淡江講座叢書㉗

自動化、言語和數碼

主講者／石輝然

淡江大學編印

淡江講座叢書㉗

自動化、言語和數碼

主講者：石 煇 然

出版者：淡江大學出版部

台北市金華街一九九巷五號

電話：（02）3419131

總經銷：鷲聲文物供應公司

台北市永康街十二號之一

郵政劃撥一四六七五

印刷者：大坤印刷企業有限公司

台北市和平西路3段302巷3弄47號

電話：（02）3015125

定價：國內新台幣180元

國外美金5元（郵資另計）

出版日期：中華民國六十九年八月

淡江講座叢書序

歷史進入知而後行的時期，凡事從知識而構成意象，從意象而生出條理，本條理而籌備計畫，按計畫而用功夫即可成爲事實。社會成爲知識的社會，世界成爲知識的世界，而經濟制度，已漸由知識經濟，取代了貨物經濟。高等學府原是知識的總滙，現在更成了生產知識、傳播知識、應用知識的中樞，這一無煙囪工廠的地位，駕乎有煙囪工廠之上。國與國間，強與弱、富與貧，事實上只是知識總和的差距。總統　蔣公，將教育與經濟、軍事三者視爲國力之源，而將教育列於經濟與軍事之上，實在是一種卓見。發展教育，提高學術的水準，實屬致國家於富強的捷徑，後來居上的良策。

在科學的群衆的時代，學術上的特色，是變化迅速，成長率高。高等學府站在時代的尖端，應當掌握這一關鍵，提高學術水準，生產知識，責無旁貸。我主持淡江學院，爲了順應這一趨勢，時時以此自勵，時時想法達成。

如何提高學術水準？我們首先想到的是遵照　國父的昭示：「恢復我一切國粹之後，還要去學歐美之所長，然後可以和歐美並駕齊驅。」因而我們以「貫通中西」作目標，以「容納衆流」、「一以貫之」，作爲我們的做法。我在五十二年發表的「淡江校風」一書中，曾藉解釋校名，說明淡江二字的三項含義，是「標舉溝通中西融鑄新學的雙重任務」、「揭示本院日進無疆繼續不斷的治學精神」和「揭示本院容納衆長淡泊寧靜的處世態度」。在是書中我曾說：「『淡水河』在鐵路公路未開通前，是主要的交通動脈，爲進出海的要道，堪作本院溝通中西文化的象徵。同時納臺灣北部溪流，自淡水入海與發揚淡江文化，加入文化洪流的情形相似。」又說：「民主世紀的處世態度，於誠實專一之

外，要有容納他人的雅量，看他人的技能和自己的技能一樣……就要學水一樣，水能溶解一切物品，像海一樣能容納衆流。」學校的一切措施，均以此爲方針，先紮下根基，再徐圖發展。

到了民國五十八年，自覺基礎漸漸鞏固，能站得住、挺得起，歐美知識爆發的程度，像幾何級數般增加，我若再踱方步，差距會愈拉愈遠。生產知識的工作，刻不容緩，於是在申請成立研究所未奉准之前，先後成立中國文學、西洋文學、數學、建築、物理、化學、區域及教育資料科學等研究室，從事於生產知識的工作，並在五十九年開始的第二期四年計畫中，把研究、教學、服務三者列爲並行的計畫，我在計畫的說明中說：「科學進步，交通發達，天下猶如一城，高等教育站在時代的尖端，所面臨的不是一市一省，須適合國情，順應世界潮流。也就是說：本院的發展不僅要與國內的大學比肩，還要與國際的大學看齊。現代的大學的任務，已不僅是傳授知識，而是『研究』、『教學』與『服務』三者並重。第一期四年計畫，雖已注意到研究，但重心仍放在教學上，爲了適應國家社會的未來的需要，對於原有計畫，已試作若干改變，期作第二期計畫的先導。」如何改變就是從研究、教學、服務三方面並重入手，在這期間曾出版不少與時代思潮頗具影響的譯著。

