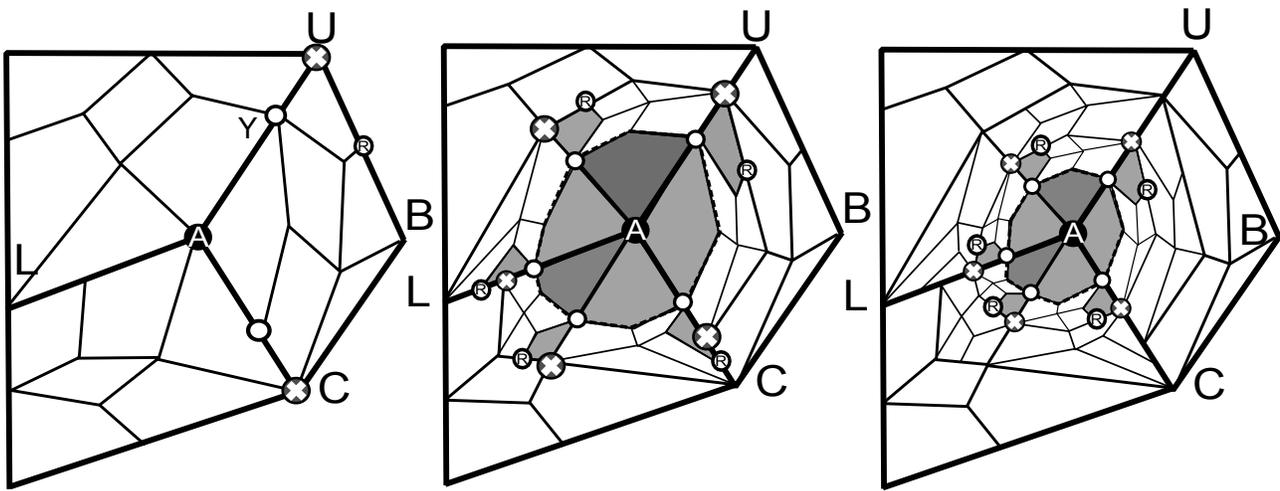


Рис. 17. Цепи узла \mathbb{A} .

Из сказанного выше следует, что для \mathbb{A} -узлов значение функции `LevelPlus` может быть установлено по ее аргументу.

На рисунке 17 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел цепи X . Можно убедиться в том, что для цепей первого и второго уровней все типы узлов с крестами, а также типы ребер, на которых они лежат, мы можем выписать, для каждого заданного X . Для 0-цепи мы можем выписать типы узлов со знаком “ \otimes ” (а также типы ребер), если знаем окружение центра (узла \mathbb{A}). Таким образом, мы можем вычислить функции `TopRightType` и `BottomLeftType`.

Серым цветом на рисунке выделены плитки, где левый нижний угол (при дальнейшем разбиении) будет попадать в вершину цепи. Можно заметить, что зная эту вершину (ее место в цепи) мы можем установить и окружение правого нижнего угла в соответствующей макроплитке. Кроме того, мы также можем установить тип левого верхнего угла, а значит и всю цепь, содержащую середину верхней стороны в соответствующей макроплитке. Все это значит, что мы можем вычислить функции `TopFromCorner`, `RightCorner`.

Функции `TopFromRight` и `BottomRightTypeFromRight` могут быть применимы только в одном случае: для 0-цепи, если аргументом является узел в середине ребра типа 1 (обозначим этот узел как Y , а макроплитку, где Y является серединой правой стороны, как T). `BottomRightTypeFromRight` в этом случае это просто наш узел A .

Для вычисления `TopFromRight` нам надо найти окружение узла в середине верхней стороны макроплитки T . Заметим, что этот узел входит в цепь с центром в левом верхнем углу T , причем уровень этой цепи на один выше, чем уровень цепи, куда входит правый верхний угол T . Окружение этой цепи мы можем установить по информации Y , так как правый верхний угол T является серединой верхней стороны в макроплитке T' , родительской к T и является первым начальником узла Y . По окружению этой цепи можно установить тип левого верхнего угла T и окружение цепи на один уровень выше, что и требуется.

Таким образом, мы вычислили значения всех функций для аргументов, входящих в A -цепи.

Цепи с центром в узле типа B . На рисунке 18 белыми точками отмечены цепи узла B . 0-цепи с центром в B не существует (все боковые узлы в соответствующей окрестности имеют тип DR). Заметим, что в 2-цепи все шесть макроплиток, левый верхний угол которых попадает в узел B , занимают левое верхнее положение в своих родительских макроплитках. Значит, если взять следующий уровень подразделения, типы ребер-границ макроплиток не изменятся. То есть окружения вершин в цепях будут те же, то есть, окружение цепи третьего и последующих уровней совпадает с окружением цепи второго уровня.

Таким образом, существует только два возможных окружения цепи с центром в узле типа B . Заметим, что по окружению цепи, мы можем установить, является ли центр цепи узлом типа B , и какого уровня цепь. Действительно, 1-цепь с центром в B содержит узел с окружением $(8B, 5B, 5B, 7B)$, который не может содержаться ни в какой другой цепи. 2-цепь с центром в B содержит узел с окружением $(8B, 5B, 1A, 3A)$, который также ни в какой другой цепи не встречается.

Заметим также, что мы можем выписать полностью все окружение 1-цепи и 2-цепи с центром в узле типа B .

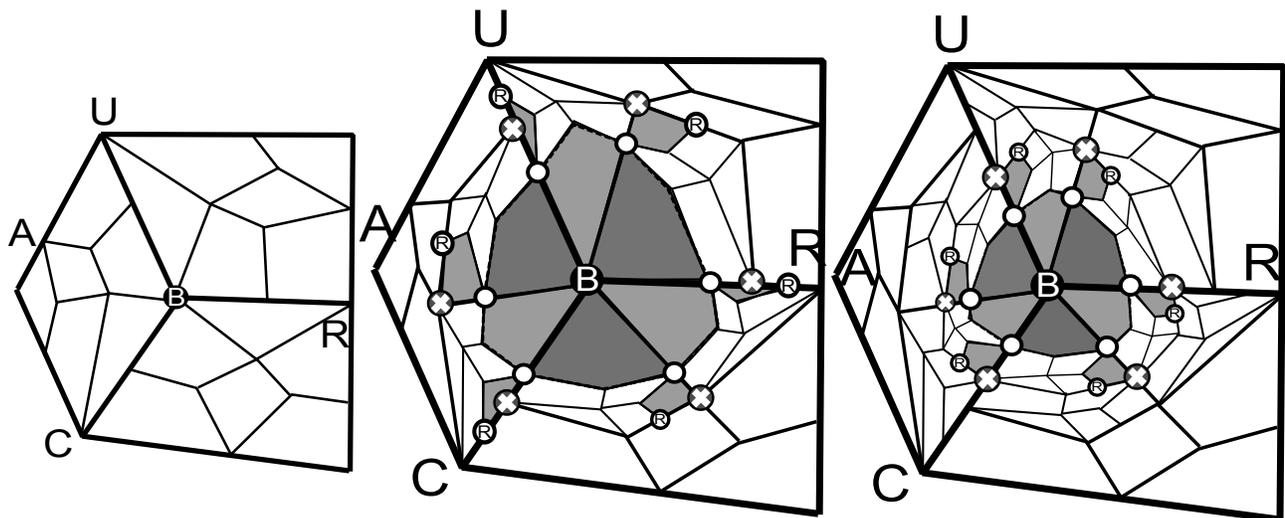


Рис. 18. Цепи узла B .

Из сказанного выше следует, что для \mathbb{B} -узлов значение функции `LevelPlus` может быть установлено по ее аргументу. На рисунке 18 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел цепи X . Можно убедиться в том, что для цепей первого и второго уровней все типы узлов с крестами, а также типы ребер, на которых они лежат, мы можем выписать, для каждого заданного X . Таким образом, мы можем вычислить функции `TopRightType` и `BottomLeftType`.

Серым цветом на рисунке выделены плитки, где левый нижний угол (при дальнейшем разбиении) будет попадать в вершину цепи. Можно заметить, что зная эту вершину (ее место в цепи мы знаем, благодаря второму аргументу – нижнему ребру) мы можем установить и окружение правого нижнего угла в соответствующей макроплитке. Кроме того, мы также можем установить тип левого верхнего угла, а значит и всю цепь, содержащую середину верхней стороны в соответствующей макроплитке. Все это значит, что мы можем вычислить функции `TopFromCorner`, `RightCorner`.

Функции `TopFromRight` и `BottomRightTypeFromRight` не могут быть применимы, то есть аргументом этих функций никогда не может быть узел из \mathbb{B} -цепи так как ни один узел из цепи с центром в \mathbb{B} не является серединой правой стороны ни в какой макроплитке.

Таким образом, мы вычислили значения всех функций для аргументов, входящих в \mathbb{B} -цепи.

Цепи с центром в узле типа \mathbb{C} . На рисунках 19 и 20 белыми точками отмечены 1-цепи и 2-цепи узла \mathbb{C} . 0-цепи с центром в \mathbb{C} не существует.

Кроме указанных на рисунках цепей первого и второго уровней существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рисунке 20.

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел \mathbb{C} , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразбиения, типы ребер-границ макроплиток не изменятся. То есть окружения вершин в цепях будут те же, то есть, окружение цепи четвертого и последующих уровней совпадает с окружением цепи третьего уровня.

Таким образом, существует только три возможных окружения цепи с центром в узле типа \mathbb{C} (для цепей первого, второго и третьего уровней). Заметим, что по окружению цепи, мы можем установить, является ли центр цепи узлом типа \mathbb{C} , и какого уровня цепь. Действительно, 1-цепь с центром в \mathbb{C} содержит узел с окружением $(8A, 5A, 1A, 3A) - (8B, 5B, 5B, 7B)$ (на ребре 5), который не может содержаться ни в какой другой цепи. 2-цепь с центром в \mathbb{C} содержит узел с окружением $(8A, 5A, 1A, 3A) - (5B, 8A, 5A, 6B)$, который также ни в какой другой цепи не встречается. Для 3-цепи таким узлом будет $(8A, 5A, 1A, 3A) - (5B, 8A, 1A, 3A)$.

Заметим также, что мы можем выписать полностью все окружение 1-цепей, 2-цепей, 3-цепей с центром в узле типа \mathbb{C} .

Из сказанного выше следует, что для \mathbb{C} -узлов значение функции `LevelPlus` может быть установлено по ее аргументу. На рисунках 19 и 20 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел цепи X . Можно убедиться в том, что для цепей первого, второго и третьего уровней все типы узлов с крестами, а также типы ребер, на которых они лежат, мы можем выписать, для каждого заданного X . Таким образом, мы можем вычислить функции `TopRightType` и `BottomLeftType`.

Серым цветом на рисунке выделены плитки, где левый нижний угол (при дальнейшем разбиении) будет попадать в вершину цепи. Можно заметить, что зная эту вершину (ее

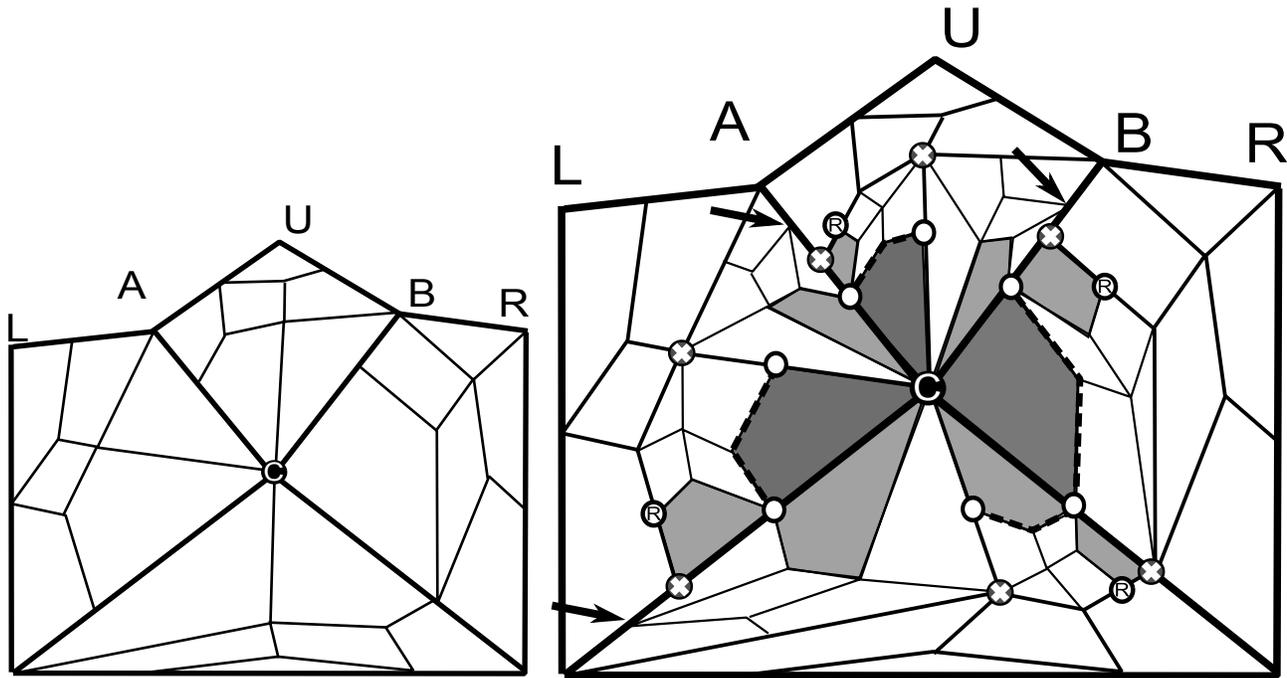


Рис. 19. 1-цепь узла C .

место в цепи) мы можем установить и окружение правого нижнего угла в соответствующей макроплитке. Кроме того, мы также можем установить тип левого верхнего угла, а значит и всю цепь, содержащую середину верхней стороны в соответствующей макроплитке. (Все это также очевидно проверяется для 3-цепи.) Все это значит, что мы можем вычислить функции $TopFromCorner$, $RightCorner$.

Функции $TopFromRight$ и $BottomRightTypeFromRight$ могут быть применимы только к 1-цепи (в остальные входят только узлы типов UL и LU). В случае 1-цепи аргументом могут быть узлы, лежащие на ребрах 1, 2 и 4 (относительно C). Пусть X такой узел. Во всех случаях $BottomRightTypeFromRight$ будет C . Узлы, на которые указывает $TopFromRight$ отмечены стрелками на рисунке 19. Таким образом, $TopFromRight$ принимает значения: 1-цепи вокруг A для 1 ребра (с указателем 2), 1-цепи вокруг B для ребра 2 (с указателем 3), 2-цепь вокруг $SBoss(X)$ для ребра 4 (с указателем, соответствующим входу ребра 7 в левый нижний угол (см параграф 6.2 “Указатели”).

Таким образом, мы вычислили значения всех функций для аргументов, входящих в C -цепи.

Цепи с центром в узлах типа UL и LU . Случаи UL и LU узлов симметричны, для определенности будем далее разбирать случай UL узла. Будем считать, что главное ребро имеет тип t , то есть с U -стороны это tA , а с L -стороны tB . На рисунках 21 и 22 белыми точками отмечены 1-цепи и 2-цепи узла UL . 0-цепи с центром в UL не существует. Кроме указанных на рисунках цепей первого и второго уровней существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рисунке 22.

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел UL , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразбиения, типы ребер-границ макроплиток не изменятся. То есть окружения вершин в цепях будут те же, то есть, окружение цепи четвертого и последующих уровней совпадает с окружением цепи третьего уровня.

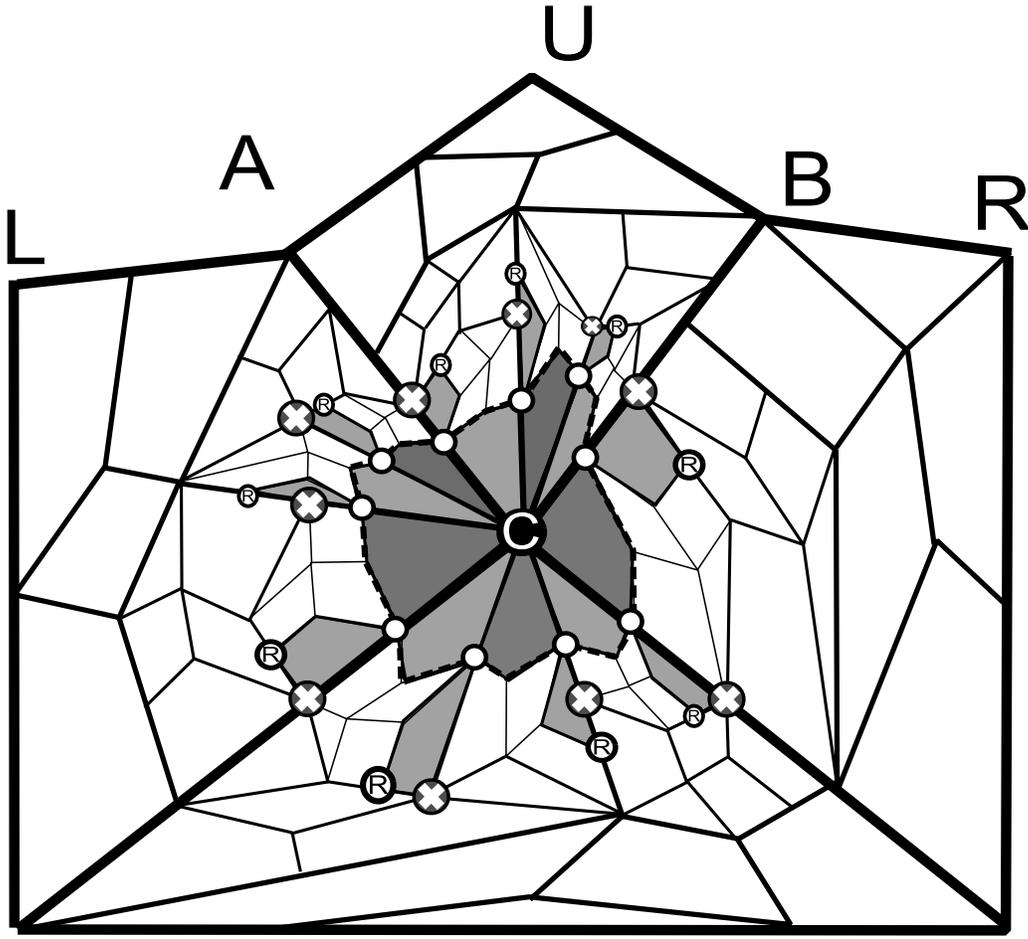


Рис. 20. 2-цепь узла C.

Таким образом, существует только три возможных конфигурации окружения цепи с центром в узле типа UL (для цепей первого, второго и третьего уровней, при этом еще может быть выбрано разное главное ребро t). Заметим, что по окружению цепи, мы можем установить, является ли центр цепи узлом типа UL , какого уровня цепь, а также сам параметр t . Действительно, 1-цепь с центром в UL содержит узлы с окружением $(tA, 1A, 6A, 2A) - (1B, 2B, 6A, 2A)$ и $(8B, 3A, 3A, 7B) - (tB, 3B, 6A, 2A)$, которые вместе ни в какой другой цепи не встречаются. 2-цепь с центром в UL содержит одновременно узлы с окружением $(tA, 1A, 1A, 3A) - (1B, 2B, 1A, 3A)$ и $(3A, 8A, 5A, 6B) - (tB, 3B, 1A, 3A)$, которые вместе также ни в какой другой цепи не встречаются. Для 3-цепи такой парой будет $(tA, 1A, 1A, 3A) - (1B, 2B, 1A, 3A)$ и $(3A, 8A, 1A, 3A) - (tB, 3B, 1A, 3A)$

Заметим также, что зная тип ребра t , мы можем выписать полностью все окружение 1-цепей, 2-цепей, 3-цепей с центром в узле типа UL .

Из сказанного выше следует, что для UL -узлов значение функции `LevelPlus` может быть установлено по ее аргументу.

На рисунках 21 и 22 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел цепи X . Можно убедиться в том, что для цепей первого, второго и третьего уровней все типы узлов с крестами, а также типы ребер на которых они лежат, мы можем выписать, для каждого заданного X . Таким образом, мы можем вычислить функции `TopRightType` и `BottomLeftType`.

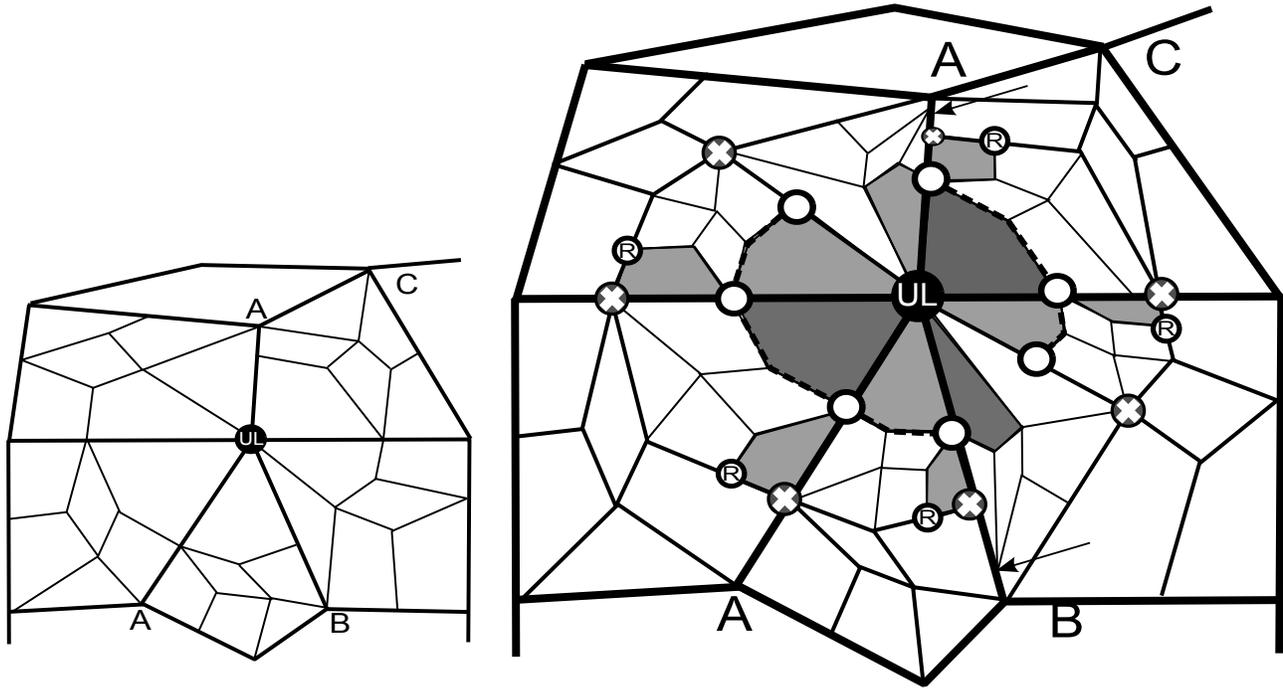


Рис. 21. 1-цепь узла UL.

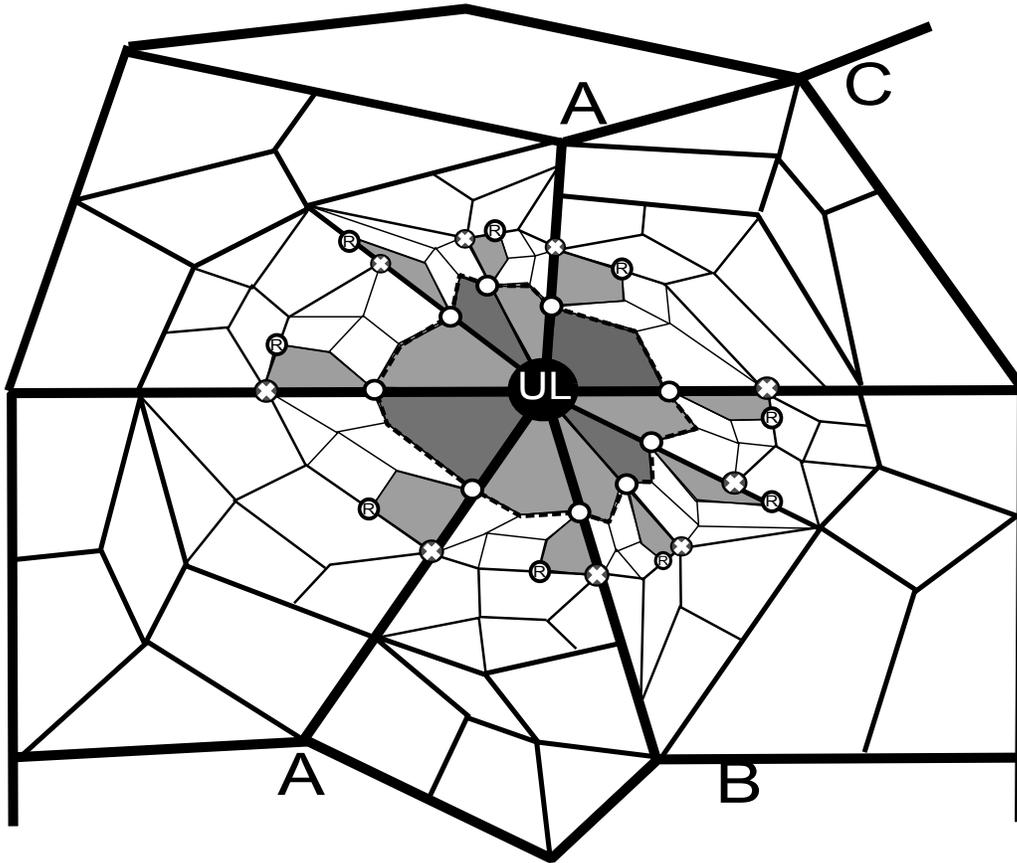
Серым цветом на рисунке выделены плитки, где левый нижний угол (при дальнейшем разбиении) будет попадать в вершину цепи. Можно заметить, что зная эту вершину (ее место в цепи) мы можем установить и окружение правого нижнего угла в соответствующей макроплитке. Кроме того, мы также можем установить тип левого верхнего угла, а значит и всю цепь, содержащую середину верхней стороны в соответствующей макроплитке. (Все это также очевидно проверяется для 3-цепи.) Все это значит, что мы можем вычислить функции `TopFromCorner`, `RightCorner`.

Функции `TopFromRight` и `BottomRightTypeFromRight` могут быть применимы только к 1-цепи (в остальные входят только узлы типов UL и LU). В случае 1-цепи аргументом могут быть узлы, лежащие на ребрах u_1 и l (относительно центрального UL-узла). Пусть X такой узел. Во всех случаях `BottomRightTypeFromRight` будет наш же узел UL. Узлы, на которые указывает `TopFromRight` отмечены стрелками на рисунке 21. Таким образом, `TopFromRight` принимает значения: 1-цепи вокруг B (указатель 1) для ребра u_1 , 1-цепи вокруг A (указатель 3) для ребра l .

Таким образом, мы вычислили значения всех функций для аргументов, входящих в UL-цепи.

Цепи с центром в узлах типа UR и RU. Случаи узлов UR и RU симметричны, для определенности будем далее разбирать случай UR узла. Будем считать, что главное ребро имеет тип t , то есть с U -стороны это tA , а с R -стороны tB .

На рисунках 23 и 24 белыми точками отмечены 1-цепи и 2-цепи узла UR. 0-цепи с центром в UR не существует. Кроме указанных на рисунках цепей первого и второго уровней существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рисунке 24.

Рис. 22. 2-цепь узла UL .

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел UR , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразделения, типы ребер-границ макроплиток не изменятся. То есть окружения вершин в цепях будут те же, то есть, окружение цепи четвертого и последующих уровней совпадает с окружением цепи третьего уровня.

Таким образом, существует только три возможных конфигурации окружения цепи с центром в узле типа UR (для цепей первого, второго и третьего уровней, при этом еще может быть выбрано разное главное ребро t). Заметим, что по окружению цепи, мы можем установить, является ли центр цепи узлом типа UR , какого уровня цепь, а также сам параметр t . Действительно, 1-цепь с центром в UR содержит одновременно узлы с окружением $(tA, 1A, 6A, 2A) - (1B, 2B, 6A, 2A)$ и $(tB, 6A, 6A, 2A) - (8B, 6B, 6B, 7B)$, которые вместе ни в какой другой цепи не встречаются. 2-цепь с центром в UR содержит одновременно узлы с окружением $(tA, 1A, 1A, 3A) - (1B, 2B, 1A, 3A)$ и $(tB, 6A, 1A, 3A) - (6B, 8A, 5A, 6B)$, которые вместе также ни в какой другой цепи не встречаются. Для 3-цепи такой парой будет $(tA, 1A, 1A, 3A) - (1B, 2B, 1A, 3A)$ и $(tB, 6A, 1A, 3A) - (6B, 8A, 1A, 3A)$

Заметим также, что зная тип ребра t , мы можем выписать полностью все окружение 1-цепей, 2-цепей, 3-цепей с центром в узле типа UR .

Из сказанного выше следует, что для UR -узлов значение функции $LevelPlus$ может быть установлено по ее аргументу.

На рисунках 23 и 24 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел

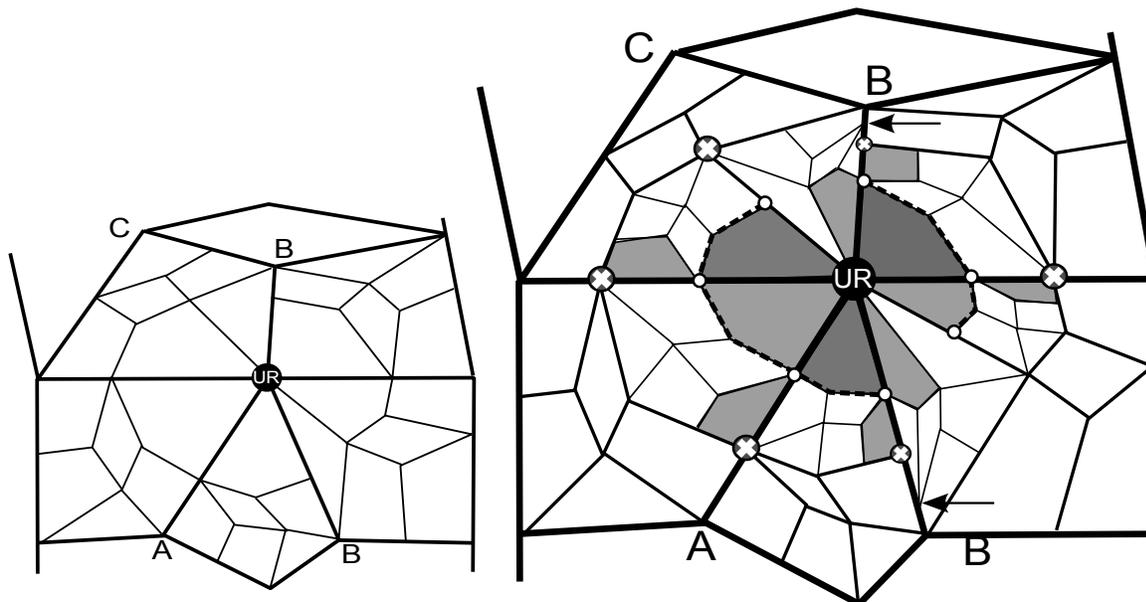


Рис. 23. 1-цепь узла UR.

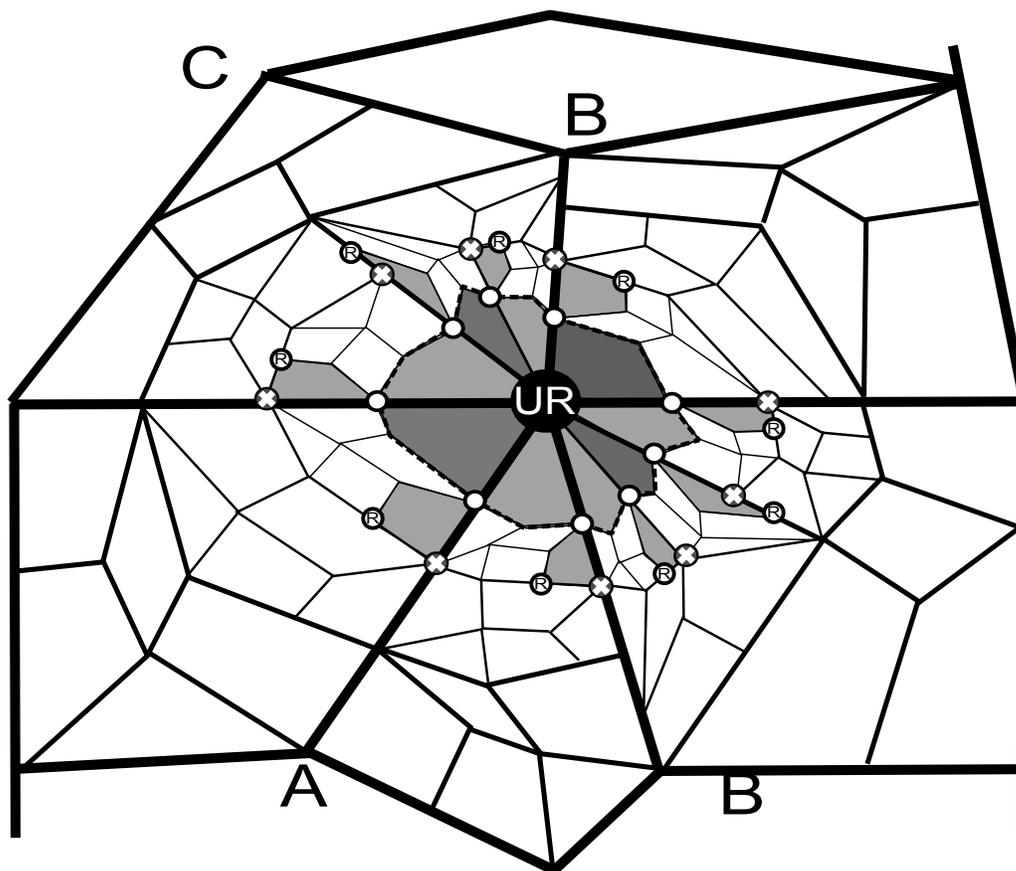


Рис. 24. 2-цепь узла UR.

цепи X . Можно убедиться в том, что для цепей первого, второго и третьего уровней все типы узлов с крестами, а также типы ребер на которых они лежат, мы можем выписать для каждого заданного X . Таким образом, мы можем вычислить функции `TopRightType` и `BottomLeftType`.

Серым цветом на рисунке выделены плитки, где левый нижний угол (при дальнейшем разбиении) будет попадать в вершину цепи. Можно заметить, что зная эту вершину (ее место в цепи) мы можем установить и окружение правого нижнего угла в соответствующей макроплитке. Кроме того, мы также можем установить тип левого верхнего угла, а значит и всю цепь, содержащую середину верхней стороны в соответствующей макроплитке. (Все это также очевидно проверяется для 3-цепи.) Все это значит, что мы можем вычислить функции `TopFromCorner`, `RightCorner`.

Функции `TopFromRight` и `BottomRightTypeFromRight` могут быть применимы только к 1-цепи (в остальные входят только узлы типов `UL` и `LU`). В случае 1-цепи аргументом могут быть узлы, лежащие на ребрах u_1 и r (относительно `UR`). Пусть X такой узел. Во всех случаях `BottomRightType` будет наш же узел `UR`. Узлы, на которые указывает `TopFromRight` отмечены стрелками на рисунке 23. Таким образом, `TopFromRight` принимает значения: 1-цепи вокруг \mathbb{B} (указатель 1) для u_1 ребра, 1-цепи вокруг \mathbb{B} (указатель 2) для r ребра.

Таким образом, мы вычислили значения всех функций для аргументов, входящих в `UR`-цепи.

Цепи с центром в узлах типа `DL` и `LD`. Случаи узлов `DL` и `LD` симметричны, для определенности будем далее разбирать случай `DL` узла. Будем считать, что главное ребро имеет тип t , то есть с D -стороны это tA , а с L -стороны tB . На рисунках 25 и 26 белыми точками отмечены 1-цепи и 2-цепи узла `DL`. 0-цепи с центром в `DL` не существует. Кроме указанных на рисунках цепей первого и второго уровней существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рисунке 26.

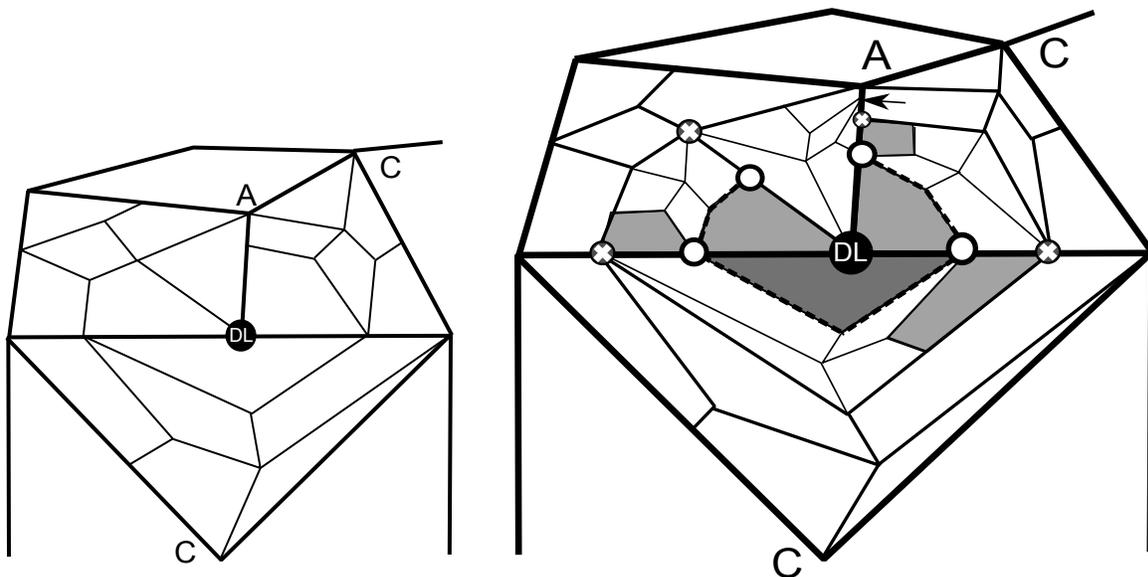


Рис. 25. 1-цепь узла `DL`.

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел \mathbb{DL} , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразделения, типы ребер-границ макроплиток не изменятся. То есть окружения вершин в цепях будут те же, то есть, окружение цепи четвертого и последующих уровней совпадает с окружением цепи третьего уровня.

Таким образом, существует только три возможных конфигурации окружения цепи с центром в узле типа \mathbb{DL} (для цепей первого, второго и третьего уровней, при этом еще может быть выбрано разное главное ребро t). Заметим, что по окружению цепи, мы можем установить, является ли центр цепи узлом типа \mathbb{DL} , какого уровня цепь, а также сам параметр t . Действительно, 1-цепь с центром в \mathbb{DL} содержит одновременно узлы с окружением $(tA, tA, 6A, 2A) - (tB, 3B, 6A, 2A)$ и $(tA, tA, 6A, 2A) - (7A, tB, 3B, 4A)$, которые вместе ни в какой другой цепи не встречаются. 2-цепь с центром в \mathbb{DL} содержит одновременно узлы с окружением $(tA, tA, 1A, 3A) - (tB, 3B, 1A, 3A)$ и $(3A, 8A, 5A, 6B) - (8B, 7B, 7B, 7B)$, которые вместе также ни в какой другой цепи не встречаются. Для 3-цепи такой парой будет $(tA, tA, 1A, 3A) - (tB, 3B, 1A, 3A)$ и $(3A, 8A, 1A, 3A) - (8B, 7B, 1A, 3A)$

Заметим также, что зная тип ребра t , мы можем выписать полностью все окружение 1-цепей, 2-цепей, 3-цепей с центром в узле типа \mathbb{DL} .

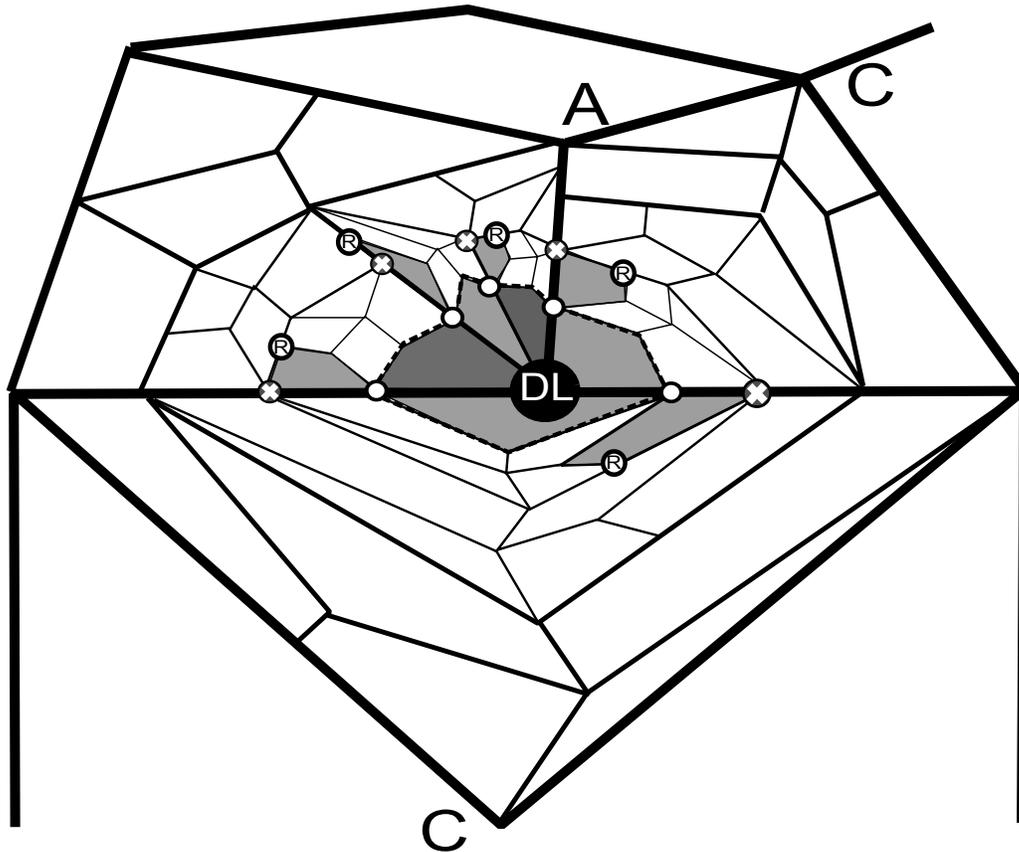


Рис. 26. 2-цепь узла \mathbb{DL} .

Из сказанного выше следует, что для \mathbb{DL} -узлов значение функции `LevelPlus` может быть установлено по ее аргументу. На рисунках 25 и 26 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел цепи X . Можно убедиться в том, что для цепей первого, второго и третьего уровней все типы узлов с крестами, а также типы ребер на которых

они лежат, мы можем выписать для каждого заданного X . Таким образом, мы можем вычислить функции `TopRightType` и `BottomLeftType`.

Серым цветом на рисунке выделены плитки, где левый нижний угол (при дальнейшем разбиении) будет попадать в вершину цепи. Можно заметить, что зная эту вершину (ее место в цепи) мы можем установить и окружение правого нижнего угла в соответствующей макроплитке. Кроме того, мы также можем установить тип левого верхнего угла, а значит и всю цепь, содержащую середину верхней стороны в соответствующей макроплитке. (Все это также очевидно проверяется для 3-цепи.) Все это значит, что мы можем вычислить функции `TopFromCorner`, `RightCorner`.

Функции `TopFromRight` и `BottomRightTypeFromRight` могут быть применимы только к 1-цепи (в остальные входят только узлы типов `UL` и `LU`). В случае 1-цепи аргументом могут быть только узел, лежащий на 1 ребре (относительно `DL` узла). Пусть X этот узел. Тогда `BottomRightTypeFromRight` будет наш же узел `DL`. Узел, на которые указывает `TopFromRight` отмечен стрелкой на рисунке 25. Таким образом, `TopFromRight` это 1-цепь вокруг A (указатель 3).

Цепи с центром в узлах типа `DR` и `RD`. На рисунках 27 и 28 белыми точками отмечены 1-цепи и 2-цепи узла `DR`. 0-цепи с центром в `DR` не существует.

Кроме указанных на рисунках цепей первого и второго уровней существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рисунке 26. Полностью аналогично цепи `DL`, можно показать, что существует только три возможных конфигурации окружения цепи с центром в узле типа `DR`. Кроме того, зная тип ребра t , мы можем выписать полностью все окружение 1-цепи, 2-цепи, 3-цепи с центром в узле типа `DR`.

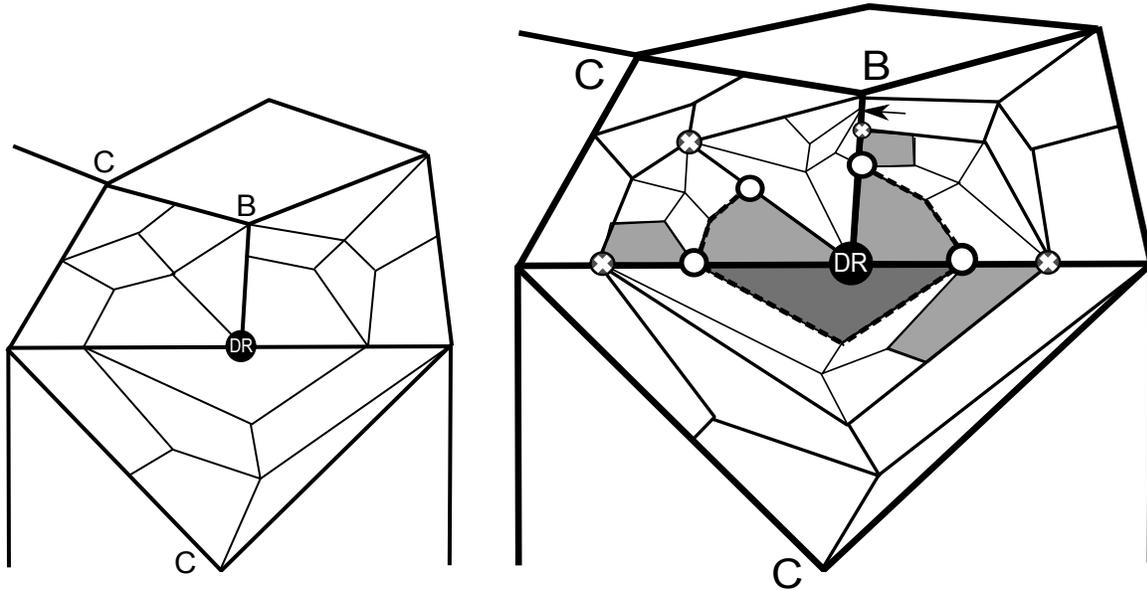


Рис. 27. 1-цепь узла `DR`.

Таким образом, что для `DR`-узлов значение функции `LevelPlus` может быть установлено по ее аргументу.

На рисунках 27 и 28 знаком “ \otimes ” отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макроплитках, где середина верхней стороны попадает в узел цепи X . Вычисление функций `TopRightType`, `BottomLeftType`, `TopFromCorner`, `RightCorner` полностью аналогично случаю `DL`-цепи.

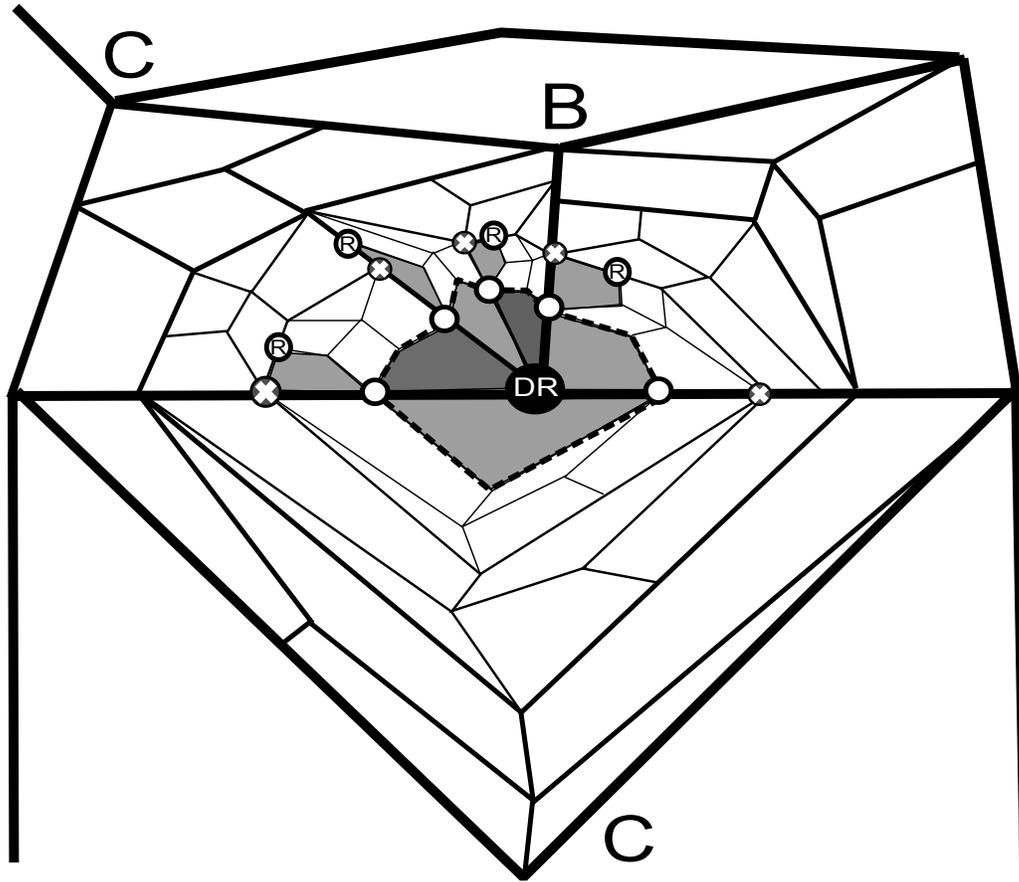


Рис. 28. 2-цепь узла \mathbb{DR} .

Функции `TopFromRight` и `BottomRightTypeFromRight` могут быть применимы только к 1-цепи (в остальные входят только узлы типов \mathbb{UL} и \mathbb{LU}). В случае 1-цепи аргументом могут быть только узел, лежащий на r ребре (относительно \mathbb{DR} узла). Пусть X этот узел. Тогда `BottomRightType` будет наш же узел \mathbb{DR} . Узел, на которые указывает `TopFromRight` отмечен стрелкой на рисунке 27. Таким образом, `TopFromRight` это 1-цепь вокруг \mathbb{B} (указатель 2).

Нам остается разобрать цепи с центрами в краевых и угловых вершинах. Важное замечание: при разборе цепи с центром в краевой вершине часть узлов в попадает на край макроплитки. Под их окружением, в данном случае, мы понимаем базовое окружение в рамках нашей макроплитки (эта макроплитка может быть подклееной и тогда вершина на ее краю одновременно имеет какое-то другое окружение в основной своей области, вот это окружение мы не рассматриваем).

С учетом этого замечания, для рассмотрения цепей с центрами в вершинах типов \mathbb{U} , \mathbb{L} , \mathbb{R} , \mathbb{D} достаточно применить те же рассуждения как и для цепей \mathbb{UL} , \mathbb{LD} , \mathbb{RD} , \mathbb{DL} , но рассматривать только половину картинку для каждой цепи. В силу полной аналогичности этого рассмотрения, мы не будем тут приводить этот разбор.

Цепи с центрами в угловых вершинах рассмотрим отдельно.

Цепи с центром в узлах типа \mathbb{CUL} , \mathbb{CUR} , \mathbb{CDR} , \mathbb{CDL} . На рисунках 29, 30, 31, 32 белыми точками отмечены цепи с вершинами в узлах типов \mathbb{CUL} , \mathbb{CUR} , \mathbb{CDR} , \mathbb{CDL} .

Аналогично предыдущим цепям, можно показать, что существует только две возможных конфигурации окружения цепи в каждом из этих четырех случаев.

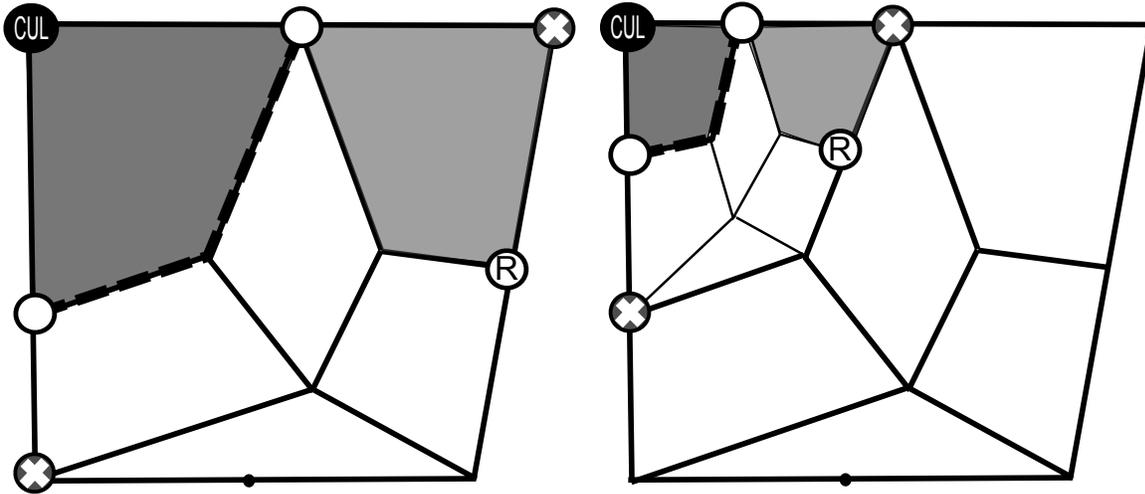


Рис. 29. 0-цепь и 1-цепь узла CUL.

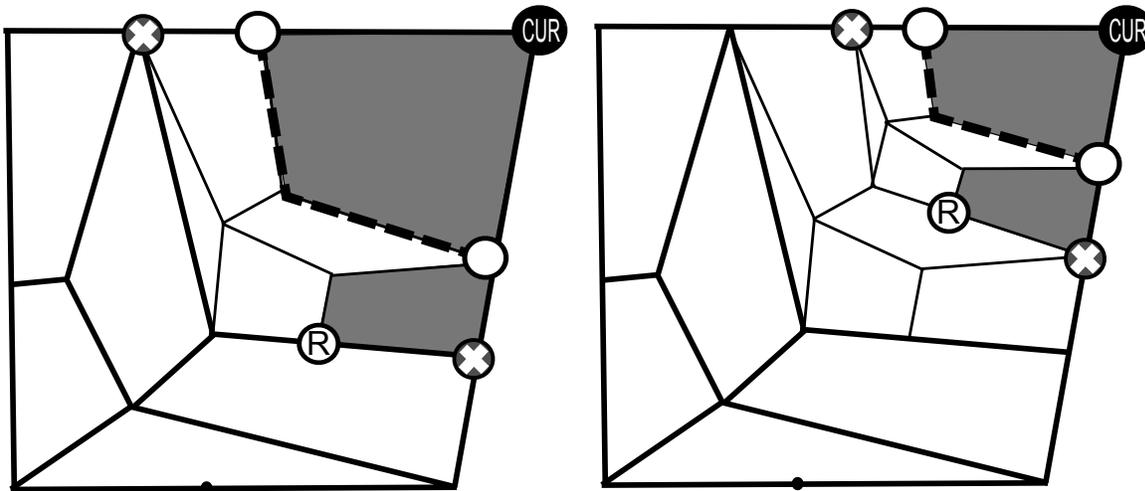


Рис. 30. 1-цепь и 2-цепь узла CUR.

Мы используем те же обозначения, что и в предыдущих случаях, крестами отмечены вершины, являющиеся правыми верхними или левыми нижними.

Вычисление функций `LevelPlus`, `TopRightType`, `BottomLeftType`, `TopFromCorner`, `RightCorner` очевидно, а функции `TopFromRight` и `BottomRightTypeFromRight` не могут быть применимы к вершинам данным цепям.

Узлы типа DR и RD. Для некоторых функций нужно также рассмотреть ситуации, когда аргументом является узел DR (или симметричный случай RD).

