On the Consistency of General Constraint-Satisfaction Problems

Philippe Jégou
L.I.U.P. - Université de Provence
3,Place Victor Hugo
F13331 Marseille cedex 3, France
jegou@gyptis.univ-mrs.fr

Abstract

The problem of checking for consistency of Constraint-Satisfaction Problems (CSPs) is a fundamental problem in the field of constraint-based reasonning. Moreover, it is a hard problem since satisfiability of CSPs belongs to the class of NP-complete problems. So, in (Freuder 1982), Freuder gave theoretical results concerning consistency of binary CSPs (two variables per constraints). In this paper, we proposed an extension to these results to general CSP (n-ary constraints). On one hand, we define a partial consistency well adjusted to general CSPs called *hyper-k-consistency*. On the other hand, we proposed a measure of the connectivity of hypergraphs called *width of hypergraphs*. Using width of hypergraphs and hyper-k-consistency, we derive a theorem defining a sufficient condition for consistency of general CSPs.

Introduction

Constraint-satisfaction problems (CSPs) involve the assignment of values to variables which are subject to a set of constraints. Examples of CSPs are map coloring, conjunctive queries in a relational databases, line drawings understanding, pattern matching in production rules systems, combinatorial puzzles... In the general case checking for the satisfiability (i.e. consistency) of a CSP is a NP-complete problem. A well known method for solving CSP is the Backtrack procedure. The complexity of this procedure is exponential in the size of the CSP, and consequently, this approach frequently induces "combinatorial explosion". So, many works try to improve the search efficiency. Three important classes of methods has been proposed:

1. Improving Backtrack search: eg. dependency-directed backtracking, Forward Checking (Haralick & Elliot 1980), etc.
2. Improving representation of the problem before search: eg. technics of achieving local consistencies using arc-consistent filtering (Mohr & Henderson 1986).
3. Decomposition methods: these technics are based on an analysis of topological features of a the constraint network related to a given CSP; these methods have generally better complexity upper bound than Backtrack methods.

The two first classes of methods do not improved theoretical complexity of solving CSP, but give on many problems, good practical results. The methods of the third class are based on theoretical results due to Freuder (Freuder 1982) (eg. the cycle-cutset method (Dechter 1990)) or research in the field of relational databases theory (Beeri et al. 1983) (eg. tree-clustering (Dechter & Pearl 1989)). These theoretical results associate a structural property of a given constraint network (eg. an acyclic network) to a semantic property related to a partial consistency (eg. arc-consistency). These two properties permit to derive a theorem concerning global consistency of the C-SP and its tractability. Intuitivly, more the network is connected, more the CSP must satisfies a large consistency, and consequently, more the problem is hard to solve. These theoretical results have two practical benefits: on one hand, to define polynomial classes of CSPs, and on the other hand, to elaborate decomposition methods.

In this paper, we propose a theoretical result that is a generalization of the results given in (Freuder 1982) and in relational databases theory (Beeri et al. 1983). Indeed, the theorem given by Freuder concerns binary CSPs (only two variables per constraint), and so this limitation induces practical problems to its application. On the contrary, the property given in the field of relational databases concerns n-ary CSPs (no limitation to the number of variables per constraint), but only CSPs with no cycle. The theorem given in this paper concerns binary and n-ary CSPs, and cyclic constraint networks. It permits to define a sufficient condition to global consistency of general CSPs. This property associates a structural measure of the connectivity of the network, called *width of hypergraph* (in the spirit of Freuder), to a semantic property of CSPs related to partial consistency of n-ary CSPs, that is called *hyper-k-consitency*.

It is known that any non-binary CSP can be treated as a binary CSP if one look at the dual representation or join-graph (this representation has been defined in the field of relational databases: constraints are variables and binary constraints impose equality on common variables). But this approach if of limited interest: it does not allow to realize extension of all theorems and algorithms to non-binary CSPs. For example, the width of an n-ary CSP cannot be defined exactly as the width of its join graph (see example in figure 3). So, original definitions are introduced in this paper.

