Алгоритмы, о которых будет идти речь в данной главе, связаны с определением различных свойств множеств прямоугольников (или ортогональных отрезков). Можно предложить одну фундаментальную классификацию этих алгоритмов, зависящую от базисного типа операций, т. е. зависящую от того, будем ли мы работать в статической или динамической среде. В первом случае вся информация, необходимая для решения задачи, полностью известна до начала вычислений; во втором случае допустимы также динамические корректировки данных путем операций вставки и удаления.

Операции этих двух типов — статические и динамические — подразумевают, как и следовало ожидать, использование существенно различающихся структур данных. В то время как работа в статическом режиме в настоящее время хорошо изучена, исследование более сложных динамических операций пребывает в стадии активного развития. Поэтому целесообразно дать здесь детальное описание алгоритмов именно статического типа; отсылки к продолжающимся исследованиям алгоритмов динамического типа будут приведены в конце настоящей главы.

Последующее изложение будет относиться в основном к двумерным приложениям. Но там, где это уместно, будет дано обобщение на d-мерный случай либо о нем будет упоминаться.

8.3. Общие замечания по алгоритмам статического типа

Уникальная особенность набора изотетичных прямоугольников (или, что то же самое, двух ортогональных наборов параллельных отрезков) состоит в том, что плоскость разбивается на полосы (например, на вертикальные), в каждой из которых задача становится одномерной. (Аналогично пространство E^d можно разбить на $(d-1)$-мерные бруски.) В частности, если взять абсциссы вертикальных сторон этих прямоугольников, то внутри каждой полосы, заключенной между смежной парой этих абсцисс, все вертикальные сечения будут одинаковыми и будут состоять из ординат тех горизонтальных сторон, которые пересекают эту полосу. Поэтому такие вертикальные сечения на плоскости разбиваются на классы эквивалентности (каждый из этих классов будет множеством сечений, заключенных между парой последовательных абсцисс вертикальных сторон прямоугольников). Кроме того, будет показано, что сечение, принадлежащее определенной полосе, можно получить в результате небольшой модификации сечений, принадлежащих одной или двум смежным полосам.

Некоторые из интересных с вычислительной точки зрения задач, связанных с множеством прямоугольников (таких, как определение площади их объединения, периметра этого объединения, его контура, регистрации пересечений и т. д.), обладают одним очень полезным свойством. Произвольная вертикальная прямая определяет собой две полуплоскости; решение же исходной задачи, как правило, соотносится весьма просто с решениями аналогичных подзадач для каждой из этих полуплоскостей. Это простое соотношение иногда выступает в виде теоретико-множественного объединения (для отчета о пересечениях), или арифметической суммы (для вычисления площади и периметра), или простого сцепления компонент (для контура). Во всех этих случаях пересечение разделяющей прямой с множеством прямоугольников содержит всю необходимую для комбинации частных решений информацию. Более того, предполагается, что решение, получаемое в результате реализации операций статического типа для, скажем, левой полуплоскости является *окончательным* (т. е. не изменяемым корректировками, возможными *в будущем*), так что глобальное решение можно получить *последовательным* расширением текущего решения вправо. Мы видим, что это типичная ситуация, когда оправданно использование метода плоского заметания, а именно: заметающая прямая параллельна одному из направлений, а движение осуществляется вдоль перпендикулярного ему направления.

Обращая внимание на характерное вертикальное сечение множества прямоугольников, которое является просто последовательностью ординат, заметим, что, поскольку пересеченные отрезки параллельны, последовательность ординат этого сечения всегда будет подпоследовательностью упорядоченной последовательности ординат всех горизонтальных отрезков. Путем предварительной нормализации, при которой множество ординат сортируется, а каждая ордината заменяется ее номером в этом упорядочении, последовательность ординат можно уподобить ряду последовательных целых чисел. Поэтому кажется, что для поддержки статуса заметающей прямой нет необходимости в повторной сортировке общей отсортированной последовательности, поскольку можно воспользоваться более эффективными структурами данных, подобранными именно для этой ситуации. Такие структуры полностью определяются (по крайней мере скелетно) множеством ординат *всех* горизонтальных отрезков, а каждое отдельное сечение можно получить, пользуясь «скелетной» метафорой, путем «наделения плотью» части скелета. Одним из примеров такой структуры является *дерево отрезков*, которое уже изучалось и использовалось ранее в данной книге (см. разд. 1.2.3.1); другой пример — *де-*

рево интервалов [McCreight (1981); Edelsbrunner (1980)], которое будет описано в разд. 8.8.1.

