관리제어기 설계

조 광 현*, 임 종 태**

*울산대학교 전기전자및자동화공학부, **한국과학기술원 전기및전자공학과

Abstract : 본 논문에서는 관리제어시스템의 사양으로 주어진 언어가 일반적으로 제어불가능이거나 관측불가능인 상황하에서 관리제어기 설계에 대한 문제를 다룬다. 먼저 사양언어가 제어불가능인 경우 제어가능성의 성질들을 이용하여 사양언어에 근사한 최고 제어가능 부언어 및 최저 닫힘 제어가능 초언어를 유도해내고 이를 바탕으로 관리제어기를 설계한다. 그리고 사양언어가 관측불가능인 경우는 관측가능성의 집합연산에 대한 성질들을 이용하여 사양언어에 가장 근사하면서 관측가능인 초언어 및 관측가능성보다 강한 조건인 표준성을 도입하여 관측가능 부언어 등을 유도하고 이를 토대로 관리제어기를 설계한다. 나아가 사양언어가 제어불가능이면서 관측불가능인 경우의 관리제어기 설계에 대해 결과를 확장한다.

1. 서론

이산사건시스템의 관리제어 (I): 이산사건모델링 및 관리제어이론[8]에서 우리는 관리제어시스템의 사양으로 언어 K가 주어졌을 때 제어가능이고 관측가능이면 이를 만족시키는 관리제어기가 존재하며 이를 구현할 수 있다는 것을 알았다. 그러나 만일 K가 제어불가능이거나 관측불가능이면 어떻게 관리제어시스템을 설계해야 할까? 이 경우 관건은 K와 가장 비슷하면서 제어가능이고 관측가능인 언어 K'을 찾아 앞서와 마찬가지로 이를 만족시키는 관리제어기를 구현하는 것이 된다. 그러므로 일반적인 상황하에서 관리제어기의 설계는 이러한 K'을 어떻게 정의하고 그 성질을 이용해 K로부터 K'을 유도해 낼 것인가의 문제로 귀착된다. 이를 위해 먼저 K가 제어불가능인 경우 제어가능성의 일반적인 성질들을 이용하여 K에 포함되고 제어가능이면서 가장 큰 최고 제어가능 부언어(supremal controllable sublanguage)와 K를 포함하고 제어가능이며 가장 작은 최저 닫힘 제어가능 초언어(infimal closed and controllable superlanguage)를 각각 정의한다. 그러면 관리제어기 설계문제의 여러 유형별로 정의된 언어들로부터 해(solution)를 유도해낼 수 있다. 또한 본 논문에서는 최고 제어가능 부언어를 체계적으로 구하기 위해 집합이론의 격자(lattice)이론에 근거한 알고리즘을 소개하고 이의 수렴성에 대해 논한다. 그리고 최저 닫힘 제어가능 초언어를 구하기 위한 언어의 연산과 계산상의 복잡도 등을 알아보고 각 연산의 성질들을 정리해 본다.

K가 관측불가능인 경우 관측가능성의 성질들을 이용하여 앞서와 마찬가지로 최고 관측가능 부언어(supremal observable sublanguage)와 최저 닫힘 관측가능 초언어(infimal closed and observable superlanguage)를 각각 정의하고 이로부터 각 유형별 관리제어기 설계문제의 해를 유도해낼 수 있다. 그러나 이 경우 관측가능성이 합집합 연산에 대해 닫혀있지 않아 최고 관측가능 부언어를 체계적으로 구하기가 어려우므로 관측가능성에 더 많은 제약조건을 인가하여 합집합 연산에 대해 닫혀있는 언어의 표준성(normality)이라는 새로운 성질을 도입한다. 그리고 이를 토대로 최고 표준 부언어(supremal normal sublanguage)를 유도하여 정규언어의 경우 유한 단계이내에 수렴하는 알고리즘을 통해 최고 관측가능 부언어 대신 활용할 수 있음을 논한다. 또한 이로부터 K가 제어불가능이면서 관측불가능인 경우 일반적으로 최적의 해이긴 하나 체계적으로 구할 수 없는 최고 제어가능 및 관측가능 부언어(supremal controllable and observable sublanguage) 대신 최적의 해는 아니지만 계산가능한 해인 최고 제어가능 및 표준 부언어(supremal controllable and normal)를 구함으로써 일반적인 상황하에서의 관리제어기 설계로 결과를 확대 해석한다.

본 논문의 구성은 다음과 같다. 2절에서는 K가 제어불가능인 경우의 관리제어기 설계문제를 다루고, 3절에서는 K가 관측불가능인 경우의 관리제어기 설계문제를 다루며 또한 K가 제어불가능이면서 관측불가능인 경우로 결과를 확장한다. 그리고 4절에서는 결론을 맺는다.

2. K가 제어불가능인 경우의 관리제어기 설계

이 절에서는 먼저 K가 제어불가능인 경우 제어가능성의 일반적인 성질들을 이용하여 K로부터 가장 근사하면서 제어가능인 부언어(sublanguage)와 초언어(superlanguage) $K^{\uparrow}$, $K^{\downarrow}$를 각각 정의하고 이를 토대로한 관리제어기의 설계에 관해 논한다.

2-1 제어가능성의 성질

[8]의 정의에 따르면 어떠한 $K \subseteq M$(이 때 K는 접두닫힌 언어일 필요는 없음)가 주어졌을 때 $\overline{K}\Sigma_{uc} \cap M \not\subseteq \overline{K}$이면 K는 $M = \overline{M} \subseteq \Sigma^*$와 $\Sigma_{uc} \subseteq \Sigma$에 대해 제어불가능이다. 이 경우 언어 K는 하나의 집합이기 때문에 이에 가장 근사한 언어로는 K에 포함되는 부분집합가운데 제어가능이면서 가장 큰 최고 제어가능 부언어(supremal controllable sublanguage) $K^{\uparrow}$와 K를 포함하는 집합가운데 제어가능이며 가장 작은 최저 닫힘 제어가능 초언어(infimal closed and controllable superlanguage) $K^{\downarrow}$를 생각할 수 있다[1], [2], [4], [7]. 즉, 이들 언어사이의 대소관계는 $\phi \subseteq K^{\uparrow} \subseteq K \subseteq \overline{K} \subseteq K^{\downarrow} \subseteq M$와 같다. 그러면 이와 같은 $K^{\uparrow}$와 $K^{\downarrow}$는 항상 존재하는 것일까? 존재한다면 K로부터 어떻게 도출해 낼 수 있을까? 이에 대한 해답을 구하기 위해 먼저 제어가능 언어의 성질들을 살펴보자.

명제 2.1〔1〕, 〔4〕: $K_1, K_2 \subseteq M$와 $\Sigma_{uc} \subseteq \Sigma$에 대해

1) 만일 K_1과 K_2가 제어가능이면 $K_1 \cup K_2$도 제어가능이고,
2) 만일 K_1과 K_2가 제어가능이라도 $K_1 \cap K_2$는 제어불가능일 수 있으며,
3) 만일 K_1과 K_2가 제어가능이면서 비충돌적이면 $K_1 \cap K_2$도 제어가능이고,
4) 만일 K_1과 K_2가 제어가능이면서 닫힌 언어이면 $K_1 \cap K_2$도 제어가능이면서 닫힘성을 지닌다.

위 명제 2.1에 나타나 있는 각각의 성질을 살펴보면 다음과 같다. 먼저 성질 1)은 $\overline{(K_1 \cup K_2)}\ \Sigma_{uc} \cap M = (\overline{K_1} \cup \overline{K_2})\Sigma_{uc} \cap M = (\overline{K_1}\Sigma_{uc} \cap M) \cup (\overline{K_2}\Sigma_{uc} \cap M) \subseteq \overline{K_1} \cup \overline{K_2} = \overline{K_1 \cup K_2}$ 이므로 성립됨을 알 수 있다. 성질 2)로부터 다음의 반례(counter example)를 생각해 보자: 즉, $\Sigma_{uc} = \{\alpha\}$, $\Sigma = \{\alpha, \beta, \gamma\}$이고 $M = \{\varepsilon, \alpha, \alpha\beta, \alpha\gamma\}$, $K_1 = \{\varepsilon, \alpha\beta\}$, $K_2 = \{\varepsilon, \alpha\gamma\}$이면 K_1, K_2는 각각 제어가능이지만, $\varepsilon\alpha \in \overline{(K_1 \cap K_2)}\ \Sigma_{uc} \cap M$이나 $\alpha \notin \overline{K_1 \cap K_2} = \{\varepsilon\}$이므로 $K_1 \cap K_2$는 제어불가능이다. 또한 성질 3)의 경우 K_1, K_2가 비충돌적이면 $\overline{K_1 \cap K_2} = \overline{K_1} \cap \overline{K_2}$이고, $\overline{(K_1 \cap K_2)}\ \Sigma_{uc} \cap M \subseteq (\overline{K_1} \cap \overline{K_2})\Sigma_{uc} \cap M = (\overline{K_1}\Sigma_{uc} \cap M) \cup (\overline{K_2}\Sigma_{uc} \cap M) \subseteq \overline{K_1} \cap \overline{K_2} = \overline{K_1 \cap K_2}$ 이므로 성립됨을 알 수 있다. 그리고 성질 4)의 경우 닫힌 언어들은 항상 비충돌적이며 닫힘성은 교집합 연산에 대해 보존되므로 성립됨을 알 수 있다.

