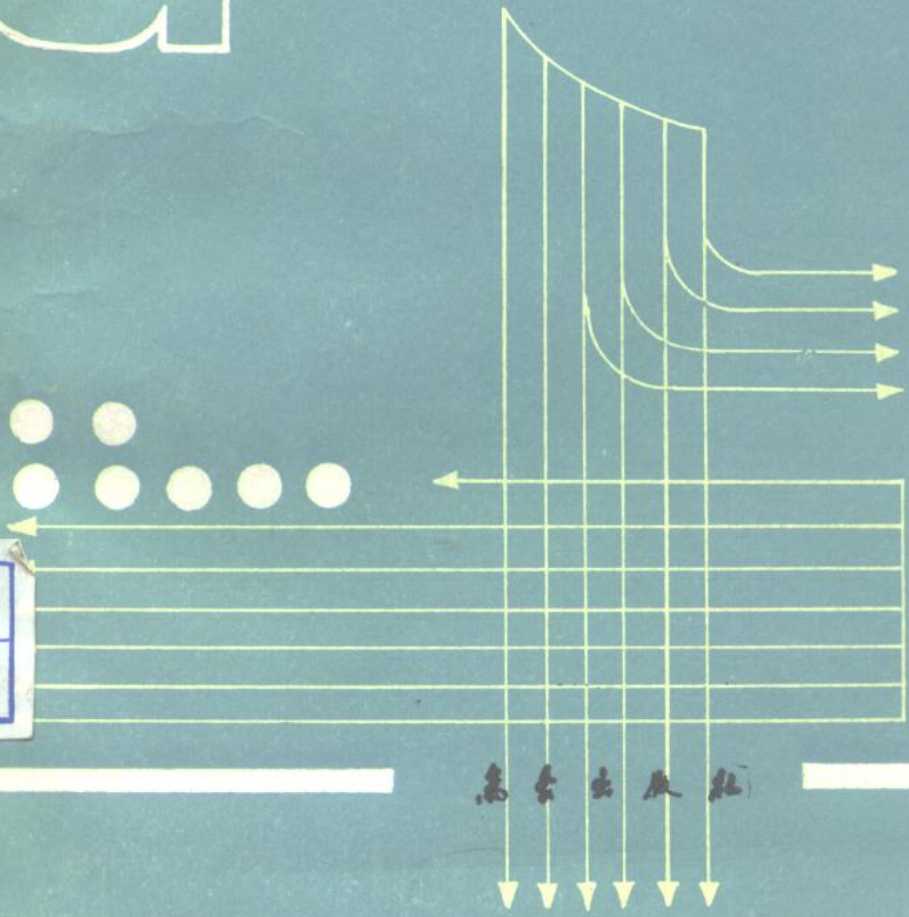
赵瑞清、孙宗智编著

计算复杂性概论



高等教育出版社

计算复杂性概论

赵瑞清
孙宗智 编著

气象出版社

内 容 简 介

计算复杂性是计算机科学的重要基础理论之一，本书力求使读者了解该领域的基本概念、问题和方法。第一章组合复杂性与第二章算法的计算复杂性与计算模型是基础，后三章重点是介绍P、NP及NP-完全问题。

本书可作为计算机专业和有关专业的教材及供有关研究工作者参考。

计算复杂性概论

赵瑞清　孙宗智　编著

责任编辑　黄丽荣

气象出版社出版

（北京西郊白石桥路46号）

北京印刷一厂印刷

气象出版社发行　新华书店经售

开本：787×1092　1/32　印张：6.375 字数：130 千字

1989年3月第一版　1989年3月第一次印刷

印数：1—3500　定价：2.80 元

ISBN 7-5029-0169-8/TP·0007

科技新书目：178-293

前　言

关于计算复杂性，尤其是其中的NP-完全问题，人们认为是计算机科学理论中最重要的问题，自从1971年S.Cook提出P类问题是否等价NP类问题以来，许多著名科学家对这个问题进行了深入探讨，有的企图证明它们等价，更多的人猜测并企图证明它们不等价。但迄今为止，人们并没有获得完全的成功，并且越来越多的迹象表明，这是一个十分困难的问题，从而带动了许多问题的研究，发展起一个庞大的理论系统。这些研究无论是对增进人们对P和NP本质的了解，还是对许多问题的复杂性的解决，都有着重大的意义，这也许是Cook最初也未预料到的。

