

Детерминированная раскраска семейства комплексов

И. А. ИВАНОВ-ПОГОДАЕВ

Московский физико-технический институт
e-mail: ivanov.pogodaev@mail.ru

А. Я. КАНЕЛЬ-БЕЛОВ

*Университет имени Бар-Илана, Израиль,
Московский государственный университет
им. М. В. Ломоносова*
e-mail: kanel@mccme.ru

УДК 512.53

Ключевые слова: детерминированность, апериодические замощения, конечно определённые полугруппы, проблемы бернсайдовского типа.

Аннотация

В знаменитых работах Ш. Мозеса «Tilings, substitution systems and dynamical systems generated by them» и Х. Гудмана-Штраусса «Matching rules and substitution tilings» исследуются важные свойства замощений на плоскости, связывающие язык подстановочных систем и локальных правил. В частности, доказываем, что практически с любой подстановочной системой плиток можно связать систему декораций и локальных правил на границы плиток так, что любое разрешённое правилами замощение плоскости будет принадлежать семейству, порождённому данной подстановочной системой.

Геометрические построения на основе замощений и геометрических комплексов из склеенных квадратов открывают новые возможности для исследования и конструирования алгебраических объектов. Важным свойством замощений, позволяющим связывать их с алгебраическими объектами, является детерминированность. Для квадратных плиток, стороны которых покрашены в конечное число цветов, это свойство означает, что цвета двух соседних сторон каждой плитки однозначно определяют цвета другой пары сторон. Дж. Кари и П. Папасоглу построили детерминированный апериодический набор. Таким образом, аналог теорем Гудмана-Штраусса и Мозеса с дополнительным условием детерминированности является важным утверждением в теории апериодических мозаик. Мы доказываем аналог этого утверждения для конкретной фиксированной подстановочной системы. Но её произвольность позволяет надеяться на возможность обобщения построения на общий случай.

Данная система важна при построении бесконечной конечно определённой нильполугруппы, решающей проблему Л. Н. Шеврина и М. В. Сапира. Работа посвящена также построению семейства геометрических комплексов и введению на них кодировки вершин и рёбер. Для полученного комплекса выполнены одновременно оба свойства: конечность набора задающих локальных правил и детерминированность на путях длины 2 на минимальных квадратах.

Данная работа является второй из трёх, составляющих цикл, посвящённый построению бесконечной конечно определённой нильполугруппы. Результаты первой работы цикла приведены в начале для удобства.

Abstract

I. A. Ivanov-Pogodaev, A. Ya. Kanel-Belov, Deterministic coloring of a family of complexes, *Fundamentalnaya i prikladnaya matematika*, vol. 24 (2022), no. 2, pp. 37–180.

In the famous works “Tilings, substitution systems and dynamical systems generated by them” by S. Mozes and “Matching rules and substitution tilings” by Ch. Goodman-Strauss, important properties of tiling on the plane connecting the language of substitution systems and local rules are investigated. In particular, it is proved that with almost any substitution system of tiles, it is possible to link a system of decorations and local rules to the boundaries of tiles in such a way that any tiling of the plane allowed by the rules will belong to the family generated by this substitution system.

Geometric constructions based on tiling and geometric complexes of glued squares open up new opportunities for the study and construction of algebraic objects. An important property of tiling that allows them to be associated with algebraic objects is determinicity. For square tiles whose sides are painted in a finite number of colors, this property means that the colors of the two adjacent sides of each tile uniquely determine the colors of the other pair of sides. J. Kari and P. Papasoglu constructed a deterministic aperiodic set. Thus, the analogue of the theorems of Goodman-Strauss and Mozes with an additional determinicity condition is an important statement in the theory of aperiodic mosaics. We prove an analog of this statement for a specific fixed substitution system. But its arbitrariness allows us to hope for the possibility of improving the construction for the general case.

This system is important when constructing an infinite finitely presented nilsemigroup, which solves the problem of L. N. Shevrin and M. V. Sapir. The work is also devoted to the construction of a family of geometric complexes and the introduction of coding of vertices and edges on them. For the resulting complex, both properties are fulfilled simultaneously: finiteness of the set of defining local rules and determinicity on paths of length 2 on minimal squares.

1. Введение

В 1961 году Хао Ван [11] рассматривал квадратные плитки с разноцветными сторонами. Разные плитки разрешалось прикладывать друг к другу сторонами одного цвета. Был поставлен вопрос, существуют ли конечные наборы таких плиток, с помощью которых могут быть получены только непериодические замощения плоскости. Первый такой набор был построен Р. Бергером [2]; 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 Р. Робинсона [10]. Таким образом, заданием конечного числа запрещённых примыканий плиток друг к другу (локальных условий) можно обеспечить глобальное свойство (апериодичность).

Одним из основных приёмов для построения апериодических замощений является самоподобие. Для каждой плитки (многоугольника) конечного набора задаётся разбиение на меньшие плитки, являющиеся меньшими подобными копиями из того же набора. Итерация таких разбиений приводит (для нетривиальной подстановки) к апериодическим замощениям.

Широко известны результаты Ш. Мозеса [9] и Х. Гудмана-Штраусса [5], связывающие язык примыканий и язык подстановок. Х. Гудман-Штраусс показал, что для любой достаточно произвольной подстановочной системы можно задать набор декораций, локальных правил, так, что все замощения, удовлетворяющие заданным правилам, будут получаться итерациями заданной подстановочной системы.

Рассмотрим самый простой вид замощений квадратными плитками и будем задавать локальные правила с помощью цветов на рёбрах. Можно рассматривать наборы плиток, каждая из которых покрашена заданным образом. Как известно из работ Р. Бергера и Р. Робинсона, существуют аperiodические наборы, для которых возможны лишь аperiodические замощения. Один из самых простых способов построения такого набора указан в [4]. Важным свойством набора является *детерминированность* — когда цвета двух соседних сторон квадрата однозначно определяют цвета двух других сторон. Это свойство является важнейшим для определения алгебраических структур, связанных с набором плиток и замощениями. Дж. Кари и П. Папасоглу [8] построили детерминированный аperiodический набор. Особый интерес к построению аperiodических наборов и свойству детерминированности объясняется также тем, что если набор обладает также свойством полноты (когда каждую плитку можно переворачивать и поворачивать), то с таким набором можно связать конечно определённую группу. Свойство детерминированности набора влечёт свойство малого сокращения $S(4)$ - $T(4)$ в полученной группе. Отсутствие периодических замощений обеспечивает отсутствие в группе двух коммутирующих элементов бесконечного порядка. А наличие аperiodических замощений влечёт негиперболичность данной группы. Вопрос о существовании такой группы поставил М. Громов [6].

С замощениями можно связывать не только группы, но и полугруппы. Для этого удобно отказаться от рассмотрения плоских мозаик и перейти к комплексам. Обычно имеет смысл рассматривать комплексы, склеенные из квадратов или кубов разных размерностей. Данный подход для теории групп был хорошо развит в работах Д. Вайса [12, 13]. Также идея связи алгебраических структур с замощениями содержалась в [3].

В [1] мы построили семейство (последовательность) комплексов, каждый из которых склеен из нескольких квадратов (плиток). Способ построения сходен с самоподобными замощениями. Комплекс уровня n строится на базе комплексов уровня $n - 1$ с помощью разбиения каждой минимальной плитки заданным способом. В настоящей работе для построенной последовательности комплексов вводится конечное множество цветов для вершин и рёбер, а также задаётся способ раскраски всех комплексов. При этом цвета рёбер и вершин вдоль пути по двум соседним вершинам любого минимального квадрата любого комплекса однозначно определяют цвета вершин и рёбер вдоль пути по двум другим сторонам. Это аналог детерминированности на языке путей. Фактически после раскраски комплексов каждому пути отвечает некоторая кодировка — последовательность букв в заданном конечном алфавите. Множество таких слов

естественно рассматривать как полугруппу. Задание конечного множества слов, кодирующих невозможные пути, является аналогом локальных правил на языке путей. Данная конструкция обладает и свойством детерминированности, и свойством задаваемости локальными правилами, хоть и для заданной, конкретной подстановки.

Основным результатом работы является следующий результат.

Теорема 1.1 (о введении раскраски со свойством детерминированности). *Введённая система цветов для рёбер и вершин семейства комплексов обладает свойством детерминированности.*

Пусть слово W является кодировкой пути P на некотором комплексе, причём путь проходит по двум сторонам минимального квадрата K и содержит цвета трёх вершин, двух выходящих рёбер и двух входящих рёбер. Кроме того, пусть путь P не является мёртвым паттерном. Тогда однозначно определяется кодировка пути Q , начало и конец которого совпадают с P и который проходит по двум другим сторонам K .

Эквивалентная формулировка: если A, B, C — вершины некоторого минимального 4-цикла $ABCD$ на комплексе и при этом путь ABC не образует мёртвый паттерн, то по кодировке пути ABC можно однозначно определить кодировку пути ADC .

Данную теорему можно рассматривать как продвижение в получении усиления теорем Ш. Мозеса и Х. Гудмана-Штраусса с дополнительным условием детерминированности правил примыкания. Нам представляется, что построенное семейство комплексов достаточно произвольно, чтобы надеяться, что тот же подход будет эффективен в общем случае.

Интерес к такой конструкции объясняется тем, что при задании некоторых дополнительных определяющих соотношений (запрещающих пути туда и обратно по некоторому ребру) задаваемая полугруппа путей на построенном семействе комплексов будет являться конечно определённой бесконечной нильполугруппой с тождеством $x^9 = 0$, которая решает проблему, поставленную Л. Н. Шевриным и М. В. Сапиром. Окончательные детали этого построения, в частности доказательство того, что слово, содержащее девятую степень, приводится к нулю, составляет содержание третьей работы цикла.

1.1. О проблеме Латышева

Построение конечно определённой нильполугруппы предшествует решению аналогичной задачи для колец, поставленной В. Н. Латышевым: существует ли конечно определённое бесконечномерное нилькольцо?

В полугрупповом случае можно рассмотреть ослабленный вопрос. Построение существенно упрощается, если допустить существование в словах фиксированной точки — буквы, которая гарантированно присутствует в слове. Фиксированная точка при этом служит знаком принадлежности к идеалу, т. е. речь идёт

о конечно определённой полугруппе с бесконечным идеалом, элементы которого не содержат квадратов. Такая конструкция была получена в [7].

Теорема 1.2 (об идеале, не содержащем квадратов). *Существует конечно определённая полугруппа с нулём H , включающая четыре буквы (a_1, a_2, a_3 и L), в которой выполнены следующие свойства:*

- 1) *множество $L\{a_1, a_2, a_3\}^*\cup\{0\}$ является бесконечным идеалом H с нулевым умножением;*
- 2) *все слова, не содержащие квадратов в $\{a_1, a_2, a_3\}^+$, соответствуют различным элементам идеала, причём $L\mathbf{U} = 0$ для любого слова $\mathbf{U} \in \{a_1, a_2, a_3\}^+$, содержащего квадрат.*

Существование фиксированной буквы в каждом слове даёт возможность использовать автоматные методы (вместо построения семейства комплексов), и доказательство упрощается и занимает 7 страниц вместо 160.

Рассматривая проблему Латышева о нилькольце, можно, по аналогии с полугруппами, поставить промежуточный вопрос: существует ли конечно определённое кольцо с нулём R , содержащее букву L , такое что в идеале LR бесконечно много элементов и для любого элемента кольца X существует такое n , что $LR^n = 0$?

В свою очередь, проблема Латышева предшествует конечно определённой проблеме Бернсайда: существуют ли конечно определённые бесконечные периодические группы?

Авторы надеются, что предложенный геометрический подход будет также полезен (по всей видимости, для будущих поколений исследователей) при решении поставленной В. Н. Латышевым проблемы построения конечно определённого нилькольца.

Работа была проведена с помощью Российского научного фонда, грант 22-11-00177. Первый автор является победителем конкурса «Молодая математика России».

Авторы признательны руководителям семинара «Теория колец» на кафедре высшей алгебры механико-математического факультета МГУ В. Н. Латышеву и А. В. Михалёву за полезные обсуждения и постоянное, в течение ряда лет, внимание к работе. Мы также благодарны И. А. Рипсу, Л. Н. Шеврину, А. Х. Шеню, Н. К. Верещагину, А. Эршлер за полезные обсуждения, Ф. Дюрранду, К. Селле, Л. А. Бокутю, Ю. Чэну, Т. Фернику за поддержку участия в конференциях, А. С. Малистову за помощь в оформлении статьи.

2. План построения

Построение состоит из трёх этапов. На первом этапе определяется последовательность геометрических комплексов, представляющих собой несколько склеенных между собой 4-циклов (квадратов). На втором этапе вершины и рёбра комплексов кодируются буквами конечного алфавита. Словам полугруппы

будут соответствовать кодировки путей на комплексах. Определяющим соотношениям отвечают локальные эквивалентности двух путей длины 2, соединяющих противоположные стороны каждого квадрата комплекса. Кроме того, вводятся мономиальные соотношения, приводящие к нулю кодировки невозможных путей, а также путей туда и обратно по некоторому ребру комплекса. На третьем этапе строится алгоритм, приводящий произвольное слово к канонической форме. При этом помогает геометрический смысл введённых соотношений. Каждое слово локально представляет собой кодировку пути на комплексе. При этом применение определяющих соотношений приводит к локальному преобразованию этого пути, не меняющего его концов и его длину. В процессе локальных преобразований в слове может появиться запрещённый участок, в этом случае оно приводится к нулю. Иначе в итоге мы получаем вложение данного пути в некоторый комплекс.

От комплекса нужны следующие свойства:

- 1) равномерная эллиптичность. Любой путь можно достаточно сильно шевелить локальными преобразованиями;
- 2) локальная конечность. Всё семейство комплексов допускает раскраску вершин и рёбер в конечное множество цветов (букв);
- 3) детерминированность. Если известны цвета рёбер и вершин вдоль пути, соединяющего противоположные вершины некоторого квадрата, то однозначно вычисляются и цвета рёбер и вершин вдоль парного к нему пути (по другим двум сторонам). Это свойство позволяет корректно ввести определяющие соотношения;
- 4) аперiodичность. Кодировки путей на комплексе не содержат периодических слов периода 9.

В итоге ненулевым словам в полугруппе соответствуют кратчайшие пути на комплексах. Кодировки, соответствующие не кратчайшим и невозможным путям, приводятся к нулю, при этом остаётся бесконечное множество неэквивалентных ненулевых путей.

3. Содержание первой работы

В первой части цикла работ «Комплексы со свойством равномерной эллиптичности» была изложена геометрическая структура семейства комплексов, служащих базой для введения определяющих соотношений. Ниже кратко излагаются основные положения первой части работы. Точные определения и доказательства можно найти в [1].

Основной конструкцией работы является семейство геометрических комплексов, каждый из которых состоит из конечного числа квадратов, приклеенных друг к другу сторона к стороне специальным образом. Нас будут интересовать свойства комплексов как графов. Комплекс уровня 1 — это просто квадрат из четырёх вершин (плитка). Комплексы уровня 2 и 3 изображены на рис. 1.

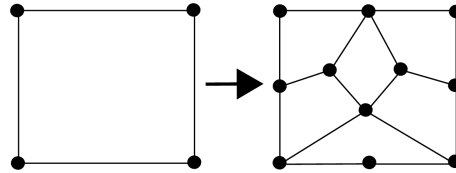


Рис. 1. Разбиение

Комплекс уровня 3 получается разбиением каждого квадрата на шесть квадратов, при этом в середине каждого ребра появляется дополнительная вершина. *Макроплиткой уровня n* называется плоская часть комплекса — результат применения $n - 1$ уровней разбиения к квадрату из четырёх вершин.

Можно заметить, что при этом образуются различные виды вершин: угловые, краевые, боковые и внутренние. Угловые и краевые лежат на краю комплекса (в углах или на сторонах), боковые лежат на ребре, проведённом при некотором разбиении, внутренние образуются при некотором разбиении.

Далее комплекс уровня n определяется индуктивно. Сначала к комплексу уровня n применяется разбиение: каждый квадрат делится на шесть квадратов, как на рис. 2, потом проводятся подклейки — к некоторым путям длины 4 на комплексе подклеиваются шесть квадратов (рис. 3).

В результате образуется комплекс следующего уровня. Структура разбиений нужна, чтобы обеспечить локальную преобразуемость: любой путь с концами на границах комплекса может быть преобразован в путь только по границам локальными заменами (когда два соседних ребра некоторого квадрата меняются на два соседних ребра).

Подклейки нужны, чтобы обеспечить равномерную эллиптичность комплек-

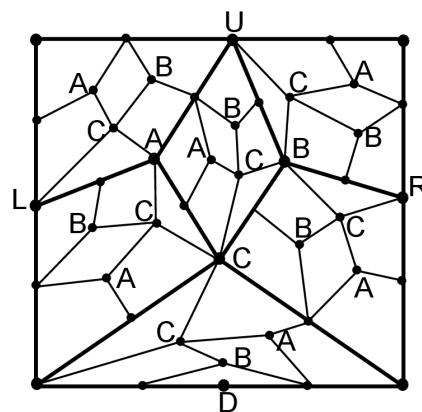


Рис. 2. Макроплитка третьего уровня. Отмечены типы внутренних вершин (A, B, C)

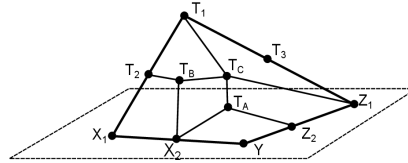


Рис. 3. Подклейка

сов: любой путь на комплексе может быть преобразован локальными заменами достаточно сильно в зависимости от его длины. Например, для естественной метрики, определённой на путях с общими концами, и некоторой константы α путь длины n локально преобразуется в путь, отличающийся на $n\alpha$.

В полугруппе, образованной кодировками путей на семействе комплексов, равномерная эллиптичность означает возможность достаточно сильно преобразовывать произвольные слова. Локальная преобразуемость же позволяет приводить слова к канонической форме.

Также нужны ещё несколько свойств, позволяющих ввести раскраску всех вершин и рёбер комплексов в конечное число цветов — они будут играть роль букв в полугруппе. Также часть свойств посвящена геометрической структуре комплексов, помогающей приводить кодировки не существующих на комплексе путей к нулю.

Нулевой формой мы называем путь по одному ребру туда-обратно. В полугруппе приравниваются к нулю слова, соответствующие *коротким* кодировкам перечисленных ниже типов:

- 1) не встречающиеся на комплексах;
- 2) пути по одному ребру туда-обратно (нулевая форма);
- 3) принадлежащие к числу «мёртвых паттернов» (это пути, которые не могут быть частью достаточно длинного кратчайшего пути на комплексе).

Если в процессе локальных преобразований в слове возникает короткое запрещённое подслово, мы приходим к нулю. Если путь лежит на комплексе, но является не кратчайшим путём, соединяющим его концы, то можно локально преобразовать его к нулевой форме и обнулить.

Ниже приводится список основных предложений первой части работы.

Лемма 3.1 (об ограниченности роста степени вершины). Для каждой вершины Z существует такое натуральное N , что начиная с уровня макроплит-

ки N степень (число входящих рёбер) вершины Z не меняется, т. е. она одинакова для макроплиток уровня N и $N + k$ для любого натурального k .

Следствие.

1. Каждая вершина заданной глубины x выступает в качестве ядра подклейки для ограниченного количества вершин.
2. В каждую вершину входит ограниченное количество рёбер различных уровней, в том числе рёбер из подклеек.

Утверждение нужно, чтобы обосновать конечность числа возможных типов вершин. Это нужно для введения конечного алфавита полугруппы.

Лемма 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 рёбер в некоторой макроплитке либо на границе подклеенной макроплитки.

Утверждение нужно для описания структуры кодировки типов вершин комплекса.

Лемма 3.3 (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 .

Кроме того, любой кратчайший путь, соединяющий противоположные углы или середины противоположных сторон макроплитки, имеет длину 2^n и может быть локально преобразован в любой из двух путей по границе (полупериметр) с теми же концами.

Это удобное утверждение для преобразований путей.

Лемма 3.4 (о выделении локального участка). Пусть n — уровень макроплитки и путь P лежит внутри неё.

1. Пусть оба края P лежат на границе T . Тогда, если путь P имеет длину не менее $5 \times 2^{n-2}$, он может быть локально преобразован в нулевую форму P' .
2. Пусть один край P лежит в углу или на середине стороны T , а второй край внутри T (часть пути может проходить по границе T). Тогда, если путь P имеет длину 2^n , он может быть локально преобразован в нулевую форму P' .

Это утверждение используется для приведения к каноническому виду. Суть в том, что достаточно длинные пути на небольшом куске комплекса гарантированно являются не кратчайшими.

Лемма 3.5 (о мёртвых паттернах). Рассмотрим некоторую макроплитку T и обозначим в ней внутренние вершины через A, B, C и боковые — через U, R, D, L (аналогично обозначениям при разбиениях). Тогда паттерны AUB, ACB, CXD (где X — нижняя левая либо нижняя правая вершина) являются мёртвыми.

Лемма 3.6 (о мёртвых путях в нижней подплитке). Рассмотрим некоторую макроплитку T уровня n . Пусть путь XYZ лежит в T , причём Y — левый нижний угол T , XU лежит на внутреннем ребре, идущем из левого нижнего угла T во внутреннюю вершину C , а YZ лежит на нижней стороне T . Тогда для любых плоских путей W_1, W_2 , длины которых более 2^{n+1} , путь W_1XYZW_2 можно локально преобразовать к нулевой форме.

Лемма 3.7 (о некорректных участках). Пусть есть некорректный участок XYZ в макроплитке T уровня n , причём T — минимальная макроплитка, содержащая XYZ в качестве некорректного участка. Тогда для любых плоских путей W_1, W_2 длины более 2^{n+2} путь W_1XYZW_2 может быть локально преобразован к нулевой форме.

Три утверждения выше выявляют суть понятия мёртвого паттерна и показывают, что некоторые геометрические пути на комплексе могут быть обнулены как не являющиеся частями длинных кратчайших путей.

Лемма 3.8 (о корректности путей). Пусть путь P представляет собой проход по двум соседним сторонам некоторой макроплитки T . Тогда никакие локальные преобразования не могут привести P к нулевой форме, а также к форме, содержащей некорректный участок.

Утверждение нужно для объяснения бесконечности полугруппы, поскольку некоторые длинные пути на комплексе не могут быть преобразованы к нулю.

Лемма 3.9 (о расстоянии от края до выхода в подклейку). Пусть вершина X лежит на краю некоторой макроплитки T , принадлежащей комплексу K , вершина Y принадлежит T , но не находится на её границе, а из Y существует выход в подклеенную макроплитку уровня $n \geq 2$, не содержащую вершин на границе T . Тогда расстояние от X до Y в комплексе K (длина кратчайшего пути по рёбрам) не менее 2^{n-1} .

Лемма 3.10 (об ограниченности пути, уходящего в подклеенную часть).

1. Пусть путь P имеет вид $V_1AV_2BV_3CV_4$, где рёбра из вершин A, B, C ведут в подклеенные области, а участки V_1, V_2, V_3, V_4 плоские. Тогда путь P не может быть плоским (лежать в одной макроплитке).
2. Пусть путь P начинается в вершине X , выход из X идёт в подклеенную макроплитку уровня n . Кроме того, пусть P не содержит рёбер — выходов

из подклеенных плиток. Тогда если длина пути не менее 2^{n+1} , то он может быть приведён к нулевой форме.

Эти утверждения нужны для доказательства, что периодические пути с периодом 9 гарантированно приводятся к нулю.

4. Строение полугруппы путей

В предыдущей части работы мы описали геометрическую структуру семейства комплексов. В следующем разделе 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 для каждого узла входящие и выходящие рёбра, 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 рёбра, Y_i — буквы, кодирующие узлы (их типы, окружения и информацию), Z_i — буквы, кодирующие выходящие рёбра (детали кодирования будут показаны в последующих разделах.)

Будем говорить, что слово W имеет **CODE-форму**, если в нём 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 нулём S с порождающими $\{X_i, Y_i, Z_i\}$.

Лемма 4.1 (о стандартной форме слова). *В полугруппе с нулём 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, введём 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 будет 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 *запрещёнными* и введём 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. \square

В последующих разделах будет показано, что для двух эквивалентных путей, не содержащих мёртвых паттернов и нулевых форм, по коду одного из них

можно восстановить код другого. Это даёт 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 после преобразования оно будет кодировкой эквивалентного пути. Допустим, после нескольких таких замен образуется запрещённое 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 запрещённое 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 (т. е. существует путь короче по длине с теми же концами), то согласно лемме 3.4 о выделении локального участка из первой части работы, 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 некротчайшим путям.

Мёртвые паттерны

Мёртвые паттерны представляют собой некоторые типы слов-кодировок, которые никогда не могут встретиться в достаточно больших ненулевых словах. Нас будут интересовать только мёртвые паттерны $AUB, ACB, CXD, BUA, BSA, DXC$, которые обсуждаются в лемме 3.5 о мёртвых паттернах из первой части. Выпишем все слова, длины не более 4, содержащие хотя бы один из перечисленных мёртвых паттернов. Для каждого такого слова W , введём соотношение $W = 0$.

Заметим, что никакие достаточно большие ненулевые слова не могут содержать мёртвый 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 в форму, содержащую мёртвый паттерн.

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

Нашей основной целью будет показать, что ненулевыми словами в полугруппе являются только регулярные слова, т. е. 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 мёртвые паттерны.

Примечание об использовании мёртвых паттернов для конструкции

Ценой некоторого усложнения кодировки можно добиться того, чтобы пути, содержащие мёртвые паттерны, допускали такие же локальные преобразования, как и обычные слова. В этом случае можно не вводить мономиальные соотношения для слов, их содержащих. 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, имеются ребро входа и ребро выхода. Рёбра 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 неё могут выходить рёбра, ведущие в подклеенные плитки/макроплитки. Таким образом, каждая вершина лежит на своей базовой плоскости и ещё участвует в нескольких подклеенных плоскостях. Фактически, вершина играет свою роль (занимает определённое положение) в своей макроплитке на базовой плоскости и выполняет какие-то другие роли в 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, причём любые две соседние вершины 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, т. е. входящие в неё рёбра либо относятся к плоскости вершины, либо классифицируются по принадлежности к различным подклеенным макроплиткам. В общем случае путь приходит в вершину из одной подклеенной макроплитки и выходит в другую. Общая кодировка вершины будет состоять из трёх *плоских* кодов: первый представляет собой плоский код вершины (без учёта 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 все четыре части.

5.1. Параметр «тип» для вершин

Тип вершины определялся в первой части работы. Вершины разделены на следующие категории.

1. *Угловые* (лежащие в углах подклеенных макроплиток или всего комплекса). Тип угловой вершины определим как один из четырёх вариантов углов, в котором она может находиться: CUL, CUR, CDR, CDL (Corner Up-Left и т. д.). Угловые вершины принадлежат макроплитке, где они являются углами.
2. *Краевые* (лежащие на стороне подклеенной макроплитки или всего комплекса). 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. Краевые вершины принадлежат макроплитке, на краю которой они лежат.
3. *Внутренние*. В этой категории определим три типа вершин: A, B, C, отвечающие внутренним узлам макроплиток; эти вершины создаются внутри разбиваемой макроплитки. Будем считать, что эти вершины принадлежат данной макроплитке.
4. *Боковые* (лежащие на границе между двумя макроплитками, на внутреннем ребре). Типы боковых вершин будут соответствовать всем упорядоченным парам из множества $\{U, R, D, L\}$, а именно DR, RD, DL и т. д. Будем считать, что в упорядоченной паре первой называется буква, соответствующая A-стороне внутреннего ребра, на котором лежит вершина. Все боковые вершины создаются в середине стороны разбиваемой макроплитки. Тип боковой вершины определяет, серединой каких именно сторон она является в двух макроплитках, где она является серединой сторон. Это как раз те макроплитки, которые разбивались при создании данной вершины. Будем считать, что боковые вершины принадлежат макроплитке, которой принадлежит данное внутреннее ребро.

В дальнейшем вершину типа A будем для простоты называть A-вершиной или A-узлом. Аналогично поступим с вершинами других типов. Вершины на

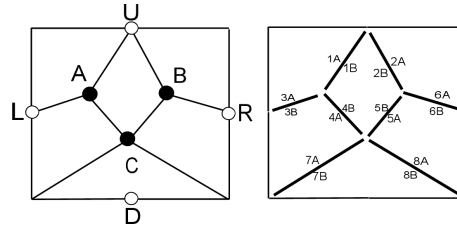


Рис. 4. Чёрные (внутренние) и белые (боковые) вершины; типы рёбер

графе мы также будем называть узлами. В левой части рис. 4 представлены внутренние и боковые вершины. Следующая лемма показывает, что возможны только указанные сочетания сторон.

Лемма 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, правая и левая, левая и нижняя, 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 рис. 4). Рассмотрим вершину F , лежащую в середине этого ребра. F тоже боковая вершина, и сочетания ориентаций сторон у неё могут быть только право и низ, верх и право, лево и низ, верх и лево, т. е. 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 рёбер YF и FZ тоже могут иметь сочетания ориентаций только право и низ, верх и право, лево и низ, верх и лево (\mathbb{RD} , \mathbb{UR} , \mathbb{LD} , \mathbb{UL} и транспозиции). Например, если макроплитки соприкасаются нижней и правой сторонами, то F имеет тип \mathbb{DR} или \mathbb{RD} , и тогда YF и FZ будут иметь типы \mathbb{RU} и \mathbb{UL} (или наоборот). Повторяя операцию разбиения, можно получить, что все боковые вершины на ребре YZ будут относиться к одному из указанных четырёх типов, т. е. будут являться серединами правой и нижней, либо верхней и правой, либо левой и нижней, либо верхней и левой сторон. \square

5.2. Типы рёбер 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), т. е. белые вершины принадлежат тем макроплиткам, в середине сторон которых они лежат, а чёрные — макроплиткам минимального уровня, внутри которых они находятся.

Типы рёбер

Всего существует 16 внутренних рёбер (с учётом стороны, рис. 4), это рёбра **1A, 1B, . . . , 8B**, и 4 краевых ребра: **left, top, right, bottom**.

Уровни вершин

Для чёрных, внутренних, вершин уровень определять не будем (можно считать, что у всех один уровень).

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

Рассмотрим чёрную вершину X . Если X имеет тип **A** или **B**, то она находится на стыке трёх макроплиток (разбивающих макроплитку, которой X принадлежит), для **C** таких макроплиток четыре. В соответствии с определением 3.8 из первой части рёбра, исходящие из X и при этом лежащие на границе каких-то двух из этих макроплиток, являются *главными рёбрами* для X .

Если X — белая вершина, лежащая на середине сторон двух макроплиток T_1 и T_2 , то два ребра, лежащих на границе между T_1 и T_2 , тоже называются *главными*. Если боковая вершина лежит на краю только одной макроплитки, то главные рёбра — это рёбра, лежащие на её границе.

Из X выходят ещё несколько неглавных рёбер (в зависимости от типа).

Рёбра входа и выхода. Обозначения

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

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

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

У вершин типа **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 **B** может быть три неглавных ребра, одно уходит в правую верхнюю подплитку, второе в правую нижнюю, третье в среднюю. Обозначим их

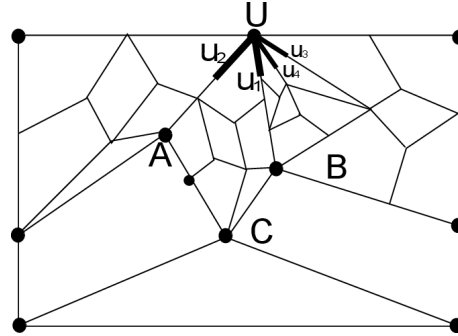


Рис. 5. Неглавные рёбра, выходящие из бокового узла

соответственно **ru**, **rd**, **mid**. У вершин типа **C** может быть шесть неглавных рёбер, два уходят в левую нижнюю подплитку, два в среднюю и два в нижнюю. В каждой паре одно из рёбер иерархически старше (появляется при разбиении раньше). Обозначим рёбра соответственно **ld₁**, **ld₂**, **mid₁**, **mid₂**, **d₁**, **d₂**. Старшему ребру даем первый номер, другому — второй.

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

- U-узел, или U-часть UL-узла или UR-узла, — **u₁**, **u₂**, **u₃**, **u₄**;
- R-узел, или R-часть UR-узла или DR-узла, — **r**, **r₂**, **r₃**;
- L-узел, или L-часть UL-узла или LD-узла, — **l**, **l₂**, **l₃**.

В D-узел неглавные рёбра не приходят.

Рёбра нумеруются по иерархическому старшинству, при равном старшинстве — по часовой стрелке. Например, если есть четыре неглавных ребра из узла типа LU, входящих внутрь плитки T, примыкающей верхней стороной (рис. 5), то **u₁** будет ребром в сторону B-узла T, **u₂** — ребром в сторону A-узла T, **u₃** и **u₄** будут входить в верхнюю правую подплитку T.

Типы рёбер, ведущих в подклеенные области

Рассмотрим вершину X. В подклеенной макроплитке она может иметь один из следующих типов: L, U, CUR, CUL, CDL.

Из вершины L могут выходить рёбра **l**, **l₂**, **l₃** (как и в плоском случае). Чтобы показать, что имеются в виду рёбра в подклеенную область, будем записывать их как **l₁**, **l₂**, **l₃** (рис. 6).

Аналогично из вершины U могут выходить рёбра **u₁**, **u₂**, **u₃**, **u₄**.

Из вершины CUR может выходить только одно ребро в подклеенную область — это ребро по правой стороне подклеенной макроплитки. Будем обозначать его **f**.

