An Isomorphism between Abstract Polyhedral Cones and Definite Boolean Functions

Florence Benoy and Andy King
Computing Laboratory, University of Kent at Canterbury,
CT2 7NF, UK. {pmb6, a.m.king}@ukc.ac.uk

Abstract

Polyhedral cones can be represented by sets of linear inequalities that express inter-variable relationships. These inequalities express inter-variable relationships that are quantified by the ratios between the variable coefficients. However, linear inequalities over a non-negative variable domain with only unit variable coefficients and no constants other than zero can represent relationships that can be valid in non-numeric domains. For instance, if variables are either non-negative or zero itself, that is, a strictly two-point domain, then $\{0 \leq x, 0 \leq y, x \leq y\}$, expresses a dependency between x and y, since if y is known to be zero, then so is x. By defining an abstraction operator that effectively puts aside the scaling coefficients whilst retaining the inter-variable aspect of these relationships polyhedral cones can express the same dependency information as *Def*, a class of Boolean function. Boolean functions are considered over a fixed finite set of variables and *Def* is a subset of the positive Boolean functions, which return the value *true* when every variable returns *true*. *Def* is a complete lattice ordered by logical consequence and it will be shown that the abstract cones also form a complete lattice, ordered by set inclusion, that is isomorphic to *Def*.

1 Introduction

Mathematical structures that allow the characterisation of inter-variable dependencies can be used to capture the results of program analyses. For example, in the context of Logic Programming, downward closed properties such as groundness, can be captured by Boolean functions; the answer pattern for the query `p(x, y)`,might be represented by the formula $x \leftrightarrow y$ indicating that if `p` succeeds, then x is ground iff y is ground. Information with respect to groundness is valuable both in its own right and as a contributing factor in other analyses. It is well known that Boolean functions are useful for characterising inter-variable dependency relationships. There are various classes of Boolean function that can be used in this way, particularly the class of functions which map to true when all the variables themselves map to true. These functions are known as positive Boolean functions and the class is referred to as *Pos*. The functions in *Pos*, that are definite are referred to as *Def*, and although generally *Pos*, is more expressive than *Def*, it has been shown [A. King, P. Hill and J.Smaus 1998] that an efficient implementation of *Def*, as a Share based set-of-sets representation can produce dependency analyses that have better scaling behaviour than some *Pos* implementations and even compare favourably with *Pos*, for speed.

Linear equations and inequalities can be viewed as a characterisation of inter-variable relationships over numeric domains. The scalar coefficients of variables in those equations or inequalities quantify the ratios that qualify the relationships. For example $x \leq 3y$ expresses a relationship between the variables x and y such that the upper bound on x is three times the upper bound on y. In fact, solving linear equations and inequalities amounts to deducing the smallest possible range of values for each variable. If both x and y are constrained to non-negative values this can be represented by the set of non-strict linear inequalities $\{0 \leq x, 0 \leq y, x \leq y\}$. Interestingly, the non-negative constraints here allow the characterisation of a dependency of x on y since if y is zero, then so is x. If all variable coefficients are unitary then the relationships involve only one-to-one ratios. Whilst in general a scaled dependency can only relate to variables with numeric domains, dependency relationships involving only one-to-one ratios can clearly characterise inter-variable dependency relationships in non-numeric domains, in the same way as Boolean functions.

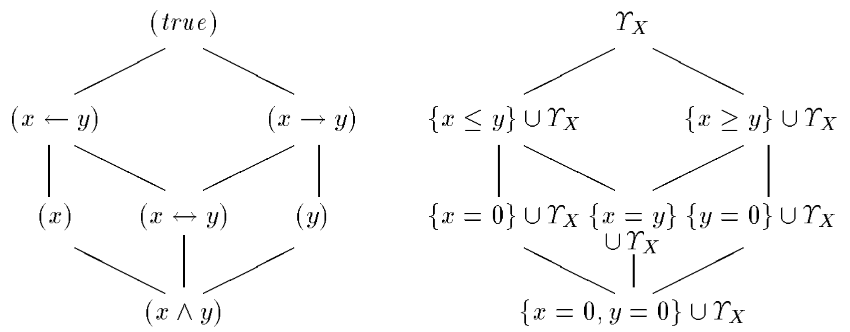
It is clear that there is a connection between those linear inequalities that can characterise inter-variable dependencies and some class of Boolean function. Consider a set of non-strict linear inequalities with unitary coefficients throughout and no constants; this set can be also be viewed as a representation of the spatial intersection of its elements. Given the constraints on the linear inequalities themselves the spaces so represented will be closed convex cones that are a particular subset of all cones and their degree of expressiveness matches exactly that of the definite Boolean functions, a sub class of positive Boolean functions.

Cones can be thought of as the union of a set of half lines emanating from the origin and with the exception of the origin, a special case, they are unbounded in at least one direction. This definition means that cones are supported, either by closed half spaces, which can be represented as non-strict inequalities, for instance $x \leq y$,; or by open half spaces which can be represented by strict inequalities, for instance $x < y$. Concern is only with closed cones as it is the non-strictness of the inequalities that allows the characterisation of dependency as illustrated in the prior example. Further, since concern is only with linear relationships, the cones will also be convex. A convex set is a set of points that delineate a space such that all linear combinations of those points are within that space and closed convex cones are known as polyhedral cones. If the inter-variable dependencies that are encoded in the delineation of these cones are to be useful, they need operators that allow their manipulation. However, if the usual mathematical operators were applied they would introduce non-unit coefficients and with them the notion of scaling and this has no meaning in a non-numeric domain. Different polyhedral cones may embody the same unscaled inter-variable dependencies. For example, $\{0 \leq x, 0 \leq y, 3x \leq 4y\}$, $\{0 \leq x, 0 \leq y, 1.2x \leq 0.01y\}$ and $\{0 \leq x, 0 \leq y, x \leq y\}$, express the same dependency with respect to the assignment of zero, namely that if y is zero then so is x. These scaled inter-variable dependencies express a level of precision relevant only in a numerical domain, but they can be abstracted to an *unscaled* form, that is, to a form with only one-to-one ratios between coefficients, by an abstraction operator that is defined solely with respect to the assignment of zero to variables. In this example, all three sets can be abstracted to the third set. These abstract cones can be manipulated with the usual operators from the more expressive domain, as any cone that is generated by their operations can be generalised by abstraction to a cone delineated by the simplest expression of variable dependency. It will be shown that abstraction collapses the infinite domain of convex cones into a finite subset of itself which, ordered by set inclusion, forms a complete lattice.

The aim of deduction in propositional logic is to deduce which variables are *true* and to capture variable dependency where it exists. Since the only deduction rule for propositional logic is *modus ponens*, a class of Boolean function that can be represented by a conjunction of definite clauses allows a representation in the form of a set of deduction rules. The aim of deduction in the domain of abstract cones is, similarly, to deduce which variables have the unique value, zero, and to capture variable dependency where it exists. The abstraction operator prescribes a representation that is unique, in the form of a complete set of deduction rules entailed by the delineation of the cone. It is not surprising then, that these abstract cones are analogous to Boolean functions that can be expressed as a conjunction of definite clauses. Functions in Pos, the set of positive Boolean functions can be represented by a conjunction of definite clauses. Def, a subset of Pos, comprises those positive functions that can be expressed without the use of disjunction and it is precisely this subset that is analagous to the abstract cones.

Both domains are complete lattices and the visual representation in Figure 1, of the lattices in the dyadic case, confirms [B.A. Davey and H.A. Priestley 1990] the analogy, as dependencies map exactly from one lattice to the other.

The remainder of this paper is in four sections. Section 2 describes the domains in question and the notation that is used to reference them. Section 3 describes the abstraction and Section 4 confirms the isomorphism. Section 5 discusses possible future work and conclusions.



(i) Def_X (ii) Abstraction of $Cone_X$ where $\Upsilon_X = \{0 \leq x, 0 \leq y\}$

Figure 1: Def_X and the abstraction of $Cone_X$ in the dyadic case.

2 Domains and Definitions

Non-strict inequalities and propositional formulae are considered over a totally ordered, finite set of variables, X, where $n = |X|$. Throughout $\vec{x}$ stands for an n-tuple denoting a point in $\mathbb{R}^n$, positive scalar multiplication of a set of points S by $\lambda > 0$, is defined $\lambda S = \{\lambda\vec{x} \mid \vec{x} \in S\}$ and $\emptyset$ denotes the empty set. Note that square brackets, [] are used to limit the scope of both universal and existential quantifiers.

Definition 2.1 A *lattice* is a partly ordered set L such that for any two elements, $l_1, l_2 \in L$ there is a meet, $l_1 \sqcap l_2$ and a join, $l_1 \sqcup l_2$, [G. Birkhoff 1948].

Theorem 2.1 Due to [G. Birkhoff 1948], the *Lattice Identities* 1 - 4 completely characterise a lattice L. For all $l_1, l_2, l_3 \in L$,

1. $l_1 \sqcap l_1 = l_1$ and $l_1 \sqcup l_1 = l_1$,
2. $l_1 \sqcap l_2 = l_2 \sqcap l_1$ and $l_1 \sqcup l_2 = l_2 \sqcup l_1$,
3. $l_1 \sqcap (l_2 \sqcap l_3) = (l_1 \sqcap l_2) \sqcap l_3$, and $l_1 \sqcup (l_2 \sqcup l_3) = (l_1 \sqcup l_2) \sqcup l_3$
4. $l_1 \sqcap (l_1 \sqcup l_2) = l_1$ and $l_1 \sqcup (l_1 \sqcap l_2) = l_1$.

A lattice L is represented as $\langle L, \sqsubseteq, \sqcup, \sqcap\rangle$ where $\sqsubseteq$ denotes the ordering, and a complete lattice as $\langle L, \sqsubseteq, \sqcup, \sqcap, \top, \bot\rangle$, where $\top$ denotes the top or greatest element of L and $\bot$ denotes the bottom or least element of L.

2.1 Polyhedra and Cones

Definition 2.2 Let $S \subseteq \mathbb{R}^n$, then the *convex hull* of S, $conv(S)$, consists of all the convex combinations of the points in S, that is, $conv(S) = \{\lambda\vec{x}_1 + (1-\lambda)\vec{x}_2 \mid \vec{x}_1, \vec{x}_2 \in S \ \wedge \ 0 \leq \lambda \leq 1\}$.

Definition 2.3 Let $S \subseteq \mathbb{R}^n$, then S is a *convex set* iff $S = conv(S)$.

Definition 2.4 The distance between any two points, $\vec{x}_1$ and $\vec{x}_2$ is denoted $d(\vec{x}_1, \vec{x}_2) = \langle \vec{x}_1 - \vec{x}_2, \vec{x}_1 - \vec{x}_2\rangle^{1/2}$ in terms of the inner product. For any point $\vec{x}_1$ in $\mathbb{R}^n$ and $\delta > 0$ the *open ball* $B(\vec{x}_1, \delta)$ with centre $\vec{x}_1$ and radius δ is $B(\vec{x}_1, \delta) \equiv \{\vec{x}_2 \in \mathbb{R}^n \colon d(\vec{x}_1, \vec{x}_2) < \delta\}$, [S. Lay 1982].

Definition 2.5 A point $\vec{x}$ is an *interior* point of the set S if there exists a $\delta > 0$ such that $B(\vec{x}, \delta) \subset S$. A set S is *open* if each of its points is an interior point of S [S. Lay 1982].

Definition 2.6 A set S is *closed* if its complement $\sim S \equiv \mathbb{R}^n/S$ is open. The *closure* of a set S is the intersection of all closed sets containing S and is denoted $cl(S)$, [S. Lay 1982].

