Path Dissolution: A Strongly Complete Rule of Inference †

Neil V. Murray
Dept. of Computer Science
State Univ. of N.Y. at Albany
Albany, NY 12222

Erik Rosenthal
Dept. of Computer Science
Wellesley College
Wellesley, MA 02181

Abstract

We introduce *path dissolution*, a rule of inference that operates on formulas in negation normal form. Path dissolution is *strongly complete*; i.e., it has the property that, given an unsatisfiable ground formula, any sequence of dissolution steps will produce the empty graph. This is accomplished by strictly reducing (at each step) the number of c-paths in the formula.

Dissolution, unlike most resolution-based inference rules, does not directly lift into first-order logic; techniques for employing dissolution at the first order level are discussed.

1. Introduction

We introduce *path dissolution*, a rule of inference that operates on quantifier-free predicate calculus formulas in negation normal form (NNF). We use techniques first developed in [Murray & Rosenthal 1985a], and in [Murray & Rosenthal 1987] employing a representation of formulas that we call *semantic graphs*. Dissolution is a generalization to NNF of the Prawitz matrix reduction rule, which operates on formulas in conjunctive normal form (CNF). One important distinction between dissolution and most other rules of inference is that one *cannot* restrict attention to CNF: A single application of dissolution generally produces a formula that is not in CNF *even if the original formula is*.

For almost a decade, the connection-graph resolution procedure had been conjectured to be *strongly complete*, i.e., to converge under *any* sequence of inferences for all contradictory ground formulas. Norbert Eisinger [Eisinger 1986] recently discovered counterexamples. Path dissolution is strongly complete: Each dissolution step strictly reduces the number of c-paths in a formula. The procedure always terminates, producing (in effect) a list of the formula's models. (If the formula is unsatisfiable, the empty graph results, representing the empty set of models.)

Bibel has presented several algorithms for determining whether a propositional formula is unsatisfiable [Bibel 1982]. He built on the work of Prawitz [Prawitz 1970] and later work of his own [Bibel 1979], [Bibel 1981] and of Andrews [Andrews 1981]. His approach was to search for paths containing links (complementary literals). The technique developed in this paper also employs links, but they are used to *remove* the paths through them.

Dissolution, unlike most resolution-based inference rules, does not directly lift into first-order logic; techniques for employing dissolution at the first order level are discussed.

Dissolution is quite different from other rules of inference, which is not surprising in view of its strong completeness and of the fact that it forces formulas away from CNF. As a result, we omit proofs and present extensive examples.

2. Preliminaries

We briefly summarize semantic graphs, including only those results that are necessary for the analysis of path dissolution. We assume the reader to be familiar with the notions of *atom, literal, formula, resolution,* and *unification*. We will consider only quantifier-free formulas in which all negations are at the atomic level.

A *semantic graph* is empty, a single node, or a triple (N,C,D) of *nodes*, *c-arcs*, and *d-arcs*, respectively, where a node is a literal occurrence, a c-arc is a conjunction of two non-empty semantic graphs, and a d-arc is a disjunction of two non-empty semantic graphs. Each semantic graph used in the construction of a semantic graph will be called an *explicit subgraph*. We use the notation $(G,H)_c$ for the c-arc between G and H and similarly use $(G,H)_d$ for a d-arc. We will consider an empty graph to be an empty disjunction, which is a contradiction. If $G = (X, Y)$, observe that every other arc is an arc in X or in Y; we call (X, Y) the *final arc* of G.

As an example, the formula

$$((A \wedge B) \vee C) \ \wedge \ (\sim A \vee (D \wedge C))$$

is the graph

$$\begin{array}{ccc} A \rightarrow B & & \bar{A} \\ \downarrow & \rightarrow & \downarrow \\ C & & D \rightarrow C \end{array}$$

Note that horizontal arrows are c-arcs, and vertical arrows are d-arcs.

The formulas we are considering are in *negation normal form* (NNF) in that all negations are at the atomic level; the only connectives used are AND and OR.

† This research was supported in part by the National Science Foundation under grant DCR-8600848.

