ABSTRACT

Using predicate transformers one can simultaneously model angelic and demonic nondeterminism. However, using standard relations one cannot simultaneously model both kinds of nondeterminism. Recently, in [5], I. Rewitzky has shown that by using up-closed binary multirelations both kinds of nondeterminism can be simultaneously modeled. In [6], Rewitzky and C. Brink note that many of the operations associated with binary multirelations are extensions of lifted operations. In this thesis, we propose a new relational model, the lifted binary multirelational model. We prove that the set of up-closed binary multirelations is embedded in the domain of relations between powersets. For that we use a Galois connection to show that there exists an isomorphism between lifted multirelations and up-closed binary multirelations. We show three properties that exist when upclosed binary multirelations are lifted to relations between powersets.
LIFTED MULTIRELATIONS AND PROGRAM SEMANTICS

A thesis submitted
to Kent State University in partial
fulfillment of the requirements for the
degree of Masters of Science

by
Jidesh Soudamini
December 2006
Thesis written by

Jidesh Soudamini
B.Tech., M.G University, 2001
M.S., Kent State University, 2006

Approved by

Dr. Austin Melton, Advisor, Computer Science Department

Dr. Robert Walker, Chair, Computer Science Department

Dr. John R D Stalvey, Dean, College of Arts and Sciences
# TABLE OF CONTENTS

1 ACKNOWLEDGMENTS iii

2 INTRODUCTION 1

3 METHODOLOGY 5

4 BINARY MULTIRELATIONS 9

5 GALOIS CONNECTIONS 10

6 LIFTED BINARY MULTIRELATIONS 16

7 APPLICATION 23

8 CONCLUSION AND FUTURE WORK 25

8 REFERENCES 26
ACKNOWLEDGEMENTS

It is a pleasure to thank the many people who made this thesis possible.

It is difficult to overstate my gratitude to my supervisor, Dr. Austin Melton. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped to make mathematics fun for me. Throughout my thesis-writing period, he provided encouragement, sound advice, good teaching, good company, and lots of good ideas. I would have been lost without him.

I wish to thank my best friends for helping me get through the difficult times, and for all the emotional support, comraderie, entertainment, and caring they provided. I am grateful to the secretaries and librarians in the computer science department for assisting me in many different ways. Marcy Curtiss deserve special mention.

Lastly, and most importantly, I wish to thank my parents. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.
CHAPTER 1

INTRODUCTION

Program semantics is a collection of techniques and methodologies which enable programmers and system designers to construct software systems in a systematic, effective and reliable manner. Mathematical techniques have assumed an increasingly important role in the development of computer systems. Discrete Mathematics is concerned with sets, functions, relations and logics. It provides a basis for comprehending and constructing mathematical arguments.

Using formal methods gives precision to the software development process but at the same time, the use of formal semantics can introduce nondeterminism in specifications. To facilitate modeling nondeterminism in specifications, special notations and semantics should be used in formal specification languages. Nondeterminism occurs when a program can choose between two or more values and this choice is not fixed in advance. Nondeterminism is an approach to deliberately leave a decision open, to abandon the exact predictability of future states. Nondeterminism is a fundamental tool for specifications to avoid laying down unnecessary details. A nondeterministic specification leaves more choice for the implementa-
tion, which can be used for optimizations. Even if this degree of freedom is not used in the envisaged first implementation, it greatly increases the likelihood that future enhancements and alternate implementations can be made compliant with the specification.

Nondeterminism often enhances the comprehensibility of specifications because the reader does not have to wonder why something has to be exactly in a certain way, when other choices would be as good. Many things are actually nondeterministic and should be acknowledged and specified as such.

Nondeterminism from an outside perspective often stems from information hiding, where the actual implementation is deterministic. A SQL database query without any sorting options returns an arbitrarily sorted list of records; a square root function returns an arbitrary value satisfying the specified precision. Both implementations are deterministic, but the outcomes are determined by hidden state components and implementation details.

There are two types of nondeterminism in a program; they are called angelic and demonic. Nondeterminism within existing components which is beyond our control is called demonic nondeterminism, and nondeterminism which we can control in our favor is called angelic nondeterminism. Angelic nondeterminism occurs when an angel makes a choice. We assume that the angel will choose the best possible outcome. On the other hand, demonic nondeterminism occurs when 'a demon' makes a choice in which we should not assume the choice made by the
demon will be for our advantage. Therefore we should be prepared for the worst possible outcome.

In this sense, we can consider program execution as a game, the rules of which are given by the specification. Demonic choices are moves made by an opponent, and angelic choices are our moves. The combined specification is correct, if we can make moves such that we can achieve the desired goal, no matter what the opponent does. Hence, such a combined specification can help to decide whether a given component is suitable to solve a certain task.

**Definition 1** Relations:

In general, if $X$ and $Y$ are sets, a set $R$ is a relation from $X$ to $Y$ if $R \subseteq X \times Y$. Often we work with relations built on one set. For example, we can let $X = Y = A$.

Then a relation on a set $A$ is a subset $R \subseteq A \times A$.

- $R$ is reflexive iff $xRx$ for all $x \in A$.
- $R$ is symmetric iff $xRy$ implies $yRx$ for all $x, y \in A$.
- $R$ is antisymmetric iff $xRy$ and $yRx$ implies $x = y$, $x, y \in A$.
- $R$ is transitive iff $xRy$ and $yRz$ implies $xRz$ for all $x, y, z \in A$.
- $R$ is a partial order iff $R$ is reflexive, antisymmetric and transitive.

**Definition 2** Partial Order:

A partially ordered set or a poset is a set taken together with a partial order on it. Formally, a partially ordered set is defined as an ordered pair $P = (X, \leq)$, where $X$ is called the ground set of $P$ and $\leq$ is the partial order of $P$. 

3
Definition 3 Partially Ordered Set:
A partially ordered set or a poset is a set taken together with a partial order on it. Formally, a partially ordered set is defined as an ordered pair $P = (X, \leq)$, where $X$ is called the ground set of $P$ and $\leq$ is the partial order of $P$.

