

杭州学军中学校 周任飞

符和布尔表达式, 提出球面复合区域面积问题的精确定义, 并介绍解决该问题的优秀算法。第五章讨论了这个算法的一些应用场景, 并展示了如何对算法的各个阶段作出适当调整, 以适应具体问题。第六章作简要的总结。

2 平面面积并问题

问题 1 (平面面积并问题) 给定平面上的 n 个圆形区域 D_1, \dots, D_n , 定义

$$D^* = D_1 \cup D_2 \cup \dots \cup D_n,$$

其中每个圆形区域 D_i 指的是一个圆的内部。计算 D^* 的面积。

本章提出一个基于格林公式的算法, 解决上述问题。

格林公式 设 $\Omega \subset \mathbb{R}^2$ 是由有限条分段光滑的曲线围成的闭区域。如果函数 $P(x, y)$ 和 $Q(x, y)$ 在 Ω 上连续, 并有连续的偏导数 $\partial Q/\partial x$ 和 $\partial P/\partial y$, 那么有

$$\oint_{\partial\Omega} Pdx + Qdy = \iint_{\Omega} \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy,$$

其中 $\partial\Omega$ 是区域 Ω 的边界。它的定向是这样被确定的: 一个人沿着 $\partial\Omega$ 的正方向行进时, 区域 Ω 总在这个人的左边。

该定理的证明可以在数学分析教材中找到, 这里不再叙述其证明, 而是叙述如何利用该定理计算 D^* 的面积。

记 S_{D^*} 表示 D^* 的面积, 则

$$S_{D^*} = \iint_{D^*} dx dy.$$

如果令

$$\begin{cases} \Omega = D^*, \\ P(x, y) = -y, \\ Q(x, y) = 0, \end{cases}$$

则 Ω, P, Q 满足格林公式的应用条件。代入格林公式, 得到

$$\begin{aligned} \oint_{\partial D^*} (-y) dx &= \iint_{D^*} dx dy \\ &= S_{D^*}, \end{aligned}$$

从而将计算面积转化为计算边界 ∂D^* 的曲线积分。

图 1 是问题 1 的一个示例, 其中 D^* 被涂成淡紫色, 圆用虚线标出, ∂D^* 用实线标出。如图所示, ∂D^* 是若干段圆弧组成的, 设这些圆弧分别为 A_1, \dots, A_m , 则根据积分可加性, 有

$$\oint_{\partial D^*} (-y) dx = \sum_{i=1}^m \int_{A_i} (-y) dx,$$

其中每个 A_i 的方向都是逆时针。我们利用上式的右侧来计算上式的左侧。于是，问题 1 被转化为两个子问题：

问题 1.1 求出 A_1, A_2, \dots, A_m 。

问题 1.2 计算 $\int_{A_i} (-y) dx$ 。

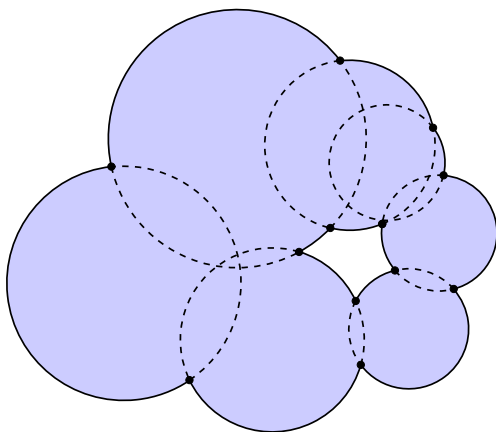


图 1

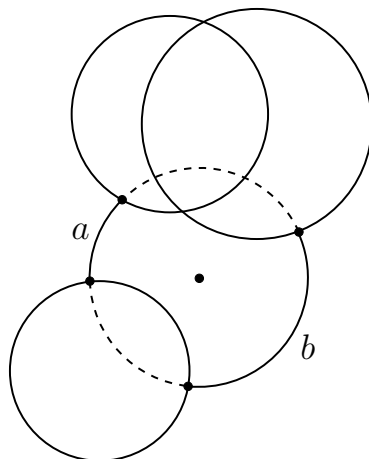


图 2

先讨论问题 1.1。不失一般性地，假设圆 D_1, D_2, \dots, D_n 两两不同；否则，对于出现多次的圆，我们可以保留一个并删去其它，而不改变问题的答案。如图 2 所示，每个圆周上未被其它圆覆盖的部分（以中央的圆为例：圆弧 a, b ）构成 ∂D^* ；被其它圆覆盖的部分（虚线部分）则不是 ∂D^* 的组成部分。求出构成 ∂D^* 的圆弧，就要求我们分别对于每一个圆，计算圆周上未被其它圆覆盖的部分。

定义“极角”运算。假设我们现在正在处理的圆的圆心是 O ；定义圆周上的点 P 的极角 $\text{Angle}(P)$ 为：从 x 轴正方向到向量 \overrightarrow{OP} 的方向所转过的角度。这是一个有向角度，逆时针转动取正号，取值范围规定为 $[-\pi, \pi)$ 。极角运算是圆周到 $[-\pi, \pi)$ 的一一映射。我们转而计算 $[-\pi, \pi)$ 上未被覆盖的部分，每个圆在圆 O 上覆盖的部分（如果有）是一段圆弧，其对应的极角是 $[-\pi, \pi)$ 上至多两个区间的并。问题进一步转化为：

问题 1.1a 给定 $[-\pi, \pi)$ 上的 n' 个区间, 求出没有被任何区间覆盖的部分, 用若干个不交的区间表示。其中 $n' \leq 2n$ 。

这是一个经典问题，这里简单介绍一下运行时间上界为 $O(n' \log n')$ 的算法。

利用扫描线的思想, 令动点 P 从 $-\pi$ 逐渐移动到 π , 在这个过程中维护有几个区间包含 P 。这个值从 1 变成 0, 以及从 0 变成 1 的位置, 就是未覆盖区间的左右端点。在该过程之前, 需要对所有区间端点进行排序, 运行时间上界为 $O(n' \log n')$ 。

求出 $[-\pi, \pi)$ 上未被覆盖的若干个区间之后, 可以在线性的时间内求出它们对应的圆弧。对于 n 个圆分别进行上述计算, 就在 $O(n^2 \log n)$ 的最坏时间复杂度内求出了 A_1, \dots, A_m ,

解决了问题 1.1。

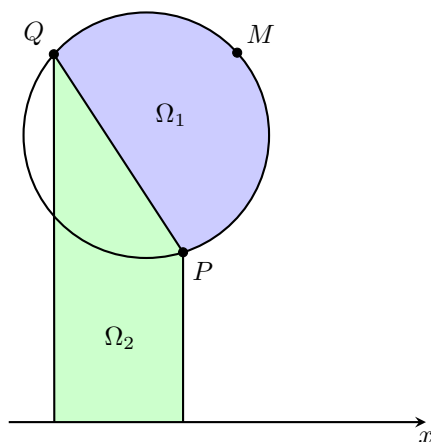


图 3

最后，讨论问题 1.2。如图 3 所示，假设我们要对于一段圆弧 $A_i = \widehat{PMQ}$ 计算 $\int_{A_i} (-y) dx$ (曲线方向为逆时针， $P \rightarrow Q$)。区域 Ω_1 是由 A_i 和 \overrightarrow{QP} 围成的弓形区域，图中用淡蓝色标出； Ω_2 是由 \overrightarrow{QP} 、 x 轴和两条竖线围成的区域。对区域 Ω_1 应用格林公式，得

$$\begin{aligned} S_{\Omega_1} &= \oint_{\partial\Omega_1} (-y) dx \\ &= \int_{A_i} (-y) dx + \int_{\overrightarrow{QP}} (-y) dx \\ \Rightarrow \int_{A_i} (-y) dx &= S_{\Omega_1} + \int_{\overrightarrow{QP}} y dx \\ &= S_{\Omega_1} + S_{\Omega_2}, \end{aligned}$$

