Simplification of Many-Valued Logic Formulas Using Anti-Links*

Bernhard Beckert and Reiner Hähnle
Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme, D-76128 Karlsruhe, Germany. Email: {beckert,haehnle}@ira.uka.de,
WWW: http://i12www.ira.uka.de/

and

Gonzalo Escalada-Imaz
Institut d'Investigació en Intel·ligència Artificial, Campus Universitat Autònoma de Barcelona, E-08193 Bellaterra, Barcelona, Spain. Email: gonzalo@sinera.iiia.csic.es,
WWW: http://www.iiia.csic.es

June 3, 1997

Abstract. We present the theoretical foundations of the many-valued generalization of a technique for simplifying large non-clausal formulas in propositional logic, that is called *removal of anti-links*. Possible applications of anti-links include computation of prime implicates of large *non-clausal* formulas as required, for example, in diagnosis. Anti-links do not compute any normal form of a given formula themselves, rather, they remove certain forms of redundancy from formulas in negation normal form (NNF). Their main advantage is that no clausal normal form has to be computed in order to remove redundant parts of a formula. In this paper, we define an anti-link operation on a generic language for expressing many-valued logic formulas called *signed NNF* and we show that all interesting properties of two-valued anti-links generalize to the many-valued setting, although in a non-trivial way.

1 Introduction

In this article we present the theoretical foundations of the many-valued generalization of a novel technique for simplifying large non-clausal formulas in propositional logic. This technique, called *removal of anti-links* (or just *anti-links*, for short) has been introduced for the two-valued case in (Ramesh *et al.*, 1997).

Possible applications of anti-links include computation of prime implicates[1] of large *non-clausal* formulas as required, for example, in logic design (Sasao, 1993) and diagnosis (de Kleer *et al.*, 1992).

Purely clausal approaches, applied after doing a polynomial time structure preserving clause form transformations (Plaisted and Greenbaum, 1986), cannot be used here, because such transformations do not preserve models. As a consequence, the set of prime implicates of the resulting clause set and of the original formula bear no straightforward relationship, see (Ramesh, 1995, Section 3.5.1) and (Ramesh *et al.*, to appear) for details.

In such settings often binary decision diagrams[2] (BDDs) (Bryant, 1986) are

---

* This research was supported in part within an Acción Integrada called "Discrete Function Manipulation Using Anti-Links" granted by DAAD (Germany) and M.E.C. (Spain).

[1] There is a strong duality between implicates and implicants. Therefore, all techniques presented in this paper can be used as well for the computation of prime implicants.

[2] Or, rather, *many-valued decision diagrams* (Srinivasan *et al.*, 1990) as the present paper deals

used. In contrast to these, anti-links do not compute any normal form of a given formula themselves, rather, they remove certain forms of redundancy from formulas in negation normal form (NNF, cf. Definition 1). Their main advantage is that no clausal normal form has to be computed in order to remove redundant parts of a formula. Although BDD implementations are storing subformulas in hash table to avoid multiple computations, a full BDD has to be computed for subsumption checking.

Viewing an NNF formula as a combinational circuit, using anti-links one can simplify circuits with unbounded nesting depth without having to compute a bounded depth circuit first. This can greatly reduce the size required for intermediate representations.

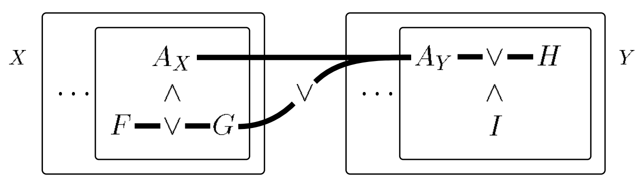
We stress, that anti-links are not intended to replace existing and successful techniques such as BDDs or dissolution (Murray and Rosenthal, 1993) (they are not even a complete inference rule for propositional logic), rather, the latter can be augmented and improved by our analysis.

In this paper, we define an anti-link operation on a generic language for expressing many-valued logic formulas called *signed NNF* and we show that all interesting properties of two-valued anti-links generalize to the many-valued setting, although in a non-trivial way.

Contrary to (Murray and Rosenthal, 1993) we do not use the special concept of *semantic graphs* for the representation of NNF formulas, but introduce an improved notation that solely relies on well-known notions like formulas, subformulas, etc.

Before giving the technical details, in the remainder of this section we briefly outline our results on an informal level.

Roughly, (two-valued) anti-links work as follows (see Sections 2 and 3 for all formal definitions): Consider the NNF formula below written down in a two-dimensional notation, where disjunctions are written horizontally and conjunctions are written vertically (F, G, H, and I, respectively, are arbitrary formulas, while A_X and A_Y are occurrences of the same literal p in the subformula X and in the subformula Y).



(1)

Let us call a maximal, disjunctively connected set of literal occurrences a path through a formula. Two of its paths are schematically displayed in (1).

Observe that all literals on any path through A_X, A_Y, and H occur as well on some path through F, G, A_Y, and H, because A_X and A_Y are occurrences of the same literal. In other words, the latter paths are all subsumed by one of the

with many-valued logic.

former paths, where subsumption on disjunctive paths coincides with the subset relation. Generalizing, we have this kind of situation, whenever

1. A_X and A_Y are two different occurrences of the same literal A in a formula,
2. A_X and A_Y are disjunctively connected, and
3. at least one of A_X and A_Y is a conjunct.

If (1)–(3) hold, then we call the pair $\langle A_X, A_Y \rangle$ a *redundant anti-link*.[3] A redundant anti-link thus always signifies the presence of subsumed paths in a formula. If the formula is converted to conjunctive normal form (CNF) such paths become non-prime implicates. It is, therefore, desirable to get rid of them.

The anti-link operator restructures a formula containing a redundant anti-link in such a way that exactly the subsumed paths are removed and, in addition, one occurrence of p is deleted on the (non-subsumed) paths where it occurs twice.

The result of applying the anti-link operator to (1) (without considering the parts of the formula indicated by "...") is displayed on the right. Observe that none of the paths containing $\{F, G, A_Y, H\}$ is present anymore.

$$\begin{array}{ccccc} F & \vee & G & \vee & I \\ & & \wedge & & \\ & & & H & \\ & A_X & \vee & \wedge & \\ & & & I & \end{array}$$

Of course, if the input formula is in (signed) CNF, the anti-link technique is applicable as well (the result, however, may not be in CNF anymore). It performs essentially a conjunctive factoring step, i.e., an application of the distributive law. While uncontrolled factoring is in general useless, the anti-link operator leads to a controlled application. Thus it can be very well beneficial to sacrifice CNF in intermediate steps.

In the many-valued case we still work with an NNF formula that is classical with respect to conjunctions and disjunctions. The difference comes in at the literal level: we use *signed* literals (sometimes called *universal* literals), that is expressions of the form S:p, where S is a subset of some truth value set and p is an atom.

It is not obvious how to extend the anti-link technique to the many-valued case; there are several possible approaches. Careful analysis reveals that, while condition (2) for anti-links can be left unchanged, conditions (1) and (3) have to be suitably altered.

1'. A_X and A_Y are two different occurrences of signed literals $S_X{:}p$ and $S_Y{:}p$, respectively, with $S_X \cap S_Y \neq \emptyset$.

