Set Intersection and Consistency in Constraint Networks

YUANLIN ZHANG
University College Cork
and
ROLAND H. C. YAP
National University of Singapore

Constraint networks provide an effective way to model and solve problems occuring in artificial intelligence, scheduling, and many other fields. The concept of consistency developed for constraint networks has shed new lights on identifying problems that can be efficiently solved. In this paper, we propose to express k-consistency, where k is a number and larger k implies higher consistency in a network, in terms of set intersection. A proof schema is then presented as a generic way to obtain consistency properties in a constraint network by lifting properties on set intersection. This approach not only eases the understanding of and unifies many existing consistency results, but also provides a theoretical tool for deriving new consistency results. We hope this will help consistency techniques to reach a broader audience and help the researchers in other areas to contribute new consistency results.

In this paper, we show a number of properties of set intersections of various special sets such as convex sets, tree convex sets, small sets whose size is smaller than a given number, and large sets whose size is larger than a given number. These results not only immediately lead to the existing consistency results, but also new results. For example, we identify a new class of *tree convex* constraints, which generalises row convex constraints, where local consistency ensures global consistency. New consistency results are obtained on a constraint network where only *some*, in contrast to *all* in the existing work, constraints are tight. We also study tightness and tree convexity in the light of relational consistency. These results significantly improve our understanding of convex, tight and loose constraints, and demonstrate that set intersection is a promising and powerful tool for studying consistency in a constraint network.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Set theory*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Relation systems*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Trees*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Constraint-based reasoning, constraint networks, local consistency, relations, tree convex sets, tree convex constraints, weakly m-tight constraint networks

---

Some of the results reported in this paper have previously appeared as Zhang, Y., and Yap, R.H.C., "Consistency and Set Intersection", *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003, pp. 263–268.

Author's addresses: Yuanlin Zhang, Cork Constraint Computation Center, University College Cork, Cork, Ireland. Email: yzhang@4c.ucc.ie; Roland H. C. Yap, Department of Computer Science, National University of Singapore, 3 Science Drive 2, Singapore 117543. Email: `ryap@comp.nus.edu.sg`

1. INTRODUCTION

A constraint network consists of a set of variables, over some finite domains, together with an associated system of constraints over those variables. An important task is to find an assignment for all the variables such that all the constraints in the network are satisfied, If such an assignment exists, the network is *satisfiable* or *globally consistent* and the assignment is called a solution. The problem of determining the global consistency of a general constraint network is known to be NP-complete. Usually a search procedure is used to find a solution. In practice, due to efficiency considerations, the search is usually equipped with a filtering algorithm which prunes values of a variable or the combinations of values of a certain number of variables which cannot be part of a solution. The filtering algorithm can make a network *locally consistent* in the sense that a consistent assignment of some variables can always be extensible to a new variable. An important and interesting question on local consistency is:

> Is the local consistency obtained sufficient to determine global consistency of the network without further search? As the filtering algorithm is of polynomial complexity, the positive answer would mean that the network can be solved in polynomial time.

Much work has been done to explore the relationship between local and global consistency (in particular) and the properties of local consistency in general. One direction is to make use of the topological structure of the graph representation of a constraint network. A classical result is that when the graph of a constraint network is a tree, arc consistency in the network is sufficient to make it globally consistent [Freuder 1982].

The second direction makes use of semantic properties of the constraints. For *monotone constraints*, path consistency implies global consistency [Montanari 1974]. Van Beek and Dechter [1995] generalize monotone constraints to a larger class of *row convex constraints*. Dechter [1992] shows that a certain level of consistency in a network whose domains are of limited size ensures global consistency. Later, Van Beek and Dechter [1997] study the consistency of a network with tight and loose constraints.

The existing work along the two approaches has used specific and different techniques to study local and global consistency. In particular, there is little commonality in the details of the existing work. In much of the existing work, the techniques and consequently the proofs given are developed specifically for the result concerned.

In this paper, we show how much of this work can be connected together. A new approach is used to study consistency in a constraint network. We unite two seemingly disparate areas: the study of *set intersection* on special sets and the study of *k-consistency* in constraint networks. We show that k-consistency can be expressed in terms of set intersection. This allows one to obtain relationships between local and global consistency in a constraint network in terms of properties of set intersection on special sets. The main result of this approach is a proof schema which can be used to lift results from set intersection, which are rather general, to particular consistency results on constraint networks. One benefit of the proof schema lies in that it provides a modular way to greatly simplify the understanding and proof of results on consistency. This benefit is considerable as often the proof of many existing results is complex and "hard-wired". Using this new approach, we show that it is precisely the various properties of set intersection that lead to the results in most existing work.

The following sketch illustrates briefly the use of our approach. One property of set intersection is that if the intersection of every pair (*2*) of *tree convex sets* (see Section 3) is not empty, the intersection of the whole collection of these sets is also not empty. From this property, we can see that local information on intersection of every pair of sets gives global information on intersection of all sets. Intuitively, this relationship between local and global information corresponds to obtaining global consistency from local consistency. The proof schema is used to lift the result on tree convex sets to the following result on consistency. For a binary network of tree convex constraints, (*2*+1)-consistency (path consistency) implies global consistency.

The usefulness of our new set-based approach are many fold. Firstly, it gives a clear picture of many of the existing results and is a unifying theoretical tool for studying consistency. For example, many well known results in the second direction based on semantic properties of the constraints (including [van Beek and Dechter 1995; 1997]) but also results from the first direction can be shown with easy proofs which make use of properties of set intersection. Secondly, it is useful for improving some of the existing results and for deriving new results as is demonstrated in sections 5–9. Thirdly, we believe it introduces a new direction to understanding consistency in constraint networks and gives a general tool for analysing new kinds of constraints, local consistency and topologies.

There is a difference between the work reported here and the work that studys the tractability of constraint languages(e.g., [Schaefer 1978; Jeavons et al. 1997]). The latter considers the problems whose constraints are from a *fixed set of relations* while the work here concerns with the problems (constraint networks) with *special properties*.

This paper is organized as follows. Section 2 gives some basic definitions. Section 3 presents properties of the intersection for *tree convex* sets, *small sets* and *large sets*. Section 4 develops a characterisation of k-consistency utilising set intersection and develops the proof schema which gives a generic way to obtain consistency results from properties of set intersection. We demonstrate the power of the new approach by applying it to derive three new classes of results on global and local consistency as detailed below as well as a number of well known results. The first is presented in Section 5. It is a generalization of row convex constraints to tree convex constraints. On a network of tree convex constraints local consistency ensures global consistency. The second is on global consistency on *weakly tight* networks and presented in Section 6. These networks only require certain constraints to be m-tight rather than all constraints as shown in [van Beek and Dechter 1997]. Section 7 revisits the results on the intrinsic consistency of loose networks. The third result in Section 8 is on networks which are properly m-tight. It advances the previous work [van Beek and Dechter 1997] since networks with certain constraints satisfying tightness restrictions can be made globally consistent by enforcing some level of relational consistency. Lastly, Section 9 gives new versions of weak tightness and tree convexity using relational consistency. Section 10 discusses related work and concludes.