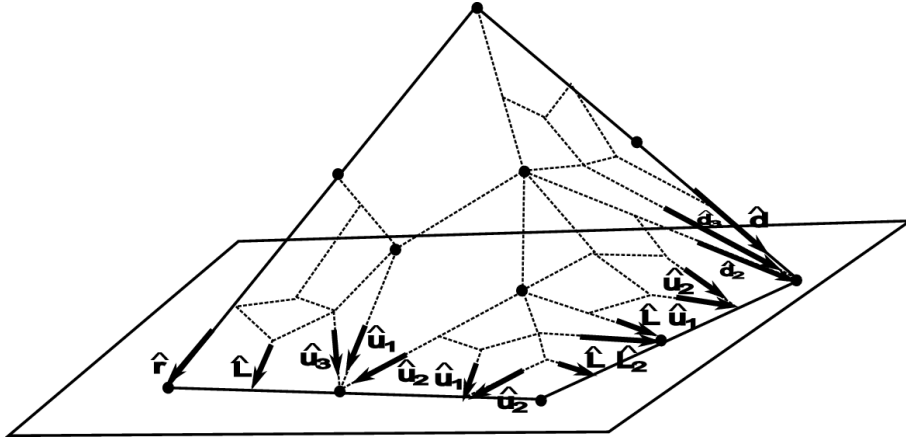


Рис. 6. Рёбра в подклеенные области

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

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

Цепи

Выберем некоторую вершину X . Обозначим через $T(X)$ 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 рёбер типов 1 и 3 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, причём центр для этой цепи определяется однозначно. Под *макроплитками из цепи* будем понимать макроплитки из соответствующего множества $T(X)$. Заметим, что между рёбрами, выходящими из X , вершина цепи, т. е. 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 ,

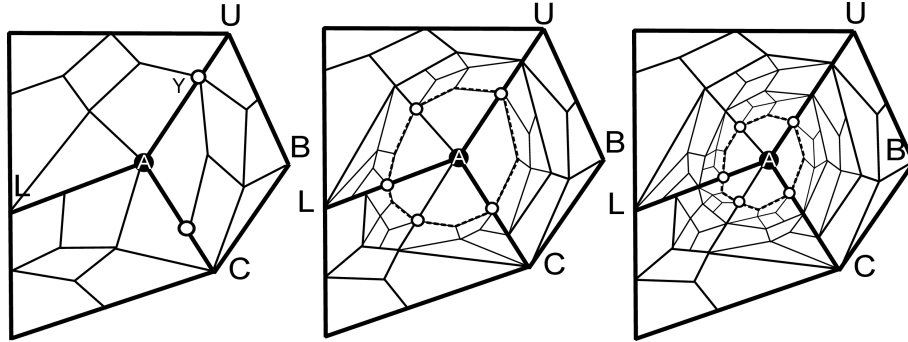


Рис. 7. Цепи узла типа А

а остальные перечисляются в соответствии с выходящими из X рёбрами по часовой стрелке. Номер вершины в её 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: самой крупной цепи — нулевой (0-цепь), затем первый и т. д. Если узел X принадлежит некоторой макроплитке T , то рёбра цепи уровня k принадлежат макроплитке, получающейся после $k + 1$ разбиения T .

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

5.3. Параметр «окружение» для узлов

Определение 5.2. Для каждой макроплитки можно записать упорядоченную четвёрку типов рёбер, на которых лежат её стороны: левая, верхняя, правая, нижняя (строго в данном порядке). Такую четвёрку 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. Определим *метку вершины относительно P*.

1. Для регулярных внутренних вершин (типов А, В, С) метка — это окружение макроплитки, которой принадлежит данная вершина.

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

2. Для регулярных боковых вершин (типов UL/LU, UR/RU, LD/DL, RD/DR) метка — упорядоченная пара окружений двух макроплиток, в середине сторон которых лежит данная вершина.

Пример. Если x, y — типы левой и верхней сторон макроплитки (рис. 7), то у узла Y в левой части рисунка будет метка $[x, y, 1A, 3A]$ -[1В, 2В, 4В, 5В].

3. Для угловых вершин метка — это окружение подклеенной макроплитки (**left, top, right, bottom**).
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 рис. 7 окружение

$$[1A, 8A, 5A, 6B]:[2B, 4B, 1A, 3A] \times [2B, 4B, 1A, 3A]:[8B, 4A, 4A, 7B] \times \\ \times [8B, 4A, 4A, 7B]:[3B, 8A, 5A, 6B] \times [3B, 8A, 5A, 6B]:[8B, 3A, 3A, 7B] \times \\ \times [8B, 3A, 3A, 7B]:[1A, 8A, 5A, 6B].$$

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

Определение 5.5. Определим *окружение вершины* относительно плоскости P .

1. Для регулярных боковых вершин типов $UL/LU, UR/RU, LD/DL$, а также для краевых вершин типов U и L окружение — это упорядоченная пара (x, y) , где x — окружение цепи, в которую входит данная вершина, y — указатель, номер данной вершины в её цепи.
2. Для регулярных боковых вершин типов RD/DR окружение — это тип ребра, на котором находится данная вершина (это ребро 2, 3, 5 или 6).
3. Для всех остальных вершин (угловых, краевых, регулярных типов A, B, C) окружение — это метка 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 через $\mathbf{SurrP}(X)$. Если из контекста ясно, о какой плоскости идёт речь, будем использовать обозначение $\mathbf{Surr}(X)$.

Пусть X — узел, входящий в некоторую цепь. Если Y — центр этой цепи, то X соответствует одному из выходящих из Y рёбер. Номер этого ребра (считая 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. Определим *расширенное окружение* относительно области P . Это упорядоченная пара: окружение относительно P и окружение относительно базовой плоскости вершины.

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

5.4. Параметр «информация» для узлов

Начальники

Определим понятие *начальников* для регулярных вершин. Пусть вершина X регулярная (т. е. не угловая и 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 рёбер T .

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

Для боковых вершин, лежащих на рёбрах 2, 5, 6 макроплитки T , *вторым начальником* будем называть узел в правом нижнем углу макроплитки T . Для боковых вершин, лежащих на рёбрах 7 и 8, а также 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 рёбрах 1, 3, 4, а также для A -узлов *информацией вершины* будем называть упорядоченную тройку, состоящую из типа, уровня и расширенного окружения её единственного начальника.

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

Для регулярных боковых вершин, лежащих на внутренних рёбрах 7, 8, а также для C -узлов *информацией вершины* будем называть упорядоченную девятку, состоящую из типов, уровней и расширенных окружений всех трёх её 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 её единственного начальника, CDL -узла в той же макроплитке.

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

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

5.5. Ядро подклейки и флаг макроплитки

Рассмотрим некоторую макроплитку T , подклеенную к рёбрам e_1 и e_2 некоторой вершины X . Вершина X является *ядром подклейки* в соответствии с определением из первой части. Она лежит в левом верхнем углу T . Упорядоченную пятёрку параметров (тип, базовое окружение X , информация X , тип ребра e_1 , тип ребра e_2) будем называть *флагом подклейки* для макроплитки T .

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

5.6. Кодировка путей

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

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 четвёрку, состоящую из типа X_i , её уровня, её окружения и её информации.

Если одно из рёбер, 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 следующий упорядоченный набор из четырёх 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) четвёртый элемент — информация X_i .

Таким образом, кодировка учитывает положение вершины и в макроплитке, откуда пришло ребро, и в макроплитке базовой плоскости. Для случая, когда входящее ребро плоское, а выходящее ведёт 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 следующий упорядоченный набор из четырёх элементов:

- 1) первый элемент — упорядоченная тройка, включающая 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, включающая 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, включающая окружение X_i в области P_{i-1} , окружение X_i в базовой плоскости, окружение X_i в области P_i ;
- 4) четвёртый элемент — информация X_i .

Замечание. Как видно, информация не подчиняется общей логике. Это происходит потому, что любая вершина в своих подклеенных макроплитках всегда занимает место на верхней или левой сторонах либо 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 через $\mathbf{Code}(X)$. 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*

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

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

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, т. е. 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 понять устройство ближайшей окрестности. Для удобства мы зададим набор функций на узлах, аргументами которых являются типы, окружения или информации некоторых специально расположенных узлов, а значениями — окружения или типы узлов в окрестности узла-аргумента. В следующем разделе 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 имеют два аргумента: первый — это окружение узла X , являющегося левым нижним углом в макроплитке T , второй — тип ребра выхода из X , соответствующего нижней стороне T .

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

Функции 6 и 7 имеют два аргумента: окружение и информацию узла X , являющегося серединой правой стороны в макроплитке T .

Ещё 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 разделе 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 её значение.

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

Цепи с центром в узле типа Δ

На рис. 8 белыми точками отмечены цепи узла типа Δ нулевого, первого и второго уровней. Заметим, что на втором уровне все пять макроплиток, левый верхний угол которых попадает в узел типа Δ , занимают левое верхнее положение в своих родительских макроплитках. Значит, если взять следующий уровень подразделения, типы рёбер — границ макроплиток не изменятся, т. е. окружения вершин в цепях будут те же, 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 Δ . Заметим, что по окружению цепи мы можем установить, является ли центр цепи узлом типа Δ и какого уровня цепь. Действительно, одна из макроплиток в цепи с центром в узле типа Δ обязательно имеет окружение

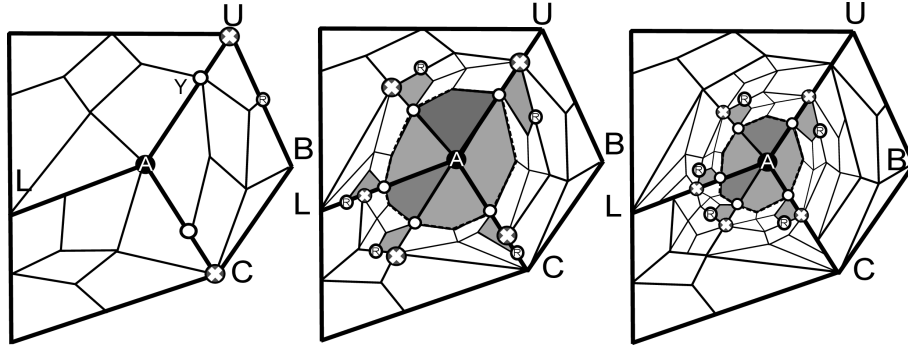


Рис. 8. Цепи узла типа A

(**4B, 1B, 2B, 5B**) или (**4B, 1B, 1A, 3A**), причём макроплитки с такими окружениями не могут входить в цепи с другими центрами. Цепь нулевого уровня содержит макроплитку с окружением (**4B, 1B, 2B, 5B**) (остальные — нет). Для цепи второго уровня все четвёрки типов рёбер имеют вид $(x, y, \mathbf{1A}, \mathbf{3A})$, а для первого уровня это не так.

Забегая вперёд, можно сказать, что возможность установить тип центра по окружению цепи имеется и для центров других типов (не только A).

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

Из сказанного выше следует, что для узлов типа A значение функции **LevelPlus** может быть установлено по её аргументу.

На рис. 8 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 рёбер, 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 рёбер), если знаем окружение центра (узла типа 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 (её 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. Всё это значит, что мы можем вычислить функции **TopFromCorner**, **RightCorner**.

Функции **TopFromRight** и **BottomRightTypeFromRight** могут быть применены только в одном случае: для 0-цепи, если аргументом является узел

в середине ребра типа 1 (обозначим этот узел через Y , а макроплитку, где Y является серединой правой стороны, через T). **BottomRightTypeFromRight** в этом случае — это просто наш узел типа \mathbb{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 , причём 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 \mathbb{A} -цепи.

Цепи с центром в узле типа \mathbb{B}

На рис. 9 белыми точками отмечены цепи узла типа \mathbb{B} . 0-цепи с центром в узле типа \mathbb{B} не существует (все боковые узлы в соответствующей окрестности имеют тип \mathbb{DR}). Заметим, что в 2-цепи все шесть макроплиток, левый верхний угол которых попадает в узел типа \mathbb{B} , занимают левое верхнее положение в своих родительских макроплитках. Значит, если взять следующий уровень подразделения, типы рёбер — границ макроплиток не изменятся, т. е. окружения вершин в цепях будут те же, значит, 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 \mathbb{B} . Заметим, что по окружению цепи мы можем установить, является ли центр цепи узлом типа \mathbb{B} и какого уровня цепь. Действительно,

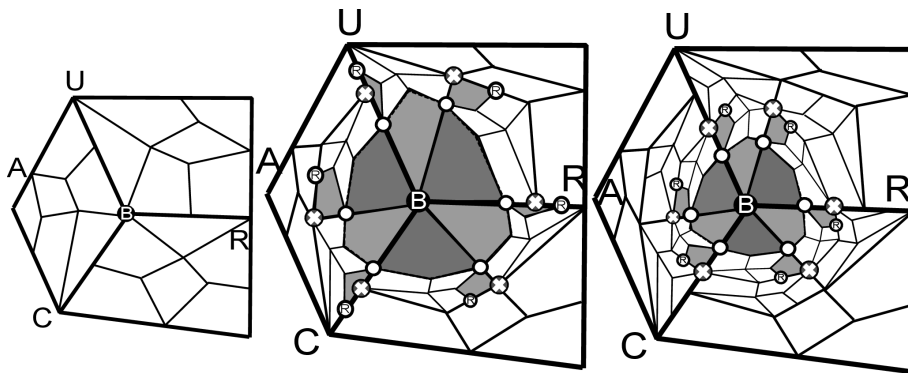


Рис. 9. Цепи узла типа \mathbb{B}

1-цепь с центром в узле типа \mathbb{B} содержит узел с окружением $(8\mathbb{B}, 5\mathbb{B}, 5\mathbb{B}, 7\mathbb{B})$, который не может содержаться ни в какой другой цепи. 2-цепь с центром в узле типа \mathbb{B} содержит узел с окружением $(8\mathbb{B}, 5\mathbb{B}, 1\mathbb{A}, 3\mathbb{A})$, не встречающийся ни в какой другой цепи.

Заметим также, что мы можем выписать полностью всё окружение 1-цепи и 2-цепи с центром в узле типа \mathbb{B} .

Из сказанного выше следует, что для узлов типа \mathbb{B} значение функции **LevelPlus** может быть установлено по её аргументу. На рис. 9 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 рёбер, 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 (её 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. Всё это значит, что мы можем вычислить функции **TopFromCorner**, **RightCorner**.

Функции **TopFromRight** и **BottomRightTypeFromRight** не могут быть применены, так как аргументом этих функций не может быть узел из \mathbb{B} -цепи, поскольку ни один узел из цепи с центром в узле типа \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 рис. 10 и 11 белыми точками отмечены 1-цепи и 2-цепи узла типа \mathbb{C} . 0-цепи с центром в узле типа \mathbb{C} не существует.

Кроме указанных на рисунках цепей первого и второго уровней, существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рис. 11. На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел типа \mathbb{C} , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразбиения типы рёбер — границ макроплиток не изменятся, т. е. окружения вершин в цепях будут те же, значит, окружение цепи четвёртого 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

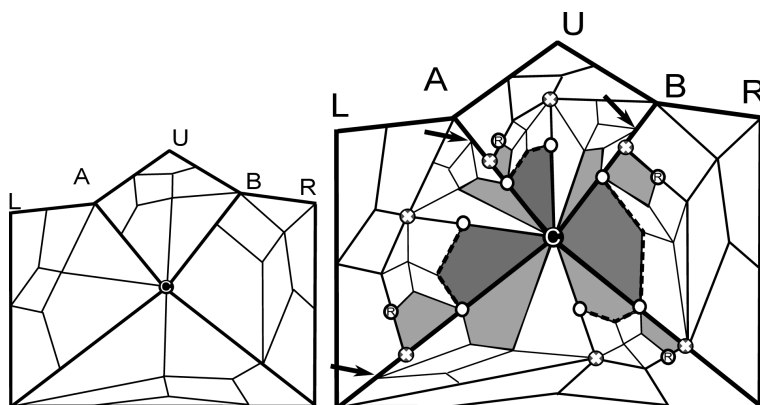


Рис. 10. 1-цепь узла типа С

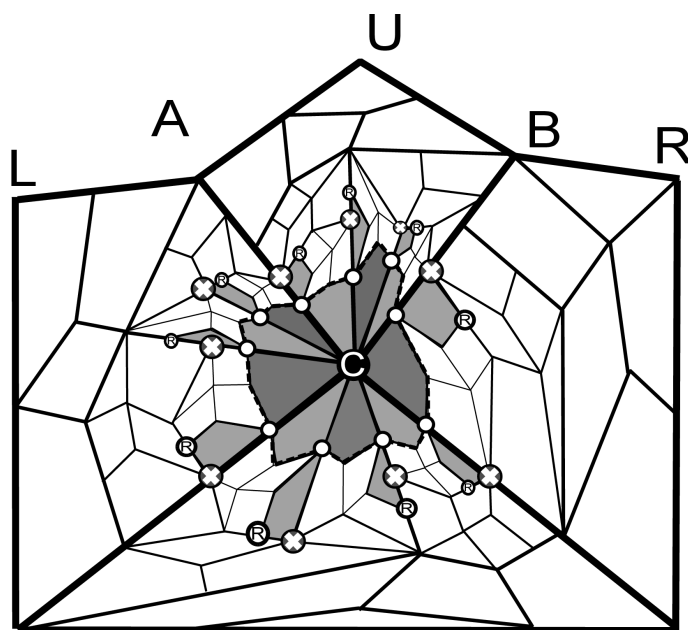


Рис. 11. 2-цепь узла типа С

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 всё окружение 1-цепей, 2-цепей, 3-цепей с центром в узле типа \mathbb{C} .

Из сказанного выше следует, что для узлов типа значение функции **LevelPlus** может быть установлено по её аргументу. На рис. 10 и 11 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 рёбер, 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 (её 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. (Всё это также очевидно проверяется для 3-цепи.) Всё это значит, что мы можем вычислить функции **TopFromCorner**, **RightCorner**.

Функции **TopFromRight** и **BottomRightTypeFromRight** могут быть применены только к 1-цепи (в остальные входят только узлы типов \mathbb{UL} и \mathbb{LU}). В случае 1-цепи аргументом могут быть узлы, лежащие на рёбрах 1, 2 и 4 (относительно \mathbb{C}). Пусть X — такой узел. Во всех случаях значением **BottomRightTypeFromRight** будет \mathbb{C} . Узлы, на которые указывает **TopFromRight** отмечены стрелками на рис. 10. Таким образом, **TopFromRight** принимает следующие значения: 1-цепь вокруг A для ребра 1 (с указателем 2), 1-цепь 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. раздел 6.2).

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

Цепи с центром в узлах типа \mathbb{UL} и \mathbb{LU}

Случаи узлов типа \mathbb{UL} и \mathbb{LU} симметричны, для определённости будем далее разбирать случай узла типа \mathbb{UL} . Будем считать, что главное ребро имеет тип t , т. е. с U -стороны это tA , а с L -стороны — tB . На рис. 12 и 13 белыми точками отмечены 1-цепи и 2-цепи узла типа \mathbb{UL} . 0-цепи с центром в узле типа \mathbb{UL}

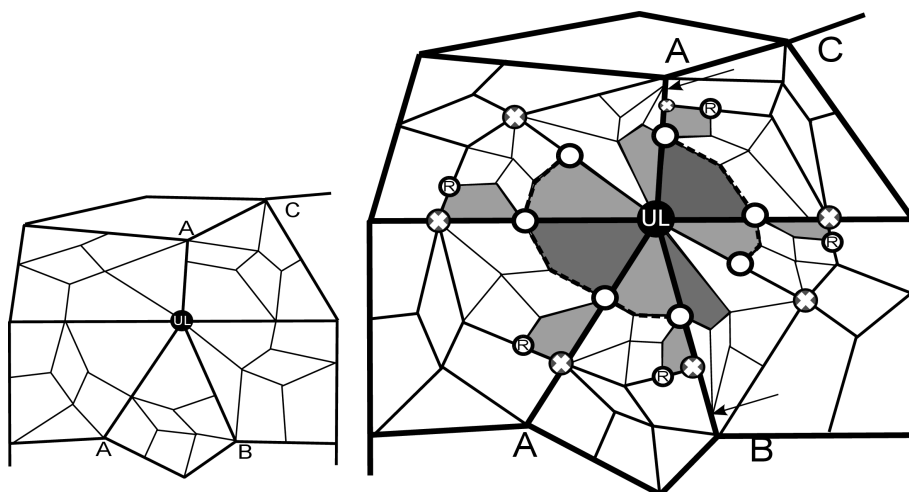


Рис. 12. 1-цепь узла типа UL

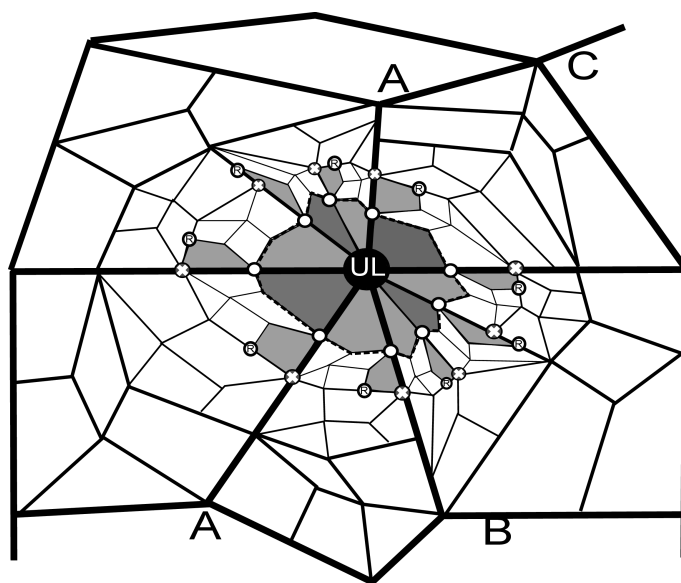


Рис. 13. 2-цепь узла типа UL

не существует. Кроме указанных на рисунках цепей первого и второго уровней, существует также 3-цепь, получаемая применением операции разбиения к макролиткам на рис. 13.

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел типа \mathbb{UL} , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразделения типы рёбер — границ макроплиток не изменятся, т. е. окружения вершин в цепях будут те же, значит, окружение цепи четвёртого и последующих уровней совпадает с окружением цепи третьего уровня.

Таким образом, существуют только три возможные конфигурации окружения цепи с центром в узле типа \mathbb{UL} (для цепей первого, второго и третьего уровней, при этом ещё может быть выбрано разное главное ребро t). Заметим, что по окружению цепи мы можем установить, является ли центр цепи узлом типа \mathbb{UL} , какого уровня цепь, а также сам параметр t . Действительно, 1-цепь с центром в узле типа \mathbb{UL} содержит узлы с окружением

$$(tA, 1A, 6A, 2A)-(1B, 2B, 6A, 2A) \text{ и } (8B, 3A, 3A, 7B)-(tB, 3B, 6A, 2A),$$

которые ни в какой другой цепи вместе не встречаются. 2-цепь с центром в узле типа \mathbb{UL} содержит одновременно узлы с окружением

$$(tA, 1A, 1A, 3A)-(1B, 2B, 1A, 3A) \text{ и } (3A, 8A, 5A, 6B)-(tB, 3B, 1A, 3A),$$

которые также ни в какой другой цепи вместе не встречаются. Для 3-цепи такой парой будет

$$(tA, 1A, 1A, 3A)-(1B, 2B, 1A, 3A) \text{ и } (3A, 8A, 1A, 3A)-(tB, 3B, 1A, 3A).$$

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

Из сказанного выше следует, что для узлов типа \mathbb{UL} значение функции **LevelPlus** может быть установлено по её аргументу.

На рис. 12 и 13 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 рёбер, 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 (её 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. (Всё это также очевидно проверяется для 3-цепи.) Всё это значит, что мы можем вычислить функции **TopFromCorner**, **RightCorner**.

Функции **TopFromRight** и **BottomRightTypeFromRight** могут быть применены только к 1-цепи (в остальные входят только узлы типов \mathbb{UL} и \mathbb{LU}). В случае 1-цепи аргументом могут быть узлы, лежащие на рёбрах u_1 и l (относительно центрального узла типа \mathbb{UL}). Пусть X — такой узел. Во всех случаях

значением **BottomRightTypeFromRight** будет наш же узел типа UL . Узлы, на которые указывает **TopFromRight**, отмечены стрелками на рис. 12. Таким образом, **TopFromRight** принимает следующие значения: 1-цепь вокруг \mathbb{B} (с указателем 1) для ребра u_1 , 1-цепь вокруг \mathbb{A} (с указателем 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 типа UR и RU симметричны, для определённости будем далее разбирать случай узла типа UR . Будем считать, что главное ребро имеет тип t , т. е. с U -стороны это tA , а с R -стороны — tB .

На рис. 14 и 15 белыми точками отмечены 1-цепи и 2-цепи узла типа UR . 0-цепи с центром в узле типа UR не существует. Кроме указанных на рисунках цепей первого и второго уровней, существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рис. 15.

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел типа UR , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразделения типы рёбер — границ макроплиток не изменятся, т. е. окружения вершин в цепях будут те же, значит, окружение цепи четвёртого и последующих уровней совпадает с окружением цепи третьего уровня.

Таким образом, существуют только три возможные конфигурации окружения цепи с центром в узле типа UR (для цепей первого, второго и третьего уровней при этом ещё может быть выбрано разное главное ребро t). Заметим,

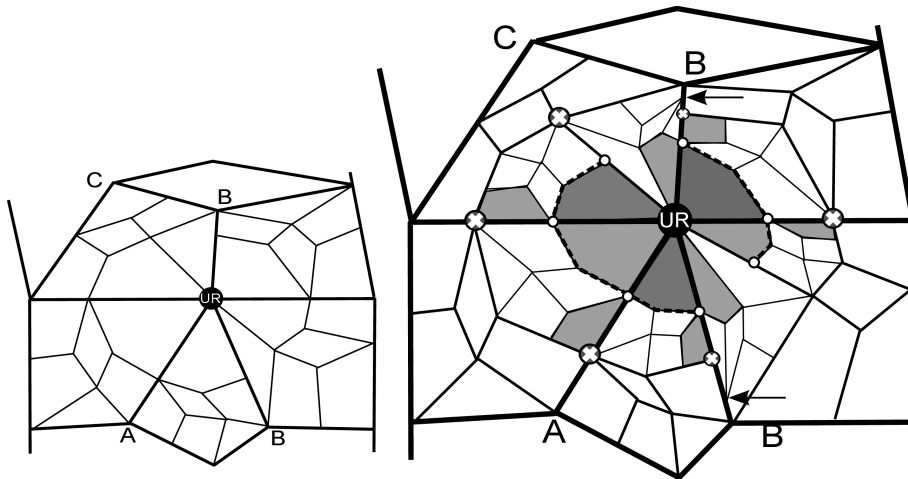


Рис. 14. 1-цепь узла типа UR

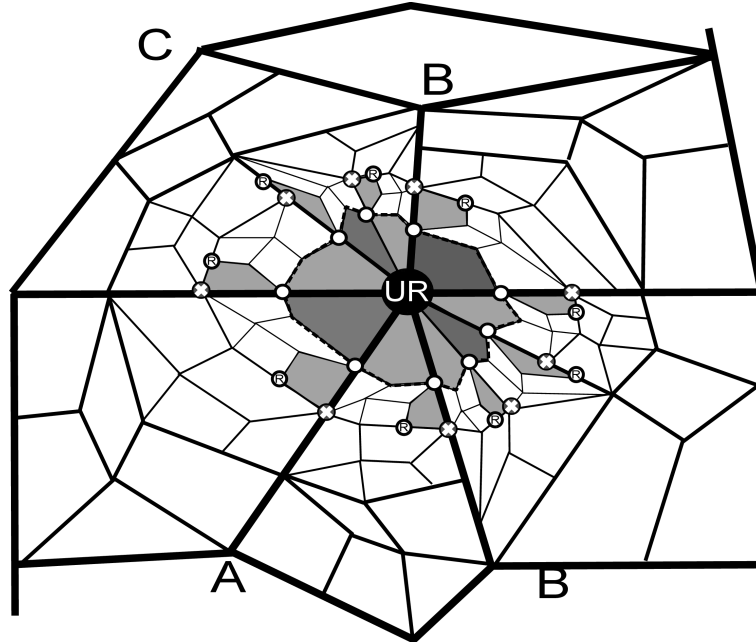


Рис. 15. 2-цепь узла типа UR

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 узле типа UR содержит одновременно узлы с окружением

$$(tA, 1A, 6A, 2A)-(1B, 2B, 6A, 2A) \text{ и } (tB, 6A, 6A, 2A)-(8B, 6B, 6B, 7B),$$

которые ни в какой другой цепи вместе не встречаются. 2-цепь с центром в узле типа UR содержит одновременно узлы с окружением

$$(tA, 1A, 1A, 3A)-(1B, 2B, 1A, 3A) \text{ и } (tB, 6A, 1A, 3A)-(6B, 8A, 5A, 6B),$$

которые также ни в какой другой цепи вместе не встречаются. Для 3-цепи такой парой будет

$$(tA, 1A, 1A, 3A)-(1B, 2B, 1A, 3A) \text{ и } (tB, 6A, 1A, 3A)-(6B, 8A, 1A, 3A).$$

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

Из сказанного выше следует, что для узлов типа UR значение функции **LevelPlus** может быть установлено по её аргументу.

На рис. 14 и 15 знаком \otimes отмечены узлы, являющиеся верхними правыми или левыми нижними углами в макрочитках, 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 рёбер, 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 (её 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. (Всё это также очевидно проверяется для 3-цепи.) Всё это значит, что мы можем вычислить функции **TopFromCorner**, **RightCorner**.

Функции **TopFromRight** и **BottomRightTypeFromRight** могут быть применены только к 1-цепи (в остальные входят только узлы типов \mathbb{UL} и \mathbb{LU}). В случае 1-цепи аргументом могут быть узлы, лежащие на рёбрах \mathbf{u}_1 и \mathbf{r} (относительно узла типа \mathbb{UR}). Пусть X — такой узел. Во всех случаях значением **BottomRightType** будет наш же узел типа \mathbb{UR} . Узлы, на которые указывает **TopFromRight** отмечены стрелками на рис. 14. Таким образом, **TopFromRight** принимает следующие значения: 1-цепь вокруг \mathbb{B} (с указателем 1) для ребра \mathbf{u}_1 , 1-цепь вокруг \mathbb{B} (с указателем 2) для ребра \mathbf{l} .

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

Цепи с центром в узлах типа \mathbb{DL} и \mathbb{LD}

Случаи узлов типа \mathbb{DL} и \mathbb{LD} симметричны, для определённости будем далее разбирать случай узла типа \mathbb{DL} . Будем считать, что главное ребро имеет тип t , т. е. с D -стороны это tA , а с L -стороны — tB . На рис. 16 и 17 белыми точками отмечены 1-цепи и 2-цепи узла \mathbb{DL} . 0-цепи с центром в \mathbb{DL} не существует. Кроме указанных на рисунках цепей первого и второго уровней, существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рис. 17.

На получающемся третьем уровне все макроплитки, левый верхний угол которых попадает в узел типа \mathbb{DL} , будут занимать левое верхнее положение в своих родительских макроплитках. Значит, на следующих уровнях подразбиения типы рёбер — границ макроплиток не изменятся, т. е. окружения вершин в цепях будут те же, значит, окружение цепи четвёртого и последующих уровней совпадает с окружением цепи третьего уровня.

Таким образом, существуют только три возможные конфигурации окружения цепи с центром в узле типа \mathbb{DL} (для цепей первого, второго и третьего уровней, при этом ещё 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 узле типа \mathbb{DL} содержит одновременно узлы с окружением

$$(tA, tA, \mathbf{6A}, \mathbf{2A})-(tB, \mathbf{3B}, \mathbf{6A}, \mathbf{2A}) \text{ и } (tA, tA, \mathbf{6A}, \mathbf{2A})-(\mathbf{7A}, tB, \mathbf{3B}, \mathbf{4A}),$$

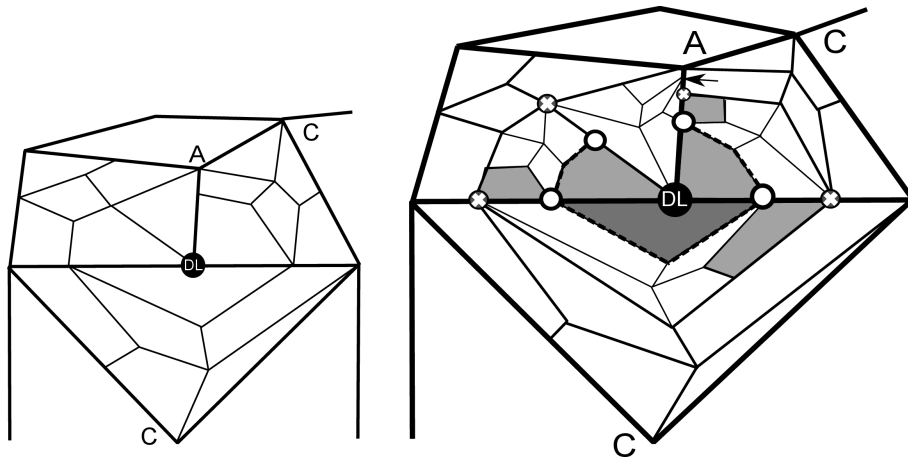


Рис. 16. 1-цепь узла DL

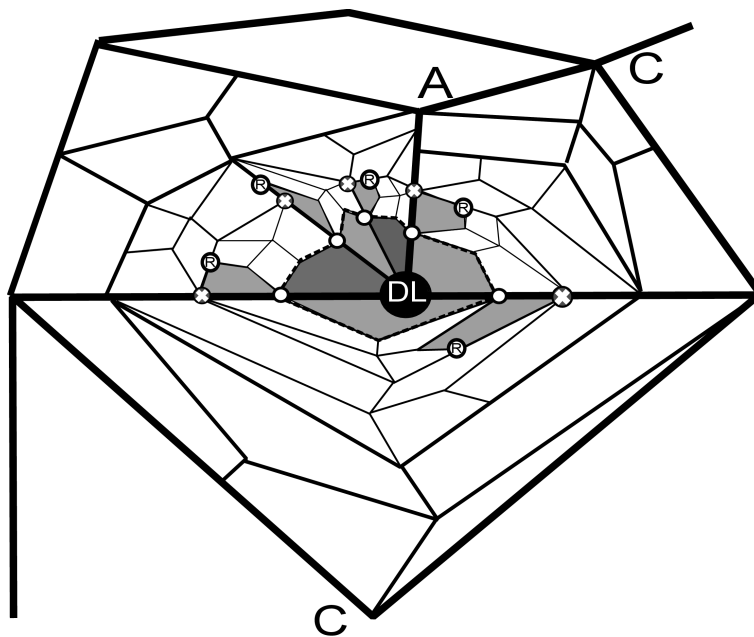


Рис. 17. 2-цепь узла DL

которые ни в какой другой цепи вместе не встречаются. 2-цепь с центром в узле типа \mathbb{DL} содержит одновременно узлы с окружением

$$(tA, tA, 1A, 3A)-(tB, 3B, 1A, 3A) \text{ и } (3A, 8A, 5A, 6B)-(8B, 7B, 7B, 7B),$$

которые также ни в какой другой цепи вместе не встречаются. Для 3-цепи такой парой будет

$$(tA, tA, 1A, 3A)-(tB, 3B, 1A, 3A) \text{ и } (3A, 8A, 1A, 3A)-(8B, 7B, 1A, 3A).$$

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

Из сказанного выше следует, что для узлов типа \mathbb{DL} значение функции **LevelPlus** может быть установлено по её аргументу. На рис. 16 и 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, второго и третьего уровней все типы узлов с крестами, а также типы рёбер, 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 (её 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. (Всё это также очевидно проверяется для 3-цепи.) Всё это значит, что мы можем вычислить функции **TopFromCorner**, **RightCorner**.