其中 S_{Ω_2} 是区域 Ω_2 的有向面积：如果 Q 的横坐标大于 P ，则 $S_{\Omega_2} < 0$ 。最后一步的推导用到了 \overrightarrow{QP} 的积分的几何意义。

Ω_1 是弓形， Ω_2 是梯形，它们的面积易于计算。使用上式可以在常数时间内计算出一段圆弧 A_i 的积分，从而解决了问题 1.2。

最后，我们再重新审视问题 1。我们在 $O(n^2 \log n)$ 的最坏时间复杂度内，求出组成 ∂D^* 的所有圆弧 A_1, \dots, A_m ，其中 m 有一个上界 n^2 。接着我们对每一个 A_i 计算其曲线积分，每一个花费常数时间，并将它们相加得到 S_{D^*} 。于是，我们得到一个运行时间上界为 $O(n^2 \log n)$ 的优秀算法，解决了问题 1。

3 球面面积并问题

在上一章中，我们利用格林公式将 D^* 上的二重积分转化为 ∂D^* 上的曲线积分，再对

组成 ∂D^* 的各段圆弧分别求曲线积分。本章我们用同样的思路解决球面面积并问题。

问题 2（球面面积并问题） 给定球面上的 n 个球冠形区域 D_1, \dots, D_n ，球冠不能大于半球。定义

$$D^* = D_1 \cup D_2 \cup \dots \cup D_n,$$

计算 D^* 的面积。

此处我们规定球冠不能大于半球，并不是说大的球冠无法被处理，而是因为在提出布尔运算后，我们可以使用较小球冠的补集来表示超过半球的球冠，从而无需讨论这种情况。作此规定可以为几何上的处理提供方便。

在介绍上述问题的解法之前，我们先简单介绍一下球面几何的基本知识，并约定一些术语的含义。

- (1) 我们假定问题中的球面是单位球面，即球心在原点 $O = (0, 0, 0)$ ，半径为 1。能够这么做是因为球的平移、缩放对问题答案的影响容易处理。
- (2) 用一个平面去截球面，截得的部分是一个圆（如图 4a 所示）。球面在该平面某一侧的部分称为球冠（如图 4b 所示）。这二者之间有对应关系，圆是球冠的边界，球冠是圆的内部。在不需要加以区分的情形下，它们的名称可以混用。

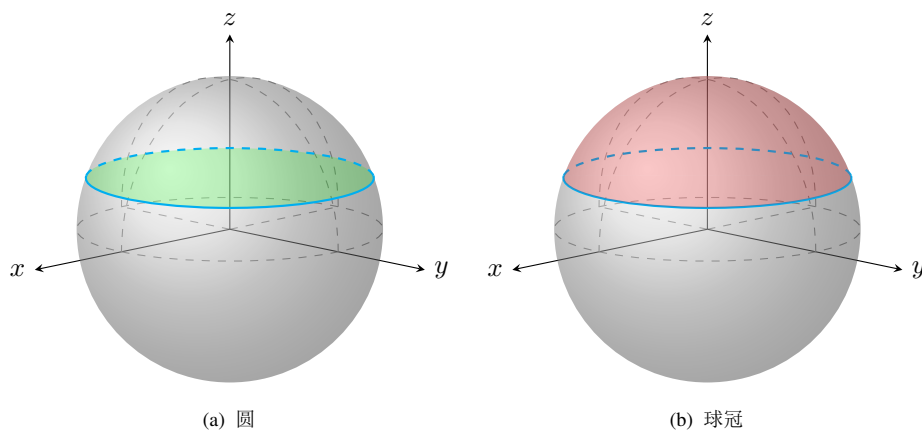


图 4

对于一个球冠，我们称其**顶点**为它在球面上的中心点。它在球冠上，且到球冠边界各点的距离相等。称其**底面**是对应的圆所在平面。

- (3) 我们称球面上的两个点是**对径的**，当且仅当它们的连线穿过球心。这等价于它们关于球心对称。
- (4) **大圆**是最大的圆，即过球心的平面去截球面时截得的圆。除大圆之外，其它的圆都称为**小圆**。大圆将整个球面分成两个大小相等的部分，即两个半球。大圆在球面几何中

的地位好比平面几何中的直线。对于球面上的两个不同的点 A, B ，如果它们不互相对径，则有唯一的大圆同时经过它们（如图 5a 所示），称为**球面直线 AB** 或大圆 AB 。此时 A, B 把这个大圆分成了长度不相等的两段，其中较短的一段称为**球面线段 AB** （如图 5b 所示），它是球面上从 A 到 B 的最短路线。

每个大圆 AB 唯一确定一个平面 OAB 。

如果 A, B 对径，则球面上从 A 到 B 的最短路线有无限条。任取一个过 A, B 的大圆， A, B 将其分为长度相等的两段，每一段都是从 A 到 B 的最短路线。我们将任意一条最短路线称为它们之间的**广义球面线段**。但为了指定一条广义球面线段，我们需要给出除 A, B 之外，它上面的另一个点 M 。

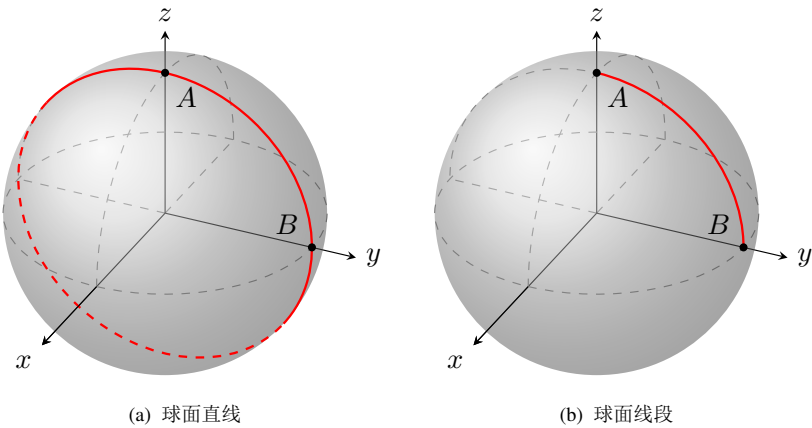


图 5

(5) 如果 AB, AC 都是球面线段，那么可以定义**球面角 $\angle BAC$** 。如图 6 所示，过 A 点作 \widehat{AB} 的切线 AF ，作 \widehat{AC} 的切线 AE ，定义 $\angle BAC = \angle FAE$ 。这等价于平面 OAB 与 OAC 之间的二面角。

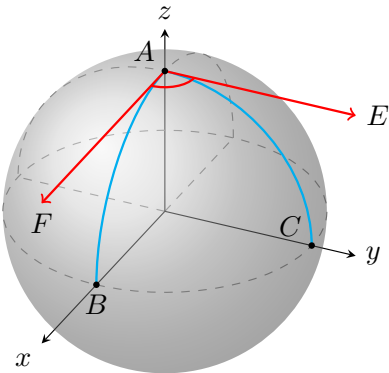


图 6

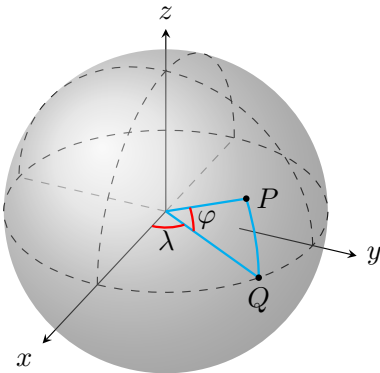


图 7

- (6) 如果球面上三个点 A, B, C 两两不同、两两不对径、不在同一个大圆上，则可以定义球面三角形 $\triangle ABC$ 。其三条边分别是球面线段 AB, BC, CA 。这三条边把整个球面分成两个部分，其中较小那个部分是 $\triangle ABC$ 的内部。球面三角形的三个内角均小于平角。
- (7) 与地球上的情形一样，定义南极点 $P_S = (0, 0, -1)$ 、北极点 $P_N = (0, 0, 1)$ ，并定义经线、纬线。具体地，连接 P_N 与 P_S 的所有球面线段被称为经线；平面 $z = z_0$ 割球面所得的圆被称为纬线。