Definition 4 Join (Least Upper Bound):
An upper bound of a subset $B \subseteq A$ of a poset $(A, \leq)$ is an element $a \in A$ such that for all $b \in B$ we have $b \leq a$. A least upper bound (LUB) or join of $B$ is an upper bound $a$ such that for all other upper bounds $a'$ we have $a \leq a'$.

Definition 5 Meet (Greatest Lower Bound):
A lower bound of a subset $B \subseteq A$ of a poset $(A, \leq)$ is an element $a \in A$ such that for all $b \in B$ we have $a \leq b$. A greatest lower bound (GLB) or meet of $B$ is a lower bound $a$ such that for all other lower bounds $a'$ we have $a' \leq a$.

Definition 6 Monotone Functions (Order-Preserving):
Let $(X, \leq)$ and $(Y, \preceq)$ be partially ordered sets, and let $f : X \rightarrow Y$ be a function. $f$ is monotone or order-preserving if and only if whenever $x \leq y$ then $f(x) \preceq f(y)$.

Definition 7 Composition of Binary relations:
The composition of a binary relations $r$ and $s$ where $r \subseteq X \times Y$ and $s \subseteq Y \times Z$ is defined as follows:
$$s \circ r = \{(x, z) | \exists y [(x, y) \in r \land (y, z) \in s]\}$$
CHAPTER 2

METHODOLOGY

Program semantics modeled using functions generally doesn’t model nondeterminism. It’s the relational models that are used to model nondeterminism. Relational models, functions and predicate transformers have been used before to model programs. However these methods have limitations. Functions only model deterministic programs. Binary relations only model programs with either angelic or demonic nondeterminism. On the other hand, a predicate transformer models program behavior backwards by mapping a set of outputs to a set of inputs. Both binary multirelations and predicate transformers relate inputs to sets of outputs and can model programs with both angelic and demonic nondeterminism.

Definition 8 Relational Semantics:

Relational semantics is the technique of describing imperative aspects of programming languages based on a simple set theoretic framework. In relational semantics relations can be considered as sets of pairs of states. Functions model determinism, meaning that outputs depend predictably on inputs. In relational semantics, a program is deterministic if the range of each input is always a singleton set. A
relation $R$ relates two values $x$ and $y$, written $xRy$, if and only if the corresponding program can terminate with output $y$ given input $x$.

For many years the realm of total functions was considered to be the natural domain in which to reason about programs. The limitation of this domain is that it can only be used to describe deterministic programs. More recently, the development of the relational calculus for program derivation has allowed programmers to reason about programs together with their specifications, which may be nondeterministic. Specifications expressed in this calculus can be manipulated and refined into functions that can be translated into an appropriate deterministic program. First, one of the beauties of the relational calculus is that specifications are simply relations. As such, they can be manipulated by all the familiar operations on relations, such as composition and inverse, as well as those developed specifically for the derivation of functional programs. The problem with the relational calculus is that the only operator that characterizes nondeterminism is union, so only one kind of nondeterminism can be described at a time [2].

**Definition 9** Predicate Transformers:

Predicate transformer semantics is an extension of Floyd-Hoare Logic invented by Dijkstra and extended and refined by other researchers. It was first introduced in Dijkstra’s paper ”Guarded commands, non-determinacy and formal derivation of programs” [7]. It is a method for defining the semantics of an imperative programming language by assigning to each command in the language a corresponding predicate transformer. A predicate transformer is a total function which maps predicates to predicates on the state space of a program. Predicates are
statements which are either true or false. We identify a predicate on a state space with the subset of all states for which the predicate is true.

An operation of a program is associated with two conditions, a precondition and a postcondition. A precondition defines the circumstances under which an operation will actually begin execution, that is, in an initial state. A postcondition defines what happens when an operation has completed its action, that is in a final state.

Weakest Preconditions: The changes of a state happen through transformations which are the results of operations. One way to describe the behavior of a program is to use logical notation and set theory. If we map the desired outcome $Q$, of an operations to the set of all states $P$ in which $S$ will achieve the desired outcome $Q$, then the set $P$ is termed as the weakest precondition. This set is called the weakest precondition of $S$ with respect to $Q$ which is denoted by $[S]Q$, where $Q$ is a logical statement for the desired outcome, $S$ is the operation to be carried out and $[S]Q$ is a logical statement describing the initial states that guarantee $Q$ after executing $S$. The sets of $[S]Q$ for various postconditions are preconditions.

Consider a program command $S$, a postcondition $Q$ and a precondition $P$ denoted jointly as $P{|[S]}|Q$. If the program $S$ is executed in an initial state where the precondition $P$ holds, then execution is guaranteed to terminate in a final state where postcondition $Q$ holds. The weakest precondition denoted by $wp(S,Q)$ (introduced by Dijkstra, 1970’s) is the predicate $P$ sufficient to ensure termination in a state described by $Q$. In other words, it is the weakest (largest) predicate $P$ such that $P{|[S]}|Q$ holds. We regard $S$ as a predicate transformer because it transforms the postcondition $Q$ into the weakest precondition $wp(S,Q)$. A pre-
condition $wp(S, Q)$ will guarantee that program $S$ terminates in a state described by the postcondition $Q$.

**Example 1** Consider a program described by the relation $R \subseteq \mathcal{N} \times \mathcal{N}$ where $R = \{(x, y)|y = x^2 + 1\}$ where $\mathcal{N}$ is the set of natural numbers, i.e., $\mathcal{N} = \{0, 1, 2, \ldots\}$.

Let the post condition $Q$ be $\{0, 1, 2, \ldots, 10\}$. Then $\{0, 1, 2, 3\}$ is the weakest precondition, that is the largest set such that each element is mapped into $Q$.