The second section presents definitions and preliminaries. In the third section, we defined hyper-k-consitency while in next section we introduce the notion of width of hypergraphs. The last section exposes the consistency theorem and give comments about its usability.

Definitions and preliminaries

Finite Constraint-Satisfaction Problems

A *General Constraint-Satisfaction Problem* involves a set X of n variables $X_1, X_2, \ldots X_n$, each defined by its finite domain values $D_1, D_2, \ldots D_n$ (d denotes the maximum cardinality over all the D_i). D is the set of all domains. C is the set of constraints $C_1, C_2, \ldots C_m$. A constraint C_i is defined as a set of variables $(X_{i_1}, X_{i_2}, \ldots X_{i_{j_i}})$. To any constraint C_i, we associate a subset of the cartesian product $D_{i_1} \times \ldots \times D_{i_{j_i}}$ that is denoted R_i (R_i specifies which values of the variables are compatible with each other; R_i is a relation, so it is a set of tuples). R is the set of all R_i. So, we denote a CSP $\mathcal{P} = (X, D, C, R)$. A solution is an assignment of value to all variables satisfying all the constraints.
Given a CSP $\mathcal{P} = (X, D, C, R)$, the hypergraph (X, C) is called the constraint hypergraph (nodes are variables and hyper-edges are defined by constraints). A binary CSP is one in which all the constraints are binary, i.e. only pairs of variables are possible, so (X, C) is a graph called constraint graph. For a given CSP, the problem is either to find all solutions or one solution, or to know if there exists any solution. The decision problem (existence of solution) is known to be NP-complete. We use two relationnal operators. *Projection of relations*: if $X' \subseteq C_i$, the projection of R_i on X' is denoted $R_i[X']$ and *join of relations* denoted $R_i \bowtie R_j$; see formal definitions in (Maier 1983).

Partial consistencies in CSPs

Different levels of consistency have been introduced in the field of CSPs. The methods to achieve these local consistencies are considered as filtering algorithms: they may lead to problem simplifications, without changing the solution set. They have been used as well to improve the representation prior the search, as to avoid backtrack during the search (Haralick & Elliot 1980). Historically, the first partial consistency proposed was *arc-consistency*. Its generalization was given in (Freuder 1978).

Definition 1 (Freuder 1978). A CSP is *k-consistent* iff for all set of $k-1$ variables, and all consistent assignments of these variables (that satisfy all the constraints among them), for all k^{th} variable X_k, there exists a value in the domain D_k that satisfies all the constraints among the k variables. A CSP is ***strongly k-consistent*** iff the CSP is j-consistent for $j = 1, \ldots k$.

Given a CSP and a value k, the complexity of the algorithm achieving k-consistencyis $O(n^k d^k)$ (Cooper 1989). But achieving k-consistency on a binary CSP generally induces new constraints, with arity equal to $k-1$. Consequently, a binary CSP can be tranformed in an n-ary CSP using this method (eg. achieving 4-consistency).
An other partial-consistency has been defined particularly for n-ary CSPs: the *pairwise-consistency* (Janssen et al. 1989) also called *inter-consistency* (Jégou 1991). This consistency is based on works concerning relational databases (Beeri et al. 1983). Whereas k-consistency is a local consistency between variables, domains and constraints, inter-consistency defines a consistency between constraints and relations. On the contrary of k-consitency, that does not consider structural features of the constraint network, inter-consistency is particularly adjusted to the connections in n-ary CSPs, because connections correspond to intersections between constraints.

Definition 2 (Beeri et al. 1983)(Janssen et al. 1989). We said that $\mathcal{P} = (X, D, C, R)$ is *inter-consistent* iff $\forall C_i, \forall C_j, R_i[C_i \cap C_j] = R_j[C_i \cap C_j]$ and $\forall R_i, R_i \neq \emptyset$.

In (Janssen et al. 1989), a polynomial algorithm achieving this consistency is given. This algorithm is based on an equivalent binary representation given in the next section.