我们可以用经纬度表示球面上的点，如图 7 所示。对于球面上的每个点 $P = (x, y, z)$ ，设实数 $\lambda \in [0, 2\pi)$ 和 $\varphi \in [-\pi/2, \pi/2]$ 满足

$$\begin{cases} x = \cos \varphi \cdot \cos \lambda, \\ y = \cos \varphi \cdot \sin \lambda, \\ z = \sin \varphi, \end{cases}$$

就称 (λ, φ) 是点 P 的经纬表示。其中 λ 称为经度， φ 称为纬度。

两个极点有多种经纬表示方法。我们将两个极点从球面中删除，形成去极球面，这并不会改变我们希望计算的面积。去极球面的每一个点 P 有唯一的经纬表示

$$(\lambda, \varphi) \in [0, 2\pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

上式右侧是平面上的一个矩形区域，称其为经纬平面。在经纬平面上，规定 λ 是横坐标、 φ 是纵坐标。

点的经纬表示法建立了从去极球面到经纬平面的一一映射。我们可以利用这个映射，将球面上区域的面积转化为经纬平面上的二重积分，再通过格林公式与曲线积分联系起来。

定理 1 假设 Ω 是球面上由有限条分段光滑的曲线围成的区域，且北极点 P_N 不在 Ω 的边界上，那么有

$$S_\Omega = \oint_{\partial\Omega} (-\sin \varphi - 1) d\lambda + [P_N \in \Omega] 4\pi.$$

证明

$$S_\Omega = \iint_{\Omega} da, \quad (1)$$

其中上式右侧是对 Ω 的曲面积分， da 是面积微元。如图 8 所示，有

$$da = \cos \varphi \cdot d\lambda \cdot d\varphi.$$

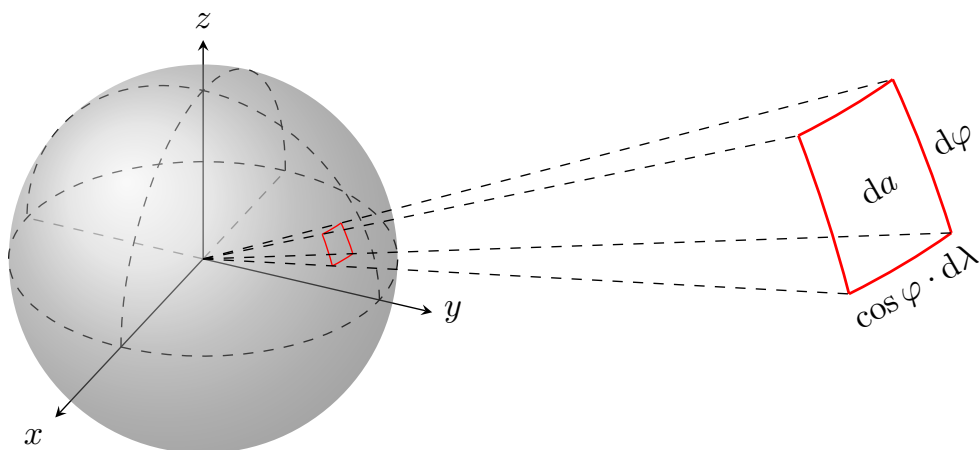


图 8

假设 $\hat{\Omega}$ 是 Ω 在经纬平面上的对应区域。(1) 式可以改写为

$$S_{\Omega} = \iint_{\hat{\Omega}} \cos \varphi \, d\lambda \, d\varphi,$$

右侧是经纬平面上的二重积分。请注意上式右侧的积分是不含两极的，在这一步中两极被去掉了，这不会改变区域的面积。

根据格林公式，经纬平面上有

$$\oint_{\partial \hat{\Omega}} P d\lambda + Q d\varphi = \iint_{\hat{\Omega}} \left(\frac{\partial Q}{\partial \lambda} - \frac{\partial P}{\partial \varphi} \right) d\lambda \, d\varphi \quad (2)$$

成立。令

$$\begin{cases} P(\lambda, \varphi) = -\sin \varphi - 1, \\ Q(\lambda, \varphi) = 0, \end{cases}$$

并代入 (2)，得到

$$\begin{aligned} \oint_{\partial \hat{\Omega}} (-\sin \varphi - 1) d\lambda &= \iint_{\hat{\Omega}} \cos \varphi \, d\lambda \, d\varphi \\ &= S_{\Omega}. \end{aligned}$$

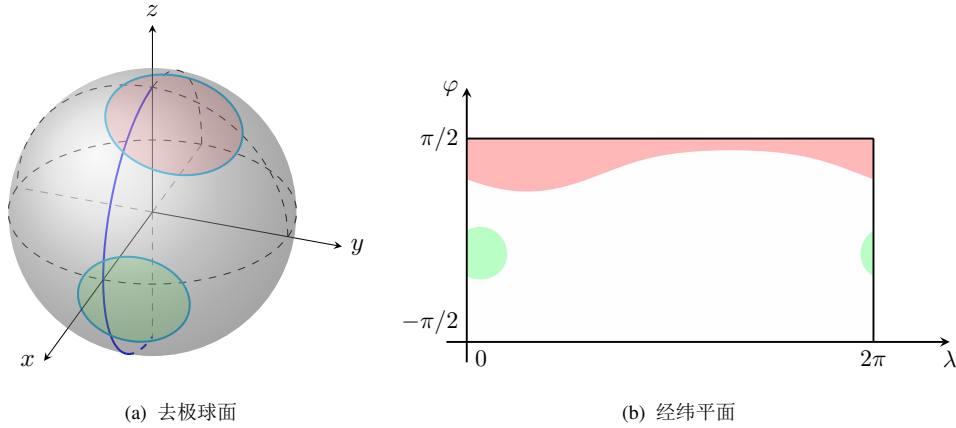


图 9

$\hat{\Omega}$ 的边界可以分为两部分：第一是经纬平面边界处的，第二是经纬平面内部的。如图 9 所示，其中第二部分与 $\partial\Omega$ 相同。第一部分中，对于左右边界有 $d\lambda = 0$ ，对于下边界有 $-\sin\varphi - 1 = 0$ ，所以沿着这三条边界的曲线积分都为 0，从而我们无需关心具体哪些部分是 $\hat{\Omega}$ 的边界。如果北极点在 Ω 内，则经纬平面的整个上边界都是 $\hat{\Omega}$ 的边界，方向为从右向左，其积分值为

$$\begin{aligned} & \int_{2\pi}^0 \left(-\sin \frac{\pi}{2} - 1 \right) d\lambda \\ &= \int_{2\pi}^0 (-2) d\lambda \\ &= 4\pi. \end{aligned}$$

反之，如果北极点不在 Ω 内，则整个上边界都不是 $\hat{\Omega}$ 的边界。综上所述，对于 $\partial\hat{\Omega}$ 的第一部分，有

$$\int_{\text{Part-1}} (-\sin\varphi - 1) d\lambda = [P_N \in \Omega] 4\pi.$$

结合 Part-2 = $\partial\Omega$ ，得到

$$\begin{aligned} \oint_{\partial\hat{\Omega}} (-\sin\varphi - 1) d\lambda &= \int_{\text{Part-1}} (-\sin\varphi - 1) d\lambda + \int_{\text{Part-2}} (-\sin\varphi - 1) d\lambda \\ &= [P_N \in \Omega] 4\pi + \oint_{\partial\Omega} (-\sin\varphi - 1) d\lambda. \end{aligned}$$

