Some Remarks on Completeness, Connection Graph Resolution and Link Deletion*

Technical report 98-1
January, 1998

Reiner Hähnle[1], Neil V. Murray[2], and Erik Rosenthal[3]

[1] Dept. of Computer Science, Univ. of Karlsruhe,
76128 Karlsruhe, Germany, `reiner@ira.uka.de`

[2] Dept. of Computer Science, University at Albany - SUNY
Albany, NY 12222, USA, `nvm@cs.albany.edu`

[3] Dept. of Mathematics, Univ. of New Haven,
West Haven, CT 06516, USA, `brodsky@charger.newhaven.edu`

Abstract. A new completeness proof that generalizes the Anderson-Bledsoe excess literal argument is developed for connection-graph resolution. The technique also provides a simplified completeness proof for semantic resolution. Some observations about subsumption and about link deletion are made. Link deletion is the basis for connection graphs. Subsumption plays an important role in most resolution-based inference systems. In some settings—for example, connection graphs in negation normal form—both subsumption and link deletion can be quite tricky. Nevertheless, a completeness result that uses both is obtained in this setting.

Keywords: *Completeness, connection-graph resolution, semantic resolution, subsumption, link deletion*

---

* This research was supported in part by the National Science Foundation under grants CCR-9404338 and CCR-9504349 and by the Deutsche Forschungsgemeinschaft within Schwerpunktprogramm Deduktion.

1 Introduction

Robinson developed semantic tree arguments [12] to provide completeness proofs for resolution and related inference systems. These arguments were not entirely transparent in that they do not directly construct a proof in the deduction system. Rather, they work with an intermediate data structure—semantic trees. Some novices find the resulting proofs difficult to follow. The excess literal technique discovered by Anderson and Bledsoe [1], which is almost completely syntactic, is a considerable simplification. This technique, which is essentially an induction on the size of the formula, was the basis for the first completeness proofs of certain refinements of resolution.

In this paper we continue work begun in [7, 6] and describe how the excess literal technique can be adapted to provide concise completeness proofs for a variety of deduction methods including for some non-clausal systems. In addition to providing simplified proofs of known results the role of subsumption in non-clausal proof systems is discussed, and a completeness proof for non-clausal connection-graph resolution is obtained.

Kowalski [8] introduced connection-graph (cg) resolution in 1975. It keeps track of all links—complementary pairs of literals—occurring in a clause set and employs deletion of the links used in a resolution step to reduce the size of the search space.

Fewer links made completeness questionable, and the semantic tree and excess literal techniques seemed to be insufficient. After six years, Bibel [3] finally proved cg-resolution to be complete. The crucial advance he made was the notion of spanning and the observation that cg-resolution steps preserve spanning (see Section 3). His approach is similar in some respects to the one presented here, but it requires a non-trivial double induction and relies on the existence of refutations that must conform to an excessively rigid structure.

Link deletion is sometimes possible with inference rules that do not rely on clause form. Path dissolution [11] and the tableau method implicitly delete links. Link deletion is also possible with semi-resolution [9], but proving that the spanning property is preserved is highly non-trivial. Some insight into link deletion in negation normal form is provided in Section 5.3.

Stickel [13] defined and implemented non-clausal cg-resolution, but he did not give a completeness proof, suspecting that it “may be difficult.” Indeed, if one tries to generalize Bibel's proof to the non-clausal case, one quickly faces seemingly insurmountable technicalities. In this paper, a variation of the Anderson-Bledsoe excess literal technique is employed to prove completeness of non-clausal cg-resolution (Theorem 4). In addition, we show that the same technique can be used to provide simplified proofs of other (known) results.

In Section 2, a variation of the Anderson-Bledsoe excess literal technique for proving completeness of resolution is described. The method is illustrated with a succinct, straightforward completeness proof for semantic resolution. In Section 3, the proof technique is used to provide a simplified proof of the completeness of clausal cg-resolution. Some new observations regarding cg-subsumption are also noted. Some prerequisites on NNF formulas are given in Section 4, and subsumption is generalized to NNF in Section 5.2. Completeness of non-clausal cg-resolution is then proven in Section 5.3.

2 A Simple Completeness Proof for Semantic Resolution

The work described here was inspired by the Anderson-Bledsoe [1] excess literal proof of the completeness of resolution. It is essentially an induction on the size of a sentence $\mathcal{S}$. An interesting variation of their proof can be obtained by applying the induction to the number of distinct atoms that appear in $\mathcal{S}$. A good example is the proof below that semantic resolution is complete. Older proofs of this result are widely recognized as quite opaque. Bachmair & Ganzinger's proof [2] is well structured, but does not yield a direct, syntactic construction.

The proof here is for the ground case; it lifts in the usual way. We make the standard convention that a set of clauses is interpreted as the conjunction of its members, and that a clause is a set of literals, interpreted as the disjunction of its members. Thus defined, duplicate clauses in a sentence and duplicate literals in a clause are automatically "merged" into one copy.

Semantic resolution is defined with respect to a given interpretation I. Let $\mathcal{S}$ be a set of clauses, and let $\mathcal{F}$ be the subset of $\mathcal{S}$ whose members are falsified by I. (Regardless of the choice of I, $\mathcal{F}$ is non-empty if $\mathcal{S}$ is unsatisfiable.) Let $\mathcal{T} = \mathcal{S} - \mathcal{F}$. Obviously, I satisfies all clauses in $\mathcal{T}$.

To define semantic resolution, let $N = \{\overline{b_1}, \ldots, \overline{b_n}\} \cup C$ be a clause in $\mathcal{T}$ in which I satisfies the $\overline{b_i}$'s and falsifies the literals in C, and suppose the clauses $B_1, B_2, ..., B_n$ are in $\mathcal{F}$ with $b_k \in B_k$. Suppose further that for any $b' \in B_i - \{b_i\}$, $b' \neq b_j, 1 \leq j \leq n$. Then the clause $R = C \cup (\bigcup_{i=1}^{n} B_i - \{b_i\})$ is the *semantic resolvent* of the *nucleus clause* N and the *satellite clauses* $B_1, \ldots, B_n$. Obviously, I falsifies R. That R can be soundly inferred from N and the B_i's can easily be seen by noting that a sequence of binary resolutions between N and the B_i's produces R. Semantically, any interpretation that satisfies the parent clauses either satisfies one literal in C (and thus R) or else some $\overline{b_i}$. But then I falsifies b_i and must satisfy $B_i - \{b_i\}$ (and thus R). Observe that hyperresolution may be regarded as a special case of semantic resolution by considering the interpretation that assigns false to every atom.

We begin by recalling the *pure rule*: A literal in a set of clauses is said to be *pure* if its complement does not occur in any other clause. In that case, the clause containing the pure literal is also said to be pure. The proof of the next lemma is straightforward.