Функции **TopFromRight** и **BottomRightTypeFromRight** могут быть применены только к 1-цепи (в остальные входят только узлы типов \mathbb{UL} и \mathbb{LU}). В случае 1-цепи аргументом может быть только узел, лежащий на 1 ребре (относительно узла типа \mathbb{DL}). Пусть X — этот узел. Тогда **BottomRightTypeFromRight** будет наш же узел типа \mathbb{DL} . Узел, на которые указывает **TopFromRight**, отмечен стрелкой на рис. 16. Таким образом, **TopFromRight** — это 1-цепь вокруг A (с указателем 3).

Цепи с центром в узлах типа \mathbb{DR} и \mathbb{RD}

На рис. 18 и 19 белыми точками отмечены 1-цепи и 2-цепи узла \mathbb{DR} . 0-цепи с центром в \mathbb{DR} не существует.

Кроме указанных на рисунках цепей первого и второго уровней, существует также 3-цепь, получаемая применением операции разбиения к макроплиткам на рис. 17. Полностью аналогично цепи с центром в узле типа \mathbb{DL} можно показать, что существуют только три возможные конфигурации окружения цепи с центром в узле типа \mathbb{DR} . Кроме того, зная тип ребра t , мы можем выписать полностью всё окружение 1-цепей, 2-цепей, 3-цепей с центром в узле типа \mathbb{DR} .

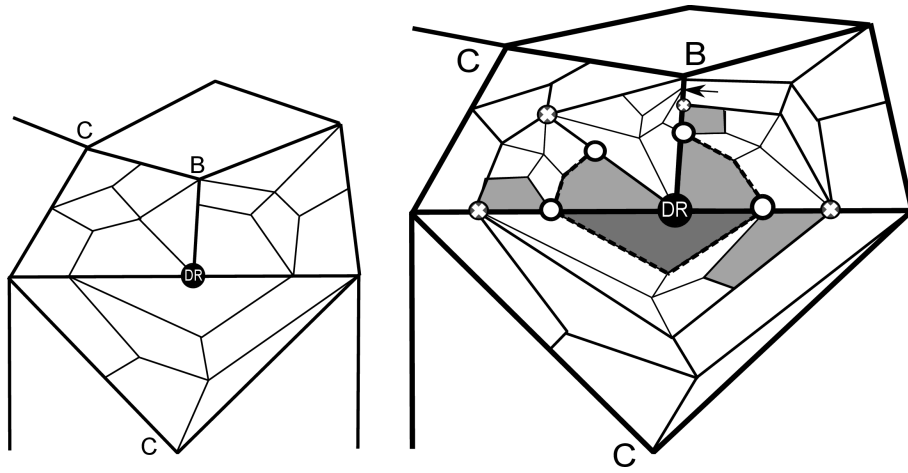


Рис. 18. 1-цепь узла DR

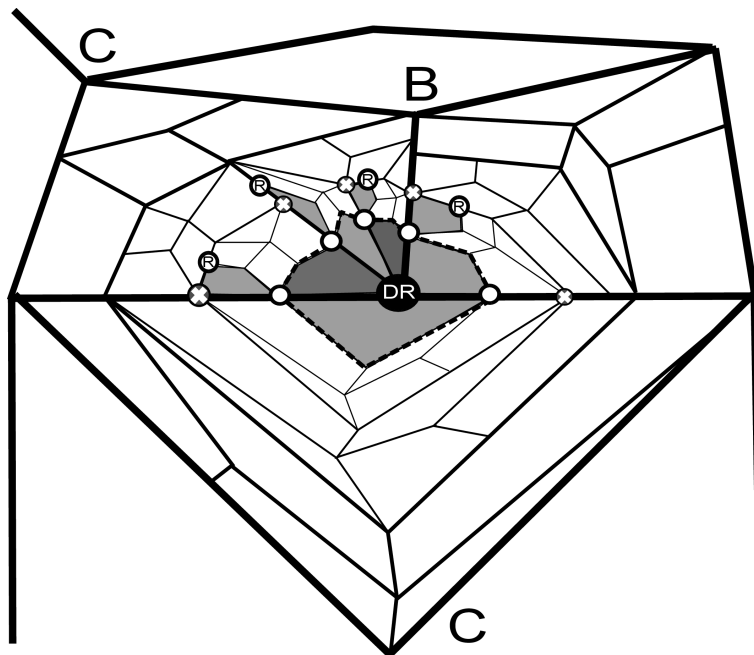


Рис. 19. 2-цепь узла DR

Таким образом, для узлов типа \mathbb{DR} значение функции **LevelPlus** может быть установлено по её аргументу.

На рис. 18 и 19 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 \mathbb{DL} -цепи.

Функции **TopFromRight** и **BottomRightTypeFromRight** могут быть применены только к 1-цепи (в остальные входят только узлы типа \mathbb{UL} и \mathbb{LU}). В случае 1-цепи аргументом может быть только узел, лежащий на ребре \mathbf{r} (относительно узла \mathbb{DR}). Пусть X — этот узел. Тогда **BottomRightType** будет наш же узел \mathbb{DR} . Узел, на которые указывает **TopFromRight**, отмечен стрелкой на рис. 18. Таким образом, **TopFromRight** — это 1-цепь вокруг \mathbb{B} (с указателем 2).

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

С учётом этого замечания для рассмотрения цепей с центрами в вершинах типов \mathbb{U} , \mathbb{L} , \mathbb{R} , \mathbb{D} достаточно применить те же рассуждения, что и для цепей \mathbb{UL} , \mathbb{LD} , \mathbb{RD} , \mathbb{DL} , но рассматривать только половину картинку для каждой цепи. Поскольку рассмотрения полностью аналогичны, мы не будем 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 рис. 20—23 белыми точками отмечены цепи с вершинами в узлах типа \mathbb{CUL} , \mathbb{CUR} , \mathbb{CDR} , \mathbb{CDL} .

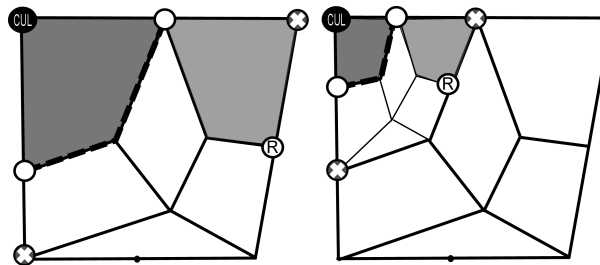


Рис. 20. 0-цепь и 1-цепь узла \mathbb{CUL}

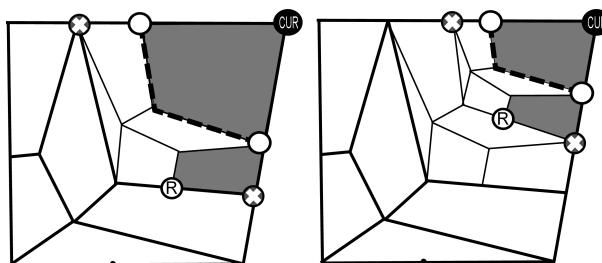


Рис. 21. 1-цепь и 2-цепь узла CUR

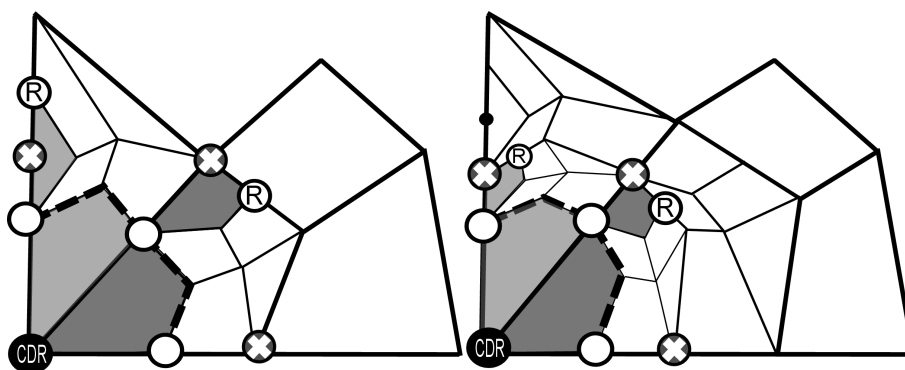


Рис. 22. 1-цепь и 2-цепь узла CDR

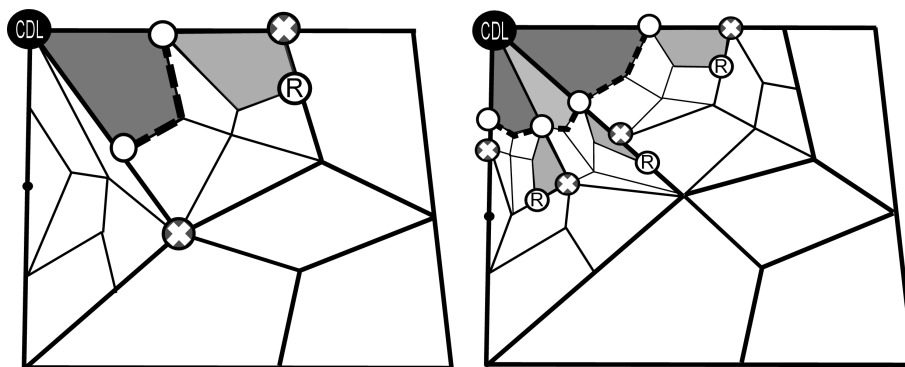


Рис. 23. 1-цепь и 2-цепь узла CDL

Как это было сделано для предыдущих цепей, можно показать, что существует только две возможные конфигурации окружения цепи в каждом из этих четырёх случаев.

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 применены к вершинам данных цепей.

Узлы типа \mathbb{DR} и \mathbb{RD}

Для некоторых функций нужно также рассмотреть ситуации, когда аргументом является узел \mathbb{DR} (или симметричный случай \mathbb{RD}).

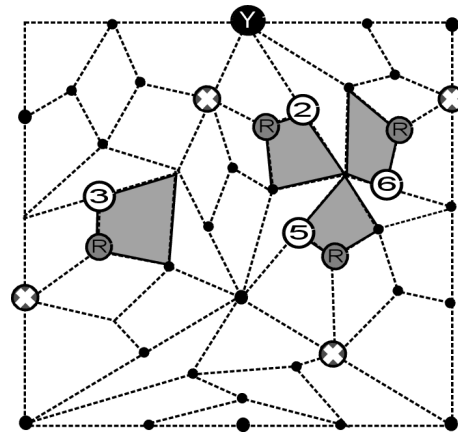


Рис. 24. Случаи расположения \mathbb{DR} -узла

Окружение \mathbb{DR} -узла может быть четырёх видов, по числу внутренних рёбер, на которых он может располагаться. На рис. 24 отмечены эти четыре возможные ситуации. Во всех случаях начальником нашего \mathbb{DR} -узла является вершина Y в середине верхней стороны. На рисунке серым цветом отмечены макроплитки, левый нижний угол которых попадает в \mathbb{DR} -узел. Таким образом, значение функции **RightCorner** попадает в узлы, отмеченные буквой R . Это \mathbb{W} -узел, и его окружение во всех случаях можно вычислить, учитывая, что мы знаем окружение узла в середине верхней стороны.

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

Теперь рассмотрим функцию **TopFromRight**. Вершины, являющиеся её 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 её центром, причём известны окружение X и тип внутреннего ребра, на котором лежит X (один из восьми типов внутренних рёбер), а также известно окружение макроплитки T , которой принадлежит это ребро. Покажем, как восстановить указатель X , т. е. тип ребра входа-выхода для Y .

Почти для всех рёбер указатель, являющийся ребром выхода из Y , устанавливается сразу по типу ребра. Для ребра типа 1 указатель будет очевидно u_2 , для ребра типа 2 — u_1 , для ребра типа 3 — l , для ребра типа 4 ответ зависит от типа Y : для A — указатель 2, для C — 1. Для ребра 5 ответ опять зависит от типа Y : для B — указатель 3, для C — 2. Для ребра 6 указатель будет r .

Пусть ребро имеет тип 7. Зная окружение макроплитки T , мы можем установить её положение в родительской макроплитке. Далее вычисляем значение указателя: для левого верхнего положения T — l_2 , для левого нижнего — ld , для среднего — mid , для правого верхнего — u_3 , для правого нижнего — rd , для нижнего — d .

Пусть ребро имеет тип 8. Зная окружение макроплитки T , мы можем установить её положение в родительской макроплитке. Далее вычисляем значение указателя: для левого верхнего положения T — 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 — l_3 , для левого нижнего — ld_2 , для среднего — mid_2 , для правого верхнего — u_4 , для правого нижнего — r_3 , для нижнего — d_2 .

6.3. Дополнительные функции

Через E_7 и E_8 будем обозначать соответственно код входящего ребра и код выходящего ребра. Очевидно, что по окружению узла в левом нижнем углу можно установить E_7 , а по окружению узла в правом нижнем углу можно установить E_8 (рис. 25).

Аналогично обозначим через E_{ld} , E_{rd} , E_{ur} коды входящих или выходящих рёбер для нижних углов по левой или правой сторонам и для верхнего угла по правой стороне. Через E_{ddl} , E_{ddr} обозначим коды рёбер для нижних углов по нижней стороне слева и справа. Очевидно, что, зная окружение соответствующего узла, можно установить букву, обозначающую нужное ребро (см. рис. 25).

Пусть узел X принадлежит некоторой цепи. Поскольку нам известны окружения всех узлов в цепи, а также типы указателей и порядок их следования, мы можем рассмотреть следующий по часовой стрелке после X узел в цепи. Будем обозначать его $Next(X)$. Аналогично можно определить следующий после X

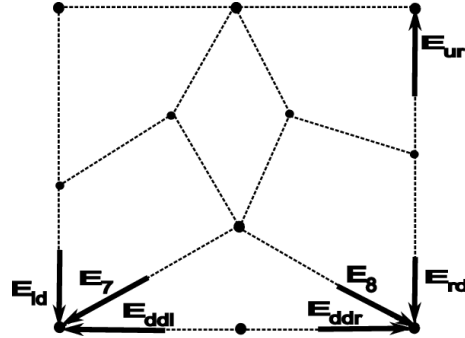


Рис. 25. Функции E_7 , E_8 , E_{1d} , E_{7d} , E_{8d} , E_{ddl} , E_{ddr} , E_{ur} , E_{rd} обозначают входящие или выходящие рёбра для углов

узел против часовой стрелки: $\mathbf{Prev}(X)$. Например, если в некоторой макроплитке известно окружение узла X в середине верхней стороны, то окружение узла в середине левой стороны будет $\mathbf{Next}(X)$.

Также отдельно определим функцию $\mathbf{BottomRightType}$, аргументами которой являются окружение и информация узла X , являющегося 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' . 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 рёбер T можно установить, какое положение занимает уже T' в своей родительской макроплитке T'' .

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

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

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

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

Если узел X в середине верхней стороны макроплитки известен, то функции $\mathbf{BottomRightType}$, $\mathbf{BottomLeftType}$, $\mathbf{TopRightType}$ дают нам типы узлов в трёх соответствующих углах. Учитывая, что мы можем установить рёбра входа в углы (E_{1d} , E_{7d} , E_{8d}), мы можем установить окружения отмеченных на рис. 26

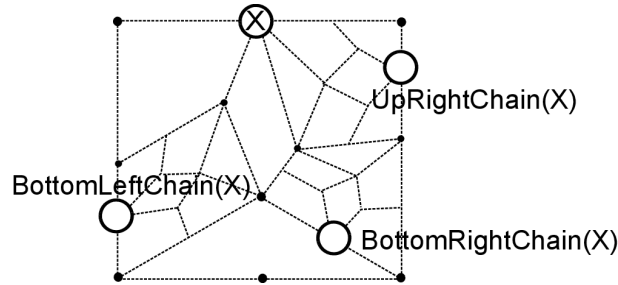


Рис. 26. Узлы, на которые указывают функции **BottomLeftChain**, **BottomRightChain**, **UpRightChain**

трёх узлов. Обозначим соответствующие функции через **BottomLeftChain**, **BottomRightChain**, **UpRightChain**.

Также определим функцию **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 $A0$ -цепи с указателем 1. Типы левой и верхней сторон в T' те же, что и в T , т. е. мы их знаем по окружению X . Для среднего, левого нижнего, правого верхнего, правого нижнего положений значением будет \mathbb{RD} -узел с окружением, соответствующим внутренним рёбрам 3, 2, 6, 5. Для нижнего положения значением будет 1-цепь вокруг второго начальника X (мы знаем его тип) с указателем E_{add} (вход по нижнему ребру в левый нижний угол).

Основным свойством введённой кодировки является детерминированность путей, позволяющая говорить о корректности введённых определяющих соотношений. Для проверки детерминированности нужно разобрать различные случаи расположения пути P . В приложении ниже проводится этот перебор.

7. Приложение: разбор случаев расположения путей

На рис. 27 изображены пары путей. Далее мы покажем, как, зная код одного пути из пары, можно определить код другого пути либо установить, что код является мёртвым. В целом для этого требуется уметь вычислять код любой вершины в любой макроплитке по известным кодам трёх 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 рёбрам входа и выхода. Мы должны установить, к какой из десяти конфигураций относится наш путь, а также показать, как провести локальное преобразование, т. е. получить код другого пути из соответствующей пары.

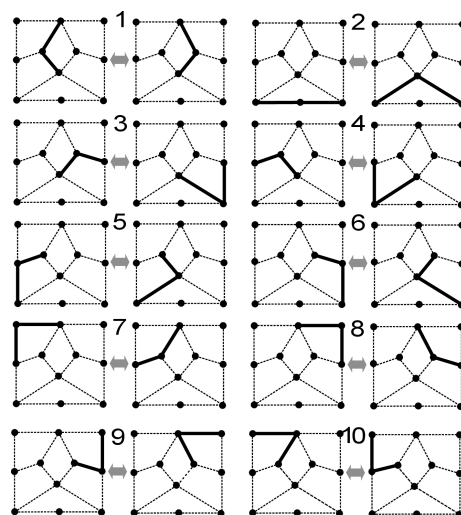


Рис. 27. По известному коду одного пути из пары можно восстановить код другого

В этом разделе мы рассмотрим случай, когда все буквы e_1, e_2, e_3, e_4 отвечают плоским рёбрам, т. е. наш путь не выходит в подклеенные области. Случай выхода в подклеенные области, когда хотя бы одна буква отвечает ребру, выходящему в подклейку, мы разберём в следующем разделе. Фактически в этом и следующем разделах мы вводим определяющие соотношения в полугруппе.

7.1. Обзор перебора случаев: сочетания типов сторон макроплитки

Чтобы провести восстановление вершин для случаев 1-10, требуется знать, что расположено вокруг плитки, в которой проходит наш путь.

Мы рассматриваем путь из двух последовательных рёбер, проходящий внутри либо по границе некоторой макроплитки T . На рис. 27 изображены макроплитки T с проходящими внутри них путями. Основная задача состоит в том, чтобы в каждом случае *восстановить путь*, т. е. по известным значениям параметров трёх вершин и входящим рёбрам вычислить параметры четвёртой вершины и код парного пути.

Для плоского пути, когда среди рёбер вдоль пути не встречаются выходы в подклейку, параметр «флаг подклейки» восстанавливается тривиально: для четвёртой вершины он будет такой же, как и для остальных вершин (среди которых можно выбрать гарантированно лежащую не на границе подклеенной макроплитки). 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 для различных расположений.

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 рёбра, соответствующие названию. Символами E_1, E_{1d}, E_{dd1} (и подобными, см. раздел 6.3) будем обозначать рёбра, получаемые в результате применения указанных функций. Все эти буквы мы будем использовать 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».

Пусть некоторые параметры вершины X зафиксированы и известны. Пусть, оперируя известными параметрами X , мы вычисляем некоторые параметры вершин X_1, X_2, X_3 . *Разрешённой комбинацией* будем называть такую упорядоченную четвёрку букв (x, x_1, x_2, x_3) , что x кодирует X с зафиксированными параметрами, x_i кодирует X_i , причём параметры совпадают с вычисленными. Заметим, что возможно несколько комбинаций разрешённых четвёрок, 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. 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). 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 разрешённых букв, кодирующих J , можно выбрать буквы из множества с фиксированными параметрами типа, базового окружения, флага подклейки и информации, но при этом с разными подклеенными 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.

Комментарий о вкладе авторов

Первому автору (И. А. Иванову-Погодаеву) принадлежит конструкция подстановочной системы семейства комплексов, система кодировки вершин, перебор для проверки детерминированности и финальная схема приведения периодического слова к нулю.

7.2. Локальное преобразование 1

Рассмотрим пару путей на рис. 28.

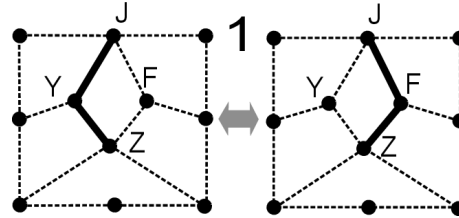


Рис. 28. Локальное преобразование 1

Определяющие соотношения

Назначение Z

Пусть Z — вершина типа C . Зафиксируем её флаг подклейки, базовое окружение, подклеенное 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 . Информация Y и F известна, так как известны окружения J и вершины в правом нижнем углу (третий начальник Z).

Таким образом, выбрав произвольно базовое и подклеенное окружение Z , подклеенное окружение и информацию J , а также флаг макроплитки, мы можем вычислить значение остальных параметров, т. е. мы можем выписать множество разрешённых четвёрок 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, т. е. J могут кодировать разные буквы, отличающиеся информацией.

Для каждой разрешённой комбинации четырёх букв J, Z, Y, F введём следующие определяющие соотношения:

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

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

Итак, Z — это произвольная буква типа \mathbb{C} (с произвольными расширенным окружением, флагом подклейки и информацией), а у J может быть произвольная информация и произвольное расширенное окружение. Остальные буквы в этом соотношении заданы однозначно.

7.3. Локальное преобразование 2

Рассмотрим пару путей на рис. 29.

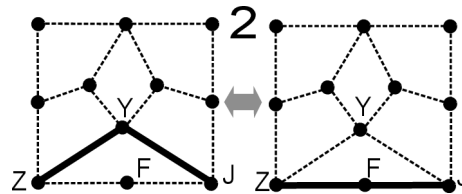


Рис. 29. Локальное преобразование 2

Определяющие соотношения

Мы рассмотрим по очереди шесть случаев расположения. Для каждого случая мы определим разрешённые множества букв, которыми могут быть вершины Z, J, F, Y , после чего для каждой комбинации букв введём два соотношения: одно отвечает прямому обходу данного пути (по стрелкам), а другое — обратному (против стрелок).

1. Левое верхнее расположение (рис. 30, левая часть)

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 рис. 30 (левая часть) чёрными кругами. Заметим, что мы можем вычислить их окружения. Одна из этих вершин 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 четырёх вершин).

Таким образом, можно сформировать множество разрешённых четвёрок букв Z, J, Y, F .

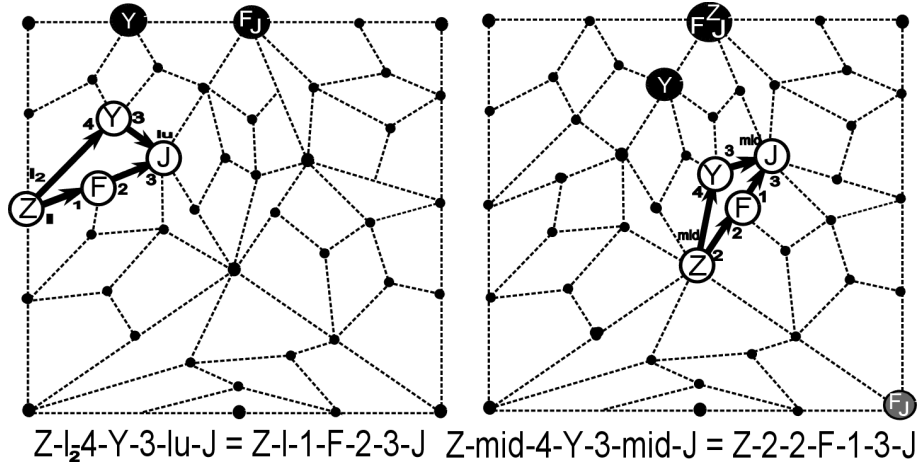


Рис. 30. Левое верхнее и среднее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned}
 Ze_{1_2}e_4Ye_3e_{1u}J &= Ze_{1_1}e_1Fe_2e_3J, \\
 Je_{1u}e_3Ye_4e_{1_2}Z &= Je_3e_2Fe_1e_1Z.
 \end{aligned}$$

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 (рис. 30, правая часть)

Назначение Z

Пусть Z — вершина с типом \mathbb{C} и произвольным расширенным окружением, флагом подклейки и информацией.

Начальники вершин J , Y , F отмечены чёрными кругами. Зная код Z , можно вычислить их окружения, т. е. можно вычислить коды J , Y , F .

Заметим, что если буква Z зафиксирована, то Y , F точно определены, а J определена с точностью до произвольного подклеенного окружения при заданном базовом, т. е. можно выделить множество разрешённых четвёрок букв, кодирующих наши вершины.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Z e_{\text{mid}} e_4 Y e_3 e_{\text{mid}} J &= Z e_2 e_2 F e_1 e_3 J, \\ J e_{\text{mid}} e_3 Y e_4 e_{\text{mid}} Z &= J e_3 e_1 F e_2 e_2 Z. \end{aligned}$$

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

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

3. Правое верхнее расположение (рис. 31, 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), т. е. коды всех остальных вершин можно вычислить.

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

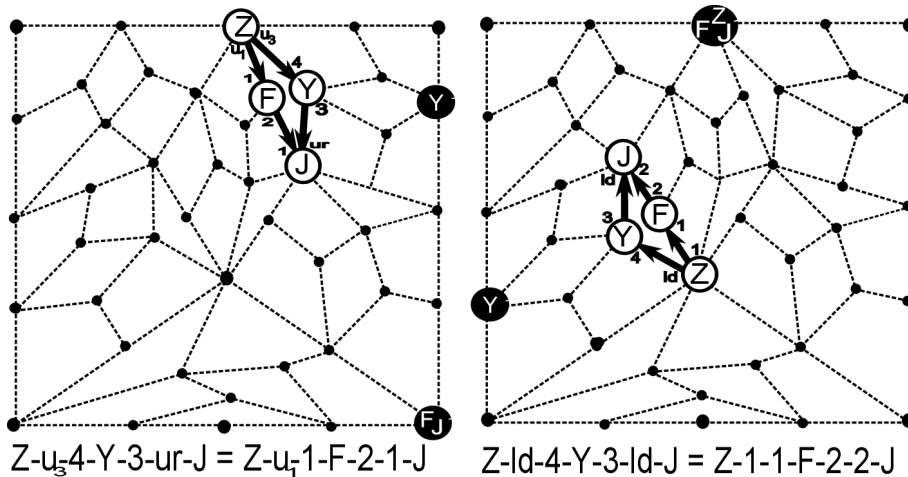


Рис. 31. Правое верхнее и левое нижнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_{\mathbf{u}_3}e_4Ye_3e_{\mathbf{ur}}J &= Ze_{\mathbf{u}_1}e_1Fe_2e_1J, \\ Je_{\mathbf{ur}}e_3Ye_4e_{\mathbf{u}_3}Z &= Je_1e_2Fe_1e_{\mathbf{u}_1}Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

4. Левое нижнее расположение (рис. 31, правая часть)

Назначение Z

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

Цепь вокруг левого нижнего угла T можно получить как $\mathbf{BottomLeftChain}(Z)$, т. е. 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 определена с точностью до произвольного подклеенного окружения при заданном базовом.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

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

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

5. Правое нижнее расположение (рис. 32, 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 $\mathbf{BottomRightChain}(Z)$, т. е. 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 определена с точностью до одного начальника и произвольного подклеенного окружения при заданном базовом.

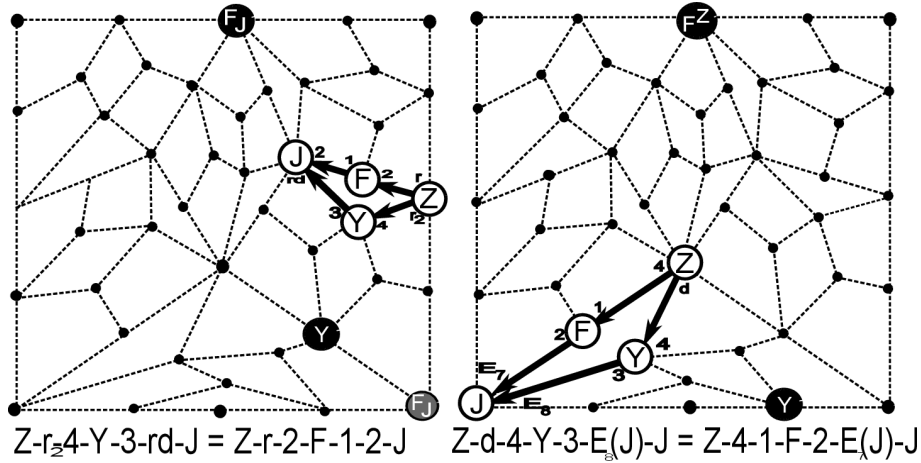


Рис. 32. Правое нижнее и нижнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Z e_{r_2} e_4 Y e_3 e_{rd} J &= Z e_r e_2 F e_1 e_2 J, \\ J e_{rd} e_3 Y e_4 e_{r_2} Z &= J e_2 e_1 F e_2 e_r Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

6. Нижнее расположение (рис. 32, 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.

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

$E_7(J)$ и $E_8(J)$ — рёберные буквы, полученные с помощью применения функций E_7 и E_8 к узлу J , т. е. это тоже буквы, кодирующие рёбра входов и выходов, но в зависимости от узла J .

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_d e_4 Y e_3 E_8(J) J &= Ze_4 e_1 F e_2 E_7(J) J, \\ JE_8(J) e_3 Y e_4 e_d Z &= JE_7(J) e_2 F e_1 e_4 Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

7.4. Локальное преобразование 3

Рассмотрим пару путей на рисунке 33.

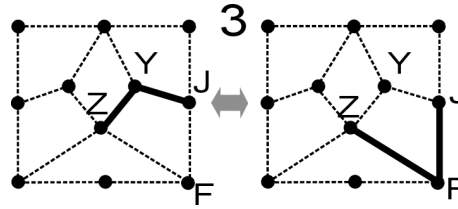


Рис. 33. Локальное преобразование 3

1. Левое верхнее расположение (рис. 34, левая часть)

Назначение F

Пусть F — вершина типа \mathbb{A} с произвольным базовым окружением, флагом подклейки и информацией. Как и в предыдущих случаях, этого достаточно, чтобы вычислить коды других вершин.

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

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_2 e_3 Y e_2 e_r J &= Ze_3 e_{1u} F e_1 e_1 J, \\ Je_r e_2 Y e_3 e_2 Z &= Je_1 e_1 F e_{1u} e_3 Z. \end{aligned}$$

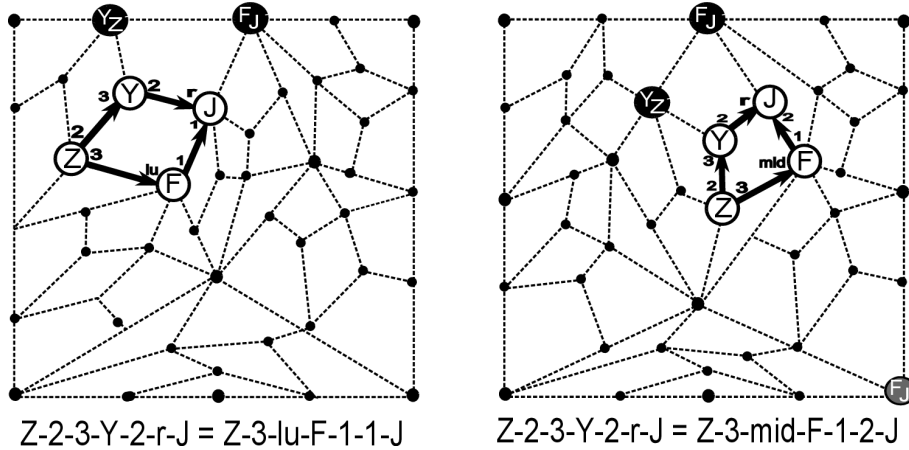


Рис. 34. Левое верхнее и среднее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

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 .