다음 2가지 부류의 언어를 생각해 보자.

- $C_{in}(K) := \{L \subseteq K \mid \overline{L}\Sigma_{uc} \cap M \subseteq \overline{L}\}$
- $CC_{out}(K) := \{K \subseteq L \subseteq M \mid (\overline{L} = L) \wedge (\overline{L}\Sigma_{uc} \cap M \subseteq \overline{L})\}$

그러면 먼저 집합 $C_{in}(K)$는 합집합 연산에 대해 닫혀있는 부분정렬집합(partially ordered set or poset)임을 알 수 있다. 이 때 부분정렬집합이란 일부 비교 불가능한 원소를 허용하는 비교관계(comparison relation)를 가지는 집합을 의미한다[3]. 이를테면, 집합 W와 이 집합의 원소 x, y ∈ W에 대해 정의된 관계(relation)≤를 생각해보자. 이 경우 x가 y에 관계되어 있으면 x≤y로 표기한다. 그리고 이 관계가 다음 3가지 조건을 만족시키면 집합 W의 부분정렬(partial order)이라고 한다:

1) 재귀적(reflexive)조건: 모든 x∈W에 대해 x≤x이다.
2) 비대칭(antisymmetric)조건: x≤y이고 y≤x이면 x=y이다.
3) 이행성(transitive)조건 : x≤y이고 y≤z이면 x≤z이다.

이와 같이 ≤가 집합 W의 부분정렬일 때, 쌍(pair) (W, ≤)를 부분정렬집합이라고 한다. 이러한 부분정렬집합은 합집합 연산에 대해 닫혀있기 때문에 부분집합의 합집합들 가운데 최대집합(집합의 원소 개수를 cardinality로 정의 하였을 때)이 존재하면 유일(unique)하게 주어지는 특성이 있다. 그러면 최고제어가능 부언어 $K^{\uparrow}$는 $C_{in}(K)$로부터 다음과 같이 구할 수 있다[1], [4]:

$$K^{\uparrow} := \bigcup_{L \in C_{in(K)}} L.$$

최악의 경우 $K^{\uparrow} = \phi$일 수 있으며, 만일 K가 제어가능이면 $K^{\uparrow} = K$가 된다. 또한 K가 닫힌 언어이면 $K^{\uparrow}$도 닫힌 언어가 되며, $K \subseteq L_m(G)$이고 $L_m(G)$-닫힘성을 지니면 $K^{\uparrow}$도 $L_m(G)$-닫힘성을 지니게 되고, 일반적으로 $\overline{K^{\uparrow}} \subseteq (\overline{K})^{\uparrow}$임을 알 수 있다.

[예] $\Sigma = \Sigma_{uc} \cup \Sigma_c = \{\beta_1, \beta_2\} \cup \{\alpha_1\}$, $M = \overline{\alpha_1\beta_2\alpha_1 + \alpha_1\beta_1\alpha_1\beta_1\alpha_1}$, 그리고 $K = \alpha_1(\beta_1 + \beta_2)\alpha_1$으로 주어졌을 때 $K^{\uparrow}$를 구해보자. 이 경우 $(\alpha_1\beta_1\alpha_1)\beta_1 \in \overline{K}\Sigma_{uc} \cap M$이지만 $\alpha_1\beta_1\alpha_1\beta_1 \notin \overline{K}$이고, $K_1 = K - \{\alpha_1\beta_1\alpha_1\} = \alpha_1\beta_2\alpha_1$이라 할 때 $(\alpha_1)\beta_1 \in \overline{K_1}\Sigma_{uc} \cap M$이지만 $\alpha_1\beta_1 \notin \overline{K}$이므로 $K^{\uparrow} = \phi = \overline{K^{\uparrow}}$임을 알 수 있다. 그러나 한편 $\overline{K} = \varepsilon + \alpha_1(\varepsilon + \beta_1 + \beta_2) + \alpha_1(\beta_1 + \beta_2)\alpha_1$이므로 $(\overline{K})^{\uparrow} = \varepsilon + \alpha_1(\varepsilon + \beta_1 + \beta_2) + \alpha_1\beta_2\alpha_1$이 되고, 따라서 $\overline{K^{\uparrow}} \subseteq (\overline{K})^{\uparrow}$가 성립됨을 알 수 있다.

집합 $CC_{out}(K)$ 또한 임의의 합집합 연산과 교집합 연산에 대해 닫혀있는 부분정렬집합이며 이로부터 최저닫힘 제어가능 초언어 $K^{\downarrow}$를 다음과 같이 구할 수 있다[1]:

$$K^{\downarrow} := \bigcap_{L \in CC_{out(K)}} L.$$

최악의 경우 $K^{\downarrow} = M$ 이 되며, 만일 K가 제어가능이면 $K^{\downarrow} = K$가 된다.

[예] $\Sigma = \Sigma_{uc} \cup \Sigma_c = \{\beta_1, \beta_2\} \cup \{\alpha_1\}$, $M = \overline{\alpha_1\beta_1\alpha_1} + \overline{\alpha_1\beta_1\alpha_1\beta_1\alpha_1}$, 그리고 $K = \alpha_1(\beta_1 + \beta_2)\alpha_1$으로 주어졌을 때 $K^{\downarrow}$는 $K^{\downarrow} = \bigcap_{L \in CC_{out(K)}} L = \varepsilon + \alpha_1(\varepsilon + \beta_1 + \beta_2) + \alpha_1(\beta_1 + \beta_2)\alpha_1 + \alpha_1\beta_1\alpha_1\beta_1$이 된다.

2-2 관리제어기 설계

관리제어기 설계문제는 주어진 상황에 따라 다음과 같이 몇 가지 유형으로 구분될 수 있다.

유형 1. 기본 관리제어기 설계문제: 이산사건시스템 G에 대해 $\Sigma_{uc} \subseteq \Sigma$와 정상(legal)언어 $L_a = \overline{L_a} \subseteq L(G)$가 주어졌을 때

1) $L(S/G) \subseteq L_a$ 이고,
2) 가능한 한 $L(S/G)$가 최대가 되도록 (즉, $L(S'/G) \subseteq L_a$인 어떠한 S'에 대해서도 $L(S'/G) \subseteq L(S/G)$가 되도록) 관리제어기 S를 설계하라.

이러한 유형 1의 관리제어기 설계문제의 해(solution)는 $L(S/G) = L_a^{\uparrow}$가 되도록 하는 S를 찾는 것이 된다. 이 때 이 해를 최소제한의 해(minimally restrictive solution)라고 한다.

유형 2. 비막힘성 관리제어기 설계문제: 이산사건시스템 G에 대해 $\Sigma_{uc} \subseteq \Sigma$와 $L_m(G)$-닫힘성의 정상표기(legal marked) 언어 $L_{am} \subseteq L_m(G)$가 주어졌을 때

1) $L_m(S/G) \subseteq L_{am}$이고,
2) 가능한 한 $L(S/G)$가 최대가 되도록 (즉, $L_m(S'/G) \subseteq L_{am}$인 어떠한 비막힘성 관리제어기 S'에 대해서도 $L(S'/G) \subseteq L(S/G)$가 되도록) 비막힘성 관리제어기 S를 설계하라.