Для удобства напомним, что каждый узел v из дерева отрезков характеризуется интервалом $[B[v], E[v]]$ и некоторыми дополнительными параметрами, необходимыми для выполнения особых вычислений. Один из этих параметров $C[v]$ — *степень* узла — возникает во всех приложениях, поскольку он обозначает число отрезков, отнесенных к узлу v; другие же параметры имеют смысл только для тех или иных приложений.

Элементарными операциями на дереве отрезков являются вставка и удаление. В частности, если $[b, e]$ обозначает интервал, то на узле v определены операции ВСТАВИТЬ$(b, e; v)$ и УДАЛИТЬ$(b, e; v)$. Обычно в следующих разделах процедура ВСТАВИТЬ будет иметь такой вид:

```
        procedure ВСТАВИТЬ(b,e; v)
1. begin if (b ≤ B[v]) and (E[v] ≤ e) then C[v] := C[v] + 1
2.       else begin if (b < ⌊B[v] + E[v])/2⌋) then
                        ВСТАВИТЬ(b,e; ЛСЫН[v]),
3.                  if (⌊(B[v] + E[v])/2⌋ < e) then
                        ВСТАВИТЬ(b,e; ПСЫН[v])
             end;
4.       КОРРЕКТИРОВАТЬ(v) (* корректировка особых
         параметров v *)
   end.
```

Важнейшей характеристикой этой процедуры, изменяющейся от приложения к приложению, является функция КОРРЕКТИРОВАТЬ (v) из строки 4, частные случаи которой будут обсуждаться в соответствующих местах. Аналогичные соображения применимы и к процедуре УДАЛИТЬ$(b, e; v)$.

В заключение заметим, что метод плоского заметания и использование структур данных, ориентированных на хранение интервалов, являются естественными средствами для решения множества задач, возникающих в геометрии прямоугольников. Приведенные ниже разделы посвящены детальному разбору специальных методов.

8.4. Мера и периметр объединения прямоугольников

В одной статье [Klee (1977)], которая представляет заметный интерес и может считаться началом целого направления исследований, Кли поставил следующий вопрос: «Заданы N интервалов $[a_1, b_1], \ldots, [a_N, b_N]$ на действительной прямой, надо найти меру их объединения. Какой может быть трудоемкость этой процедуры?».

Такая постановка дает простейший (одномерный) пример задачи о мере объединения. (Будем называть задачу о вычислении меры объединения для произвольного числа измерений задачей О.1.[1]) Кли сразу же построил алгоритм решения этой задачи с оценкой $O(N \log N)$, основанный на предварительной сортировке абсцисс $a_1, b_1, \ldots, a_N, b_N$ в массиве $X[1:2N]$. Дополнительное свойство этого массива состоит в том, что если a_i помещено в $X[h]$, b_j — в $X[k]$ и $a_i = b_j$, то $h < k$ (т. е. правая концевая точка помещается в нем после левой точки, обладающей такой же абсциссой). Вычисление завершается простым просмотром массива $X[1:2N]$ за линейное время (нижеследующий алгоритм представляет собой адаптацию алгоритма Кли, удобную для последующих обобщений):

```
      procedure МЕРА-ОБЪЕДИНЕНИЯ-ИНТЕРВАЛОВ
begin X[1 : 2N] := упорядоченная последовательность
      абсцисс концов интервалов;
      X[0] := X[1];
      m := 0; (* m — мера *)
      C := 0; (* C — число перекрывающихся интервалов *)
      for i := 1 until 2N do
          begin if (C ≠ 0) then m := m + X[i] − X[i − 1];
                if (X[i] — левая концевая точка) then
                C := C + 1 else C := C − 1
          end
end.
```

В своей статье Кли также отметил, что, хотя сортировка является ключом к получению вышеизложенного результата, она не является *априорно* необходимой, и поставил вопрос о существовании какого-нибудь метода решения, требующего $o(N \log N)$ операций.

Ответ на этот вопрос был получен Фредменом и Вайде [Fredman, Weide (1978)] для модели вычислений с линейным деревом решений (к которой относится и вышеизложенный алгоритм); применение общего метода Бен-Ора (разд. 1.4) расширяет пределы этого результата на произвольные алгебраические деревья решений. Закономерно, что доказательства подобного типа укладываются в уже знакомую схему, встречавшуюся ранее в связи с исследованиями выпуклых оболочек (разд. 3.2), максимумов на множестве векторов (разд. 4.1.3), уникальности элементов (разд. 5.2) и максимального промежутка (разд. 6.4). В основе лежит симметрическая группа степени N, которая вновь является ключом к доказательству. Здесь будет использован следующий вычислительный прототип:

---

[1]) О — аббревиатура слова «ортогональность». — *Прим. перев.*

Задача О.2 (ε-БЛИЗОСТЬ). Дано $N+1$ действительных чисел $x_1, x_2, \ldots, x_N$ и $\varepsilon > 0$. Надо определить, будут ли какие-нибудь два числа x_i и x_j $(i \neq j)$ находиться на расстоянии, меньшем чем ε одно от другого.