Lemma 1 (Pure Rule). Let $\mathcal{S}$ be an unsatisfiable clause set in which the clause C is pure. Then $\mathcal{S}' = \mathcal{S} - \{C\}$ is unsatisfiable.

Lemma 2. Let $\mathcal{S} = \{C_0, C_1, C_2, \ldots, C_k\}$ be a minimally unsatisfiable set of clauses (i.e., no proper subset of $\mathcal{S}$ is unsatisfiable), and suppose $C_0 = \{p\} \cup \{q_1, \ldots, q_n\}$, where $n \geq 0$. Obtain $\mathcal{S}'$ from $\mathcal{S}$ by deleting every occurrence of p in $\mathcal{S}$. Then $\mathcal{S}'$ is unsatisfiable, and every minimally unsatisfiable subset of $\mathcal{S}'$ contains $C_0' = \{q_1, \ldots, q_n\}$ but contains no clauses that contain $\overline{p}$.

Proof. Every clause in $\mathcal{S}'$ subsumes a clause in $\mathcal{S}$, so $\mathcal{S}'$ is unsatisfiable. By the pure rule, any minimally unsatisfiable subset of $\mathcal{S}'$ cannot contain any clauses that contain $\overline{p}$. Also, since $\mathcal{S}$ is minimally unsatisfiable, there must be an interpretation I_0 that falsifies C_0 but satisfies every other clause in $\mathcal{S}$. Thus, I_0 satisfies every clause in $\mathcal{S}'$ other than C_0'. But then C_0' must be in any minimally unsatisfiable subset of $\mathcal{S}'$. □

Theorem 1. Semantic resolution is refutation complete for propositional logic.

Proof. Let $\mathcal{S} = \{C_1, C_2, \ldots, C_m\}$ be an unsatisfiable set of clauses. We assume that $\mathcal{S}$ is minimally unsatisfiable; otherwise, restrict attention to a minimally unsatisfiable subset. We must show that, given an arbitrary interpretation I, there is a refutation of $\mathcal{S}$ using semantic resolution with respect to I.

Proceed by induction on the number n of distinct atoms in $\mathcal{S}$. If there are none, then $\mathcal{S}$ contains the empty clause, and we are done.

So suppose that all unsatisfiable sets of clauses with at most n atoms can be refuted with semantic resolution, and assume that $\mathcal{S}$ has $n+1$ atoms including the atom p. Let $q = p$ if I falsifies p, $q = \overline{p}$ otherwise. Since $\mathcal{S}$ is minimal, the Pure Rule implies that p cannot be pure, so both q and $\overline{q}$ occur in $\mathcal{S}$.

If $\mathcal{S}$ contains the unit clause $\{q\}$, fine; otherwise, remove all occurrences of q from $\mathcal{S}$. This formula is unsatisfiable by Lemma 2. Consider a minimally unsatisfiable subset; by Lemma 2, every clause that had contained q in $\mathcal{S}$ is in this set and no clause containing $\overline{q}$ is present. By the induction hypothesis, there is a refutation $\mathcal{R}_q$ by semantic resolution. Now apply that refutation to $\mathcal{S}$; call the resulting refutation $\mathcal{R}_q'$. The effect is to reintroduce q into some clauses, so that, with merging, the clause $\{q\}$ rather than the empty clause may be produced by $\mathcal{R}_q'$.

Nevertheless, each step is a semantic resolution step. The reason is that I falsifies q, and so, by reintroducing q, the membership in $\mathcal{F}$ or $\mathcal{T}$ of the resulting clauses is unchanged. Also, none of the clauses containing $\overline{q}$ are resolved upon, and the result is either the empty clause[1] or the clause $\{q\}$. This clause is the last semantic resolvent and is in $\mathcal{F}$.

Analogously, if we begin by deleting $\overline{q}$, a refutation $\mathcal{R}_{\overline{q}}$ by semantic resolution can be found. Let the proof that results from applying $\mathcal{R}_{\overline{q}}$ to $\mathcal{S}$ be denoted by $\mathcal{R}_{\overline{q}}'$. This proof yields either the empty clause or the unit $\{\overline{q}\}$. However, it may not be a semantic resolution proof with respect to I, because reintroducing $\overline{q}$ into a member of $\mathcal{F}$ produces a clause satisfied by I and thus in $\mathcal{T}$. But a semantic resolution proof can be constructed from $\mathcal{R}_{\overline{q}}'$ using the unit $\{q\}$ derived by $\mathcal{R}_q'$.

Consider an arbitrary step in $\mathcal{R}_{\overline{q}}$ with nucleus $N = \{\overline{b_1}, \ldots, \overline{b_n}, c_1, ..., c_m\}$ and satellites $B_k \in \mathcal{F}$ with $b_k \in B_k, 1 \leq k \leq n$. Suppose that $\overline{q}$ is reintroduced into some of these clauses. First, if $\overline{q}$ is in N, simply use $\{q\}$ as an additional satellite, and it will be resolved away. If $\overline{q}$ is also reintroduced into some B_k, the resulting clause $B_k \cup \{\overline{q}\}$ is a member of $\mathcal{T}$ and cannot be a satellite. But we may first use it as a nucleus clause in a semantic resolution with satellite $\{q\}$, and

---

[1] In fact, this cannot happen because of minimality, but this is not really relevant.

the semantic resolvent is simply B_k. This in turn can be used in the step that employed nucleus N, and so that step is unchanged. Note that this construction assures that the last step that produced $\overline{q}$ in $\mathcal{R}'_{\overline{q}}$ now produces the empty clause.

Combining $\mathcal{R}'_q$ and the modified $\mathcal{R}'_{\overline{q}}$ produces the required semantic resolution proof. $\square$

The above induction is somewhat reminiscent of the Davis-Putnam-Loveland procedure [5]: Refutations are obtained from the induction hypothesis by removing all occurrences of a given atom. A simplified completeness proof for connected CNF tableaux can also be obtained with this technique [7, 6].

3 Connection Graphs

Let $\mathcal{S}$ be a set of clauses. A *link* in $\mathcal{S}$ is a pair of complementary literals from different clauses, and a *c-path* through $\mathcal{S}$ is a set containing exactly one literal occurrence from each clause in $\mathcal{S}$. A c-path may also be thought of as a maximal conjunction of literal occurrences in $\mathcal{S}$ and corresponds to a clause in a DNF equivalent of $\mathcal{S}$. Any satisfying interpretation for $\mathcal{S}$ will satisfy every literal on some c-path of $\mathcal{S}$. Obviously, if $\mathcal{S}$ is unsatisfiable, no c-path can be satisfied, and thus every c-path will contain a link.

