View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by HAL-Artois

HAL
archives-ouvertes.fr

A Framework for Decision-based Consistencies

Jean-Francois Condotta, Christophe Lecoutre

▶ To cite this version:

Jean-Francois Condotta, Christophe Lecoutre. A Framework for Decision-based Consistencies. 17th International Conference on Principles and Practice of Constraint Programming (CP'11), 2011, Perugla, Italy. Springer, 6876, pp.172-186, 2011, Lecture Notes in Computer Science (LNCS). <hal-00865549>

HAL Id: hal-00865549

https://hal.archives-ouvertes.fr/hal-00865549

Submitted on 24 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Framework for Decision-based Consistencies

Jean-François Condotta and Christophe Lecoutre

CRIL - CNRS, UMR 8188,
Univ Lille Nord de France, Artois
F-62307 Lens, France
{condotta,lecoutre}@cril.fr

Abstract. Consistencies are properties of constraint networks that can be enforced by appropriate algorithms to reduce the size of the search space to be explored. Recently, many consistencies built upon taking decisions (most often, variable assignments) and stronger than (generalized) arc consistency have been introduced. In this paper, our ambition is to present a clear picture of decision-based consistencies. We identify four general classes (or levels) of decision-based consistencies, denoted by S^{ϕ}_{Δ}, E^{ϕ}_{Δ}, B^{ϕ}_{Δ} and D^{ϕ}_{Δ}, study their relationships, and show that known consistencies are particular cases of these classes. Interestingly, this general framework provides us with a better insight into decision-based consistencies, and allows us to derive many new consistencies that can be directly integrated and compared with other ones.

1 Introduction

Consistencies are properties of constraint networks that can be used to make inferences. Such inferences are useful to filter the search space of problem instances. Most of the current constraint solvers interleave inference and search. Typically, they enforce generalized arc consistency (GAC), or one of its partial form, during the search of a solution. One avenue to make solvers more robust is to enforce strong consistencies, i.e., consistencies stronger than GAC. Whereas GAC corresponds to the strongest form of local reasoning when constraints are treated separately, strong consistencies necessarily involve several constraints (e.g., path inverse consistency [12], max-restricted path consistency [8] and their adaptations [20] to non-binary constraints) or even the entire constraint network (e.g., singleton arc consistency [9]).

A trend that emerges from recent works on strong consistencies is the resort to taking decisions before enforcing a well-known consistency (typically, GAC) and making some deductions. Among such decision-based consistencies, we find SAC (singleton arc consistency), partition-k-AC [2], weak-k-SAC [22], BiSAC [4], and DC (dual consistency) [15]. Besides, a partial form of SAC, better known as shaving, has been introduced for a long time [6, 18] and is still an active subject of research [17, 21]; when shaving systematically concerns the bounds of each variable domain, it is called BoundSAC [16]. What makes decision-based consistencies particularly attractive is that they are (usually) easy to define and understand, and easy to implement since they are mainly based on two concepts (decision, propagation) already handled by constraint solvers. The increased interest perceived in the community for decision-based consistencies has motivated our study.

In this paper, our ambition is to present a clear picture of decision-based consistencies that can derive nogoods of size up to 2; i.e., inconsistent values or inconsistent pairs of values. The only restriction we impose is that decisions correspond to unary constraints. The four classes (or levels) of consistencies, denoted by S^{ϕ}_{Δ}, E^{ϕ}_{Δ}, B^{ϕ}_{Δ} and D^{ϕ}_{Δ}, that we introduce are built on top of a consistency ϕ and a so-called decision mapping Δ. These are quite general because:

1. Δ allows us to introduce a specific set of decisions for every variable x and every possible (sub)domain of x,
2. decisions are membership decisions (of the form $x \in D_x$ where D_x is a set of values taken from the initial domain of x) that generalize both variable assignments (of the form $x = a$) and value refutations (of the form $x \neq a$),
3. decisions may ignore some variables and/or values, and decisions may overlap each other,
4. ϕ is any well-behaved nogood-identifying consistency.

We study the relationships existing between them, including the case where Δ covers every variable and every value. We also show that SAC, partition-k-AC, BiSAC and DC are particular cases of S^{ϕ}_{Δ}, S^{ϕ}_{Δ}+E^{ϕ}_{Δ} (the two consistencies combined), B^{ϕ}_{Δ} and D^{ϕ}_{Δ}, respectively. BoundSAC, and many other forms of shaving, are also elements of the class S^{ϕ}_{Δ}. The general framework we depict provides a better insight into decision-based consistencies while allowing many new combinations and comparisons of such consistencies. For example, the class of consistencies S^{ϕ}_{Δ} induces a complete lattice where the partial order denotes the relative strength of every two consistencies.

2 Technical Background

This section provides technical background about constraint networks and consistencies, mainly taken from [1, 11, 3, 13].

Constraint Networks. A *constraint network* (CN) P is composed of a finite set of n variables, denoted by $vars(P)$, and a finite set of e constraints, denoted by $cons(P)$. Each variable x has a domain which is the finite set of values that can be assigned to x. Each constraint c involves an ordered set of variables, called the *scope* of c and denoted by $scp(c)$, and is defined by a relation which is the set of tuples allowed for the variables involved in c. The initial domain of a variable x is denoted by $dom^{init}(x)$ whereas the current domain of x (in the context of P) is denoted by $dom^P(x)$, or more simply $dom(x)$. Assuming that the initial domain of each variable is totally ordered, $min(x)$ and $max(x)$ will denote the smallest and greatest values in $dom(x)$. The initial and current relations of a constraint c are denoted by $rel^{init}(c)$ and $rel(c)$, respectively.

A constraint is *universal* iff $rel^{init}(c) = \Pi_{x \in scp(c)} dom^{init}(x)$. For simplicity, a pair (x, a) with $x \in vars(P)$ and $a \in dom(x)$ is called a *value* of P, which is denoted by $(x, a) \in P$. A *unary* (resp., *binary*) constraint involves 1 (resp., 2) variable(s), and a *non-binary* one strictly more than 2 variables. Without any loss of generality, we only consider CNs that do not involve unary constraints, universal constraints and constraints of similar scope. The set of such CNs is denoted by $\mathscr{P}$. An *instantiation* I of a set $X = \{x_1, \ldots, x_k\}$ of variables is a set $\{(x_1, a_1), \ldots, (x_k, a_k)\}$ such that $\forall i \in 1..k, a_i \in dom^{init}(x_i)$; X is denoted by $vars(I)$ and each a_i is denoted by $I[x_i]$. An instantiation I *on* a CN P is an instantiation of a set $X \subseteq vars(P)$; it is *complete* if $vars(I) = vars(P)$. I is *valid* on P iff $\forall (x, a) \in I, a \in dom(x)$. I *covers* a constraint c iff $scp(c) \subseteq vars(I)$, and I *satisfies* a constraint c with $scp(c) = \{x_1, \ldots, x_r\}$ iff (i) I covers c and (ii) the tuple $(I[x_1], \ldots, I[x_r]) \in rel(c)$. An instantiation I on a CN P is *locally consistent* iff (i) I is valid on P and (ii) every constraint of P covered by I is satisfied by I. A *solution* of P is a complete locally consistent instantiation on P; $sols(P)$ denotes the set of solutions of P. An instantiation I on a CN P is *globally inconsistent*, or a *nogood*, iff it cannot be extended to a solution of P. Two CNs P and P' are *equivalent* iff $vars(P) = vars(P')$ and $sols(P) = sols(P')$.