Прежде всего покажем, что можно легко установить преобразование

ε-БЛИЗОСТЬ $\propto_N$ МЕРА ОБЪЕДИНЕНИЯ ИНТЕРВАЛОВ.

Действительно, построим интервалы $[x_i, x_i+\varepsilon]$ для $i = 1, 2, \ldots, N$. Эти интервалы будут входом для процедуры МЕРА-ОБЪЕДИНЕНИЯ, результатом работы которой будет вещественное число m. Теперь ясно, что никакие два числа из

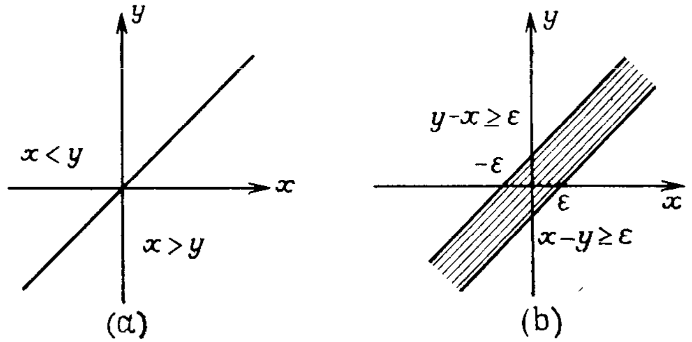

Рис. 8.7. Иллюстрация (в случае E^2) различия между множествами истинности в задачах об УНИКАЛЬНОСТИ ЭЛЕМЕНТОВ (a) и ε-БЛИЗОСТИ (b).

$\{x_1, \ldots, x_N\}$ не будут находиться на расстоянии, меньшем чем ε друг от друга, тогда и только тогда, когда $m = N\varepsilon$.

Существует близкое сходство между ε-БЛИЗОСТЬЮ и УНИКАЛЬНОСТЬЮ ЭЛЕМЕНТОВ, изложенной в разд. 5.2. Однако УНИКАЛЬНОСТЬ ЭЛЕМЕНТОВ не является частным случаем ε-БЛИЗОСТИ, поскольку значение $\varepsilon = 0$, необходимое для получения указанного частного случая, недопустимо. Различие этих задач выразительно проиллюстрировано на рис. 8.7 для двумерного случая: если речь идет об УНИКАЛЬНОСТИ ЭЛЕМЕНТОВ, то непересекающиеся связные компоненты множества истинности W (см. разд. 1.4 и 5.2) представляют собой открытые множества, разделенные только их общей границей; если же речь идет об ε-БЛИЗОСТИ, то компоненты представляют собой замкнутые множества и расстояние между ними положительно. Если исключить это различие, то доказательство того, что множество W для задачи об ε-БЛИЗОСТИ распадается на $N!$ непересекающихся компонент связности, будет аналогично доказательству, приведенному в разд. 5.2; отсюда вытекает следствие из теоремы 1.2:

Следствие 8.1. *Любой алгоритм, который использует модель алгебраического дерева решений, для определения существования некоторой пары чисел из N-элементного множества, удаленных друг от друга на расстояние, меньшее чем ε, затратит $\Omega(N \log N)$ проверок.*

Следствие 8.1 доказывает оптимальность результата Кли для задачи о вычислении меры в случае одного измерения, но оставляет открытым вопрос об оптимальной оценке при $d \geqslant 2$.

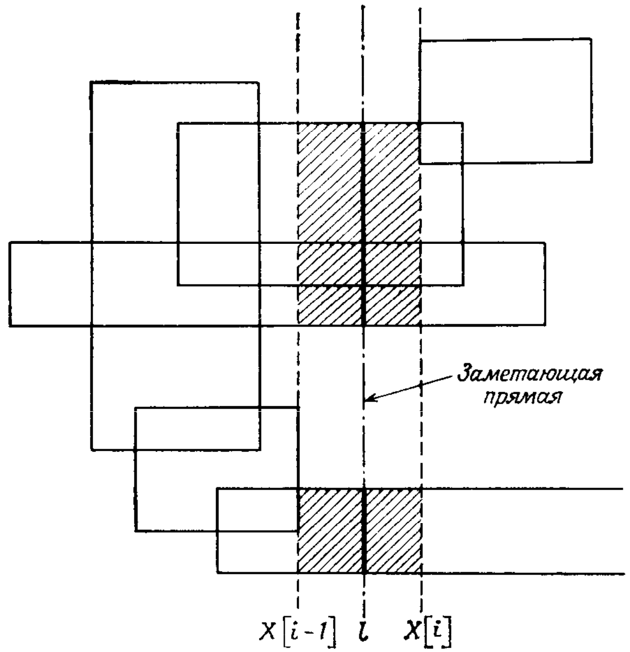

Рис. 8.8. Применение метода плоского заметания к задаче определения меры объединения прямоугольников для случая двух измерений.