Lemma 1. Let G be a semantic graph, and let A and B be nodes in G. Then there is a unique arc connecting A and B.

One of the keys to our analysis is the notion of *path.* Let G be a semantic graph. A *partial c-path through G* is a set c of nodes such that any two are connected by a c-arc. A *c-path* is a partial c-path that is not properly contained in any partial c-path. We similarly define d-path using d-arcs instead of c-arcs. Several other authors have employed paths; for example see [Andrews 1981], [Bibel 1979], [Bibel 1981], [Bibel 1982], [Eisinger 1986], [Murray 1982], [Prawitz 1970]. They generally concentrated on c-paths.

Lemma 2. Let G be a semantic graph. Then an interpretation I satisfies (falsifies) G iff I satisfies (falsifies) every literal on some c-path (d-path) through G.

We will frequently find it useful to consider subgraphs that are not explicit; that is, given any set of nodes, we would like to define that part of the graph that consists of exactly the given set of nodes. The previous example is shown below on the left. The subgraph relative to the set {A, $\bar{A}$, D} is the graph on the right.

$$\begin{array}{ccc} \mathbf{A} \rightarrow \mathrm{B} & & \bar{\mathbf{A}} \\ \downarrow & \rightarrow & \downarrow \\ \mathrm{C} & & \mathbf{D} \rightarrow \mathrm{C} \end{array} \qquad\qquad \begin{array}{ccc} & & \bar{\mathbf{A}} \\ \mathbf{A} & \rightarrow & \downarrow \\ & & \mathbf{D} \end{array}$$

If N is the node set of a graph G, and if $N' \subset N$, we define $G_{N'}$, *the subgraph of G relative to* N', as follows: If $N' = N$, then $G_{N'} = G$. If the final arc of G is (X, Y), and if no node in N' appears in Y (or in X), then $G_{N'} = X_{N'}$ (or $G_{N'} = Y_{N'}$). Otherwise, $G_{N'} = (X_{N'}, Y_{N'})$, where this arc is of the same type as (X, Y). In practice, we typically will not distinguish between N' and $G_{N'}$.

A *c-block C* is a subgraph of a semantic graph with the property that any c-path p that includes at least one node from C passes through C; that is, the subset of p consisting of the nodes that are in C is a c-path through C. A *d-block* is similarly defined with d-paths, and a *full block* is a subgraph that is both a c-block and a d-block. We define *a strong c-block* in a semantic graph G to be a subgraph C of G with the property that every c-path through G contains a c-path through C. A *strong d-block* is similarly defined.

The *fundamental subgraphs* of a semantic graph G are defined recursively as follows. If $G = (X, Y)_c$, and if the final arc of X is a d-arc, then X is a fundamental subgraph of G. Otherwise, the fundamental subgraphs of X are fundamental subgraphs of G. (The dual case when $G = (X, Y)_c$ is obvious.)

An isomorphism from (N, C, D) to (N', C', D') is a bijection f: $N \rightarrow N'$ that preserves c- and d-paths such that for each A in N, A=f(A). We call Theorems 1 and 1′ and their corollaries the *Isomorphism Theorem.*

Theorem 1. Let G be a semantic graph, and let B be a full block in G. Then B is a union of fundamental subgraphs of some explicit subgraph of G.

Theorem 1′. If G and H are isomorphic semantic graphs, then H can be formed by reassociating and commuting some of the arcs in G.

Corollary 1. Let G be a semantic graph, and let B be a full block in G. Then there is a semantic graph G' and an isomorphism f: $G \rightarrow G'$ such that f(B) is an explicit subgraph of G'.

Corollary 2. The intersection of two full blocks is a full block.

Corollary 3. Given a semantic graph G and a collection of mutually disjoint full blocks, there is a graph isomorphic to G in which each full block is an explicit subgraph. Moreover, given any two of the blocks, each node in one is c-connected to each node in the other or each node in one is d-connected to each node in the other.

