计算的数学理论

(上 册)

ZOHAR MANNA 著

王 冬 生 译

长沙铁道学院科技情报室

一九七九年九月

前　言

什么叫计称的数学理论？我相信没有两个计称机科学工作者用恰好相同的方法来定义这种概念。根据John McCarthy的开创性论文"计称的数学理论基础"[*]，我认为这种理论是企图把我们对计称的理解形式化，特别是要使检验计称机程序（熟知的调试技术）的艺术变成一种科学。

我打称在本书中同时处理实践和理论的问题，为了使得这本书是自给的，我把可计称理论和数理逻辑的一些经过选择的概念包括在书中。

我的目的是把这些论题介绍给广大的读者：大学高、低年级的学生、一年级研究生以及有兴趣的程序员。我故意省略了冗长的细节，而对某些结果只叙述不证明，用节省下来的空间补充一些例子。我的意图是：对属于这个领域的方法和结果给出一个全貌，而不是一个"定义——定理——证明"式样的标准数学课本。因此，我请求那些喜欢从理论上弄清楚全部形式的读者见谅。

近几年来，发表了许多关于计称的数学理论方面的论文，而且这个领域的活跃程度每年都在增加。因此，在本书的写作中，我面临的主要问题之一就是决定在一本导论性的课本中应当收入什么内容。我使用两条取舍标准。第一，只包括我感到对每个计称机科学工作者必须知道的那些材料，这主要根据我近几年来在斯坦福（Stanford）大学和威茨曼（Weizmann）学院讲授计称的数学理论这个课程的经验；第二，因为本书是导论性质的，我舍去了那些我期望它在不久的将来会发生重大变化和改革的课题，少数几个这样的课题是：自动程序的归纳，部分函数逻辑，对大

[*] In P. Braffort and D. Hirschberg (eds), "Computer Programming and Formal Systems," North-Holland Publishing Company, Amsterdam 1962.

型计算机程序的检验技术（如编译程序）以及并行程序。

大多数章是自给的。除了要理解 2—1.6 节的内容，必须先掌握 1—5 节的内容外，第一章和第二章都不要求任何先决条件。对这个领域的实践方面感兴趣的读者将发现只有 2—1.1 和 2—1.2 节的内容对第三章和第五章是予先要求的，这两章包含着要适用的技术。第四章在本质上更加理论化，为了理解它要求先掌握 1—5 节和 2—1 节的内容。

我在每一章都附了文献评论和参考文献目录。这里的取材原则是包括那些书所涉及的每个题目最有关的文献，而不是开列出现有出版物的清单。每章之后附有 20 到 30 道习题（关表示难题），本人认为这些问题是课文的不可缺少的部分，并且大力鼓励读者至少尝试着去解其中的一部分。

许多人直接或间接地对本书的完成作了贡献。首先感谢 David Cooper, Robert Floyd 和 John McCarthy，他们把我引导到这个领域。还要感谢我的同事，他们对不同的手稿样本提出了最有帮助的评论和建议。在这方面我对 Peter Andrews, Edward Ashcroft, Jean-Marie Cadiou, Ashok Chandra, Nissim Fsances, Shmuel Katz, Lookwood Morris, Stephen Ness, David Plaisted, Amir Pnueli, Adi Shamir, Mark Smith 和 Jean Vuillemin 表示最大的感激，对 John McCarthy 和 Lester Earnest 表示特别的感谢，因为当我写这本书的时候，他们始终不倦地鼓励我。最后感谢 Phyllis Winkler，他非常耐心地打印了许多份手稿的样本。