2. PRELIMINARIES

A *constraint network* $\mathcal{R}$ is defined as a set of variables $N = \{x_1, x_2, \cdots, x_n\}$; a set of finite domains $D = \{D_1, D_2, \cdots, D_n\}$ where domain D_i, for all $i \in 1..n$, is a set of values that variable x_i can take; and a set of constraints $C = \{c_{S_1}, c_{S_2}, \cdots, c_{S_e}\}$ where S_i, for all $i \in 1..e$, is a subset of $\{x_1, x_2, \cdots, x_n\}$ and each constraint c_{S_i} is a relation defined on domains of all variables in S_i. The *arity* of constraint c_{S_i} is the number of variables in S_i. For a variable x, D_x denotes the domain of variable x.

An instantiation of variables $Y = \{x_1, \cdots, x_j\}$ is denoted by $\bar{a} = (a_1, \cdots, a_j)$ where $a_i \in D_i$ for $i \in 1..j$. An *extension* of $\bar{a}$ to a variable $x (\notin Y)$ is denoted by $(\bar{a}, u)$ where $u \in D_x$. An instantiation of a set of variables Y is *consistent* if it satisfies all constraints in $\mathcal{R}$ which don't involve any variable outside Y.

A constraint network $\mathcal{R}$ is *k-consistent* if and only if for any consistent instantiation $\bar{a}$ of any distinct $k-1$ variables, and for any new variable x, there exists $u \in D_x$ such that $(\bar{a}, u)$ is a consistent instantiation of the k variables. $\mathcal{R}$ is *strongly* k-consistent if and only if it is j-consistent for all $j \leq k$. A strongly n-consistent network is called *globally consistent*.

More information on constraint networks and consistency can be found in [Mackworth 1977; Freuder 1978; Dechter 2003].

3. PROPERTIES ON SET INTERSECTION

In this section, we develop the underlying set intersection results which are useful for the later results on consistency. The set intersection property which we are concerned with is:

> *Given a collection of l finite sets, under what conditions is the intersection of all l sets not empty?*

This property is not very useful for collections of arbitrary sets. Here, we study sets with two restrictions: convexity and cardinality.

3.1 Sets with Convexity Restrictions

We first define the convexity of sets, especially including discrete sets.

Definition 1. Given a set U and a total ordering "$\preceq$" on it, a set $A \subset U$ is *convex* if the elements in it are consecutive under the ordering, that is if $u, w \in A$ then for any $v \in U$ and $u \preceq v \preceq w$, $v \in A$.

Definition 2. Given a collection of sets $\mathcal{S}$, let the union of all the sets in $\mathcal{S}$ be U. The sets in $\mathcal{S}$ are *convex* under a total ordering on U if every set in $\mathcal{S}$ is convex under the ordering. The sets in $\mathcal{S}$ are said to be convex if they are convex under some total ordering on U.

Example 1. The set of real numbers between 1 and 2 is convex under the usual ordering of numbers. $\{1, 9\}$ and $\{3, 9\}$ are convex with a total ordering $1 \preceq 9 \preceq 3$. However, $\{1, 9\}, \{3, 9\}$ and $\{5, 9\}$ are not convex under any total ordering on $\{1, 3, 5, 9\}$.

The following result shows a property on the intersection of convex sets.

LEMMA 1 CONVEX SETS INTERSECTION. *Given a finite collection of convex sets* $\mathcal{S}$, $\bigcap_{E \in \mathcal{S}} E \neq \emptyset$ *iff for all* $E_1, E_2 \in \mathcal{S}$, $E_1 \bigcap E_2 \neq \emptyset$.

This result lies in the heart of Lemma 3.1 in [van Beek and Dechter 1995] which is based on the concept of constraints and the matrix representation of constraints. However, Lemma 3.1 does not introduce the concept of convex sets explicitly.

The concept of convex sets imposes a strong requirement that the sets are "dense" under a common total ordering. In fact, the total ordering here can be generalized to a tree for discrete sets as follows.

Definition 3. Given a discrete set U and a tree $\mathcal{T}$ with vertices U. A set $A \subseteq U$ is *tree convex* under $\mathcal{T}$ iff there exists a subtree of $\mathcal{T}$ whose set of vertices is exactly A.

Definition 4. Given a collection of discrete sets $\mathcal{S}$, let the union of the sets in $\mathcal{S}$ be U. The sets in $\mathcal{S}$ are tree convex under a tree $\mathcal{T}$ on U if every set in S is tree convex under $\mathcal{T}$.

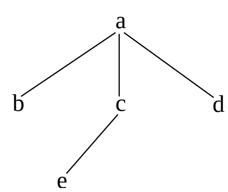

Fig. 1. A tree with nodes a, b, c, d, e

Example 2. Consider a set $U = \{a, b, c, d, e\}$ and a tree given in Fig 1. The subset $\{a, b, c, d\}$ is tree convex. So is the set $\{b, a, c, e\}$ since the elements in the set consists of a subtree. However, $\{b, c, e\}$ is not tree convex as it does not form a subtree of the given tree.

Consider $\mathcal{S} = \{\{1, 9\}, \{3, 9\}, \{5, 9\}\}$. A tree can be constructed on $\{1, 3, 5, 9\}$ with 9 being the root and $1, 3, 5$ being its children. Each set in $\mathcal{S}$ covers the nodes of exactly one branch of the tree. Hence, the sets in $\mathcal{S}$ are tree convex.

Tree convex sets have the following intersection property.

LEMMA 2 TREE CONVEX SETS INTERSECTION. *Given a finite collection of finite sets* $\mathcal{S}$, *assume the sets in* $\mathcal{S}$ *are tree convex.* $\bigcap_{E \in \mathcal{S}} E \neq \emptyset$ *iff for all* $E_1, E_2 \in \mathcal{S}$, $E_1 \bigcap E_2 \neq \emptyset$.

Proof. Let l be the number of sets in $\mathcal{S}$, and $\mathcal{T}$ a tree such that there exists a subtree T_i for each $E_i \in \mathcal{S}$ such that the set of vertices of T_i is E_i. We can regard $\mathcal{T}$ as a rooted tree and thus every T_i $(i \in 1..l)$ can be regarded as a rooted tree whose root is exactly the node nearest to the root of $\mathcal{T}$. Let r_i denote the root of T_i for $i \in 1..l$.

To prove $\bigcap_{i \in 1..l} E_i \neq \emptyset$, we want to show the intersection of the trees $\{T_i \mid i \in 1..l\}$ is not empty. The following propositions on subtrees are necessary in our main proof.