이러한 유형 2의 관리제어기 설계문제의 해는 $L(S/G) = \overline{L_{am}^{\uparrow}}$가 되도록 하는 S를 찾는 것이 된다. 그러면 $L_m(S/G) = \overline{L_{am}^{\uparrow}} \cap L_m(G) = L_{am}^{\uparrow}$이므로 S는 비막힘성이 된다. 이 때 이 해를 최소제한 비막힘성의 해(minimally restrictive and nonblocking solution)라고 한다.

유형 3. 기본 관리제어기 설계의 이원(dual)문제: 이산사건시스템 G에 대해 $\Sigma_{uc} \subseteq \Sigma$와 최소허용가능(minimum acceptable) 언어 $L_{min} \subseteq L(G)$가 주어졌을 때

1) $L(S/G) \supseteq L_{min}$이고,
2) 가능한 한 $L(S/G)$가 최소가 되도록 (즉, $L(S'/G) \supseteq L_{min}$인 어떠한 S'에 대해서도 $L(S'/G) \supseteq L(S/G)$가 되도록) 관리제어기 S를 설계하라.

이러한 유형 3의 관리제어기 설계문제의 해는 $L(S/G) = L_{min}^{\downarrow}$가 되도록 하는 S를 찾는 것이 된다. 이 경우 $L_{min} \subseteq L_m(G)$로 주어지면 $L_m(S/G) = L_{min}^{\downarrow} \cap L_m(G) \supseteq L_{min}$임을 알 수 있다.

유형 4. 허용(tolerance)범위내의 관리제어기 설계문제: 이산사건시스템 G에 대해 $\Sigma_{uc} \subseteq \Sigma$와 $L_m(G)$-닫힘성의 원하는 표기언어 $L_{des} \subseteq L_m(G)$, 그리고 $\overline{L_{des}} \subseteq L_{tol}$인 허용범위내의 정상언어 $L_{tol} = \overline{L_{tol}} \subseteq L(G)$가 각각 주어졌을 때

1) $L(S/G) \subseteq L_{tol}$ 이고,
2) 모든 닫힌 제어가능 언어 $K \subseteq L_{tol}$ 에 대해 $K \cap L_{des} \subseteq L(S/G) \cap L_{des}$ 이며,
3) 모든 닫힌 제어가능 언어 $K \subseteq L_{tol}$ 에 대해 $K \cap L_{des} \subseteq L(S/G) \cap L_{des} \Rightarrow L(S/G) \subseteq K$이 성립하도록 관리제어기 S를 설계하라.

이러한 유형 4의 관리제어기 설계문제의 해는 $L(S/G) = (L_{tol}^{\uparrow} \cap L_{des})^{\downarrow}$가 되도록 하는 S를 찾는 것이 된다.

2-3 ↑연산

앞서 주어진 언어 K에 대한 최고 제어가능 부언어 $K^{\uparrow}$의 정의와 $C_{in}(K)$로부터 구하는 방법에 대해 언급하였다. 이 절에서는 보다 자세하게 $K^{\uparrow}$를 구하는 절차와 알고리즘, 그리고 계산상의 복잡도(computational complexity) 등에 대해서 알아본다.

닫힌 언어 K에 대해 $K^{\uparrow}$는 몫(quotient)연산을 이용해 다음과 같이 구해질 수 있다[4]:

$$K^{\uparrow} := K - [\ (M\text{-}K)\ /\ \Sigma_{uc}^{*}]\ \Sigma^{*}.$$

여기서 몫 연산부분 $(M\text{-}K) / \Sigma_{uc}^{*}$는 [8]의 정의를 따르면 $(M\text{-}K) / \Sigma_{uc}^{*} := \{s \in \Sigma^{*} \mid (\exists t \in \Sigma_{uc}^{*})\ st \in (M\text{-}K)\}$을 의미한다. 일반적으로 정규언어 부류는 이러한 몫 연산에 대해 닫혀 있으므로 K와 M이 정규언어이면 $K^{\uparrow}$ 또한 정규언어가 된다. 한편 이와 같은 ↑연산의 계산상의 복잡도를 정량적으로 가늠하기 위해 먼저 일반적인 언어 L의 또 다른 기수(cardinality)의 정의 $\|L\|$을 생각해 보자. 어떠한 두 문자열 s, $t \in \Sigma^{*}$에 대해 만일 $\{s' \mid s' \in \Sigma^{*}$ 그리고 $ss' \in L\} = \{t' \mid t' \in \Sigma^{*}$ 그리고 $tt' \in L\}$이면 이 문자열들은 서로 모드 L로서 동등(equivalent with mod L)하다고 하며 $s \equiv t \pmod L$ 또는 $s \equiv_L t$ 로 표기한다. 이때 $\|L\|$는 $\|L\| := \mathrm{card}(\Sigma^{*}/\equiv_L)$, 즉 Σ^{*}내의 $\equiv_L$에 대한 동등클래스(equivalent class)의 개수를 나타낸다. 만일 L이 정규언어이면(즉 $\|L\| < \infty$이면), $\|L\|$는 L의 인식기 오토마톤 상태집합의 최소 기수를 의미한다. 위↑연산에서 $\|K\| = n$, $\|M\| = m$이라 할 때 $K^{\uparrow}$을 구하기 위한 계산상의 복잡도는 $O(n^2 m)$으로서 주어진 언어의 기수에 다항식 차수승(ploynomial order)으로 비례함을 알 수 있다.

[예] $\Sigma = \Sigma_c \cup \Sigma_{uc} = \{c\} \cup \{u_1, u_2\}$, $M = \overline{cu_1c(u_1c + u_2u_1)}$, 그리고 $K = \overline{cu_1c(u_1c + u_2)}$로 주어졌을 때 $K^{\uparrow}$을 구해보자. 먼저 $M\text{-}K = cu_1cu_2u_1$이고, $(M\text{-}K) / \Sigma_{uc}^{*} = cu_1c(u_2u_1 + u_2 + \varepsilon)$이므로 $K^{\uparrow} = K\text{-}[(M\text{-}K) / \Sigma_{uc}^{*}]\Sigma^{*} = cu_1$ 임을 알 수 있다.

[예] $\Sigma = \Sigma_c \cup \Sigma_{uc} = \{c\} \cup \{u\}$, $M = \overline{cucuc}$, 그리고 $K = c(\varepsilon + uc(\varepsilon + uc))$로 주어졌을 때 $K^{\uparrow}$을 구해보자. 이 경우는 $M\text{-}K = cu(\varepsilon + cu)$이고 $(M\text{-}K)/\Sigma_{uc}^{*} = c(\varepsilon + u(\varepsilon + c(\varepsilon + u)))$이므로 $K^{\uparrow} = K\text{-}[(M\text{-}K)/\Sigma_{uc}^{*}]\Sigma^{*} = \phi$가 됨을 알 수 있다.

$K^{\uparrow}$을 구함에 있어서 $\|K\|$와 $\|M\|$이 큰 경우 위와 같은 방식을 통해 해석적인(analytic) 해를 구하기가 실제로 쉽지 않다. 따라서 체계적으로 $K^{\uparrow}$을 구해내는 알고리즘이 필요하게 된다. 이러한 알고리즘을 구현하기에 앞서 먼저 집합이론(set theory)으로부터 ↑연산의 수렴성을 밝혀내기 위해 격자(lattice)이론을 소개한다[2], [3].

$(X, \|\cdot\|)$를 집합 X와 노움(norm) $\|\cdot\|$상에 정의된 노움을 가지는 완전 노움 선형공간(complete normed linear space)이라고 하고, $T : X \mapsto X$를 이 공간에서 정의된 연산자(operator)라고 하자. 그러면 $Tx^*=x^*$인 원소 $x^* \in X$를 T에 의한 연산 후에도 그대로 원래 값이 유지되므로 연산자 T의 고정점(FP: Fixed Point)이라고 한다. 다음의 K, M, 그리고 Σ_{uc}에 의해 매개변수화(parameterized)되어 있는 언어상의 연산자 $\Omega^{\uparrow} : 2^{\Sigma^*} \mapsto 2^{\Sigma^*}$를 생각해 보자[1], [4]:

$$\Omega^{\uparrow}(K) := \{t \in K \mid \bar{t}\Sigma_{uc} \cap M \subseteq \overline{K}\}.$$

이 때 $\bar{t} := \{\bar{t}\}$, 즉 문자열 t의 접두어의 집합을 의미한다. 정의로부터 이 연산자에 의해 언어 상호간의 포함관계는 보존됨을 알 수 있다. 이를테면 $A \subseteq B$인 두 언어 A, B에 대해 $\Omega^{\uparrow}(A) \subseteq \Omega^{\uparrow}(B)$가 성립된다. 또한 이 연산자의 정의역(domain)과 공변역(codomain)에 해당하는 2^{Σ^*}는 집합의 포함관계(set inclusion)를 부분정렬로 가지는 완전격자(complete lattice)를 형성하게 되며 임의의 합집합 연산에 대해 닫혀 있으므로 격자이론으로부터 이 연산자 $\Omega^{\uparrow}$는 최대 고정점(LFP: Largest Fixed Point)을 가지게 됨을 알 수 있다. 즉, $\Omega^{\uparrow}(LFP)=LFP$이고 $\Omega^{\uparrow}(FP)=FP$인 모든 FP에 대해 $FP \subseteq LFP$인 LFP가 존재한다. 한편 격자이론에 따르면 완전격자 상의 단조 연산자(monotone oprator)에 대한 모든 고정점들의 집합 또한 동일한 부분정렬을 가지는 완전격자를 이루게 된다.