The main idea for handling many-valued anti-links is to replace the occurrence of $S_Y{:}p$ with the equivalent formula $(S_X \cap S_Y){:}p \vee (S_Y \backslash S_X){:}p$, to replace the occurrence of $S_X{:}p$ with $(S_X \cap S_Y){:}p \vee (S_X \backslash S_Y){:}p$, and then to apply the classical results to the resulting two different occurrences of $(S_X \cap S_Y){:}p$. Obviously, making the replacement can destroy property (3), which must be changed as well; this is discussed in detail in Section 4.

[3] The phrase "anti-link" is motivated by the fact that a *link* is a pair of *complementary* and *conjunctively connected* literals.

In Section 5 we give an extended example. The paper is closed by pointing out the next stages of work.

2 Prerequisites and Concepts Related to Anti-Links

DEFINITION 1. Let Σ be a **propositional signature** that is a countable set of propositional variables $\{p, q, \ldots\}$ which are also called **atoms**. Let $N = \{i_1, \ldots, i_n\}$ be a finite set of **truth values** disjoint with Σ. If $p \in \Sigma$ and $S \subseteq N$, then the expression $S{:}p$ is called a **signed literal**.[4] Signed literals of the form $\emptyset{:}p$, respectively, $N{:}p$ are identified with the expressions *false*, respectively, *true*.

Signed formulas in negation normal form (NNF formulas, for short) are inductively defined as the smallest set with the following properties:

1. signed literals, and *true*, *false* are NNF formulas;
2. if $F_1, \ldots, F_m$ are NNF formulas, so are $F_1 \wedge \cdots \wedge F_m$ and $F_1 \vee \cdots \vee F_m$.

If $N = \{0, 1\}$, then we speak also of a **classical NNF formula**. In this case we abbreviate signed literals as follows: $\{0\}{:}p$ with $\overline{p}$, $\{1\}{:}p$ with p.

DEFINITION 2. The **subformulas** of an NNF formula G are defined as the smallest set having the following properties:

1. if G is a signed literal then its only subformula is G itself;
2. if $G = F_1 \wedge \cdots \wedge F_m$ ($G = F_1 \vee \cdots \vee F_m$) then, for any $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, m\}$, $F_{i_1} \wedge \cdots \wedge F_{i_r}$ ($F_{i_1} \vee \cdots \vee F_{i_r}$) is a subformula of G;
3. if F is a subformula of H and H is a subformula of G, then F is also a subformula of G.

DEFINITION 3. Let G, H be subformulas of an NNF formula F. We say that G and H are **disjunctively (conjunctively) connected**—d-(c-)connected, for short—if there exists a subformula $X \vee Y$ ($X \wedge Y$) of F such that G is a subformula of X and H is a subformula of Y.

A **partial disjunctive path** through an NNF formula F is a set of mutually d-connected occurrences of *true* and literals in F (occurrences of *false* are omitted). A **disjunctive path**—d-path, for short—through F is a partial d-path through F which is maximal and does not contain *true*. The set of all d-paths through an NNF formula F is denoted with $dp(F)$. **(Partial) conjunctive paths** are defined dually (using c- instead of d- and *true*, *false* exchanged). They are denoted $cp(F)$.

Observe that paths are defined as sets of *literal occurrences* A and do not contain the constants *true* and *false*. $\ell(A)$ denotes the literal of which A is an occurrence.

[4] As signed literals are the only kind of literals we deal with, we often simply say "literal" instead of "signed literal".

The set $\{\ell(A) \ : \ A \in \pi\}$ of *literals* on a path π is denoted with $\ell(\pi)$. One may think of a literal occurrence as a uniquely labelled subformula.

The above definition of paths is the same as in (Murray and Rosenthal, 1993; Ramesh *et al.*, 1997). In the following a different, but equivalent definition of paths through a formula will be convenient. As we make use of some results on paths contained in the papers mentioned above, we formally state their equivalence:

LEMMA 1. Let F be an NNF formula.

$$dp(F) = \begin{cases} \emptyset & \text{if } F = \mathit{true} \\ \{\emptyset\} & \text{if } F = \mathit{false} \\ \{\{F\}\} & \text{if } F \text{ is a literal} \\ \{\bigcup_{i=1}^{m} \pi_i \mid \pi_i \in dp(F_i) \text{ for } 1 \leq i \leq m\} & \text{if } F = F_1 \vee \cdots \vee F_m \\ \bigcup_{i=1}^{m} dp(F_i) & \text{if } F = F_1 \wedge \cdots \wedge F_m \end{cases}$$

$$cp(F) = \begin{cases} \{\emptyset\} & \text{if } F = \mathit{true} \\ \emptyset & \text{if } F = \mathit{false} \\ \{\{F\}\} & \text{if } F \text{ is a literal} \\ \bigcup_{i=1}^{m} cp(F_i) & \text{if } F = F_1 \vee \cdots \vee F_m \\ \{\bigcup_{i=1}^{m} \pi_i \mid \pi_i \in cp(F_i) \text{ for } 1 \leq i \leq m\} & \text{if } F = F_1 \wedge \cdots \wedge F_m \end{cases}$$

Proof. A straightforward induction on the depth of F. ∎

There are two different notions of subsumption: either paths are simply sets of literal occurrences or else the signs inside their literals are taken into account.

DEFINITION 4. Let π, π' be d-paths through a formula F. π **classically subsumes** π' iff $\ell(\pi) \subseteq \ell(\pi')$. π **MV-subsumes** π' iff for each $S{:}p \in \ell(\pi)$ there are $\{S_1{:}p, \ldots, S_m{:}p\} \subseteq \ell(\pi')$ such that $S \subseteq \bigcup_{i=1}^{m} S_i$.

Let F, G be NNF formulas. Then F **classically subsumes** G iff for each $\pi \in dp(G)$ there is a $\pi' \in dp(F)$ such that π' classically subsumes π. F **MV-subsumes** G iff for each $\pi \in dp(G)$ there is a $\pi' \in dp(F)$ such that π' MV-subsumes π.

A path or a formula **properly subsumes** (classically or MV-) another iff it subsumes the latter, but not vice versa.

A d-path is **tautological** iff it contains signed literals $S_1{:}p, \ldots, S_m{:}p$ such that $\bigcup_{i=1}^{m} S_i = N$.

Two NNF formulas are **classically (MV-)path equivalent** iff they classically (MV-)subsume each other.

It is obvious that classical subsumption (path equivalence) implies MV-subsumption (path equivalence).

EXAMPLE 1. Let $N = \{0, 1, 2, 3\}$. Consider d-paths $\pi = \{\{0,1\}{:}p, \{2\}{:}p, \{3\}{:}q\}$ and $\pi' = \{\{0,2\}{:}p\}$. Neither classically subsumes the other, but π' properly MV-subsumes π.

The NNF formula $F = \{0,1\}{:}p$ classically (and thus MV-) subsumes $G = (\{0\}{:}p \wedge \{3\}{:}q) \vee \{0,1\}{:}p$. G does not classically subsume F, but it MV-subsumes F. Hence, F and G are MV-, though not classically, path equivalent.

DEFINITION 5. Relative to a signature Σ and a truth value set N one defines an **(MV) interpretation** as a function $\mathbf{I} : \Sigma \to N$.