该等式左侧等于 S_Ω ，右侧与定理 1 是一样的。 \square

请注意在上述证明过程中去极球面与经纬平面的关系。我们在球面上进行的积分仅含变量 λ, φ ，这和经纬平面是一样的。对于点、曲线、区域，它们在球面上与在经纬平面上没有区别，两种形式下它们的积分值也相同。唯一的区别是在经纬平面上应用格林公式的过程中，用区域定义其边界时 $\partial\Omega$ 与 $\partial\hat{\Omega}$ 的区别——后者比前者多出一部分，即上文所述 $\partial\hat{\Omega}$ 的第一部分。

另一点需要补充的是，当曲线穿过子午线的时候， λ 有一个间断点。这时我们可以该间断点的位置上把曲线切分为两段，这两段的 λ 都是连续的，从而其积分有良好定义；再利用积分的可加性，定义整条曲线的积分为这两段积分之和。为了叙述方便，文中仍然使用曲线积分的一般符号来表示它。

推论 S_Ω 与 $\oint_{\partial\Omega} (-\sin\varphi - 1) d\lambda$ 之间可以互相推出，即只要知道其中一个，就可以在常数的时间内计算出另一个的值。

回到问题 2。我们假定 D_1, \dots, D_n 两两不同。

在解决这个问题之前，以随机的角度旋转整个球面。具体来说，在球面上均匀随机选择一个点 P'_N ，并通过旋转使该点到达 $(0, 0, 1)$ 。旋转完成后，再按前文的方法定义南北极、经纬度。这使得 D_1, \dots, D_n 中的某个圆穿过北极的概率是 0，我们假定它不会发生。另外，对于一条圆弧，如果其位置由 D_1, \dots, D_n 确定，而与旋转点的选取无关（例如某两个球冠顶点的连线），那么我们也假定它不穿过北极点。

根据推论，问题 2 转化为计算

$$\oint_{\partial D^*} (-\sin\varphi - 1) d\lambda.$$

与上一章同理， ∂D^* 是由一些圆弧——即每个圆未被其它圆覆盖的部分——构成的。于是该问题转化为两个子问题：

问题 2.1 求出每个圆未被其它圆覆盖的若干圆弧。

问题 2.2 对于某一段有向圆弧 C ，计算其曲线积分 $\int_C (-\sin\varphi - 1) d\lambda$ 。

先讨论问题 2.1。假设我们要对于圆 O' 求出圆周上未被覆盖的部分。该圆在平面 α 上，且点 O' 是平面 α 上该圆的圆心。在平面 α 上建立直角坐标系 $xO'y$ ，并以 O' 为中心来定义极角运算，在圆周上的点与 $[-\pi, \pi)$ 之间建立一一映射。使用上一章中解决问题 1.1 的方法，即可求出圆周上未被覆盖的部分，它们作为 $\partial\Omega$ 的定向也容易确定，这里不再赘述。

再讨论问题 2.2。如图 10a 所示，假设我们要计算圆弧 $C_1 = \widehat{PQ}$ 的曲线积分

$$\int_{C_1} (-\sin\varphi - 1) d\lambda,$$

其中曲线的方向是 $P \rightarrow Q$ 。

第一种情况： C_1 是小圆的一部分。作有向球面线段 $C_2 = \overrightarrow{QP}$ ，则 C_1 与 C_2 围成了一个弓形区域 A 。根据推论，可以由 S_A 推出

$$\int_{C_1} (-\sin\varphi - 1) d\lambda + \int_{C_2} (-\sin\varphi - 1) d\lambda,$$

再结合 C_2 的积分来推出 C_1 的积分。问题转化为两部分：

问题 2.2a 计算球面上弓形的面积。

问题 2.2b 给定有向球面线段 $C_2 = \overrightarrow{QP}$ ，计算

$$\int_{C_2} (-\sin\varphi - 1) d\lambda.$$

第二种情况： C_1 是大圆的一部分。如果 C_1 大于或等于半个大圆，则取 C_1 的中点 M ，将其分为两个球面线段 \overrightarrow{QM} 和 \overrightarrow{MP} ；否则， C_1 本身就是球面线段。从而转化为问题 2.2b。

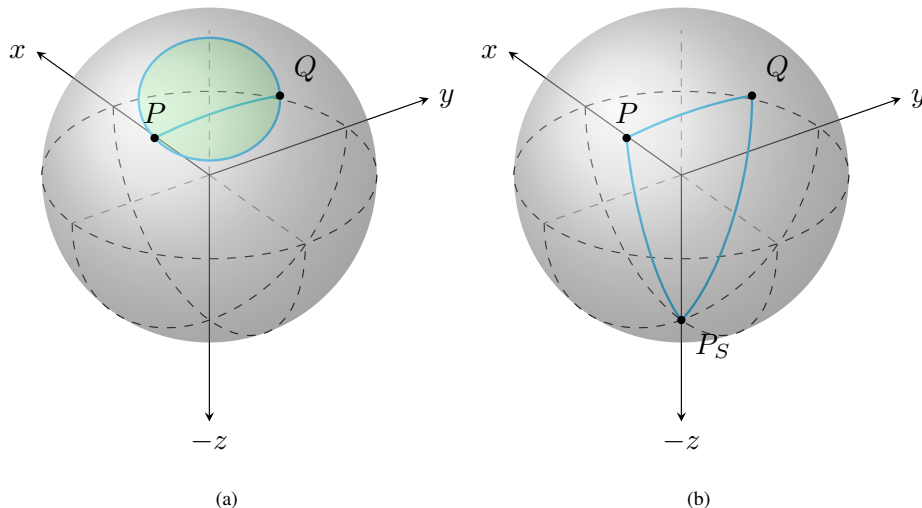


图 10

先讨论问题 2.2a。通过连接弓形的两个端点和它所在的球冠的顶点，我们将弓形表示为一个扇形与一个三角形的和或差。该问题被转化为下面两个：

问题 2.2c 计算球面扇形的面积。

问题 2.2d 计算球面三角形的面积。

接着讨论问题 2.2b。连接 $P_S P$ 和 $P_S Q$ ，并判断 $\Delta P_S P Q$ 在 \overrightarrow{QP} 的哪一侧。如果它在右侧，如图 10b 所示，则计算 \overrightarrow{PQ} 的积分，再取其相反数。根据推论，有

$$\begin{aligned} S_{\Delta P_S P Q} &= \oint_{\partial \Delta P_S P Q} (-\sin \varphi - 1) d\lambda \\ &= \int_{\overrightarrow{P_S P}} (-\sin \varphi - 1) d\lambda + \int_{\overrightarrow{PQ}} (-\sin \varphi - 1) d\lambda + \int_{\overrightarrow{QP_S}} (-\sin \varphi - 1) d\lambda \\ &= \int_{\overrightarrow{PQ}} (-\sin \varphi - 1) d\lambda. \end{aligned}$$

最后一步是因为对于 $\overrightarrow{P_S P}$ 和 $\overrightarrow{QP_S}$ 而言， $d\lambda = 0$ ，所以其积分值为 0。

至此将问题 2.2b 转化为 2.2d。

通过简单的计算得到球冠的面积公式 $S = 2\pi h$ ，其中 h 是球冠顶点到底面的距离。利用球冠的对称性，我们容易导出扇形的面积公式

$$S = 2\pi h \theta,$$

其中 θ 是扇形的角度。

关于球面三角形的面积, [4] 指出

$$S_{\triangle ABC} = \angle A + \angle B + \angle C - \pi.$$

其中 $\angle A, \angle B, \angle C$ 分别表示以 A, B, C 为顶点的球面角。[7] 给出了一个数值稳定性更好的公式

$$\tan \frac{S_{\triangle ABC}}{2} = \frac{\tan \frac{1}{2}a \tan \frac{1}{2}b \sin \angle C}{1 + \tan \frac{1}{2}a \tan \frac{1}{2}b \cos \angle C},$$