2. Среднее расположение (рис. 34, 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 разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned}
 Ze_2e_3Ye_2e_rJ &= Ze_3e_{mid}Fe_1e_2J, \\
 Je_re_2Ye_3e_2Z &= Je_2e_1Fe_{mid}e_3Z.
 \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

3. Левое нижнее расположение (рис. 35, 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 разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Z e_2 e_3 Y e_2 e_r J &= Z e_3 e_{1d} F e_3 e_2 J, \\ J e_r e_2 Y e_3 e_2 Z &= J e_2 e_3 F e_{1d} e_3 Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

4. Правое верхнее расположение (рис. 35, правая часть)

Назначение F

Пусть F — вершина типа \mathbb{B} 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

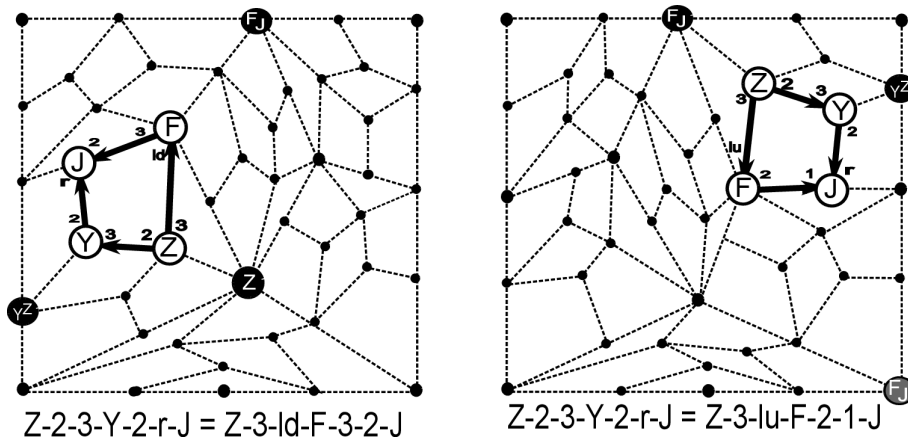


Рис. 35. Левое-нижнее и правое-верхнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

вершин (J и Z — с точностью до произвольного подклеенного окружения при заданном базовом).

Для каждой разрешённой комбинации заданных букв введём 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

Рассуждения аналогичны случаю левого верхнего расположения.

5. Правое нижнее расположение (рис. 36, 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 разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_3Ye_2e_rJ &= Ze_3e_{rd}Fe_3e_1J, \\ Je_re_2Ye_3e_2Z &= Je_1e_3Fe_{rd}e_3Z. \end{aligned}$$

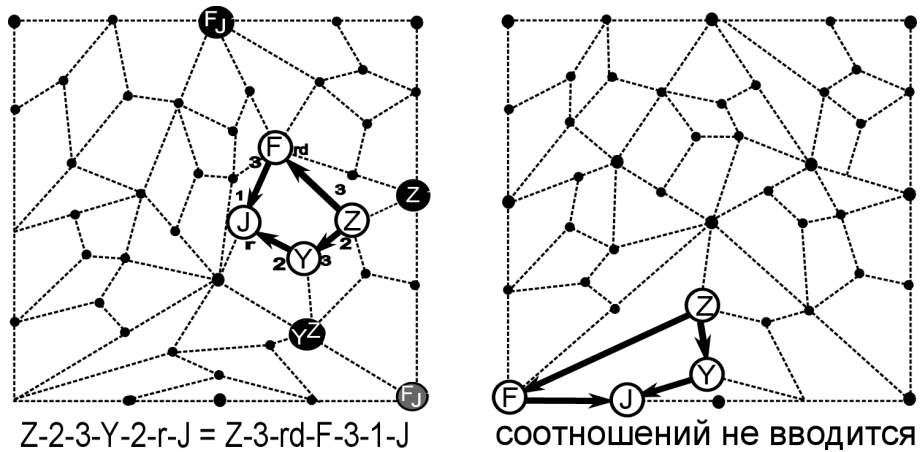


Рис. 36. Правое нижнее и нижнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

6. Нижнее расположение (рис. 36, правая часть)

В этом случае соотношений мы не вводим, так как расположение пути удовлетворяет признакам мёртвого паттерна, т. е. участок пути с данным кодом не может являться подпутём достаточно длинного ненулевого пути.

7.5. Локальное преобразование 4

Рассмотрим пару путей на рисунке 37.

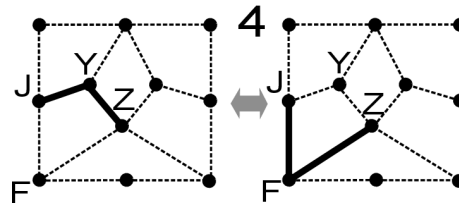


Рис. 37. Локальное преобразование 4

1. Левое верхнее расположение (рис. 38, 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 \mathbb{L} , \mathbb{LD} или \mathbb{LU} , лежащую в цепи вокруг T .

Заметим, что, учитывая расположение на рисунке, по известному коду F мы можем вычислить окружения и начальников остальных трёх вершин, т. е. мы можем выписать все комбинации четырёх букв Z, Y, J, F , такие что их окружения и начальники соответствуют окружению и начальникам изображённых вершин. Будем называть такие комбинации *разрешёнными*. 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 подклеенными окружениями.

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

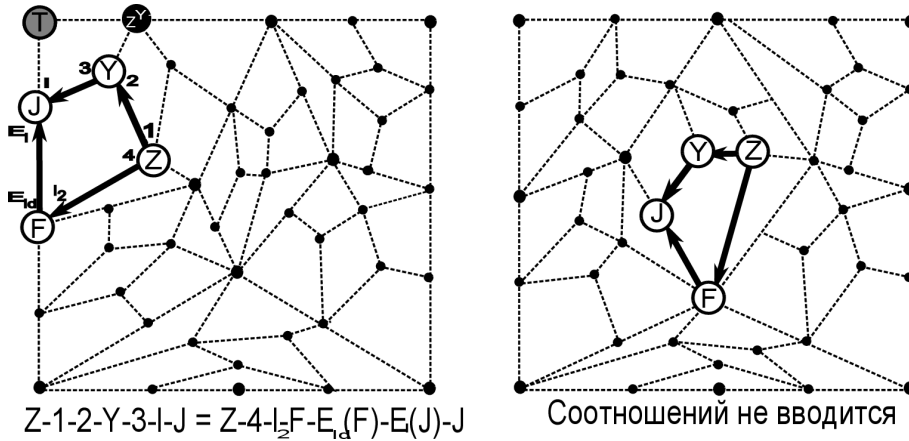


Рис. 38. Левое верхнее и среднее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер. В ситуации справа соотношений не вводится

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$Ze_1e_2Ye_3e_1J = Ze_4e_1_2FE_{1d}E_1J,$$

$$Je_1e_3Ye_2e_1Z = JE_1E_{1d}Fe_{1_2}e_4Z.$$

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 .

2. Среднее расположение (рис. 38, правая часть)

В этом случае соотношений мы не вводим, так как расположение пути удовлетворяет признакам мёртвого паттерна, т. е. участок пути с данным кодом не может являться подпутём достаточно длинного ненулевого пути.

3. Левое нижнее расположение (рис. 39, левая 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников F . Таким образом, мы можем аналогично определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие

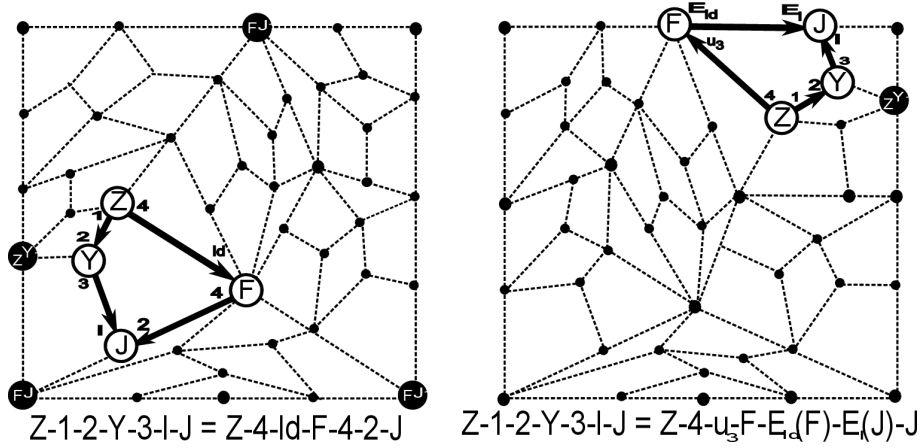


Рис. 39. Левое нижнее и правое верхнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём 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

Рассуждения аналогичны случаю левого верхнего расположения.

4. Правое верхнее расположение (рис. 39, правая часть)

Назначение F

Пусть F — буква, кодирующая вершину типа U , UL или UR . Тогда базовые окружения вершин Z, Y, J легко вычисляются, поскольку 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 определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие 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 рёбра входа и выхода на рисунке) легко вычисляются по известным кодам J и F .

Для каждой разрешённой комбинации заданных букв введём 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

Рассуждения аналогичны случаю левого верхнего расположения.

5. Правое нижнее расположение (рис. 40, левая часть)

Назначение F

Пусть F — буква, кодирующая вершину типа \mathbb{R} , \mathbb{RD} или \mathbb{RU} с произвольным окружением и начальниками. Тогда базовые окружения вершин Z , Y , J легко вычисляются. Начальники всех вершин отмечены чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников F . Таким образом, мы можем определить все разрешённые четвёрки букв Z , Y , J , F , соответствующие 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 рёбра входа и выхода на рисунке) легко вычисляются по известным кодам J и F .

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

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

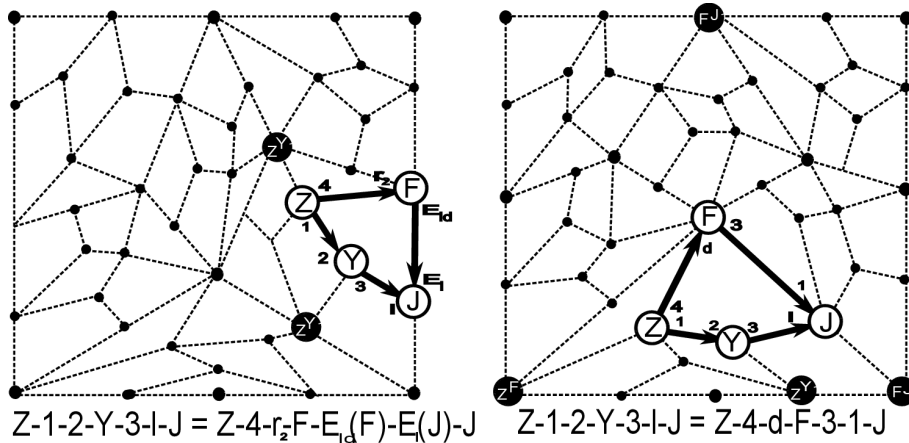


Рис. 40. Правое нижнее и нижнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

6. Нижнее расположение (рис. 40, 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников F . Таким образом, мы можем аналогично определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_1e_2Ye_3e_1J &= Ze_4e_dFe_3e_1J, \\ Je_1e_3Ye_2e_1Z &= Je_1e_3Fe_de_4Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

7.6. Локальное преобразование 5

Рассмотрим пару путей на рисунке 41.

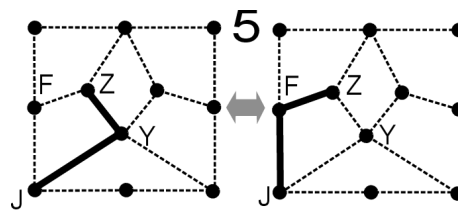


Рис. 41. Локальное преобразование 5

1. Левое верхнее расположение (рис. 42, 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 .

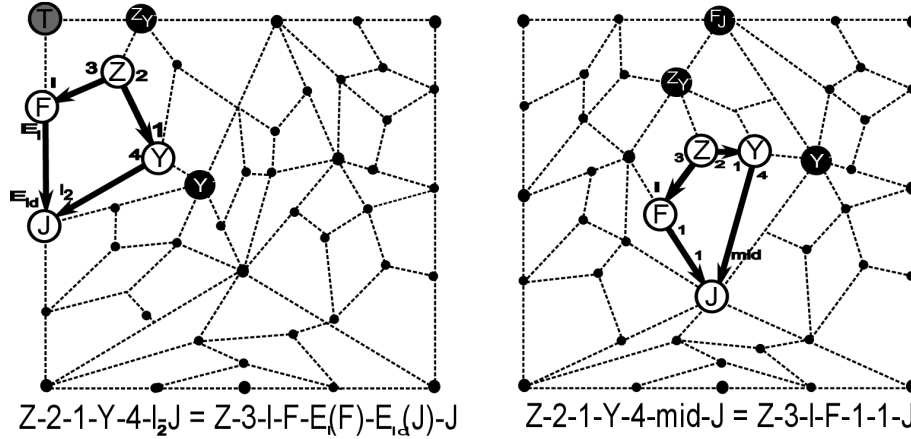


Рис. 42. Левое верхнее и среднее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Назначение J

Пусть J — буква, кодирующая вершину типа \mathbb{L} , \mathbb{LD} или \mathbb{LU} , лежащую в цепи вокруг T .

Аналогично предыдущему случаю по известному коду J мы можем вычислить окружения и начальники остальных трёх вершин, т. е. мы можем выписать все разрешённые комбинации четырёх букв Z, Y, J, F (такие, что их окружения и начальники соответствуют окружению и начальникам изображённых 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 подклеенными окружениями.

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

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_{1_2}J &= Ze_3e_1FE_1E_{1d}J, \\ Je_{1_2}e_4Ye_1e_2Z &= JE_{1d}E_1Fe_1e_3Z. \end{aligned}$$

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 (рис. 42, правая 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_{\text{mid}}J &= Ze_3e_1Fe_1e_1J, \\ Je_{\text{mid}}e_4Ye_1e_2Z &= Je_1e_1Fe_1e_3Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

3. Левое нижнее расположение (рис. 43, 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_2e_1Ye_4e_{\text{ld}}J &= Ze_3e_1Fe_2e_4J, \\ Je_{\text{ld}}e_4Ye_1e_2Z &= Je_4e_2Fe_1e_3Z. \end{aligned}$$

Восстановление кода

Рассуждения аналогичны случаю левого верхнего расположения.

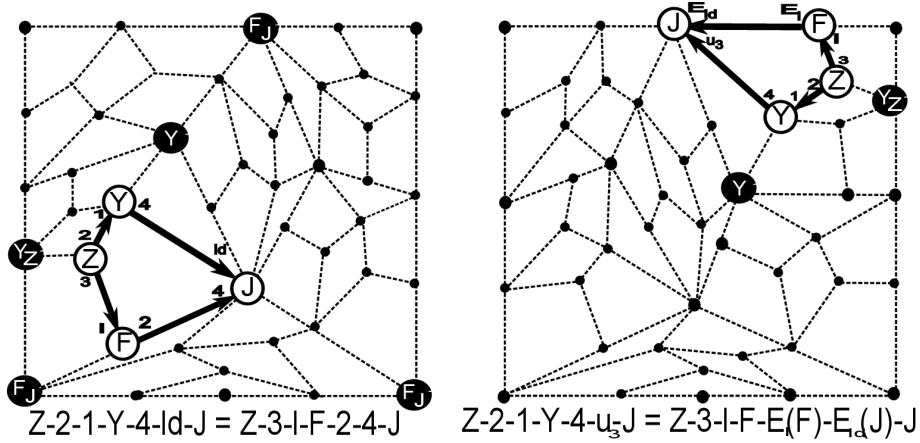


Рис. 43. Левое нижнее и правое верхнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

4. Правое верхнее расположение (рис. 43, правая часть)

Назначение J

Пусть J — буква, кодирующая вершину типа U , UL или UR . Тогда базовые окружения вершин Z , Y , J легко вычисляются, поскольку 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 определить все разрешённые четвёрки букв Z , Y , J , F , соответствующие 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 рёбра входа и выхода на рисунке) легко вычисляются по известным кодам J и F .

Для каждой разрешённой комбинации заданных букв введём 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

Рассуждения аналогичны случаю левого верхнего расположения.

5. Правое нижнее расположение (рис. 44, 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 очевидны. Начальники всех вершин отмечены чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z , Y , J , F , соответствующие 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 рёбра входа и выхода на рисунке) легко вычисляются по известным кодам J и F .

Для каждой разрешённой комбинации заданных букв введём 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

Рассуждения аналогичны случаю левого верхнего расположения.

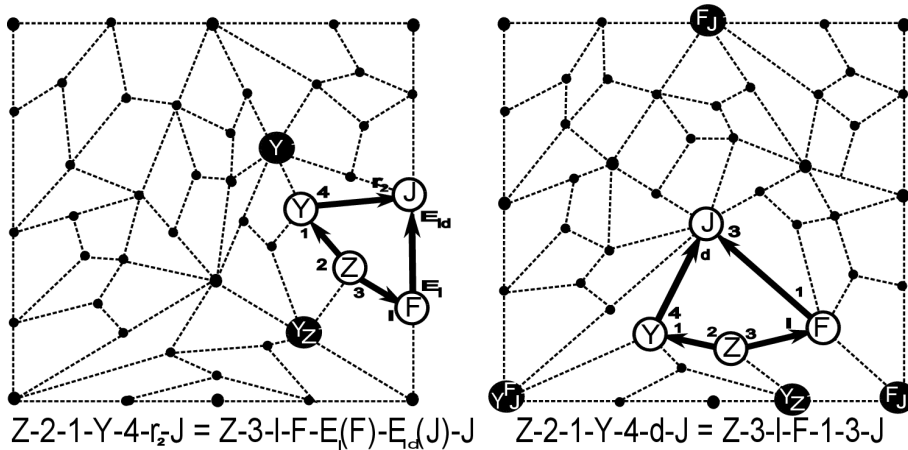


Рис. 44. Правое нижнее и нижнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

6. Нижнее расположение (рис. 44, правая часть)

Назначение J

Пусть J — буква, кодирующая вершину типа \mathbb{C} с произвольным окружением и начальниками. Тогда базовые окружения вершин Z, Y, F очевидны. Начальники всех вершин отмечены чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём 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

Рассуждения аналогичны случаю левого верхнего расположения.

7.7. Локальное преобразование 6

Рассмотрим пару путей на рисунке 45.

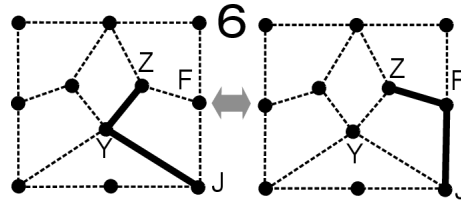


Рис. 45. Локальное преобразование 6

1. Левое верхнее расположение (рис. 46, 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 разрешённые комбинации четырёх букв Z, Y, J, F (такие, что их окружения и начальники соответствуют окружению и начальникам изображённых на рисунке 46 вершин). При этом буквы Y и F будут определены однозначно, а Z и J могут быть любыми

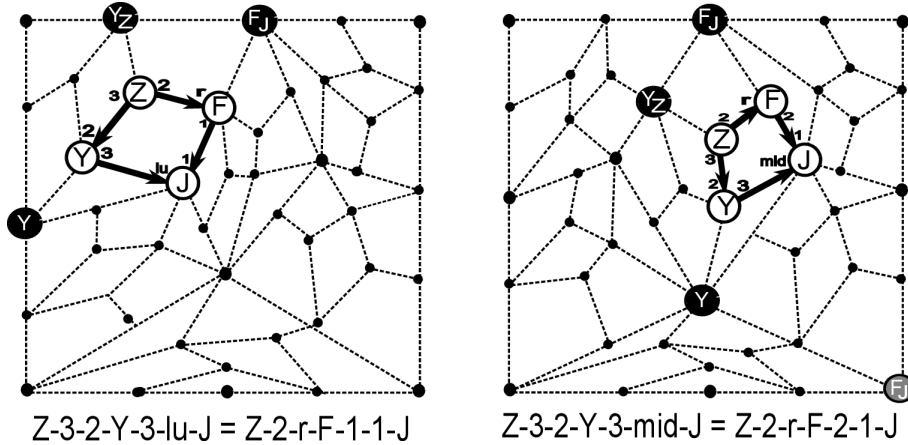


Рис. 46. Левое верхнее и среднее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

буквами с заданным базовым окружением и начальниками, но с различными подклеенными окружениями.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned}
 Ze_3e_2Ye_3e_{lu}J &= Ze_2e_rFe_1e_1J, \\
 Je_{lu}e_3Ye_2e_3Z &= Je_1e_1Fe_re_2Z.
 \end{aligned}$$

Восстановление кода

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

2. Среднее расположение (рис. 46, 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z , Y , J , F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_3e_2Ye_3e_{\text{mid}}J &= Ze_2e_rFe_2e_1J, \\ Je_{\text{mid}}e_3Ye_2e_3Z &= Je_1e_2Fe_re_2Z. \end{aligned}$$

Восстановление кода

Зная коды J и Z , можно вычислить код как Y , так и F . Окружение и первый начальник у Y 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 (рис. 47, 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем код J . Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_3e_2Ye_3e_{\text{ld}}J &= Ze_2e_rFe_2e_3J, \\ Je_{\text{ld}}e_3Ye_2e_3Z &= Je_3e_2Fe_re_2Z. \end{aligned}$$

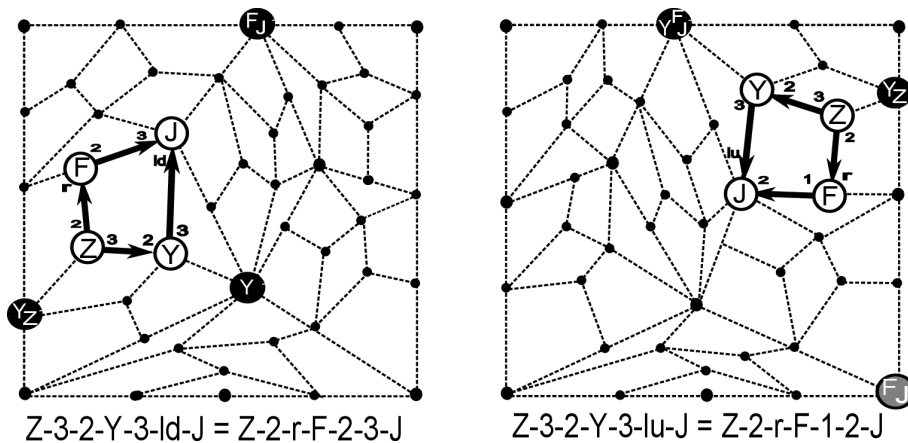


Рис. 47. Левое нижнее и правое верхнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

Восстановление кода

Зная коды J и Z , можно вычислить код как Y , так и F . Окружение и первый начальник у Y 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 .

4. Правое верхнее расположение (рис. 47, 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 $\mathbf{UpRightChain}(F)$ и узнать тип вершины в правом верхнем углу, т. е. начальники всех вершин также вычисляются. Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_3e_2Ye_3e_{1u}J &= Ze_2e_rFe_1e_2J, \\ Je_{1u}e_3Ye_2e_3Z &= Je_2e_1Fe_re_2Z. \end{aligned}$$

Восстановление кода

Зная коды J и Z , можно вычислить код как Y , так и F . Окружение и первый начальник у Y 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 (рис. 48, левая 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 чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned} Ze_3e_2Ye_3e_{rd}J &= Ze_2e_rFe_1e_3J, \\ Je_{rd}e_3Ye_2e_3Z &= Je_3e_1Fe_re_2Z. \end{aligned}$$

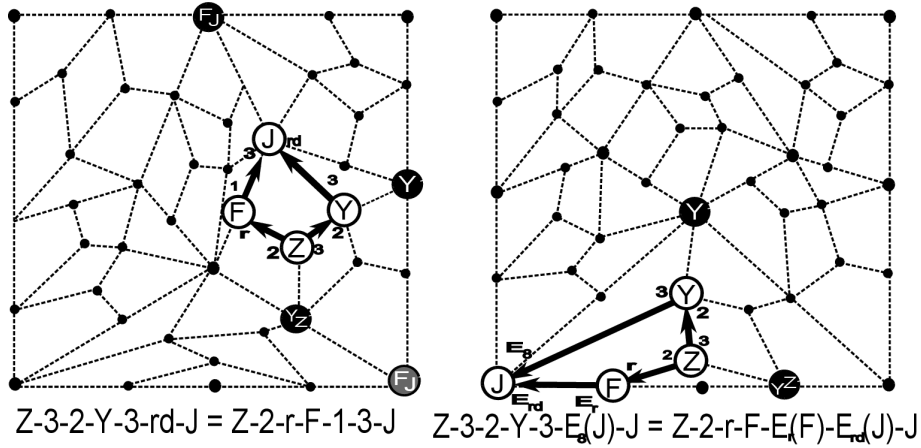


Рис. 48. Правое нижнее и нижнее положения. У вершин при входе и выходе указаны коды входящих и выходящих рёбер

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 **RightFromB**(J), а третий — вершина J . Окружение F вычисляется по известному окружению J , начальники те же, что и у J .

6. Нижнее расположение (рис. 48, правая часть)

Назначение J

Пусть J — буква, кодирующая вершину типа \mathbb{C} с произвольным окружением и начальниками. Тогда базовые окружения вершин Z, Y, F очевидны. Начальники всех вершин отмечены чёрными кругами, и все они очевидно вычисляются, если мы знаем начальников J . Таким образом, мы можем определить все разрешённые четвёрки букв Z, Y, J, F , соответствующие вычисленным значениям окружений и начальников. У вершин Z и J могут быть разные подклеенные окружения.

Для каждой разрешённой комбинации заданных букв введём следующие определяющие соотношения:

$$\begin{aligned}
 Ze_2e_1Ye_4e_dJ &= Ze_3e_1Fe_1e_3J, \\
 Je_d e_4Ye_1e_2Z &= Je_3e_1Fe_1e_3Z.
 \end{aligned}$$

Восстановление кода

Зная коды J и Z , можно вычислить код как Y , так и F . Окружение и первый начальник у Y 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 $E_{ddr}(\mathbf{TBoss}(J))$. Начальники те же, что и у J .

7.8. 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, поэтому может быть три варианта цепи вокруг центра, который может иметь следующий тип: C, B, A, UL/UL, UR/RU, DL/LD, DR/RD, U, L, D, R, CUL, CUR, CDL, CDR. Мы разберём все эти случаи и введём соотношения в соответствии с устройством всех возможных цепей.

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

7.9. Случай цепи C1; преобразования 8 и 9

В правой части рис. 49 изображены локальные преобразования 8 и 9. Обозначим через T макроплитку, в которой проходят пути. Вершина Z попадает в середину верхней стороны T .

Итак, пусть середина верхней стороны T входит в C1-цепь. Различные случаи расположения интересующих нас путей показаны в левой части рис. 49.

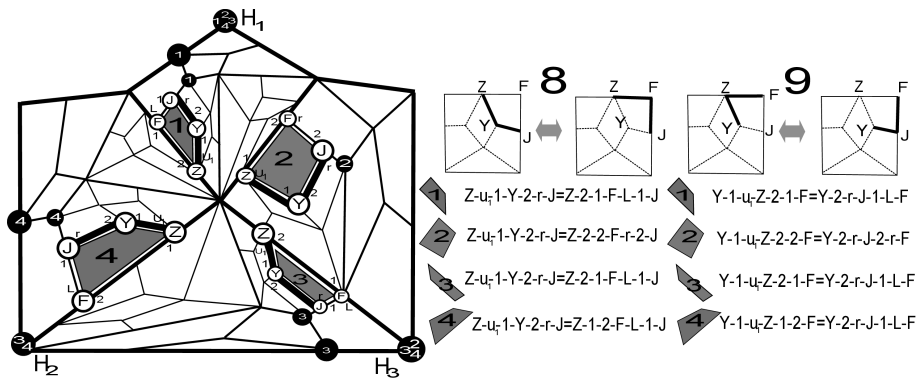


Рис. 49. Случаи расположения пути вокруг C1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Сначала введём определяющие соотношения для данного случая. Зафиксируем \mathcal{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 четырёх 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 \mathcal{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 четырёх случаях расположения, что видно на рис. 49. Например, окружение F в случае 4 соответствует 1-цепи вокруг H_2 с указателем $E_7(H_2)$ и уровень F равен 2.

На рис. 49 чёрными кругами отмечены вершины, являющиеся начальниками хотя бы одной из вершин Z, Y, F, J в каждом из четырёх случаев, число в круге обозначает номер случая. Окружение всех вершин, отмеченных чёрными кругами, также можно выписать, зная начальников центральной \mathcal{C} -вершины, т. е. мы можем вычислить начальников каждой из четырёх вершин во всех случаях.

Уровни вершин F во всех случаях равны 2, а уровни 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 всё наоборот.

Соотношения для преобразования 8

Назначение Z

Для каждого из четырёх случаев расположения обозначим через Z буквы в алфавите, соответствующие кодам вершин с заданными типом, уровнем, базовым окружением и информацией, т. е. эти буквы отличаются друг от друга только различными подклеенными 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 подклеенным 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 подклеенном

окружении. Мы это делаем, так как путь проходит через F и Y по плоским рёбрам.

Теперь для каждого случая и для каждого разрешённого набора букв Z , F , Y , J выпишем соотношение, представленное в правой части рис. 49. Буквы Z , F , Y , J мы только что определили, а остальные символы в соотношении обозначают входящие и выходящие рёбра.

Соотношения для преобразования 9

Буквы Z и J определены однозначно, а Y и F — с точностью для подклеенного окружения. Рассуждения аналогичны рассмотренному случаю.

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} , т. е. число сочетаний «тип, уровень, окружение, информация».

Характеризация

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

На рис. 49 отмечены входящие и выходящие рёбра в каждом из четырёх случаев. Исходя из этого легко выписать свойства пути, позволяющие нам установить, с каким случаем мы имеем дело. Всё показано в табл. 1 и 2.

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

Условие на буквы	Симметричное условие (проход в обратном порядке)	Случай	Локальное преобразование
$\text{Surr}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = u_112r$	$\text{Surr}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = r21u_1$	1	8 (левая)
$\text{Surr}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 2111$	$\text{Surr}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 1112$	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)

Таблица 2. Характеристические условия на принадлежность пути к случаю C1 цепи

Условие на буквы	Симметричное условие (проход в обратном порядке)	Случай	Локальное преобразование
$\text{Type}(X_1) = \mathbb{B},$ $\text{FBoss}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 1u_121$	$\text{Type}(X_3) = \mathbb{B},$ $\text{FBoss}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 12u_11$	1	9 (левая)
$\text{Type}(X_1) = \mathbb{B},$ $\text{FBoss}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 2r1l$	$\text{Type}(X_3) = \mathbb{B},$ $\text{FBoss}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 1l1r2$	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)
$\text{Type}(X_1) = \mathbb{B},$ $\text{FBoss}(X_1) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 2r1l$	$\text{Type}(X_3) = \mathbb{B},$ $\text{FBoss}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 1l1r2$	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)

$\mathbf{Type}(X_1) = \mathbb{B},$ $\mathbf{FBoss}(X_1) = \mathbb{C}14,$ $e_1 e_2 e_3 e_4 = 2r1l$	$\mathbf{Type}(X_3) = \mathbb{B},$ $\mathbf{FBoss}(X_3) = \mathbb{C}14,$ $e_1 e_2 e_3 e_4 = l1r2$	4	9 (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.

Таким образом, 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 четырёх. Допустим, мы знаем коды трёх из четырёх вершин (из числа Z, Y, J, F). Тогда, учитывая характеристику, мы знаем, с каким из случаев расположения мы имеем дело. 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 — второго и третьего начальников 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 виден из расположения. В случаях расположений 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, $\mathbf{FBoss}(Z)$. Тип второго начальника J — это \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.

Пусть слово W подходит под один из перечисленных случаев. Тогда мы знаем, с каким случаем имеем дело, а также какую конфигурацию имеет путь. Слово W должно кодировать этот путь, т. е. 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 введённое нами соотношение из числа указанных справа на рис. 49 и получить код для другой части

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

7.10. Случай цепи C1; преобразования 7 и 10

В правой части рис. 50 изображены локальные преобразования 7 и 10.

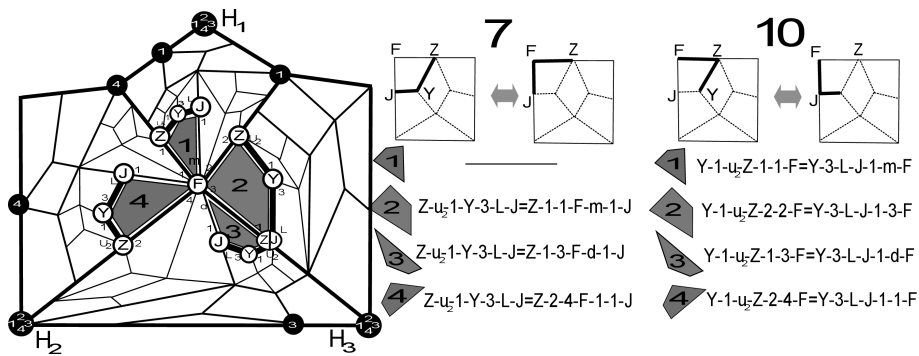


Рис. 50. Случаи расположения пути вокруг C1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Итак, зафиксируем C-узел с некоторым окружением и начальниками. Различные случаи расположения интересующих нас путей показаны в левой части рис. 50.

Зная окружение и начальников центрального C-узла, можно вычислить базовые окружения и начальников вершин Z, Y, F, J во всех случаях расположения. Таким образом, мы можем определить коды вершин Z, Y, F, J во всех четырёх случаях расположения и выписать определяющие соотношения для локальных преобразований 7 и 10.

Уровни боковых вершин (т. е. J и Z) во всех случаях равны 1.

Соотношения для преобразования 7

Назначение Z

Для каждого из четырёх случаев расположения обозначим через Z буквы в алфавите, соответствующие кодам вершин Z с заданными типом, уровнем, базовым окружением и информацией, эти буквы отличаются друг от друга только различными подклеенными 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 подклеенным 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 подклеенном окружении. Мы это делаем, так как путь проходит через F и Y по плоским рёбрам.

Теперь для каждого случая и для каждого разрешённого набора букв Z , F , Y , J выпишем соотношение, представленное в правой части рис. 50. Буквы Z , F , Y , J мы только что определили, а остальные символы в соотношении обозначают входящие и выходящие рёбра.