Binary representation for n-ary CSPs

In this representation, the vertices of the constraint graph are n-ary constraints C_i, their domains are the associated relations R_i, and the edges, that are new constraints, are given by intersections between C_i. The compatibility relations are then given by the equality constraints between the connected R_i. This binary representation is called the *constraint intergraph associated to a constraint hypergraph* (Jégou 1991).

Definition 3. A *hypergraph H* is a pair (X, C) where X is a finite set of vertices and C a set of hyper-edges, i.e. subsets of X. When the cardinality of any hyper-edges is two, the hypergraph is a graph (necessary undirected). Given a CSP (X, D, C, R), we consider its associated hypergraph denoted (X, C).

Definition 4 (Bernstein & Goodman 1981). Given a hypergraph $H = (X, C)$, an *intergraph* of H is a graph $G(H) = (C, E)$ such as:

- $E \subseteq \{\{C_i, C_j\} \subset C / i \neq j$ and $C_i \cap C_j \neq \emptyset\}$
- $\forall C_i, C_j \in C$, if $C_i \cap C_j \neq \emptyset$, there is a chain $(C_i = C_1, C_2, \ldots C_q = C_j)$ in $G(H)$ such as $\forall k, 1 \leq k < q, C_i \cap C_j \subseteq C_k \cap C_{k+1}$

Intergraphs are also called *line-graphs*, *join-graphs* (Maier 1983) and *dual-graphs* (Dechter & Pearl 1989).

Definition 5 (Jégou 1991). Given a CSP (X, D, C, R), we defined an equivalent (equivalent sets of solutions) binary CSP (C, R, E, Q):

- (C, E) is an intergraph of the hypergraph (X, C).
- $C = \{C_1, \ldots C_m\}$ is a set of variables defined on domains $R = \{R_1, \ldots R_m\}$.
- if $\{C_i, C_j\} \in E$, then we have an equality constraint: $Q_k = \{(r_i, r_j) \in R_i \times R_j / r_i[C_i \cap C_j] = r_j[C_i \cap C_j]\}$

Given a hypergraph, it can exists several associated intergraphs. Some of them can contain redundant edges that can be deleted to obtain an other intergraph. The maximal one is called *representative graph*. In the field of CSPs, we are naturally interested with minimal intergraphs: all edges are necessary, i.e. no edge can be deleted conserving the property of chains in intergraphs. So an algorithms have been proposed to find minimal intergraphs in (Janssen et al. 1989). A study of combinatorial properties of minimal intergraphs is given in (Jégou & Vilarem 1993). In the next example two minimal intergraphs are given:

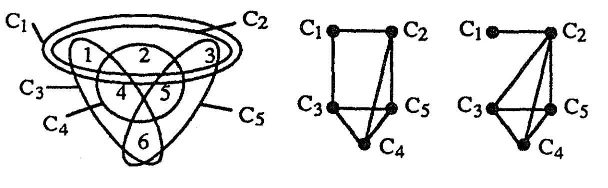

Figure 1. Hypergraph (a) and two minimal intergraphs.

A sufficient condition for CSPs consistency

Freuder has identified sufficient conditions for a binary CSP to satisfy consistency, ie. satisfiability. These conditions associate topology of the constraint graph with partial consistency.

Definition 6 (Freuder 1982). An *ordered constraint graph* is a constraint graph in which nodes are linearly ordered. The *width of a node* is the number of edges that link that node to previous nodes. The *width of an order* is the maximum width of all nodes. The *width of a graph* is the minimum width of all orderings of that graph.

This definition is illustrated in figure 1: the width of the graph (b) is 2 and the width of the graph (c) is 3. On this example, we can remark that the width of an hypergraph cannot be defined as the width of its minimal intergraph, because all minimal intergraphs has not the same width.

Theorem 7 (Freuder 1982). Given a CSP, if the level of strong consistency is greater than the width of the constraint graph, then the CSP is consistent and it is possible to find solutions without backtracking (in polynomial time).

