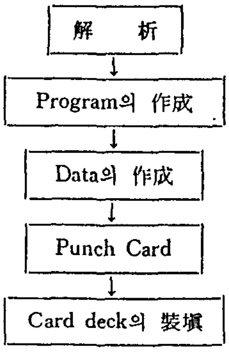
電子計算機의 使用法

崔 榮 博*

머 릿 말

電子計算機로서 計算을 할 경우 우리들은 다음과 같은 各段階의 作業을 할 필요가 있다.



이것을 다시 具體的인 作業으로 細分하면 다음과 같다.

解析
Flow-Chart의 作成
Data Sheet의 設計
結果表의 設計
Program Coding
Data Sheet 의 記入
Punch Card 와 Card deck의 作成
Test欄과 修正
本欄

여기서 이들 作業을 行하는 方法 및 注意事項을 다음 例題로서 說明하고저 한다.

"2次方程式 $AX^2+BX+C=0$를 풀기 위한 Program을 作成하고 $A=3.20$, $B=-6.30$, $C=2.86$의 값을 가질 때 X를 求하라."

1. 解 析

$D\ \ B^2\ \ 4AC$

(1) $D \geqq 0$의 경우

$$X_1=\frac{-B+\sqrt{D}}{2A} \qquad X_2=\frac{-B-\sqrt{D}}{2A}$$

(2) $D \leqq 0$의 경우

$$X_1=-\frac{B}{2A}+\frac{\sqrt{-D}}{2A}$$

$$X_2=-\frac{B}{2A}-\frac{\sqrt{-D}}{2A}$$

2. Flow Chart의 作成

電子計算機로서 演算을 할 경우 그 內容은 다음과 같이 區分된다.

(1) Data의 읽기
(2) 加減乘除
(3) 條件(大小比較로서 表示됨)에 依한 演算經路의 選擇
(4) 結果表示

여기서 演算이 어떤 順序로 行해지는가를 一連의 흐름圖 即 Flow Chart로 表示한다.

Flow Chart를 表示 하는데는 一定한 記號가 사용된다.

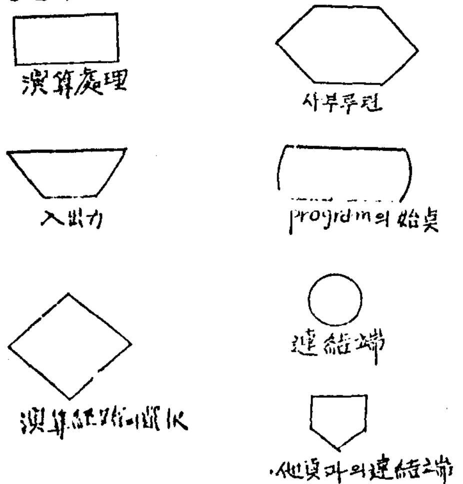


그림 1. Flow Chart의 記號

* 技術士(建設部門)·理學博士
高麗大學校 理工大敎授

Flow Chart는 Program을 作成할 때 사용될 뿐만 아니라 後日 Program을 修正한다든가 他人에게 Program의 內容을 理解시키는 경우에 必要하므로 簡單한 計算이라도 반드시 만드는 習慣을 가지게 하는 것이 所望된다. Flow Chart는 簡單한 計算에서는 처음부터 順序있게 써내려 갈 수 있으나 큰 計算인 경우는 여러 Block로 나눈 것이 보통이다. 이 경우 각 Block에 이름를 부치서 Program을 쓸 때 注記에 그 이름을 써두면 乃終에 Flow Chart와 Program을 對照하는데 便利이다. Flow Chart를 表示하는데는 一定한 記號가 사용된다. 그림—2는 그 한 例이다. 이 例에서 判別式 D의 正負에 의해서 演算經路가 나누어지는 것에 注目할 것이다.