PROPOSITION 3. *Let T_1, T_2 be two subtrees of a tree $\mathcal{T}$, and $T = T_1 \cap T_2$. T is a tree.*

If $T = \emptyset$, it is a trivial tree. Now let $T \neq \emptyset$. Since T is a portion of T_1, there is no circuit in it. It is only necessary to prove T is connected. That is to show, for any two nodes $u, v \in T$, there is a path between them. $u, v \in T_1$ and $u, v \in T_2$ respectively imply that there exist paths $P_1 : u, \cdots, v$ in T_1 and $P_2 : u, \cdots, v$ in T_2 respectively. Recall that *there is a unique path from u to v in $\mathcal{T}$* and that T_1 and T_2 are subtrees of $\mathcal{T}$. Therefore, P_1 and P_2 cover the same nodes and edges, and they are in T, the intersection of T_1 and T_2. P_1 is the path we want.

PROPOSITION 4. *Let T_1, T_2 be two subtrees of a tree $\mathcal{T}$, and $T = T_1 \cap T_2$. T is not empty if and only if at least one of the roots of T_1 and T_2 is in T.*

Let r_1 and r_2 be the roots of T_1 and T_2 respectively. If $r_1 \in T$, the proposition is correct. Otherwise, we show $r_2 \in T$. Assume the contrary $r_2 \notin T$. Let r be the root of $\mathcal{T}$ and v the root of T (T is a tree in terms of Proposition 3). We have paths $P_1 : r_1, \cdots, v$ in T_1; $P_2 : r_2, \cdots, v$ in T_2; and $P_3 : r, \cdots, r_1$, and $P_4 : r, \cdots, r_2$ in $\mathcal{T}$. The assumption tells that $r_1 \neq r_2$. From the closed walk $P_3 P_1 P_2' P_4'$ where P_2' and P_4' are the reverse of P_2 and P_4 respectively, we can construct a circuit containing at least r_1 and r_2. It contradicts that there is no circuit in $\mathcal{T}$.

Further we have the following observation.

PROPOSITION 5. *Let the root of $\mathcal{T}$ be r. Given two subtrees T_1 and T_2 of $\mathcal{T}$ with roots r_1 and r_2 respectively. Let r_1 be not closer to r than r_2, and T the intersection of T_1 and T_2. r_1 is the root of T if T is not empty.*

Let r_1 be farther to the r than r_2. Assume r_2 is the root of T. Since r_1 is farther to r than r_2, r_2 is not possible to be a node of T_1. It contradicts that $r_2 \in T$.

Let $T = \bigcap_{i \in 1..l} T_i$. We are ready now to prove our main result $T \neq \emptyset$. We select a tree $T_{\max}$ from $T_1, T_2, \cdots, T_l$ such that its root $r_{\max}$ is the farthest away from r of $\mathcal{T}$ among the roots of the concerned trees. In terms of Proposition 5, that $T_{\max}$ intersect with every other trees implies that $r_{\max}$ is a node of every T_i $(i \in 1..l)$. Therefore, $r_{\max} \in T$. $\square$

Remark. Recall that a partial order can be represented by an acyclic directed graph. It is tempting to further generalize the tree convexity to partial convexity in the following way.

Definition 5. Given a set U and a partial order on it. A set $A \subset U$ is *partially convex* if and only if A is the set of nodes of a connected subgraph of the partial order. Given a collection of sets $\mathcal{S}$, let the union of the sets of $\mathcal{S}$ be U. The sets in $\mathcal{S}$ are *partially convex* if there is a partial ordering on U such that every set in $\mathcal{S}$ is convex under the ordering.

However, with this generalization, we do not get a result similar to Lemma 2. This is illustrated by the following counterexample. Consider three sets $\{c, b, d\}$,

$\{d, f, a\}$ and $\{a, e, c\}$ which are the nodes of some subgraph given in Fig 2. These sets are partially convex and intersect pairwise. However, the intersection of all three sets is empty.

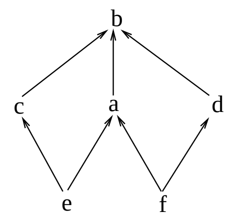

Fig. 2. A partial order with nodes $\{a, b, c, d, e, f\}$

3.2 Sets with Cardinality Restrictions

Another useful restriction which we will use on sets is to restrict their cardinality. The following results are on arbitrary finite sets where the only restriction is on the size of sets.

LEMMA 6 SMALL SET INTERSECTION. *Given a number m and a finite collection of l sets $\mathcal{S}$, let $m < l$. Assume one set from $\mathcal{S}$ has at most m elements.*

$$\bigcap_{E \in \mathcal{S}} E \neq \emptyset$$

iff the intersection of any $m+1$ sets from $\mathcal{S}$ is not empty.

Proof. The necessary condition is immediate.

The sufficient condition is proved by induction on l. It is obviously true when $l \leq m+1$. Assuming that k $(> m)$ sets intersect, we show that any $k+1$ sets intersect. Without loss of generality, the subscripts of the $k+1$ sets are numbered from 1 to $k+1$ and let $|E_1| \leq m$. Let A_i be the intersection of all the $k+1$ sets except E_i:

$$A_i = E_1 \cap \cdots \cap E_{i-1} \cap E_{i+1} \cap \cdots \cap E_{k+1}, \; for \; 1 < i \leq k+1.$$

If $A_i \cap A_j \neq \emptyset$ for some $i, j \in 2..k+1, i \neq j$,

$$\bigcap_{i \in 1..k+1} E_i = A_i \cap A_j \neq \emptyset.$$

Assume the contrary that $A_i \cap A_j = \emptyset$ for all distinct i and j. According to the construction of A_i's,

$$E_1 \supseteq \bigcup_{i \in 2..k+1} A_i,$$

and $|A_i| \geq 1$ by the induction assumption. Henceforth,

$$|E_1| \geq \sum_{i \in 2..k+1} |A_i| \geq k > m$$

which contradicts $|E_1| \leq m$. □

An immediate corollary of this lemma is when all the sets of concern have at most m elements.

COROLLARY 7 SMALL SETS INTERSECTION. *Given a number m and a finite collection of l sets $\mathcal{S}$, let $m < l$. For every $E \in \mathcal{S}$, assume E is finite and $|E| \leq m$. $\bigcap_{E \in \mathcal{S}} E \neq \emptyset$ iff the intersection of any $m+1$ sets from $\mathcal{S}$ is not empty.*

The small sets intersection lemma singles out the essential component of Lemma 3.2 in [van Beek and Dechter 1997]: The property of sets determines the behavior of the corresponding constraints (or relations). This observation is not explicit in Lemma 3.2 and its proof since Lemma 3.2 takes constraints (or relations) as a primitive concept. By shifting the primitive concept from constraints to sets, the foundamental properties (leading to consistency results) become more prominent, and both the results and reasoning process are simplified and made more focused. It is the emphasis on set intersection properties that results in the discovery of Lemma 6.

Another special case is that some set has only *one* element.

COROLLARY 8 SINGLETON SET INTERSECTION. *Given a collection of sets $\mathcal{S}$, assume one set from $\mathcal{S}$ has only one element. $\bigcap_{E \in \mathcal{S}} E \neq \emptyset$ iff all sets mutually intersect.*

Based on sets, rather than constraints, this result is straightforward. Since E has only one element and its intersection with any other set is not empty, the element in E is the one shared by all sets. In a simple and clear way, this corollary presents the foundamental property leading to the consistency results reported in [David 1993].