Соотношения для преобразования 10

Буквы Z и J определены однозначно, а Y и F — с точностью для подклеенного окружения. Рассуждения аналогичны рассмотренному случаю.

Оценка числа соотношений

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

В последующих случаях отношения считаются аналогично.

Характеризация

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

На рис. 50 отмечены входящие и выходящие ребра в каждом из четырёх случаев. Исходя из этого легко выписать свойства пути, позволяющие нам установить, с каким случаем мы имеем дело. Всё показано в табл. 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 преобразование
$\mathbf{Surr}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = u_1l3l$	$\mathbf{Surr}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = l31u_1$	1	7 (левая)
$\mathbf{Surr}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 11m1$	$\mathbf{Surr}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 1m11$	1	7 (правая)
$\mathbf{Surr}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = u_2l3l$	$\mathbf{Surr}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = l31u_2$	2	7 (левая)
$\mathbf{Surr}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 2231$	$\mathbf{Surr}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 1322$	2	7 (правая)

$\mathbf{Surr}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = u_213l$	$\mathbf{Surr}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = l31u_2$	3	7 (левая)
$\mathbf{Surr}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 13d1$	$\mathbf{Surr}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 1d31$	3	7 (правая)
$\mathbf{Surr}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = u_213l$	$\mathbf{Surr}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = l31u_2$	4	7 (левая)
$\mathbf{Surr}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 2411$	$\mathbf{Surr}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 1142$	4	7 (правая)

Таблица 4. Характеристические условия на принадлежность пути к случаю C1 цепи

Условие на буквы	Симметричное условие (проход в обратном порядке)	Случай	Локальное преобразование
$\mathbf{Type}(X_1) = \mathbb{B},$ $\mathbf{FBoss}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 1u_111$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 11u_11$	1	10 (левая)
$\mathbf{Type}(X_1) = \mathbb{A},$ $\mathbf{FBoss}(X_1) = \mathbb{C}11,$ $e_1e_2e_3e_4 = 3l1m$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}11,$ $e_1e_2e_3e_4 = m1l3$	1	10 (правая)
$\mathbf{Type}(X_1) = \mathbb{A},$ $\mathbf{FBoss}(X_1) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 1u_222$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 22u_21$	2	10 (левая)
$\mathbf{Type}(X_1) = \mathbb{A},$ $\mathbf{FBoss}(X_1) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 3l13$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}12,$ $e_1e_2e_3e_4 = 31l3$	2	10 (правая)
$\mathbf{Type}(X_1) = \mathbb{A},$ $\mathbf{FBoss}(X_1) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 1u_213$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 31u_21$	3	10 (левая)
$\mathbf{Type}(X_1) = \mathbb{A},$ $\mathbf{FBoss}(X_1) = \mathbb{C}13,$ $e_1e_2e_3e_4 = 3l1d$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}13,$ $e_1e_2e_3e_4 = d1l3$	3	10 (правая)
$\mathbf{Type}(X_1) = \mathbb{A},$ $\mathbf{FBoss}(X_1) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 1u_224$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 42u_21$	4	10 (левая)
$\mathbf{Type}(X_1) = \mathbb{B},$ $\mathbf{FBoss}(X_1) = \mathbb{C}14,$ $e_1e_2e_3e_4 = 3l1ld$	$\mathbf{Type}(X_3) = \mathbb{A},$ $\mathbf{FBoss}(X_3) = \mathbb{C}14,$ $e_1e_2e_3e_4 = ld1l3$	4	10 (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.

Таким образом, 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

Зная код F , можно легко вычислить все остальные коды. Значит, остаётся разобраться, как определить код F по известным кодам остальных вершин в локальном преобразовании 7 для случаев расположения 2, 3, 4. Окружение F совпадает с U -частью окружения H_1 , первого начальника Z . Заметим, что Z в случаях 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 случаю мы можем определить, с каким случаем мы имеем дело, а также какую конфигурацию имеет путь. Слово W должно кодировать этот путь, т. е. W должно входить в Q . В противном случае W приводится к нулю.

Если же W входит в Q , то к W можно применить введённое нами соотношение из числа указанных справа на рис. 50 и получить код для другой части локального преобразования.

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

7.11. Случай цепи $\mathbb{C}2$; преобразования 8 и 9

В правой части рис. 51 изображены локальные преобразования 8 и 9.

Аналогично предыдущим случаям мы можем ввести определяющие соотношения. Фиксируем \mathbb{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 подклеенных окружений крайних вершин в пути).

Чёрными кругами отметим вершины, являющиеся начальниками вершин Z , Y , F , J . Зная код центрального \mathbb{C} -узла, мы можем найти их типы, уровни и окружения, т. е. 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 рис. 51.

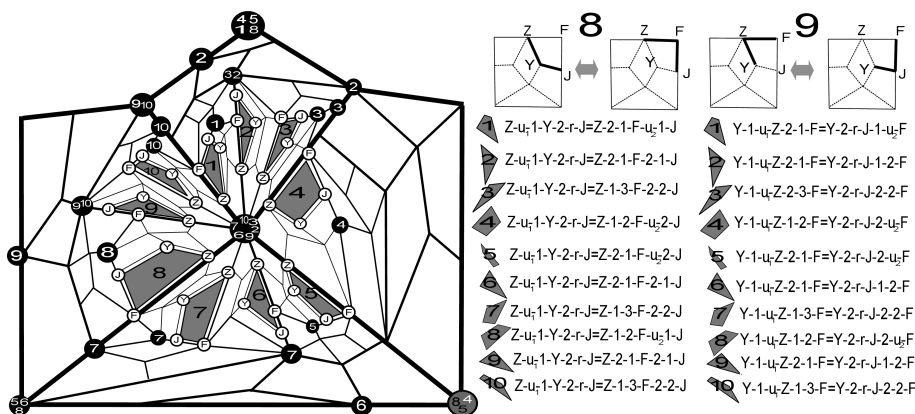


Рис. 51. Случаи расположения пути вокруг C2-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация

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

Мы не будем приводить здесь таблицу, как для случая C1-цепи, её можно составить полностью аналогично, опираясь на рис. 51.

Восстановление кода

Учитывая характеристику, мы знаем, с каким случаем расположения имеем дело. Кроме того, ясно, что окружение каждой из четырёх 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. Остаётся 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. 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 введённое нами соотношение из числа указанных справа на рис. 51 и получить код для другой части локального преобразования.

7.12. Случай цепи C2; преобразования 7 и 10

В правой части рис. 52 изображены локальные преобразования 7 и 10.

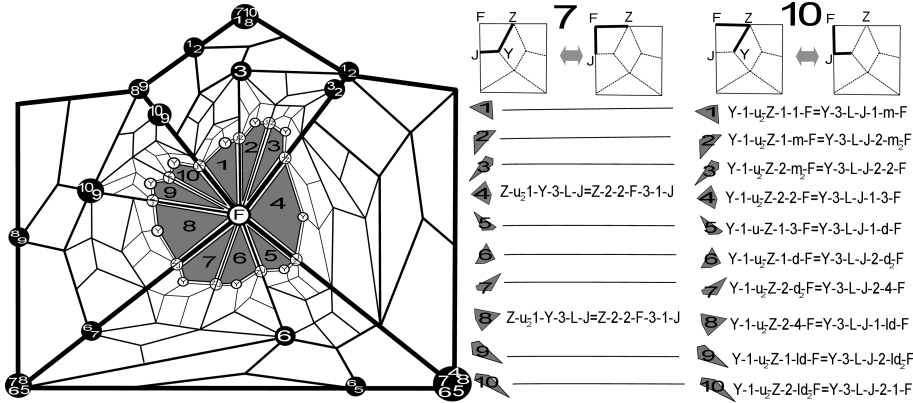


Рис. 52. Случаи расположения пути вокруг C2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Чёрными кругами отметим вершины, являющиеся начальниками вершин Z , Y , F , J . Заметим, что, зная код центрального C-узла, мы можем найти их окружения, т. е. коды вершин Z , Y , F , J (включая типы, уровни, окружения и информацию), во всех десяти случаях мы можем назвать их явно, с точностью до подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рис. 52.

Характеризация

Составление таблиц полностью аналогично случаю C1-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 код F , можно вычислить все остальные вершины. 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. Также, зная код пути FZY , можно вычислить код пути FJY и наоборот.

Таким образом, к W можно применить введённое нами соотношение из числа указанных справа на рис. 52 и получить код для другой части локального преобразования.

7.13. 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, т. е. $32FP^2\text{Num}(C)$.

7.14. Случай цепи B1; преобразования 8 и 9

В правой части рис. 53 изображены локальные преобразования 8 и 9.

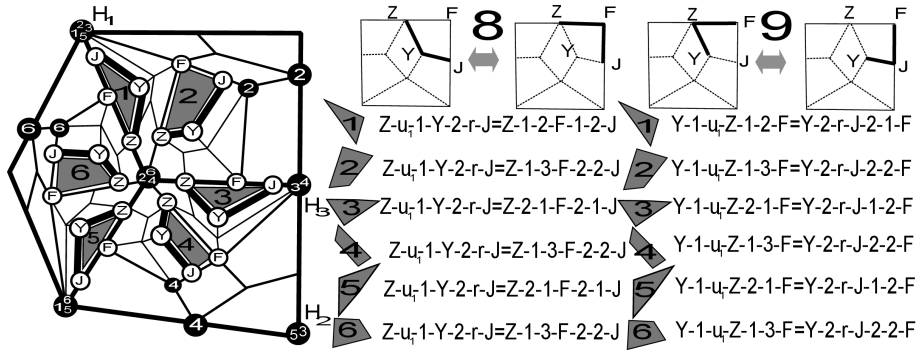


Рис. 53. Случаи расположения пути вокруг B1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Аналогично предыдущим случаям мы можем ввести определяющие соотношения. Фиксируем вершину типа B и её начальников, вершины H_1 и H_2 , причём для H_2 мы фиксируем только тип. Чёрными 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 **RightFromB**. Окружения для вершин Z, Y, F, J в каждом случае вычисляются с учётом известных нам окружения H_1 и типа H_2 , так 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 рис. 53.

Характеризация

Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода

Пусть мы знаем коды трёх 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 понятен 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.

7.15. Случай цепи $\mathbb{B}1$; преобразования 7 и 10

В правой части рис. 54 изображены локальные преобразования 7 и 10.

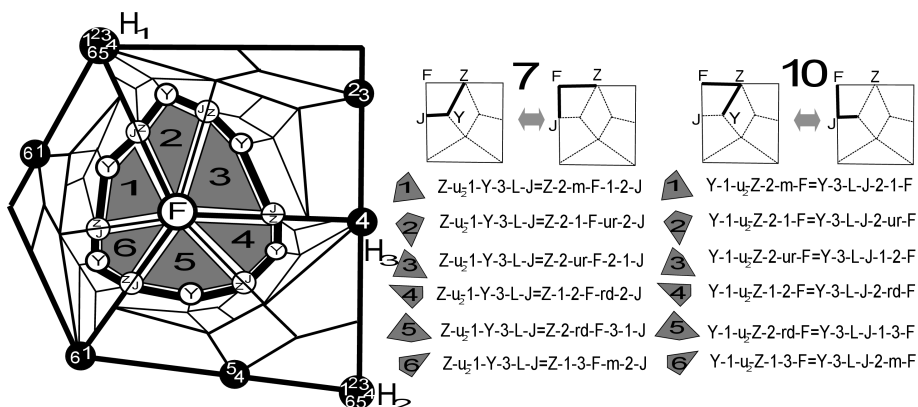


Рис. 54. Случаи расположения пути вокруг $\mathbb{B}1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Аналогично предыдущим случаям мы можем ввести определяющие соотношения. Фиксируем вершину типа \mathbb{B} и её начальников, вершины H_1 и H_2 , причём для H_2 мы фиксируем только тип. Чёрными 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 неё можем найти только тип). Типы, уровни и окружения для вершин Z, Y, F, J в каждом случае вычисляются с учётом известных нам окружения H_1 и типа H_2 , так 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 рис. 54.

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.

7.16. Случай цепи $\mathbb{B}2$; преобразования 8 и 9

В правой части рис. 55 изображены локальные преобразования 8 и 9.

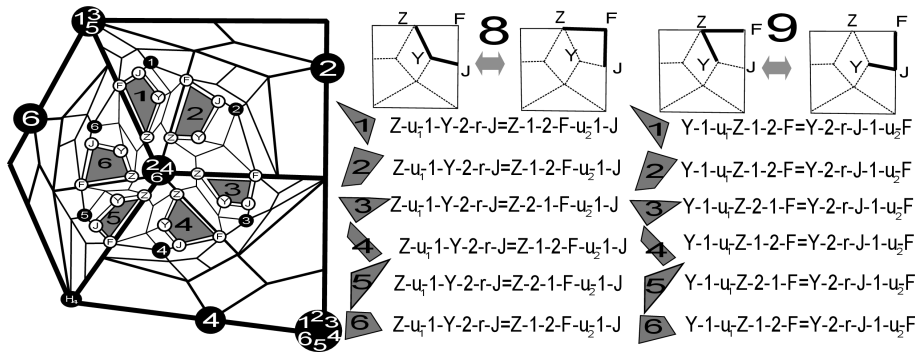


Рис. 55. Случаи расположения пути вокруг $\mathbb{B}2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Введение определяющих соотношений полностью аналогично предыдущим случаям, можно легко проверить, что, зная центральный \mathbb{B} -узел, коды всех четырёх вершин в каждом из случаев расположения можно легко выписать. Это позволяет ввести определяющие отношения, записанные в правой части рис. 55.

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 четырёх вершин мы можем вычислить. Кроме того, заметим, что каждая из четырёх вершин Z, J, Y, F в каждом из случаев либо имеет начальников среди других трёх вершин, либо у неё общие

начальники с какой-то из трёх других вершин. Значит, в каждом из случаев, зная коды трёх вершин, можно вычислить код четвёртой.

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.

7.17. Случай цепи $\mathbb{B}2$; преобразования 7 и 10

В правой части рис. 56 изображены локальные преобразования 7 и 10.

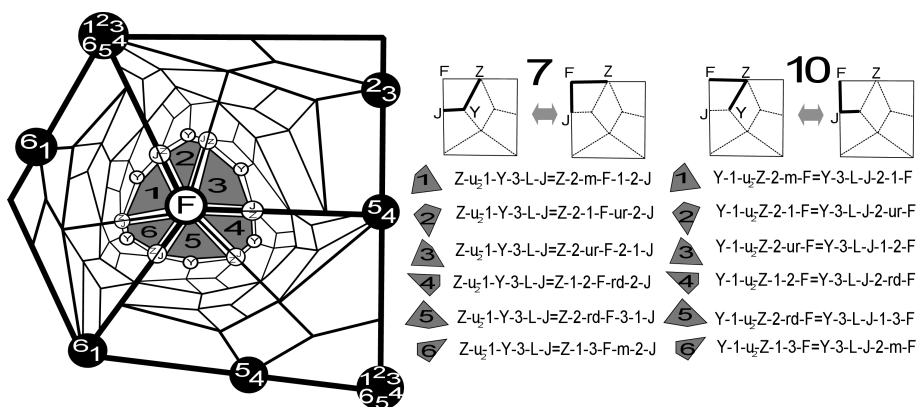


Рис. 56. Случаи расположения пути вокруг $\mathbb{B}2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Зафиксируем вершину типа \mathbb{B} и её 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.

7.18. Случай цепи $\mathbb{A}0$; преобразования 8 и 9

В правой части рис. 57 изображены локальные преобразования 8 и 9.

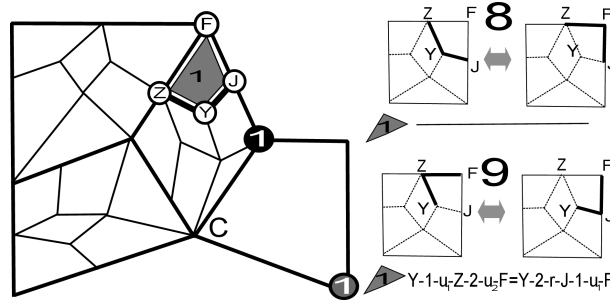


Рис. 57. Случаи расположения пути вокруг $\mathbb{A}0$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

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 мёртвый паттерн, для него определяющие соотношения вводить не требуется. По коду X вычисляются коды остальных трёх вершин. Например, тип второго начальника J — вершины в правом нижнем углу — может быть найден с помощью функции $\text{BottomRightType}(X)$.

Характеризация

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

Восстановление кода

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

7.19. Случай цепи $\mathbb{A}0$; преобразования 7 и 10

В правой части рис. 58 изображены локальные преобразования 7 и 10.

Зафиксируем вершину типа \mathbb{A} и её начальника (это вершина, отмеченная чёрным 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 рис. 58.

Характеризация

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

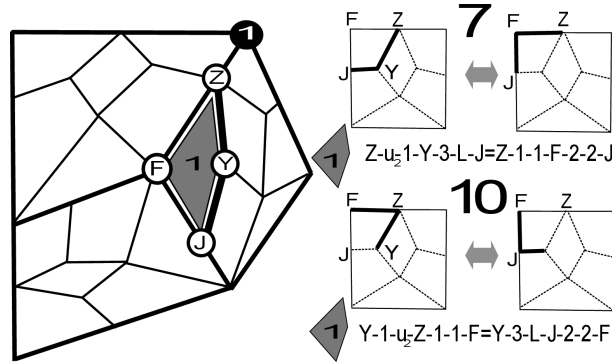


Рис. 58. Случаи расположения пути вокруг $\Delta 0$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Восстановление кода

Вершина, отмеченная чёрным 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.

7.20. Случай цепи $\Delta 1$; преобразования 8 и 9

В правой части рис. 59 изображены локальные преобразования 8 и 9.

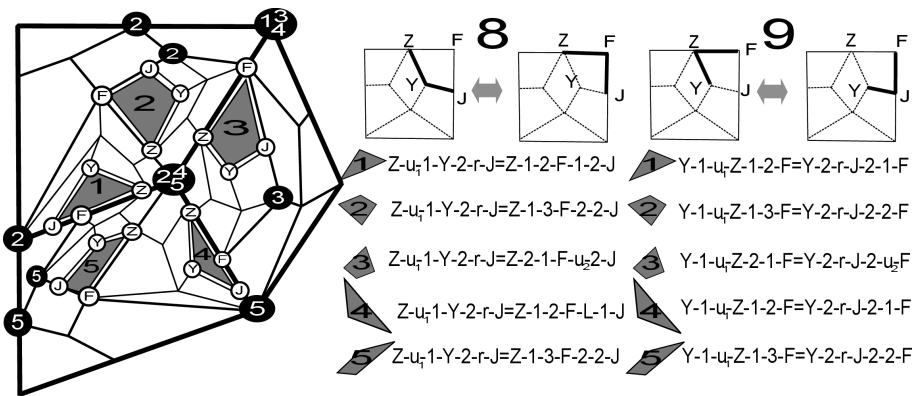


Рис. 59. Случаи расположения пути вокруг $\Delta 1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа Δ и её начальника (третьего уровня). Чёрными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J .

Заметим, что, зная окружение центрального Δ -узла, мы можем найти их типы, уровни и окружения. Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных чёрными 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 рис. 59.

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 вершин мы можем вычислить, зная остальные три. Кроме того, заметим, что каждая из четырёх вершин Z, J, Y, F в каждом из случаев либо имеет начальников среди других трёх вершин, либо у неё общие начальники с какой-то из трёх других вершин. Значит, в каждом из случаев, зная коды трёх вершин, можно вычислить код четвёртой.

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.

7.21. Случай цепи $\Delta 1$; преобразования 7 и 10

В правой части рис. 60 изображены локальные преобразования 7 и 10.

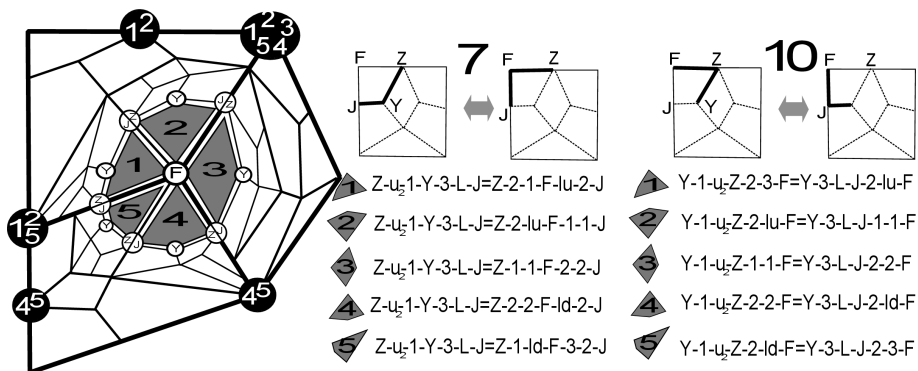


Рис. 60. Случаи расположения пути вокруг $\Delta 1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа Δ и её начальника. Чёрными 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 рис. 60.

Характеризация

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

Восстановление кода

Вершина F имеет общего начальника с одной из вершин J или Z , поэтому, 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.

7.22. Случай цепи $\mathbb{A}2$; преобразования 8 и 9

В правой части рис. 61 изображены локальные преобразования 8 и 9.

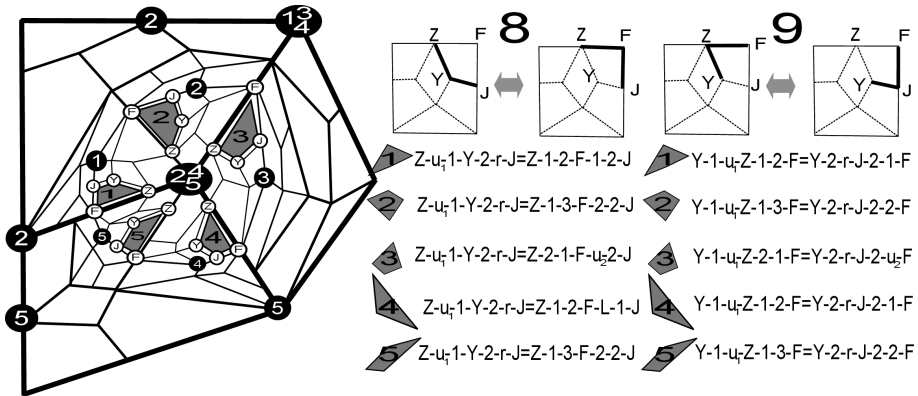


Рис. 61. Случаи расположения пути вокруг $\mathbb{A}2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Определяющие соотношения вводятся полностью аналогично случаю $\mathbb{A}1$ -цепи. 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.

7.23. Случай цепи $\mathbb{A}2$; преобразования 7 и 10

В правой части рис. 62 изображены локальные преобразования 7 и 10.

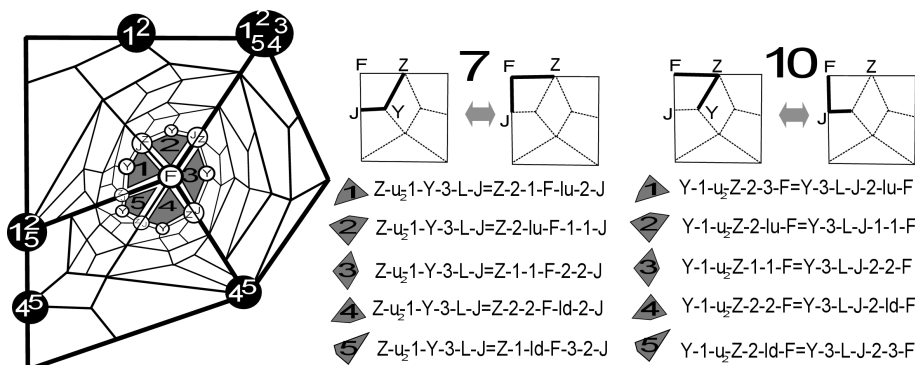


Рис. 62. Случаи расположения пути вокруг A2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Определяющие соотношения вводятся полностью аналогично случаю A1-цепи. 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.

7.24. Случай цепи UL1; преобразования 8 и 9

В правой части рис. 63 изображены локальные преобразования 8 и 9.

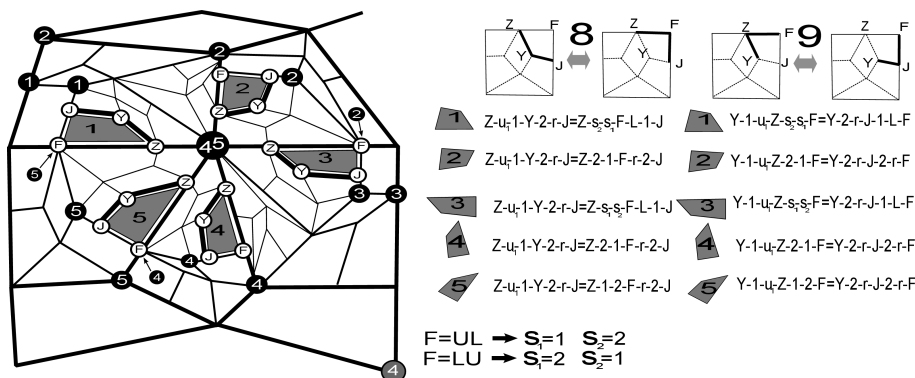


Рис. 63. Случаи расположения пути вокруг UL1-цепи и определяющие соотношения для локальных преобразований 8 и 9

Зафиксируем вершину типа UL третьего уровня и её начальников. Чёрными 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 \mathbb{UL} -узла, мы можем найти типы, уровни и окружения вершин с чёрными кругами (кроме вершины в правом нижнем углу, для неё можем найти только тип). Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных чёрными 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 рис. 63.

Характеризация

Составление таблиц полностью аналогично случаю $\mathbb{C1}$ -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 трёх. Кроме того, заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. У Y во всех случаях первым начальником является Z , а тип второго в каждом случае ясен из расположения.

Начальники J вычисляются следующим образом. В случае 1 это $\mathbf{Prev}(F)$; в случае 2 это 1-цепь вокруг $\mathbf{TypeBottomLeft.FBoss}(Z)$, указатель 1 (первый начальник), \mathbb{A} — тип второго начальника; в случае 3 это $\mathbf{Prev}(F)$; в случае 4 это 0-цепь с указателем 1 вокруг \mathbb{A} , окружение которого как \mathbb{U} — часть $\mathbf{FBoss}(Z)$ (первый начальник), \mathbb{B} — тип второго начальника; в случае 5 это $\mathbf{Plus.FBoss}(Z)$ (первый начальник), \mathbb{A} — тип второго начальника.

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.

7.25. Случай цепи $\mathbb{UL1}$; преобразования 7 и 10

В правой части рис. 64 изображены локальные преобразования 7 и 10.

Фиксируем некоторую вершину типа \mathbb{UL} третьего уровня и её начальников. Чёрными 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 рис. 64.

Характеризация

Составление таблиц полностью аналогично случаю $\mathbb{C1}$ -цепи.

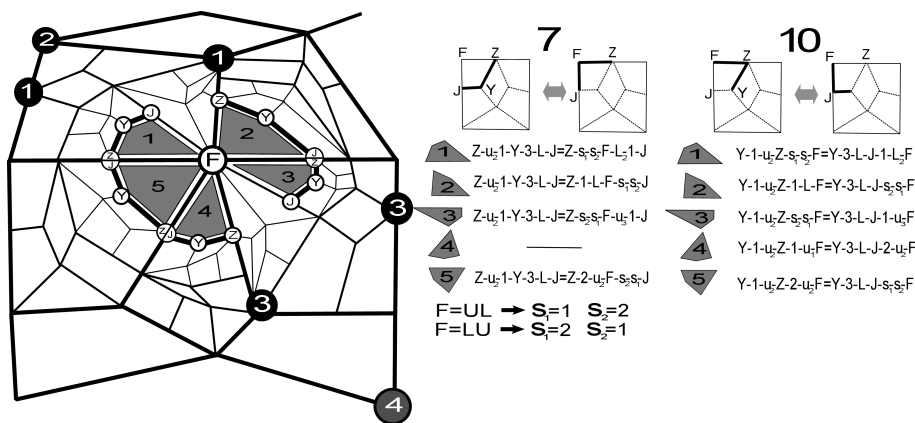


Рис. 64. Случаи расположения пути вокруг UL_1 -цепи и определяющие соотношения для локальных преобразований 7 и 10

Восстановление кода

Вершина F имеет общего начальника с одной из вершин J или Z , в каждом из случаев, кроме случая 4 (а в случае 4 нам не надо её вычислять), т. е., 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.

7.26. Случай цепи UL_2 ; преобразования 8 и 9

В правой части рис. 65 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа UL третьего уровня и её начальников. Чёрными 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 чёрными 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 рис. 65.

Характеризация

Составление таблиц полностью аналогично случаю C_1 -цепи.

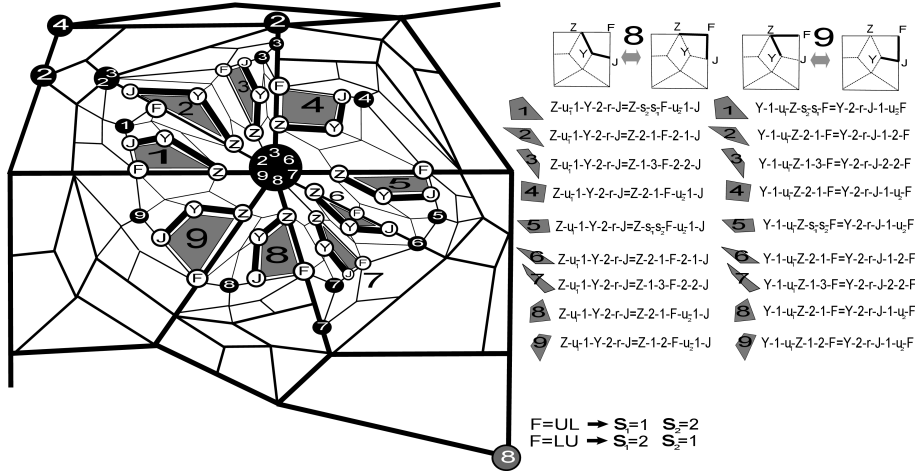


Рис. 65. Случаи расположения пути вокруг UL_2 -цепи и определяющие соотношения для локальных преобразований 8 и 9

Восстановление кода

Заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y также вычисляется легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае легко виден из рисунка.

Начальники J вычисляются следующим образом: в случаях 1, 4, 5, 8, 9 узел F — единственный начальник; в случаях 2 и 6 начальники те же, что и у F ; в случаях 3 и 7 первый и второй начальники такие, 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.

7.27. Случай цепи UL_2 ; преобразования 7 и 10

В правой части рис. 66 изображены локальные преобразования 7 и 10.

Фиксируем некоторую вершину типа UL третьего уровня и её начальников. Чёрными 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 рис. 66.

Для локального преобразования 7 пути в случаях 2, 3, 6, 7, 8 удовлетворяют условиям мёртвого паттерна, и в этих случаях мы соотношения не вводим.

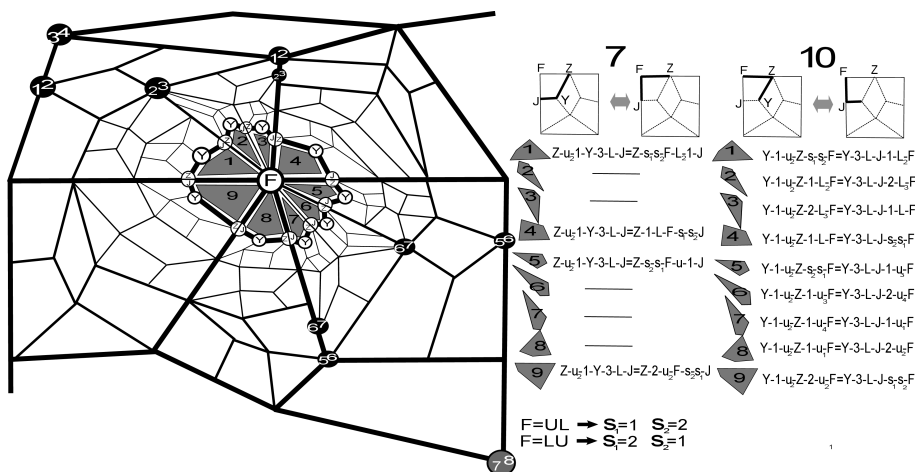


Рис. 66. Случаи расположения пути вокруг UL_2 -цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация

Составление таблиц полностью аналогично случаю C_1 -цепи.

Восстановление кода

В случаях 1, 4, 5, 9 вершина F имеет общего начальника с одной из вершин J или Z , т. е., 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.

7.28. Случай цепи UL_3 ; преобразования 7, 8, 9, 10

Случай UL_3 -цепи полностью аналогичен UL_2 -случаю, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают UL_3 -цепи. Все рассуждения о вычислении путей полностью аналогичны. Соотношений вводится столько же, сколько для UL_2 -случая.

7.29. Случай цепи UR_1 ; преобразования 8 и 9

В правой части рис. 67 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа UR третьего уровня и её начальников. Чёрными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J . Три круга попадают в F -узлы других расположений, это отмечено стрелками.

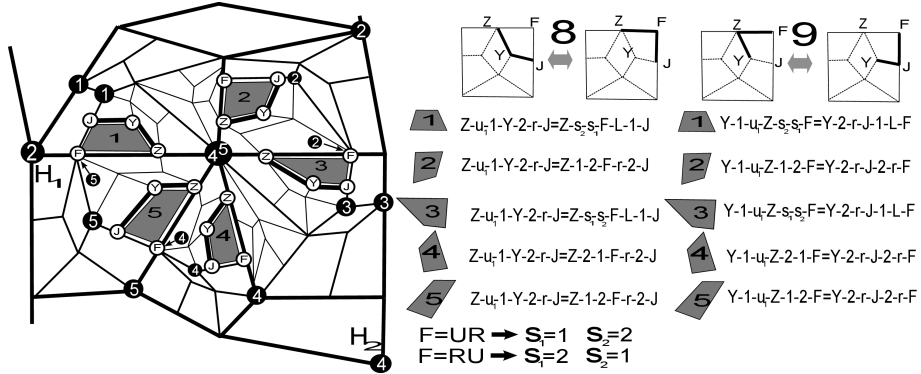


Рис. 67. Случаи расположения пути вокруг UR1-цепи и определяющие соотношения для локальных преобразований 8 и 9

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 чёрными 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 рис. 67.