Let’s modify the program in the last example. Let’s define $S \subseteq \mathcal{N} \times \mathcal{N}$ such that $S = \{(x, y)|y = x^2 + 1 \text{ or } y = x\}$. Now if $Q = \{0, 1, \ldots, 10\}$, and we want to find the weakest precondition. We have a choice between angelic weakest precondition and demonic precondition. The angelic precondition is $\{0, 1, \ldots, 10\}$ because we assure an angel will always make the right choice. Thus, with an input of 6, the angel will choose 6 as the output so that the postcondition is satisfied. However, the demonic weakest precondition is $\{1, 2, 3\}$ because given an input of 6, for example, we assume no control and no ”right-choosing” angel has control. Thus, we cannot let 6 be in the precondition because its output might be 37.
CHAPTER 3

BINARY MULTIRELATIONS

**Definition 10** Let $X$ and $Y$ be sets. A binary multirelation $R$ from $X$ to $Y$ is a subset of $X \times \mathcal{P}(Y)$, that is, a set of ordered pairs $(x, Q)$ where $x \in X$ and $Q \subseteq Y$.

**Definition 11** Let $X$, $Y$ and $Z$ be sets, and let $R$ be a binary multirelation from $X$ to $Y$, and $S$ a binary multirelation from $Y$ to $Z$. The composition of $R$ followed by $S$ is a binary multirelation from $X$ to $Z$. The composition is denoted by $R; S$ and is such that

$$R; S = \{(x, C) \mid \exists B \subseteq Y. xRB \land B \subseteq \{y \in Y \mid ySC\}\}.$$
GALOIS CONNECTIONS

Let \((P, \leq)\) and \((Q, \preceq)\) be partially ordered sets, and \(f: P \to Q, g: Q \to P\) be functions. \((f, P, Q, g)\) or \((f, g)\) is a Galois connection iff \(\forall p \in P\) and \(\forall q \in Q\), \(p \leq g(q)\) iff \(f(p) \preceq q\).

**Proposition 1** Let \((P, \leq)\) and \((Q, \preceq)\) be partially ordered sets, and let \(f: P \to Q\) and \(g: Q \to P\) be functions. The following are equivalent.

i) \(\forall p \in P\) and \(\forall q \in Q\), \(p \leq g(q)\) iff \(f(p) \preceq q\)

ii) a) \(f\) and \(g\) are order preserving functions and

b) \(\forall p \in P, p \leq g(f(p))\) and \(\forall q \in Q, f(g(q)) \preceq q\)

**Proof:** Showing that i) implies ii). We suppose for \(\forall p \in P\) and \(\forall q \in Q\), that \(p \leq g(q)\) iff \(f(p) \preceq q\). Let \(p \in P\). Since \(f(p) = f(p)\), then \(f(p) \preceq f(p)\).

Hence, \(p \leq g(f(p))\). Likewise for \(q \in Q\), \(g(q) = g(q)\). So \(g(q) \leq g(q)\), and thus, \(f(g(q)) \preceq q\).

Let \(p_1, p_2 \in P\) such that \(p_1 \leq p_2\). Since \(p_2 \leq g(f(p_2))\), then \(p_1 \leq g(f(p_2))\). By i) with \(f(p_2) = q\), \(f(p_1) \preceq f(p_2)\). Let \(q_1, q_2 \in Q\) such that \(q_1 \preceq q_2\). Since \(f g(q_1) \preceq q_1\), then \(f g(q_1) \preceq q_2\). By i) with \(g(q_2) = p\), \(g(q_1) \preceq g(q_2)\). Therefore, i) implies ii).
Showing that ii) implies i). We suppose that \( f \) and \( g \) are order-preserving functions and that \( \forall p \in P, p \leq g(f(p)) \) and \( \forall q \in Q, f(g(q)) \leq q \). Let \( p \in P \) and \( q \in Q \), and suppose that \( p \leq g(q) \). Since \( f \) is order-preserving, \( f(p) \leq f(g(q)) \). However, by ii), \( f(g(q)) \leq q \). Thus, \( f(p) \leq q \). Let \( p \in P \) and \( q \in Q \), and suppose that \( f(p) \leq q \). Since \( g \) is order-preserving, \( g(f(p)) \leq g(q) \). However, by ii), \( p \leq g(f(p)) \). Thus, \( p \leq g(q) \).

Thus we now have an alternative definition for a Galois connection.

**Definition 12** Let \((P, \leq)\) and \((Q, \preceq)\) be partially ordered sets, and let \( f : P \to Q \) and \( g : Q \to P \) be functions. \((f, P, Q, g)\) or \((f, g)\) is a Galois connection, if \( f \) and \( g \) are order preserving functions and if \( \forall p \in P, p \leq g(f(p)) \) and \( \forall q \in Q, f(g(q)) \preceq q \).

We list and prove properties of Galois connections from [4], [1] and [3].

**Definition 13** Let \( f : P \to Q \) and \( g : Q \to P \) be functions. \( g \) is quasi-inverse for \( f \) iff \( fgf = f \), and \( f \) is a quasi-inverse for \( g \), i.e., \( gfg = g \).

**Proposition 2** Let \((f, P, Q, g)\) be a Galois connection. \( fgf = f \) and \( gfg = g \) i.e., \( f \) and \( g \) are quasi-inverses.

**Proof:** Let \( p \in P \). By the alternate definition \( p \leq g(f(p)) \). Since \( f \) is order preserving, \( f(p) \preceq f(g(f(p))) \). Also, \( g(f(p)) \preceq g(f(p)) \). Hence, by the first definition and the facts that \( g(f(p)) \in P \) and that \( f(p) \in Q \), we have \( f(g(f(p))) \preceq f(p) \). Hence \( f(p) \preceq f(g(f(p))) \) and \( f(g(f(p))) \preceq f(p) \). Therefore, \( f(p) = f(g(f(p))) \).

Since \( p \) was arbitrarily chosen \( f = fgf \).

Showing that \( gfg = g \). Let \( q \in Q \). By the alternate definition \( f(g(q)) \preceq q \).