Definition 2.7 A set of points in $\mathbb{R}^n$ which can be expressed as the intersection of finitely many closed half-spaces is called a *polyhedron.*

$Poly_X$ denotes the set of all polyhedra defined over the variable set X. Polyhedra are ordered by set inclusion.

Proposition 2.1 $\langle Poly_X, \subseteq, \overline{\cup}, \cap \rangle$ is a lattice, where $\forall\, P_1,\, P_2 \in Poly_X$
$[P_1 \,\overline{\cup}\, P_2 = cl(conv(P_1 \cup P_2))]$.
Proof
It is sufficient to show that the described lattice identities hold. Let $P_1, P_2, P_3 \in Poly_X$,

1. $P_1 \cap P_1 = P_1$, by definition of intersection. and $P_1 \,\overline{\cup}\, P_1 = P_1$, by Definition 2.2.
2. $P_1 \cap P_2 = P_2 \cap P_1$, by definition of intersection and $P_1 \,\overline{\cup}\, P_2 = P_2 \,\overline{\cup}\, P_1$, by Definition 2.2.
3. (i) $P_1 \cap (P_2 \cap P_3) = (P_1 \cap P_2) \cap P_3$, by definition of intersection.
   (ii) By [R. Rockafellar 1970, Theorem 19.6], the closure of the convex hull is associative, hence, $P_1 \,\overline{\cup}\, (P_2 \,\overline{\cup}\, P_3 = (P_1 \,\overline{\cup}\, P_2) \,\overline{\cup}\, P_3$.
4. (i) $P_1 \cap (P_1 \,\overline{\cup}\, P_2) = P_1$. Let $P' = P_1 \,\overline{\cup}\, P_2$. By Definition 2.2 $P_1 \subseteq P'$, therefore, by definition of intersection $P_1 \cap P' = P_1$,
   (ii) $P_1 \,\overline{\cup}\, (P_1 \cap P_2) = P_1$.
   Let $P'' = P_1 \cap P_2$. By definition of intersection $P'' \subseteq P_1$, therefore, by Definition 2.2 $P_1 \,\overline{\cup}\, P'' = P_1$.

■

Definition 2.8 A subset C of $\mathbb{R}^n$ is called a *cone* if it is closed under positive scalar multiplication, that is $C = \lambda C$ forall $\lambda > 0$.

Definition 2.9 A subset C of $\mathbb{R}^n$ is a *polyhedral cone* iff it can be expressed as a finite set of closed half spaces whose boundary hyperplanes pass through the origin [R. Rockafellar 1970].

The preceding definition means that all polyhedral cones are closed, convex and include the origin and the set of all polyhedral cones is denoted $Cone_X$.

Definition 2.10 If P_1, P_2 represent convex spaces in $\mathbb{R}^n$, then their *sum* is defined:

$$P_1 + P_2 = \{\vec{x_1} + \vec{x_2} \mid \vec{x_1} \in P_1, \vec{x_2} \in P_2\}$$

Proposition 2.2 $\langle Cone_X, \subseteq, \overline{\cup}, \cap \rangle$ is a lattice.
Proof
Since all polyhedral cones are in $Poly_X$, the lattice identities hold and it is sufficient to show that (i) the closure of the convex hull of an arbitrary number of polyhedral cones is a polyhedral cone, (ii) the intersection of an arbitrary number of polyhedral cones is a polyhedral cone.
Let C_1, $C_2 \in Cone_X$.

- (i) $conv(C_1 \cup C_2) = C_1 + C_2$ and $C_1 + C_2$ is a convex cone, [R. Rockafellar 1970, Theorem 3.8]. Since addition is both associative and commutative, it follows that the convex hull of an arbitrary collection of n polyhedral cones is a convex cone. Therefore the closure of the convex hull, $cl(\Sigma_{i=1}^{n} C_i)$, is a closed convex cone, that can be expressed as a finite set of closed half spaces whose boundary planes pass through the origin, that is a polyhedral cone, by Definition 2.9.

- (ii) The intersection of an arbitrary collection of convex cones is a convex cone [R. Rockafellar 1970, Theorem 2.5], and the intersection of an arbitrary collection of polyhedra is a polyhedra. Therefore since all closed convex cones are polyhedra, the intersection of an arbitrary collection of closed convex cones, is itself a polyhedral cone.

■

The *non-negative orthant* of $\mathbb{R}^n$, is $\{\langle x_1, \ldots x_n\rangle \mid 0 \leq x_1, \ldots, 0 \leq x_n\}$, [R. Rockafellar 1970], and this is a polyhedral cone.

2.2 Def_X a Class of Boolean Function

Definition 2.11 Let $Bool = \{false, true\}$, and the set of Boolean functions over a totally ordered, finite set of variables X, where $n = |X|$, be $Bool_X$. Therefore, if $f \in Bool_X$, then $f : Bool^n \rightarrow Bool$.

A propositional formula or the Boolean function it represents is denoted by f, without distinction.

Definition 2.12 $model_X : Bool_X \rightarrow \wp(\wp(X))$ and the set of models for a function f is:

$$model_X(f) = \{M \subseteq X \mid \quad y = \left\{ \begin{array}{ll} true & x_i \in M \\ false & x_i \notin M \end{array} \right\}, \quad f(\langle x_1, \ldots x_n\rangle) = true\}$$

Note that $model_X$ is bijective.

Example 2.1 Let $X = \{x, y\}$, the function $\{\langle true, true\rangle \mapsto true, \langle true, false\rangle \mapsto false, \langle false, true\rangle \mapsto false, \langle false, false\rangle \mapsto false\}$, can be represented as the formula $x \wedge y$, and $model_X(x \wedge y) = \{\{x, y\}\}$.

Since Boolean functions can be distinguished by their sets of models, ordering is defined with respect to models. Let $f_1, f_2 \in Bool_X$. If $model_X(f_1) = model_X(f_2)$, then f_1 is logically equivalent to f_2 and this is denoted $f_1 \equiv f_2$. If $model_X(f_1) \subset model_X(f_2)$, then f_1 is strictly more precise than f_2 and this is denoted $f_1 \prec f_2$. If $model_X(f_1) \subseteq model_X(f_2)$, then f_2 is a logical consequence of f_1, that is f_1 entails f_2 and this relation prescribes the ordering of Boolean functions and is denoted $\models$. If $model_X(f_1) \not\models model_X(f_2)$, and $model_X(f_2) \not\models model_X(f_1)$, then f_1 and f_2 are said to be incomparable and this is denoted $f_1 \not\parallel f_2$.

Definition 2.13 A function $f \in Bool_X$ is *positive* iff $X \in model_X(f)$.

The set of positive Boolean functions over X is denoted Pos_X.

Definition 2.14 A function $f \in Pos_X$ is *definite* iff $\forall M, M' \in model_X(f) \colon (M \cap M') \in model_X(f)$.

The set of definite, Boolean functions over X is denoted Def_X. Note that a Boolean function with models that are closed under intersection may or may not be positive.

Example 2.2 Let $X = \{x, y\}$ and $f = x \wedge \neg y$. Therefore, $model_X(f) = \{\{x\}\}$ and $model_X(f)$ is closed under intersection, but $X \notin model_X(f)$ and therefore $f \notin Pos_X$.

Definition 2.15 Let $\uparrow S = \{M' \mid M \in S \wedge M \subseteq M' \subseteq X\}$. A function $f \in Bool_X$ is said to be *monotonic* iff $\uparrow model_X(f) = model_X(f)$.

The set of monotonic Boolean functions over X is denoted Mon_X.

Example 2.3 Let $X = \{x, y\}$, then $(x \leftarrow y) \in Def_X$, since $X \in model_X(x \leftarrow y) = \{\emptyset, \{x\}, \{y\}, \{x, y\}\}$ and $model_X(x \leftarrow y)$ is closed under intersection. However, $(x \vee y) \in Pos_X$ but $(x \vee y) \notin Def_X$, as $\{x\}, \{y\} \in model_X(x \vee y)$ but $(\{x\} \cap \{y\}) = \emptyset$ and $\emptyset \notin model_X(x \vee y)$.

Both $\langle Def_X, \models, \dot{\vee}, \wedge\rangle$ and $\langle Pos_X, \models, \vee, \wedge, \rangle$ are complete lattices, where the join in Def_X is denoted by $\dot{\vee}$ [T. Armstrong *et al.* 1992]. The meet in both cases is classical conjunction, however, whilst the join in Pos_X is classical disjunction this is not the case in Def_X, see Example 2.3.

2.2.1 Representation in Def_X

Various representations of Boolean functions can be derived from the conjunctive normal form of a definite sentence, $\wedge_{i=1}^{m}(\vee_{i=1}^{n} x_{ij})$ where each x_{ij} is either a propositional variable or the negation of a propositional variable [A.G. Hamilton 1988]. Reduced Monotonic Body Form [T. Armstrong *et al.* 1992, P. Dart 1991] is such a variant where each variable occurs exactly once as a head and each body is not only monotonic, but the variable in the head does not occur in the body. A function $f \in Def_X$ iff f can be represented in Reduced Monotonic Body Form [T. Armstrong *et al.* 1992, P. Dart 1991].

Definition 2.16 A formula,

$$\bigwedge_{i=1}^{n} x_i \leftarrow M_i$$

is in *Reduced Monotonic Body Form* (RMBF), iff each $M_i \in Mon_{X \setminus \{x_i\}}$.

Orthogonal RMBF (ORMBF) is such that transitive dependencies are explicit, for example, $(x \leftarrow y) \wedge (y \leftarrow z)$ becomes $(x \leftarrow (y \vee z)) \wedge (y \leftarrow z)$.

Definition 2.17 A formula,

$$\bigwedge_{i=1}^{n} x_i \leftarrow M_i$$

in Reduced Monotonic Body Form (RMBF), is in *orthogonal* form iff for every set $Y \subseteq X$

$$(f \wedge \bigwedge Y) \models x_i) \text{ iff } (\bigwedge Y \models (M_i \vee x_i))$$

[T. Armstrong *et al.* 1992].

ORMBF is such that every deduction rule embodied in the function is explicit in the representation, but there are no pair of deduction rules such that one entails the other. DMBF is derived from ORMBF, by throwing away tautologies and reformulating disjunction in the form of distinct conjunctions such that the body of every deduction rule is itself a positive function with a set of models that is closed under intersection.

Proposition 2.3 If a formula $f \in Def_X$ then f can be represented in Definite Monotonic Body Form.
See A.2 for proof.

Definition 2.18 A function $f \in Def_X$ is described by a formula, $\bigwedge F$, in *Definite Monotonic Body Form* (DMBF) where,

$$F = \{y \leftarrow \bigwedge Y \mid (f \models y \leftarrow \bigwedge Y) \wedge \forall Y' \subset Y\, [f \not\models y \leftarrow \bigwedge Y']\}$$

and where $y \in X$ and $Y \subseteq X/\{y\}$.

3 The Abstraction of $Cone_X$

Interest here is confined to non-strict linear inequalities defining closed half planes that pass through the origin and therefore have no constants.

By constraining variables to non-negative values and restricting variable assignment only to zero, non-numeric dependencies can be characterised by considering only non-strict linear inequalities with variables that have unitary coefficients, for example $\{0 \leq x, 0 \leq y, x \leq y\}$, which captures a dependency of x on y, as if $y = 0$ then $x = 0$. The intersection of a set of such non-strict linear inequalities with unit variable coefficients throughout, spatially represents a polyhedral cone. Hereinafter, dependencies characterised by these non-strict linear inequalities with only unit variable coefficients will be referred to as *unscaled*. However, many non-strict inequalities can embody the same unscaled dependencies between variables, for example, $x \leq 3y$, and $x \leq 0.6y$ express the same dependency as $x \leq y$. By defining an abstraction operator with respect to the propagation of variable assignment to zero in a domain with variables constrained to non-negative values, this infinite domain of polyhedral cones is collapsed into a finite subset of itself. Since the abstraction maps from $Cone_X$ to $Cone_X$ it is clear that the elements can be ordered by set inclusion and will retain their relative ordering as in $Cone_X$.