Бентли [Bentley (1977)] [1]) взялся за эту задачу и преуспел в создании оптимального алгоритма для вычисления меры объединения при $d = 2$. Его подход является простой и разумной модификацией одномерного метода. В частности, при одном измерении длина интервала $[X[i-1], X[i]]$ добавляется к искомой мере (строка 6 в алгоритме МЕРА-ОБЪЕДИНЕНИЯ-ИНТЕРВАЛОВ) в зависимости от того, будет ли хотя бы один отрезок накрывать его или нет, или, что то же самое, будет ли параметр C ненулем или нулем соответственно. Следовательно,

[1]) Этот результат изложен также в работе [Van Leeuwen, Wood (1981)].

необходимо знать только значение C (строка 7). В случае двух измерений (рис. 8.8) вертикальная полоса, заключенная между $X[i-1]$ и $X[i]$, вносит в меру объединения прямоугольников вклад, равный величине $(X[i]-X[i-1])m_i$, где m_i — длина пересечения произвольной вертикальной прямой из этой полосы с объединением прямоугольников. Поэтому величина m_i (являющаяся константой и равная 1 в случае одного измерения) — это ключевой параметр для двумерной техники. Если бы m_i можно было вычислять и корректировать за время, не превышающее $O(\log N)$, то времена сканирования и предварительной обработки были бы равны и был бы получен оптимальный алгоритм с оценкой $\theta(N \log N)$.

Поскольку ординаты горизонтальных сторон прямоугольников известны *заранее*, то определение m_i можно реализовать посредством *дерева отрезков* (см. разд. 8.3). Вспоминая общий формат элементарных операций на дереве отрезков ВСТАВИТЬ И УДАЛИТЬ (см. разд. 8.3), определим для текущего приложения следующий дополнительный параметр узла:

$m(v)$:= вклад интервала $[B[v], E[v]]$ в величину m_i.

Вычисление $m[v]$ проводится следующей процедурой (вызываемой в строке 4 из процедуры ВСТАВИТЬ):

```
        procedure КОРРЕКТИРОВАТЬ(v)
begin   if (C[v] ≠ 0) then m[v] := E[v] − B[v]
        else if (v не лист) then m[v] := m[ЛСЫН[v]] +
                                        m[ПСЫН[v]]
             else m[v] := 0
end.
```

Ясно, что $m_i = m$ [корень дерева отрезков]. Как общий параметр $C[v]$, так и специальный параметр $m[v]$ можно легко корректировать (за константное время на один узел) при вставке или удалении одного отрезка (вертикальной стороны прямоугольника) из дерева отрезков. Итак, корректировку дерева отрезков и вычисление m_i можно реализовать за время $O(\log N)$ для одной вертикальной стороны прямоугольника; тем самым достигается ранее поставленная цель. Теперь можно сформулировать алгоритм, где b_i и e_i обозначают соответственно минимальную и максимальную ординаты вертикальной стороны, имеющей абсциссу $X[i]$. (Заметим, что этот алгоритм является непосредственным обобщением одномерного алгоритма, приведенного ранее.)

```
        procedure МЕРА-ОБЪЕДИНЕНИЯ-
                  ПРЯМОУГОЛЬНИКОВ
1.  begin X[1 : 2N] := упорядоченная последовательность
          абсцисс вертикальных сторон;
2.        X[0] := X[1];
3.        m := 0;
4.        построить и инициализовать дерево отрезков T
          для ординат сторон прямоугольника;
5.        for i := 1 until 2N do
6.            begin m* := m[корень(T)];
7.                  m := m + m* (X[i] − X[i − 1]);
8.                  if (X[i] — абсцисса левой стороны) then
                        ВСТАВИТЬ(b_i, e_i; корень(T))
9.                  else УДАЛИТЬ(b_i, e_i; корень(T))
              end
    end.
```


Завершим это обсуждение следующей теоремой:

Теорема 8.1. *Меру объединения N изотетичных прямоугольников на плоскости можно вычислить за оптимальное время* $\theta(N \log N)$.

