Extremal Solutions of Inequations over Lattices with Applications to Supervisory Control [1]

Ratnesh Kumar
Department of Electrical Engineering
University of Kentucky
Lexington, KY 40506-0046
Email: kumar@engr.uky.edu

Vijay K. Garg
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78712-1084
Email: vijay@pine.ece.utexas.edu

[1]This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, in part by the National Science Foundation under Grant NSF-CCR-9110605, in part by a grant from IBM, and in part by a TRW faculty assistantship award.

Abstract

We study the existence and computation of extremal solutions of a system of inequations defined over lattices. Using the Knaster-Tarski fixed point theorem, we obtain sufficient conditions for the existence of supremal as well as infimal solution of a given system of inequations. Iterative techniques are presented for the computation of the extremal solutions whenever they exist, and conditions under which the termination occurs in a single iteration are provided. These results are then applied for obtaining extremal solutions of various inequations that arise in computation of maximally permissive supervisors in control of logical discrete event systems (DESs) first studied by Ramadge and Wonham. Thus our work presents a unifying approach for computation of supervisors in a variety of situations.

Keywords: Fixed points, lattices, inequations, discrete event systems, supervisory control, language theory.

1 Introduction

Given a set X and a function $f : X \to X$, $x \in X$ is called a *fixed* point of the function if $f(x) = x$. Existence and computation of fixed points of functions defined over lattices have been studied in computer science literature for applications such as theory of recursive functions, program termination, algorithm design, etc., [24]. Lattices are partially ordered sets with the property that least upper bound and greatest lower bound of any pair of lattice elements is defined. A commonly encountered example of lattices is a power set—the set of all subsets of a given set—together with the containment partial order.

One of the initial results on extremal fixed points of functions defined over lattices is due to Knaster-Tarski [31]. It states that every monotone function possesses an infimal as well as a supremal fixed point. Another result provides methods of computing the infimal and supremal fixed points under stronger conditions than monotonicity. (Refer to Theorems 1 and 2 in section 2.) The paper by Lassez-Nguyen-Sonenberg [18] provides a nice historical account of these fixed point theorems. Several other fixed point results have since been discovered and are reported in papers such as [4, 23, 30, 9, 1, 6]. The notion of optimal fixed points and their properties are discussed in [22, 17].

In this paper we study the existence and computation of extremal solutions of a system of inequations $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ defined over lattices, where $x \in X$ is the variable of inequations and $n \in \mathcal{N}$ is a fixed number. Since a fixed point equation $f(x) = x$ can be written as a pair of inequations, it is clear that the computation of extremal fixed points of a certain function is a special case of that of computation of extremal solutions of a system of inequations. We show that the converse is also true. We use the two fixed point theorems informally described above for determining the existence and computation of extremal solutions of inequations.

Our interest in studying the extremal solutions of a system of inequations stems from computations of supervisors in control of logical behavior of DESs. DESs are systems that involve quantities that are discrete and which evolve according to the occurrence of certain discrete qualitative changes, called *events*, in the system. At the *logical* level of abstraction [7], the behavior of a DES can be described using the set of all possible sequences of events that it can execute [33, 8]. Thus the space of logical behaviors of DESs is a certain power set, and so it can be studied from a lattice theoretic perspective.

The framework of supervisory control was introduced by Ramadge and Wonham [27, 28] for developing the techniques for controlling the qualitative behavior of such systems. A supervisor in this setting is event driven, and dynamically disables some of the events from occurring so that certain *desired* or *target* behavior constraint is satisfied. It is desirable that such a supervisor be *maximally permissive* so that a maximal behavior satisfying the desired behavior constraint is achieved under control. Computation of such supervisors requires computation of extremal solutions of a certain system of inequations defined over the power set lattice of behaviors of DESs.

We first introduce the notions of dual, co-dual, inverse, and converse of a function and study their properties. Some of these terminology is taken from the work of Dijkstra-Scholten

[5] who first investigated these concepts in the setting of predicates and predicate transformers. Kumar-Garg-Marcus [15] applied some of this work to supervisory control of DESs represented as programs consisting of a finite number of conditional assignment statements. In this paper, we further extend these work and apply it for obtaining conditions under which supremal and infimal solutions of a system of inequations $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ exists. We also provide iterative techniques for computing the solutions whenever they exist, and present conditions under which termination occurs in a single iteration. These techniques are then used for computation of maximally permissive supervisors in a variety of settings.

Rest of the paper is organized as follows: Section 2 introduces several concepts from lattice theory, and reviews basic results on extremal fixed points. In section 3 we define the notions of dual, co-dual, conjugate, inverse, and converse of a function, and study some of their properties. Section 4 studies existence and computation of extremal solutions of a given system of inequations. Each result presented in sections 3 and 4 has a "*primal*" and a "*dual*" version; we *only* present a proof for the primal version, as the dual version can be proved analogously. These results are applied in section 5 to supervisory control of DESs. In section 6 we conclude the work presented here, and in Appendix A we give an alternative interpretation for the inverse operation.

2 Notation and Preliminaries

In this section we introduce the relevant notations and concepts of lattice theory and review some of the basic results on existence and computation of extremal fixed points. Given a set X, a *partial order relation*, denoted $\leq$, over X is a reflexive, anti-symmetric and transitive relation. For $x, y \in X$ if $x \leq y$, then x is said to be *smaller* than y, and y is said to be *greater* than x. $\leq$ is said to be *total order* if for each $x, y \in X$, either $x \leq y$ or $y \leq x$. The containment relation defined on a power set is an example of a partial order which is not a total order.

Definition 1 The pair $(X, \leq)$, where X is a set and $\leq$ is a partial order over X, is called a *partially ordered set* or a *poset*. A totally ordered subset of X is called a *chain*.

Given $Y \subseteq X$, $x \in X$ is said to be *supremal* of Y if

- (upper bound): $\forall y \in Y : y \leq x$, and
- (least upper bound): $\forall z \in X : [\forall y \in Y : y \leq z] \Rightarrow [x \leq z]$.

It is easy to check that the supremal of Y is unique whenever it exists. The notation $sup Y$ is used to denote the supremal of Y. We also use $\sqcup Y$ to denote $sup Y$. In particular, given $x, y \in X$, $x \sqcup y$ is used to denote $sup\{x, y\}$.

Similarly, $x \in X$ is called *infimal* of $Y \subseteq X$ if

- (lower bound): $\forall y \in Y : x \leq y$, and
- (greatest lower bound): $\forall z \in X : [\forall y \in Y : z \leq y] \Rightarrow [z \leq x]$.

It is easy to check that the infimal of Y is unique whenever it exists. The notation $\mathit{inf}\, Y$ is used to denote the infimal of Y. We also use $\sqcap Y$ to denote $\mathit{inf}\, Y$. In particular, given $x, y \in X$, $x \sqcap y$ is used to denote $\mathit{inf}\,\{x, y\}$. For example, consider the interval $[0, 2) := \{x \in \mathcal{R} \mid 0 \leq x < 2\}$. Then $\mathit{sup}\,[0, 2) = 2$ and $\mathit{inf}\,[0, 2) = 0$. Note that it follows from the above the definitions that $\mathit{sup}\,\emptyset = \mathit{inf}\, X$ (whenever it exists), and $\mathit{inf}\,\emptyset = \mathit{sup}\, X$ (whenever it exists).

Definition 2 A poset $(X, \leq)$ is said to be a *lattice* if $\mathit{sup}\, Y, \mathit{inf}\, Y \in X$ for any finite $Y \subseteq X$. If $\mathit{sup}\, Y, \mathit{inf}\, Y \in X$ for arbitrary $Y \subseteq X$, then $(X, \leq)$ is called a *complete lattice*. If $\mathit{inf}\, X \in X$, and $\mathit{sup}\, Y \in X$ for any *chain* $Y \subseteq X$, then $(X, \leq)$ is called a *complete partial order (cpo)*.

Example 1 Given a set X, $(2^X, \subseteq)$—the set of all subsets of X together with the containment partial order—is called the *power set lattice* of X. A power set lattice is an example of a complete lattice. The set of natural numbers with the natural ordering is an example of a chain which is not a cpo.

As mentioned in introduction, at the qualitative or *logical* level of abstraction, the behavior of a DES is described using the set of all possible sequences of events that it can execute. Let Σ denote the set of events that can occur in a DES, then the notation Σ^* denotes the set of all finite length sequences of events, including the zero length sequence ϵ. Logical behavior of a DES is a subset of Σ^*, also called a *language*. Each member of Σ^* is called a *string* or a *trace*. In section 5 we study extremal solutions of inequations defined over the power set lattice $(2^{\Sigma^*}, \subseteq)$—the set of all languages together with the containment partial order.

We next define a few useful properties of functions defined over lattices. Given a poset $(X, \leq)$, a function $f : X \to X$ is said to be *idempotent* if

$$\forall x \in X : f(x) = f(f(x));$$

it is said to be *monotone* if

$$\forall x, y \in X : [x \leq y] \Rightarrow [f(x) \leq f(y)].$$

Given a complete lattice $(X, \leq)$, a function $f : X \to X$ is said to be *disjunctive* if

$$\forall Y \subseteq X : f(\sqcup_{y \in Y} Y) = \sqcup_{y \in Y} f(y);$$

it is said to be *conjunctive* if

$$\forall Y \subseteq X : f(\sqcap_{y \in Y} Y) = \sqcap_{y \in Y} f(y).$$

It is readily verified that disjunctive and conjunctive functions are also monotone. Note that since $\mathit{sup}\,\emptyset = \mathit{inf}\, X$ and $\mathit{inf}\,\emptyset = \mathit{sup}\, X$, by setting $Y = \emptyset$ in the last two definitions we obtain for a disjunctive function that $f(\mathit{inf}\, X) = \mathit{inf}\, X$, and for a conjunctive function that $f(\mathit{sup}\, X) = \mathit{sup}\, X$.