Окружение DR-узла может быть четырех видов, по числу внутренних ребер, на которых он может располагаться. На рисунке 33 отмечены эти четыре возможные ситуации. Во всех случаях начальником нашего DR-узла является вершина Y в середине верхней стороны. На рисунке отмечены серым цветом макроплитки, левый нижний угол которых попадает в DR-узел. Таким образом, значение функции `RightCorner` попадает в узлы отмеченные как "R". Это B-узел, и его окружение во всех случаях можно вычислить, учитывая, что мы знаем окружение узла в середине верхней стороны.

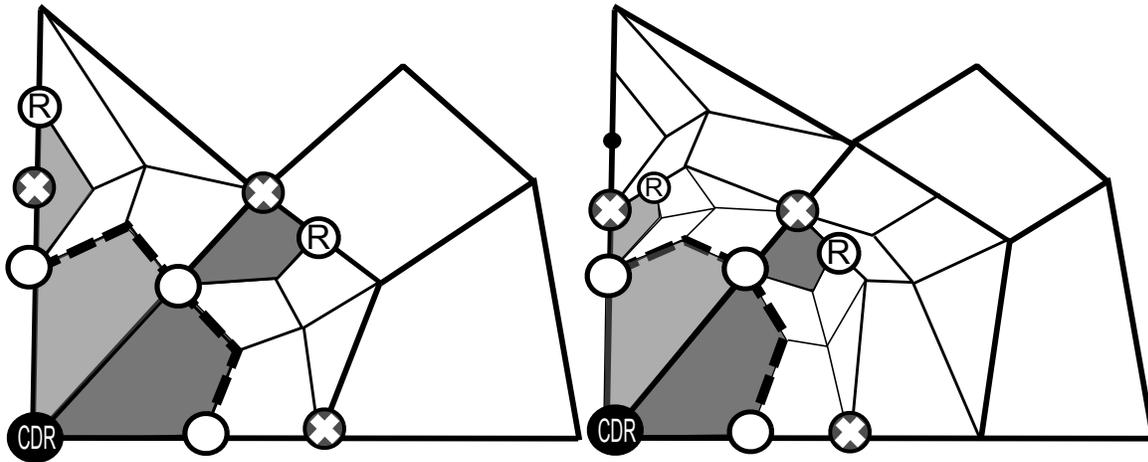


Рис. 31. 1-цепь и 2-цепь узла CDR.

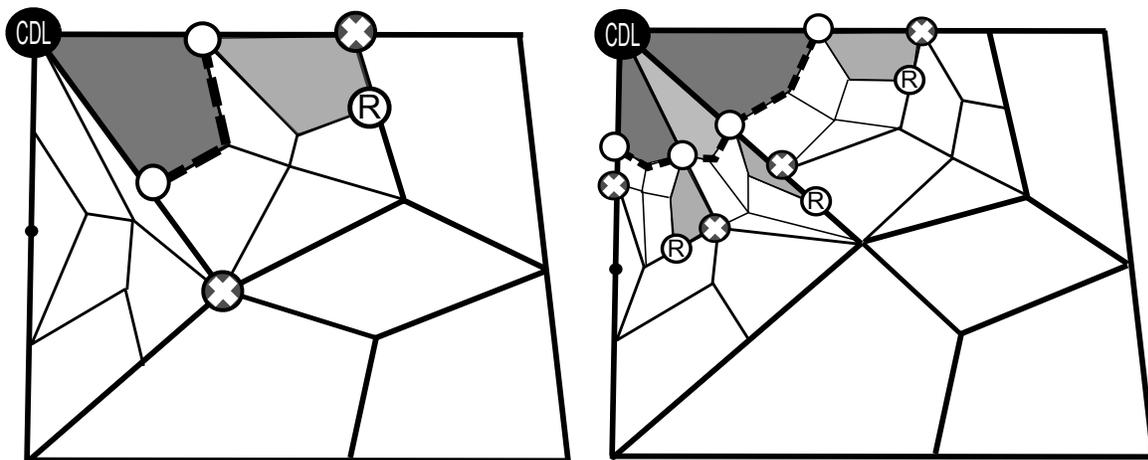


Рис. 32. 1-цепь и 2-цепь узла CDL.

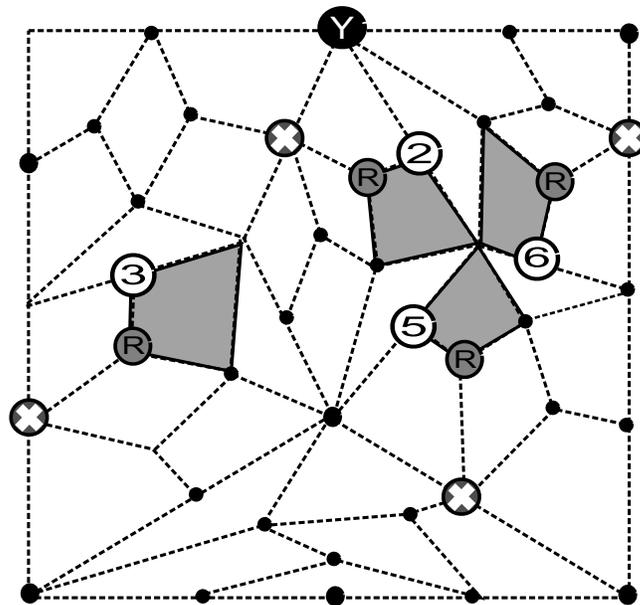


Рис. 33. Случаи расположения DR-узла.

Значение `TopFromCorner` указывает на 1 цепь вокруг \mathbb{A} -узла с указателем `ld` для одного из случаев расположения, и 1-цепь вокруг \mathbb{B} -узла для трех остальных случаев, указатели ясны из рисунка. Значение `BottomRightTypeFromRight` также очевидно во всех четырех случаях.

Теперь рассмотрим функцию `TopFromRight`. Вершины, являющиеся ее значениями, отмечены знаком “ \otimes ” на рисунке. Поскольку мы знаем окружение вершины Y в середине верхней стороны, мы можем вычислить и все требуемые окружения.

6.2. Указатели. Пусть узел X принадлежит некоторой цепи, а узел Y является ее центром, причем известно окружение X и тип внутреннего ребра, на котором лежит X (один из восьми типов внутренних ребер), а также известно окружение макроплитки T , которой принадлежит это ребро. Покажем, как восстановить *указатель* X , то есть тип ребра входа-выхода для Y .

Почти для всех ребер *указатель*, являющийся ребром выхода из Y устанавливается сразу по типу ребра: Для ребра типа 1, *указатель* будет очевидно `u2`, для ребра типа 2 будет `u1`, для ребра типа 3 будет `l`, для ребра типа 4 – в зависимости от типа Y : для \mathbb{A} указатель `2`, для \mathbb{C} будет `1`. Для 5 ребра опять в зависимости от типа Y – для \mathbb{B} – указатель `3`, для \mathbb{C} будет `2`. Для 6 ребра указатель `r`.

Пусть ребро имеет тип 7. Зная окружение макроплитки T , мы можем установить ее положение в родительской макроплитке. Далее вычисляем значение указателя: для левого-верхнего положения T это `l2`, для левого нижнего `ld`, для среднего `mid`, для правого верхнего `u3`, для правого нижнего – `rd`, для нижнего – `d`.

Пусть ребро имеет тип 8. Зная окружение макроплитки T , мы можем установить ее положение в родительской макроплитке. Далее вычисляем значение указателя: для левого-верхнего `lu`, для левого нижнего `ld`, для среднего `mid`, для правого верхнего `ru`, для правого нижнего – `rd`. В случае нижнего положения T , по типу верхней (и правой) стороны T можно узнать, какое положение уже родительская макроплитка для T занимает в своей родительской макроплитке и в зависимости от этого установить значение указателя: для левого-верхнего положения – `l3`, для левого нижнего `ld2`, для среднего `mid2`, для правого верхнего `u4`, для правого нижнего – `r3`, для нижнего – `d2`.

6.3. Дополнительные функции. Символами E_7 и E_8 будем обозначать код соответствующего входящего и выходящего ребра. Очевидно, что зная по окружению узла в левом нижнем углу можно установить E_7 , а по окружению узла в правом нижнем углу можно установить E_8 – (рисунок 34).

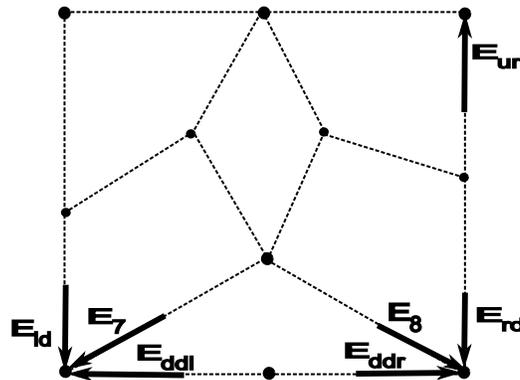


РИС. 34. Функции E_7 , E_8 , E_{ld} , E_{rd} , E_{ur} , E_{ddl} , E_{ddr} обозначают входящие или выходящие ребра в углы.

Аналогично, обозначим символами E_{1d} , E_{rd} , E_{ur} коды входящих или выходящих ребер в нижние углы по левой или правой сторонам и в верхний угол по правой стороне. Также символами E_{adl} , E_{adr} обозначим коды ребер в нижние углы по нижней стороне, слева и справа. Очевидно, что зная окружение соответствующего узла, можно установить букву, обозначающую нужное ребро (рисунок 34).

Пусть узел X принадлежит некоторой цепи. Поскольку нам известны окружения всех узлов в цепи, а также типы указателей и их порядок следования, мы можем рассмотреть следующий по часовой стрелке узел в цепи, после X . Будем обозначать его как $\text{Next}(X)$. Аналогично можно определить следующий после X узел против часовой стрелки: $\text{Prev}(X)$. Например, если в некоторой макроплитке известно окружение X узла в середине верхней стороны, то окружение узла в середине левой стороны будет $\text{Next}(X)$.

Также отдельно определим функцию BottomRightType , аргументами которой являются окружение и информация узла X , являющимся серединой верхней стороны в некоторой макроплитке T , а значением – тип узла в правом нижнем углу T и ребро, на котором он расположен, если это боковой узел.

Значение этой функции вычисляется следующим образом. Мы можем установить положение T в родительской макроплитке T' . Далее, для левого верхнего и левого нижнего положений тип правого нижнего угла будет \mathbb{A} , для среднего, правого верхнего и правого нижнего положений тип правого нижнего угла будет \mathbb{B} .

В случае нижнего положения T , правый нижний угол попадает в левый нижний угол T' . По типу правого и верхнего ребер T можно установить, какое положение занимает уже T' в своей родительской макроплитке T'' .

Для левого верхнего положения правый нижний угол T попадет в середину левой стороны T'' и его тип будет вычисляться функцией $\text{Next.FBoss}(X)$.

Для левого нижнего, среднего и нижнего положений правый нижний угол T попадет в \mathbb{C} -узел T'' .

Для правого верхнего положения правый нижний угол T попадет в середину верхней стороны T'' и его тип будет вычисляться как $\text{FBoss}(X)$.

Для правого нижнего положения правый нижний угол T попадет в \mathbb{B} узел T'' .

Если узел X в середине верхней стороны макроплитки известен, то функции BottomRightType , BottomLeftType , TopRightType дают нам типы узлов в трех соответствующих углах. Учитывая, что мы можем установить ребра входа в углы (символы E_{1d} , E_{rd} , E_{ur}), мы можем установить окружения отмеченных на рисунке 35 трех узлов. Обозначим соответствующие функции как BottomLeftChain , BottomRightChain , UpRightChain .

Также определим функцию $\text{RightFromB}(X)$, которая вычисляет окружение узла в середине правой стороны в макроплитке, по известному \mathbb{B} -узлу X . Значение этой функции вычисляется следующим образом.

Мы можем установить положение T в родительской макроплитке T' . Для левого верхнего положения, значением будет \mathbb{UR} -узел с окружением, соответствующим $\mathbb{A0}$ -цепи, с указателем 1. Типы левой и верхней стороны в T' те же что и в T , то есть мы их знаем по окружению X . Для среднего, левого нижнего, правого верхнего, правого нижнего положений значением будет \mathbb{RD} -узел с окружением соответственно 3, 2, 6, 5 внутренним ребрам. Для нижнего положения значением будет 1-цепь вокруг второго начальника X (мы знаем его тип), с указателем E_{adl} (вход по нижнему ребру в левый нижний угол).

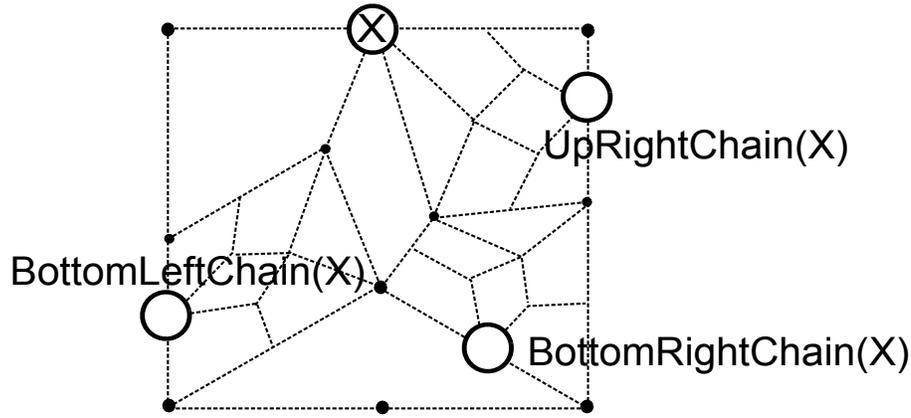


РИС. 35. Узлы, на которые указывают функции `BottomLeftChain`, `BottomRightChain`, `UpRightChain`.

7. ОЦЕНКА КОЛИЧЕСТВА БУКВ

Алфавит состоит из вершинных букв и реберных букв. Реберные буквы кодируют все возможные ребра входа и выхода, включая возможные выходы в подклеенные области. Плоских ребер, выходящих из вершины – не более 10 (больше всего в \mathbb{C} -вершине), всех подклеенных ребер не более 11 (\hat{x} , \hat{u}_i ; где $i = 1, \dots, 4$, \hat{l}_1 , \hat{l}_2 , \hat{l}_3 , \hat{d} , \hat{d}_2 , \hat{d}_3). То есть реберных букв всего 21. Каждая вершинная буква кодирует одну комбинацию значений параметров вершины, то есть тип, уровень, расширенное окружение, информация и флаг подклейки.

Посчитаем количество вершинных букв, используемых при кодировании. Четверок типов ребер всего существует 210, подробно этот подсчет произведен в приложении 11. То есть, существует 210 вариантов базовых окружений для типов \mathbb{A} , \mathbb{B} и \mathbb{C} . В цепях с центром в \mathbb{UL} -вершине существует 25 возможных вершин (7 в 1-цепи, и по 9 в 2-цепи и 3-цепи). Каждая может быть одного из трех уровней, то есть всего 75 возможных сочетаний базового окружения и уровня.

В таблице 1 ниже представлено, сколько различных базовых окружений может быть у вершины заданного типа, входящей в цепь указанного уровня.

Таким образом, всего возможно не более 698 окружений для вершин, входящих в некоторую цепь, причем для каждой из этих вершин также известен ее тип. Каждая из них может быть одного из трех уровней, итого 2094 сочетаний *тип-уровень-окружение*.

Теперь посчитаем количество вершин, не входящих в цепи. В таблице 2 ниже подсчитаны все возможные окружения вершин, не входящих в цепи.

Заметим, что у \mathbb{D} -вершины может быть только окружение (`left, top, right, bottom`), так как нижней стороной макроплитка примыкает к краю только являясь при этом подклеенной плиткой. Аналогично, у \mathbb{R} -вершины может быть два окружения (`left, top, right, bottom`) и (`8B, bottom, bottom, 7B`), так как она макроплитка может примыкать правой стороной только к правому или нижнему краям и только в двух этих случаях.

То есть всего существует не более 655 различных окружений для вершин не из цепей. Итого всех сочетаний тип-уровень-базовое окружение- не более $698 \times 3 + 655 = 2749$.

Рассмотрим теперь тип в подклеенной области. Он может быть один из следующих: \mathbb{CUL} , \mathbb{CUR} , \mathbb{CDL} , \mathbb{U} , \mathbb{L} . У первых трех по одному возможному подклеенному окружению. У \mathbb{U} -вершины всего 6 вариантов (по одному узлу в $\mathbb{U1}$, $\mathbb{U2}$, $\mathbb{U3}$, $\mathbb{L1}$, $\mathbb{L2}$, $\mathbb{L3}$, цепях). Аналогично у

ТАБЛИЦА 1. Количество вершин с различными окружениями, сгруппированных по цепям

тип центра цепи	0-цепь	1-цепь	2-цепь	3-цепь	всего
UL/LU	–	7	9	9	50
UR/RU	–	7	9	9	50
DL/LD	–	4	5	5	28
DR/RD	–	4	5	5	28
A	420	5	5	–	430
B	–	6	6	–	12
C	–	7	10	10	27
D	–	2	2	–	4
L	–	4	5	5	14
R	–	4	5	5	14
U	–	5	6	6	17
CUL	–	2	2	–	4
CUR	–	2	2	–	4
CDL	–	2	4	4	10
CDR	–	3	3	–	6

ТАБЛИЦА 2. Количество вершин с разными окружениями, сгруппированных по типам

тип вершины	число окружений	число уровней	всего
DR/RD	4	3	12
A	210	–	210
B	210	–	210
C	210	–	210
D	1	3	3
R	2	3	6
CUL	1	–	1
CUR	1	–	1
CDL	1	–	1
CDR	1	–	1

L-вершины тоже 6 вариантов подклеенных окружений. С учетом трех возможных уровней, получается $3 + 6 \times 3 + 6 \times 3 = 39$ сочетаний.

Из определения операции подклейки следует, что к вершине типов A, B, C и к угловым вершинам подклеенные макроплитки не могут примыкать левой или верхней стороной (только левым-верхним углом). Значит, вершина с нетривиальным подклееным окружением, не являющаяся ядром, в базовой плоскости либо имеет тип DR/RD, D, R, либо входит в некоторую цепь. То есть для таких вершин базовых окружений всего $698 \times 3 + 12 + 3 + 6 = 2115$. То есть расширенных окружений (сочетаний базового и подклееного) имеется не более $2115 \times 39 = 82485$.

Число значений параметра “информация”. Для простоты будем говорить не “число значений параметра информация”, а просто “число информации”. Оценим это число. Пусть сначала все начальники не лежат на левом или верхнем краю подклеенных плиток (то есть у них только базовое окружение).

Тогда первый начальник обязательно является вершиной, входящей в некоторую цепь, то есть 2094 варианта. Второй начальник (левый нижний угол) либо попадает в вершину из цепи, либо DIR/RD -вершину, либо C -вершину либо CDL -вершину. Всего $2094 + 12 + 210 + 1 = 2317$ вариантов. Окружение третьего начальника (правый нижний угол) может быть вычислено по окружению второго (функция `RightCorner`).

Итак, если начальник один, есть 2094 вариантов информации. Можно проверить, что тип правого нижнего угла можно восстановить по окружению узла в середине верхней стороны, для всех случаев, кроме двух – цепи A0 с указателем 3 и цепи B1 с указателем 2. То есть, всего возможных сочетаний окружения первого начальника и типа второго существует не более $2092 + 19 + 19 = 2130$.

Если начальника три, то окружение первого и третьего восстанавливаются по окружению второго (функции `TopFromCorner` и `RightCorner`), то есть всего вариантов информации не более 2317. Итого вариантов информации для плоского случая не более $2094 + 2317 + 2130 = 6541$.

Теперь рассмотрим случай, когда хотя бы один начальник лежит на левом или верхнем краю подклеенной макроплитки. Тогда в случае одного начальника вариантов информации будет не более $2115 \times 39 = 82485$ (любое сочетание базового и подклеенного окружения). Для вершин с двумя начальниками заметим, что когда первый начальник имеет краевой тип, мы можем вычислить подклеенный тип правого нижнего угла по подклеенному окружению первого начальника. То есть тут столько же вариантов информации, 82485.

Для вершин с тремя начальниками, третий начальник не может попасть на левый или верхний край подклеенной макроплитки, то есть у него будет только базовое окружение, которое можно вычислить по функции `RightCorner`. Возможны три случая:

1. И первый и второй начальники имеют и подклеенное и базовое окружение (то есть они оба попали на левую и верхнюю стороны подклеенной макроплитки). Тогда расширенное окружение второго начальника можно выбрать не более чем 82485 способами. Подклеенное окружение первого начальника можно вычислить по функции `TopFromCorner`. Если считать, что базовое окружение первого начальника – любое, то всего вариантов информации $82485 \cdot 2115$.

2. Только первый начальник попадает на край. Тогда число вариантов информации не превосходит $2749 \cdot 2115$. (Произвольное базовое окружение первого начальника и произвольное базовое окружение второго).

3. Только второй начальник попадает на край. Окружение первого начальника можно вычислить по функции `TopFromCorner`. Тогда число вариантов информации не превосходит 82485. Таким образом, общее число всех вариантов не более $82485 \cdot 2115 + 2115 \cdot 2749 + 82485 = 180352395$.

Любых возможных информации не более $181 \cdot 10^6$. Всего сочетаний (тип, уровень, расширенное окружение, информация) не более чем $181 \cdot 10^6 \cdot (82485 + 2749)$ (то есть либо произвольное расширенное окружение, либо базовое окружение, без подклеенного). Это число не превосходит $16 \cdot 10^{12}$.

Число флагов макроплиток. Ядро, то есть вершина в левом верхнем углу подклеенной макроплитки, может иметь произвольное базовое окружение и произвольную информацию. Флагом макроплитки называется сочетание типа, уровня, базового окружения и информации ядра этой макроплитки и упорядоченной пары двух выходящих ребер, которые определяют стороны макроплитки. Одно из ребер может вести в подклеенные области. Плоских выходящих ребер у любой вершины не более 10. Различных подклеенных ребер существует не более 11. То есть возможных флагов существует не более $2749 \cdot 11 \cdot 10 \cdot 16 \cdot 10^{12} < 5 \cdot 10^{18}$.

Таким образом, число всех возможных букв в алфавите не превосходит $82485 \cdot 16 \cdot 10^{12} \cdot 5 \cdot 10^{18} < 7 \cdot 10^{36}$.

8. РАЗБОР СЛУЧАЕВ РАСПОЛОЖЕНИЯ ПУТЕЙ

На рисунке 36 изображены пары путей. Далее мы покажем, как, зная код одного пути из пары, можно определить код другого пути либо установить, что код является мертвым. В целом для этого требуется уметь вычислять код любой вершины в любой макроплитке по известным кодам трех остальных вершин. Рассмотрим произвольный путь $X_1e_1e_2X_2e_3e_4X_3$, где X_1, X_2, X_3 – буквы, отвечающие кодам вершин, а e_1, e_2, e_3, e_4 – буквы, отвечающие ребрам входа и выхода. Мы должны установить, к какой из десяти конфигураций относится наш путь, а также показать, как провести локальное преобразование, то есть получить код другого пути из соответствующей пары.

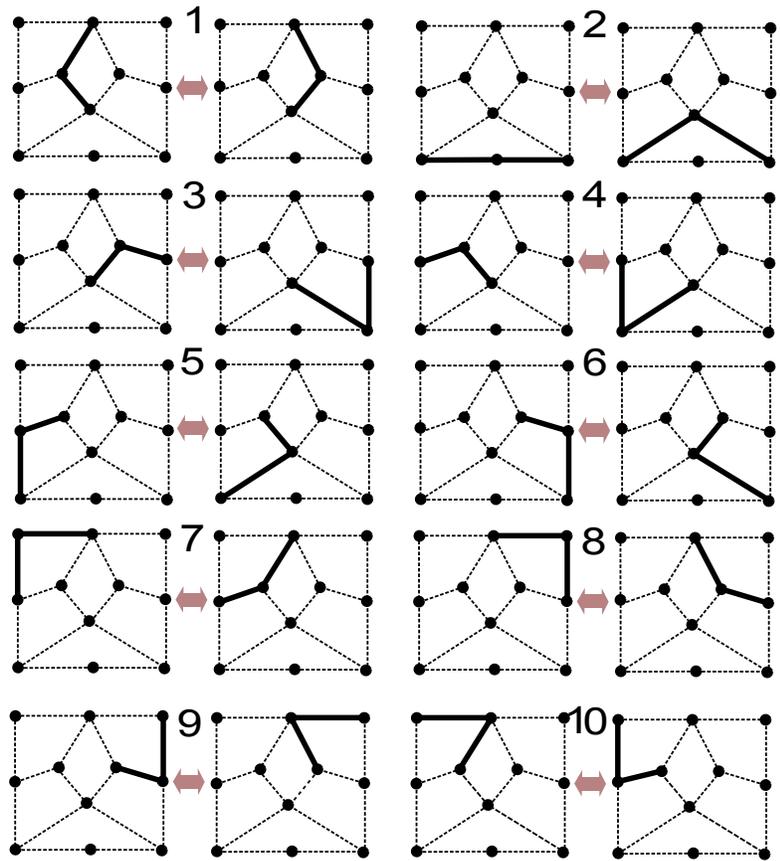


Рис. 36. По известному коду одного пути из пары можно восстановить код другого.

В этой главе мы рассмотрим случай, когда все буквы e_1, e_2, e_3, e_4 отвечают плоским ребрам, то есть наш путь не выходит в подклеенные области. Случай выхода в подклеенные области, когда хотя бы одна буква отвечает ребру, выходящему в подклейку, мы разберем в следующей главе. Фактически, в этой и следующей главах мы вводим определяющие соотношения в полугруппе.

8.1. Обзор перебора случаев: сочетания типов сторон макроплитки. Чтобы провести восстановление вершин для случаев (1) – (10), требуется знать что расположено вокруг плитки, в которой идет наш путь.

Итак, мы рассматриваем путь из двух последовательных ребер, проходящий внутри либо по границе некоторой макроплитки T . На рисунке 3б изображены макроплитки T с проходящими внутри них нашими путями. Основная задача состоит в том, чтобы в каждом случае *восстановить путь*, то есть по известным значениям параметров трех вершин и входящим ребрам вычислить параметры четвертой вершины и код парного пути.

Для плоского пути, когда среди ребер вдоль пути не встречаются выходы в подклейки, параметр *флаг подклейки* восстанавливается тривиально – для четвертой вершины он будет такой же, как и для остальных вершин (среди которых можно выбрать гарантированно лежащую не на границе подклеенной макроплитки). То есть нужно вычислить остальные параметры – тип, уровень, окружение и информацию.

Локальные преобразования с 1 по 6 мы рассмотрим каждое отдельно. При этом мы исследуем различные варианты расположения макроплитки, в которой проходит путь, и покажем, как вводятся определяющие соотношения в различных случаях расположения.

Локальные преобразования с 7 по 10 мы будем рассматривать все вместе для каждого случая расположения макроплитки.

Обозначения. Буквами $e_1, e_2, e_3, e_4, e_{u_1}, e_{u_2}, e_{u_3}, e_{1d}, e_r, e_{r_2}, e_{r_3}$ (и все подобные) будем обозначать входящие и выходящие ребра, соответствующие названию. Символами E_1, E_{1d}, E_{dd1} , (и подобными, указанными в разделе 6.3 “дополнительные функции”) будем обозначать ребра, получаемые в результате применения указанных функций. Все эти буквы мы будем использовать эти буквы при введении определяющих соотношений.

Иногда нам нужно будет зафиксировать часть параметров некоторой вершины X , чтобы потом использовать их для вычисления параметров других вершин. В этом случае мы будем использовать слово *назначение X* .

Пусть часть параметров вершины X зафиксированы и известны. Пусть также оперируя известными параметрами X , мы вычисляем некоторые параметры вершин X_1, X_2, X_3 . *Разрешенной* комбинацией будем называть такую упорядоченную четверку букв (x, x_1, x_2, x_3) , что x кодирует X с зафиксированными параметрами, x_i кодирует X_i , причем параметры совпадают с вычисленными. Заметим, что возможно несколько комбинаций разрешенных четверок, например, при разных параметрах флага подклейки. Часто кодирующие буквы мы будем обозначать так же, как сами вершины, например, Z, J, Y, F .

Замечание. Рассматриваемые нами пути могут быть кусками более длинных путей. В том числе возможна ситуация, когда после прохождения нашего подпути, далее путь уходит по подклеенному ребру, например, после вершины J . В этой ситуации, согласно определению, окружением J является расширенное окружение, состоящее из базового окружения (в плоскости нашего пути) и подклеенного окружения (в плоскости той подклеенной плитки, куда уходит наш путь). При этом подклеенное окружение часто может быть произвольным. Это отражается в том, что даже если тип, базовое окружение, флаг подклейки и информация у J вычислены нами из условия расположения пути и остальных вершин, подклеенное окружение J часто может быть произвольным. В этом случае в качестве *разрешенных* букв, кодирующих J , можно выбрать буквы из множества с фиксированными параметрами типа, базового окружения, флага подклейки и информации, но при этом с разными подклееными окружениями.

В этом случае мы будем говорить, что J выбрана с точностью до подклеенного окружения. Указанная ситуация может быть применима и к другим параметрам, например, информации.

8.2. Локальное преобразование 1. Рассмотрим пару путей на рисунке 37.

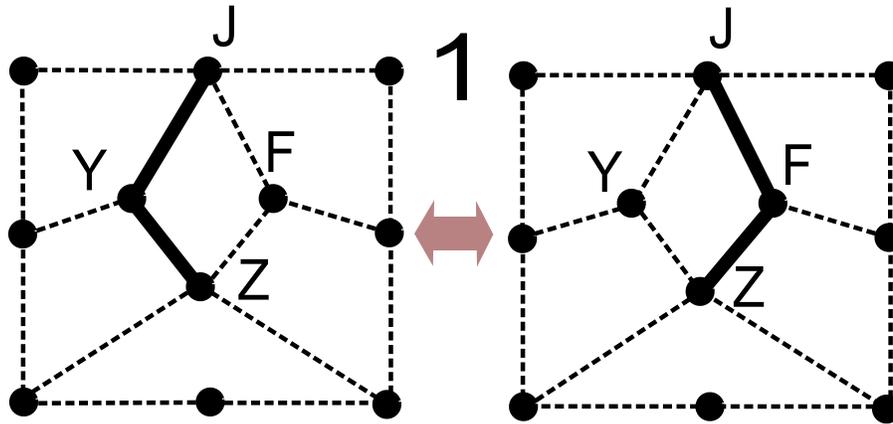


Рис. 37. Локальное преобразование 1

Определяющие соотношения.

Назначение Z . Пусть Z – вершина типа С. Зафиксируем ее флаг подклейки, базовое окружение, подклееное окружение и информацию.

Теперь можно вычислить базовые окружения вершин J, Y, F , так как J – это просто первый начальник Z , а базовые окружения Y и F совпадают с окружением макроплитки, которое содержится в базовом окружении J . Зафиксируем некоторую информацию J . Информацию Y и F известна, так как известны окружения J и вершины в правом нижнем углу (третий начальник Z).

Таким образом, выбрав произвольно базовое и подклееное окружение Z , подклееное окружение и информацию J , а также флаг макроплитки, мы можем вычислить значение остальных параметров. То есть, мы можем выписать множество разрешенных четверок букв (Z, J, Y, F) , кодирующих описанное положение вершин.

Заметим, что если буква Z зафиксирована, то буквы Y, F – точно определены, а буква J – определена с точностью до информации, то есть J могут кодировать разные буквы, отличающиеся информацией.

Для каждой разрешенной комбинации четырех букв J, Z, Y, F , введем следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_1e_{u_2}J &= Ze_2e_3Fe_1e_{u_1}J \\ Je_{u_2}e_1Ye_2e_1J &= Ze_{u_1}e_1Fe_3e_2J \end{aligned}$$

Два соотношения вводятся, потому что путь может быть пройден как в прямом порядке, так и в обратном.

Итак, Z – это произвольная буква типа С (произвольные расширенное окружение, флаг подклейки и информация), а у J может быть произвольная информация и произвольное расширенное окружение. Остальные буквы в этом соотношении заданы однозначно.

8.3. Локальное преобразование 2. Рассмотрим пару путей на рисунке 38.

Определяющие соотношения. Мы рассмотрим по очереди шесть случаев расположения. Для каждого случая мы определим разрешенные множества букв, которыми могут

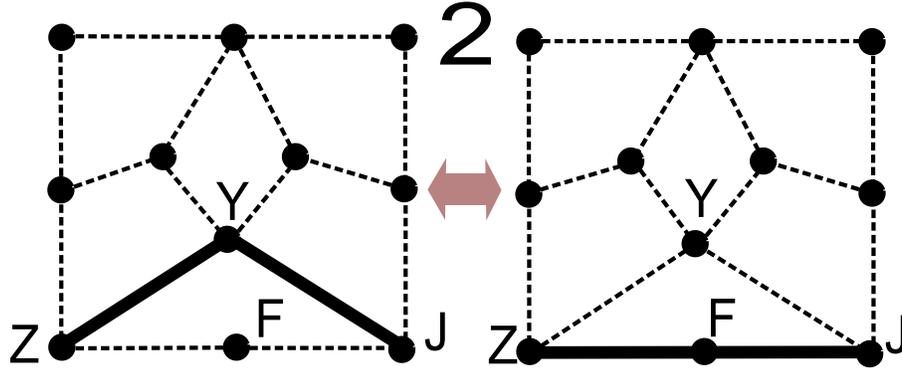


Рис. 38. Локальное преобразование 2

быть вершины Z , J , F , Y . После чего для каждой комбинации букв, введем два соотношения, одно отвечает прямому обходу данного пути (по стрелкам), а другое – обратному (против стрелок).

1. Левое-верхнее расположение (левая часть рисунка 39).

Назначение Z . Пусть Z – вершина типа LU/UL, LD/DL или L с произвольным расширенным окружением и информацией.

Начальники вершин J , Y , F отмечены на рисунке 39 (левая часть) черными кругами. Заметим, что мы можем вычислить их окружения. Одна из них входит в ту же цепь что и Z , а другая вычисляется с помощью функции `LevelPlus`. Значит, мы можем определить базовые окружения и информации вершин J , Y , F . Флаг подклейки может быть назначен произвольно (одинаковое значение для всех четырех вершин).

Таким образом, можно сформировать множество разрешенных четверок букв Z , J , Y , F .

Для каждой разрешенной комбинации заданных букв, введем следующее определяющие соотношения:

$$\begin{aligned} Ze_{1_2}e_4Ye_3e_{1_u}J &= Ze_1e_1Fe_2e_3J \\ Je_{1_u}e_3Ye_4e_{1_2}Z &= Je_3e_2Fe_1e_1Z \end{aligned}$$

Два соотношения отвечают двум направлениям обхода заданного участка, по стрелкам или против стрелок.

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F , так как их начальники и базовые окружения легко вычисляются по известному окружению и начальникам Z , в частности, у F общий набор начальников с J .

2. Среднее расположение (правая часть рисунка 39).

Назначение Z . Пусть Z – вершина с типом C и произвольным расширенным окружением, флагом подклейки и информацией.

Начальники вершин J , Y , F отмечены черными кругами. Зная код Z , можно вычислить их окружения. То есть можно вычислить коды J , Y , F .

Заметим, что если буква Z зафиксирована, то Y , F – точно определены, а J – с точностью до произвольного подклеенного окружения, при заданном базовом. То есть можно выделить множество разрешенных четверок букв, кодирующих наши вершины.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

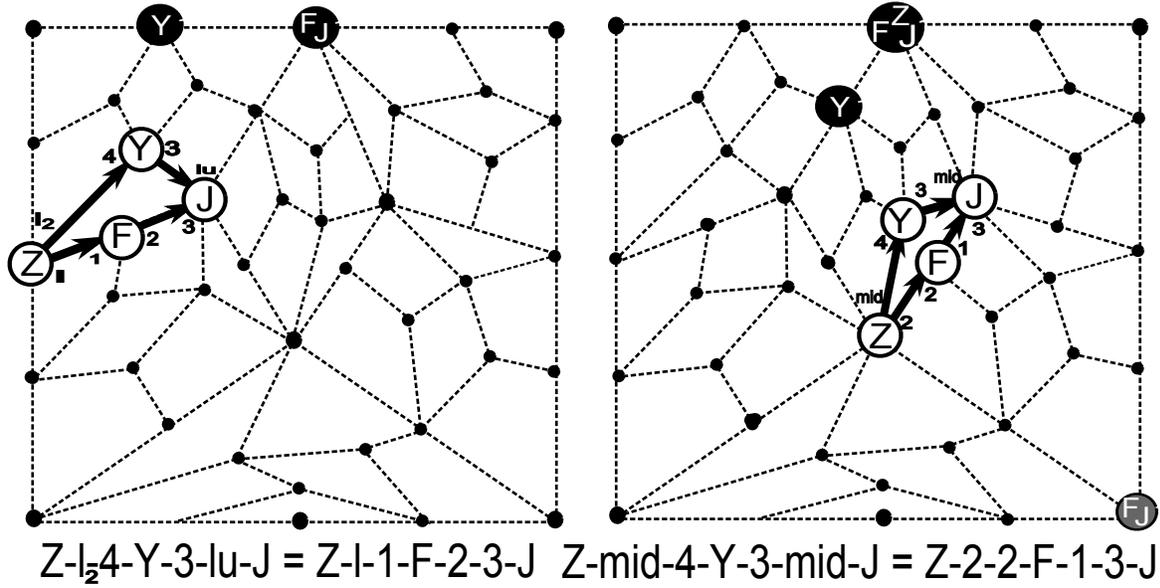


Рис. 39. Левое-верхнее и среднее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

$$\begin{aligned}
 Ze_{mid}e_4Ye_3e_{mid}J &= Ze_2e_2Fe_1e_3J \\
 Je_{mid}e_3Ye_4e_{mid}Z &= Je_3e_1Fe_2e_2Z
 \end{aligned}$$

Два соотношения отвечают двум направлениям обхода заданного участка, по стрелкам или против стрелок.

Восстановление кода. Аналогично левому-верхнему расположению.

3. Правое-верхнее расположение (рисунок 40, левая часть).

Назначение Z . Пусть Z – вершина типа UL/LU или UR/RU с произвольным расширенным окружением, флагом подклейки и информацией.

Окружение первого начальника Y мы получаем, используя функцию $UpRightChain(Z)$. Тип узла в правом нижнем углу – $BottomRightType(Z)$. То есть, коды всех остальных вершин можно вычислить.

Заметим, что если буква Z зафиксирована, то буквы F , Y – точно определены, а буква J – с точностью до произвольного подклеенного окружения, при заданном базовом.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned}
 Ze_{u_3}e_4Ye_3e_{ur}J &= Ze_{u_1}e_1Fe_2e_1J \\
 Je_{ur}e_3Ye_4e_{u_3}Z &= Je_1e_2Fe_1e_{u_1}Z
 \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

4. Левое-нижнее расположение (рисунок 40, правая часть).

Назначение Z . Пусть Z вершина типа C с произвольным расширенным окружением, флагом подклейки и информацией.

Цепь вокруг левого нижнего угла T можно получить как $BottomLeftChain(Z)$. То есть, окружения вершин, отмеченных кругами можно вычислить, и коды всех вершин J , Y , F можно вычислить, зная код Z .

Заметим, что если буква Z зафиксирована, то буквы F , Y – точно определены, а буква J – с точностью до произвольного подклеенного окружения, при заданном базовом.

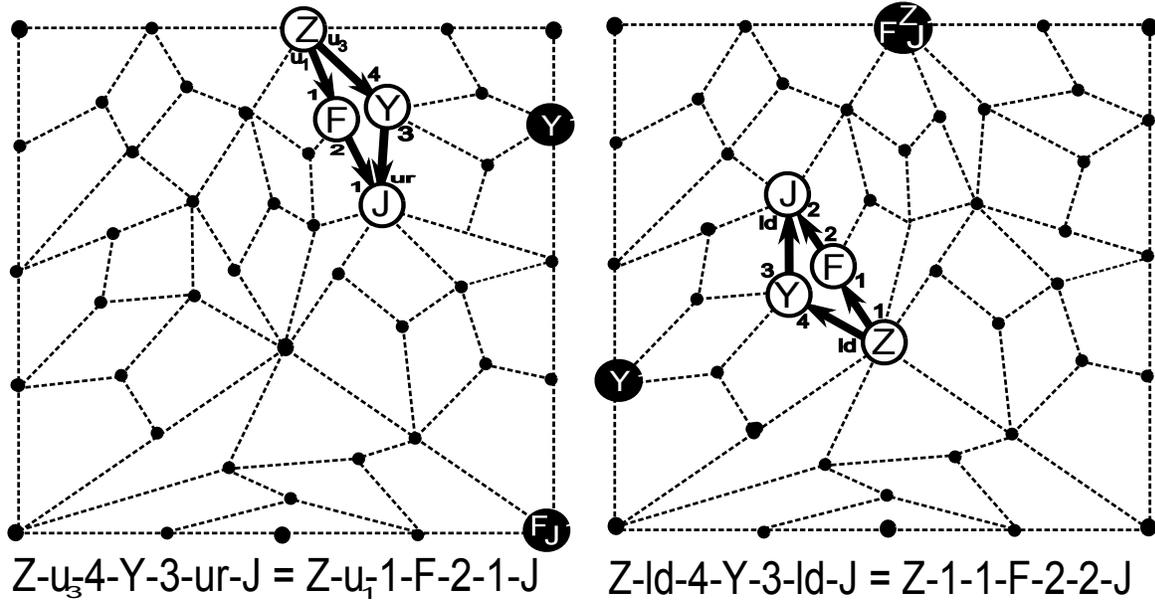


Рис. 40. Правое-верхнее и левое нижнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_{1d}e_4Ye_3e_{1d}J &= Ze_1e_1Fe_2e_2J \\ Je_{1d}e_3Ye_4e_{1d}Z &= Je_2e_2Fe_1e_1Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

5. Правое-нижнее расположение (рисунок 41, левая часть).

Назначение Z . Пусть Z – вершина типа \mathbb{RD}/\mathbb{DR} , \mathbb{RU}/\mathbb{UR} или \mathbb{R} с произвольным расширенным окружением, флагом подклейки и информацией.

Цепь вокруг правого нижнего угла T можно получить как $\text{BottomRightChain}(Z)$. То есть, можно вычислить окружения всех начальников, значит и все коды вершин J , Y , F .

Заметим, что если буква Z зафиксирована, то F , Y – точно определены, а J – с точностью до одного начальника и произвольного подклеенного окружения, при заданном базовом.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_{r_2}e_4Ye_3e_{rd}J &= Ze_re_2Fe_1e_2J \\ Je_{rd}e_3Ye_4e_{r_2}Z &= Je_2e_1Fe_2e_rZ \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

6. Нижнее расположение (рисунок 41, правая часть).

Назначение Z . Пусть Z – вершина типа \mathbb{C} с произвольным расширенным окружением, флагом подклейки и информацией.

Заметим, что J – второй начальник Z . Окружения и информации F и Y мы также можем вычислить.

Назначение J . Тип, окружение и флаг подклейки для J уже определены. Назначим произвольную информацию.

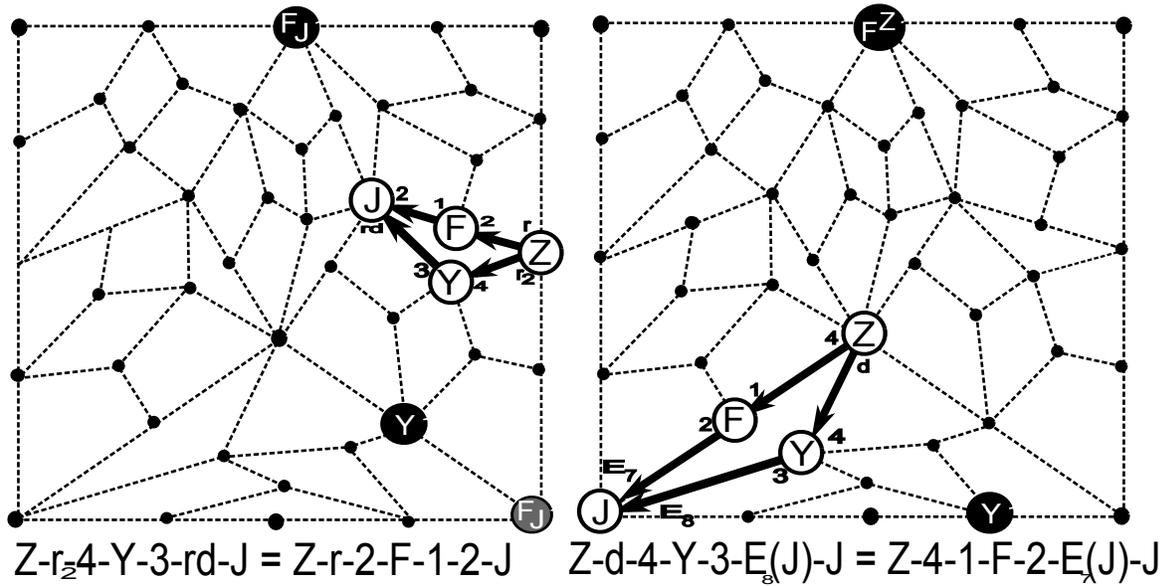


Рис. 41. Правое-нижнее и нижнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

Теперь буквы F , Y – точно определены, а буква J – с точностью до произвольной информации.

$E_7(J)$ и $E_8(J)$ – реберные буквы, полученные с помощью применения функций E_7 и E_8 к узлу J , то есть это тоже буквы, кодирующие ребра входов и выходов, но в зависимости от узла J .

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_4e_4Ye_3E_8(J)J &= Ze_4e_1Fe_2E_7(J)J \\ JE_8(J)e_3Ye_4e_4Z &= JE_7(J)e_2Fe_1e_4Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

8.4. Локальное преобразование 3. Рассмотрим пару путей на рисунке 42.

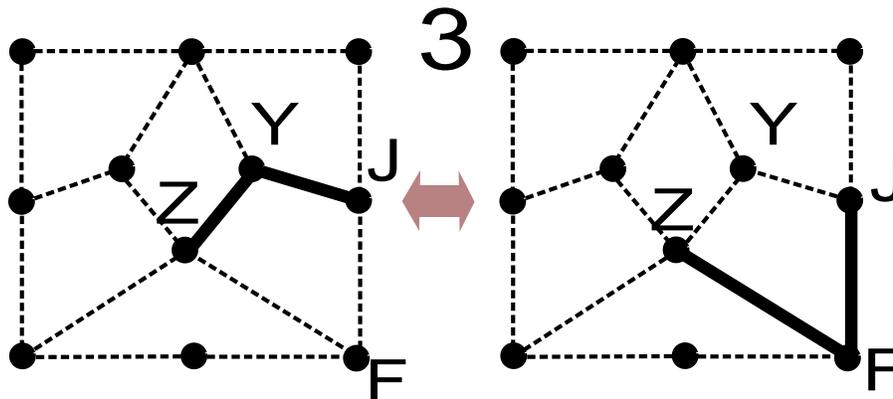


Рис. 42. Локальное преобразование 3

1. Левое-верхнее расположение (рисунок 43, левая часть).

Назначение F . Пусть F – вершина типа \mathbb{A} с произвольным базовым окружением, флагом подклейки и информацией.

Аналогично предыдущим случаям, этого достаточно, чтобы вычислить коды других вершин.

Заметим, что буквы F, Y – точно определены, а множества разрешенных букв J и Z – различаются произвольным подклеенным окружением, при заданном базовом.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_2e_3Ye_2e_rJ = Ze_3e_{1u}Fe_1e_1J$$

$$Je_re_2Ye_3e_2Z = Je_1e_1Fe_{1u}e_3Z$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F : у Y базовое окружение и первый начальник как у Z , а F является третьим начальником Z и имеет общий набор начальников с J .

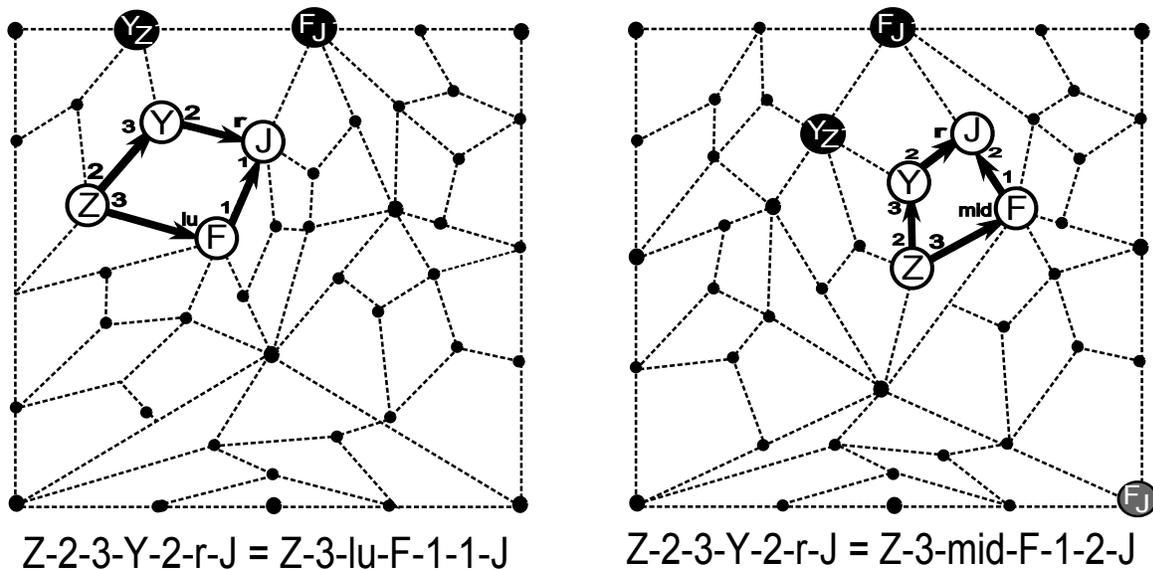


Рис. 43. Левое верхнее и среднее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

2. Среднее расположение (рисунок 43, правая часть).

Назначение F . Пусть F – вершина типа \mathbb{B} с произвольным базовым окружением, флагом подклейки и информацией.

Аналогично предыдущим случаям, мы можем вычислить коды всех вершин (J и Z – с точностью до произвольного подклеенного окружения, при заданном базовом).

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_2e_3Ye_2e_rJ = Ze_3e_{mid}Fe_1e_2J$$

$$Je_re_2Ye_3e_2Z = Je_2e_1Fe_{mid}e_3Z$$

Восстановление кода. Аналогично левому-верхнему расположению.

3. Левое нижнее расположение (рисунок 44, левая часть).

Назначение F . Пусть F – вершина типа \mathbb{A} с произвольным базовым окружением, флагом подклейки и информацией.

Цепь вокруг левого нижнего угла T можно получить как $\text{BottomLeftChain.FBoss}(F)$. Учитывая это, можно вычислить коды всех вершин (J и Z – с точностью до произвольного подклеенного окружения, при заданном базовом).

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_3Ye_2e_rJ &= Ze_3e_{1d}Fe_3e_2J \\ Je_re_2Ye_3e_2Z &= Je_2e_3Fe_{1d}e_3Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

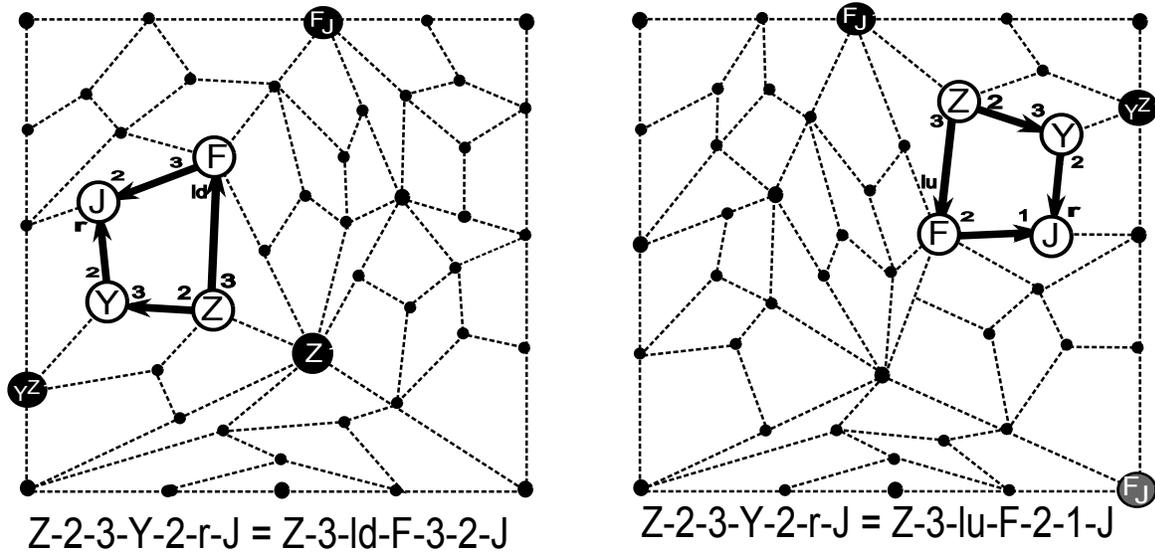


Рис. 44. Левое-нижнее и правое-верхнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

4. Правое-верхнее расположение (рисунок 44, правая часть).

Назначение F . Пусть F – вершина типа \mathbb{W} с произвольным базовым окружением, флагом подклейки и информацией.

Цепь вокруг правого верхнего угла можно получить как $\text{UpRightChain.FBoss}(F)$. Учитывая это, можно вычислить коды всех вершин (J и Z – с точностью до произвольного подклеенного окружения, при заданном базовом).

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_3Ye_2e_rJ &= Ze_3e_{1u}Fe_2e_1J \\ Je_re_2Ye_3e_2Z &= Je_1e_2Fe_{1u}e_3Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

5. Правое-нижнее расположение (рисунок 45, левая часть).

Назначение F . Пусть F – вершина типа \mathbb{A} с произвольным базовым окружением, флагом подклейки и информацией.

Окружение вершины в середине правой стороны можно вычислить с помощью функции RightFromB . Учитывая это, мы можем вычислить коды всех вершин (J и Z – с точностью до произвольного подклеенного окружения, при заданном базовом).

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_2e_3Ye_2e_rJ = Ze_3e_{rd}Fe_3e_1J$$

$$Je_re_2Ye_3e_2Z = Je_1e_3Fe_{rd}e_3Z$$

Восстановление кода. Аналогично левому-верхнему расположению.

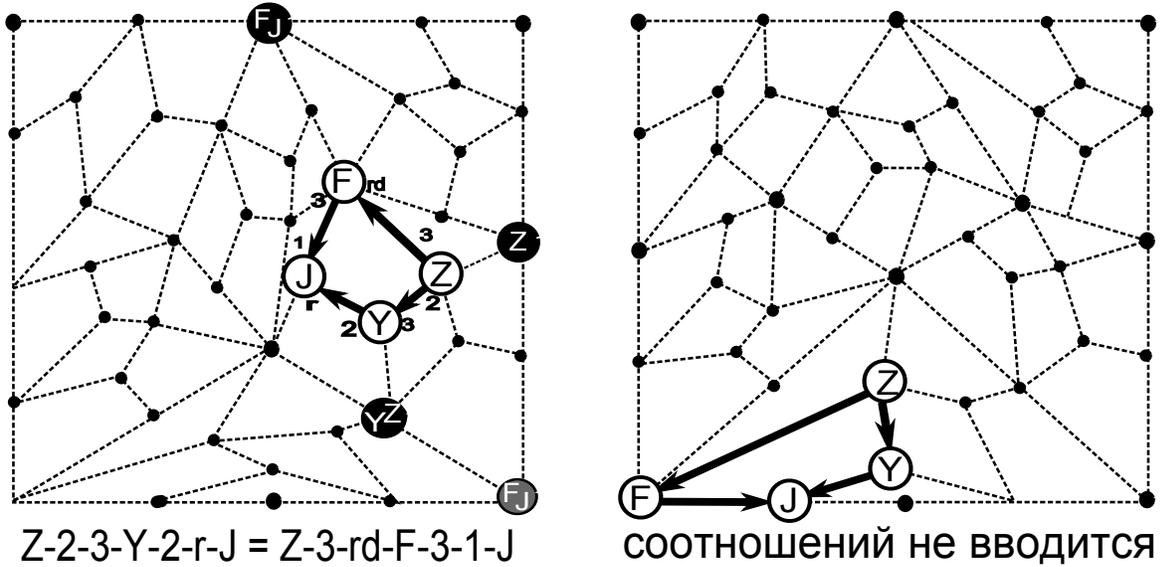


Рис. 45. Правое-нижнее и нижнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

6. Нижнее расположение (рисунок 45, правая часть).