其中 a, b 分别指 $\angle A, \angle B$ 的对边 BC 与 CA 的长度。

至此, 问题 2 已被完整解决。该算法的大体结构与上一章解决问题 1 的方法类似, 且具有相同的运行时间上界 $O(n^2 \log n)$ 。

4 布尔运算

现实应用中, 我们要计算的通常不是 n 个区域的并。比如, 地球周围的空间中有 n 颗定位卫星, 希望求出地表有多大的区域能与至少三颗卫星直连。再比如, 在辅助设计工艺品时, 用户反复使用交、并、补等运算, 人为地规定一个复杂区域。即使只考虑面积并问题, 也可以用三个半球面的交来表示球面三角形, 进而用三角形的并表示球面多边形。这就提出了计算更一般的复合区域面积的要求。

k 元布尔函数 $F(x_1, \dots, x_k)$ 是一个 $\{0, 1\}^k \mapsto \{0, 1\}$ 的映射。 x_1, \dots, x_k 称为其参数。

假设 D_1, \dots, D_n 是给定的球冠。可以用一个 n 元布尔函数 $F(\lambda_1, \dots, \lambda_n)$ 来定义复合区域 D^* :

$$D^* = \{P \mid F(\lambda_1, \lambda_2, \dots, \lambda_n) = 1\},$$

即

$$[P \in D^*] = F(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (\forall P),$$

其中 λ_i 表示点 P 是否在 D_i 中, 即 $\lambda_i = [P \in D_i]$ 。严格地说, λ_i 是一个关于 P 的函数 $\lambda_i(P)$, 但为了清晰, 在我们考察的点 P 十分明确的前提下, 省略了其参数 P , 下文将沿用这种省略。这两个式子指出布尔函数 F 定义了 D_1, \dots, D_n 的哪些组合在复合区域 D^* 中。

有一类基本的、简单的布尔函数, 称为布尔运算符。对于一个布尔运算符 $f(x_1, \dots, x_k)$, 必须存在一种数据结构, 支持:

- (1) INITIALIZE。初始化数据结构, 令 $x_i = 0$ ($\forall i \in [1, k]$)。
- (2) MODIFY(i, b)。修改参数的值, 令 $x_i = b$ 。
- (3) QUERY。查询函数的值, 即 $f(x_1, \dots, x_k)$ 。

其中 INITIALIZE 操作需要在 $O(k)$ 的时间复杂度内完成，而 MODIFY 和 QUERY 必须在常数运行时间内完成。

我们将布尔运算符的内部逻辑视为一个黑盒，而仅通过它的数据结构来获得信息。所有 $\Theta(1)$ 元的、可计算的布尔函数，都属于布尔运算符。

我们用递归的方式来定义布尔表达式。一个布尔表达式 α 要么是某个 λ_i ($i \in [1, n]$)，要么有

$$\alpha = f(\beta_1, \beta_2, \dots, \beta_k),$$

其中 f 是一个布尔运算符， β_1, \dots, β_k 分别是一个布尔表达式。

定义表达式 α 的长度

$$|\alpha| = \begin{cases} 1, & \text{if } \alpha = \lambda_i, \\ 1 + \sum_{j=1}^k |\beta_j|, & \text{if } \alpha = f(\beta_1, \dots, \beta_k). \end{cases}$$

我们以布尔表达式的形式来给出布尔函数 F ，正式提出以下问题：

问题 3 (球面复合区域面积问题) 给定球面上 n 个球冠形区域 D_1, \dots, D_n 。再给定一个长度为 m 的布尔表达式 α_f ，定义

$$D^* = \{P \mid F(\lambda_1, \lambda_2, \dots, \lambda_n) = 1\},$$

其中 $F(\lambda_1, \dots, \lambda_n) = \alpha_f$ 是由布尔表达式 α_f 定义的 n 元布尔函数。希望计算 D^* 的面积。

仍然假定 D_1, \dots, D_n 两两不同，且没有球冠大于半球。能够如此假设是因为大于半球的球冠能够用一个较小球冠的补集来表示。进一步地，我们假设它们的边界也互不重合，即不存在两个球冠 D_i 和 D_j ，它们的并等于整个球面。

利用上一章的方法，我们只需求出 ∂D^* 就可以算出 D^* 的面积，为此分别考察每一个圆 ∂D_i 。

在上一章中，

$$F = \lambda_1 \vee \lambda_2 \vee \dots \vee \lambda_n,$$

我们判断出，一段圆弧在 ∂D^* 中，当且仅当它没有被其它球冠覆盖，且方向相对于球冠内部是逆时针。这是因为，对于一段满足条件的有向圆弧，它的左侧（即相对于球冠而言的内侧）在 D^* 内，而右侧在 D^* 外，于是它是 ∂D^* 的成分；对于被其它球冠覆盖的圆弧，它的左右两侧均在 D^* 内，于是它就不是 ∂D^* 的成分。沿用此思路，如果 A 是 ∂D_i 的一段有向圆弧， A 的左侧在 D^* 内，右侧在 D^* 外，那么 $A \subset \partial D^*$ ；如果 A 的左侧在 D^* 外，右侧在 D^* 内，那么 A 的反向是 ∂D^* 的成分。为此，我们计算以下问题：

问题 3a 计算 ∂D_i 的所有部分，满足其左侧在 D^* 内。

问题 3b 计算 ∂D_i 的所有部分，满足其右侧在 D^* 内。

上述两个问题的答案可以快速合并，成为我们想要的信息。这两个问题唯一的区别是，对于 ∂D_i 左侧有 $\lambda_i = 1$ ；对于其右侧有 $\lambda_i = 0$ 。对于另外的 j ($j \neq i$)， ∂D_i 上被 D_j 覆盖的部

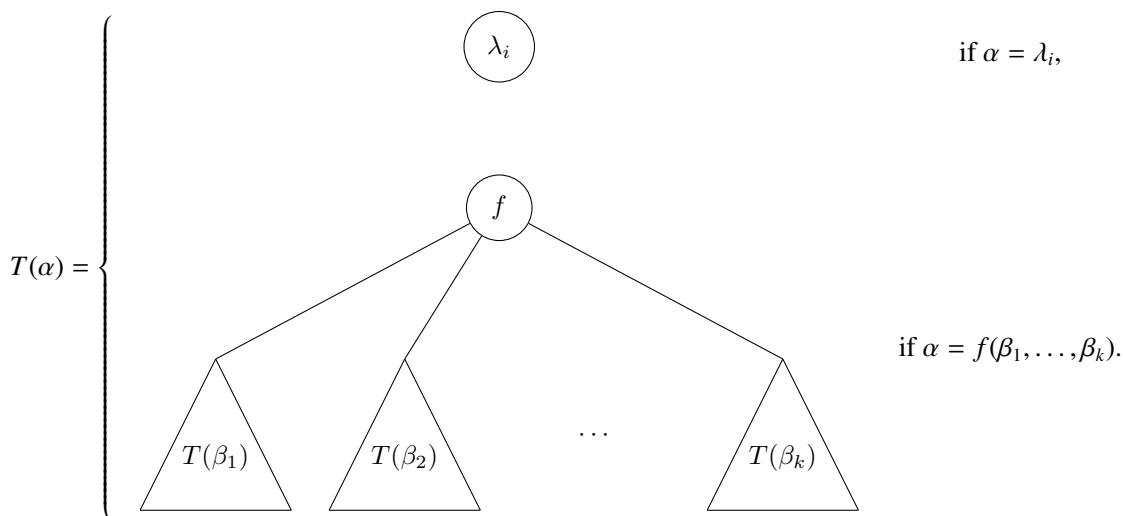
分有 $\lambda_j = 1$ ，未被 D_j 覆盖的部分有 $\lambda_j = 0$ 。这两个问题的解决方法也是一致的，下面只讨论问题 3a 的解法。