Several additional definitions are necessary to define the dissolution operation. From the isomorphism theorem we know that any full block U is a conjunction or a disjunction of fundamental subgraphs of some explicit subgraph H. If the final arc of H is a conjunction, then we define the *c-extension* of U to be H and the *d-extension* of U to be U itself. (The situation is reversed if the final arc of H is a d-arc.) We define the *c-path extension* of an arbitrary subgraph H in a semantic graph G as follows (note that this is different from the c-extension of a full block): Let $F_1, \ldots, F_k$ be the fundamental subgraphs of G that meet H, and let $F_{k+1}, \ldots, F_n$ be those that do not. Then

$$\mathrm{CPE}(\emptyset, G) = \emptyset \quad \text{and} \quad \mathrm{CPE}(G, G) = G.$$

$$\mathrm{CPE}(H, G) = \mathrm{CPE}(H_{F_1}, F_1) \vee \cdots \vee \mathrm{CPE}(H_{F_n}, F_n)$$
$$\text{if the final arc of } G \text{ is a d-arc}$$

$$\mathrm{CPE}(H, G) = \mathrm{CPE}(H_{F_1}, F_1) \wedge \cdots \wedge \mathrm{CPE}(H_{F_k}, F_k) \wedge F_{k+1} \wedge \cdots \wedge F_n$$
$$\text{if the final arc of } G \text{ is a c-arc}$$

Lemma 3. The c-paths of CPE(H, G) are precisely the c-paths of G that pass through H.

Using the same notation we define the *strong split graph of H in G*, denoted SS(H,G), as follows:

$$\mathrm{SS}(\emptyset, G) = G \quad \text{and} \quad \mathrm{SS}(G, G) = \emptyset.$$

$$\mathrm{SS}(H, G) = \mathrm{SS}(H, F_1) \vee \cdots \vee \mathrm{SS}(H, F_n)$$
$$\text{if the final arc of } G \text{ is a d-arc}$$

$$\mathrm{SS}(H, G) = \mathrm{SS}(H, F_1) \vee \cdots \vee \mathrm{SS}(H, F_k) \wedge F_{k+1} \wedge \cdots \wedge F_n$$
$$\text{if the final arc of } G \text{ is a c-arc}$$

Lemma 4. If H is a c-block in G, then SS(H, G) is isomorphic to the subgraph of G relative to the nodes that lie on c-paths that miss H.

Define the *auxillary subgraph* Aux(H, G) of a subgraph H in a semantic graph G to be the subgraph of G relative to the set of all nodes in G that lie on extensions of d-paths through H to d-paths through G.

Lemma 5. If H is a non-empty subgraph of G, then Aux(H, G) is empty iff H is a strong c-block. Moreover, Aux(H, G) cannot contain a d-path through G; if H is a c-block, then so is Aux(H, G).

Lemma 6. If H is a c-block then CPE(H, G) $=$ SS(Aux(H, G), G).

3. Path Dissolution

We define a *chain* in a graph to be a set of pairs of c-connected nodes such that each pair can simultaneously be made complementary by an appropriate substitution. A *link* is an element of a chain, and a chain is *full* if it is not properly contained in any other chain. A graph G is *spanned* by the chain K if every c-path through G contains a link from K; in that case, we call K a *resolution chain* for G.

Intuitively, path dissolution operates on a resolution chain by constructing a semantic graph whose c-paths are exactly those that do not pass through the chain. Not all resolution chains are candidates for dissolution: A special type of chain that we call a *dissolution chain* (what else?) is required. Since single links always form dissolution chains, the class is not too specialized. The construction of the dissolvent from such a chain is straightforward.

A resolution chain H is a *dissolution chain* if it is a single c-block or if it has the following form: If M is the smallest full block containing H, then $M = (X, Y)_c$ where $H \cap X$ and $H \cap Y$ are each c-blocks.