An interpretation $\mathbf{I}$ **satisfies** a signed literal $S{:}p$ if $\mathbf{I}(p) \in S$. It satisfies a d-(c-)path iff it satisfies at least one (all) of the literals occurring on it. No interpretation satisfies *false* and all interpretation satisfy *true*. Satisfaction is extended to complex NNF formulas in a natural way:

$$\mathbf{I} \text{ satisfies } F \text{ iff } \begin{cases} F = F_1 \wedge \cdots \wedge F_m \text{ and } \mathbf{I} \text{ satisfies all } F_i \\ F = F_1 \vee \cdots \vee F_m \text{ and } \mathbf{I} \text{ satisfies at least one } F_i \end{cases}$$

A formula is **satisfiable** iff there exists a satisfying interpretation for it. Two formulas are **logically equivalent** iff they are satisfied by exactly the same interpretations.

Observe that for classical NNF formulas our notion of satisfaction coincides with the usual one. The following lemma is obvious.

LEMMA 2. $\mathbf{I}$ satisfies an NNF formula F iff it satisfies all literals in one of its c-paths iff it satisfies at least one literal in each of its d-paths.

LEMMA 3. If two NNF formulas are classically or MV-path equivalent, then they are also logically equivalent.

Proof. Classical path equivalence implies MV-path equivalence, so assume the latter of F, G. We show that every interpretation that satisfies F also satisfies G, the other direction is symmetric.

Assume $\mathbf{I}$ satisfies F and $\mathbf{I}$ does not satisfy G. Then there is a d-path π through G which is not satisfied by $\mathbf{I}$. Because F MV-subsumes G, there is a d-path π' through F which MV-subsumes π.

By the previous lemma $\mathbf{I}$ satisfies at least one literal, say L_ψ, in each d-path ψ of F, in particular, $\mathbf{I}$ satisfies an $L_{\pi'} = S{:}p$ in π', hence $\mathbf{I}(p) \in S$. Because π' MV-subsumes π, there are $\{S_1{:}p, \ldots, S_m{:}p\} \subseteq \ell(\pi)$ such that $S \subseteq \bigcup_{i=1}^{m} S_i$. Thus $\mathbf{I}(p) \in S_i$ for some i. But then $\mathbf{I}$ satisfies a literal in π—contradiction. ∎

Given an occurrence of a subformula G of an NNF formula F and an NNF formula H, $F\{G \leftarrow H\}$ denotes the result of replacing this occurrence of G in F by H.

LEMMA 4. Let G be a subformula of F and let H be an MV-path equivalent of G. Then $F\{G \leftarrow H\}$ is an MV-path equivalent of F.

Let G be a subformula of F and let H be an NNF formula such that $dp(G) = dp(H)$. Then $dp(F\{G \leftarrow H\}) = dp(F)$.

Proof. Using Lemma 1 one proves with a straightforward induction on the formula structure using $dp(H)$ instead of $dp(G)$ preserves MV-path equivalence.

The second claim is an immediate consequence of Lemma 1. ∎

LEMMA 5. For all $S, S' \subseteq N$, and atoms p:

1. $(S \cup S')$:p is MV-path equivalent to S:$p \vee S'$:p;
2. $\emptyset$:p is MV-path equivalent to *false*;
3. N:p is MV-path equivalent to *true*.

Proof. Straightforward from the definitions. ∎

Finally we need some special terminology:

DEFINITION 6. Given an NNF formula F, a **subformula with respect to a set ρ of literal occurrences** is obtained from F by deleting all literal occurrences not in ρ.

Let G be a subformula of an NNF formula F. A d-path π in $dp(F)$ **passes through** an occurrence of G iff the subset of π which consists of literal occurrences in G is a d-path through G. c-paths passing through a formula occurrence are defined dually.

DEFINITION 7. Let G be an NNF formula. The **c-extension** and the **d-extension** of a subformula occurrence H in G, denoted by $CE(H)$ resp. $DE(H)$, are inductively defined as follows:

1. $CE(G) = DE(G) = G$.
2. If M is the occurrence of a conjunction $F_1 \wedge \ldots \wedge F_m$ $(m > 1)$ in G then
$$CE(F_i) = CE(M) \text{ and } DE(F_i) = F_i \qquad (1 \leq i \leq m) \ .$$
3. If M is the occurrence of a disjunction $F_1 \vee \ldots \vee F_m$ $(m > 1)$ in G then
$$CE(F_i) = F_i \text{ and } DE(F_i) = DE(M) \qquad (1 \leq i \leq m) \ .$$

Note, that the operators CE and DE have an implicit second argument that is always the entire formula G in which the first argument occurs. Contrary to that the operators CPE and DPE (see the following definition) have an explicit second argument, that does not have to be the entire formula.

DEFINITION 8. Let X and H be arbitrary occurrences of subformulas in an NNF formula.[5] The **c-path complement** of H with respect to X, written $CC(H, X)$, is the subformula of X with respect to all literals in X that lie on c-paths that do not pass through H. If no such literal exists, $CC(H, X) = \mathit{false}$. The **c-path extension** of H with respect to X, written $CPE(H, X)$, is the subformula of X containing all literals that lie on c-paths that pass through H. If no such literal exists, $CPE(H, X) = \mathit{false}$.

In the development of anti-link operations, we will use operations that are the duals of CC and CPE. We use DC for the **d-path complement** and DPE for the **d-path extension** operators. Their definitions are straightforward by duality (but note that then the base case is defined as $DC(H, X) = DPE(H, X) = \mathit{true}$).

EXAMPLE 2. In (2) on page 9,

$$\begin{aligned} DC(A_X, X) &= B & DC(A_Y, Y) &= E \vee C_Y \\ DPE(A_Y, Y) &= A_Y & CE(A_X) &= A_X \\ DC(CE(A_X), X)) &= B & DPE(A_X, X) &= A_X \vee C_X \\ CC(A_Y, Y) &= \mathit{false} \end{aligned}$$

3 Anti-Links in Two-Valued Logic

In this section we restate formally the discussion of the introduction on two-valued anti-links. It is partly taken from (Ramesh *et al.*, 1997), where also proofs of all the results in this section can be found. *All formulas in this section are classical NNF formulas. Likewise, subsumed means always classically subsumed, path equivalent means classically path equivalent, etc.*

DEFINITION 9. A **disjunctive (conjunctive) anti-link** is a pair $\langle A_X, A_Y \rangle$ of disjunctively (conjunctively) connected occurrences of the same literal $p = \ell(A_X) = \ell(A_Y)$ in an NNF formula F such that A_X occurs in X, A_Y occurs in Y, and $X \vee Y$ $(X \wedge Y)$ is a subformula of F.

In the rest of the paper we deal mainly with disjunctive anti-links; thus, when we write "anti-link" the intended meaning is always "disjunctive anti-link".

The following theorem relates subsumed paths to both kinds of anti-links. The theorem is immediate for classical CNF formulas; there is an obvious dual theorem regarding subsumed c-paths that is immediate for DNF formulas.

THEOREM 1. Let F be an NNF formula in which a non-tautological d-path π subsumes a distinct d-path π' in F. Then F contains either a disjunctive anti-link or a conjunctive anti-link.

[5] H usually is (but does not have to be) a subformula with respect to some set of literal occurrences of X.