The *nogood representation* of a CN is a set of nogoods, one for every value removed from the initial domain of a variable and one for every tuple forbidden by a constraint. More precisely, the nogood representation $\widetilde{x}$ of a variable x is the set $\{\{(x, a)\} \mid a \in \overline{dom}(x)\}$ with $\overline{dom}(x) = dom^{init}(x) \setminus dom(x)$. The nogood representation $\widetilde{c}$ of a constraint c is $\{\{(x_1, a_1), \ldots, (x_r, a_r)\} \mid (a_1, \ldots, a_r) \in \overline{rel}(c)\}$, with $scp(c) = \{x_1, \ldots, x_r\}$ and $\overline{rel}(c) = \Pi_{x \in scp(c)} dom^{init}(x) \setminus rel(c)$. The nogood representation $\widetilde{P}$ of a CN P is $(\cup_{x \in vars(P)} \widetilde{x}) \cup (\cup_{c \in cons(P)} \widetilde{c})$. Based on nogood representations, a general partial order can be introduced to relate CNs. Let P and P' be two CNs such that $vars(P) = vars(P')$, we have $P' \preceq P$ iff $\widetilde{P'} \supseteq \widetilde{P}$ and we have $P' \prec P$ iff $\widetilde{P'} \supsetneq \widetilde{P}$. $(\mathscr{P}, \preceq)$ is the partially ordered set (poset) considered in this paper. The search space of a CN can be reduced by a filtering process (called constraint propagation) based on some properties (called consistencies) that allow us to identify and record explicit nogoods in CNs; e.g., identified nogoods of size 1 correspond to inconsistent values that can be safely removed from variable domains. In $\mathscr{P}$, there is only one manner to discard an instantiation from a given CN, or equivalently to "record" a new explicit nogood. Given a CN P in $\mathscr{P}$, and an instantiation I on P, $P \setminus I$ denotes the CN P' in $\mathscr{P}$ such that $vars(P') = vars(P)$ and $\widetilde{P'} = \widetilde{P} \cup \{I\}$. $P \setminus I$ is an operation that retracts I from P and builds a new CN. If $I = \{(x, a)\}$, we remove a from $dom(x)$. If I corresponds to a tuple allowed by a constraint c of P, we remove this tuple from $rel(c)$. Otherwise, we introduce a new constraint allowing all possible tuples (from initial domains) except the one that corresponds to I.

Consistencies. A consistency is a property defined on CNs. When a consistency ϕ holds on a CN P, we say that P is *ϕ-consistent*; if ψ is another consistency, P is *ϕ+ψ-consistent* iff P is both ϕ-consistent and ψ-consistent. A consistency ϕ is *nogood-identifying* iff the reason why a CN P is not ϕ-consistent is that some instantiations, which are not in $\widetilde{P}$, are identified as globally inconsistent by ϕ; such instantiations are said to be *ϕ-inconsistent*. A *kth-order consistency* is a nogood-identifying consistency that allows the identification of nogoods of size k. A *domain-filtering consistency* [10, 5] is a first-order consistency. A nogood-identifying consistency is *well-behaved* when for any CN P, the set $\{P' \in \mathscr{P} \mid P'$ is ϕ-consistent and $P' \preceq P\}$ admits a greatest element, denoted by $\phi(P)$, equivalent to P. Enforcing ϕ on a CN P means computing $\phi(P)$. Any well-behaved consistency ϕ is *monotonic*: for any two CNs P and P', we have: $P' \preceq P \Rightarrow \phi(P') \preceq \phi(P)$. To compare the pruning capability of consistencies, we use a preorder. A consistency ϕ is *stronger* than (or equal to) a consistency ψ, denoted by $\phi \trianglerighteq \psi$, iff whenever ϕ holds on a CN P, ψ also holds on P. ϕ is *strictly stronger* than ψ, denoted by $\phi \triangleright \psi$, iff $\phi \trianglerighteq \psi$ and there is at least a CN P such that ψ holds on P but not ϕ. ϕ and ψ are *equivalent*, denoted by $\phi \approx \psi$, iff both $\phi \trianglerighteq \psi$ and $\psi \trianglerighteq \phi$.

Now we introduce some concrete consistencies, starting with GAC (Generalized Arc Consistency). A value (x, a) of P is *GAC-consistent* iff for each constraint c of P involving x there exists a valid instantiation I of $scp(c)$ such that I satisfies c and $I[x] = a$. P is GAC-consistent iff every value of P is GAC-consistent. For binary constraints, GAC is often referred to as AC (Arc Consistency). Now, we introduce known consistencies based on decisions. When the domain of a variable of P is empty, P is unsatisfiable (i.e., $sols(P) = \emptyset$), which is denoted by $P = \bot$; to simplify, we consider that no value is present in a CN P such that $P = \bot$. The CN $P|_{x=a}$ is obtained from P by removing every value $b \neq a$ from $dom(x)$. A value (x, a) of P is *SAC-consistent* iff $GAC(P|_{x=a}) \neq \bot$ [9]. A value (x, a) of P is *1-AC-consistent* iff (x, a) is SAC-consistent and $\forall y \in vars(P) \setminus \{x\}, \exists b \in dom(y) \mid (x, a) \in GAC(P|_{y=b})$ [2]. A value (x, a) of P is *BiSAC-consistent* iff $GAC(P^{ia}|_{x=a}) \neq \bot$ where P^{ia} is the CN obtained after removing every value (y, b) of P such that $y \neq x$ and $(x, a) \notin GAC(P|_{y=b})$ [4]. P is SAC-consistent (resp., 1-AC-consistent, BiSAC-consistent) iff every value of P is SAC-consistent (resp., 1-AC-consistent, BiSAC-consistent). P is BoundSAC-consistent iff for every variable x, $min(x)$ and $max(x)$ are SAC-consistent [16]. A decision-based second-order consistency is dual consistency (DC) defined as follows. A locally consistent instantiation $\{(x, a), (y, b)\}$ on P, with $y \neq x$, is DC-consistent iff $(y, b) \in GAC(P|_{x=a})$ and $(x, a) \in GAC(P|_{y=b})$ [14]. P is *DC-consistent* iff every locally consistent instantiation $\{(x, a), (y, b)\}$ on P is DC-consistent. P is *sDC-consistent* (strong DC-consistent) iff P is GAC+DC-consistent, i.e. both GAC-consistent and DC-consistent. All consistencies mentioned above are well-behaved. Also, we know that sDC $\triangleright$ BiSAC $\triangleright$ 1-GAC $\triangleright$ SAC $\triangleright$ BoundSAC $\triangleright$ GAC.

3 Decision-based Consistencies

In this section, we introduce decisions before presenting general classes of consistencies.