Характеризация

Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода

Заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y также вычисляется легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае виден из рисунка.

Начальники J вычисляется следующим образом. В случае 1 это $\text{Prev}(F)$; в случае 2 это $\text{UpRightChain.FBoss}(Z)$ и \mathbb{B} — тип второго начальника; в случае 3 это $\text{Prev}(F)$; в случае 4 это 0-цепь с указателем 1 вокруг \mathbb{A} , окружение которого как \mathbb{U} — часть $\text{FBoss}(Z)$, и \mathbb{B} — тип второго начальника; в случае 5 это $\text{Plus.FBoss}(Z)$, \mathbb{B} — тип второго начальника.

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.

7.30. Случай цепи UR1; преобразования 7 и 10

В правой части рис. 68 изображены локальные преобразования 7 и 10.

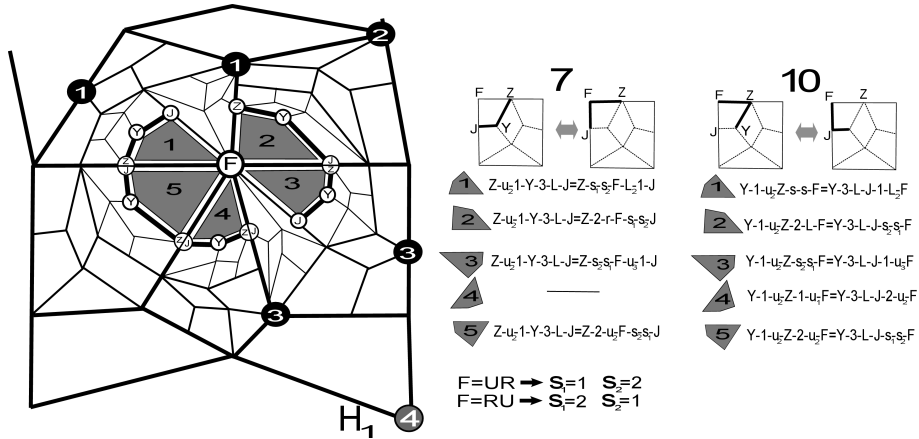


Рис. 68. Случаи расположения пути вокруг UR1-цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа UR третьего уровня и её начальников. Чёрными 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 подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рис. 68.

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 не надо её 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, т. е. мы можем осуществить локальное преобразование пути через операцию с его кодом.

7.31. Случай цепи UR2; преобразования 8 и 9

В правой части рис. 69 изображены локальные преобразования 8 и 9.

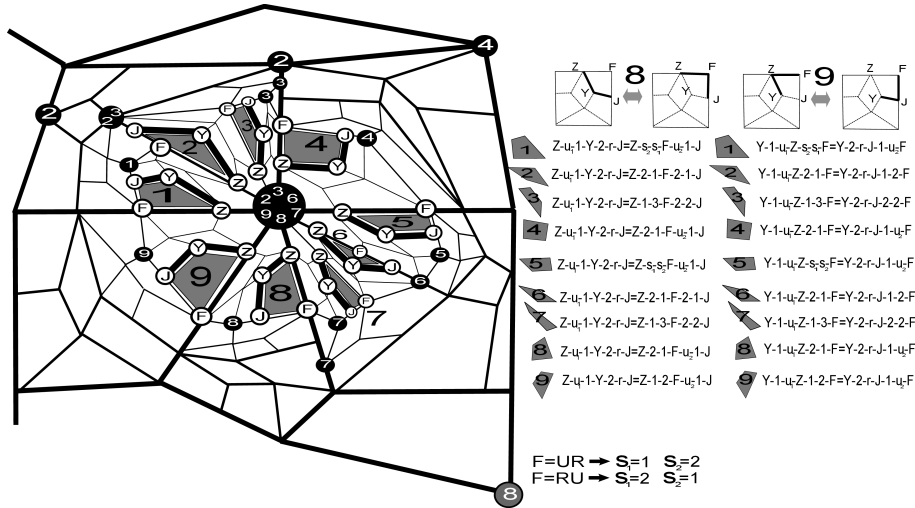


Рис. 69. Случаи расположения пути вокруг UR2-цепи и определяющие соотношения для локальных преобразований 8 и 9

Фиксируем вершину типа UR третьего уровня и её начальников. Чёрными 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 чёрными 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 69.

Характеризация

Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода

Заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y вычисляется также легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляются следующим образом: в случаях 1, 4, 5, 8, 9 узел F — единственный начальник; в случаях 2 и 6 начальники те же, что и у F ; в случаях 3 и 7 первый и второй начальники такие, 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.

7.32. Случай цепи UR2; преобразования 7 и 10

В правой части рис. 70 изображены локальные преобразования 7 и 10.

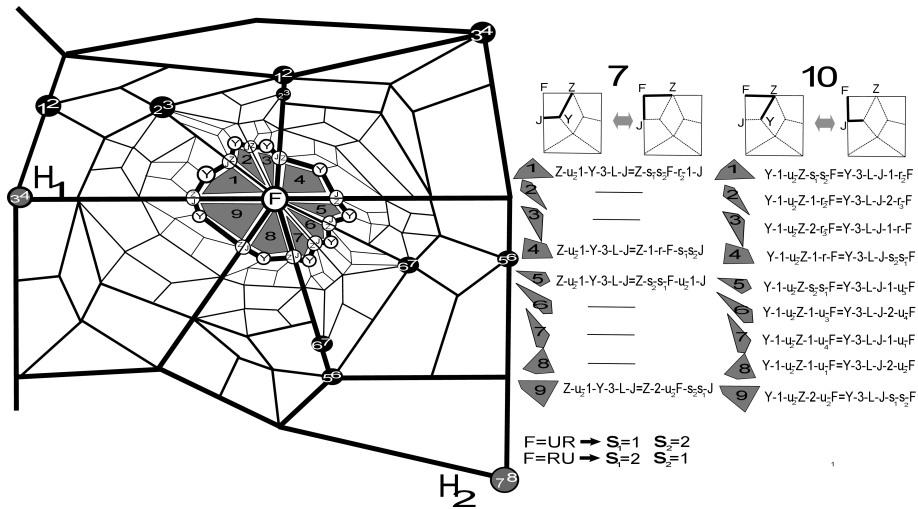


Рис. 70. Случаи расположения пути вокруг UR2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа UR и её начальников. Чёрными 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 рис. 70.

Для локального преобразования 7 в случаях 2, 3, 6, 7, 8 пути удовлетворяют условиям мёртвого 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 , т. е., 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.

7.33. 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 повторяются. Соотношений вводится столько же, сколько для UR2-случая.

7.34. Случай цепи DR1; преобразования 8 и 9

Аналогично предыдущим случаям мы можем ввести определяющие соотношения. Сначала зафиксируем вершину типа DR третьего уровня и её начальников. Всего возможно четыре случая расположения DR-вершины, на четырёх внутренних рёбрах (типов 2, 3, 5, 6). Все эти случаи расположения показаны на рис. 71.

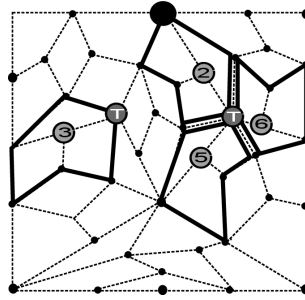


Рис. 71. Случаи расположения DR-вершины

Первым начальником нашей DR-вершины во всех случаях будет вершина в середине верхней стороны. Теперь рассмотрим цепь с центром в этой DR-вершине. В правой части рис. 72 изображены локальные преобразования 8 и 9.

Чёрными 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 DR-узел: если это ребро 3, то A , в остальных случаях (2, 5 или 6) — B .

Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных чёрными 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 рис. 72.

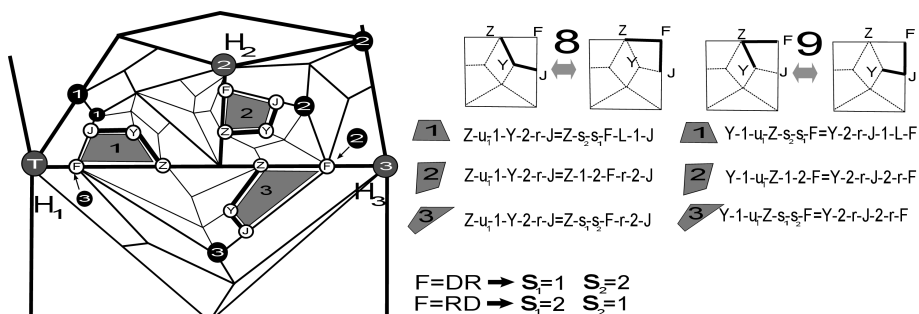


Рис. 72. Случаи расположения пути вокруг $\mathbb{DR}1$ -цепи и определяющие соотношения для локальных преобразований 8 и 9

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 окружение каждой вершины мы можем вычислить, зная код остальных трёх. Кроме того, заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y также вычисляется легко, так как во всех случаях первым начальником является Z , а тип правого нижнего угла в каждом случае ясен.

Начальники J вычисляются следующим образом. В случае 1 это $\text{Prev}(F)$; в случае 2 первый начальник — $\text{UpRightChain.FBoss}(Z)$ и \mathbb{B} — тип второго начальника; в случае 3 это 1-цепь вокруг узла T с указателем в зависимости от типа ребра центрального \mathbb{DR} -узла: 3 для типа 3, 1 для типа 2, 2 для типа 6, 3 для типа 5.

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.

7.35. Случай цепи $\mathbb{DR}1$; преобразования 7 и 10

В правой части рис. 73 изображены локальные преобразования 7 и 10.

Фиксируем некоторую вершину типа \mathbb{DR} третьего уровня и её начальников. Чёрными 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

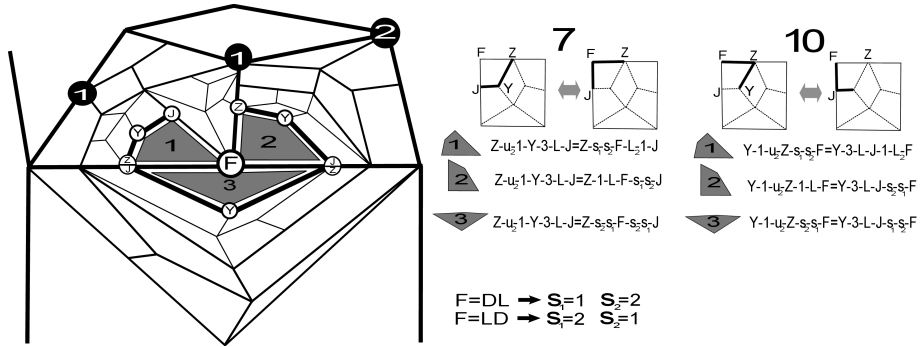


Рис. 73. Случаи расположения пути вокруг $\mathbb{DR}1$ -цепи и определяющие соотношения для локальных преобразований 7 и 10

подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рис. 73.

Характеризация

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

Восстановление кода

Вершина F имеет общего начальника с одной из вершин J или Z в каждом из случаев, т. е., 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.

7.36. Случай цепи $\mathbb{DR}2$; преобразования 8 и 9

В правой части рис. 74 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа \mathbb{DR} третьего уровня и её начальников. Чёрными 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 чёрными 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 рис. 74.

Характеризация

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

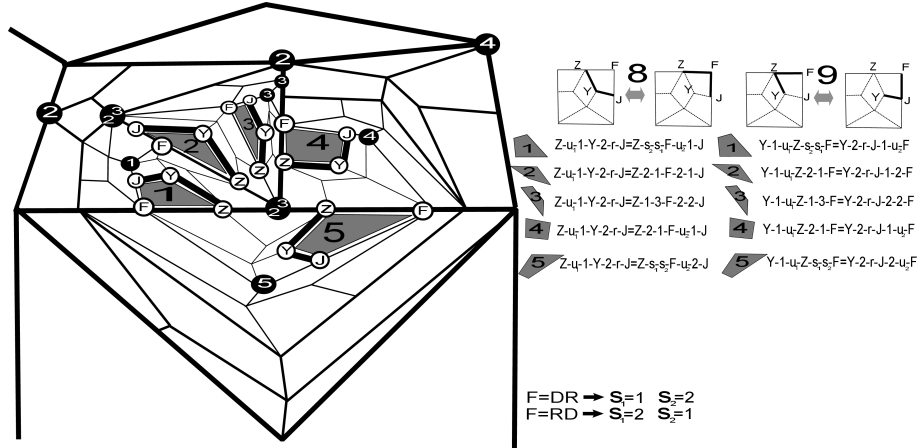


Рис. 74. Случаи расположения пути вокруг $\mathbb{DR}2$ -цепи и определяющие соотношения для локальных преобразований 8 и 9

Восстановление кода

Заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y также вычисляется легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляются следующим образом: в случаях 1, 4, 5 узел F — единственный начальник; в случае 2 начальники те же, что и у F ; в случае 3 первый и второй начальники такие, 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.

7.37. Случай цепи $\mathbb{DR}2$; преобразования 7 и 10

В правой части рис. 75 изображены локальные преобразования 7 и 10.

Зафиксируем некоторую вершину типа \mathbb{DR} третьего уровня и её начальников. Чёрными 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 рис. 75.

Для локального преобразования 7 в случаях 2, 3, 6, 7, 8 пути удовлетворяют условиям мёртвого паттерна, и в этих случаях мы соотношения не вводим.

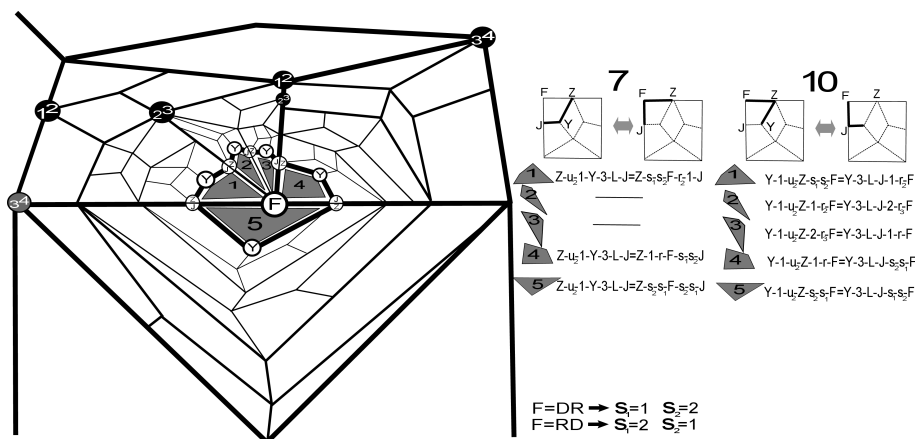


Рис. 75. Случаи расположения пути вокруг $\mathbb{DR}2$ -цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация

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

Восстановление кода

В случаях 1, 4, 5 вершина F имеет общего начальника с одной из вершин J или Z , т. е., 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.

7.38. Случай цепи $\mathbb{DR}3$; преобразования 7, 8, 9, 10

Случай $\mathbb{DR}3$ -цепи полностью аналогичен $\mathbb{DR}2$ -случаю, соотношения выглядят идентично, только кодировки вершин J, F, Z, Y отвечают $\mathbb{DR}3$ -цепи. Все рассуждения о вычислении путей полностью повторяются. Соотношений вводятся столько же, сколько для $\mathbb{DR}2$ -случая.

7.39. Случай цепи $\mathbb{DL}1$; преобразования 8 и 9

В правой части рис. 76 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа \mathbb{DL} третьего уровня и её начальников. Чёрными 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 являются

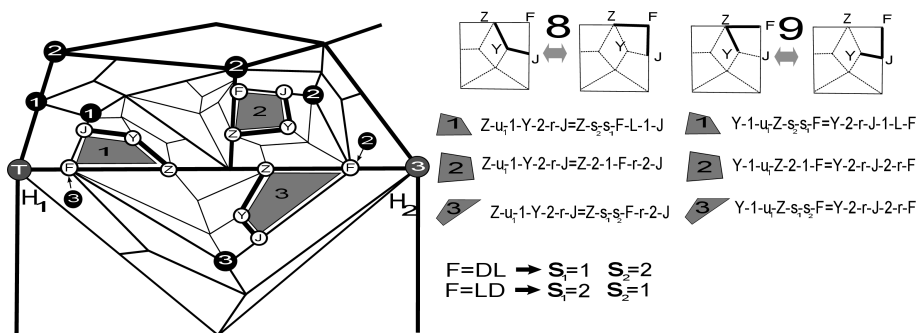


Рис. 76. Случаи расположения пути вокруг \mathbb{DL} -цепи и определяющие соотношения для локальных преобразований 8 и 9

данные вершины. Зная окружение центрального \mathbb{DL} -узла, мы можем найти их типы, уровни и окружения (для вершин H_1 и H_2 можем найти только тип).

Так как начальники вершин Z, Y, F, J содержатся среди вершин, отмеченных чёрными 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 рис. 76.

Характеризация

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

Восстановление кода

Заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y также вычисляется легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляются следующим образом. В случае 1 это $\mathbf{Prev}(F)$; в случае 2 первый начальник — $\mathbf{BottomLeftChain.FBoss}(Z)$ и \mathbb{A} — тип второго начальника. Для случая 3 заметим, что тип узла, отмеченного символом T , зависит от типа ребра, на котором лежат Z и F . Так как это ребро с центром в \mathbb{DL} -вершине, то тип его может быть либо 4, либо 7, в соответствии со структурой разбиения макроплитки на подплитки. Если ребро имеет тип 4, 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 ребру 7.

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.

7.40. Случай цепи DL1; преобразования 7 и 10

В правой части рис. 77 изображены локальные преобразования 7 и 10.

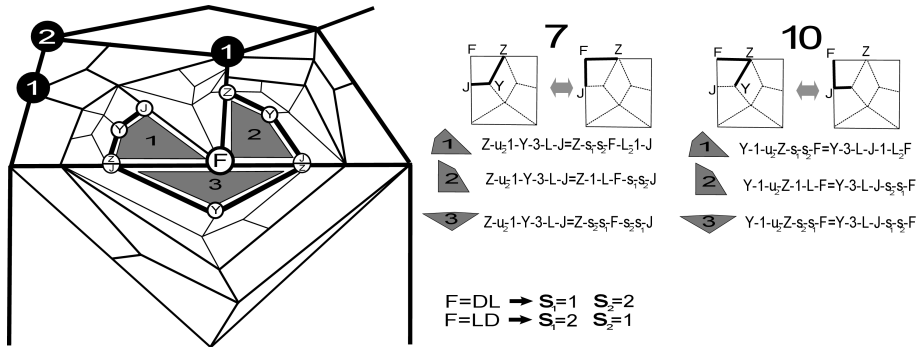


Рис. 77. Случаи расположения пути вокруг DL1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа DL третьего уровня и её начальника. Чёрными 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 подклеенных окружений. Это позволяет ввести определяющие отношения, записанные в правой части рис. 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, т. е., 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.

7.41. Случай цепи DL2; преобразования 8 и 9

В правой части рис. 78 изображены локальные преобразования 8 и 9.

Зафиксируем вершину типа DR третьего уровня и её начальников. Чёрными кругами отметим вершины, являющиеся начальниками вершин Z, Y, F, J.

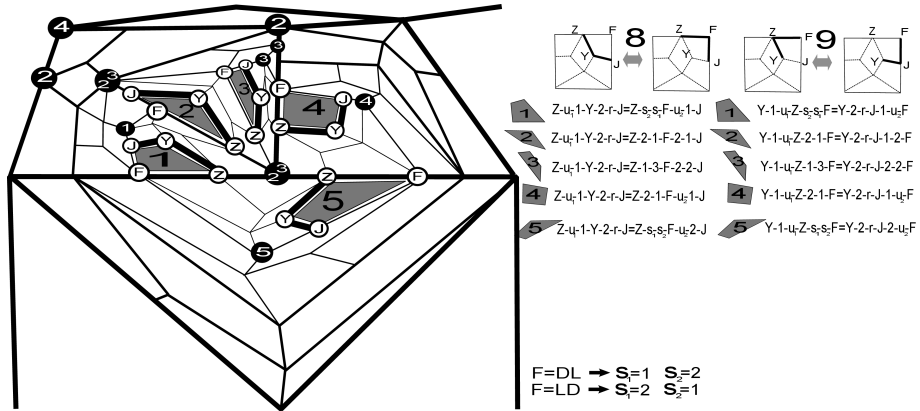


Рис. 78. Случаи расположения пути вокруг DL2-цепи и определяющие соотношения для локальных преобразований 8 и 9

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 чёрными 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 рис. 78.

Характеризация

Составление таблиц полностью аналогично случаю C1-цепи.

Восстановление кода

Заметим, что у Z и F в каждом из случаев общие начальники, т. е. код каждой из этих вершин может быть вычислен, исходя из кодов остальных трёх вершин. Y также вычисляется легко, так как во всех случаях первым начальником является Z , а тип второго в каждом случае ясен.

Начальники J вычисляются 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 такие же, 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.

7.42. Случай цепи DL2; преобразования 7 и 10

В правой части рис. 79 изображены локальные преобразования 7 и 10.

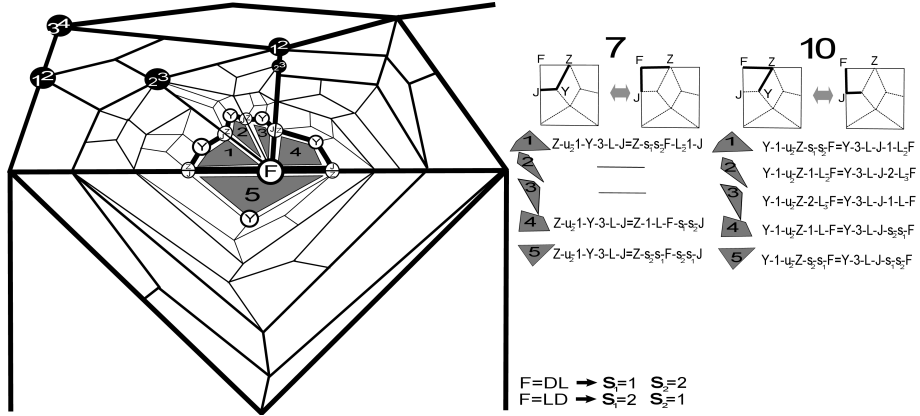


Рис. 79. Случаи расположения пути вокруг DL2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Фиксируем некоторую вершину типа DL третьего уровня и её начальников. Чёрными 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 рис. 79.

Для локального преобразования 7 пути в случаях 2 и 3 удовлетворяют условиям мёртвого 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, т. е., 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.

7.43. 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, т. е. 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 приведём вводимые определяющие соотношения.

7.44. Случай цепи $\mathbb{D}1$; преобразования 8 и 9

В правой части рис. 80 изображены локальные преобразования 8 и 9.

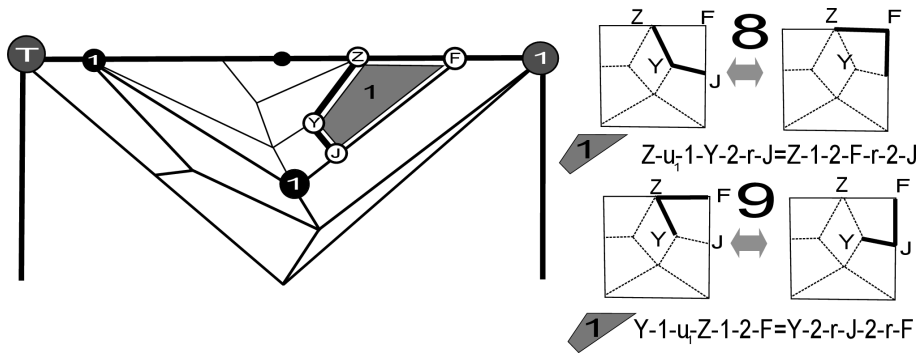


Рис. 80. Случаи расположения пути вокруг $\mathbb{D}1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DL}1$ -или $\mathbb{DR}1$ -цепи.

7.45. Случай цепи $\mathbb{D}1$; преобразования 7 и 10

В правой части рис. 81 изображены локальные преобразования 7 и 10.

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DL}1$ -или $\mathbb{DR}1$ -цепи.

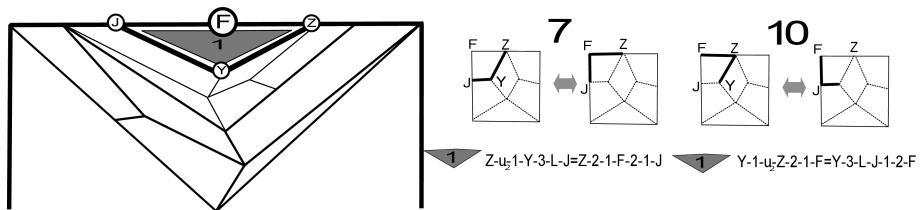


Рис. 81. Случаи расположения пути вокруг $\mathbb{D}1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

7.46. Случай цепи $\mathbb{D}2$; преобразования 8 и 9

В правой части рис. 82 изображены локальные преобразования 8 и 9.

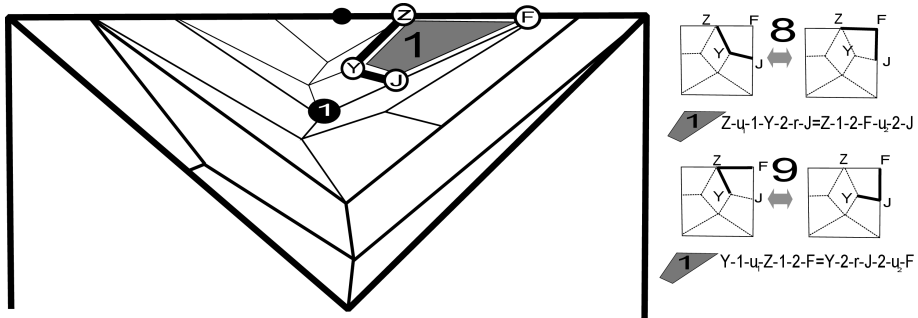


Рис. 82. Случай расположения пути вокруг $\mathbb{D}2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DL}2$ -или $\mathbb{DR}2$ -цепи.

7.47. Случай цепи $\mathbb{D}2$; преобразования 7 и 10

В правой части рис. 83 изображены локальные преобразования 7 и 10.

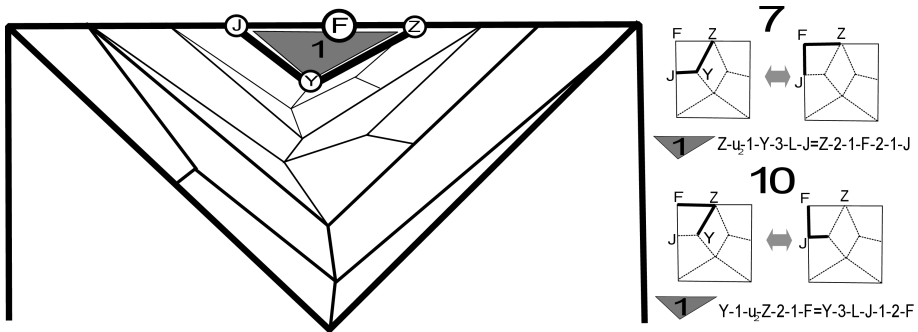


Рис. 83. Случай расположения пути вокруг $\mathbb{D}2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DL}2$ -или $\mathbb{DR}2$ -цепи.

7.48. Случай цепи U1; преобразования 8 и 9

В правой части рис. 84 изображены локальные преобразования 8 и 9.

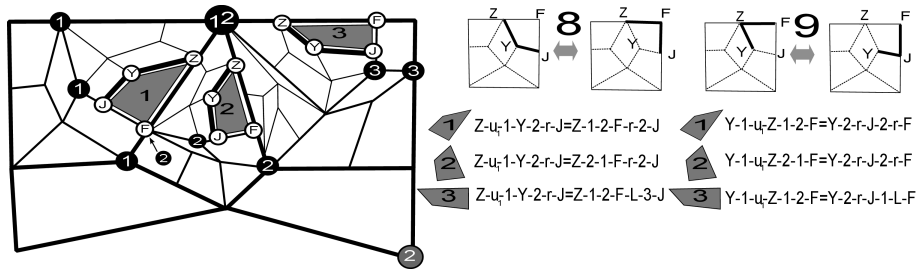


Рис. 84. Случаи расположения пути вокруг U1-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю UL1- или UR1-цепи.

7.49. Случай цепи U1; преобразования 7 и 10

В правой части рис. 85 изображены локальные преобразования 7 и 10.

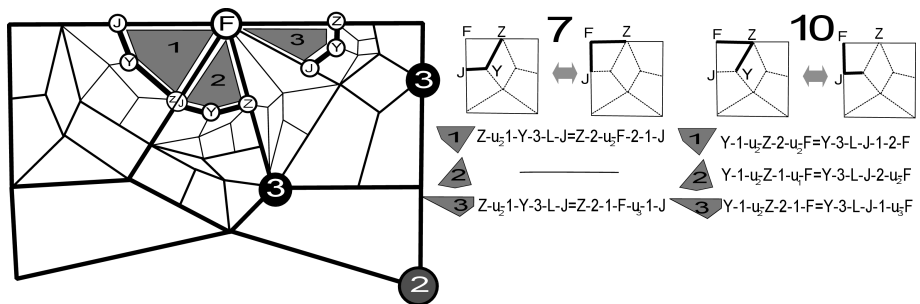


Рис. 85. Случаи расположения пути вокруг U1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю UL1- или UR1-цепи.

7.50. Случай цепи U2; преобразования 8 и 9

В правой части рис. 86 изображены локальные преобразования 8 и 9.

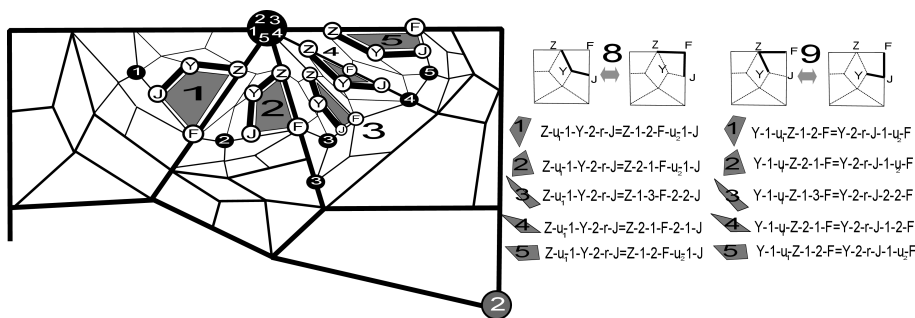


Рис. 86. Случаи расположения пути вокруг U2-цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю UL2- или UR2-цепи.

7.51. Случай цепи U2; преобразования 7 и 10

В правой части рис. 87 изображены локальные преобразования 7 и 10.

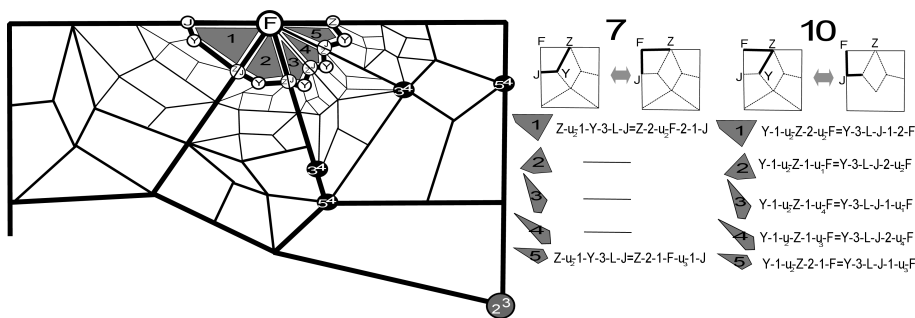


Рис. 87. Случаи расположения пути вокруг U2-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю UL2- или UR2-цепи.

7.52. 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 повторяются. Соотношений вводится столько же, сколько для $\mathbb{U}2$ -случая.

7.53. Случай цепи $\mathbb{L}1$; преобразования 8 и 9

В правой части рис. 88 изображены локальные преобразования 8 и 9.

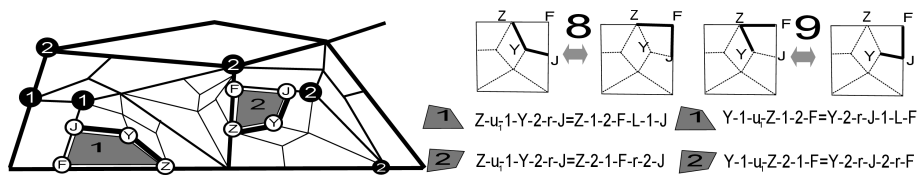


Рис. 88. Случай расположения пути вокруг $\mathbb{L}1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{U}\mathbb{L}1$ -или $\mathbb{D}\mathbb{L}1$ -цепи.

7.54. Случай цепи $\mathbb{L}1$; преобразования 7 и 10

В правой части рис. 89 изображены локальные преобразования 7 и 10.

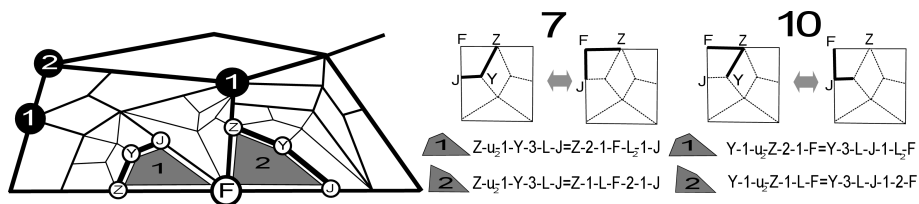


Рис. 89. Случай расположения пути вокруг $\mathbb{L}1$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{U}\mathbb{L}1$ -или $\mathbb{D}\mathbb{L}1$ -цепи.