3.1 Redundant Anti-links

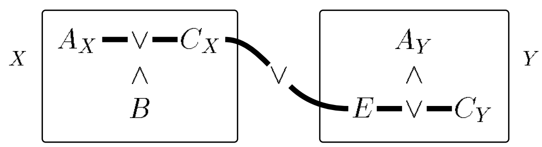
Unfortunately, the presence of anti-links does not imply the presence of subsumed paths, and hence the converse of the above theorem is not true.

It turns out, however, that it is possible to identify such disjunctive anti-links which do imply the presence of subsumed paths:

Definition 10. An anti-link $\langle A_X, A_Y \rangle$ is called **redundant** if $CE(A_X) \neq A_X$ or if $CE(A_Y) \neq A_Y$.

Definition 11. Let $\langle A_X, A_Y \rangle$ be an anti-link in F, where $M = X \vee Y$ is the smallest subformula of F containing the anti-link (the unique subformula of F containing the anti-link such that no proper subformula of M contains the anti-link). $DP(\langle A_X, A_Y \rangle, F)$ is defined as the set of all d-paths of M which pass through both $CE(A_X) - \{A_X\}$[6] and A_Y or through both $CE(A_Y) - \{A_Y\}$ and A_X.

Example 3. Consider the following formula $F = X \vee Y$:



(2)

The two occurrences of A in F form a redundant anti-link.

We proceed to show that $DP(\langle A_X, A_Y \rangle, F)$ consists solely of subsumed paths: Since $CE(A_X) - \{A_X\} = \mathit{true}$ there are no paths through it. Therefore, the only paths in $DP(\langle A_X, A_Y \rangle, F)$ are those which go through $CE(A_Y) - \{A_Y\} = E \vee C_Y$ and A_X. Since $DPE(A_X, X) = A_X \vee C_X$, there is only one such d-path, namely $\pi = \{A_X, C_X, E, C_Y\}$ (indicated by a line). π is subsumed by $\pi' = \{A_X, C_X, A_Y\}$ (with literal set $\{A, C\}$). In the example, the smallest subformula of F containing the anti-link is F itself. Notice that when F is a proper subformula of a formula G, then every d-path ψ in G containing π is subsumed by a corresponding d-path ψ' differing from ψ only in that ψ' contains π' instead of π.

In general, one or both of the literals in a redundant anti-link $\langle A_X, A_Y \rangle$ is an argument of a conjunction, and $DP(\langle A_X, A_Y \rangle, F) \neq \emptyset$. In the above example, the two occurrences of C are both arguments of disjunctions, and thus comprise a non-redundant anti-link for which accordingly $DP(\langle C_X, C_Y \rangle, F) = \emptyset$.

Although only redundant anti-links contribute directly to subsumed d-paths, non-redundant anti-links do not prohibit the existence of subsumed paths. However, such non-redundant anti-links do not themselves provide any evidence that such paths are in fact present.

Theorem 2. Let $\langle A_X, A_Y \rangle$ be a redundant anti-link in an NNF formula F. Then each d-path in $DP(\langle A_X, A_Y \rangle, F)$ is properly subsumed by a d-path through F that contains the anti-link.

[6] $CE(A_X) - \{A_X\}$ is used here and in the future as a shorthand for $CE(A_X)\{A_X \leftarrow \mathit{true}\}$.

3.2 An Anti-Link Operator

The identification of redundant anti-links can be done easily by checking to see if $CE(A_X) \neq A_X$ or $CE(A_Y) \neq A_Y$. After identifying a redundant anti-link, it is possible to remove it using the **disjunctive anti-link dissolvent** (DADV) operator defined below; in the process, all d-paths in $DP(\langle A_X, A_Y \rangle, F)$ are eliminated, and the two occurrences of the anti-link literal are collapsed into one.

Definition 12. Let $\langle A_X, A_Y \rangle$ be an anti-link and let $M = X \vee Y$ be the smallest subformula containing the anti-link. Then

$$DADV(\langle A_X, A_Y \rangle, M) \;=\; \begin{array}{c} DC(A_X, X) \;\vee\; DC(A_Y, Y) \\ \wedge \\ DC(CE(A_X), X) \;\vee\; DPE(A_Y, Y) \\ \wedge \\ DPE(A_X, X) \;\vee\; CC(A_Y, Y) \end{array}$$

Example 4. Consider again formula (2) from Example 3. In Example 2 we computed $DC(A_X, X)$ and $DC(A_Y, Y)$, so the upper conjunct in $DADV$ is $(B \vee E \vee C_Y)$. For the middle conjunct use $DC(CE(A_X), X)$ and $DPE(A_Y, Y)$ which yields $(B \vee A_Y)$. Finally, in the lower conjunct, $DPE(A_X, X)$ and $CC(A_Y, Y)$ give $(A_X \vee C_X)$. The result is:

$$DADV(A_X, A_Y, M) \;=\; \begin{array}{c} B \vee E \vee C_Y \\ \wedge \\ B \vee A_Y \\ \wedge \\ A_X \vee C_X \end{array}$$

We point out that although $DADV$ produces a CNF formula in the above simple example, in general it does not. In particular, the above formula can be simplified as the consequence of easily recognizable conditions, and the resulting formula is not in CNF. For the details, see Case 1 of Section 3.4.

3.3 Correctness of DADV

Theorem 3 below states that $DADV(\langle A_X, A_Y \rangle, F)$ is logically equivalent to F and does not contain the d-paths of $DP(\langle A_X, A_Y \rangle, F)$.

Theorem 3. Let $M = X \vee Y$ be the smallest subformula containing $\langle A_X, A_Y \rangle$, an anti-link in the NNF formula F. Then $DADV(\langle A_X, A_Y \rangle, M)$ is logically equivalent to M and $F\{M \leftarrow DADV(\langle A_X, A_Y \rangle, M)\}$, i.e., the result of applying the anti-link operator, differs in d-paths from F as follows: d-paths in $DP(\langle A_X, A_Y \rangle, F)$ are not present, and any d-path of F containing the anti-link is replaced by a path with the same literal set having only one occurrence of the anti-link literal.

Theorem 3 gives us a method to remove anti-links and some subsumed d-paths: Simply identify a redundant anti-link $\langle A_X, A_Y \rangle$ and the smallest subformula M containing it, and then replace M by $DADV(\langle A_X, A_Y \rangle, M)$. The cost of performing $DADV(\langle A_X, A_Y \rangle, M)$ is proportional to the size of the formula replacing M, and this is linear in M. Also, c-connected literals in M do not become d-connected in $DADV(\langle A_X, A_Y \rangle, M)$. Thus truly new disjunctive anti-links are not introduced. However, parts of the formula may be duplicated, and this may give rise to additional copies of anti-links not yet removed.

Nevertheless, persistent removal of disjunctive anti-links is a terminating process, because at each step

1. if the removed anti-link is redundant (in which case $DP(\langle A_X, A_Y \rangle, M) \neq \emptyset$), then the number of d-paths is strictly reduced;
2. else, if the anti-link is not redundant, then the d-paths in the formula remain unchanged with the exception of those going through the anti-link on which one literal occurrence is deleted.

This proves:

THEOREM 4. Finitely many applications of the $DADV$ operation will result in a formula without disjunctive anti-links, and termination of this process is independent of the choice of anti-link at each step.

Although we can remove all the redundant disjunctive anti-links in the formula, this process can introduce new conjunctive anti-links. Such anti-links may indicate the presence of subsumed d-paths, but the sufficient requirement for redundancy is much stronger as in Definition 10, see (Ramesh *et al.*, 1997, Section 3.7).