Freuder also gave an algorithm to compute the width of any graph (in $O(n+m)$). So given a CSP, it is sufficient to know the width of the constraint graph, denoted $k-1$, then to achieve k-consistency. But a problem appears: this approach is possible only for acyclic constraint graphs (width equal to one) and a subclass of graphs the width of which is two (called regular graphs of width two in (Dechter & Pearl 1988)). The cause of that problem: achieving k-consistency generally induces n-ary constraints (arity can be equal to $k-1$), so the corresponding problem is a constraint hypergraph, and the theorem can not be applied. Nevertheless, the result concerning acyclic CSPs is applied in the cycle-cutset method (Dechter 90) and we can consider Freuder's theorem as a vehicle to give a lower bound for complexity of a binary CSP: to the order of d^k if its width is $k-1$, because complexity of achieving k-consistency is $O(n^k d^k)$ (Cooper 1989).

A result of relational database theory

A similar property has been derived in the field of relational databases. This property is related to acyclic hypergraphs.

Definition 8 (Beeri et al. 1983). A hypergraph is acyclic iff $\exists$ a linear order $(C_1, C_2, \ldots C_m)$ such as $\forall i, 1 < i \leq m. \exists j_i < i / (C_i \bigcap \cup_{k=1}^{i-1} C_k) \subseteq C_{j_i}$ (this property is called *running intersection property*)

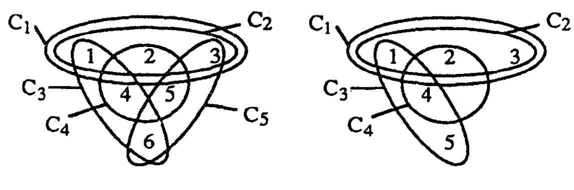

Figure 2. Cyclic (a) and acyclic (b) hypergraphs.

(Beeri et al. 1983) gave a fundamental property of acyclic database schemes that concerns consistency of such databases, namely *global consistency*. This result is presented below using CSPs terminology:

Definition 9. Let $\mathcal{P} = (X, D, C, R)$ be a CSP. We say that $\mathcal{P}$ is *globally consistent* if there is a relation S over the variables X (S is the set of solutions) such as $\forall i, 1 \leq i \leq m, R_i = S[C_i]$. It is equivalent to $\forall i, 1 \leq i \leq m, (\bowtie_{j=1}^{m} R_j)[C_i] = R_i$ since $S = (\bowtie_{j=1}^{m} R_j)$.

Note that global consistency of CSPs implies satisfiability of CSPa. Indeed, a CSP is globally consistent iff every tuple of relations appears at least in one solution. Furthermore, it is clear that global consistency implies inter-consistency but the converse is false. We give the interpretation in the field of CSP to the property given by (Beeri et al. 1983):

Theorem 10 (Beeri et al. 1983). If $\mathcal{P}$ is such as its constraints hypergraph is acyclic, then

$\mathcal{P}$ is inter-consistent $\Leftrightarrow \mathcal{P}$ is globally consistent.

An immediate application of this theorem concerns the consistency checking of CSPs. We know polynomial algorithms to achieve inter-consistency while to check global consistency is a NP-complete problem. So, knowing this theorem, if the database scheme is acyclic, it is possible to check global consistency in a polynomial time achieving inter-consistency. This result is applied in the tree-clustering method (Dechter & Pearl 1989).

Some remarks

Theorems 7 and 10 are significant. First, they can be used to solve CSP, immediatly if the considered CSP is acyclic, or for all CSP, using decomposition methods as the cycle-cutset method, or tree-clustering scheme. Second, because they define polynomial subclasses of CSPs. Nevertheless, there is two significant limitations to these theoretical results. On one hand, theorem 7 is only defined on binary CSPs, and so, it can not be applied, nor on n-ary CSPs, nor on constraints graph with width greater than 2. On the other hand, theorem 11 concerns only acyclic n-ary CSPs. So, a generalization to cyclic n-ary CSPs is necessary to extend this kind of theoretical approach to all CSPs.