A link set containing every link in $\mathcal{S}$ is said to be *full*, and $\mathcal{S}$ is said to be *spanned* by a set of links $\mathcal{L}$ if every c-path of $\mathcal{S}$ contains a member of $\mathcal{L}$. If $\mathcal{S}$ is unsatisfiable, then it is spanned by its links; note, however, that a link set $\mathcal{L}$ may span $\mathcal{S}$ and yet not be full. Intuitively, if $\mathcal{L}$ spans $\mathcal{S}$, then $\mathcal{L}$ contains enough information to demonstrate the unsatisfiability of $\mathcal{S}$. A *connection graph* for $\mathcal{S}$ consists of $\mathcal{S}$ along with a set $\mathcal{L}$ of links from $\mathcal{S}$; it is denoted $\mathcal{G}(\mathcal{S}, \mathcal{L})$. We write $\mathcal{G}$ when the clause and link set are obvious from context. We say that $\mathcal{G}$ (or $\mathcal{S}$) is *minimally spanned* if removal of any clause and its associated links produces a graph that is not spanned.

Example 1. Consider the connection graph $\mathcal{G}(\mathcal{S}, \mathcal{L})$ with unsatisfiable clause set $\mathcal{S}$ consisting of $C = \{p, \overline{q}\}$, $D = \{\overline{p}, \overline{q}\}$, $E = \{q\}$ and full link set $\mathcal{L} = \{\{p_C, \overline{p}_D\}\}\ \{\overline{q}_C, q_E\}\ \{\overline{q}_D, q_E\}\}$. $\mathcal{G}$ is minimally spanned.

3.1 The Pure Rule and Subsumption

Care is required with links in a connection graph since a complementary pair of literal occurrences may not be in the link set. Thus, if p is a literal in the clause C, we often use p_C to denote the occurrence of p in C. Care is also required for the notion of purity: Suppose $\mathcal{G}(\mathcal{S}, \mathcal{L})$ is spanned and contains the clause $C = A \cup \{p\}$. Suppose further that no link in $\mathcal{L}$ contains the literal p_C. Then the literal occurrence p_C and the clause C are said to be *pure* in $\mathcal{G}$. The next lemma is the pure rule for connection graphs; its proof is straightforward.

Lemma 3 (Pure Rule). Let $\mathcal{G}(\mathcal{S}, \mathcal{L})$ be spanned and contain the clause $C = A \cup \{p\}$, where p_C is pure. Then $\mathcal{G}' = \mathcal{G}(\mathcal{S}', \mathcal{L}')$ is spanned, where $\mathcal{S}' = \mathcal{S} - \{C\}$ and $\mathcal{L}'$ is the result of removing from $\mathcal{L}$ all links that meet clause C.

In the next lemma, and in subsequent developments, objects are removed from the clauses of a connection graph. In such situations, it is implicitly assumed that links associated with the removed objects are also removed.

Lemma 4. Let $\mathcal{S} = \{C_0, C_1, C_2, \ldots, C_k\}$ be a minimally spanned set of clauses, and suppose $C_0 = \{p\} \cup \{q_1, \ldots, q_n\}$, where $n \geq 0$. Obtain $\mathcal{S}'$ from $\mathcal{S}$ by deleting p from C_0, Then $\mathcal{S}'$ is spanned, and every minimally spanned subset of $\mathcal{S}'$ contains $C_0' = \{q_1, \ldots, q_n\}$ but contains no clauses that contain an unlinked $\overline{p}$. More generally, p may be removed from all clauses in $\mathcal{S}$ to produce a spanned set of clauses with one fewer atom.

Proof. Removing a literal from a clause in $\mathcal{S}$ simply removes the c-paths containing that literal, so $\mathcal{S}'$ is spanned. By the pure rule, any minimally spanned subset of $\mathcal{S}'$ cannot contain pure clauses.

Observe that not all c-paths through $\mathcal{S} - \{C_0\}$ are linked since $\mathcal{S}$ is minimally spanned. Thus, there is a linkless c-path P_0 through $\mathcal{S} - \{C_0\}$. Since $\mathcal{S}$ is spanned, there is a link connecting every literal in C_0 to a literal in P_0. Since P_0 is also a c-path (without links) through $\mathcal{S}' - \{C_0'\}$, $\mathcal{S}' - \{C_0'\}$ is not spanned, and C_0' is in any minimally spanned subset of $\mathcal{S}'$.

Finally, successive applications of the first part of the lemma can be used to produce a minimally spanned set of formulas in which p has been completely removed. □

The classical notion of subsumption carries over to the connection-graph setting in a spanning-preserving manner [3]. It must be modified because a link set may not be full. Suppose connection graph $\mathcal{G}(\mathcal{S}, \mathcal{L})$ contains clauses C and D, and suppose that $C \subseteq D$; i.e., C "classically" subsumes D. For each literal p_C in C, we denote by $\mathcal{L}_{p_C}$ the set of literal occurrences linked to p_C in $\mathcal{L}$. Suppose further that for every literal p_C, $\mathcal{L}_{p_C} \supseteq \mathcal{L}_{p_D}$. Then we say clause C *cg-subsumes* clause D.

Lemma 5 (Cg-subsumption). Suppose that $\mathcal{G}(\mathcal{S}, \mathcal{L})$ is spanned and contains clauses C and D, and suppose that C cg-subsumes D. Then $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$ is spanned, where $\mathcal{S}' = \mathcal{S} - \{D\}$, and $\mathcal{L}'$ is the result of removing from $\mathcal{L}$ all links that meet clause D.

We will subsequently investigate and further develop cg-subsumption. However, the tools already described are sufficient to produce a quite elegant completeness proof for cg-resolution.

3.2 Cg-resolution

To define connection graph resolution, let $\mathcal{G}(\mathcal{S}, \mathcal{L})$ be a connection graph containing clauses $C = \{p\} \cup A$ and $D = \{\overline{p}\} \cup B$. Suppose further that $\mathcal{L}$ contains the link $L = \{p_C, \overline{p}_D\}$. Then we may cg-resolve clauses C and D to produce the connection graph $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$, where $\mathcal{S}' = \mathcal{S} \cup \{E\}$, $E = A \cup B$, and $\mathcal{L}' = \mathcal{L} - \{L\} \cup INH$. The set INH consists of *inherited links*. Intuitively, each literal in E comes from one in C or from one in D and so inherits the links the corresponding literal had in the parent clause. When the same literal occurs in both parents, the occurrences are merged, and the resolvent inherits the links of both. Formally,

$$INH = \{\{p_E, \overline{p}_Q\} \mid \{p_C, \overline{p}_Q\} \in \mathcal{L} \text{ or } \{p_D, \overline{p}_Q\} \in \mathcal{L}\}$$

Example 2. In $\mathcal{G}(\mathcal{S}, \mathcal{L})$ from Example 1 we may cg-resolve clauses C and D via $L = \{p_C, \overline{p}_D\}$ with resolvent $F = \{\overline{q}\}$ and inherited link set $INH = \{\{\overline{q}_F, q_E\}\}$. The resulting link set $\mathcal{L}'$ spans $\mathcal{S}' = \mathcal{S} \cup \{F\}$, but $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$ is not minimally spanned, since $\{E, F\} \subset \mathcal{S}'$ is spanned by INH.