Example 2 Consider the power set lattice of languages defined over the event set Σ. The *prefix closure* operation $pr : 2^{\Sigma^*} \to 2^{\Sigma^*}$ is defined as: Given $K \subseteq \Sigma^*$, $pr(K) \subseteq \Sigma^*$ is the set of prefixes of strings belonging to K, i.e.,

$$pr(K) := \{s \in \Sigma^* \mid \exists t \in K \text{ s.t. } s \leq t\},$$

where the notation $s \leq t$ is used to denote that the string s is a prefix of string t. Clearly, $K \subseteq pr(K)$; K is said to be *prefix closed* if $K = pr(K)$. The extension closure operation $ext : 2^{\Sigma^*} \to 2^{\Sigma^*}$ is defined as : Given $K \subseteq \Sigma^*$, $ext(K) \subseteq \Sigma^*$ is the set of extensions of strings in K, i.e.,

$$ext(K) := \{s \in \Sigma^* \mid \exists t \in K \text{ s.t. } t \leq s\}.$$

Clearly, $K \subseteq ext(K)$; K is said to be *extension closed* if $K = ext(K)$.

Some other interesting functions are the concatenation and the quotient operations (with a fixed language) defined as follows: Given a fixed language $H \subseteq \Sigma^*$, and a language $K \subseteq \Sigma^*$, the *concatenation* of K with H, denoted KH, is the language obtained by concatenating strings from K and H, i.e.,

$$KH := \{st \in \Sigma^* \mid s \in K, t \in H\};$$

and the K *quotient* with H, denoted K/H, is the language obtained by removing suffixes belonging to H from strings in K, i.e.,

$$K/H := \{s \in \Sigma^* \mid \exists t \in H \text{ s.t. } st \in K\}.$$

It is easily verified that for a language $K \subseteq \Sigma^*$, $pr(K) = K/\Sigma^*$ and $ext(K) = K\Sigma^*$. Thus prefix closure operation is an example of the quotient operation and extension closure operation is an example of concatenation operation. It can be checked that both concatenation and quotient operations are disjunctive and thus monotone, however, none of them are conjunctive. The prefix and extension closure operations are both idempotent.

The following fixed point theorem is due to Knaster and Tarski:

Theorem 1 [4, Theorem 4.11] Let $(X, \leq)$ be a complete lattice and $f : X \to X$ be a monotone function. Let $Y := \{x \in X \mid f(x) = x\}$ be the set of fixed points of f. Then

1. $\mathit{inf}\, Y \in Y$, and $\mathit{inf}\, Y = \mathit{inf}\, \{x \in X \mid f(x) \leq x\}$.

2. $\mathit{sup}\, Y \in Y$, and $\mathit{sup}\, Y = \mathit{sup}\, \{x \in X \mid x \leq f(x)\}$.

It follows from Theorem 1 that a monotone function defined over a complete lattice always has an infimal and a supremal fixed point. The following theorem provides a technique for computing such fixed points under stronger conditions. We first define the notion of disjunctive and conjunctive closure. Given a complete lattice $(X, \leq)$, and a function $f : X \to X$, the *disjunctive closure* of f, denoted f^*, is the map $f^* : X \to X$ defined as:

$$\forall x \in X : f^*(x) := \sqcup_{i \geq 0} f^i(x);$$

and the *conjunctive closure* of f, denoted f_*, is the map $f_* : X \to X$ defined as:

$$\forall x \in X : f_*(x) := \sqcap_{i \geq 0} f^i(x),$$

where f^0 is defined to be the identity function, and for each $i \geq 0$, $f^{i+1} := ff^i$. It is easy to see that the disjunctive as well conjunctive closures of f are idempotent.

Theorem 2 [4, Theorem 4.5] Let $(X, \leq)$ be a complete lattice and $f : X \to X$ be a function. Let $Y := \{x \in X \mid f(x) = x\}$ be the set of fixed points of f.

1. If f is disjunctive, then $\mathit{inf}\, Y = f^*(\mathit{inf}\, X)$.

2. If f is conjunctive, then $\mathit{sup}\, Y = f_*(\mathit{sup}\, X)$.

Remark 1 In the first part of Theorem 2, it is possible to compute the infimal fixed point of a function in a weaker setting when it is defined over a cpo (not necessarily a complete lattice) and it commutes with the supremal operation taken over a countable chain (so it is not necessarily disjunctive) [4].

3 Dual, Co-Dual, Inverse, and Converse Operations

In this section we develop the notion of dual, co-dual, inverse, and converse operations and study some of their properties. These concepts are used in the next section for obtaining extremal solutions of a system of inequations. We begin by providing conditions for existence of extremal solutions of simple inequations.

Lemma 1 [5] Consider a complete lattice $(X, \leq)$ and functions $f, g : X \to X$.

1. If f is disjunctive, then the supremal solution of the inequation $f(x) \leq y$, in the variable x, exists for each $y \in X$.

2. If g is conjunctive, then the infimal solution of the inequation $y \leq g(x)$, in the variable x, exists for each $y \in X$.

Proof: Since $(X, \leq)$ is complete, $\mathit{inf}\, X \in X$, and by definition $\mathit{inf}\, X \leq y$. Using the disjunctivity of f we obtain $f(\mathit{inf}\, X) = \mathit{inf}\, X \leq y$. Thus the set of solutions of the inequation $f(x) \leq y$ is nonempty. Let I be an indexing set such that for each $i \in I$, $x_i \in X$ is a solution of the inequation $f(x) \leq y$. Then it suffices to show that $\sqcup_{i \in I} x_i$ is also a solution of the inequation. Since $(X, \leq)$ is complete, it follows that $\sqcup_{i \in I} x_i \in X$. Also, $f(\sqcup_{i \in I} x_i) = \sqcup_{i \in I} f(x_i) \leq y$, where the equality follows from the fact that f is disjunctive and the inequality follows from the fact that $f(x_i) \leq y$ for each $i \in I$. $\blacksquare$

Lemma 1 can be used to define the notion of dual of a disjunctive function and co-dual of a conjunctive function.

Definition 3 Consider a complete lattice $(X, \leq)$ and functions $f, g : X \to X$. If f is disjunctive, then its *dual*, denoted $f^{\perp}(\cdot)$, is defined to be the supremal solution of the inequation $f(x) \leq (\cdot)$. If g is conjunctive, then its *co-dual*, denoted $g^{\top}(\cdot)$, is defined to be the infimal solution of the inequation $(\cdot) \leq g(x)$. ($x \in X$ is the variable of the inequation.)

Example 3 Consider the power set lattice of languages defined over the event set Σ and the prefix and extension closure operations. Since these operations are disjunctive, their dual exist. It follows from the definition of duality that for $K \subseteq \Sigma^*$, $pr^{\perp}(K)$ is the supremal language whose prefix closure is contained in K. Thus $pr^{\perp}(K)$ is the *supremal prefix closed sublanguage* of K, which we denote as $sup P(K)$. Similarly, $ext^{\perp}(K)$ is the *supremal extension closed* sublanguage of K, which we denote as $sup E(K)$.

The following proposition provides an alternative definition of duality as well as of co-duality.

Proposition 1 Consider a complete lattice $(X, \leq)$, a disjunctive function $f : X \to X$, and a conjunctive function $g : X \to X$. Then the following are equivalent.

1. $f^{\perp} = g$.
2. $\forall x, y \in X : [f(x) \leq y] \Leftrightarrow [x \leq g(y)]$.
3. $g^{\top} = f$.

Proof: We only prove the equivalence of the first and the second assertion; the equivalence of the second and the third assertion can be proved analogously. Since f is disjunctive, $f^{\perp}$ is defined. Suppose the first assertion is true. In order to see the forward implication of the second assertion, suppose $f(x) \leq y$, which implies x is a solution of the inequation. Since $f^{\perp}(y) = g(y)$ is the supremal solution of the inequation, it follows that $x \leq g(y)$. Next in order to see the backward implication, suppose $x \leq g(y)$. So from monotonicity of f we obtain that $f(x) \leq f(g(y))$. Since $g(y) = f^{\perp}(y)$ is a solution of the inequation, we have $f(g(y)) \leq y$. So $f(x) \leq f(g(y)) \leq y$, as desired.

Next suppose the second assertion holds. By setting $x = g(y)$ in the second assertion, we obtain that for all $y \in X$, $f(g(y)) \leq y$. This shows that $g(y)$ is solution of the inequation. Finally using the forward implication of the second assertion we conclude that if x is a solution of the inequation, then $x \leq g(y)$. This shows that $g(y)$ is the supremal solution of the inequation. So $g(y) = f^{\perp}(y)$ for all $y \in X$. ∎

Note that the equivalence of the first two assertions in Proposition 1 does not require g to be conjunctive. Hence if we replace g by $f^{\perp}$, then the first assertion is identically true; consequently, the second assertion is also identically true. Similarly it can be argued that the second assertion is identically true with f replaced by $g^{\top}$. This is stated in the following corollary.

Corollary 1 Consider a complete lattice $(X, \leq)$, and functions $f, g : X \to X$.

1. If f is disjunctive, then $\forall x, y \in X : [f(x) \leq y] \Leftrightarrow [x \leq f^{\perp}(y)]$.
2. If g is conjunctive, then $\forall x, y \in X : [g^{\top}(x) \leq y] \Leftrightarrow [x \leq g(y)]$.

Corollary 1 can be used to obtain several interesting properties of the dual and co-dual operations. We first show that dual of a disjunctive function is conjunctive, and co-dual of a conjunctive function is disjunctive.

Lemma 2 Consider a complete lattice $(X, \leq)$ and functions $f, g : X \to X$.