A new consistency for n-ary CSPs: Hyper-k-consistency

When Freuder defined k-consistency, the definition is related to assignments: "given a consistent assignment of variables $X_1, X_2, \ldots X_{k-1}$, it is possible to extend this assignment for all k^{th} variable". To generalize k-consistency to n-ary CSPs, we consider the same approach but with constraints and relations: we can consider "assignment" of constraints $C_1, C_2, \ldots C_{k-1}$, and their extension to any k^{th} constraint. Our definition of hyper-k-consistency is given in this spirit:

Definition 11. A CSP $\mathcal{P} = (X, D, C, R)$ is *hyper-k-consistent* iff $\forall R_i, R_i \neq \emptyset$ and $\forall C_1, C_2, \ldots C_{k-1} \in C$,

$$(\bowtie_{i=1}^{k-1} R_i)[(\cup_{i=1}^{k-1} C_i) \cap C_k] \subseteq R_k[(\cup_{i=1}^{k-1} C_i) \cap C_k]$$

$\mathcal{P}$ is *strongly hyper-k-consistent* iff $\forall i, 1 \leq i \leq k$, $\mathcal{P}$ is hyper-i-consistent.

We can note that hyper-2-consistency is equivalent to inter-consistency. So, hyper-k-consistency constitutes really a generalization of inter-consistency to greater levels. Actually, this definition can be considered as a formulation of k-consistency on the constraint intergraph: $(\bowtie_{i=1}^{k-1} r_i) \in (\bowtie_{i=1}^{k-1} R_i)$ signifies that $(r_1, r_2, \ldots r_{k-1})$ is a consistent assignment of constraints $C_1, C_2, \ldots C_{k-1}$, and if there is $r_k \in R_k$ such as:

$$(\bowtie_{i=1}^{k-1} r_i)[(\cup_{i=1}^{k-1} C_i) \cap C_k] = r_k[(\cup_{i=1}^{k-1} C_i) \cap C_k]$$

then $(r_1, r_2, \ldots r_{k-1}, r_k)$ is a consistent assignment of constraints $C_1, C_2, \ldots C_{k-1}, C_k$, i.e. k variables of the constraint intergraph. This particularity induces a method to achieve hyper-k-consitency, that is based on the same approach of achieving k-consistency on binary CSPs. So, we have the same kind of problems: achieving hyper-k-consistency can modified constraint hypergraph. A second problem concerns the complexity of achieving hyper-k-consistency: the complexity is in the order of r^k if r is the maximum size of R_i. These problems are discussed in (Jégou 1991).
Another remark about hyper-k-consistency concerns its links with global consistency; we easily verify that if a CSP $\mathcal{P}$ is hyper-m-consistent, then $\mathcal{P}$ is globally consistent while the converse is generally false.

The definition of hyper-k-consistency in n-ary CSP concerns connections in hypergraphs, i.e. intersections between hyper-edges. So the definition of width of hypergraph is based on the same principles.

Width of Hypergraphs

Connections in hypergraphs concerns intersections between hyper-edges. So, the consistency of a n-ary CSP is intimately connected to the intersections between hyper-edges. The definition of width of hypergraph allows us to define a degree of cyclicity of hypergraphs. Using this width, we shall work out links between structural properties of hypergraphs and the global consistency of n-ary CSPs.

Before to give our definition of the width of an hypergraph, we must explain why this definition is not immediatly related to intergraphs. A first reason has already been given: all minimal intergraphs of a hypergraph have not necessary the same width (see figure 1). An other reason is the next one: if we define the width of an hypergraph as the width of one of its intergraph (not necessary a minimal one), we cannot obtain the same properties than we have with Freuder's theorem that is based on a good order for the assignment of the variables:

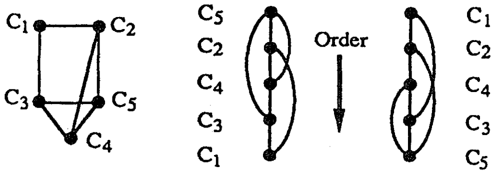

Figure 3. Problem of order on variables.