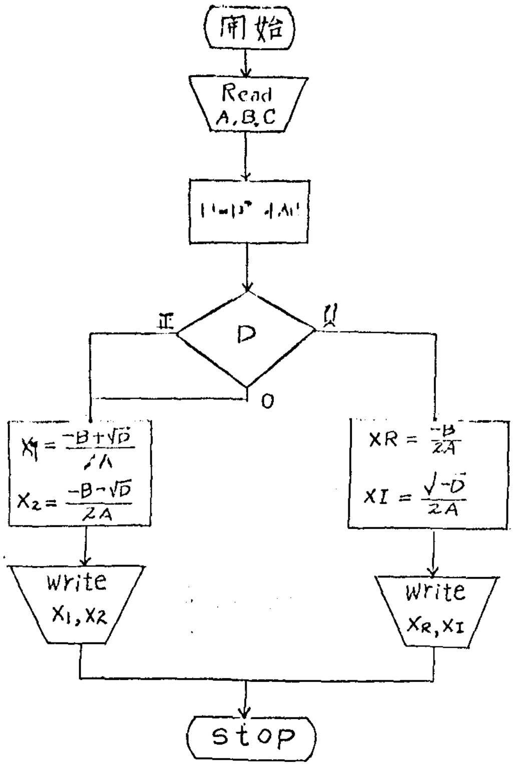


그림 2.

3. Data Sheet의 設計

Punch Card는 80개까지의 數字를 收容할 수 있는데 1매의 Punch Card에 몇개의 Data를 整理하여 Punch하는 것이 보통이다. 이 Data의 順序와 크기는 Program中에 指定된 形에 一致하지 않으면 안된다. 이 까닭에 Program을 쓰기 앞서서 Punch Card에 Punch되는 Data의 形을 設計한다. 이 設計는 將來 Data를 記入하기 위하여 사용되는 Data Sheet위에 行하여진다. 그림—3. 은 例題의 Data Sheet의 設計이다. Data Sheet의 1行이 Punch Card의 1매에 상당한다.

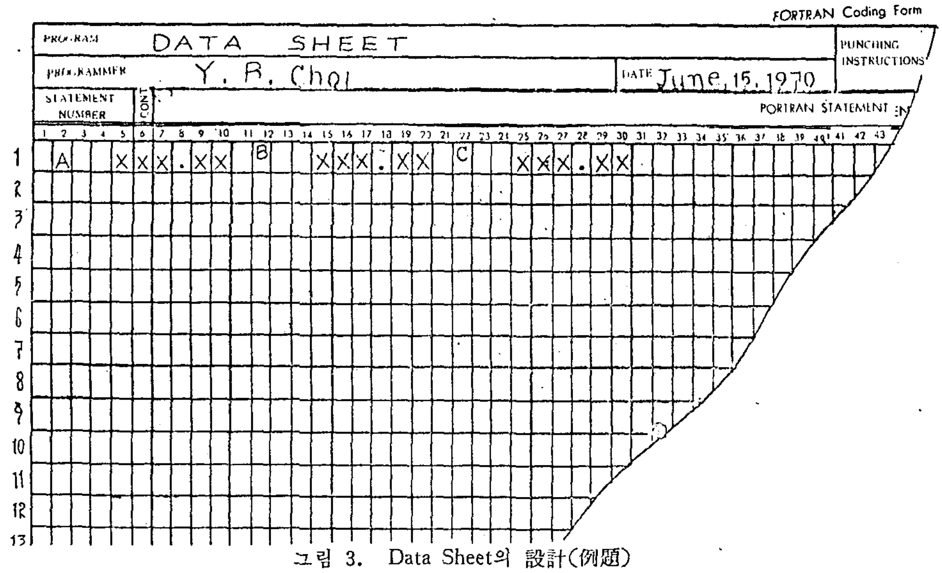


그림 3. Data Sheet의 設計(例題)

4. 結果表의 設計

Progrom을 쓰기 앞서서 高速印刷機로 적어 내는 結果表의 形을 設計한다. 1枚의 結果表用紙에 印刷되는 活字의 位置는 高速印刷機의 種類에 따라 決定되므로 그 區域나누기를 印刷한 用紙 위에서 字割當을 決定한다. 이와같은 目的으로 사용되는 用紙를 Spacing Sheet라 한다. 이 Spacing Sheet에서 1行은 133Column으로 된다.

最初의 Column은 結果表用紙의 送付를 指示하기 위하여 使用되며 印刷에는 關係가 없다. 第4行 및 第12行의 最初에 쓰인 "1"은 "結果表用紙를 改正하고 第1行을 印刷하라"라는 것을 意味한다.

