第二章　研究背景

第一節　公設系統

在建構一套代數系統的步驟當中，通常我們先對此代數系統操作的場(field)做一個完整的定義與描述。例如，線性代數開宗明義先定義了所謂的 Vector spaces:

Definition. A vector sapce (or linear space) consists of following:

1. a field F of scalars;
2. a set V of objects, called vectors
3. a rule (or operation), called vector addition .....

接著，對一個想要研究的系統，收集一群很明顯、合理的、不需證明就能使我們相信為真的敘述(true statement)，且這群敘述同時又滿足獨立(independence)及一致(consistency)兩個性質，我們稱這群敘述為這個系統的公設(axioms)。如果沒有任何一個公設是其他公設邏輯推論的結果稱為獨立性。任何兩個公設彼此間都不矛盾稱為一致性。

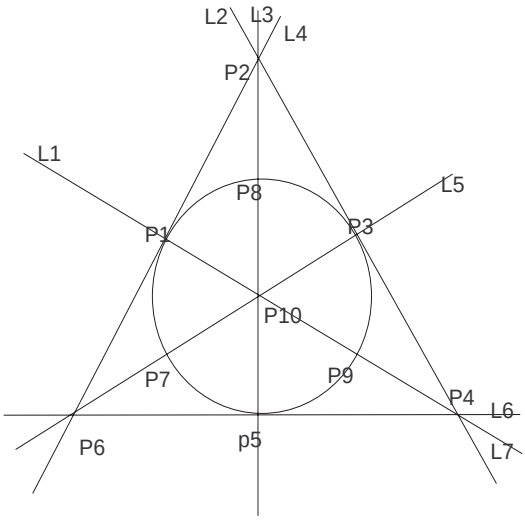

圖 1-1 公設系統的例子.

例如圖 1.滿足下列六個公設，其中 P 代表點，L 代表線(不一定是直線)，

Axiom 1. 對任何相異兩點，至少有一條線包含這兩點

Axiom 2. 對任何相異兩點，最多有一條線包含這兩點

Axiom 3. 至少有一點同時在兩條線上

Axiom 4. 至少存在一條線

Axiom 5. 每一條線一定包含三個不同的點

Axiom 6. 並非所有的點都在相同的線上

由公設再經由一系列 logic arguments 得到的真的敘述(true statement)數學上稱為定理(theorem)。例如由上述六個公設可推出：Theorem 1 任何相異兩點在唯一的線上(由 Axiom 1、Axiom 2)、Theorem 2 存在三點不在相同的線上(依 Axiom 4 存在一條線 L，Axiom 5 L 包含三個不同的點，Axiom 6 存在一點不在 L 上，最後由 Theorem 1 得證)。

我們也可以找到許多圖形滿足上列六個公設，同時 Theorem 1、2 亦成立，因為 Theorem 的證明不是依賴圖形，而是公設。

第二節　代數 (Algebra)的簡介與應用

若 E 為一群具有 t1=t2 形式的等式之集合，Σ 包含所有可出現在 E 中之常數(constant)、變數及函數(function)的集合，我們稱(Σ，E)為等式規格(equational specification)。E 常被稱為公設(axioms)，Σ 稱為 signature。例如，我們要為自然數定義一套等式規格 (Σ_1, E_1)。它的 signature Σ_1 包含一個常數 0、三個函數 s()，a()，m()。s() 稱為後繼函數 (successor)。其定義為 s(x) = x+1。運用後繼函數與常數 0, 我們可以定義出任何大小的自然數。例如 3 = s(s(s(0)))。signature 中的 a() 為加法，m()為乘法。其中 s 為一維(unary)函數、a 及 m 為二維函數。

接著表 2-1 為規範自然數的一組公設 E_1，Σ_1 包含一個常數 0、三個函數：s 為後繼函數(successor)、a 為加、m 為乘，其中 s 為一維(unary)函數、a 及 m 為二維函數。

1.	a(x , 0) = x
2.	a(x , s(y)) = s(a(x , y))
3.	m(x , 0) = 0
4.	m(x , s(y)) = a(m(x , y) , x)