The link set in the example is not full after only one cg-resolution step. Nevertheless, spanning is preserved. It is not difficult to prove but was quite difficult to see; it was the key observation that Bibel made that led to the first completeness proof of cg-resolution. Below a new proof is presented that is included here because it is shorter and simpler than Bibel's original proof, and because it will make it easier for the reader to follow the proof of the NNF case in Section 5.3. Note, however, that the new proof relies heavily on Bibel's lemma:

Lemma 6 (Bibel [3]). Suppose that a link in the spanned connection graph $\mathcal{G}(\mathcal{S}, \mathcal{L})$ is selected and cg-resolution produces the connection graph $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$. Then $\mathcal{G}'$ is spanned, i.e., $\mathcal{S}'$ is spanned by $\mathcal{L}'$.

Theorem 2. Connection-graph resolution is refutation complete for propositional logic.

Proof. Let $\mathcal{G}(\mathcal{S}, \mathcal{L})$ be a connection graph spanned by $\mathcal{L}$; we must show that there is a refutation of $\mathcal{S}$ using cg-resolution. We proceed by induction on the number n of distinct atoms in $\mathcal{S}$. If there are none, then $\mathcal{S}$ contains the empty clause, and we are done. Otherwise, assume that all spanned connection graphs with at most n atoms can be refuted with cg-resolution, and suppose that $\mathcal{S}$ has $n + 1$.

If $\mathcal{S}$ is not minimally spanned, then restrict attention to a minimally spanned subset. Let p be any atom in $\mathcal{S}$; we begin by deriving the unit clause $\{p\}$ from $\mathcal{S}$. If $\mathcal{S}$ contains $\{p\}$, we have it; otherwise, by Lemma 4, we can remove all occurrences of p in $\mathcal{S}$, producing the spanned $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$. By the induction hypothesis, there is a refutation of $\mathcal{S}'$. This refutation applied to $\mathcal{S}$ produces either the empty clause, in which case the proof is complete, or the unit clause p, as promised.

The unit $\{p\}$ cannot be pure because no occurrence of p in $\mathcal{S}$ was pure, and no resolution step involved any clause from $\mathcal{S}$ that contained $\overline{p}$. In particular, there are no new clauses containing $\overline{p}$. Let $C = \{\overline{p}\} \cup C'$ be a clause that is linked to p. The cg-resolvent of $\{p\}$ and C is C', which cg-subsumes C since no link to C' has been deleted. Thus, a minimally spanned subset of the resulting connection graph is a spanned clause set in which the number of occurrences of $\overline{p}$ is reduced.

We now iterate this process, successively reducing the number of occurrences of $\overline{p}$ while preserving the spanning property. Eventually a spanned graph with no occurrences of $\overline{p}$ is produced. By the pure rule, a minimally spanned subset contains no occurrences of p. Thus, we have a spanned connection graph with at most n distinct atoms. By the induction hypothesis, there is a refutation of the empty clause. $\square$

Remark. In [3], Bibel uses a somewhat restricted inheritance rule to further reduce the number of links. The above proof applies with that inheritance rule virtually without modification.

3.3 Cg-subsumption

In the proof of Theorem 2, we made rather straightforward use of cg-subsumption. The requirements on the links of the clauses involved are necessary to ensure that spanning is preserved. In fact, whenever C subsumes D, C can be forced to cg-subsume D by performing enough cg-resolutions on D. For example, if the literal p occurs in both clauses, and if p_D is linked to $\overline{p_E}$ but p_C is not, then cg-resolve on the $\{p_D, \overline{p_E}\}$ link. This deletes the link; with enough cg-resolutions involving a subsumed clause, excess links that disallow cg-subsumption can be deleted, and the subsumed clause will be cg-subsumed.

This offers the advantage of deleting the subsumed clause, but the penalty is severe: A number of extra clauses are introduced into the search space. If these extra clauses turn out to be necessary for a proof, then the penalty is really no penalty at all. However, this seems unintuitive. On the other hand, if the extra cg-resolutions are avoided, the subsumed clause must be kept.

There may be a better approach. Suppose that in connection graph $\mathcal{G}$, clause C subsumes clause D but does not cg-subsume D. Suppose further that for each literal p_C in C such that $\mathcal{L}_{p_C} \not\supseteq \mathcal{L}_{p_D}$, we add to $\mathcal{L}_{p_C}$ exactly those links required to ensure that $\mathcal{L}_{p_C} \supseteq \mathcal{L}_{p_D}$. By cg-subsumption, D may now be deleted from the graph. We say that D has been removed by *augmented cg-subsumption.*

First observe that augmented cg-subsumption preserves spanning. Adding links to the subsuming clause cannot harm the spanning property, nor can the subsequent removal of the cg-subsumed clause. However, we would like to know that we have left completeness unaffected and not increased either the proof or the search space.

Investigation of the proof and search spaces is beyond the scope of this paper and left for future work. Completeness is, however, easily settled. First, cg-subsumption is a special case of augmented cg-subsumption; so the current proof of Theorem 2 goes through with the latter. But that proof could in fact be greatly simplified: All occurrences of literal p could be removed at once, and a derivation of (at least one) unit clause $\{p\}$ results. *All* other clauses containing p are then deleted by augmented cg-subsumption. As a result, *all* clauses with $\overline{p}$ are either already pure or become so after being resolved with the unit. Then the unit is pure. Voilá: The induction hypothesis provides a refutation.

4 Negation Normal Form

Formulas in *negation normal form* (NNF) are defined inductively: (i) Literals, true, false are NNF formulas; (ii) if $\mathcal{F}_1, \ldots, \mathcal{F}_m$ are NNF formulas, so are $\mathcal{F}_1 \wedge \cdots \wedge \mathcal{F}_m$ and $\mathcal{F}_1 \vee \cdots \vee \mathcal{F}_m$. We identify formulas that are equal up to associativity.

The *subformulas* of an NNF formula $\mathcal{G}$ are defined as follows: (i) Literals have only themselves as subformulas; (ii) if $\mathcal{G} = \mathcal{F}_1 \circ \cdots \circ \mathcal{F}_m$, where $\circ \in \{\wedge, \vee\}$, then for any $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, m\}$, $\mathcal{F}_{i_1} \circ \cdots \circ \mathcal{F}_{i_r}$ is a subformula of $\mathcal{G}$; (iii) the subformula relation is transitive.

