

制御入力과 複合機械 (Control Input and Polymachine)

金 玄 在*
(Kim, Hyon-Jae)

Abstract

In the description and design of certain kinds of sequential machines, we often meet inadequacies of traditional models which we adopt to use.

To lessen these inconveniencies, this paper aims to define polymachine as a new basic model of sequential machine.

Polymachine, defined on the level of state model, is composed of an index which is an independent component and a polymer which, as a dependent component, consists of many element machines.

In the meantime, some basic characters of the machine are illustrated and discussed in detail, in order to present the usefulness of polymachine.

1. 緒 論

順序機械(Sequential Machine)¹⁾를 記述하거나 設計함에 있어서 順序機械를 表現하는 모델(Model)의 不適合性 때문에 가끔 많은 不便을 겪는데, 이와같은 不便을 덜기 爲하여, 여기에 새로운 基礎모델인 複合機械(Polymachine)의 概念을 定義한다.

2. 定 義

複合機械(Polymachine)를 定義하기 위하여 몇가지의 새 術語가 쓰이게 되므로, 複合機械의 概念을 定義하기에 앞서, 우선 複合機械를 構成하기 위하여 導入되는 各 部分機械(Submachine)²⁾의 定義에서부터 始作하기로 한다.

a) 素機械(Element Machine)

複合機械는 여러개의 素機械(Element Machines)가 모여서 이루어졌는데, 各 素機械는 그 素機械 固有의 傳達特性, 卽 入出力特性을 完備한 하나의 機能素子(Functional Unit)이다. (그림 1 참조)

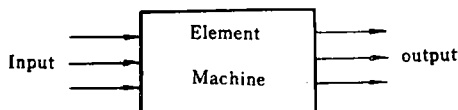


그림 1. 素機械
Fig. 1. An Element Machine

b) 複合體(Polymer)

複合體(Polymer)는 複合機械의 部分機械(Submachine)이며, 여러가지 素機械의 集合體이다.

複合體內에 包含되는 素機械의 種類數를 複合體(또는 複合機械)의 複合度(Polyness)라 하며, m 개의 素機械들이 모여서 하나의 複合體를 이루고 있을 때, 이 複合體의 複合度는 m 이라 한다.

複合體 P 가 M_1, M_2, \dots, M_m 과 같은 m 개의 素機械로 이루어졌다면,

$$P = \{M_1, M_2, \dots, M_m\}$$

複合體가 받아들이는 入力은 데이터入力(Data Input)이며, 그 入力記號의 集合을 I_d , 出力記號의 集合을 Z 라 할 때, 各 素機械의 特性을 그 入出力雙(I/O pair)으로 表示할 수도 있으므로,

$$P \subseteq I_d \times Z$$

c) 指示器(Index)

指示器(Index)는 制御入力(Control Input) I_c 에 依

* 正會員 全南大學校 計測工學科
接受日字: 1977年 1月19日

해 驅動되는 무우어型機械(Moore Machine)이며, 指示器의 한 狀態出力은 어느 素機械의 動作을 指示한다.

指示器의 狀態集合을 Q , 또 指示器의 各 狀態와 素機械 사이의 對應關係를 指示函數(Index Function) η 라 할 때,

$$Q = \{q_1, q_2, \dots, q_n\}$$

$$\eta(Q) = P$$

d) 複合機械 (Polymachine)

複合機械(Polymachine)는 指示器(Index)와 複合體(Polymer)의 두 部分機械(Submachine)로 構成된 順序機械(Sequential Machine)이며, 複合機械의 入力(Input) I 는 制御入力(Control Input) I_c 와 데이터入力(Data Input) I_d 로 區分되는데, 그 두가지 入力中에서 制御入力は 指示器가 받아들이는 入力이며, 데이터入力は 複合體가 받아들이는 入力이다.

한편, 複合體의 出力은 바로 複合機械의 出力이 된다.

指示器의 狀態는 어느 素機械의 動作을 選擇指示하게 되므로, 複合機械의 데이터入力特性은 어느 時點에서 指示器에 依하여 選擇된 素機械의 入出力特性이 된다. 卽, 指示器의 制御에 依하여 여러가지 素機械의 入出力特性이 어떤 順序에 따라 時間적으로 交代되는 特性을 가진 것이 複合機械이다.