ZOHAR MANNA

目　　录

第一章 可计称性

引言

在近几十年内，对定义一种一般的、足以计称每一个"可计称"函数的计称装置，已取得了实质性的成效。在1936年，Turing曾经建议使用一种被认为是最一般的计称装置（因此称为Turing机）。后来，Church提出一个假说[Church命题(1936)]：任何可计称函数都能用Turing机计称。因为可计称函数的概念在数学上是不精确的，所以，我们提本不能希望形式地去证明Church的命题。然而十分明显，由Turing机的定义，凡能通过Turing机描述的计称都可以机械地实现；反过来，在现代数字计称机上能够实现的任何计称，都可用Turing机予以描述。如之现已证明了后来提出的所有其它的一般的计称装置[如Church(1936)，Kleene(1936)和Post(1936)提出的计称装置]† 都和Turing机有相同的计称能力，这就加强了我们对Church命题的信心。

我们在本章讨论几种机口，如Post机，具有下推存贮口的有穷机，它们都和Turing机有相同的计称功能。我们强调指出，实际上存在三种不同的方法去看待Turing机：(1)当作接收口看待（接受一个递归集合或递归可枚举集合）；(2)当作发生口看待（计称一个全递归函数或部分递归函数）；及(3)当作一个称法看待（解或部分解一类是/否问题）。

我们在本章的绝大部分篇幅来讨论Turing机（最一般可能的计称装置）的种类；但是，为了比较起见，我们在第一节讨论"最简可能的"计称装置，即有穷自动机。

---

† 这三篇论文都出现在DAVIS(1965)的选集中。

1—1 有穷自动机

设Σ是任何有限的符号集合，我们把该集合叫做一个字母表；这些符号叫做字母表中的字母。Σ上的一个字（串）是由Σ上的字母形成的任何有穷序列。空字（用Λ表示）是没有字母的字。Σ^*表示Σ上所有字的集合，其中包括空字Λ。于是，如果$\Sigma=\{a,b\}$，那么$\Sigma^*=\{\Lambda, a, b, aa, ab, ba, bb, aaa, \cdots\}$，$\Phi$代表空集合$\{\}$，即不含任何字的集合。*

我们用

$$UV=\{x \mid x=uv,\ u\text{在}U\text{中}，v\text{在}V\text{中}\}$$

来定义Σ^*的两个子集合U和V的乘积（连结）UV，因此，集合UV中的每一个字是通过把U中的一个字和V中的一个字连结起来而形成的。例如，若$U=\{a, ab, aab\}$，而$V=\{b, bb\}$，则集合UV是$\{ab, abb, aabb, abbb, aabbb\}$。注意$abb$是用两种方法得到的：(1) a和bb连结，(2) ab和b连结，集合VU是$\{ba, bab, baab, bba, bbab, bbaab\}$。注意$UV \neq VU$，表明乘积运算不是交换的。然而乘积运算是结合的，即对于Σ^*的任何子集合U，V和W，有

$$(UV)W=U(VW)$$

一个集合S的星闭包（用S^*表示），是由空字和所有那些把S中的字通过有限次连结运算而形成的字组成。于是，若$S=\{ab, bb\}$，则

$$S^*=\{\Lambda, ab, bb, abab, abbb, bbab, bbbb, ababab, \cdots\}$$

---

*注意、Λ，$\{\}$，$\{\Lambda\}$之间的差别。Λ是一个空字；$\{\}$是字的一个集合（不包含任何字的集合）；$\{\Lambda\}$是字的一个集合（该集合由唯一的字：即空字，组成）

还可采用如下定义：

$$S^* = S^0 \cup S^1 \cup S^2 \cup S^3 \cup \cdots,$$

其中 $S^0 = \{\Lambda\}$，而 $S^i = S^{i-1} S$，$i > 0$

现在我们将讨论 Σ 上的字的一类特殊的集合，称为正则集合。Σ 上的正则集合类可递归地定义如下：

1. Σ 上字的每一个有限集合（包括中——空集）是一个正则集合。

2. 若 U 和 V 是正则集合，则它们的联合 $U \cup V$ 和乘积 UV 同样也是正则集合。

3. 如果 S 是一个正则集合，那末它的闭包 S^* 也是正则集合。

这就是意味着：一个集合是正则的，当且仅当这个集合可以经过有限次应用 1 到 3 而得到。换句话说，Σ 上的正则集合类是最小的一类集合，它包含（Σ 上）字的所有有限集合并在联合，乘积和星运标下是封闭的。