If p and q are literals in an NNF formula $\mathcal{F}$, and if $p \in \mathcal{G}$ and $q \in \mathcal{H}$ where $\mathcal{G} \wedge \mathcal{H}$ is a subformula of $\mathcal{F}$, then p and q are said to be *c-connected*; if $\mathcal{G}$ and $\mathcal{H}$ are disjoined, then p and q are said to be *d-connected.* A *link* is a complementary pair of c-connected literals. Unless otherwise stated, we will assume that for any formula, the following simplification rules have been applied; in them, p is a literal and $\mathcal{S}$ is an arbitrary formula.

$$
\begin{array}{lll}
p \vee \cdots \vee p = p \wedge \cdots \wedge p = p & p \vee \neg p = \text{true} & p \wedge \neg p = \text{false} \\
\mathcal{S} \vee \text{false} = \mathcal{S} \wedge \text{true} = \mathcal{S} & \mathcal{S} \wedge \text{false} = \text{false} & \mathcal{S} \vee \text{true} = \text{true}
\end{array}
$$

A *c-path through* $\mathcal{F}$ is a maximal set of mutually c-connected literals, and a *d-path through* $\mathcal{F}$ is a maximal set of mutually d-connected literals. The c-paths of a formula correspond to the clauses of one of its disjunctive normal form (DNF) equivalents. Similarly, the d-paths correspond to the clauses of a CNF equivalent. If $\mathcal{F} = \mathcal{G} \wedge \mathcal{H}$, and if P and Q are c-paths through $\mathcal{G}$ and $\mathcal{H}$, respectively, then $P \cup Q$ is a c-path through $\mathcal{F}$; it is denoted PQ. (If P and Q are d-paths through $\mathcal{G}$ and $\mathcal{H}$, respectively, then P and Q are each a d-path through $\mathcal{F}$.) The next lemma (from [10]) is easy to prove.

Lemma 7. A formula is satisfiable if and only if some c-path in it is satisfiable, and a c-path is satisfiable if and only if it does not contain a link.

We use *a set* of NNF formulas to denote the conjunction of the its members. In particular, we assume the formulas in the set to be either disjunctions or literals. Thus, complementary literals residing in different formulas are in fact c-connected and constitute a link. Links may also reside within a single formula. As in the clausal case, a spanned set of NNF formulas is *minimally spanned* if removing any member produces a set that is not spanned.

Our completeness arguments require the removal of syntactic objects from formulas in a spanning preserving way. This is more complicated for NNF formulas than it is for clause form, where all that is necessary is the removal of literal occurrences from the clauses in which they appear. One way to remove an object $\mathcal{P}$ from a spanned formula $\mathcal{S}$ to produce a spanned formula $\mathcal{S}_\mathcal{P}$ is to remove all c-paths containing $\mathcal{P}$. Since all c-paths of $\mathcal{S}$ contain a link, and since the c-paths of $\mathcal{S}_\mathcal{P}$ will be a subset of the c-paths of $\mathcal{S}$, $\mathcal{S}_\mathcal{P}$ will also be spanned. It turns out that a good choice for $\mathcal{S}_\mathcal{P}$ is the *c-path complement* of $\mathcal{P}$ in $\mathcal{S}$, denoted $CC(\mathcal{P}, \mathcal{S})$. We use a definition of $CC(\mathcal{P}, \mathcal{S})$ which is tailored to the special case required in this paper; for the general case, see [11].

Definition 1. *Let $\mathcal{H}$ be a subformula of the formula $\mathcal{G}$. Then the* c-path complement *of $\mathcal{H}$ with respect to $\mathcal{G}$, denoted $CC(\mathcal{H},\mathcal{G})$, is the subformula that results from replacing $\mathcal{H}$ by* false *and making obvious simplifications.*

Alternatively, $CC(\mathcal{H},\mathcal{G})$ can be simply characterized with the CE operator defined as follows. Let $\mathcal{P}$ be a subformula of an NNF formula $\mathcal{G}$. The *c-extension* $\mathcal{P}$, denoted $CE(\mathcal{P})$ is the largest subformula of $\mathcal{G}$ of the form $\mathcal{P} \wedge \mathcal{P}'$. The *d-extension* is denoted $DE(\mathcal{P})$ and is similarly defined. Observe that one of $DE(\mathcal{P})$ and $CE(\mathcal{P})$ must consist of $\mathcal{P}$ alone, and the other one cannot (unless $\mathcal{F} = \mathcal{P}$). The following Theorem is proved in [11].

Theorem 3. If $\mathcal{P}$ is a subformula of formula $\mathcal{S}$ then $CC(\mathcal{P},\mathcal{S}) = \mathcal{S} - CE(\mathcal{P})$.

The pure rule is valid for NNF formulas [11] but requires some care. In essence, the NNF equivalent of the clause containing a given literal p is $DE(p)$.

Lemma 8 (Pure Rule for Negation Normal Form). If p is a pure literal in an unsatisfiable NNF formula $\mathcal{S}$, then $\mathcal{S} - DE(p)$ is unsatisfiable.

The previous lemma is adapted to the cg setting in a straightforward way:

Lemma 9 (Pure Rule for NNF Connection Graphs). If p is a pure literal in a spanned NNF formula $\mathcal{S}$, then $\mathcal{S} - DE(p)$ is spanned.

Proof. Removal of $DE(p)$ cannot harm spanning unless for some c-path Q, all the links on Q meet $DE(p)$. We may write Q as $Q'Q_{DE(p)}$. Note that $DE(p)$ is a disjunction, and $Q_{DE(p)}$ is a c-path through one of its disjuncts other than p. But then $Q'p$ is a linkless c-path through $\mathcal{S}$, contrary to hypothesis. □

Lemma 10. Let $\mathcal{S} = \{\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\}$ be a minimally spanned set of NNF formulas, and p a literal in $\mathcal{F}_0$. Obtain $\mathcal{S}'$ from $\mathcal{S}$ by removing CE(p) for every occurrence of p in $\mathcal{F}_0$. Obtain $\mathcal{S}''$ by applying the pure rule to $\mathcal{S}'$ (possibly repeatedly). *$\mathcal{F}_i'$ are the formulas in $\mathcal{S}'$ corresponding to $\mathcal{F}_i$ in $\mathcal{S}$.*

1. *Both $\mathcal{S}'$ and $\mathcal{S}''$ are spanned;*
2. *$\mathcal{F}_0'$ is a member of any minimally spanned subset of $\mathcal{S}'$.*

Proof. 1. Removal of $CE(p)$ removes c-paths from the formula (Theorem 3). Since $\mathcal{S}$ is spanned, every c-path contains a link, and that is unchanged; $\mathcal{S}''$ is spanned by Lemma 9.