7.55. Случай цепи $\mathbb{L}2$; преобразования 8 и 9

В правой части рис. 90 изображены локальные преобразования 8 и 9.

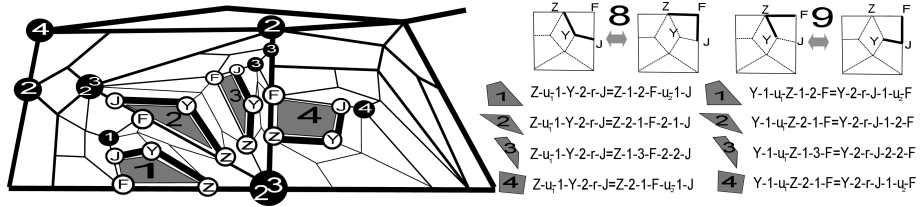


Рис. 90. Случаи расположения пути вокруг $\mathbb{L}2$ -цепи и соответствующие им определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{UL}2$ -или $\mathbb{DL}2$ -цепи.

7.56. Случай цепи $\mathbb{L}2$, преобразования 7 и 10

В правой части рис. 91 изображены локальные преобразования 7 и 10.

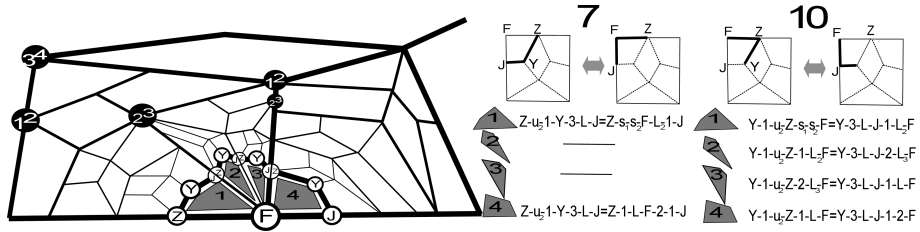


Рис. 91. Случаи расположения пути вокруг $\mathbb{L}2$ -цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{UL}2$ -или $\mathbb{DL}2$ -цепи.

7.57. 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 повторяются. Соотношений вводится столько же, сколько для $\mathbb{L}2$ -случая.

7.58. Случай цепи $\mathbb{R}1$; преобразования 8 и 9

В правой части рис. 92 изображены локальные преобразования 8 и 9.

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DR}1$ -или $\mathbb{UR}1$ -цепи.

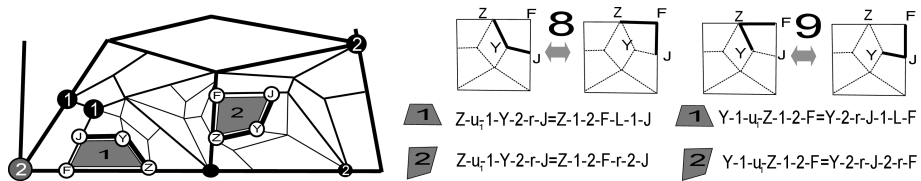


Рис. 92. Случаи расположения пути вокруг $\mathbb{R}1$ -цепи и определяющие соотношения для локальных преобразований 8 и 9

7.59. Случай цепи $\mathbb{R}1$; преобразования 7 и 10

В правой части рис. 93 изображены локальные преобразования 7 и 10.

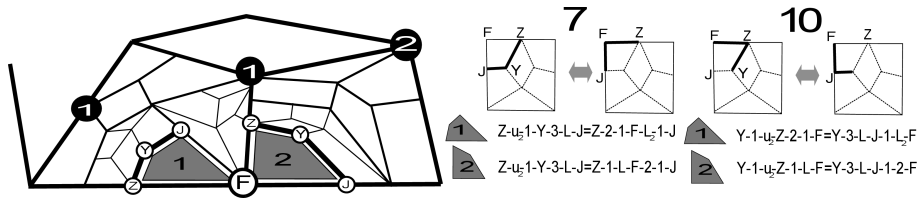


Рис. 93. Случаи расположения пути вокруг $\mathbb{R}1$ -цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DR}1$ -или $\mathbb{UR}1$ -цепи.

7.60. Случай цепи $\mathbb{R}2$; преобразования 8 и 9

В правой части рис. 94 изображены локальные преобразования 8 и 9.

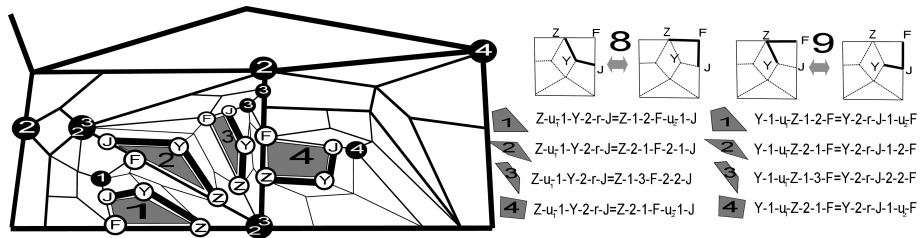


Рис. 94. Случаи расположения пути вокруг $\mathbb{R}2$ -цепи и определяющие соотношения для локальных преобразований 8 и 9

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DR}2$ -или $\mathbb{UR}2$ -цепи.

7.61. Случай цепи $\mathbb{R}2$; преобразования 7 и 10

В правой части рис. 95 изображены локальные преобразования 7 и 10.

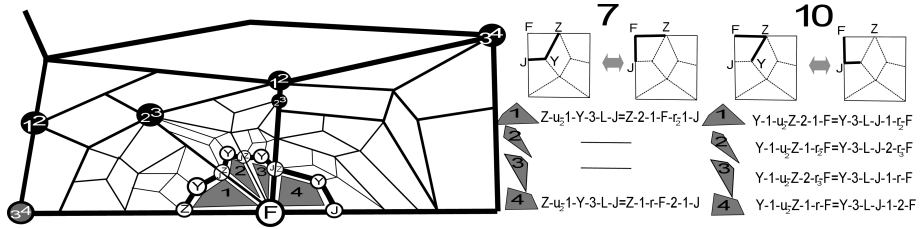


Рис. 95. Случаи расположения пути вокруг $\mathbb{R}2$ -цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация и восстановление кода полностью аналогично случаю $\mathbb{DR}2$ -или $\mathbb{UR}2$ -цепи.

7.62. 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 повторяются. Соотношений вводится столько же, сколько для $\mathbb{R}2$ -случая.

7.63. Случай цепи $\mathbb{CUL}0$; преобразования 7, 8, 9, 10

В правой части рис. 96 изображены локальные преобразования 7, 8, 9 и 10.

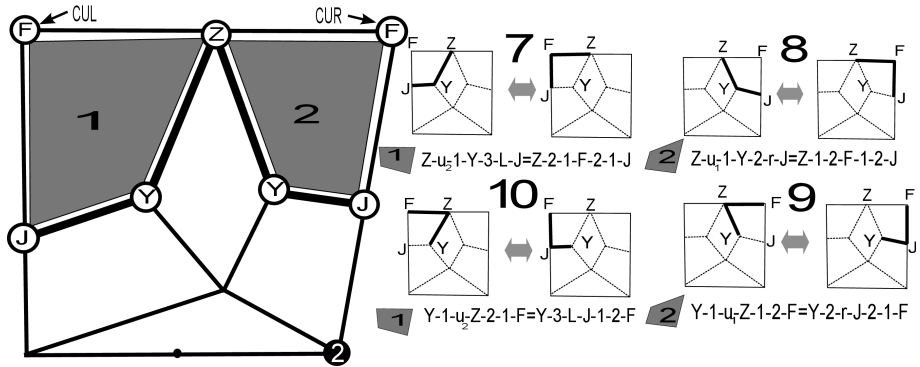


Рис. 96. Случаи расположения пути вокруг $\mathbb{CUL}0$ -цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвертой вершины по известным трём другим также ясно во всех случаях.

7.64. Случай цепи CUL1; преобразования 7, 8, 9, 10

В правой части рис. 97 изображены локальные преобразования 7, 8, 9 и 10.

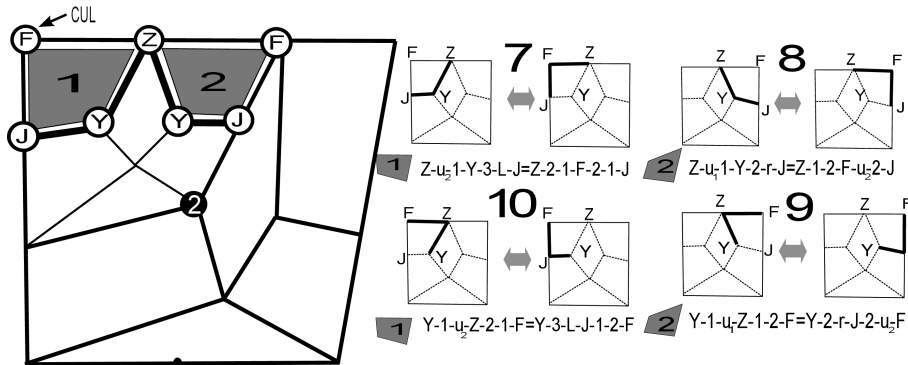


Рис. 97. Случаи расположения пути вокруг CUL1-цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвертой вершины по известным трём другим также ясно во всех случаях.

7.65. Случай цепи CUR1; преобразования 7, 8, 9, 10

В правой части рис. 98 изображены локальные преобразования 7, 8, 9 и 10.

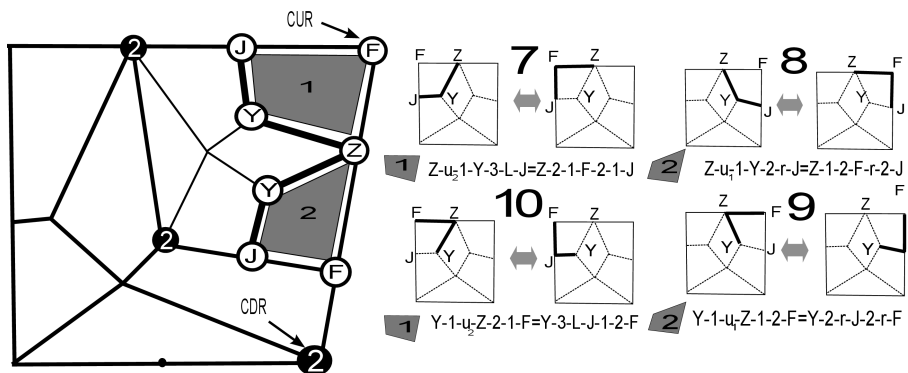


Рис. 98. Случаи расположения пути вокруг CUR1-цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвертой вершины по известным трём другим также ясно во всех случаях.

7.66. Случай цепи CUR2; преобразования 7, 8, 9, 10

В правой части рис. 99 изображены локальные преобразования 8 и 9.

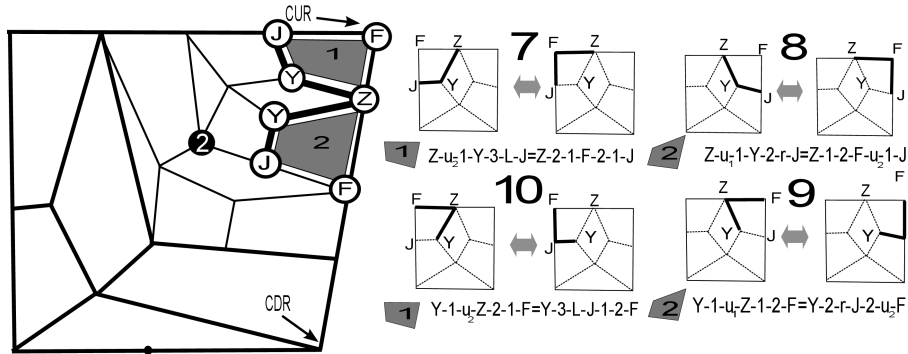


Рис. 99. Случай расположения пути вокруг CUR2-цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

Характеризация очевидна. Восстановление кода четвёртой вершины по известным трём другим также ясно во всех случаях.

7.67. Случай цепи CDL1; преобразования 7, 8, 9, 10

В правой части рис. 100 изображены локальные преобразования 7, 8, 9 и 10.

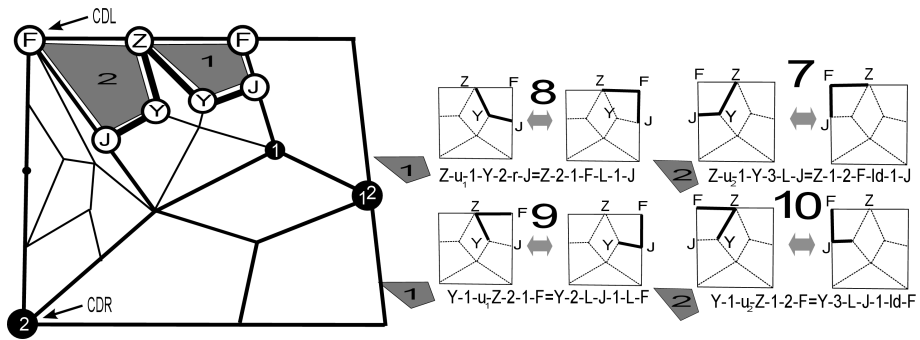


Рис. 100. Случай расположения пути вокруг CDL1-цепи и определяющие соотношения для локальных преобразований 7, 8, 9, 10

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 — A, а Z — краевая вершина без начальников.

7.68. Случай цепи CDL2; преобразования 8 и 9

В правой части рис. 101 изображены локальные преобразования 8 и 9.

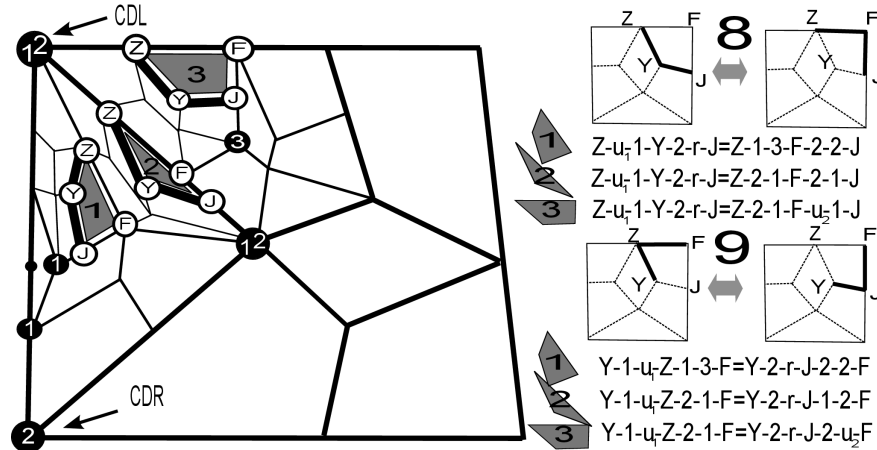


Рис. 101. Случаи расположения пути вокруг CDL2-цепи и определяющие соотношения для локальных преобразований 8 и 9

Характеризация очевидна. Восстановление кода также ясно, учитывая, что у F и Z всегда общее множество начальников, у Y первый начальник — Z , а тип

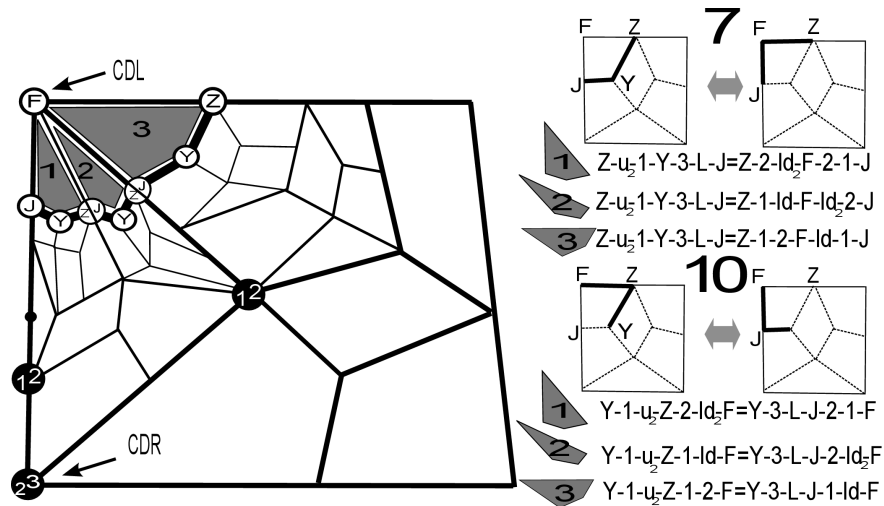


Рис. 102. Случаи расположения пути вокруг CDL2-цепи и определяющие соотношения для локальных преобразований 7 и 10

второго ясен из расположения. Начальники J во всех случаях либо содержатся среди начальников F , либо сам F является начальником.

7.69. Случай цепи CDL2; преобразования 7 и 10

В правой части рис. 102 изображены локальные преобразования 7 и 10.

Характеризация очевидна. Для доказательства восстановления кода достаточно заметить, что окружения четырёх вершин в каждом случае очевидны, а окружение начальников (вершины, отмеченных чёрными кругами) можно явным образом написать.

7.70. 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 повторяются. Соотношений вводится столько же, сколько для CDL2-случая.

7.71. Случай цепи CDR1; преобразования 8 и 9

В правой части рис. 103 изображены локальные преобразования 8 и 9.

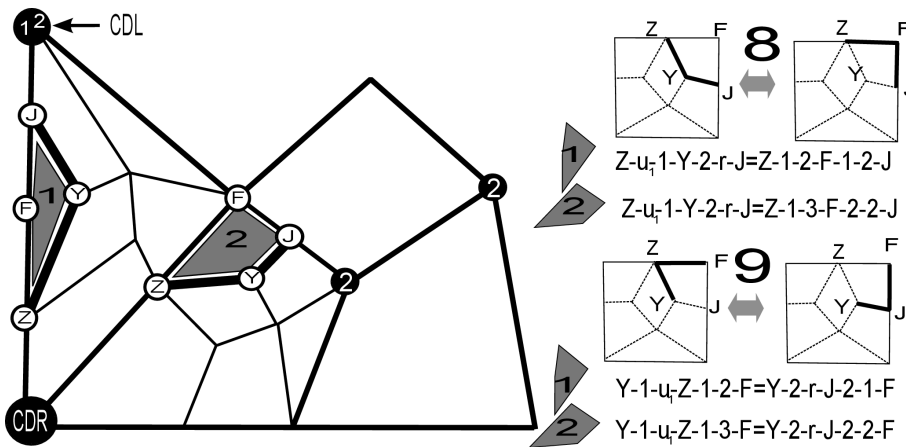


Рис. 103. Случаи расположения пути вокруг CDR1-цепи и определяющие соотношения для локальных преобразований 8 и 9

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырёх, либо является начальником другой вершины из четырёх.

7.72. Случай цепи CDR1; преобразования 7 и 10

В правой части рис. 104 изображены локальные преобразования 7 и 10.

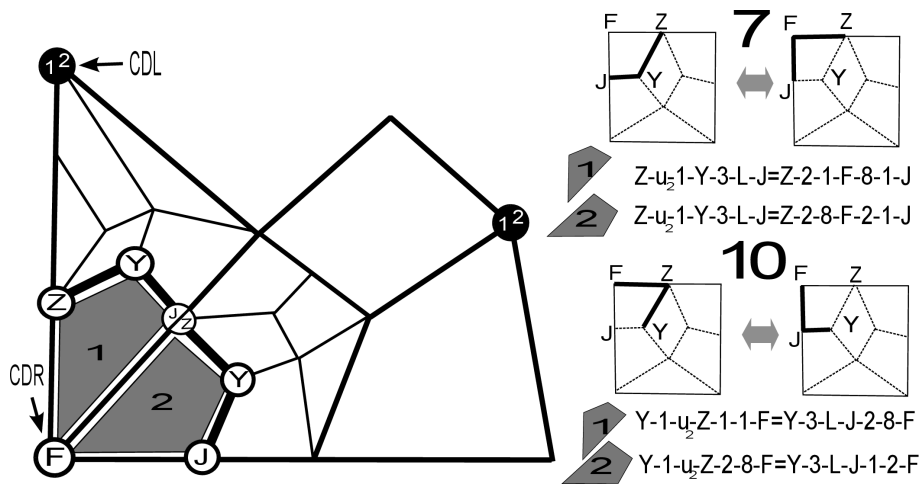


Рис. 104. Случаи расположения пути вокруг CDR1-цепи и соответствующие им определяющие соотношения для локальных преобразований 7 и 10

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырёх, либо является начальником другой вершины из четырёх.

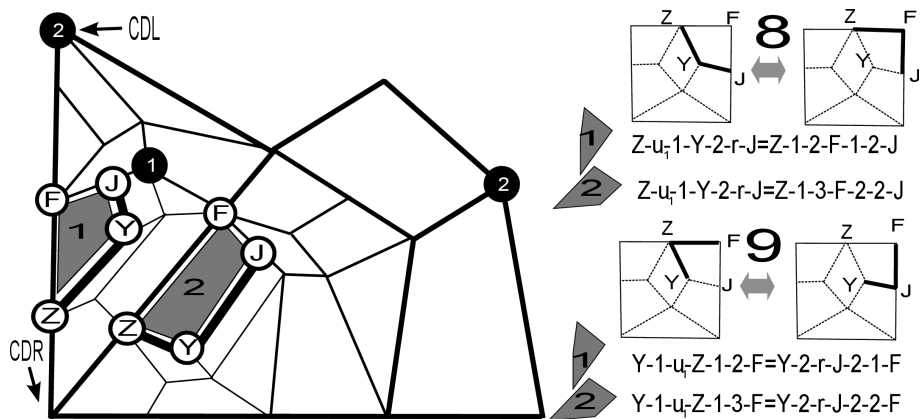


Рис. 105. Случаи расположения пути вокруг CDR2-цепи и определяющие соотношения для локальных преобразований 8 и 9

7.73. Случай цепи CDR2; преобразования 8 и 9

В правой части рис. 105 изображены локальные преобразования 8 и 9.

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырёх, либо является начальником другой вершины из четырёх.

7.74. Случай цепи CDR2; преобразования 7 и 10

В правой части рис. 106 изображены локальные преобразования 7 и 10.

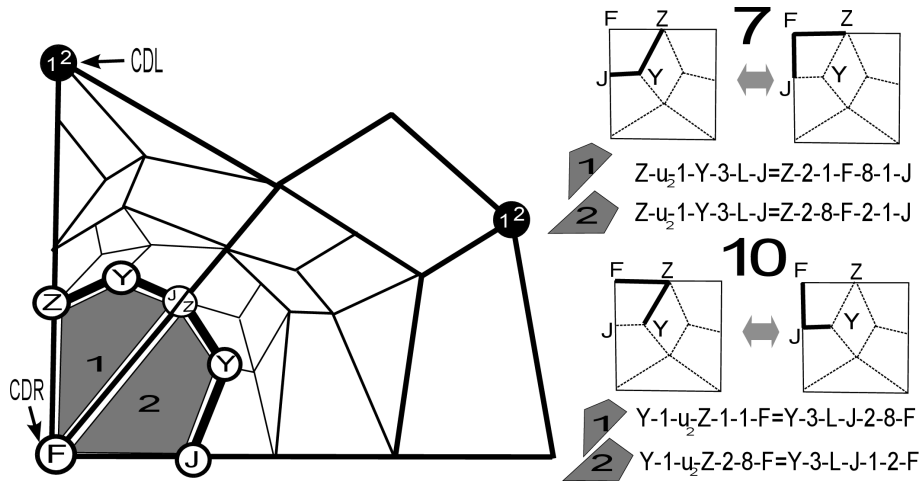


Рис. 106. Случай расположения пути вокруг CDR2-цепи и определяющие соотношения для локальных преобразований 7 и 10

Характеризация очевидна. Для восстановления кода достаточно заметить, что начальник каждой вершины либо является другой вершиной среди четырёх, либо является начальником другой вершины из четырёх.

7.75. Подсчёт введённых соотношений

Пусть F — число различных флагов макроплиток, P — число различных подклеенных окружений, I — число различных информации. Пусть также $\text{Num}(A)$ — число базовых окружений вершин типа A , аналогичные обозначения будут использоваться для других типов. Крайние вершины в пути могут иметь произвольное подклеенное окружение, кроме того, соотношения вводятся для каждого заданного флага подклейки.

Для каждого из локальных преобразований 1–6 мы фиксировали один узел с окружением, флагом макроплитки и информацией. Также ещё для одного узла могла быть зафиксирована информация. Затем в каждом из шести случаев

расположения мы вводили не более двух соотношений для каждого сочетания подклеенных окружений крайних узлов пути и флага макроплитки, т. е. общее число соотношений для локальных преобразований 1—6 не превосходит $72 \cdot I \cdot F \cdot P \cdot \mathbf{Num}$, где \mathbf{Num} — число всевозможных сочетаний типа, уровня, окружения и информации плоской вершины. Таким образом, для локальных преобразований 1—6 было введено не более

$$72 \cdot 181 \cdot 10^6 \cdot 5 \cdot 10^{18} \cdot 39 \cdot 16 \cdot 10^{12} < 41 \cdot 10^{42}$$

соотношений.

Теперь посчитаем число введённых соотношений для локальных преобразований 7—10. Напомним, что мы рассматривали вершины из цепей, фиксировали для них информацию, флаг подклейки, а также два подклеенных окружения. Затем вводилось не более двух соотношений для каждого локального преобразования. Таким образом, число введённых соотношений для локальных преобразований 7—10 не превосходит $8 \cdot F \cdot P^2 \cdot N$, где N — общее число вершин всех типов, всех базовых окружений, всех возможных информационных. Учитывая, что $F \leq 16 \cdot 10^{18}$ и $P = 39$, получаем, что сочетаний базовых окружений и информационных не более $2749 \cdot 181 \cdot 10^6$. Значит, число введённых соотношений для локальных преобразований 7—10 не превосходит $13 \cdot 10^{33}$.

8. Локальные преобразования при выходе в подклеенную макроплитку

В предыдущем разделе мы описали, как вводятся локальные преобразования (аналог определяющих соотношений) для плоских участков путей. В этом разделе мы опишем, как это сделать для путей, содержащих выход в подклеенную макроплитку. Мы рассмотрим 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 рёбрам входа и выхода. Хотя бы одно из рёбер входа и выхода отвечает ребру, выходящему в подклейку.

Для начала докажем следующее утверждение, позволяющее узнать рёбра входа и выхода в пути из трёх узлов, для случая, когда средний узел является ядром некоторой подклейки.

Ядро подклейки

Пусть Y — ядро некоторой подклеенной макроплитки T . Напомним, что в каждом узле макроплитки 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 рёбер, которые лежат на сторонах T . Например, если T подклеена по рёбрам 1 и 3 к узлу типа \mathbb{A} с окружением $(7\mathbb{A}, 7\mathbb{A}, 3\mathbb{B}, 4\mathbb{A})$, информацией \mathbf{Info} , то флаг подклейки для узлов T выглядит как $[\mathbb{A}, \mathbf{Info}, (7\mathbb{A}, 7\mathbb{A}, 3\mathbb{B}, 4\mathbb{A}), 1, 3]$.

Этот параметр (для узла X внутри T) обозначается через $\mathbf{Core}(X)$. Тип выходящего из ядра ребра, соответствующего верхней стороне T , обозначается через $\mathbf{T.Core}(X)$, ребра, являющегося левой стороной T , через $\mathbf{L.Core}(X)$. Для окружения или типа Y мы используем напрямую обозначение $\mathbf{Core}(X)$.

Лемма 8.1 (о рёбрах около ядра макроплитки). Пусть XYZ — некоторый плоский путь, причём узел Y — 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 рёбер входов и выходов можно выписать рёбра выхода из X и входа в Z (приведены в таблицах 5–8 для разных типов Y).

Таблица 5. Рёбра выхода для A -узла

ребро входа в Y (или выхода из Y)	ребро выхода из X (или входа в Z)
1	1
2	2
3	2
lu	2
ld	2

Таблица 6. Рёбра выхода для 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

Таблица 7. Рёбра выхода для 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 8. Рёбра 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

Таким образом, зная хотя бы один узел подклеенной макроплитки T (из тех, которые лежат внутри неё, т. е. не на верхней или левой стороне), мы можем узнать рёбра входа и выхода для путей вдоль левой или верхней стороны T . В дальнейшем, если нам потребуется выписать типы рёбер вдоль левой или верхней стороны подклеенной макроплитки, мы будем использовать ссылку на лемму 8.1.

Лемма 8.2 (об информации узла рядом с ядром макроплитки). Пусть XU — некоторый плоский путь, причём узел U — ядро подклеенной макроплитки, на одной из сторон которой лежит X . Будем считать, что нам известен код U (тип, уровень, окружение и информация), а также тип входящего в U 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 — A . Рассмотрим ребро входа в U . Есть пять возможных типов рёбер для узла типа A , это рёбра 1, 2, 3, **lu**, **ld**. Если ребро входа в U — это 1, 2 или 3, информация у X будет такой же, как у U . Для ребра **lu** первым начальником X будет **LevelPlus.FBoss**(U), вторым — **Next.FBoss**(U), третьим — сам узел U . В случае **ld** первым начальником X будет **BottomLeftChain**(U), вторым — C с окружением U , третьим — сам узел U .

Пусть тип U — B . Есть шесть возможных рёбер из узла типа B , это рёбра 1, 2, 3, **ru**, **rd**, **mid**. Если ребро входа в U имеет тип 1, 2 или 3, то информация у X такая же, как у U . Для ребра **rd** первым начальником будет узел в середине ребра 7 для макроплитки T , где U выступает в роли B -узла, т. е. 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 U (это входит в информацию для U). Второй начальник X — это узел в середине правой стороны T , т. е. первый начальник U — это **TopFromRight.SBoss**(X). Третьим начальником X будет сам узел U . 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 **TopRightType**(U). Второй начальник X совпадает с первым начальником U . Третьим начальником X будет сам узел U . Для ребра **mid** первым начальником будет 1-цепь вокруг A -узла T (т. е. окружение такое же, как у U) с указателем 1. Второй начальник X — это C -узел T (окружение такое же, как у U). Третьим начальником X будет сам узел U .

Пусть тип U — C . Есть десять возможных рёбер из узла типа C (4 главных и 6 неглавных). Можно заметить, что во всех случаях окружения начальников X могут быть восстановлены по информации и окружению U , этот процесс полностью аналогичен описанному выше.

Пусть теперь X имеет боковой тип. Если входящее в U ребро главное, то X и U лежат на одном ребре в некоторой макроплитке, и поэтому у них одинаковая информация. Пусть входящее в U 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 U , при этом для u_1 тип второго начальника для X может быть восстановлен по информации и окружению U . Для l первым и единственным начальником X будет **Prev**(U), а для r первый начальник X может быть восстановлен по процедуре **TopFromRight**(U),

тип второго — по процедуре **BottomRightTypeFromRight**(Y). Для остальных рёбер все начальники X восстанавливаются аналогично случаям внутренних вершин Y . \square

8.1. Обзор случаев

Пусть есть путь, участвующий в локальном преобразовании, лежащий в макроплитке T , причём одно из его рёбер — ребро, входящее в подклеенную макроплитку. Нужно показать, как происходит преобразование во всех случаях. Заметим, что один конец ребра в подклеенную область всегда лежит на верхней или левой стороне макроплитки.

Итак, нужно рассмотреть случаи локальных преобразований, где макроплитка T примыкает к границе левой или верхней стороны подклеенной макроплитки. Таким образом, нужно рассмотреть те из случаев B1–B20, которые включают граничные стороны **top** или **left** (табл. 9). Пронумеруем их как P1–P10 и рассмотрим каждый отдельно.

Таблица 9. 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),	$x = \mathbf{left, top}$	P3
B4	(7A, x, 1A, 3A),	$x = \mathbf{left, top}$	P4
B5	(top, right, 6A, 2A)		P5
B6	(top, right, 1A, 3A)		P6
B11	($x, \mathbf{3B, 6A, 2A}$),	$x = \mathbf{left, top}$	P7
B12	($x, \mathbf{3B, 1A, 3A}$),	$x = \mathbf{left, top}$	P8
B17	($x, \mathbf{1A, 6A, 2A}$),	$x = \mathbf{left, top}$	P9
B18	($x, \mathbf{1A, 1A, 3A}$),	$x = \mathbf{left, top}$	P10

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 (рис. 107). В остальных случаях (3 и 6) путь не может содержать ребро в подклеенную область, так как пути не касаются левой или верхней стороны.

Замечание. Ниже мы не будем выписывать симметричное к введённому 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.

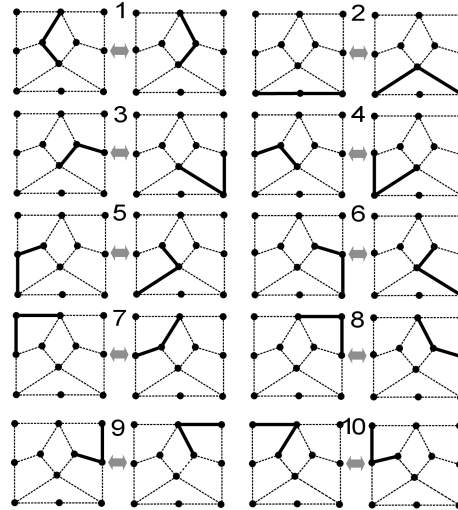


Рис. 107. Локальные преобразования

8.2. P1: Макроплитка с окружением (left, top, right, bottom)

В этом разделе мы обращаемся к случаю P1, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (**left, top, right, bottom**). Это происходит, когда макроплитка сама является только что подклеенной макроплиткой.

Зафиксируем узел Q — ядро 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 к нему не была бы подклеена макроплитка). Сторонами подклейки могут быть любые два ребра, выходящие из нашего узла.

Напомним, что в параметр «ядро подклейки» входят тип, уровень, расширенное окружение, информация и типы двух выходящих рёбер, являющихся сторонами подклейки, т. е. наше ядро подклейки задано.