Motivated by [van Beek and Dechter 1997, lemma 4.1 in page 561], we consider the following restrictions on a collection of sets: (1) each set has size larger than some number m; but (2) there is a small number of sets in the collection; and (3) the union U of all sets has limited size d. The name of the lemma, large sets intersection, is after the first restriction. In this case, if the intersection of all sets is empty, then for any $a \in U$, a is excluded by some set E_i. However, since E_i is large, it can exclude at most $d-m$ elements in U. All sets in $\mathcal{S}$ can exclude at most $l \times (d-m)$ elements in U. When l is also small (such that $l(d-m) < d$), some element in U may not be excluded by any set, which means that the intersection of all sets is not empty.

LEMMA 9 LARGE SETS INTERSECTION. *Given a number m and a collection of l discrete sets $\mathcal{S}$, for all $E \in \mathcal{S}$, assume E is finite and $|E| \geq m$. Let $|\bigcup_{E \in \mathcal{S}} E| = d$. If $l \leq \lceil d/(d-m) \rceil - 1$, then $\bigcap_{E \in \mathcal{S}} E \neq \emptyset$.*

Proof. Let $\mathcal{S} = \{E_1, E_2, \ldots, E_l\}$, $U = \bigcup_{i \in 1..l} E_i$, and $A_i = U - E_i$ for all $i \leq l$. It is immediate that

$$\bigcup_{i \in 1..l} A_i \subseteq U.$$

We know

$$|\bigcup_{i\in 1..l} A_i| \le \sum_{i\in 1..l} |A_i|.$$

For $|A_i| \le d-m$, we have

$$\sum_{i\in 1..l} |A_i| \le \sum_{i\in 1..l} (d-m) = l(d-m) < d.$$

Hence, $\bigcup_{i\in 1..l} A_i$ is a proper subset of U. There exists $x \in U$ such that $x \notin A_i$ for all $i \le l$, which implies that $x \in E_i$ for all $i \le l$. □

4. SET INTERSECTION AND CONSISTENCY

In this section, we first relate consistency in constraint networks to set intersection. Using this result, we present a proof schema which allows us to study the relationship between local and global consistency from the properties of set intersection.

Underlying the concept of k-consistency is whether an instantiation of some variables can be extended to a new variable such that all relevant constraints to the new variable are satisfied. A *relevant* constraint to a variable x_i is a constraint where only x_i is uninstantiated (and the others are instantiated). Each relevant constraint allows a set (possibly empty) of values for the new variable. This set is called *extension set* below. The satisfiability of all relevant constraints depends on whether the intersection of their extension sets is non-empty (see lemma 10).

Definition 6. Given a constraint c_{S_i}, a variable $x \in S_i$ and any instantiation $\bar{a}$ of $S_i - \{x\}$, the *extension set* of $\bar{a}$ to x with respect to c_{S_i} is defined as

$$E_{i,x}(\bar{a}) = \{b \in D_x \mid (\bar{a}, b) \ satisfies \ c_{S_i}\}.$$

An extension set is *trivial* if it is empty; otherwise it is *non-trivial*.

Remember that D_x refers to the domain of variable x. Throughout the paper, it is often the case that an instantiation $\bar{a}$ of $S - \{x\}$ is already given, where $S - \{x\}$ is a superset of $S_i - \{x\}$. Let $\bar{b}$ be the instantiation obtained by restricting $\bar{a}$ to the variables only in $S_i - \{x\}$. For ease of presentation, we continue to use $E_{i,x}(\bar{a})$, rather than $E_{i,x}(\bar{b})$, to denote the extension of $\bar{b}$ to x under constraint c_{S_i}. To make the presentation easy to follow, some of the three parameters i, $\bar{a}$ and x may be omitted from an expression hereafter whenever they are clear from the context. For example, given an instantiation $\bar{a}$ and a new variable x, to emphasize different extension sets with respect to different constraints R_{S_i}, we write E_i instead of $E_{i,x}(\bar{a})$ to simply denote an extension set.

Example 3. Consider a network with variables $\{x_1, x_2, x_3, x_4, x_5\}$:

$$\begin{array}{rcll} c_{S_1} &=& \{(a,b,d),(a,b,a)\}, & S_1 = \{x_1,x_2,x_3\}; \\ c_{S_2} &=& \{(b,a,d),(b,a,b)\}, & S_2 = \{x_2,x_4,x_3\}; \\ c_{S_3} &=& \{(b,d),(b,c)\}, & S_3 = \{x_2,x_3\}; \\ c_{S_4} &=& \{(b,a,d),(b,a,a)\}, & S_4 = \{x_2,x_5,x_3\}; \\ \multicolumn{4}{c}{D_1 = D_4 = D_5 = \{a\}, D_2 = \{b\}, D_3 = \{a,b,c,d\}.} \end{array}$$

Let $\bar{a} = (a, b, a)$ be an instantiation of variables $Y = \{x_1, x_2, x_4\}$. The relevant constraints to x_3 are c_{S_1}, c_{S_2}, and c_{S_3}. c_{S_4} is not relevant since it has two uninstantiated variables. The extension sets of $\bar{a}$ to x_3 with respect to the relevant constraints are:

$$E_1(\bar{a}) = \{d, a\}, E_2(\bar{a}) = \{d, b\}, E_3(\bar{a}) = \{d, c\}.$$

The intersection of the extension sets above is not empty, implying that $\bar{a}$ can be extended to satisfy all relevant constraints c_{S_1}, c_{S_2} and c_{S_3}.

Let $\bar{a} = (b, c)$ be an instantiation of $\{x_2, x_3\}$. $E_{1,x_1}(\bar{a}) = \emptyset$ and thus it is trivial. In other words, being a trivial extension set, an instantiation can not be extended to satisfy the constraint of concern.

The relationship between *k-consistency* and set intersection is characterized by the following lemma which is a direct consequence of the definition of *k-consistency.*

LEMMA 10 SET INTERSECTION AND CONSISTENCY; LIFTING. *A constraint network $\mathcal{R}$ is k-consistent if and only if for any consistent instantiation $\bar{a}$ of any $(k-1)$ distinct variables $Y = \{x_1, x_2, \cdots, x_{k-1}\}$, and any new variable x_k,*

$$\bigcap_{j \in 1..l} E_{i_j} \neq \emptyset$$

where E_{i_j} is the extension set of $\bar{a}$ to x_k with respect to $c_{S_{i_j}}$, and $c_{S_{i_1}}, \cdots, c_{S_{i_l}}$ are all relevant constraints.

Proof. It follows directly from the definition of k-consistency (in Section 2) and the definition of extension set. □

The insight behind this lemma is to examine consistency from the perspective of set intersection.

Example 4. Consider again the previous example. We would like to check whether the network is 4-consistent. Consider the instantiation $\bar{a}$ of Y again. This is a trivial consistent instantiation since the network doesn't have a constraint among the variables in Y. To extend it to x, we need to check the first three constraints. The extension is feasible because the intersection of E_1, E_2, and E_3 is not empty. We show the network is 4-consistent, by exhausting all consistent instantiations of any three variables. Conversely, if we know the network is 4-consistent, we can immediately say that the intersection of the three extension sets of $\bar{a}$ to x is not empty.