정리 2.1[1]: $K^{\uparrow}$는 연산자 $\Omega^{\uparrow}$의 최대고정점이다. 즉, $K^{\uparrow}=LFP$.

이 정리 2.1은 ↑연산의 성질로부터 최대고정점이 곧 $K^{\uparrow}$임을 보여준다. 즉, $K^{\uparrow} \subseteq K$이고 $K^{\uparrow}$는 제어가능($\overline{K}^{\uparrow}\Sigma_{uc} \cap M \subseteq \overline{K}^{\uparrow}$)이므로 $T = \overline{T}$이며 $T\Sigma_{uc} \cap M \subseteq \overline{K}^{\uparrow}$인 $T \subseteq \Sigma^*$에 대해 $\overline{K}^{\uparrow} \subseteq T^{\uparrow}$가 성립한다. 따라서 $K^{\uparrow} \subseteq K \cap \overline{K}^{\uparrow} \subseteq \Omega^{\uparrow}(K^{\uparrow})$이 된다. 한편, $t \in \Omega^{\uparrow}(K^{\uparrow})$에 대해 $t \in K$이고 $\bar{t}\Sigma_{uc} \cap M \subseteq \overline{K}^{\uparrow}$이며 $K'=K^{\uparrow} \cup \{t\}$라고 하면 $K' \subseteq K$이 되고, $\overline{K'}\Sigma_{uc} \cap M=(\overline{K}^{\uparrow} \cup \bar{t})\Sigma_{uc} \cap M=(\overline{K}^{\uparrow}\Sigma_{uc} \cap M) \cup (\bar{t}\Sigma_{uc} \cap M) \subseteq \overline{K}^{\uparrow} \subseteq \overline{K'}$가 성립된다. 그러므로 $K' \in \Omega^{\uparrow}(K)$이고, $K' \supset K^{\uparrow}$가 된다. 그런데 $K^{\uparrow}$는 정의로부터 상한치(supremum)이므로 $K'=K^{\uparrow}$이다. 즉, $t \in K^{\uparrow}$이며 $\Omega^{\uparrow}(K^{\uparrow}) \subseteq K^{\uparrow}$가 성립된다. 따라서 $K^{\uparrow}=\Omega^{\uparrow}(K^{\uparrow})$이고 $K^{\uparrow}$는 $\Omega^{\uparrow}$의 최대고정점임을 알 수 있다.

한편, 정리 2.1로부터 $K^{\uparrow}$는 K에 연산자 $\Omega^{\uparrow}$를 차례로 적용해 나감으로써 구해낼 수 있음을 짐작하게 된다. 이러한 취지에서 다음의 반복(iteration)알고리즘을 생각해 보자.

<u>알고리즘</u>

1) $K_0 = K$,
2) $K_{i+1} = \Omega^{\uparrow}(K_i)$, $i \geq 0$.

이 경우 연산자 $\Omega^{\uparrow}$의 정의로부터 $K_0 \supseteq K_1 \supseteq K_2 \supseteq \cdots \supseteq K_i \supseteq K_{i+1}$의 포함관계가 성립됨을 알 수 있다. 또한 위 알고리즘으로부터 정의되는 수열 $\{K_i,\ i \geq 0\}$의 극한값(limit) K_∞가 존재하며 이는 $K_\infty := \bigcap_{i=1}^{\infty} K_i$로 주어진다. 그러나 이 경우 우리의 예상과는 달리 항상 $K_\infty=K^{\uparrow}$가 성립되지는 않으며 일반적으로 $K^{\uparrow} \subseteq K_\infty$가 된다. 즉, $K^{\uparrow} \subseteq K_0$과 수학적 귀납법으로부터 다음 정리 2.2에서와 같이 $K^{\uparrow} \subseteq K_\infty$임을 알 수 있다.

정리 2.2[1]: $K^{\uparrow} \subseteq K_\infty$이다.

[예] $\Sigma = \Sigma_c \cup \Sigma_{uc} = \{\beta\} \cup \{\alpha, \gamma\}$, $M = \overline{\alpha\beta + \alpha\gamma\gamma^*\beta\gamma^*}$, 그리고 $K = \alpha\beta + \alpha\{\gamma^{n+1}\beta\gamma^m \mid 0 \leq m \leq n, n \geq 0\}$로 주어졌을 때 $K^{\uparrow}$와 K_∞를 비교해 보자. 이 경우 $K^{\uparrow}$는 정의로부터 $K^{\uparrow} = \phi$이며, $K_1 = \alpha\beta + \alpha\gamma\{\gamma^{n+1}\beta\gamma^m \mid 0 \leq m \leq n,\ n \geq 0\}$, $\cdots$, $K_i = \alpha\beta + \alpha\gamma^i\{\gamma^{n+1}\beta\gamma^m \mid 0 \leq m \leq n,\ n \geq 0\}$이므로 $K_\infty = \alpha\beta$이고, 따라서 $K^{\uparrow} \subseteq K_\infty$가 성립됨을 알 수 있다.

위와 같이 일반적으로 $K_\infty \supseteq K^{\uparrow}$이지만, 우리의 주된 관심대상인 정규언어에 대해서는 $K_\infty = K^{\uparrow}$가 성립되므로 대부분의 경우 위와 같은 알고리즘을 통해 K로부터 $K^{\uparrow}$를 도출해 낼 수 있다. 이는 격자상의 기수 상호간의 관계와 정규언어에 대한 연산자 $\Omega^{\uparrow}$의 적용에 따른 수열의 수렴성질 등으로부터 유도될 수 있다.

정리 2.3[1]: 만일 K와 M이 정규언어이면

(1) $K_\infty = K^{\uparrow}$이고,
(2) 수열 $\{K_i, i \geq 0\}$는 유한 단계 이내에 K_∞에 수렴하며,
(3) $K^{\uparrow}$가 정규언어인 경우 $\|K^{\uparrow}\| \leq \|K\|\ \|M\|+1$ 가 성립된다.

지금까지 다룬 ↑연산의 성질들을 정리하면 다음과 같다:

1) K가 닫힌 언어이면 $K^{\uparrow}$도 닫힌언어이다.
2) K가 $L_m(G)$-닫힘성을 지니면 $K^{\uparrow}$도 $L_m(G)$-닫힘성을 가진다.
3) K와 M이 정규언어이면 $K^{\uparrow}$도 정규언어이다.
4) $K_1 \subseteq K_2$이면 $K_1^{\uparrow} \subseteq K_2^{\uparrow}$이다.
5) $(K_1 \cap K_2)^{\uparrow} \subseteq K_1^{\uparrow} \cap K_2^{\uparrow}$이다.
6) $(K_1 \cap K_2)^{\uparrow} = (K_1^{\uparrow} \cap K_2^{\uparrow})^{\uparrow}$이다.
7) $K_1^{\uparrow}$와 $K_2^{\uparrow}$가 비충돌적이면 $(K_1 \cap K_2)^{\uparrow} = K_1^{\uparrow} \cap K_2^{\uparrow}$이다.
8) $(K_1 \cup K_2)^{\uparrow} \supseteq K_1^{\uparrow} \cup K_2^{\uparrow}$이다.