2. Let P_0 be a linkless c-path through $\{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$. Such a P_0 must exist since otherwise $\mathcal{S}$ would not be minimally spanned. However, since $\mathcal{S}$ is spanned, every extension of P_0 through $\mathcal{F}_0$ must contain a link. Since P_0 is also an unlinked c-path through $\mathcal{S}' - \mathcal{F}_0'$, $\mathcal{S}' - \mathcal{F}_0'$ is not spanned. Thus, any spanned subset of $\mathcal{S}'$ must contain $\mathcal{F}_0'$. □

5 Non-Clausal Resolution

We begin by providing a precise definition of non-clausal (NC) resolution on literals for NNF formulas. Let $\mathcal{F}$ and $\mathcal{G}$ be arbitrary ground NNF formulas, where p is an atom occurring in both $\mathcal{F}$ and $\mathcal{G}$. We denote by $\mathcal{F}[p/\beta]$ the result of replacing all occurrences of p in $\mathcal{F}$ by β. If $\beta = \text{true}$ or $\beta = \text{false}$, we assume that simplifications are performed. Then the *NC-resolvent* of $\mathcal{F}$ and $\mathcal{G}$ on the atom p is:

$$\mathcal{F}[p/\text{false}] \ \vee \ \mathcal{G}[p/\text{true}] \ .$$

5.1 Polarity

Although the definition above is symmetric with respect to $\mathcal{F}$ and $\mathcal{G}$, we have the following *polarity restriction*:

If p occurs only positively in $\mathcal{F}$ or only negatively in $\mathcal{G}$, then we need not consider the dual NC-resolvent

$$\mathcal{F}[p/\text{true}] \ \vee \ \mathcal{G}[p/\text{false}] \ .$$

The atom p occurs positively in $\mathcal{F}$ if it occurs as the literal p; p occurs negatively in $\mathcal{F}$ if it occurs in the literal $\overline{p}$. Note that p may occur both positively and negatively in $\mathcal{F}$.

5.2 NNF subsumption

In the discussion that follows, we will often refer to subsumption of d- and c-paths rather than of disjuncts and conjuncts. Paths are defined as sets of literal occurrences, but with regard to subsumption, we consider the literal set of a path P. In this way, no change in the basic definitions is needed. Clausal subsumption is generalized to the NNF case in a fairly straightforward manner.

Recalling that the d-paths of a formula correspond to the clauses of a CNF equivalent, we say that $\mathcal{F}$ *d-subsumes* $\mathcal{F}'$ if for every non-tautological d-path P' in $\mathcal{F}'$, there is a d-path P in $\mathcal{F}$, such that $P \subseteq P'$. The following lemma is easy.

Lemma 11. Let $\mathcal{S}$ be an unsatisfiable set of NNF formulas in which $\mathcal{F}$ d-subsumes $\mathcal{F}'$. Then $\mathcal{S} - \{\mathcal{F}'\}$ is unsatisfiable.

Clearly, if $\mathcal{F}$ d-subsumes $\mathcal{F}'$, then $\mathcal{F} \models \mathcal{F}'$. There is a dual (but not equivalent) syntactic characterization of subsumption for NNF formulas. We say that $\mathcal{F}$ *c-subsumes* $\mathcal{F}'$ if for every non-contradictory c-path P in $\mathcal{F}$, there is a c-path P' in $\mathcal{F}'$ such that $P \supseteq P'$. Of course, if $\mathcal{F}$ c-subsumes $\mathcal{F}'$, then $\mathcal{F} \models \mathcal{F}'$.

Lemma 12. Let $\mathcal{S}$ be an unsatisfiable set of NNF formulas in which $\mathcal{F}$ c-subsumes $\mathcal{F}'$. Then $\mathcal{S} - \{\mathcal{F}'\}$ is unsatisfiable.

To see that neither of these subsumption rules capture implication completely, consider $\mathcal{F}_1 = p$, $\mathcal{F}_2 = (p \vee q)$, $\mathcal{F}_3 = ((p \vee q) \wedge (r \vee \overline{r}))$. It is easy to see that $\mathcal{F}_1$ implies both $\mathcal{F}_2$ and $\mathcal{F}_3$. But $\mathcal{F}_1$ both d- and c-subsumes $\mathcal{F}_2$; $\mathcal{F}_1$ also d-subsumes $\mathcal{F}_3$ but does not c-subsume it.

The conditions of Lemmas 11 and 12 are quite strong; they are in fact necessary if we wish to delete an entire formula from a set of formulas. However, subsumption can more usefully be generalized further, so that more modest reductions can be captured. Lemma 13 below captures as a special case, Lemma 12. First, we define *c-reduction* for NNF formulas.

Given formulas $\mathcal{F}$ and $\mathcal{F}'$, suppose there is a subformula $\mathcal{H}'$ of $\mathcal{F}'$, such that $\mathcal{F}$ c-subsumes $\mathcal{H}'$. Then we say that $\mathcal{F}$ *c-reduces* $\mathcal{F}'$ via $\mathcal{H}'$.

Lemma 13. Let $\mathcal{S}$ be an unsatisfiable set of NNF formulas in which $\mathcal{F} \in \mathcal{S}$ c-reduces $\mathcal{F}'$ via $\mathcal{H}'$. Then $\mathcal{S}_{\mathcal{H}'} = (\mathcal{S} - \{\mathcal{F}'\}) \cup \{\mathcal{F}' - DE(\mathcal{H}')\}$ is unsatisfiable.

Proof. Suppose interpretation I satisfies $\mathcal{S}_{\mathcal{H}'}$. Then I satisfies some c-path P_I of $\mathcal{S}$, and P_I contains a c-path through $\mathcal{F}$. If P_I does not meet $DE(\mathcal{H}')$, then P_I is a c-path through $\mathcal{S}$, and I satisfies $\mathcal{S}$.

If P_I meets $DE(\mathcal{H}')$, then P_I may be written $PP_{DE(\mathcal{H}')}P_{\mathcal{F}}$, where $P_{DE(\mathcal{H}')}$ is P_I restricted to $DE(\mathcal{H}')$, and $P_{\mathcal{F}}$ is P_I restricted to $\mathcal{F}$. Since $\mathcal{F}$ c-subsumes $\mathcal{H}'$, $P_{\mathcal{F}} \supseteq P'$, for some c-path P' of $\mathcal{H}'$. Obviously, I satisfies the literals of P' and thus satisfies $PP'P_{\mathcal{F}}$. But this latter c-path is a c-path through $\mathcal{S}$, and so I satisfies $\mathcal{S}$. □

Of course, the dual notion of *d-reduction* is defined in the obvious way, and the dual of Lemma 13 holds. To adapt these notions to the connection-graph setting, it is most convenient to restrict attention to c-reduction.