複合機械를 S 라 할 때, S 는 다음과 같은 6重組(6-tuple)로 표시할 수 있다.

$$S = \langle I_c, I_d, Q, Z, \delta, \omega \rangle$$

여기서 I_c 는 制御入力記號의 有限集合, I_d 는 데이터入力記號의 有限集合, Q 는 狀態集合, Z 는 出力記號의 有限集合, 寫像 δ 는 狀態遷移函數(Next State Function)이며 $\delta : I_c \times Q \rightarrow Q$, ω 는 出力函數(Output Function)이며,

$$\omega : I_d \times Q \rightarrow Z$$

그런데,

$$I = I_c \vee I_d$$

$$I_c \wedge I_d = \phi$$

$$P = \{M_1, M_2, \dots, M_m\}$$

$$\eta(Q) = P \subseteq I_d \times Z$$

한편, 複合機械를 構成하는 各 素機械의 入出力特性은 個別的으로 規定되어야 하며, 各 素機械 相互間에 入出力線을 共有하거나 入出力領域을 共有하는 일에 對해서도 아무런 制限이 없으므로 別途로 規定되어야 한다.

그림 2는 複合機械의 構成圖이다.

또 그림 2(b)는 指示圖(Index)만을 따로 그린 것이다.

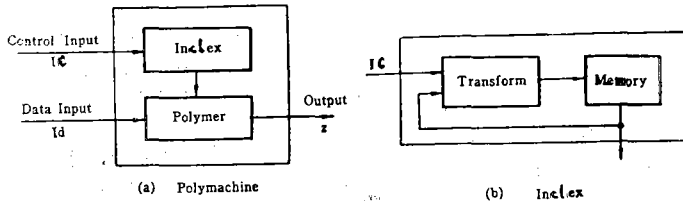


그림 2. 複合機械와 指示器
Fig. 2. Polymachine and Index

3. 應用例

이번에는 몇가지의 간단한 應用例를 가지고 複合機械의 性質을 吟味해 본다.

例 1 소프트웨어와의 對應關係

(Parallelism with software)

마이크로프로그램(Microprogram)의 형태로 記述된 소프트웨어(Software)가 있을 때, 이것을 複合機械로 表現하는 문제로 생각해 본다. 마이크로프로그램의 各 단계(Step)에서 行해지는 個別的인 作業(Job)에 對하여 그 各 作業을 遂行하는 素機械를 作成하고, 그 素機械들을 모두 모아서 複合體를 構成하는 한편, 마이크로프로그램 各 단계間의 遷移特性을 指示器의 遷移特性에 付與해서 複合機械를 만들 때, 이 複合機械는 주어진 소프트웨어에 依해 記述된 機能과 꼭 같은 機能을

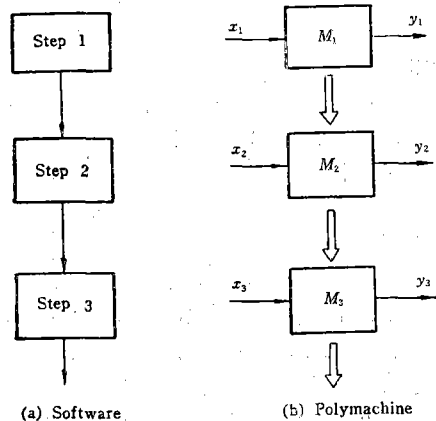


그림 3. 소프트웨어 및 複合機械
Fig. 3. Software and Polymachine

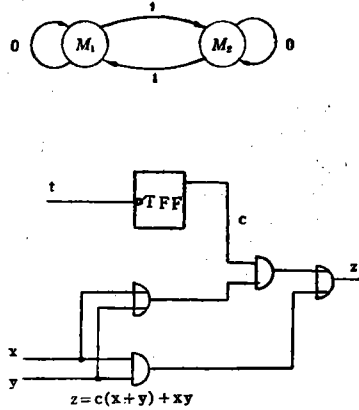
맞는다. 그림 3은 소프트웨어와 複合機械의 順序動作 機能(Sequential Behavior)을 서로 比較한 것이다.