计算复杂性是一门很重要的多课程，各校计算机系都将纷纷开出这一门课，可是至今还未见关于计算复杂性的专门中文著作。基于这种情况，我们在几年教学教材的基础之上，经过几次修改，编写成此书，企图引起人们对计算复杂性的关注。

本书共分五章，前二章：组合复杂性与算法的计算复杂性和计算模型，是计算复杂性中的一些基本概念与理论。后三章重点是讨论P与NP及NP-完全问题。

计算复杂性自1971年以来，发展得异常迅速，取得了许多成果。在本书中，限于篇幅，不能一一详细介绍，有兴趣的读者，可参阅书中所列出的有关参考文献。

由于水平所限，不当之处在所难免，敬请读者批评指正。

作者

1987年4月12日

引　言

早在30年代计算机问世之前，人们就致力于算法及计算模型的探讨。Alan Turing在1937年在“可计算数以及对Entscheidungs问题的应用”中，引入了著名的图灵机，提供了有效可计算函数(或算法)的概念的最令人信服的形式表示。在可计算理论的早期工作中，一是描述和刻划了那些算法可解问题的特征，二是展示了一些算法不可解问题。在充分阐明了哪些问题能用计算机而哪些问题不能用计算机解决的理论之后，很自然地就要询问可计算函数的相对难度，这就是计算复杂性的主题。

到了本世纪60年代，计算复杂性概念日趋明确，至今已成为计算机科学的基本理论之一。那末什么是计算复杂性呢？

我们知道，仅仅从理论上证明一类问题是可计算的还不够，因为有这样的问题，它在理论上是可计算的，但由于计算太复杂，以至于实际上成为不可计算。这是因为：一方面计算常常是有时间要求的，例如，明天的天气预报，若计算结果后天才能出来，这就无意义了；另一方面，任何一台电子计算机都有损耗问题，如果计算机平均三年出一次毛病，那么30年才能计算出来的问题，也应该认为实际上是不可计算的了。

计算复杂性就是研究问题的计算有多“难”，在这里，这个“难”字很模糊，让我们先看一个复利计算的例子：

$$30(\text{元})\times(1+0.06)^{20}=S$$

算法1：$30\times1.06\times1.06\times\cdots\times1.06$(20个1.06)；

算法 2：令 $a=1.06, a^2=a\times a, a^4=a^2\times a^2$，将 a^4 的值存入寄存器，$a^8=a^4\times a^4$，那末有：

$$S=a^8\times a^8\times a^4\times 30$$

算法 3：令 $a=1.06, a^2=a\times a, a^4=a^2\times a^2, a^5=a^4\times a, a^{10}=a^5\times a^5, a^{20}=a^{10}\times a^{10}$，那末有：

$$S=30\times a^{20}$$

同样一个问题，算法 1 用了 20 次乘法；算法 2 用了 6 次乘法，但需要一个存贮单元；而算法 3 也只用了 6 次乘法，又不用存贮单元。我们要问：这个问题到底有多么复杂？因为从上面可以看出，复杂程度与算法有关，所以人们要问：这个问题最少需要多少个运算呢？算法 3 是否是最好的呢？等等。这也就是要研究这个问题的复杂性，这是针对这一具体问题而言的，那末对于一般问题，应该怎样衡量计算的复杂性呢？