Возвращаясь к последнему алгоритму, мы сразу же увидим, что он относится к категории плоского заметания (см. разд. 8.3), где *списком точек событий* служит массив $X[1:2N]$, а *статус заметающей прямой* задается деревом отрезков. Подобная реализация допускает непосредственное обобщение этого метода на случаи более чем двух измерений. Действительно, техника заметания позволяет преобразовать исходную d-мерную задачу (на N гиперпрямоугольниках в E^d) в «последовательность», состоящую из N штук $(d-1)$-мерных подзадач (каждая на не более чем N гиперпрямоугольниках в E^{d-1}). Для $d = 3$ эти подзадачи становятся двумерными, и по теореме 8.1 каждую из них можно решить за время $O(N \log N)$; поэтому задачу о мере объединения при трех измерениях можно решить за время $O(N^2 \log N)$. Это устанавливает базис для индуктивного доказательства, приводящего к следствию 8.2.

Следствие 8.2. *Меру объединения N изотетичных гиперпрямоугольников в пространстве $d \geqslant 2$ измерений можно вычислить за время* $O(N^{d-1} \log N)$.

Сильным недостатком описанного метода при $d \geqslant 3$ является тот факт, что при заметании мы не смогли использовать «когерентность» [1]) между двумя последовательными сечениями. (При $d = 2$ дерево отрезков позволяет воспользоваться когерентностью смежных сечений путем эффективного вычисления текущего сечения за счет простой корректировки предыдущего сечения.) Эта идея была впоследствии перенесена на случай

[1]) То есть согласованность, взаимозависимость. — *Прим. перев.*

трех измерений Ван Леювеном и Вудом [Van Leeuwen, Wood (1981)], которые предложили использовать в качестве структуры данных «квадродерево» [Finkel, Bentley (1974)]. Квадродерево можно считать двумерным обобщением дерева отрезков, но весьма курьезно, что оно было предложено раньше, чем эта более простая структура. Сейчас мы кратко опишем его организацию.

Квадродерево — это способ организации ортогональной сетки, состоящей из $N \times M$ ячеек, определенных $(N+1)$ горизонтальными и $(M+1)$ вертикальными прямыми. (Дерево отрезков — это соответственно способ организации N смежных интервалов.) Для простоты изложения предположим, что $N = M = 2^k$. Квадродерево T — это дерево со степенями исхода, кратными четырем, у которого с каждым узлом связан участок сетки размером $2^i \times 2^i$ (именуемый 2^i-квадратом) $(i = 0, 1, \ldots, k)$. 2^i-квадрат, связанный с заданным узлом v из T (при $i > 0$), разбивается на четыре 2^{i-1}-квадрата вертикальным и горизонтальным делениями пополам (рис. 8.9); каждый из этих четырех квадратов (именуемых СЗ, СВ, ЮВ и ЮЗ квадратами) связывается с одним из четырех потомков узла v. Данное построение начинается со связывания всей сетки с корнем T, а заканчивается тогда, когда процесс четвертования дает 2^0-квадраты (листья T). Поскольку T обладает N^2 листьями, то его построение занимает $O(N^2)$ времени и использует $O(N^2)$ памяти.

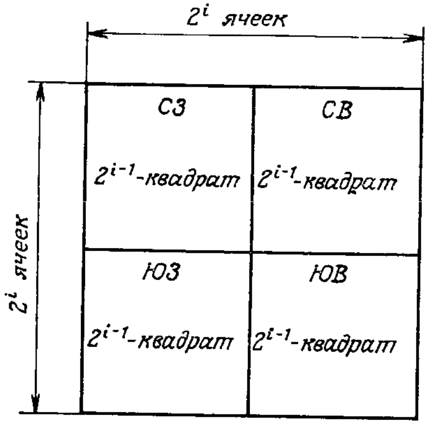

Рис. 8.9. Пример разбиения плоской сетки, осуществляемого одним из узлов квадродерева, который соответствует 2^i-квадрату.