1. If f is disjunctive, then $f^{\perp}$ is conjunctive.
2. If g is conjunctive, then $g^{\top}$ is disjunctive.

Proof: Pick $Y \subseteq X$. We need to show that $f^{\perp}(\sqcap_{y \in Y} y) = \sqcap_{y \in Y} f^{\perp}(y)$. The forward inequality can be shown as follows:

$$\begin{aligned}
[f^{\perp}(\sqcap_{y\in Y} y) \leq f^{\perp}(\sqcap_{y\in Y} y)] &\Leftrightarrow [f(f^{\perp}(\sqcap_{y\in Y} y)) \leq \sqcap_{y\in Y} y] \\
&\Leftrightarrow [\forall y \in Y : f(f^{\perp}(\sqcap_{y\in Y} y)) \leq y] \\
&\Leftrightarrow [\forall y \in Y : f^{\perp}(\sqcap_{y\in Y} y) \leq f^{\perp}(y)] \\
&\Leftrightarrow [f^{\perp}(\sqcap_{y\in Y} y) \leq \sqcap_{y\in Y} f^{\perp}(y)],
\end{aligned}$$

where the first and the third equivalence follow from Corollary 1.

Next the reverse inequality can be obtained as follows:

$$\begin{aligned}
[\forall y \in Y : \sqcap_{y\in Y} f^{\perp}(y) \leq f^{\perp}(y)] &\Leftrightarrow [\forall y \in Y : f(\sqcap_{y\in Y} f^{\perp}(y)) \leq y] \\
&\Leftrightarrow [f(\sqcap_{y\in Y} f^{\perp}(y) \leq \sqcap_{y\in Y} y] \\
&\Leftrightarrow [\sqcap_{y\in Y} f^{\perp}(y) \leq f^{\perp}(\sqcap_{y\in Y} y)],
\end{aligned}$$

where the first and the final equivalence follow from Corollary 1. ■

It follows from Lemma 2 that it is possible to define co-dual of the dual of a disjunctive function and dual of the co-dual of a conjunctive function. The following proposition describes some other properties of dual and co-dual operations.

Proposition 2 Consider a complete lattice $(X, \leq)$, disjunctive functions $f, f_1, f_2 : X \to X$, and conjunctive functions $g, g_1, g_2 : X \to X$.

1. ($^{\perp}$ and $^{\top}$ inverses)	(a) $(f^{\perp})^{\top} = f$	(b) $(g^{\top})^{\perp} = g$
2. (composition)	(a) $(f_1 f_2)^{\perp} = f_2^{\perp}(f_1^{\perp})$	(b) $(g_1 g_2)^{\top} = g_2^{\top}(g_1^{\top})$
3. (idempotence)	(a) $[ff = f] \Leftrightarrow [f^{\perp}f^{\perp} = f^{\perp}]$	(b) $[gg = g] \Leftrightarrow [g^{\top}g^{\top} = g^{\top}]$

Proof: 1. Since f is disjunctive, it follows from Lemma 2 that $f^{\perp}$ is conjunctive, so $(f^{\perp})^{\top}$ is defined. By replacing g with $f^{\perp}$ in Proposition 1 we obtain from its third assertion that $(f^{\perp})^{\top} = f$, as desired.

2. Since disjunctivity is preserved under composition of functions, $f_1 f_2$ is disjunctive, so that its dual is defined. Fix $x, y \in X$. Then the repeated application of Corollary 1 yields the following series of equivalences:

$$\begin{aligned}[f_1 f_2(x) \leq y] &\Leftrightarrow [f_2(x) \leq f_1^{\perp}(y)] \\ &\Leftrightarrow [x \leq f_2^{\perp} f_1^{\perp}(y)].\end{aligned}$$

Since $f_2^{\perp} f_1^{\perp}$ is conjunctive (follows from Lemma 2, and the fact that conjunctivity is preserved under composition of functions), if we replace f by $f_1 f_2$ and g by $f_2^{\perp} f_1^{\perp}$ in Proposition 1, we obtain $(f_1 f_2)^{\perp} = f_2^{\perp} f_1^{\perp}$, as desired.

3. The forward implication can be shown as follows: $f^{\perp} = (ff)^{\perp} = f^{\perp} f^{\perp}$, where the first equality follows from hypothesis, and the second from part 2. The backward implication can be obtained as follows: $f = (f^{\perp})^{\top} = (f^{\perp} f^{\perp})^{\top} = ff$, where the first equality follows from part 1, the second from hypothesis, and the final from parts 2 and 1. ∎

Example 4 Consider the power set lattice of languages defined over the event set Σ. We showed in Example 3 that $pr^{\perp} = sup\, P$ and $ext^{\perp} = sup\, E$. Then it follows from Lemma 2 that $sup\, P$ as well as $sup\, E$ are conjunctive. Moreover, Proposition 2 implies that $(sup\, P)^{\top} = (pr^{\perp})^{\top} = pr$ and $(sup\, E)^{\top} = (ext^{\top})^{\perp} = ext$. Finally, since pr and ext are idempotent, it follows from Proposition 2 that $pr^{\perp} = sup\, P$ and $ext^{\top} = sup\, E$ are idempotent.

3.1 Conjugate Operation

The notions of duality and co-duality can be defined for functions defined over complete lattices. However, if the lattice is also a Boolean lattice, so that each lattice element can be *uniquely complemented*, then the notion of conjugate of a function can also be defined. This is then used to define inverse of a disjunctive function and converse of a conjunctive function.

Definition 4 A lattice $(X, \leq)$ is said to be a *Boolean* lattice, if

- (Bounded): $inf\, X, sup\, X \in X$, and
- (Distributive): $\forall x, y, z \in X : x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$, and
- (Complement): $\forall x \in X : \exists$ unique $x^c \in X$ s.t. $x \sqcap x^c = inf\, X;\ x \sqcup x^c = sup\, X$.

For a pair x, y of elements of a Boolean lattice $(X, \leq)$, the notation $x - y$ is used to denote $x \sqcap y^c$. A power set lattice is an example of a Boolean lattice that is also complete. The following hold for a Boolean lattice:

Lemma 3 [4, Lemma 7.3] Let $(X, \leq)$ be a Boolean lattice. Then

1. $(inf\, X)^c = sup\, X$ and $(sup\, X)^c = inf\, X$.
2. $\forall x \in X : (x^c)^c = x$.

3. (de Morgan's Law): $\forall x, y \in X : (x \sqcap y)^c = x^c \sqcup y^c$; $\quad (x \sqcup y)^c = x^c \sqcap y^c$.

4. $\forall x, y \in X : [x \leq y] \Leftrightarrow [x \sqcap y^c = inf\, X]$.

Definition 5 Given a complete Boolean lattice (CBL) $(X, \leq)$ and a function $f : X \to X$, the *conjugate* of f, denoted $\overline{f}$, is defined as:

$$\forall x \in X : \overline{f}(x) := (f(x^c))^c.$$

If f is disjunctive, then its *inverse*, denoted f^{-1}, is defined to be the function $\overline{f^{\perp}}$; and if f is conjunctive, then its *converse*, denoted $f^{\sharp}$, is defined to be the function $(\overline{f})^{\perp}$.

Note that if f is disjunctive (respectively, conjunctive), then it follows from de Morgan's law that $\overline{f}$ is conjunctive (respectively, disjunctive). Hence inverse (respectively, converse) of a disjunctive (respectively, conjunctive) function is well defined. Moreover, it follows from Lemma 2 that the inverse of a disjunctive function is disjunctive, and converse of a conjunctive function is conjunctive. Appendix A provides a justification for the choice of the name inverse for the operation conjugate of dual.

Example 5 Consider the power set lattice of languages defined over the event set Σ and the prefix and extension closure operations. It follows from the definition of conjugate that for a language $K \subseteq \Sigma^*$, $\overline{pr}(K) = \Sigma^* - pr(\Sigma^* - K) = sup\, E(K)$, the supremal extension closed sublanguage of K. Similarly, $\overline{ext}(K) = \Sigma^* - ext(\Sigma^* - K) = sup\, P(K)$, the supremal prefix closed sublanguage of K. These relations between prefix and extension closure operations are not coincidental, rather they can be derived as we show below.

The following lemma lists a few properties of the conjugate operation.

Lemma 4 Consider a CBL $(X, \leq)$ and functions $f, g : X \to X$.

1. (self-inverse) $\quad \overline{\overline{f}} = f$
2. (composition) $\quad \overline{fg} = \overline{f}\,\overline{g}$
3. (idempotence) $\quad [ff = f] \Leftrightarrow [\overline{f}\,\overline{f} = \overline{f}]$

Proof: The first assertion is obvious. In order to see the second assertion, pick $x \in X$. Then we have $\overline{fg}(x) = (fg(x^c))^c = (f(\overline{g}(x))^c)^c = \overline{f}\,\overline{g}(x)$. The forward implication of the third assertion is obtained as follows: $\overline{f}\,\overline{f} = \overline{ff} = \overline{f}$, where the first equality follows from part 2, and the second from hypothesis. The backward implication of the third assertion is obtained as follows: $f = \overline{\overline{f}} = \overline{\overline{f}\,\overline{f}} = ff$, where the first equality follows from part 1, the second from hypothesis, and the third from parts 2 and 1. ∎

Next we provide a few properties of the inverse and the converse operations. The following proposition provides an alternative definition of inverse as well as converse.

Proposition 3 Consider a CBL $(X, \leq)$, and functions $f, g, h : X \to X$.