В этом случае соотношений мы не вводим, так как расположение пути удовлетворяет признакам мертвого паттерна. То есть участок пути с данным кодом не может являться подпутем достаточно длинного ненулевого пути.

8.5. Локальное преобразование 4. Рассмотрим пару путей на рисунке 46.

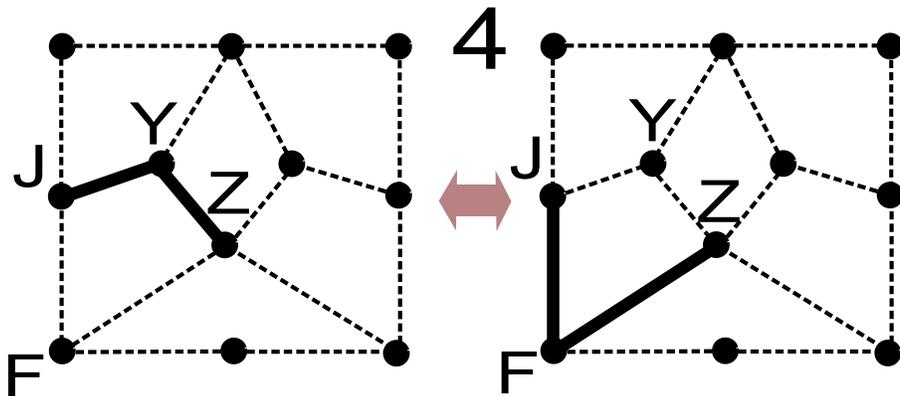


Рис. 46. Локальное преобразование 4

1. Левое-верхнее расположение (рисунок 47, левая часть).

В этом случае сначала определим тип узла T в левом верхнем углу. При этом J и F будут лежать на ребре выхода из T .

Назначение F . Пусть F – буква, кодирующая вершину типа L , LD или LU , лежащую в цепи вокруг T .

Заметим, учитывая расположение на рисунке, по известному коду F , мы можем вычислить окружения и начальников остальных трех вершин. То есть мы можем выписать все комбинации четырех букв Z, Y, J, F такие, что их окружения и начальники соответствуют окружению и начальникам изображенных вершин. Будем называть такие комбинации *разрешенными*. При этом, буквы Y и F будут определены однозначно, а Z и J могут быть любыми буквами с заданным базовым окружением и начальниками, но с различными подклееными окружениями.

Кроме того, буквы E_1 и E_{1d} (кодирующие соответствующие на рисунке ребра входа и выхода) легко вычисляются по известным кодам J и F .

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_3e_1J &= Ze_4e_{1_2}FE_{1d}E_1J \\ Je_1e_3Ye_2e_1Z &= JE_1E_{1d}Fe_{1_2}e_4Z \end{aligned}$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F : для Y это очевидно, а F является вторым начальником Z и имеет общий набор начальников с J .

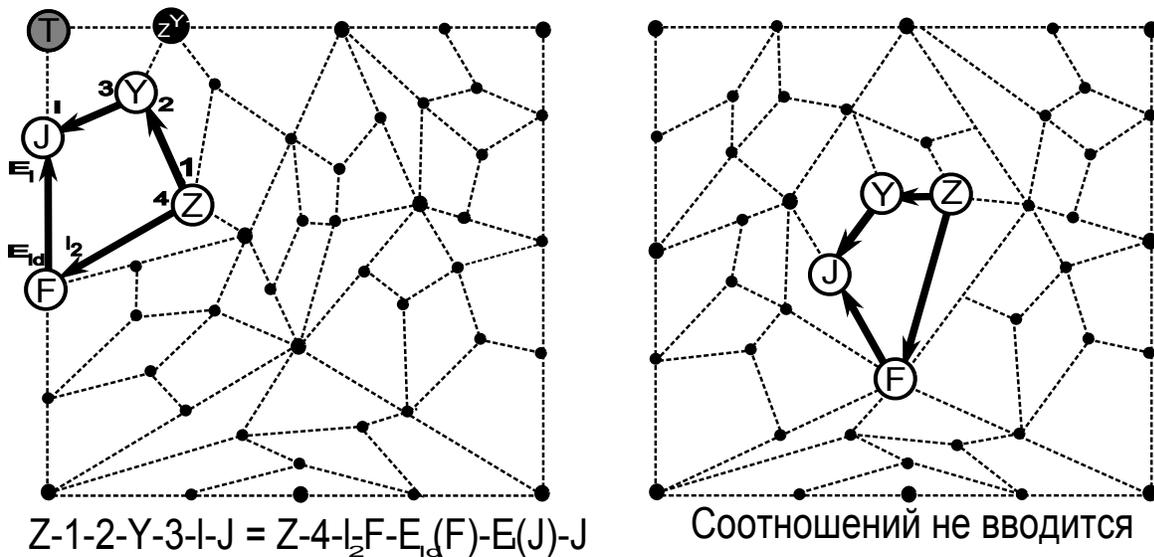


Рис. 47. Левое верхнее и среднее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

2. Среднее расположение, рисунок 47, правая часть.

В этом случае соотношений мы не вводим, так как расположение пути удовлетворяет признакам мертвого паттерна. То есть участок пути с данным кодом не может являться подпутем достаточно длинного ненулевого пути.

3. Левое нижнее расположение (рисунок 48, левая часть).

Назначение F . Пусть F – буква, кодирующая вершину типа C , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, J легко вычисляются. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников

F . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z , Y , J , F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_3e_1J &= Ze_4e_{1d}Fe_4e_2J \\ Je_1e_3Ye_2e_1Z &= Je_2e_4Fe_{1d}e_4Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

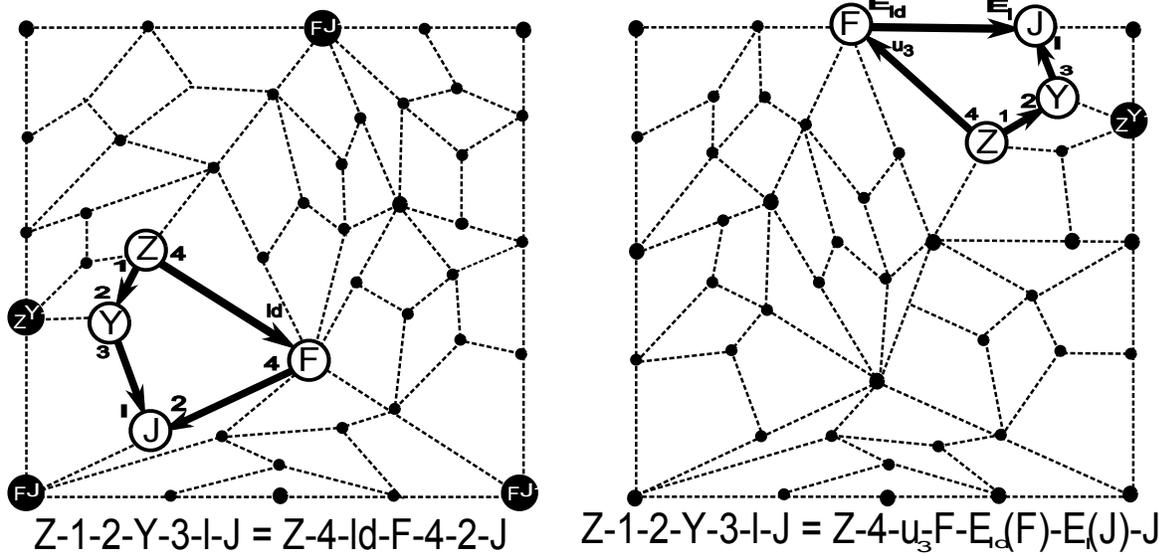


Рис. 48. Левое-нижнее и правое-верхнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

4. Правое верхнее расположение (рисунок 48, правая часть).

Назначение F . Пусть F – буква, кодирующая вершину типа U , UL или UR .

Тогда базовые окружения вершин Z , Y , J легко вычисляются, учитывая, что мы можем применить функцию $UpRightChain(F)$ и узнать тип вершины в правом верхнем углу.

Ясно также что начальники всех вершин также вычисляются, например у J они такие же, как у F . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z , Y , J , F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Кроме того, буквы E_1 и E_{1d} (кодирующие соответствующие на рисунке ребра входа и выхода) легко вычисляются по известным кодам J и F .

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_3e_1J &= Ze_4e_{u_3}FE_{1d}E_1J \\ Je_1e_3Ye_2e_1Z &= JE_1E_{1d}Fe_{u_3}e_4Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

5. Правое-нижнее расположение (рисунок 49, левая часть).

Назначение F . Пусть F – буква, кодирующая вершину типа R , RD или RU с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, J легко вычисляются. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников F . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Кроме того, буквы E_1 и E_{1d} (кодирующие соответствующие на рисунке ребра входа и выхода) легко вычисляются по известным кодам J и F .

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_3e_1J &= Ze_4e_{r_2}FE_{1d}E_1J \\ Je_1e_3Ye_2e_1Z &= JE_1E_{1d}Fe_{r_2}e_4Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

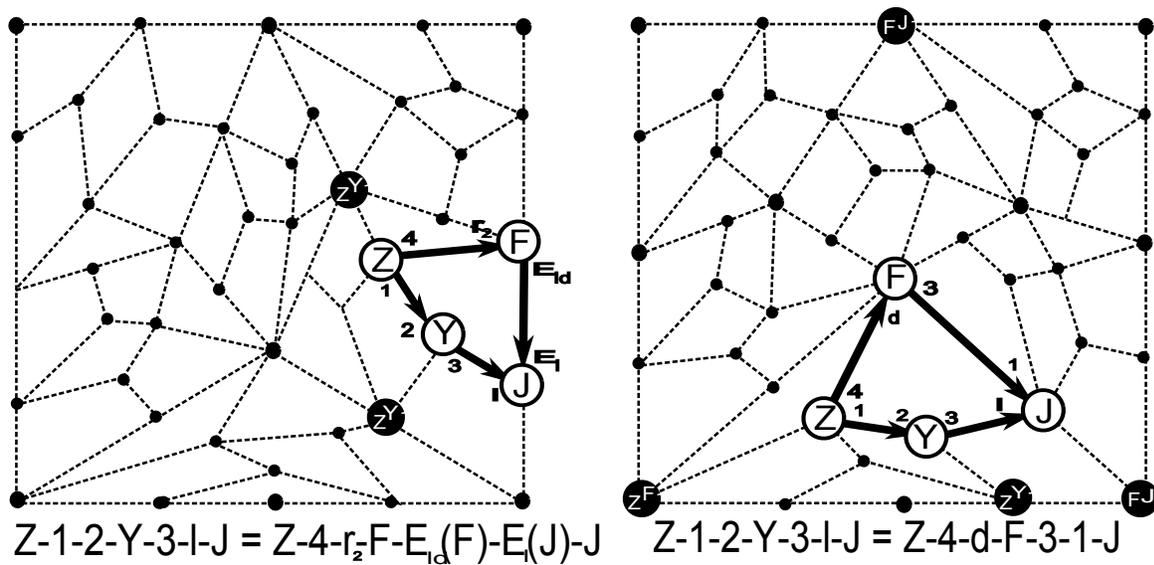


Рис. 49. Правое-нижнее и нижнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

6. Нижнее расположение (рисунок 49, правая часть).

Назначение F . Пусть F – буква, кодирующая вершину типа \mathbb{C} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, J легко вычисляются. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников F . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_3e_1J &= Ze_4e_dFe_3e_1J \\ Je_1e_3Ye_2e_1Z &= Je_1e_3Fe_d e_4Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

8.6. Локальное преобразование 5. Рассмотрим пару путей на рисунке 50.

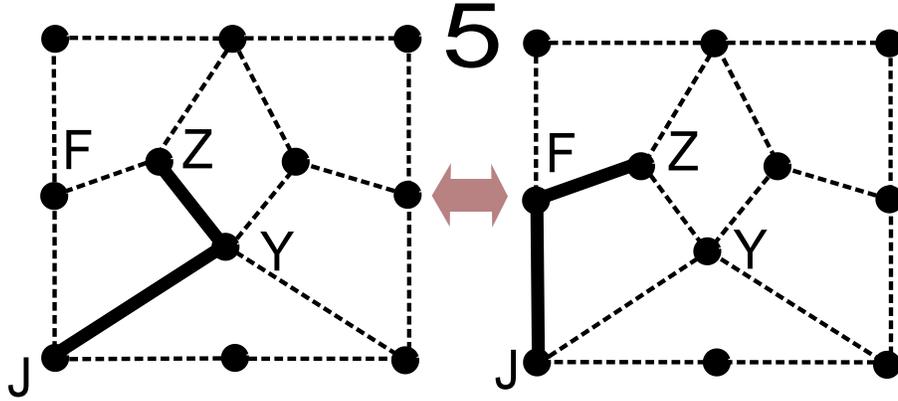


Рис. 50. Локальное преобразование 5

1. Левое-верхнее расположение (рисунок 51, левая часть).

В этом случае сначала определим тип узла T в левом верхнем углу. При этом J и F будут лежать на ребре выхода из T .

Назначение J . Пусть J – буква, кодирующая вершину типа L , LD или LU , лежащую в цепи вокруг T .

Аналогично предыдущему случаю, по известному коду J , мы можем вычислить окружения и начальников остальных трех вершин. То есть мы можем выписать все *разрешенные* комбинации четырех букв Z, Y, J, F то есть такие, что их окружения и начальники соответствуют окружению и начальникам изображенных вершин. При этом, буквы Y и F будут определены однозначно, а Z и J могут быть любыми буквами с заданным базовым окружением и начальниками, но с различными подклееными окружениями.

Кроме того, буквы E_1 и E_{1d} (кодирующие соответствующие на рисунке ребра входа и выхода) легко вычисляются по известным кодам J и F .

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_{12}J &= Ze_3e_1FE_1E_{1d}J \\ Je_{12}e_4Ye_1e_2Z &= JE_{1d}E_1Fe_1e_3Z \end{aligned}$$

Восстановление кода. Зная коды J и Z можно вычислить код Y и F : для Y это очевидно, а F соответствует $\text{Next.FBoss}(Z)$ и имеет общий набор начальников с J .

2. Среднее расположение (рисунок 51, правая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа C , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, F легко вычисляются. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

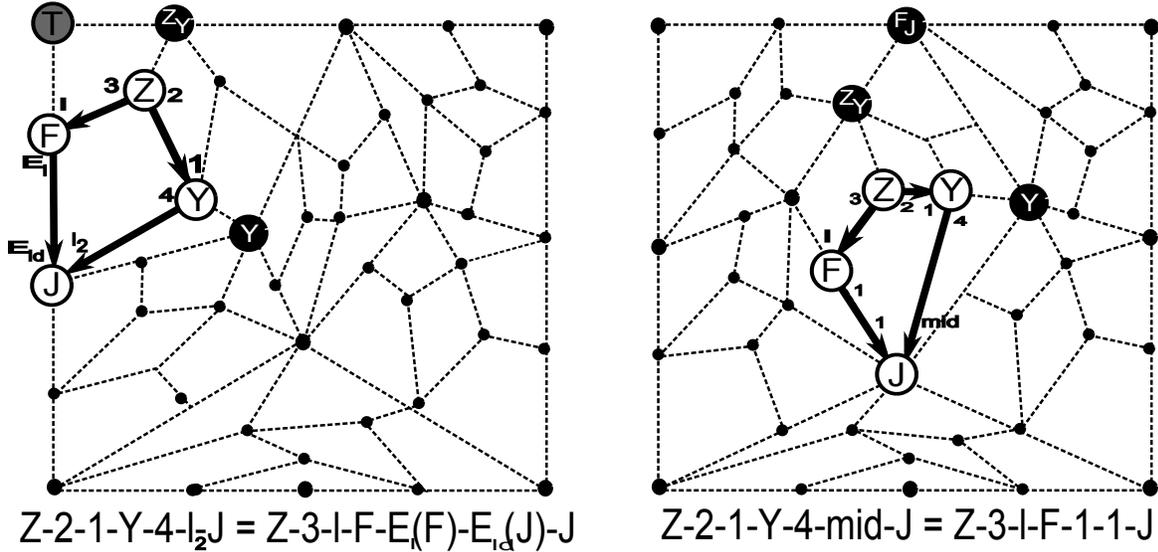


Рис. 51. Левое верхнее и среднее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

$$Ze_2e_1Ye_4e_{mid}J = Ze_3e_1Fe_1e_1J$$

$$Je_{mid}e_4Ye_1e_2Z = Je_1e_1Fe_1e_3Z$$

Восстановление кода. Аналогично левому-верхнему расположению.

3. Левое нижнее расположение (рисунок 52, левая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{C} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, F легко вычисляются. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_2e_1Ye_4e_{1d}J = Ze_3e_1Fe_2e_4J$$

$$Je_{1d}e_4Ye_1e_2Z = Je_4e_2Fe_1e_3Z$$

Восстановление кода. Аналогично левому-верхнему расположению.

4. Правое верхнее расположение (рисунок 52, правая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{U} , \mathbb{UL} или \mathbb{UR} .

Тогда базовые окружения вершин Z, Y, J легко вычисляются, учитывая, что мы можем применить функцию $\text{UpRightChain}(F)$ и узнать тип вершины в правом верхнем углу.

Начальники всех вершин также вычисляются, например у F они такие же, как у J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Кроме того, буквы E_1 и E_{1d} (кодирующие соответствующие на рисунке ребра входа и выхода) легко вычисляются по известным кодам J и F .

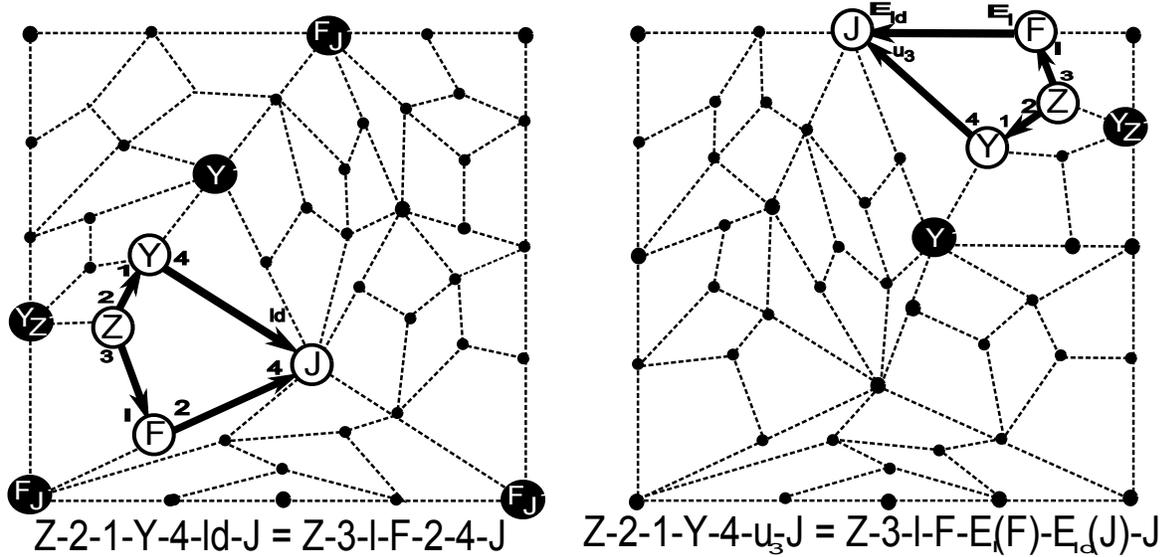


Рис. 52. Левое-нижнее и правое-верхнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_{u_3}J &= Ze_3e_1FE_1E_{1d}J \\ Je_{u_3}e_4Ye_1e_2Z &= JE_{1d}E_1Fe_1e_3Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

5. Правое-нижнее расположение (рисунок 53, левая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{R} , \mathbb{RD} или \mathbb{RU} с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z , Y , F (рисунок 53, левая часть), очевидны. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z , Y , J , F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Кроме того, буквы E_1 и E_{1d} (кодирующие соответствующие на рисунке ребра входа и выхода) легко вычисляются по известным кодам J и F .

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_{r_2}J &= Ze_3e_1FE_1E_{1d}J \\ Je_{r_2}e_4Ye_1e_2Z &= JE_{1d}E_1Fe_1e_3Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

6. Нижнее расположение (рисунок 53, правая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{C} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z , Y , F на вышеуказанном рисунке очевидны. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если

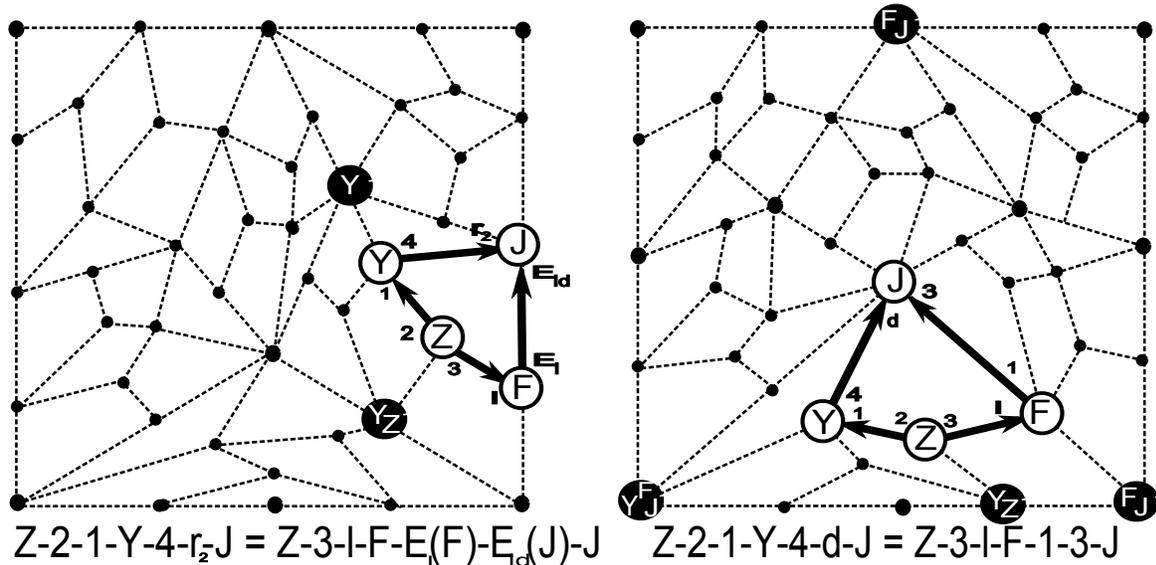


Рис. 53. Правое-нижнее и нижнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_dJ &= Ze_3e_1Fe_1e_3J \\ Je_4e_4Ye_1e_2Z &= Je_3e_1Fe_1e_3Z \end{aligned}$$

Восстановление кода. Аналогично левому-верхнему расположению.

8.7. Локальное преобразование 6. Рассмотрим пару путей на рисунке 54.

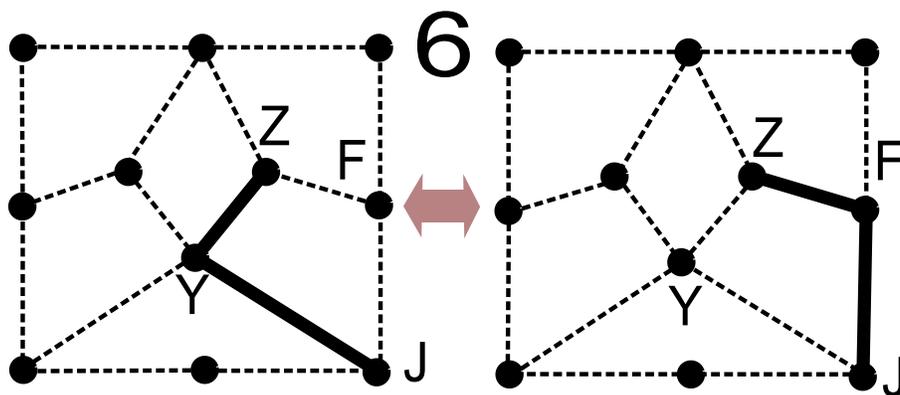


Рис. 54. Локальное преобразование 6

1. Левое-верхнее расположение (рисунок 55, левая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{A} , с произвольным окружением и произвольным начальником.

По известному коду J , мы можем выписать все *разрешенные* комбинации четырех букв Z, Y, J, F , то есть такие, что их окружения и начальники соответствуют окружению и начальникам изображенных на рисунке 55 вершин. При этом, буквы Y и F будут определены однозначно, а Z и J могут быть любыми буквами с заданным базовым окружением и начальниками, но с различными подклееными окружениями.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_3e_2Ye_3e_{1u}J = Ze_2e_rFe_1e_1J$$

$$Je_{1u}e_3Ye_2e_3Z = Je_1e_1Fe_re_2Z$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F . Окружение Y и первый начальник такие же как у Z , второй начальник соответствует $\text{Next.FBoss}(J)$, а третий – вершина J . Окружение F соответствует 0-цепи вокруг J с указателем 1, начальник тот же что и у J .

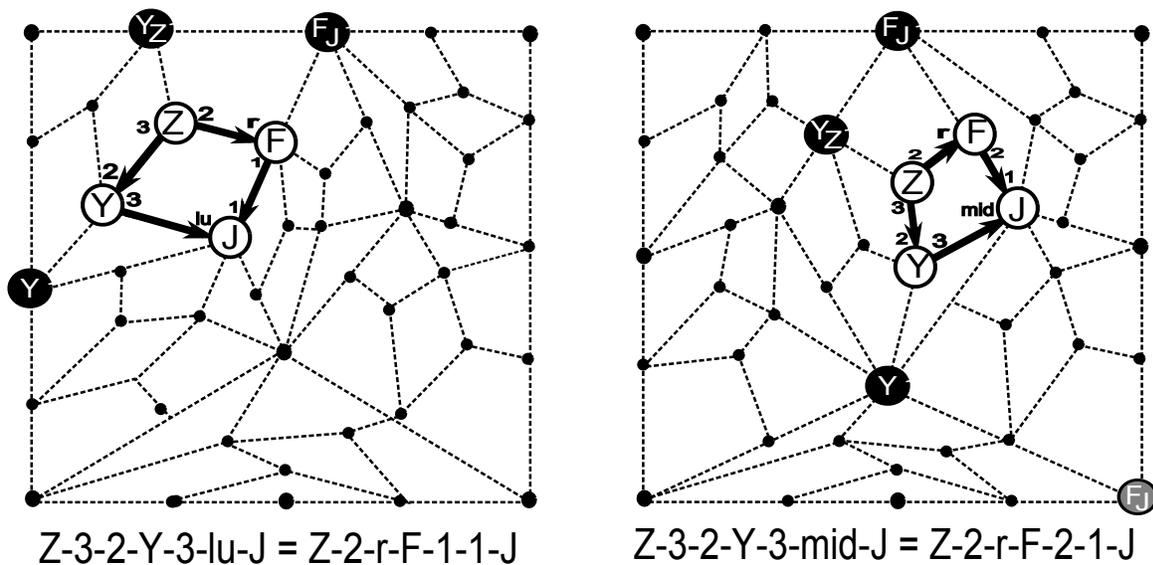


Рис. 55. Левое верхнее и среднее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

2. Среднее расположение (рисунок 55, правая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{B} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, F очевидны. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_3e_2Ye_3e_{\text{mid}}J = Ze_2e_rFe_2e_1J$$

$$Je_{\text{mid}}e_3Ye_2e_3Z = Je_1e_2Fe_re_2Z$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F . Окружение Y и первый начальник такие же как у Z , второй начальник это вершина типа \mathbb{C}

с окружением как у J , а третий – вершина J . Окружение F вычисляется по известному окружению J , начальник тот же что и у J .

3. Левое нижнее расположение (рисунок 56, левая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{A} , с произвольным окружением и начальником.

Тогда базовые окружения вершин Z, Y, F очевидны. Начальники всех вершин отмечены черными кружками, и очевидно все они вычисляются, если мы знаем код J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_3e_2Ye_3e_{1d}J &= Ze_2e_rFe_2e_3J \\ Je_{1d}e_3Ye_2e_3Z &= Je_3e_2Fe_re_2Z \end{aligned}$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F . Окружение Y и первый начальник такие же как у Z , второй начальник соответствует узлу \mathbb{C} с окружением как у J , а третий – сама вершина J . Окружение F вычисляется по известному окружению J , начальник тот же что и у J .

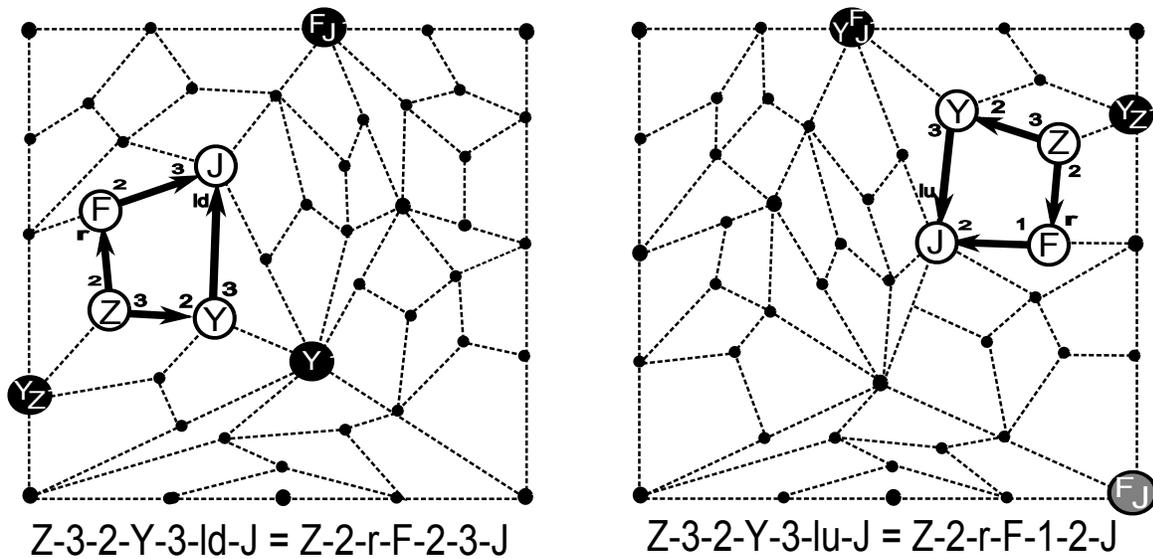


Рис. 56. Левое-нижнее и правое-верхнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

4. Правое верхнее расположение (рисунок 56, правая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{B} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, J , очевидны. Заметим, что мы можем применить функцию $UpRightChain(F)$ и узнать тип вершины в правом верхнем углу. То есть, начальники всех вершин, также вычисляются. Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_3e_2Ye_3e_{1u}J = Ze_2e_rFe_1e_2J$$

$$Je_{1u}e_3Ye_2e_3Z = Je_2e_1Fe_re_2Z$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F . Окружение Y и первый начальник такие же как у Z , второй начальник соответствует $\mathbf{FBoss}(J)$, а третий – вершина J . Окружение F вычисляется по известному окружению J , начальники те же что и у J .

5. Правое-нижнее расположение (рисунок 57, левая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{B} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, F (рисунок 57, левая часть), очевидны. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$Ze_3e_2Ye_3e_{rd}J = Ze_2e_rFe_1e_3J$$

$$Je_{rd}e_3Ye_2e_3Z = Je_3e_1Fe_re_2Z$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F . Окружение Y и первый начальник такие же как у Z , второй начальник вычисляется с помощью процедуры $\mathbf{RightFromB}(J)$, а третий – вершина J . Окружение F вычисляется по известному окружению J , начальники те же что и у J .

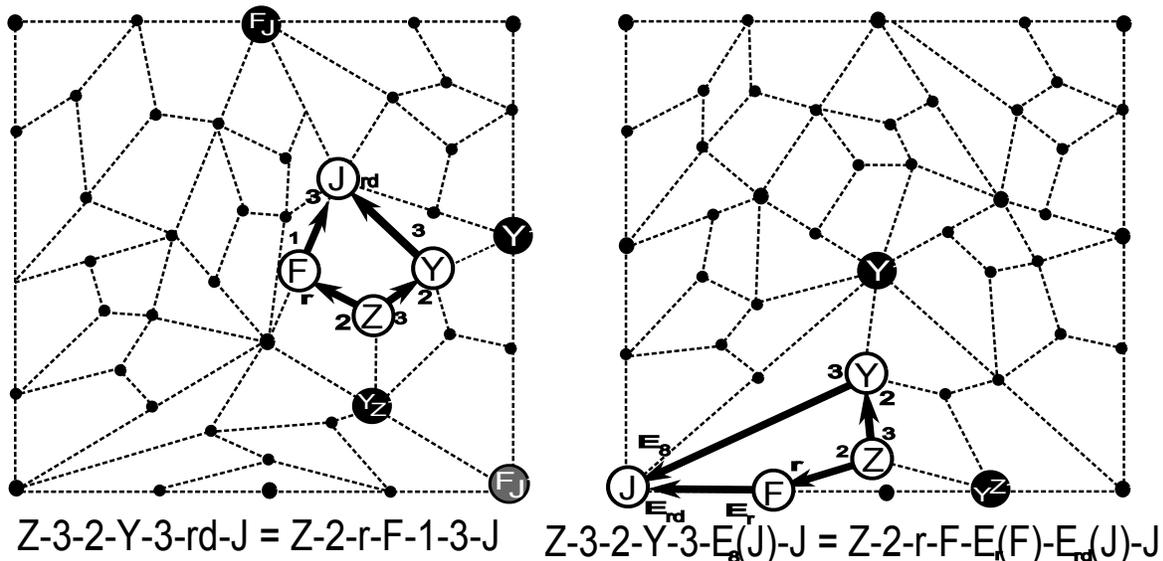


Рис. 57. Правое-нижнее и нижнее положения. При входе и выходе в вершины обозначены коды входящих и выходящих ребер

6. Нижнее расположение (рисунок 57, правая часть).

Назначение J . Пусть J – буква, кодирующая вершину типа \mathbb{C} , с произвольным окружением и начальниками.

Тогда базовые окружения вершин Z, Y, F на вышеуказанном рисунке очевидны. Начальники всех вершин отмечены черными кругами, и очевидно все они вычисляются, если мы знаем начальников J . Таким образом, мы можем аналогично определить все разрешенные четверки букв Z, Y, J, F , соответствующих вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешенной комбинации заданных букв, введем следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_4J &= Ze_3e_1Fe_1e_3J \\ Je_4e_4Ye_1e_2Z &= Je_3e_1Fe_1e_3Z \end{aligned}$$

Восстановление кода. Зная коды J и Z можно вычислить код как Y , так и F . Окружение Y и первый начальник такие же как у Z , второй начальник – вершина J , третий соответствует $\mathbf{SBoss}(J)$. Окружение F соответствует 1-цепи вокруг третьего начальника J с указателем, соответствующим входу в узел по нижней стороне, функция $E_{addr}(\mathbf{TBoss}(J))$. Начальники те же что и у J .

Локальные преобразования 7, 8, 9, 10.

Рассмотрим узел, лежащий в середине верхней стороны макроплитки, где расположен наш путь. Этот узел должен входить в некоторую цепь. То есть может быть три варианта цепи вокруг центра, который может иметь следующий тип:

$$\mathbb{C}, \mathbb{B}, \mathbb{A}, \mathbb{UL}/\mathbb{UL}, \mathbb{UR}/\mathbb{RU}, \mathbb{DL}/\mathbb{LD}, \mathbb{DR}/\mathbb{RD}, \mathbb{U}, \mathbb{L}, \mathbb{D}, \mathbb{R}, \mathbb{CUL}, \mathbb{CUR}, \mathbb{CDL}, \mathbb{CDR}.$$

Мы разберем все эти случаи и введем соотношения в соответствии с устройством всех возможных цепей.

Замечание. Ниже мы не будем выписывать симметричное к введенному соотношение, отвечающее проходу пути в обратном порядке. Просто будем считать, что соотношений вводится в два раза больше.

8.8. Случай цепи $\mathbb{C}1$; преобразования 8 и 9. В правой части рисунка 58 изображены локальные преобразования 8 и 9. Обозначим макроплитку, в которой проходят пути как T . Вершина Z попадает в середину верхней стороны T .

Итак, пусть середина верхней стороны T входит в $\mathbb{C}1$ -цепь. Различные случаи расположения интересующих нас путей показаны в левой части рисунка 58.

Сначала введем определяющие соотношения для данного случая. Зафиксируем \mathbb{C} -вершину с некоторым базовым окружением и начальниками $\mathbf{FBoss}(Z)$, $\mathbf{SBoss}(Z)$, $\mathbf{TBoss}(Z)$. Исходя из этих данных, мы можем определить коды вершин Z, Y, F, J во всех четырех случаях расположения и выписать определяющие соотношения для локальных преобразований 8 и 9.

Покажем, как можно определить эти коды. Вершины H_1, H_2, H_3 являются начальниками центральной \mathbb{C} -вершины. Зная типы, окружения и уровни вершин H_1, H_2, H_3 , можно выписать типы, окружения и уровни Z, Y, F, J во всех четырех случаях расположения, что видно на рисунке 58. Например, окружение F в случае 4 соответствует 1-цепи вокруг H_2 с указателем $E_7(H_2)$, и уровень F равен 2.

На рисунке 58 черными кругами отмечены вершины, являющиеся начальниками хотя бы одной из вершин Z, Y, F, J в каждом из четырех случаев, число в круге обозначает номер случая. Окружение всех вершин, отмеченных черными кругами, также можно выписать,

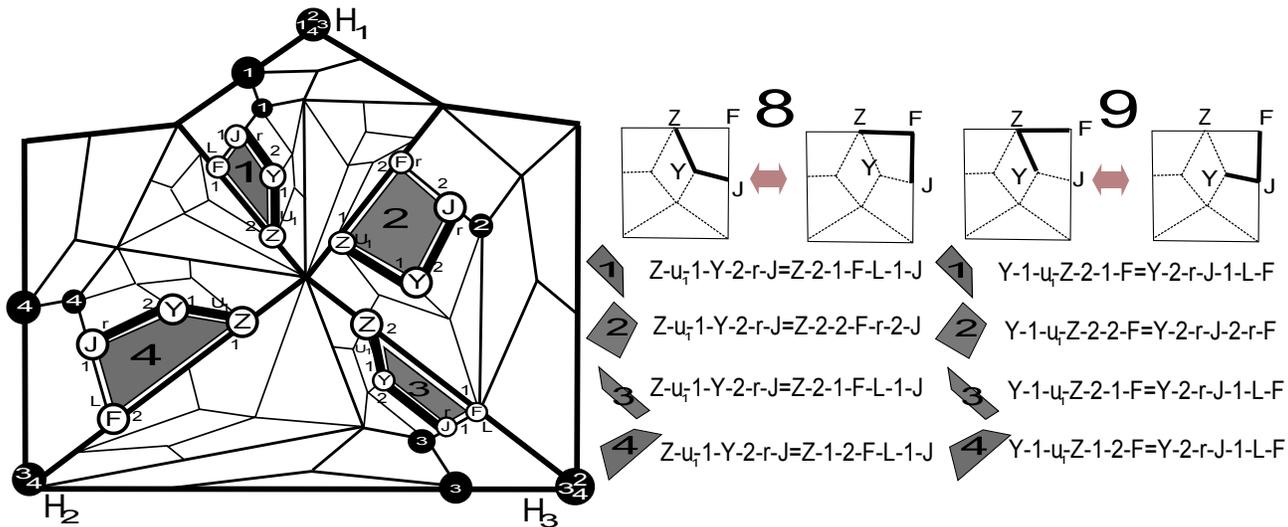


Рис. 58. Случаи расположения пути вокруг $C1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

зная начальников центральной C -вершины. То есть, мы можем вычислить начальников каждой из четырех вершин во всех случаях.

Уровни вершин F во всех случаях равны 2, а остальных боковых вершин (J и Z) равны 1.

Таким образом, мы можем определить множество букв Z , кодирующих вершины с заданным базовым окружением и начальниками, аналогично с остальными вершинами. Для локального преобразования 8 буквы F и Y будут выбраны однозначно, а Z и J – с точностью до подклеенного окружения. Для локального преобразования 9 все наоборот.

Соотношения для преобразования 8.

Назначение Z . Для каждого из четырех случаев расположения, обозначим символом Z буквы в алфавите, соответствующие кодам вершин с заданными типом, уровнем, базовым окружением и информацией. То есть эти буквы отличаются друг от друга только различными подклееными окружениями.

Назначение J . Аналогично определим буквы J , как буквы в алфавите, соответствующие кодам вершин J с заданными типом, уровнем, базовым окружением и информацией, и произвольным подклееным окружением.

Назначение Y и F . Буквы Y и F мы определим как конкретные буквы в алфавите с заданными типом, уровнем, базовым окружением и информацией, при пустом подклееном окружении. Мы это делаем, так как путь проходит через F и Y по плоским ребрам.

Теперь для каждого случая и для каждого разрешенного набора букв Z, F, Y, J выпишем соотношение, представленное в правой части рисунка 58. Буквы Z, F, Y, J мы только что определили, а остальные символы в соотношении обозначают входящие и выходящие ребра.

Соотношения для преобразования 9. Тут уже Z и J определены однозначно, а Y и F с точностью для подклеенного окружения. В остальном, все аналогично.

Оценка числа соотношений. Так как в каждом соотношении две буквы определены однозначно, а другие две – с точностью для подклеенных окружений, то для каждого

случая мы выписываем число соотношений, равное квадрату числа возможных подклеенных окружений. Кроме того, все соотношения могут быть выписаны для любого значения параметра *флаг макроплитки*.

Мы вводим не более $8FP^2\text{Num}(\mathbb{C})$ соотношений, где F – число различных флагов макроплиток, P – число различных подклеенных окружений, $\text{Num}(\mathbb{C})$ – число вершин типа \mathbb{C} , то есть число сочетаний “тип-уровень-окружение-информация”.

Характеризация. Пусть есть слово W представляющее собой код пути $X_1e_1e_2X_2e_3e_4X_3$. Покажем, как по нему установить, имеем ли мы дело с локальным преобразованием 8 или 9, а также как его провести.

На рисунке 58 отмечены входящие и выходящие ребра в каждом из четырех случаев. Исходя из этого легко выписать свойства пути, позволяющие нам установить, с каким случаем мы имеем дело. Все показано в таблицах 3 и 4.

ТАБЛИЦА 3. Характеристические условия на принадлежность пути к случаю $\mathbb{C}1$ цепи.

Условие на буквы	Симметричное условие (проход в обратном порядке)	случай	Локальное преобр
$\text{Surr}(X_1) = \mathbb{C}11, e_1e_2e_3e_4 = u_112r$	$\text{Surr}(X_3) = \mathbb{C}11, e_1e_2e_3e_4 = r21u_1$	1	8 (левая)
$\text{Surr}(X_1) = \mathbb{C}11, e_1e_2e_3e_4 = 21l1$	$\text{Surr}(X_3) = \mathbb{C}11, e_1e_2e_3e_4 = 1l12$	1	8 (правая)
$\text{Surr}(X_1) = \mathbb{C}12, e_1e_2e_3e_4 = u_112r$	$\text{Surr}(X_3) = \mathbb{C}12, e_1e_2e_3e_4 = r21u_1$	2	8 (левая)
$\text{Surr}(X_1) = \mathbb{C}12, e_1e_2e_3e_4 = 12r2$	$\text{Surr}(X_3) = \mathbb{C}12, e_1e_2e_3e_4 = 2r21$	2	8 (правая)
$\text{Surr}(X_1) = \mathbb{C}13, e_1e_2e_3e_4 = u_112r$	$\text{Surr}(X_3) = \mathbb{C}13, e_1e_2e_3e_4 = r21u_1$	3	8 (левая)
$\text{Surr}(X_1) = \mathbb{C}13, e_1e_2e_3e_4 = 21l1$	$\text{Surr}(X_3) = \mathbb{C}13, e_1e_2e_3e_4 = 1l12$	3	8 (правая)
$\text{Surr}(X_1) = \mathbb{C}14, e_1e_2e_3e_4 = u_112r$	$\text{Surr}(X_3) = \mathbb{C}14, e_1e_2e_3e_4 = r21u_1$	4	8 (левая)
$\text{Surr}(X_1) = \mathbb{C}14, e_1e_2e_3e_4 = 12l1$	$\text{Surr}(X_3) = \mathbb{C}14, e_1e_2e_3e_4 = 1l21$	4	8 (правая)

Ясно, что указанные условия на буквы полностью определяют конфигурацию пути, а также его код, с точностью до подклеенных окружений. Иначе говоря, не бывает никакого другого пути с заданными условиями на буквы, кроме пути, указанного нами.

То есть, для заданной конфигурации есть конечное множество слов Q , которые могли бы кодировать такой путь. Два слова в этом множестве отличаются только кодами подклеенных окружений для крайних вершинных букв.

Восстановление кода. Зафиксируем некоторый случай расположения n , один из четырех. Допустим, мы знаем коды трех из четырех вершин (из числа Z, Y, J, F). Тогда, учитывая характеризацию, мы знаем с каким из случаев расположения мы имеем дело. Далее, окружение вершины H_1 мы вычисляем, так как она первый начальник Z и F . В первом и втором случаях расположения знания окружения H_1 достаточно, чтобы определить окружение оставшейся вершины. Для третьего и четвертого случаев, мы знаем также окружение H_2 и H_3 – второй и третий начальник F и Z . Этого достаточно, чтобы определить оставшееся окружение.

Теперь покажем, как определить начальников. У F и Z общее множество начальников. Первым начальником Y всегда является Z , а тип второго очевиден из расположения. В случаях расположения 1, 3, 4 начальники J лежат в той же цепи, что и F .

В случае 2, первый начальник J соответствует 1-цепи вокруг H_3 . Тип H_3 мы знаем, так как это второй начальник F и Z . Указатель можно установить по окружению макроплитки, $\text{FBoss}(Z)$. Тип второго начальника J – это \mathbb{B} .

ТАБЛИЦА 4. Характеристические условия на принадлежность пути к случаю $\mathbb{C}1$ цепи.

Условие на буквы	Симметричное условие (проход в обратном порядке)	случай	Локальное преобр
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 1u_121$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 12u_11$	1	9 (левая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 2r1l$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = l1r2$	1	9 (правая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 1u_112$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 21u_11$	2	9 (левая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 2r2r$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = r2r2$	2	9 (правая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 1u_121$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 12u_11$	3	9 (левая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 2r1l$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = l1r2$	3	9 (правая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 1u_112$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 21u_11$	4	9 (левая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 2r1l$	$\text{Type}(X_3) = \mathbb{B}, \text{FBoss}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = l1r2$	4	9 (правая)

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

Пусть слово W подходит под один из перечисленных случаев. Тогда мы знаем с каким случаем мы имеем дело, а также какую конфигурацию имеет путь. Слово W должно кодировать этот путь, то есть W должно входить в Q . Если слово W не удовлетворяет этому условию, например, если код средней вершины не такой, какой должен быть исходя из конфигурации пути, то W кодирует невозможный путь. В этом случае $W = 0$, так как мы ввели обнуляющие соотношения для всех достаточно коротких невозможных путей.

Если же W входит в Q , то к W можно применить соотношение введенное нами, из числа указанных справа на рисунке 58, и получить код для другой части локального преобразования. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.9. Случай цепи $\mathbb{C}1$; преобразования 7 и 10. В правой части рисунка 59 изображены локальные преобразования 7 и 10.

Итак, зафиксируем \mathbb{C} -узел с некоторым окружением и начальниками. Различные случаи расположения интересующих нас путей показаны в левой части рисунка 59.

Зная окружение и начальников центрального \mathbb{C} -узла, можно вычислить базовые окружения и начальников вершин Z, Y, F, J во всех случаях расположения. То есть мы можем определить коды вершин Z, Y, F, J во всех четырех случаях расположения и выписать определяющие соотношения для локальных преобразований 7 и 10.

Уровни боковых вершин (то есть J и Z) во всех случаях равны 1.

Соотношения для преобразования 7.

Назначение Z . Для каждого из четырех случаев расположения, обозначим символом Z буквы в алфавите, соответствующие кодам вершин Z с заданными типом, уровнем,

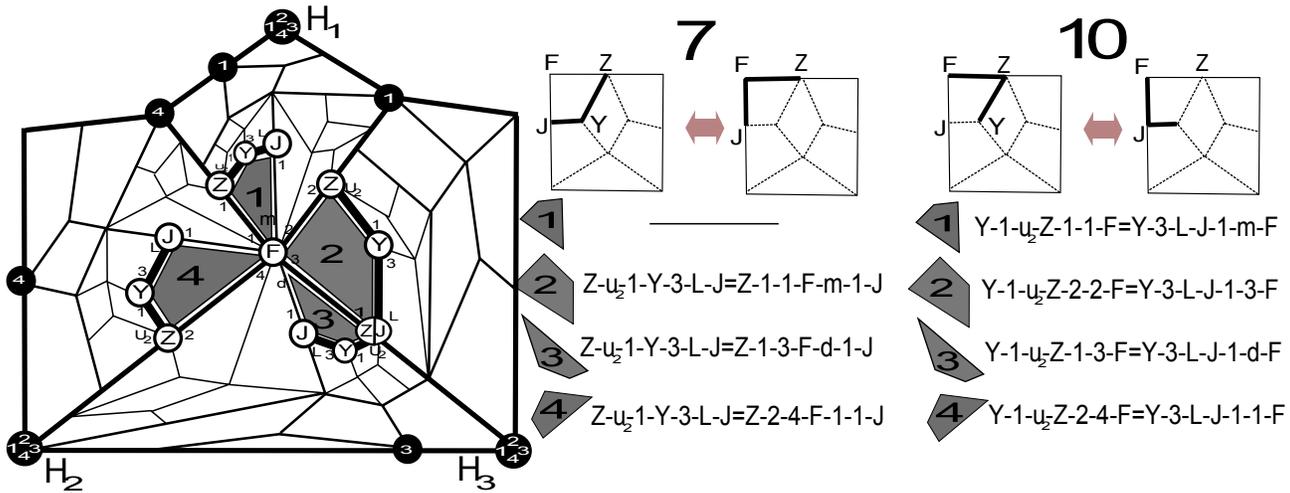


Рис. 59. Случаи расположения пути вокруг S_1 -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

базовым окружением и информацией. То есть эти буквы отличаются друг от друга только различными подклееными окружениями.

Назначение J . Аналогично определим буквы J , как буквы в алфавите, соответствующие кодам вершин J с заданными типом, уровнем, базовым окружением и информацией, и произвольным подклееным окружением.

Назначение Y и F . Буквы Y и F мы определим как конкретные буквы в алфавите с заданными типом, уровнем, базовым окружением и информацией, при пустом подклееном окружении. Мы это делаем, так как путь проходит через F и Y по плоским ребрам.

Теперь для каждого случая и для каждого разрешенного набора букв Z, F, Y, J выпишем соотношение, представленное в правой части рисунка 59. Буквы Z, F, Y, J мы только что определили, а остальные символы в соотношении обозначают входящие и выходящие ребра.

Соотношения для преобразования 10. Тут уже Z и J определены однозначно, а Y и F с точностью для подклееного окружения. В остальном, все аналогично.

Оценка числа соотношений. Мы вводим не более $7FP^2\text{Num}(\mathbb{C})$ соотношений, где F – число различных флагов макроплиток, P – число различных подклеенных окружений, $\text{Num}(\mathbb{C})$ – число вершин типа \mathbb{C} .

В последующих случаях отношения считаются аналогично.

Характеризация. Пусть есть слово W представляющее собой код пути $X_1e_1e_2X_2e_3e_4X_3$. Покажем, как по нему установить, имеем ли мы дело с локальным преобразованием 7 или 10, а также как его провести.

На рисунке 59 отмечены входящие и выходящие ребра в каждом из четырех случаев. Исходя из этого легко выписать свойства пути, позволяющие нам установить, с каким случаем мы имеем дело. Все показано в таблицах 5 и 6.

Ясно, что указанные условия на буквы полностью определяют конфигурацию пути, а также его код, с точностью до подклеенных окружений. Иначе говоря, не бывает никакого другого пути с заданными условиями на буквы, кроме пути, указанного нами.

То есть, для заданной конфигурации есть конечное множество слов Q , которые могли бы кодировать такой путь. Два слова в этом множестве отличаются только кодами подклеенных окружений для крайних вершинных букв.

ТАБЛИЦА 5. Характеристические условия на принадлежность пути к случаю $\mathbb{C}1$ цепи.

Условие на буквы	Симметричное условие (проход в обратном порядке)	случай	Локальное преобр
$\text{Surr}(X_1) = \mathbb{C}11, e_1e_2e_3e_4 = u_1l3l$	$\text{Surr}(X_3) = \mathbb{C}11, e_1e_2e_3e_4 = l3l u_1$	1	7 (левая)
$\text{Surr}(X_1) = \mathbb{C}11, e_1e_2e_3e_4 = 11m1$	$\text{Surr}(X_3) = \mathbb{C}11, e_1e_2e_3e_4 = 1m11$	1	7 (правая)
$\text{Surr}(X_1) = \mathbb{C}12, e_1e_2e_3e_4 = u_2l3l$	$\text{Surr}(X_3) = \mathbb{C}12, e_1e_2e_3e_4 = l3l u_2$	2	7 (левая)
$\text{Surr}(X_1) = \mathbb{C}12, e_1e_2e_3e_4 = 2231$	$\text{Surr}(X_3) = \mathbb{C}12, e_1e_2e_3e_4 = 1322$	2	7 (правая)
$\text{Surr}(X_1) = \mathbb{C}13, e_1e_2e_3e_4 = u_2l3l$	$\text{Surr}(X_3) = \mathbb{C}13, e_1e_2e_3e_4 = l3l u_2$	3	7 (левая)
$\text{Surr}(X_1) = \mathbb{C}13, e_1e_2e_3e_4 = 13d1$	$\text{Surr}(X_3) = \mathbb{C}13, e_1e_2e_3e_4 = 1d31$	3	7 (правая)
$\text{Surr}(X_1) = \mathbb{C}14, e_1e_2e_3e_4 = u_2l3l$	$\text{Surr}(X_3) = \mathbb{C}14, e_1e_2e_3e_4 = l3l u_2$	4	7 (левая)
$\text{Surr}(X_1) = \mathbb{C}14, e_1e_2e_3e_4 = 2411$	$\text{Surr}(X_3) = \mathbb{C}14, e_1e_2e_3e_4 = 1142$	4	7 (правая)

ТАБЛИЦА 6. Характеристические условия на принадлежность пути к случаю $\mathbb{C}1$ цепи.

Условие на буквы	Симметричное условие (проход в обратном порядке)	случай	Локальное преобр
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}11, e_1e_2e_3e_4 = 1u_111$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}11, e_1e_2e_3e_4 = 11u_11$	1	10 (левая)
$\text{Type}(X_1) = \mathbb{A}, \text{FBoss}(X_1) = \mathbb{C}11, e_1e_2e_3e_4 = 3l1m$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}11, e_1e_2e_3e_4 = m1l3$	1	10 (правая)
$\text{Type}(X_1) = \mathbb{A}, \text{FBoss}(X_1) = \mathbb{C}12, e_1e_2e_3e_4 = 1u_222$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}12, e_1e_2e_3e_4 = 22u_21$	2	10 (левая)
$\text{Type}(X_1) = \mathbb{A}, \text{FBoss}(X_1) = \mathbb{C}12, e_1e_2e_3e_4 = 3l13$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}12, e_1e_2e_3e_4 = 31l3$	2	10 (правая)
$\text{Type}(X_1) = \mathbb{A}, \text{FBoss}(X_1) = \mathbb{C}13, e_1e_2e_3e_4 = 1u_2l3$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}13, e_1e_2e_3e_4 = 3l u_21$	3	10 (левая)
$\text{Type}(X_1) = \mathbb{A}, \text{FBoss}(X_1) = \mathbb{C}13, e_1e_2e_3e_4 = 3l1d$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}13, e_1e_2e_3e_4 = d1l3$	3	10 (правая)
$\text{Type}(X_1) = \mathbb{A}, \text{FBoss}(X_1) = \mathbb{C}14, e_1e_2e_3e_4 = 1u_224$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}14, e_1e_2e_3e_4 = 42u_21$	4	10 (левая)
$\text{Type}(X_1) = \mathbb{B}, \text{FBoss}(X_1) = \mathbb{C}14, e_1e_2e_3e_4 = 3l1ld$	$\text{Type}(X_3) = \mathbb{A}, \text{FBoss}(X_3) = \mathbb{C}14, e_1e_2e_3e_4 = ld1l3$	4	10 (правая)

Восстановление кода. Зная код F , все остальные коды легко вычисляются. То есть остается разобраться, как определить код F по известным кодам остальных вершин, в 7 локальном преобразовании, случаи расположения 2, 3, 4. Окружение F совпадает с U -частью окружения H_1 первого начальника Z . Заметим, что Z в случае 3 и 4, и J в случае 2 имеют тех же начальников, что и F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

Пусть слово W подходит под один из перечисленных случаев. Аналогично предыдущему параграфу, мы можем определить с каким случаем мы имеем дело, а также какую конфигурацию имеет путь. Слово W должно кодировать этот путь, то есть W должно входить в Q . В противном случае W приводится к нулю.