Теперь мы рассмотрим вопрос о хранении прямоугольника R в квадродереве. Исходная сетка образована $2N$ абсциссами и $2N$ ординатами сторон заданных N прямоугольников, и квадродерево T строится на этой сетке. Каждый узел v из T обладает одним целочисленным параметром $C[v]$, который вначале равен 0 (такое квадродерево называется *скелетным*). Прямоугольник R вносит 1 в $C[v]$ тогда и только тогда, когда (1) R содержит квадрат, относящийся к v, и (2) ни один из предков v в T не обладает таким же свойством. Очевидно, что таким образом определяется однозначный способ разбиения R на квадраты, каждый из которых связан с одним узлом v. В то время как в дереве отрезков каждый отрезок разбит на не более чем

$O(\log N)$ интервалов, для квадродерева можно относительно просто показать[1]), что один прямоугольник разбивается на не более чем $O(N)$ квадратов. Поэтому процедуры вставки и удаления в квадродереве, а также определение меры объединения (площади) можно реализовать каждую за $O(N)$ операций. Отсюда следует, что последовательная корректировка сечений

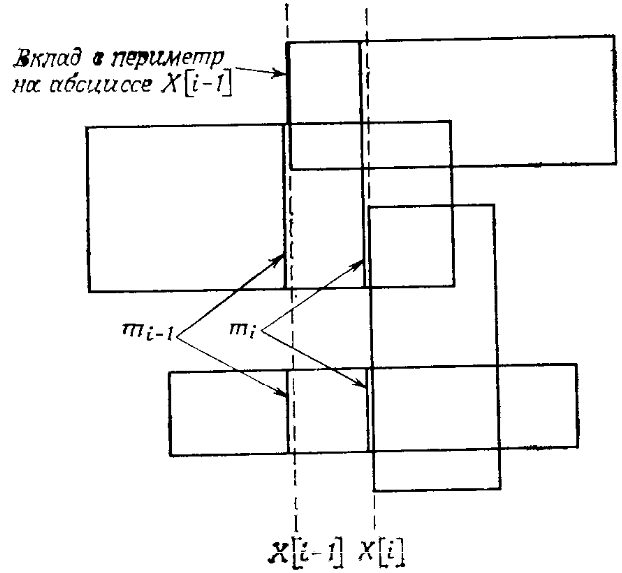

Рис. 8.10. Иллюстрация к вычислению суммарной длины вертикальных сторон.

при пространственном заметании занимает $O(N)$ времени, а поскольку существует $2N$ сечений, то решение задачи определения меры объединения в трехмерном пространстве займет в сумме $O(N^2)$ времени. Именно данный метод — а не двумерный, основанный на дереве отрезков, — можно использовать в качестве базиса индукции и доказать следующий результат:

Теорема 8.2. *Меру объединения N изотетичных гиперпрямоугольников при $d \geqslant 3$ измерениях можно вычислить за время* $O(N^{d-1})$.

Хотя кажется, что улучшить этот результат будет весьма трудно, нет никаких данных о его возможной оптимальности.

Замечание. Приведенный алгоритм вычисления меры объединения прямоугольников можно модифицировать для решения следующей задачи.

[1]) Доказательство можно найти в статье [Van Leeuwen, Wood (1981)].

Задача О.3 (ПЕРИМЕТР ОБЪЕДИНЕНИЯ ПРЯМОУГОЛЬНИКОВ). Дано объединение F, состоящее из N изотетичных прямоугольников. Найти *периметр* F, т. е. длину его границы.

Метод решения основан на следующем наблюдении. Предположим, что на i-й итерации некий прямоугольник вставляется в дерево отрезков или удаляется из него (мы условились говорить, что прямоугольник вставляется или удаляется, если заметающая прямая касается его левой или правой стороны соответственно). Обращаясь к рис. 8.10, напомним, что m_i обозначает суммарную длину вертикального сечения непосредственно слева от $X[i]$ (см. строку 6 алгоритма МЕРА-ОБЪЕДИНЕНИЯ-ПРЯМОУГОЛЬНИКОВ). Суммарная длина вертикальных кусков границы F на сечении $X[i-1]$ равна разности длин вертикальных сечений F непосредственно слева и справа от $X[i-1]$. В то время как первая из этих длин равна m_{i-1} по определению, вторая равна m_i, поскольку сечение неизменно на всем интервале $[X[i-1], X[i]]$. Поэтому искомая суммарная длина равна $|m_i - m_{i-1}|$[1]. Кроме того, необходимо учесть вклад горизонтальных граничных сторон, расположенных внутри вертикальной полосы. В интервале $[X[i-1], X[i]]$ этот вклад, очевидно, равен $(X[i] - X[i-1]) \times \alpha_i$, где целое число α_i равно количеству горизонтальных сторон границы F внутри рассматриваемой вертикальной полосы. Параметр α_i, по сути, весьма похож на m_i. Действительно, введем для каждого узла v в дереве отрезков три новых специальных параметра, один четный целый параметр $\alpha[v]$ и два двоичных параметра ЛКР$[v]$ и ПКР$[v]$[2]. Обозначая через $\mathscr{I}$ текущее вертикальное сечение F ($\mathscr{I}$ — набор не связанных между собой интервалов), определяем эти параметры следующим образом:

$\alpha[v] :=$ удвоенное число компонент связности $\mathscr{I} \cap [B[v], E[v]]$;

ЛКР$[v] := 1$ или 0 в зависимости от того, будет или нет $B[v]$ нижним концом интервала из $\mathscr{I} \cap [B[v], E[v]]$;

ПКР$[v] := 1$ или 0 в зависимости от того, будет или нет $E[v]$ верхним концом интервала из $\mathscr{I} \cap [B[v], E[v]]$.