表 2-1 自然數的公設

公設中除了唯一的常數 0 之外，出現的變數 x、y 為項(term)，項的定義如下：

1. 變數是項(term)
2. 常數是項(term)
3. 如 f 為一 n 維函數、t1,t2…tn 都是項(term)則 f(t1,t2,…tn)為一項

含變數的項叫 *open term*；不含變數的項叫 *closed term*。

在表 2-1 我們能將任何一項(term)取代 x，例如將第四個等式之 x 換為 s(s(0))且 y 換為 0 得到 m(s(s(0)),s(0))=a(m(s(s(0)),0),s(s(0)))。這樣的替換過程，我們可以記錄成記成

$$E_1 \square\ m(s(s(0)),s(0)) = a(m(s(s(0)),0),s(s(0))) \qquad (1)$$

也就是說 E_1 可推導出(1)式，導出(derive)的規則如下：

i. $(s=t)\in E$ implies $E \square (s=t)$

ii. **substitution**

$E\square s(x1,x2,\cdots xn)=t(x1,x2,\cdots xn)$ implies $E\square s(t1,t2,\cdots tn)=t(t1,t2,\cdots tn)$

(若 s() = t() 的等式可為公設推導出，則我們把其參數取代，公設亦成立)

iii. **forming context**

$E\square\ s=t$ implies $C[s]=C[t]$

如果 $E \square\ s=t$ 而 s 為某一項(term)子項(subterm)，則 t 可取代 s。

iv. **the equivalence properties of operator "="**

symmetry: $E \square\ s=t$ implies $E \square\ t=s$

reflexivity: $E \square\ t=t$

transitivity: $E \square\ s=t$ and $E \square\ t=u$ implies $E \square\ s=u$

v. **conditional equations**

$G \Rightarrow s=t$ (有時寫成 $\frac{G}{s=t}$)

$\Rightarrow$ 為 logical implication，G 為 E 導出的等式集合

以上所討論僅是(Σ，E)的語法(syntax):一群沒有意義的等式及規則，接下來討論(Σ，E)的語意(semantic)。一個代數 *A* 由一群元素及函數所組成：函數的運作必需具有封閉性(closure)、元素即常數與函數運作產生的項(term)，元素的集合記成 A，A 為 *A* 之 domain。例如 *A* =(N,+,・,s,0)為自然數代數，N={0,1,2,….}，+,・為普通算術的加乘，s 為後繼函數(successor)。B=({0,1},xor,and,not,0)為 Boolean 代數。

若代數 *A* 之常數、函數與Σ常數、函數都分別一一對應，則 *A* 稱為一個Σ-algebra，也稱Σ的一種解釋(interpretation)。例如我們把 *A* =(N,+,・,s,0)作如下的解釋，*A* 就是 Σ_1-algebra，因為 *A* 的運算子與 Σ_1 中的 function 有下面的一一對應關係：

a → +
m → ・
s → s
0 →0

當然我們也可以作另一種解釋，它也是 Σ_1-algebra 如下：

a → ・
m → +
s → s
0 → 0

這也是 Σ_1 的一種解釋，不過此種解釋不符合 Σ_1 公設系統，也不具實際意義。

若 *A* 為一Σ-algebra 且滿足 E 之所有等式，記成 *A*□E，則稱 *A* 為 E 之**代數**或**模型**，E 稱為 *A* 之 sound axiomatization，E 的等式含有變數 x, y, 變數可被取代為 *A* 的 domain 中任意元素。例如 *A*=(N,+,・,s,0)在上文第一種解釋下滿足 E_1 的 a(x, s(y)) = s(a(x, y))，這個等式在 *A* 被解釋為 x + s(y)=s(x+y)， ∀x,y∈N。