Suppose $\mathcal{F}$ c-reduces $\mathcal{F}'$ via $\mathcal{H}'$. Suppose furthermore that for each non-contradictory c-path P' in $\mathcal{H}'$, each literal p is linked to a subset of the literals linked to occurrences of p on $P \supseteq P'$ for some non-contradictory c-path P in $\mathcal{F}$. Then we say that $\mathcal{F}$ *cg-c-reduces* $\mathcal{F}'$ via $\mathcal{H}'$.

Lemma 14. Let $\mathcal{S}$ be a spanned set of NNF formulas in which $\mathcal{F} \in \mathcal{S}$ cg-c-reduces $\mathcal{F}'$ via $\mathcal{H}'$. Then $\mathcal{S}_{\mathcal{H}'} = (\mathcal{S} - \{\mathcal{F}'\}) \cup \{\mathcal{F}' - DE(\mathcal{H}')\}$ is spanned.

Proof. Suppose for some c-path P through $\mathcal{S}$, every link on P meets $DE(\mathcal{H}')$. We may write $P = P_1P_{DE(\mathcal{H}')}P_{\mathcal{F}}$, where $P_{DE(\mathcal{H}')}$ is the part of P in $DE(\mathcal{H}')$ and $P_{\mathcal{F}}$ is the part of P in $\mathcal{F}$. Since $\mathcal{F}$ c-subsumes $\mathcal{H}'$, $P_{\mathcal{F}} \supseteq P'$, for some c-path P' of $\mathcal{H}'$. The c-path $P_1P'P_{\mathcal{F}}$ is a c-path of $\mathcal{S}$ and contains a link. But this link must be to P'. But then $P_{\mathcal{F}}$ is also linked to the same literal by the definition of c-reduction, contrary to the hypothesis that every link of P was to $DE(\mathcal{H}')$. □

As with augmented cg-subsumption for CNF formulas, we may define *augmented c-reduction* for NNF formulas. Suppose that in NNF connection graph $\mathcal{G}$, formula $\mathcal{F}$ c-reduces formula $\mathcal{F}'$ via $\mathcal{H}'$ but does not cg-c-reduce $\mathcal{F}'$. Suppose further that for each literal $p_{\mathcal{F}}$ on c-path $P_{\mathcal{F}}$, corresponding to literal $p_{\mathcal{F}'}$ on c-path $P_{\mathcal{F}'} \subseteq P_{\mathcal{F}}$, such that $\mathcal{L}_{p_{\mathcal{F}}} \not\supseteq \mathcal{L}_{p_{\mathcal{F}'}}$, we add to $\mathcal{L}_{p_{\mathcal{F}}}$ exactly those links required to ensure that $\mathcal{L}_{p_{\mathcal{F}}} \supseteq \mathcal{L}_{p_{\mathcal{F}'}}$. We may now apply cg-c-reduction. We say that $\mathcal{F}'$ has been *augmented cg-c-reduced.*

Of course, augmented cg-c-reduction preserves spanning. Adding links to the c-reducing formula cannot harm the spanning property, nor can the subsequent application of cg-c-reduction. As in the clausal case, we believe that completeness is unaffected, and neither the proof nor the search space have increased. As we have said, the NNF case is complex; an in depth investigation of completeness and of the proof and search spaces is beyond the scope of this paper and left for future work. In the next subsection, we do take some preliminary steps in this direction.

5.3 Non-clausal connection-graph resolution

To define cg-resolution for NNF formulas, let $\mathcal{G}(\mathcal{S}, \mathcal{L})$ be a connection graph containing NNF formulas $\mathcal{E}$ and $\mathcal{F}$, and suppose the atom p occurs positively in $\mathcal{E}$ and negatively in $\mathcal{F}$. Suppose further that $\mathcal{L}$ contains a non-empty set L of links of the form $\{p_{\mathcal{E}}, \overline{p}_{\mathcal{F}}\}$. Then we may cg-resolve formulas $\mathcal{E}$ and $\mathcal{F}$ to produce the connection graph $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$, where $\mathcal{S}' = \mathcal{S} \cup \{\mathcal{H}\}$, and

$$\mathcal{H} = \mathcal{E}[p/\text{false}] \ \vee \ \mathcal{F}[p/\text{true}] \ .$$

The resulting set of links $\mathcal{L}'$ is defined as follows: If p occurs only positively in $\mathcal{E}$ and only negatively in $\mathcal{F}$, then $\mathcal{L}' = \mathcal{L} - L \cup INH$; otherwise, $\mathcal{L}' = \mathcal{L} \cup INH$. The set INH consists of *inherited links.* As in the CNF case, each literal in $\mathcal{H}$ comes from one in $\mathcal{E}$ or from one in $\mathcal{F}$ and so inherits the links the corresponding literal had in the parent formula. Formally,

$$INH = \{\{p_{\mathcal{H}}, \overline{p}_{\mathcal{Q}}\} \mid \{p_{\mathcal{E}}, \overline{p}_{\mathcal{Q}}\} \in \mathcal{L} \text{ or } \{p_{\mathcal{F}}, \overline{p}_{\mathcal{Q}}\} \in \mathcal{L}\} \ .$$

Example 3. Consider the NNF connection graph consisting of four numbered formulas in Fig. 1.

The graph is fully linked except for $\{p_2, \overline{p}_4\}$, which is not present. (In the figure to the right, links are denoted by arcs connecting the linked literals). The remaining links span the graph.

If we cg-resolve on $\{p_3, \overline{p}_4\}$ (highlighted by the dashed arc), the resolvent is $(\text{false} \vee (\text{true} \wedge r)) \vee (\text{false} \vee q)$. Simplifying the latter yields (5) $r \vee q$.

Observe that the activated link must not be deleted, because p has mixed polarity in (3).

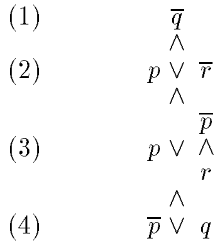

Fig. 1. NNF connection graph.

The careful reader will have noticed that whenever the atom resolved upon has mixed polarity in either parent formula, no links are deleted. This may at first seem surprising; however, past studies have shown that link deletion in the NNF setting is complicated. See, for example, [10]. More to the point, this restriction is crucial to the spanning-preservation lemma below. To see this, consider again Example 3 (Fig. 1). If the activated link were deleted, the c-path $\{\overline{q}_1, p_2, p_3, \overline{p}_4, r_5\}$ would be unlinked. In fact, successive applications of the Pure Rule would remove every formula from the graph, resulting in the (satisfiable!) empty conjunction.

Lemma 15. Suppose that a link $\{p_{\mathcal{E}}, \overline{p}_{\mathcal{F}}\}$ between formulas $\mathcal{E}$ and $\mathcal{F}$ in the NNF connection graph $\mathcal{G}(\mathcal{S}, \mathcal{L})$ is selected and cg-resolution produces the connection graph $\mathcal{G}'(\mathcal{S}', \mathcal{L}')$. Then $\mathcal{G}'$ is spanned, i.e., $\mathcal{S}'$ is spanned by $\mathcal{L}'$.