一般地，考虑一个问题类的计算，对于这个问题类中的每一个问题都要给它的大小以一个度量，通常是用一个自然数 n 来度量它的大小，这 n 视不同的问题类有不同的含义。如对函数来说，这 n 可能是自变量的个数；对矩阵来说，这 n 可能是其元素数目；对图来说，这 n 可能是顶点数，对多项式来说，这 n 可能是其阶数；对集合来说，这 n 可能是该集合中元素的个数等等。

对于大小为 n 的问题，如果计算它最多需要时间为 $g(n)$，就说这个问题类的时间复杂性为 $g(n)$；如果计算它最多需要内存为 $h(n)$，就说这个问题类的空间复杂性为 $h(n)$。所以一般说来，一个问题的复杂性是其大小 n 的函数。

另外，有各种不同的复杂性概念，以计算机为例，比较容易理解的有时间复杂性与空间复杂性。而对一般计算模型而

言，每个模型都有它各种不同资源，对于每种资源都有复杂性概念，它是问题大小为 n 所消耗的资源多少。

也许有人会认为，随着电子计算机速度及可靠性的提高，计算复杂性理论的研究就越来越不重要了。事实上，恰恰相反，正因为计算机速度的提高，人们才越来越要求用它来计算大的问题。

如果一个问题类，它具有多项式算法［这样的问题类称“易型”(tractible)问题］，这时可以认为这个问题是实际可计算的，它具有一个“好算法”。这是因为这类问题随着问题大小的增加，所须时间增加不太快；如果该问题只能有指数时间的算法［该问题为“顽型”(intractable)问题］，这时这个问题被认为在计算机上实际是不可计算的，虽然也有一个算法，但被认为是“坏算法”。这是因为这类问题随着问题大小的增加，计算时间增加很快。是否大大加速计算机运算速度，就能解决问题呢？如复杂性为 2^n 时，即使机器运行速度加快了1000倍，所解决问题的大小增大还不到 10。从而可以看出，改进算法是多么重要了。

计算复杂性有很宽阔的领域。本书将概括介绍的是由 Shannon (1949) 提出 Savage[1] 做了大量工作的组合复杂性理论；以实际计算机的简单模型 RAM 机为工具，介绍一些有效的算法的设计及分析；NP-完全问题及近似算法，这是近来研究的热门课题。这些仅仅是计算复杂性理论的几个基本方面。对该领域的比较全面的介绍，在 S.A.Cook 的《计算复杂性综述》[2]一文中可以找到。

目　录

第一章　组合复杂性

二进制函数的组合复杂性的研究，最早见于1938年Shannon的论文，现在它也是计算复杂性理论中值得注意的分支，在这方面有许多重要著作。

在这一章里将引进一种计算模型组合机；讨论布尔函数的组合复杂性的一般概念。并对一些基本的布尔函数的组合复杂性进行研究，从中介绍基本方法和主要结果。

这章的内容也是第二章问题的基础。

§1.1　布尔函数

一、布尔函数及其表示

定义　定义域为$\{0,1\}^n$，n正整数，值域为$\{0,1\}$的函数，称为布尔函数，并表为：

$$y=f(x_1,x_2,\cdots,x_n)$$

其中x_i称为f的变量，$i=1,2,\cdots,n$，$\{0,1\}^n$表示$\{0,1\}$的n重笛卡儿积，即二进制的n维集，$x_i\in\{0,1\}$，$i=1,2,\cdots,n$，$y\in\{0,1\}$。