除了(N,+,・,s,0)為(Σ_1, E_1)模型外，尚有 *B*=({0,1},xor,and,not,0)為其模型，其解釋如下：

a------xor
m-----and
s------not
0-----0

B 不但滿足 E_1 之所有公設，另外它也滿足 s(s(x))=x，而(N,+,・,s,0)□s(s(x))=x，

因此一般而言一組規格(公設)擁有超過一個以上的模型。

若以 alg(Σ, E)表示所有解釋(Σ, E)代數的集合, 其中有一類代數由封閉項(closed term)組成非常特殊，這些 terms 都是由Σ所產生出來。而這些 terms 可以被公設 E 分成許多等價類。而且在任一個等價類中的 terms 都可以經由 E 推出它們是相等的。除此之外，不存在一種等價類是 E 推不出來的[1]。也就是說它恰好僅滿足 E 沒有其他的了，即它只滿足 E 及 E 導出的等式，我們將它記成 I(Σ，E)，稱為(Σ，E)之 initial algebra。I(Σ，E)為(Σ，E)之解釋，(Σ，E)導出的等式對應在 I(Σ，E)亦成立，現在 I(Σ，E)又只滿足 E 沒有其他的了，所以 I(Σ，E)與 (Σ，E)之關係用數學表示為：

I(Σ，E) □ t=s ⇔ (Σ，E) □ t=s ∀ closed terms t,s∈ I(Σ，E)。

其中 (Σ, E)　B 代表規格可以推出 B。(Σ, E)稱為 I(Σ, E)的 complete axiomatization。

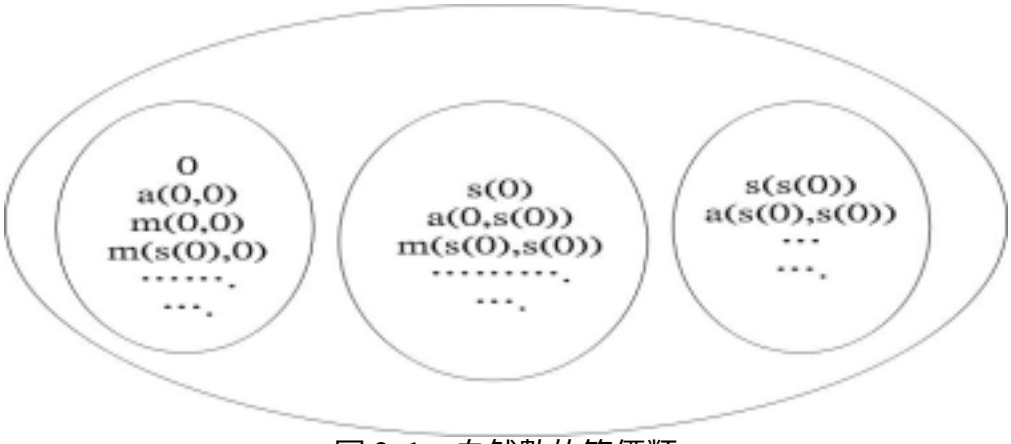

圖 2-1　自然數的等價類

例如表 2-1 其對應之 I(Σ_1，E_1)如圖 2-1，其元素就是等價類。最左邊的一類都是 0,中間都是 1,右邊為 2。每一等價類中任何兩項都可由 E_1證明它們是相等的，所以 I(Σ_1，E_1)與(N,+,・,s,0)是同構的(isomorphic)，也就是說存在一函數 f 為一對一且映成(onto)，由 I(Σ_1，E_1)至(N,+,・,s,0)滿足 f(x。y)=f(x)□f(y)，。、□分別代表 I(Σ_1，E_1)及(N,+,・,s,0)中的有對應關係的運算符號。對(Σ_1，E_1)而言可能有許多模型為 initial algebra，它們之間都是同構的，所以我們只需其中之一即可。

[1]若Σ所產生出來的 terms 可形成某種等價類是 E 推導不出來的，則(Σ，E) 就不是此實體的一個 complete axiomation。該實體稱為此(Σ，E)的一個模型(model)。

若 E 可證明兩項(term)相同是最強的等價關係，可否找到其他較弱的等價關係 R，而又能滿足：R(a1,b1),R(a2,b2)….R(an,bn)$\Rightarrow$R(f(a1,a2,…an),f(b1,b2,…bn))，f 為Σ之函數，如此之 R 稱 congruence。這樣我們就能找到更多(Σ，E)的模型。