Since \( g \) is order preserving, \( g(f(g(q))) \leq g(q) \). Also, \( f(g(q)) \leq f(g(q)) \). Hence,
by the first definition and the facts that \( f(g(q)) \in Q \) and that \( g(q) \in P \), we have 
\[ g(q) \leq g(f(g(q))). \]
Hence \( g(q) \leq g(f(g(q))) \) and \( g(f(g(q))) \leq g(q) \). Therefore, 
\[ g(q) = g(f(g(q))). \]
Since \( q \) was arbitrarily chosen \( g = gfg. \)

**Proposition 3** Let \( f : P \to Q, g : Q \to P \) be functions. Then the following are equivalent.

a) \( fgf = f \) and \( gfg = g \)

b) For all \( p \in P, p \in g(Q) \) iff \( p = gf(p) \) and for all \( q \in Q, q \in f(P) \) iff \( q = fg(q) \)

**Proof:** Showing a) implies b). Suppose \( fgf = f \) and \( gfg = g \). Let \( p \in P \).

Suppose \( p \in g(Q) \). Then there exists \( q \in Q \) such that \( p = g(q) \). Since \( gfg = g \),
then \( gfg(p) = g(q) \). And since \( g(q) = p \), then \( gf(p) = p \). Thus, \( p \in g(Q) \) implies
\( p = gf(p) \). Now suppose that \( p \in P \), such that \( p = gf(p) \). Since \( f(p) \in Q \), then
\( p \in g(Q) \).

Suppose \( fgf = f \) and \( gfg = g \). Let \( q \in Q \). Suppose \( q \in f(P) \). Then there
exists \( p \in P \) such that \( q = f(p) \). Since \( fgf = f \), then \( fgf(p) = f(p) \). And since
\( f(p) = q \), then \( fg(q) = q \). Thus, \( q \in f(P) \) implies \( q = fg(q) \). Now suppose that
\( q \in Q \), such that \( q = fg(q) \). Since \( g(q) \in P \), then \( q \in f(P) \).

Showing b) implies a). Let \( p \in P \), then \( f(p) \in f(P) \). Therefore, \( f(p) = fgf(p) \).

Since \( p \) was an arbitrary element of \( P \), \( f(p) = fgf(p) \) for every \( p \in P \). Hence
\( f = fgf \). Now showing \( g = gfg \). Let \( q \in Q \), then \( g(q) \in g(Q) \). Therefore,
g \( g(q) = g(f(g(q))) \). Since \( q \) was an arbitrary element of \( Q \), \( g(q) = gfg(q) \) for every
\( q \in Q \).

This result leads us to a new proposition.
Proposition 4 If \((f, P, Q, g)\) is a Galois connection, then for every \(p \in P\), \(p \in g[Q]\) iff \(p = gf(p)\), and for every \(q \in Q\), \(q \in f[P]\) iff \(q = fg(q)\).

Notation 1 Let \(P^* = g[Q]\) and let \(Q^* = f[P]\). Since the image of \(P\) under \(f\) is \(Q^*\), we can define \(f^* : P^* \to Q^*\) such that for each \(p \in P^*, f^*(p) = f(p)\). Likewise we can define \(g^* : Q^* \to P^*\) such that for each \(q \in Q^*, g^*(q) = g(q)\).

Proposition 5 \(P^*\) and \(Q^*\) are isomorphic partially ordered sets with \(f^*\) and \(g^*\) being inverse partial order isomorphisms.

Proof: Since \(f\) and \(g\) are order-preserving functions, so are \(f^*\) and \(g^*\). We need to show for each \(p \in P^*\) that \(p = g^*f^*(p)\) and for each \(q \in Q^*\) that \(q = f^*g^*(q)\).

However, these are the results from the proposition 4. 

Proposition 6 Let \((f, P, Q, g)\) be a Galois connection. \(f\) preserves joins, that is, if \(A \subseteq P\) and if \(\bigvee A\) exists in \(P\) then \(\bigvee f[A]\) exists in \(Q\), and \(f(\bigvee A) = \bigvee f[A]\).

Proof: Let \(A \subseteq P\), and suppose \(\bigvee A\) exists in \(P\). Since \(\bigvee A\) is an upper bound for \(A\) and since \(f\) is order-preserving, then \(f(\bigvee A)\) is an upper bound for \(f[A]\). Let \(q\) be an upper bound for \(f[A]\). Since \(g\) is order-preserving, then \(g(q)\) is an upper bound for \(gf[A]\), i.e., \(\forall p \in A, gf(p) \leq g(q)\). However, for each \(p \in A, p \leq gf(p)\) and \(gf(p) \leq g(q)\). Thus, \(\forall p \in A, p \leq g(q)\) and so \(g(q)\) is an upper bound for \(A\). Since \(\bigvee A\) is the least upper bound for \(A\), then \(\bigvee A \leq g(q)\), and since \(f\) is order-preserving, \(f(\bigvee A) \leq fg(q)\). Recalling that \(\forall q \in Q, fg(q) \leq q\), gives us that \(f(\bigvee A) \leq q\), hence \(f(\bigvee A) = \bigvee f[A]\).

Proposition 7 Let \((f, P, Q, g)\) be Galois connection. \(g\) preserves meets, that is, if \(B \subseteq Q\) and if \(\bigwedge B\) exists in \(Q\), then \(\bigwedge g[B]\) exists in \(P\) and \(g(\bigwedge B) = \bigwedge g[B]\).