Определив Q , можно определить и узлы Q_1, Q_2, Q_3, Q_4 , лежащие на сторонах подклейки, т. е. мы рассматриваем всевозможные сочетания кодов для пятёрки вершин, существующих на комплексе. Для каждой такой пятёрки можно определить буквы — коды вершин X, Y, F, Z для каждого из восьми указанных на рис. 108 случаев. Также введём обозначения $e_{t.core}$ и $e_{l.core}$ для типов двух выходящих рёбер, являющихся сторонами подклейки.

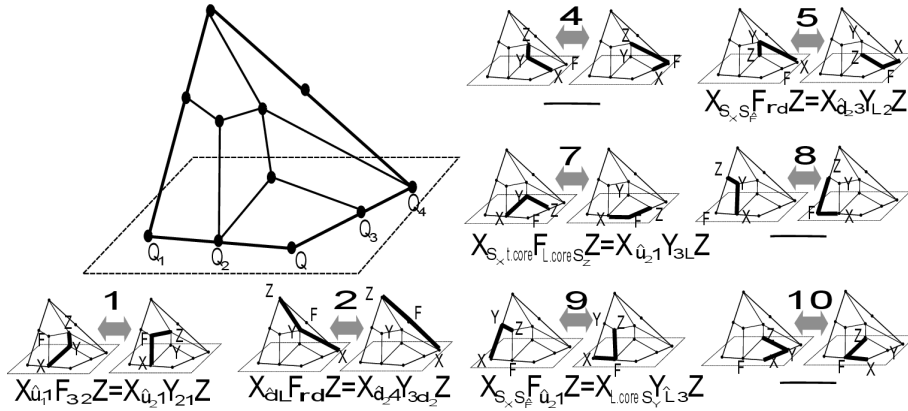


Рис. 108. Случай P1

Характеризация

По выданному нам коду пути $P_1 P_2 P_3$, включающему переход по подклеенному ребру, мы можем установить, встречается ли этот путь среди перечисленных на рис. 108. Действительно, первый узел после перехода по подклеенному ребру должен иметь параметр «ядро подклейки», соответствующий рассматриваемому нами. Ясно также, что по конфигурации входящих и выходящих рёбер и типов вершин можно выявить, какой именно случай из шестнадцати представленных на рис. 108 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 мёртвого 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 (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ .

Заметим, что коды X и Z мы и так знаем (из пути XYZ). Остаётся выписать рёбра выхода из X , ребро входа в F , код F , ребро выхода из F и ребро входа в Z (табл. 10).

Замечание. Через $\mathbf{T.Core}(Y)$ и $\mathbf{L.Core}(Y)$ мы обозначаем типы выходящих рёбер, соответствующих сторонам подклеенной макроплитки. Они содержатся в параметре «флаг подклейки» у вершины Y .

Таблица 10. 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	\hat{u}_1	1	\mathbb{B} (окруж. как у Y)	1-й босс — X	3	2
2	\hat{d}	1	\mathbb{D} , 1 (окруж. как у Y)	—	\mathbf{r}	\mathbf{d}
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	\mathbf{l}	3
7	лемма 8.1	T.Core (Y)	Core (Y)	Core (Y)	L.Core (Y)	лемма 8.1
9	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	\hat{u}_1	1

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X , F , Z , а также рёбра входа в F , Z и рёбра выхода из X , F вдоль пути XFZ), можно выписать код XYZ . Заметим, что коды X и Z мы и так знаем (из пути XFZ). Остаётся выписать рёбра выхода из X , ребро входа в Y , код Y , ребро выхода из Y и ребро входа в Z (табл. 11).

Таблица 11. 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	\hat{u}_2	1	\mathbb{A} (окружение как у F)	как у F	2	1
2	\hat{d}_2	4	\mathbb{C} (окружение как у F)	1-й босс как у Z , 2-й босс — X , 3-й — Z	3	\mathbf{d}_2
5	\hat{d}_2	3	\mathbb{C} (окружение как у Z)	1-й босс как у Z , 2-й босс — X , 3-й босс — \mathbb{CDR}	1	2
7	\hat{u}_2	1	\mathbb{A} , окружение — (left, right, top, bottom)	1-й босс — X	3	1
9	\hat{r}	2	\mathbb{R} , 1, окружение — (left, right, top, bottom)	—	\mathbf{r}	2

Когда все коды определены, можно выписать определяющие соотношения, переводящие один путь в другой.

Обозначим результат применения процедуры, описанной в лемме 8.1, к вершине Q , связанной с этим ребром, через s_Q . Через $s_{\bar{Q}}$ обозначим противоположное главное ребро, т. е. если $s_Q = 1$, то $s_{\bar{Q}} = 2$, и если $s_Q = 2$, то $s_{\bar{Q}} = 1$. Обозначение **Past**(X) используем для подклеенного окружения узла X .

Соотношения

Введём соотношения, реализующие описанные переходы.

1. $Xe_{\hat{u}_1}e_1Fe_3e_2Z = Xe_{\hat{u}_2}e_1Ye_2e_1Z.$
2. $Xe_{\hat{d}}e_1Fe_re_dZ = Xe_{\hat{d}_2}e_4Ye_3e_{d_2}Z.$
5. $Xe_{s_x}e_{s_f}Fe_1e_3Z = Xe_{\hat{d}_2}e_3Ye_1e_2Z.$
7. $Xe_{s_x}e_{t.core}Fe_{1.core}e_zZ = Xe_{\hat{u}_2}e_1Ye_3e_1Z.$
9. $Xe_{s_x}e_{s_f}Fe_{\hat{u}_1}e_1Z = Xe_{\hat{e}_u}Ye_re_2Z.$

8.3. P2: Макроплитка с окружением (left, top, 1A, 3A)

В этом разделе мы обращаемся к случаю 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 рис. 109, т. е. Q является ядром подклейки, а остальные вершины лежат на сторонах.

Для зафиксированного сочетания кодов этих пяти вершин можно вычислить коды всех путей, указанных на рис. 109. Кроме того, по выданному слову (коду пути) можно установить, действительно ли этот путь относится к случаю P2 и какую конфигурацию из изображённых на рис. 109 он имеет. Действительно, по окружению узлов после перехода по подклеенному ребру можно установить, что мы имеем дело именно со случаем P2, а последовательность входящих и выходящих рёбер помогает установить нужную конфигурацию.

Теперь покажем, как по этому коду определить код парного к нему пути.

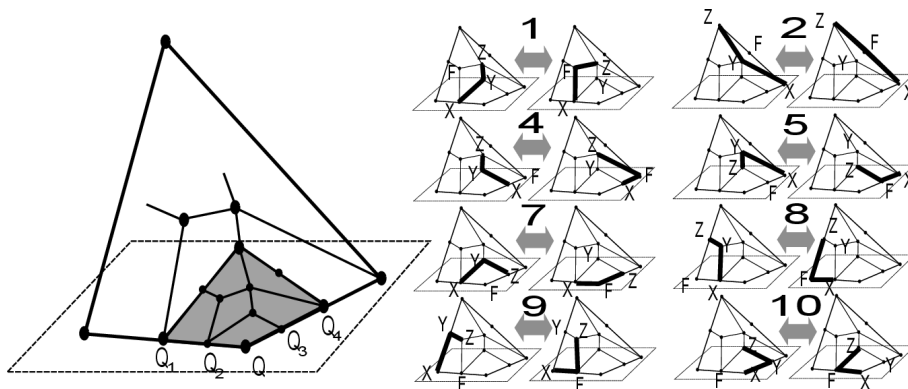


Рис. 109. Случай P2

Вычисление кода XFZ

Аналогично случаю P1, зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 12).

Таблица 12. 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-й босс — X	3	2
2	$\hat{1}$	1	\mathbb{DR} , 1, окружение ребро 3	как у Z	2	3
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	l_2	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	1	3
7	лемма 8.1	T.Core (Y)	Core (Y)	Core (Y)	L.Core (Y)	лемма 8.1
8	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	u_2	2
9	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	\hat{u}_1	1
10	Top.Core (Z)	лемма 8.1	FBoss (Z)	лемма 8.2	\hat{u}_2	1

Таблица 13. 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-й босс — X	2	1
2	\hat{d}_2	4	\mathbb{C} (окружение как у F)	1-й босс как у Z , 2-й босс — X , 3-й босс — Z	3	d_2
4	$\hat{1}$	3	\mathbb{A} (окружение как у Z)	1-й босс как у Z	2	1
5	\hat{d}_2	3	\mathbb{C} (окружение как у Z)	1-й босс как у Z , 2-й босс — X , 3-й босс — \mathbb{A} , окружение — Past (X)	1	2
7	\hat{u}_2	1	\mathbb{A} , окружение — (left, right, 1A, 3A)	1-й босс — X	3	1
8	\hat{u}_1	1	\mathbb{B} , окружение — (left, right, 1A, 3A)	1-й босс — X тип 2-го — \mathbb{A}	2	r
9	\hat{u}_2	2	\mathbb{RU} , окружение — 0-цепь вокруг \mathbb{A} (ук. 1) с окружением Past (X), уровень 1	1-й босс — X	r	2
10	L.Core (Z)	лемма 8.1	Next.Past.FBoss (Z)	лемма 8.2	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также ребра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 13).

Замечание. Запись **Next.Past.FBoss**(Z) означает, что берётся следующий узел в цепи для первого начальника Z , причём рассматривается его подклеенное окружение (а не основное).

Соотношения

Соотношения вводятся аналогично случаю P1, коды вершин и рёбер указаны в таблицах выше.

8.4. P3: Макроплитка с окружением (7A, x , 3B, 4A), $x = \text{left, top}$

В этом разделе мы обращаемся к случаю 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 четыре вершины Q, Q_1, Q_2, Q_3 , занимающие на комплексе положения, как на рис. 110, т. е. Q является ядром подклейки, а остальные вершины лежат на сторонах.

Для зафиксированного сочетания кодов этих четырёх вершин можно вычислить коды всех путей, указанных на рис. 110. Кроме того, по выданному слову (коду пути) можно установить, действительно ли этот путь относится к случаю P2 и какую конфигурацию из изображённых на рис. 110 он имеет.

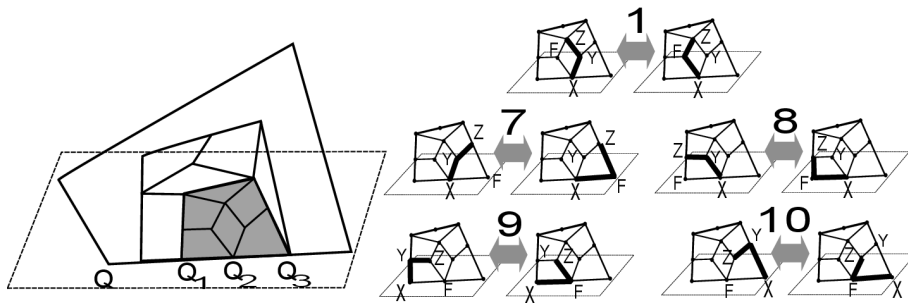


Рис. 110. Случай P3

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ . Обозначение $\mathbf{Past}(X)$ используем для подклеенного окружения узла X (табл. 14).

Таблица 14. 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} (окруж. как у Y)	1-й босс — X тип 2-го — \mathbb{A}	3	2
7	лемма 8.1	лемма 8.1	$\mathbf{SBoss}(Z)$	как у X	\hat{u}_3 , если $\mathbf{Past}(F) = \mathbb{U}$; $\hat{1}_2$, если $\mathbf{Past}(F) = \mathbb{L}$	1
8	лемма 8.1	лемма 8.1	$\mathbf{Next.FBoss}(Z)$	как у X	1	1
9	лемма 8.1	лемма 8.1	$\mathbf{FBoss}(Z)$	как у X	\hat{u}_1	1
10	лемма 8.1	лемма 8.1	$\mathbf{FBoss}(Z)$	как у X	\hat{u}_2	1

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 15).

Таблица 15. 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} (окруж. как у F)	1-й босс — X	2	1
7	\hat{u}_2	1	\mathbb{A} , окружение — $(7\mathbb{A}, x, 3\mathbb{B}, 4\mathbb{A})$, $x = \mathbf{left, top}$	1-й босс — X	3	1
8	\hat{u}_1	1	\mathbb{B} , окружение — $(7\mathbb{A}, x, 3\mathbb{B}, 4\mathbb{A})$, $x = \mathbf{left, top}$	1-й босс — X	2	\mathbf{r}
9	$\hat{1}$	2	$\mathbb{DR}, 1$, окружение — 3	1-й босс — Y	\mathbf{r}	2
10	$\hat{1}_2$ или \hat{u}_1	1	$\mathbf{Next.Past.FBoss}(Z)$	1-й босс — $\mathbf{TopFromCorner.Past}(X)$ 2-й босс — X , 3-й босс — $\mathbf{RightCorner.Past}(X)$	1	3

Замечание. Ребро из X в случае 10 имеет тип $\hat{1}_2$, если $\mathbf{Past}(X) = \mathbb{L}$, и \hat{u}_1 , если $\mathbf{Past}(X) = \mathbb{U}$.

Запись $\mathbf{Next.Past.FBoss}(Z)$ означает, что берётся следующий узел в цепи для первого начальника Z , причём рассматривается его подклеенное окружение (а не основное).

8.5. P4: Макроплитка с окружением (7A, x , 1A, 3A), $x = \text{left, top}$

В этом разделе мы обращаемся к случаю 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 — прямой потомок 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.

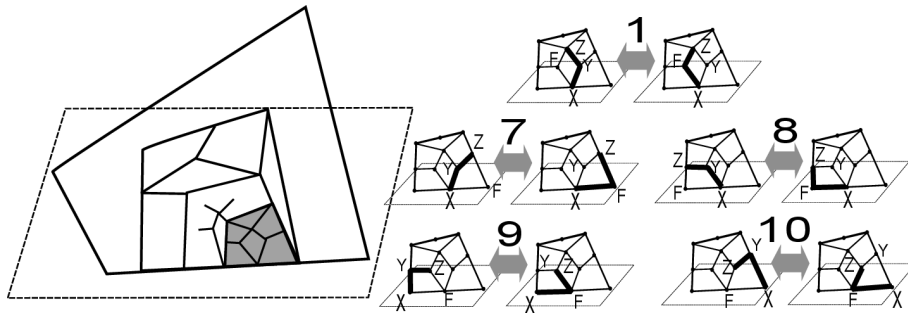


Рис. 111. Случай P4

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 16).

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 17).

Замечание. Ребро из X в случае 10 имеет тип $\hat{1}_2$, если $\mathbf{Past}(X) = \mathbb{L}$, и \hat{u}_1 , если $\mathbf{Past}(X) = \mathbb{U}$.

Таблица 16. 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} (окруж. как у Y)	1-й босс — X тип 2-го — \mathbb{A}	3	2
7	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	\hat{u}_3 , если Past (F) = \mathbb{U} ; $\hat{1}_2$, если Past (F) = \mathbb{L}	1
8	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	\hat{u}_2	2
9	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	\hat{u}_1	1
10	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	\hat{u}_2	1

Таблица 17. 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} (окруж. как у F)	1-й босс — X	2	1
7	\hat{u}_2	1	\mathbb{A} , окружение — (7A , x , 1A , 3A), $x = \text{left, top}$	1-й босс — X	3	1
8	\hat{u}_1	1	\mathbb{B} , окружение — (7A , x , 1A , 3A), $x = \text{left, top}$	1-й босс — X	2	r
9	\hat{u}_2	2	\mathbb{RU} , 1, окружение — $\mathbb{A}0$ -цепь (ук. 1) с окруж. Past (X)	1-й босс — X	r	2
10	$\hat{1}_2$ или \hat{u}_1	1	Next.Past.FBoss (Z)	1-й босс — TopFromCorner.Past (X), 2-й босс — X , 3-й босс — RightCorner.Past (X)	1	3

Запись **Next.Past.FBoss**(Z) означает, что берётся следующий узел в цепи для первого начальника Z , причём рассматривается его подклеенное окружение (а не основное).

8.6. P5: Макроплитка с окружением (top, right, 6A, 2A)

В этом разделе мы обращаемся к случаю 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.

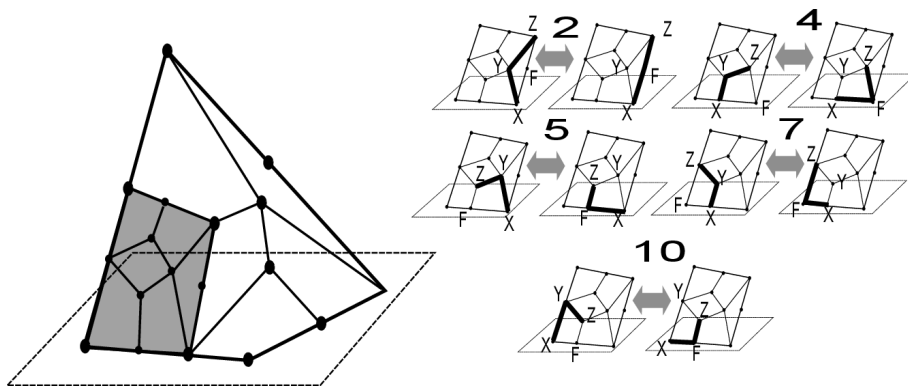


Рис. 112. Случай P5

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 18).

Для преобразования 7 пути удовлетворяют условиям мёртвого паттерна. Для них мы соотношения не вводим.

Таблица 18. Случай P5: вычисление кода XFZ

	Рёбро из X	Рёбро в F	Тип, уровень и окружение F	Информация F	Рёбро из F	Рёбро в Z
2	\hat{u}_1	1	$\mathbb{DR}, 1$, окружение — 3	1-й босс — X , тип 2-го — \mathbb{CDR}	1	2
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	\hat{u}_3	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3
10	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 19).

Таблица 19. Случай P5: вычисление кода XYZ

	Рёбро из X	Рёбро в Y	Тип, уровень и окружение Y	Информация Y	Рёбро из Y	Рёбро в Z
2	\hat{u}_3	4	C, окружение как у T	1-й босс — U, окружение как у T, (top, right, 6A, 2A), 2-й босс — X, 3-й босс — B, окружение (left, top, right, bottom)	2	1
4	$\hat{1}$	3	A, окружение как у T	1-й босс как у Z	2	1
5	\hat{u}_3	4	C, окружение как у T	1-й босс — как у Z, 2-й босс — X, 3-й босс — B, окружение (left, top, right, bottom)	1	2
10	\hat{r}	2	FBoss(Z)	—	u_2	1

8.7. P6: Макроплитка с окружением (top, right, 1A, 3A)

В этом разделе мы обращаемся к случаю P6, когда макроплитка T, внутри которой мы рассматриваем локальное преобразование пути, имеет окруже-

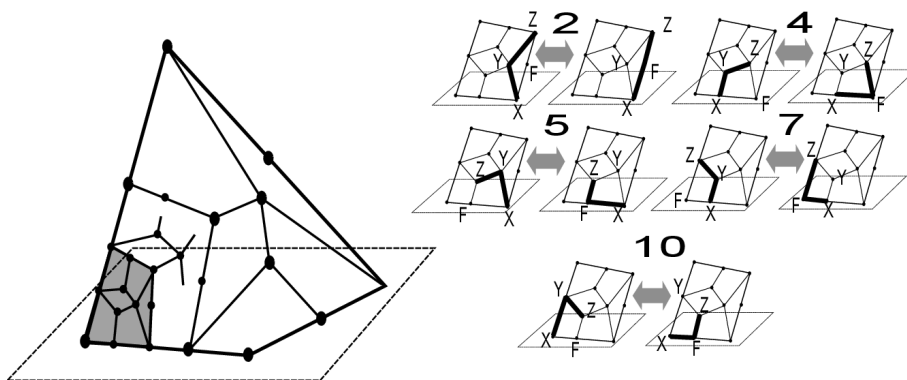


Рис. 113. Случай P6

ние (**top, right, 1A, 3A**). Это происходит, когда макроплитка — прямой поток некоторой макроплитки, являющейся правой верхней подплиткой некоторой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 20).

Для преобразования 7 пути удовлетворяют условиям мёртвого паттерна. Для них мы соотношения не вводим.

Таблица 20. Случай P6: вычисление кода XFZ

	Рёбро из X	Рёбро в F	Тип, уровень и окружение F	Информация F	Рёбро из F	Рёбро в Z
2	$\hat{1}$	1	$\mathbb{DR}, 1,$ окружение — 3	1-й босс как у Z	2	3
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	$\hat{1}_3$	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3
10	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 21).

Таблица 21. Случай P6: вычисление кода XYZ

	Рёбро из X	Рёбро в Y	Тип, уровень и окружение Y	Информация Y	Рёбро из Y	Рёбро в Z
2	$\hat{1}_2$	4	$\mathbb{C},$ окружение как у T	1-й босс — $\mathbb{U},$ окружение как у $T,$ 2-й босс — $X,$ 3-й босс — $\mathbb{A},$ окружение как у FBoss (F)	2	1
4	$\hat{1}$	3	$\mathbb{A},$ окружение как у T	1-й босс как у Z	2	1
5	\hat{u}_3	4	$\mathbb{C},$ окружение как у T	1-й босс как у $Z,$ 2-й босс — $X,$ 3-й босс — $\mathbb{A},$ окружение RightCorner.Past (X)	1	2
10	\hat{r}	2	FBoss (Z)	1-й босс — X	u_2	1

**8.8. P7: Макроплитка с окружением $(x, 3B, 6A, 2A)$;
 $x = \text{left, top}$**

В этом разделе мы обращаемся к случаю P7, когда макроплитка T , внутри которой мы рассматриваем локальное преобразование пути, имеет окружение $(x, 3B, 6A, 2A)$; $x = \text{left, top}$. Это происходит, когда макроплитка — правая верхняя подплитка некоторой макроплитки, являющейся левой нижней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

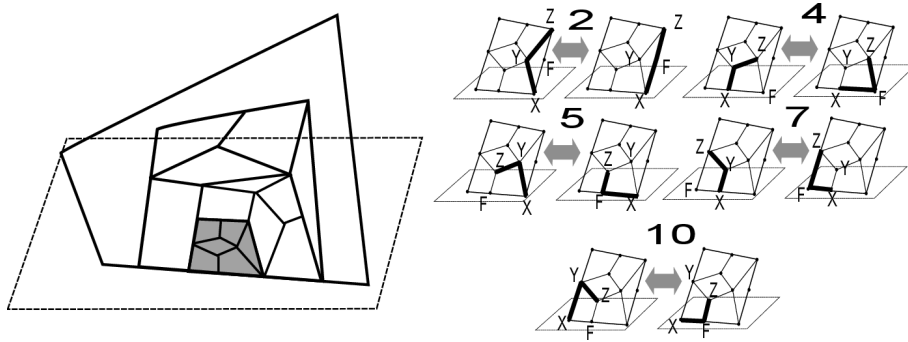


Рис. 114. Случай P7

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 22).

Таблица 22. 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-й босс — X , тип 2-го — \mathbb{A}	2	1
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	\hat{u}_3	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3
7	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	1
10	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 23).

Таблица 23. 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	C, окружение как у T	1-й босс — TopFromCorner.Past(X) , 2-й босс — X, 3-й босс — Z	3	ru
4	$\hat{1}$	3	A, окружение как у T	1-й босс как у Z	2	1
5	\hat{u}_3	4	C, окружение как у T	1-й босс как у Z, 2-й босс — X, 3-й босс — B, окружение (7A, x, 3B, 4A)	2	1
7	$\hat{1}$	3	A, окружение как у T	1-й босс — Z	1	u₂
10	$\hat{1}$	1	FBoss(Z)	1-й босс — Prev(X)	u₂	1

**8.9. P8: Макроплитка с окружением (x, 3B, 1A, 3A);
x = left, top**

В этом разделе мы обращаемся к случаю 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 — прямой потомок правой верхней подплитки некоторой макроплитки, являющейся левой нижней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

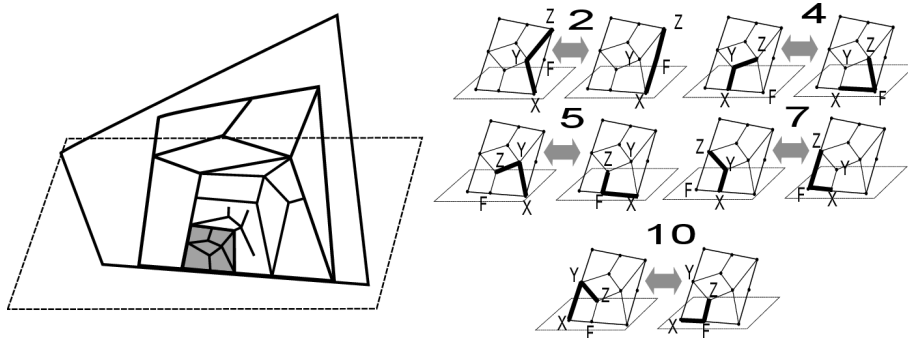


Рис. 115. Случай P8

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 24).

Таблица 24. 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	$\mathbb{DR}, 1,$ окружение — 3	1-й босс как у Z	2	3
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	$\hat{1}_2$	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3
7	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	1
10	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 25).

Таблица 25. 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-й босс — TopFromCorner.Past (X), 2-й босс — X , 3-й босс — Z	3	ru
4	$\hat{1}$	3	\mathbb{A} , окружение как у T	1-й босс как у Z	2	1
5	$\hat{1}_2$	4	\mathbb{C} , окружение как у T	1-й босс как у Z , 2-й босс — X , 3-й босс — \mathbb{A} , окружение RightCorner.Past (X)	1	2
7	$\hat{1}$	3	\mathbb{A} , окружение как у T	1-й босс — Z	1	u ₂
10	$\hat{1}$	2	FBoss (Z)	1-й босс — Prev (X)	u ₂	1

**8.10. P9: Макроплитка с окружением $(x, 1A, 6A, 2A)$;
 $x = \text{left, top}$**

В этом разделе мы обращаемся к случаю 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 — правая верхняя подплитка некоторой макроплитки, являющейся левой верхней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

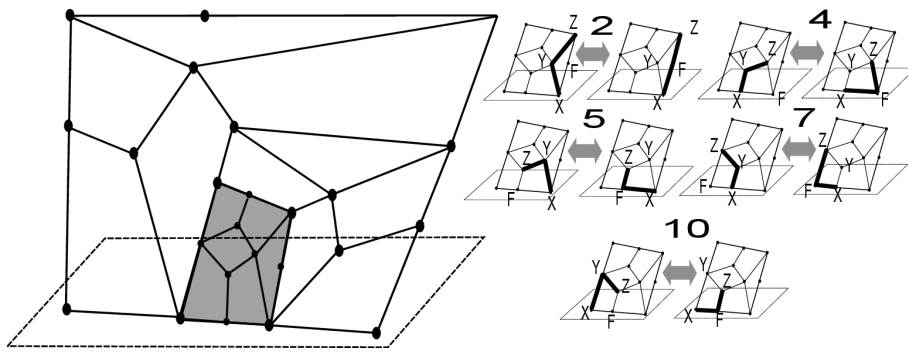


Рис. 116. Случай P9

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 26).

Таблица 26. 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-й босс — X , тип 2-го — \mathbb{A}	2	1
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	\hat{u}_3	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3
7	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	\hat{u}_2	1
10	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 27).

Таблица 27. Случай P9: вычисление кода XYZ

	Рёбро из X	Рёбро в Y	Тип, уровень и окружение Y	Информация Y	Рёбро из Y	Рёбро в Z
2	\hat{u}_3	4	C, окружение как у T	1-й босс — TopFromCorner.Past(X) , 2-й босс — X, 3-й босс — Z	3	ru
4	\hat{i}	3	A, окружение как у T	1-й босс как у Z	2	1
5	\hat{u}_3	4	C, окружение как у T	1-й босс как у Z, 2-й босс — X, 3-й босс — B, окружение (left, top, 1A, 3A)	1	2
7	\hat{i}	3	A, окружение как у T	1-й босс — Z	1	u₂
10	\hat{i}	1	FBoss(Z)	1-й босс — X	u₂	1

8.11. P10: Макроплитка с окружением (x, 1A, 1A, 3A); x = left, top

В этом разделе мы обращаемся к случаю P10, когда макроплитка T, внутри которой мы рассматриваем локальное преобразование пути, имеет окружение (x, 3B, 1A, 3A); x = left, top. Это происходит, когда макроплитка — прямой

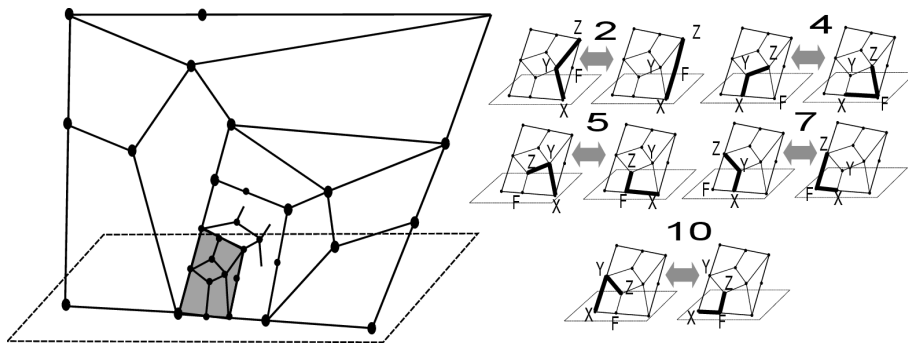


Рис. 117. Случай P10

потомок правой верхней подплитки некоторой макроплитки, являющейся левой верхней подплиткой подклеенной макроплитки.

Определение кодов вершин можно провести аналогично предыдущим случаям.

Вычисление кода XFZ

Зная код пути XYZ (т. е. коды X, Y, Z , а также рёбра входа в Y, Z и рёбра выхода из X, Y вдоль пути XYZ), можно выписать код XFZ (табл. 28).

Таблица 28. 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	$\mathbb{D}\mathbb{R}$, окружение -3	1-й босс как у Z	2	3
4	лемма 8.1	лемма 8.1	SBoss (Z)	как у X	$\hat{1}_2$	4
5	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3
7	лемма 8.1	лемма 8.1	FBoss (Z)	как у X	\hat{u}_2	2
10	лемма 8.1	лемма 8.1	Next.FBoss (Z)	как у X	$\hat{1}$	3

Вычисление кода XYZ

Зная код пути XFZ (т. е. коды X, F, Z , а также рёбра входа в F, Z и рёбра выхода из X, F вдоль пути XFZ), можно выписать код XYZ (табл. 29).

Таким образом, во всех случаях можно провести локальное преобразование, как и для плоских путей. Это даёт нам возможность преобразовывать пути, ме-

Таблица 29. 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-й босс — LevelPlus.FBoss (Z), 2-й босс — X , 3-й босс — Z	3	ru
4	$\hat{1}$	3	\mathbb{A} , окружение как у T	1-й босс как у Z	2	1
5	$\hat{1}_2$	4	\mathbb{C} , окружение как у T	1-й босс как у Z , 2-й босс — X , 3-й босс — \mathbb{A} , окружение RightCorner.Past (X)	1	2
7	$\hat{1}$	3	\mathbb{A} , окружение как у T	1-й босс — Z	1	u ₂
10	\hat{u}_2	2	FBoss (Z)	1-й босс — X	u ₂	1

няя локальные участки на им эквивалентные, и приводить слово к канонической форме.

8.12. Оценка числа введённых соотношений

В каждом случае мы выбирали ядро подклейки, после чего выбирали не более четырёх вершин (Q_1, Q_2, Q_3, Q_4) на сторонах подклеенной макроплитки, для которых фиксировали их коды. Затем мы 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 рёбрам-сторонам, т. е. будет не более чем $16 \cdot F \cdot H^4$ соотношений, где F — количество флагов подклейки, H — число расширенных окружений, т. е. не более чем $16 \cdot 82485^4 \cdot 5 \cdot 10^{18} < 4 \cdot 10^{39}$ соотношений.

Литература

- [1] Иванов-Погодаев И. А., Канель-Белов А. Я. Конечно определённая нильполугруппа: комплексы с равномерной эллиптичностью // Изв. РАН. Сер. матем. — 2021. — Т. 85, № 6. — С. 126—163.
- [2] Berger R. The undecidability of the domino problem: Ph.D. Thesis. — Harvard University, 1964.
- [3] Conway J., Lagarias J. Tiling with polyminoes and combinatorial group theory // J. Combin. Theory. Ser. A. — 1990. — Vol. 53, no. 2. — P. 183—208.
- [4] Durand B., Levin L., Shen A. Local rules and global order, or aperiodic tilings // Math. Intelligencer. — 2005. — Vol. 27. — P. 64—68.
- [5] Goodman-Strauss Ch. Matching rules and substitution tilings // Ann. Math. (2). — 1998. — Vol. 147, no. 1. — P. 181—223.
- [6] Gromov M. Hyperbolic groups // Essays in Group Theory / S. M. Gersten, ed. — Springer, 1987. — (MSRI; Vol. 8). — P. 75—263.
- [7] Ivanov-Pogodaev I., Malev S., Sapir O. A construction of a finitely presented semigroup containing an infinite squarefree ideal with zero multiplication // Int. J. Algebra Comput. — 2018. — Vol. 28, no. 8. — P. 1565—1573.
- [8] Kari J., Papasoglu P. Deterministic aperiodic tile sets // Geom. Funct. Anal. — 1999. — Vol. 9. — P. 353—369.
- [9] Mozes Sh. Tilings, substitution systems and dynamical systems generated by them // J. Anal. Math. — 1989. — Vol. 53, no. 1. — P. 139—186.
- [10] Robinson R. M. Undecidability and nonperiodicity for tilings of the plane // Invent. Math. — 1971. — Vol. 12. — P. 177—209.
- [11] Wang Hao. Proving theorems by pattern recognition. II // Bell System Tech. J. — 1961. — Vol. 40, no. 1. — P. 1—41.
- [12] Wise D. Non-positively curved squared complexes, aperiodic tilings, and non-residually finite groups: Ph.D. Thesis. — Princeton University, 1996.
- [13] Wise D. Cubulating small cancellation groups // Geom. Funct. Anal. — 2004. — Vol. 14. — P. 150—214.