3.1 The Abstraction Operator

The abstraction operator is defined with respect to the propagation of variable assignment of zero, allowing extraneous precision to be discarded. Throughout, let non-negative constraints on all variables in X be denoted $\Upsilon_X = \{0 \leq x \mid \forall x \in X\}$, and E^α denote a set of non-strict inequalities as mapped by the abstraction. Hence, the union, $\Upsilon_X \cup E^\alpha$, represents a polyhedral cone in the non-negative orthant of n-dimensional space.

Definition 3.1 Let $\Upsilon_X \cup E \in Cone_X$, $y \in X$, $Y \subseteq (X \backslash \{y\})$.
$\alpha(\Upsilon_X \cup E) = \Upsilon_X \cup E^\alpha$, where,

$$E^\alpha = \{y \leq \Sigma Y \mid \begin{array}{l} (\Upsilon_X \cup E \wedge \bigcup_{y' \in Y} y' = 0) \models y = 0 \quad \wedge \\ \forall Y' \subset Y [(\Upsilon_X \cup E \wedge \bigcup_{y' \in Y'} y' = 0) \not\models y = 0]\} \end{array}$$

Note that the structure of E^α mandates that, $(y \leq \Sigma Y \in E^\alpha) \rightarrow (\forall Y' \subset Y : y \leq \Sigma Y' \notin E^\alpha)$

Example 3.1 Let $X = \{x_1, x_2, x_3\}$ and $C = \Upsilon_X \cup \{3x_1 \leq x_2\}$ then, $\alpha(C) = \Upsilon_X \cup \{x_1 \leq x_2\}$. Clearly $(\Upsilon_X \cup \{x_1 \leq x_2\} \cup \{x_2 = 0\}) \models (x_1 = 0)$, but also, $(\Upsilon_X \cup \{x_1 \leq x_2\} \cup \{x_2 = 0, x_3 = 0\}) \models (x_1 = 0)$. and whilst $x_1 \leq x_2$ and $x_1 \leq x_2 + x_3$ are both in E', only $x_1 \leq x_2$, is in E^α.

In E^α there may be more than one non-strict inequality with the same variable on the left hand side of the inequality sign. If this is the case then all non-strict inequalities in the abstraction set, E^α that can infer zero assignment for that variable will be incomparable.

Example 3.2 Let $C = \Upsilon_X \cup \{3x \leq y, 0.5x \leq 2z\}$, then $\alpha(C) = \Upsilon_X \cup \{x \leq y, x \leq z\}$, and $\{x \leq y\} \not\parallel \{x \leq z\}$.

The abstraction operator will have the effect of relaxing the scaled relationships firstly by making all coefficients unitary and secondly by relaxing equations of the form $\Sigma Y \leq \Sigma Y'$, to $\{y \leq \Sigma Y' \mid \forall y \in Y\}$.

Example 3.3 Let $X = \{x_1, x_2, x_3, x_4\}$, and $C \in Cone_X$, where $C = \Upsilon_X \cup \{2x_1 + x_2 \leq 4x_3 + 1.5x_4\}$.

$$\alpha(C) = \Upsilon_X \cup \{x_1 \leq x_3 + x_4, x_2 \leq x_3 + x_4\}$$

The abstraction operator is such that α is a many-to-one, idempotent mapping with no inverse and it follows that $\alpha(Cone_X)$ is a finite, strict subset of $Cone_X$.

3.2 The meet and join in $\alpha(Cone_X)$

The cones in $\alpha(Cone_X)$ are also in $Cone_X$ and there are prescribed join and meet operators for $Cone_X$. Despite the fact that these operators take into account a level of precision that is not relevant in $\alpha(Cone_X)$ the intersection of cones in $\alpha(Cone_X)$ will not result in a cone that is not itself in $\alpha(Cone_X)$. The reason for this can be seen by considering the Definition 2.8 of a polyhedral cone. A polyhedral cone can be represented by the intersection of a set of closed half spaces with boundary hyperplanes that all pass through the origin. The intersection of any number of polyhedral cones is also a polyhedral cone and by definition of intersection it will be bound by some subset of all the boundary hyperplanes of the original polyhedral cones. It follows that such a cone will be in $\alpha(Cone_X)$. However, the closure of a convex hull of two or more cones in $\alpha(Cone_X)$ may not be in $\alpha(Cone_X)$, but, an unscaled dependency will be encapsulated in the ouput from this operation and can be abstracted using the abstraction operator.

Example 3.4 Let $C_1^\alpha, C_2^\alpha \in \alpha(Cone_X)$, where $X = \{x_1, x_2, x_3, x_4\}$, $C_1^\alpha = \Upsilon_X \cup \{x_1 \leq x_3, x_2 \leq x_4\}$ and $C_2^\alpha = \Upsilon_X \cup \{x_1 \leq x_4, x_2 \leq x_3\}$. Hence,

$$C_1^\alpha \,\overline{\cup}\, C_2^\alpha = \Upsilon_X \cup \{x_1 + x_2 \leq x_3 + x_4\},$$

but $\Upsilon_X \cup \{x_1+x_2 \leq x_3+x_4\} \notin \alpha(Cone_X)$. Now, $\alpha(C_1^\alpha \,\overline{\cup}\, C_2^\alpha) = \Upsilon_X \cup \{x_1 \leq x_3+x_4, x_2 \leq x_3+x_4\}$, and although $C_1^\alpha \,\overline{\cup}\, C_2^\alpha \prec \alpha(C_1^\alpha \,\overline{\cup}\, C_2^\alpha)$, if $x_3 = 0$ and $x_4 = 0$ then in both cases it can be deduced that both $x_1 = 0$ and $x_2 = 0$, and abstraction preserves the propagation of variable assignment to zero.

It should be noted that output from these operators will not introduce constants other than zero since zero is the only constant that can be present in the operands. Cones in $\alpha(Cone_X)$ are denoted C^α to distinguish them from those that are not in $\alpha(Cone_X)$.
It is precisely because $C_1^\alpha \,\overline{\cup}\, C_2^\alpha$ is not always in $\alpha(Cone_X)$ that $\alpha(Cone_X)$ is not a sublattice of $Cone_X$, [B.A. Davey and H.A. Priestley 1990]. However, the propagation facility with respect to variable assignment to zero embodied in $C_1^\alpha \,\overline{\cup}\, C_2^\alpha$ can be safely abstracted by α. Since all cones in $Cone_X$ can be reduced to their abstraction, the application of α to the output of the $\overline{\cup}$ operator on cones in $\alpha(Cone_X)$ ensures that this bound remains within $\alpha(Cone_X)$. This effectively allows the definition of both the join and meet in this more general domain allowing it to be viewed as a lattice in its own right.

Definition 3.2 Let C_1^α, $C_2^\alpha \in \alpha(Cone_X)$, then the *join in* $\alpha(Cone_X)$, is defined:

$$C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha = \alpha(C_1^\alpha \,\overline{\cup}\, C_2^\alpha)$$

Lemma 3.1 Let $C_1 = \Upsilon_X \cup E_1$, $C_2 = \Upsilon_X \cup E_2 \in Cone_X$ and $X_1 = \{y_{11}, \ldots y_{n1}\}$, $X_2 = \{y_{12}, \ldots y_{n2}\}$ so that $|X| = |X_1| = |X_2|$, then $C_1 \,\overline{\cup}\, C_2 = \{\vec{X} \mid \vec{X} = \vec{X_1} + \vec{X_2} \ \wedge\ \Upsilon_{X1} \cup E_{X_1} \ \wedge\ \Upsilon_{X2} \cup E_{X_2}\}$, where $\Upsilon_{X1} \cup E_{X_1}$, denotes the projection of the constraints on C_1 over X_1 and $\Upsilon_{X2} \cup E_{X_2}$ the projection of the constraints on C_2 over X_2.
Proof
Since polyhedral cones are closed under positive scalar multiplication, from [R. Rockafellar 1970, Theorem 19.5, Theorem 19.6,] the convex hull of two polyhedral cones reduces to their sum and it follows that $C_1 \,\overline{\cup}\, C_2 = C_1 + C_2$, that is $C_1 \,\overline{\cup}\, C_2 = \{\vec{X} \mid \vec{X} = \vec{X_1} + \vec{X_2} \wedge \Upsilon_{X1} \cup E_{X_1} \wedge \Upsilon_{X2} \cup E_{X_2}\}$.
■

Definition 3.3 Forall $C_1^\alpha = \Upsilon_X \cup E_1^\alpha$, $C_2^\alpha = \Upsilon_X \cup E_2^\alpha \in \alpha(Cone_X)$ if $\exists y_i \leq \Sigma Y_i^1$, $y_{i'} \leq \Sigma Y_{i'}^1 \in E_1^\alpha \ \wedge\ \exists y_i \leq \Sigma Y_i^2$, $y_{i'} \leq \Sigma Y_{i'}^2 \in E_2^\alpha \ \wedge\ Y_i^1 = Y_{i'}^2 \ \wedge\ Y_{i'}^1 = Y_i^2$ then y_i and $y_{i'}$ are said to be *interchangeable*.

In example 3.4, x_1 and x_2 are *interchangeable*, and when there are pairs of interchangeable variables in the representations of the operands the convex hull of the two abstract cones will not be an abstract cone itself.

The lemma that follows confirms the intuition that convexity will determine that if a variable y_i is constrained by the summation of a set of variables Y_{i1} in one cone and y_i is constrained by the summation of a set of variables Y_{i2}, in another cone then y_i will be constrained by the summation of the union $Y_{i1} \cup Y_{i2}$ in the convex hull of the two cones; and further that if a variable y_i is not constrained in both of two cones, then it will not be constrained at all in the convex hull of the two cones.

Lemma 3.2 Let $C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha = \Upsilon_X \cup E_h^\alpha$, and $C_i^\alpha = \Upsilon_X \cup E_i^\alpha$.

$$(y_i \leq \Sigma Y_{i1} \in E_1^\alpha \ \wedge\ y_i \leq \Sigma Y_{i2} \in E_2^\alpha) \ \leftrightarrow\ (y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha)$$

Proof
I. $(y_i \leq \Sigma Y_{i1} \in E_1^\alpha \ \wedge\ y_i \leq \Sigma Y_{i2} \in E_2^\alpha) \ \leftarrow\ (y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha)$.

The convex hull of two polyhedral cones reduces to their sum [R. Rockafellar 1970] as in definition 2.10, therefore if a variable is unconstrained in either or both of the operands it will be unconstrained in the convex hull.
II. $(y_i \leq \Sigma Y_{i1} \in E_1^\alpha \ \wedge \ y_i \leq \Sigma Y_{i2} \in E_2^\alpha) \ \rightarrow \ (y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha)$.
From Lemma 3.1 $C_1^\alpha \ \overline{\cup} \ C_2^\alpha = \{\vec{X} \mid \vec{X} = \vec{X_1} + \vec{X_2} \ \wedge \ \Upsilon_{X1} \cup E_{X_1} \ \wedge \ \Upsilon_{X2} \cup E_{X_2}\}$, where $X_1 = \{y_{11}, \ldots y_{n1}\}$, $X_2 = \{y_{12}, \ldots y_{n2}\}$. It will be shown that any variable constrained in both operands will be constrained in the convex hull and it is sufficient to consider all the pairwise combinations of constraints on any y_i that is constrained in both operands. There are two cases to consider, where two such variables are interchangeable and where the constrained variable is not interchangeable with another variable.
i) A variable is not interchangeable with another and is constrained in both operands.