Given a dissolution chain H, define DV(H, M), the *dissolvent of* H *in* M, as follows (using the above notation): If H is a single c-block, then DV(H, M) $=$ SS(H, M). Otherwise (i.e., if H consists of two c-blocks), then

$$\mathrm{DV}(H, M) \quad = \quad \left\{ \begin{array}{ccc} \mathrm{CPE}(H, X) & \rightarrow & \mathrm{SS}(H, Y) \\ & \downarrow & \\ \mathrm{SS}(H, X) & \rightarrow & \mathrm{CPE}(H, Y) \\ & \downarrow & \\ \mathrm{SS}(H, X) & \rightarrow & \mathrm{SS}(H, Y) \end{array} \right.$$

Intuitively, DV(H, M) is a semantic graph whose c-paths miss at least one of the c-blocks of the dissolution chain. The only paths left out are those that go through the dissolution chain and hence are unsatisfiable. Notice that we may express DV(H, M) in either of the two more compact forms shown below (since CPE(H, X) $\cup$ SS(H, X) $= X$ and CPE(H, Y) $\cup$ SS(H, Y) $= Y$):

$$\begin{array}{ccc} X & \rightarrow & \mathrm{SS}(H,Y) \\ \downarrow & & \\ \mathrm{SS}(H,X) & \rightarrow & \mathrm{CPE}(H,Y) \end{array} \qquad\qquad \begin{array}{ccc} \mathrm{SS}(H,X) & \rightarrow & Y \\ \downarrow & & \\ \mathrm{CPE}(H,X) & \rightarrow & \mathrm{SS}(H,Y) \end{array}$$

Note that the three representations are semantically equivalent but are not in general isomorphic; in particular their d-paths need not be the same. The c-paths of all three representations, however, are identical; they consist of exactly those c-paths in M that do not pass through H.

Theorem 2. Let H be a ground dissolution chain in a graph G, and let M be the smallest full block containing H. Then M and DV(H, M) are equivalent.

We may therefore select an arbitrary dissolution chain H in G and replace the smallest full block containing H by its dissolvent, producing (in the ground case) an equivalent graph. We call the resulting graph the *dissolution of* G *with respect to* H and denote it Diss(G,H); links are inherited in the obvious way.

The graph formed by dissolution has strictly fewer c-paths than the old one: All remaining c-paths were present in the old graph, and the two graphs are semantically equivalent. The original graph has only finitely many c-paths, and each dissolution operation preserves its meaning. As a result, finitely many dissolutions (bounded above by the number of c-paths in the original graph) will yield a graph without links. If this graph is empty, then the original graph was spanned; if not, then every (necessarily linkless) c-path characterizes a model of the original graph.

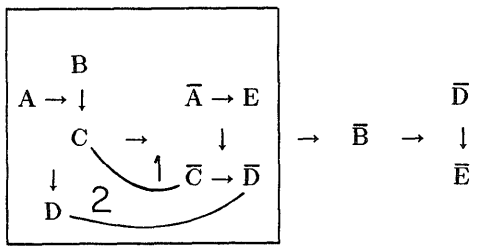
If we dissolve on a link $\{A, \bar{A}\}$ in a graph in CNF, then $H = \{A, \bar{A}\}$, X and Y are the two clauses containing A and $\bar{A}$, respectively, $M = X \cup Y$, $H_X = \{A\} =$ CPE(H_X, X), and $H_Y = \{\bar{A}\} =$ CPE(H_Y, Y). Since CPE(H_X, X) $= \overline{\mathrm{CPE}(H_Y, Y)}$ the Prawitz matrix reduction rule [Prawitz 1970] may then be used. The resulting graph is

$$\begin{array}{ccc} \mathrm{CPE}(H, X) & \rightarrow & \mathrm{SS}(H, Y) \\ & \downarrow & \\ \mathrm{SS}(H, X) & \rightarrow & \mathrm{CPE}(H, Y) \end{array}$$

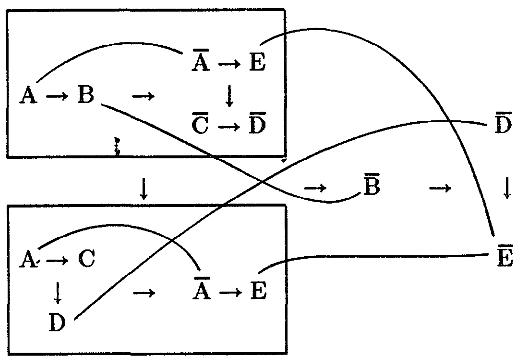
Note that Theorem 2 does not apply in this case (i.e., the Prawitz rule preserves unsatisfiability but not equivalence).