"×××·××"는 演算結果의 數字를 印刷하는 位置가 表示되고 其他는 說明用으로서 印刷되는 文字이다. 이 모습은 Program을 쓸 때 "FORM-(AT)"라 하는 表現의 括弧안에 定해진 記

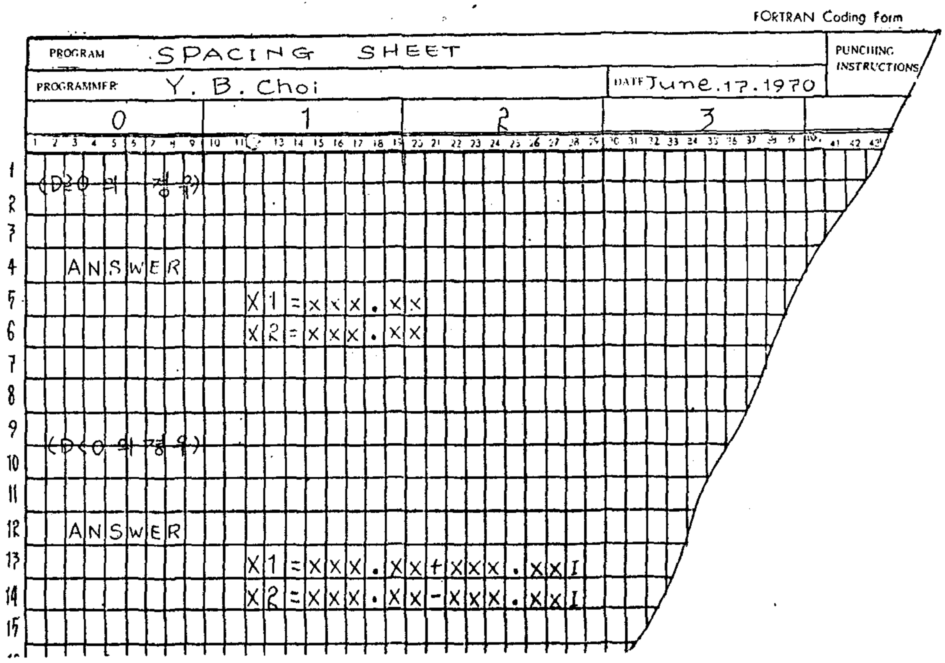


그림 4. 結果表 設計(例題)

號를 써서 나타낸다.

5. Program Coding과 Data Sheet의 記入

以上으로 準備가 끝나고 다음에 Program을 쓸 段階로 된다. Program 쓰는 것을 Program Coding이라 한다. Program의 表現은 그 內容에 따라 다음의 6種類로 나누어진다.

(1) 數學的 演算의 表現

數學的演算의 表現은 보통의 代數式과 조금도 다름이 없다. 定數, 變數 등은 加減乘除의 記號로서 組合하고 等號로서 表現한다(=表現).

(2) 入出力의 表現

Data를 읽어 結果를 써 낼때에 사용되는 表現으로 "읽어라" 또는 "써라"라는 命令과 "어떤 形으로"라 하는 指定이 組合되어 사용된다.

(READ 表現) (WRITE 表現) (FOMAT 表現)

以外에 磁氣테이프에 써 넣을 때 "끝났다"는 記號를 써 넣기 위한 命令 및 磁氣테이프의 뒤감기를 하기 위한 命令이 있다.

(END-FILE 表現) (REWIND 表現)

(3) Program의 實施順序를 指定하는 表現

Program의 表現은 特別한 指定이 없는 限, 쓰여진 順序에 따라서 실시된다. 이 項에 包含되는 表現은 다음에 行할 表現을 指示하는 것으로 單히 그 行先을 指示하는 것과 條件에 따라 그 行先을 選擇시키는 것과 反復을 行하는 것등이 있다.

(GO—TO 表現)

(條件附 GO—TO 表現)

(IF 表現)

(DO 表現)

以外에 命令 또는 指定도 하지 않고 行先을 指示하는 경우의 行先으로서 使用되는 表現 및 Program의 끝을 指定하는 表現이 이 種의 表現에 包含된다.

(CONTINUE 表現)

(END 表現)

(4) 記憶裝置를 確保하기 위한 表現

Data를 Program 속에 表示하는데 한變數名으로서 한Data를 代表시킬 경우(A,X, STRESS 등)과 變數名과 別系인 變數名을 組合해서 一連의 Data를 代表시킬 경우(變數群이라 하며 A(I),

X(I), STRESS(I) 등)가 있다. 後者의 경우 이 一連의 Data를 連續한 番地에 넣으므로 먼저 그 變數群名과 Data의 個數를 電子計算機에 알리어 수이 빈터를 確保하여 두기 위한 表現이 必要하다(DIMENSION 表現).