令 $\Sigma = \{a, b\}$。作为例子，我们将说明 Σ 上所有那些或者含有两个连续的 a 或者含有两个连续的 b 的字的集合，以及 Σ 上所有那些含有偶数个 a 和偶数个 b 的字的集合都是 Σ 上的正则集合。另一方面，我们将指出，集合 $\{a^n b^n / n \geq 0\}$［即所有那些由 n 个 a 后面跟着 n 个 b 所组成的全部字（任何 $n \geq 0$）的集合］不是 Σ 上的正则集合，同样地，集合 $\{a^n b a^n / n \geq 0\}$ 和 $\{a^{n^2} / n \geq 0\}$ 都不是 Σ 上的正则集合。*

在本节，我们叙述表示正则集合的三个不同的方法。Σ^* 的一个子集合是正则的，当且仅当：(1) 它可以用某个正则表达式表示，(2) 它被某个有穷自动机接受，或 (3) 它被某个转换图接

*注意字（例如 $bbaba$），字的集合（例如 $\{a^n b^n / n \geq 0\}$ 和字的集合类（例如 $\{a^n / n \geq 0\}$，$\{a^n b^n / n \geq 0\}$，$\{a^n b^n a^n / n \geq 0\}$，$\{a^n b^n a^n b^n / n \geq 0\}$ …）之间的区别。

要。最后我们証明存在一种确定Σ上的两个已知正则集（用上述形式的任何一种表达）是否相等的办法，这是有关正则集合的最主要的结果之一。

1.1 正则表达式

首先我们考虑用正则表达式表示正则集合的方法。这种表示法特别有趣，因为它允许对正则集引进代数的处理办法。Σ上的正则表达式类可递归地定义如下：

1. Λ和Φ是Σ上的正则表达式。
2. 每一个字母$\sigma\in\Sigma$是Σ上的一个正则表达式。
3. 如R_1和R_2是Σ上的正则表达式，那么，(R_1+R_2)，$(R_1\cdot R_2)$和$(R_1)^*$也是Σ上的正则表达式。

例如，若$\Sigma=\{a,b\}$，则$(((a+(b\cdot a)))^*\cdot a)$是$\Sigma$上的一个正则表达式。

Σ上的每一正则表达式R描述Σ上的一个字集合$\tilde{R}$，即：$\tilde{R}\subseteq\Sigma^*$，我们如下递归地定义$\tilde{R}$：

1. 若$R=\Lambda$，则$\tilde{R}=\{\Lambda\}$，即由空字组成的集合；

   如果$R=\Phi$，那末$\tilde{R}=\Phi$，这就是空集合。
2. 如果$R=\sigma$，那末$\tilde{R}=\{\sigma\}$，即由字母组成的集合。
3. 设R_1和R_2是Σ上分别描述字集合$\tilde{R_1}$和$\tilde{R_2}$的正则表达式。

   如果$R=(R_1+R_2)$，那末$\tilde{R}=\tilde{R_1}\cup\tilde{R_2}=\{x\mid x\in\tilde{R_1}$或$x\in\tilde{R_2}\}$即两集合之并。

   如果$R=(R_1\cdot R_2)$，那末$\tilde{R}=\tilde{R_1}\tilde{R_2}=\{xy\mid x\in\tilde{R_1}$而$y\in\tilde{R_2}\}$即两集合之乘积。

   如果$R=(R_1)^*$，那末$\tilde{R}=\tilde{R_1}^*=\{\Lambda\}\cup\{x\mid x$是将$\tilde{R_1}$中的字经过有限次连结运算得到的$\}$，即集合$\tilde{R_1}$的闭包。