2-4 ↓연산

일반적으로 주어진 사양 K에 대한 최저 닫힘 제어가능 초언어 $K^{\downarrow}$는 사양의 허용범위를 벗어나는 사건열을 포함하므로 최고 제어가능 부언어 $K^{\uparrow}$보다 바람직하지 못한 것으로 받아들여지나 경우에 따라서는 [이를테면 $K^{\downarrow}$가 자명한 해(trivial solution)만을 포함하거나 사양의 허용범위가 적용의 임계치(critical value)를 의미하지 않을 때] 보다 좋은 제어결과를 초래할 수도 있다. 이 절에서는 이러한 ↓연산으로부터 $K^{\downarrow}$를 구하는 상세한 방법과 이를 이

용한 관리제어시스템의 설계, 그리고 ↓연산의 설명 등에 대해 알아본다. 먼저 다음 정리 2.4를 생각해 보자.

정리 2.4〔4〕: $K^{\downarrow} = \overline{K}\Sigma_{uc}^{*} \cap M$이다.

이 경우 이를테면 $K' := \overline{K}\Sigma_{uc}^{*} \cap M$라고 하자. 그러면 K'은 닫힌 언어이므로 $K' \supseteq K$이고, $K'\Sigma_{uc} \cap M = \overline{K}\Sigma_{uc}^{*}\Sigma_{uc} \cap M\Sigma_{uc} \cap M \subseteq \overline{K}\Sigma_{uc}^{*} \cap M = K'$으로부터 K'은 제어 가능이므로 $K^{\downarrow} \subseteq K'$이다. 한편, $L \in CC_{out}(K)$라고 하자. 그러면 $\overline{K} \subseteq L$ 이므로 $\overline{K} \cap M \subseteq L$ 이고, $\overline{K}\Sigma_{uc} \cap M \subseteq L\Sigma_{uc} \cap M \subseteq L$, $\overline{K}\Sigma_{uc}^{2} \cap M \subseteq L\Sigma_{uc} \cap M \subseteq L$, $\cdots$, $\overline{K}\Sigma_{uc}^{r} \cap M \subseteq L\Sigma_{uc} \cap M \subseteq L$ $(\forall r \geq 0)$이 각각 성립하므로 $K' \subseteq L$ 이며 따라서 $K' \subseteq \bigcap_{L \in CC_{out(K)}} L = K^{\downarrow}$가 된다. 그러므로 $K^{\downarrow} = K'$임을 알 수 있다.

정리 2.4로부터 $K^{\downarrow}$를 구할 때 계산상의 복잡도는 $\|K\| = n$, $\|M\| = m$ 이라고 하면 $O(nm|\Sigma_{uc}|)$이 된다. 이러한 $K^{\downarrow}$를 이용하면 다음 정리 2.5와 같이 주어지는 관리제어기 설계문제의 해를 구할 수 있다.

정리 2.5〔1〕, 〔2〕: $L_m(G)$ - 닫힘성의 언어 $L_{min} \subseteq L_m(G)$가 주어지고 이산사건시스템 G가 다음의 성질을 만족시킨다고 가정하자: 모든 $q \in Q$ 와 $q_m \in Q_m$ 에 대해 $\delta(q, \sigma) = q_m$ 이면 $\sigma \in \Sigma_c$ 이다. 그러면 $L_m(S/G) = L_{min}$인 비막힘성 관리제어기 S가 존재한다.

이와 같은 경우는 위 정리 2.4의 결과를 토대로 관리제어기 S를 $L(S/G) = L_{min}^{\downarrow}$ 이 되도록 설계하면 된다. 그러면, $L_m(S/G) = L_{min}^{\downarrow} \cap L_m(G) = \overline{L}_{min}\Sigma_{uc}^{*} \cap L(G) \cap L_m(G) = \overline{L}_{min}\Sigma_{uc}^{*} \cap L_m(G) = \overline{L}_{min} \cap L_m(G) = L_{min}$이 됨을 알 수 있다.

위에서 다룬 ↓연산의 성질들을 정리하면 다음과 같다 :

1) K와 M이 정규언어이면 $K^{\downarrow}$도 정규언어이다.
2) $K_1 \subseteq K_2$이면 $K_1^{\downarrow} \cap K_2^{\downarrow}$ 이다.
3) $(K_1 \cap K_2)^{\downarrow} \subseteq K_1^{\downarrow} \cap K_2^{\downarrow}$ 이다.
4) K_1과 K_2 가 비충돌적이면 $(K_1 \cap K_2)^{\downarrow} = K_1^{\downarrow} \cap K_2^{\downarrow}$ 이다.
5) $(K_1 \cup K_2)^{\downarrow} = K_1^{\downarrow} \cup K_2^{\downarrow}$ 이다.

[예] $\Sigma = \Sigma_c \cup \Sigma_{uc} = \{\alpha_1, \alpha_2\} \cup \{\beta\}$, $M = \overline{(\alpha_1\beta^2 + \alpha_2)\beta^*}$, 그리고 $K = \overline{\alpha_1\beta^2} + \alpha_2\beta^*$로 주어졌을 때 앞서 소개한 알고리즘을 이용하여 $K^{\uparrow}$을 구하고 이를 토대로 관리제어기를 구현하여 보자. 먼저 직관적으로 문자열 $\alpha_1\beta^2$는 정상언어이나 $\beta \in \Sigma_{uc}$이므로 $\alpha_1\beta^3$는 정상언어가 아님을 알 수 있다. 이런 직관적인 방식으로 계속 유추하면 $K^{\uparrow} = \varepsilon + \alpha_2\beta^*$가 됨을 예측할 수 있다. 이에 대해 앞의 알고리즘을 통해 체계적으로 계산하여 보면 $K_0 = K$, $K_1 = \overline{\alpha_1\beta} + \alpha_2\beta^*$, $K_2 = \overline{\alpha_1} + \alpha_2\beta^*$, $K_3 = \varepsilon + \alpha_2\beta^*$, $K_4 = K_3$, $K_5 = K_3$, $\cdots$이므로 $K^{\uparrow} = K_3 = \varepsilon + \alpha_2\beta^*$가 됨을 알 수 있다. 따라서 $K^{\uparrow}$을 인식하는 인식기 오토마톤을 토대로 [8]에서처럼 관리제어기를 구현할 수 있다.

[예] 다음 그림 1과 같이 직렬로 연결된 2대의 기계(M_1, M_2)와 사이에 놓여진 1개의 버퍼(B)로 이루어진 간략화된 생산시스템을 생각해 보자[1], [8].

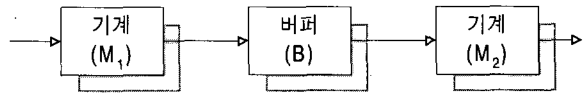

그림 1. 생산시스템의 예 : 2대의 기계와 1개의 버퍼[1].

이 경우 각각의 기계 M_1, M_2는 '대기(I_i: idle)' 또는 '동작(W_i: working)'의 두가지 상태를 가지며 버퍼 B는 '비어있음(E: empty)' 또는 '차있음(F: full)' 상태를 가지는 1개의 슬롯(slot)으로 구성되어 있다고 가정하자. 그러면 이 생산시스템의 부분 이산사건모델은 그림 2와 같다.

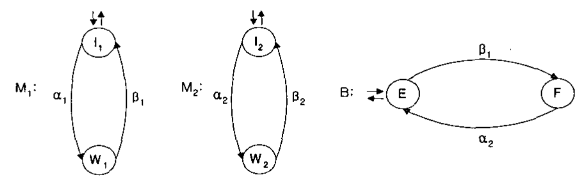

그림 2. 생산시스템의 부분 이산사건모델.

이 시스템은 초기상태 (I_1, I_2, E)에 놓여 있으며 B는 M_1의 사건열 $\alpha_1\beta_1$에 의해 상태 F로 천이 되고 M_2의 사건열 $\alpha_2\beta_2$에 의해 다시 상태 E로 천이 된다. 이 때 $\Sigma = \Sigma_c \cup \Sigma_{uc} = \{\alpha_1, \alpha_2\} \cup \{\beta_1, \beta_2\}$이다.

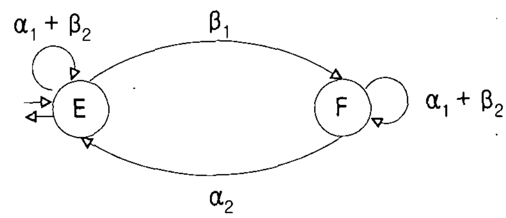

그림 3. L_f의 인식기.