(5) Subprogram의 利用에 관한 表現

Program過程에서 그 Program과 別途로 쓰이진 一連의 Program을 利用한 경우가 있다. 이 別途의 Program을 Subprogram이라고 한다. Subprogram은 函數 Program과 Subroutine으로 나누어진다. 函數 Program은 Library로서 內藏되어 있는 것으로 主 Program中에서 그 이름을 쓰기만 하면 간단히 사용할 수 있다. Subroutine은 主 Program外에 別個로 쓰여진 特別한 Program으로서 主 Program에 必要할 때 이것을 呼出해서 利用할 수 있다. 이때 主 Program과 Subroutine의 兩者에 이 關聯을 指定하는 表現이 必要하다.

(CALL 表現)

(SLBROUTINE 表現)

(RETURN 表現)

(COMMON 表現)

(6) 倍精度를 指定하는 表現

取扱하는 數桁이 크고 1語로서 收容할 수 없는 경우에는 連續한 2個의 番地에 收容된다. 이 경우 이들의 數를 代表하는 變數에 對하여는 事前 倍精度인 것을 Program의 表現에 의해서 指定해 두지 않으면 안된다.

(DOUBLE-PRECISION 表現)

이외 서로 複雜한 Program도 以上 6種의 表現의 組合에 의하여 구성된다.

FORTRAN에는 上記 (　　) 안에 쓰여진 表現보다 더 많은 表現이 規定되고 있으나 普通의 土木技術計算에서 以上의 것만 알아두면 充分하다.

Program을 記入樣式에 印刷한 一定한 用紙위에 쓴다. 이 用紙를 Coding Sheet라 한다. Coding Sheet의 1行이 Punch Card 1枚에 穿孔된다.

그림—5는 앞의 例題의 Program을 Coding에 써 넣은 것이다. Sheet Coding의 記入方法은 다음과 같다.

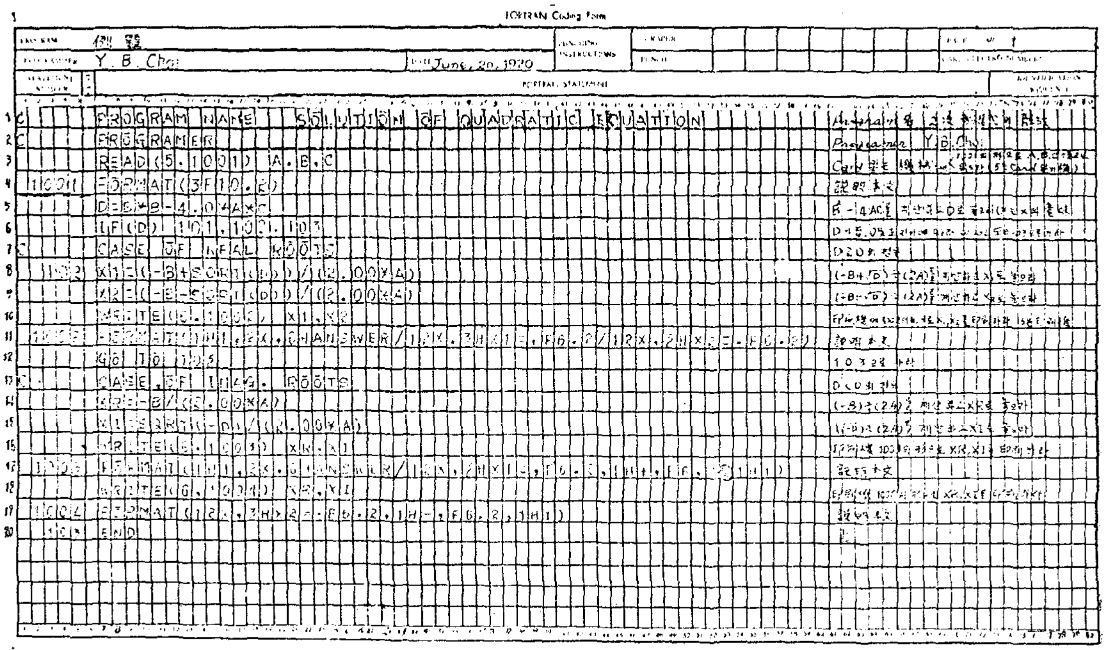


그림 5. program(例題)

(1) 1 Column에는 必要에 따라 "C"라는 文字가 記入된다. "C"에 記入된 行의 內容은 演算에는 관계하지 않는다. 이 行은 뒤에 다시 뒤돌아 읽을 경우 Program의 內容을 알기

쉽게 하기위한 說明(COMMENT)에 사용된다.