在民主的世紀，知識是社會共同創造的結晶，也是大家分享的資源。學術的研究，已不能自封於象牙塔裡。也就是說，應當揚棄「十年窗下無人問，一舉成名天下知」的舊方式。同行與同行間，固須切磋琢磨，不同行的人，也要相互參證，切長補短，才能取精用宏。於是又興建會文館，作招待校外學人，以利交換研究心得的需要。自六十年起，乃由國內推展到國際，由個人的交換心得邁向到國際，每年主辦一種國際性的學術會議，先後舉辦了「國際比較文學會議」、「開發中國家經濟穩定會議」、「美國研究會議」、「亞洲圖書館合作會議」、「泛太平洋區域科學會議」等，在學術風氣的倡導上，發生了很大的影響。

六十四年　蔣總統崩殂，我淡江文理學院建築中正紀念堂，以資紀念，闢了一個現代化會議廳，深切體會到　蔣總統所昭示的「大學要講授，更要研究，要與各企業和機關的研究室取得連繫，而構成各種研究的中心」「專題與常識兼顧」之重要，特創辦「淡江講座」。經常利用會議廳作學術性講演，從專題的介紹着眼，藉以增進學術探討的水準，而達到生產知識、服務社會的目的。是項講座，分爲專題講座與定期講演兩種。專題講座，每年舉辦十次。定期講演經常舉行，均聘學有專精、而負衆望的國內外知名之士爲之。爲鄭重起見，並組織講座委員會，負責講座人選及研議有關事宜，由我親任召集人。專題講座的講稿，均予以出版，藉流傳而擴大其影響。以吾人之所知者，分享於社會，共謀學術水準的提高，致力於國家的進步、人類的光明前途，達成我們這一代人的歷史任務。

張　建　邦
民國六十五年三月十日

目　錄

自動化機械（Machines　）…………………………………………… 1

自動化機械和言語體系（Automata and Languages）…… 10

言語之運算……………………………………………………… 14

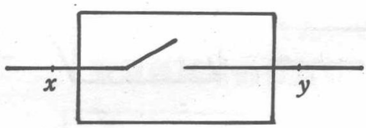
數碼…………………………………………………………… 16

數碼與言語的代數性質……………………………………… 19

本文要旨在於介紹自動化機械（如電腦等）（Machines），言語（Languages）和數碼（Codes）。關於這一方面有興趣及研究者，包括電機工程師、電腦專家、數學家、資訊專家和邏輯學者。研究論文，多發表在資訊學報、電腦專刊及數學學刊。

自動化機械 (Machines)

在定義自動化機械時需要一很重要的觀念——“狀態”（states）。所以我們先來解釋狀態這一名詞。我們考慮一個箱子（box），裡面有一電門或者是開關，如圖所示：

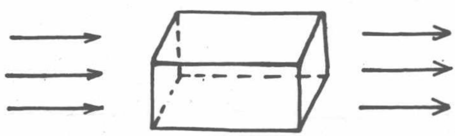


假設這一電門僅有一性質，開（on）或閉（off），其他什麼觀念都沒有。那麼當電門是“閉”時，x 和 y 兩點可相通，電流可由 x 通至 y 點。若是“開”時，自然無電流可通過。像這樣的東西，我們就稱它是兩個狀態的裝置（two-state device）。像上面所說的簡單電門可爲有用電子計算機的部份細胞，諸如可做加法之計算機。

現在假如我們有兩個單純電門，以某種方法連結在一起時，由於一個開－閉和另一個開－閉，便有四種不同的花樣，因此我們可以視由此兩電門所成者爲“四個狀態”之裝置。由此類推，n 個電門可構成 2^n 之狀態。

上面所說的是狀態的一例。另外如電視機有一選擇電台的裝置。歐美一般的電視可有十二個不同電台供自由選擇視、聽。這也是一種十二狀態的裝置。我們再來舉一個更簡單的狀態的例子：時鐘為一十二狀態的裝置，應該是極易了解之事。

有了狀態之觀念，現在我們可以來定義自動化機械或簡稱為機械。首先讓我們來問一個問題，“計算機（Computer）是什麼？”，為了回答這一問題，讓我們來假設一個“黑箱子（black box）”和幾個箭頭在black box的左邊和右邊（如下圖），這個box能替我們做一些我們所要求它做的事情，所以一定有某些方法，使人和box可以交談。其意即人可以給box一些資料，而box有方法可以還給我們工作後的結果。