第三節　項轉移系統（Term Rewriting System）

在這一節，我們介紹所謂的項轉移系統(Term Rewriting System,TRS)。項轉移系統可以用來檢驗一組公設是否是最少的，而且都是最基本的。一個項轉移系統(TRS)為(Σ，*R*)，Σ包含所有可出現在 *R* 中之常數(constant)、變數及函數(function)的集合，*R* 為一群像 t$\rightarrow$s 轉移規則(rewrite rule)的集合, 每一個轉移規則同時滿足下列兩個條件：

1. Var(s)$\subseteq$Var(t) ；s 若含有變數，這個變數一定在 t 中出現過。
2. t 決不是單一變數；例如 x$\rightarrow$x + a + b 左邊為單一變數，所以不是轉移規則。

若已知 u 為一項(term)、s$\rightarrow$t 為轉移規則，轉移方式是在 u 中找一子項(subterm：出現在 u 中而本身為項)，而此子項剛好配合 s，以 t 的型式取代 u 中子項。舉例如下：我們想轉移 s(s(0)) × s(s(0))，TRS 為 1.至 4.

1. 0+x$\rightarrow$x
2. s(x)+y$\rightarrow$s(x+y)
3. 0×x$\rightarrow$0
4. s(x)×y$\rightarrow$(x × y) + y

s(s(0)) × s(s(0))$\rightarrow$ (s(0) × s(s(0))) + s(s(0))　;用規則 4.

$\rightarrow$ ((0 × s(s(0))) + s(s(0))) + s(s(0))　;用規則 4.

$\rightarrow$ (0 + s(s(0))) + s(s(0))　;用規則 3.

$\rightarrow$ s(s(0)) + s(s(0))　;用規則 1.

$\rightarrow$ s(s(0) + s(s(0)))　;用規則 2.

$\rightarrow$ s(s(0 + s(s(0))))　;用規則 2.

$\rightarrow$ s(s(s(s(0))))　;用規則 1.

如果 term *s* 已沒有轉移規則可應用，*s* 稱為 norm form。對一個 TRS 我們希望可將每一個 term 轉移至唯一的 norm form，TRS 滿足下列兩條件即可將每一個 term 轉移至唯一的 norm form：

1.strong normalization：沒有無限步驟的轉移

2.confluent：t 同時經兩種不同的一連串轉移可達 t1 及 t2 時，存在 t3 使得 t1 及 t2 都可轉移至 t3。

若 t3 不存在，t1 與 t2 稱 critical pairs，換句話說沒有 critical pairs 即為 confluent。

另外 Rosen[13]已證明任何具有 orthogonal 性質的 TRS 為 confluent，orthogonal 性質為 1.每一轉移規則為 left-linear：左邊為線性函數，2.TRS 不會產生 critical pairs。

實際上一群公設就是 TRS，僅需將"="改為"$\rightarrow$"。但他們之間還是有些微差異需要注意，例如公設 0+x=x 可變成 0+x$\rightarrow$x，變成 x$\rightarrow$0 + x 就非轉移規則。

第四節　等價關係的簡介

令 A 為一集合，在 A 上定義一關係 R，滿足反身(reflexivity)、對稱(symmetry)、遞移(transitivity)三種性質，R 即為等價關係。例如 " = " 就是一種等價關係。等價關係最大的優點為可將集合 A 分割成等價類：等價類記成[a]={x|xRa, $\forall$ x$\in$A}，所有等價類聯集為 A，且任意兩個等價類交集為空集合。集合 A 經等價關係分割後，我們視等價類為 A 之元素，集合 A 的結構變成 A/R(A modulo R)。

在實際應用時，不可能只有集合、等價關係而沒有其他作用在集合中元素上的運算。當考慮等價類運算時，我們想知道是否等價類中任一元素與集合中元素運算的結果都一樣，即 a1,a2$\in$[a]，b1,b2 $\in$[b]滿足 a1+b1=a1+b2=a2+b2，我們稱這種等價關係對這種運算為 congruence（可替代的）關係。