4. A Dissolution Refutation

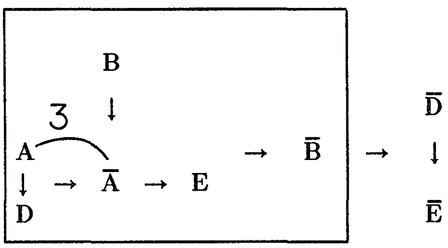
The graph below is unsatisfiable and has 12 c-paths. We box the smallest full block containing a dissolution chain about to be activated.



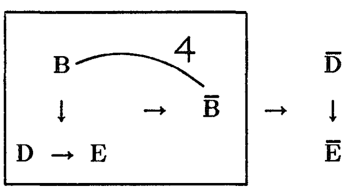
Links 1 and 2 form a dissolution chain; $M = (X, Y)_c$ where X and Y are the two leftmost fundamental subgraphs of the entire graph. Notice that SS({C,D}, X) $= X_{\{A,B\}}$ and SS($\{\bar{C},\bar{D}\}$, Y) $= Y_{\{\bar{A},E\}}$; also CPE({C,D}, X) $= X_{\{A,C,D\}}$. Dissolution removes 4 c-paths resulting in the following graph (we use the second of the two compact versions of dissolvent throughout this section):



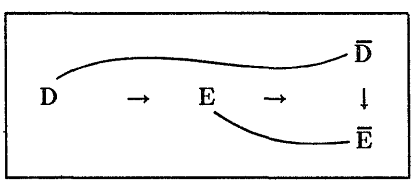
The subgraph $\overline{C} \to \overline{D}$ and the single occurrence of C are both linkless full blocks. (We have not deleted links in the ordinary sense of [Bibel 1981] or [Murray & Rosenthal 1985b]. With path dissolution, links simply disappear because their associated nodes, although c-connected in the original graph, become d-connected in the dissolvent.) We may therefore apply the Pure Lemma [Bibel 1981], [Murray & Rosenthal 1985b], [Murray & Rosenthal 1987] and delete the d-extensions of these full blocks, which in turn renders the upper occurrence of A pure. The result is:



Now we activate link 3 and apply the Pure Lemma to $\overline{A}$:



We next dissolve on link 4 to produce:

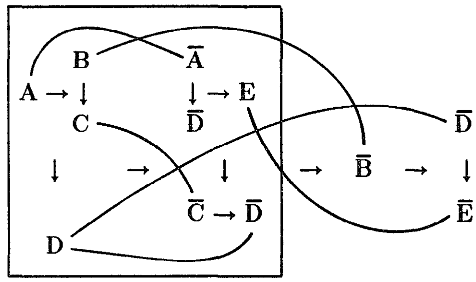


The remaining two links constitute a single strong c-block and they span the entire graph. Dissolving on them results in the empty graph.

5. Applying dissolution to a satisfiable graph.

We may always apply dissolution to a ground semantic graph until the graph is without links. The remaining c-paths, if any, characterize exactly the interpretations satisfying the graph. We must not, however, apply the Pure Lemma if that is our objective since it, unlike dissolution, preserves only satisfiability, not equivalence.

The graph below is similar to that of the previous example but is satisfiable.



The details, which are similar to the previous example, are left to the reader because of space considerations. After six dissolutions (activating a total of 10 links), the graph is reduced to $A \to C \to \overline{D} \to E \to \overline{D} \to \overline{B}$, which specifies those interpretations that satisfy the original graph.

6. First Order Dissolution

The usual arguments (involving the application of Robinson's Unification Theorem) allow us to lift ground chains to the general level. More stringent conditions, however, must be satisfied if we wish to replace the smallest full block containing a dissolution chain by its dissolvent. (Were this not the case, we would have a decision procedure for first order satisfiability.) The difficulty arises from ground instances (possibly crucial to a proof) in which the chain does not exist, i.e., instances that are not consistent with the mgsu of the chain. Of course, the dissolvent can always be soundly conjoined to the existing graph. Dissolution (with replacement) may then be applied freely to the newly inferred portion of the graph. A partial replacement technique may be applied to chains that link the old and new sections of the graph. These ideas are discussed in this section.