1. If f is disjunctive, then $f^{-1} = h$ if and only if

$$\forall x, y \in X : [f(x) \sqcap y = \mathit{inf}\, X] \Leftrightarrow [x \sqcap h(y) = \mathit{inf}\, X]. \tag{1}$$

2. If g is conjunctive, then $g^{\sharp} = h$ if and only if

$$\forall x, y \in X : [g(x) \sqcup y = \mathit{sup}\, X] \Leftrightarrow [x \sqcup h(y) = \mathit{sup}\, X].$$

Proof: Since $f^{-1} := \overline{(f^{\perp})}$, it follows from the first part of Lemma 4 that $f^{-1} = h$ if and only if $f^{\perp} = \overline{h}$. Thus it suffices to show that $f^{\perp} = \overline{h}$ is equivalent to (1). From Proposition 1, $f^{\perp} = \overline{h}$ is equivalent to

$$\forall x, y \in X : [f(x) \leq y] \Leftrightarrow [x \leq \overline{h}(y)]. \tag{2}$$

Hence it suffice to show the equivalence of (2) and (1). Replacing y by y^c in (2) we obtain

$$\forall x, y \in X : [f(x) \leq y^c] \Leftrightarrow [x \leq (h(y))^c].$$

Thus the desired equivalence follows from the part 4 of Lemma 3. ■

The following proposition provides additional properties of inverse and converse operations.

Proposition 4 Consider a CBL $(X, \leq)$, disjunctive functions $f, f_1, f_2 : X \to X$, and conjunctive functions $g, g_1, g_2 : X \to X$.

1. (commutation)	(a) $\overline{f^{\perp}} = \overline{f}^{\top}$	(b) $\overline{g^{\top}} = \overline{g}^{\perp}$
2. (self-inverse)	(a) $(f^{-1})^{-1} = f$	(b) $(g^{\sharp})^{\sharp} = g$
3. (composition)	(a) $(f_1 f_2)^{-1} = f_2^{-1} f_1^{-1}$	(b) $(g_1 g_2)^{\sharp} = g_2^{\sharp} g_1^{\sharp}$
4. (idempotence)	(a) $[ff = f] \Leftrightarrow [f^{-1} f^{-1} = f^{-1}]$	(b) $[gg = g] \Leftrightarrow [g^{\sharp} g^{\sharp} = g^{\sharp}]$

Proof: 1. Since $\overline{f^{\perp}} = f^{-1}$, it follows from the first part of Proposition 3 that it suffices to show that (1) holds with h replaced by $\overline{f}^{\top}$. This can be shown as follows:

$$\begin{aligned} [f(x) \sqcap y = \mathit{inf}\, X] &\Leftrightarrow [f(x) \leq y^c] \\ &\Leftrightarrow [y \leq (f(x))^c] \\ &\Leftrightarrow [y \leq \overline{f}(x^c)] \\ &\Leftrightarrow [\overline{f}^{\top}(y) \leq x^c] \\ &\Leftrightarrow [\overline{f}^{\top}(y) \sqcap x = \mathit{inf}\, X], \end{aligned}$$

where the fourth equivalence follows from the second part of Corollary 1, and the other equivalences follow from Lemma 3.

2. From part 1, $f^{-1} = \overline{f}^{\top}$. Hence from Proposition 1, $(f^{-1})^{\perp} = \overline{f}$. By applying conjugate operation on both sides of this identity, we obtain $(f^{-1})^{-1} = f$.

3. We have $(f_1 f_2)^{-1} = \overline{((f_1 f_2)^{\perp})} = \overline{f_2^{\perp} f_1^{\perp}} = \overline{f_2^{\perp}}\,\overline{f_1^{\perp}} = f_2^{-1} f_1^{-1}$, as desired.

4. The forward implication can be obtained as follows: $f^{-1} = (ff)^{-1} = f^{-1}f^{-1}$, where the first equality comes from hypothesis, and the second from part 3. The backward implication can be obtained as follows: $f = (f^{-1})^{-1} = (f^{-1}f^{-1})^{-1} = ff$, where the first equality comes from part 2, the second from hypothesis, and the final from parts 3 and 2. ∎

Remark 2 The commutative diagram of Figure 1 summarizes the relationship among the various operations that we have obtained above. Note that given any disjunctive func-

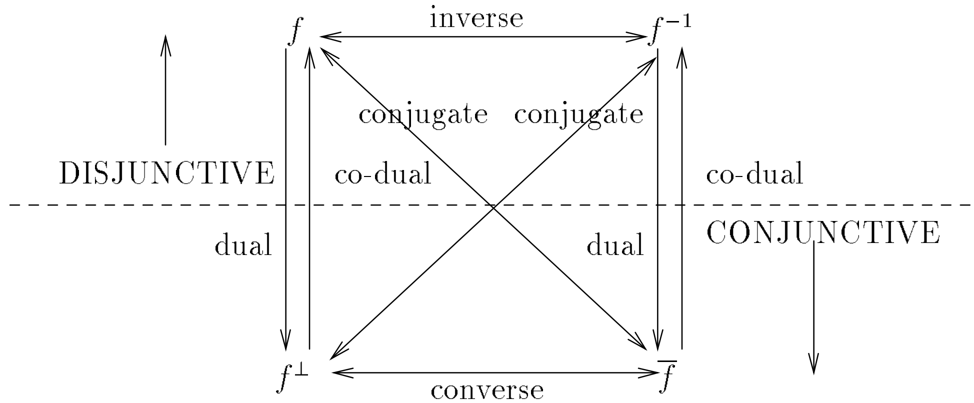

Figure 1: Commutative diagram for dual, co-dual, conjugate, inverse, and converse

tion f, by applying $(\cdot)^{\perp}, (\cdot)^{\top}$ and $\overline{(\cdot)}$ any number of times, we get four unique functions: $(f, f^{-1}, f^{\perp}, \overline{f})$, the first two of which are disjunctive and the last two are conjunctive.

The commutative diagram of Figure 1 yields the commutative diagram shown in Figure 2 for operations of pr, ext, $sup\,P$, and $sup\,E$.

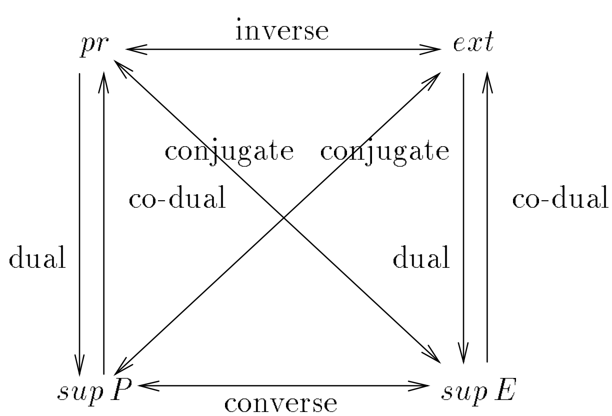

Figure 2: Commutative diagram for pr, ext, $sup\,P$, and $sup\,E$ operations

4 Extremal Solutions of Inequations

Given a complete lattice $(X, \leq)$ and a finite family of functions $\{f_i, g_i : X \to X\}_{i \leq n}$, where $n \in \mathcal{N}$, we next consider computation of extremal solutions of the system of inequations:

$$[\forall i \leq n : f_i(x) \leq g_i(x)],$$

where $x \in X$ is the variable of the system of inequations. Note that this also allows us to obtain extremal solutions of a system of *equations*, as each equation can equivalently be written as a pair of *inequations*. We show that the computation of extremal solutions of the above system of inequations can be reduced to extremal fixed point computations of certain induced functions. We need the result of the following lemma:

Lemma 5 Consider the system of inequations $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ over a complete lattice $(X, \leq)$. Define functions $h_1, h_2 : X \to X$ as:

$$\forall y \in X : h_1(y) := \sqcap_{i \leq n} f_i^{\perp}(g_i(y)); \quad \forall y \in X : h_2 := \sqcup_{i \leq n} g_i^{\top}(f_i(y)). \tag{3}$$

1. If f_i is disjunctive and g_i is monotone for each $i \leq n$, then h_1 is monotone, and $\forall y, z \in X : [y \leq h_1(z)] \Leftrightarrow [\forall i \leq n : f_i(y) \leq g_i(z)]$.

2. If f_i is monotone and g_i is conjunctive for each $i \leq n$, then h_2 is monotone, and $\forall y, z \in X : [h_2(y) \leq z] \Leftrightarrow [\forall i \leq n : f_i(y) \leq g_i(z)]$.

Proof: In order to show the monotonicity of h_1, it suffices to show that for each $i \leq n$, $f_i^{\perp} g_i$ is monotone. This follows from the facts that g_i is given to be monotone, $f_i^{\perp}$ is conjunctive (refer to Lemma 2), so that it is also monotone, and monotonicity is preserved under composition of functions.

In order to see the second claim, fix $y, z \in X$. Then we have the following series of equivalences:

$$\begin{aligned} [y \leq h_1(z)] &\Leftrightarrow [y \leq \sqcap_{i \leq n} f_i^{\perp}(g_i(z))] \\ &\Leftrightarrow [\forall i \leq n : y \leq f_i^{\perp}(g_i(z))] \\ &\Leftrightarrow [\forall i \leq n : f_i(y) \leq g_i(z)], \end{aligned}$$

where the final equivalence follows from the first part of Corollary 1. ∎

Theorem 3 Consider the system of inequations $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ over a complete lattice $(X, \leq)$. Let

$$Y := \{y \in X \mid \forall i \leq n : f_i(y) \leq g_i(y)\} \tag{4}$$

be the set of all solutions of the system of inequations; and

$$Y_1 := \{y \in X \mid h_1(y) = y\}, \quad Y_2 := \{y \in X \mid h_2(y) = y\} \tag{5}$$

be the sets of all fixed points of h_1 and h_2, respectively, where h_1 and h_2 are defined by (3).