Let $y_i \leq \Sigma Y_{i1} \in E_1^\alpha$ and $y_i \leq \Sigma Y_{i2} \in E_2^\alpha$ then by Corollary 3.1 $\forall y_i \in X[y_i = y_{i1} + y_{i2}]$, and in general, $y_i \leq \Sigma Y_{iX_1} + \Sigma Y_{iX_2}$. There are four cases to consider:

1. $Y_{i1} = Y_{i2}$, that is, $\forall y_{k1} \in Y_{iX_1}[\exists y_{k2} \in Y_{iX_2}] \ \wedge \ \forall y_{k2} \in Y_{iX_2}[\exists y_{k1} \in Y_{iX_1}]$. For example, $Y_{iX_1} = \{y_{21}, y_{31}, y_{51}\}$, $Y_{iX_2} = \{y_{22}, y_{32}, y_{52}\}$. In this case $y_i \leq \Sigma Y_{iX_1} + \Sigma Y_{iX_2}$ reduces to $y_i \leq \Sigma Y_i$ where $Y_i = Y_{i1}$, but since $Y_{i1} = Y_{i2}$, the above premise holds.

2. $Y_{i1} \subset Y_{i2}$, that is $\forall y_{k1} \in Y_{iX_1}[\exists y_{k2} \in Y_{iX_2}] \ \wedge \ \exists y_{k2} \in Y_{iX_2}[\nexists y_{k1} \in Y_{iX_1}]$. For example, $Y_{i1} = \{y_{21}, y_{31}\}$, $Y_{i2} = \{y_{22}, y_{32}, y_{52}\}$. In this case $y_i \leq \Sigma Y_{iX_1} + \Sigma Y_{iX_2}$ reduces to $y_i \leq \Sigma Y_i' + \Sigma Y_i''$ where $Y_i' = \{y_k | y_k \in Y_{i1} \ \wedge \ y_k \in Y_{i2}\}$ and therefore $Y_i' = Y_{i1} \cap Y_{i2}$, and $Y_i'' = \{y_{k2} | \exists y_{k2} \in Y_{iX_2} \wedge \ \nexists y_{k1} \in Y_{iX_1}\}$, and since $\forall y_{k2}[y_k \geq y_{k2}]$ it follows that Y_i'' can be relaxed to $\{y_k | y_k \in Y_{i2} \wedge y_k \notin Y_{i1}\}$. It is clear then that from $y_i \leq \Sigma Y_i' + \Sigma Y_i''$ it can be deduced that $y_i \leq \Sigma(Y_{i1} \cup Y_{i2})$.

3. $(Y_{i1} \cap Y_{i2} \neq \emptyset) \wedge (Y_{i1} \not\subseteq Y_{i2})$, that is, $\exists y_{k1} \in Y_{iX_1}[\exists y_{k2} \in Y_{iX_2}] \ \wedge \ \exists y_{k1} \in Y_{iX_1}[\nexists y_{k2} \in Y_{iX_2}] \wedge \exists y_{k2} \in Y_{iX_2}[\nexists y_{k1} \in Y_{iX_1}]$. For example, $Y_{iX_1} = \{y_{11}, y_{21}, y_{31}\}$, $Y_{iX_1} = \{y_{22}, y_{32}, y_{42}\}$. Following the same principle as in the previous case, $y_i \leq \Sigma Y_{iX_1} + \Sigma Y_{iX_2}$ reduces to $y_i \leq \Sigma Y_i' + \Sigma Y_i''$ where $Y_i' = \{y_k | y_k \in Y_{i1} \ \wedge \ y_k \in Y_{i2}\}$ and therefore $Y_i' = Y_{i1} \cap Y_{i2}$, and here, $Y_i'' = \{y_{k2} | y_{k2} \in Y_{iX_2} \wedge \ \nexists y_{k1} \in Y_{iX_1}\} \cup \{y_{k1} | y_{k1} \in Y_{iX_1} \wedge \ \nexists y_{k2} \in Y_{iX_2}\}$ and since $\forall y_{k2}[y_k \geq y_{k2}]$ and $\forall y_{k1}[y_k \geq y_{k1}]$ it follows that Y_i'' can be relaxed to $\{y_k | y_k \in Y_{i2} \wedge y_k \notin Y_{i1}\} \cup \{y_k | y_k \in Y_{i1} \wedge y_k \notin Y_{i2}\}$. It is clear then that from $y_i \leq \Sigma Y_i' + \Sigma Y_i''$ it can be deduced that $y_i \leq \Sigma(Y_{i1} \cup Y_{i2})$.

4. $Y_{i1} \cap Y_{i2} = \emptyset$, that is, $\forall y_{k1} \in Y_{iX_1}[\nexists y_{k2} \in Y_{iX_2}] \ \wedge \ \forall y_{k2} \in Y_{iX_2}[\nexists y_{k1} \in Y_{iX_1}]$. Here, following the same principle in part as in the previous case, since there are no variables in common, $y_i \leq \Sigma Y_{iX_1} + \Sigma Y_{iX_2}$ reduces to $y_i \leq \Sigma Y_i''$ where $Y_i'' = \{y_{k2} | y_{k2} \in Y_{iX_2} \wedge \ \nexists y_{k1} \in Y_{iX_1}\} \cup \{y_{k1} | y_{k1} \in Y_{iX_1} \wedge \ \nexists y_{k2} \in Y_{iX_2}\}$ and since $\forall y_{k2}[y_k \geq y_{k2}]$ and $\forall y_{k1}[y_k \geq y_{k1}]$ it follows that Y_i'' can be relaxed to $\{y_k | y_k \in Y_{i2} \wedge y_k \notin Y_{i1}\} \cup \{y_k | y_k \in Y_{i1} \wedge y_k \notin Y_{i2}\}$. It is clear then that from $y_i \leq \Sigma Y_i''$ it can be deduced that $y_i \leq \Sigma(Y_{i1} \cup Y_{i2})$.

By the definition of α, if $y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h$ then $y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha$.
ii) y_i and $y_{i'}$, are interchangeable variables, constrained such that $y_i \leq \Sigma Y_{i1}$, $y_{i'} \leq \Sigma Y_{i'1} \in E_1^\alpha$ and $y_i \leq \Sigma Y_{i2}$, $y_{i'} \leq \Sigma Y_{i'2} \in E_2^\alpha$, where $Y_{i1} = Y_{i'2}$ and $Y_{i'1} = Y_{i2}$. There are three cases to consider, in the first when $Y_{i1} = Y_{i'1}$, the convex hull is not more precise than its abstraction, but in the other two cases it is and abstraction generalises, allowing the premise that holds when there are no interchangeable variables to hold in all cases.

1. $Y_{i1} = Y_{i'1}$

   In this case the closure of the convex hull of the two abstract cones is equal to its abstraction. From the previous part, i), since $Y_{i1} = Y_{i'1}$, it follows that $y_i \leq \Sigma Y_{i1}$, $y_{i'} \leq \Sigma Y_{i1} \in E_h$ and this is unchanged in the abstraction, so $y_i \leq \Sigma Y_{i1}$, $y_{i'} \leq \Sigma Y_{i1} \in E_h^\alpha$.

2. $(Y_{i1} \neq Y_{i'1}) \wedge (Y_{i1} \cap Y_{i'1} \neq \emptyset)$

   Here the closure of the convex hull of the two abstract cones is more precise than its abstraction as it introduces a scaled constraint on the sum of the interchangeable variables. That is

$y_i + y_{i'} \leq \Sigma\lambda(Y_{i1} \cup Y_{i'1}) \in E_h^\alpha$, where in an abuse of notation, λ is a vector of coefficients in $\{1, 2\}$, each associated with the variables in the union of the two sets. Variables that are not common to both sets have the coefficient 1, and variables that are common to both sets, the coefficient 2. Here $\alpha(\Upsilon_X \cup E_h) = \Upsilon_X \cup E_h^\alpha$ is such that the constraint on the sum is relaxed to the individual constraints on y_i and $y_{i'}$ so that $y_i \leq \Sigma(Y_{i1} \cup Y_{i'1})$, $y_{i'} \leq \Sigma(Y_{i1} \cup Y_{i'1})$, $\in E_h^\alpha$.

3. $Y_{i1} \cap Y_{i'1} = \emptyset$:

   Once more the closure of the convex hull of the two abstract cones is more precise than its abstraction as it introduces a constraint on the sum of the interchangeable variables. In this case, similar in part to the previous one, $y_i + y_{i'} \leq \Sigma(Y_{i1} \cup Y_{i'1}) \in E_h$. The abstraction $\alpha(\Upsilon_X \cup E_h) = \Upsilon_X \cup E_h^\alpha$ is such that this relationship is relaxed and $y_i \leq \Sigma(Y_{i1} \cup Y_{i'1})$, $y_{i'} \leq \Sigma(Y_{i1} \cup Y_{i'1}) \in E_h^\alpha$.

Hence, in all cases

$$(y_i \leq \Sigma Y_{i1} \in E_1^\alpha \wedge y_i \leq \Sigma Y_{i2} \in E_2^\alpha) \rightarrow (y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha)$$

By the definition of α, if $y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h$ then $y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha$.
Therefore,

$$(y_i \leq \Sigma Y_{i1} \in E_1^\alpha \wedge y_i \leq \Sigma Y_{i2} \in E_2^\alpha) \leftrightarrow (y_i \leq \Sigma(Y_{i1} \cup Y_{i2}) \in E_h^\alpha)$$

■

Proposition 3.1 $\overline{\cup}^\alpha$ is associative, that is, $\forall\, C_1^\alpha, C_2^\alpha, C_3^\alpha \in \alpha(Cone_X)$,

$$C_1^\alpha \,\overline{\cup}^\alpha\, (C_2^\alpha \,\overline{\cup}^\alpha\, C_3^\alpha) = (C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha) \,\overline{\cup}^\alpha\, C_3^\alpha$$

Proof

1. Let $C_2^\alpha \,\overline{\cup}^\alpha\, C_3^\alpha = \Upsilon_X \cup E_{h1}^\alpha$ and $C_1^\alpha \,\overline{\cup}^\alpha\, C_{h1}^\alpha = \Upsilon_X \cup E_{h2}^\alpha$.

   $\forall y_i \in X[(y_i \leq \Sigma Y_{i2} \in E_2^\alpha \wedge y_i \leq \Sigma Y_{i3} \in E_3^\alpha) \rightarrow (y_i \leq \Sigma(Y_{i2} \cup Y_{i3}) \in E_{h1}^\alpha)]$, by Lemma 3.2.

   $\forall y_i \in X[(y_i \leq \Sigma Y_{i1} \in E_1^\alpha \wedge y_i \leq \Sigma Y_{ih1} \in E_{h1}^\alpha) \rightarrow (y_i \leq \Sigma(Y_{i1} \cup (Y_{i2} \cup Y_{i3})) \in E_{h2}^\alpha)]$, and therefore $\forall y_i \in X[(y_i \leq \Sigma Y_{ih2} \in E_{h2}^\alpha) \rightarrow (Y_{ih2} = (Y_{i1} \cup Y_{i2} \cup Y_{i3})])$. by Lemma 3.2.

2. Let $(C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha) \,\overline{\cup}^\alpha\, C_3^\alpha = \Upsilon_X \cup E_{h3}^\alpha$.