The hypergraph considered in the figure 4 is the one of the figure 1. We consider two possible orders on this intergraph. The first one is not a possible order for the assignment of the variables. Indeed, when the variable corresponding to C_3 is assigned, the variable 1 of the hypergraph has already been assigned and so, it is possible to assign C_3 with an other value for 1. This is possible because C_2 is given before C_3 in the order and because there is no edge between C_2 and C_3. So, we can obtain two different assignments for the variable 1, and finally, we can obtain a consistent assignment on $C_1, C_2, \ldots C_5$, and consequently an assignment on variables $X_1, X_2, \ldots X_6$ that is not a solution of the problem. The second order of width 3 does not induce such problems.

Definition 12. Given a hypergraph $H = (X, C)$, $\mathcal{O}$ the set of linear orders on C, and a linear order $\tau = (C_1, \ldots C_m) \in \mathcal{O}$:

- the *width of C_i in order τ on H* is the number of maximal intersections with predecessors of C_i in τ; $\mathcal{L}_\tau(C_i)$ denotes the width of C_i in order τ: $\mathcal{L}_\tau(C_i) = \mid \{C_i \cap C_j / j < i \wedge \neg\exists k, k < i / C_i \cap C_j \subset\neq C_i \cap C_k\} \mid$
- the *width of τ* is $\mathcal{L}_H(\tau) = max\ \{\mathcal{L}_\tau(C_i)/C_i \in C\}$
- the *width of H* is $\mathcal{L}(H) = min\ \{\mathcal{L}_H(\tau)/\tau \in \mathcal{O}\}$

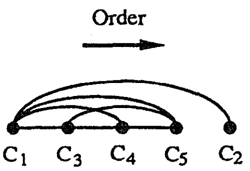

Figure 4. Width of the hypergraph.

In the fugure 4, we see the width of the hypergraph H given in figure 1-a. Here, $\mathcal{L}_H(\tau) = 3$ since $\mathcal{L}_\tau(\{3, 5, 6\}) = 3$; we can verify that $\forall \tau \in \mathcal{O}, \mathcal{L}_H(\tau) = 3$, and consequently that $\mathcal{L}(H) = 3$. A property relies width and cyclicity:

Proposition 13. *H is acyclic $\Leftrightarrow \mathcal{L}(H) = 1$*
Proof. *H acyclic satisfies the running intersection property*
$\Leftrightarrow \exists order\ (C_1, C_2, \ldots C_m)$ *such as*
$\forall i, 1 < i \leq m, \exists j_i < i / (C_i \bigcap (\cup_{k=1}^{i-1} C_k)\ ) \subseteq C_j$,
$\Leftrightarrow \exists order\ (C_1, C_2, \ldots C_m)$ *such as* $\forall i, 1 < i \leq m$,
$\mid \{C_i \cap C_j / j < i \wedge \neg\exists k, k < i / C_i \cap C_j \subset\neq C_i \cap C_k\} \mid \leq 1$
$\Leftrightarrow \exists order\ (C_1, C_2, \ldots C_m) / \mathcal{L}_H(\tau) \leq 1 \Leftrightarrow \mathcal{L}(H) \leq 1$.
Moreover, it is clear that if H is connected and if C possesses more than one hyper-edge, the inequality $\mathcal{L}(H) \leq 1$ necessary holds.

Given a hypergraph (X, C) and an order on C, it is not hard to find the width of this order. It is just necessary to compute for each C_i the number of maximal intersections with predecessors, and to select the greater. On the contrary, finding an order that give the minimal width of a hypergraph is an optimization problem; this problem seems us to be open concerning its complexity: does it belong to NP-hard problems ? This question is at present an open question. In (Jégou 1991), an heuristic is proposed to find small width, ie. just an approximation of the width of a given hypergraph.