Если же W входит в Q , то к W можно применить соотношение введенное нами, из числа указанных справа на рисунке 59, и получить код для другой части локального преобразования.

Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.10. **Случай цепи C_2 ; преобразования 8 и 9.** В правой части рисунка 60 изображены локальные преобразования 8 и 9.

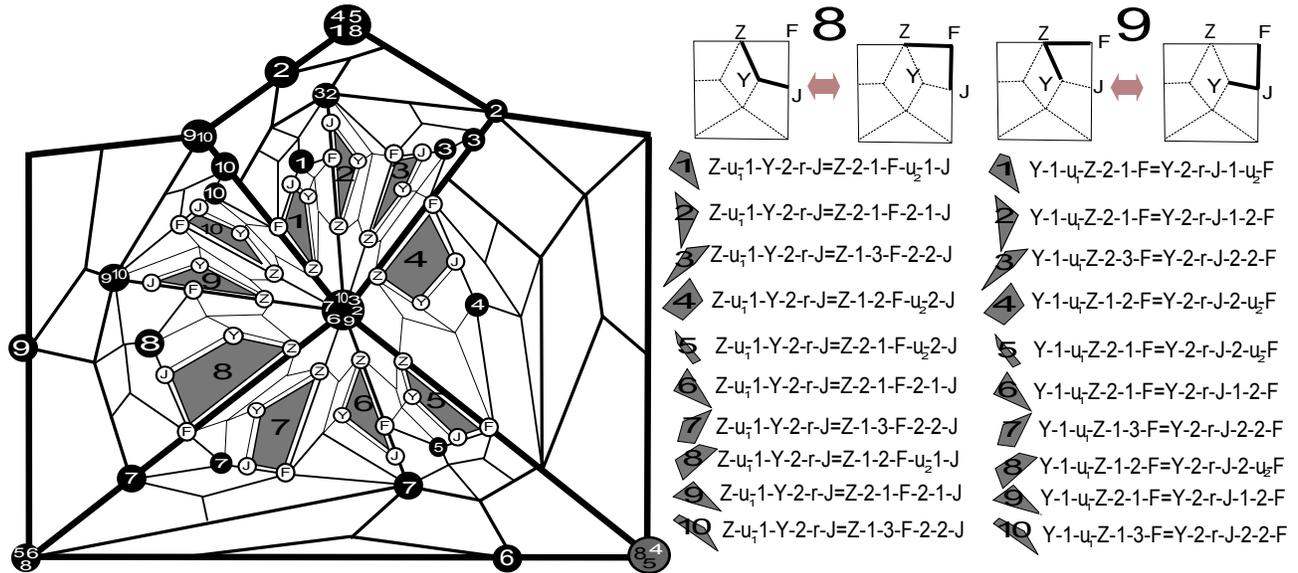


Рис. 60. Случаи расположения пути вокруг C_2 -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Аналогично предыдущим случаям, мы можем ввести определяющие соотношения. Фиксируем C -узел с базовым окружением и тремя начальниками. Ясно, что зная код центрального узла, типы, уровни и окружения вершин Z, Y, F, J можно вычислить (с точностью до подклееными окружениями крайних вершин в пути).

Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная код центрального C -узла мы можем найти их типы, уровни и окружения. То есть коды вершин Z, Y, F, J во всех десяти случаях мы можем назвать явно, с точностью до подклеенных окружений.

Это позволяет ввести определяющие отношения, записанные в правой части рисунка 60.

Характеризация. Пусть есть слово W представляющее собой код пути $X_1e_1e_2X_2e_3e_4X_3$. Аналогично предыдущим случаям, мы можем выписать характеризующую последовательность кодов вершин и входящих-выходящих ребер. Это позволит для любого слова определить, является ли он кодом локального преобразования 8 или 9.

Мы не будем приводить здесь таблицу как для случая C_1 -цепи, ее можно составить полностью аналогично, опираясь на рисунок 60.

Восстановление кода. Учитывая характеристику, мы знаем, с каким случаем расположения имеем дело. Кроме того, ясно, что окружение каждой из четырех вершин можно определить во всех десяти случаях. Заметим, что у Z и F одно и то же множество начальников во всех случаях. Первым начальником Y всегда является Z , а тип второго очевиден из расположения. Остается определить начальников J . В случаях 1, 4, 5, 8 это просто

F , а в остальных случаях у J либо те же начальники что и у F , либо один начальник, совпадающий с первым начальником F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

Таким образом, если слово W подходит под один из перечисленных случаев то к нему можно применить соотношение введенное нами, из числа указанных справа на рисунке 60, и получить код для другой части локального преобразования.

8.11. Случай цепи С2; преобразования 7 и 10. В правой части рисунка 61 изображены локальные преобразования 7 и 10.

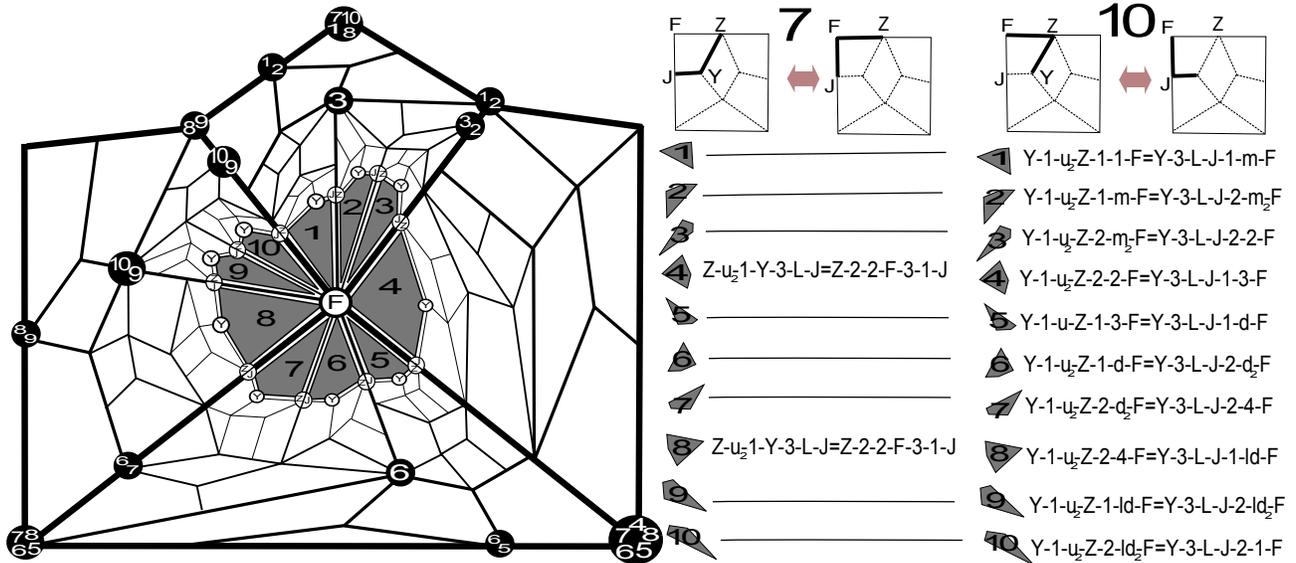


Рис. 61. Случаи расположения пути вокруг С2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная код центрального C -узла мы можем найти их окружения. То есть коды вершин Z, Y, F, J (включающие типы, уровни, окружения и информацию), во всех десяти случаях мы можем назвать явно, с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 61.

Характеризация. Составление таблиц полностью аналогично случаю С1-цепи.

Восстановление кода. Рассмотрим случаи расположения 4 и 8 для преобразования 7. Заметим, что Z в случае 8 и J в случае 4 имеют тех же начальников, что и F . Окружение F совпадает с U -частью его первого начальника. Значит, зная коды Z, J, Y , можно вычислить код F . Зная же код F , очевидно, все остальные вершины тоже вычисляются. Для преобразования 10, зная коды F, J, Y , можно вычислить Z , а зная F, Z, Y , можно вычислить J .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

Таким образом, к W можно применить соотношение введенное нами, из числа указанных справа на рисунке 61, и получить код для другой части локального преобразования.

8.12. **Случай цепи C3; преобразования 7, 8, 9, 10.** Случай C3-цепи полностью аналогичен случаю C2-цепи, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают C3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для C2 случая, то есть $32FP^2\text{Num}(\mathbb{C})$.

8.13. **Случай цепи W1; преобразования 8 и 9.** В правой части рисунка 62 изображены локальные преобразования 8 и 9.

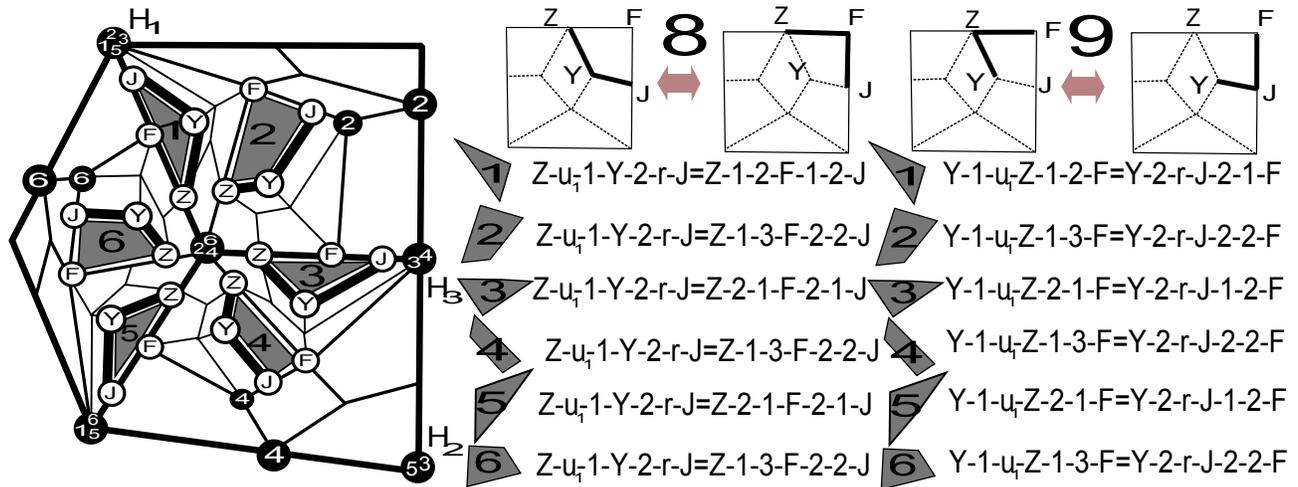


Рис. 62. Случаи расположения пути вокруг W1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Аналогично предыдущим случаям, мы можем ввести определяющие соотношения. Фиксируем вершину типа \mathbb{W} и ее начальников, вершины H_1 и H_2 , причем от H_2 мы фиксируем только тип. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Числа в них обозначают номер случая расположения. Заметим, что мы можем найти их окружения (кроме вершины H_2). Например, H_3 вычисляется с помощью RightFromW . Окружения для вершин Z, Y, F, J в каждом случае вычисляются с учетом известных нам окружения H_1 и типа H_2 . То есть что коды вершин Z, Y, F, J во всех случаях мы можем назвать явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 62.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. Пусть мы знаем коды трех вершин из Z, J, Y, F . Поскольку мы можем установить, с каким именно случаем расположения мы имеем дело, то и окружение оставшейся вершины мы можем вычислить.

F и Z имеют общее множество начальников во всех случаях. Первым начальником Y всегда является Z , тип второго очевиден из расположения в каждом случае. В случаях 1, 3, 5 у Y те же начальники что и у F , в случаях 2, 4, 6 у J первый начальник как у F , а второй – как третий у F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

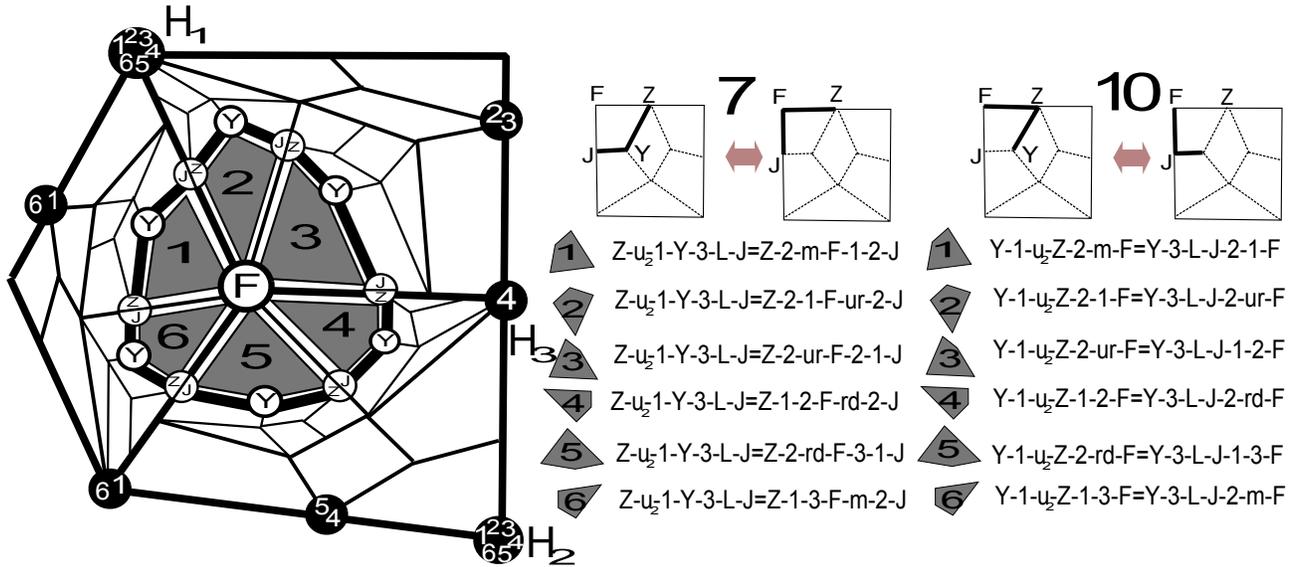


Рис. 63. Случаи расположения пути вокруг $\mathbb{V}1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

8.14. **Случай цепи $\mathbb{V}1$; преобразования 7 и 10.** В правой части рисунка 63 изображены локальные преобразования 7 и 10.

Аналогично предыдущим случаям, мы можем ввести определяющие соотношения. Фиксируем вершину типа \mathbb{V} и ее начальников, вершины H_1 и H_2 , причем от H_2 мы фиксируем только тип. Черными кругами отметим вершины, являющиеся начальниками вершин Z , Y , F , J . Числа в них обозначают номер случая расположения. Заметим, что мы можем найти их типы, уровни и окружения (кроме вершины H_2 , для нее только тип). Типы, уровни и окружения для вершин Z , Y , F , J , в каждом случае вычисляются с учетом известных нам окружения H_1 и типа H_2 . То есть что коды вершин Z , Y , F , J во всех случаях мы можем назвать явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 63.

Характеризация. Составление таблиц полностью аналогично случаю $\mathbb{C}1$ -цепи.

Восстановление кода. Для локального преобразования 7, если нам нужно вычислить код F по известным кодам Z , J , Y , можно заметить, что хотя бы у одной вершины из Z , J , Y начальником будет F , а у какой-либо другой начальники будут совпадать с F . Если же F нам известно, все остальные вершины, очевидно, вычисляются.

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

8.15. **Случай цепи $\mathbb{V}2$; преобразования 8 и 9.** В правой части рисунка 64 изображены локальные преобразования 8 и 9.

Введение определяющих соотношений полностью аналогично предыдущим случаям, можно легко проверить, что зная центральный \mathbb{V} -узел, коды всех четырех вершин в каждом из случаев расположения легко можно выписать. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 64.

Характеризация. Составление таблиц полностью аналогично случаю $\mathbb{C}1$ -цепи.

Восстановление кода. Поскольку мы можем установить, с каким именно случаем расположения мы имеем дело, то и окружение каждой из четырех вершин мы можем

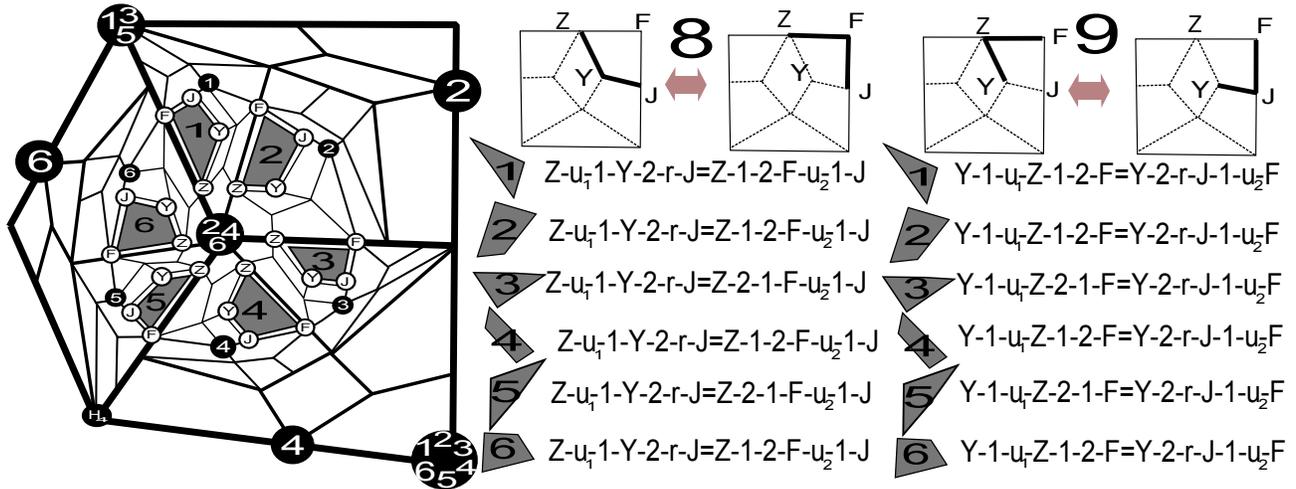


Рис. 64. Случаи расположения пути вокруг В2-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

вычислить. Кроме того, заметим, что каждая из четырех вершин Z, J, Y, F в каждом из случаев либо имеет начальников среди других трех вершин, либо у нее общие начальники с какой-то из трех других вершин. Значит, в каждом из случаев, зная коды трех вершин, можно вычислить код четвертой.

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

8.16. Случай цепи $\mathbb{B}2$; преобразования 7 и 10. В правой части рисунка 65 изображены локальные преобразования 7 и 10.

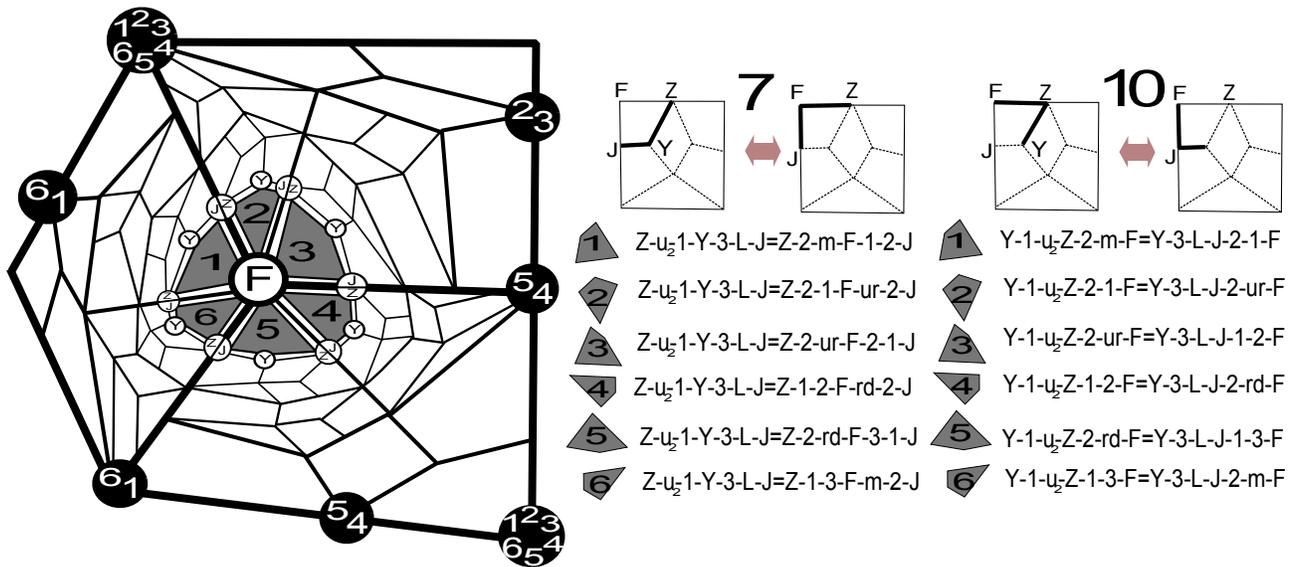


Рис. 65. Случаи расположения пути вокруг В2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Зафиксируем вершину типа \mathbb{B} и ее начальников. Аналогично предыдущим случаям, мы можем вычислить коды всех вершин и ввести определяющие соотношения.

Характеризация. Составление таблиц полностью аналогично случаю $\mathbb{C}1$ -цепи.

Восстановление кода. Для локального преобразования 7, если нам нужно вычислить код F по известным кодам Z, J, Y можно заметить, что хотя бы у одной вершины из Z, J, Y начальники будут совпадать с F . Если же F нам известно, все остальные вершины, очевидно, вычисляются.

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

8.17. **Случай цепи A_0 ; преобразования 8 и 9.** В правой части рисунка 66 изображены локальные преобразования 8 и 9.

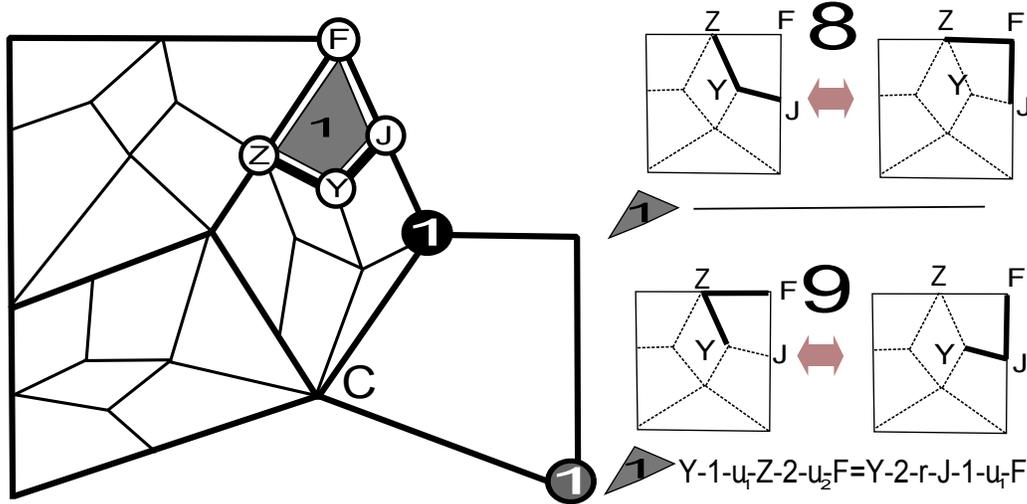


Рис. 66. Случаи расположения пути вокруг A_0 -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину X второго уровня с типом UL, UR или U . Мы будем вводить соотношения только для преобразования 9. Путь в преобразовании 8 подпадает под мертвый паттерн, для него определяющие соотношения вводить не требуется. По коду X вычисляются коды остальных трех вершин, например тип второго начальника J – вершины в правом нижнем углу может быть найден с помощью функции $BottomRightType(X)$.

Характеризация. Составление таблиц полностью аналогично случаю C_1 -цепи.

Восстановление кода. В преобразовании 9, зная код F , мы можем вычислить и J и Z .

8.18. **Случай цепи A_0 ; преобразования 7 и 10.** В правой части рисунка 67 изображены локальные преобразования 7 и 10.

Зафиксируем вершину типа A и ее начальника (вершина, отмеченная черным кругом с “1”, уровень которой равен 2). Заметим, что зная окружение этой вершины, мы можем вычислить коды вершин Z, Y, F, J с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 67.

Характеризация. Составление таблиц полностью аналогично случаю C_1 -цепи.

Восстановление кода. Вершина, отмеченная черным кругом с 1, является единственным начальником Z, F, J . Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

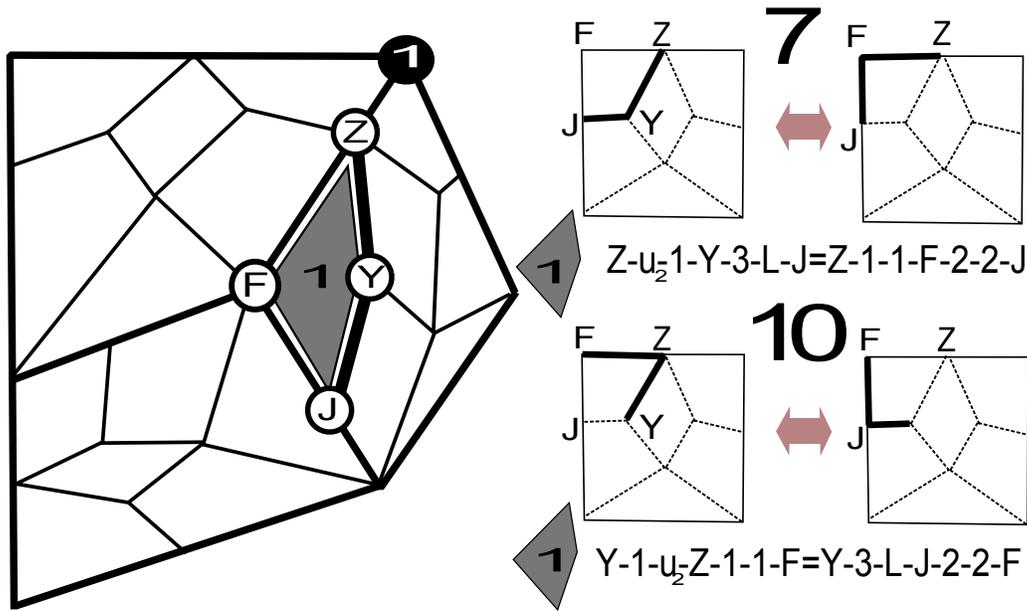


Рис. 67. Случаи расположения пути вокруг A0-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

8.19. Случай цепи A1; преобразования 8 и 9. В правой части рисунка 68 изображены локальные преобразования 8 и 9.

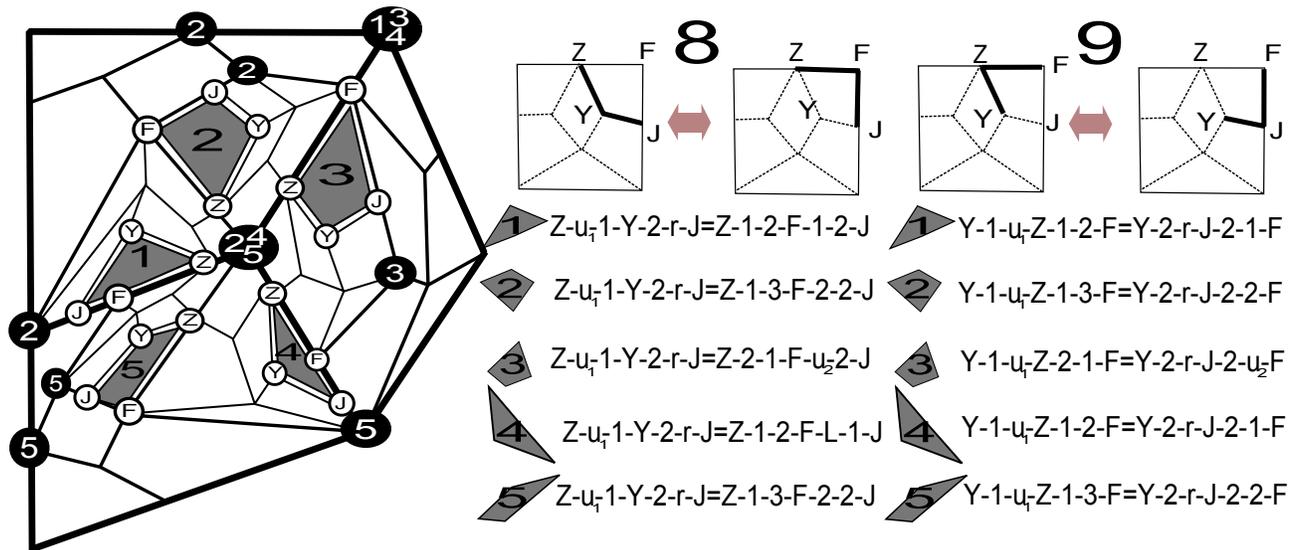


Рис. 68. Случаи расположения пути вокруг A1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа \mathbb{A} и ее начальника (третьего уровня). Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центрального \mathbb{A} -узла, мы можем найти их типы, уровни и окружения. Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J во всех случаях мы можем назвать явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 68.

Характеризация. Составление таблиц полностью аналогично случаю С1-цепи.

Восстановление кода. Поскольку мы можем установить, с каким именно случаем расположения мы имеем дело, то и окружение каждой из вершин мы можем вычислить, зная остальные три. Кроме того, заметим, что каждая из четырех вершин Z, J, Y, F в каждом из случаев либо имеет начальников среди других трех вершин, либо у нее общие начальники с какой-то из трех других вершин. Значит, в каждом из случаев, зная коды трех вершин, можно вычислить код четвертой.

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот.

8.20. **Случай цепи А1; преобразования 7 и 10.** В правой части рисунка 69 изображены локальные преобразования 7 и 10.

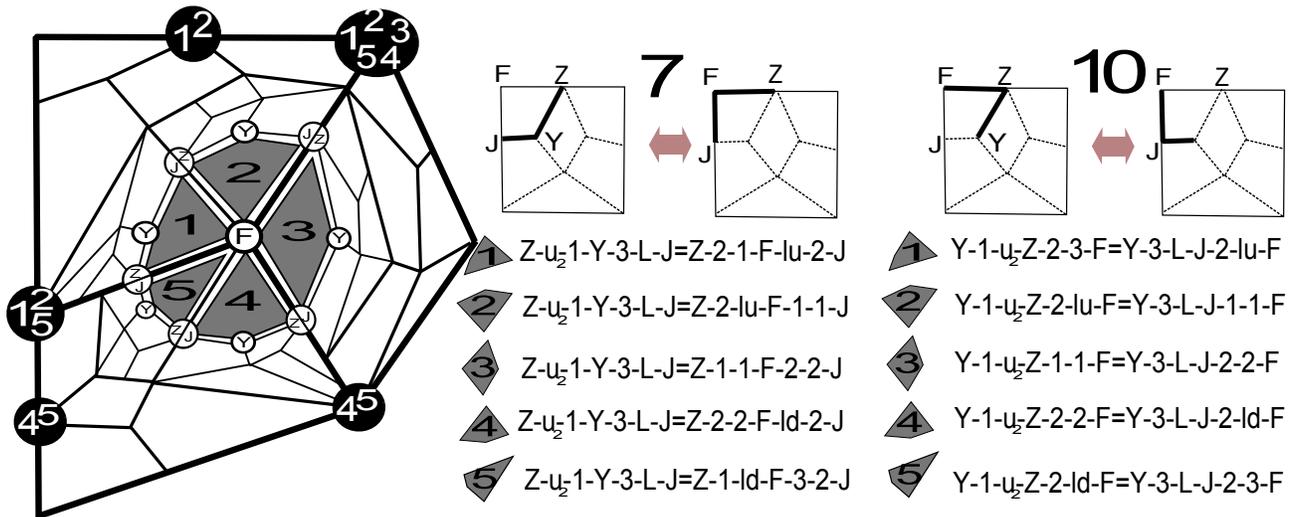


Рис. 69. Случаи расположения пути вокруг А1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа Δ и ее начальника. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение этой центральной Δ -вершины, мы можем вычислить коды вершин Z, Y, F, J с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 69.

Характеризация. Составление таблиц полностью аналогично случаю С1-цепи.

Восстановление кода. Вершина F имеет общего начальника с одной из вершин J или Z . То есть, зная код пути ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.21. **Случай цепи А2; преобразования 8 и 9.** В правой части рисунка 70 изображены локальные преобразования 8 и 9.

Введение определяющих соотношений полностью аналогично случаю с А1-цепью. Также аналогично можно установить, что зная код пути ZYJ , можно вычислить код пути

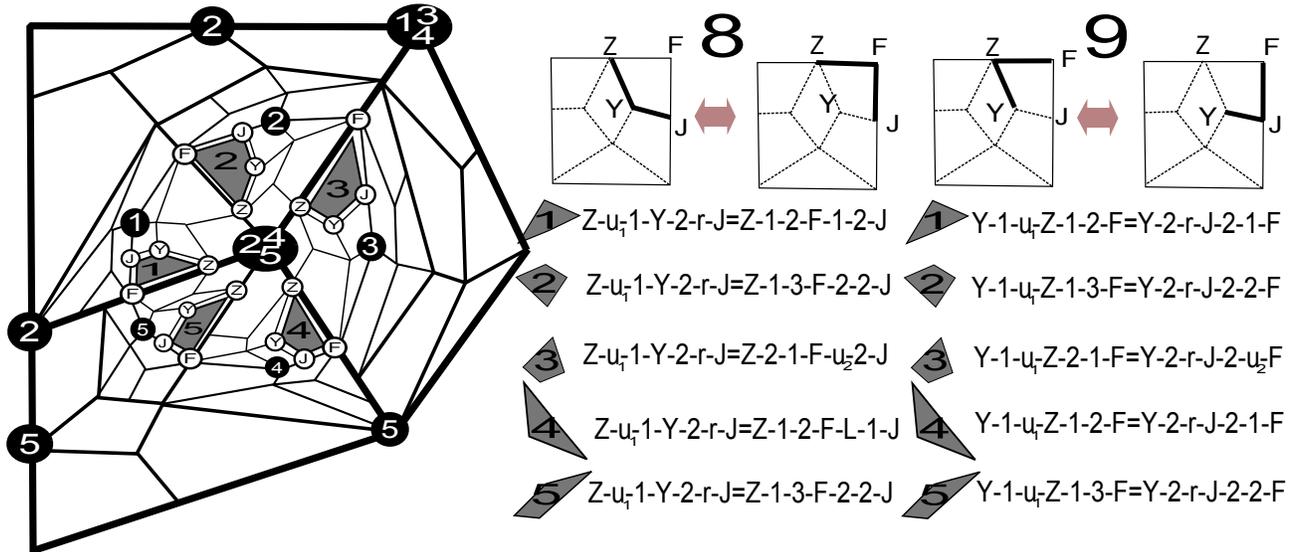


Рис. 70. Случаи расположения пути вокруг A2-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Аналогично можно установить последовательности букв, характеризующие данную цепь и локальные преобразования.

8.22. Случай цепи A2; преобразования 7 и 10. В правой части рисунка 71 изображены локальные преобразования 7 и 10.

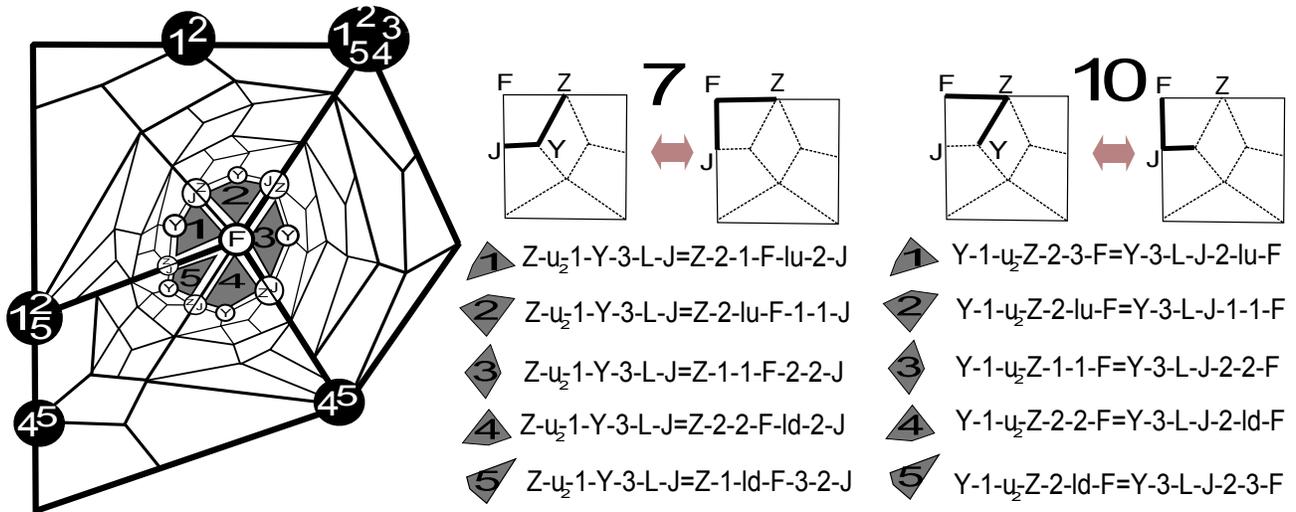


Рис. 71. Случаи расположения пути вокруг A2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Введение определяющих соотношений полностью аналогично случаю с A1-цепью. Также аналогично можно установить, что зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Аналогично можно установить последовательности букв, характеризующие данную цепь и локальные преобразования.

8.23. **Случай цепи UL1; преобразования 8 и 9.** В правой части рисунка 72 изображены локальные преобразования 8 и 9.

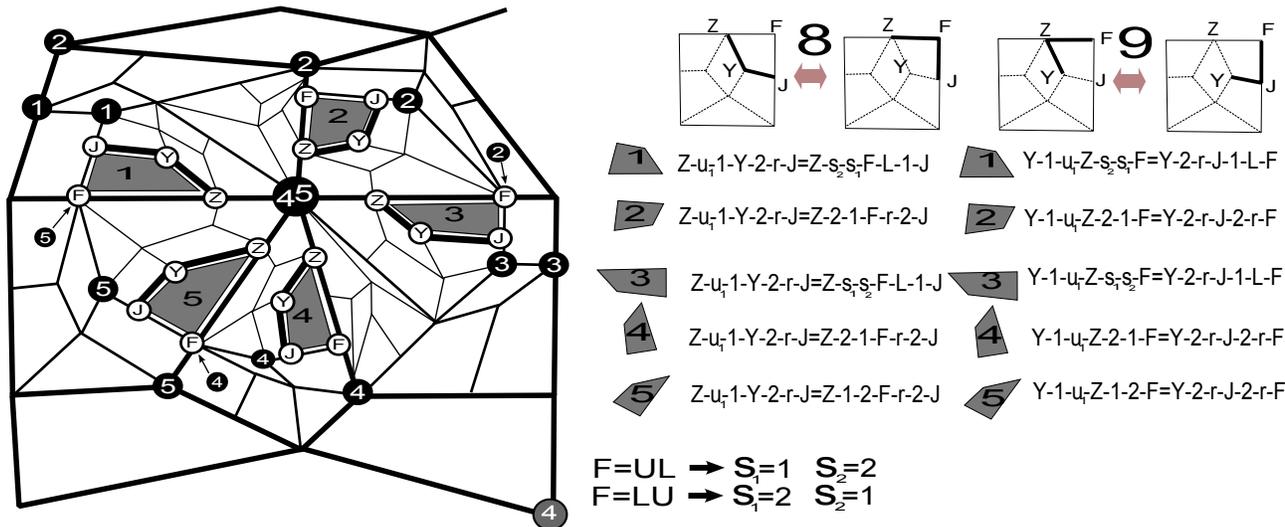


Рис. 72. Случаи расположения пути вокруг UL1-цепи и определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа UL третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Три круга попадают в F -узлы других расположений, это отмечено стрелками. Заметим, что зная окружение центрального UL -узла, мы можем найти типы, уровни и окружения вершин с черными кругами (кроме вершины в правом нижнем углу, для нее только тип). Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J во всех случаях мы можем назвать явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 72.

Характеризация. Составление таблиц полностью аналогично случаю $C1$ -цепи.

Восстановление кода. Поскольку мы можем установить, с каким именно случаем расположения мы имеем дело, то окружение оставшейся вершины мы можем вычислить, зная коды остальных трех. Кроме того, заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. У Y во всех случаях первым начальником является Z , а тип второго в каждом случае ясен из расположения.

Начальники J вычисляются следующим образом: случай 1 – это $Prev(F)$; случай 2 – это 1-цепь вокруг $TypeBottomLeft.FBoss(Z)$, указатель 1 (первый начальник), и \mathbb{A} -тип второго начальника; случай 3 – это $Prev(F)$; случай 4 – это 0-цепь с указателем 1 вокруг \mathbb{A} , окружения которого как U -часть $Fboss(Z)$ (первый начальник), и тип второго начальника \mathbb{B} ; случай 5 – это первый $Plus.FBoss(Z)$, тип второго \mathbb{A} .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

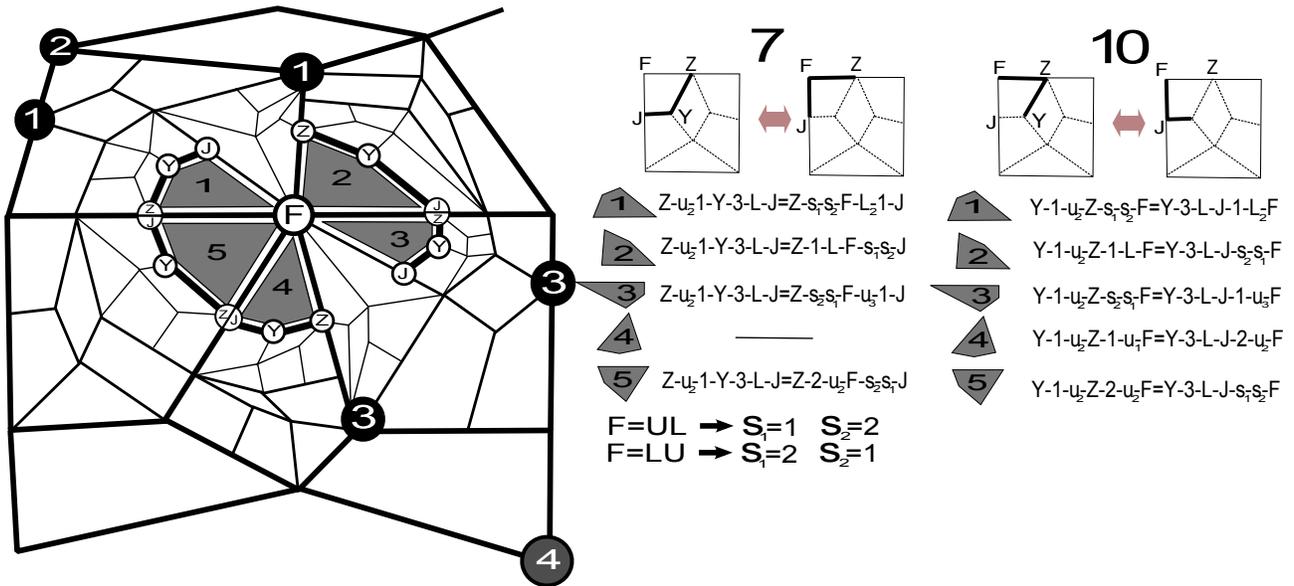


Рис. 73. Случаи расположения пути вокруг UL1-цепи и определяющие соотношения для локальных преобразований 7 и 10

8.24. **Случай цепи UL1; преобразования 7 и 10.** В правой части рисунка 73 изображены локальные преобразования 7 и 10.

Фиксируем некоторую вершину типа UL третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центральной вершины (которая совпадает с F), мы можем вычислить коды вершин Z, Y, F, J с точностью до подклеенных окружений во всех случаях. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 73.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. Вершина F имеет общего начальника с одной из вершин J или Z , в каждом из случаев, кроме случая 4 (а в случае 4 нам ее не надо вычислять). То есть, зная код пути ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.25. **Случай цепи UL2; преобразования 8 и 9.** В правой части рисунка 74 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа UL третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центрального UL-узла, мы можем найти их типы, уровни и окружения. Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J , во всех случаях мы можем назвать явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 74.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. Заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных

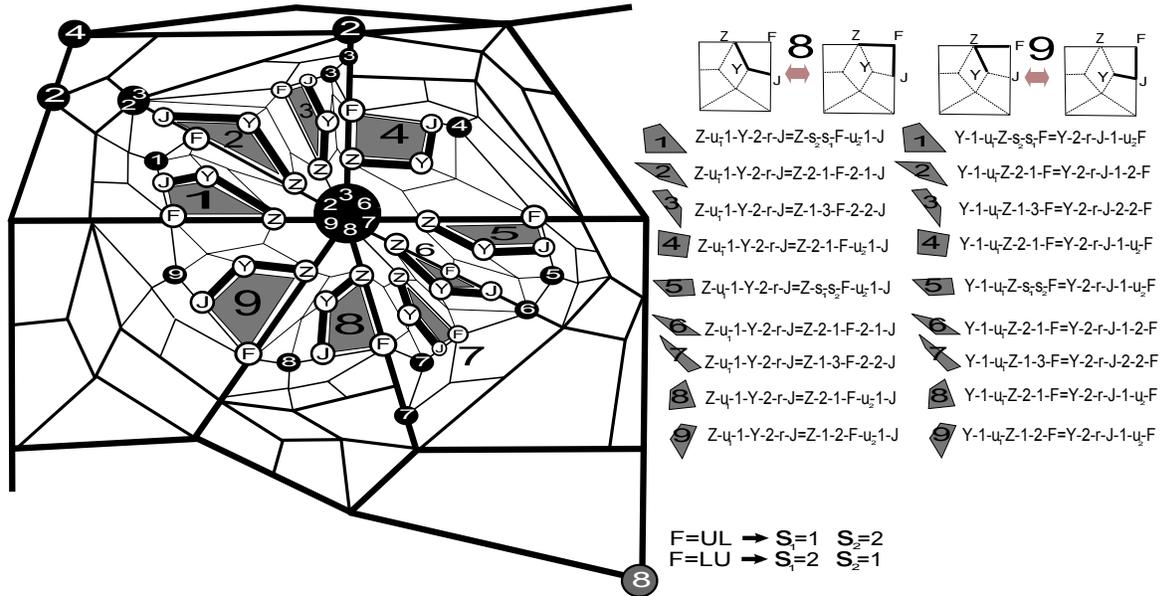


Рис. 74. Случаи расположения пути вокруг UL2-цепи и определяющие соотношения для локальных преобразований 8 и 9

трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае легко виден из рисунка.

Начальники J вычисляется следующим образом: в случаях 1, 4, 5, 8, 9 – единственный начальник, узел F ; в случаях 2, 6 – начальники те же что и у F ; в случаях 3 и 7 – первый и второй начальники как первый и третий у F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.26. Случай цепи UL2; преобразования 7 и 10. В правой части рисунка 75 изображены локальные преобразования 7 и 10.

Фиксируем некоторую вершину типа UL третьего уровня и ее начальников. Черными и серыми кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центральной вершины (совпадающей с F), мы можем вычислить коды вершин Z, Y, F, J , с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 75.

Для локального преобразования 7 пути в случаях 2, 3, 6, 7, 8 удовлетворяют условиям мертвого паттерна, и в этих случаях мы соотношения не вводим.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. В случаях 1, 4, 5, 9 вершина F имеет общего начальника с одной из вершин J или Z . То есть, зная код пути ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.27. Случай цепи UL3; преобразования 7, 8, 9, 10. Случай UL3-цепи полностью аналогичен UL2 случаю, соотношения выглядят идентично, только кодировки вершин J ,

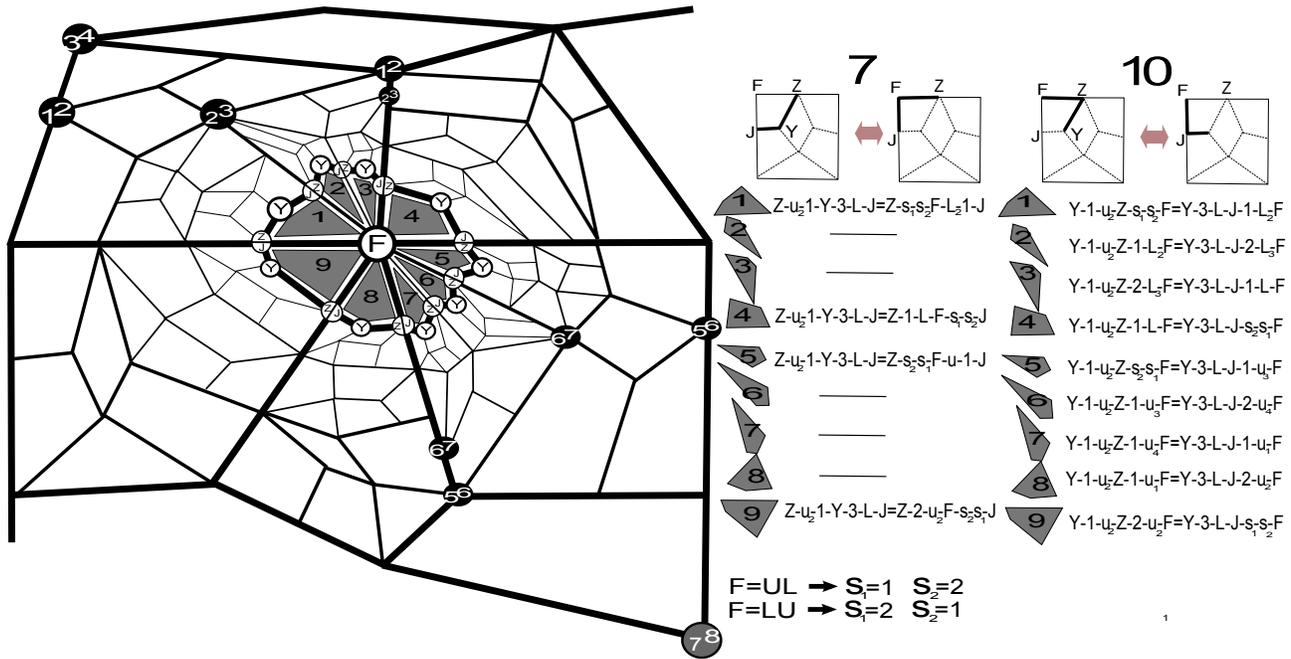


Рис. 75. Случаи расположения пути вокруг UL2-цепи и определяющие соотношения для локальных преобразований 7 и 10

F, Z, Y отвечают UL3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для UL2 случая.

8.28. Случай цепи UR1; преобразования 8 и 9. В правой части рисунка 76 изображены локальные преобразования 8 и 9.

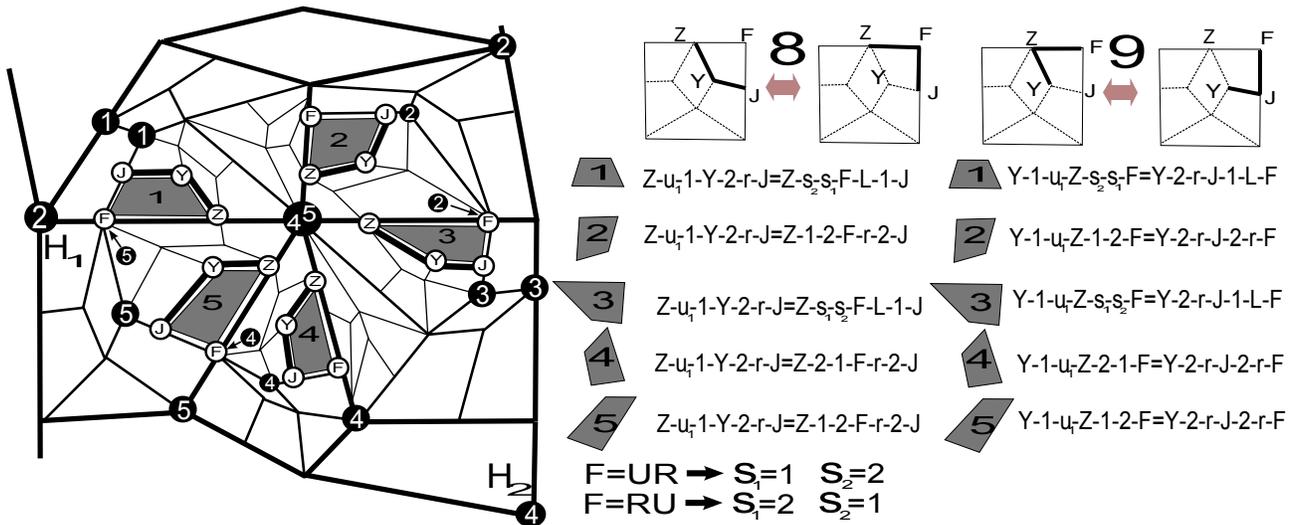


Рис. 76. Случаи расположения пути вокруг UR1-цепи и определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа UR третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Три круга попадают в F -узлы других расположений, это отмечено стрелками. Заметим, что зная окружение центрального UR-узла, мы можем найти их типы, уровни и окружения (Кроме вершин H_1

и H_2 , для которых можем найти тип). Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J , во всех случаях мы можем назвать явно, с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 76.

Характеризация. Составление таблиц полностью аналогично случаю $C1$ -цепи.

Восстановление кода. Заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае виден из рисунка.

Начальники J вычисляется следующим образом: случай 1 – это $\text{Prev}(F)$; случай 2 – это $\text{UpRightChain.FBoss}(Z)$ и \mathbb{B} -тип второго начальника; случай 3 – это $\text{Prev}(F)$; случай 4 – это 0-цепь с указателем 1 вокруг \mathbb{A} , окружения которого как U -часть $\text{Fboss}(Z)$, и тип второго начальника \mathbb{B} ; случай 5 – это $\text{Plus.FBoss}(Z)$ и тип второго начальника \mathbb{B} .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.29. **Случай цепи UR1; преобразования 7 и 10.** В правой части рисунка 77 изображены локальные преобразования 7 и 10.

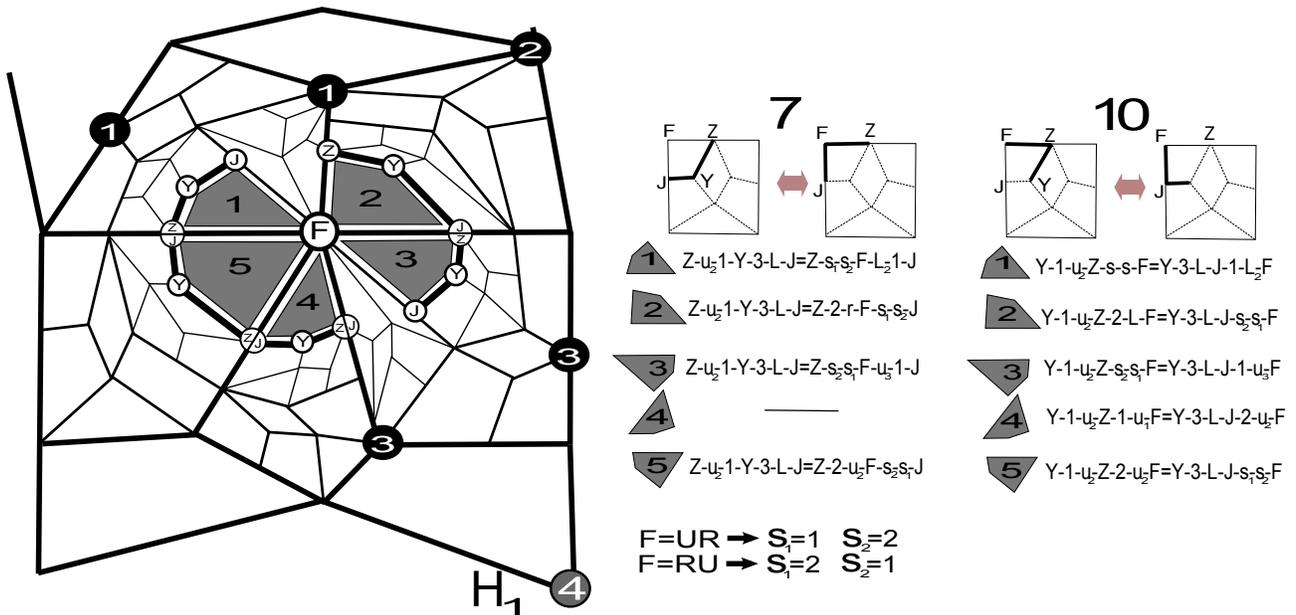


Рис. 77. Случаи расположения пути вокруг UR1-цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа UR третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центральной вершины, мы можем вычислить коды вершин Z, Y, F, J , с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 77.