函数f也常表示为：

$$f:\{0,1\}^n\to\{0,1\}$$

在实际中，经常考虑的是m个布尔函数集$f_1,f_2,\cdots,f_m$，其中，$f_i:\{0,1\}^n\to\{0,1\}$，$i=1,2,\cdots,m$。在这种情况下，将这个函数集表示为：

$$f:\{0,1\}^n\to\{0,1\}^m$$

是方便的，它的值域为二进制m维集。

下面是四个常用函数及其表示，一个函数 $f:\{0,1\}^n\to\{0,1\}^m$ 完全可以由一个表来表示，表 1.1 表示四个重要函数：

表 1.1 四个重要布尔函数表

“否”或“布尔补”

x_1	f_-
0	1
1	0

“析取”或“布尔或”

x_1	x_2	f_+
0	0	0
0	1	1
1	0	1
1	1	1

“合取”或“布尔积”

x_1	x_2	$f_\cdot$
0	0	0
0	1	0
1	0	0
1	1	1

“异或”(与模 2 等价)

x_1	x_2	$f_\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

二、当变量的个数固定时，布尔函数有多少？

1．用枚举法可以证明一共有四个一个变量的函数

- 恒等函数 $f_1(x)=x$；
- 布尔补 $f_-(x)=\bar{x}$；
- 常数函数 $f_0(x)=0$；
- 常数函数 $f_1(x)=1$.

2．可以证明有 2^{2^n} 个 n 个变量的布尔函数

这是因为 n 个布尔变量的不同排列有 2^n 种，而每种排列既可对应 0，也可对应 1，所以有 2^{2^n} 个。

3．同理有 $(2^m)^{2^n}$ 个 n 个变量的函数

$$f:\{0,1\}^n\to\{0,1\}^m.$$

三、复合函数

经常由一些简单函数，组成一些复杂的函数，其中方法之一是复合的方法。

定义 2　给定

$g:\{0,1\}^p\to\{0,1\}^m$

$h_i:\{0,1\}^n\to\{0,1\}$　　$1\leqslant i\leqslant p$

g 的复合表示为：$g(h_1,h_2,\cdots,h_p)$定义一个函数

$$f:\{0,1\}^n\to\{0,1\}^m$$

其规律如下：

$$f(x)=g(h_1(x),h_2(x),\cdots,h_p(x))$$

对所有的 $x=(x_1,x_2,\cdots,x_n)\in\{0,1\}^n$.

例 1　函数 $f(x_1,x_2,x_3)=f.(f.(x_1,x_2),x_3)$ 当且仅当 $x_1=x_2=x_3=1$ 才有值为 1.

四、函数 $f_-,f_+,f.,f_\oplus$ 的某些性质

这四个函数在形成复合函数时经常运用，所以简记为：

$f_-(x_1)=\bar{x}_1$,　　$f_+(x_1,x_2)=x_1+x_2$,

$f.(x_1,x_2)=x_1\cdot x_2$,　　$f_\oplus(x_1,x_2)=x_1\oplus x_2$.

还有一个重要函数：投影函数 p_i：

$$\{p_i(x_1,x_2,\cdots,x_n)=x_i\mid 1\leqslant i\leqslant n\}$$

下面是性质：

定理 1　下述规律成立

结合律：$x_1+(x_2+x_3)=(x_1+x_2)+x_3$

$x_1\cdot(x_2\cdot x_3)=(x_1\cdot x_2)\cdot x_3$

$x_1\oplus(x_2\ominus x_3)=(x_1\oplus x_2)\oplus x_3$

交换律：$x_1+x_2=x_2+x_1$,　$x_1\cdot x_2=x_2\cdot x_1$

$x_1\oplus x_2=x_2\oplus x_1$

分配律：$x_1\cdot(x_2+x_3)=(x_1\cdot x_2)+(x_1\cdot x_3)$

$$x_1\cdot(x_2\oplus x_3)=(x_1\cdot x_2)\oplus(x_1\cdot x_3)$$

$$x_1+(x_2\cdot x_3)=(x_1+x_2)\cdot(x_1+x_3)$$

D·摩根律：$\overline{x_1+x_2}=\bar{x}_1\cdot\bar{x}_2$，$\overline{x_1\cdot x_2}=\bar{x}_1+\bar{x}_2$