這一節剩下部份介紹在程序代數常用到的等價關係，我們以最基本的有限程序集合為模型，介紹幾種常用到的等價關係。若 closed term 被解釋為程序(process)時，這種程序謂之有限程序(finite process)，以 *P* 表示有限程序之集合，每個程序由循序算子・、選擇算子+以及不可分割的原子行為(atomic action)a、b、c 等所組成，另加一個不可觀察的

內部行為 τ，每個程序的行為方式如 CCS[1]所描述，為了方便敘述我們定義 A 為 a、b、c…之集合且 $\tau \notin A$。接著在 *P* 上定義四種等價關係如下：

1. 只有公設規範的等價關係

   以 E 表示公設之集合，E 推論出兩程序相等為最基本的等價關係，不但具有“取代”的 congruence 關係而且在任何以 E 為公設的模型上定義另外具有 congruence 的等價關係，都不能違反 E 推論出相等的結果。換言之另外定義的等價關係僅是將 *P*/E 中某些等價類合併形成包含更少的等價類。例如在 *P* 中另外定義等價關係 R_x，先用 E 將程序 p、q 分別簡化成惟一的 norm form p’及 q’後，如果 p’=q’則 p’與 q’一定具有 p’ R_x q’關係。如果採用 E 作為 *P* 上惟一等價關係，(*P*，E)即為 initial algebra，E 為 *P* 的 complete axiomatisation。

2. **Strong bisimulation( )**

   strong bisimulation 為一在 *P* 上的等價關係記成 ，滿足下列條件：

   I． 如果 p q 且 $p \xrightarrow{a} p'$，∃一個 q' ∋ $q \xrightarrow{a} q'$且 p' q'

   II．如果 p q 且 $q \xrightarrow{a} q'$，∃一個 p' ∋ $p \xrightarrow{a} p'$且 p' q'

   上述條件中，$\forall a \in A \cup \{\tau\}$都成立。 也具有 congruence 的等價關係，在應用方面較少採用 ，因為它太強，形成最多的等價類。不過去除 τ 及 τ 的公設第三章表一之公設與 是一致的，即 p q ⇔ 表一公設□p=q。

3. **Weak bisimulation($\underline{\leftrightarrow}_\tau$)**

   首先以 $\overset{s}{\Rightarrow}$ 表示執行一連串可觀察的行為，s 中含有 τ 或一連串 τ，因 τ 是不可觀察的行為，所以都被刪除，若 s 全都由 τ 組成稱之為空行為以 $\Rightarrow$ 表示，$p \Rightarrow p'$意為 p 會自動到達 p'狀況。讓 $\underline{\leftrightarrow}_\tau$ 代表 weak bisimulation 關係，定義如下：

   I． 如果 $p \underline{\leftrightarrow}_\tau q$ 且 $p \overset{s}{\Rightarrow} p'$，∃一 q' ∋ $q \overset{s}{\Rightarrow} q'$且 $p' \underline{\leftrightarrow}_\tau q'$

   II．如果 $p \underline{\leftrightarrow}_\tau q$ 且 $q \overset{s}{\Rightarrow} q'$，∃一 p' ∋ $p \overset{s}{\Rightarrow} p'$且 $p' \underline{\leftrightarrow}_\tau q'$

weak bisimulation 主要用於觀察程序的實驗，內部行為 τ 也是觀察的一部分，如沒有內部行為，我們會把 a・0 + τ ・0 $\underline{\leftrightarrow}_{\tau}$ a・0。但是當有內部行為時，a・0 + τ ・0 $\underline{\not\leftrightarrow}_{\tau}$ a・0，因不等式的左邊只能作 a，右邊亦如此。但是當不等式的左邊可選擇內部行為，不等式右邊卻不能做出相對等的行為，故 a・0 + τ ・0 與 a・0 不為 $\underline{\leftrightarrow}_{\tau}$。$\underline{\leftrightarrow}_{\tau}$對 CCS[1]所提及有關程序間的運算，僅有"+"這個運算子沒有 congruence 關係，例如，b・0 $\underline{\leftrightarrow}_{\tau}$ τ ・b・0，但 a・0 +b・0 與 a・0 + τ ・b・0 卻不為 $\underline{\leftrightarrow}_{\tau}$，因前式執行 a 實時，後式可能先執行內部行為而到達一個只能執行 b 的狀態。使用 weak bisimulation 時公設集合必需包含下列三個有關 τ 的公設：$\alpha \in A \cup \{\tau\}$，P,Q∈*P*