   Similarly, it can be shown that $\forall y_i \in X[(y_i \leq \Sigma Y_{ih3} \in E_{h3}^\alpha) \rightarrow (Y_{ih3} = (Y_{i1} \cup Y_{i2} \cup Y_{i3})])$.

Hence, $C_1^\alpha \,\overline{\cup}^\alpha\, (C_2^\alpha \,\overline{\cup}^\alpha\, C_3^\alpha) = (C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha) \,\overline{\cup}^\alpha\, C_3^\alpha$. ■

Proposition 3.2 $\langle \alpha(Cone_X), \subseteq, \bot, \top, \overline{\cup}^\alpha, \cap \rangle$ is a complete lattice.
Proof
(i) The lattice identities hold, and therefore $\alpha(Cone_X)$ is a lattice.
Let $C_{1^\alpha}, C_2^\alpha, C_3^\alpha \in \alpha(Cone_X)$.

1. $C_1^\alpha \cap C_1^\alpha = C_1^\alpha$, by definition of intersection, and $C_1^\alpha \,\overline{\cup}^\alpha\, C_1^\alpha = C_1^\alpha$, by Definitions 2.2 and 3.2.

2. $C_1^\alpha \cap C_2^\alpha = C_2^\alpha \cap C_1^\alpha$, by definition of intersection and $C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha = C_2^\alpha \,\overline{\cup}^\alpha\, C_1^\alpha$, by Definitions 2.2 and 3.2.

3. (i) $C_1^\alpha \cap (C_2^\alpha \cap C_3^\alpha) = (C_1^\alpha \cap C_2^\alpha) \cap C_3^\alpha$, by definition of intersection.

   (ii) $C_1^\alpha \,\overline{\cup}^\alpha\, (C_2^\alpha \,\overline{\cup}^\alpha\, C_3^\alpha) = (C_1^\alpha \,\overline{\cup}^\alpha\, C_2^\alpha) \,\overline{\cup}^\alpha\, C_3^\alpha$, since $\overline{\cup}^\alpha$ is associative by Proposition 3.1.

4. (i) $C_1^\alpha \cap (C_1^\alpha \, \overline{\cup}^\alpha \, C_2^\alpha) = C_1^\alpha$.

   Let $C^{\alpha'} = C_1^\alpha \, \overline{\cup}^\alpha \, C_2^\alpha$. By Definitions 2.2 and 3.2 $C_1^\alpha \subseteq (C_1^\alpha \, \overline{\cup} \, C_2^\alpha) \subseteq C^{\alpha'}$, therefore, by definition of intersection $C_1 \cap C^{\alpha'} = C_1^\alpha$.

   (ii) $C_1^\alpha \, \overline{\cup}^\alpha \, (C_1^\alpha \cap C_2^\alpha) = C_1^\alpha$.

   Let $C^{\alpha''} = C_1^\alpha \cap C_2^\alpha$. By definition of intersection $C^{\alpha''} \subseteq C_1^\alpha$, therefore, by Definition 2.2, $C_1^\alpha \, \overline{\cup} \, C^{\alpha''} = C_1^\alpha$, and since α is idempotent, $C_1^\alpha \, \overline{\cup}^\alpha \, C^{\alpha''} = C_1^\alpha$.

(ii) Since $\alpha(Cone_X)$ is finite, it is a complete lattice, [G. Szasz 1963], with top element: the origin, and bottom element: the non-negative orthant in n-dimensional space. ■

3.3 Representation in $\alpha(Cone_X)$

Let $C_1^\alpha, C_2^\alpha \in \alpha(Cone_X)$ and, $C_1^\alpha = \Upsilon_X \cup E_1^\alpha$ and $C_2^\alpha = \Upsilon_X \cup E_2^\alpha$. It is clear that if $E_1^\alpha = E_2^\alpha$, then C_1 and C_2 each represent the same set of points. It can be shown that the abstraction operator maps to a representation of cones in $\alpha(Cone_X)$ that is unique. In the discussion and lemmas that follow it is shown that if the representation of any two cones in $\alpha(Cone_X)$ is not equal, up to reordering of the elements of the sets of inequalities in the representations then the representations do not describe the same space.

In general, different syntactic representations may represent the same set of points, and linear combination can disclose hidden entailed dependencies. The definition of α has a significant impact on the possible outcome of linear combination in this context. Each non-strict inequality in E^α is of the form $y \leq \Sigma Y$ that describes a deduction rule for some variable with respect to the assignment of zero and by definition of α every rule is explicit including those derived from transitive dependencies. Direct dependencies and those derived from transitive dependencies are considered to be non-redundant (the formal definition follows). It can be shown that any non-redundant linear combination of a set of non-strict inequalities, $\Upsilon_X \cup E^\alpha$ that represents a cone in $\alpha(Cone_X)$, will already be explicit in the representation. This means that the representation of any cone in $\alpha(Cone_X)$ as prescribed by α, is unique up to ordering of the elements in $\Upsilon_X \cup E^\alpha$.

3.4 Linear Combination in $\alpha(Cone_X)$

Linear combination allows the explicit expression of information that is entailed in the combination of some or all of a set of linear inequalities that describe a space. All but one of the examples in this discussion are binary combinations, since addition is associative and commutative, they illustrate without loss of generality. Note that throughout the discussion that follows, the index i associated with a variable, indicates its position in the ordered set of variables over which the non-strict inequalities are considered, and the index j indicates the jth element of a set of m non-strict inequalities. Consider the motives for linear combination in the context of $\alpha(Cone_X)$:

- rather than assigning a variable to an arbitrary constant, variables are only ever assigned the value zero,
- since variables are only assigned to zero, and all variables are greater than or equal to zero, restricting a variable to a range of constants is not applicable in $\alpha(Cone_X)$,
- deduction of inter variable dependencies facilitates variable assignment to zero, since where $y \leq \Sigma Y$, if all the variables in Y are known to be zero, then y is zero. This deduction rule is analogous to *modus ponens* in Def_X.

Definition 3.4 The *linear combination* of a set of m non-strict inequalities with n variables constrained to non-negative values can be described in the following way, with λ_i, and λ_j, non-negative scalars, and at least two λ_i or λ_j are greater than zero.

$$\Sigma_{i=1}^{n} \lambda_i .0 + \Sigma_{j=n+1}^{n+m} \lambda_j y_j \;\leq\; \Sigma_{i=1}^{n} \lambda_i y_i + \Sigma_{j=n+1}^{n+m} \lambda_j (\Sigma Y_j)$$

Let l_i be a linear inequality and the symbol $+\!\!+$ describe any binary linear combination of inequalities. Hence the linear combination of k linear inequalities, $1 < k$ is $l_1 +\!\!+ \ldots +\!\!+ l_k = l'$.

Definition 3.5 A linear combination, $\Sigma Z \leq \Sigma Z'$ of the $n+m$ non-strict linear inequalities $\Upsilon_X \cup E^\alpha \in \alpha(Cone_X)$, is considered *redundant* iff,

$$[\overline{\exists}(Z \cup Z').(\Upsilon_X \cup \{\Sigma Z \leq \Sigma Z'\}) \not\models \overline{\exists}(Z \cup Z').(\Upsilon_X \cup E^\alpha)]$$

Example 3.5 Consider $X = \{x_1, x_2, x_3\}$ and $C^\alpha = \Upsilon_X \cup \{x_1 \leq x_2, x_2 \leq x_3, x_1 \leq x_3\}$, and the linear combination, $0.0 \leq 0.x_1 +\!\!+ 0.0 \leq 0.x_2 +\!\!+ 0.0 \leq 0.x_3 +\!\!+ x_1 \leq x_2 +\!\!+ 2x_2 \leq 2x_3 +\!\!+ 0.x_1 \leq 0.x_3 = x_1 + x_2 \leq 2x_3$. Since, $\exists X' \subseteq X[\overline{\exists} X'.\Upsilon_X \cup \{x_1 + x_2 \leq 2x_3\} \not\models \overline{\exists} X'.\Upsilon_X \cup E^\alpha]$, it follows that $\{x_1 + x_2 \leq 2x_3\}$ is *redundant*.

Example 3.6 Consider $X = \{x_1, x_2, x_3\}$ and $C^\alpha = \Upsilon_X \cup \{x_1 \leq x_2, x_2 \leq x_3, x_1 \leq x_3\}$, and the linear combination, $0.0 \leq 0.x_1 +\!\!+ 0.0 \leq 0.x_2 +\!\!+ 0.0 \leq 0.x_3 +\!\!+ x_1 \leq x_2 +\!\!+ x_2 \leq x_3 +\!\!+ 0.x_1 \leq 0.x_3 = x_1 + x_2 \leq x_2 + x_3 = x_1 \leq x_3$. Since, $\exists X' \subseteq X[\overline{\exists} X'.\Upsilon_X \cup \{x_1 \leq x_3\} \not\models \overline{\exists} X'.\Upsilon_X \cup E^\alpha]$, it follows that $\{x_1 \leq x_3\}$ is *non-redundant*.

Definition 3.6 Let S_d be a *system of non-cyclic transitive dependencies* of depth $d > 2$ and $p, q \in \{1, \ldots, d\}$ ordered such that,

$$S_d = \bigcup_{i=1}^{d} \{y_i \leq \Sigma Y_i \mid (y_i \notin Y_i) \wedge (\forall i : 2 \leq i \leq d\, [\exists p : 1 \leq p < i : y_i \in Y_p])\}$$

A non-redundant linear combination of such a system will be of the form $y_1 \leq \Sigma Y'$ where $Y' \subset \bigcup_{i=1}^{d} Y_i$. In the context of $\alpha(Cone_X)$, when the variables in Y' are zero, y_1 is also zero, and therefore this expresses a variable dependency, or deduction rule for y_1, and y_1 is said to be the *root* of the system.

Example 3.7 Let $\{x, y, z, p, q, r, s\} \subseteq X$, and Υ_X hold,

$$\begin{array}{rcll} x & \leq & y + z & +\!\!+ \\ y & \leq & p + q & +\!\!+ \\ p & \leq & r & +\!\!+ \\ q & \leq & s & \\ \hline x & \leq & z + r + s & \end{array}$$

In Example 3.7 linear combination of the system is a variable dependency, $x \leq z + r + s$ for the root of the system, x.

To show that the representations of cones in $\alpha(Cone_X)$ are unique, the circumstances in which a linear combination of the non-strict inequalities, that represent a cone in $\alpha(Cone_X)$, is non-redundant must be considered. Linear combination with variable elimination allows the explicit representation of entailed constraints on variables. If no variables are eliminated in the combination process then the linear combination is a relaxation of the explicit constraints on variables. Further, since the inequalities in E^α are of the form $y_j \leq \Sigma Y_j$, any linear combination of the form $\Sigma Y \leq \Sigma Y'$, where $y_j \in Y$ and $Y_j \subset Y'$ will be redundant, since the constraints on y_j will be relaxed. Therefore a non-redundant linear combination will be of the form $y \leq \Sigma Y$. Since positive scalar multiplication aids variable elimination in linear combination (see Definition 3.4 where $\lambda_i, \lambda_j \geq 0$), its effectiveness in the context of $\alpha(Cone_X)$, where scalar values are of no consequence, is also considered.

3.5 Scalar multiplication in $\alpha(Cone_X)$

Consider how scalar multiplication facilitates variable elimination.

1. A variable can be eliminated if it occurs on the same side of each inequality with coefficients of the same cardinality, but different signs.

Example 3.8

$$\begin{array}{rcl} x & \leq & 3y + 2z \;+\!\!+ \\ q & \leq & p - 2z \\ \hline x + q & \leq & 3y + p \end{array}$$

2. A variable can be eliminated if it occurs on both sides of the resulting inequality with coefficients of the same cardinality and sign. This can only occur if there is at least one instance of transitive dependency. See Example 3.7.