Характеризация. Составление таблиц полностью аналогично случаю $C1$ -цепи.

Восстановление кода. Вершина F имеет общего начальника с одной из вершин J или Z , в каждом из случаев, кроме случая 4 (а в случае 4 нам ее не надо вычислять). Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. То есть, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.30. **Случай цепи UR2; преобразования 8 и 9.** В правой части рисунка 78 изображены локальные преобразования 8 и 9.

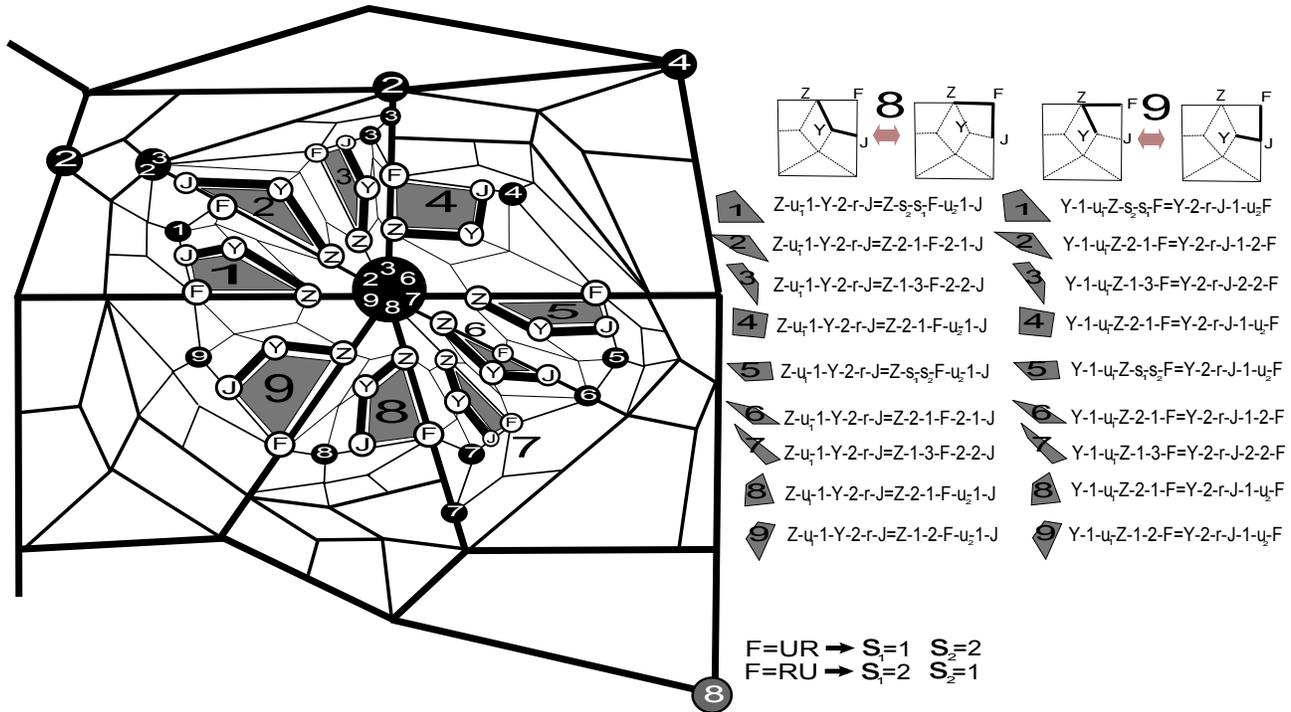


Рис. 78. Случаи расположения пути вокруг UR2-цепи и определяющие соотношения для локальных преобразований 8 и 9

Фиксируем вершину типа UR третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z , Y , F , J . Заметим, что зная окружение центрального UR-узла, мы можем найти их типы, уровни и окружения (для вершины в правом нижнем углу – только тип). Так как начальники вершин Z , Y , F , J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z , Y , F , J , во всех случаях мы можем назвать явно, с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 78.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. Заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляется следующим образом: в случаях 1, 4, 5, 8, 9 – единственный начальник, узел F ; в случаях 2, 6 – начальники те же что и у F ; в случаях 3 и 7 – первый и второй начальники как первый и третий у F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.31. **Случай цепи UR2; преобразования 7 и 10.** В правой части рисунка 79 изображены локальные преобразования 7 и 10.

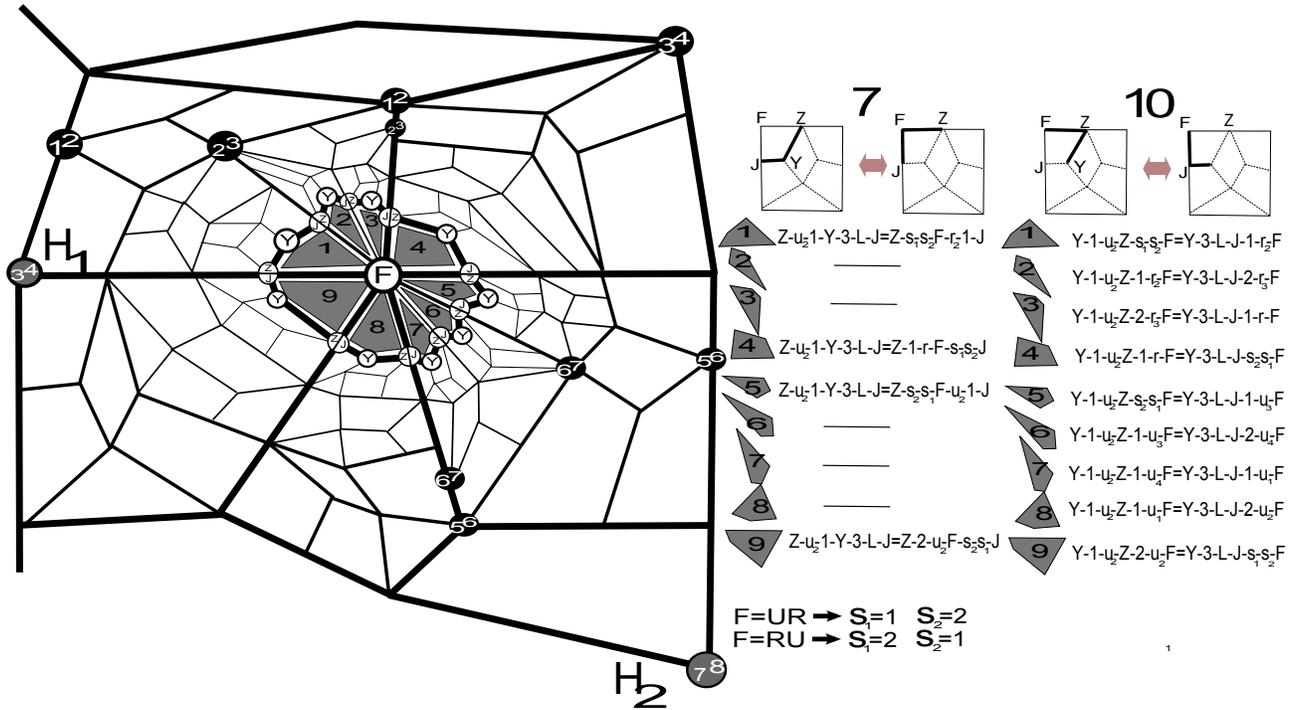


Рис. 79. Случаи расположения пути вокруг UR2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа UR и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение этой вершины, мы можем вычислить коды вершин Z, Y, F, J , с точностью до подклеенных окружений (кроме вершин H_1 и H_2 , для которых можно найти тип). Это позволяет ввести определяющие отношения, записанные в правой части рисунка 79.

Для локального преобразования 7 в случаях 2, 3, 6, 7, 8 пути удовлетворяют условиям мертвого паттерна, и в этих случаях мы соотношения не вводим.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. В случаях 1, 4, 5, 9 вершина F имеет общего начальника с одной из вершин J или Z . То есть, зная код пути ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.32. **Случай цепи UR3; преобразования 7, 8, 9, 10.** Случай UR3-цепи полностью аналогичен UR2 случаю, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают UR3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для UR2 случая.

8.33. **Случай цепи DR1; преобразования 8 и 9.** Аналогично предыдущим случаям, мы можем ввести определяющие соотношения. Сначала зафиксируем вершину типа DR третьего уровня и ее начальников. Всего возможно четыре случая расположения DR-вершины, на четырех внутренних ребрах (типов 2, 3, 5, 6). Все эти случаи расположения показаны на рисунке 80.

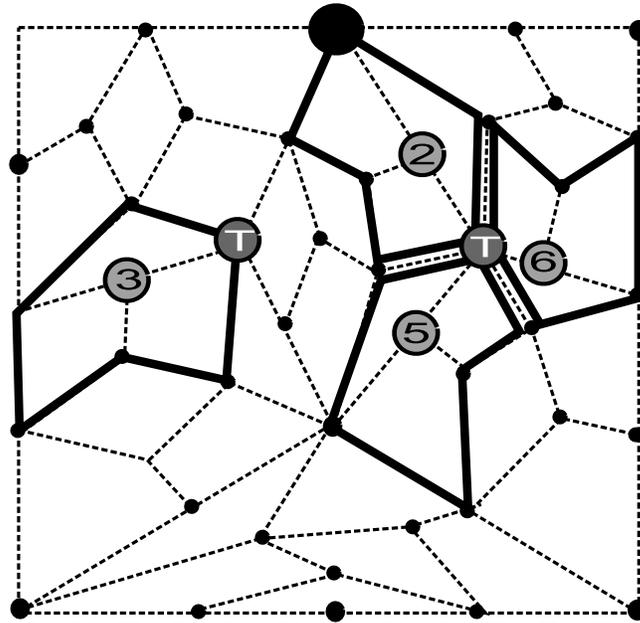


Рис. 80. Случаи расположения DR-вершины

Первым начальником нашей DR-вершины во всех случаях будет вершина в середине верхней стороны. Теперь непосредственно рассмотрим цепь с центром в этой DR-вершине. В правой части рисунка 81 изображены локальные преобразования 8 и 9.

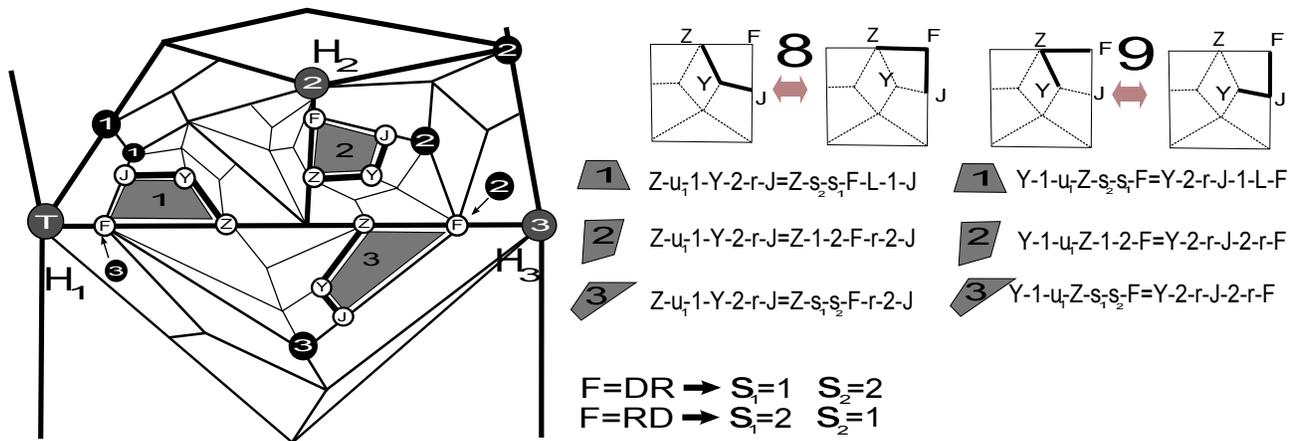


Рис. 81. Случаи расположения пути вокруг DR1-цепи и определяющие соотношения для локальных преобразований 8 и 9

Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Два круга попадают в F -узлы других расположений, это отмечено стрелками. Зная окружение центрального DR-узла, мы можем найти их типы, уровни и окружения (для вершин H_1, H_2, H_3 можем найти только тип).

Тип узла, отмеченного как T , мы можем определить исходя из типа внутреннего ребра, на котором лежит центральный \mathbb{DR} -узел. Если это ребро 3 – то \mathbb{A} , в остальных случаях ($2, 5$ или 6) – \mathbb{B} .

Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J , во всех случаях мы можем назвать явно, с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 81.

Характеризация. Составление таблиц полностью аналогично случаю $S1$ -цепи.

Восстановление кода. Поскольку мы можем установить, с каким именно случаем расположения мы имеем дело, то окружение каждой вершины мы можем вычислить, зная код остальных трех. Кроме того, заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип правого нижнего угла в каждом случае ясен.

Начальники J вычисляется следующим образом: случай 1 – это $\text{Prev}(F)$; случай 2 – первый начальник – $\text{UpRightChain.FBoss}(Z)$ и \mathbb{B} -тип второго; случай 3 – это 1-цепь вокруг узла T с указателями в зависимости от типа ребра центрального \mathbb{DR} -узла: 3 для 3 типа, 1 для 2 типа, 2 для 6 типа, 3 для 5 типа.

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.34. **Случай цепи $\mathbb{DR}1$; преобразования 7 и 10.** В правой части рисунка 82 изображены локальные преобразования 7 и 10.

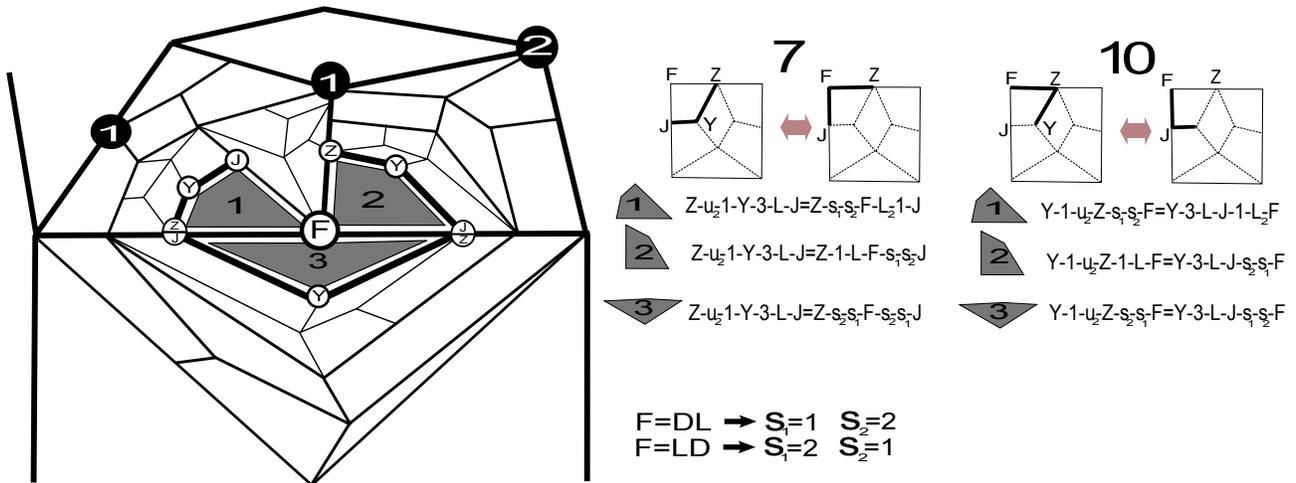


Рис. 82. Случаи расположения пути вокруг $\mathbb{DR}1$ -цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа \mathbb{DR} третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центральной \mathbb{DR} -вершины, мы можем в каждом случае вычислить коды вершин Z, Y, F, J с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 82.

Характеризация. Составление таблиц полностью аналогично случаю $S1$ -цепи.

Восстановление кода. Вершина F имеет общего начальника с одной из вершин J или Z , в каждом из случаев. То есть, зная код пути ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для 10 локального преобразования: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.35. **Случай цепи \mathbb{DR}_2 ; преобразования 8 и 9.** В правой части рисунка 83 изображены локальные преобразования 8 и 9.

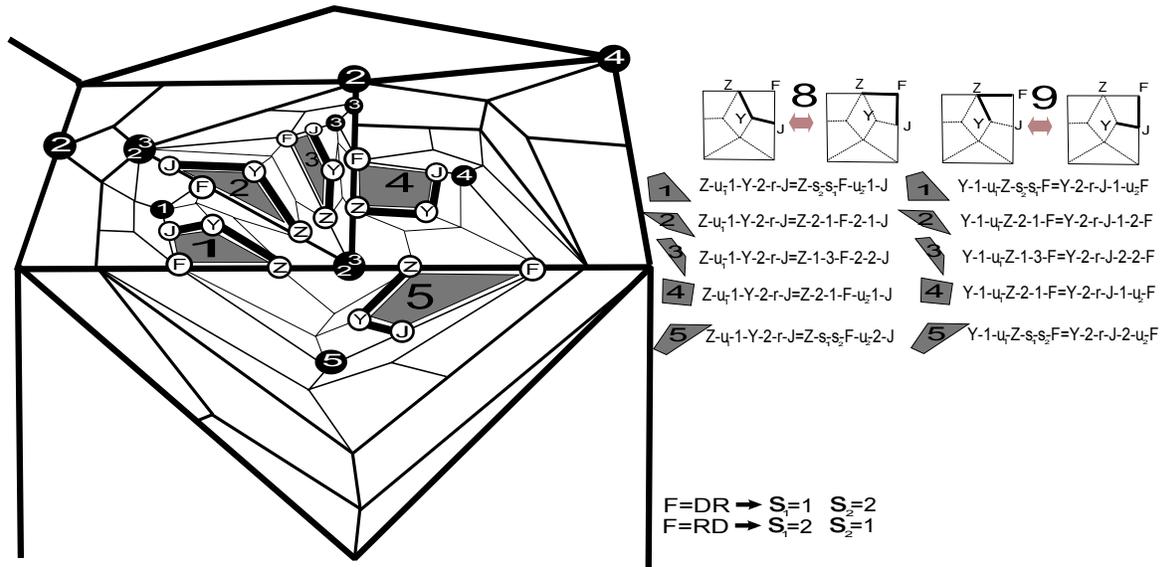


Рис. 83. Случаи расположения пути вокруг \mathbb{DR}_2 -цепи и определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа \mathbb{DR} третьего уровня и ее начальники. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центрального \mathbb{DR} -узла, мы можем найти их типы, уровни и окружения. Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J , во всех случаях мы можем назвать явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 83.

Характеризация. Составление таблиц полностью аналогично случаю $\mathbb{C}1$ -цепи.

Восстановление кода. Заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляется следующим образом: в случаях 1, 4 и 5 – единственный начальник, узел F ; в случае 2 – начальники те же что и у F ; в случае 3 – первый и второй начальники как первый и третий у F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.36. **Случай цепи DR2; преобразования 7 и 10.** В правой части рисунка 84 изображены локальные преобразования 7 и 10.

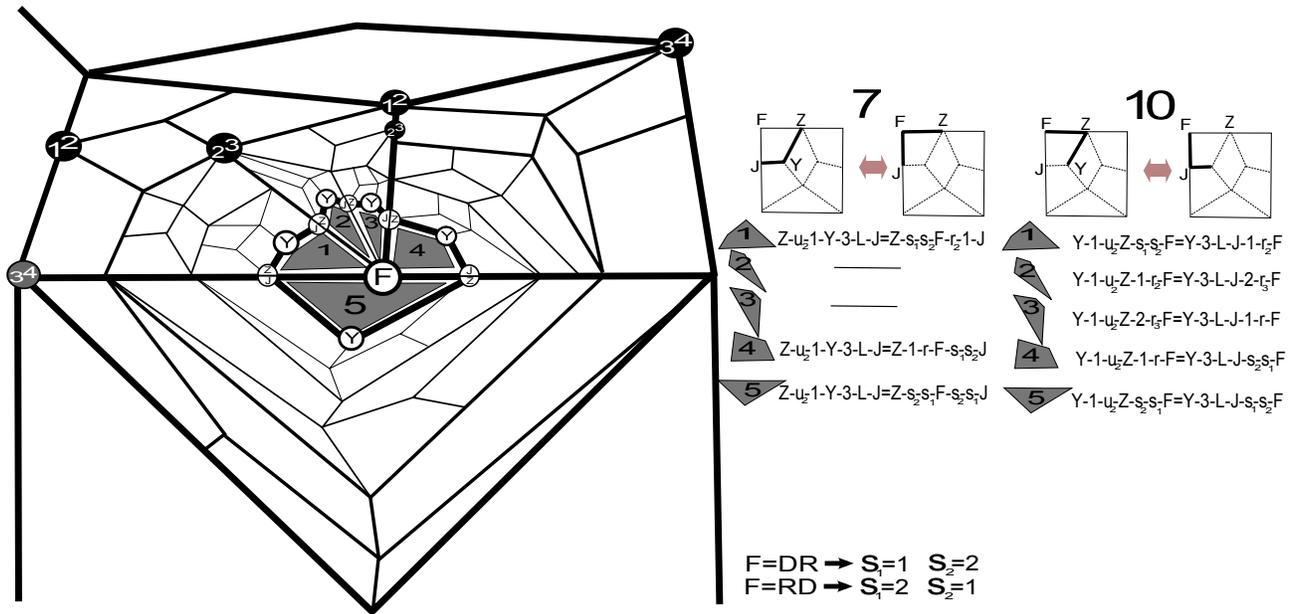


Рис. 84. Случаи расположения пути вокруг DR2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Зафиксируем некоторую вершину типа DR третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение этой вершины, мы можем вычислить коды вершин Z, Y, F, J , с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 84.

Для локального преобразования 7 в случаях 2, 3, 6, 7, 8 пути удовлетворяют условиям мертвого паттерна, и в этих случаях мы соотношения не вводим.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. В случаях 1, 4, 5 вершина F имеет общего начальника с одной из вершин J или Z . То есть, ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.37. **Случай цепи DR3; преобразования 7, 8, 9, 10.** Случай DR3-цепи полностью аналогичен DR2 случаю, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают DR3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для DR2 случая.

8.38. **Случай цепи DL1; преобразования 8 и 9.** В правой части рисунка 85 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа DL третьего уровня и ее начальников. Черными и серыми кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Два круга попадают в F -узлы других расположений, это отмечено стрелками. Числа в них обозначают, начальниками вершин какого случая является данная вершина. Зная окружение

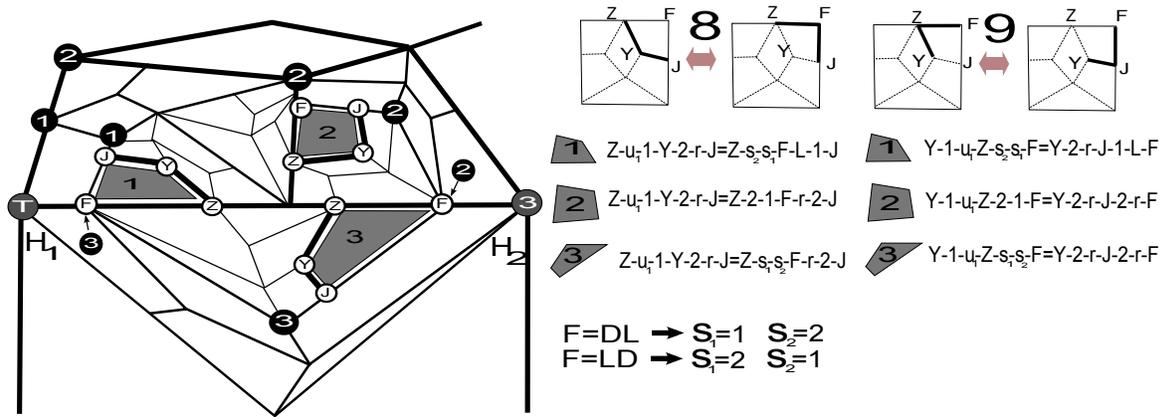


Рис. 85. Случаи расположения пути вокруг DL1-цепи и определяющие соотношения для локальных преобразований 8 и 9

центрального DL-узла, мы можем найти их типы, уровни и окружения (для вершин H_1 и H_2 – тип).

Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J , во всех случаях мы можем назвать явно, с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 85.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода. Заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляется следующим образом: случай 1 – $\text{Prev}(F)$; случай 2 – первый – $\text{BottomLeftChain.FBoss}(Z)$ и \mathbb{A} -тип второго. Для случая 3 заметим, что тип узла, отмеченного символом T может быть установлен, в зависимости от типа ребра, на котором лежат Z и F . Так как это ребро, с центром в DL-вершине, то тип его может быть либо 4 либо 7, в соответствии со структурой разбиения макроплитки на подплитки. Если ребро имеет 4 тип, то вершина T имеет тип \mathbb{A} , а первый начальник J – это вершина из $\mathbb{A}1$ цепи с указателем 2. Если же ребро имеет тип 7, то T -вершина является вторым начальником F и Z , а первый начальник J – это вершина из 2-цепи вокруг него с указателем, соответствующим 7 ребру.

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.39. Случай цепи DL1; преобразования 7 и 10. В правой части рисунка 86 изображены локальные преобразования 7 и 10.

Фиксируем некоторую вершину типа DL третьего уровня и ее начальника. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центральной вершины, мы можем вычислить коды вершин Z, Y, F, J , с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 86.

Характеризация. Составление таблиц полностью аналогично случаю C1-цепи.

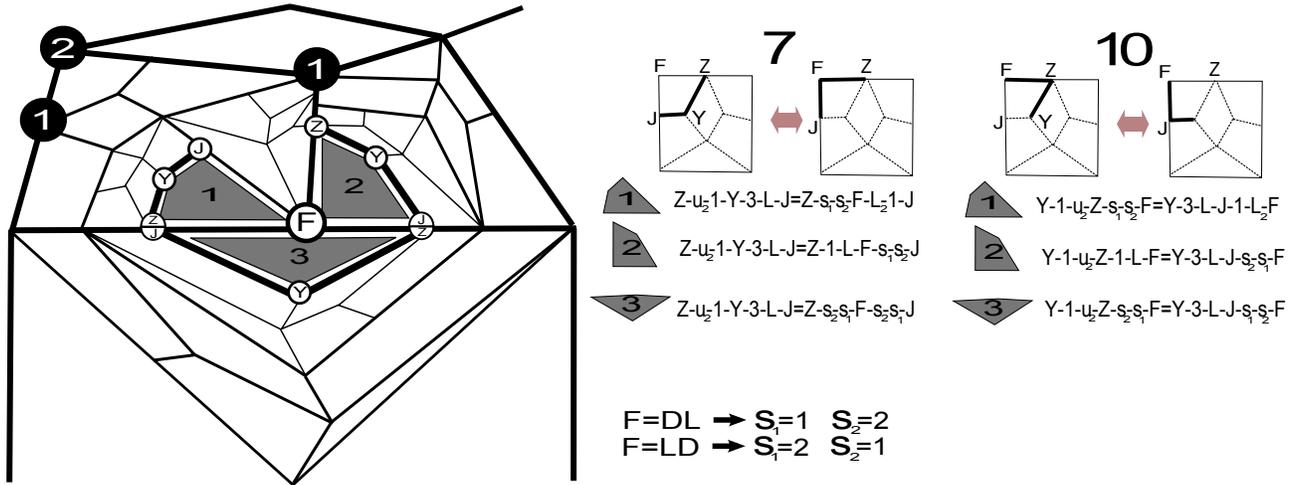


Рис. 86. Случаи расположения пути вокруг DL1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Восстановление кода. Вершина F имеет общего начальника с одной из вершин J или Z , в каждом из случаев. То есть, ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.40. **Случай цепи DL2; преобразования 8 и 9.** В правой части рисунка 87 изображены локальные преобразования 8 и 9.

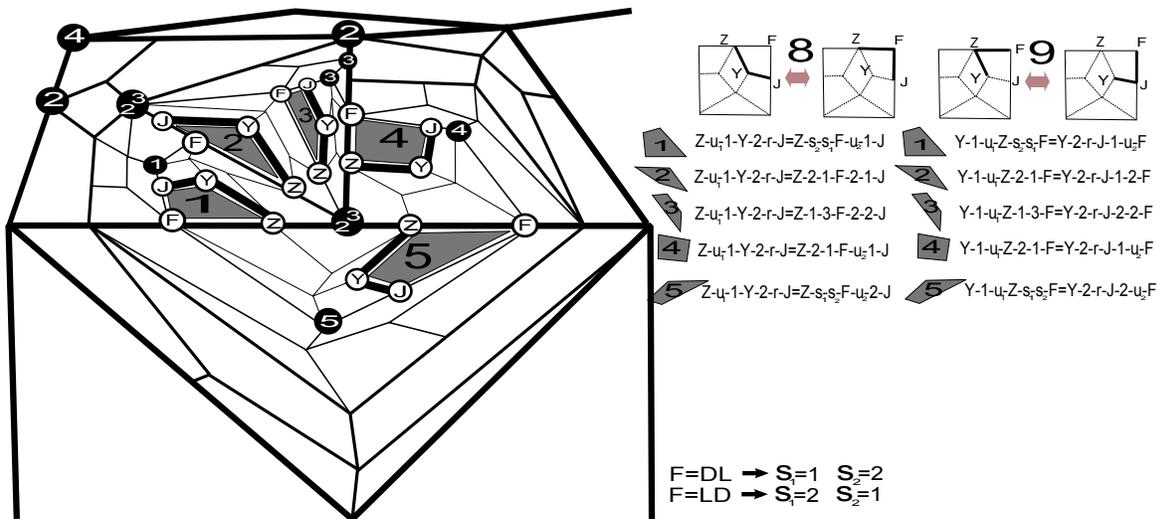


Рис. 87. Случаи расположения пути вокруг DL2-цепи и определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа \mathbb{DR} третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение центрального DL-узла, мы можем найти их типы, уровни и окружения. Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных черными кругами, то коды вершин Z, Y, F, J , во всех случаях мы можем назвать явно, с точностью до

подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 87.

Характеризация. Составление таблиц полностью аналогично случаю S_1 -цепи.

Восстановление кода. Заметим, что у Z и F в каждом из случаев общие начальники, то есть код каждой из этих вершин может быть вычислен, исходя из кодов остальных трех вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляется следующим образом: в случаях 1, 4, 5 – единственный начальник, узел F ; в случае 2 – начальники те же что и у F ; в случае 3 – первый и второй начальники как первый и третий у F .

Таким образом, зная код пути ZYJ , можно вычислить код пути ZFJ , и наоборот. А также зная код пути FZY , можно вычислить код пути FJY , и наоборот. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.41. **Случай цепи DL2; преобразования 7 и 10.** В правой части рисунка 88 изображены локальные преобразования 7 и 10.

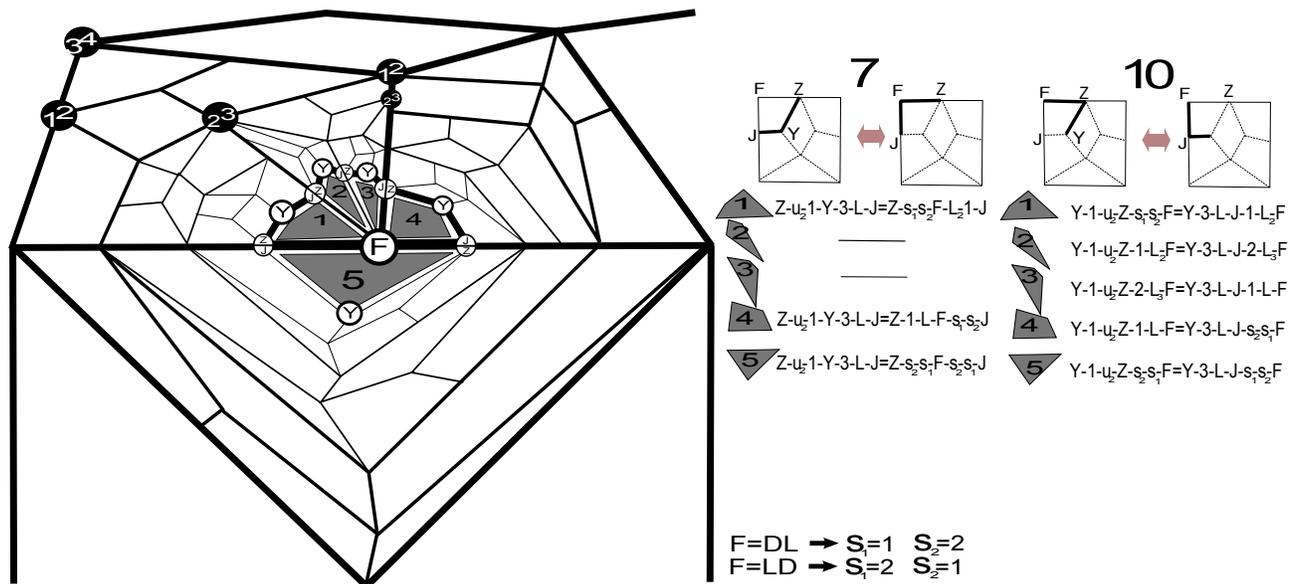


Рис. 88. Случаи расположения пути вокруг DL2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа DL третьего уровня и ее начальников. Черными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Заметим, что зная окружение этой вершины, мы можем вычислить коды вершин Z, Y, F, J , с точностью до подклееных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рисунка 88.

Для локального преобразования 7 пути в случаях 2 и 3 удовлетворяют условиям мертвого паттерна, и в этих случаях мы соотношения не вводим.

Характеризация. Составление таблиц полностью аналогично случаю S_1 -цепи.

Восстановление кода. В случаях 1, 4, 5 вершина F имеет общего начальника с одной из вершин J или Z . То есть, ZYJ , можно вычислить код пути ZFJ . В обратную сторону, а также для локального преобразования 10: очевидно, что зная F , можно вычислить коды

остальных вершин. Таким образом, мы можем осуществить локальное преобразование пути через операцию с его кодом.

8.42. **Случай цепи DL3; преобразования 7, 8, 9, 10.** Случай DL3-цепи полностью аналогичен DL2 случаю, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают DL3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для DL2 случая.

Далее рассмотрим случаи цепей вблизи края макроплиток, то есть цепи с центрами в вершинах типов $\mathbb{D}, \mathbb{U}, \mathbb{R}, \mathbb{L}$. Рассмотрение этих случаев полностью аналогично случаям обычных боковых узлов, просто используются не все из локальных преобразований. Тем не менее, мы кратко приведем вводимые определяющие соотношения.

8.43. **Случай цепи D1; преобразования 8 и 9.** В правой части рисунка 89 изображены локальные преобразования 8 и 9.

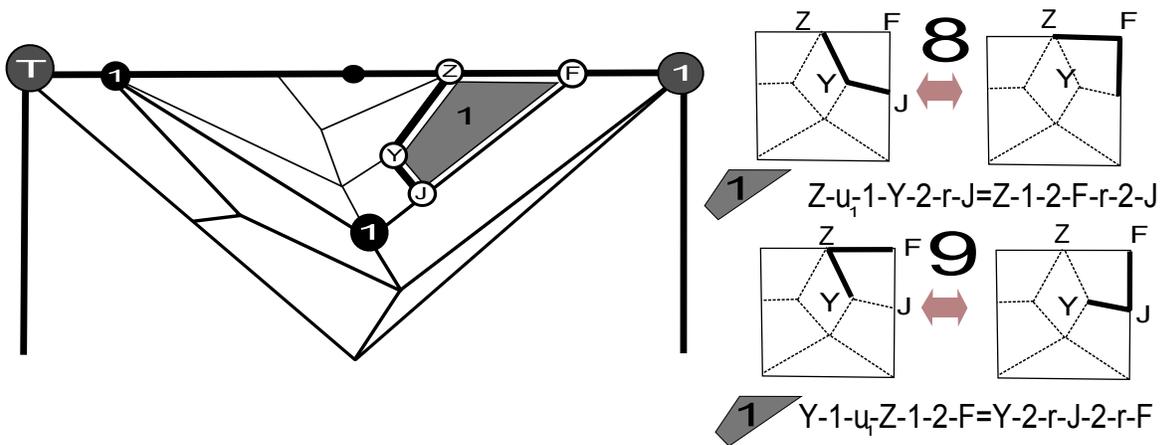


Рис. 89. Случаи расположения пути вокруг D1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю DL1 или DR1 цепи.

8.44. **Случай цепи D1; преобразования 7 и 10.** В правой части рисунка 90 изображены локальные преобразования 7 и 10.

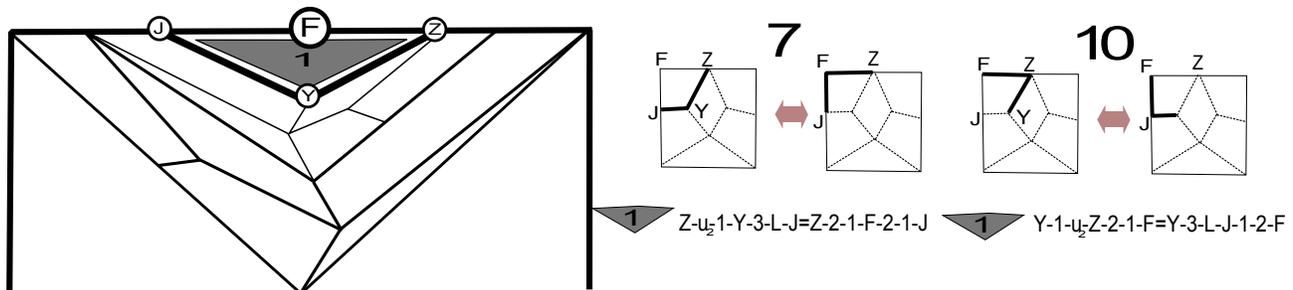


Рис. 90. Случаи расположения пути вокруг D1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю DL1 или DR1 цепи.

8.45. **Случай цепи \mathbb{D}_2 ; преобразования 8 и 9.** В правой части рисунка 91 изображены локальные преобразования 8 и 9.

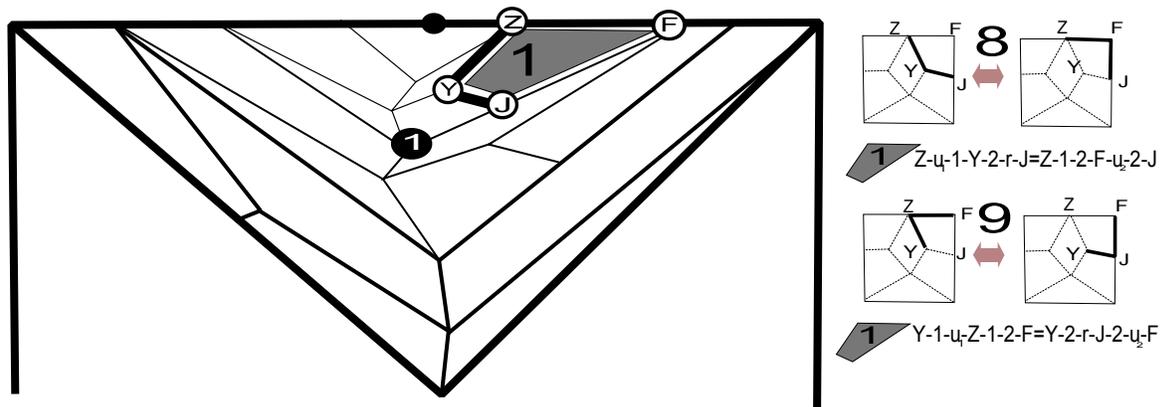


Рис. 91. Случаи расположения пути вокруг \mathbb{D}_2 -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю \mathbb{DL}_2 или \mathbb{DR}_2 цепи.

8.46. **Случай цепи \mathbb{D}_2 ; преобразования 7 и 10.** В правой части рисунка 92 изображены локальные преобразования 7 и 10.

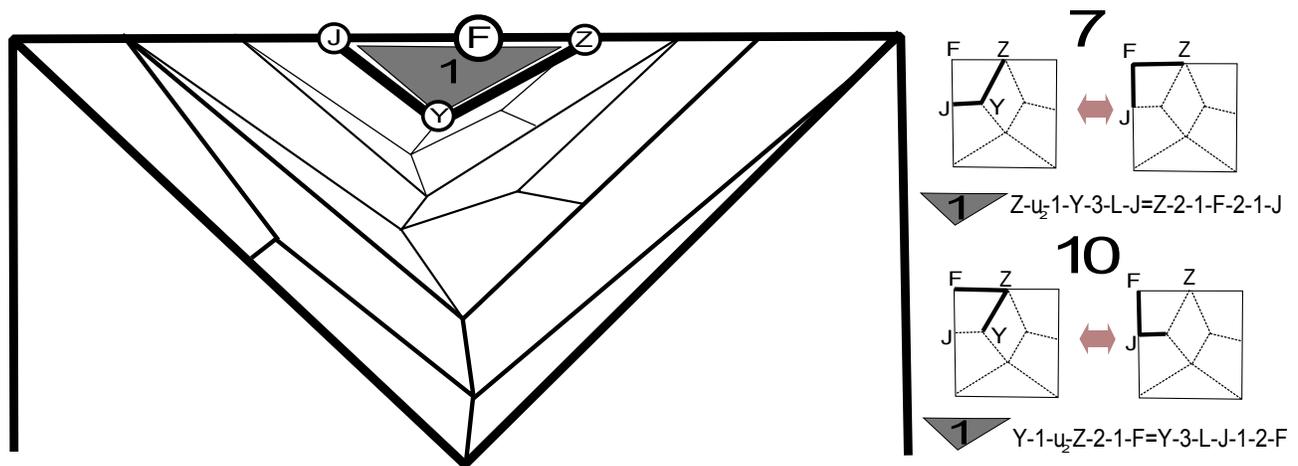


Рис. 92. Случаи расположения пути вокруг \mathbb{D}_2 -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю \mathbb{DL}_2 или \mathbb{DR}_2 цепи.

8.47. **Случай цепи \mathbb{U}_1 ; преобразования 8 и 9.** В правой части рисунка 93 изображены локальные преобразования 8 и 9.

Характеризация и восстановление кода полностью аналогично случаю \mathbb{UL}_1 или \mathbb{UR}_1 цепи.

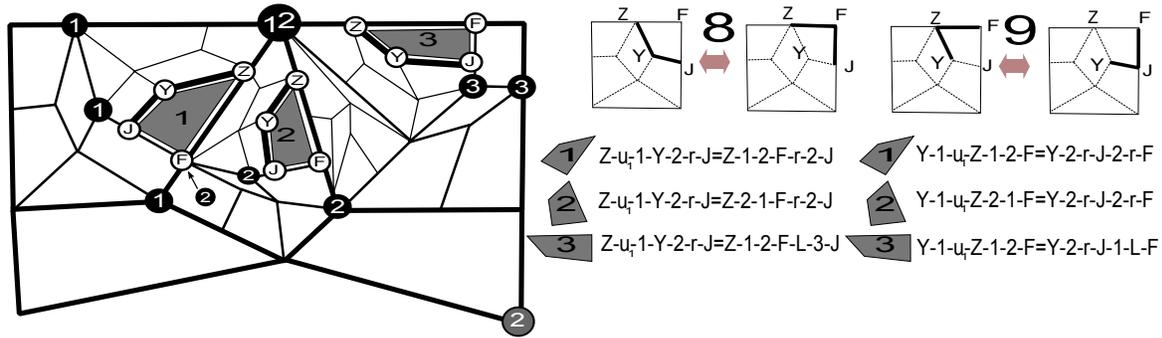


Рис. 93. Случаи расположения пути вокруг U1-цепи и соответствующие им определяющие соотношения для 8 и 9 локальных преобразований

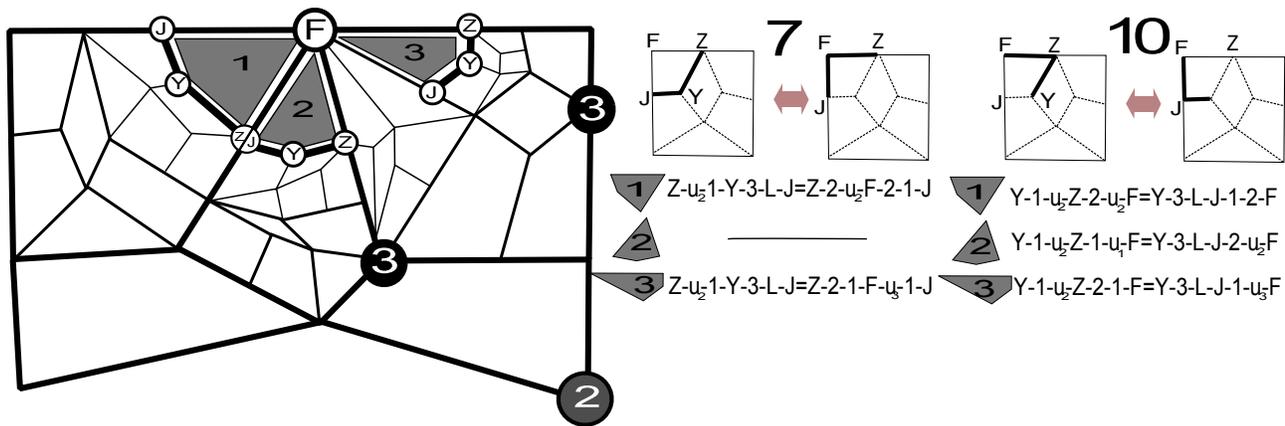


Рис. 94. Случаи расположения пути вокруг U1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

8.48. **Случай цепи U1; преобразования 7 и 10.** В правой части рисунка 94 изображены локальные преобразования 7 и 10.

Характеризация и восстановление кода полностью аналогично случаю UL1 или UR1 цепи.

8.49. **Случай цепи U2; преобразования 8 и 9.** В правой части рисунка 95 изображены локальные преобразования 8 и 9.

Характеризация и восстановление кода полностью аналогично случаю UL2 или UR2 цепи.

8.50. **Случай цепи U2; преобразования 7 и 10.** В правой части рисунка 96 изображены локальные преобразования 7 и 10.

Характеризация и восстановление кода полностью аналогично случаю UL2 или UR2 цепи.

8.51. **Случай цепи U3; преобразования 7, 8, 9, 10.** Случай U3-цепи полностью аналогичен случаю U2-цепи, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают U3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для U2 случая.

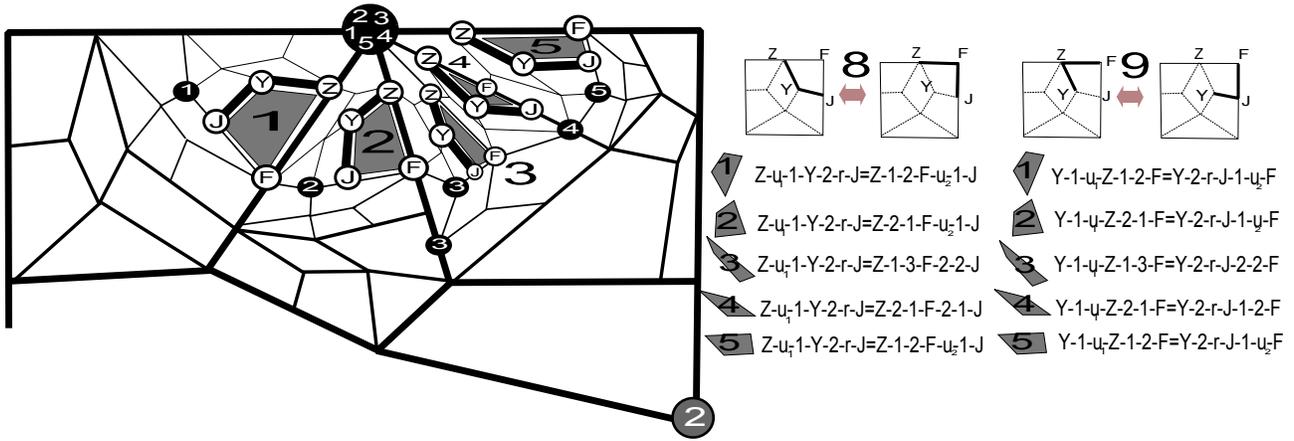


Рис. 95. Случаи расположения пути вокруг U2-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

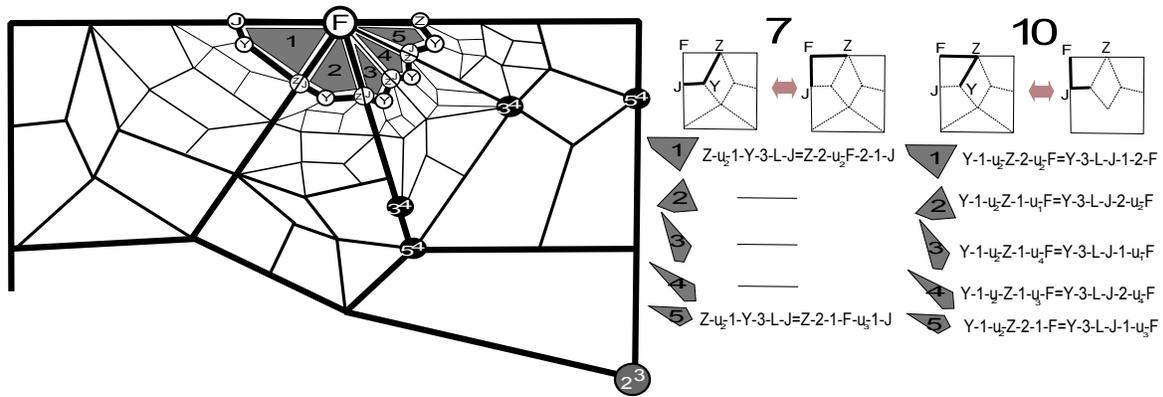


Рис. 96. Случаи расположения пути вокруг U2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

8.52. Случай цепи L1; преобразования 8 и 9. В правой части рисунка 97 изображены локальные преобразования 8 и 9.

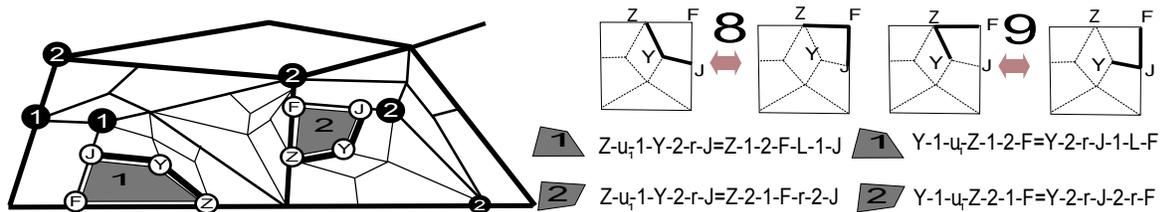


Рис. 97. Случаи расположения пути вокруг L1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю UL1 или DL1 цепи.

8.53. Случай цепи L1; преобразования 7 и 10. В правой части рисунка 98 изображены локальные преобразования 7 и 10.

Характеризация и восстановление кода полностью аналогично случаю UL1 или DL1 цепи.

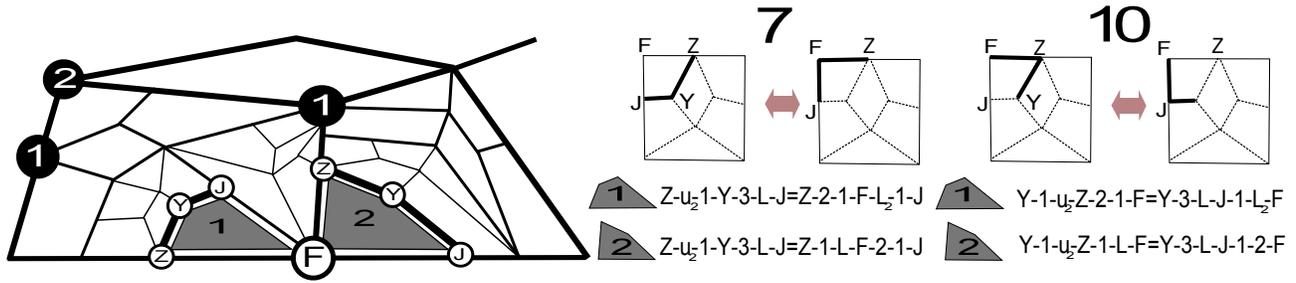


Рис. 98. Случаи расположения пути вокруг L1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

8.54. Случай цепи L2; преобразования 8 и 9. В правой части рисунка 99 изображены локальные преобразования 8 и 9.

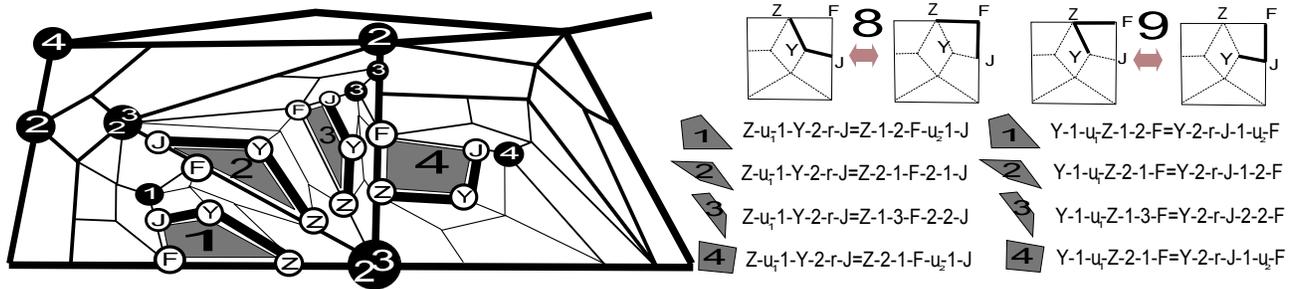


Рис. 99. Случаи расположения пути вокруг L2-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю UL2 или DL2 цепи.

8.55. Случай цепи L2, преобразования 7 и 10. В правой части рисунка 100 изображены локальные преобразования 7 и 10.

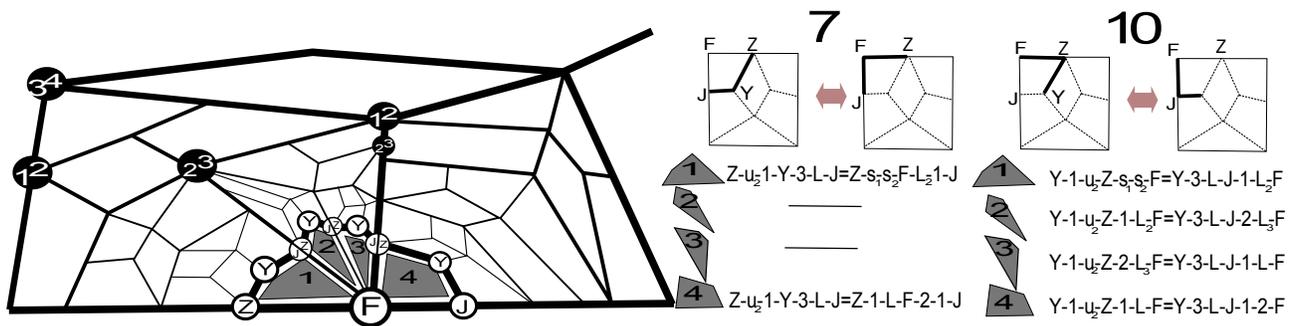


Рис. 100. Случаи расположения пути вокруг L2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю UL2 или DL2 цепи.

8.56. **Случай цепи \mathbb{L}_3 ; преобразования 7, 8, 9, 10.** Случай \mathbb{L}_3 -цепи полностью аналогичен случаю \mathbb{L}_2 -цепи, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают \mathbb{L}_3 -цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для \mathbb{L}_2 случая.

8.57. **Случай цепи \mathbb{R}_1 ; преобразования 8 и 9.** В правой части рисунка 101 изображены локальные преобразования 8 и 9.