在不致引起混同时，正则表达式中的圆括号可以省略。为了恢复被省去的括号，有几条优先规则：'*'比'·'或'+'更优

先，即'*'总是附加到最小可能的范围（Λ，Φ，一个字母，或用括号括起来的表达式），同时'·'优先于'+'。'·'总被省略掉。例如，$a+ba^*$就是$(a+(b\cdot(a)^*))$，而$(a+ba)^*a$就是$(((a+(b\cdot a)))^*\cdot a)$。存在两个以上连续的'·'时，括号总是向左边结合，对加法也是如此。例如，aba代表$((a\cdot b)\cdot a)$，而$a^*(aa+bb)^*b$代表$((a)^*\cdot(((a\cdot a)+(b\cdot b))]^*)\cdot b)$。

例1—1 考虑下述正则表达式，其中$\Sigma=\{a,b\}$。

R	$\tilde{R}$
ba^*	Σ上以一个b开头后面只跟上一些a的全部字
$a^*ba^*ba^*$	Σ上所有那些恰好含有两个b的字
$(a+b)^*$	Σ上的所有字
$(a+b)^*(aa+bb)(a+b)^*$	Σ上所有包含两个连续的a或两个连续的b的字
$[aa+bb+(ab+ba)(aa+bb)^*(ab+ba)]^*$	Σ上所有包含偶数个a和偶数个b的字
$(b+abb)^*$	Σ上所有这样的字：其中每一个a后面至少立即跟上两个b。

根据正则表达式的定义，直接有：一个集合在Σ上是正则的，

当且仅当它们能用Σ上的正则表达式表示。注意，一个正则集合可用多个正则表达式描述。例如，$\Sigma=\{a,b\}$上所有使a和b交替出现（以b开头和结尾）的字集合既可用表达式$b(ab)^*$，也可用表达式$(ba)^*b$来描述。Σ上的两个正则表达式R_1和R_2称为是等价的（记号：$R_1=R_2$）当且仅当$\tilde{R_1}=\tilde{R_2}$；于是，$b(ab)^*$和$(ba)^*b$是等价的。在本节的末尾我们将描述一个确定两个已知的正则表达式是否等价的标法。在某种情况下，两个正则表达式的等价性可运用已知的恒等式来说明。下面开列了一些比较有意义的恒等式。为简单起见，我们令$R_1=R_2$代表$\tilde{R_1}=\tilde{R_2}$，又$W\in R$代表$W\in\tilde{R}$

对Σ上的任何正则表达式R，S和T有

1. $R+S=S+R$，$R+\phi=\phi+R$，$R+R=R$

   $(R+S)+T=R+(S+T)$

2. $R\Lambda=\Lambda R=R$，$R\phi=\phi R=\phi$，$(RS)T=R(ST)$

   注意一般来说$RS\neq SR$

3. $R(S+T)=RS+RT$，$(S+T)R=SR+TR$

4. $R^*=R^*R^*=(R^*)^*=(\Lambda+R)^*$，$\phi^*=\Lambda^*=\Lambda$

5. $R^*=\Lambda+R+R^2+\cdots+R^K+R^{K+1}R^*\ (K\geq 0)$

   特殊情况：$R^*=\Lambda+RR^*$

6. $(R+S)^*=(R^*+S^*)^*=(R^*S^*)^*=(R^*S)^*R^*$

   $=R^*(SR^*)^*$

   注意，一般说来$(R+S)^*\neq R^*+S^*$

7. $R^*R=RR^*$，$R(SR)^*=(RS)^*R$.

8. $(R^*S)^*=\Lambda+(R+S)^*S$，$(RS^*)^*=\Lambda+R(R+S)^*$.

9. （Arden规则）设$\Lambda\notin S$；那末

   $R=SR+T$ 当且仅当 $R=S^*T$

   $R=RS+T$ 当且仅当 $R=TS^*$

这些恒式式的部分可采用一般的技巧予以证明，这种技术有时叫

做用正反分析法证明（Proof by reparsing）. 让我们通过证明 $R(SR)^* = (RS)^*R$（恒等式 3）来说明这种技巧。设想任意字 $W \in R(SR)^*$ 也就是 $W = r_0(s_1r_1)(s_2r_2)\cdots(s_nr_n)$, $n \geq 0$, 其中每个 $r_i \in R$, $s_i \in S$.