例 2 AND-OR 回路

入出力線을 共有하는 두個의 素機械 M_1 및 M_2 로 構成된 複合機械 S 가 있다.

O/I state	Input			output z
	c	x	y	
M_1	0	0	0	0
	0	0	1	0
	0	1	0	0
	0	1	1	1
M_2	1	0	0	0
	1	0	1	1
	1	1	0	1
	1	1	1	1

(a) Truth Table



(b) Logical Circuit

그림 4. S 의 構成
Fig. 4. Construction of S

例 3 미이리型機械의 變換

그림 5(a)와 같이 간단한 미이리型機械가 주어졌을 때, 이것을 複合機械의 形態로 表現해 본다.

이 미이리型機械가 갖는 入出力雙의 種類는 0/0, 0/1, 1/0, 1/1이므로, 이것을 複合機械의 狀態記號 A, B, C, D 에 對應시킨다. 이 미이리型機械는 別途로 指定된 制御入力を 갖고 있지 않으므로, 미이리型機械에 印加되는 데이터入력을 그대로 制御入력으로 兼用해서 그림 5(b)와 같은 遷移表를 얻는다.

단 A 라는 狀態는 두가지 位置를 가지며, 그 位置에 따라 遷移해가는 다음 狀態가 서로 다르므로, 이것을 A_1, A_2 의 두가지로 分離시켰다.

따라서 이 遷移表대로 複合機械의 狀態圖를 그리면 그림 5(c)와 같다.

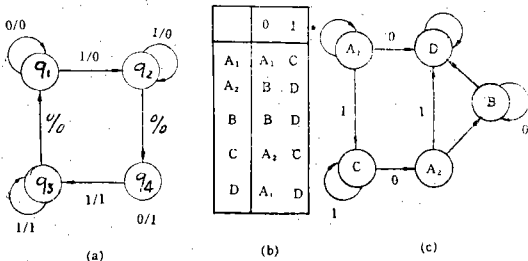


그림 5. 미이리型機械와 等價複合機械
Fig. 5. Mealy machine and equivalent polymachine

M_1 은 AND機能을, M_2 는 OR機能을 가졌으며, S 는 狀態轉換用 制御入力펄스 t 를 받아서 그 狀態를 反轉하게 된다고 한다. 指示器의 出力을 c , 데이터入력을 x 및 y , 出力을 z 라 한다면 그림 4와 같은 複合機械로 된다.

그림 5(c)의 複合機械에서 A_1 을 初期狀態로 삼고 010100110011과 같은 入力系列(Input string)을 가했을 때, $A_1CA_2DA_1A_1CCA_2BDD$ 와 같은 狀態遷移가 일어나므로, 각 狀態에 매겨진 出力值를 따서 000100000111과 같은 出力系列(Output String)이 얻어진다.

그런데 이는 그림 5(a)의 미이리型機械에서 Q_1 을 初期狀態로 삼았을 때의 入出力特性과 꼭 같음을 알 수 있다.

例 4, 무우어型機械의 類似性

무우어型機械와의 類似性을 檢討하기 爲하여, 한 미이리型機械를 무우어型機械와 複合機械의 두가지로 表現해서 比較해 본다.

그림 6(a)와 같은 미이리型機械가 주어졌을 때, 이것을 무우어型機械로 等價變換시키면 그림 6(b)와 같이 된다.

한편, 그림 6(a)의 미이리型機械를 複合機械로 表現하면 그림 7(a)와 같다.

그런데 그림 7(a)의 複合機械에서 데이터入력을 無視해버리고, 각 狀態의 出力值만이 고려된다면 $A_2=C, B_2=D_1$ 으로 놓을 수 있으므로, 이 遷移表를 더 간단화시켜 그림 7(b)와 같은 遷移表가 얻어지며, 그 狀態圖는 그림 7(c)와 같다. 여기서 그림 6(b)와 그림 7(c)를 比較해 볼 때 두 狀態圖가 서로 같은 構造를 갖고 있음을 알 수 있다.