During the construction of the mgsu of a chain, some care must be taken regarding the familiar process of standardizing variables apart. If χ is any variable, two occurrences of χ cannot be standardized apart if they appear in d-connected nodes. In CNF this is a sufficient condition for determining whether variables may be standardized apart; in semantic graphs (NNF), this is not the case. What is required is the transitive closure of the relation 'are d-connected', which provides all the occurrences of χ that are in fact the same variable.

6.1. Partial replacement

Let G be a (first order) semantic graph, and let H be a dissolution chain with mgsu σ in the full block $M = (X, Y)_c$. Let $U = \mathrm{DV}(H, M)$ and consider the graph

$$G \to U\sigma$$

If we have a dissolution chain in $U\sigma$, we may dissolve with replacement since anything lost due to successive instantiations is present in G. If we have a dissolution chain from G to U, the smallest full block containing it will consist of a fundamental subgraph of G and a fundamental subgraph of U. (This full block could be larger if the chain contains a strong c-block.) We of course cannot replace the fundamental in G, but we may replace the fundamental in U by the entire dissolvent. The following example illustrates these ideas. It consists of five fundamental c-connected subgraphs, labeled F_1 through F_5.

$$\begin{array}{ccccccccc}
\overline{A(x)} & & & & 1 & & & & \\
\downarrow & & \overline{E(x)} & & \overline{G(x)} & & \overline{B(x)} & & A(g(x)) \\
E(x) & \to & \downarrow & \to & \downarrow & \to & \downarrow & \to & \downarrow \\
\downarrow & & B(x)\to C(x) & & B(x)\to D(x) & & \overline{C(x)}\to\overline{D(x)} & & E(h(x)) \\
G(f(x)) & & & & & & & & \\
 & & & & & & & & \\
F_1 & & F_2 & & F_3 & & F_4 & & F_5
\end{array}$$

Suppose that we first dissolve on link 1; the smallest full block containing it (F_1 conjoined with F_5), and its dissolvent, are shown below.

$$\begin{array}{ccccccc}
\overline{A(x)} & 1 & & \qquad & E(g(v)) & & A(g(v)) \\
\downarrow & & A(g(x)) & & \downarrow & \to & \downarrow \\
E(x) & \to & \downarrow & & G(f(g(v))) & & E(h(v)) \\
\downarrow & & E(h(x)) & & & \downarrow & \\
G(f(x)) & & & & \overline{A(g(v))} & \to & E(h(v)) \\
 & & & & & & \\
F_1 & & F_5 & & & F_6 &
\end{array}$$

In the original graph, several occurrences of x can be standardized apart (although we have not done so), but in the dissolvent, all occurrences of v are d-related. The dissolution operation has created two d-connected occurrences of the literal E(h(v)), both of which are linked to $\overline{E(x)}$ in the original graph. Therefore these two links are descendants of the original link, and they form a dissolution chain that is somewhat easier to find (given the appropriate bookkeeping) than an arbitrary two-link chain. Shown below is F_7, the result of dissolving on this chain.

$$\begin{array}{ccccccc}
 & \overline{E(h(v))} & & & E(g(v)) & & \\
 & \downarrow & & \to & \downarrow & \to & A(g(v)) \\
B(h(v)) & \to & C(h(v)) & & G(f(g(v))) & & \\
 & & & \downarrow & & & \\
 & & & & E(g(v)) & & \\
 & & & & \downarrow & \to & E(h(v)) \\
 & & & & G(f(g(v))) & & \\
B(h(v)) & \to & C(h(v)) & \to & & \downarrow & \\
 & & & & \overline{A(g(v))} & \to & E(h(v)) \\
 & & & & & & \\
 & & & F_7 & & &
\end{array}$$