3.1 Decisions

A *positive decision* δ is a restriction on a variable x of the form $x = a$ whereas a *negative decision* is a restriction of the form $x \neq a$, with $a \in dom^{init}(x)$. A *membership decision* is a decision of the form $x \in D_x$, where x is a variable and $D_x \subseteq dom^{init}(x)$ is a non-empty set of values; note that D_x is not necessarily $dom(x)$, the current domain of x. Membership decisions generalize both positive and negative decisions as a positive (resp., negative) decision $x = a$ (resp., $x \neq a$) is equivalent to the membership decision $x \in \{a\}$ (resp., $x \in dom^{init}(x) \setminus \{a\}$). The variable involved in a decision δ is denoted by $var(\delta)$.

For a membership decision δ, we define $P|_\delta$ to be the CN obtained (derived) from P such that, if δ denotes $x \in D_x$ and if x is a variable of P then each value $b \in dom^P(x)$ with $b \notin D_x$ is removed from $dom^P(x)$. If Γ is a set of decisions, $P|_\Gamma$ is obtained by restricting P by means of all decisions in Γ, and $vars(\Gamma)$ denotes the set of variables occurring in Γ. Enforcing a given well-behaved consistency ϕ after taking a decision δ on a CN P may be quite informative. As seen later, analyzing the CN $\phi(P|_\delta)$ allows us to identify nogoods. Computing $\phi(P|_\delta)$ in order to make such inferences is called a decision-based ϕ-check on P from δ, or more simply a *decision-based check*. For SAC, a decision-based check from a pair (x, a), usually called a singleton check, aims at comparing $GAC(P|_{x=a})$ with $\bot$.

From now on, Δ will denote a mapping, called *decision mapping*, that associates with every variable x and every possible domain $dom_x \subseteq dom^{init}(x)$, a (possibly empty) set $\Delta(x, dom_x)$ of membership decisions on x such that for every decision $x \in D_x$ in $\Delta(x, dom_x)$, we have $D_x \subseteq dom_x$. For example, an illustrative decision mapping Δ^{ex} may be such that $\Delta^{ex}(x, \{a, b, c, d\}) = \{x \in \{a, b\}, x \in \{d\}\}$. For the current domain of x, i.e., the domain of x in the context of a current CN P, $\Delta(x, dom(x)) = \Delta(x, dom^P(x))$ will be simplified into $\Delta(x)$ when this is unambiguous. To simplify, we shall also refer to Δ as the set of all "current" decisions w.r.t. P, i.e., Δ will be considered as $\cup_{x \in vars(P)}\Delta(x)$. This quite general definition of decision mapping will be considered as our basis to perform decision-based checks. Sometimes, we need to restrict sets of decisions in order to have each value occurring at least once in a decision. A set of decisions Γ on a variable x is said to be a *cover* of $\cup_{(x \in D_x) \in \Gamma} D_x$. For example, $\Delta^{ex}(x, \{a, b, c, d\})$, as defined above, is a cover of $\{a, b, d\}$. Δ is a *cover* for (x, dom_x), where $dom_x \subseteq dom^{init}(x)$, iff $\Delta(x, dom_x)$ is a cover of dom_x. For example, Δ^{ex} is not a cover for $(x, \{a, b, c, d\})$. Δ is a *cover* for x iff for every $dom_x \subseteq dom^{init}(x)$, Δ is a *cover* for (x, dom_x). Δ is *covering* iff for every variable x, Δ is a *cover* for x.

As examples of decision mappings, we have for every variable x:

- $\Delta^{id}(x)$ containing only $x \in dom(x)$;
- $\Delta^{=}(x)$ containing $x = a$, $\forall a \in dom(x)$;
- $\Delta^{\neq}(x)$ containing $x \neq a$, $\forall a \in dom(x)$;
- $\Delta^{bnd}(x)$ containing $x = min(x)$ and $x = max(x)$;
- $\Delta^{P_2}(x)$ containing $x \in D_x^1$ and $x \in D_x^2$ where D_x^1 and D_x^2 resp. contain the first and last $|dom(x)|/2$ values of $dom(x)$.

For example, if P is a CN such that $vars(P) = \{x, y\}$ with $dom(x) = dom^P(x) = \{a, b, c\}$ and $dom(y) = dom^P(y) = \{a, b\}$ then:

- $\Delta^{id}(x) = \{x \in \{a, b, c\}\}$ and $\Delta^{id}(y) = \{y \in \{a, b\}\}$;
- $\Delta^{=}(x) = \{x = a, x = b, x = c\}$ and $\Delta^{=}(y) = \{y = a, y = b\}$;
- $\Delta^{\neq}(x) = \{x \neq a, x \neq b, x \neq c\}$ and $\Delta^{\neq}(y) = \{y \neq a, y \neq b\}$;
- $\Delta^{bnd}(x) = \{x = a, x = c\}$ and $\Delta^{bnd}(y) = \{y = a, y = b\}$;
- $\Delta^{P_2}(x) = \{x \in \{a, b\}, x = c\}$ and $\Delta^{P_2}(y) = \{y = a, y = b\}$.

Note that, except for Δ^{bnd}, all these decision mappings are covering. Also, the reader should be aware of the dynamic nature of decision mappings. For example, if P' is obtained from P after removing a from $dom^P(x)$ then we have $\Delta^{bnd}(x, dom^{P'}(x)) = \{x = b, x = c\}$.

3.2 Two Classes of First-order Consistencies

Informally, a decision-based consistency is a property defined from the outcome of decision-based checks. From now on, we consider given a well-behaved nogood-identifying consistency ϕ and a decision mapping Δ. A first kind of inferences is made possible by considering the effect of a decision-based check on the domain initially reduced by the decision that has been taken.

Definition 1 (Consistency S^{ϕ}_{Δ}). *A value (x, a) of a CN P is S^{ϕ}_{Δ}-consistent iff for every membership decision $x \in D_x$ in $\Delta(x)$ such that $a \in D_x$, we have $(x, a) \in \phi(P|_{x \in D_x})$.*

The following result can be seen as a generalization of Property 1 in [2].