1. If f_i is disjunctive and g_i is monotone, then $sup\, Y \in Y$, $sup\, Y_1 \in Y_1$ and $sup\, Y = sup\, Y_1$.

2. If f_i is monotone and g_i is conjunctive, then $inf\, Y \in Y$, $inf\, Y_2 \in Y_2$ and $inf\, Y = inf\, Y_2$.

Proof: It follows from the first part of Lemma 5 that h_1 is monotone. Hence it follows from the second part of Theorem 1 that $sup\, Y_1 \in Y_1$, and

$$sup\, Y_1 = sup\, Y_1', \text{ where } Y_1' := \{y \in X \mid y \leq h_1(y)\}. \tag{6}$$

It remains to show that $sup\, Y \in Y$ and $sup\, Y = sup\, Y_1$. In view of (6), it suffices to show that $Y = Y_1'$, i.e, $y \in X$ is a solution of the system of inequations if and only if $y \leq h_1(y)$. This follows from the first part of Lemma 5 by setting $z = y$. ∎

The following corollary follows from Theorems 3 and 2.

Corollary 2 Consider the system of inequations $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ over a complete lattice $(X, \leq)$; the set Y of all solutions of the system of inequations as defined by (4); and functions h_1 and h_2 defined by (3). If f_i is disjunctive and g_i is conjunctive for each $i \leq n$, then $sup\, Y = (h_1)_*(sup\, X)$ and $inf\, Y = (h_2)^*(inf\, X)$.

Proof: Since f_i is disjunctive and g_i is conjunctive (which implies f_i as well as g_i are monotone) Theorem 3 implies that $sup\, Y = sup\, Y_1$ and $inf\, Y = inf\, Y_2$, where Y_1, Y_2 are the sets of fixed points of h_1, h_2, respectively. Using the facts that $f_i^{\perp}$ is conjunctive, and g_i is conjunctive, it is easily shown that h_1 is conjunctive. Hence it follows from the second part of Theorem 2 that $sup\, Y_1 = (h_1)_*(sup\, X)$. Similarly, using the facts that $g_i^{\top}$ is disjunctive, and f_i is disjunctive, it is easily shown that h_2 is disjunctive. Hence it follows from the first part of Theorem 2 that $inf\, Y = (h_2)^*(inf\, X)$, as desired. ∎

The result of Corollary 2 is applicable whenever f_i is disjunctive and g_i is conjunctive for each $i \leq n$. The following theorem provides techniques for the computation of extremal solutions of the system of inequations under different conditions.

Theorem 4 Consider the system of inequations $\{f_i(x) \leq g_i(x)\}_{i \leq n}$ over a complete lattice $(X, \leq)$; and the set Y of all solutions of the system of inequations as defined by (4).

1. Let f_i be disjunctive and g_i be monotone. Consider the following iterative computation:

   - $y_0 := sup\, X$,
   - $\forall k \geq 0 : y_{k+1} := h_1(y_k)$,

   where h_1 is defined by (3). Suppose $m \in \mathcal{N}$ is such that $y_{m+1} = y_m$; then $y_m = sup\, Y$.

2. Let f_i be monotone and g_i be conjunctive. Consider the following iterative computation:

   - $y_0 := inf\, X$,
   - $\forall k \geq 0 : y_{k+1} := h_2(y_k)$,

where h_2 is defined by (3). Suppose $m \in \mathcal{N}$ is such that $y_{m+1} = y_m$; then $y_m = inf\, Y$.

Proof: It follows from Theorem 3 that the supremal solution of the system of inequations, $sup\, Y$, exists. We first show that $y_m \in Y$, i.e., it is a solution of the system of inequations. Since $y_{m+1} = y_m$, we have $h_1(y_m) = y_m$, which implies that $y_m \leq h_1(y_m)$. Thus by setting $y = z = y_m$ in the equivalence of the first part of Lemma 5, we obtain that $y_m \in Y$.

Next we show that if $z \in Y$ is another solution of the system of inequations, then $z \leq y_m$. We use induction to show that for each $k \geq 0$, $z \leq y_k$. If $k = 0$, then $y_k = sup\, X$, so that $z \leq y_k = sup\, X$. Thus the base step trivially holds. Suppose for induction hypothesis that $z \leq y_k$ for some $k \geq 0$. From the first part of Lemma 5 we have that h_1 is monotone. This together with the induction hypothesis implies that $h_1(z) \leq h_1(y_k) = y_{k+1}$. Thus it suffices to show that $z \leq h_1(z)$. Since $z \in Y$, $f_i(z) \leq g_i(z)$ for each $i \leq n$. Thus by setting $y = z$ in the first part of Lemma 5, we obtain that $z \leq h_1(z)$. ■

4.1 Specializations of Extremal Solutions of Inequations

In many applications we are interested in finding the supremal solution smaller than a given element $w \in X$, and/or the infimal solution greater than the given element w, of a system of inequations: $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq n}$, where $\{v_i \in X\}_{i \leq n}$ is a given family of fixed elements. Note that if $w = v_i = sup\, X$, then this problem reduces to the problem analyzed in Theorems 3 and 4. Conversely, we show that the problem just described can be analyzed using techniques developed in Theorems 3 and 4 provided the lattice is also Boolean, so that lattice elements can be uniquely complemented.

First note that the constraint that the supremal solution be smaller than w can be captured by adjoining the following additional inequation:

$$f_a(x) \leq g_a(x),$$

where f_a is the identity function, i.e., $f_a(x) := x$, and g_a is the constant function $g_a(x) := w$. Similarly the constraint that the infimal solution should be greater than w can be captured by adjoining the following additional inequation:

$$f_b(x) \leq g_b(x),$$

where f_b is the constant function $f_b(x) := w$, and $g_b(x)$ is the identity function. Next note that

$$[f_i(x) \sqcap v_i \leq g_i(x)] \Leftrightarrow [f_i(x) \leq g_i(x) \sqcup v_i^c].$$

Thus if we define $g_i'(x) := g_i(x) \sqcup v_i^c$ and $f_i'(x) := f_i(x) \sqcap v_i$, then we obtain

$$[f_i(x) \sqcap v_i \leq g_i(x)] \Leftrightarrow [f_i(x) \leq g_i'(x)] \Leftrightarrow [f_i'(x) \leq g_i(x)].$$

It can be checked that (i) g_i' is monotone if and only if g_i is monotone; f_i' is monotone if and only if f_i is monotone, (ii) the identity function is disjunctive as well as conjunctive,

(iii) a constant function is monotone, (iv) $f_a^\top(g_a(x)) = g_b^\top(f_b(x)) = w$, and (v) $(g'_i(x))^c = (g_i(x) \sqcup v_i^c)^c = v_i \sqcap (g_i(x))^c = v_i - g_i(x)$. We also have

$$\begin{aligned} [\sqcap_{i\leq n} f_i^\perp(g'_i(x))] \sqcap f_a^\perp(g_a(x)) &= \sqcap_{i\leq n}[f_i^{-1}[v_i - g_i(x)]]^c \sqcap w \\ &= w \sqcap [\sqcup_{i\leq n} f_i^{-1}[v_i - g_i(x)]]^c \\ &= w - \sqcup_{i\leq n} f_i^{-1}[v_i - g_i(y)]. \end{aligned}$$

Thus the following result can be obtained as a corollary of Theorems 3 and 4.

Corollary 3 Given a CBL $(X, \leq)$, a fixed $w \in X$, and a family of lattice elements $\{v_i \in X\}_{i\leq n}$, consider the system of inequations $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i\leq n}$ over the given CBL. Define functions $h'_1, h'_2 : X \to X$ as:

$$\forall y \in X : h'_1(y) := w - [\sqcup_{i\leq n} f_i^{-1}(v_i - g_i(y))]; \quad h'_2(y) := w \sqcup [\sqcup_{i\leq n} g_i^\top(v_i \sqcap f_i(y))]. \tag{7}$$

1. Suppose f_i is disjunctive and g_i is monotone for each $i \leq n$. Then
   (a) h'_1 is monotone; and the supremal solution smaller than w of the given system of inequations exists and equals the supremal fixed point of the function h'_1.
   (b) Consider the following iterative computation:
   - $y_0 := w$,
   - $\forall k \geq 0 : y_{k+1} := h'_1(y_k)$.

   Suppose $m \in \mathcal{N}$ is such that $y_{m+1} = y_m$; then y_m equals the supremal solution smaller than w of the given system of inequations.

2. Suppose f_i is monotone and g_i is conjunctive for each $i \leq n$. Then
   (a) h'_2 is monotone; and the infimal solution greater than w of the given system of inequations exists and equals the infimal fixed point of the function h'_2.
   (b) Consider the following iterative computation:
   - $y_0 := w$,
   - $\forall k \geq 0 : y_{k+1} := h'_2(y_k)$.

   Suppose $m \in \mathcal{N}$ is such that $y_{m+1} = y_m$; then y_m equals the infimal solution greater than w of the given system of inequations.

We next apply Theorem 4 and Corollary 3 to compute the extremal solution of a single inequation.

Theorem 5 Consider a CBL $(X, \leq)$, a disjunctive and idempotent function $f : X \to X$, and fixed lattice elements $w, v \in X$.