分析（by reparsing）最后的表达式（运用连结是可结合的，这一事实），我们能够建立 $W = (r_0s_1)(r_1s_2)\cdots(r_{n-1}s_n)r_n$; 因此，$W \in (RS)^*R$. 由于 W 是任意地选择的，这就意味着 $R(SR)^* \subseteq (RS)^*R$; 同样地可证 $R(SR)^* \supseteq (RS)^*R$, 这就建立了恒等式。

证明这些恒等式的另一个普通方法是简单地运用预先已知的恒等式。例如，$R = S^*T$ 蕴涵着 $R = SR + T$（恒等式 9）. 因为 $R = S^*T \overset{(5)}{=\!=} (1+SS^*)T \overset{(3)}{=\!=} 1T + SS^*T \overset{(2)}{=\!=} T + SR \overset{(1)}{=\!=} SR + T$.

最后证明它们证明：如果 $1 \notin S$ 而 $R = SR + T$, 那末 $R = S^*T$. 首先注意到，如果 $R = SR + T$ 并且我们重复地用 $SR + T$ 代替 R, 我们得到（运用上述恒等式（2）和（3）之后）

$$R = SR + T = S^2R + (ST + T) = S^3R + (S^2T + ST + T) = \cdots$$

$$= S^{k+1}R + (S^kT + S^{k-1}T + \cdots + ST + T), \text{对于 } k \geq 0$$

首先我们证明 $R \subseteq S^*T$。令 $x \in R$ 并假定 $|x| = l$。然后我们考虑等式 $R = S^{l+1}R + (S^lT + S^{l-1}T + \cdots + ST + T)$。因为 $1 \notin R$, 所以 $S^{l+1}R$ 中的字至少应该有 $l+1$ 个符号；因此 $x \notin S^{l+1}R$. 但因为 $x \in R$, 这就推出 $x \in (S^lT + S^{l-1}T + \cdots + ST + T)$; 因此 $x \in S^*T$. 为了证明 $S^*T \subseteq R$, 我们令 $x \in S^*T$. 因此必存在 $i \geq 0$ 使得 $x \in S^iT$. 然而等式 $R = S^{i+1}R + (S^iT + S^{i-1}T + \cdots + ST + T)$ 蕴涵着 $R \supseteq S^iT$, 因此 $x \in R$.

现在我们将说明上述恒等式对证明正则表达式的等价性的用处。

例 1－2　证明下面的式子：

$(b+aa^*b)+(b+aa^*b)(a+ba^*b)^*(a+ba^*b)=a^*b(a+ba^*b)^*$

证 $(b+aa^*b)+(b+aa^*b)(a+ba^*b)^*(a+ba^*b)$

$\overset{(3)}{=}(b+aa^*b)[1+(a+ba^*b)^*(a+ba^*b)]$

$\overset{(5)}{=}(b+aa^*b)(a+ba^*b)^*$

$\overset{(3)}{=}(1+aa^*)b(a+ba^*b)^*$

$\overset{(5)}{=}a^*b(a+ba^*b)^*$

1—1.2 有穷自动机

令$\Sigma=\{\sigma_1,\sigma_2,\sigma_3,\cdots,\sigma_n\}$. Σ上的一个有穷自动机A是一个有限的方向图[注1]，图中的每一个顶点有从它引出的n个箭头，每一个箭头旁边标有不同的σ_i $(1\le i\le n)$. 存在一个标有"—"号的顶点（叫起始顶点），和标有"+"号的顶点集（可能是空集）（叫做终结顶点集）起始顶点也可以是终结顶点。顶点有时也称为状态。