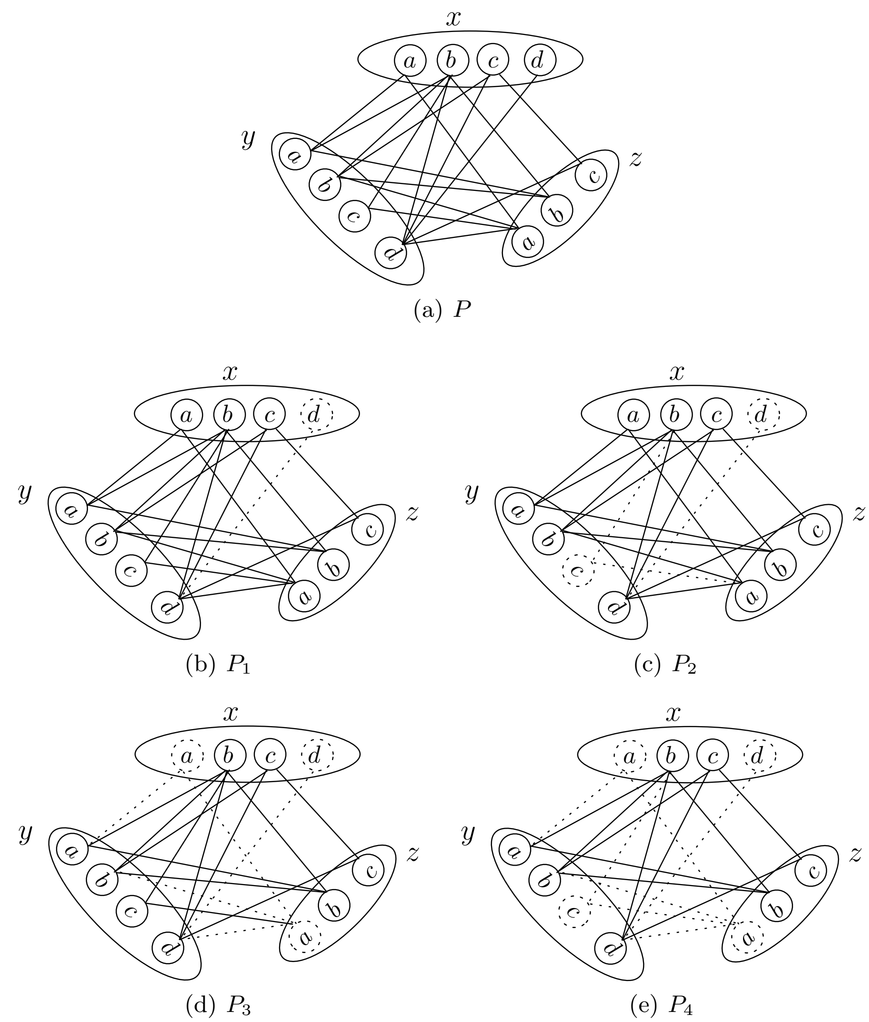
Proposition 1. *Any S^{ϕ}_{Δ}-inconsistent value is globally inconsistent.*

Proof. If (x, a) is an S^{ϕ}_{Δ}-inconsistent value, then we know that there exists a decision $x \in D_x$ in $\Delta(x)$ such that $a \in D_x$ and $(x, a) \notin \phi(P|_{x \in D_x})$. We deduce that $x \in D_x \wedge x = a$ cannot lead to a solution because ϕ is nogood-identifying. This simplifies into $x = a$ being a nogood because $a \in D_x$. □

SAC is equivalent to $S^{GAC}_{\Delta^{=}}$ (because no value belongs to $\perp$), and BoundSAC[1] is equivalent to $S^{GAC}_{\Delta^{bnd}}$. Note also that GAC is equivalent to $S^{GAC}_{\Delta^{id}}$. As a simple illustration of S^{ϕ}_{Δ}, let us consider the five binary CNs depicted in Figure 1; each vertex denotes a value, each edge denotes an allowed tuple and each dotted vertex (resp., edge) means that the value (resp., tuple) is removed (resp., no more relevant). P_1, P_2, P_3 and P_4 are obtained from P by removing values that are S^{AC}_{Δ}-inconsistent when Δ is set to Δ^{id}, Δ^{P_2}, Δ^{bnd} and $\Delta^{=}$, respectively. For example, for Δ^{P_2}, we find that $(y, c) \notin AC(P|_{y \in \{c,d\}})$. Note that the CN P_4 is also obtained when setting Δ to $\Delta^{\neq}$.

---

[1] Another related consistency is Existential SAC [16], which guarantees that some value in the domain of each variable is SAC-consistent. However, there is no guarantee about the network obtained after checking Existential SAC due to the non-deterministic nature of this consistency. Existential SAC is not an element of S^{ϕ}_{Δ}.



(a) P

(b) P_1 (c) P_2

(d) P_3 (e) P_4

Fig. 1. Illustration of S_{Δ}^{GAC}.

In [2], it is also shown that inferences regarding values may be obtained by considering the result of several decision-based checks. This is generalized below. The idea is that a value (x, a) of P can be safely removed when there exist a variable y and a cover $\Gamma \subseteq \Delta(y)$ of $dom(y)$ such that every decision-based check, performed from a decision in Γ, eliminates (x, a).

Definition 2 (Consistency E_{Δ}^{ϕ}). *A value (x, a) of a CN P is E_{Δ}^{ϕ}-consistent w.r.t. a variable $y \neq x$ of P iff for every cover Γ of $dom(y)$ such that $\Gamma \subseteq \Delta(y)$, there exists a decision $y \in D_y$ in Γ such that $(x, a) \in \phi(P|_{y \in D_y})$. (x, a) is E_{Δ}^{ϕ}-consistent iff (x, a) is E_{Δ}^{ϕ}-consistent w.r.t. every variable $y \neq x$ of P.*

Proposition 2. *Any E^{ϕ}_{Δ}-inconsistent value is globally inconsistent.*

Proof. If (x, a) is an E^{ϕ}_{Δ}-inconsistent value, then we know that there exists a variable $y \neq x$ of P and a set $\Gamma \subseteq \Delta(y)$ such that (i) $dom^P(y) = \cup_{(y \in D_y) \in \Gamma} D_y$ and (ii) every decision $y \in D_y$ in Γ entails $(x, a) \notin \phi(P|_{y \in D_y})$. As Γ is a cover of $dom(y)$, we infer that $sols(P) = \cup_{(y \in D_y) \in \Gamma} sols(P|_{y \in D_y})$. Because ϕ preserves solutions, we have $sols(P) = \cup_{(y \in D_y) \in \Gamma} sols(\phi(P|_{y \in D_y}))$. For every $y \in D_y$ in Γ, we know that $(x, a) \notin \phi(P|_{y \in D_y})$. We deduce that (x, a) cannot be involved in any solution. □

As an illustration, let us consider the CN of Figure 1(a) and $\Delta(x) = \{x \in \{a, c\}, x \in \{b, d\}\}$. We can show that (z, a) is E^{GAC}_{Δ}-inconsistent because $(z, a) \notin AC(P|_{x \in \{a,c\}})$ and $(z, a) \notin AC(P|_{x \in \{b,d\}})$. The consistency P-k-AC, introduced in [2], corresponds to S^{ϕ}_{Δ}+E^{ϕ}_{Δ} where $\phi = AC$ and Δ necessarily corresponds to a partition of each domain into pieces of size at most k.

3.3 Classes Related to Nogoods of Size 2

Decision-based consistencies introduced above are clearly domain-filtering: they allow us to identify inconsistent values. However, decision-based consistencies are also naturally orientated towards identifying nogoods of size 2. $NG2(P)^{\phi}_{\Delta}$ denotes the set of nogoods of size 2 that can be directly derived from checks on P based on the consistency ϕ and the decision mapping Δ. From this set, together with a decision $x \in D_x$, we obtain a set $ND1(P, x \in D_x)^{\phi}_{\Delta}$ of negative decisions that can be used to make further inferences.

Definition 3. *Let P be a CN and $x \in D_x$ be a membership decision in $\Delta(x)$.*

- *$NG2(P)^{\phi}_{x \in D_x}$ denotes the set of locally consistent instantiations $\{(x, a), (y, b)\}$ on P such that $a \in D_x$ and $(y, b) \notin \phi(P|_{x \in D_x})$.*
- *$NG2(P)^{\phi}_{\Delta}$ denotes the set $\cup_{\delta \in \Delta} NG2(P)^{\phi}_{\delta}$.*
- *$ND1(P, x \in D_x)^{\phi}_{\Delta}$ denotes the set of negative decisions $y \neq b$ such that every value $a \in D_x$ is such that $\{(x, a), (y, b)\} \in \widetilde{P}$ or $\{(x, a), (y, b)\} \in NG2(P)^{\phi}_{\Delta \setminus \{x \in D_x\}}$.*

From *ND1* sets, we can define a new class B^{ϕ}_{Δ} of consistencies.