The usefulness of this lemma is that it allows consistency information to be obtained from the intersection of extension sets, and also vice versa. Using this view of consistency as set intersection, some results on set intersection properties, including all those in section 3, can be *lifted* to get various consistency results for a constraint network by making use of the following *proof schema.*

Proof Schema

1. (*Consistency to Set*) From a certain level of consistency *in* the constraint network, we derive information on the intersection of the extension sets by Lemma 10.

2. (*Set to Set*) From the *local* intersection information of sets, information may be obtained on intersection of more sets.

3. (*Set to Consistency*) From the new information on intersection of extension sets, higher level of consistency is obtained according to Lemma 10.

4. (*Formulate conclusion on the consistency of the constraint network*). □

In the proof schema, step 1 (consistency to set), step 3 (set to consistency), and step 4 are straightforward in many cases. So, Lemma 10 is also called the *lifting* lemma because once we have a set intersection results (step 3), we can easily have consistency results on a network (step 4). The proof schema establishes a direct relationship between set intersection and consistency properties in a constraint network.

In the following sections, we demonstrate how the set intersection properties and the proof schema are used to obtain new and also well known results on consistency of a network.

5. APPLICATION I: GLOBAL CONSISTENCY OF TREE CONVEX CONSTRAINTS

The notion of *extension set* plays the role of a bridge between the restrictions to set(s) and properties of special constraints. The sets in Lemma 2 are restricted to be tree convex. A constraint is *tree convex* if all extension sets with respect to the constraint are tree convex.

Definition 7. A constraint c_S is *tree convex* with respect to x_i and a tree T_i on D_i if and only if the sets in

$$A = \{E_{S,x_i} \mid E_{S,x_i} \text{ is a non-trivial extension of some instantiation of } S - \{x_i\}\}$$

are tree convex under T_i. A constraint c_S is *tree convex* under a tree T on the union of the domains of the variables in S, if it is tree convex wrt every $x \in S$ under T.

Example 5. Tree convex constraints could occur in practice where there is a structure among the values of a variable. Consider the constraint on the accessibility of a set of facilities by a set of persons. The personnel includes a network engineer, web server engineer, application engineer, and a team leader. The relationship among the staffs is that the leader manages the rest, which forms a tree structure shown in Fig. 3(b). There are different accessibilities to a system which include basic access, access to the network routers, access to the web server, and access to the file server. In order to access the routers and servers, one has to have the basic access right, impying a tree structure (Fig. 3(c)) on the accessibilities. The constraint is that the leader is able to access all the facilities while each engineer can access the corresponding facility (e.g., the web server engineer can access the web server). This tree convex constraint is shown in Fig. 3(a) where the rows are named by (the initials of) staffs and the columns by (the initials of) accessibilities. The tree on the union of personnel and the accessibilities can be obtained from their respective trees (in Fig. 3(b) and (c)) by adding an edge, say between web server and leader.

Tree convex constraints can also be used to model scene labeling problems naturally as shown in [Zhang and Freuder 2004].

Definition 8. A constraint network is *tree convex* if there exists a tree $\mathcal{T}$ on the union of all its variable domains such that all constraints are tree convex under $\mathcal{T}$.

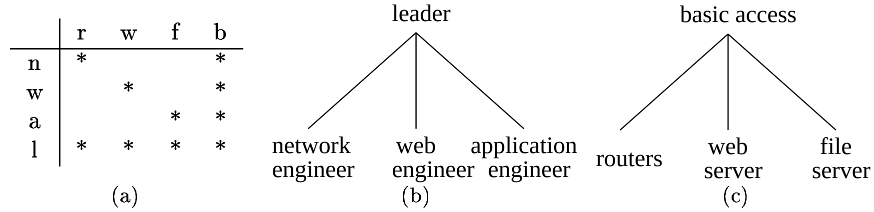

Fig. 3. A tree convex constraint between accessibilities and staffs

Convex sets naturally give rise to convex constraints which is a special case of tree convex constraints.

Definition 9. A constraint c_S is *row convex* with respect to x if and only if the sets in

$$A = \{E_{S,x} \mid E_{S,x} \text{ is a non-trivial extension of some instantiation of } S - \{x\}\}$$

are convex. It is row convex if under a total ordering on the union of involved domains, it is row convex wrt every $x \in S$. A constraint network is *row convex* iff there exist a total ordering on the union of all domains such that all constraints are row convex under the ordering.

Example 6. For the constraint c in Example 5 to be row convex, b(basic access) has to be the neighbor of r(routers), w(web server), and f(file server). However, in a total ordering, a value can be the neighbor of at most two other values. Hence, c is not row convex.

The consistency results on these networks can be derived from the property of set intersection using the proof schema. We obtain the main result of this section.

THEOREM 11 TREE CONVEXITY. *Let $\mathcal{R}$ be a network of constraints with arity at most r and strongly $2(r-1)+1$ consistent. If $\mathcal{R}$ is tree convex then it is globally consistent.*

Proof. The network is strongly $2(r-1)+1$ consistent by assumption. We prove by induction that the network is k consistent for any $k \in \{2r, \cdots, n\}$.

Consider any instantiation $\bar{a}$ of any $k-1$ variables and any new variable x. Let the number of relevant constraints be l. For each relevant constraint there is one extension set of $\bar{a}$ to x. So we have l extension sets. If the intersection of all l sets is not empty, we have a value for x such that the extended instantiation satisfies all relevant constraints.

(*Consistency to Set*) Consider any two of the l extension sets: E_1 and E_2. The two corresponding constraints involve at most $2(r-1)+1$ variables since the arity of a constraint is at most r and each of the two constraints has x as a variable. According to the consistency lemma, that $\mathcal{R}$ is $2(r-1)+1$-consistent implies that the intersection of E_1 and E_2 is not empty.

(*Set to Set*) Since all relevant constraints are tree convex under the given tree, the extension sets of $\bar{a}$ to x are tree convex. Henceforth, the fact that every two of the extension sets intersect shows that the intersection of all l extension sets is not empty, in terms of the *tree convex sets intersection* lemma.

(*Set to Consistency*) From the consistency lemma, we have that $\mathcal{R}$ is k-consistent. □

Since a row convex constraint is tree convex, we have the following result.

COROLLARY 12 ROW CONVEXITY. [van Beek and Dechter 1995] *Let $\mathcal{R}$ be a network of constraints with arity at most r and strongly $2(r-1)+1$ consistent. If there exists an ordering of the domains $D_1, \cdots, D_n$ of $\mathcal{R}$ such that $\mathcal{R}$ is row convex, $\mathcal{R}$ is globally consistent.*

This can also be proved directly by lifting the convex sets Lemma 1. In other words, by the property of the intersection of convex sets, we can straightforwardly have the row convexity result by the lifting lemma and proof schema. This straightforwardness is not obvious in [van Beek and Dechter 1997].