13
Proof: Let $B \subseteq Q$, and suppose $\bigwedge B$ exists in $Q$. Since $\bigwedge B$ is a lower bound for $B$ and since $g$ is order-preserving, then $g(\bigwedge B)$ is a lower bound for $g[B]$. Let $p$ be a lower bound for $g[B]$. Since $f$ is order-preserving, then $f(p)$ is a lower bound for $fg[B]$. However, for each $q \in B$, $fg(q) \preceq q$ and $f(p) \preceq fg(q)$. Thus, $f(p)$ is a lower bound for $B$. Since $\bigwedge B$ is the greatest lower bound for $B$, then $f(p) \preceq \bigwedge B$, and since $g$ is order-preserving, $gf(p) \preceq g(\bigwedge B)$. Recalling that $p \preceq gf(p)$, gives us that $p \preceq g(\bigwedge B)$. Hence $g(\bigwedge B) = \bigwedge g[B]$. •

Proposition 8 Let $(f, P, Q, g)$ be a Galois connection. Then $f$ and $g$ determine each other. In fact, for each $p \in P$, $f(p) = \bigwedge \{q \in Q | p \preceq g(q)\}$, and, for each $q \in Q$, $g(q) = \bigvee \{p \in P | f(p) \preceq q\}$.

Proof: For $p \in P$, let $Q_p = \{q \in Q | p \preceq g(q)\}$. We want to show that $f(p)$ is a lower bound for $Q_p$. For each $q \in Q_p$, $p \preceq g(q)$. Thus, $f(p) \preceq fg(q)$. However, $fg(q) \preceq q$, and therefore, $f(p) \preceq q$ for each $q \in Q_p$. It follows that $f(p)$ is a lower bound for $Q_p$. Also note that $p \preceq g(f(p))$. Therefore, $f(p) \in Q_p$. It follows that any lower bound for $Q_p$ must be less than or equal to $f(p)$. Thus, $f(p)$ is the greatest lower bound, i.e., $f(p) = \bigwedge \{q \in Q | p \preceq g(q)\}$.

For $q \in Q$, let $P_q = \{p \in P | f(p) \preceq q\}$. We want to show that $g(q)$ is an upper bound for $P_q$. For each $p \in P_q$, $f(p) \preceq q$. Thus, $gf(p) \preceq g(q)$. However, $p \preceq gf(p)$, and therefore, $p \preceq g(q)$ for each $p \in P_q$. It follows that $g(q)$ is an upper bound for $P_q$. Also note that $fg(q) \preceq q$. Therefore, $g(q) \in P_q$. It follows that any upper bound for $P_q$ must be greater than or equal to $g(q)$. Thus, $g(q)$ is the least upper bound, i.e., $g(q) = \bigvee \{p \in P | f(p) \preceq q\}$. •

Proposition 9 Let $(f, P, Q, g)$ be a Galois connection $f$ is one-to-one iff $g$ is onto
iff $gf = id_P$, and $g$ is one-to-one iff $f$ is onto iff $fg = id_Q$.

**Proof:** We want to show that $f$ being one-to-one implies that $g$ is onto. Let $p \in P$. We know that $f(p) = gfg(p)$. Since $f$ is one-to-one, $p = gf(p)$. Thus, $p \in g[Q]$, and $g$ is onto. We want to show that $g$ being onto, implies $gf = id_P$, i.e., we want to show for each $p \in P$, that $p = ggf(p)$. Let $p \in P$. Since $g$ is onto, $p = g[Q]$. Thus, $p \in g[Q]$, and $p = gf(p)$. Therefore, $gf = id_P$. We want to show if $gf = id_P$, then $f$ is one-to-one. Let $p_1, p_2 \in P$ such that $p_1 \neq p_2$. Since $gf(p_1) = p_1$ and $gf(p_2) = p_2$, then $gf(p_1) \neq gf(p_2)$. Since $g$ is a well-defined function, $f(p_1) \neq f(p_2)$ and hence, $f$ is one-to-one. The second series of equivalence can be proved similarly. •

**Proposition 10** Let $(f, (P, \leq), (Q, \sqsubseteq), g)$ and $(h, (Q, \sqsubseteq), (R, \preceq), k)$ are Galois connections, then $(h \circ f, (P, \leq), (R, \preceq), g \circ k)$ is also a Galois connection.

**Proof:** Since $f, g, h$ and $k$ are order-preserving, then $h \circ f$ and $g \circ k$ are also order-preserving. Let $p \in P$. Since $(h, (Q, R, k)$ is a Galois connection, then $f(p) \sqsubseteq khf(p)$ and since $g$ is order-preserving, $gf(p) \leq (g \circ k)(h \circ f)(p)$. However, $p \leq gf(p)$, and so, $p \leq (g \circ k)(h \circ f)(p)$. Let $r \in R$. Since $(f, P, Q, g)$ is a Galois connection, then $fgk(r) \sqsubseteq k(r)$ and since $h$ is order-preserving, $(h \circ f)(g \circ k)(r) \preceq hk(r)$. However, $hk(r) \preceq r$, and so, $(h \circ f)(g \circ k)(r) \preceq r$. 

15
CHAPTER 5

LIFTED BINARY MULTIRELATIONS

For given sets $X$ and $Y$, we show that the set of relations from $X$ to $Y$ can be embedded into the set of binary multirelations from $X$ to $(Y)$ and into the set of relations from $\mathcal{P}(X)$ to $\mathcal{P}(Y)$ and that the set of binary multirelations from $X$ to $Y$ can be embedded into the set of relations from $\mathcal{P}(X)$ to $\mathcal{P}(Y)$. Further, composition of binary multirelations corresponds to the usual composition of relations from $\mathcal{P}(X)$ to $\mathcal{P}(Y)$ when the binary multirelations are assumed to be up-closed. Since the results in [5] and [6] assume up-closed binary multirelations when showing that binary multirelations are equivalent to monotone predicate transformers, then the results obtained using binary multirelations may also be obtained using the images of binary multirelations in $\mathcal{P}(X) \times \mathcal{P}(Y)$. We call these images lifted relations.

**Notation 2** Let $X$ and $Y$ be sets. Let $\mathcal{A}$ be the set of all relations from $\mathcal{P}(X)$ to $\mathcal{P}(Y)$; let $\mathcal{B}$ be the set of all binary multirelations from $X$ to $Y$; and let $\mathcal{C}$ be the set of all relations from $X$ to $Y$. Thus, $\mathcal{A} = \mathcal{P}(\mathcal{P}(X) \times \mathcal{P}(Y))$; $\mathcal{B} = \mathcal{P}(X \times \mathcal{P}(Y))$ and $\mathcal{C} = \mathcal{P}(X \times Y)$. 