利用上一章定义的极角运算，将 ∂D_i 上的点映射到 $[-\pi, \pi)$ 上，问题 3a 转化为：

问题 3.1 $\forall i \in [1, n]$ ，区间 $[-\pi, \pi)$ 上至多两个给定区间使 $\lambda_i = 1$ ，对于其它位置有 $\lambda_i = 0$ 。求出使 $F(\lambda_1, \dots, \lambda_n) = 1$ 的所有位置，用 $[-\pi, \pi)$ 上若干个不交的区间来表示。

利用扫描线的思想，令动点 P 从 $-\pi$ 向 π 移动，并在移动的过程中维护 $F(\lambda_1, \dots, \lambda_n)$ 的值。移动过程中会有 $O(n)$ 个“事件”——某个 λ_i 的取值发生了变化， F 值也随之更新。 F 值从 1 变化为 0，以及从 0 变化为 1 的位置，就是某个所求区间的左右端点。

用表达式树来显示布尔表达式的结构。对于一个布尔表达式 α ，定义其表达式树



在第一种情况中， $T(\alpha)$ 唯一的结点称为“输入结点”；第二种情况中的根节点称为“运算结点”， f 称为该结点的运算符。表达式树是有根树，输入结点都是叶结点。运算结点的孩子是一个有序的列表，对应运算符的各个参数，顺序不可调换。

建出 α_f 的表达式树，下面所有讨论都在这棵树上进行。容易验证，一个表达式的长度和其表达式树的结点数相等，所以 $T(\alpha_f)$ 有 m 个结点。对于每一个结点 u ，称子树 u 表示 u 和它的全体后代形成的子树，子树 u 规定了一个布尔表达式，有时也用结点的名称 u 来指代该表达式，或是该表达式的值。

当 $\lambda_1, \dots, \lambda_n$ 取值确定后，定义结点 u 的值 $W(u)$ 为子树 u 规定的表达式的值，即

$$W(u) = \begin{cases} \lambda_i, & \text{if } u \text{ is an input node,} \\ f_u(W(v_1), W(v_2), \dots, W(v_k)), & \text{otherwise,} \end{cases}$$

其中 v_1, \dots, v_k 是 u 的子结点。在第二种情况中， f_u 指结点 u 的运算符。自底向上反复应用上式，就可以计算出根结点的值，即 $F(\lambda_1, \dots, \lambda_n)$ 。

当 $\lambda_1, \dots, \lambda_n$ 取值确定后，输入结点的值也被确定。这时，包含变量 λ_i 的输入结点等价于包含 0 元运算符的运算结点，该节点的值要么恒为 0，要么恒为 1，被 λ_i 的取值确定。

我们在表达式树上将它们表示为运算结点，这样表达式树就与 $\lambda_1, \dots, \lambda_n$ 没有直接关系了。但为了行文方便，我们仍然将这些结点称为“包含 λ_i 的输入结点”。当某个 λ_i 被修改时，包含 λ_i 的若干输入结点的运算符需要被修改。每个结点至多被修改 $O(1)$ 次。我们需要解决如下问题：

问题 3.2 设计一个数据结构，支持三种操作：

- (1) INITIALIZE-TREE。初始化数据结构。建出表达式树，令所有输入结点的值为 0。
- (2) MODIFY-TREE(u, b)。修改输入结点 u 的值，令 $W(u) = b$ 。
- (3) QUERY-TREE。查询根节点的值 $W(\text{root})$ 。

当一个叶子被修改的时候，它所有祖先的值也需要重新计算。在有些特殊的问题中，树的深度不超过 $O(\log m)$ ，则可以使用朴素方法，自底向上依次更新这些结点的值。

算法 1 数据结构的朴素实现

```

1: procedure INITIALIZE-TREE-SIMPLE
2:   for each node  $u$  bottom-up do
3:     assume  $u = f_u(v_1, \dots, v_k)$  ▷ 此处用结点名称指代其值
4:     let  $u.\text{processor}$  be a new data structure of operator  $f$ 
5:      $u.\text{processor}.\text{INITIALIZE}$ 
6:     for each  $v_i$  do
7:        $u.\text{processor}.\text{MODIFY}(i, W(v_i))$ 
8:     end for
9:      $W(u) \leftarrow u.\text{processor}.\text{QUERY}$ 
10:  end for
11: end procedure
12:
13: procedure MODIFY-TREE-SIMPLE( $u, b$ )
14:    $W(u) \leftarrow b$ 
15:   if  $u = \text{root}$  then exit
16:    $p \leftarrow u.\text{parent}$ 
17:   assume  $p = f_p(v_1, \dots, v_k)$  and  $v_i = u$ 
18:    $p.\text{processor}.\text{MODIFY}(i, b)$ 
19:   MODIFY-TREE-SIMPLE( $p, p.\text{processor}.\text{QUERY}$ )
20: end procedure
21:
22: function QUERY-TREE-SIMPLE
23:   return  $W(\text{root})$ 
24: end function

```

上面的伪代码展示了数据结构的朴素实现。INITIALIZE-TREE-SIMPLE 为每个结点创建其运算符的数据结构黑盒 $u.processor$ ，并将黑盒的参数设置为正确的值。子过程 MODIFY-TREE-SIMPLE 递归地依次访问需要重新计算的结点。对于每个结点而言，其参数中至多有一个改变了。第 18、19 行利用黑盒，在常数运行时间内将参数调整为正确的值，并取得新的运算结果。INITIALIZE-TREE-SIMPLE 的运行时间是 $\Theta(m)$ ，MODIFY-TREE-SIMPLE 的运行时间以树的最大深度为上界，QUERY-TREE-SIMPLE 的运行时间是常数。如果树的最大深度不超过 $O(\log m)$ ，则该算法已经足够高效。

在树深较大的情况下，使用树链剖分。对每个结点 u ，记 s_u 为子树 u 的结点数。对于非叶结点 u ，若它的孩子中 s 最大的结点是 v ，则称 (u, v) 是重边， v 是 u 的重孩子； u 到其它孩子的边是轻边，它们是 u 的轻孩子。以下性质成立：

- 重边形成若干条链，每个结点恰好属于一条链（链可以只有一个结点）。这些链称为重链。
- 从任意结点到根的路径至多经过 $O(\log m)$ 条轻边，至多经过 $O(\log m)$ 条重链。

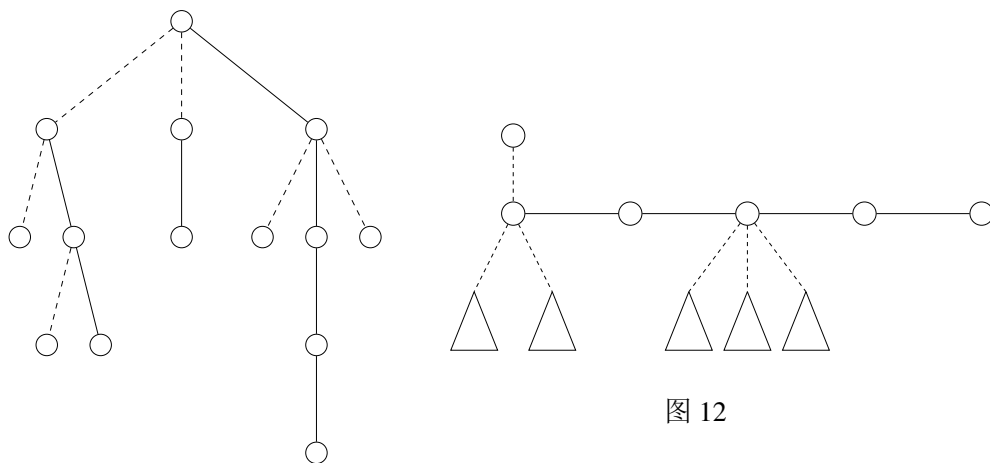


图 12

图 11

图 11 展示了一个树链剖分的结构。图中用实线表示重边，用虚线表示轻边。图 12 展示了一条重链的结构，图中为了清晰，将轻孩子画在链的下方，而实际上轻孩子与重孩子之间有固定的顺序。