Generally, positive scalar multipliers are applied in linear combination to equate the cardinality of like variables. However, since all variables in any system of non-strict inequalities have positive unit coefficients, further positive scalar multiplication is ineffective, as the following lemma demonstrates. It can be shown that variable elimination can only be effected, if at each step the current coefficient of the root, y_1 is equated with that of the next linear inequality in S_d to be linearly combined. Therefore, the only possible non-redundant linear combination can, in fact, be obtained by using $\lambda_j = 1$ throughout, where $y_j \leq \Sigma Y_j \in S_d$.

Lemma 3.3 Let $\Upsilon_X \cup E^\alpha \in \alpha(Cone_X)$ and S_d be a non-cyclic system of transitive dependencies such that $S_d \subseteq E^\alpha$. The only non-redundant linear combination, $\Sigma_{i=1}^n \lambda_i.0 + \Sigma_{j=n+1}^{n+m} \lambda_j y_j \leq \Sigma_{i=1}^n \lambda_i y_i + \Sigma_{j=n+1}^{n+m} \lambda_j (\Sigma Y_j)$ where $\lambda_i = 0 \wedge (\lambda_j > 0 \leftrightarrow y_j \leq \Sigma Y_j \in S_d)$ can be derived with all non-zero $\lambda_j = 1$.

Proof

Since the linear combination of an *ordered* system of non-cyclic transitive dependencies is considered, the proof is by induction on the depth d of the system.

Let $S_d = \{y_1 \leq \Sigma Y_1, \ldots, y_d \leq \Sigma Y_d\}$, be a system of non-cyclic transitive dependencies, and therefore, $S_d \equiv \{\lambda_1 y_1 \leq \lambda_1 \Sigma Y, \ldots, \lambda_d y_d \leq \lambda_d \Sigma Y_d\}$.

- Base step: Consider the linear combination of a system of non-cyclic transitive dependencies of depth $d = 2$.

$$\begin{array}{rcl} \lambda_1 y_1 & \leq & \lambda_1 Y_1 \;+\!\!+ \\ \lambda_2 y_2 & \leq & \lambda_2 Y_2 \\ \hline \lambda_1 y_1 \; + \; \lambda_2 y_2 & \leq & \lambda_1 Y_1 + \lambda_2 Y_2 \end{array}$$

  By Definition 3.6 y_1 is the root and $y_2 \in Y_1$. Therefore the result of linear combination can be expressed as,

$$\lambda_1 y_1 + \lambda_2 y_2 \leq \lambda_1 (Y_1 / \{y_2)\} + \lambda_1 y_2 + \lambda_2 Y_2$$

  In order to facilitate the elimination of y_2 from both sides of the inequality λ_1 must equal λ_2.

- Induction step: Consider the linear combination of a system of non-cyclic transitive dependencies of depth $d > 2$. Let the result of linear combination of a system of non-cyclic transitive dependencies of depth $d-1$ be $R = \lambda_1 y_1 \leq \lambda_1(\Sigma Y_1') + \ldots + \lambda_{d-1}(\Sigma Y_{d-1}')$, where Y_i' is the residue of Y_i after variable elimination. By the induction hypothesis,

$$R \equiv \lambda_1 y_1 \leq \lambda_1(\Sigma(Y_1' \cup \ldots Y_{d-1}')).$$

  Now consider a system of depth d and its linear combination, the result of linear combination of $d-1$ elements combined with the dth element in the system.

$$\begin{array}{rcl} \lambda_1 y_1 & \leq & \lambda_1(\Sigma(Y_1' \cup \ldots \cup Y_{d-1}')) \;+\!\!+ \\ \lambda_d y_d & \leq & \lambda_d Y_d \\ \hline \lambda_1 y_1 \; + \; \lambda_d y_d & \leq & \lambda_1(\Sigma(Y_1' \cup \ldots \cup Y_{d-1}')) + \lambda_d Y_d \end{array}$$

  By definition, $y_d \in (Y_1' \cup \ldots \cup (Y_{d-1}')$, therefore the result can be expressed as,

$$\lambda_1 y_1 + \lambda_d y_d \leq \lambda_1(\Sigma(Y_1' \cup \ldots \cup Y_{d-1}')/\{y_d\}) + \lambda_1 y_d + \lambda_d Y_d$$

  As before, in order to facilitate elimination, here of y_d, λ_1 must equal λ_d.

Therefore, by the principle of mathematical induction, in order to facilitate variable elimination in a linear combination of a system of non-cyclic transitive dependencies, the scalar multipliers employed at each combination step must be the same. Since variables throughout the system have unit coefficients to begin with, the only linear combination of a system of non-cyclic transitive dependencies that is non-redundant, can be derived with multipliers equal to one. ∎

It can be shown that linear combination of the representation of a cone in $\alpha(Cone_X)$, cannot yield a non-redundant result that is not already explicit in its representation. The proof is by contradiction.

Lemma 3.4 Let $\Upsilon_X \cup E^\alpha$ represent a polyhedral cone, in $\alpha(Cone_X)$, then $(\Upsilon_X \cup E^\alpha \models (y \leq \Sigma Y)) \leftrightarrow \exists (y \leq \Sigma Y) \in E^\alpha$ where $y \leq \Sigma Y$ is non-redundant.
Proof

1. $\Upsilon_X \cup E^\alpha \models (y \leq \Sigma Y) \leftarrow \exists (y \leq \Sigma Y) \in E^\alpha$ Elementary.
2. $\Upsilon_X \cup E^\alpha \models (y \leq \Sigma Y) \rightarrow \exists (y \leq \Sigma Y) \in E^\alpha$

   Let $\Upsilon_X \cup E^\alpha$ represent a cone in $\alpha(Cone_X)$, suppose that $(\Upsilon_X \cup E^\alpha) \models (y \leq \Sigma Y)$, where $y \leq \Sigma Y$ is non-redundant and assume that $\nexists (y \leq \Sigma Y) \in E^\alpha$. This assumption asserts that there is some linear combination of elements in $\Upsilon_X \cup E^\alpha$ that is equal to $y \leq \Sigma Y$. Consider possible linear combinations of the of the elements in $\Upsilon_X \cup E^\alpha$:

   (a) In all cases, linear combination that includes non-negative constraints will be redundant.

   $$\text{(i)}\quad \begin{array}{rcl} 0 & \leq & y_1 \quad ++ \\ y_1 & \leq & \Sigma Y_1 \\ \hline 0 & \leq & \Sigma Y_1 \end{array} \qquad \text{(ii)}\quad \begin{array}{rcl} 0 & \leq & y_1 \quad ++ \\ y_2 & \leq & \Sigma Y_2 \\ \hline y_2 & \leq & y_1 + \Sigma Y_2 \end{array}$$

   (b) *Without variable elimination*

   The non-strict inequalities are of the form $y \leq \Sigma Y$ and hence, any linear combination of non-strict inequalities, without variable elimination, will be of the form $\Sigma Y \leq \Sigma Y'$ and this will relax the upper bound on each of the variables in Y. Hence, in these circumstances, linear combination will be redundant.

   (c) *With variable elimination*

   i. A variable can be eliminated if it occurs on the same side of each inequality with coefficients of the same cardinality, but different signs. This method is inapplicable, since in any linear combination,

   $$\Sigma_{i=1}^{n} \lambda_i.0 + \Sigma_{j=n+1}^{n+m} \lambda_j y_j \;\leq\; \Sigma_{i=1}^{n} \lambda_i y_i + \Sigma_{j=n+1}^{m+n} \lambda_j (\Sigma Y_j)$$

   there are no variables with negative coefficients on the right hand side of the inequality. Re-arrangement by subtraction of a variable on the left hand side of the inequality sign will render the linear combination redundant.

   ii. Variable elimination can occur when facilitated by a system of non-cyclic transitive dependencies, however, by Lemma 3.3, any linear combination of a system of non-cyclic transitive dependencies that is non-redundant can be derived with unit multipliers (λ_js).

   Hence, the only non-redundant linear combinations are derived through transitive dependencies, and all transitive dependencies are explicit in E^α, by definition of α. Therefore, it follows that where $\Upsilon_X \cup E^\alpha \models (y \leq \Sigma Y)$, and $y \leq \Sigma Y$ is non-redundant, the assumption $\nexists (y \leq \Sigma Y) \in E^\alpha$, is false, since no linear combination of non-strict inequalities can derive a non-redundant non-strict inequality that is not already explicit in E^α. That is, where $y \leq \Sigma Y$ is non-redundant, $\Upsilon_X \cup E^\alpha \models y \leq \Sigma Y \rightarrow \exists (y \leq \Sigma Y) \in E^\alpha$.

Hence,

$$\Upsilon_X \cup E^\alpha \models (y \leq \Sigma Y) \leftrightarrow \exists (y \leq \Sigma Y) \in E^\alpha$$

■

Corollary 3.1 By relaxing the non-redundant condition on $y \leq \Sigma Y$, Lemma 3.4 generalises to, $(\Upsilon_X \cup E \models y \leq \Sigma Y) \leftrightarrow \exists (y \leq \Sigma Y') \in E^\alpha : (Y' \subseteq Y)$ ■

To recap, deduction in $\alpha(Cone_X)$ is driven by these aims, to assign variables to zero or to deduce dependency relationships, in the form $y \leq \Sigma Y$, for as many variables as possible and thereby facilitate their assignment to zero. Since no further non-redundant non-strict inequality can be derived from linear combination of the prescribed representations, it follows that the representations embody a set of deduction rules that are entailed by the delineation of the abstract cones. Further, it is clear that in this context, the only deduction rule is of the form $y \leq \Sigma Y$, and the prescribed representation of a cone in $\alpha(Cone_X)$ as the union of non-negative variable constraints with a set of deduction rules is unique.

4 A Lattice Isomorphism

There is clearly a connection between the variable dependencies that are encoded in the representation of abstract cones in $\alpha(Cone_X)$ and Boolean functions in Def_X. Abstract cones are uniquely defined as sets of points and functions in Def_X as sets of models. In both cases set intersection is the meet operator but generalisation of set union is required to achieve a join operator that is closed. The significant characteristic of these abstract cones that indicates the necessity for generalisation is their convexity. This characteristic disallows a straightforward join that comprises the union of those points that are in either or both operands, as the resulting set must conform to the constraint that any linear combination of points in the join must itself be in the join. There are instances when the union of two particular polyhedra cannot be represented in terms of a single polyhedra, that is, the union can only be represented as two distinct, albeit possibly abutting, polyhedra. Hence, the union cannot be uniformly represented in terms of a single polyhedra without generalisation and this is precisely what the closure of the convex hull is, a generalisation that is the smallest convex space that contains all the points in both operands. Similarly, there are instances when the join of two functions in Def_X that in terms of models is the union of the sets of models for each function, is such that the union of models represents a function that is not itself in Def_X, since its models are not closed under intersection. The generalisation required is analogous to that required in $\alpha(Cone_X)$.
Representations in $\alpha(Cone_X)$ are an explicit set of non-strict inequalities that prescribe the means of deducing that a variable on the left hand side of each inequality is zero. There may be more than one such rule, for each variable, or there may be none for a particular variable. This condition is consistent with that of a DMBF representation in Def_X. DMBF is such that where $f \in Def_X$ and $f \equiv \bigwedge F$, each implication in F is a deduction rule for the variable that occurs in the head. Every possible rule is explicit, but no rule is entailed by any other, and if nothing is known about a particular variable it will not occur in the head of any implication. Deduction and propagation follow one from the other and Dart [P. Dart 1991] considers that such implications can be thought of, alternatively as propagation rules. It is clear that however these rules are viewed, the inequalities in the representation of elements in the abstraction of $Cone_X$ serve the same purpose as the implications in DMBF representations of functions in Def_X.