Define $f_1 : \mathcal{A} \to \mathcal{B}$ such that for $R \subseteq \mathcal{P}(X) \times \mathcal{P}(Y)$, $(x, N) \in f_1(R)$ if and only if there exists $M \subseteq X$ with $x \in M$ and $(M, N) \in R$.

Define $g_1 : \mathcal{B} \to \mathcal{A}$ such that for $S \subseteq X \times \mathcal{P}(Y)$, $(M, N) \in g_1(S)$ if and only if $(m, N) \in S$ for each $m \in M$.

Define $f_2 : \mathcal{B} \to \mathcal{C}$ such that for $S \subseteq \mathcal{B}$, $(m, n) \in f_2(S)$ if and only if $(m, N) \in S$ for each $n \in N$.

Define $g_2 : \mathcal{C} \to \mathcal{B}$ such that for $r \in \mathcal{C}$, $(m, N) \in g_2(r)$ if and only if $(m, n) \in r$ for each $n \in N$.

Define $f_3 : \mathcal{A} \to \mathcal{C}$ such that for $R \in \mathcal{A}$, $(m, n) \in f_3(R)$ if and only if there exist $M \subseteq X$ and $N \subseteq Y$ with $m \in M$, $n \in N$, and $(M, N) \in R$.

Define $g_3 : \mathcal{C} \to \mathcal{A}$ such that for $r \in \mathcal{C}$, $(M, N) \in g_3(r)$ if and only if for each $m \in M$ and for each $n \in N$, $(m, n) \in r$. 

![Diagram](image-url)
For the following lemmas and Theorem 1, the partial orderings on $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are subset ordering.

**Lemma 1** $f_1 : \mathcal{A} \to \mathcal{B}$ is order preserving.

**Proof:** Let $R, R' \in \mathcal{A}$ such that $R \subseteq R'$. Let $(x, N) \in f_1(R)$. Then there exists $M \subseteq X$ such that $x \in M$ and $(M, N) \in R$. Since $R \subseteq R'$ then $(M, N) \in R'$, and therefore, $(x, N) \in f_1(R')$. Thus, $f_1(R) \subseteq f_1(R')$ and $f_1$ is order-preserving.

**Lemma 2** $g_1 : \mathcal{B} \to \mathcal{A}$ is order preserving.

**Proof:** Let $S, S' \in \mathcal{B}$ such that $S \subseteq S'$. Let $(M, N) \in g_1(S)$. Then for each $m \in M$, $(m, N) \in S$. Since $S \subseteq S'$, we also have $(m, N) \in S'$ for each $m \in M$. Therefore $(M, N) \in g_1(S')$. Thus, $g_1(S) \subseteq g_1(S')$ and $g_1$ is order-preserving.

**Lemma 3** $f_2 : \mathcal{B} \to \mathcal{C}$ is order preserving.

**Proof:** Let $S, S' \in \mathcal{B}$ such that $S \subseteq S'$. Then for each $m \in M$, $(m, N) \in S$. Since $S \subseteq S'$, $(m, N) \in S'$. Therefore $(m, n) \in f_2(S')$. Thus $f_2(S) \subseteq f_2(S')$ and $f_2$ is order-preserving.

**Lemma 4** $g_2 : \mathcal{C} \to \mathcal{B}$ is order preserving.

**Proof:** Let $(m, N) \in g_2(r)$. Then there exists $(m, n) \in r$ for each $n \in N$. Since $r \subseteq r'$, then $(m, n) \in r'$ for each $n \in N$. Therefore $(m, N) \in g_2(r')$. Thus $g_2(r) \subseteq g_2(r')$, and $g_2$ is order-preserving.

1 $(f_1, (\mathcal{A}, \subseteq), (\mathcal{B}, \subseteq), g_1)$, $(f_2, (\mathcal{B}, \subseteq), (\mathcal{C}, \subseteq), g_2)$, and $(f_3, (\mathcal{A}, \subseteq), (\mathcal{C}, \subseteq), g_3)$ are all Galois connections.
Proof: Lemmas 1 and 2 show that \( f_1 \) and \( g_1 \) are order-preserving functions. Let \( R \in \mathcal{A} \), i.e., \( R \subseteq \mathcal{P}(X) \times \mathcal{P}(Y) \). Let \((M, N) \in R\). Then \( \forall m \in M, (m, N) \in f_1(R) \), which implies that \((M, N) \in g_1 f_1(R)\). Thus, \( R \subseteq g_1 f_1(R) \).

Let \( S \in \mathcal{B} \), i.e., \( S \subseteq X \times \mathcal{P}(Y) \). Let \((m, N) \in f_1 g_1(S)\). Since \((m, N) \in f_1 g_1(S)\), then there exists \( M \subseteq X \) such \( m \in M \) and \((M, N) \in g_1(S)\). Since \( m \in M \), it follows that \((m, N) \in S\). Therefore, \( f_1 g_1(S) \subseteq S \). Thus, \((f_1, \mathcal{A}, \mathcal{B}, g_1)\) is a Galois connection.

Lemmas 3 and 4 show that \( f_2 \) and \( g_2 \) are order-preserving functions. Let \( S \in \mathcal{B} \).

Let \((m, N) \in S\). Since \((m, N) \in S\), then \((m, n) \in f_2(S)\) for each \( n \in N \). Thus it follows that \((m, N) \in g_2 f_2(S)\). Therefore, \( S \subseteq g_2 f_2(S) \).

Let \( r \in \mathcal{C} \), i.e., \( r \subseteq X \times Y \). Let \((m, n) \in f_2 g_2(r)\). Since \((m, n) \in f_2 g_2(r)\), then there exists \( N \subseteq Y \) such \( n \in N \) and \((m, N) \in g_2(r)\). Further, since \( n \in N \), then \((m, n) \in r\). Thus, \( f_2 g_2(r) \subseteq r \).