对于任意结点 u ，假设 $u = f_u(v_1, \dots, v_k)$ ，其中 v_i 是其重儿子（即重链上的下一个结点）。那么，我们可以将除 v_i 外的 $k-1$ 个参数视作常量，定义

$$g_u(v_i) = f_u(v_1, \dots, v_i, \dots, v_k),$$

那么 $g_u(x)$ 便是一个一元布尔函数。特别地，如果 f_u 是 0 元的，没有参数，那么我们添加一个虚拟参数 x ，并令

$$g_u(x) = f_u(),$$

此时 $g_u(x)$ 虽然形式上是一元函数，但有 $g_u(0) = g_u(1)$ 成立，即函数值与参数值无关。此外，定义两个一元函数 f, g 的复合

$$(f \circ g)(x) = f(g(x)),$$

它满足结合律。

对于一条重链，按照顺序复合链上的 g_u 将得到一个一元函数 g^* ，它满足

$$g^*(0) = g^*(1) = W(r),$$

其中 r 是链顶，即这条链最顶部的结点。 g_u 的定义直接说明了上式的正确性。

对每一条重链，用线段树维护一元函数 g_u 的序列，并维护链顶的值 $W(r)$ 。当叶结点的 0 元运算符被修改时，它在重链上的一元函数也被修改。在这棵线段树上作出这个修改，并重新查询全体一元函数的复合，来计算 $W(r)$ 的值。 $W(r)$ 更新后， r 的父亲（如果存在）有一个轻孩子的值发生了变化，从而引起它的一元函数发生变化，进一步重复这个过程。算法 2 中的伪代码展示了一种实现。

在子过程 **MODIFY-CHILD-VALUE** 中，我们仍然利用运算符的黑盒来计算一元函数 g_u ，这个子过程的运行时间是常数。*segment-tree*.**QUERY** 查询的总是全体一元函数的复合，该信息已经存储在线段树的根结点中，所以运行时间也是常数。在一次 **MODIFY-TREE** 中，*segment-tree*.**MODIFY** 的调用次数不超过 $O(\log m)$ ，每次运行时间不超过 $O(\log m)$ ，故 **MODIFY-TREE** 的运行时间有上界 $O(\log^2 m)$ 。

在此基础上，使用“全局平衡二叉树”代替线段树，可以使 **MODIFY-TREE** 的运行时间上界降至 $O(\log m)$ ，见 [1]。另外，**INITIALIZE-TREE** 的运行时间是 $\Theta(m)$ ，**QUERY-TREE** 的运行时间是 $\Theta(1)$ 。我们已经得到问题 3.2 的一个高效解法。

接着，应用上述数据结构解决问题 3.1。在“扫描”开始之前，对数据结构进行初始化需要 $\Theta(m)$ 的时间。在“扫描”的过程中，表达式树上的每个叶子会被修改常数次，这一部分的运行时间上界是 $O(m \log m)$ 。于是，问题 3.1 以 $O(m \log m)$ 的运行时间上界被解决。

在本章的开头，我们把问题 3 转化为 $\Theta(n)$ 个互相独立的问题 3.1。从而利用上述方法得到问题 3 的一个 $O(nm \log m)$ 运行时间上界的算法。

算法 2 数据结构的高效实现

```

1: procedure INITIALIZE-TREE
2:   INITIALIZE-TREE-SIMPLE
3:   build heavy-light decomposition
4:   calculate  $g_u$  ( $\forall u$ )
5:   for each chain  $c$  do
6:     let  $c.segment-tree$  be a new segment tree of  $g_u$ 
7:     initialize  $c.segment-tree$ 
8:   end for
9: end procedure
10:
11: procedure MODIFY-CHILD-VALUE( $u, v, b$ )
12:   assume  $v_1, \dots, v_k$  are children of  $u$ 
13:   assume  $v = v_i$ , and  $v_j$  is the heavy-child of  $u$  ( $i \neq j$ )
14:    $u.processor.MODIFY(i, b)$ 
15:    $u.processor.MODIFY(j, 0)$ 
16:    $g_u(0) \leftarrow u.processor.QUERY$ 
17:    $u.processor.MODIFY(j, 1)$ 
18:    $g_u(1) \leftarrow u.processor.QUERY$ 
19: end procedure
20:
21: procedure RECALCULATE( $u$ )
22:   assume  $u$  is the  $i^{\text{th}}$  node of chain  $c$ 
23:    $r \leftarrow c.top$ 
24:    $c.segment-tree.MODIFY(i, g_u)$ 
25:    $g^* \leftarrow c.segment-tree.QUERY$ 
26:    $W(r) \leftarrow g^*(0)$   $\triangleright$  此时  $g^*(0) = g^*(1)$ 
27:   if  $r \neq \text{root}$  then
28:      $p \leftarrow r.parent$ 
29:     MODIFY-CHILD-VALUE( $p, r, W(r)$ )
30:     RECALCULATE( $p$ )
31:   end if
32: end procedure
33:
34: procedure MODIFY-TREE( $u, b$ )
35:    $g_u(0) \leftarrow b, g_u(1) \leftarrow b$ 
36:   RECALCULATE( $u$ )
37: end procedure
38:
39: function QUERY-TREE
40:   return  $W(\text{root})$ 
41: end function

```

5 应用与讨论

在本章，我们讨论球面复合区域面积问题的几种变形、推广，以及它们的应用。

球面多边形

球面多边形是由若干条球面线段首尾相连而围成的区域，现实中常用来表示地表的某个区域。

我们的算法能够处理球面多边形。我们可以用三个“半球面”的交表示三角形，其中半球面是指大圆某一侧的区域。进一步地，使用球面三角形的并来表示球面多边形，在这里我们并不反复使用常见的二元“逻辑或”运算符，而是直接使用 k 元逻辑或运算符。上述方法中，球面多边形是由某个表达式规定的，而这个表达式树的深度是常数。表示出球面多边形之后，它们可以继续参与布尔运算，来形成更复杂的复合区域。

平面复合区域

与实际应用不同的是，在算法竞赛中，常常将问题抽象为平面上的面积问题。这是因为球面几何代码编写繁琐，且通常与思维考察点关系不大，故通过抽象为平面问题将其省去了。

通过将第二章与第四章的内容进行结合，我们能够直接得到一个算法，它能够计算平面上若干个圆进行布尔运算规定的复合区域的面积。但是这还不够，因为球面上的“直线”是一种特殊的圆，而平面上不是。为此，除了圆，我们再引入另一种区域——半平面。

半平面是指一条有向直线左侧的区域。这条有向直线就是半平面的边界，且拥有正确的定向。

模仿极角运算，我们可以定义关于有向直线的点积运算。假设一条有向直线上的点集是

$$\vec{a} + \vec{b}t \quad (t \in \mathbb{R}),$$

其中 t 沿直线方向增大。定义直线上一个点 P 的点积

$$\text{Dot}(P) = (P - \vec{a}) \cdot \vec{b},$$

右侧是两个向量的点积。点积运算建立了有向直线上的所有点到区间 $(-\infty, +\infty)$ 的一一映射。容易验证, 对于任意两个区域 D_i 和 D_j , 它们都是圆或者半平面, D_j 覆盖 ∂D_i 的部分映射到数轴上是至多两个区间的并。这样, 我们就可以继续沿用第四章的方法, 以扫描线的思想来求出复合区域的边界 ∂D^* 。对 ∂D^* 的积分是简单的。