左邊的箭頭是我們給box的輸入系統，而右邊的箭頭為輸出系統。當這樣的box，收到我們給它的資料後，每次都給我們正確的我們所要它做的結果時，這box便是所謂的計算機（Computer）了。

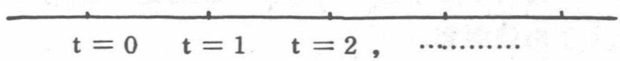
現在假設這個black box，每次給它資料後，替我們工作，並且給予我們所要的正確的結果，那麼我們會開始問一個問題，到底這個box裡面有什麼東西？為什麼它會替我們做我們所要它做的事情呢？裡面到底情況是如何而它為什麼會替我們工作？就因為它能替我們做那麼偉大的工作，所以我們稱這類black box為自動化機械M。自動化機械的進一步的構造和功能，現在

讓我們來做進一步的探討。

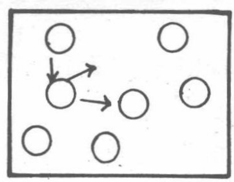
首先要知道的是自動化機械（簡稱機械）對於時間的觀念是 discrete ，它是利用電流來運轉，而電流爲連續的，故以特種裝置將連續的東西轉變爲 discrete 。所以我們可假設時間 t 爲

$$t = 0 , 1 , 2 , \cdots\cdots\cdots$$

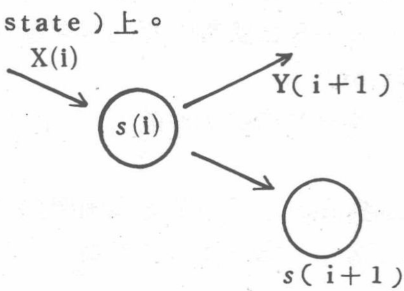
而機械的動作僅以 t 爲 0 ，1 ，2 ，……等等時間時做動作。在時間 t 爲 n 及 $n+1$ 之時間內，則視爲完全停止狀態，任何事情都沒有發生。



其次，在 box 內有有限個不同的狀態（ states ），所謂不同，其意即兩個狀態（ states）的行爲，作用不完全相同，因爲狀態（ states）爲有限，我們可以設其集合爲 $S = \{ s_1 , s_2 , s_3 , \cdots\cdots , s_n \}$ 整個機械M的行爲可介紹如下：



在每一個時間（時間爲 discrete ），機械一定在某一個而且僅有一個狀態（ state ）上。



因此，我們可以用 i 代表其時間，$s(i)$ 代表在時間 i 時之狀態（state），所以 $s(i+1)$ 代表在 $i+1$ 時間時之狀態（state）。如果在每一時間內，我們給機械資料（input），我們可以 $X(i)$ 代表在 i 時間時所給的資料，同理在 i 時間時機械給我們的資料假設爲 $Y(i+1)$。上圖所表示的意思是在 i 時間時之state爲 $s(i)$，在 i 時間之input爲 $X(i)$。同樣道理，在 $i+1$ 時間時之state爲 $s(i+1)$，而output 爲 $Y(i+1)$。由此可見機械在時間變動時，由一個狀態（state）跳動到另一狀態，我們一連串地給input而機械不斷地給我們output。我們了解了這一個事實，那就是一個機械固定下來後，便產生了兩個函數，δ 及 λ：

$$\delta:(s(i),X(i))\rightarrow s(i+1)$$

$$\lambda:(s(i),X(i))\rightarrow Y(i+1)$$

輸入的input是取自一個叫做input alphabet 的集合X，而輸出的output 則取自output alphabet的集合Y，下面是機械的一個正式定義。