1. The supremal solution smaller than w of $f(x) \sqcap v \leq x$ equals $w - f^{-1}(v - w)$.

2. The infimal solution greater than w of $f(x) \sqcap v \leq x$ equals $w \sqcup [f(w) \sqcap v]$.

Proof: Since f is disjunctive, and identity function is monotone, it follows from the first part of Corollary 3 that the supremal solution smaller than w of $f(x) \sqcap v \leq x$ exists. Also, the function h_1' defined by (7) simplifies to:

$$\forall y \in X : h_1'(y) = w - f^{-1}(v - y).$$

We apply the iterative computation of Corollary 3 for computing the desired supremal solution. Then $y_0 = w$. This implies

$$y_1 = h_1'(y_0) = h_1'(w) = w - f^{-1}(v - w) \tag{8}$$

$$y_2 = h_1'(y_1) = w - f^{-1}(v - y_1) \tag{9}$$

We show that $y_1 = y_2$. First note that $y_1 = w - f^{-1}(v - w) \leq w = y_0$. Hence monotonicity of h_1' (refer to part 1(a) of Corollary 3) implies that $y_2 = h_1'(y_1) \leq h_1'(y_0) = y_1$. Thus it suffices to show that $y_2 \geq y_1$. Using (8) we obtain

$$\begin{aligned}
f^{-1}(v - y_1) &= f^{-1}(v - [w - f^{-1}(v - w)]) \\
&= f^{-1}(v \sqcap [w \sqcap [f^{-1}(v - w)]^c]^c) \\
&= f^{-1}(v \sqcap [w^c \sqcup f^{-1}(v - w)]) \\
&= f^{-1}[(v \sqcap w^c) \sqcup (v \sqcap f^{-1}(v - w)] \\
&= f^{-1}(v - w) \sqcup f^{-1}(v \sqcap f^{-1}(v - w)),
\end{aligned}$$

where the last equality follows from the fact that f^{-1} is disjunctive. Hence from (9) we obtain

$$\begin{aligned}
y_2 &= w - f^{-1}(v - w) - f^{-1}(v \sqcap f^{-1}(v - w)) \\
&\geq w - f^{-1}(v - w) - f^{-1}(f^{-1}(v - w)) \\
&= w - f^{-1}(v - w) \\
&= y_1,
\end{aligned}$$

where the inequality follows from the fact that f^{-1} is disjunctive, so that it is also monotone, and the second equality follows from the fact that f^{-1} is idempotent. Thus it follows from part 1(b) of Corollary 3 that $y_1 = y_2 = w - f^{-1}(v - w)$ is the supremal solution smaller than w of $f(x) \sqcap v \leq x$. ■

In Theorem 5, it is required that the function f be idempotent. However, if this is not the case, then we can replace f by its disjunctive closure f^*. We need the result of the following lemma:

Lemma 6 Consider a CBL $(X, \leq)$, a disjunctive function $f : X \to X$, and a fixed $v \in X$. Then

$$\forall x \in X : [f(x) \sqcap v \leq x] \Leftrightarrow [f^*(x) \sqcap v \leq x].$$

Proof: It suffices to show the forward inequality, as the reverse inequality is obvious. Again, it suffices to show that for each $i \in \mathcal{N}$,

$$[f(x) \sqcap v \leq x] \Rightarrow [f^i(x) \sqcap v \leq x].$$

We show this using induction on i. If $i = 0$, then $f^i(x) \sqcap v = x \sqcap v \leq x$. Thus the base step trivially holds. Suppose for induction hypothesis that $f^i(x) \sqcap v \leq x$, i.e., $f^i(x) \leq x \sqcup v^c$. Then by applying f on both sides of the last inequation and using monotonicity of f we obtain:

$$f^{i+1}(x) \leq f(x \sqcup v^c) = f(x) \sqcup f(v^c) \leq (x \sqcap v^c) \sqcup (v^c \sqcup v^c) = x \sqcup v^c,$$

where the first equality follows from disjunctivity of f, and the second inequality follows from the hypothesis that for each $x \in X$, $f(x) \sqcap v \leq x$, i.e., $f(x) \leq x \sqcup v^c$. This establishes the induction step and completes the proof. ∎

Since the disjunctive closure of any function is idempotent, and preserves disjunctivity, the following result can be obtained as a corollary of Theorem 5 and Lemma 6.

Corollary 4 Consider a CBL $(X, \leq)$, a disjunctive function $f : X \to X$, and fixed lattice elements $w, v \in X$.

1. The supremal solution smaller than w of $f(x) \sqcap v \leq x$ equals $w - (f^*)^{-1}(v - w)$.

2. The infimal solution greater than w of $f(x) \sqcap v \leq x$ equals $w \sqcup [f^*(w) \sqcap v]$.

It is evident from Theorem 5 and Corollary 4 that the extremal solution of a single inequation of the type $f(x) \sqcap v \leq x$ can be easily obtained. We show in the following theorem that under certain conditions this can be used for computing the extremal solutions of a system of inequations in a *modular* fashion, i.e., it is possible to first compute the extremal solution of inequations of the type $f(x) \sqcap v \leq x$ as in Theorem 5 and Corollary 4, and then compute the extremal solutions of the remaining inequations. Thus the iterative computation scheme can be considerably simplified. For simplicity of illustration we only consider a pair of inequations.

Theorem 6 Given a CBL $(X, \leq)$, a fixed $w \in X$, and a pair of lattice elements $\{v_i \in X\}_{i \leq 2}$, consider a pair of inequations $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq 2}$ over the given CBL, where f_1 is disjunctive and idempotent and g_1 is the identity function.

1. Let f_2 be disjunctive and g_2 be monotone, and $f_1^\perp f_2^\perp = f_2^\perp$ (or equivalently, $f_1^{-1} f_2^{-1} = f_2^{-1}$). Consider the following iterative computation:

   - $y_0 := w - f_1^{-1}(v_1 - w)$
   - $\forall k \geq 0 : y_{k+1} := y_k - f_2^{-1}(v_2 - g_2(y_k))$

   Suppose $m \in \mathcal{N}$ is such that $y_{m+1} = y_m$, then y_m is the supremal solution smaller than w of $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq 2}$.

2. Let f_2 be monotone and g_2 be conjunctive, and $f_1 g_2^\top = g_2^\top$. Consider the following iterative computation:

   - $y_0 := w \sqcup [f_1(w) \sqcap v_1]$
   - $\forall k \geq 0 : y_{k+1} := y_k \sqcup g_2^\top(v_2 \sqcap f_2(y_k))$

   Suppose $m \in \mathcal{N}$ is such that $y_{m+1} = y_m$, then y_m is the infimal solution greater than w of $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq 2}$.

Proof: It follows from part 1(a) of Corollary 3 that the required supremal solution exists. Also, it follows from the first part of Theorem 5 that the supremal solution smaller than w of $f_1(x) \sqcap v_1 \leq g_1(x) = x$ is $w - f^{-1}(v_1 - w)$. Hence we obtain from part 1(b) of Corollary 3 that $y_m \leq w$ equals the supremal solution of $f_2(x) \sqcap v_2 \leq g_2(x)$ smaller than the supremal solution of $f_1(x) \sqcap v_1 \leq g_1(x) = x$. Hence if $z \leq w$ denotes the supremal solution of $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq 2}$, then $z \leq y_m$. Since $z \leq w$ is the supremal solution of $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq 2}$, in order to show that $y_m \leq z$, it suffices to show that $y_m \leq w$ is a solution of $\{f_i(x) \sqcap v_i \leq g_i(x)\}_{i \leq 2}$.

As noted above $y_m \leq w$ is a solution of $f_2(x) \sqcap v_2 \leq g_2(x)$. We show using induction that for each $k \geq 0$, y_k is a solution of $f_1(x) \sqcap v_1 \leq g_1(x) = x$, i.e.,

$$\begin{aligned}
[f_1(y_k) \sqcap v_1 \leq y_k] &\Leftrightarrow [f_1(y_k) \sqcap (v_1 - y_k) = \mathit{inf}\, X] \\
&\Leftrightarrow [y_k \sqcap f_1^{-1}(v_1 - y_k) = \mathit{inf}\, X] \\
&\Leftrightarrow [f_1^{-1}(v_1 - y_k) \leq y_k^c],
\end{aligned}$$

where the second equivalence follows from (1). Since $y_0 \leq w$ is the supremal solution of $f_1(x) \sqcap v_1 \leq x$ (refer to the first part of Theorem 5), it follows that the base step holds. Assume for induction hypothesis that $f_1^{-1}(v_1 - y_k) \leq y_k^c$ for some $k \geq 0$. Then the induction step can be established as follows:

$$\begin{aligned}
f_1^{-1}(v_1 - y_{k+1}) &= f_1^{-1}(v_1 - [y_k - f_2^{-1}(v_2 - g_2(y_k))]) \\
&= f_1^{-1}[(v_1 - y_k) \sqcup (v_1 \sqcap f_2^{-1}(v_2 - g_2(y_k)))] \\
&= f_1^{-1}(v_1 - y_k) \sqcup f_1^{-1}[v_1 \sqcap f_2^{-1}(v_2 - g_2(y_k))] \\
&\leq y_k^c \sqcup f_1^{-1}(f_2^{-1}(v_2 - g_2(y_k))) \\
&= y_k^c \sqcup f_2^{-1}(v_2 - g_2(y_k)) \\
&= y_{k+1}^c,
\end{aligned}$$

where the first equality follows from the definition of y_k, the third equality follows from the fact that f_1^{-1} is disjunctive, the inequality follows from induction hypothesis and the fact that f_1^{-1} is monotone (as it is disjunctive), the fourth equality follows from the fact that $f_1^\perp f_2^\perp = f_2^\perp$, which is equivalent to $f_1^{-1} f_2^{-1} = f_2^{-1}$, and the final equality follows from the definition of y_{k+1}. ■

5 Applications to DES Supervisory Control

In this section we demonstrate how the techniques for computation of extremal solution of inequations developed above can be applied for computation of maximally permissive supervisors in control of logical behavior of DESs.

Given a discrete event plant G with event set Σ, its logical behavior is described using a pair of languages $(L_m(G), L(G))$ satisfying $L_m(G) \subseteq L(G) = pr(L(G)) \neq \emptyset$. $L(G) \subseteq \Sigma^*$ is called the *generated* language of G, and consists of the event sequences that the plant can execute. It is prefix closed since for a sequence of events to occur, all its prefixes must also occur. $L_m(G) \subseteq L(G)$ is called the *marked language* of G and consists of those strings whose execution imply completion of a certain task. A desired generated behavior is a certain sublanguage of $L(G)$ and a desired marked behavior is a certain sublanguage of $L_m(G)$. A control mechanism is needed so that the plant executes only those sequences of events which are desired.