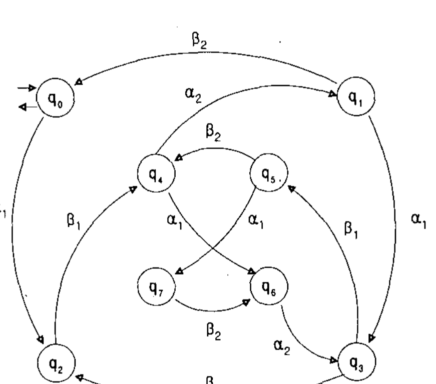

그림 4. K의 인식기.

관리제어시스템에 대한 사양으로 버퍼가 E 또는 F의 상태만을 유지하며 과잉(overflow)을 방지하는 것으로 생각하여 이를 언어 L_f로 표현하면 L_f의 인식기는 그림 3과 같다. 그러면 관리제어시스템의 사양언어 K는 $K=L_f \cap L(\mathrm{G})$가 된다(여기서 $\mathrm{G}=\mathrm{M}_1 \| \mathrm{M}_2$이다). 이러한 사양언어 K의 인식기는 그림 4와 같다. 그림 4의 각 상태 q_i의 정의는 표 1에 나타나 있다.

표 1. 그림 4의 상태정의.

상태	내용	상태	내용
q_0	(I_1, I_2, E)	q_4	(I_1, I_2, F)
q_1	(I_1, W_2, E)	q_5	(I_1, W_2, F)
q_2	(W_1, I_2, E)	q_6	(W_1, I_2, F)
q_3	(W_1, W_2, E)	q_7	(W_1, W_2, F)

이 경우 앞서 알고리즘을 적용하면 $K_0=K$, $K_2=K_3=\cdots=K_1$이므로 $K^{\uparrow}=K_1$이 되고 $K^{\uparrow}$의 인식기는 그림 5와 같다.

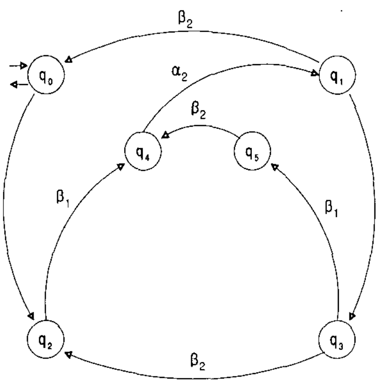

그림 5. $K^{\uparrow}$의 인식기.

표 2. 관리제어기능.

상태	허용사건	억제사건	상태	허용사건	억제사건
q_0	α_1	α_2	q_3	β_1,β_2	ϕ
q_1	α_1,β_2	ϕ	q_4	α_2	α_1
q_2	β_1	α_2	q_5	β_2	α_1

따라서 $K^{\uparrow}$을 토대로 앞서와 마찬가지로 관리제어기를 구현할 수 있다. 이 때 $K^{\uparrow}$의 인식기 각 상태에서의 관리제어기능은 표 2와 같다.

3. *K*가 관측불가능인 경우의 관리제어기 설계

이 절에서는 K가 관측불가능인 경우 관측가능성의 집합연산에 대한 성질들을 이용하여 K에 가장 근사하면서 관측가능인 초언어 및 관측가능성보다 강한 조건인 표준성(normality)[4]을 도입하여 K에 근사하면서 관측가능인 부언어 등을 유도하고 이를 토대로한 관리제어기의 설계에 관해 알아본다[4]-[7]. 또한 K가 제어불가능이면서 관측불가능인 경우의 관리제어기 설계에 대해 결과를 확대 해석한다.

3-1 관측가능성의 성질 및 관리제어기의 설계

관측가능성이 제어가능성에 비해 특이한 차이점은 합집합 연산에 대해 닫혀있지 않다는 데에 있다. 다음의 예를 살펴보자.

[예] $\Sigma=\Sigma_c=\Sigma_o\cup\Sigma_{uo}=\{\beta\}\cup\{\alpha\}$, $M=\alpha+\beta+\alpha\beta$로 주어졌을 때 $K_1=\alpha$, $K_2=\beta$는 각각 관측가능이다. 그러나 $K=K_1\cup K_2=\alpha+\beta$ 는 $s=\varepsilon$, $s'=\alpha$, $\sigma=\beta$라 하면 $s\sigma=\beta\in\overline{K}$이고 $s'\in\overline{K}$이며 $s'\sigma=\alpha\beta\in M$이지만 $s'\sigma=\alpha\beta\notin\overline{K}$ 이므로 $(s, \sigma, s')\notin \mathrm{NEXTACT}_{K, M}$ 이어서 관측불가능이다.

위와 같이 관측가능성이 합집합 연산에 대해 닫혀있지 않기 때문에 관측불가능 언어 K로부터 최고 관측가능 부언어(supremal observable sublanguage) $K^{o\uparrow}$를 체계적으로 구하기가 어렵다. 그러면 먼저 다음의 부분관측 상황하에서 관리제어기 설계문제의 유형을 살펴보자.

유형 1. 부분관측하에서의 기본 관리제어기의 설계문제: 이산사건시스템 G에 대해 $\Sigma_{uc}\subseteq\Sigma$, $\Sigma_o\subseteq\Sigma$와 이에 해당하는 투영사상 $P:\Sigma^*\mapsto\Sigma_o^*$, 그리고 정상언어 $L_a=\overline{L_a}\subseteq L(\mathrm{G})$가 주어졌을 때

1) $L(S_p/\mathrm{G})\subseteq L_a$ 이고,
2) 가능한 한 $L(S_p/\mathrm{G})$가 최대가 되도록 (즉, $L(S_p'/\mathrm{G})\subseteq L_a$인 어떠한 S_p'에 대해서도 $L(S_p'/\mathrm{G})\subseteq L(S_p/\mathrm{G})$가 되도록) P-관리제어기 S_p를 설계하라.

유형 2. 부분관측하에서의 비막힘성 관리제어기 설계문제: 이산사건시스템 G에 대해 $\Sigma_{uc}\subseteq\Sigma$, $\Sigma_o\subseteq\Sigma$와 이에 해당하는 투영사상 $P:\Sigma^*\mapsto\Sigma_o^*$, 그리고 $L_m(\mathrm{G})$-닫힘성의 정상표기 언어 $L_{am}\subseteq L_m(\mathrm{G})$가 주어졌을 때

1) $L_m(S_p/\mathrm{G})\subseteq L_{am}$ 이고,
2) 가능한 한 $L(S_p/\mathrm{G})$가 최대가 되도록 (즉, $L_m(S_p'/\mathrm{G})\subseteq L_{am}$인 어떠한 S_p'에 대해서도 $L(S_p'/\mathrm{G})\subseteq L(S_p/\mathrm{G})$가 되도록) 비막힘성 P-관리제어기 S_p를 설계하라.

앞서 언급한 바와 같이 $K^{o\uparrow}$를 체계적으로 구하기가 어렵기 때문에 위와 같은 관리제어기 설계문제의 일반적인 해도 구하기가 어렵다. 따라서 위 문제에 접근하기 위한 절차를 정리하면 다음과 같다 :

(1) 먼저 경우에 따라 L_a(또는 L_{am})의 부언어이며 관측가능이고 제어가능인 것들 가운데 최대집합이 계산가능하면 이를 구한다.
(2) 위 (1)의 상황을 특징지워 유사한 경우 같은 방식을 적용한다.
(3) 관측가능성보다 강한 조건을 인가하여 합집합 연

산에 대해 닫혀있는 새로운 성질을 찾아내어 이를 만족시키는 최대 부언어를 체계적으로 구하는 방법을 유도한다.

위 접근방식 가운데 다음 부절(subsection)에서는 보다 일반적인 (3)의 방식에 대해 자세히 논한다.

관측가능성이 합집합 연산에 대해 닫혀있지 않는 반면, 닫힌 언어의 경우 교집합 연산에 대해 닫혀있어 관측불가능 언어 K가 주어졌을 때 최저 닫힘 관측가능 초언어(infimal closed and observable superlanguage) $K^{o\downarrow}$를 바탕으로 P-관리제어기를 용이하게 설계할 수 있다. 먼저 관측가능성에 대한 다음 성질을 생각해 보자.

명제 3.1[5], [6]: K_1, K_2 가 각각 닫힌 언어이고 M, Σ_o, Σ_c 에 대해 관측가능이면 $K_1 \cap K_2$도 관측가능이다.