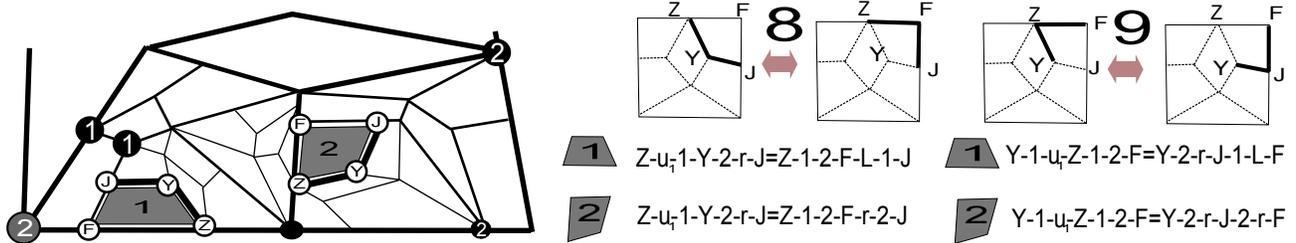


Рис. 101. Случаи расположения пути вокруг \mathbb{R}_1 -цепи и определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю \mathbb{DR}_1 или \mathbb{UR}_1 цепи.

8.58. **Случай цепи \mathbb{R}_1 ; преобразования 7 и 10.** В правой части рисунка 102 изображены локальные преобразования 7 и 10.

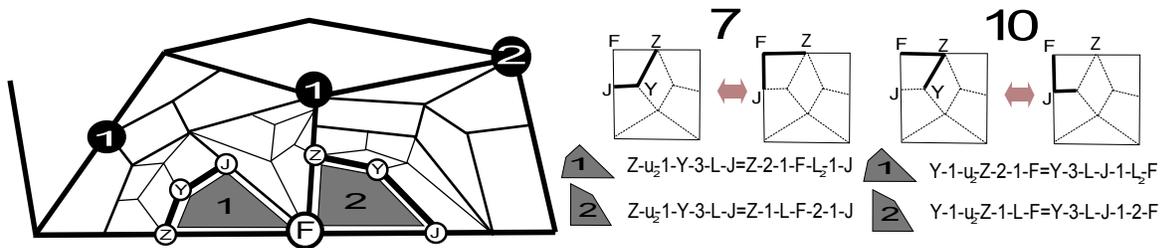


Рис. 102. Случаи расположения пути вокруг \mathbb{R}_1 -цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю \mathbb{DR}_1 или \mathbb{UR}_1 цепи.

8.59. **Случай цепи \mathbb{R}_2 ; преобразования 8 и 9.** В правой части рисунка 103 изображены локальные преобразования 8 и 9.

Характеризация и восстановление кода полностью аналогично случаю \mathbb{DR}_2 или \mathbb{UR}_2 цепи.

8.60. **Случай цепи \mathbb{R}_2 ; преобразования 7 и 10.** В правой части рисунка 104 изображены локальные преобразования 7 и 10.

Характеризация и восстановление кода полностью аналогично случаю \mathbb{DR}_2 или \mathbb{UR}_2 цепи.

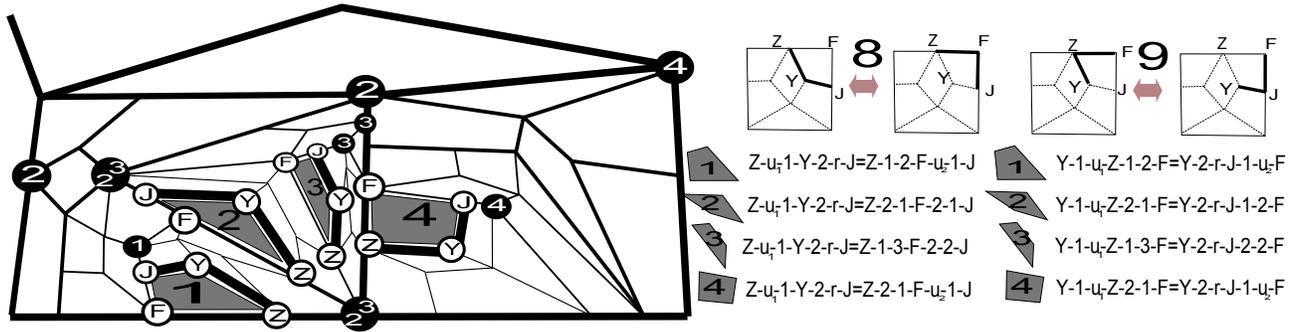


Рис. 103. Случаи расположения пути вокруг R2-цепи и определяющие соотношения для локальных преобразований 8 и 9

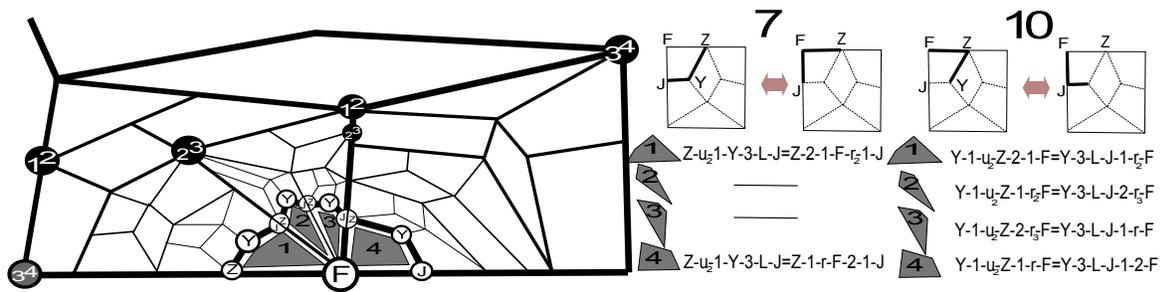


Рис. 104. Случаи расположения пути вокруг R2-цепи и определяющие соотношения для локальных преобразований 7 и 10

8.61. Случай цепи $\mathbb{R}3$; преобразования 7, 8, 9, 10. Случай $\mathbb{R}3$ -цепи полностью аналогичен случаю $\mathbb{R}2$ -цепи, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают $\mathbb{R}3$ -цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для $\mathbb{R}2$ случая.

8.62. Случай цепи $CUL0$; преобразования 7, 8, 9, 10. В правой части рисунка 105 изображены локальные преобразования 7, 8, 9 и 10.

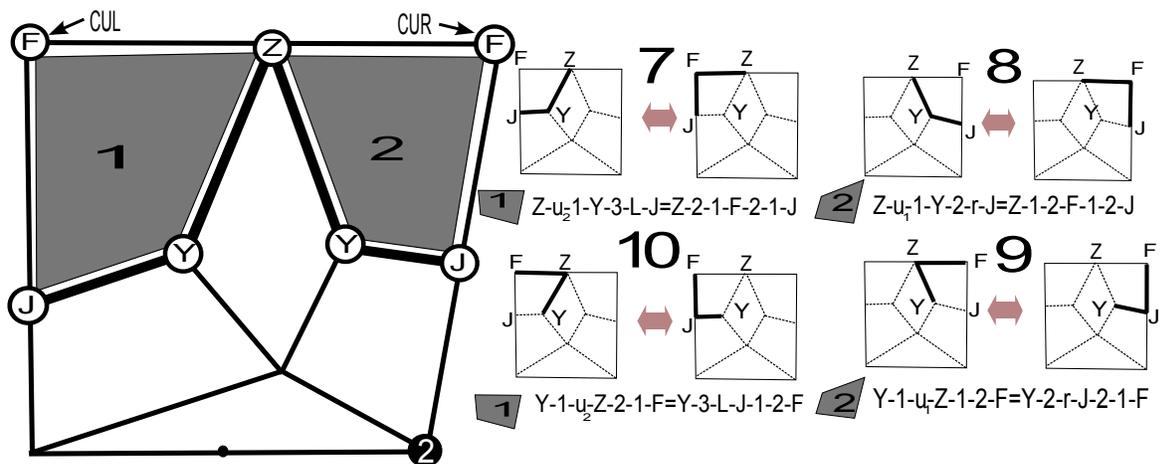


Рис. 105. Случаи расположения пути вокруг $CUL0$ -цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвертой вершины по известным трем другим также ясно во всех случаях.

8.63. **Случай цепи CUL1; преобразования 7, 8, 9, 10.** В правой части рисунка 106 изображены локальные преобразования 7, 8, 9 и 10.

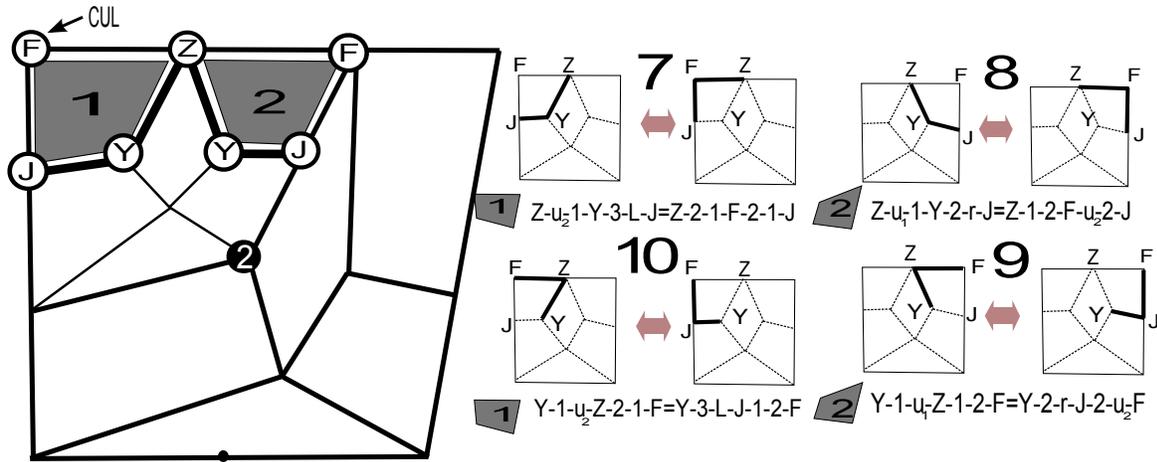


Рис. 106. Случаи расположения пути вокруг CUL1-цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвертой вершины по известным трем другим также ясно во всех случаях.

8.64. **Случай цепи CUR1; преобразования 7, 8, 9, 10.** В правой части рисунка 107 изображены локальные преобразования 7, 8, 9 и 10.

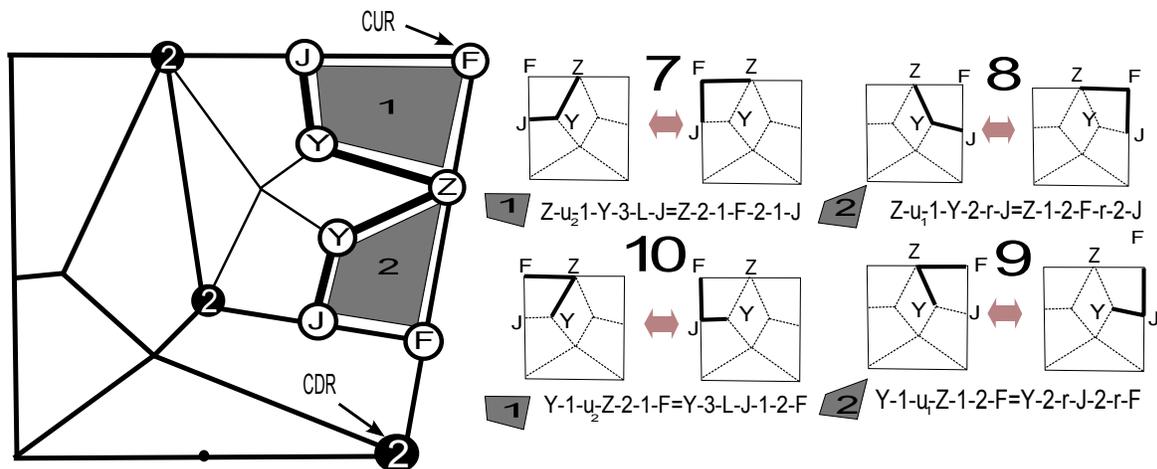


Рис. 107. Случаи расположения пути вокруг CUR1-цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвертой вершины по известным трем другим также ясно во всех случаях.

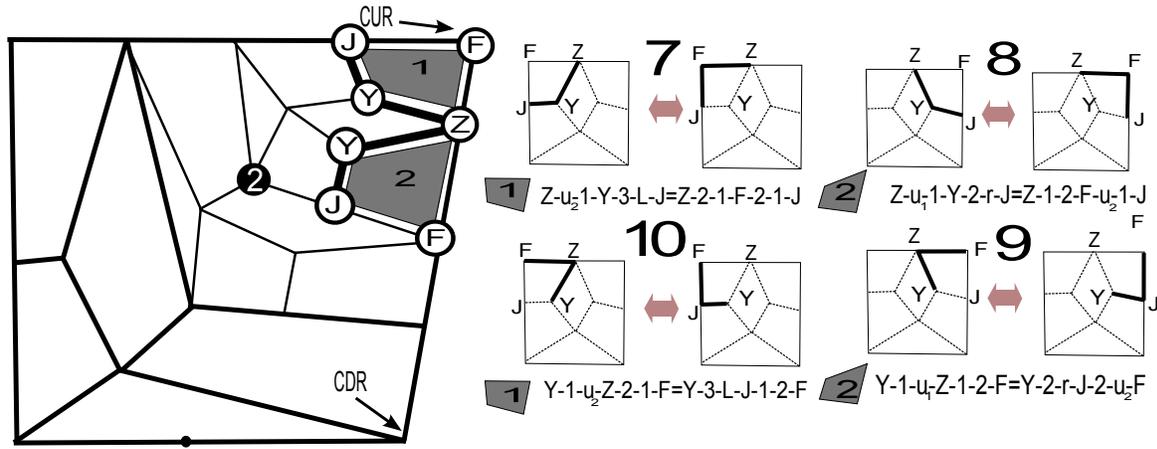


Рис. 108. Случаи расположения пути вокруг $CUR2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7, 8, 9, 10

8.65. **Случай цепи $CUR2$; преобразования 7, 8, 9, 10.** В правой части рисунка 108 изображены локальные преобразования 8 и 9.

Характеризация очевидна. Восстановление кода четвертой вершины по известным трем другим также ясно во всех случаях.

8.66. **Случай цепи $CDL1$; преобразования 7, 8, 9, 10.** В правой части рисунка 109 изображены локальные преобразования 7, 8, 9 и 10.

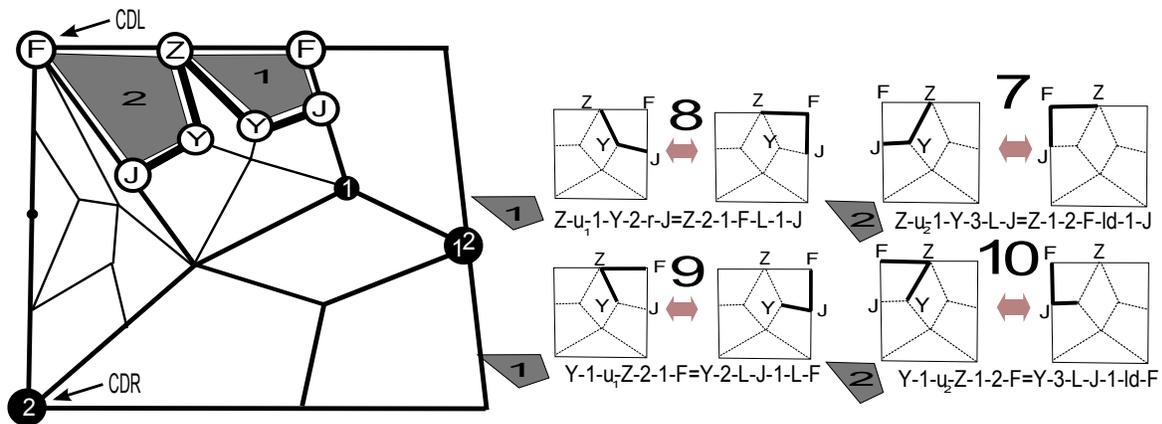


Рис. 109. Случаи расположения пути вокруг $CDL1$ -цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода ясно, так как у F и J – общее множество начальников, у Y первый начальник – Z , и тип второго во втором случае – \mathbb{A} , а Z – краевая вершина без начальников.

8.67. **Случай цепи $CDL2$; преобразования 8 и 9.** В правой части рисунка 110 изображены локальные преобразования 8 и 9.

Характеризация очевидна. Восстановление кода также ясно, учитывая, что у F и Z всегда общее множество начальников, у Y первый начальник – Z , а тип второго ясен из

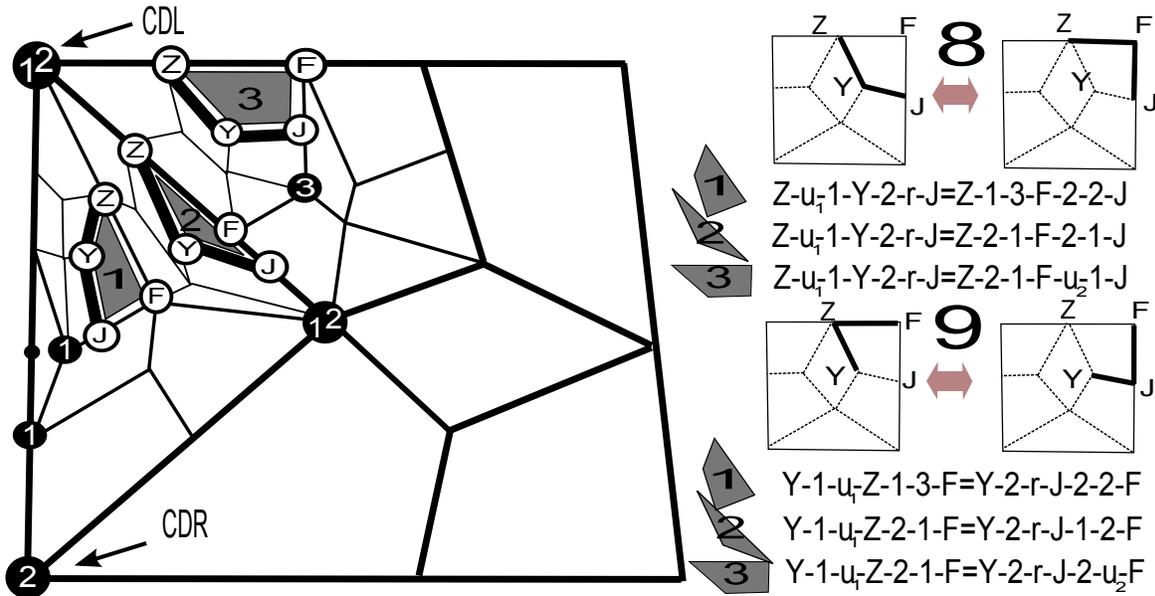


Рис. 110. Случаи расположения пути вокруг CDL2-цепи и определяющие соотношения для локальных преобразований 8 и 9

расположения. Начальники J во всех случаях либо содержатся среди начальников F , либо сам F является начальником.

8.68. Случай цепи CDL2; преобразования 7 и 10. В правой части рисунка 111 изображены локальные преобразования 7 и 10.

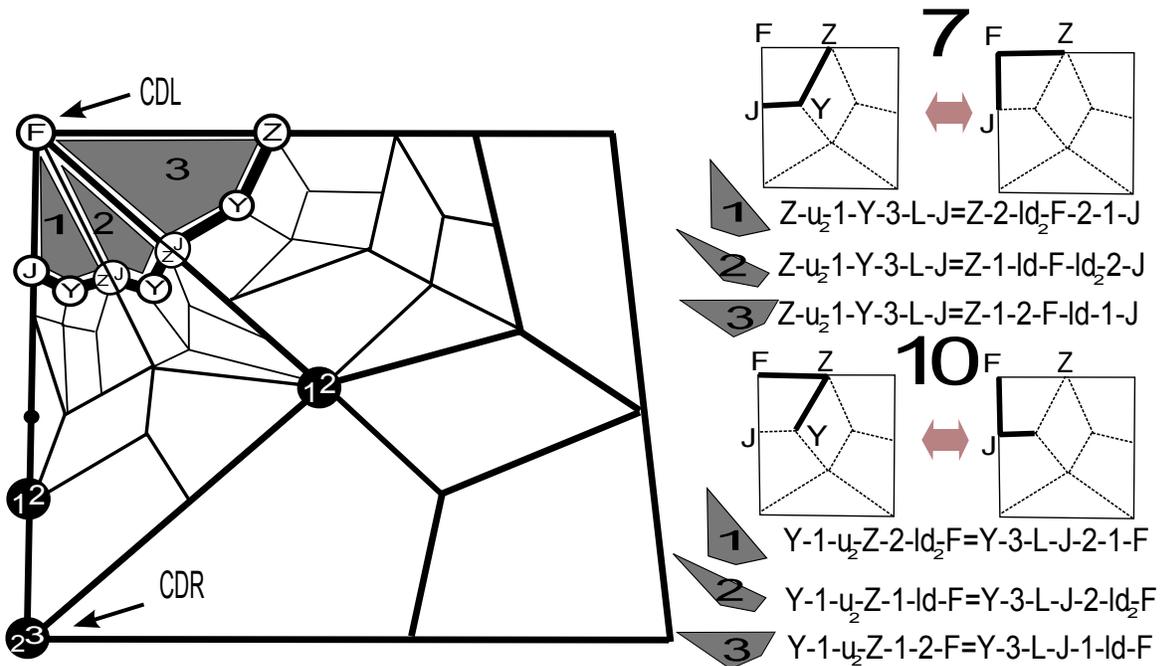


Рис. 111. Случаи расположения пути вокруг CDL2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация очевидна. Для доказательства восстановления кода достаточно заметить, что окружения четырех вершин в каждом случае очевидны, а окружение начальников (вершины, отмеченных черными кругами), можно явным образом написать.

8.69. **Случай цепи CDL3; преобразования 7 и 10.** Случай CDL3-цепи полностью аналогичен случаю CDL2-цепи, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают CDL3-цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для CDL2 случая.

8.70. **Случай цепи CDR1; преобразования 8 и 9.** В правой части рисунка 112 изображены локальные преобразования 8 и 9.

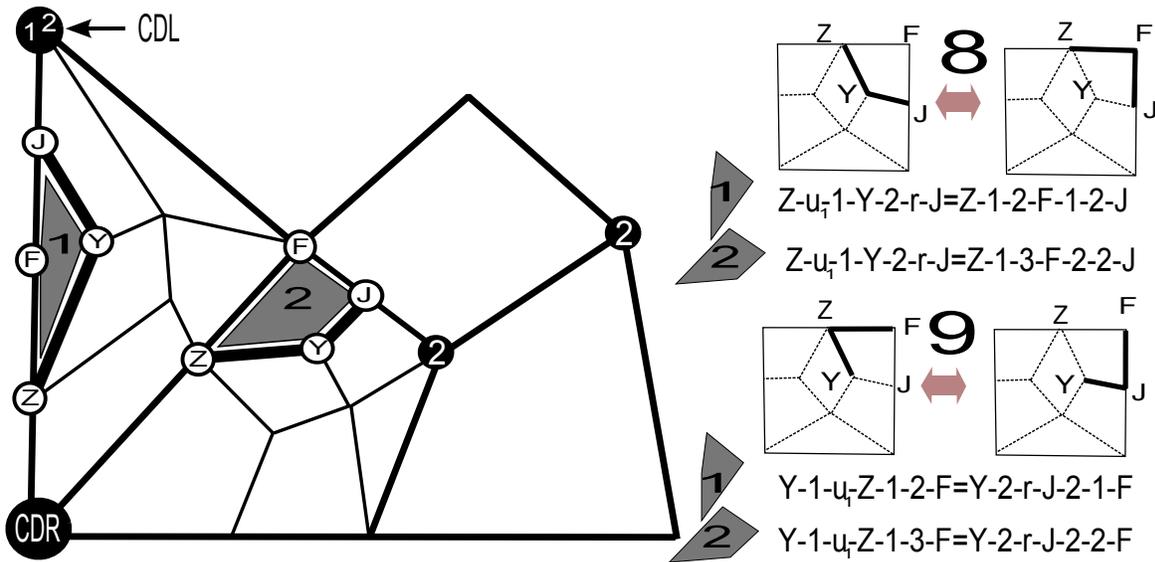


Рис. 112. Случаи расположения пути вокруг CDR1-цепи и определяющие соотношения для локальных преобразований 8 и 9

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырех, либо является начальником другой вершины из четырех.

8.71. **Случай цепи CDR1; преобразования 7 и 10.** В правой части рисунка 113 изображены локальные преобразования 7 и 10.

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырех, либо является начальником другой вершины из четырех.

8.72. **Случай цепи CDR2; преобразования 8 и 9.** В правой части рисунка 114 изображены локальные преобразования 8 и 9.

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырех, либо является начальником другой вершины из четырех.

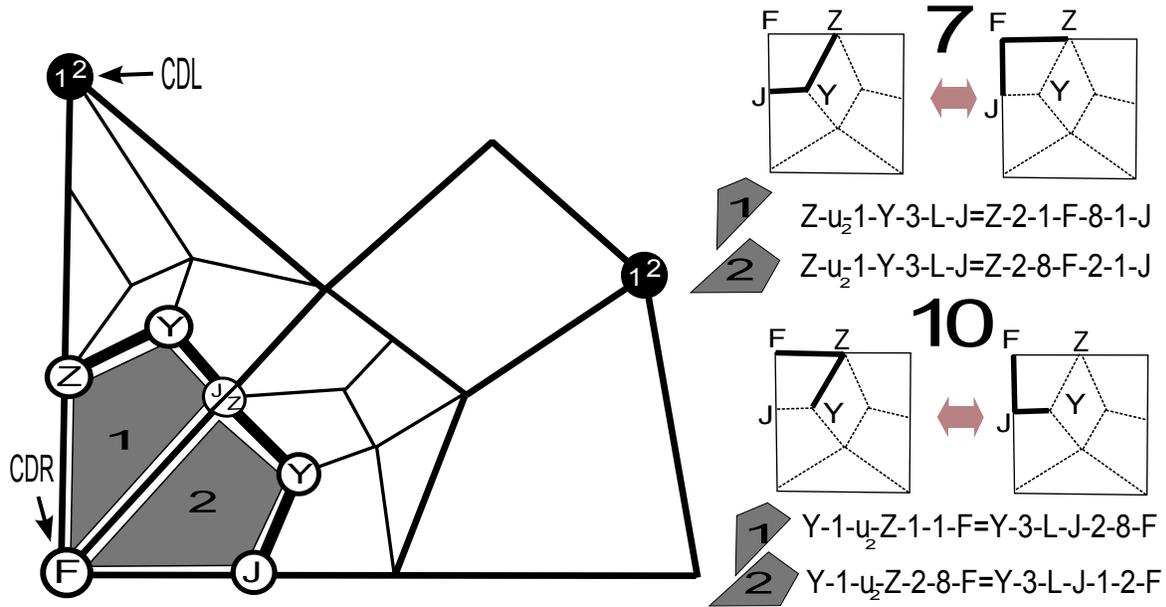


Рис. 113. Случаи расположения пути вокруг CDR1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

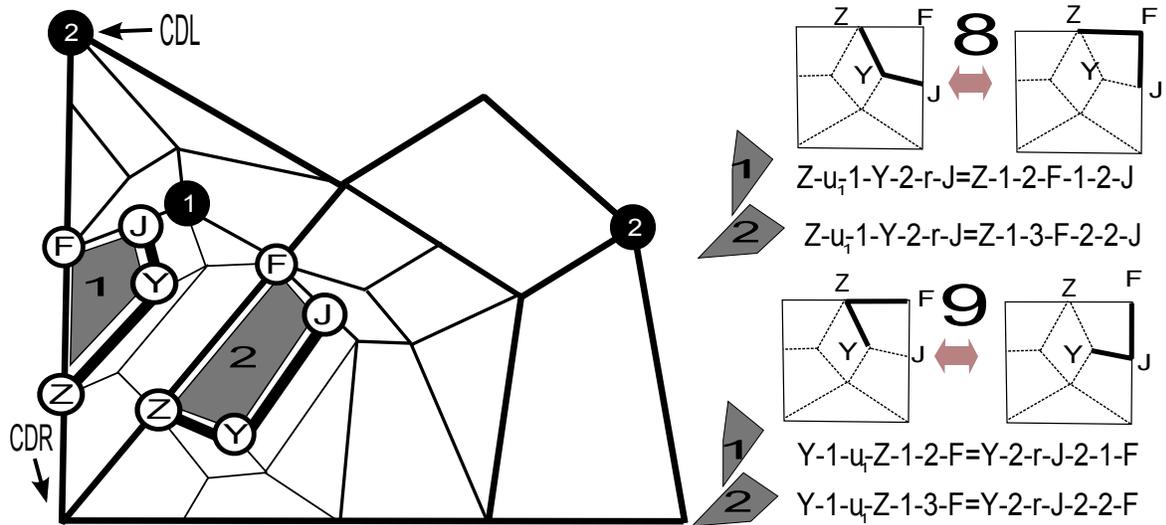


Рис. 114. Случаи расположения пути вокруг CDR2-цепи и определяющие соотношения для локальных преобразований 8 и 9

8.73. **Случай цепи CDR2; преобразования 7 и 10.** В правой части рисунка 115 изображены локальные преобразования 7 и 10.

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырех, либо является начальником другой вершины из четырех.

8.74. **Подсчет введенных соотношений.** Пусть F – число различных флагов макроплиток, P – число различных подклеенных окружений, I – число различных информаций. Пусть также $\text{Num}(\mathbb{A})$ – число базовых окружений вершин типа \mathbb{A} , аналогично для других типов. Крайние вершины в пути могут иметь произвольное подклеенное окружение, кроме того, соотношения вводятся для каждого заданного флага подклейки.

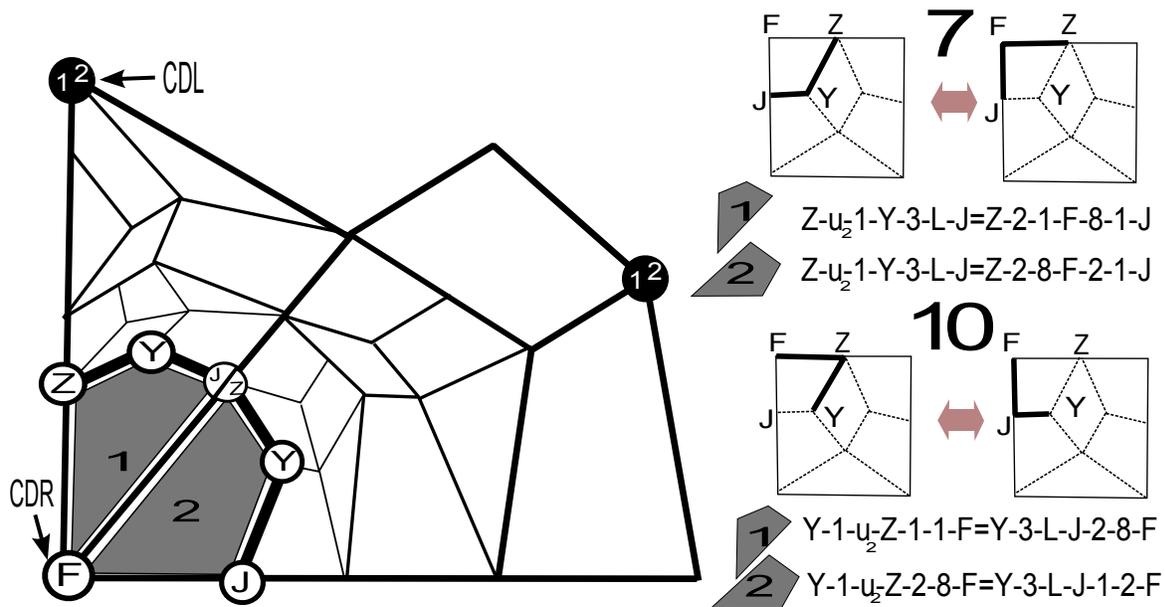


Рис. 115. Случаи расположения пути вокруг CDR2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Для каждого из локальных преобразований (1) – (6) мы фиксировали один узел с окружением, флагом макроплитки и информацией. Также еще для одного узла могла быть зафиксирована информация. После чего, в каждом из шести случаев расположения мы вводили не более 2 соотношений для каждого сочетания подклеенных окружений крайних узлов пути и флага макроплитки. То есть общее число соотношений для локальных преобразований (1) – (6) не превосходит $72 \cdot I \cdot F \cdot P \cdot \text{Num}$, где Num – число всевозможных сочетаний типа, уровня, окружения и информации плоской вершины. То есть для локальных преобразований (1) – (6) соотношений было введено не более

$$72 \cdot 181 \cdot 10^6 \cdot 5 \cdot 10^{18} \cdot 39 \cdot 16 \cdot 10^{12} < 41 \cdot 10^{42}.$$

Теперь посчитаем число введенных соотношений для локальных преобразований 7, 8, 9, 10. Напомним, что мы рассматривали вершины из цепей, фиксировали для них информацию, флаг подклейки, а также два подклеенных окружения. После этого вводилось не более двух соотношений для каждого локального преобразования. Таким образом, число введенных соотношений для локальных преобразований 7, 8, 9, 10 не более $8 \cdot F \cdot P^2 \cdot N$, где N – общее число вершин всех типов, всех базовых окружений, всех возможных информаций. Учитывая $F \leq 16 \cdot 10^{18}$ и $P = 39$, получаем, что сочетаний базовых окружений и информаций не более $2749 \cdot 181 \cdot 10^6$. Значит число введенных соотношений для 7, 8, 9, 10 локальных преобразований не превосходит $13 \cdot 10^{33}$.

9. ЛОКАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ ПРИ ВЫХОДЕ В ПОДКЛЕЕННУЮ МАКРОПЛИТКУ

В предыдущей главе мы описали, как вводятся локальные преобразования (аналог определяющих соотношений) для плоских участков путей. В этой главе мы опишем, как это делается для путей, содержащих выход в подклеенную макроплитку. То есть мы рассматриваем произвольный путь $X_1 e_1 e_2 X_2 e_3 e_4 X_3$, где X_1, X_2, X_3 – буквы, отвечающие кодам вершин, а e_1, e_2, e_3, e_4 – буквы, отвечающие ребрам входа и выхода. Хотя бы одно из ребер входа и выхода отвечает ребру, выходящему в подклейку.

Для начала докажем следующее утверждение, позволяющее узнать ребра входа и выхода в пути из трех узлов, для случая когда средний узел является ядром некоторой подклейки.

Ядро подклейки. Пусть Y – ядро некоторой подклеенной макроплитки T . Напомним, что в каждом узле T хранится параметр “*флаг подклейки*”, содержащий в себе полный код (тип, уровень, окружение и информацию) соответствующего ядра макроплитки Y , а также сочетания типов выходящих из Y ребер, которые лежат на сторонах T . например, если T подклеена по ребрам 1 и 3 к узлу типа \mathbb{A} с окружением $(7A, 7A, 3B, 4A)$, информацией Info , то параметр *флаг подклейки* для узлов T выглядит как $[\mathbb{A}, \text{Info}, (7A, 7A, 3B, 4A), 1, 3]$. Этот параметр (для узла X внутри T) обозначается как $\text{Core}(X)$. Тип выходящего из ядра ребра, соответствующего верхней стороне T обозначается как $T.\text{Core}(X)$. Соответственно, для ребра являющегося левой стороной T : $L.\text{Core}(X)$. Для окружения или типа Y мы используем напрямую обозначение $\text{Core}(X)$.

Предложение 9.1 (О ребрах около ядра макроплитки.). *Пусть XYZ – некоторый плоский путь, причем Y – узел-ядро подклеенной макроплитки, на сторонах которой лежат X и Z . Тогда ребро выхода из X (это всегда ребро 1 или ребро 2) можно установить, зная ребро входа в Y . Аналогично, ребро входа в Z устанавливается по известному ребру выхода из Y .*

Напомним, у каждого внутреннего ребра есть A -сторона и B -сторона. При этом первое главное ребро – то, для которого A -сторона лежит справа. В соответствии с этим правилом и с определением кодировок ребер входов и выходов можно выписать ребра выхода из X и входа в Z (приведены таблицы 7, 8, 9, 10 для разных типов Y):

ТАБЛИЦА 7. Ребра выхода для \mathbb{A} -узла

ребро входа в Y (или выхода из Y)	ребро выхода из X (или входа в Z)
1	1
2	2
3	2
1u	2
1d	2

ТАБЛИЦА 8. Ребра выхода для \mathbb{B} -узла

ребро входа в Y (или выхода из Y)	ребро выхода из X (или входа в Z)
1	2
2	1
3	1
ru	2
mid	2
rd	2

Таким образом, зная хотя бы один узел подклеенной макроплитки T (из тех, которые лежат внутри нее, то есть не на верхней или левой стороне), мы можем узнать ребра входа

ТАБЛИЦА 9. Ребра выхода для \mathbb{C} -узла

ребро входа в Y (или выхода из Y)	ребро выхода из X (или входа в Z)
1	1
2	2
3	1
4	2
ld ₁	1
ld ₂	2
mid ₁	1
mid ₂	2
d ₁	1
d ₂	2

ТАБЛИЦА 10. Ребра выхода для UL, UR, DR, DL узлов

ребро входа в Y (или выхода из Y)	ребро выхода из X (или входа в Z)
1	2
2	1
u ₁	1
u ₂	2
u ₃	1
u ₄	2
l	1
l ₂	1
l ₃	2
r	2
r ₂	1
r ₃	2

и выхода для путей вдоль левой или верхней стороны T . В дальнейшем, если нам потребуется выписать типы ребер вдоль левой или верхней стороны подклеенной макроплитки, мы будем использовать ссылку на предложение 9.1.

Предложение 9.2 (Об информации узла рядом с ядром макроплитки). Пусть XU – некоторый плоский путь, причем Y – узел-ядро подклеенной макроплитки, на одной из сторон которой лежит X . Будем считать, что нам известен код Y (тип, уровень, окружение и информация), а также тип входящего в Y ребра и тип, уровень и окружение X . Тогда информация у X может быть восстановлена по этим данным.

Доказательство. Допустим сначала, что X имеет один из внутренних типов (A, B, C).

Пусть тип X это A. Рассмотрим ребро входа в Y . Имеется пять возможных типов ребер для узла типа A, это ребра 1, 2, 3, lu, ld. Если ребро входа в Y это 1, 2 или 3, информация у X будет такой же, как у Y . Для ребра lu, первым начальником X будет `LevelPlus.FBoss(Y)`, вторым – `Next.FBoss(Y)`, третьим – сам узел Y . В случае ld, первым начальником X будет `BottomLeftChain(Y)`, вторым – C, с окружением Y , третьим – сам узел Y .

Пусть тип Y это \mathbb{B} . Имеется шесть возможных ребер из узла типа \mathbb{B} , это ребра $1, 2, 3, ru, rd, mid$. Если ребро входа в Y имеет тип $1, 2$ или 3 , то информация у X такая же как у Y . Для ребра rd , первым начальником будет узел в середине ребра 7 для макроплитки T , где Y выступает в роли \mathbb{B} -узла. То есть это 1-цепь вокруг правого нижнего угла T , с указателем соответствующим входу по ребру 7 . Тип правого нижнего угла T нам известен, так как это второй начальник для Y (это входит в информацию для Y). Вторым начальник X – это узел в середине правой стороны T . То есть первый начальник Y это $\text{TopFromRight.SBoss}(X)$. Третьим начальником X будет сам узел Y . Для ребра ru , первым начальником будет 1-цепь вокруг правого верхнего угла T , с указателем соответствующим входу по правому ребру. Тип правого верхнего угла T мы можем определить, применив функцию $\text{TopRightType}(Y)$. Вторым начальник X совпадает с первым начальником Y . Третьим начальником X будет сам узел Y . Для ребра mid , первым начальником будет 1-цепь вокруг \mathbb{A} -узла T (то есть окружение как у Y), с указателем 1 . Вторым начальник X это \mathbb{C} -узел T (окружение как у Y). Третьим начальником X будет сам узел Y .

Пусть тип Y это \mathbb{C} . Имеется десять возможных ребер из узла типа \mathbb{C} (4 главных и 6 неглавных). Можно заметить, что во всех случаях окружения начальников X могут быть восстановлены по информации и окружению Y , этот процесс полностью аналогичен описанному выше.

Пусть теперь X имеет боковой тип. Если входящее в Y ребро – главное, то X и Y лежат на одном ребре в некоторой макроплитке, и поэтому у них одинаковая информация. Пусть входящее в Y ребро – неглавное. Тогда тип ребра может быть одним из следующих: $u_1, u_1, u_1, u_1, r, r_2, r_3, l, l_2, l_3$. Для u_1, u_2 первым начальником X будет Y , при этом для u_1 тип второго начальника для X может быть восстановлен по информации и окружению Y . Для l , первым и единственным начальником X будет $\text{Prev}(Y)$, а для r первый начальник X может быть восстановлен по процедуре $\text{TopFromRight}(Y)$, тип второго – по процедуре $\text{BottomRightTypeFromRight}(Y)$. Для остальных ребер все начальники X восстанавливаются аналогично случаям внутренних вершин Y .

9.1. Обзор случаев. Пусть есть путь, участвующий в локальном преобразовании, лежащий в макроплитке T , причем одно из его ребер – входящее в подклееную макроплитку. Нужно показать, как происходит преобразование во всех случаях.

Заметим, что один конец ребра в подклееную область всегда лежит на верхней или левой стороне макроплитки.

Итак, нужно рассмотреть случаи локальных преобразований, где макроплитка T смыкается к границе левой или верхней стороны подклеенной макроплитки. Таким образом, нужно рассмотреть те из случаев $B1 – B20$, которые включают граничные стороны top или left , таблица 11.

Пронумеруем их как $P1 – P10$ и рассмотрим каждый отдельно.

Для каждого из этих случаев мы рассмотрим те локальные преобразования, где хотя бы один путь имеет узел на верхней или левой стороне. Это преобразования $1, 2, 4, 5, 7, 8, 9, 10$ (рисунок 116). В остальных случаях (3 и 6) путь не может содержать ребро в подклееную область, так как пути не касаются левой или верхней стороны.

Замечание. Ниже мы не будем выписывать симметричное к введенному соотношение, отвечающее проходу пути в обратном порядке. Просто будем считать, что соотношений вводится в два раза больше. Для экономии места в таблицах ниже, “начальников” будем называть “боссами”.

ТАБЛИЦА 11. Десять окружений макроплиток, содержащих граничные стороны *top* или *left*.

старое обозначение	окружение макроплитки		новое обозначение
B1.	(left, top, right, bottom)		P1
B2.	(left, top, 1A, 3A)		P2
B3.	(7A, x , 3B, 4A),	$x = \text{left, top}$	P3
B4.	(7A, x , 1A, 3A),	$x = \text{left, top}$	P4
B5.	(top, right, 6A, 2A)		P5
B6.	(top, right, 1A, 3A)		P6
B11.	(x , 3B, 6A, 2A),	$x = \text{left, top}$	P7
B12.	(x , 3B, 1A, 3A),	$x = \text{left, top}$	P8
B17.	(x , 1A, 6A, 2A),	$x = \text{left, top}$	P9
B18.	(x , 1A, 1A, 3A),	$x = \text{left, top}$	P10

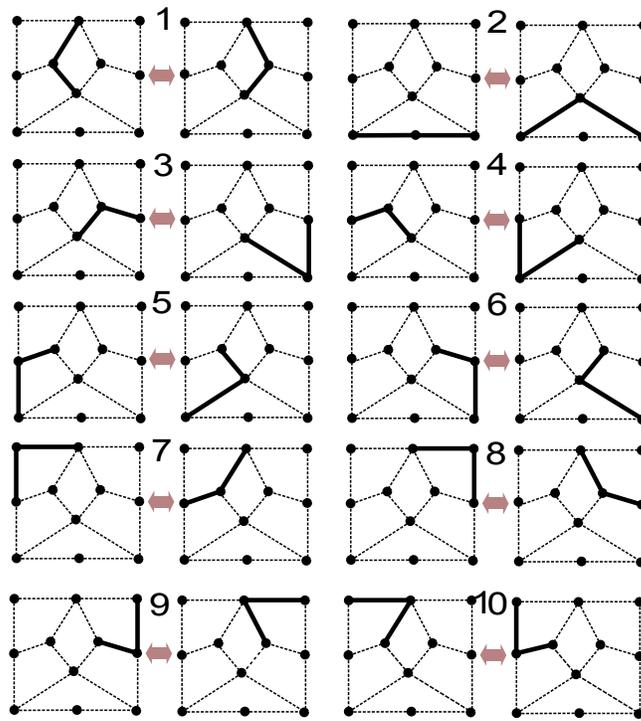


Рис. 116. Локальные преобразования

9.2. **P1: Макроплитка с окружением (left, top, right, bottom).** В этом параграфе мы рассмотрим случай **P1**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (left, top, right, bottom). Это происходит, когда макроплитка сама по себе является только что подклеенной макроплиткой.

Зафиксируем узел Q , являющийся ядром нашей подклеенной макроплитки. Его подклеенное окружение это (left, top, right, bottom). У него может быть произвольный тип, кроме углового, произвольное базовое окружение и произвольная информация. Если это боковой или краевой узел, то уровень у него должен быть третий (иначе бы к нему не была подклеена макроплитка). Сторонами подклейки могут быть любые два ребра, выходящие из нашего узла.

Напомним, что в параметр *ядро подклейки* входит тип, уровень, расширенное окружение, информация и типы двух выходящих ребер, являющихся сторонами подклейки. То есть наше ядро подклейки задано.

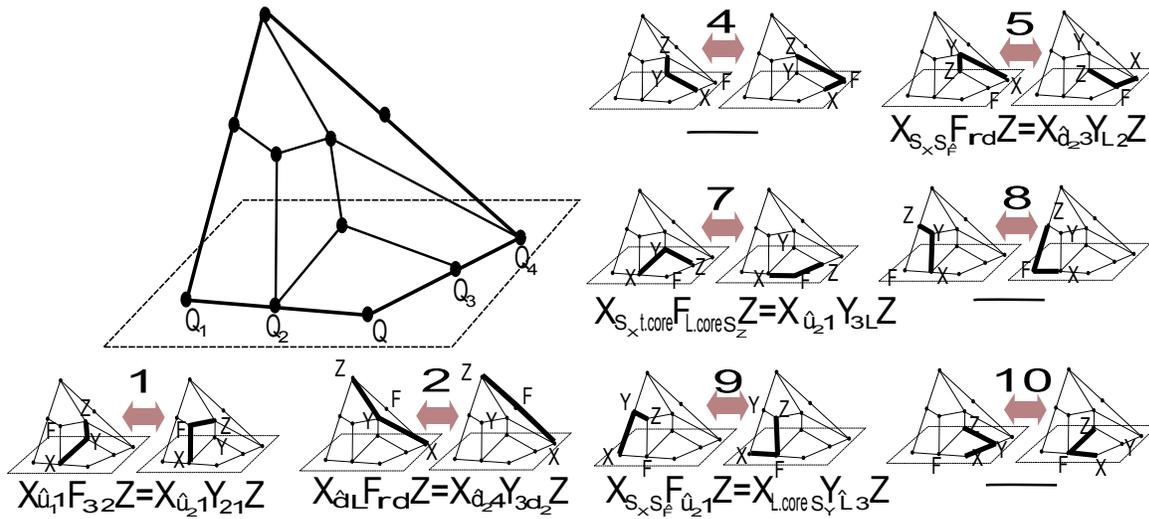


Рис. 117. Случай P1

Определив Q , можно также определить и узлы Q_1, Q_2, Q_3, Q_4 , лежащие на сторонах подклейки. То есть, мы рассматриваем всевозможные сочетания кодов для пятерки вершин, существующих на комплексе. Для каждой такой пятерки, можно определить буквы коды вершин X, Y, F, Z , для каждого из указанных на рисунке 117 восьми случаев. Также введем обозначения $e_{t.core}$ и $e_{1.core}$ для типов двух выходящих ребер, являющихся сторонами подклейки.

Характеризация. По выданному нам коду пути $P_1P_2P_3$, включающему переход по подклеенному ребру, мы можем установить, встречается ли этот путь среди перечисленных на рисунке 117. Действительно, первый узел после перехода по подклеенному ребру должен содержать параметр *ядро подклейки*, соответствующий рассматриваемому нами. Ясно также, что по конфигурации входящих-выходящих ребер и типов вершин можно выявить, какой именно случай из шестнадцати представленных на рисунке 117 имеет место.

Итак, мы можем определить по коду пути к какому именно случаю расположения он относится. Теперь покажем, как по этому коду определить код парного к нему пути.

В случаях локальных преобразований 4, 8 и 10 путь удовлетворяет условиям мертвого паттерна и не может быть частью достаточно длинного ненулевого пути, и для этих локальных преобразований мы соотношений не вводим.

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ .

Заметим, что коды X и Z мы и так знаем (из пути XYZ). Остается выписать ребра выхода из X , ребро входа в F , код F , ребро выхода из F и ребро входа в Z (таблица 12).

Примечание. Через $T.Core(Y)$ и $L.Core(Y)$ мы обозначаем типы выходящих ребер, соответствующих сторонам подклеенной макроплитки. Они содержатся в параметре флаг подклейки у вершины Y .

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ .

ТАБЛИЦА 12. Случай P1: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
1.	\widehat{u}_1	1	\mathbb{B} (окруж. как у Y)	1 босс – X	3	2
2.	\widehat{d}	1	$\mathbb{D}, 1$ (окруж. как у Y)	–	r	d
5.	Предл.9.1	Предл.9.1	Next.FBoss(Z)	как у X	1	3
7.	Предл.9.1	T.Core(Y)	Core(Y)	Core(Y)	L.Core(Y)	Предл.9.1
9.	Предл.9.1	Предл.9.1	FBoss(Z)	как у X	\widehat{u}_1	1

Заметим, что коды X и Z мы и так знаем (из пути XFZ). Остается выписать ребра выхода из X , ребро входа в Y , код Y , ребро выхода из Y и ребро входа в Z (таблица 13).

ТАБЛИЦА 13. Случай P1: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
1.	\widehat{u}_2	1	\mathbb{A} (окружение как у F)	как у F	2	1
2.	\widehat{d}_2	4	\mathbb{C} (окружение как у F)	1 босс – как у Z 2 босс - X , 3 - Z	3	d_2
5.	\widehat{d}_2	3	\mathbb{C} (окружение как у Z)	1 босс как у Z 2 босс - X , 3 – CDR	1	2
7.	\widehat{u}_2	1	\mathbb{A} , окружение – (left, right, top, bottom)	1 босс – X	3	1
9.	\widehat{r}	2	$\mathbb{R}, 1$ окружение – (left, right, top, bottom)	–	r	2

Когда все коды определены можно выписать определяющие соотношения, переводящие один путь в другой.

Обозначим результат применения процедуры, описанной в Предложении 9.1 к вершине Q , связанной с этим ребром как s_Q . Как $s_{\widehat{Q}}$ обозначим противоположное главное ребро, то есть, если $s_Q = 1$, то $s_{\widehat{Q}} = 2$ и $s_Q = 2$, то $s_{\widehat{Q}} = 1$. Обозначение $\text{Past}(X)$ используем для обозначения подклеенного окружения узла X .

Соотношения. Введем соотношения, реализующие описанные переходы.

$$1) X e_{\widehat{u}_1} e_1 F e_3 e_2 Z = X e_{\widehat{u}_2} e_1 Y e_2 e_1 Z$$

$$2) X e_{\widehat{d}} e_l F e_r e_d Z = X e_{\widehat{d}_2} e_4 Y e_3 e_{d_2} Z$$

$$5) X e_{s_X} e_{s_{\widehat{F}}} F e_l e_3 Z = X e_{\widehat{d}_2} e_3 Y e_l e_2 Z$$

$$7) X e_{s_X} e_{t.core} F e_{l.core} e_Z Z = X e_{\widehat{u}_2} e_1 Y e_3 e_l Z$$

$$9) X e_{s_X} e_{s_{\widehat{F}}} F e_{\widehat{u}_1} e_1 Z = X e_{\widehat{r}} e_u Y e_r e_2 Z$$

9.3. P2: Макроплитка с окружением (left, top, 1A, 3A). В этом параграфе мы рассмотрим случай **P2**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (left, top, 1A, 3A). Это происходит, когда макроплитка является прямым потомком подклеенной макроплитки.

Аналогично случаю **P1**, зафиксируем пять вершин Q, Q_1, Q_2, Q_3, Q_4 , занимающих на комплексе положения как на рисунке 118. То есть Q является ядром подклейки, а остальные вершины лежат на сторонах.

Для зафиксированного сочетания кодов этих пяти вершин, можно вычислить коды всех путей, указанных на рисунке 118. Кроме того, по выданному слову (коду пути) можно установить, действительно ли этот путь относится к случаю **P2** и какую конфигурацию из изображенных на рисунке 118 он имеет. Действительно по окружению узлов после перехода по подклеенному ребру можно установить, что мы имеем дело именно со случаем $P2$, а последовательность входящих-выходящих ребер помогает установить нужную конфигурацию.

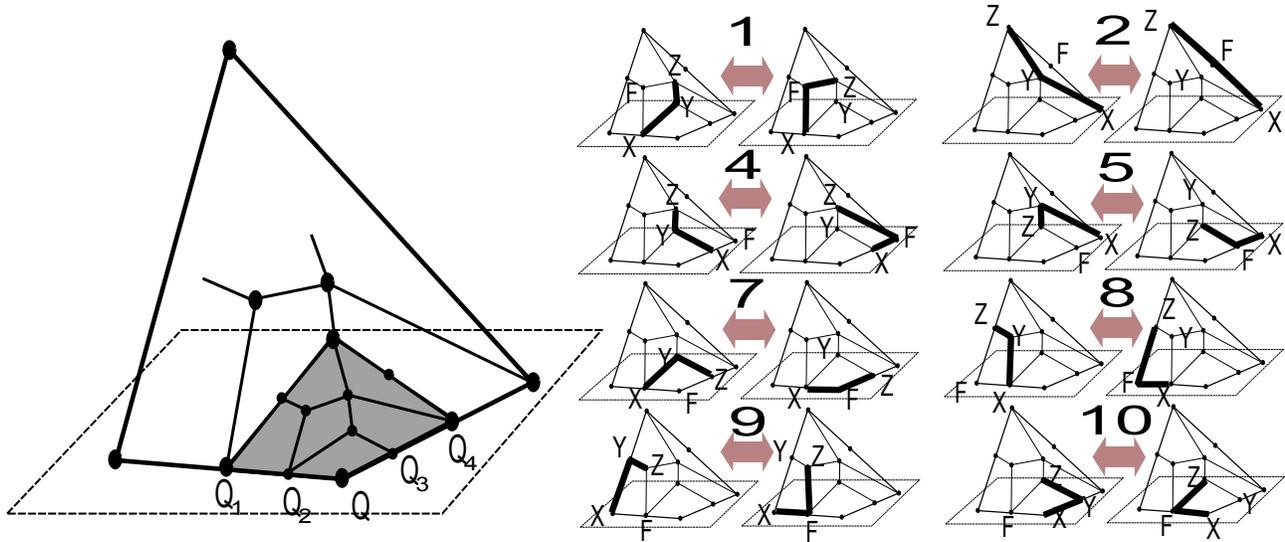


Рис. 118. Случай P2

Теперь покажем, как по этому коду определить код парного к нему пути.

Вычисление кода XFZ . Аналогично случаю **P1**, зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 14).

ТАБЛИЦА 14. Случай P2: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
1.	\hat{u}_1	1	\mathbb{B} (окружение как у Y)	1 босс – X	3	2
2.	$\hat{1}$	1	$\mathbb{DR}, 1$, окружение 3 ребро	как у Z	2	3
4.	Предл.9.1	Предл.9.1	$SBoss(Z)$	как у X	1_2	4
5.	Предл.9.1	Предл.9.1	$Next.FBoss(Z)$	как у X	1	3
7.	Предл.9.1	$T.Core(Y)$	$Core(Y)$	$Core(Y)$	$L.Core(Y)$	Предл.9.1
8.	Предл.9.1	Предл.9.1	$FBoss(Z)$	как у X	u_2	2
9.	Предл.9.1	Предл.9.1	$FBoss(Z)$	как у X	\hat{u}_1	1
10.	$Top.Core(Z)$	Предл.9.1	$FBoss(Z)$	Предл.9.2	\hat{u}_2	1

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X , F , Z , а также ребра входа в F , Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 15).