Consistency Theorem

In this section, we derive a sufficient condition to consistency of general CSPs. This condition concerns the width of the hypergraph associated to the CSP, and the hyper-k-consistency that the relations satisfy (i.e. the value k).

Theorem 14. Let $\mathcal{P} = (X, D, C, R)$ be a CSP and $H = (X, C)$. If $\mathcal{P}$ is strongly hyper-k-consistent and if $\mathcal{L}(H) \leq k - 1$, then $\mathcal{P}$ is consistent.
Proof. *We must show that $\mathcal{P}$ is consistent, i.e. $\forall i, 1 \leq i \leq m, \exists r_i \in R_i / (\bowtie_{i=1}^{m} r_i) \in (\bowtie_{i=1}^{m} R_i)$. That is $(r_1, r_2, \ldots r_m)$ can be considered as a consistent assignment on $(C_1, C_2, \ldots C_m)$: $\forall i, j, 1 \leq i, j \leq m, r_i[C_i \cap C_j] = r_j[C_i \cap C_j]$.*

We proove this property by induction on p, such as $1 \leq p \leq m$.

If $p = 1$, the property trivially holds.

We consider now a linear order $(C_1, C_2, \ldots C_m)$ associated to the width $\mathcal{L}(H) \leq k - 1$. Suppose that the property holds for $p - 1$ such as $1 < p \leq m$. That is we have $(r_1, r_2, \ldots r_{p-1}) / \forall i, 1 \leq i \leq p - 1, r_i \in R_i$ and $\forall i, j, 1 \leq i, j \leq p - 1, r_i[C_i \cap C_j] = r_j[C_i \cap C_j]$.

By definition of the width, C_p possesses at most $k - 1$ maximal intersections with predecessors in the order. Let $C_{i_1}, C_{i_2}, \ldots C_{i_q}$ be the corresponding C_i, with necessary $q \leq k - 1$, considering only one C_i for every maximal intersection. $\mathcal{P}$ being strongly hyper-k-consistent, and since $q \leq k - 1$, we have

$$(\bowtie_{j=1}^{q} R_{i_j})[(\cup_{j=1}^{q} C_{i_j}) \cap C_p] \subseteq R_p[(\cup_{j=1}^{q} C_{i_j}) \cap C_p]$$

and for the r_{i_j}'s appearing in $(r_1, r_2, \ldots r_{p-1})$

$$(\bowtie_{j=1}^{q} r_{i_j})[(\cup_{j=1}^{q} C_{i_j}) \cap C_p] \in R_p[(\cup_{j=1}^{q} C_{i_j}) \cap C_p]$$

So, $\exists r_p \in R_p$ such as r_p is consistent with $(r_{i_1}, r_{i_2}, \ldots r_{i_q})$, that is: $\forall j, 1 \leq j \leq q, r_{i_j}[C_{i_j} \cap C_p] = r_p[C_{i_j} \cap C_p]$.

We show now, that r_p is also consistent for all the r_i's, i.e. $r_i[C_i \cap C_p] = r_p[C_i \cap C_p]$.

Consider C_i such as $1 \leq i < p$ and $C_i \cap C_p \neq \emptyset$. By the definition of the width, $\exists j, 1 \leq j \leq q$ such as $C_i \cap C_p \subseteq C_{i_j} \cap C_p$ because the C_{i_j}'s are maximal for the intersection with C_p. Consequently, we have $C_i \cap C_p \subseteq C_i \cap C_{i_j}$.

By hypothesis, we have $r_i[C_i \cap C_{i_j}] = r_{i_j}[C_i \cap C_{i_j}]$, a fortiori, we have $r_i[C_i \cap C_p] = r_{i_j}[C_i \cap C_p]$. We seen that $r_{i_j}[C_{i_j} \cap C_p] = r_p[C_{i_j} \cap C_p]$; this emplies that $r_{i_j}[C_i \cap C_p] = r_p[C_i \cap C_p]$. Consequently, $r_i[C_i \cap C_p] = r_{i_j}[C_i \cap C_p] = r_p[C_i \cap C_p]$. So, $(r_1, r_2, \ldots r_{p-1}, r_p)$ is consistent assignment on $(C_1, C_2, \ldots C_{p-1}, C_p)$.