이 경우 먼저 $K := K_1 \cap K_2$ 라고 하자. 그리고 이 명제의 결론을 부정하여 K가 관측불가능이라고 하면 $P(s) = P(s')$인 문자열 s, s'과 $\sigma \in \Sigma_c$가 존재하여 $(s\sigma \in \overline{K}) \wedge (s' \in \overline{K}) \wedge (s'\sigma \in M - \overline{K})$이지만 $s'\sigma \in M - \overline{K}$를 만족시킨다. 즉, $(s'\sigma \in M) \wedge (s'\sigma \notin \overline{K_1} \vee s'\sigma \notin \overline{K_2})$이 된다. 만일 $s'\sigma \notin \overline{K_1}$ 이면, $(s\sigma \in \overline{K_1}) \wedge (s' \in \overline{K_1}) \wedge (s'\sigma \in M - \overline{K_1})$, 즉 $(s, \sigma, s') \notin NEXTACT_{K_1, M}$ 이므로 K_1은 관측불가능이며 이는 가정에 위배되므로 모순이고, 만일 $s'\sigma \notin \overline{K_2}$ 이면 마찬가지로 K_2가 관측불가능이 되어 역시 모순이 성립한다. 따라서 K는 관측가능임을 알 수 있다.

다음과 같은 언어부류를 생각해보자.

$$CO_{out}(K) := \{K \subseteq L \subseteq M \mid (L = \overline{L}) \wedge (L\text{은 관측가능})\}.$$

이 부류의 모든 언어는 교집합 연산에 대해 닫혀 있으며 앞서 $\Omega^{\uparrow}$연산자의 고정점과 마찬가지로 이러한 교집합 연산을 반복 수행하면 집합 내의 유일한 언어에 수렴하게 된다. 이 언어가 최저 닫힘 관측가능 초언어 $K^{O\downarrow}$ 이다. 최악의 경우 $K^{O\downarrow} = M$ 이고 K가 관측가능이면 $K^{O\downarrow} = \overline{K}$ 가 된다. 만일 K가 제어불가능이면서 관측불가능이면 이러한 연산을 통해서 관리제어기를 설계하기 위해서는 $K^{CIO\downarrow}$, 즉 최저 닫힘 제어가능 및 관측가능 초언어를 구해야 한다. 다음 관리제어기 설계문제의 한 유형을 살펴보자.

유형 3. 부분관측하에서의 기본 관리제어기 설계의 이원문제: 이산사건시스템 G에 대해 $\Sigma_{uc} \subseteq \Sigma$, $\Sigma_o \subseteq \Sigma$와 이에 해당하는 투영사상 $P : \Sigma^* \mapsto \Sigma_o^*$, 그리고 $L_m(G)$-닫힘성의 최소 사양 표기언어 $L_{min} \subseteq L_m(G)$가 주어졌을 때

1) $L_m(S_p/G) \supseteq L_{min}$ 이고,
2) 가능한 한 $L(S_p/G)$가 최소가 되도록 (즉, $L_m(S_p'/G) \supseteq L_{min}$ 인 어떠한 S_p'에 대해서도 $L(S_p'/G) \supseteq L(S_p/G)$가 되도록) P-관리제어기 S_p를 설계하라.

이러한 유형 3의 관리제어기 설계문제의 해는 $L(S_p/G) = L_{min}^{CIO\downarrow}$가 되도록 하는 S_p를 찾는 것이 된다.

그러면 $L(S_p/G) = L_{min}^{CIO\downarrow} \cap L_m(G) \supseteq \overline{L}_{min} \cap L_m(G) = L_{min}(G)$가 성립되어 위 조건을 만족시키게 된다.

3-2 표준성과 관리제어기 설계

앞서 관측가능성이 합집합 연산에 대해 닫혀 있지 않아 관측불가능한 K로부터 $K^{O\uparrow}$를 체계적으로 구할 수 없음을 알았다. 이 부절에서는 이러한 관측가능성에 더 많은 제약조건을 인가하여 합집합 연산에 대해 닫혀있는 새로운 성질을 생각해본다. 이러한 성질을 언어의 표준성(normality)이라고 한다[4], [5].

정의 3.1(표준성): $M = \overline{M} \subseteq \Sigma^*$와 투영사상 $P : \Sigma^* \mapsto \Sigma_o^*$가 주어졌을 때 언어 $K \subseteq M$이 $\overline{K} = P^{-1}[P(\overline{K})] \cap M$을 만족시키면 P와 M에 대해 표준(normal)이라고 한다.

다음 명제 3.2는 이러한 언어의 표준성이 관측가능성을 내포함을 보여준다.

명제 3.2[4], [5]: 만일 언어 $K \subseteq M$이 P와 M에 대해 표준이면 K는 M, P, 그리고 모든 $\Sigma_o \subseteq \Sigma$에 대해 관측가능이다.

명제 3.2의 경우 $\Sigma_c = \Sigma$이라 하고 K는 표준이지만 관측불가능이라고 가정하자. 그러면 $P(s) = P(s')$인 문자열 s, s'과 $\sigma \in \Sigma$에 대해 $(s\sigma \in \overline{K}) \wedge (s' \in \overline{K}) \wedge (s'\sigma \in M - \overline{K})$가 성립한다. 이 때 $P(s'\sigma) = P(s\sigma) \in P(\overline{K})$이므로 $P(s'\sigma) \in P(\overline{K})$이고, $s'\sigma \in M$ 과 $s'\sigma \in P^{-1}[P(\overline{K})] \cap M$으로부터 $s'\sigma \in \overline{K}$이 되어 모순이 된다. 따라서 K는 관측가능임을 알 수 있다. 그러나 일반적으로 위 명제 3.2의 역은 성립되지 않는다. 다음의 반례를 살펴보자.

[예] $\Sigma = \{\alpha, \beta\}$, $\Sigma_o = \{\beta\}$, $\Sigma_c = \{\alpha\}$, $M = \overline{\beta\alpha}$, 그리고 $K = \beta$주어진 경우 K는 관측가능이지만 $P^{-1}[P(K)] \cap M = M \supset \overline{K}$이므로 표준이 아님을 알 수 있다.

다음 명제 3.3은 이러한 언어의 표준성이 합집합 연산에 대해 닫혀 있음을 보여준다.

명제 3.3[4], [5]: 만일 언어 K_1, $K_2 \subseteq M$ 이 P와 M에 대해 표준이면 $K_1 \cup K_2$도 P와 M에 대해 표준이다.

이 경우 $P^{-1}[P(\overline{K_1 \cup K_2})] \cap M = P^{-1}[P(\overline{K_1} \cup \overline{K_2})] \cap M = P^{-1}[P(\overline{K_1}) \cup P(\overline{K_2})] \cap M = [P^{-1}[P(\overline{K_1})] \cap M] \cup [P^{-1}[P(\overline{K_2})] \cap M] = \overline{K_1} \cup \overline{K_2} = \overline{K_1 \cup K_2}$ 이므로 위와 같이 언어의 표준성은 합집합 연산에 대해 닫혀있음을 알 수 있다.

언어 $K \subseteq M$에 대해 다음과 같은 부류의 언어를 생각해 보자.

$$N_{in}(K) := \{L \subseteq K \mid \overline{L} = P^{-1}[P(\overline{L})] \cap M\}.$$

그러면 위 명제 3.3으로부터 이 부류의 언어는 합집합 연산에 대해 닫혀있으므로 부분집합의 합집합가운데 최대집합이 존재하면 유일하게 주어진다. 이러한 최고 표준 부언어(supremal normal sublanguage)를 $K^{N\uparrow}$라고

하면 최악의 경우 $K^{N\uparrow}=\phi$이고 K가 표준이면 $K^{N\uparrow}=K$가 된다. 만일 K가 닫힌 언어이면 $K^{N\uparrow}$는 다음 식으로부터 구해질 수 있다[4].

$$K^{N\uparrow} = K - [P^{-1} [P (M-K)]]\Sigma^{*}.$$

또한 앞절의 최고 제어가능 부언어 $K^{\uparrow}$를 격자상의 연산자 $\Omega^{\uparrow}$에 대한 최대 고정점으로부터 유도하였듯이 $K^{N\uparrow}$를 구하는 일반적인 알고리즘을 비슷한 방식으로 유도해낼 수 있다[4]. 이러한 알고리즘은 특히 K와 M이 정규언어이면 유한단계이내에 수렴하여 이 경우 $K^{N\uparrow}$도 정규언어가 된다.