4.1 From Polyhedral Cones to Boolean Functions

The propagation of the assignment of zero to a variable in the non-negative orthants of n-dimensional space in $\alpha(Cone_X)$ has an analogy to the the propagation of the assignment of *true* to a variable in Def_X. Since elements from both domains can be expressed as sets of deduction rules the mapping is described in these terms. φ'_X maps a cone $C^\alpha \in \alpha(Cone_X)$ to a set of implications, definite clauses, the conjunction of which represents a formula in Def_X.

Definition 4.1 The mapping $\varphi'_X: \alpha(Cone_X) \to Bool_X$, is defined:

$$\varphi'_X\left(\Upsilon_X \cup \bigcup_{j=1}^m \{y_j \leq \Sigma Y_j\}\right) = \bigwedge_{j=1}^m y_j \leftarrow \bigwedge Y_j$$

Example 4.1 Let $X = \{x_1, x_2, x_3\}$,

$$\varphi'_X(\Upsilon_X \cup \{x_1 \leq x_2,\ x_3 \leq 0\}) = (x_1 \leftarrow x_2) \wedge (x_3 \leftarrow true)$$

Note that for any variable $y \in X$, when $y \leq 0 \in E^\alpha$ this is equivalent to $y \leq \Sigma Y$, where $Y = \emptyset$ and φ'_X maps $y \leq \Sigma\emptyset$ to the implication $y \leftarrow true$. Since all variables are constrained to non-negative values, $(y \leq 0 \wedge 0 \leq y) \equiv (y = 0)$, giving a mapping from $y = 0$ to $y \leftarrow true$ as expected.

The implications in $\varphi'_X(\alpha(Cone_X))$ are clearly representative of Boolean functions and it can be shown that the constraints imposed on the representations of cones in $\alpha(Cone_X)$ by α are such that when mapped to $Bool_X$ the definite clauses in the image conform to DMBF and a Boolean function can be represented in DMBF iff it is in Def_X.

Proposition 4.1 $\varphi'_X(\alpha(Cone_X)) = Def_X$.
Proof

- $\varphi'_X(\alpha(Cone_X)) \subseteq Def_X$
  A cone in $\alpha(Cone_X)$ can be represented by $\Upsilon_X \cup E^\alpha$. By definition of α, E^α is in orthogonal form, that is, transitive dependencies are explicit and $y \leq \Sigma Y \in E^\alpha \to (\forall Y' \subset Y : y \leq \Sigma Y' \notin E^\alpha)$. Therefore, by definition of φ'_X, F is also in orthogonal form and $y \leftarrow \bigwedge Y \in F \to (\forall Y' \subset Y : y \leftarrow \bigwedge Y' \notin F)$. Since these are precisely the conditions that describe DMBF it follows that $\varphi'_X(C^\alpha) \subseteq Def_X$.

- $Def_X \subseteq \varphi'_X(\alpha(Cone_X))$
  Similarly, a formula $f \in Def_X$ can be represented in DMBF and $f = \bigwedge F$. By definition of DMBF, F is in orthogonal form and $y \leftarrow \bigwedge Y \in F \to (\forall Y' \subset Y : \bigwedge F \not\models y \leftarrow \bigwedge Y')$. Since the elements of F are definite clauses it follows that $y \leftarrow \bigwedge Y \in F \to (\forall Y' \subset Y : y \leftarrow \bigwedge Y' \notin F)$. By definition of α and φ'_X these conditions also apply to every element of $\varphi'_X(C^\alpha)$. Therefore, $Def_X \subseteq \varphi_X(\alpha(Cone_X))$.

■

Hence, φ_X is defined:

Definition 4.2 $\varphi_X: \alpha(Cone_X) \to Def_X$.

$$\varphi_X(\Upsilon_X \cup E^\alpha) = \varphi'_X(\Upsilon_X \cup E^\alpha)$$

Proposition 4.2 φ_X is injective.
Proof
Let $\varphi_X(C_1^\alpha) = \bigwedge F_1$, $\varphi_X(C_2^\alpha) = \bigwedge F_2$, where $C_1^\alpha = \Upsilon_X \cup E_1^\alpha$, $C_2^\alpha = \Upsilon_X \cup E_2^\alpha \in \alpha(Cone_X)$. If $\varphi_X(C_1^\alpha) \equiv \varphi_X(C_2^\alpha)$, then $F_1 = F_2$, since, by definition of DMBF, $(\varphi_X(C_1^\alpha) \equiv \varphi_X(C_2^\alpha)) \leftrightarrow (F_1 = F_2)$. Therefore, $E_1^\alpha = E_2^\alpha$, by definition of φ. It then follows that $C_1^\alpha \equiv C_2^\alpha$, since by definition of α and Lemma3.4 $(C_1^\alpha \equiv C_2^\alpha) \leftrightarrow (E_1^\alpha = E_2^\alpha)$.
Hence, φ_X is injective since,

$$(\varphi(C_1^\alpha) \equiv \varphi(C_2^\alpha)) \to (C_1^\alpha \equiv C_2^\alpha).$$

■

Proposition 4.3 φ_X is bijective.
Proof
Since by Proposition 4.2 φ_X is injective and by Definition 4.2 $\varphi_X(\alpha(Cone_X)) = Def_X$ it follows that φ_X is bijective. ■

Proposition 4.4 Let C_1^α, $C_2^\alpha \in \alpha(Cone_X)$, then $(C_1^\alpha \prec C_2^\alpha) \leftrightarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha))$.
Proof
Let $C_1^\alpha = \Upsilon_X \cup E_1^\alpha$, $C_2^\alpha = \Upsilon_X \cup E_2^\alpha$, $\varphi_X(C_1^\alpha) = \bigwedge F_1$ and $\varphi_X(C_2^\alpha) = \bigwedge F_2$. Note that by definition, φ_X maps cones to representations of Boolean functions in Def_X that are in DMBF.

- $(C_1^\alpha \prec C_2^\alpha) \rightarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha)$
  Let $C_1^\alpha \prec C_2^\alpha$, consider E_1^α and E_2^α.
  * If $E_2^\alpha = \emptyset$, then $\Upsilon_X \cup E_2^\alpha = \Upsilon_X$ and $\forall(y_1 \leq \Sigma Y_1) \in E_1^\alpha[\Upsilon_X \cup (y_1 \leq \Sigma Y_1) \models C_2^\alpha]$.
  * Otherwise by Corollary 3.1 the following conditions both hold:
    $\forall(y_2 \leq \Sigma Y_2) \in E_2^\alpha[\exists(y_1 \leq \Sigma Y_1) \in E_1^\alpha : (y_1 = y_2) \wedge (Y_1 \subseteq Y_2)] \wedge$
    $\exists(y_1 \leq \Sigma Y_1) \in E_1^\alpha[\forall(y_2 \leq \Sigma Y_2) \in E_2^\alpha : (y_1 \neq y_2) \vee ((y_1 = y_2) \wedge (Y_1 \subseteq Y_2))]$
  Consider $\varphi_X(C_1^\alpha)$ and $\varphi_X(C_2^\alpha)$. Given the definition of φ_X, where $E_2^\alpha \neq \emptyset$, the following conditions both hold:
  1. $\forall(y_2 \leftarrow \bigwedge Y_2) \in F_2[\exists(y_1 \leftarrow \bigwedge Y_1) \in F_1 : (y_1 = y_2) \wedge (Y_1 \subseteq Y_2)] \wedge$
  2. $\exists(y_1 \leftarrow \bigwedge Y_1) \in F_1[\forall(y_2 \leftarrow \bigwedge Y_2) \in F_2 : (y_1 \neq y_2) \vee ((y_1 = y_2) \wedge (Y_1 \subset Y_2))]$

  Hence, by 2., $\exists(y_1 \leftarrow \bigwedge Y_1) \in F_1 : (\bigwedge F_2 \not\models (y_1 \leftarrow \bigwedge Y_1))$, therefore by definition of DMBF, $\varphi_X(C_2^\alpha) \not\models \varphi_X(C_1^\alpha)$. Since, by 1, $\varphi_X(C_1^\alpha) \models \varphi_X(C_2^\alpha)$, it follows that, $\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha)$, and therefore, $(C_1^\alpha \prec C_2^\alpha) \rightarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha))$.

- $(C_1^\alpha \prec C_2^\alpha) \leftarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha))$
  Let $\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha)$, now consider F_1 and F_2.
  * If $F_2 = \emptyset$, then $\bigwedge F_2 = true$ and $\forall(y_1 \leftarrow \bigwedge Y_1) \in F_1[(y_1 \leftarrow \bigwedge Y_1) \models \varphi_X(C_2^\alpha)]$.
  * Otherwise from the definition of DMBF, the following conditions both hold:
    $\forall(y_2 \leftarrow \bigwedge Y_2) \in F_2[\exists(y_1 \leftarrow \bigwedge Y_1) \in F_1 : (y_1 = y_2) \wedge (Y_1 \subseteq Y_2)] \wedge$
    $\exists(y_1 \leftarrow \bigwedge Y_1) \in F_1[\forall(y_2 \leftarrow \bigwedge Y_2) \in F_2 : (y_1 \neq y_2) \vee ((y_1 = y_2) \wedge (Y_1 \subset Y_2))]$
  Consider C_1^α and C_2^α. Given the definition of φ_X, where $E_2^\alpha \neq \emptyset$, the following conditions both hold,
  1. $\forall(y_2 \leq \Sigma Y_2) \in E_2^\alpha[\exists(y_1 \leq \Sigma Y_1) \in E_1^\alpha : (y_1 = y_2) \wedge (Y_1 \subseteq Y_2)] \wedge$
  2. $\exists(y_1 \leq \Sigma Y_1) \in E_1^\alpha[\forall(y_2 \leq \Sigma Y_2) \in E_2^\alpha : (y_1 \neq y_2) \vee ((y_1 = y_2) \wedge (Y_1 \subset Y_2))]$

  Hence, by 2. and Lemma 3.4, $\exists(y_1 \leq \Sigma Y_1) \in E_1^\alpha : (\Upsilon_X \cup E_2^\alpha) \not\models (\Upsilon_X \cup (y_1 \leq \Sigma Y_1))$, and $C_2^\alpha \not\models C_1^\alpha$. Since, by 1. and Corollary 3.1, $C_1^\alpha \models C_2^\alpha$, it follows that, $C_1^\alpha \prec C_2^\alpha$, and therefore, $C_1^\alpha \prec C_2^\alpha) \leftarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha))$.

Hence,
$$(C_1^\alpha \prec C_2^\alpha) \leftrightarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha))$$
■

Proposition 4.5 There is a lattice isomorphism between Def_X and $\alpha(Cone_X)$.
Proof
By Proposition 4.3, φ_X is bijective, and by Proposition 4.4, for all C_1^α, $C_2^\alpha \in \alpha(Cone_X)$, $(C_1^\alpha \prec C_2^\alpha) \leftrightarrow (\varphi_X(C_1^\alpha) \prec \varphi_X(C_2^\alpha))$. Therefore, it follows that there is a lattice isomorphism between $\alpha(Cone_X)$ and Def_X, the image set under φ_X [B.A. Davey and H.A. Priestley 1990]. ■

5 Conclusion

Def_X lacks the precision of Pos_X, since information is lost with the join operator and since Def_X is not condensing it is not suited to goal-independent analyses [T. Armstrong *et al.* 1992]. However, despite these apparent drawbacks, in practice, the performance of Def_X for goal-dependent analyses can compare favourably with that of Pos_X, particularly if the implementation is efficient [A. King, P. Hill and J.Smaus 1998].