3.4 Simplifications

Obviously, $DADV(\langle A_X, A_Y \rangle, M)$ can be syntactically larger than $M = X \vee Y$. Under certain conditions we may use simplified alternative definitions for $DADV$. These definitions result in formulas which are syntactically smaller than those that result from the general definition. The following is a list of possible simplifications.

1. If
$$CE(A_X) = A_X \qquad (\text{and } CE(A_X) \neq X) \ ,$$
then $DC(CE(A_X), X) = DC(A_X, X)$. Therefore by (possibly non atomic) factoring on $DC(A_X, X)$ and observing that $(DC(A_Y, Y) \wedge DPE(A_Y, Y))$ has the same d-paths as Y, $DADV(\langle A_X, A_Y \rangle, M)$ becomes
$$\begin{array}{rcl} DC(A_X, X) & \vee & Y \\ & \wedge & \\ DPE(A_X, X) & \vee & CC(A_Y, Y) \end{array}$$

It turns out that this rule applies to (2) in Example 3; the simplified rule for this case results in the following formula:

$$\begin{array}{c} B \vee \begin{array}{c} A \\ \wedge \\ E \vee C \end{array} \\ \wedge \\ A \vee C \end{array}$$

2. If
$$CE(A_X) = X \ ,$$
then $DC(CE(A_X), X) = true$, $DPE(A_X, X) = A_X$ and $DC(A_X, X) = (X - \{A_X\})$. Hence $DADV(\langle A_X, A_Y \rangle, M)$ becomes
$$\begin{array}{c} X - \{A_X\} \vee DC(A_Y, Y) \\ \wedge \\ A_X \vee CC(A_Y, Y) \end{array}$$

3. If both Case 1 and Case 2 apply, then $CE(A_X) = X = A_X$, and the above formula simplifies to
$$A_X \vee CC(A_Y, Y) \ .$$

Note that in all the above versions of $DADV$, the rôles of X and Y can be interchanged.

4 Anti-Links in Many-Valued Logic

By definition, an anti-link in classical logic consists of two occurrences of the same literal. In many-valued logics the definition has to be more general, because there are redundancies as well if literals are not identical but consist of the same propositional variable and non-disjoint truth signs:

Definition 13. A **disjunctive (conjunctive) many-valued anti-link** consists of disjunctively (conjunctively) connected occurrences A_X and A_Y of literals in a many-valued formula in NNF such that

1. $\ell(A_X) = S_X{:}p$ and $\ell(A_Y) = S_Y{:}p$ for some atom p and $S_X, S_Y \subseteq N$;
2. $S_X \cap S_Y \neq \emptyset$.

4.1 Redundant Anti-Links in Many-Valued Logics

The analogue of Theorem 1 holds for many-valued anti-links, i.e., if a formula contains subsumed d-paths this implies the presence of anti-links; and the converse of the theorem is not true: only *redundant* anti-links indicate the existence of subsumed d-paths.

The classical anti-link operator, when applied to a redundant anti-link, reduces a formula in two ways. First, if the anti-link literal A_X is a conjunct, d-paths that go through the other anti-link literal A_Y and through $CE(A_X) - \{A_X\}$ are removed (cf. Figure 1 on page 18). These paths are of the form

$$\pi = \pi_{CE} \cup \pi_r \cup \{A_Y\}$$

(where π_{CE} is the part going through $CE(A_X) - \{A_X\}$ and π_r is the rest of the path except A_Y). Such a path π is classically subsumed by a path

$$\pi' = \{A_X\} \cup \pi_r \cup \{A_Y\}$$

in the formula, because $\ell(\pi') \subseteq \ell(\pi)$ for a classical anti-link. π' is identical to π except that it goes through A_X instead of $CE(A_X) - \{A_X\}$ (and, thus, through both anti-link literals).

In the many-valued case, where $\ell(A_X) = S_X{:}p$ and $\ell(A_Y) = S_Y{:}p$, this type of reduction is possible iff $S_X \subseteq S_Y$, because then a path $\pi' = \{A_X\} \cup \pi_r \cup \{A_Y\}$ MV-subsumes a path $\pi = \pi_{CE} \cup \pi_r \cup \{A_Y\}$. The same type of reduction can be found if A_Y is a conjunct instead of A_X and—in the many-valued case—provided $S_Y \subseteq S_X$.

These considerations justify the following definitions:

Definition 14. A many-valued anti-link $\langle A_X, A_Y\rangle$, where $\ell(A_X) = S_X{:}p$ and $\ell(A_Y) = S_Y{:}p$, is **redundant** if either one of the following conditions holds:

- A_X is a conjunct, i.e. $CE(A_X) \neq A_X$, and $S_X \subseteq S_Y$
- A_Y is a conjunct, i.e. $CE(A_Y) \neq A_Y$, and $S_Y \subseteq S_X$.

Definition 15. Let $\langle A_X, A_Y\rangle$ be a many-valued anti-link in F, where $\ell(A_X) = S_X{:}p$ and $\ell(A_Y) = S_Y{:}p$, and $M = X \vee Y$ is the smallest subformula containing the anti-link. Then

$$DP_{MV}(\langle A_X, A_Y\rangle, F) = \begin{cases} DP(\langle A_X, A_Y\rangle, F) & \text{if } S_X = S_Y \\ \{\pi \in dp(F) \mid \pi \text{ passes through} & \\ \qquad CE(A_X) - \{A_X\} \text{ and } A_Y\} & \text{if } S_X \subsetneq S_Y \\ \{\pi \in dp(F) \mid \pi \text{ passes through} & \\ \qquad CE(A_Y) - \{A_Y\} \text{ and } A_X\} & \text{if } S_Y \subsetneq S_X \\ \emptyset & \text{otherwise} \end{cases}$$

The following theorem is the many-valued version of Theorem 2 (and the proof for Theorem 2 given in (Ramesh *et al.*, 1997) can easily be adapted):

THEOREM 5. Let $\langle A_X, A_Y \rangle$ be a redundant many-valued anti-link in an NNF formula F. Then each d-path in $DP_{MV}(\langle A_X, A_Y \rangle, F)$ is properly MV-subsumed by a d-path through F that contains the anti-link.

The second type of reduction of the (classical) anti-link operator is to remove one anti-link literal occurrence A_Y from all paths that go through both anti-link literals A_X and A_Y, which is justified by the fact that $\ell(A_X) = \ell(A_Y)$. In the many-valued case this second reduction is only possible if $S_Y \subseteq S_X$ or $S_X \subseteq S_Y$.

4.2 A Many-Valued Anti-Link Operator

It is not obvious how to extend the anti-link technique to the many-valued case; there are several possible approaches. Careful analysis shows that the following is a successful method for developing a many-valued anti-link operator for simplifying signed NNF formulas from the classical operator.

The following assertions are obvious for all sets S_X and S_Y of truth values:

1. $S_X = (S_X \cap S_Y) \cup (S_X \setminus S_Y)$.
2. $S_X{:}p$ is MV-path equivalent to $(S_X \cap S_Y){:}p \vee (S_X \setminus S_Y){:}p$.
3. If $S_X \subseteq S_Y$, then $S_X{:}p$ is identical to $(S_X \cap S_Y){:}p$.