ТАБЛИЦА 15. Случай P2: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
1.	\hat{u}_2	1	\mathbb{A} (окружение как у F)	1 босс – X	2	1
2.	\hat{d}_2	4	\mathbb{C} (окружение как у F)	1 босс – как у Z 2 босс – X , 3 босс – Z	3	d_2
4.	$\hat{1}$	3	\mathbb{A} (окружение как у Z)	1 босс как у Z	2	1
5.	\hat{d}_2	3	\mathbb{C} (окружение как у Z)	1 босс как у Z 2 босс – X , 3 босс – \mathbb{A} , окружение – $\text{Past}(X)$	1	2
7.	\hat{u}_2	1	\mathbb{A} , окружение – (left, right, 1A, 3A)	1 босс – X	3	1
8.	\hat{u}_1	1	\mathbb{B} окружение – (left, right, 1A, 3A)	1 босс – X тип 2 – \mathbb{A}	2	r
9.	\hat{u}_2	2	\mathbb{RU} , окружение – 0-цепь вокруг \mathbb{A} (ук 1) с окружением $\text{Past}(X)$ уровень 1	1 босс – X	r	2
10.	$L.\text{Core}(Z)$	Предл.9.1	$\text{Next.Past.FBoss}(Z)$	Предл.9.2	$\hat{1}$	3

Примечание. Запись $\text{Next.Past.FBoss}(Z)$ означает, что берется следующий узел в цепи для первого начальника Z , причем рассматривается его подклеенное окружение (а не основное).

Соотношения. Вводятся аналогично случаю $P1$, коды вершин и ребер указаны в таблицах выше.

9.4. P3: Макроплитка с окружением (7A, x , 3B, 4A); $x = \text{left, top}$. В этом параграфе мы рассмотрим случай **P3**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (7A, x , 1A, 3A); $x = \text{left, top}$. Это происходит, когда макроплитка является левой нижней подплиткой и верхней стороной выходит на левую или верхнюю сторону некоторой подклеенной макроплитки.

Зафиксируем пять вершин Q , Q_1 , Q_2 , Q_3 , занимающих на комплексе положения как на рисунке 119. То есть Q является ядром подклейки, а остальные вершины лежат на сторонах.

Для зафиксированного сочетания кодов этих пяти вершин, можно вычислить коды всех путей, указанных на рисунке 119. Кроме того, по выданному слову (коду пути) можно установить, действительно ли этот путь относится к случаю $P2$ и какую конфигурацию из изображенных на рисунке 119 он имеет.

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X , Y , Z , а также ребра входа в Y , Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ .

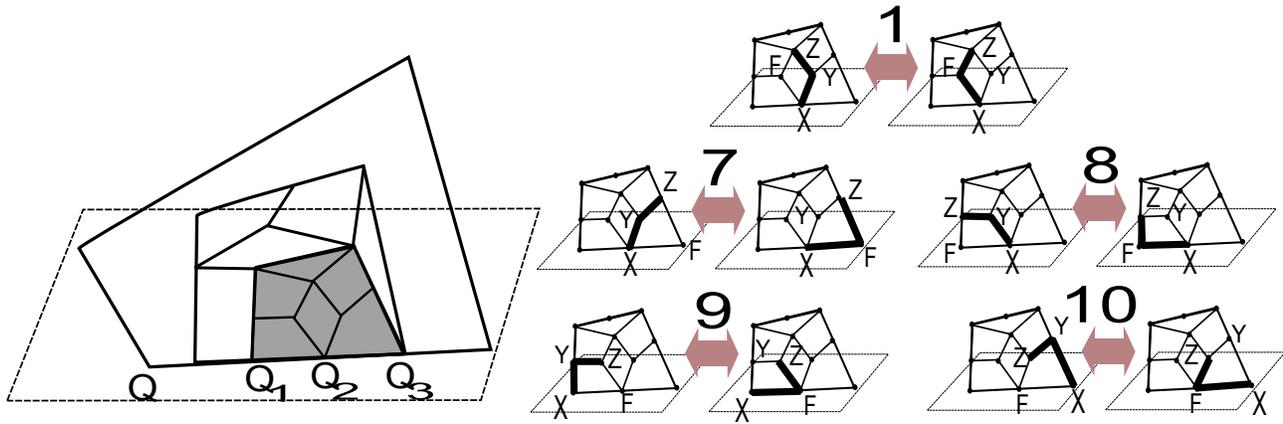


Рис. 119. Случай P3

Обозначение $\text{Past}(X)$ используем для обозначения подклеенного окружения узла X (таблица 16).

ТАБЛИЦА 16. Случай P3: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
1.	\hat{u}_1	1	\mathbb{B} (окоуж как у Y)	1 босс – X тип 2 – \mathbb{A}	3	2
7.	Предл.9.1	Предл.9.1	$\text{SBoss}(Z)$	как у X	\hat{u}_3 если $\text{Past}(F) = \mathbb{U}$ $\hat{1}_2$ если $\text{Past}(F) = \mathbb{L}$	1
8.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	1	1
9.	Предл.9.1	Предл.9.1	$\text{FBoss}(Z)$	как у X	\hat{u}_1	1
10.	Предл.9.1	Предл.9.1	$\text{FBoss}(Z)$	как у X	\hat{u}_2	1

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 17).

Примечание. Ребро из X в случае 10 имеет тип $\hat{1}_2$ если $\text{Past}(X) = \mathbb{L}$ и \hat{u}_1 если $\text{Past}(X) = \mathbb{U}$.

Запись $\text{Next.Past.FBoss}(Z)$ означает, что берется следующий узел в цепи для первого начальника Z , причем рассматривается его подклеенное окружение (а не основное).

9.5. P4: Макроплитка с окружением (7A, x, 1A, 3A); $x = \text{left, top}$. В этом параграфе мы рассмотрим случай **P4**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (7A, x, 1A, 3A); $x = \text{left, top}$. Это происходит, когда макроплитка является прямым потомком некоторой макроплитки, являющейся левой нижней подплиткой. Верхней стороной T выходит на левую или верхнюю сторону некоторой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

ТАБЛИЦА 17. Случай P3: вычисление кода XYZ.

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
1.	\hat{u}_2	1	\mathbb{A} (окоуж. как у F)	1 босс – X	2	1
7.	\hat{u}_2	1	\mathbb{A} , окружение – (7A, x, 3B, 4A), x = left, top	1 босс – X	3	1
8.	\hat{u}_1	1	\mathbb{B} , окружение – (7A, x, 3B, 4A), x = left, top	1 босс – X	2	r
9.	$\hat{1}$	2	\mathbb{DR} , 1 окружение – 3	1 босс – Y	r	2
10.	$\hat{1}_2$ или \hat{u}_1	1	Next.Past.FBoss(Z)	1 босс – TopFromCorner.Past(X) 2 босс – X, 3 босс – RightCorner.Past(X)	1	3

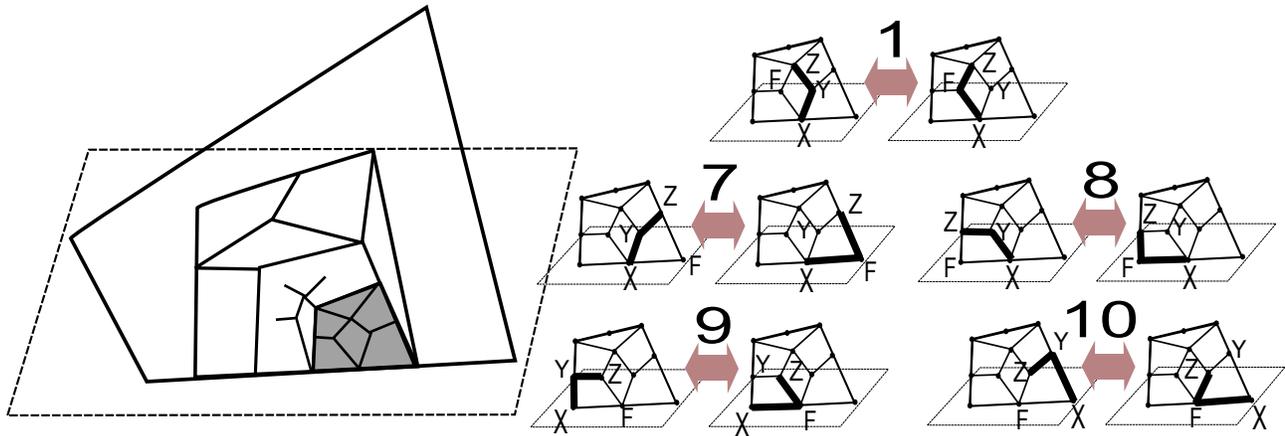


Рис. 120. Случай P4

Вычисление кода XFZ. Зная код пути XYZ (то есть коды X, Y, Z, а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 18).

ТАБЛИЦА 18. Случай P4: вычисление кода XFZ.

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
1.	\hat{u}_1	1	\mathbb{B} (окоуж как у Y)	1 босс – X тип 2 – \mathbb{A}	3	2
7.	Предл.9.1	Предл.9.1	SBoss(Z)	как у X	\hat{u}_3 если Past(F) = \mathbb{U} $\hat{1}_2$ если Past(F) = \mathbb{L}	1
8.	Предл.9.1	Предл.9.1	FBoss(Z)	как у X	\hat{u}_2	2
9.	Предл.9.1	Предл.9.1	FBoss(Z)	как у X	\hat{u}_1	1
10.	Предл.9.1	Предл.9.1	FBoss(Z)	как у X	\hat{u}_2	1

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 19).

ТАБЛИЦА 19. Случай P4: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
1.	\hat{u}_2	1	\mathbb{A} , (окружение как у F)	1 босс – X	2	1
7.	\hat{u}_2	1	\mathbb{A} , окружение – (7A, $x, 1A, 3A$), $x = \text{left, top}$	1 босс – X	3	1
8.	\hat{u}_1	1	\mathbb{B} , окружение – (7A, $x, 1A, 3A$), $x = \text{left, top}$	1 босс – X	2	r
9.	\hat{u}_2	2	$\mathbb{RU}, 1$, окружение – $\mathbb{A}0$ -цепь (указ 1) с окруж $\text{Past}(X)$	1 босс – X	r	2
10.	$\hat{1}_2$ или \hat{u}_1	1	$\text{Next.Past.FBoss}(Z)$	1 босс – $\text{TopFromCorner.Past}(X)$ 2 босс – X , 3 босс – $\text{RightCorner.Past}(X)$	1	3

Примечание. Ребро из X в случае 10 имеет тип $\hat{1}_2$ если $\text{Past}(X) = \mathbb{L}$, и \hat{u}_1 если $\text{Past}(X) = \mathbb{U}$.

Запись $\text{Next.Past.FBoss}(Z)$ означает, что берется следующий узел в цепи для первого начальника Z , причем рассматривается его подклееное окружение (а не основное).

9.6. P5: Макроплитка с окружением (top, right, 6A, 2A). В этом параграфе мы рассмотрим случай **P5**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (top, right, 6A, 2A). Это происходит, когда макроплитка является правой верхней подплиткой некоторой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 20).

Для преобразования 7 пути удовлетворяют условиям мертвого паттерна. Для них мы соотношения не вводим.

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 21).

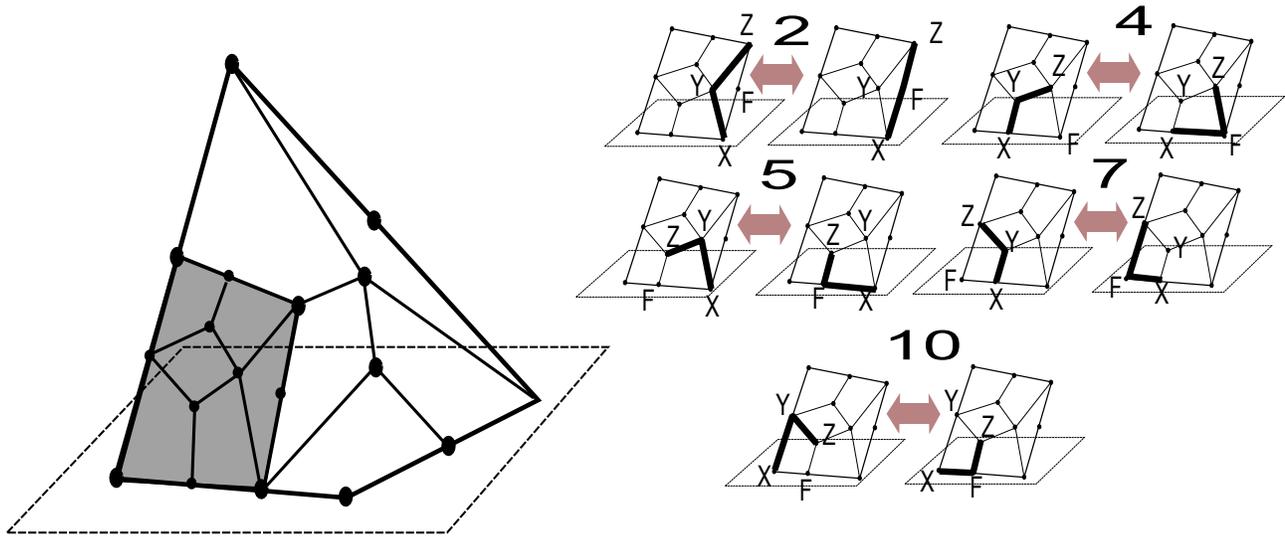


Рис. 121. Случай P5

ТАБЛИЦА 20. Случай P5: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
2.	\hat{u}_1	1	$\mathbb{DR}, 1,$ окружение – 3	1 босс – X тип 2 – \mathbb{CDR}	1	2
4.	Предл.9.1	Предл.9.1	$\mathbb{SBoss}(Z)$	как у X	\hat{u}_3	4
5.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3
10.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3

ТАБЛИЦА 21. Случай P5: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
2.	\hat{u}_3	4	\mathbb{C} , окружение как у T	1 босс – \mathbb{U} , окружение как у T , (top, right, 6A, 2A), 2 босс – X , 3 босс – \mathbb{B} , окружение (left, top, right, bottom)	2	1
4.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс как у Z	2	1
5.	\hat{u}_3	4	\mathbb{C} , окружение как у T	1 босс – как у Z 2 босс – X , 3 босс – \mathbb{B} , окружение (left, top, right, bottom)	1	2
10.	\hat{r}	2	$\mathbb{FBoss}(Z)$	–	u_2	1

9.7. **P6: Макроплитка с окружением (top, right, 1A, 3A).** В этом параграфе мы рассмотрим случай **P6**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (top, right, 1A, 3A). Это происходит, когда макроплитка является прямым потомком некоторой макроплитки, являющейся правой верхней подплиткой некоторой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

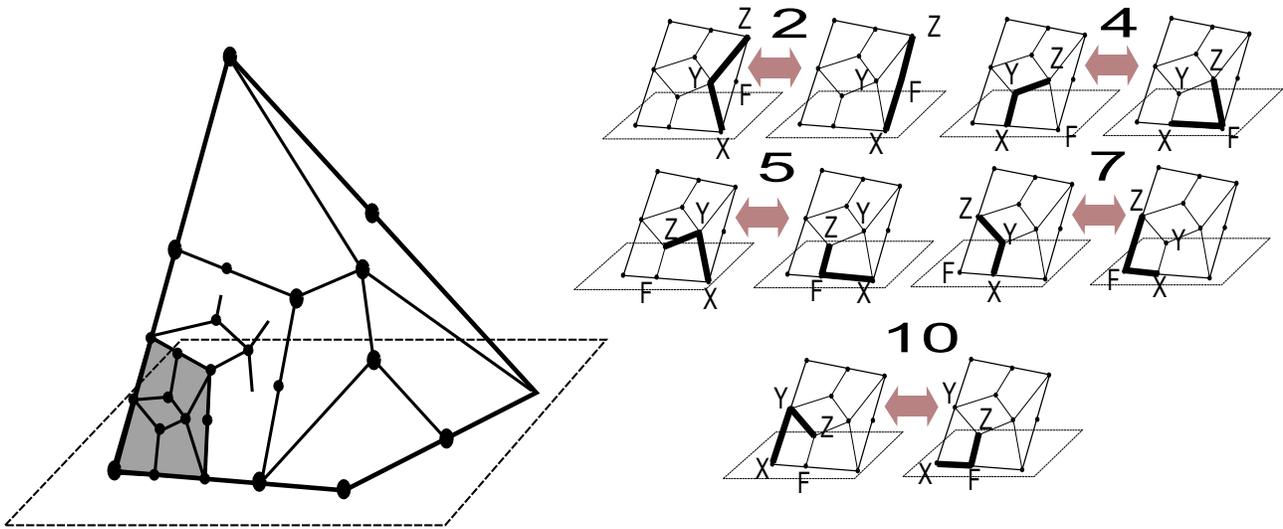


Рис. 122. Случай P6

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 22).

Для преобразования 7 пути удовлетворяют условиям мертвого паттерна. Для них мы соотношения не вводим.

ТАБЛИЦА 22. Случай P6: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
2.	$\hat{1}$	1	$\mathbb{DR}, 1,$ окружение -3	1 босс как у Z	2	3
4.	Предл.9.1	Предл.9.1	$SBoss(Z)$	как у X	$\hat{1}_3$	4
5.	Предл.9.1	Предл.9.1	$Next.FBoss(Z)$	как у X	$\hat{1}$	3
10.	Предл.9.1	Предл.9.1	$Next.FBoss(Z)$	как у X	$\hat{1}$	3

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 23).

9.8. P7: Макроплитка с окружением $(x, 3B, 6A, 2A)$; $x = left, top$. В этом параграфе мы рассмотрим случай **P7**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение $(x, 3B, 6A, 2A)$; $x = left, top$. Это происходит, когда макроплитка является правой верхней подплиткой некоторой макроплитки, являющейся левой нижней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

ТАБЛИЦА 23. Случай P6: вычисление кода XYZ.

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
2.	$\hat{1}_2$	4	\mathbb{C} , окружение – как у T	1 босс – \mathbb{U} , окружение как у T 2 босс – X, 3 босс – \mathbb{A} , окружение как у $\text{FBoss}(F)$	2	1
4.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс как у Z	2	1
5.	\hat{u}_3	4	\mathbb{C} , окружение как у T	1 босс – как у Z 2 босс – X, 3 босс – \mathbb{A} , окружение $\text{RightCorner.Past}(X)$	1	2
10.	\hat{r}	2	$\text{FBoss}(Z)$	1 босс – X	u_2	1

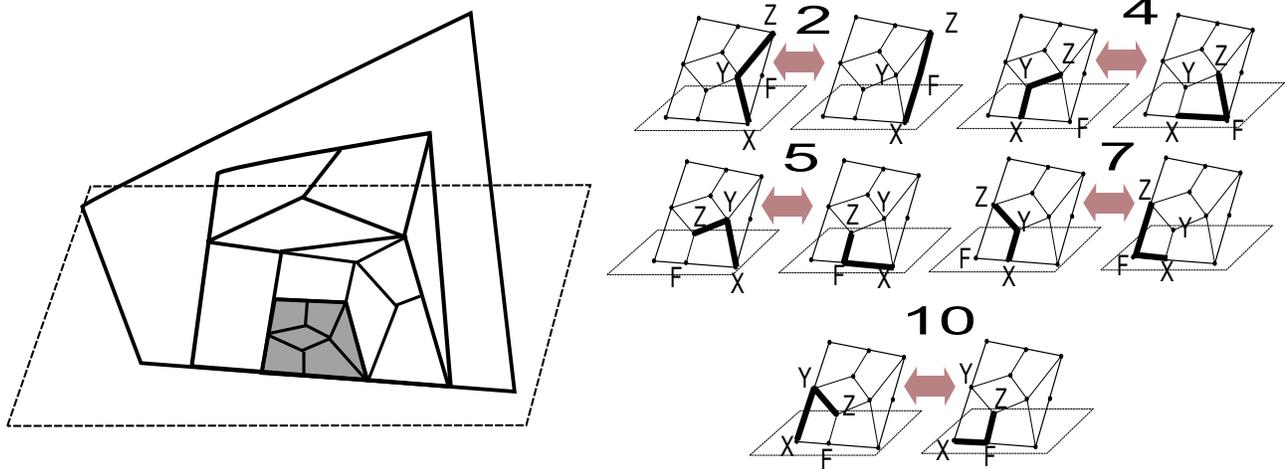


Рис. 123. Случай P7

Вычисление кода XFZ. Зная код пути XYZ (то есть коды X, Y, Z, а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 24).

ТАБЛИЦА 24. Случай P7: вычисление кода XFZ.

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
2.	\hat{u}_1	1	$\mathbb{DR}, 1$, окружение – 2	1 босс – X тип 2 – \mathbb{A}	2	1
4.	Предл.9.1	Предл.9.1	$\text{SBoss}(Z)$	как у X	\hat{u}_3	4
5.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3
7.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	1
10.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 25).

ТАБЛИЦА 25. Случай P7: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
2.	\hat{u}_3	4	\mathbb{C} , окружение – как у T	1 босс – $\text{TopFromCorner.Past}(X)$, 2 босс – X , 3 босс – Z	3	ru
4.	$\hat{1}$	3	\mathbb{A} окружение как у T	1 босс как у Z	2	1
5.	\hat{u}_3	4	\mathbb{C} окружение как у T	1 босс – как у Z 2 босс – X , 3 босс – \mathbb{B} , окружение (7A, x , 3B, 4A)	2	1
7.	$\hat{1}$	3	\mathbb{A} окружение как у T	1 босс – Z	1	u_2
10.	$\hat{1}$	1	$\mathbb{F}\text{Boss}(Z)$	1 босс – $\text{Prev}(X)$	u_2	1

9.9. P8: Макроплитка с окружением $(x, 3B, 1A, 3A)$; $x = \text{left, top}$. В этом параграфе мы рассмотрим случай P8, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение $(x, 3B, 1A, 3A)$; $x = \text{left, top}$. Это происходит, когда макроплитка является прямым потомком правой верхней подплитки некоторой макроплитки, являющейся левой нижней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

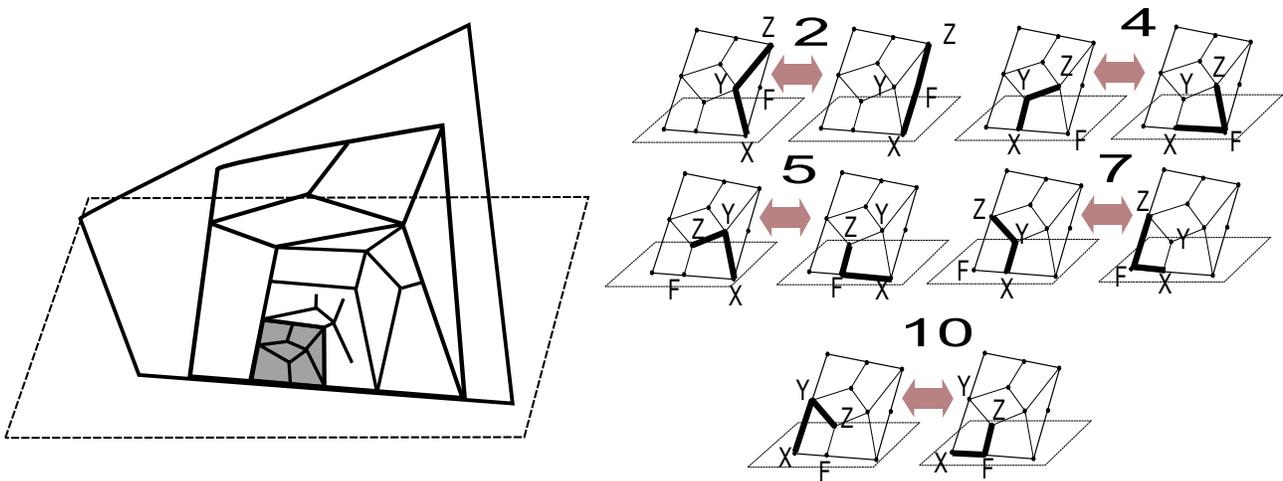


Рис. 124. Случай P8

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 26).

ТАБЛИЦА 26. Случай P8: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
2.	$\hat{1}$	1	$\mathbb{D}\mathbb{R}, 1,$ окружение – 3	1 босс – как у Z	2	3
4.	Предл.9.1	Предл.9.1	$\text{SBoss}(Z)$	как у X	$\hat{1}_2$	4
5.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3
7.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	1
10.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 27).

ТАБЛИЦА 27. Случай P8: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
2.	$\hat{1}_2$	4	\mathbb{C} , окружение – как у T	1 босс – $\text{TopFromCorner.Past}(X)$, 2 босс – X 3 босс – Z	3	ru
4.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс как у Z	2	1
5.	$\hat{1}_2$	4	\mathbb{C} , окружение как у T	1 босс – как у Z 2 босс – X , 3 босс – \mathbb{A} , окружение $\text{RightCorner.Past}(X)$	1	2
7.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс – Z	1	u_2
10.	$\hat{1}$	2	$\text{FBoss}(Z)$	1 босс – $\text{Prev}(X)$	u_2	1

9.10. **P9: Макроплитка с окружением $(x, 1A, 6A, 2A)$; $x = \text{left, top}$.** В этом параграфе мы рассмотрим случай **P9**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение $(x, 3B, 6A, 2A)$; $x = \text{left, top}$. Это происходит, когда макроплитка является правой верхней подплиткой некоторой макроплитки, являющейся левой верхней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 28).

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 29).

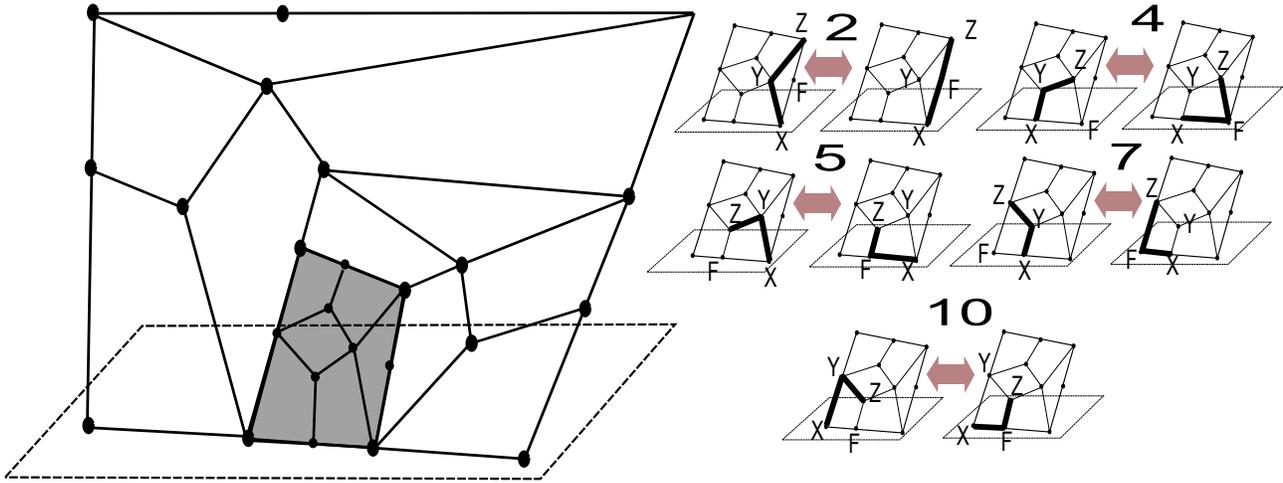


Рис. 125. Случай P9

ТАБЛИЦА 28. Случай P9: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
2.	\hat{u}_1	1	$\mathbb{DR}, 1,$ окружение $- 2$	1 босс $- X$ тип 2 $- \mathbb{A}$	2	1
4.	Предл.9.1	Предл.9.1	$\mathbb{SBoss}(Z)$	как у X	\hat{u}_3	4
5.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3
7.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	\hat{u}_2	1
10.	Предл.9.1	Предл.9.1	$\text{Next.FBoss}(Z)$	как у X	$\hat{1}$	3

ТАБЛИЦА 29. Случай P9: вычисление кода XYZ .

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
2.	\hat{u}_3	4	\mathbb{C} , окружение $-$ как у T	1 босс $-$ $\text{TopFromCorner.Past}(X),$ 2 босс $- X,$ 3 босс $- Z$	3	ru
4.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс как у Z	2	1
5.	\hat{u}_3	4	\mathbb{C} , окружение как у T	1 босс $-$ как у Z 2 босс $- X,$ 3 босс $- \mathbb{B}$, окружение (left, top, 1A, 3A)	1	2
7.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс $- Z$	1	u_2
10.	$\hat{1}$	1	$\mathbb{FBoss}(Z)$	1 босс $- X$	u_2	1

9.11. **P10: Макроплитка с окружением** $(x, 1A, 1A, 3A); x = \text{left, top}$. В этом параграфе мы рассмотрим случай **P10**, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение $(x, 3B, 1A, 3A); x = \text{left, top}$. Это происходит, когда макроплитка является прямым потомком правой верхней подплитки некоторой макроплитки, являющейся левой верхней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

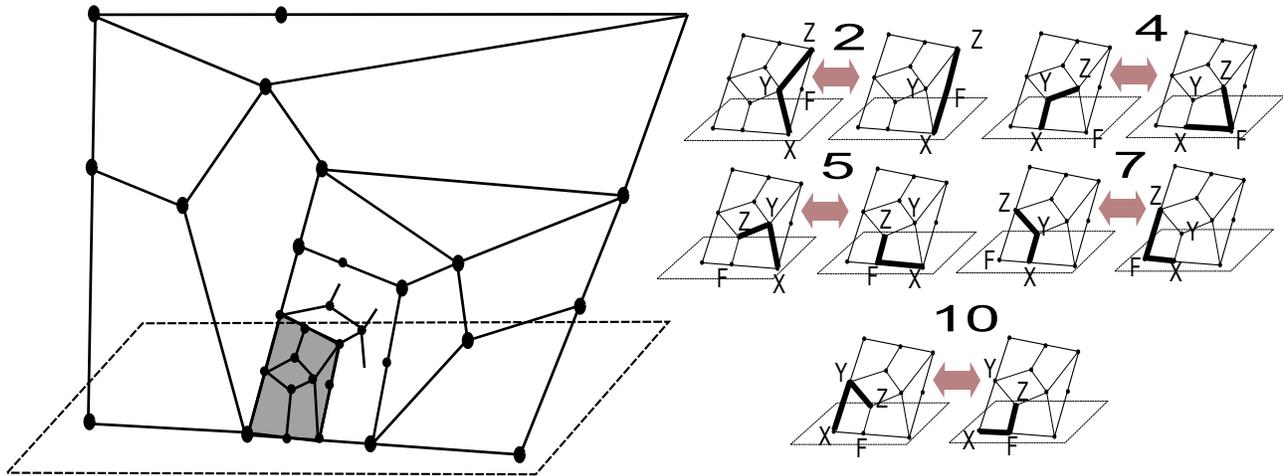


Рис. 126. Случай P10

Вычисление кода XFZ . Зная код пути XYZ (то есть коды X, Y, Z , а также ребра входа в Y, Z и ребра выхода из X и Y вдоль пути XYZ), можно выписать код XFZ (таблица 30).

ТАБЛИЦА 30. Случай P10: вычисление кода XFZ .

	ребро из X	ребро в F	тип, уровень и окружение F	информация F	ребро из F	ребро в Z
2.	$\hat{1}$	1	\mathbb{DR} окружение - 3	1 босс как у Z	2	3
4.	Предл.9.1	Предл.9.1	$SBoss(Z)$	как у X	$\hat{1}_2$	4
5.	Предл.9.1	Предл.9.1	$Next.FBoss(Z)$	как у X	$\hat{1}$	3
7.	Предл.9.1	Предл.9.1	$FBoss(Z)$	как у X	\hat{u}_2	2
10.	Предл.9.1	Предл.9.1	$Next.FBoss(Z)$	как у X	$\hat{1}$	3

Вычисление кода XYZ . Зная код пути XFZ (то есть коды X, F, Z , а также ребра входа в F, Z и ребра выхода из X и F вдоль пути XFZ), можно выписать код XYZ (таблица 31).

Таким образом, во всех случаях можно провести локальное преобразование, как и для плоских путей. Это дает нам возможности преобразовывать пути, меняя локальные участки на им эквивалентные и приводить слово к канонической форме.

9.12. Оценка числа введенных соотношений. В каждом случае мы выбирали ядро подклейки, после чего выбирали не более четырех вершин (Q_1, Q_2, Q_3, Q_4) на сторонах подклеенной макроплитки, для которых фиксировали их коды. После чего записывали не более 16 соотношений, для каждого выбранного случая. Учитывая конструкцию подклейки, можно заметить, что информация вершин Q_i может быть вычислена по информации ядра подклейки и известным ребрам-сторонам. То есть соотношений будет не более чем $16 \cdot F \cdot H^4$, где F – количество флагов подклейки, H – число расширенных окружений. То есть не более чем $16 \cdot 82485^4 \cdot 5 \cdot 10^{18} < 4 \cdot 10^{39}$ соотношений.

ТАБЛИЦА 31. Случай P10: вычисление кода XYZ.

	ребро из X	ребро в Y	тип, уровень и окружение Y	информация Y	ребро из Y	ребро в Z
2.	$\hat{1}_2$	4	\mathbb{C} , окружение – как у T	1 босс – <code>LevelPlus.FBoss(Z)</code> , 2 босс – X , 3 босс – Z	3	ru
4.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс как у Z	2	1
5.	$\hat{1}_2$	4	\mathbb{C} , окружение как у T	1 босс – как у Z 2 босс – X , 3 босс – \mathbb{A} , окружение <code>RightCorner.Past(X)</code>	1	2
7.	$\hat{1}$	3	\mathbb{A} , окружение как у T	1 босс – Z	1	u_2
10.	\hat{u}_2	2	$\mathbb{F}\text{Boss}(Z)$	1 босс – X	u_2	1

10. ПРИВЕДЕНИЕ К КАНОНИЧЕСКОЙ ФОРМЕ

В этой главе мы покажем, как проводить преобразования для приведения слова к канонической форме. Сначала мы рассмотрим пути, лежащие на одном ребре некоторой макроплитки, а затем разберем общий случай.

Пусть X_i – буквы, кодирующие входящие ребра, Y_i – буквы, кодирующие узлы (их типы, окружения и информации), Z_i – буквы, кодирующие выходящие ребра. Фактически, X_i и Z_i выбираются из одного алфавита ребер входа-выхода. Напомним, что слово W имеет *CODE-форму*, если в нем справа от любой (не последней в слове) буквы семейства X обязательно стоит буква семейства Y , справа от любой (не последней в слове) буквы семейства Y стоит буква семейства Z , а справа от любой (не последней в слове) буквы семейства Z стоит буква семейства X . Рассмотрим полугруппу S с нулем 0 и порождающими $\{X_i, Y_i, Z_i\}$. Будем считать, что в полугруппе S введены определяющие соотношения следующих категорий:

Категория 1: соотношения, обеспечивающие CODE-форму: $X_i Z_j = 0$, $X_i X_j = 0$, $Y_i X_j = 0$, $Z_i Y_j = 0$, $Y_i Y_j = 0$, $Z_i Z_j = 0$, для всевозможных пар (i, j) .

Категория 2: $U_i = 0$, где U_i пробегает всевозможные слова длины не более 4, не являющиеся кодировкой никакого пути на комплексе.

Категория 3: $U_i = 0$, где U_i пробегает всевозможные слова, являющиеся кодировкой какого-либо пути, представляющего нулевую форму (путь туда и обратно по некоторому ребру).

Категория 4: $U_i = 0$, где U_i пробегает всевозможные слова, длины не более 6, являющиеся кодировкой какого-либо пути, представляющего путь с мертвым паттерном (одним из мертвых паттернов, перечисленных в предложении 3.8).

Категория 5: $U_i = V_j$, где U_i и V_j – кодировки путей, участвующих в некотором локальном преобразовании. Конечное число таких пар приведено в главах 8 и 9.

Оценка числа соотношений. Букв в алфавите не более $7 \cdot 10^{36}$. Всевозможных слов длины не более 4 существует не более $2,401 \cdot 10^{147}$. Учитывая, что в других категориях соотношений значительно меньше, можно считать, что общее число соотношений, введенных в полугруппе, не превосходит $3 \cdot 10^{147}$.

Замечание. Введение соотношений можно значительно оптимизировать, сильно уменьшив их число. Но это выходит за рамки этой работы.

Далее мы рассмотрим следствия из этих соотношений.

10.1. Свойства путей, лежащих на одном ребре. Пусть W – последовательность боковых узлов, принадлежащих одному внутреннему ребру некоторой макроплитки. Поскольку мы ввели мономиальные соотношения для всех слов, не длиннее 10, не являющихся кодировками путей на комплексе, мы можем установить, какие слова в принципе могут быть разрешены в полугруппе S .

Определение 10.1. Определим *последовательность уровней на ребре* $\text{EdgeLevels}(k)$ в зависимости от натурального параметра k следующим образом:

$$\text{EdgeLevels}(1)=1;$$

$$\text{EdgeLevels}(2)=121;$$

$$\text{EdgeLevels}(3)=1213121;$$

$$\text{Для } k > 3, \text{ EdgeLevels}(k)=\text{EdgeLevels}(k-1) 3 \text{ EdgeLevels}(k-1).$$

Рассмотрим последовательность $\text{EdgeLevels}(k)$. Заменяем 1 на 2, 2 на 3, 3 оставляем как есть. После этого в начале и конце последовательности, а также между любыми двумя членами добавим 1. Можно показать по индукции, что при этом получится $\text{EdgeLevels}(k+1)$. также можно заметить, что последовательность периодическая, с периодом 3121 и предпериодом 121. Ясно, что для любого бокового ребра существует такое k , что последовательность $\text{EdgeLevels}(k)$ отвечает значениям уровней боковых узлов на этом ребре.

Предложение 10.1. Пусть W – слово в S , в котором все буквы, соответствующие узлам, кодируют боковые узлы, лежащие на одном и том же ребре, одинаковым для всех параметром информации. Кроме того, пусть все реберные буквы одинаковы и равны 1 или 2 (то есть все переходы по главным ребрам). Тогда:

1. Уровни узлов на ребре идут в последовательности $\text{EdgeLevels}(k)$ для некоторого натурального k .

2. Для каждого узла X , уровень которого выше 1, существует подклееная макроплитка T , такая что тип X относительно T (тип в подклейке) равен либо CUR либо CDL, причем оба таких типа (для разных макроплиток) могут быть только у одного узла в W , находящегося в середине внутреннего ребра макроплитки.

Доказательство. Первое утверждение следует из того, что в любом комплексе уровни идут именно в такой последовательности. То есть, все слова длины не более 20, не являющиеся подсловами такой периодической последовательности, присутствуют среди мономиальных соотношений. Из этого следует, что если последовательность узлов не содержит запрещенных подслов, последовательность уровней узлов соответствует указанной.

Для доказательства второго утверждения рассмотрим, как устроены последовательности узлов на внутренних ребрах построенного комплекса. После операции разбиения, узлы первого уровня становятся узлами второго. При операции подклейки, каждый узел второго уровня становится либо правым верхним, либо левым нижним углом новой подклеенной макроплитки, причем в дальнейшем к этим узлам макроплитки углами не подклеиваются. Единственным узлом, в подклеенном типе которого может присутствовать и CUR и CDL, является узел в середине внутреннего ребра, так как он может участвовать в нескольких подклеенных макроплитках.

Ранги боковых узлов. Пусть W – последовательность боковых узлов, принадлежащих одному внутреннему ребру некоторой макроплитки. Будем говорить, что *боковой узел*

имеет ранг 1 (или, соответственно, ранг 2), если этот боковой узел – первого (или, соответственно, второго) уровня. *Регулярным словом* ранга 1 будем считать слово из одного узла ранга 1.

Теперь индуктивно определим остальные ранги узлов. *Регулярным словом* ранга n будем называть последовательность узлов $X Y X$, где X – регулярное слово ранга $n - 1$, а Y – узел ранга n . То есть, например, если ранги узлов идут в последовательности 121, то это регулярное слово ранга 2. Будем говорить, что *боковой узел X имеет ранг n* , если выполнены следующие условия:

1. X находится в середине подслова $\widehat{W} \subset W$, состоящего из $2^{n-1} + 1$ букв;
2. Слово \widehat{W} представляется в форме $Y U X V Z$, где Y, Z – боковые узлы ранга $n - 1$, U, V – регулярные слова ранга $n - 2$.

Слово \widehat{W} будем называть *словом-представителем* для буквы X ранга n .

Диаграммой рангов слова U будем называть слово \widehat{U} в счетном алфавите $\{1, 2, \dots\}$, получаемое из слова U заменой каждой буквы на ее ранг. Например, узел X имеет ранг 3, если в W есть подслово $Z_1 Y_1 X Y_2 Z_2$, где у Z_1 и Z_2 ранг 2, а у Y_1 и Y_2 ранг 1. Тогда диаграмма рангов выглядит так: 21312.

Замечание. Смысл понятия *слово-представитель X* в том что слово-представитель является половиной периметра (верхняя плюс левая сторона) подклеенной макроплитки, ядром которой является X .

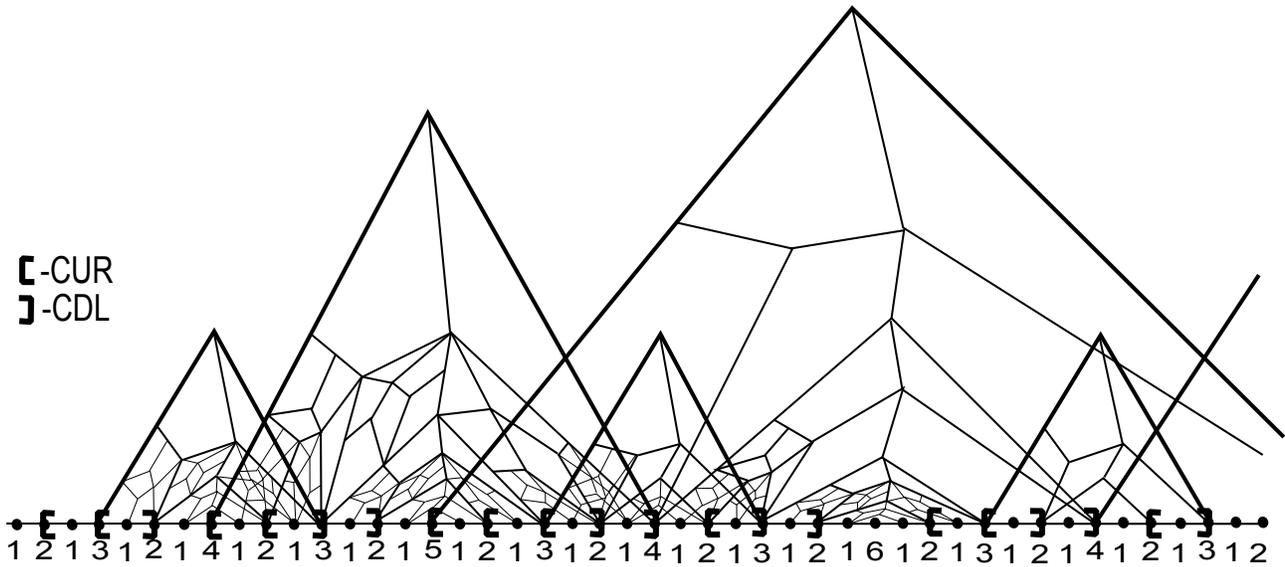


Рис. 127. Устройство подклеек на ребре

Предложение 10.2. Пусть W – последовательность узлов на одном внутреннем ребре некоторой макроплитки и узел X имеет ранг n . Рассмотрим слова длиной $2^k + 1$, с центром в X . Его слово-представитель, имеющее длину $2^{n-1} + 1$, является наименьшим среди таких слов, левый узел которых имеет в подклеенном типе CUR, а правый – CDL.

Это утверждение доказывается индукцией по n .

Предложение 10.3. Пусть W – слово в S , в котором все буквы, соответствующие узлам, кодируют боковые узлы, причем все они лежат на ребре одного и того же типа, и параметр информации у всех одинаковый.

Пусть X – буква в слове W , уровень которой равен 3, причем между краями W и X находятся, как минимум, 4 буквы, отвечающие кодам узлов. Тогда либо некоторое подслово $\widehat{W} \in W$, содержащее X , приводится к нулю с помощью определяющих соотношений в S , либо существует подслово в W , являющееся словом-представителем X и оно может быть реализовано в качестве кодировки пути на некотором ребре комплекса.

Доказательство. Будем считать, что мы читаем слова в положительном (относительно несущего ребра) направлении. Пусть уровень X (то есть и ранг тоже) равен 3. Рассмотрим подслово W , содержащее 5 узловых букв, с центром в X .

Допустим, первая и последняя буквы содержат в подклеенном типе, соответственно, CDL и CUR . Если W не встречается на комплексе, оно должно присутствовать среди мономимальных определяющих соотношений. Иначе оно автоматически реализуется в качестве кодировки пути. В этом случае W – кодировка пути по верхней и левой стороне подклеенной макроплитки уровня 2, и W – слово-представитель X .

Пусть, теперь, первая и последняя буквы содержат в подклеенном типе, соответственно, CUR и CDL . (Иные варианты не встречаются на комплексе в качестве кодировок и потому приводятся к нулю).

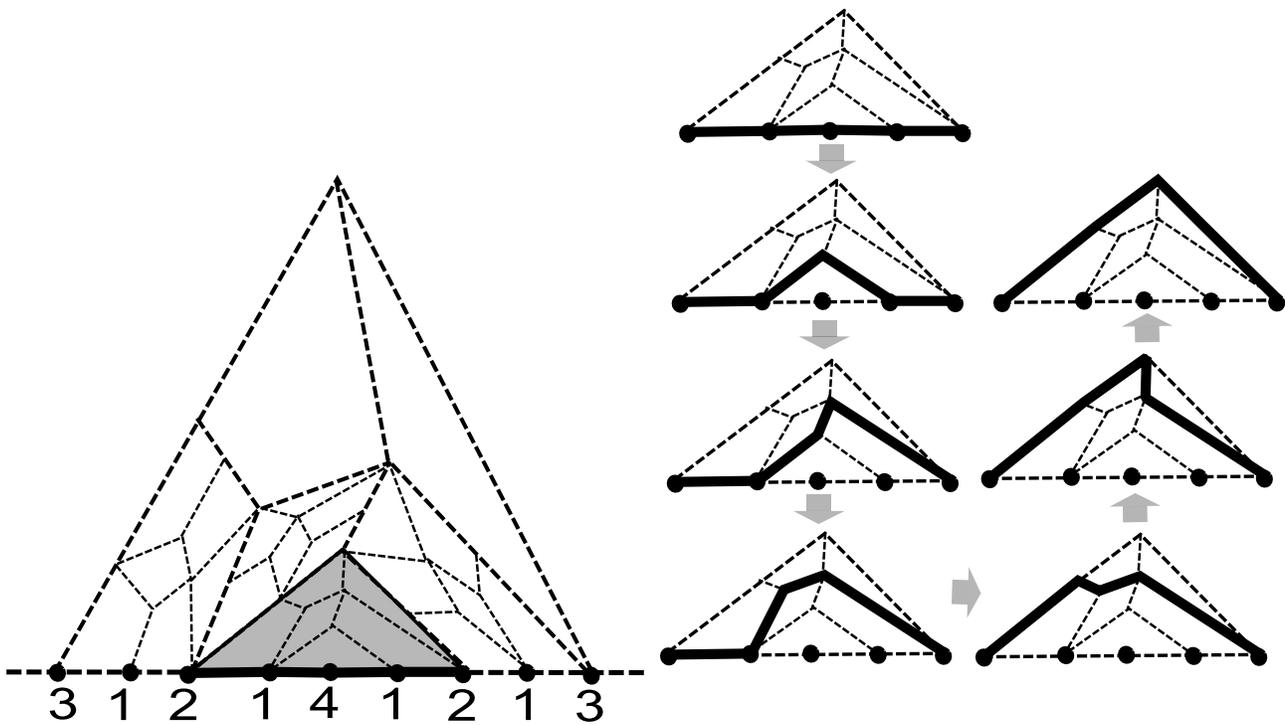


Рис. 128. Преобразование слова-представителя

В этом случае ранг центральной буквы – 4 или выше. Рассмотрим слово V_1 из девяти букв с центром в X . Диаграмма рангов V_1 имеет вид $3121x1213$. Слово V_1 встречается на комплексе, иначе все можно сразу привести к нулю. Рассмотрим типы (в подклеенной части) для четвертой и шестой букв этого слова. Это должны быть разные типы, а учитывая положительное прочтение слова, четвертая должна иметь тип \mathbb{U} , а шестая \mathbb{L} (речь о типах в подклеенной области). То есть, к подслову из четвертой, пятой и шестой букв может быть

применено соотношение для локального преобразования из числа введенных в случае $P2$. Затем можно применить другие локальные преобразования 7 из числа введенных в случае $P2$, для соответствующих трехбуквенных слов. В результате мы преобразуем слово V_1 в кодировку пути, изображенного на рисунке 128.

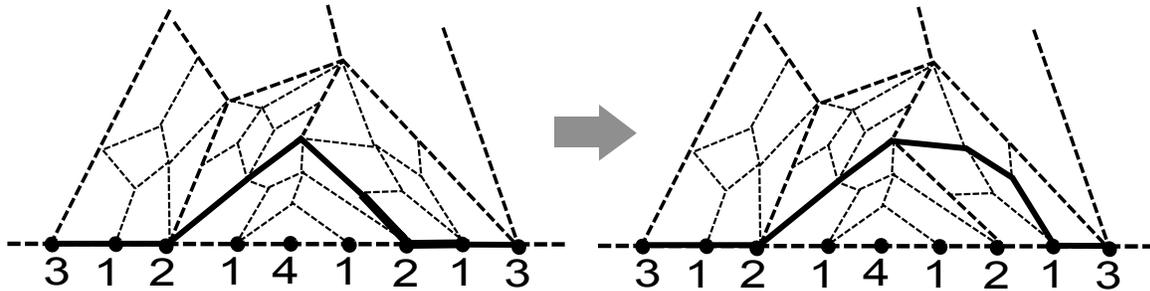


Рис. 129. Преобразование слова-представителя

Рассмотрим трехбуквенное подслово, отмеченное в левой части на рисунке 129. Заметим, что либо путь с таким кодом не может встретиться на комплексе (и тогда оно равно нулю), либо к нему можно применить соотношение $P3 - 8$. Действительно, узел X обязан иметь уровень 1, значит, базовый тип его UL или LU , а тип в подклейке – U (если будет L , то такого участка из трех узлов не может встретиться на комплексе). Затем, аналогично применяем соотношения $B3 - 6$, $P3 - 1$, и получаем кодировку пути, изображенного в правой части рисунка 129.

Далее возможно два варианта. Последняя буква слова V_1 может содержать в подклеенном типе либо CDL либо CUR .

В первом случае мы можем применить соотношение $P3 - 10$, далее $B3 - 5$, $B3 - 2$. Применяя далее подходящие соотношения, можно получить из слова кодировку пути, изображенного в левой части на рисунке 130. Согласно предложению 10.2, в этом случае рассмотренное девятибуквенное слово и есть слово-представитель для X . Применив далее несколько соотношений, мы можем преобразовать это слово в кодировку пути, указанного в правой части рисунка 130.

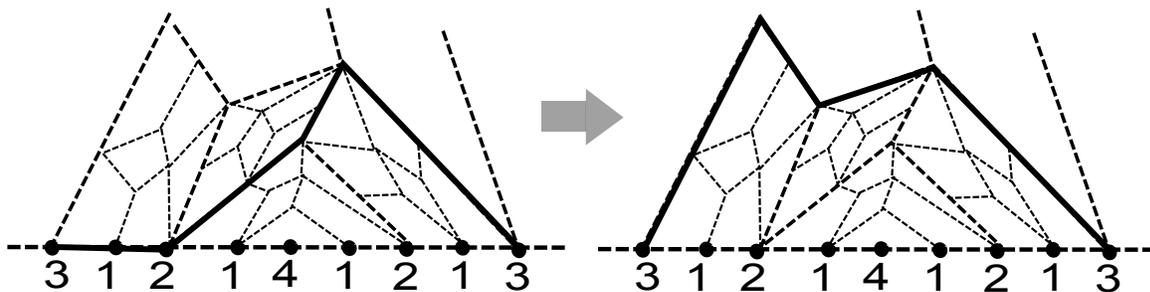


Рис. 130. Преобразование слова-представителя

Во втором случае мы продолжаем применять соотношения, пока не получим из нашего девятибуквенного слова кодировку указанного на рисунке 131 пути.

Теперь рассмотрим 17-буквенное слово V_2 с центром в X (рисунок 131). Мы опять можем брать трехбуквенные слова на границе 17-буквенного слова так, что либо эти трехбуквенные слова будут встречаться в мономиальных соотношениях, либо в соотношениях

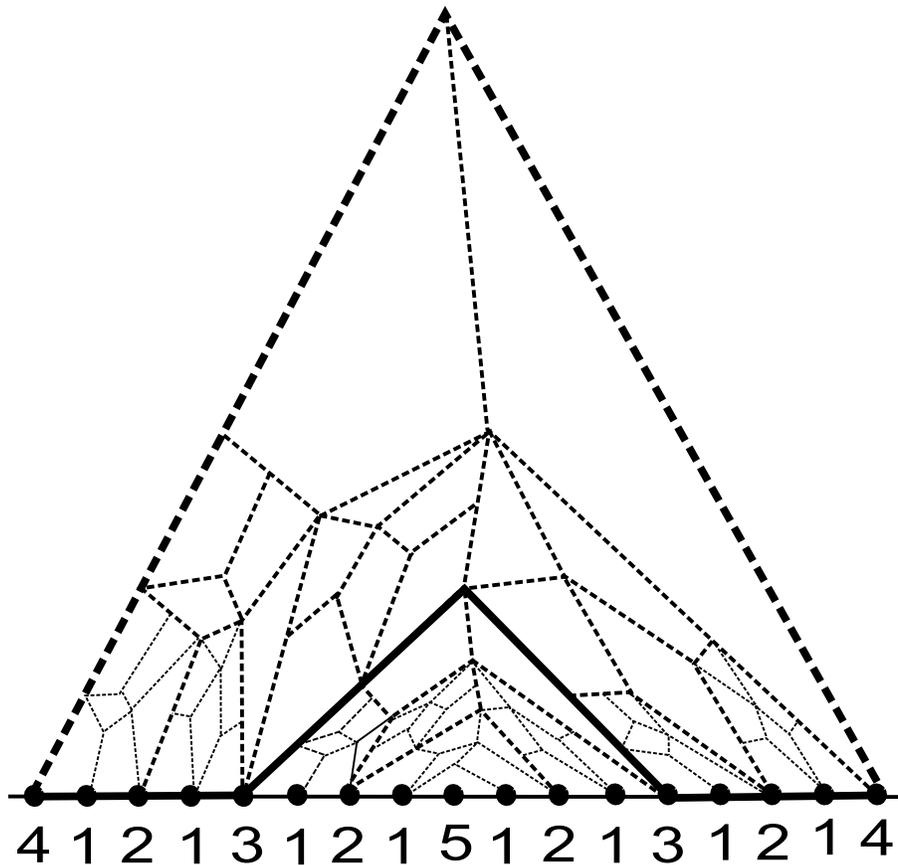


Рис. 131. Преобразование слова-представителя

описанных в главах про локальные преобразования путей. После проведения необходимых преобразований, мы опять получим альтернативу: последняя буква слова V_2 может содержать в подклеенном типе либо CDL либо CUR. Аналогично 9-буквенному слову, мы либо получаем кодировку пути, полученного из слова-представителя, либо возможность продолжить процесс и перейти к рассмотрению 33 буквенного слова.

Продолжая процесс преобразования, мы дойдем до того момента, когда сможем окончательно выписать кодировку пути, полученного из слово-представителя, либо получим в какой-то момент нулевое слово.

Предложение 10.4. Пусть W – слово в S , удовлетворяющее следующим условиям:

1. W состоит из трех частей $W = V_1 X V_2$, где

X – узловая буква, соответствующая внутреннему узлу (A, B или C), либо X – любая узловая буква, но ребро входа в X или ребро выхода из X – неглавное;

V_1, V_2 – слова, все узловые буквы которых обладают одной и той же информацией (своей для V_1 и для V_2), а все реберные буквы (кроме, возможно, входящего в X и выходящего из X) соответствуют главным ребрам, причем тип ребра постоянен для V_1 и для V_2 .

Тогда либо некоторое подслово $\widehat{W} \in W$, содержащее X , приводится к нулю с помощью определяющих соотношений в S , либо существует подслово в W являющееся словом-представителем X и оно может быть реализовано в качестве кодировки пути на некотором ребре комплекса.

Доказательство полностью аналогично предложению 10.3. В данном случае в процессе доказательства используются соотношения, связанные с подклеенной макроплиткой с ядром в X .

Предложение 10.5. Пусть W – слово в S и каждая узловая буква W кодирует боковой узел, причем информация всех таких узлов совпадает, а каждая реберная буква соответствует главному ребру (то есть для всех 1 или для всех 2). Тогда либо W можно привести к нулю с помощью определяющих соотношений в S , либо W представляется в виде кодировки некоторого пути на комплексе.

Доказательство. Заметим, что если уровень всех узловых букв не выше 2, то W не может быть содержать более 3 узловых букв, в этом случае W имеет малую длину и если в нем не содержится запрещенных слов, значит оно реализуется в качестве кодировки пути.

