

# 制式系統淺說(上)

科學月刊第十七卷第一期

中華民國 75 年 1 月

謝清俊

制式系統是山 formal system 譯來的。也有人譯作：形式系統、正規系統或是正系統。在計算機系統裡，無論軟體硬體，制式系統用得非常多。它在不知不覺中影響我們思考的方式，因為許多結構模式都是製式系統的形態。譬如，各種計算機語言的結構就是典型的制式系統。若能掌握制式系統的觀念和共通性，對於外計算機科學實事半功倍的效果。

本文是為從未接觸過制式系統的朋友所做的介紹。

本文主要內容改寫自 D.R.Hofstadter 所著的：Godel, Escher, Bach: An Eternal Golden Braid 一書中的第一、二章。為了引起你的興趣，讓我們從一則點題的遊戲——「愛你」之謎開始。

## 「愛你」之謎

這是一個字串的遊戲。字串是依順序排列的一串字。在這遊戲裡，只准用三個字（這是此系統制式的規定之一），我們姑且用：「愛」、「你」、「我」。換言之在此遊戲中，所有的字串都是以「愛」「你」「我」組成的，譬如：

「愛你」

「你我愛」

「愛你你愛你你」

「你我我你愛我你你我愛你我我你」

等都是符合制式規定的字串。

現在，我們提供你一個字串：「愛我」。這是你目前推有的唯一字串。然後，我們再提供四條規則。這些規則可以隨你選擇重複的一一使用，但是不可以自己發明規（這也是制式的要求）。給你的題目是：看你能不能變出「愛你」這個字串！這四條規則是：

規則一：若你擁有一個字串的末尾一字是「我」，你可以在此字串後加一個「你」字，成為你擁有的一個新字串。

規則二：若你擁有一個字串「愛X」，那麼你可擁有「愛XX」的新字串。此中“X”代表任何的字串。譬如：

- 由「愛我你」可得「愛我你我你」， $X = \text{「我你」}$
- 由「愛你愛」可得「愛你愛你愛」， $X = \text{「你愛」}$
- 由「愛你」可得「愛你你」， $X = \text{「你」}$

請注意：X不是此系充中的一員。它像是代數中的變數（或函數），用以表示某一字串罷了。換言之，X只是一個代表任何字串的符號，我們可藉以便利地說明規則。

規則三：若字串中含有「我我我」，則此「我我我」可為「你」取代，並成為你的新字串。譬如：

- 由「你愛我我我愛你」可得「你愛你愛你」
- 由「愛我我我我」可得「愛我你」或「愛你我」
- 由「愛我我我」可得「愛你」
- 由「我我愛我我」則什麼都得不到，因為隔中沒有連續的「我我我」。

以上這些規則都是不可逆的。例如：由「愛你」變成「愛我我我」就錯了，制式的規則就是這麼僵硬的，不可權通變達。

規則四：若字串中含有「你你」，可以把它去掉以得到新字串。譬如：

- 由「你你你」可得「你」

· 由「愛你你你我我我」可得「愛你我我我」

至此，所有的規則已介紹完畢，你可以放手一試了。最好先試試再看下文。希望你玩得愉快。

註：為了便於閱讀，所有的字串均以「」括出。

## 定理、公理、規則

嘗試過了嗎？其實，找不找得到答案並不重要。重要的是你自己親身的體驗。相信你已做出了不少的字串。對制式系統已有些體會了。

為了便於稱呼，先讓我們界定些名詞。。這此經由規則所產生的字串叫作「定理」（theorem）。當然，此定理與數學中的定理不同。數學中的定理是經過嚴謹的推理，並經證明為真的一些陳述。在這個系統中的定理，並未經過上述的證明過程，而是由系統中既有的規則所產生的。但其「正確性」一如數學中之定理。這個產生的過程非常機械化，因為所用的規則都是很刻板的。

在剛開始的時候，我們送給你一個字串「愛我」。這種不經產生就有的字串稱之為「公理」（axiom）。當然，它和數學中的公理也不一樣。但是二者皆毋需證明就可存在是互致的。在一個制式系統中，公理的個數並沒有限制，可以從無到有、到無窮多個。

規則是用來變換字串形態用的。它可以產生新字串，由新字串更可帶來新的訊息。所以稱它們為產生的規則（rules of production）或是推理的規則（rules of inference）。產生新字串的過程稱為推導（derivation）。推導和證明實似表親關係，它們分別在制式系統和數學中扮演非常類似的角色。

下面就是個推導的例子

公理                   「愛我」.....（1）  
由（1）用規則二得 「愛我我」.....（2）  
由（2）用規則二得 「愛我我我我」.....（3）  
由（3）用規則一得 「愛我我我我你」.....（4）  
由（4）用規則三得 「愛你我你」.....（5）  
由（5）用規則二得 「愛你我你你我你」...（6）  
由（6）用規則四得 「愛你我我你」.....（7）