吸收律：$x_1+\bar{x}_1=1$，$x_1\cdot x_1=x_1$

$$x_1\cdot\bar{x}_1=0,\quad x_1\cdot x_1=x_1$$

$$x_1+x_1\cdot x_2=x_1\cdot(x_1+x_2)=x_1$$

常数代换：$x_1+0=x_1$，$x_1+1=1$，$x_1\cdot 0=0$

$$x_1\cdot 1=x_1,x_1\oplus 0=x_1,\quad x_1\oplus 1=\bar{x}_1$$

这定理的证明留作练习。

由于结合律成立，所以对+，.，⊕的复合运算可不用括号。

五、布尔函数的典型表示

1．析取范式(DNF)

它是由变量 x_i 和 +，.，－复合而成，具体进行如下：

若 $f:\{0,1\}^n\to\{0,1\}$ 在点 $(x_1,x_2,\cdots,x_n)=(c_1,c_2,\cdots,c_n)$有值为1，其中 $c_i\in\{0,1\}$，则我们形成极小项函数：

$$m_{c_1,c_2,\ldots,c_n}(x_1,x_2,\cdots,x_n)=x_1^{c_1}\cdot x_2^{c_2}\cdot\cdots\cdot x_n^{c_n}$$

其中 $x_i^1=x_i$，$x_i^0=\bar{x}_i$。这函数仅当$(x_1,x_2,\cdots,x_n)=(c_1,c_2,\cdots,c_n)$有值为1。则 f 的 DNF 由下式给出：

$$f(x_1,x_2,\cdots,x_n)$$

$$=\sum_{(c_1,\cdots,c_n)\in f(c_1,\cdots,c_n)=1}M_{c_1,\ldots,c_n}(x_1,\ x_2,\cdots,x_n)$$

其中∑表示布尔 OR 运算，其和取遍二进制 n 维向量。

例 2　求 $f(x_1,x_2,x_3)=x_1\oplus x_2\oplus x_3$ 的 DNF

解：∵f 仅在点：(0,0,1)，(0,1,0)，(1,0,0)，(1,1,1)上有值为1，∴它的 DNF 为：

$$x_1\oplus x_2\oplus x_3=x_1^0\cdot x_2^0\cdot x_3^1+x_1^0\cdot x_2^1\cdot x_3^0+x_1^1\cdot x_2^0\cdot x_3^0+x_1^1\cdot x_2^1\cdot x_3^1$$

$$= \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

2．合取范式(CNF)

先定义极大项函数：

$$s_{c_1,c_2,\cdots,c_n}(x_1, x_2, \cdots, x_n) = x_1^{c_1} + x_2^{c_2} + \cdots + x_n^{c_n}$$

则它仅当 $x_i = c_i, 1 \leqslant i \leqslant n$ 时其值为 0，这样，f 的 CNF 为：

$$f(x_1, x_2, \cdots, x_n) = \prod_{(c_1,\cdots,c_n) \in f(c_1,\cdots,c_n)=0} s_{c_1,\cdots,c_n}(x_1, x_2, \cdots, x_n)$$

其中 Π 表示 AND，其积遍取二进制 n 维向量。

可以证明，这展开式可从 $\bar{f}$ 的 DNF 应用 D·摩根律而得，而且 $s_{c_1,\cdots,c_n}(x_1, x_2, \cdots, x_n)$ 是极小项 $m_{c_1,\cdots,c_n}(x_1, x_2, \cdots, x_n)$ 的否定。

3．环和展开式(RSE)

它可从 f 的 DNF 中用恒等式 $\bar{x}$ 用 $x \oplus 1$ 及 $x_1 + x_2$ 用 $x_1 \oplus x_2 \oplus x_1 x_2$ 代替经过整理而得。如极小项 $x_1 \bar{x}_2 \bar{x}_3$，可化为：

$$x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = x_1 \cdot (x_2 \oplus 1)(x_3 \oplus 1) = x_1 \oplus x_1 x_2 \oplus x_1 x_3 \oplus x_1 x_2 x_3$$