The event set Σ is partitioned into $\Sigma_u \cup (\Sigma - \Sigma_u)$, the sets of *uncontrollable* and *controllable* events. A supervisor is *event driven*, and at each event execution epoch it dynamically disables some of the controllable events from occurring so that the behavior of the controlled plant satisfies the desired behavior constraint. Due to the inability of a supervisor to prevent uncontrollable events from occurring, a certain behavior can be achieved under control only if it is controllable [27]. A language $H \subseteq L(G)$ is said to be *controllable* if

$$pr(H)\Sigma_u \cap L(G) \subseteq pr(H). \tag{10}$$

If in addition it is also desired that the supervisor be *non-blocking* so that any string in the generated language of the controlled plant can be extended to a string in its marked language, then the desired behavior must also be relative closed (also known as L_m-closed) [27]. A language $H \subseteq L_m(G)$ is called *relative closed* if

$$pr(H) \cap L_m(G) \subseteq H. \tag{11}$$

Furthermore, if the supervisor's observation is filtered through a mask function M defined over the event set Σ, then the desired behavior can be achieved under control only when it is also observable [20, 3]. A language $H \subseteq L(G)$ is said to be *observable* if

$$\forall s, t \in pr(H), \sigma \in \Sigma : [M(s) = M(t), s\sigma \in pr(H), t\sigma \in L(G)] \Rightarrow [t\sigma \in pr(H)]. \tag{12}$$

In case the supervisor is *local* so that it is only able to control those events that it observes [11, 19, 21], then the desired behavior must satisfy a condition stronger than observability, called normality. A language $H \subseteq L(G)$ is said to be *normal* if

$$\mathcal{M}(pr(H)) \cap L(G) \subseteq pr(H), \tag{13}$$

where $\mathcal{M} : 2^{\Sigma^*} \to 2^{\Sigma^*}$ is the map induced by the mask function M and is defined as:

$$\forall H \subseteq \Sigma^* : \mathcal{M}(H) := \{s \in \Sigma^* \mid \exists t \in H \text{ s.t. } M(t) = M(s)\}.$$

Thus $\mathcal{M}$ maps a certain string to all those string which look alike under the mask M. It can be verified that $\mathcal{M}$ is disjunctive and idempotent. Also, $\mathcal{M}^{-1} = \mathcal{M}$.

In case the desired behavior fails to satisfy one or more of the required conditions, a supervisor is synthesized that achieves a supremal sublanguage or infimal superlanguage of the desired behavior satisfying the required conditions. We next discuss computation of such languages; this requires computation of extremal solutions of inequations of the type (10)-(13) defined over the power set lattice $(2^{\Sigma^*}, \subseteq)$. Since a power set lattice is also a CBL, the results developed in the previous section can be applied.

5.1 Extremal Relative Closed Languages

Consider the definition of relative closure given by (11). It is of the form

$$f(H) \cap L_m(G) \subseteq g(H),$$

where f is the prefix closure operation which is disjunctive and idempotent, and g is the identity function which is conjunctive and idempotent. Hence it follows from Corollary 3 that the *supremal relative closed sublanguage* as well as *infimal relative closed superlanguage* of a given language $K \subseteq L_m(G)$, denoted $sup\, R(K)$ and $inf\, \overline{R}(K)$, respectively, exist, as expected [12]. Furthermore it follows from the first part of Theorem 5 that

$$sup\, R(K) = K - pr^{-1}(L_m(G) - K) = K - (L_m(G) - K)\Sigma^*.$$

On the other hand, it follows from the second part of Theorem 5 that

$$inf\, \overline{R}(K) = K \cup (pr(K) \cap L_m(G)) = pr(K) \cap L_m(G),$$

where the second equality follows from the fact that $K \subseteq L_m(G)$.

5.2 Extremal Controllable Languages

Consider the definition of controllability given by (10). It is of the form

$$f(H) \cap L(G) \subseteq g(H),$$

where f is the composition of the prefix closure operation and the operation of concatenation with the event set Σ_u, and g is the prefix closure operation, which is monotone but not conjunctive. Since prefix closure as well as concatenation operations are disjunctive, and disjunctivity is preserved under composition of functions, it follows that f is disjunctive. Hence it follows from Corollary 3 that the supremal controllable sublanguage of a language $K \subseteq L(G)$, denoted $sup\, C(K)$, exists, as expected [26]. However, since g is not conjunctive, the infimal controllable superlanguage of K need not exist, as expected.

Since $f(\cdot) = pr(\cdot)\Sigma_u$, it follows from Figure 2 and Proposition 4 that $f^{-1}(\cdot) = [(\cdot)/\Sigma_u]\Sigma^*$. Hence the following iterative scheme of Corollary 3 can be used to compute $sup\, C(K)$:

- $K_0 := K$,
- $K_{i+1} := K_i - [(L(G) - K_i)/\Sigma_u]\Sigma^*$.

If there exists $m \in \mathcal{N}$ such that $K_{m+1} = K_m$, then $K_m = sup\,C(K)$. Using the arguments similar to those given in [26] it can be shown that such an m exists whenever K and $L(G)$ are *regular* languages [10].

If we require that the extremal language be controllable as well as prefix closed, then we must consider the extremal solution of the following two inequations:

$$pr(H)\Sigma_u \cap L(G) \subseteq pr(H); \quad pr(H) \subseteq H, \tag{14}$$

where $H \subseteq L(G)$ is the variable of inequation. Using the fact that $H \subseteq L(G)$, we show that the two inequations of (14) is equivalent to the following single inequation:

$$pr(H)\Sigma_u \cap L(G) \subseteq H. \tag{15}$$

It is clear that (14) implies (15). Also, since $H \subseteq pr(H)$, the first inequation of (14) follows from (15). It remains to show that (15) implies $pr(H) \subseteq H$. First note that if $f(\cdot) = pr(\cdot)\Sigma_u$, then $f^*(\cdot) = pr(\cdot)\Sigma_u^*$. Hence it follows from Lemma 6 that (15) is equivalent to

$$pr(H)\Sigma_u^* \cap L(G) \subseteq H. \tag{16}$$

This equivalence was first demonstrated in [2] under the assumption that K is prefix closed, and without this assumption in [13]. Since $pr(H) \subseteq pr(H)\Sigma_u^*$ and $pr(H) \subseteq L(G)$ (as $H \subseteq L(G)$), it follows from (16) that $pr(H) \subseteq pr(H)\Sigma_u^* \cap L(G) \subseteq H$, as desired.

It follows from the above discussions that (15), or equivalently (16), can be used to compute the extremal prefix closed and controllable language. We consider (16), which is of the form

$$f(H) \cap L(G) \subseteq g(H),$$

where $f(\cdot) = pr(\cdot)\Sigma_u^*$, which is disjunctive as well as idempotent, and g is the identity function. Consequently, it follows from Corollary 3 that the *supremal prefix closed and controllable sublanguage* and *infimal prefix closed controllable superlanguage* of $K \subseteq L(G)$, denoted $sup\,PC(K)$ and $inf\,\overline{PC}(K)$, respectively, exist. Furthermore, it follows from Theorem 5 that

$$\begin{aligned} sup\,PC(K) &= K - f^{-1}(L(G) - K) = K - [(L(G) - K)/\Sigma_u^*]\Sigma^* \\ inf\,\overline{PC}(K) &= K \cup (f(K) \cap L(G)) = K \cup (pr(K)\Sigma_u^* \cap L(G)) = pr(K)\Sigma_u^* \cap L(G). \end{aligned}$$

The above formula for $sup\,PC(K)$ was first reported in [2] under the assumption that K is prefix closed. Our derivation shows that we do not need to impose this assumption. The formula for $inf\,\overline{PC}(K)$ was first reported in [16].

We can also study the computation of extremal languages that are relative closed (rather than prefix closed) and controllable. In this case we must consider the extremal solutions of the following two inequations:

$$pr(H) \cap L_m(G) \subseteq H; \quad pr(H)\Sigma_u \cap L(G) \subseteq pr(H),$$

where $H \subseteq L(G)$ is the variable of inequations. It can be argued as above that the *supremal relative closed and controllable sublanguage* of $K \subseteq L_m(G)$, denoted $sup\, RC(K)$, exists; however, the infimal relative closed and controllable sublanguage need not exist, as expected. Moreover, if we define

$$f_1(\cdot) := pr(\cdot); \quad f_2(\cdot) := pr(\cdot)\Sigma_u,$$

then

$$f_1^{-1}(f_2^{-1}(\cdot)) = f_1^{-1}[((\cdot)/\Sigma_u)\Sigma^*] = [((\cdot)/\Sigma_u)\Sigma^*]\Sigma^* = ((\cdot)/\Sigma_u)\Sigma^* = f_2^{-1}(\cdot).$$

Hence it follows from the first part of Theorem 6 that it is possible to modularly compute $sup\, RC(K)$, i.e.,

$$sup\, RC(K) = sup\, C(sup\, R(K)).$$

So the following iterative computation of Theorem 6 can be used for computing $sup\, RC(K)$:

- $K_0 := K - (L_m(G) - K)\Sigma^*$,
- $K_{i+1} := K_i - [(L(G) - K_i)/\Sigma_u]\Sigma^*$.

If $m \in \mathcal{N}$ is such that $K_{m+1} = K_m$, then $K_m = sup\, RC(K)$. It can be shown using arguments similar to those given in [26] that such an m exists whenever K and $L(G)$ are regular.