These factors extend to $\alpha(Cone_X)$, and its usefulness in an environment where the solvers are already in place, for example in the context of constraint logic programming will depend on the efficiency of the solvers, particularly with regard to the join calculation.

Close inspection has revealed a natural affinity between the two-point domain Def_X, where variables are assumed *false* until proven to be *true* and its two-point counterpart, $\alpha(Cone_X)$, where variables are assumed to be non-negative until proven to be zero. Definite Monotonic Body Form offers a unique representation, as a set of definite clauses, for any function in Def_X, that is an explicit expression of all the deduction rules that are entailed by the function. Similarly, the abstraction operator prescribes a unique representation as a set of non-strict linear inequalities, for an abstract cone in $\alpha(Cone_X)$ that is an explicit expression of all the deduction rules that are entailed by the delineation of the abstract cone.

References

[A. King, P. Hill and J.Smaus 1998] A. King, P. Hill and J.Smaus (1998) Practical Dependency Analysis through a *Share* Quotient. Technical Report 587, Computing Laboratory, University of Kent, UK.

[G. Birkhoff 1948] G. Birkhoff (1948) Lattice Theory. *American Mathematical Society*, 190 Hope Street, Providence, Rhode Island, USA.

[G. Szasz 1963] G. Szasz (1963) Introduction to Lattice Theory. Academic Press, New York and London and The Publishing House of the Hungarian Academy of Sciences, Budapest.

[P. Dart 1991] P. Dart (1991) On Derived Dependencies and Connected Databases. *Journal of Logic Programming.*

[B.A. Davey and H.A. Priestley 1990] B.A. Davey and H.A. Priestley (1990) Introduction to Lattices and Order. Cambridge University Press.

[A.G. Hamilton 1988] A.G. Hamilton (1988) Logic for Mathematicians. Cambridge University Press.

[R. Rockafellar 1970] R. Rockafellar (1970) Convex Analysis. Princeton University Press, New Jersey.

[S. Lay 1982] S. Lay (1982) Convex Sets and their Applications. Wiley and Sons.

[T. Armstrong *et al.* 1992] T. Armstrong and K. Marriot and P. Schachte and H. Sondergaard (1992) Two Classes of Boolean Functions for Dependency Analysis.

[P. Cousot and R. Cousot 1992] P. Cousot and R. Cousot (1992) Abstract Interpretation and Application to Logic Programs. Laboratoire d'Informatique de l'Ecole Normale Superiéure 45 Rue d'Ulm, 75230 Paris Cédex 05, France.

A PROOFS

Proposition A.1 Let $f \equiv \bigwedge MB$ and

$$MB = \{x_i \leftarrow M_i \mid \forall Y \subseteq X\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i)]\},$$

where $M_i \in Mon_X/\{x_i\}$. It can be shown that

$$\begin{array}{c} \forall Y \subseteq X\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i))] \\ \leftrightarrow \\ \forall Y' \subseteq X/\{x_i\}\,[((f \wedge \bigwedge Y') \models x_i) \leftrightarrow (\bigwedge Y' \models M_i)] \end{array}$$

It follows then that MB can be also be defined:

$$MB = \{x_i \leftarrow M_i \mid \forall Y \subseteq X/\{x_i\}\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models M_i)]\}$$

Proof

1. To show that

$$\begin{array}{c} \forall Y \subseteq X\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i))] \\ \rightarrow \\ \forall Y' \subseteq X/\{x_i\}\,[((f \wedge \bigwedge Y') \models x_i) \leftrightarrow (\bigwedge Y' \models M_i)] \end{array}$$

   Since $((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i))$, then $(f \wedge \bigwedge Y) \models x_i) \leftrightarrow ((\bigwedge Y \models M_i) \vee (\bigwedge Y \models x_i))$. Consider then $Y' \subseteq X/\{x_i\}$, since $(\bigwedge Y \models x_i) \leftrightarrow (x_i \in Y)$ it follows that $((f \wedge \bigwedge Y') \models x_i) \leftrightarrow (\bigwedge Y' \models M_i)$.

2. To show that

$$\begin{array}{c} \forall Y \subseteq X\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i))] \\ \leftarrow \\ \forall Y' \subseteq X/\{x_i\}\,[((f \wedge \bigwedge Y') \models x_i) \leftrightarrow (\bigwedge Y' \models M_i)] \end{array}$$

   Consider $Y = Y' \cup \{x_i\}$, then if $((f \wedge \bigwedge Y') \models x_i) \leftrightarrow (\bigwedge Y' \models M_i))$, it follows that $((f \wedge \bigwedge Y) \models x_i) \leftrightarrow true$. Similarly, since $(\bigwedge Y' \models M_i)$, it follows that $(\bigwedge Y \models (M_i \vee x_i)) \leftrightarrow true$. Therefore, $((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i))$.

Hence

$$(((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (M_i \vee x_i))) \leftrightarrow (((f \wedge \bigwedge Y') \models x_i) \leftrightarrow (\bigwedge Y' \models M_i)))$$

Therefore MB can also be defined:

$$MB = \{x_i \leftarrow M_i \mid \forall Y \subseteq X/\{x_i\}\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models M_i)]\}$$

■

Proposition A.2 If a function, $f \in Def_X$, then f can be represented in Definite Monotonic Body Form.
Proof
A function $f \in Def_X$ iff f can be represented in ORMBF [T. Armstrong *et al.* 1992]. Let $f = \bigwedge MB$ be in ORMBF where, by Proposition A.1,

$$MB = \{x_i \leftarrow M_i \mid \forall Y \subseteq X/\{x_i\}\,[((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models M_i)]\}$$

and $M_i \in Mon_{X/\{x_i\}}$.
By relaxing the reduced condition, that each variable occurs at least and only once as a head, a set of implications F can be derived from those in MB such that $\bigwedge F \equiv \bigwedge_{i=1}^{n} MB$. Therefore, the aim is (i) to derive a set of implications, F, such that the body of each implication in F is a conjunction of propositional variables and (ii) to identify the constraints that qualify those implications.
(i) The proof is by induction on the depth, k, of the formulae that compose M_i.

- Base Step: Consider $x_i \leftarrow M_i$ where $k = 0$, that is, there are no connectives.
  - $(x_i \leftarrow x_j)$ where x_j is a statement variable and $i \neq j$, is in the required form.
  - $(x_i \leftarrow true) \equiv (x_i \leftarrow \bigwedge \emptyset)$ [P. Dart 1991].
  - $(x_i \leftarrow false)$ is a tautology and can be discarded.
- Induction Step: Now consider $x_i \leftarrow M_i$, where $k > 0$, and by the induction hypothesis, let every formula M_i of depth $k-1$ or less be expressible as a conjunction of variables in the set X_i, such that $x_i \notin X_i$. Since M_i can only be constructed from Mon_X there are two cases to consider, $M_i = f_1 \wedge f_2$ and $M_i = f_1 \vee f_2$:
  - Case 1: $x_i \leftarrow (f_1 \wedge f_2)$. By the induction hypothesis $f_1 = \bigwedge X_i^1$ and $f_2 = \bigwedge X_i^2$. Therefore,
    $$x_i \leftarrow \bigwedge(X_i^1 \cup X_i^2)$$
    and is in the required form.
  - Case 2: $x_i \leftarrow (f_1 \vee f_2)$. By the induction hypothesis $f_1 = \bigwedge X_i^1$ and $f_2 = \bigwedge X_i^2$. Therefore, since
    $$x_i \leftarrow (\bigwedge X_i^1 \vee \bigwedge X_i^2) \equiv (x_i \leftarrow \bigwedge X_i^1) \wedge (x_i \leftarrow \bigwedge X_i^2),$$
    it follows that $x_i \leftarrow (f_1 \vee f_2)$ can be replaced by the conjunction of the two implications on the right hand side of the equivalence, that are both in the required form.

(ii) Consider the four possible forms that M_i can take. If $x_i \leftarrow false$ is in MB it is discarded, otherwise each implication with a single variable in the head is replaced by a set of implications each with that same single variable in the head. Let this set be F_i where the index i associates each set with a particular variable. The conditions that apply to each element of MB, where $M_i \neq false$, map across to the conjunction of a set of implications F_i, that is logically equivalent to its counterpart in MB. The mapping is simple in cases 2 and 3, where F_i has only one element, but case 4 is not so straightforward.

1. $x_i \leftarrow false$ is a tautology and is discarded.
2. $x_i \leftarrow true$ is replaced with $F_i = \{x_i \leftarrow \bigwedge \emptyset\}$.
3. $x_i \leftarrow \bigwedge X_i$, where $X_i \subseteq X/\{x_i\}$, is replaced by $F_i = \{x_i \leftarrow \bigwedge X_i\}$.
4. $x_i \leftarrow \bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu}$, where $X_{i\mu} \subseteq X/\{x_i\}$, is replaced by $F_i = \bigcup_{\mu=1}^{\eta}\{x_i \leftarrow \bigwedge X_{i\mu}\}$.

In case 4, the conditions that apply to a single implication map across to a conjoined set of implications. Consider the conditions that qualify each element in MB :

$$MB = \{x_i \leftarrow M_i \mid \forall Y \subseteq X/\{x_i\} [((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models M_i)]\}$$

and the case where $x_i \leftarrow M_i \in MB$ and $M_i = \bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu}$. In this case then:

$$((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\bigwedge Y \models (\bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu}))$$

However, $(\bigwedge Y \models (\bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu})) \leftrightarrow (\exists \mu' \in \{1 \ldots \eta\} [\bigwedge Y \models \bigwedge X_{i\mu'}])$ and $(\bigwedge Y \models \bigwedge X_{i\mu'}) \leftrightarrow (X_{i\mu'} \subseteq Y)$. Therefore,

$$(\bigwedge Y \models (\bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu})) \leftrightarrow (\exists \mu' \in \{1 \ldots \eta\} [X_{i\mu'} \subseteq Y])$$

It follows then that where $x_i \leftarrow \bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu} \in MB$,

$$((f \wedge \bigwedge Y) \models x_i) \leftrightarrow (\exists \mu' \in \{1 \ldots \eta\} [X_{i\mu'} \subseteq Y])$$

Hence, since $\bigwedge F_i \equiv (x_i \leftarrow \bigvee_{\mu=1}^{\eta} \bigwedge X_{i\mu}$, F_i can be defined:

$$F_i = \bigcup_{\mu=1}^{\eta} \{x_i \leftarrow X_{i\mu} \,|\, \forall Y \subseteq X/\{x_i\} \,[((f \wedge \bigwedge Y \models x_i) \leftrightarrow (\exists \mu' \in \{1 \ldots \eta\} \,[X_{i\mu'} \subseteq Y])]\}$$

It follows then that:

$$(y_i \leftarrow \bigwedge Y) \in F_i \;\rightarrow\; (\forall Y' \subset Y \,[f \not\models y_i \leftarrow \bigwedge Y'])$$

Let $\bigcup_{i \in \{1 \ldots n\}} F_i = F$ and then $\bigwedge F \equiv \bigwedge MB$. Since no single implication in F can entail another with a different single variable in the head, overall $(y \leftarrow \bigwedge Y) \in F \;\rightarrow\; (\forall Y' \subset Y \,[f \not\models y \leftarrow \bigwedge Y'])$, which allows this definition of DMBF as $\bigwedge F$ where:

$$F = \{y \leftarrow \bigwedge Y \,|\, (f \models y \leftarrow \bigwedge Y) \;\wedge\; \forall Y' \subset Y \,[f \not\models y \leftarrow \bigwedge Y'])\}$$

■