There are now two d-connected occurrences of the c-block $B(h(v)) \to C(h(v))$; the two taken together also form a c-block. Each is linked to $(\overline{B(x)}, \overline{C(x)})_d$, a strong c-block in F_4. Dissolving results in replacing F_7 with:

$$\begin{array}{ccccccc}
\overline{B(h(v))} & & & & E(g(v)) & & \\
\downarrow & \to & \overline{E(h(v))} & \to & \downarrow & \to & A(g(v)) \\
\overline{C(h(v))}\to\overline{D(h(v))} & & & & G(f(g(v))) & & \\
 & & & & & & \\
F_8 & & F_9 & & F_{10} & & F_{11}
\end{array}$$

Next, a dissolvent is computed from the link $\{\overline{G(x)}, G(f(g(v)))\}$; we replace only F_{10}, the fundamental that meets G(f(g(v))).

$$\begin{array}{ccccccc}
 & & & & E(g(v)) & & \\
 & & & & \downarrow \;\to\; B(f(g(v))) \to D(f(g(v))) & & \\
\overline{B(h(v))} & & & & G(f(g(v))) & & \\
\downarrow & \to & \overline{E(h(v))} & \to & \downarrow & \to & A(g(v)) \\
\overline{C(h(v))}\to\overline{D(h(v))} & & & & E(g(v)) \to \overline{G(f(g(v)))} & & \\
 & & & & & & \\
F_8 & & F_9 & & F_{12} & & F_{11}
\end{array}$$

The c-block consisting of $B(f(g(v))) \to D(f(g(v)))$ in F_{12} is linked to the strong c-block $(\overline{B(x)}, \overline{D(x)})_d$ within F_4 of the original graph. Replacing F_{12} by the dissolvent yields (we omit F_8):

$$\begin{array}{ccccccccc}
 & & \overline{B(f(g(v)))} & & & & & & \\
\overline{E(h(v))} & \to & \downarrow & \to & E(g(v)) & \to & \overline{G(f(g(v)))} & \to & A(g(v)) \\
 & & \overline{C(f(g(v)))} \to \overline{D(f(g(v)))} & & & & & & \\
 & & & & & & & & \\
F_9 & & F_{13} & & F_{14} & & F_{15} & & F_{11}
\end{array}$$

The proof may be completed using F_2, F_4, and F_{14}:

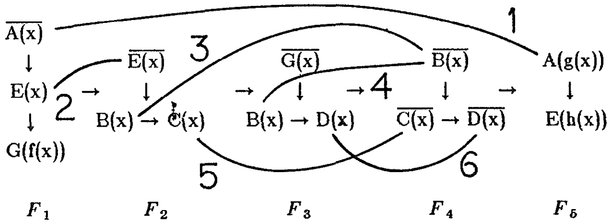
$$\begin{array}{ccccc}
\overline{E(x)} & & 2 \quad \overline{B(x)} & & \\
\downarrow & \to & \downarrow & \to & E(g(v)) \\
B(x) \to C(x) & & \overline{C(x)} \to \overline{D(x)} & & \\
 & & & & \\
F_2 & & F_4 & & F_{14}
\end{array}$$

Dissolving on link 2 produces $B(g(v)) \to C(g(v))$, which is linked to F_4 by a dissolution chain that spans the entire graph.

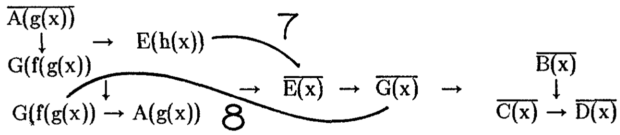
6.2. Dissolution on copies of graphs

In the previous technique, dissolution was used once within the original graph to create an inferred graph on which replacement could safely be performed. Another strategy would be to create a copy of the original graph, and then dissolve *with replacement* on the copy as much as possible. The idea is to drive the copy toward some instantiated linkless consequence, which is then conjoined to the original graph. (If we are lucky, the consequence will be empty!) The process can then be repeated, with preference given to those links (if any) not used in previous iterations. Note that in general, as dissolution is applied, some links not yet used in the copy will simply vanish, their literals having become instantiated in ways inconsistent with their original unifiers.