由上可知，推導是一板一眼很制式化的。在制式系充中，如何運用既有的規則才是容許自由變化之處，但也經常是問題的所在。

名詞介紹至此，前即遊戲的問題可以用這麼一句話來說：在上述的遊戲中，「愛你」是定理嗎？

## 圈內、圈外

許多人在剛開始試「愛你」之謎的時候，都是隨意的先試著產生幾個定理。但不久，就會注意到這些定理間有些重要的共同性質。這現象是你的智慧參與工作的表現。例如，你或午已發現所有的定理都是以「愛」作第一個字。起先，你或許對這現象還有些懷疑。此時很可能為了對付這個疑問，你會暫時放下產生「愛你」的王嘗試，先去解決心中那個疑團。因此，你檢視規則，逆著定理產生的方向追蹤，直到正本清源地回到「愛我」的公理為止。至此，你相信你已證明了任何一個定理都是以「愛」為出發點的現象。於是，心中疑雲盡消，你的注意力又回到原來的問題：如何產生「愛你」上面來了。

以上的現象說明了一件非常值得注意的事：人和機器的區別。我們可以輕而易舉地寫一程式，令計算機逐一產生定理，當「愛你」出現時便停止。但是它決不會主動去觀察每個定理是什麼字起頭的。然而，即使是一個非常遲鈍的人，在一些嘗試之後，會對面臨的問題做些觀察和較深入的體認。換言之，人的智慧很不「安分」，它在做一件事的時候，經常會做到一半，就跳出了做事的圈子，在圈外看看圈內，回顧一下他做了些什麼，蒐集一些資料後，再跳回圈子裡去繼續未完的事。

在處理制式系統時，分清什麼工作是圈內的和什麼是圈外的是一件很重要的事。你可以如此試試看，在解「愛你」之謎時，準備二張紙；一張記錄圈內的工作—屬於「愛你」之謎這個系統內的，一張記錄圈外的工作—是你對此系統的觀察和心得。你將發現你會經常往返二張紙之間。這樣做的好處之一是可以更有條理地觀察和了解相關的問題，其次是使你較不易犯規。除非你是個訓練有素的研究人員，否則經常會不自覺的把制式系統以外的東西帶到制式系統中，而這是不允許的。

為了便於稱呼，我們且稱圈內工作為機械形態（ mechanical mode ），圈外的為智慧形態（ interligent mode ）。其實，我們在求知時，亦常在此二者之間往返。做習題過程就是個好例子。將圈內和圈外的事務混為一談更是出錯的泉源，比方說在雨天觀察玻璃窗，一滴水滴加上加一滴水滴還是一滴水滴，你能說這是  $1+1=1$  嗎？

## 判定的程序

上文所談的「愛你」之謎實是一種制式系統。這種制式系統是1920年代一位美國羅輯學者波斯特（ E.Ppst ）發明的。故又稱波氏產生系統（ Post production system ）。我們且稱這「愛你」之謎為「愛你我」系統，因為系統中所有的字串全由這三個字組成。

然而，根據以上的討論，並非所有僅含「愛」「你」「我」的字串均在「愛你我」系統之中。譬如，「你」「我」這些字串就不是，因為它們不是以「愛」起頭的。那麼，「愛你」是不是呢？記得嗎？我們還沒有把這個問題解決呢！說到這兒，引起了一個基本的問題，那就是：這個制式系統中的字串究竟有什麼特性？或說：我們如何判定一個字串是或不是在這個系統中？若是我們能清楚地了解系統中字串的特性，我們就有方法依據這些特性來判別一字串是否在系統之中。反之，我們若能判斷任一字串是否在系統之中，就可依判斷的條件歸納出系統中字串的特性了。所以上面二個問題實是一個問題的二種問法。

在制式系統中，判定一字串是否是定理（在系統中）的方法和過程，就叫做此系統的「判定程序」（**decision procedure**）。如果一個系統有判定程序，那麼，此系統中定理的特性就可明白地刻畫出來。若否，則定理的特性將隱含於公理和規則之中。即使是簡單如「愛你我」系統；只有一個公理和簡單的四條規則，但是其定理的特性即不易明。

判定程序也可說一個「識別過程」（**recognition process**）。通常，制式系統的問題可歸入二大類；一是產生（**generation**），其次就是識別。這二類問題解決的交會之處，就是認清定理特性的判定程序。

## 暫停

討論至此，我還是捨不得把「愛你」之謎的答案現在告訴你。真是很抱歉，且留待下回分解吧！在結束之前，讓我們再做個小小的試驗：在「愛你我」系統中，請將「咬」取代「愛」，「人」取代「你」，「狗」取代「我」，也就是把「愛你我」變成「咬人狗」，而所有字串也都變成「咬」、「人」、「狗」三個字的組合。此時，再回頭重新看看本文中的系統（「咬人

狗」系統），你是否感覺很滑稽呢？在一笑之餘試問「愛你」之謎和「咬人」之謎有什麼不同嗎？你願意花點時間想想這個問題嗎？