(2) READ 또는 WRITE 表現의 "5" 또는 "6"은 Control Card 로서 指定한 數와 一致시키지 않으면 안된다. 그런데 既成의 Control Card를 사용할 때는 注意를 要한다.

(3) 1001의 FORMAT 表現의 ( ) 안에는 그림-3으로서 設計 모습이 記號를 사용해서 쓰여지고 있다. F는 小數點數를 意味하고 "F 10.2"는 小數點 및 記號를 包含해서 10 Column에 收容되고 끝에서 2 Column이 小數가 되는 그와 같은 小數點數를 表示하고 있다. "F" 앞의 3은 이와 같은 小數點數가 順番으로 3回 줄을 지우고 있는 것을 意味한다. 이것에서 計算機는 처음의 10 Column을 A, 다음의 10 Column을 B, 그 다음의 10 Column을 C로서 읽어 넣는다.

(4) SQRT는 內藏되고 있는 Library 中의 平方根을 求하는 函數 Program의 이름이다. 函數 Program은 이와 같이 이름을 쓴 것만 빼낼 수 있다.

(5) 1002의 FORMAT 表現의 ( ) 안에는 그림-4의 4行에서 6行에 쓰여진 모습이 記號로서 表示되고 있다. 記號의 意味는 다음과 같다.

ㄱ) "H"—다음의 英數字를 쓰라는 意味. 英數字의 個數를 앞에, 쓰여야 할 英數字를 뒤에 쓴다. "1 HI"는 "I"이라는 英數字를 1字쓰라. "6 HANSWER"는 "ANSWER"라 하는 6個의 英數字를 쓰라. "2HXI"는 "XI"이라 하는 2個의 英數字를 쓰라. "3HX2="는 "X2"라 하는 3個의 英數字를 쓰라. 그리고 "IH1"는 印刷가 行하여지는 것이 아니고 用紙를 새로 바꾸어라는 意味가 된다.

ㄴ) "X"—空白(Blank)을 意味한다. 앞의 數는 그 個數, "2X"는 2個의 空白 "12X"는 12個의 空白

ㄷ) "1"—은 行을 바꾸어라는 意味(FORMAT)中에 演算을 쓰는 일은 없으므로 나누기 計算의 記號와 착오를 이르키는 일은 없다.

ㄹ) "F"—는 (2)와 같다. 이 경우 처음의 X1의 값이 다음에 F 6.2에 X2의 値가 써 넣어진다.

(6) 1003의 FORMAT 表現의 ( ) 안에는 그림-4의 12 및 13行에 쓰여진 모습이 表示되고 있다.

(7) 1004의 FORMAT 表現의 ( ) 안에는 그림-4의 14行에 쓰여진 모습이 表示되고 있다.

(8) Program의 最後에는 반드시 "END"라고 쓴다.

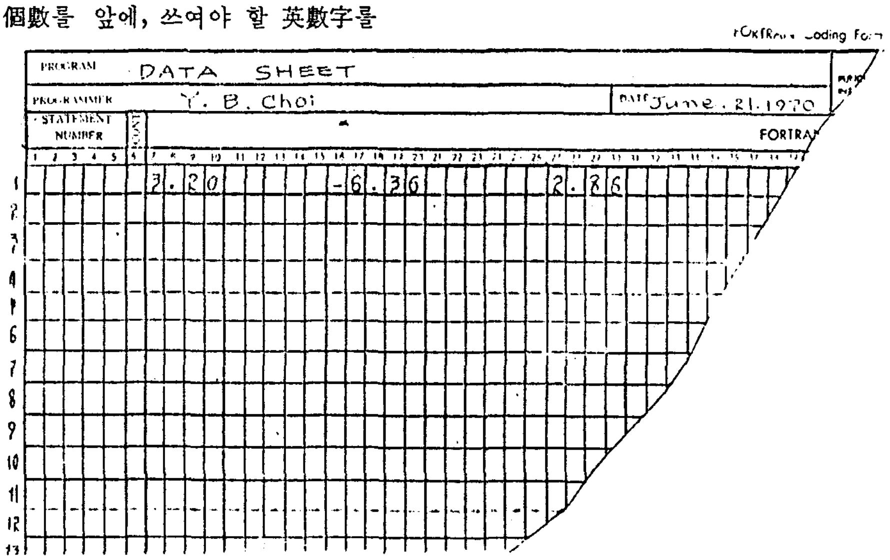


그림 6. Data(例題)

그리고 Program 을 Punch 할 때 數字의 0과 英字의 O는 誤讀하기 쉬우므로 이 경우에 英字를 Ō 및 ϕ로 쓰고 數字는 0를 써서 區別한다.