An associated question with Theorem 11 is how efficient it is to check whether a constraint network is tree convex. Recently, Yosiphon [2003] has proposed an algorithm to recognize a tree convex constraint network in polynomial time.

6. APPLICATION II: GLOBAL CONSISTENCY ON WEAKLY TIGHT NETWORKS

In this section, we study networks with some tight constraints. The *m-tight* property of a constraint is related to the cardinality of the extension set in the following way.

Definition 10. [van Beek and Dechter 1997] A constraint c_{S_i} is *m-tight* with respect to $x \in S_i$ iff for any instantiation $\bar{a}$ of $S_i - \{x\}$,

$$|E_{i,x}| \leq m \text{ or } |E_{i,x}| = |D_x|.$$

A constraint c_{S_i} is *m-tight* iff it is m-tight with respect to every $x \in S_i$.

Given an instantiation, if its extension set with respect to x is the same as the domain of variable x ($|E_{i,x}| = |D_x|$), the instantiation is supported by all values of x and thus easy to be satisfiable. Hence, in the definition above, these instantiations do not affect the m-tightness of a constraint.

Definition 11. A constraint network is *weakly* m-tight at level k iff for every set of variables $\{x_1, \cdots, x_l\}(k \leq l < n)$ and a new variable, there exists an m-tight constraint in the relevant constraints after the instantiation of the l variables.

The small set intersection lemma (Lemma 6) gives the following theorem.

THEOREM 13 WEAK TIGHTNESS. *If a constraint network $\mathcal{R}$ with constraints of arity at most r is strongly $((m+1)(r-1)+1)$-consistent and weakly m-tight at level $((m+1)(r-1)+1)$, it is globally consistent.*

Proof. Let $j = (m+1)(r-1)+1$. The constraint network $\mathcal{R}$ will be shown to be k-consistent for all k ($j < k \leq n$).

Let $Y = \{x_1, \cdots, x_{k-1}\}$ be a set of any $k-1$ variables, and $\bar{a}$ be an instantiation of all variables in Y. Consider any additional variable x_k. Without loss of generality, let the relevant constraints be $c_{S_1}, \cdots, c_{S_l}$, and E_i be the extension set of $\bar{a}$ to x_k with respect to c_{S_i} for $i \leq l$.

(*Consistency to Set*) Consider any $m+1$ of the l extension sets. All the corresponding $m+1$ constraints contain at most $(m+1)(r-1)+1$ variables including x_k. Since $\mathcal{R}$ is $(m+1)(r-1)+1$-consistent, according to the *set intersection and consistency* lemma, the intersection of $m+1$ extension sets is not empty.

(*Set to Set*) The network is weakly m-tight at level $((m+1)(r-1)+1)$. So, there must be an m-tight constraint among the relevant constraints $c_{S_1}, \cdots, c_{S_l}$. Let it be c_{S_i}. We know its extension set $|E_i| \leq m$. For the intersection of every $m+1$ of the extension sets is not empty, all l extension sets share a common element in terms of the *small set intersection* lemma.

(*Set to Consistency*) From the lifting lemma, $\mathcal{R}$ is k-consistent. □

Immediately we have the following result which is a main result in [van Beek and Dechter 1997].

COROLLARY 14 TIGHTNESS. [van Beek and Dechter 1997] *If a constraint network $\mathcal{R}$ with constraints that are m-tight and of arity at most r is strongly $((m+1)(r-1)+1)$-consistent, then it is globally consistent.*

This result can of course be directly lifted from the small sets Corollary 7. There is no difference between this proof and the proofs for other resutls (e.g., tree convexity and weak tightness theorems). This uniformness is absent from the proofs in the existing work (e.g., [Dechter 1992; van Beek and Dechter 1995; 1997; David 1993]).

Corollary 14 requires every constraint to be m-tight. The weak tightness theorem, on the other hand, does not require all constraints to be m-tight. The following example illustrates this difference.

extension	relevant constraints									
1234 → 5,	125*,	135 ,	145 ,	235,	245,	345,	15+,	25 ,	35 ,	45
2345 → 1,	231 ,	241 ,	251*,	341,	351,	451,	21 ,	31 ,	41 ,	51+
3451 → 2,	132 ,	142 ,	152*,	342,	352,	452,	12 ,	32+,	42 ,	52
4512 → 3,	123 ,	143*,	153 ,	243,	253,	453,	13 ,	23+,	43 ,	53
5123 → 4,	124 ,	134*,	154 ,	234,	254,	354,	14 ,	24 ,	34+,	54

Table I. Relevant constraints in extending an instantiation of four variables to a new variable

Example 7. For a weakly m-tight network, we are interested in its topological structure. Thus we have omitted the domains of variables here. Consider a network with five variables labelled $\{1, 2, 3, 4, 5\}$. In this network, for any pair of variables and for any three variables, there is a constraint. Assume the network is already strongly 4-consistent.

Since the network is already strongly 4-consistent, we can simply ignore the instantiations with less than 4 variables. This is why we introduce the level at which the network is weakly m-tight. The interesting level here is 4. Table I shows the relevant constraints for each possible extension of four instantiated variables to the other one. In the first row, $1234 \to 5$ stands for extending the instantiation of variables $\{1, 2, 3, 4\}$ to variable 5. Entries in its second column denote a constraint. For example, 125 denotes c_{125}. If the constraints on $\{1, 2, 5\}$ and $\{1, 3, 4\}$ (suffixed by * in the table) are m-tight, the network is weakly m-tight at level 4. Or, if the constraints $\{1, 5\}$, $\{2, 3\}$ and $\{3, 4\}$ (suffixed by +) are m-tight, the network will also be weakly m-tight. However, the tightness corollary requires all binary and ternary constraints to be m-tight. The weak m-tightness theorem needs significantly less constraints to be m-tight. Further results are given in Section 9.

7. APPLICATION III: CONSTRAINT LOOSENESS

The next result is a consequence of the large sets intersection lemma. For large sets, their intersection is not empty as long as they are large enough. It means that there is certain level of consistency in a constraint network characterized by a large set. This is in contrast to the previous results where global consistency is implied by certain level of local consistency.

The *m-loose* property of a constraint is related to the cardinality of the extension set in the following way.

Definition 12. [van Beek and Dechter 1997] A constraint c_{S_i} is *m-loose* with respect to $x \in S_i$ if and only if for any instantiation $\bar{a}$ of $S_i - \{x\}$,

$$|E_i| \geq m.$$

A constraint c_{S_i} is *m-loose* if and only if it is m-loose with respect to every $x \in S_i$.

For example, the constraint $x \leq y$, where $x \in \{1, 2, \cdots, 10\}$ and $y \in \{1, 2, \cdots, 10\}$, is 1-loose.