Therefore, given a formula F in NNF that contains an anti-link $\langle A_X, A_Y \rangle$, where $\ell(A_X) = S_X{:}p$, $\ell(A_Y) = S_Y{:}p$, the result of replacing A_X by $(S_X \cap S_Y){:}p \vee (S_X \setminus S_Y){:}p$ if $S_X \not\subseteq S_Y$ and replacing $S_Y{:}p$ by $(S_X \cap S_Y){:}p \vee (S_Y \setminus S_X){:}p$ if $S_Y \not\subseteq S_X$ is a formula F' that is MV-path equivalent to F.

F' contains a classical anti-link: the two occurrences of $(S_X \cap S_Y){:}p$. Thus, the classical anti-link operator can be applied to F' if F' is viewed as a *classical* NNF formula over the signature consisting of the many-valued literals (including their signs) that occur in F'.

The result of this application is a formula F'' that is classically path equivalent to F' and thus MV-path equivalent to F. By definition of the classical anti-link operator, F'' is constructed by replacing the smallest subformula M' in F' containing the anti-link by $M'' = DADV(\langle A_X, A_Y \rangle, M')$.

M'' (and thus F'') can be expressed in terms of the original formula; the result is a d-path equivalent formula that can be seen as the result of applying a many-valued anti-link operator to the original formula, and in fact we use it as the definition of our operator:

DEFINITION 16. Let $\langle A_X, A_Y \rangle$ be a many-valued anti-link in an NNF formula F, where $\ell(A_X) = S_X{:}p$ and $\ell(A_Y) = S_Y{:}p$, and let $M = X \vee Y$ be the smallest subformula of F containing the anti-link. Then

$$MVDADV(\langle A_X, A_Y\rangle, M) \;=\; \begin{array}{rcl} DC(A_X,X) & \vee & DC(A_Y,Y) \\ & \wedge & \\ T & \vee & DPE(A_Y,Y) \\ & \wedge & \\ DPE(A_X,X) & \vee & S \end{array}$$

where

$$T = \begin{cases} DC(CE(A_X),X) & \text{if } S_X \subseteq S_Y \\ DC(A_X,X) & \text{otherwise} \end{cases}$$

and

$$S = \begin{cases} CC(A_Y,Y) & \text{if } S_Y \subseteq S_X \\ Y\{A_Y \leftarrow (S_Y \setminus S_X){:}p\} & \text{otherwise} \end{cases}$$

As in the classical case, the cost of computing $MVDADV(\langle A_X, A_Y\rangle, M)$ is linear in M.

The following theorem, that states correctness of the many-valued anti-link operator is the analogue of Theorem 3.

THEOREM 6. Let $M = X \vee Y$ be the smallest subformula containing $\langle A_X, A_Y\rangle$, a many-valued anti-link in the NNF formula F, where $\ell(A_X) = S_X{:}p$ and $\ell(A_Y) = S_Y{:}p$. Then $MVDADV(\langle A_X, A_Y\rangle, M)$ is MV-path equivalent to M and differs in d-paths from M as follows:

1. d-paths in $DP_{MV}(\langle A_X, A_Y\rangle, M)$ are not present;
2. any d-path π of M containing the anti-link is replaced by $\pi' = \pi \setminus \{A_Y\}$ if $S_Y \subseteq S_X$, and else by $\pi' = (\pi \setminus \{A_Y\}) \cup \{A_Y^{\setminus}\}$, where $A_Y^{\setminus}$ is an occurrence of $(S_Y \setminus S_X){:}p$.

Proof. The proof follows closely the description given at the beginning of this section of how the many-valued anti-link operator can be constructed from the classical one.

1st case: $S_X = S_Y$
In this case, where both $S_X \subseteq S_Y$ and $S_Y \subseteq S_X$, the definition of $MVDADV$ is identical to that of $DADV$, and the theorem follows immediately from the correctness of the classical operator (Theorem 3).

2nd case: $S_X \subsetneq S_Y$
The set of paths in M can be separated into four disjoint subsets: the set of paths that (a) do not go through A_Y, (b) go through A_Y but not through $CE(A_X)$, (c) go through A_Y and through $CE(A_X) - \{A_X\}$, (d) go through both anti-link literals A_X and A_Y (cf. Figure 1).

Because of $S_X \subsetneq S_Y$ the formula M' is constructed from M by replacing A_Y by $(S_X \cap S_Y){:}p \vee (S_Y \setminus S_X){:}p$. Paths that do not go through A_Y remain unchanged;

in paths that contain A_Y this is replaced by occurrences $A_Y^{\cap}$ of $(S_X \cap S_Y){:}p$ and $A_Y^{\setminus}$ of $(S_Y \setminus S_X){:}p$. Thus, $dp(M') = (a) \cup (b') \cup (c') \cup (d')$ where

$$\begin{aligned}(b') &= \{(p \setminus \{A_Y\}) \cup \{A_Y^{\cap}, A_Y^{\setminus}\} \mid p \in dp(M) \text{ goes through } A_Y \\ &\qquad\qquad \text{and not through } CE(A_X)\} \\ (c') &= \{(p \setminus \{A_Y\}) \cup \{A_Y^{\cap}, A_Y^{\setminus}\} \mid p \in dp(M) \text{ goes through } A_Y \\ &\qquad\qquad \text{and through } CE(A_X) - A_X\} \\ (d') &= \{(p \setminus \{A_Y\}) \cup \{A_Y^{\cap}, A_Y^{\setminus}\} \mid p \in dp(M) \text{ goes through } A_Y \\ &\qquad\qquad \text{and through } A_X\}\end{aligned}$$

Because $A_Y^{\cap}$ is part of a disjunction, $CE(A_Y^{\cap}) = A_Y^{\cap}$ in M'; therefore the set (c') is identical to $DP(\langle A_X, A_Y^{\cap}\rangle, M')$. This means that (c') is—according to Theorem 3—the set of paths that is removed when the classical anti-link operator is applied to the anti-link $\langle A_X, A_Y^{\cap}\rangle$ in M' (recall that $S_X{:}p$ is identical to $(S_X \cap S_Y){:}p$ by $S_X \subseteq S_Y$). In addition, by applying the classical operator, the second occurrence $A_Y^{\cap}$ is removed from the paths going through the anti-link, i.e., from all paths in (d'). The set of paths in the result M'' of applying $DADV$ to M' is thus $dp(M'') = (a) \cup (b') \cup (d'')$ where

$$\begin{aligned}(d'') = \{(p \setminus \{A_Y\}) \cup \{A_Y^{\setminus}\} \mid p \in dp(M) &\text{ goes through } A_Y \\ &\text{and through } A_X\}\end{aligned}$$

According to the definition of the classical anti-link operator, M'' has the form

$$M'' = \begin{array}{ccc} DC(A_X, X') & \vee & DC(A_Y^{\cap}, Y') \\ & \wedge & \\ DC(CE(A_X), X') & \vee & DPE(A_Y^{\cap}, Y') \\ & \wedge & \\ DPE(A_X, X') & \vee & CC(A_Y^{\cap}, Y') \end{array}$$

Because

1. $X = X'$;
2. $DC(A_Y^{\cap}, Y')$ is identical to $DC(A_Y, Y)$ since the disjunctive complement DC consists of those paths that do not contain the anti-link literal;
3. $CC(A_Y^{\cap}, Y') = Y\{A_Y \leftarrow (S_Y \setminus S_X){:}p\}$ using the definition of the conjunctive complement and since $Y' = Y\{A_Y \leftarrow ((S_X \cap S_Y){:}p \vee (S_Y \setminus S_X){:}p)\}$;

the only difference between M'' and $MVDADV(\langle A_X, A_Y\rangle, M)$ is that M'' contains the subformula $DPE(A_Y^{\cap}, Y')$ in its middle part instead of $DPE(A_Y, Y)$. This difference only affects the paths in the subset (b') of $dp(M'')$. Instead of the two occurrences $A_Y^{\cap}$ and $A_Y^{\setminus}$ they contain A_Y in $MVDADV(\langle A_X, A_Y\rangle, M)$.