명제 3.4〔4〕, 〔5〕:

1) 만일 K가 닫힌 언어이면, $K^{N\uparrow}$도 닫힌 언어이고,
2) 만일 $K \subseteq L_m(G)$가 $L_m(G)$-닫힘성을 지니면, $K^{N\uparrow}$도 $L_m(G)$-닫힘성을 지닌다.

위 명제 3.4의 의미를 살펴보자. 먼저 1)의 경우 $K^{N\uparrow}$은 표준이고 $\overline{K^{N\uparrow}}$도 표준이며 $\overline{K^{N\uparrow}} \subseteq \overline{K} = K$이므로 $\overline{K^{N\uparrow}} \subseteq K^{N\uparrow}$이고 또한 $\overline{K^{N\uparrow}} \supseteq K^{N\uparrow}$이므로 $\overline{K^{N\uparrow}} = K^{N\uparrow}$이다. 따라서 $K^{N\uparrow}$는 닫힌 언어임을 알 수 있다. 다음 2)의 경우 먼저 $K^{N\uparrow} \subseteq \overline{K^{N\uparrow}} \cap L_m(G)$은 자명하다. 따라서 $K^{N\uparrow} \supseteq \overline{K^{N\uparrow}} \cap L_m(G)$이 성립되는지 살펴보자. $K' := \overline{K^{N\uparrow}} \cap L_m(G)$라고 하면, $K^{N\uparrow} \subseteq \overline{K^{N\uparrow}} \cap L_m(G)$이므로 $\overline{K^{N\uparrow}} \subseteq \overline{K^{N\uparrow} \cap L_m(G)} = \overline{K'}$가 된다. 그리고 $\overline{K^{N\uparrow}} \cap L_m(G) \subseteq \overline{K^{N\uparrow}}$으로부터 $\overline{\overline{K^{N\uparrow}} \cap L_m(G)} \subseteq \overline{K^{N\uparrow}}$이고, 그래서 $\overline{K'} \subseteq \overline{K^{N\uparrow}}$가 된다. 즉, $\overline{K'} \subseteq \overline{K^{N\uparrow}}$이며 $K' \subseteq \overline{K} \cap L_m(G) = K$이므로 $K' \subseteq K^{N\uparrow}$이다. 결국 $\overline{K^{N\uparrow}} \cap L_m(G) \subseteq K^{N\uparrow}$이고, 따라서 $\overline{K^{N\uparrow}} \cap L_m(G) = K^{N\uparrow}$가 되어 $K^{N\uparrow}$도 $L_m(G)$-닫힘성을 지님을 알 수 있다.

앞서 언급한 바와 같이 만일 K가 제어불가능이면서 관측불가능이면 최고 제어가능 및 관측가능 부언어 (supremal controllable and observable sublanguage) $K^{C\uparrow O\uparrow}$를 구하여 이를 토대로 관리제어기를 설계할 수 있지만 관측가능성이 합집합 연산에 대해 닫혀있지 않아서 일반적으로 $K^{C\uparrow O\uparrow}$를 구할 수가 없다. 이 경우 $K^{C\uparrow O\uparrow}$ 대신 최고 제어가능 및 표준 부언어(supremal controllable and normal sublanguage)를 $K^{C\uparrow N\uparrow}$라고 하면 $K^{C\uparrow N\uparrow}$는 관측가능성의 측면에서 최적의 해는 아니지만 제어가능성과 표준성이 모두 합집합 연산에 대해 닫혀 있기 때문에 계산가능한 해를 제공하게 된다. 이를테면, $L_m(G)$-닫힘성의 정상 표기언어 $L_{am} \subseteq L_m(G)$가 주어졌을 때 $L(S_p/G)=\overline{L_{am}^{C\uparrow N\uparrow}}$인 P-관리제어기 S_p를 설계하면 명제 3.4로부터 $L_m(S_p/G)=\overline{L_{am}^{C\uparrow N\uparrow}} \cap L_m(G)=L_{am}^{C\uparrow N\uparrow}$이므로 S_p는 비막힘성이며 주어진 L_{am}내에서 제어가능이며 관측가능인 언어를 생성토록 한다. 즉, 일반적으로 $K \supseteq K^{C\uparrow O\uparrow} \supseteq K^{C\uparrow N\uparrow}$의 관계를 만족시키게 된다. 한편, 실제로 $K^{C\uparrow N\uparrow}$을 구하는 과정에 있어서는 먼저 $K^{C\uparrow}=K'$을 구한 뒤 $(K')^{N\uparrow}=K''$을 구하더라도 K''이 관측불가능일 수 있으므로 다시 이 과정들을 반복하여 두 가지 조건을 모두 만족시킬 때까지 반복수행을 통해 구하게 된다. 앞서 언어의 표준성은 곧 관측가능성을 내포하지만 그 역은 성립되지 않음을 알았다 (명제 3.2의 역이 성립되지 않는 반례를 살펴보았다). 그러나 다음 명제 3.5는 특정한 경우, 즉 $\Sigma_c \subseteq \Sigma_o$일 때 제어가능이면서 관측가능이면 동시에 표준임을 보여준다.

명제 3.5〔5〕, 〔6〕: 만일 $\Sigma_c \subseteq \Sigma_o$이고 K가 M과 Σ_{uc}에 대해 제어가능이며 M, P 그리고 Σ_c에 대해 관측가능이면 K는 M과 P에 대해 표준이다.

이 경우 명제의 조건이 성립될 때 $K \neq \phi$가 표준이 아니라고 가정해 보자. 그러면 $t \in M$, $t \neq \varepsilon$이 존재하여 $(t \notin \overline{K}) \wedge (P(t) \in P(\overline{K}))$를 만족시킨다. $s'\sigma$를 그러한 문자열 t라고 하자. 즉, $(s' \in \overline{K}) \wedge (s'\sigma \in M) \wedge (s'\sigma \notin \overline{K}) \wedge (P(s'\sigma) \in P(\overline{K}))$이라고 하자. 이 때 K는 제어가능이므로 $\overline{K}\Sigma_{uc} \cap M \subseteq \overline{K}$이고 따라서 $\sigma \in \Sigma_c \subset \Sigma_o$이다. 결국 $(\sigma \in \Sigma_o) \wedge (P(s'\sigma) \in P(\overline{K}))$이므로 $P(s) = P(s')$인 $s\sigma \in \overline{K}$가 존재하게 되며 이러한 $s\sigma$에 대해 $(s\sigma \in \overline{K}) \wedge (s' \in \overline{K}) \wedge (s'\sigma \in M)$이지만 $s'\sigma \notin \overline{K}$이므로 K는 관측불가능이 되어 가정에 위배된다. 따라서 모순이 성립되므로 K는 M과 P에 대해 표준임을 알 수 있다.

위 명제 3.5를 통해 $\Sigma_c \subset \Sigma_o$이면 $K^{C\uparrow N\uparrow}$이 곧 $K^{C\uparrow O\uparrow}$를 의미하게 되어 원래 구하고자 했던 제어가능이며 관측가능인 언어를 이러한 경우에는 제어가능이며 표준인 언어를 통해 구해낼 수 있음을 알았다.

4. 결론

본 논문에서는 이산사건시스템의 관리제어 (I): 이산사건모델링 및 관리제어이론[8]에서 소개한 이산사건시스템의 관리제어이론을 바탕으로 실제 상황하에서 주어진 관리제어시스템의 사양언어에 대한 관리제어기의 설계문제를 다루었다. 이를 위해 사양언어가 제어불가능이거나 관측불가능인 경우 각각에 대해 사양언어에 가장 근사하면서 제어가능이며 관측가능인 부언어 또는 초언어 등을 정의하였고 아울러 언어의 성질과 집합이론의 결과들을 이용해 그러한 언어들을 유도하는 여러 해석적 기법들을 소개하였다. 특히 관측가능성의 성질을 보완하기 위해 언어의 표준성이라는 새로운 개념을 도입해 사양언어가 제어불가능이면서 관측불가능인 경우의 관리제어기 설계로 결과를 확장하였다.

참고문헌

[1] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language", *SIAM J. of Control and Optimization*, vol. 25, pp. 637-659, 1987.

[2] R. Kumar and V. K. Garg, *Modeling and Control of Logical Discrete Event Systems*, MA: Kluwer