Выберем некоторую букву X , уровня 3. Пусть ее ранг равен k . Применим к X предложение 10.3. Получим $V(X)$ – слово-представитель X . Обозначим крайнюю справа букву $V(X)$ как X_1 . Ее ранг будет меньше на 1, чем ранг X . Применим к X_1 предложение 10.3, опять возьмем в $V(X_1)$ крайнюю справа букву. Продолжим так применять предложение 10.3, пока ранг и уровень всех буквы X_i , которой равны 3. Если в какой-то момент слово W закончилось, мы можем рассмотреть аналогичный процесс, взяв на каждом шаге крайние слева буквы слов-представителей.

Будем пока считать, что слово не заканчивается. Тогда мы в итоге получим букву X_i , ранг и уровень которой равны 2. Следующая за X_i узловая буква должна иметь уровень 1, иначе возникнет запрещенное слово. Рассмотрим еще одну следующую узловую букву Y . Ее уровень равен 3, применим к ней предложение 10.3. Можно заметить, что левый конец полученного слова-представителя $V(Y)$ – это буква X . То есть, ранг Y на один выше, чем ранг X .

Таким образом, с помощью указанного процесса, мы можем по заданной узловой букве либо найти букву с большим рангом, либо дойти до конца слова (в обе стороны). Во втором случае указанная буква будет обладать наивысшим рангом из всех узловых букв.

Пусть буква X – обладает максимальным рангом из всех узловых букв W . Теперь применим предложение 10.3 к X и рассмотрим крайние буквы X_L и X_R полученного слова-представителя. Заметим, что их ранг на 1 меньше, чем ранг X . Для каждой из этих букв зафиксируем пару соседних с ней реберных букв так, что одно ребро уходит в подклейку (соответствующую слову-представителю X), а другая – соответствует главному ребру. Применим к этим буквам с такими парами ребер предложение 10.4, получая расширение подслова, представимого в качестве кодировки некоторого пути. Продолжим далее аналогичным образом, пока крайние буквы не будут иметь ранг 2. Следующие две буквы с обоих концов должны иметь ранг 1, и полученное таким образом слово представляется в виде кодировки пути на комплексе. Заметим, что если после этого слева или справа имеется какая-либо буква, она должна иметь больший ранг, чем X (в этом можно убедиться, применив для этой буквы предложение 10.3). Таким образом, все слово может быть представлено в виде кодировки пути на комплексе.

Пусть у нас получился путь P на комплексе. Теперь можно проделать над ним все локальные преобразования в обратном порядке, возвращая слову форму, где все узловые буквы принадлежат одному ребру и все переходы – главные. Путь P преобразуется в эквивалентный путь, идущий по одному внутреннему ребру некоторой макроплитки.

Предложение 10.6. Пусть W – слово в S , удовлетворяющее следующим условиям:
 W состоит из трех частей $W = V_1XV_2$, где

X – узловая буква, соответствующая внутреннему узлу (\mathbb{A} , \mathbb{B} или \mathbb{C}), либо X – любая узловая буква, но ребро входа в X или ребро выхода из X – неглавное;

V_1, V_2 – слова, все узловые буквы которых обладают одной и той же информацией (своей для V_1 и для V_2), а все реберные буквы (кроме, возможно, входящего в X и выходящего из X) соответствуют главным ребрам, причем тип ребра постоянен для V_1 и для V_2 .

Тогда либо W можно привести к нулю с помощью определяющих соотношений в S , либо W представляется в виде кодировки некоторого пути на комплексе.

Доказательство аналогично предложению 10.5.

Предложение 10.7 (О пути внутри макроплитки). Пусть W – слово в S , отвечающее следующим условиям:

1. В слове W отсутствуют буквы, кодирующие входы-выходы в подклейку;
2. Первая и последняя буквы слова W – узловые узлы;
3. Вторая буква слова W кодирует неглавное выходящее ребро;
4. Предпоследняя буква слова W кодирует неглавное входящее ребро;
5. Кроме второй и предпоследней букв в слове W ни одна буква не кодирует неглавных входящих или выходящих ребер (то есть все остальные буквы, кодирующие ребра – отвечают главным ребрам).

Тогда W либо может быть приведено к нулю, либо W является кодировкой некоторого пути, соединяющего два узла на периметре некоторой макроплитки.

Доказательство. Можно считать, что слово W имеет CODE-форму, то есть буквы, кодирующие входящие ребра, узлы и выходящие ребра, встречаются в слове в правильном порядке. В противном случае слово сразу можно привести к нулю, с помощью соотношений категории 1. Также, будем считать что всякое подслово длины 4 в нашем слове является регулярным, иначе, можно применить мономиальное соотношение из категории 2.

Согласно CODE-форме, буквы $1, 4, 7, \dots, 3k + 1$ слова W кодируют узлы. Буквы $2, 5, 8, \dots, 3k + 2$ – кодируют выходящие ребра, а буквы $3, 6, 9, \dots, 3k$ – кодируют входящие ребра.

Неглавное ребро входа означает, что путь входит внутрь некоторой макроплитки. Заметим, всего существует шесть возможных “входов” внутрь макроплитки: по ребрам 1 и 2 (из середины верхней стороны), по ребру 3 из середины левой стороны, по ребру 6 из середины правой стороны, по ребру 7 из левого нижнего угла и по ребру 8 из правого нижнего угла.

По второй букве W , кодирующей вход в макроплитку, можно установить, с каким из указанных шести входов мы имеем дело. Допустим, сначала, что это вход по ребру 3 из середины левой стороны, а выход (предпоследняя буква) – по ребру 1 в середину верхней стороны.

Допустим, четвертая буква W соответствует черному внутреннему узлу (\mathbb{A} , \mathbb{B} или \mathbb{C}). Рассмотрим седьмую букву, соответствующую третьему по очередности узлу вдоль нашего пути. Если эта буква является последней в W , то W представляет собой путь из середины левой стороны через узел типа \mathbb{A} в середину верхней стороны. Все слова не соответствующие этому, не кодируют никакого пути на комплексе и поэтому содержатся среди мономиальных соотношений категории 2. Значит, седьмая буква не является последней. Заметим, что если она соответствует боковому узлу, то подслово W из первых семи букв будет кодировать несуществующий путь. В случае, когда седьмая буква соответствует внутреннему узлу, можно рассмотреть возможные типы десятой и тринадцатой букв. Можно заметить, что если путь с такой кодировкой существует, в нем содержится

участок, проходимый туда и обратно (нулевая форма) и в этом случае, в нашем слове будет подслово из категории 3 мономиальных соотношений.

Теперь пусть четвертая буква W кодирует боковой узел. Пусть в слове W все буквы кодируют боковые узлы. Участок пути из одного бокового узла в другой по главному ребру не меняет тип этого ребра. В нашем случае это ребро 3, причем с левой стороны – сторона 3А, а с правой – 3В. Но при выходе (когда мы приходим в середину верхней стороны) ребро должно быть типа 1, то есть найдется локальный участок, где тип ребра меняется, значит можно применить соотношение из категории 2.

Итак, после цепочки боковых узлов, нам должен встретиться внутренний узел. Поскольку у нас ребро 3, это может быть только узел типа А, для других не будет реализующих путей на комплексе. После узла А, мы можем продолжать путь по ребру 1, либо по ребру 4. (Если путь далее продолжается по ребру 3, то образуется участок с нулевой формой и опять же можно применить мономиальное соотношение.)

В случае ребра 4, мы продолжим изучение последующих букв, кодирующих узлы. Аналогично можно установить, что после нескольких боковых узлов, мы должны встретить узел типа С, иначе можно будет применить мономиальное соотношение. После узла С, дальнейший путь может продолжиться по одному из ребер 7, 8 или 5. В первых двух случаях, если после цепочки боковых, мы встретим внутренний узел, можно будет применить мономиальное соотношение, так как такого участка пути не может существовать на комплексе.

Если же дальнейший путь идет по ребру 5, опять рассмотрим цепочку боковых узлов до первого внутреннего узла (опять же, боковые узлы не могут поменять тип ребра, а в конце ребро должно быть первого типа). Это может быть только узел типа В и дальнейший путь может идти только по ребрам 2 или 6. В обоих случаях, для любого встреченного узла внутреннего типа можно будет применить мономиальное соотношение.

Таким образом, наше слово должно содержать цепочку боковых узлов на ребре 1, потом А-узел, потом цепочка боковых узлов типа 3. Если указанные цепочки содержат не более 10 букв, кодирующих узлы, то можно напрямую либо найти все наше слово среди запрещенных, а если его там нет, значит слово реализуется в качестве кодировки. Пусть указанные цепочки достаточно длинные. Временно выбросим из слова три первые и три последние буквы. Останется слово, к которому мы можем применить предложение 10.6, выбрав в качестве X наш А-узел.

То есть, урезанное слово без первых трех и последних трех букв может быть представлено в качестве кодировки пути на комплексе. Пусть это будет путь внутри некоторой макроплитки T , с заданными окружениями тех узлов на границе T , которые являются начальниками узлов из W .

При этом первые три и последние три буквы кодируют вход и выход в макроплитку, плюс информацию двух узлов на границе. Если вход или выход не соответствует ребрам, по которым далее должен идти путь, мы сразу можем применить мономиальное соотношение. Значит, окружение макроплитки и окружение и информация двух входных узлов могут сочетаться в одном объекте на комплексе. Тогда получаем, что наше слово реализуется в качестве кодировки некоторого пути на комплексе.

Замечание 1. Если вход и выход осуществляются не по ребрам 3 и 1, можно провести полностью аналогичный побуквенный анализ слова, и найти подслово категорий 2 или 3, либо установить, что слово есть кодировка некоторого пути в макроплитке. Таким образом, предложение можно доказать для входов внутрь макроплитки по ребрам любых внутренних типов.

Замечание 2. Неглавные ребра в начале и конце слова могут также быть ребрами в подклейку, рассуждения при этом не меняются.

Предложение 10.8 (О сокращении подпути по подклеенной макроплитке). Пусть W – слово в S , причем первая и последняя буквы кодируют некоторые узлы (то есть это некоторые из букв Y_i), а вторая и предпоследняя – кодируют, соответственно входящее ребро в подклейку и выходящее ребро из подклейки, причем кроме этих двух, больше входов и выходов в подклеенные области в слове W не встречаются. Тогда W либо может быть приведено к нулю, либо к форме, содержащей менее двух входов или выходов в подклеенные области.

Доказательство. Можно считать, что W имеет CODE-форму, то есть буквы, кодирующие входящие ребра, узлы и выходящие ребра, встречаются в слове в правильном порядке.

Скобочной структурой слова будем называть слово в алфавите из двух букв: $\{[,]\}$ открывающей и закрывающей скобок, составленное следующим образом: – каждая буква, кодирующая выходящее неглавное ребро, или ребро, ведущее в подклеенную макроплитку, заменяется на “[”, а каждая буква, кодирующая входящее неглавное ребро, или ребро, ведущее из подклеенной макроплитки, заменяется на “]”. Остальные буквы заменяются пустыми буквами.

Рассмотрим скобочную структуру W . Она начинается открывающей скобкой и заканчивается закрывающей. Очевидно, что существует подслово в W , которому в скобочной структуре соответствует подслово “[]” (из открывающей и закрывающей скобок). Это подслово \widehat{W} удовлетворяет условиям предложения 10.7, то есть, в некоторой макроплитке есть путь, соединяющий точки периметра, с таким кодом. Но тогда есть цепочка локальных преобразований, приводящая этот путь к эквивалентному, но идущему по периметру той же макроплитки. Поскольку локальным преобразованиям отвечают соотношения из категории 5, подслово \widehat{W} можно привести к форме, скобочная структура которой содержит не более одной скобки (а не две). Отметим также, что начальная и конечная скобки не исчезают при такой операции.

Будем проводить такие операции, пока во всем слове W не останется только открывающей и закрывающей скобок. Теперь наше слово соответствует пути, все ребра которого – главные, кроме первого и последнего. Данная ситуация аналогична условиям предложения 10.7. Таким образом, если наше слово ненулевое, оно представляет некоторый путь и мы можем провести цепочку локальных преобразований, чтобы этот путь шел по периметру макроплитки. Такой путь не может содержать более одного ребра в подклейки. Таким образом, в результате преобразований, в слове понизилось количество ребер в подклейки.

Предложение 10.9 (О двух выходах в подклейку подряд). Пусть W – слово в полугруппе S , представляющееся в виде $W = xV_1yV_2$, где x, y, z – буквы, кодирующие выходящие в подклейку ребра, а V_1, V_2 – слова, не содержащие букв, кодирующих возвращающиеся из подклеек ребра.

Тогда если количество узловых букв в V_2 более чем в 5 раз превышает количество узловых букв в V_1 , то слово может быть приведено к нулю в S .

Доказательство. Рассмотрим подслово V_2 . Допустим, все его реберные буквы кодируют главные ребра. Если среди узловых букв V_2 нет букв, соответствующих узлам A, B, C, R, D, CD, R , то все узлы V_2 боковые и можно применить предложение 10.6 к слову $W = xV_1yV_2$, где в качестве узла X из формулировки предложения 10.6 используется последний узел в V_1 (узел, где мы выходим в подклейку второй раз). Так как одно ребро длиннее другого, процесс преобразований приведет к ситуации, когда подслово из трех

узлов, к которому мы хотим применить локальное преобразование, представляет собой запрещенный участок пути, то есть можно применить мономиальное соотношение.

Если среди узловых букв V_2 есть буквы, соответствующие узлам A, B, C, R, D, CDR , то можно заметить, что этих узлов не более 4. Иначе можно проанализировать, как путь ходит по макроплитке и привести кусок пути к нулевой форме (рассуждения, аналогичные используемым в предложении 10.7). Заметим также, что расстояние между двумя такими узлами (если считать его в количестве узловых букв) будет одинаковым. Иначе можно применить предложение 10.6 взяв в качестве X букву на границе между неравными участками. Таким образом, общая длина V_2 не более чем в пять раз превышает длину начального участка (от начала V_2 до первого узла из числа указанных). Длина начального участка не превышает длины V_1 (по тем же причинам, как если в V_2 есть только боковые узлы).

Пусть среди реберных букв V_2 есть переходы по неглавному ребру (то есть входы в подплитку). Будем считать, что есть только входы по неглавному ребру, так как если есть выходы, то можно применить сокращение пары скобок (предложение 10.7). Отметим букву с каждым таким входом угловой скобкой “ $<$ ”. Согласно предложению 3.6, путь, проходящий по макроплитке уровня n , имеет длину менее $5 \cdot 2^{n-2}$. То есть, каждый кусок пути между узлами, отмеченными угловой скобкой, будет менее $5 \cdot 2^{n-2}$, где n – уровень макроплитки, где проходит этот участок. С каждым прохождением угловой скобки мы попадаем в макроплитку, уровень которой как минимум, на 1 меньше чем был до этого. Значит, общая длина пути не может превышать $5 \cdot 2^{k-2}$, где k – уровень макроплитки, в которую мы переходим в узле V_2 .

При этом, учитывая предложение 3.12, количество узловых букв (или длина по ребрам) в подслове V_1 не менее 2^{k-1} . Отсюда следует наше утверждение.

Предложение 10.10 (О безквадратности пути по одному ребру). *Пусть W – путь на комплексе, целиком проходящий внутри некоторого ребра (внутреннего или граничного) и не содержащий узлов с типами $DL, LD, DR, RD, RU, UR, R, D$. Тогда соответствующее слово в полугруппе S не содержит двух одинаковых подслов, идущих подряд.*

Доказательство. Рассмотрим подстановочную систему (или DOLL-систему) в алфавите $\{U_1, L_1, U, L\}$, определяемую следующим образом: $U_1 \rightarrow U_1UL_1, L_1 \rightarrow U_1LL_1, U \rightarrow U, L \rightarrow L$.

Такая система, в частности, генерирует следующую последовательность слов:

$$U_1 \rightarrow U_1UL_1 \rightarrow U_1UL_1UU_1LL_1 \rightarrow U_1UL_1UU_1LL_1UU_1UL_1LU_1LL_1 \rightarrow \dots$$

Пусть X – некоторое слово. Обозначим описанную подстановочную замену как $f(X)$.

Пусть сначала путь отвечающий W проходит по внутреннему ребру. Рассмотрим кодировку нашего слова W в полугруппе S . Проведем над ним следующие операции:

- каждую букву, кодирующую UL -узел, уровня выше первого, заменим на букву U ;*
- каждую букву, кодирующую LU -узел, уровня выше первого, заменим на букву L ;*
- каждую букву, кодирующую UL -узел первого уровня, заменим на букву U_1 ;*
- каждую букву, кодирующую LU -узел первого уровня, заменим на букву L_1 ;*
- все буквы, кодирующие не узловые буквы, удалим из слова.*

Получившееся слово в алфавите $\{U_1, L_1, U, L\}$ будем называть W' . Заметим, что если W содержит два одинаковых подслова, идущих подряд, то таким же свойством будет обладать и W' .

Покажем, что W' является подсловом $f^n(U_1)$ или $f^n(L_1)$ для некоторого n . Действительно, выберем в нашем слове W букву, X кодирующую узел с самым большим рангом n (узел, появившийся ранее всех при разбиениях). Такая буква только одна, так как если бы их было хотя бы две, то согласно правилам разбиения, между ними нашелся бы узел с более высоким рангом.

Допустим, ради определенности, что узел X имеет тип UL . Можно заметить, что последовательность узловых букв в слове W' является подсловом в слове $f^{n-1}(U_1)$, это легко можно установить индукцией по n , подстановочная последовательность f полностью соответствует образованию узловых вершин при разбиениях.

Таким образом, для доказательства предложения, достаточно показать, что $f^n(U_1)$ не содержит двух одинаковых подслов подряд для любого n .

Допустим, это не так. Возьмем наименьшее n , при котором такие подслова появляются. Пусть $f^n(U_1)$ содержит подслово QQ . Разметим наше слово следующим образом. Выделим в $f^n(U_1)$ обычными (не квадратными) скобками участки, соответствующие подсловам U_1UL_1 , U_1LL_1 , U , V , образовавшимся при последнем переходе $f^{n-1}(U_1) \rightarrow f^n(U_1)$. Сами скобки не принадлежат алфавиту, а являются просто инструментом разметки. Если при этом в Q вошло целое количество участков, начинающихся и заканчивающихся скобкой, то в слове $f^{n-1}(U_1)$ также будет два одинаковых подслова $f^{-1}(Q)$ идущих подряд, что дает противоречие с выбором n .

Значит есть подслово (U_1UL_1) или (U_1LL_1) , часть которого лежит вне Q , а часть внутри. То есть, Q начинается с UL_1) или LL_1), либо с U_1) или L_1).

Рассмотрим первый случай. Q должно заканчиваться на $(U_1$, то есть Q представляется в виде $UL_1)Y(U_1$, где Y – слово содержащее целое количество участков в скобках и к нему можно применить f^{-1} . Тогда $f^n(U_1)$ представляется в виде

$$W_1(U_1UL_1)Y(U_1UL_1)Y(U_1UL_1)W_2,$$

где W_1, W_2 – некоторые подслова, содержащие целое число участков в скобках.

Можно заметить, что при применении f^{-1} к подслову $(U_1UL_1)Y(U_1UL_1)Y(U_1UL_1)$ там также образуется два одинаковых подслова идущих подряд. То есть такие слова встретятся в $f^{n-1}(U_1)$, что невозможно.

Остальные случаи: Q начинается с LL_1); U_1); L_1) рассматриваются полностью аналогично.

Предложение 10.11. Пусть W – произвольное слово в S . Тогда слово W^9 приводится к нулю.

Доказательство. Отметим в слове W открывающими скобками – ребра, кодирующие входы в подклейки, а закрывающими – выходы. Если подслово начинается с открывающей скобки, заканчивается закрывающей, и не содержит внутри скобок, будем применять предложение 10.8. Таким образом, в какой-то момент мы преобразуем наше слово W в форму $\widehat{W} = W_1W_2$, где W_1 содержит только закрывающие скобки, а W_2 только открывающие.

Заметим, что $W^{n+1} = W_1(W_2W_1)^nW_2$. Рассмотрим слово W_2W_1 . Применяя к нему такой же процесс (предложение 10.8 к каждой паре скобок), мы в итоге получим слово V , либо не содержащее скобок вообще, либо содержащее только один их вид.

Пусть, например, V содержит только открывающие скобки, то для $n \leq 7$, V^n содержит подслово xV_1yV_2 , где x, y – открывающие скобки (буквы, кодирующие ребра в подклейки), а количество узловых букв в V_2 превышает число узловых букв в V_1 в более чем в пять раз.

Таким образом, учитывая предложение 10.9, в этом случае V^n можно привести к нулю, а значит и W^9 тоже.

Пусть V не содержит скобок вообще. В этом случае мы имеем дело с плоской ситуацией, без выходов в подклейки. Расставим скобки новым образом: открывающие для неглавных выходящих из узла ребер, а закрывающие – для неглавных входящих в узел ребер. Фактически, теперь открытие и закрытие скобок символизирует вход и выход в макроплитку. Теперь будем применять предложение 10.7 для участков внутри пары правильно расположенных скобок. Таким образом, мы приведем V к форме V_1V_2 , где в V_1 нет открывающих скобок, в V_2 нет закрывающих.

Заметим, что $V^n = V_1(V_2V_1)^{n-1}V_2$. Рассмотрим слово V_2V_1 . Применяя к нему такой же процесс (предложение 10.7 к каждой паре скобок), мы в итоге получим слово Q , либо не содержащее скобок вообще, либо содержащее только один их вид.

Пусть, например, Q содержит только открывающие скобки, то Q^7 содержит подслово xQ_1yQ_2 , где x, y – открывающие скобки (буквы, кодирующие ребра в подклейки), а количество узловых букв в Q_2 превышает число узловых букв в Q_1 более чем в пять раз. Таким образом, учитывая предложение 10.9, в этом случае Q^7 можно привести к нулю, а значит и V^8 тоже.

Пусть Q не содержит скобок вообще. В этом случае, все ребра главные. Будем следить, по каким ребрам проходит предполагаемый путь. Допустим, Q содержит узловую букву кодирующую не боковой узел (то есть, узел типа А, В, С, или угловой), тогда путь должен проходить по нескольким ребрам одной макроплитки. Заметим, что нельзя посетить более четырех внутренних ребер макроплитки не повернув ни в одном месте назад и не выйдя на периметр. Но Q^7 должен содержать больше четырех таких участков, значит, в каком-то из них можно будет применить мономиальное соотношение.

Пусть теперь все узловы буквы Q кодируют боковые узлы. То есть предполагаемый путь проходит по одному и тому же ребру некоторой макроплитки. Для такого случая можно применить предложение 10.5. То есть мы применяем несколько преобразований, и если в процессе нельзя было применить мономиальное соотношение, в конце получится представление нашего слова в виде кодировки некоторого пути на комплексе. То есть, наш путь лежит полностью внутри некоторого ребра на комплексе.

Внутри каждого ребра на комплексе нет более трех узлов с типами, отличными от UL и LU. Так как наш путь содержит девять одинаковых участков подряд, в них не может встретиться вершин с типами, отличными от UL и LU. Но тогда в слове не может быть даже двух одинаковых участков подряд (предложение 10.10). Таким образом, W^9 всегда приводится к нулю.

Теперь мы можем завершить доказательство нашей основной теоремы:

Теорема 10.1. *В полугруппе S , заданной конечным числом определяющих соотношений, существует бесконечное число различных слов, не равных нулю. При этом для любого слова W его девятая степень приводится к нулю: $W^9 = 0$.*

Доказательство. Каждому регулярному пути на комплексе можно привести в соответствие его кодировку – слово в полугруппе. При этом, при преобразовании пути его длина не меняется. Таким образом, если некоторый путь приводится к нулю, это значит, что либо в нем нашлось запрещенное подслово, либо локально некратчайший кусок пути, либо мертвый паттерн. Во всех этих случаях путь не является регулярным. То есть, все регулярные пути к нулю привести нельзя.

Вторая часть утверждения следует из предложения 10.11. Таким образом, построенная конечно определенная полугруппа S содержит бесконечное множество различных слов, и

при этом является нильполугруппой, где каждое слово в девятой степени приводится к нулю.

11. ПРИЛОЖЕНИЕ. ПОДСЧЕТ ЧИСЛА ОКРУЖЕНИЙ

В этой главе мы посчитаем число различных четверок-комбинаций типов ребер.

Предложение 11.1 (О нижней стороне макроплитки). Пусть X и Y – вершины, соответствующие левому нижнему и правому нижнему углу макроплитки T . Кроме того, пусть Z – боковая вершина, лежащая на нижней стороне T . Тогда начальники у Z – это пара (X, Y) .

Доказательство. Рассмотрим положение T в ее родительской макроплитке T' . Заметим, что нижняя сторона T в любом случае не попадает на границу T' , то есть обязательно является одним из внутренних ребер в T' . Тогда его концы X, Y являются начальниками для всех боковых вершин лежащих внутри ребра.

Определение. Код пути P будем называть *мертвым*, если существует натуральное N , такое что для любых путей W_1, W_2 длины более N путь W_1PW_2 может быть приведен к нулевой форме.

11.1. Варианты расположения макроплиток. В этой главе мы разберем различные случаи расположения макроплитки на мозаике и определим, какие возможные комбинации плиточной информации (то есть сочетания типов ее граничных ребер) она может иметь.

Определение. Назовем макроплитку X *прямым потомком* для макроплитки Y , если X является левой верхней подплиткой Y , либо является левой верхней подплиткой некоторой макроплитки Z , которая уже является прямым потомком для Y .

Предложение 11.2 (О комбинациях ребер при примыкании к краю). Рассмотрим некоторую макроплитку T с типами граничных ребер $t = \text{Top}(T)$, $l = \text{Left}(T)$, $r = \text{Right}(T)$ и $b = \text{Bottom}(T)$. Пусть также среди типов ребер t и l хотя бы один обозначает краевое ребро (*top, right, bottom, left*). Тогда всевозможные варианты значений t, l, r, b исчерпываются описанными ниже пятью случаями:

1. T – подклеенная макроплитка. $l = \text{left}$, $t = \text{top}$, $r = \text{right}$, $b = \text{bottom}$. То есть, вся четверка (*left, top, right, bottom*).

2. T – непосредственная подплитка подклеенной плитки. В зависимости от ее типа четверка определяется следующим образом:

- (1) левое нижнее положение – (*7A, left, 3B, 4A*)
- (2) левое верхнее положение – (*left, top, 1A, 3A*)
- (3) правое верхнее положение – (*top, right, 6A, 2A*)
- (4) правое нижнее положение – (*right, 8A, 5A, 6B*)
- (5) нижнее положение – (*8B, bottom, bottom, 7B*)

3. T – прямой потомок некоторой макроплитки T' , которая является непосредственной подплиткой подклеенной макроплитки, либо T – правая нижняя подплитка макроплитки, являющейся нижней подплиткой в подклеенной макроплитке, либо прямой потомок такой макроплитки.

- (*7A, left, 1A, 3A*)
- (*top, right, 1A, 3A*)
- (*right(bottom), 8A, 1A, 3A*)

(bottom, 8A, 5A, 6B)
 (8B, bottom, 1A, 3A)

4. T примыкает левой стороной к краю подклеенной макроплитки, то есть $l = left, top, right, bottom$. Тогда четверка (l, t, r, b) может быть следующей:

(l, 3B, 6A, 2A), (right, 6A, 6A, 2A), (bottom, 6A, 6A, 2A),
 (l, 1A, 6A, 2A), (right, 6A, 1A, 3A), (bottom, 6A, 1A, 3A),
 (l, 3B, 1A, 3A), (bottom, bottom, 6A, 2A),
 (l, 1A, 1A, 3A), (bottom, bottom, 1A, 3A)
 где $l = left, top, right, bottom$

5. T примыкает верхней стороной к краю подклеенной макроплитки, то есть $t = left, top, right, bottom$. Тогда четверка (l, t, r, b) может быть следующей:

(7A, t, 3B, 4A), (bottom, bottom, 6A, 2A),
 (7A, t, 1A, 3A), (bottom, bottom, 1A, 3A),
 где $t = left, top, right, bottom$

Общий список возможных сочетаний перечислен в таблице 32.

ТАБЛИЦА 32. Все варианты окружений, включающие примыкание к краю

B1.	(left, top, right, bottom)	
B2.	(left, top, 1A, 3A)	
B3.	(7A, x, 3B, 4A),	$x = left, top, right, bottom$
B4.	(7A, x, 1A, 3A),	$x = left, top, right, bottom$
B5.	(top, right, 6A, 2A)	
B6.	(top, right, 1A, 3A)	
B7.	(x, 8A, 5A, 6B)	$x = right, bottom$
B8.	(x, 8A, 1A, 3A)	$x = right, bottom$
B9.	(8B, bottom, bottom, 7B)	
B10.	(8B, bottom, 1A, 3A)	
B11.	(x, 3B, 6A, 2A),	$x = left, top, right, bottom$
B12.	(x, 3B, 1A, 3A),	$x = left, top, right, bottom$
B13.	(right, 6A, 6A, 2A)	
B14.	(right, 6A, 1A, 3A)	
B15.	(bottom, 6A, 6A, 2A)	
B16.	(bottom, 6A, 1A, 3A)	
B17.	(x, 1A, 6A, 2A),	$x = left, top, right, bottom$
B18.	(x, 1A, 1A, 3A),	$x = left, top, right, bottom$
B19.	(bottom, bottom, 6A, 2A)	
B20.	(bottom, bottom, 1A, 3A)	

Доказательство. Очевидно, что если T сама является подклеенной плиткой, то типы ребер будут как в случае 1. Если T является непосредственной подплиткой подклеенной макроплитки, то возможны пять случаев (кроме средней подплитки, так как ее границы не выходят на край) и типы ребер будут как в случае 2.

Теперь рассмотрим вариант, когда T не является ни подклеенной плиткой, ни непосредственной подплиткой подклеенной плитки. Пусть левый верхний угол T является углом

подклееной макроплитки (но сама T подклееной не является). Тогда возможны два случая. Первый – T является правой нижней подплиткой некоторой плитки, которая является нижней подплиткой в подклееной макроплитке. Тогда четверка (l, t, r, b) принимает значения **bottom, 8A, 5A, 6B**. Вторым – T есть прямой потомок некоторой плитки X , которая является одной из четырех непосредственных подплиток подклееной макроплитки (X не может быть средней подплиткой, так как она не примыкает к краевым сторонам, и левой верхней подплиткой, так как тогда T была бы прямым потомком подклееной плитки). В этом случае пара (l, t) принимает следующие значения, соответствующие четырем положениям X : **(7A, left)**, **(top, right)**, **(right, 8A)**, **(8B, bottom)**. Так как T является прямым потомком X , то правое и нижнее ребро будут иметь типы **(1A, 3A)**. Получаем перечисленные в случае 3 варианты.

ТАБЛИЦА 33. Комбинации ребер, возможные при внутреннем расположении макроплитки

для любого внутреннего типа t	(7A, t, 3B, 4A)
если $t \in \{2A, 3A, 4A, 5B, 6B, 7B\}$ (примыкание нижней стороной)	(8B, $t, t, 7B)$, ($t, t, 6A, 2A)$,
если $t = 1A$	($l, 1A, 6A, 2A)$, (1-любой внутренний тип)
если $t = 1B$	(4B, 1B, 2B, 5B),
если $t = 2B$	(1B, 2B, 6A, 2A),
если $t = 3B$	($l, 3B, 6A, 2A)$, (1-любой внутренний тип)
если $t = 5A$	(8A, 5A, 6A, 2A),
если $t = 6A$	($l, 6A, 6A, 2A)$, (1-любой внутренний, кроме 1B, 4B, 7A, 8A, 8B.)
если $t = 8A$	($l, 8A, 5A, 6B)$, (1-любой внутренний, кроме 1B, 4B, 7A, 8A, 8B.)
если (l, t, r, d) - одна из четверок выше	($l, r, 1A, 3A)$

Итак, теперь пусть теперь левый верхний угол T не является углом подклееной плитки. Тогда одновременно левая и верхняя стороны T примыкать к краям не могут. Пусть сначала T примыкает левой стороной к краю подклееной макроплитки (но не в условиях случаев 1-3). Пусть X это максимальная плитка, прямым потомком которой является наша макроплитка T (либо, если T – не левая верхняя подплитка, то $X = T$). Заметим, что X не может быть подклееной плиткой или одной из шести подплиток подклееной плитки, так как тогда реализовался бы один из предыдущих случаев. Заметим, что X может занимать только правое верхнее положение в своей родительской макроплитке X' , так как в остальных положениях левая сторона не будет примыкать к краю, либо можно было бы выбрать более крупную X , либо левый верхний угол T попадет в угол подклееной плитки. То есть, X' примыкает к краю своей верхней стороной, а верхняя сторона нашей плитки T лежит на правой стороне X' . В свою очередь, X' может занимать левое нижнее, левое верхнее, правое верхнее либо нижнее положение в своей родительской макроплитке. В зависимости от этого положения t принимает значения **3B, 1A, 6A, bottom**. Заметим, что если X' занимает нижнее положение в своей родительской макроплитке, (последний случай) то эта родительская макроплитка должна быть подклееной, так как иначе нижняя сторона не будет выходить на край. Аналогично, если X' занимает правое верхнее положение, то

либо родительская плитка – подклееная, и тогда край – правый, либо родительская плитка – нижняя подплитка подклеенной и тогда край – нижний. Значения (r, b) определяются как $(1A, 3A)$, если $X \neq T$, и $(6A, 2A)$, если $T = X$. Получаем весь список значений из случая 4.

Таблица 34. Все варианты окружений для внутренних расположений

A1.	$(7A, t, 3B, 4A)$	для любого внутреннего типа t
A2.	$(7A, t, 1A, 3A)$	для любого внутреннего типа t
A3.	$(8B, t, t, 7B)$,	где $t \in \{2A, 3A, 4A, 5B, 6B, 7B\}$
A4.	$(8B, t, 1A, 3A)$,	где $t \in \{2A, 3A, 4A, 5B, 6B, 7B\}$
A5.	$(t, t, 6A, 2A)$,	где $t \in \{2A, 3A, 4A, 5B, 6B, 7B\}$
A6.	$(t, t, 1A, 3A)$,	где $t \in \{2A, 3A, 4A, 5B, 6B, 7B\}$
A7.	$(l, 1A, 6A, 2A)$,	$(l$ -любой внутренний тип)
A8.	$(l, 1A, 1A, 3A)$,	$(l$ -любой внутренний тип)
A9.	$(4B, 1B, 2B, 5B)$,	
A10.	$(4B, 1B, 1A, 3A)$,	
A11.	$(1B, 2B, 6A, 2A)$,	
A12.	$(1B, 2B, 1A, 3A)$,	
A13.	$(l, 3B, 6A, 2A)$,	$(l$ -любой внутренний тип)
A14.	$(l, 3B, 1A, 3A)$,	$(l$ -любой внутренний тип)
A15.	$(8A, 5A, 6A, 2A)$,	
A16.	$(8A, 5A, 1A, 3A)$,	
A17.	$(l, 6A, 6A, 2A)$,	$(l$ -любой внутренний, кроме $1B, 4B, 7A, 8A, 8B$.)
A18.	$(l, 6A, 1A, 3A)$,	$(l$ -любой внутренний, кроме $1B, 4B, 7A, 8A, 8B$.)
A19.	$(l, 8A, 5A, 6B)$,	$(l$ -любой внутренний, кроме $1B, 4B, 7A, 8A, 8B$.)
A20.	$(l, 8A, 1A, 3A)$,	$(l$ -любой внутренний, кроме $1B, 4B, 7A, 8A, 8B$.)

Итак **остался случай**, когда T примыкает верхней стороной к краю. Пусть X это максимальная плитка, прямым потомком которой является наша макроплитка T (либо, если T – не левая верхняя подплитка, то $X = T$). Заметим, что X может занимать левое нижнее либо правое верхнее положение в своей родительской макроплитке X' , так как в остальных положениях левая сторона не будет примыкать к краю, либо можно было бы выбрать более крупную X , либо левый верхний угол T попадет в угол подклеенной плитки. В случае правого верхнего положения, X' должна быть нижней подплиткой подклеенной плитки, иначе левый верхний угол T попадет в угол подклеенной плитки. Но тогда левая сторона T тоже попадет на край, и это будет один из уже описанных случаев: **(bottom, bottom, 1A, 3A)** и **(bottom, bottom, 1A, 3A)**. Итак, остается вариант, когда X является левой нижней подплиткой X' . Тогда $l = 7A$, а пара r, b определяется опять в зависимости от того, совпадает ли T с X : либо $(3B, 4A)$ если совпадает, либо $(1A, 3A)$ если нет. Получаем перечисленные варианты в случае 5.

Предложение 11.3. О комбинациях ребер при внутреннем расположении. Рассмотрим некоторую макроплитку T с типами граничных ребер $t = Top(T)$, $l = Left(T)$, $r = Right(T)$ и $b = Bottom(T)$, причем типы l и t ребер – внутренние. Тогда возможны комбинации типов ребер перечисленные в таблице 33.

Общий список возможных сочетаний перечислен в таблице 34.

Доказательство. Плитка T может занимать одно из 6 возможных мест в родительской макроплитке. В четырех из них можно сразу определить сочетание ребер и случай, к которому относится данная ситуация (таблица 35).

ТАБЛИЦА 35. все варианты окружений, включающие примыкание к краю

положение T	комбинация ребер		итоговый случай
левое нижнее	$(7A, x, 3B, 4A)$	для любого внутреннего типа x	A1
правое нижнее	$(x, 8A, 5A, 6B)$	x -правая сторона (любая, кроме 1B, 4B, 7A, 8A, 8B)	A19
среднее	$(4B, 1B, 2B, 5B)$		A9
нижнее	$(8B, x, x, 7B)$	x -нижняя сторона, (то есть 2A, 3A, 4A, 5B, 6B, 7B)	A3

ТАБЛИЦА 36. все варианты окружений, включающие примыкание к краю

положение левого верхнего угла	комбинация ребер		
середина верхней стороны	$(x, 1A, 6A, 2A)$ $(1B, 2B, 6A, 2A)$	x – верхняя сторона, то есть любой тип	A7 A11
середина левой стороны	$(x, 3B, 6A, 2A)$	x – левая сторона, то есть любой тип	A13
середина правой стороны	$(x, 6A, 6A, 2A)$	x – правая сторона, (любой, кроме 1B, 4B, 7A, 8A, 8B)	A17
внутренний C-узел	$(8A, 5A, 6A, 2A)$		A15
середина нижней стороны	$(x, x, 6A, 2A)$	x – нижняя сторона, то есть 2A, 3A, 4A, 5B, 6B, 7B	A5

Пусть T занимает правое верхнее положение. Тогда возможны ситуации, указанные в таблице 36.

Если T занимает левое верхнее положение, рассмотрим максимального макроплитку T' , являющуюся прямым предком T (то есть такую, что из T' переходами к левой верхней подплитке можно за несколько операций получить T). Заметим, что в T' левая и верхняя стороны такие же как в T , и при этом у T' есть родительская макроплитка, в которой она занимает одно из пяти положений (кроме левого верхнего).

Таким образом, тип левого и верхнего ребер T' будет как в одном из случаев рассмотренных выше, и общая четверка для T будет состоять из этих двух ребер плюс 1A и 3A для правого и нижнего ребер. То есть получим один из четных случаев выписанных выше.

12. ДАЛЬНЕЙШИЕ ВОПРОСЫ И СООБРАЖЕНИЯ

Методы данной работы, возможно, помогут построить и иные объекты. В частности, представляет интерес следующий

Вопрос. Существует ли конечно определенная полугруппа с размерностью Гельфанда–Кириллова 2.5.

Л. Н. Шеврин поставил следующие вопросы:

Вопрос 1. *Каково наименьшее возможное число элементов базиса бесконечной конечно определенной нильполугруппы конечного индекса?*

Вопрос 2. *Каков наименьший возможный индекс бесконечной конечно определенной нильполугруппы конечного индекса?*

Можно ли опустить в формулировке обоих вопросов слова «конечного индекса»? А именно, эти слова нужно будет опустить, если будет дан положительный ответ на следующий вопрос:

Вопрос 3. *Любая ли бесконечная конечно определенная полугруппа имеет конечный индекс?*

Нам представляются чрезвычайно важным для дальнейшего обобщение результата Х. Гудмана-Штраусса. В частности, представляют интерес два вопроса:

А. Топологическое обобщение результата Х. Гудмана-Штраусса. Мозаика рассматривается как плоский граф, где плитками являются простые циклы. Нужно доказать, что заданием конечного числа локальных правил можно добиться заданного иерархического правила построения.

В. Обобщение иерархического правила. Нужно доказать, что заданием конечного числа локальных правил можно добиться заданного рекурсивного правила построения мозаики.

Последнее нам представляется важным как в связи с третьим вопросом Л. Н. Шеврина (мы предполагаем отрицательный ответ), так и для построения иных рекурсивных объектов. По всей видимости, эти вопросы должны предшествовать теоретико-кольцевым исследованиям.

И, наконец, возникает вопрос, на который обратил наше внимание Л. Н. Шеврин: *какие максимальные ненулевые степени элементов возможны в построенной нами полугруппе S ?*

Все перечисленные вопросы представляются полезными для получения идей, позволяющих работать в кольцевой тематике.

СПИСОК ЛИТЕРАТУРЫ

- [1] Адян С. И. *Проблема Бернсайда и связанные с ней вопросы*, УМН, 65:5(395) (2010), 5–60
- [2] Адян С. И. *Новые оценки нечетных периодов бесконечных бернсайдовских групп*. Труды Математического института им. В. А. Стеклова, 2015, т. 289, 41–82
- [3] Адян С. И. *Проблема Бернсайда и тождества в группах*, Наука, М., 1975, 335 с.
- [4] Белов А. Я. *Линейные рекуррентные уравнения на дереве*. Математические заметки, 78, N5, 643–651.
- [5] Белов А. Я. *Проблемы бернсайдовского типа, теоремы о высоте и о независимости*, Фундамент. и прикл. матем., 13:5 (2007), 19–79 А. Ya. Belov, *Burnside-type problems, theorems on height, and independence*, J. Math. Sci., NY 156:2 (2009), 219–260
- [6] Белов А. Я., Харитонов М. И. *Субэкспоненциальные оценки в теореме Ширшова о высоте*, Матем. сб., 203:4 (2012), 81–102 А. Ya. Belov, M. I. Kharitonov, *Subexponential estimates in Shirshov's theorem on height*, Sb. Math., 203:4 (2012), 534–553 .
- [7] А. Я. Белов-Канель, И. А. Иванов-Погодаев, А. Малистов, И. Митрофанов, М. Харитонов, *Замощения, раскраски и плиточные группы*, <http://www.turgor.ru/lktg/2009/4/index.php>, 21-я летняя конференция международного математического Турнира городов, 2009 , Российская академия наук, Департамент образования города Москвы, Московский центр непрерывного математического образования, Карачаево-Черкесский государственный университет им. У. Д. Алиева, Журнал “Квант”, Теберда, Карачаево-Черкессия, 02.08.2009–10.08.2009.
- [8] Е. С. Голод *О ниль-алгебрах и финитно-аппроксимлируемых p -группах*. Изв. АН СССР. Сер. матем., 28:2 (1964), 273–276

- [9] Е. С. Голод, И. Р. Шафаревич *О башне полей классов*. Изв. АН СССР. Сер. матем., 28:2 (1964), 261–272
- [10] *Днестровская тетрадь: оперативно-информац. сборник* — 4-е изд. — Новосибирск: изд. ин-та матем. СО АН СССР, 1993, 73.
- [11] Иванов-Погодаев И. А., *Алгебра с конечным базисом Грёбнера и неразрешимой проблемой делителей нуля*, *Фундамент. и прикл. матем.*, 12:8 (2006), 79–96
- [12] Иванов-Погодаев И. А. *Машина Минского, свойства нильпотентности и размерность Гельфанда-Кириллова в конечно определенных полугруппах*. Дисс. на соиск. уч. ст. канд. физ. мат. наук — Москва, 2006, 77с.
- [13] T. Fernique, Иванов-Погодаев И. А., Белов А. Я, И. В. Митрофанов. *Аперiodические замощения*, <http://www.turgor.ru/lktg/2013/2/index.htm>, 2013, Российская академия наук, Департамент образования города Москвы, Московский центр непрерывного математического образования, Журнал “Квант”, 25-я летняя конференция международного математического Турнира городов Боровка, Беларусь / 04.08.2013–12.08.2013; 25 summer conference International mathematical Tournament of towns Boroovka, Belarus, 04.08.2013–12.08.2013 Aperiodic tilings. [http : //lipn.univ – paris13.fr/ ~ fernique/info/turgorod_rus2.pdf](http://lipn.univ-paris13.fr/~fernique/info/turgorod_rus2.pdf)
- [14] Игуду Н. К. *Алгоритмическая разрешимость проблемы распознавания делителей нуля в одном классе алгебр*. *Фунд. и прикл. матем.*, 1995, 2, 1, 541–544.
- [15] Игуду Н. К. *Стандартные базисы и распознаваемость свойств алгебр, заданных копредставлением*. Дисс. на соиск. уч. ст. канд. физ. мат. наук — Москва, 1996, 73.
- [16] Кострикин А. И. *Вокруг Бернсайда*. — М.: Наука, 1986, 232.
- [17] Лысёнок И. Г. *Бесконечность бернсайдовых групп периода 2^k при $k > 13$* , *УМН*, 47:2 (1992), 201–202; I. G. Lysenok *The infinitude of Burnside groups of period 2^k for $k > 13$* , *Russian Math. Surveys*, 47:2 (1992), 229–230.
- [18] Лысёнок И. Г. *Бесконечные бернсайдовы группы четного периода*, *Изв. РАН. Сер. матем.*, 60:3 (1996), 3–224; англ. пер.: I. G. Lysenok, *Infinite Burnside groups of even exponent*, *Izv. Math.*, 60:3 (1996), 453–654.
- [19] Новиков П. С., Адян С. И. *О бесконечных периодических группах.*, (I–III), *Изв. АН СССР. Сер. матем.*, 32 (1968)
- [20] Ольшанский А. Ю. *О теореме Новикова–Адяна*, *Матем. сб.*, 118(160):2(6) (1982), 203–235
- [21] Ольшанский А. Ю. *Геометрия определяющих соотношений в группах.*, М.: Наука, 1989.
- [22] Пионтковский Д. И. *Базис Грёбнера и когерентность мономиальной ассоциативной алгебры.*, *Фунд. и прикл. матем.*, 1996, 2, 2, 501–509.
- [23] Пионтковский Д. И. *Некоммутативные базисы Гребнера, когерентность ассоциативных алгебр и делимость в полугруппах.*, *Фунд. и прикл. матем.*, 2001, 7, 2, 495–513.
- [24] Санов И. Н., *Решение проблемы Бернсайда для показателя 4*, *Уч. зап. Ленингр. ун-та, сер. матем.*, 10 (1940), 166–170.
- [25] *Свердловская тетрадь: Нерешённые задачи теории полугрупп. Выпуск третий*, Свердловск, 1989. — 40 с
- [26] Уфнарковский В. А. *Комбинаторные и асимптотические методы в алгебре.*, *Итоги науки и техн. Сер. Современ. пробл. мат. Фундам. направления.* М.: ВИНТИ, 1990, 57, 5–177.
- [27] Уфнарковский В. А. *О росте алгебр*. *Вестник МГУ.* вып 1, 1978, 4, 59–65.
- [28] Л. М. Глускин, Б. М. Шайн, Л. Н. Шеврин. *Полугруппы*. *Итоги науки. Сер. Мат. Алгебра. Топол. Геом.* 1966, 1968, 9–56
- [29] Ширшов А. И. *О некоторых неассоциативных ниль-кольцах и алгебраических алгебрах.*, *Мат. сб.*, 1957, 41, 3, 381–394. (*РЖМат*, 1958, 164)
- [30] Ширшов А. И. *О кольцах с тождественными соотношениями.*, *Мат. сб.*, 1957, 43, 2, 277–283. (*РЖМат*, 1958, 7544)
- [31] Щиголев В. В., *О ниль и нильпотентных конечноопределённых алгебрах*, *Фунд. и прикл. матем.*, 2000, т. 6, выпуск 4, стр. 1239–1245.
- [32] N.Bédaride, Th.Fernique, *When periodicities enforce aperiodicity?*, *Communications in Mathematical Physics* 335 (2015), pp. 1099–1120.
- [33] Kanel-Belov A., Rowen Louis H. *Perspectives on Shirshov’s Height Theorem*. in book: selected papers of A.I.Shirshov, Birkhäuser Verlag AG (2009), 3–20.
- [34] Belov A. J., Borisenko V. V., Latysev V. N. *Monomial Algebras*. NY. Plenum, 1997.
- [35] Belov A. Ya. Ivanov I. A. *Construction of Semigroups with Some Exotic Properties*, *Comm. in Algebra*, Volume 31, Num 2, 2003. 673–696.

- [36] Belov A. Ya. Ivanov I. A. *Construction of Semigroups with Some Exotic Properties*, Acta Appl. Math 85 (2005), no 1–3, 49–56.
- [37] Belyaev V. Ya. *Imbeddability of recursively defined inverse semigroups in finitely presented semigroups*. Sibirsk. Math. Journal 25 no. 2., 1984. 50–54.
- [38] R. Berger, *The undecidability of the domino problem*, Ph.D. thesis, Harvard University, July 1964.
- [39] Bergman G. *The Diamond Lemma for ring theory*. Adv. Math., 1978, 29, 2, 178–218.
- [40] Burnside W., *On an unsettled question in the theory of discontinuous groups*, Q. J. Pure Appl. Math., 33 (1902), 230–238
- [41] Bokut L. A., Kukin G. P. *Algorithmic and combinatorial algebra*. Math. and its appl. 233.
- [42] Conway J., Lagarias J.. *Tiling with polyminoes and combinatorial group theory*./ J. Combin. Theory Ser. A. 1990. V. 53, 2, 183–208.
- [43] Ivanov S. V., *The free Burnside groups of sufficiently large exponents*, Internat. J. Algebra Comput., 4:1–2 (1994), 1–308.
- [44] Ivanov S. V., “On the Burnside problem on periodic groups”, Bul. Amer. Math. Soc. (N. S.), 27:2 (1992), 257–260; arXiv: math/9210221.
- [45] Gelfand I. M., Kirillov A. A. *Sur les corps liés aux algèbres enveloppantes des algèbres de Lie*. Publ. math. Inst. hautes études sci., 1966, 31, 509–523.
- [46] C. Goodman-Strauss, *Matching rules and substitution tilings*, Ann. of Math. (2) 147 (1998), no. 1, 181–223.
- [47] Grunbaum B., Shephard G. C. *Tilings and patterns*, Freeman, NY 1986.
- [48] Hall M.Jr., *Solution of the Burnside problem for exponent 6*, Proc. Nat. Acad. Sci. USA, 43 (1957), 751–753.
- [49] Ivanov I., *Construction of Semigroups with Some Exotic Properties*, // Proceedings of the 65th Workshop on General Algebra, Potsdam University, Germany
- [50] Kemer, A. R., *Comments on the Shirshov’s Height Theorem*. in book: selected papers of A.I.Shirshov, Birkhäuser Verlag AG (2009), 41–48.
- [51] Krause G. R., Lenagan T. H. *Growth of algebras and Gelfand-Kirillov dimension*. London: Pitman Adv. Publ. Program, 1985, 182.
- [52] Kukin G. P. *The variety of all rings has Higman’s property*. Algebra and Analysis. Irkutsk. 1989 91–101
- [53] Latyshev V. *On the recognizable properties of associative algebras*. Special vol. J.S.C.: On computational aspects of commutative algebras. London: Acad. Press, 1988, 237–254.
- [54] Kharlampovich O. G., Sapir M. V.. *Algorithmic problems in varieties*. International Journal of Algebra and Computation, Vol. 5, Nos. 4–5 (1995), 379–602
- [55] Ol’shanskii Alexander Yu., Sapir Mark V. *Non-amenable finitely presented torsion-by-cyclic groups*. Publications Mathématiques de l’Institut des Hautes Etudes Scientifiques May 2003, Volume 96, Issue 1, pp 43–169.
- [56] Robinson R. M., *Undecidability and nonperiodicity for tilings of the plane*, Inventiones Mathematicae 12 (1971), 177–209.
- [57] Sapir M. V. *Algorithmic Problems for Amalgams of Finite Semigroups*.
- [58] Mark V. Sapir with contributions by Victor S. Guba and Mikhail V. Volkov *Combinatorial algebra: syntax and semantics* Springer, 2014, 355pp.
- [59] Wang Hao. *Proving theorems by pattern recognition—II*, Bell System Tech. Journal 40(1):1–41, 1961
- [60] Zelmanov E. I. *On the nilpotency of nilalgebras*. Lect. Notes Math., 1988, 1352, 227–240. PЖМат, 1989, 7A188)

- [61] Belov-Kanel Alexey, Ivanov-Pogodaev Ilya, *Construction of infinite finitely presented nilsemigroup*, First Joint International Meeting of the Israel Mathematical Union and the Mexican Mathematical Society (Oaxaca, Mexico, September 7 to 11, 2015.), Instituto Tecnológico de Oaxaca, 2015, 13 (Published online).
- [62] Belov-Kanel Alexey, Ivanov-Pogodaev Ilya *Construction of infinite finitely presented nilsemigroup*, Supported by Grant RFBR N 14-01-00548, Groups and Rings, Theory and Applications, GRiTA2015 (July 15–22, 2015 – Sofia, BULGARIA), Institute of Mathematics and Informatics at the Bulgarian Academy of Sciences, Sofia, 2015, 10.
- [63] Kanel-Belov Alexey, Ivanov-Pogodaev Ilya, *Construction of infinite finitely presented nilsemigroup*, Growth, Symbolic Dynamics and Combinatorics of Words in Groups, Ecole normale supérieure, Paris, 1 - 5 June 2015.

- [64] Канель-Белов А. Я. *О существовании бесконечных конечно определённых ниль-полугрупп (По совместной работе с И.А. Ивановым-Погодаевым)*, Алгоритмические вопросы алгебры и логики, 14 апреля 2015 г. 18:30, г. Москва, ГЗ МГУ, ауд. 16-04
- [65] Белов А. Я., Иванов-Погодаев И. А. *Конечно-определённые ниль-полугруппы, иные объекты и непериодические мозаики*, Дискретная и вычислительная геометрия 23 сентября 2014 г. 13:45, г. Москва, ИПИ РАН, Большой Каретный переулок, 19, ауд. 307
- [66] Иванов-Погодаев И. А., Белов А. Я., *Построение конечно определённой ниль-полугруппы*, Алгебра и теория чисел: современные проблемы и приложения (Тула, 21–25 апреля 2014), Материалы XII межд. конф., посв. 80-летию проф. В. Н. Латышева, Тул. гос. пед. ун-т им. Л. Н. Толстого, Тула, 2014, 30.
- [67] Belov Alexei Ya., Ivanov-Pogodaev Ilya A., *Construction of finitely presented infinite nil-semigroup*, Classical Aspects of Ring Theory and Module Theory, Abstracts (Bedlewo, Poland, July 14–20, 2013), Stefan Banach International Mathematical Center, 2013, 14.
- [68] Belov-Kanel Alexey, Ivanov-Pogodaev Ilya, *Finitely presented nil-semigroups and aperiodic tilings*, CIRM, Marseille, 2013, SubTile 2013 Pavages : systemes dynamiques, combinatoire, theorie des nombres, decidabilite, geometrie discrete, geometrie non-commutative. 14-18 janv. 2013 Marseille (France)
- [69] Иванов-Погодаев И. А., Канель-Белов А. Я. *Конечно-определённые нильполугруппы и непериодические мозаики*, Межкафедральный семинар МФТИ по дискретной математике 17 октября 2012г.
- [70] Ivanov-Pogodayev I., Kanel-Belov A., *Construction of finitely presented infinite nil-semigroups.*, Special session dedicated to prof. B.Plotkin, International Conference “Modern Algebra and Its Applications”(Batumi, September 2010), Journal of Mathematical Sciences, 186, no. 5, 2012, 751–752 (Translated from *Sovremennaya Matematika i Ee Prilozheniya (Contemporary Mathematics and Its Applications)*, Vol. 74, (Batumi, 2010), Part 1, 2011, 84–86., Georgian Science foundation)
- [71] Belov Alexey, Ivanov-Pogodaev Ilya, *Construction of finitely presented Nil but not nilpotent semigroup*, International Conference on Algebra and Related Topics (ICCA) (23–27 June 2009, Guangzhou), The Center of Combinatorial Algebra, South China Normal University, 2009.