This, finally, shows that $dp(MVDADV(\langle A_X, A_Y\rangle, M))$ consists of the paths in (a), (b), and (d''). The paths in $DP_{MV}(\langle A_X, A_Y\rangle, M) = (c)$ have been removed, and in the paths in (d) (the paths going through $\langle A_X, A_Y\rangle$) the occurrence A_Y has been replaced by $A_Y^{\setminus}$; this concludes the proof of the second part of the theorem for this case.

It remains to be shown that $MVDADV(\langle A_X, A_Y\rangle, M)$ is MV-path equivalent to M: This, however, is obvious using Theorem 5 and the fact that for any ψ paths $\pi = \psi \cup A_X \cup A_Y$, $\pi' = \psi \cup A_X \cup A_Y^{\setminus}$ subsume each other provided $S_X \subseteq S_Y$.

3rd case: $S_Y \subsetneq S_X$
The proof for this subcase proceeds analogously to that for the previous subcase. The only differences are:

- If $S_Y \subsetneq S_X$, then $Y' = Y$ and therefore $CC(A_Y, Y') = CC(A_Y, Y)$, etc.
- A_X is replaced by the disjunction $(S_X \cap S_Y){:}p \vee (S_Y \setminus S_X){:}p$ to construct M' from M. Therefore, $A_X^{\cap}$ (the occurrence of $(S_X \cap S_Y){:}p$ in M') is a disjunct. This implies $CE(A_X^{\cap}) = A_X^{\cap}$ and $DC(CE(A_X^{\cap}), X') = DC(A_X, X)$.

4th case: otherwise
The proof for this subcase is a combination of the proofs for the two previous subcases. ∎

Observing the definition of disjunctive paths, the result of Theorem 6 for the smallest subformula M containing the anti-link can easily be extended to any formula containing an anti-link.

Corollary. Let $M = X \vee Y$ be the smallest subformula containing $\langle A_X, A_Y\rangle$, a many-valued anti-link in the NNF formula F. Then the result

$$F\{M \leftarrow MVDADV(\langle A_X, A_Y\rangle, M)\}$$

of applying the many-valued anti-link operator to F is MV-path equivalent to F and differs in d-paths from F in the same way as $MVDADV(\langle A_X, A_Y\rangle, M)$ differs from M.

As in the classical case iterative application of the many-valued anti-link operator is a terminating process:

THEOREM 7. Finitely many applications of the $MVDADV$ operation will result in a formula without many-valued disjunctive anti-links, and termination of this process is independent of the choice of anti-link at each step.

Proof. We use the following complexity measure $|\cdot|$ for the size of a many-valued formula F, that in the classical case is identical to the sum of the lengths of all d-paths of F:

$$|F| = \sum_{\pi \in dp(F)} \sum_{S:p \in \pi} |S| \ ,$$

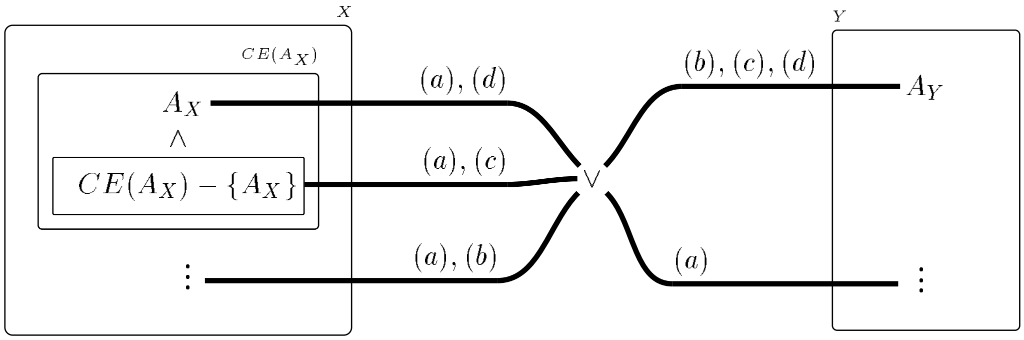


Fig. 1. The different types of paths if $S_X \subsetneq S_Y$ (see proof of Theorem 6).

where $|S|$ is the cardinality of S. This measure is finite for all formulas in *finitely*-valued logics.[7]

The second part of the corollary implies

$$|F\{M \leftarrow MVDADV(\langle A_X, A_Y\rangle, M)\}| < |F| \ ,$$

even if the anti-link is not redundant (note that $S_X \cap S_Y \neq \emptyset$ by Definition 13).

This implies the termination of the process of applying the anti-link operator iteratively, because at each step the complexity measure strictly decreases. ∎

Since the anti-link operator is not symmetric, there are always two possibilities for its application (by interchanging A_X and A_Y). How to choose is not obvious; note that in both cases the number of d-paths in the result is the same. Other things have to be considered, for example the syntactic size of the result. In general, applications are preferable that make use of the simplified versions of $MVDADV$ described in the following subsection.

4.3 Simplifications

Similar to the classical operator (see Section 3.4), the $MVDADV$ operator can be simplified in certain cases. Here S and T are the same subformulas as in Definition 16.

1. If
$$S_X \not\subseteq S_Y \text{ or } CE(A_X) = A_X \ ,$$
then
$$MVDADV(\langle A_X, A_Y\rangle, M) = \begin{array}{c} DC(A_X, X) \ \vee \ Y \\ \wedge \\ DPE(A_X, X) \ \vee \ S \end{array}$$

[7] The theorem holds for infinitely-valued logics as well; to prove this, however, a more elaborate complexity measure has to be used.

2. If

$$S_X \subseteq S_Y \text{ and } CE(A_X) = X \ ,$$

then

$$MVDADV(\langle A_X, A_Y\rangle, M) = \begin{array}{c} X - \{A_X\} \ \vee \ DC(A_Y, Y) \\ \wedge \\ A_X \ \vee \ S \end{array}$$

3. If

$$CE(A_X) = A_X = X \ ,$$

then

$$MVDADV(\langle A_X, A_Y\rangle, M) = A_X \ \vee \ S$$

5 Extended Example

We apply the many-valued anti-link operator to the formula

$$F \ = \ \begin{array}{ccc} \begin{array}{l} S_1{:}p \\ \ \wedge \quad \vee \ C \\ B \\ \qquad \wedge \\ \qquad D \end{array} & \vee & \begin{array}{c} S_2{:}p \ \vee \ G \\ \wedge \\ H \end{array} \end{array}$$

F contains six paths and seven literals. In the left parts of Figures 2 and 3 the result of applying the many-valued anti-link operator to $\langle A_X, A_Y\rangle$ is shown, where $\ell(A_X) = S_1{:}p$ and $\ell(A_Y) = S_2{:}p$, i.e., $F = X \vee Y$. The formulas on the right are the result when the second possibility is used, where $\ell(A_X) = S_2{:}p$ and $\ell(A_Y) = S_1{:}p$, i.e., $F = Y \vee X$.