如此，我们就解决了含圆和半平面的“平面复合区域面积”问题，运行时间有同样的上界 $O(nm \log m)$ 。用半平面表示平面上的多边形是容易的，这里不再赘述。该方法可以直接通过三道 NOI 中出现过的此类问题：

- NOI 2004 《降雨量》;
- NOI 2005 《月下柠檬树》;
- NOI 2009 《描边》。

容易计算边界的情形

在某些题目中，因为题目具有特殊性质，我们可以采用针对题目特殊性的方法，以快速计算 ∂D^* 。分段计算 ∂D^* 的曲线积分不会成为运行效率的瓶颈，因为这一部分的运行时间正比于 ∂D^* 中圆弧的数量，这是前一个阶段（即快速计算 ∂D^* ）的输出。

例 1 《月下柠檬树》^[2] 给定平面上的 n 个圆 A_1, \dots, A_n ，它们的圆心都在 x 轴上，且圆心的 x 坐标递增。对于相邻两个圆 A_i 与 A_{i+1} ，如果它们互不包含，则作出它们的两条外公切线，并将四个切点连成梯形，记这个梯形区域为 B_i 。定义

$$D^* = \bigcup_i A_i \cup \bigcup_i B_i,$$

计算 D^* 的面积。

在本题中， ∂D^* 关于 x 轴对称，只需要计算 x 轴上方的部分，记为 C 并忽略方向。将每一个圆 A_i 与它左侧的梯形 B_{i-1} （如果存在）的并记为 A'_i 。初始状态令 C 为空，从左到右依次加入 A'_i ，并在每一个图形加入后计算新的边界 C' 。容易证明，每个 A'_i 加入时，原先的 C 恰有一个“后缀”（最右侧连续的一段）被 A'_i 覆盖，从而被删除，代之以 $\partial A'_i$ 的某个部分。用栈维护这一过程，就可以在 $\Theta(n)$ 的运行时间内计算出 C ，从而得到 ∂D^* 。

综上，我们得到原问题的一个 $\Theta(n)$ 运行时间的算法。该算法达到了本题的时间复杂度理论下界，远快于出题者参考算法的 $O(n^3)$ 。

DAG 形式的布尔函数

在第四章中，我们用树形结构来表示从各个 λ_i 到最终布尔函数的计算过程。我们用这个结构解决实际问题，很大程度上依靠了布尔运算符的灵活性。例如，我们知道

$$a \oplus b = (a \vee b) \wedge \neg(a \wedge b),$$

即我们可以用基本的逻辑运算符 \neg 、 \wedge 、 \vee 去定义“异或”运算。但右侧的展开形式在我们的算法中是行不通的，因为 a 、 b 各出现了两次，而树形结构没办法将一个结点的值作为多个运算符的输入。如果将整棵子树复制一份，时间复杂度将上升为指数级，无法接受。我们采取的方法是直接定义一个“异或”运算符，在这个运算符内处理中间变量的复用。

上述方法只能处理“局部的”变量复用。在计算机辅助设计中，常常出现以下情况：整个运算过程都由用户给定，每一次运算结果都会被存储为中间变量，它们在将来能够被

任意访问。对于这样的高度自由的运算过程，我们只能用 DAG（有向无环图）来表示它。在图中，每一个结点表示一个中间变量，它是一个运算符的直接结果，运算符的各个参数对应结点的各条入边。此处，一个结点的入边是有序的列表，对应运算符的各个参数的顺序，不能随意调换。

通过牺牲一些运行效率，我们的算法能够处理这种情况。假定 DAG 的边数是 m 。那么当 $\lambda_1, \dots, \lambda_n$ 的值确定后，最终的布尔函数 F 的值可以在 $O(m)$ 的时间内计算得到。我们不再使用数据结构去维护 F 值，而是在某个变量发生变化后，以 $O(m)$ 的时间完全重新计算 F 值。整个算法的时间复杂度为 $O(n^2m)$ 。

例 2 《计算体积》^[3] 输入 n 条指令，每条指令为以下几种之一：

- (1) 创建一个球；
- (2) 创建一个每条边都和坐标轴平行的长方体；
- (3) 输入在这条指令前面的两条指令编号 i 、 j ，创建一个立体图形为第 i 条指令与第 j 条指令的交/并/差。

对于每一条指令，输出它创建的立体图形的体积。

离线后对所有图形一起处理。本题的数据规模、值域、精度要求都比较低，可以对一个维度使用数值积分，将问题转化为二维问题。这个二维问题的输入只包含圆和长方形，运算由 DAG 给出，DAG 共有 $O(n)$ 条边。特别的是，我们需要求出 DAG 上每一个结点 u 的值规定的复合区域 D_u^* 的面积，即所有结点都是输出结点。当某个变量 λ_i 改变时，重新计算 DAG 上所有结点的值总共需要 $O(n)$ 的时间，故解决二维子问题需要 $O(n^3)$ 的时间。解决原问题的时间复杂度是 $O(n^3I)$ ，其中 I 是数值积分方法调用求值函数的次数。

球并的表面积

例 3 给定空间中 n 个球，计算它们的并的表面积。

对于一个球面而言，其中未被其它球覆盖的部分，就成为球并的表面。分别对于每一个球面，计算未被覆盖部分的面积，即球面上

$$D^* = U - \bigcup_i D_i$$

的面积，其中 U 是整个球面， D_i 是该球面被第 i 个球覆盖的区域，是一个球冠。这就将问题转化为球面复合区域面积问题，得到一个 $O(n^3 \log n)$ 的算法。

另一个有趣的问题是计算球并的体积。虽然上述方法可以求出球并的边界，但笔者尚未找到计算球并的体积的精确算法。这个问题尚待继续研究。

6 总结

本文介绍了球面复合区域面积问题，并提出格林公式方法来解决它。该问题在现实中有广泛的应用，主要是在地球表面上的面积问题。球面几何因其代码冗长、细节繁多，不适合用于算法竞赛的命题，但我们的算法能够作一些修改以适应平面情形，从而在算法竞赛中体现其优势。

然而，对这个问题的研究还没有结束。在实际应用中，如果我们需要一块区域更精确的面积，就应该将地球表面抽象成一个椭球面，从而需要解决椭球面复合区域面积问题。此外，第五章中对于 DAG 形式的布尔函数，我们仅仅得到一个运行时间上界为 $O(n^2m)$ 的算法，它的运行效率或许有进步空间。上述两个相关问题尚待进一步研究。

致谢

感谢周航锐同学提供《月下柠檬树》^[2]的线性算法及其程序实现，并协助了部分图片的绘制工作。

感谢徐哲安同学与我讨论格林公式方法的应用。

参考文献

- [1] 杨哲. SPOJ375 QTREE 解法的一些研究. 中国信息学奥林匹克国家集训队作业, 2007: 9-11.
- [2] 胡伟栋. 月下柠檬树. 全国青少年信息学奥林匹克竞赛, 2005.
<https://www.luogu.com.cn/problem/P4207>.
- [3] 徐明宽. 二维计算几何相关算法与实战应用. 全国青少年信息学奥林匹克冬令营, 2018: 70-72.
- [4] 人民教育出版社, 课程教材研究所, 中学数学教材实验研究组. 普通高中课程标准实验教科书数学选修 3-3 (B 版): 球面上的几何[M]. 人民教育出版社, 2008: 27-28.
- [5] 常庚哲, 史济怀. 数学分析教程[M]. 第 3 版. 中国科学技术大学出版社, 2012.
- [6] Area. ACM/ICPC Asia Regional Nanjing Online, 2013.
<http://acm.hdu.edu.cn/showproblem.php?pid=4748>.
- [7] Todhunter I. *Spherical Trigonometry, for the use of colleges and schools: with numerous examples*[M]. Macmillan, 1863: 67-71.