这样在f的DNF 中每个极小项经过变形，于是 f 的 RSE 如下：

$$f(x_1, x_2, \cdots, x_n) = a_0 \oplus \sum_{j=1}^{n} \sum_{1 \leqslant i_1 < i_2 < \cdots < i_j \leqslant n} a_{i_1,\cdots,i_j} \cdot x_{i_1} \cdot x_{i_2} \cdots x_{i_j}$$

其中 $a_0, a_{i_1,i_2,\cdots,i_j} \in \{0,1\}$，$\sum$ 表示模 2 加的重复。

4．积和展开式(SOPE)

f 被展成 $x_{i_1}^{c_1} \cdot x_{i_2}^{c_2} \cdot \cdots \cdot x_{i_l}^{c_l}, 1 \leqslant l \leqslant n$ 的布尔积的 OR。

实际上，这是对 f 的 DNF 的一个改善，即减少操作数，如 $x_1 \bar{x}_2 \bar{x}_3$ 和 $x_1 \bar{x}_2 x_3$ 可合并为 $x_1 \bar{x}_2$。一般说来，操作数确是减少了，但也有的 f 的 SOPE 与 f 的 DNF 恒等，如

$$f(x_1, x_2, \cdots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

它的 SOPE 与 DNF 等价，其证明如下。

定理 2　函数 $f(x_1,x_2,\cdots,x_n)=x_1\oplus x_2\oplus\cdots\oplus x_n$ 有最少的文字出现的积和展开式，恒等于 f 的 DNF，而且包含 $n\,2^{n-1}$ 个文字。

证：这个函数在 2^{n-1} 个点上有值为 1，而且如果它的任何一个变量变的话(其它变量任取后不变)，它的值也变，将 f 的 SOPE 记为：

$$f=p_1+p_2+\cdots+p_k$$

其中 $p_i, i=1,2,\cdots,k$ 是文字之积。

若 p_i 中不包含某一个变量或其补，比方说不包含 x_1，这样就存在其它 $x_j(j\neq 1)$ 的一种取值方法，使变量 x_1 不管取值为 0 或 1，都有 p_i 等于 1。从而使 f 的值与 x_1 无关，这与 f 的定义矛盾，所以每一个 p_i 是一极小项。于是它的 SOPE 与 DNF 一致，即 $k=2^{n-1}$，而每个 p_i 都包含 n 个文字，所以一共有 $n\,2^{n-1}$ 个文字。

§ 1.2　计算链和组合机

计算复杂性与计算模型有一定关系，我们先介绍组合机下的复杂性。

一、计算链(简称为链)

链是计算一个布尔函数的依赖于数据的计算步的序列，它是逻辑电路的抽象表示。

定义 1　给定布尔函数集 $\Omega=\{h_i\mid h_i:\{0,1\}^{n_i}\to\{0,1\}\}$，称为基和 n 个变量 $x_1,x_2,\cdots,x_n$ 及常数 0，1 的数据集 $\Gamma=\{x_1,x_2,\cdots,x_n,0,1\}$，一个 k 步链：

$$\beta=(\beta_1,\beta_2,\cdots,\beta_k)$$

是一个有序的 k 步：$\beta_1,\beta_2,\cdots,\beta_k$ 的集，其中：

或 $\beta_j\in\Gamma$ 或

$$\beta_j=(h_i;\beta_{j_1},\beta_{j_2},\cdots,\beta_{j_{ni}})$$

这里，$1\leqslant j_r<j$，$1\leqslant r\leqslant n_i$，$h_i\in\Omega$.

第一型的步称为数据步(deta steps)，另一类的步称为计算步(computation steps).

所以链是一个描述函数计算步骤的一个序列，每一步对应于一个布尔函数 $\tilde{\beta}_j$：

若 $\beta_j\in\Gamma$，就给出一个函数 $\tilde{\beta}_j=\beta_j$.