Эти три специальных параметра вначале равны нулю, а их вычисление выполняет нижеследующая конкретизация подпро-

[1] Заметим, что возможно равенство $X[i-1]$ и $X[i]$, т. е. две вертикальные стороны имеют одинаковые абсциссы. В этом случае введем для них удобное лексикографическое упорядочение (например, по ординатам их нижних концов) и представим себе, что две такие стороны фиктивно раздвинуты на величину ε в направлении оси x.

[2] ЛКР и ПКР — аббревиатуры выражений «левый край» и «правый край». — *Прим. перев.*

граммы КОРРЕКТИРОВАТЬ, вызываемой в строке 4 процедуры ВСТАВИТЬ (см. разд. 8.2):

```
        procedure КОРРЕКТИРОВАТЬ(v);
1.  begin if (C[v] > 0) then
2.          begin α[v] := 2;
3.                ЛКР[v] := 1;
4.                ПКР[v] := 1
            end
5.      else begin α[v] := α[ЛСЫН[v]] + α[ПСЫН[v]] —
                  —2 · ПКР[ЛСЫН[v]] ЛКР[ПСЫН[v]];
6.                ЛКР[v] := ЛКР[ЛСЫН[v]];
7.                ПКР[v] := ПКР[ПСЫН[v]]
             end
    end.
```

Корректность процедуры КОРРЕКТИРОВАТЬ доказывается легко. Если $C[v] \geqslant 1$, т. е. интервал $[B[v], E[v]]$ полностью накрыт, то в составе $\mathscr{I} \cap [B[v], E[v]]$ будет ровно один элемент, причем $\alpha[v] = 2$, ЛКР$[v] =$ ПКР$[v] = 1$, как указано в

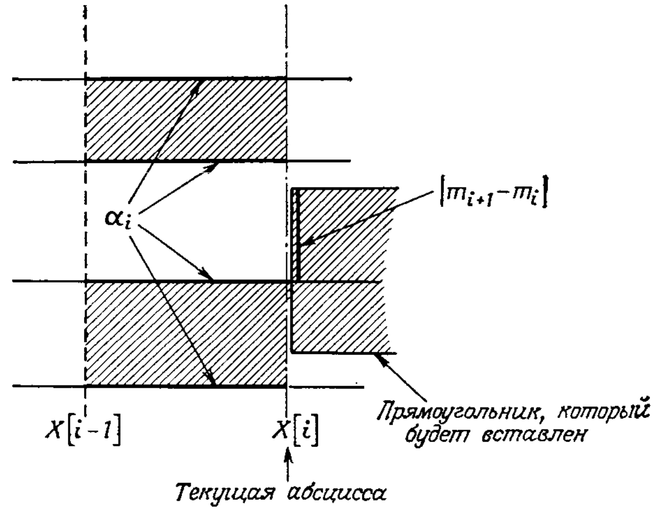

Рис. 8.11. Вклад в величину периметра на текущем шаге состоит из длин горизонтальных и вертикальных ребер.

строках 2—4. Если же $C[v] = 0$, то число элементов в $\mathscr{I} \cap [B[v], E[v]]$ равно сумме чисел таких элементов для двух потомков узла v, за исключением того случая, когда сечение $\mathscr{I}$ содержит один интервал, соединяющий точку E[ЛСЫН$[v]] = B$[ПСЫН$[v]]$; в последнем случае ПКР[ЛСЫН$[v]] =$ ЛКР[ПСЫН$[v]] = 1$. Этим доказывается корректность строк 5—7 алгоритма. Кроме того, не составит большого труда показать, что дополнительные параметры α, ЛКР и ПКР вычисляются за константное время в расчете на один пройденный узел.