i. α ・ τ ・P = α ・P

ii. P + τ ・P = τ ・P

iii. α ・(P + τ ・Q) + α ・Q = α ・(P + τ ・Q)

4. Congruent bisimulation(=)

R.Milner 希望能找到一種具有 congruence 等價關係且分割出來的每一個等價類含有最多等價的程序(process)，也就是說找一種等價關係比　弱但又改進 $\underline{\leftrightarrow}_{\tau}$ 對+沒有 congruence 的缺憾，congruent bisimulation 即為其所要的，R.Milner 把它記成=。=定義如下：首先解釋用到的符號，$\alpha \in A \cup \{\tau\}$，以 $\overset{\alpha}{\Rightarrow}$ 表示執行一連串 α 與 τ 任意組合的行為，這一連串行為中僅有一個 α、0 個或多個 τ。

I. 如果 p=q 且 $p \overset{\alpha}{\rightarrow} p'$，∃一個 $q' \ni q \overset{\alpha}{\Rightarrow} q'$且 $p' \underline{\leftrightarrow}_{\tau} q'$

II. 如果 p=q 且 $q \overset{\alpha}{\rightarrow} q'$，∃一個 $p' \ni p \overset{\alpha}{\Rightarrow} p'$且 $p' \underline{\leftrightarrow}_{\tau} q'$

R.Milner 為解決 $\underline{\leftrightarrow}_{\tau}$ 對"+"沒有 congruence 而想出的定義，$\underline{\leftrightarrow}_{\tau}$ 對+沒有 congruence 最大的原因為前導 τ，因此定義中對兩程序(process)有=關係，第一個 action(包含 τ)必需配合，往後行為有 $\underline{\leftrightarrow}_{\tau}$ 就好。=與　及 $\underline{\leftrightarrow}_{\tau}$ 關係為：若 p　q 成立，可以推出 p=q 成立、若 p=q 成立，可以推出 $p \underline{\leftrightarrow}_{\tau} q$。

第五節　局部性分析(compositional analysis)的簡介

對一個由許多程序組成之系統，直接按系統組成的方式，將所有程序的所有可能到達的狀態(reachable state)一次展開來，以驗證系統是否滿足某些性質，稱為整體性分析(global analysis)。整體性分析用於複雜系統時，常因系統的狀態空間(state space)成指數成長而失敗。因應組態爆炸(state explosion)的問題而發展出許多方法，其中以局部性分析法最有可能有效的解決組態爆炸的問題。

局部性分析法將複雜系統按其原本性質分解成許多子系統，將這些子系統組成階層式架構(hierarchy)，然後一步一步從底層開始向上將子系統組合起來，在組合的過程中，不斷的以簡化或最小化(reduction and minimization)之後的,只展現外部行為的子系統取代原子系統，以控制組態爆炸，直到整個系統被完全分析為止。局部性分析要能有效的防止組態爆炸依賴子系統內元件間只有簡單的溝通，也就是當一個元件擁有簡單的介面行為或簡單的外部行為，局部性分析可以隱藏元件的內在狀態，改以介面行為取代原元件的複雜行為。然後再進一步與別的元件作狀態展開時，自然能減少狀態空間的指數成長，達成防止組態爆炸的問題。但是在實際應用時，對於某些已架構完好的系統時，因某種原因系統內元件間的界面可能非常複雜，並不符合局部性分析的要求。局部性分析也就無法應用在此類系統。