Proposition 11 \((f_3, g_3)\) is a Galois connection.

Proof: We claim that \( f_3 = f_2 \circ f_1 \). Let \( R \in \mathcal{A} \). \((m, n) \in (f_2 \circ f_1)(R)\) iff \((m, n) \in f_2(f_1(R))\) iff \( \exists M \subseteq X \) with \( m \in M \) and \((m, N) \in f_1(R)\) iff \( \exists M \subseteq X \) with \( m \in M \) and \( \exists N \subseteq Y \) with \( n \in N \) and \((m, n) \in R\) iff \( \exists M \subseteq X \) and \( N \subseteq Y \) with \( m \in M \) and \( n \in N \) such that \((m, n) \in R\) iff \((m, n) \in f_2 g_2(r)\).

We claim \( g_3 = g_1 \circ g_2 \). Let \( r \in \mathcal{C} \). \((M, N) \in (g_1 \circ g_2)(r)\) iff \((M, N) \in g_1(g_2(r))\) iff \((M, N) \in g_2(r)\) \( \forall m \in M \) iff \((m, n) \in r \) \( \forall n \in N \) and \( \forall m \in M \) iff \((m, n) \in r \) \( \forall m \in M \) and \( \forall n \in N \) iff \((M, N) \in g_3(r)\).

Therefore, \((f_3, g_3)\) is a Galois connection by Proposition 10.
Corollary 1 \( f_3 : \mathcal{A} \to \mathcal{C} \) is order preserving.

Corollary 2 \( g_3 : \mathcal{C} \to \mathcal{A} \) is order preserving.

Remark 1 As is done in [5], we restrict our attention to the case where \( X = Y \).

We will call this common set \( S \). We will continue to use the Galois connection of Theorem 1, except now, for example, \( \mathcal{A} \) is the set of all relations \( R \) such that \( R \subseteq \mathcal{P}(S) \times \mathcal{P}(S) \); i.e., \( \mathcal{A} = \mathcal{P}(\mathcal{P}(S) \times \mathcal{P}(S)) \).

Proposition 12 Let \( R \) and \( T \) be up-closed binary multirelations from \( S \) to \( S \). Then \( g_1(T) \circ g_1(R) = g_1(R;T) \).

Proof Suppose \((A, C) \in g_1(T) \circ g_1(R)\). Then there exists \( B \subseteq S \) such that \((A, B) \in g_1(R)\) and \((B, C) \in g_1(T)\). Thus, for each \( a \in A \), \( aRB \), and for each \( b \in B \), \( bTC \). Thus, \( B \subseteq \{y \in S|yTC\} \). Therefore, for each \( a \in A \), \( aRB \), and \( B \subseteq \{y \in S|zTC\} \). Hence, for each \( a \in A \), \((a, C) \in R;T \). Thus, \((A, C) \in g_1(R;T)\), and \( g_1(T) \circ g_1(R) \subseteq g_1(R;T) \).

Suppose \((D, F) \in g_1(R;T)\). Then for each \( d \in D \), \((d, F) \in R;T \). Thus, for each \( d \in D \), there exists \( E_d \subseteq S \) such that \( dRE_d \) and \( E_d \subseteq \{y \in S|yTF\} \). Since for each \( d \in D \), \( E_d \subseteq \{y \in S|yTF\} \), then if \( E = \bigcup_{d \in D} E_d \), then \( E \subseteq \{y \in S|yTF\} \). Hence, \((E, F) \in g(T)\). Further, since, for each \( d \in D \), \( dRE_d \) and since \( R \) is up-closed, then \( dRE \) for each \( d \in D \). Thus, \((D, E) \in g_1(R)\). It follows that \((D, F) \in g_1(T) \circ g_1(R)\) and \( g_1(R;T) \subseteq g_1(T) \circ g_1(R) \). Hence, \( g_1(T) \circ g_1(R) = g_1(R;T) \). •

Definition 14 Let \( R \in \mathcal{A} \). \( R \) is down-closed in the first coordinate if whenever \((M, N) \in R \) and \( M' \subseteq M \), then \((M', N) \in R \). \( R \) is up-closed in the second coordinate if whenever \((M, N) \in R \) and \( N \subseteq N' \), then \((M, N') \in R \).
Proposition 13 If $R$ is a binary multirelation, then $g_1(R)$ is down-closed in the first coordinate.

Proof: Let $(M, N) \in g_1(R)$ and let $M' \subseteq M$. Since $(M, N) \in g_1(R)$, then $\forall m \in M$, $mRN$. Since $M' \subseteq M$, then $mRN$, $\forall m \in M'$. Therefore $(M', N) \in g_1(R)$, and $g_1(R)$ is down-closed in the first coordinate.

Proposition 14 If $R$ is an up-closed binary multirelation, then $g_1(R)$ is up-closed in the second coordinate.

Proof: Let $(M, N) \in g_1(R)$ and suppose that $N \subseteq N'$. Since $(M, N) \in g_1(R)$, then $\forall m \in M$, $mRN$. However, since $R$ is up-closed, then $\forall m \in M$, $mRN'$. Hence $(M, N') \in g_1(R)$, and thus, $g_1(R)$ is up-closed in the second coordinate.

Proposition 15 $g_1$ is one-to-one.

Proof: Let $R$ and $T$ be multirelations from $S$ to $S$ such that $R \neq T$. Thus, without loss of generality there exists $(a, N) \in R - T$. Then $(\{a\}, N) \in g_1(R)$ but $(\{a\}, N) \notin g_1(T)$, and $g_1$ is one-to-one.

Notation 3 Let $B \uparrow$ be the set of up-closed binary multirelations on $S$.

Definition 15 Let $R \in A$. We say $R$ has the union property if whenever $\mathcal{M} \subseteq \mathcal{P}(S)$ with $(M, N) \in R$, $\forall M \in \mathcal{M}$, then $(\bigcup \mathcal{M}, N) \in R$.