5.3 Extremal Normal Languages

Consider the definition of normality given by (13). It is of the form

$$f(H) \cap L(G) \subseteq g(H),$$

where $f(\cdot) = \mathcal{M}(pr(\cdot))$ which is disjunctive as well as idempotent (as disjunctivity and idempotency is preserved under composition of functions), and $g = pr$ which is monotone but not conjunctive. So the *supremal normal sublanguage* of a language $K \subseteq L(G)$, denoted $sup\, N(K)$ exists, as expected [20], whereas the infimal normal superlanguage need not exist, also as expected [20]. Since $\mathcal{M}^{-1} = \mathcal{M}$, the following iterative computation for computing $sup\, N(K)$ results from Corollary 3:

- $K_0 := K$,
- $K_{i+1} := K_i - [\mathcal{M}(L(G) - K_i)]\Sigma^*$.

If there exists $m \in \mathcal{N}$ such that $K_{m+1} = K_m$, then $K_m = sup\, N(K)$. That such an m exists can be shown whenever K and $L(G)$ are regular languages.

It can be argued as in the previous subsection that the *supremal prefix closed and normal sublanguage* and *infimal prefix closed and normal superlanguage* of a language $K \subseteq L(G)$, denoted $sup\, PN(K)$ and $inf\, \overline{PN}(K)$, respectively, exist. Moreover,

$$sup\, PN(K) = K - [\mathcal{M}(L(G) - K)]\Sigma^*; \quad inf\, \overline{PN}(K) = \mathcal{M}(pr(K)) \cap L(G).$$

The formula for $sup\, PN(K)$ was reported in [2, 13] under the assumption that K is prefix closed. We do not need this assumption here. Similarly, one can argue that the supremal relative closed and normal sublanguage of $K \subseteq L_m(G)$, denoted $sup\, RN(K)$, exists. However, the infimal relative closed and normal superlanguage need not exist. It can be verified that the hypothesis of Theorem 6 holds so that $sup\,RN(K) = sup\,N(sup\,R(K))$, i.e., a modular computation is possible.

6 Conclusion

We have studied the existence and computation of extremal solutions of a system of inequations defined over complete lattices. We have shown that under certain conditions our techniques provide closed form formulas for extremal solutions of a single inequation. We have demonstrated the applicability of our work to computation of supervisors in control of logical behaviors of DESs, represented as languages over a certain event set, under complete as well as partial observation. The results presented here can also be applied for computation of modular [34] and decentralized [19, 29] supervisors, and also for computing supervisors for controlling the non-terminating behaviors [25, 14, 32]. The work presented here thus presents a unifying approach for existence and computation of supervisory control policies in a variety of settings.

A Remark on Inverse Operation

In this appendix we present a justification for using the terminology *inverse* for the operation of conjugate of dual. We present an intuitive definition of inverse of a function defined over a power set lattice, say $(2^X, \subseteq)$, and show that the this definition coincides with one given earlier. Elements of the set X are called the *atoms* of the lattice. A power set lattice possesses the additional feature that its lattice elements can be described using the atoms of the lattice, i.e.,

$$\forall Y \subseteq X : Y = \{x \in X \mid \{x\} \cap Y \neq \emptyset\}.$$

Given a function $f : 2^X \to 2^X$, we have the following intuitive definition of its inverse:

$$\forall Y \subseteq X : f^{-1}(Y) := \{x \in X \mid f(\{x\}) \cap Y \neq \emptyset\}.$$

Then we have the following proposition:

Proposition 5 Consider a power set lattice $(2^X, \subseteq)$ and a function $f : 2^X \to 2^X$. If f is disjunctive, then

$$\forall Y, Z \subseteq X : [f(Y) \cap Z = \emptyset] \Leftrightarrow [Y \cap f^{-1}(Z) = \emptyset]. \tag{17}$$

Proof: We begin by proving the contrapositive of (17). First suppose $f(Y) \cap Z \neq \emptyset$. Since f is disjunctive, $f(Y) = \bigcup_{y \in Y} f(\{y\})$. Hence $f(Y) \cap Z \neq \emptyset$ implies there exists $y \in Y$ such that $f(\{y\}) \cap Z \neq \emptyset$. Consequently, $y \in f^{-1}(Z)$, i.e., $Y \cap f^{-1}(Z) \neq \emptyset$. Next suppose $Y \cap f^{-1}(Z) \neq \emptyset$. Then there exists $y \in Y$ such that $y \in f^{-1}(Z)$. This implies that $f(\{y\}) \cap Z \neq \emptyset$. Since f is disjunctive, it is also monotone. Hence $f(Y) \cap Z \neq \emptyset$. ∎

It follows from the first part of Proposition 3 that (17) uniquely defines the inverse of a disjunctive function. Thus the two definitions of inverse coincide. Note that the first definition of inverse is only defined for a disjunctive function over a CBL, whereas the second definition of inverse is defined for any function over a power set lattice. However, the equivalence of (17) only holds when the function is also disjunctive.

Example 6 Consider for example the prefix closure operation defined over the power set lattice of languages. Then it follows from above that

$$\forall H \subseteq \Sigma^* : pr^{-1}(H) = \{s \in \Sigma^* \mid pr(\{s\}) \cap H \neq \emptyset\} = ext(H),$$

as expected.

References

[1] F. Bracho. Continuously generated fixed points. *Theoretical Computer Science*, 68:303–317, 1989.

[2] R. D. Brandt, V. K. Garg, R. Kumar, F. Lin, S. I. Marcus, and W. M. Wonham. Formulas for calculating supremal controllable and normal sublanguages. *Systems and Control Letters*, 15(8):111–117, 1990.

[3] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya. Supervisory control of discrete event processes with partial observation. *IEEE Transactions on Automatic Control*, 33(3):249–260, 1988.

[4] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, UK, 1990.

[5] E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semantics*. Springer-Verlag, New York, 1990.

[6] P. Hitchcock and D.Park. Induction rules and termination proofs. In *Proceedings of IRIA Conference on Automata Languages and Programming Theory*, France, 1972.

[7] Y. C. Ho. Scanning the issue. In Y. C. Ho, editor, *Proceedings of IEEE: Special Issue on Discrete Event Systems*, volume 77, pages 3–6. IEEE press, New York, NY, 1989.

[8] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, Inc., Englewood Cliffs, NJ, 1985.

[9] C. A. R. Hoare. Fixed points of increasing functions. *Information Processing Letters*, 34:111–112, 1990.

[10] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.

[11] R. Kumar. Formulas for observability of discrete event systems. Technical report, Department of Electrical Engineering, University of Kentucky, Lexington, KY, 1883.

[12] R. Kumar. Formulas for observability of discrete event dynamical systems. In *Proceedings of 1993 Conference on Information Sciences and Systems*, pages 581–586, Johns Hopkins University, Baltimore, MD, March 1993.

[13] R. Kumar, V. K. Garg, and S. I. Marcus. On controllability and normality of discrete event dynamical systems. *Systems and Control Letters*, 17(3):157–168, 1991.

[14] R. Kumar, V. K. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Transactions on Automatic Control*, 37(12):1978–1985, December 1992.

[15] R. Kumar, V. K. Garg, and S. I. Marcus. Predicates and predicate transformers for supervisory control of discrete event systems. *IEEE Transactions on Automatic Control*, 38(2):232–247, February 1993.

[16] S. Lafortune and E. Chen. On the infimal closed and controllable superlanguage of a given language. *IEEE Transactions on Automatic Control*, 35(4):398–404, 1990.

[17] J. L. Lassez and M. L. Maher. Optimal fixed points of logic programs. *Theoretical Computer Science*, 39:15–25, 1985.

[18] J.L. Lassez, V.L. Nguyen, and E.A. Sonenberg. Fixed point theorems and semantics: A folk tale. *Information Processing Letters*, 14(3):112–116, 1982.

[19] F. Lin and W. M. Wonham. Decentralized supervisory control of discrete event systems. *Information Sciences*, 44:199–224, 1988.

[20] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, 1988.

[21] F. Lin and W. M. Wonham. Decentralized control and coordination of discrete-event systems with partial observation. *IEEE Transactions of Automatic Control*, 35(12):1330–1337, December 1990.

[22] Z. Manna and A. Shamir. The theoretical aspects of the optimal fixedpoint. *SIAM Journal of Computing*, 5(3), 1976.

[23] Z. Manna and A. Shamir. The convergence of functions to fixed points of recursive definitions. *Theoretical Computer Science*, 6(2):109–141, 1978.

[24] M. C. Paull. *Algorithm design: a recursion transformation framework*. Wiley Interscience, New York, NY, 1988.

[25] P. J. Ramadge. Some tractable supervisory control problems for discrete event systems modeled by buchi automata. *IEEE Transactions on Automatic Control*, 34(1):10–19, 1989.

[26] P. J. Ramadge and W. M. Wonham. On the supremal controllable sublanguage of a given language. *SIAM Journal of Control and Optimization*, 25(3):637–659, 1987.

[27] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.

[28] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of IEEE: Special Issue on Discrete Event Systems*, 77:81–98, 1989.

[29] K. Rudie and W. M. Wonham. Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, November 1992.

[30] D. Scott. Data types as lattices. *SIAM Journal of Computing*, 5:522–587, 1976.

[31] A. Tarski. A lattice-theoretical fixedpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[32] J. G. Thistle. *Control of infinite behavior of discrete-event systems*. PhD thesis, University of Toronto, Toronto, Canada, January 1991.

[33] P. Varaiya. Preface. In P. Varaiya and A. B. Khurzhanski, editors, *Discrete Event Systems: Models and Applications*, volume 103 of *Lecture Notes in Control and Information Sciences*, pages iv–vii. Springer-Verlag, New York, NY, 1988.

[34] W. M. Wonham and P. J. Ramadge. Modular supervisory control of discrete event systems. *Mathematics of Control Signals and Systems*, 1(1):13–30, 1988.