Definition 4 (Consistency B^{ϕ}_{Δ}). *A value (x, a) of a CN P is B^{ϕ}_{Δ}-consistent iff for every membership decision $x \in D_x$ in $\Delta(x)$ such that $a \in D_x$, we have $(x, a) \in \phi(P|_{\{x \in D_x\} \cup ND1(P, x \in D_x)^{\phi}_{\Delta}})$.*

Proposition 3. *Any B^{ϕ}_{Δ}-inconsistent value is globally inconsistent.*

Proof. The proof is similar to that of Proposition 1. The only difference is that the network P is made smaller by removing some additional values by means of negative decisions. However, in the context of a decision $x \in D_x$ taken on P, the inferred negative decisions correspond to inconsistent values because they are derived from nogoods of size 2 (showing that elements of $NG2(P)^{\phi}_{\Delta}$ are nogoods is immediate). □

As an illustration of B^{ϕ}_{Δ}, let us consider the binary CN P in Figure 2(a). For $\phi = AC$ and $\Delta = \Delta^{P_2} = \{x \in \{a,b\}, x = c, y \in \{a,b\}, y = c, z = a, z = b\}$ we obtain $NG2(P)^{\phi}_{\Delta} = \{\{(x,a),(y,a)\}, \{(x,a),(z,b)\}, \{(x,b),(y,c)\}\}$ since for example $(x,a) \notin AC(P|_{y\in\{a,b\}})$. Because $\{(x,b),(z,b)\} \in \widetilde{P}$ and $\{(x,a),(z,b)\} \in NG2(P)^{\phi}_{\Delta}$, $ND1(P, x \in \{a,b\})^{\phi}_{\Delta} = \{z \neq b\}$, and (x,a) is B^{ϕ}_{Δ}-inconsistent as $(x,a) \notin AC(P|_{x\in\{a,b\}\cup\{z\neq b\}})$. Here, P is S^{ϕ}_{Δ}-consistent, but not B^{ϕ}_{Δ}-consistent.

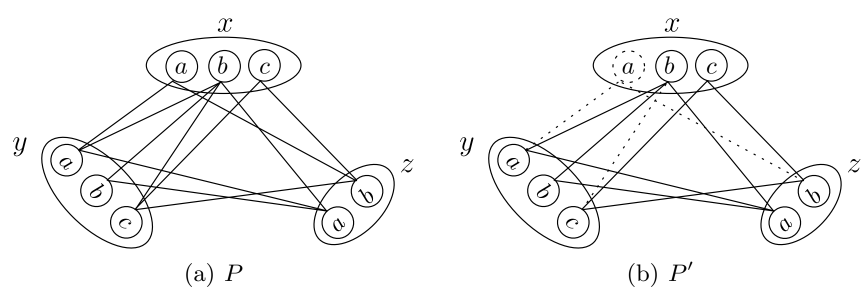


Fig. 2. Illustration of B^{GAC}_{Δ} and D^{GAC}_{Δ}.

Note that BiSAC [4] is equivalent to $B^{GAC}_{\Delta^=}$. On the other hand, there is a 2-order consistency that can be naturally defined as follows.

Definition 5 (Consistency D^{ϕ}_{Δ}). *A locally consistent instantiation* $\{(x,a),(y,b)\}$ *on a CN P is D^{ϕ}_{Δ}-consistent iff for every membership decision $x \in D_x$ in $\Delta(x)$ such that $a \in D_x$, $(y,b) \in \phi(P|_{x\in D_x})$ and for every membership decision $y \in D_y$ in $\Delta(y)$ such that $b \in D_y$, $(x,a) \in \phi(P|_{y\in D_y})$.*

Proposition 4. *Any D^{ϕ}_{Δ}-inconsistent instantiation is globally inconsistent.*

Proof. D^{ϕ}_{Δ}-inconsistent instantiations are exactly those in $NG2(P)^{\phi}_{\Delta}$, which are nogoods. □

Note that DC [15] is equivalent to $D^{GAC}_{\Delta^=}$, and recall that DC is equivalent to PC (Path Consistency) for binary CNs. D^{ϕ}_{Δ} (being 2-order) is obviously incomparable with previously introduced domain-filtering consistencies. However, a natural practical approach is to benefit from decision-based checks to record both S^{ϕ}_{Δ}-inconsistent values and D^{ϕ}_{Δ}-inconsistent instantiations. This corresponds to the combined consistency $S^{\phi}_{\Delta}+D^{\phi}_{\Delta}$.

As an illustration of D^{ϕ}_{Δ}, let us consider again Figure 2. For $\phi = AC$ and $\Delta = \Delta^{P_2} = \{x \in \{a,b\}, x = c, y \in \{a,b\}, y = c, z = a, z = b\}$, we have that P is S^{ϕ}_{Δ}-consistent, not B^{ϕ}_{Δ}-consistent and not D^{ϕ}_{Δ}-consistent. Enforcing $S^{\phi}_{\Delta} + D^{\phi}_{\Delta}$ on P yields the CN P', which is also the strong DC-closure (here, AC+PC-closure) of P.

4 Qualitative Study

In this section, we study the relationships between the different classes of consistencies (as well as some of their combinations), and discuss refinements and well-behavedness of consistencies.

4.1 Relationships between Consistencies

From Definitions 1 and 4, it is immediate that any S^ϕ_Δ-inconsistent value is necessarily B^ϕ_Δ-inconsistent.

Proposition 5. *$B^\phi_\Delta \trianglerighteq S^\phi_\Delta$.*

In order to relate B^ϕ_Δ with E^ϕ_Δ, we need to consider covering sets of decisions.

Proposition 6. *If Δ is covering, $B^\phi_\Delta \trianglerighteq E^\phi_\Delta$.*

Proof. We show that every E^ϕ_Δ-inconsistent value in a CN P is necessarily B^ϕ_Δ-inconsistent. Assume that (x, a) is a E^ϕ_Δ-inconsistent value. It means that there exists a variable $y \neq x$ of P and $\Gamma \subseteq \Delta(y)$ such that $dom^P(y) = \cup_{(y \in D_y) \in \Gamma} D_y$ and every decision $y \in D_y$ in Γ is such that $(x, a) \notin \phi(P|_{y \in D_y})$. We deduce that for every value $b \in dom^P(y)$, we have $\{(x, a), (y, b)\}$ in $NG2(P)^\phi_\Delta$. On the other hand, we know that there exists a decision $x \in D_x$ in Δ such that $a \in D_x$ (since Δ is covering). Hence, $ND1(P, x \in D_x)^\phi_\Delta$ contains a negative decision $y \neq b$ for each value in $dom^P(y)$. It follows that $\phi(P|_{\{x \in D_x\} \cup ND1(P, x \in D_x)^\phi_\Delta}) = \bot$, and (x, a) is B^ϕ_Δ-inconsistent. □

As a corollary, we have $B^\phi_\Delta \trianglerighteq S^\phi_\Delta + E^\phi_\Delta$ when Δ is covering. Note that there exist consistencies ϕ and decision mappings Δ such that B^ϕ_Δ is strictly stronger ($\triangleright$) than S^ϕ_Δ and E^ϕ_Δ (and also $S^\phi_\Delta + E^\phi_\Delta$). For example, when $\phi = AC$ and $\Delta = \Delta^=$, we have $B^\phi_\Delta = BiSAC$, $S^\phi_\Delta = SAC$ and $S^\phi_\Delta + E^\phi_\Delta = 1\text{-}AC$, and we know that $BiSAC \triangleright 1\text{-}AC$ [4], and $1\text{-}AC \triangleright SAC$ [2].