若 $\beta_j\notin\Gamma$，则 $\tilde{\beta}_j=h_i(\tilde{\beta}_{j_1},\cdots,\tilde{\beta}_{j_{ni}})$

显然，复合是这里唯一使用的规则。

二、链的图解

链的图可如下构成：每个结点标上 β 的一步，若 β_j 使用 β_{j_r}，则从 β_{j_r} 至 β_j 联上一条有向边，标上变量或常量的结点称为原始结点，另一些结点称为计算结点。

三、完备基

一个链 β 被说成去计算 $f_1,f_2,\cdots,f_m$，其中每个 $f_r:\{0,1\}^n\to\{0,1\}$，若存在 m 步：$\beta_{i_1},\beta_{i_2},\cdots,\beta_{im}$，使 $\tilde{\beta}_{ir}=f_r$，$1\leqslant r\leqslant m$。

若每一个函数 $f:\{0,1\}^n\to\{0,1\}^m$，在基 Ω 和数据集 Γ 上，存在一个链 β 能计算 f，则称 Ω 为完备基。

如基$\{+,\cdot,-\}$和基$\{\oplus,\cdot\}$是完备的。

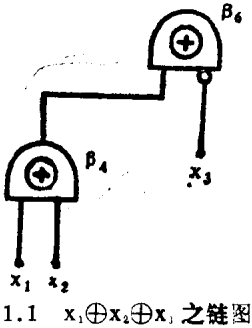


图 1.1 $x_1\oplus x_2\oplus x_3$ 之链图
小圈表示“非”

四、例

例 1　函数 $x_1\oplus x_2\oplus x_3$ 在基 $\Omega=\{\oplus,\cdot,-\}$ 上由链 $\beta=(\beta_1,\beta_2,\cdots,\beta_6)$ 实现(其链图如图 1.1)，其中：

$\beta_1=x_1,\beta_2=x_2,\beta_3=x_3$，

$$\beta_4=(\oplus;\beta_1,\beta_2),$$
$$\beta_5=(-;\beta_3)$$
$$\beta_6=(\oplus;\beta_4,\beta_5)$$

实际上，$\tilde{\beta}_4=\beta_1\oplus\beta_2=x_1\oplus x_2$

$$\tilde{\beta}_5=\bar{x}_3,\ \tilde{\beta}_6=\beta_4\oplus\beta_5=x_1\oplus x_2\oplus\bar{x}_3$$

例 2　$c_j=c_{j-1}\cdot x_j+c_{j-1}\cdot y_j+x_j\cdot y_j$

$$s_j=c_{j-1}\oplus x_j\oplus y_j$$

这是由 c_{j-1},x_j,y_j 给出的两个函数 c_j,s_j 描述一全加器。它是二进制加的组件，在基 $\Omega=\{\oplus,+,\cdot\}$上，它们的链图如图 1.2 所示。

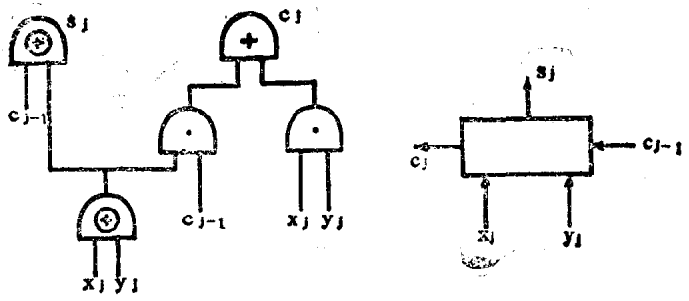


图 1.2　全加器图

五、组合机

1．什么是组合机

组合机就是由计算链的图，把其中结点代以对应逻辑元件，边代以导线所得到由逻辑元件组装而成的逻辑电路。

组合机的逻辑元件实现一个布尔函数，实质上，组合机是能进行运算操作的无存贮的机器，这是本书讨论的第一种计算模型。

2．结点的扇出与链的扇出