Data 는 設計된 모습에 따라서 Data Sheet 에 記入된다. 記入에 있어서 數字, 正負, 記號, 小數點 등은 모두 1 Columm 이 使用되고 正負記號가 없는 것은 모두 正으로서 읽어진다. 그림-6은 例題의 Data 를 Data Sheet 에 記入한 것이다.

6. Punch Crad 와 Crad deck 의 作成

記入된 Coding Sheet 와 Data Sheet 는 Kee-puncher 에 주어져서 Punch Card 가 行하여 진다. 한번 뚫인 Punch Card 는 다른 Kee-puncher 에 주어지서 Punch 檢査機로서 檢査된다. 檢査에 error 가 없는 Punch Card 에는 右端에 小半圓形의 切取가 行하여진다. 完成된 各 Punch Card 群은 Coding Sheet 또는 *Data Sheet* 에 쓰여진 順으로 바르게 겹쳐있지 않으면 안된다.

7. Test 欄과 修正

完成된 Card deck 를 電子計算機에 넣으면 되는 것인데 實際로 한꺼번에 結果가 얻어지는 것은 熟練한 Programmer 라도 거의 없다. 그래서 本欄에 들어가기 前에 Test 欄과 修正이 行하여진다.

Test 欄에서는 Data 의 全部 또는 一部가 使用되는 경우가 많은데 Test 欄用의 特別한 Data 가 使用될 경우도 있다. 例를 들면 例題의 경우 그 Data 만으로서는 IF 文의 分岐의 Test 를 行할 수 없다.

Program 이 大型인 경우로서 몇 個의 Block로 나누어져 있을 경우는 各 Block 에 對하여 Test 欄의 修正이 反復되어 各 Block 가 바른 것이 된 후에 全 Program 의 Test 欄이 行하여진다.

8. 本欄

몇 回의 Test 欄의 修正結果 바른 Program 이 얻어지면 이것과 全 Data 에 依하여 本欄이 行하여진다. 但 Test 欄에 全部의 Data 가 使用될 경우는 最後의 Test 欄이 本欄이 된다. Program 은 本欄 또는 Test 欄에 立會하는 것이 所望된다.

<41p 에서 繼續>

組立하여야 된다. 이때 部分 組立에서 綜合組立으로 다시 나누어지고, 部分品間에 相互組立에 있어 各己 特性을 살려 所要系統에 合致시켜야 된다는 煩雜性이 있고, 一般的으로 調整, 試驗等과 같은 習達技術等이 所要되는 妙味를 살리는 工程管理가 되어야 한다.

③ 部分品과 組立生産은 部分品에서 組立까지 一貫作業으로 生産하는 것으로 앞 ①과 ②인 경우를 合친 工程管理가 되어야 하고 더 한층 어려운 管理面을 內包하고 있는 것이다.

以上의 것을 一般的으로 綜合하고 보면 그림 4와 같이 되고 그림의 下部로 옮겨갈 수록 管理의 어려움도 加增한다.

따라서 같은 工程管理라하여도 크게 여섯개의 要素가 있고 그 中의 (1) 形態는 I, II, III의 3

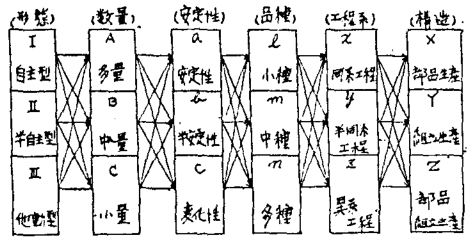


그림 4. 工程管理의 一般表示式

$$_{\text{I.II.III}}S_{1.A.B.C}S_{1.a.b.c}S_{1.l.m.n}S_{1.x.y.z}S_{X.Y.Z_n}S_1$$

그림 5. 工程管理의 體系

條件이 있고, (2) 數量은 A, B, C 의 3 條件이, (3) 安定性에는 a, b, c 3 條件, (4) 品種에는 l, m, n 3 條件, (5) 工程系에는 x, y, z 3 條件, (6) 生産構造에는 X, Y, Z 의 3 條件이 있는 것이다. 即 工程管理에는 6 個要素中 18 個條件으로 各各 組合한다면 $3^6=729$ 種이나 된다고 할 수 있다.