定義：自動化機械（Machine）是由五項東西 $M=(S,X,Y,\delta,\lambda)$ 所構成，其中

(1) S 爲一有限狀態（states）之集合；

(2) X 爲一有限輸入之集合（input alphabet）；

(3) Y 爲一有限輸出之集合（output alphabet）；

(4) $\delta: S\times X\rightarrow S$ 爲狀態之變化或跳動函數；（state transition）

(5) $\lambda: S\times X\rightarrow Y$ 爲輸出函數（output function）。

在此定義中，集合S，X，Y均爲有限集合，其有限與否，甚爲重要。當S，X，Y這三集合之計數不大時，我們可將狀態

之變化函數及輸出函數，用表或圖形來表示，讓我們先舉一例來幫助了解。

例一：本例爲一由二狀態（2 - state）而成之機械（machine），其中 $S=\{s_0, s_1\}$，$X=\{00, 01, 10, 11\}$，$Y=\{0, 1\}$

δ	00	01	10	11
s_0	s_0	s_0	s_0	s_1
s_1	s_0	s_1	s_1	s_1

λ	00	01	10	11
s_0	0	1	1	0
s_1	1	0	0	1

上表之意爲

$\delta(s_0, 00)=s_0$　　$\delta(s_1, 00)=s_0$

$\delta(s_0, 01)=s_0$　　$\delta(s_1, 01)=s_1$

$\delta(s_0, 10)=s_0$　　$\delta(s_1, 10)=s_1$

$\delta(s_0, 11)=s_1$　　$\delta(s_1, 11)=s_1$

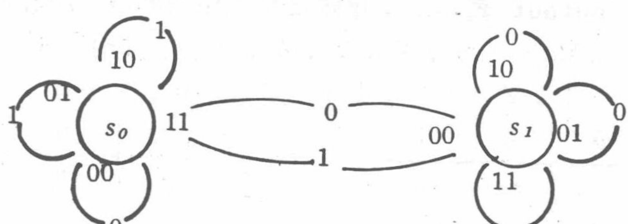
$\lambda(s_0, 00)=0$　　$\lambda(s_1, 00)=1$

$\lambda(s_0, 01)=1$　　$\lambda(s_1, 01)=0$

$\lambda(s_0, 10)=1$　　$\lambda(s_1, 10)=0$

$\lambda(s_0, 11)=0$　　$\lambda(s_1, 11)=1$

其圖形之表示法如下：



我們現在再舉幾個例子，以介紹機械如何計算或如何工作：

例二：（a Memory machine） $M_1 = (S, X, Y, \delta, \lambda)$，其中 $S = \{s_0, s_1\}$，$X = \{0, 1\}$，$Y = \{0, 1\}$，且

δ	0	1
s_0	s_0	s_1
s_1	s_0	s_1

λ	0	1
s_0	0	0
s_1	1	1

由下面的計算可看出，機械 M_1 確實可將資料記憶一單位時間。

	1	0	1	1	1	0		← inputs
s_0	s_1	s_0	s_1	s_1	s_1	s_0		← states
		*	1	0	1	1	1	0 ←outputs

我們亦可設計一類似的機械，以便將資料記憶一更長的時間，但欲達此目的，則需要更多的狀態（states）。

例三：（a parity checking machine）

假如我們有一個數列，而欲將其輸入於機械裡，（此數列係由 0 和 1 所構成），當機械獲得此數列後，我們要它告訴我們所輸入之數列裡有偶數個 1 或奇數個 1，若此數列有偶數個 1 時，機械以 0 爲 output 表示之，否則以 1 爲 output 表示之；像做這樣工作的機械可設計如下：

令 $M_2 = (S, X, Y, \delta, \lambda)$，其中 $S = \{s_0, s_1\}$，$X = \{0, 1\}$，$Y = \{0, 1\}$ 且

δ	0	1
s_0	s_0	s_1
s_1	s_1	s_0

λ	0	1
s_0	0	1
s_1	1	0

現在我們用設計出來的機械來看一個實例。假如輸入數列為1 1 0 1 1 1 0 1，

	1	1	0	1	1	1	0	1		← input
s_0	s_1	s_0	s_0	s_1	s_0	s_0	s_0	s_1		
		1	0	0	1	0	0	0	1	←output