Lemma 5 Let $T \in A$ such that $T$ is up-closed in the second coordinate. Then $f_1(T)$ is an up-closed binary multirelation.

Proof: Let $(a, N) \in f_1(T)$ and let $N \subseteq N'$. Since $(a, N) \in f_1(T)$, then there exists $M \subseteq S$ such that $a \in M$ and $(M, N) \in T$. Since $T$ is up-closed in the second
coordinate, then \((M, N') \in T\). Since \(a \in M\), then \((a, N') \in f_1(T)\). Therefore, 
\(f_1(T)\) is up-closed.

2 \(R \in g_1[\mathcal{B} \uparrow]\) iff \(R\) is down-closed in the first coordinate, up-closed in the second coordinate and has the union property.

**Proof:** Let \(T \in \mathcal{B} \uparrow\). Then \(g_1(T)\) is down-closed in the first coordinate and up-closed in the second by propositions 13 and 14. Suppose \(\mathcal{M} \subseteq \mathcal{P}(S)\) such that 
\(\forall M \in \mathcal{M}, (M, N) \in g_1(T)\). Let \(M \in \mathcal{M}\), since \((M, N) \in g_1(T)\), then \(\forall m \in M, (m, N) \in T\). Since \(M\) was arbitrarily chosen in \(\mathcal{M}\), then \(\forall M \in \mathcal{M}\) and \(\forall m \in M, (m, N) \in T\). Hence, \(\forall m \in \bigcup \mathcal{M}, (m, N) \in T\) and thus, \((\bigcup \mathcal{M}, N) \in g_1(T)\).

Let \(T\) be a subset of \(\mathcal{P}(S) \times \mathcal{P}(S)\) such that \(T\) is up-closed in the first coordinate, down-closed in the second, and has the union property. Clearly \(T \subseteq g_1 f_1(T)\) we want to show that \(g_1 f_1(T) \subseteq T\), and hence, \(T = g_1 f_1(T)\). Then from Proposition 4, we will have that \(T \in g_1[\mathcal{B}]\).

Let \((M, N) \in g_1 f_1(T)\). Then \(\forall m \in M, (m, N) \in f_1(T)\). Thus, for each \(m \in M\), there exists \(L_m \subseteq S\) such that \(m \in L_m\) and \((L_m, N) \in T\). Let \(\mathcal{L} = \{L_m | m \in M\}\).

Since \(T\) has the union property then \((\bigcup \mathcal{L}, N) \in T\). However, since \(M \subseteq \bigcup \mathcal{L}\) and \(T\) is down-closed in the first coordinate, \((M, N) \in T\). Hence, \(T = g_1 f_1(T)\).

Therefore, \(T \in g_1[\mathcal{B}]\). Further, since \(T\) is up-closed in second coordinate, then by Lemma 5, \(f_1(T)\) is an up-closed binary multirelation, i.e., \(f_1(T) \in \mathcal{B} \uparrow\). It follows that \(T = g_1 f_1(T) \in g_1[\mathcal{B} \uparrow]\).
Tic-Tac-Toe:

Tic-Tac-Toe is a well known game played by two persons who alternately place X’s and O’s upon a $3 \times 3$ playing board. The players first decide who will go first. Play proceeds with the opponents alternately placing their marks in any unoccupied cell with the first player always using an X and the opponent always using an O. The object of the game is to be the first player with 3 marks in a row, where a row can be either vertical, horizontal, or diagonal. If all the cells become filled without someone winning, the game is a draw.

We want to define a relation so that whenever possible on every move our agent can ensure of a win or a draw. We define the sample space $S$ as $S = \{(s_1, \ldots, s_9) | s_i \in \{b, X, O\} \text{ for } 1 \leq i \leq 9\}$ such that $(s_1, \ldots, s_9)$ is a row dominated $3 \times 3$ matrix.

Let $R$ be the relation that defines the next move whenever a win or a draw can be ensured, $R \subseteq \mathcal{P}(S) \times \mathcal{P}(S)$. Usually $A$ contains a single element which represents the board after the opponent has made his move. $A$ is associated to set $B$ that has all possible next moves which ensure a win or a draw. Before making a move
our agent looks for an $A$ that has a element matching to the current game state and selects a move $b$ in $B$. In fact $B$ contains all moves which will ensure a win or a draw.

For example:

$$A=\left\{ \begin{array}{|c|c|}
X & 0 \\
X & X \\
0 & X \\
\end{array} \right\}, \quad B=\left\{ \begin{array}{|c|c|}
X & X \\
0 & 0 \\
X & X \\
\end{array} \right\}, \quad \left\{ \begin{array}{|c|c|}
X & 0 \\
X & 0 \\
0 & X \\
\end{array} \right\}$$

There are cases with 2 elements in $A$ and a next move can take to a common state.

For example:

$$A=\left\{ \begin{array}{|c|c|}
X & O \\
X & X \\
0 & X \\
\end{array} \right\}, \quad B=\left\{ \begin{array}{|c|c|}
X & 0 \\
X & 0 \\
0 & X \\
\end{array} \right\}$$

The relation $\mathcal{R}$ is a generating relation, and that $\mathcal{R}$ can be used to get a corresponding relation that is down-closed in the first coordinate and up-closed in the second coordinate.
CHAPTER 7

CONCLUSION AND FUTURE WORK

Using Galois connections, we have shown that there exists an isomorphism between up-closed binary multirelations and lifted multirelations where the image of up-closed binary multirelation is is down-closed in the first coordinate, up-closed in the second coordinate and has the union property. As the new model takes predicates to predicates, there exists new possibilities for relations which simultaneously model angelic and demonic nondeterminism and "act like" predicate transformers. We suggest as future work, defining properties of lifted multirelations comparable to binary multirelation's properties and modelling angelic and demonic nondeterminism using lifted multirelations.
REFERENCES