Because D^ϕ_Δ captures all 2-sized nogoods while S^ϕ_Δ can eliminate inconsistent values, it follows that the joint use of these two consistencies is stronger than B^ϕ_Δ.

Proposition 7. *$S^\phi_\Delta + D^\phi_\Delta \trianglerighteq B^\phi_\Delta$.*

Proof. Let P be a CN that is $S^\phi_\Delta + D^\phi_\Delta$-consistent. As P is S^ϕ_Δ-consistent, for every decision $x \in D_x$ in Δ and every $a \in D_x$, we have $(x, a) \in \phi(P|_{x \in D_x})$. But $\phi(P|_{x \in D_x}) = \phi(P|_{\{x \in D_x\} \cup ND1(P, x \in D_x)^\phi_\Delta})$ since P being D^ϕ_Δ-consistent entails $NG2(P)^\phi_\Delta = \emptyset$ and $ND1(P, x \in D_x)^\phi_\Delta = \emptyset$. We deduce that P is B^ϕ_Δ-consistent. □

One may expect that $S^\phi_\Delta \trianglerighteq \phi$. However, to guarantee this, we need both ϕ to be domain-filtering and Δ to be covering, For example, $S^{AC}_\Delta \trianglerighteq AC$ does not hold if for every $dom_x \subseteq dom^{init}(x)$, we have $\Delta(x, dom_x) = \emptyset$: it suffices to build a CN P with a value (x, a) being arc-inconsistent.

Proposition 8. *If ϕ is domain-filtering and Δ is covering, $S^{\phi}_{\Delta} \trianglerighteq \phi$.*

Proof. Assume that (x, a) is a ϕ-inconsistent value of a CN P. This means that $(x, a) \notin \phi(P)$. As Δ is covering, there exists a decision $x \in D_x$ in Δ with $a \in D_x$. We know that $P|_{x \in D_x} \preceq P$. By monotonicity of ϕ, $\phi(P|_{x \in D_x}) \preceq \phi(P)$. Since $(x, a) \notin \phi(P)$, we deduce that $(x, a) \notin \phi(P|_{x \in D_x})$. So, (x, a) is S^{ϕ}_{Δ}-inconsistent, and S^{ϕ}_{Δ} is stronger than ϕ. □

Figure 3 shows the relationships between the different classes of consistencies introduced so far. There are many ways to instantiate these classes because the choice of Δ and ϕ is left open. If we consider binary CNs, and choose $\phi = AC$ and $\Delta = \Delta^{=}$, we obtain known consistencies. We directly benefit from the relationships of Figure 3, and have just to prove strictness when it holds. Figure 4 shows this where an arrow denotes now $\triangleright$ (instead of $\trianglerighteq$). An extreme instantiation case is when $\Delta = \Delta^{id}$ and ϕ is domain-filtering. In this case, all consistencies collapse: we have $S^{\phi}_{\Delta^{id}} = E^{\phi}_{\Delta^{id}} = B^{\phi}_{\Delta^{id}} = D^{\phi}_{\Delta^{id}} = \phi$. This means that our framework of decision-based consistencies is general enough to encompass all classical local consistencies. Although this is appealing for theoretical reasons (e.g., see Proposition 11 later), the main objective of decision-based consistencies remains to learn relevant nogoods from nontrivial decision-based checks.

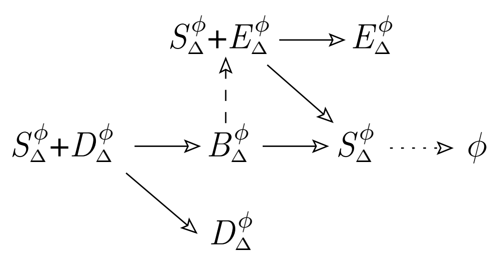


Fig. 3. Summary of the relationships between (classes of) consistencies. An arrow from φ to ψ means that $\varphi \trianglerighteq \psi$. A dashed (resp., dotted) arrow means that the relationship is guaranteed provided that Δ is covering (resp., Δ is covering and ϕ is domain-filtering).

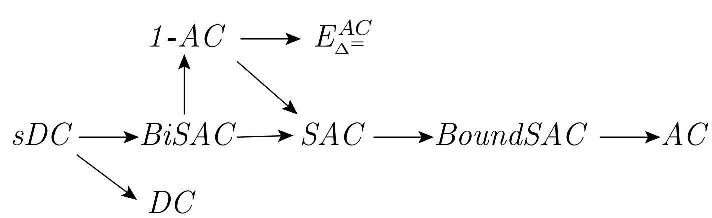


Fig. 4. Relationships between consistencies when $\phi = AC$ and $\Delta = \Delta^{=}$ (except for BoundSAC which is derived from Δ^{bnd}). An arrow from φ to ψ means that $\varphi \triangleright \psi$.

4.2 Refinements

Now, we show that two consistencies of the same class can be naturally compared when a refinement connection exists between their decision mappings.

Definition 6. *A decision mapping Δ' is a refinement of a decision mapping Δ iff for each decision $x \in D_x$ in Δ there exists a subset $\Gamma \subseteq \Delta'(x)$ that is a cover of D_x.*

For example, $\{x \in \{a,b\}, x = c\}$ is a refinement of $\{x \in \{a,b,c\}\}$, and $\{x \in \{a,b\}, x = c, y = a, y = b, y = c\}$ is a refinement of $\{x \in \{a,b,c\}, y \in \{a,b\}, y \in \{b,c\}\}$. Unsurprisingly, using refined sets of decisions improves inference capability as shown by the following proposition.