The *large set intersection* lemma is lifted to the following result on *constraint looseness.*

THEOREM 15 LOOSENESS. *Given a constraint network with domains that are of size at most d and constraints that are m-loose and of arity r, $r \geq 2$. It is strongly k-consistent, where k is the maximum value such that*

$$binomial(k-1, r-1) \leq \lceil d/(d-m) \rceil - 1.$$

Proof. Let $Y = \{x_1, x_2, \cdots, x_{K-1}\}$ be a set of any $K-1$ variables where $K \leq k$, $\bar{a}$ a consistent instantiation of the variables in Y, and x_K be any new variable. Let l be the number of relevant constraints to x_K. It can be shown that (see [van Beek and Dechter 1997])

$$l \leq binomial(K-1, r-1) \leq binomial(k-1, r-1) \leq \lceil d/(d-m) \rceil - 1.$$

So, according to Lemma 9, the intersection of extension sets to x_K is not empty. Hence, the constraint network is strongly k-consistent. □

We remark that Theorem 15 is a revised version of the one in [van Beek and Dechter 1997] which may overestimate the level of consistency. For a further discussion on the looseness of constraints and a tighter bound on the inherent level of consistency, see [Zhang and Yap 2003].

8. APPLICATION IV: MAKING WEAKLY TIGHT NETWORKS GLOBALLY CONSISTENT

Consider the weak m-tightness Theorem 13 in Section 6. Generally, a weakly m-tight network may not have the level of local consistency required by the theorem. It is tempting to enforce such a level of consistency on the network to make it globally consistent. However, this procedure may result in constraints with higher arity.

Example 8. Consider a network with variables $\{x, x_1, x_2, x_3\}$. Let the domains of x_1, x_2, x_3 be $\{1, 2, 3\}$, the domain of x be $\{1, 2, 3, 4\}$, and the constraints be that all the variables should take different values:

$$x \neq x_1, x \neq x_2, x \neq x_3, x_1 \neq x_2, x_1 \neq x_3, x_2 \neq x_3.$$

This network is strongly path consistent. In checking the 4-consistency of the network, we know that the instantiation $(1, 2, 3)$ of $\{x_1, x_2, x\}$ is consistent but can not be extended to x_3. To enforce 4-consistency, it is necessary to introduce a constraint on $\{x_1, x_2, x\}$ to make $(1, 2, 3)$ no longer a valid instantiation.

To make the new network globally consistent, the newly introduced constraints with higher arity may in turn require higher local consistency in accordance with Theorem 13. Therefore it is difficult to predict an exact level of consistency (variable based) to *enforce* on the network to make it globally consistent.

In this section, relational consistency will be used to make a constraint network globally consistent.

Definition 13. [van Beek and Dechter 1997] A constraint network is *relationally m-consistent* iff given (1) any m distinct constraints $c_{S_1}, \cdots, c_{S_m}$, and (2) any $x \in \cap_{i=1}^m S_i$, and (3) any consistent instantiation $\bar{a}$ of the variables in $(\cup_{i=1}^m S_i - \{x\})$, there exists an extension of $\bar{a}$ to x such that the extension is consistent with the m relations. A network is *strongly* relationally m-consistent if it is relationally j-consistent for every $j \leq m$.

Variables are no longer of concern in relational consistency. Instead, constraints are the basic unit of consideration. Intuitively, relational m-consistency concerns whether all m constraints agree at every one of their shared variables. It makes sense because different constraints interact with each other exactly through the shared variables.

Relationally 1-, and 2-consistency are also called relationally arc, and path consistency, respectively.

Using relational consistency, it is possible to obtain global consistency by enforcing local consistency on the network. In order to achieve our main result we need a stronger version of m-tightness — *proper* m-tightness.

Definition 14. A constraint c_{S_i} is *properly m-tight* with respect to $x \in S_i$ iff for any instantiation $\bar{a}$ of $S_i - \{x\}$,

$$|E_{i,x}| \leq m.$$

A constraint c_{S_i} is *properly m-tight* iff it is properly m-tight with respect to every $x \in S_i$.

A constraint is m-tight if it is properly m-tight. The converse may not be true. For example, the constraint $x \leq y$, where $x \in \{1, 2, \cdots, 10\}$ and $y \in \{1, 2, \cdots, 10\}$, is 9-tight but not properly 9-tight. It is properly 10-tight since $|E_x(10)| = 10$ when $y = 10$.

A *weakly properly m-tight network* is defined by replacing "m-tight" with "properly m-tight" in definition 11 (section 6).

Definition 15. A constraint network is *weakly properly* m-tight at level k if and only if for every set of variables $\{x_1, \cdots, x_l\} (k \leq l < n)$ and a new variable, there exists a properly m-tight constraint among the relevant constraints after an instantiation of the l variables.

We have the following observation on the weak m-tightness and weak proper m-tightness of a network.

Proposition 16. *A constraint network is weakly properly m-tight (and weakly m-tight respectively) at any level if the constraint between every* two *variables in the network is properly m-tight (m-tight respectively).*

Proof. Consider any level k, any set of variables $Y = \{x_1, x_2, \cdots, x_l\} (k \leq l \leq n)$, and any new variable $x \notin Y$. Since the constraint between any two variable is properly m-tight, the constraint $c_{\{x_1,x\}}$ on x_1 and x is properly m-tight. Therefore, there is a properly m-tight constraint $c_{\{x_1,x\}}$ among the relevant constraints after an instantiation of Y. □

Now we have the main result of this section.

Theorem 17 Weak Proper-Tightness. *Given a constraint network whose constraint on every two variables is properly m-tight, it is globally consistent after it is made relationally $m+1$-consistent.*

Proof. It can be verified that the *proper m-tightness* of the binary constraints is preserved during the procedure to enforce certain level of consistency in the network. So, after enforcing strong relational $m+1$-consistency on the network, it is still weakly properly m-tight. The theorem follows immediately from Theorem 18 in the next section. □

The implication of this theorem is that as long as we have certain properly m-tight constraints on certain combinations of variables, the network can be made globally consistent by enforcing relational $m+1$-consistency.

Remark. Proposition 16 and Theorem 17 assume there is a constraint between every two variables. If there is no constraint between some two variables, a universal constraint is introduced. In this case, we can apply path consistency to the constraint network to make binary constraints tighter so that lower level of relational consistency is sufficient to make the network globally consistent.

9. APPLICATION V: TIGHTNESS AND CONVEXITY REVISITED

All the results on *small set intersection* and *tree convex set intersection* in section 3 can be rephrased in a relational consistency setting. For example, a new version of weak tightness based on relational consistency is given as follows.

Theorem 18 Weak Tightness. *If a constraint network $\mathcal{R}$ of constraints with arity of at most r is strongly relationally $(m+1)$-consistent and weakly m-tight at level of $(m+1)(r-1)+1$, it is globally consistent.*

Proof. Let $j = (m+1)(r-1)+1$. The constraint network $\mathcal{R}$ will be shown to be k-consistent for all k $(j < k \leq n)$.

Let $Y = \{x_1, \cdots, x_{k-1}\}$ a set of any $k-1$ variables, and $\bar{a}$ be an consistent instantiation of all variables in Y. Consider any new variable x_k. Without loss of generality, let $R_{S_1}, \cdots, R_{S_l}$ be the relevant constraints, and E_i be the extension set of $\bar{a}$ to x_k with respect to R_{S_i} for $i \leq l$.