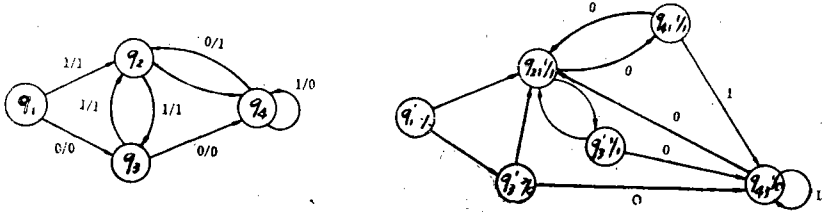


그림 6. 미이리型機械 및 等價무우어機械
Fig. 6. Mealy machine and equivalent Moore machine.

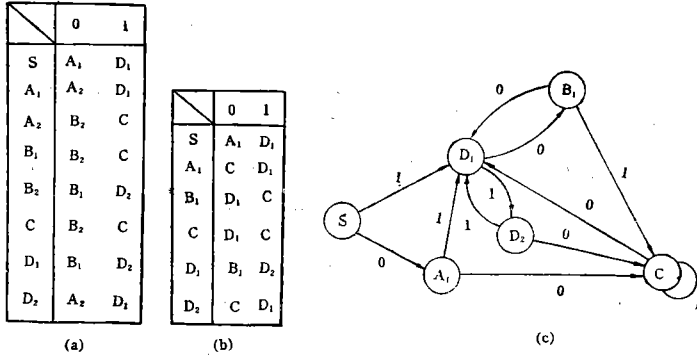


그림 7. 等價複合機械
Fig. 7. Equivalent polymachine

即 무우어型機械는 데이터入力이 退化된 複合機械이다.

4. 結 論

이 複合機械는 미이리型機械나 무우어型機械等과 함께 順序機械모델의 한 種類로 생각할 수 있으며, 알 예에서 提示한 바와 같이 그들 사이에는 互換性이 있다.

또 例1에 提示한 바와 같이 어떤 소프트웨어(Software), 또는 마이크로프로그램(Microprogram)의 形態로 記述된 順序機械가 주어졌을 때, 이와 같은 소프트웨어가 갖고 있는 알고리즘的特徵(Algorithmic character)을 複合機械로 表現할 수 있다. 即 複合機械는 소프트웨어(Software)와 하드웨어(Hardware) 사이에 介在하는 界面(Interface)에서의 整合條件을 自己의 表現方式안에 이미 갖추고 있다.

또 複合機械의 定義에서, 各素機械의 種類와 特性에 關係서는 特別히 限定한 바가 없으므로, 그 個個의 部分機械(Submachine)를 다시 各己 獨立된 새로운 複合機械로 構成할 수 있다.

即 論理設計過程에서, 어떤 複雜한 順序機械를 實現해야 할 때, 이것을 表現하는 複合機械 또는 指示器의 階構造(Hierarchy)가 許容된다.

한편, 論理設計過程을 살핀 때, 狀態圖로 表現되는

狀態機械(State Machine)가 求해지기 以前에 그에 先行하는 다른 言語表現이 있으므로, 그 先行言語表現을 狀態機械의 一種인 複合機械로 變換시키는 문제와, 또 주어진 複合機械를 論理圖等에 의해 表現되는 論理機械로 變換시키는 具體的인 問題等은, 미이리型機械나 무우어型機械의 경우를 참고하면 된다.

그러나 어떤 種類의 順序機械를 設計할 경우, 順序機械모델의 不適合性에 緣由하는 論理設計過程의 複雜性和 不便을 덜고, 더 合理的이며 一貫性있는 設計過程을 實現하기 위하여 提示된 모델이 바로 이 複合機械이다.

參 考 文 獻

1. Taylor L. Booth, "Sequential Machines and Automata Theory" John Wiley, 1967.
2. Frederick C. Herrie, "Finite-state Models for Logical Machines" John Wiley, 1968.
3. M. Phister, "Logical design of digital computers" John Wiley, 1958.
4. Christopher R. Clare, "Designing logical systems using state machines" McGraw-Hill, 1972.
5. 本多波雄 著, "オートマトン・言語理論" コロナ社 1972.
6. 深尾 毅 著, "システム理論入門" 昭晃堂, 1972.