设$W\in\Sigma^*$是一个字，从A的顶点i到顶点j的W路径是这样一个路径：该路径从i到j并且把沿途标号连结起来形成一个字W（注意，路径可以和同一顶点相交多次）。如果W路径从起始顶点引向最终结顶点，就说一个字$W\in\Sigma^*$是被有穷自动机A接受的。空字1被接受，当且仅当起始顶点也是终结顶点。被有穷自动机接受的字集合记作$\overline{A}$。Kleene定理（1—1.4节介绍）

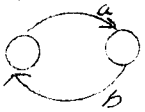
[注] 在两种情况下都只在证明从左到右的蕴涵式时要求条件1和5.

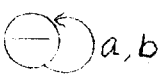
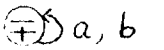
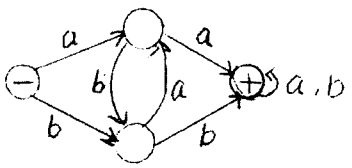
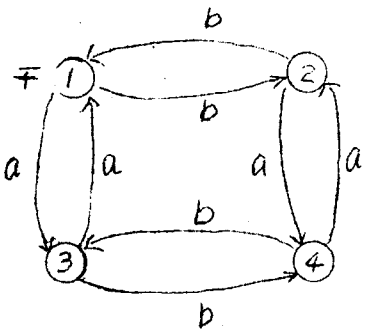
[注] 对任何字X，$|X|$表示X中的符号数；特别地，$|1|=0$。$|X|$有时也被称为X的长度。

[注] 有限方向图由（称为顶点）的有限集合和顶点的有序偶(V,V')（称为箭）组成。箭(V,V')表作 Ⓥ — Ⓥ′：顶点的有限序列$V_1,V_2,\cdots,V_k$（不必相异）称为从V_1到V_k的一个路径，如果每一有序偶(V_i,V_{i+1}) $1\le i\le k$在图上都用箭头表示。

蕴涵着：Σ上一个集合是正则的，当且仅当它被Σ上的某个有穷自动机接受。

例 1 —— 3

考虑$\Sigma=\{a,b\}$上的下列有穷自动机，我们应用形如  的箭头代表双箭头  的记法。

A	$\tilde{A}$
	ϕ
	Σ^*
	Σ上所有包含两个连续的a或两个连续的b的字
	Σ上所有那些包含偶数个a和偶数个b的字

我们怎么知道上述那些$\tilde{A}$是正确的呢？给出一个非形式证明的一个方法是：把第一个顶点i和一个字集合S_i结合起来，其中，对于S_i中的每一个字w，它在图上的路径是从起始顶点引向i顶点的。通过把每一个S_i和用单箭头引向i的所有顶点j的S_j比较，我们对S_i的选择就定下来了。所有用终结顶相应的S_i的并就是$\tilde{A}$。例如对例 1 — 3 中最后那个自动机S_i的恰

当的集合是：

S_1：所有具有偶数个 a 和偶数个 b 的字。

S_2：所有具有偶数个 a 和奇数个 b 的字。

S_3：所有具有奇数个 a 和偶数个 b 的字。

S_4：所有具有奇数个 a 和奇数个 b 的字。

我们可以用两个正则集合能被某个有穷自动机接受的事实去证明某些集合不是正则的。作为例子，我们来证明 $\{a^n b^n \mid n \geq 0\}$ 不是一个正则的集合。假定它是正则集合，那末必须存在一个有穷自动机 A 使得 $T = \{a^n b^n \mid n \geq 0\}$。我们假定 A 有 N 个状态，$N > 0$，考虑字 $a^N b^N$。因为该字被 A 接受而其长度大于 A 的顶点数，所以在 A 中至少存在 a 的一个循环，或至少存在 b 的一个循环，才能使 A 能够接受字 $a^N b^N$。假设它是长度为 k（$k>0$）的 a 的循环。而因为我们可以对任何 $i \geq 0$ 经过 i 次循环，所以 A 也应该接受所有形如 $a^{N+ik} b^N$（$i \geq 0$）的字，这就与事实 $T = \{a^n b^n \mid n \geq 0\}$ 矛盾。