(*Consistency to Set*) Consider any $m+1$ of the l extension sets. Since the $\mathcal{R}$ is relationally $(m+1)$-consistent, the intersection of $m+1$ extension sets is not empty.

(*Set to Set*) The network is weakly m-tight. So, there must be an m-tight constraint in the relevant constraints $R_{S_1}, \cdots, R_{S_l}$. Let it be R_{S_i}. We know its extension set $|E_i| \leq m$. For every $m+1$ of the extension sets have a non-empty intersection, all l extension sets share a common element in terms of the *small set intersection lemma* (Lemma 6).

(*Set to Consistency*) From the lifting lemma, we have that $\mathcal{R}$ is k-consistent. □

Compared with the weak tightness theorem in the previous section, the exposition of the result is neater and the proof is simpler.

For completeness, we also include here a new version of the tree convex theorem using relational consistency. The proof is omitted since it is a simplified version of the one in Section 5 as hinted by the proof above.

THEOREM 19 TREE CONVEXITY. *Let $\mathcal{R}$ be a tree convex constraint network. $\mathcal{R}$ is globally consistent if it is strongly relationally path consistent.*

10. DISCUSSION

The lifting lemma and proof schema proposed in this paper allows us to study consistency in constraint networks through properties of set intersection. We have demonstrated how to infer properties of consistency on a network purely by making use of set intersection properties. There are a few advantages for this approach.

Firstly, although this approach does not offer "new" way to prove consistency results, it does provide a uniform way to understand many seemingly different results on the impact of convexity and tightness on global consistency. In addition to the results shown here, some other results can also be obtained easily by the lifting lemma and proof schema. For example, the work of David [1993] can be obtained by lifting the singleton set Corollary 8. The work of Faltings and Sam-Haroud [1996] is on convex constraint networks in continuous domains and the idea there is to lift Helly's theorem [Eckhoff 1993] on intersection of convex sets in Euclidean spaces.

Furthermore, this approach singles out the fact that set intersection properties play a fundamental role in determining the consistency of a constraint network. With the help of this perspective, we identify a number of new consistency results which we believe are significant progress to convexity and tightness of constraints since van Beek and Dechter's work [1995; 1997]. We identify a new class of tree convex constraints which is a generalization of row convex constraints [van Beek and Dechter 1995]. In a network of tree convex constraints, global consistency is ensured by a certain level of local consistency. We also show that *in a network of arbitrary constraints, local consistency implies global consistency whenever there are m-tight constraints on certain variables* (e.g. Theorem 13). However, when the network does not have the required local consistency, global consistency may not be simply obtained by enforcing such a level of local consistency. A surprising result is that as long as the constraint between every pair of variables is *properly* m-tight in an arbitrary network, global consistency can be achieved by enforcing a certain level of *relational* consistency (Theorem 17). In previous work (e.g. [van Beek and Dechter 1997]), all constraints are required to be m-tight which may be violated by newly introduced constraints in the process of enforcing the intended relational consistency.

We would like to emphasize that the proof schema greatly ease the study of the consistency of a network. In the case of tightness, from the point view of set intersetion, the tight set intersection lemma is technically a straightforward extension of the tight sets intersection lemma (an essential but not explicitly expressed middle result in [van Beek and Dechter 1997]). This small improvement on set intersection result has resulted in significant improvement on consistency results.

Secondly, the establishment of the relationship between set intersection and consistency in a constraint network does not only make it easier for researchers outside the constraint network community to understand the existing consistency results, but also makes it possible for them to contribute to consistency results by exploiting their knowledge on set intersection properties.

In addition to k-consistency and relational consistency, there are other types of consistency like directional consistency [Dechter and van Beek 1997] and adaptive consistency [Dechter and Pearl 1987] where the proof schema are equally applicable. Under these types of consistency, the conditions of many theorems in this paper can be further relaxed and thus the results are more effective in practice since they need less computation and require weaker properties on the constraint network. More work on tightness of constraints can be found in [Zhang 2004].

In the future, we plan to explore existing properties on set intersection and hope to derive from them useful consistency results.

Acknowledgment

We are indebted to Dr. Peter van Beek for valuable discussions. This work is partially supported by a grant under the Academic Research Fund, National Univ. of Singapore and Science Foundation Ireland under Grant 00/PI.1/C075.

REFERENCES

David, P. 1993. When functional and bijective constraints make a CSP polynomial. In *Proceedings of Thirteenth International Joint Conference on Artificial Intelligence.* Vol. 1. Chambery, France, 224–229.

Dechter, R. 1992. From local to global consistency. *Artificial Intelligence 55*, 87–107.

Dechter, R. 2003. *Constraint Processing.* Morgan Kaufmann, San Francisco, CA.

Dechter, R. and Pearl, J. 1987. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence 34*, 1–38.

Dechter, R. and van Beek, P. 1997. Local and global relational consistency. *Theoretical Computer Science 173*, 283–308.

Eckhoff, J. 1993. Helly, Radon, and Carathéodory type theorems. In *Handbook of Convex Geometry*, P. M. Gruber and J. M. Wills, Eds. North Holland, Amsterdam, 389–448.

Freuder, E. 1978. Synthesizing constraint expressions. *Communications of ACM 21,* 11, 958–966.

Freuder, E. 1982. A sufficient condition for backtrack-free search. *Journal of The ACM 29,* 1, 24–32.

Jeavons, P. G., Cohen, D. A., and Gyssens, M. 1997. Closure properties of constraints. *Journal of The ACM 44,* 4, 527–548.

Mackworth, A. K. 1977. Consistency in networks of relations. *Artificial Intelligence 8,* 1, 118–126.

Montanari, U. 1974. Networks of constraints: Fundamental properties and applications. *Information Science 7,* 2, 95–132.

SAM-HAROUD, D. AND FALTINGS, B. V. 1996. Solving non-binary convex CSPs in continous domains. In *Proceedings of International Conference on Principles and Practice of Constraint Programming*. Cambridge, Massachusetts, 410–424.

SCHAEFER, T. J. 1978. The complexity of satisfiability problems. In *Proceedings of 10th ACM Symposium on the Theory of Computing*. 216–226.

VAN BEEK, P. AND DECHTER, R. 1995. On the minimality and global consistency of row-convex constraint networks. *Journal of The ACM 42,* 3, 543–561.

VAN BEEK, P. AND DECHTER, R. 1997. Constraint tightness and looseness versus local and global consistency. *Journal of The ACM 44,* 4, 549–566.

YOSIPHON, G. 2003. Efficient algorithm for identifying tree convex constraints. Manuscript.

ZHANG, Y. 2004. On the tightness of constraints. To appear in Proceedings of CP-04.

ZHANG, Y. AND FREUDER, E. C. 2004. Tractable tree convex constraints. To appear in Proceedings of AAAI-04.

ZHANG, Y. AND YAP, R. H. C. 2003. Erratum: P. van Beek and R. Dechter's theorem on constraint looseness and local consistency. *Journal of The ACM 50,* 3, 1–3.

...