Let us try this approach on the previous example. Links 1 through 6 are compatible. Regardless of the order in which they are activated, the resulting graph contains all c-paths except those through any of the six links.



We omit the individual dissolution steps, and present only the result:



The semantic graph above has 6 c-paths, whereas the original one has 48. Dissolving on links 7 and 8 yields the empty graph.

These techniques look promising, but both are primitive; much remains to be investigated at the first order level. Our intuition is strong that dissolution at the ground level is likely to be an effective technique, and we cannot help but believe that, if properly lifted, it would also be effective for first order logic.

Acknowledgements

We would like to thank Stacia Quimby and Scott Shurr, without whose help we could not have dealt with multiple systems in multiple geographic locations.

References

[Andrews 1981] P.B. Andrews, "Theorem proving via general matings," *J.ACM*, vol. 28,2, pp. 193-214, April 1981.

[Bibel 1979] W. Bibel, "Tautology testing with a generalized matrix reduction method," *Theoretical Computer Science*, vol. 8, pp. 31-44, 1979.

[Bibel 1981] W. Bibel, "On matrices with connections," *J.ACM*, vol. 28,4, pp. 633-645, Oct. 1981.

[Bibel 1982] W. Bibel, "A comparative study of several proof procedures," *Artificial Intelligence*, vol. 18,3, pp. 269-293, 1982.

[Brown 1976] F. Brown, "Notes on chains and connection graphs," *Personal notes*, Department of Computation and Logic, Edinburgh University, 1976.

[Chang & Slagle 1979] C.L. Chang and J.R. Slagle, "Using rewriting rules for connection graphs to prove theorems," *Artificial Intelligence*, vol. 12, pp. 159-178, Aug. 1979.

[Eisinger 1986] N. Eisinger, "What you always wanted to know about clause graph resolution," *Proceedings of the Eighth International Conference on Automated Deduction*, Oxford, England, July 1986. In *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 230, 316-336.

[Kowalski 1975] R. Kowalski, "A proof procedure using connection graphs," *J.ACM*, vol. 22,4, pp. 572-595, Oct. 1975.

[Murray 1982] N.V. Murray, "An experimental theorem prover using fast unification and vertical path graphs," *Proc. of the Fourth National Conf. of Canadian Society of Computational Studies of Intelligence*, pp. 125-131, U. of Saskatchewan, May 1982.

[Murray & Rosenthal 1985a] N.V. Murray and E. Rosenthal, "Path Resolution and Semantic Graphs," *Proceedings of EUROCAL '85*, Linz Austria, April 1-3, 1985. In *Lecture Notes in Computer Science*, Springer-Verlag, vol. 204, 50-63.

[Murray & Rosenthal 1985b] N.V. Murray and E. Rosenthal, "Path Resolution With Link Deletion," *Proceedings of IJCAI-85*, pp. 1187-1193, UCLA, Aug. 18-24, 1985.

[Murray & Rosenthal 1987] N.V. Murray and E. Rosenthal, "Inference With Path Resolution and Semantic Graphs," To appear in *J.ACM*, vol. 34,2, April 1987.

[Prawitz 1970] D. Prawitz, "A proof procedure with matrix reduction," *Lecture Notes in Mathematics*, vol. 125, pp. 207-213, Springer-Verlag, 1970.

[Robinson 1965] J.A. Robinson, "A machine oriented logic based on the resolution principle," *J.ACM*, vol. 12,1, pp. 23-41, 1965 .

[Stephan & Siekmann 1978] W. Stephan and J. Siekmann, "Completeness and soundness of the connections graph proof procedure," *AISB/GI Conference on Artificial Intelligence* (D. Sleeman, ed.), pp. 340-344, Leeds University Press, 1978, Hamburg, July 18-20, 1978.

[Stickel 1982] M.L. Stickel, "A nonclausal connection-graph resolution theorem-proving program," *Proc. AAAI-82 Nat. Conf. on Artificial Intelligence*, pp. 229-233, Pittsburgh, Pennsylvania, Aug. 1982.