1—1.3 转换图

现在我们将介绍有穷自动机概念的一个推广（称为转换图），用它来表示正则集合是非常方便的。而且我们要表明：虽然有穷自动机类是转换图类的真子类，但是，每个被转换图接受的正则集合也被某个有穷自动机接受。

Σ 上的转换图 T 是一个有限方向图，其中每个箭头用某个字 $w \in \Sigma^*$ 标记（也可能标上空字 Λ）。至少存在一个标有"—"号的顶点（这样的顶点称为起始顶点）存在一个（可能为空）用"+"标记的顶点（称为终结顶点）的集合。一个顶点可以既是起始顶点又是终结顶点。

对于一个字 $w \in \Sigma^*$，从 T 的顶点 x 到顶点 y 的 w 路径是从 x 到 y 的这样一个路径：沿着这个路径连结箭头上的标志，形成字 w

（略去空字 Λ，如果有的话）如果存在从起始顶点到终结顶点的 W 路径，就说字 $W\in\Sigma^*$ 被转换图 T 接受。如果存在 T 中的一个顶点它既是起始顶点又是终结顶点 Λ 路径从起始顶点引向终结顶点，那么，空字 Λ 被 T 接受。被转换图 T 接受的字的集合用 $\mathcal{T}$ 表示。Kleene 定理（在 1－1.4 介绍）蕴涵着：一个集合在 Σ 上是正则的，当且仅当它被 Σ 上的某个转换图所接受。

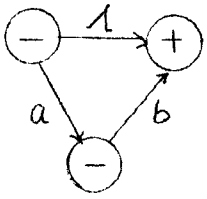
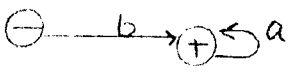
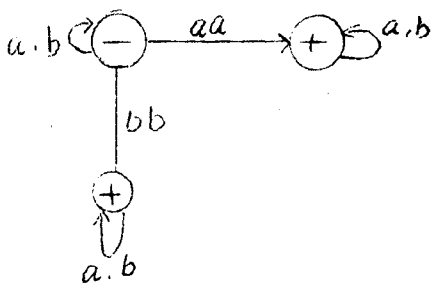
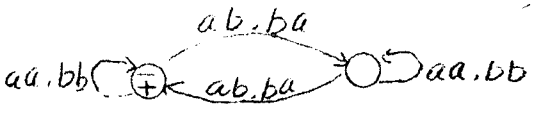
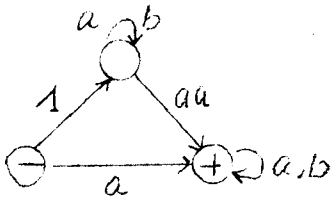
注意，每一个有穷自动机是一个转换图，但反之不然。两者之间的基础差别之一是有穷自动机在对于每一个字 $W\in\Sigma$ 和顶点 v 存在唯一的从 v 出发的 W 路径这个意义上是确定的。另一方面，转换图是不确定的，因为它可以有一个以上的从 v 出发的 W 路径（或根本没有）。

例 1－4

考虑 $\Sigma=\{a,b\}$ 上的下列转换图

---

† 设 $\sigma\in\Sigma$。σ 的一个循环是一个第一个顶点和最后顶点相同的闭路径并使得所有的箭头上都用 σ 标记。

T	$\tilde{T}$
	$\emptyset$
	$\{1\}$
	Σ^*
	$\{1, ab, b\}$
	Σ 上所有以一个 b 开头后面只跟一些 a 的字
	Σ 上所有包含两个连续的 a 或两个连续的 b 的字
	Σ 上所有包含偶数个 a 和偶数个 b 的字
	Σ 上所有或者以 a 开头或包含 aa 的字