Proposition 9. *If Δ and Δ' are two decision mappings such that Δ' is a refinement of Δ, then $X^{\phi}_{\Delta'} \trianglerighteq X^{\phi}_{\Delta}$ where $X \in \{S, E, B, D\}$.*

Proof. Due to lack of space, we only show that $S^{\phi}_{\Delta'} \trianglerighteq S^{\phi}_{\Delta}$. Assume that (x, a) is an S^{ϕ}_{Δ}-inconsistent value of a CN P. This means that there exists a decision $x \in D_x$ in $\Delta(x)$ such that $a \in D_x$ and $(x, a) \notin \phi(P|_{x \in D_x})$. We know, by hypothesis, that there exists a subset $\Gamma \subseteq \Delta'(x)$ such that $D_x = \cup_{(x \in D'_x) \in \Gamma} D'_x$. Hence, there exists (at least) a decision $x \in D'_x$ in Γ such that $a \in D'_x$ and $D'_x \subseteq D_x$. As $D'_x \subseteq D_x$, we have $P|_{x \in D'_x} \preceq P|_{x \in D_x}$, and by monotonicity of ϕ, $\phi(P|_{x \in D'_x}) \preceq \phi(P|_{x \in D_x})$. Consequently, $(x, a) \notin \phi(P|_{x \in D_x})$ implies $(x, a) \notin \phi(P|_{x \in D'_x})$. We deduce that there exists a decision $x \in D'_x$ in $\Delta'(x)$ such that $a \in D'_x$ and $(x, a) \notin \phi(P|_{x \in D''_x})$. Then (x, a) is $S^{\phi}_{\Delta'}$-inconsistent. We conclude that $S^{\phi}_{\Delta'} \trianglerighteq S^{\phi}_{\Delta}$. □

As a corollary, for any decision mapping Δ, we have: $X^{\phi}_{\Delta^=} \trianglerighteq X^{\phi}_{\Delta} \trianglerighteq X^{\phi}_{\Delta^{id}}$ where $X \in \{S, E, B, D\}$. In particular, if $\phi = GAC$, we have $SAC = S^{GAC}_{\Delta^=} \trianglerighteq S^{GAC}_{\Delta} \trianglerighteq S^{GAC}_{\Delta^{id}} = GAC$.

Because, consistencies S^{ϕ}_{Δ} identify inconsistent values on the basis of a single decision, we obtain the two following results. In the spirit of our set view of decision mappings, for any two decision mappings Δ_1 and Δ_2, $\Delta_1 \cup \Delta_2$ is the decision mapping such that for every variable x and every $dom_x \subseteq dom^{init}(x)$, $(\Delta_1 \cup \Delta_2)(x, dom_x) = \Delta_1(x, dom_x) + \Delta_2(x, dom_x)$.

Proposition 10. *Let Δ_1 and Δ_2 be two decision mappings. We have $S^{\phi}_{\Delta_1} + S^{\phi}_{\Delta_2} = S^{\phi}_{\Delta_1 \cup \Delta_2}$.*

Proof. Let P be a CN and (x, a) be a value of P. (x, a) is $S^{\phi}_{\Delta_1 \cup \Delta_2}$-inconsistent $\Leftrightarrow$ there exists a decision $x \in D_x$ in $\Delta_1 \cup \Delta_2$ such that $(x, a) \notin \phi(P|_{x \in D_x}) \Leftrightarrow (x, a)$ is $S^{\phi}_{\Delta_1}$-inconsistent or (x, a) is $S^{\phi}_{\Delta_2}$-inconsistent $\Leftrightarrow (x, a)$ is $S^{\phi}_{\Delta_1} + S^{\phi}_{\Delta_2}$-inconsistent. □

$\mathscr{S}^{\phi}$ denotes the set of equivalence classes modulo $\approx$ of the consistencies S^{ϕ}_{Δ} that can be built from ϕ and all possible decision mappings Δ. It forms a complete lattice, in a similar way to what has been shown for qualitative constraint networks [7].

Proposition 11. *$(\mathscr{S}^\phi, \trianglerighteq)$ is a complete lattice with $S^\phi_{\Delta^=}$ as greatest element and $S^\phi_{\Delta^{id}}$ as least element.*

Proof. Let $S^\phi_{\Delta_1}$ and $S^\phi_{\Delta_2}$ be two consistencies in $\mathscr{S}^\phi$.
(Existence of binary joins) From Proposition 10, we can infer that $S^\phi_{\Delta_1 \cup \Delta_2}$ is the least upper bound of $S^\phi_{\Delta_1}$ and $S^\phi_{\Delta_2}$.
(Existence of binary meets) Let us define the set E as $E = \{S^\phi_\Delta \in \mathscr{S}^\phi : S^\phi_\Delta \trianglelefteq S^\phi_{\Delta_1}$ and $S^\phi_\Delta \trianglelefteq S^\phi_{\Delta_2}\}$. Note that $E \neq \emptyset$ since $S^\phi_{\Delta^{id}} \in E$. Next, let us define $S^\phi_{\Delta^E}$ such that $\Delta^E = \bigcup_{S^\phi_{\Delta_i} \in E} \Delta_i$. For every $S^\phi_{\Delta_i} \in E$, Δ^E is a refinement Δ_i, and so, from Proposition 9, we know that $S^\phi_{\Delta^E}$ is an upper bound of E. We now prove by contradiction that $S^\phi_{\Delta^E} \trianglelefteq S^\phi_{\Delta_1}$. Suppose that there is a value (x, a) of a CN P that is $S^\phi_{\Delta^E}$-inconsistent and $S^\phi_{\Delta_1}$-consistent. This means that there exists a decision $x \in D_x$ in $\Delta(x)$ such that $(x, a) \notin \phi(P|_{x \in D_x})$. From construction of Δ, we know that there exists a decision mapping Δ_i such that $S^\phi_{\Delta_i} \in E$ and $x \in D_x$ is in Δ_i. By definition of E, we know that $S^\phi_{\Delta_i} \trianglelefteq S^\phi_{\Delta_1}$. Consequently, (x, a) is $S^\phi_{\Delta_i}$-consistent and $(x, a) \in \phi(P|_{x \in D_x})$. This is a contradiction, so $S^\phi_\Delta \trianglelefteq S^\phi_{\Delta_1}$. Similarly, we have $S^\phi_\Delta \trianglelefteq S^\phi_{\Delta_2}$. Then S^ϕ_Δ is the greatest lower bound of $S^\phi_{\Delta_1}$ and $S^\phi_{\Delta_2}$. □

4.3 Well-behavedness

Finally, we are interested in well-behavedness of consistencies. Actually, in the general case, the consistencies S^ϕ_Δ, E^ϕ_Δ, B^ϕ_Δ and D^ϕ_Δ are not necessarily well-behaved for $(\mathscr{P}, \preceq)$. Consider as an illustration three CNs P, P_1 and P_2 which differ only by the domain of the variable x: $dom^P(x) = \{a, b, c, d\}$, $dom^{P_1}(x) = \{a, b, c\}$ and $dom^{P_2}(x) = \{d\}$. Now, consider a decision mapping Δ defined for the variable x and the domains $\{a, b, c, d\}$, $\{a, b, c\}$ and $\{d\}$ by: $\Delta(x, \{a, b, c, d\}) = \{x \in \{a\}\}$, $\Delta(x, \{a, b, c\}) = \{x \in \{a, b, c\}\}$ and $\Delta(x, \{d\}) = \{x \in \{d\}\}$. Despite the fact that $dom^P(x) = dom^{P_1}(x) \cup dom^{P_2}(x)$, one can see that the value (x, a) could be S^ϕ_Δ-consistent in P_1 and P_2, whereas S^ϕ_Δ-inconsistent in P. With such a Δ, S^ϕ_Δ is not guaranteed to be well-behaved.

Nevertheless, there exist decision mappings for which consistencies are guaranteed to be well-behaved, at least those of the class S^ϕ_Δ. Informally, a relevant decision mapping is a decision mapping that keeps its precision (in terms of decisions) when domains are restricted.