當輸出為0時，表示到此刻輸入了偶數個1，而當輸出為1時，則表示到此刻輸入了奇數個1。

最後讓我們來設計一個做加法之機械，並且查看它如何做二個正整數的加法。

比如說要加二數，其一為25，另一數為19，我們首先要做的工作是將25和19改變為二進位。那就是

25＝1 1 0 0 1　　　　19＝1 0 0 1 1

機械要將這兩數相加起來時，也是一位一位地加，並且也是從個位數開始加起。

兩個二進位數的兩個同一數位相加的情形為

0＋0＝0，　0＋1＝1，　1＋0＝1，　1＋1＝11

這時我們只有兩個不同的情形發生，一為進位，另一為不進位，因此，若利用機械來做加法，則只需要兩種不同的狀態（states），一個狀態做不進位的工作，另一個狀態做進位的工作。

由上可知，能做加法的自動化機械，其實是本文中的第一個例子之機械，若利用例1中的圖形表示法來觀查其運算方法極其方便。狀態 s_0 做不進位的工作，而狀態 s_1 做進位的工作。機械開始工作時的狀態是 s_0，輸入的alphabet 為X＝{00，01，10，11}。

因　　　25＝1 1 0 0 1

　　　　19＝1 0 0 1 1

而機械是一位一位相加，故先做1＋1，而後0＋1，0＋0，

1＋0，1＋1，因此輸入的數列為

11，10，00，01，11 →

當輸入資料已輸入完畢時，所得的結果可能尚未完全輸出，若沒有輸入而機械仍在工作時，我們假設輸入為00，故無輸入資料時，我們假定輸入為00，00，00，………，這個數列不斷地進行。

下表即為此機械運算的過程：

$t=$	0	1	2	3	4	5	6	7
25	1	0	0	1	1			
19	1	1		0	1			
input	11	01	00	10	11	00	00	00 …
state	s_0	s_1	s_1	s_0	s_0	s_1	s_0	
output		0	0	1	1	0	1	

$$\begin{array}{r} 25 \\ +)\ 19 \\ \hline 44=101100 \end{array}$$

故此答案即為正確的答案。

凡是人類所能製造出來的機械，都是由有限個零件所構成，所以任何人類造出來的機械都是有限狀態的機械。因此實際之自動化機械都是有限狀態的機械。在市面上運用的大機械，如電子計算機等大約有 $2^{1,000,000}$ 個狀態（states）。

假如一機械所收到的輸入是一constant（固定值），（例如一機械在運轉而我們不給它輸入任何東西），那就等於機械收到固定不變的信號。因為機械只有有限個狀態，經過相當的一段時間後，狀態的變化就成一循環圈（loop），就因為這個道理，我們有辦法可證得，任給一機械，總是存在有二個正整數 x 及 y，使得此機械無法替人類做此兩數的相乘運算，現在我們就來

證明這個事實。

假設給予一機械，而我們要此機械做二數 $x = 2^u$ 和 $y = 2^v$ 的相乘積，若把 x 和 y 轉為二進位數時，則

$$x = \underbrace{100000\cdots\cdots0}_{u} \qquad y = \underbrace{1000\cdots\cdots0}_{v}$$

x 和 y 相乘的正確答案是

$$xy = 2^u \times 2^v = 2^{u+v}$$

轉變為二進位時即為

$$xy = \underbrace{1000\cdots\cdots\cdots0}_{u+v}$$

為方便起見，可設 $u = v$，若要運用此機械計算，則其輸入應為

$$11, 00, 00, \cdots\cdots\cdots\cdots\cdots, 00 \rightarrow$$

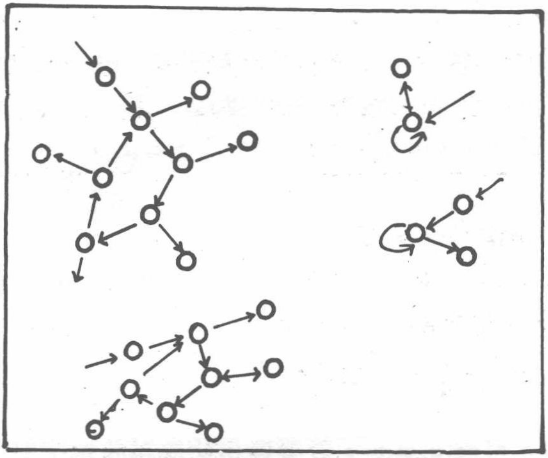
而輸出則為 $\underbrace{0, 0, 0, \cdots\cdots\cdots, 0, 1}_{u+v}$

$$\text{input} \rightarrow \underbrace{00, 00, \cdots\cdots\cdots, 00, 11, 00, 00,}_{u}\cdots\cdots$$

states 的變化

$$\text{output} \rightarrow \underbrace{0, 0, \cdots\cdots\cdots, 0, 0, \cdots\cdots\cdots\cdots\cdots\cdots 0}_{u+u}\cdots\cdots$$