Теперь очевидно, что $\alpha_i = \alpha[\text{корень } (T)]$. Следовательно, алгоритм вычисления периметра F получается путем простой модификации соответствующего алгоритма вычисления меры этого объединения. Заметим, однако, что в то время, как в алгоритме вычисления меры объединения к площади F добавляется площадь только что заметенной вертикальной полосы (так что корректировка дерева отрезков следует за суммированием этой площади), в обсуждаемом алгоритме ситуация несколько иная. Обратимся к рис. 8.11; вклад текущего шага в периметр состоит из двух частей: от горизонтальных ребер в полосе $[X[i-1], X[i]]$ он равен $\alpha_i \times (X[i] - X[i-1])$ и от вертикальных ребер на абсциссе $X[i]$ он равен $|m_{i+1} - m_i|$. Поэтому величину α_i необходимо извлечь из дерева отрезков до его корректировки, а величину m_{i+1} — после. Итак, получаем:

procedure ПЕРИМЕТР-ОБЪЕДИНЕНИЯ-ПРЯМОУГОЛЬНИКОВ
begin $X[1:2N]$:= упорядоченная последовательность абсцисс вертикальных сторон;
    $X[0] := X[1]$;
    $m_0 := 0$;
    $p := 0$;
    построить и инициализировать дерево T, состоящее из ординат сторон прямоугольников;
    for $i := 1$ **until** $2N$ **do**
        begin $\alpha^* := \alpha$ [корень(T)];
            if ($X[i]$ — абсцисса левого конца)
            then ВСТАВИТЬ$(b_i, e_i,$ корень$(T))$
            else УДАЛИТЬ$(b_i, e_i;$ корень$(T))$;
            $m^* := m$[корень(T)];
            $p := p + \alpha^* \times (X[i] - X[i-1]) + |m^* - m_0|$;
            $m_0 := m^*$
        end
end.

В заключение получаем:

Теорема 8.3. *Периметр объединения N изотетичных прямоугольников можно вычислить за время $O(N \log N)$.*

8.5. Контур объединения прямоугольников

Тот же самый общий подход (плоское заметание, основанное на дереве отрезков) можно с успехом применить для решения другой интересной задачи: определения контура

объединения F, состоящего из N изотетичных прямоугольников $R_1, \ldots, R_N$. Вновь $F = R_1 \cup \ldots \cup R_N$ может состоять из одной или более не пересекающихся компонент связности, а каждая компонента может содержать или не содержать внутренние дыры (заметим, что дыра может содержать внутри себя некоторые из компонент связности F). Контур (граница) F состоит из набора непересекающихся циклов, образованных (чередующимися) вертикальными и горизонтальными ребрами. Условимся, что любое ребро ориентировано так, что фигура расположена слева от него; другими словами, цикл ориентирован по часовой стрелке, если он ограничивает дыру, и против часовой стрелки, если он является внешней границей компоненты связности. Мы имеем:

Задача О.4 (КОНТУР ОБЪЕДИНЕНИЯ ПРЯМОУГОЛЬНИКОВ). Дан набор из N изотетичных прямоугольников. Требуется определить контур их объединения.

Во-первых, заметим, что периметр (для вычисления которого в предыдущем разделе был описан довольно простой алгоритм) является не чем иным, как длиной искомого контура. Однако мы увидим, что вычислить периметр значительно проще, чем сам контур, как набор непересекающихся циклов.

Алгоритм лучше всего описать неформально, на примере. Он состоит из двух главных этапов. На первом этапе определяется множество V, состоящее из вертикальных ребер контура (ребра от e_1 до e_{10} на рис. 8.12); на втором этапе эти вертикальные ребра соединяются горизонтальными ребрами для формирования ориентированных циклов контура [Lipski, Preparata (1980)].

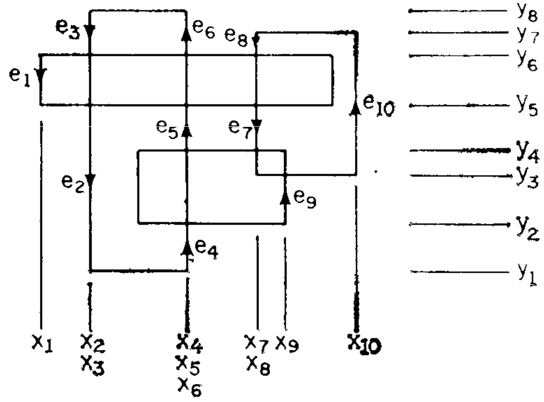

Рис. 8.12. Иллюстрация к задаче построения контура объединения прямоугольников.

Обозначим через $(x; b, t)$ вертикальное ребро с абсциссой x, для которого b и t $(b < t)$ являются соответственно нижней и верхней ординатами; аналогично $(y; l, r)$ обозначает горизонтальное ребро с ординатой y, для которого l и r $(l < r)$ являются соответственно левой и правой абсциссами.

Для получения множества V просканируем слева направо абсциссы, соответствующие вертикальным сторонам прямо-