So the property holds for p, and consequently, $\mathcal{P}$ is consistent.

If we recall the property given about acyclic database schemes (Beeri et al. 1983), it is clear that theorem 10 is a corollary of theorem 14 (because k = 2). A more interesting result is the next corollary:

Corollary 15. Let $\mathcal{P} = (X, D, C, R)$ be a CSP such as (X, C) is a graph (i.e. all hyper-edges have cardinality 2). If $\mathcal{P}$ is strongly hyper-3-consistent, then $\mathcal{P}$ is consistent.
Proof. *It is sufficient to remark that if (X, C) is a graph, its width is at most 2, because all hyper-edges of (X, C) are edges, and an edge has no more than 2 maximal intersections.*

Nevertheless, this surprising corollary is not allways usable, because achieving hyper-k-consistency can modify the hypergraph associated to a n-ary CSP. Freuder's theorem has the same kind of problems: try to obtain its preconditions can modify these preconditions. So, concerning the practical use of the theorem, a problem is given by the verification of hyper-k-consistency in a CSP. On one hand, the theorem gives a sufficient condition to consistency, and not a necessary condition; on the other hand, given a value k, it is possible to obtain hyper-k-consistency using filtering mecanisms (Jégou 1991) in polynomial time in k, in the size of the CSP. But this process can modify the hypergraph with additions of new hyper-edges, and so modify the width. Nevertheless, contrary to Freuder's theorem, the consistency theorem can be tried to apply after modification of the width because it is directly defined on n-ary CSPs.
Consequently, the theorem must be considered in a first time as a theoretical result, with, at this moment, only one practical application: the corollary given in (Beeri et al. 1983), and not as a directly usable result.
The next research must be to exploit the theorem, on one hand to try to find new polynomial classes of C-SP, and on the other hand to propose new methods to solve practically n-ary CSPs.

References

Beeri, C., Fagin, R., Maier, D. and Yannakakis, M. 1983. On the desirability of acyclic database schemes. *Journal of the ACM* 30:479-513.

Dechter, R. 1990. Enhancement Schemes fo Constraint Processing: Backjumping, Learning and Cutset Decomposition. *Artificial Intelligence* 41:273-312.

Dechter, R. and Pearl, J. 1988. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence* 34:1-38.

Dechter, R. and Pearl, J. 1989. Tree Clustering for Constraint Networks. *Artificial Intelligence* 38:353-366.

Freuder, E.C. 1978. Synthesizing constraint expressions. *Communications of the ACM* 21:958-967.

Freuder, E.C. 1982. A sufficient condition for backtrack-free search. *Journal of the ACM* 29(1):24-32.

Haralick, R.M. and Elliot, G.L. 1980. Increasing tree search efficiency for constraint-satisfaction problems. *Artificial Intelligence* 14:263-313.

Janssen, P., Jégou, P., Nouguier, B. and Vilarem, M.C. 1989. A filtering process for general constraint satisfaction problems: achieving pairwise-consistency using an associated binary representation. In Proceedings of the IEEE Workshop on Tools for Artificial Intelligence, 420-427. Fairfax, USA

Jégou, P. 1991. Contribution à l'étude des probèmes de satisfaction de contraintes... Thèse de Doctorat, Université Montpellier II, France.

Jégou, P. and Vilarem, M.C. 1993. On some partial line graphs of a hypergraph and the associated matroid. *Discrete Mathematics.* To appear.

Maier, D. 1983. The Theory of Relational Databases. Computing Science Press.

Mohr, R. and Henderson, T.C. 1986. Arc and path consistency revisited. *Artificial Intelligence* 28(2):225-233.