當 u 足夠大時，經過 u 個00 input 之後有一11之 input，然後就變成了00 之相同 input 永遠不變。output 則為打出 $u + u$ 個 0 之後，最後打出 1。可是因為 u 是個很大的數字，機械裡面的狀態之變化早已成循環圈（如圖），而在此循環圈裡輸出均為 0，到最後要機械輸出一個 1 是不可能的。這就證明了此一機械無法做 x 和 y 的相乘積。



自動化機械和言語體系
(Automata and Languages)

在1950和1960年代，許多人在努力於從事以數學的方式來翻譯自然語言，例如：英文、俄文、中文及德文等的研究，爲了要做這個工作，首先得要了解這些語言的陳述（statement）的構造。在1959那一年，Chomsky 和其他的一些人，發展出一些講述語言的模式，在那些模式當中，有幾種則對於計算機言語的翻譯（compilers）是有用的。在這一節裡，我們將介紹四種不同的言語族和四種不同的自動化機械及其相互關係的一些學問。

首先來定義一些專門術語，設X爲一有限集合，稱它爲alphabet，每一由X中字母而成的有限序列均叫做由X所產生的字。令X* 爲所有由X所產生的字的集合，當一個字裡，連一個字母也沒有時，我們稱這個特別的字爲空字，以1代表之。任何X* 之集合都稱它爲一言語，當一個言語裡的字的個數爲有限時，我們稱它爲有限言語。設A爲一由X而成之有限言語，假如

A爲很小時，亦即A的字數不多時，我們可將A的所有的字都寫出來，那麼A這個言語就一目了然，而且A的性質大致可以了解。但是若A的集合相當大，或者A根本就是一無限集合，則我們得想一辦法來描述它。自然我們得運用有限條規定或規則來完成，或者用有限條的法則將A裡面的每一個字，一字不漏地製造出來。例如以數學表示法能定義的言語有如 $\{a^n b^n \mid n \geqq 1\}$ 或 $\{a^n b^m \mid n \geqq 1, m \geqq 1\}$

但有一些很重要並且有趣的言語，則無法如同上述的表示法表出之。

下面是一個有限的描述（或法則），可將其轉變爲一生成言語的工具：

定義：一改寫系統（Rewriting System）是由兩個集合而成，$R=(X, F)$，其中X爲字母的集合（alphabet），F爲 $X^* \times X^* = \{(x, y) \mid x \in X^*, y \in X^*\}$ 之有限子集合，每一F中之元素（p，q）稱爲改寫規則，通常以 $p \rightarrow q$ 表示之。

註：$R=(X, F)$ 爲一個有序的集合偶。

設 p, q 爲由字母集合X所產生的兩個字，我們說 p 直接生成 q，用 $p \Rightarrow q$ 代表之，當，僅當 $p=p' p_1 p''$，$q=p' q_1 p''$，其中 p'，$p'' \in X^*$，而且 $p_1 \rightarrow q_1 \in F$，我們說 p 生成 q（用 $p \Rightarrow^* q$ 代表之），若存在一有限序列的字 $p_1, p_2, p_3, \cdots\cdots, p_k \in X^*$，使得

$$p \Rightarrow p_1 \Rightarrow p_2 \Rightarrow \cdots\cdots\cdots\cdots \Rightarrow p_k \Rightarrow q$$

成立。

令 $L_g(R, A)=\{q \mid p \Rightarrow^* q, p \in A\}$，其中 $A \subseteq X^*$，我們稱 $L_g(R, A)$ 爲由（R，A）生成的言語。