Definition 7. *A decision mapping Δ is said to be* relevant *if and only if for any variable x, any two sets of values dom_x and dom'_x such that $dom'_x \subsetneq dom_x \subseteq dom^{init}(x)$ and any decision $x \in D_x$ in $\Delta(x, dom_x)$, we have:*

$$D_x \cap dom'_x \neq \emptyset \Rightarrow \exists \Gamma \subseteq \Delta(x, dom'_x) \mid D_x \cap dom'_x = \cup_{(x \in D'_x) \in \Gamma} D'_x.$$

We can notice that Δ^{id}, $\Delta^=$, $\Delta^{\neq}$, Δ^{bnd} are relevant decision mappings. For our proposition, we need some additional definitions. A CN P' is a sub-CN of a CN P if P' can be obtained from P by simply removing certain values. If P_1 and P_2 are two CNs that only differ by the domains of their variables, then $P = P_1 \cup P_2$ is the CN such that P_1 and P_2 are sub-CNs of P and for every variable x, $dom^P(x) = dom^{P_1}(x) \cup dom^{P_2}(x)$.

Proposition 12. *Let Δ be a relevant decision mapping and let P, P_1, and P_2 be three CNs such that $P = P_1 \cup P_2$. If P_1 and P_2 are S^{ϕ}_{Δ}-consistent then P is S^{ϕ}_{Δ}-consistent.*

Proof. Let (x, a) be a value of $P = P_1 \cup P_2$. Let us show that this value is S^{ϕ}_{Δ}-consistent. Consider a membership decision $x \in D_x$ in $\Delta(x, dom^P(x))$ such that $a \in D_x$. We have to show that $(x, a) \in \phi(P|_{x \in D_x})$. We know that $dom^P(x) = dom^{P_1}(x) \cup dom^{P_2}(x)$. Hence, $a \in dom^{P_1}(x)$ or $x \in dom^{P_2}(x)$. Assume that $a \in dom^{P_1}(x)$ (the case $a \in dom^{P_2}(x)$ can be handled in a similar way). Since Δ is a relevant decision mapping, there exists $\Gamma \subseteq \Delta(x, dom^{P_1}(x))$ such that $D_x \cap dom^{P_1}(x) = \cup_{(x \in D'_x) \in \Gamma} D'_x$. It follows that there exists a decision $x \in D^1_x$ in $\Delta(x, dom^{P_1}(x))$ such that $a \in D^1_x$ and $D^1_x \subseteq D_x$. From the fact that P_1 is S^{ϕ}_{Δ}-consistent we know that $(x, a) \in \phi(P_1|_{x \in D^1_x})$. Since $a \in D^1_x$, $D^1_x \subseteq D_x$ and P_1 is a sub-CN of P we can assert that $(x, a) \in \phi(P|_{x \in D_x})$. We conclude that (x, a) is a S^{ϕ}_{Δ}-consistent value of P. □

Corollary 1. *If Δ is a relevant decision mapping then S^{ϕ}_{Δ} is well-behaved.*

Indeed, to obtain the closure of a CN P, it suffices to take the union of all sub-CNs of P which are S^{ϕ}_{Δ}-consistent. Hence, the consistency S^{ϕ}_{Δ} for which Δ is a relevant decision mapping is well-behaved for $(\mathscr{P}, \preceq)$.

5 Conclusion

In this paper, our aim was to give a precise picture of decision-based consistencies by developing a hierarchy of general classes. This general framework offers the user a vast range of new consistencies. Several issues have now to be addressed. First, me must determine the conditions under which overlapping between decisions may be beneficial. Overlapping allows us to cover domains while considering weak decisions (e.g., decisions in $\Delta^{\neq}$) that are quick to propagate, and might also be useful to tractability procedures (e.g., in situations where only some decisions lead to known tractable networks). Second, we must seek to elaborate dynamic procedures (heuristics) so as automatically select the right decision-based consistency (set of membership decisions) at each step of a backtrack search as in [19]; many new combinations are permitted. Finally, bound consistencies and especially singleton checks on bounds may be revisited by checking several values at once (using intervals at bounds with the mechanism of detecting X^{ϕ}_{Δ}-inconsistent values), so as to speed up the inference process in shaving procedures. These are some of the main perspectives.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by OSEO (project ISI PAJERO).

References

1. K.R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.
2. H. Bennaceur and M.S. Affane. Partition-k-AC: An efficient filtering technique combining domain partition and arc consistency. In *Proceedings of CP'01*, pages 560–564, 2001.
3. C. Bessiere. Constraint propagation. In *Handbook of Constraint Programming*, chapter 3. Elsevier, 2006.
4. C. Bessiere and R. Debruyne. Theoretical analysis of singleton arc consistency and its extensions. *Artificial Intelligence*, 172(1):29–41, 2008.
5. C. Bessiere, K. Stergiou, and T. Walsh. Domain filtering consistencies for non-binary constraints. *Artificial Intelligence*, 72(6-7):800–822, 2008.
6. J. Carlier and E. Pinson. Adjustments of heads and tails for the job-shop problem. *European Journal of Operational Research*, 78:146–161, 1994.
7. J.-F. Condotta and C. Lecoutre. A class of df-consistencies for qualitative constraint networks. In *Proceedings of KR'10*, pages 319–328, 2010.
8. R. Debruyne and C. Bessiere. From restricted path consistency to max-restricted path consistency. In *Proceedings of CP'97*, pages 312–326, 1997.
9. R. Debruyne and C. Bessiere. Some practical filtering techniques for the constraint satisfaction problem. In *Proceedings of IJCAI'97*, pages 412–417, 1997.
10. R. Debruyne and C. Bessiere. Domain filtering consistencies. *Journal of Artificial Intelligence Research*, 14:205–230, 2001.
11. R. Dechter. *Constraint processing*. Morgan Kaufmann, 2003.
12. E.C. Freuder and C. Elfe. Neighborhood inverse consistency preprocessing. In *Proceedings of AAAI'96*, pages 202–208, 1996.
13. C. Lecoutre. *Constraint networks: techniques and algorithms*. ISTE/Wiley, 2009.
14. C. Lecoutre, S. Cardon, and J. Vion. Conservative dual consistency. In *Proceedings of AAAI'07*, pages 237–242, 2007.
15. C. Lecoutre, S. Cardon, and J. Vion. Second-order consistencies. *Journal of Artificial Intelligence Research (JAIR)*, 40:175–219, 2011.
16. C. Lecoutre and P. Prosser. Maintaining singleton arc consistency. In *Proceedings of CPAI'06 workshop held with CP'06*, pages 47–61, 2006.
17. O. Lhomme. Quick shaving. In *Proceedings of AAAI'05*, pages 411–415, 2005.
18. P. Martin and D.B. Shmoys. A new approach to computing optimal schedules for the job-shop scheduling problem. In *Proceedings of IPCO'96*, pages 389–403, 1996.
19. K. Stergiou. Heuristics for dynamically adapting propagation. In *Proceedings of ECAI'08*, pages 485–489, 2008.
20. K. Stergiou and T. Walsh. Inverse consistencies for non-binary constraints. In *Proceedings of ECAI'06*, pages 153–157, 2006.
21. R. Szymanek and C. Lecoutre. Constraint-level advice for shaving. In *Proceedings of ICLP'08*, pages 636–650, 2008.
22. M.R.C. van Dongen. Beyond singleton arc consistency. In *Proceedings of ECAI'06*, pages 163–167, 2006.