If $S_1 \subseteq S_2$, $S_2 \not\subseteq S_1$ the anti-link is redundant. By applying the many-valued operator, the MV-subsumed path $\{B, C, A_Y, G\}$ (resp. $\{B, C, A_X, G\}$) is removed. In case $\ell(A_X) = S_2{:}p$, $\ell(A_Y) = S_1{:}p$, the first simplified version of $MVDADV$ (see Section 4.3) can be used. The two possible results are shown in Figure 2. They both have the same five paths. However, the formula on the right, that results from using the simplified version of $MVDADV$ is syntactically smaller: it consists of nine instead of twelve literals.

If $S_2 \subseteq S_1$, $S_1 \not\subseteq S_2$, the anti-link is not redundant, and the number of paths is not reduced. The formula on the left in Figure 3, that is the result of applying $MVDADV$ if $\ell(A_X) = S_1{:}p$, $\ell(A_Y) = S_2{:}p$, is syntactically smaller, because in that case the occurrence A_Y can be removed from paths going through the anti-link. In the formula on the right A_Y has been replaced by $(S_Y \setminus S_X){:}p$ in paths through the anti-link.

The two possible results of applying the anti-link operator to either $\langle A_X, A_Y\rangle$ or to $\langle A_Y, A_X\rangle$ have always identical d-paths (except the one going through the anti-link). However, as the example shows, they can be quite different syntactically.

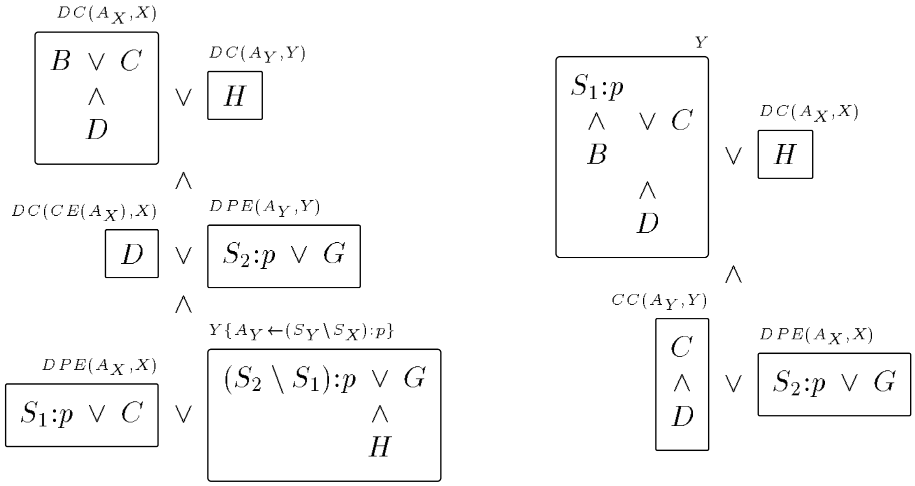


Fig. 2. The two possible results of applying the anti-link operator to F if $S_1 \subseteq S_2$.

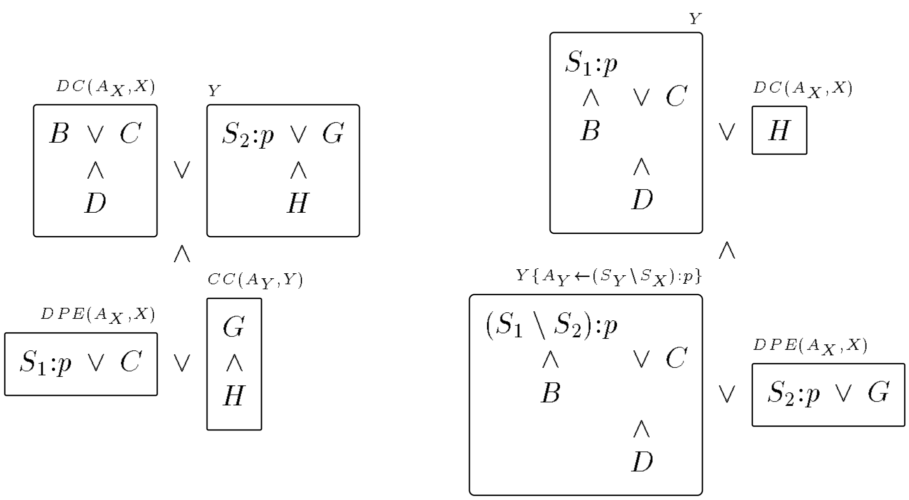


Fig. 3. The two possible results of applying the anti-link operator to F if $S_2 \subseteq S_1$, $S_1 \not\subseteq S_2$.

Here the result is larger than the original formula F, but in general it does not have to be; and in all cases $MVDADV(\langle A_X, A_Y\rangle, F)$ is much smaller than the result of transforming F to disjunctive normal form which contains 19 literals.

Summary and Future Work

We extended the concept anti-links from classical to many-valued logic and defined a many-valued anti-link operator. This operator can be employed so as to strictly reduce the number of d-paths in a many-valued NNF formula. Anti-link operations remove subsumed paths without any direct subsumption checks. This is significant for computing prime implicates, since such computations tend to be dominated by subsumption checks.

Anti-link techniques are not restricted to many-valued formulas in NNF. Principally, they can be adapted to work with other normal forms as well, for example, XOR-normal form (Sasao, 1993) or normal forms based on T-norms and S-norms (Gottwald, 1993), as well as with other logics such as modal logics. Necessary conditions are that path subsumption and the subset relation coincide, and that an adequate distributivity law can be formulated for the chosen logical connectives. The details will be subject of forthcoming work.

ACKNOWLEDGEMENT

We would like to thank N. Murray for his valuable comments on an earlier version of this paper and K. Geiss for finding several typos.

References

Randal E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35:677–691, 1986.

Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2–3):192–222, 1992.

Siegfried Gottwald. *Fuzzy Sets and Fuzzy Logic*. Vieweg, Braunschweig, 1993.

Neil V. Murray and Erik Rosenthal. Dissolution: Making Paths Vanish. *Journal of the ACM*, 3(40):504–535, 1993.

David A. Plaisted and Steven Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2:293–304, 1986.

Anavai Ramesh, Bernhard Beckert, Reiner Hähnle, and Neil V. Murray. Fast subsumption checks using anti-links. *Journal of Automated Reasoning*, 18(1):47–84, 1997.

Anavai Ramesh, George Becker, and Neil V. Murray. CNF and DNF considered harmful for computing prime implicants/implicates. *Journal of Automated Reasoning*, to appear.

Anavai G. Ramesh. *Some Applications of Non-Clausal Deduction*. PhD thesis, Department of Computer Science, State University of New York at Albany, 1995.

Tsutomu Sasao, editor. *Logic Synthesis and Optimization*. Kluwer, Norwell/MA, USA, 1993.

A. Srinivasan, T. Kam, S. Malik, and R. E. Brayton. Algorithms for discrete function manipulation. In *Proceedings, IEEE International Conference on CAD, Santa Clara/CA, USA*, pages 92–95. IEEE Press, 1990.