Proof. If p occurs negatively in $\mathcal{E}$ or positively in $\mathcal{F}$, then no links are deleted and the result is immediate. Otherwise, let the cg-resolvent be denoted by $\mathcal{H}$. Consider a c-path P through $\mathcal{G}$, for which the only links on P are in L. We must show that every extension of P through $\mathcal{H}$ is linked.

Observe that the cg-resolvent $\mathcal{H}$ is constructed as the disjunction of $\mathcal{E}'$ and $\mathcal{F}'$, where $\mathcal{E}'$ has all c-paths in $\mathcal{E}$ that do not contain p, and $\mathcal{F}'$ has all c-paths in $\mathcal{F}$ that do not contain $\overline{p}$. (All literals containing p are replaced by false, and this amounts to computing the c-path complement of those occurrences; see Theorem 3.) All such c-paths in $\mathcal{H}$ have isomorphic c-paths in either $\mathcal{E}$ or in $\mathcal{F}$ from which a link is inherited.

More precisely, let $P = P'P_{\mathcal{E}}P_{\mathcal{F}}$ be a c-path containing only links of the form $\{p_{\mathcal{E}}, \overline{p}_{\mathcal{F}}\}$, which are deleted in the cg-resolution. Without loss of generality, extend P through $\mathcal{E}'$ in $\mathcal{H}$ to $PP_{\mathcal{H}} = P'P_{\mathcal{E}}P_{\mathcal{F}}P_{\mathcal{E}'}$. Let $P^{\mathcal{E}'}$ be the c-path in $\mathcal{E}$ isomorphic to $P_{\mathcal{E}'}$. Since $\mathcal{G}$ is spanned, the c-path $P'P^{\mathcal{E}'}P_{\mathcal{F}}$ must have a link to $P^{\mathcal{E}'}$. But this link is inherited on $P_{\mathcal{E}'}$. ☐

Theorem 4. Non-clausal connection-graph resolution is refutation complete for propositional logic.

Proof. Let $\mathcal{G}(\mathcal{S}, \mathcal{L})$ be an NNF formula spanned by $\mathcal{L}$; we must show that there is a refutation of $\mathcal{S}$ using cg-resolution. Proceed by induction on the number n of distinct atoms in $\mathcal{S}$. If there are none, then $\mathcal{S}$ contains false, and we are done. Otherwise, assume that all spanned connection graphs with at most n atoms can be refuted with cg-resolution, and suppose that $\mathcal{S}$ has $n + 1$.

If $\mathcal{S}$ is not minimally spanned, then restrict attention to a minimally spanned subset. Let p be any atom in $\mathcal{S}$; we begin by deriving the unit formula $\{p\}$ from $\mathcal{S}$. If $\mathcal{S}$ contains the unit $\{p\}$, fine. Otherwise, successive applications of Lemma 10 produces a minimally spanned set of formulas $\mathcal{S}'$ in which all occurrences of $CE(p)$ and $DE(\overline{p})$ have been removed.

By the induction hypothesis, there is a refutation $\mathcal{R}$ by cg-resolution. Were we dealing with clauses, we could argue that if $\mathcal{R}$ is applied to $\mathcal{S}$, either false or the literal p would be derived. However, in the NNF case, this is not obvious. The reason is that a given formula $\mathcal{F}_i$ of $\mathcal{S}$ may contain several occurrences of p and of $\overline{p}$; the c-extension of each occurrence of p and the d-extension of each occurrence of $\overline{p}$ are missing from the corresponding $\mathcal{F}_i'$'s in $\mathcal{S}'$. The effect of $\mathcal{R}$ on each of these structures must be considered when applying $\mathcal{R}$ to $\mathcal{S}$.

Consider first the c-extensions of p. The steps of $\mathcal{R}$ can change $\mathcal{F}_i$ outside $CE(p)$ and can change $CE(p)$ itself. Changes outside $CE(p)$ either leave it unchanged or delete it in its entirety due to a simplification rule. Changes inside

$CE(p)$ can also delete it entirely. The other possibility is that what remains is a formula of the form $p \wedge \psi$. The point is, if $CE(p)$ is not completely deleted, then a formula of the form $p \wedge \psi$ remains.

The d-extensions of $\overline{p}$ are disjunctions. When $\overline{p}$ occurs in $\mathcal{F}_i$, $DE(\overline{p})$ is either $\mathcal{F}_i$ itself or a proper subformula of $\mathcal{F}_i$. In the former case, $\mathcal{F}_i$ would have been removed in forming $\mathcal{S}'$ and not have participated in $\mathcal{R}$. Otherwise, $DE(\overline{p})$ is one conjunct in some conjunction of $\mathcal{F}_i$. This conjunction is eliminated in $\mathcal{R}$ and will still be eliminated when $\mathcal{R}$ is applied to $\mathcal{S}$, unless it also contains p; in that case the conjunction is simply one of the c-extensions of p that remain as discussed above.

In any case, the result of applying $\mathcal{R}$ to $\mathcal{S}$ is either false, in which case we are done, or a formula set $\mathcal{S}''$ made up of formulas of the form $p \wedge \psi$. Observe that $\mathcal{S}''[\text{false}/p]$ will reduce to false via the simplification rules since every $p \wedge \psi$ has this property.

Thus, $\mathcal{R}$ applied to $\mathcal{S}$ produces either false or the unit $\{p\}$. This unit cannot be pure, and we may proceed as in the CNF case: Resolve this unit with $\mathcal{F}'$'s that contain $\{\overline{p}\}$. The resolvent will cg-subsume the other parent formula. Thus we can produce a formula set containing fewer occurrences of $\overline{p}$. Repeating this process produces a formula set with one less atom, and the induction hypothesis gives us a refutation. □

We note that with augmented cg-c-subsumption, the above proof can be simplified in much the way that the proof for the CNF case was. It is worth mentioning that by using the distributive laws or d-path complement operators, a formula containing p both positively and negatively can be replaced by two, in each of which p occurs with only one polarity. In other words, we can force link deletion to be enabled, although it is not at all clear that doing so provides an advantage.

5.4 Strong vs. weak completeness

We close with a brief discussion of strong completeness (any sequence of resolution steps chosen according to some easily decidable condition and starting with an unsatisfiable formula does end with the empty formula) vs. weak completeness (*there is* a sequence of resolution steps ending with the empty formula). Strong completeness of clausal cg-resolution was recently proven [4]. Unfortunately, we cannot hope to obtain a proof of strong completeness easily from the method employed in the present paper. The reason is that it is inherently existential, as are many completeness arguments. The *existence* of a proof combined with some simple fairness criteria yields strong completeness trivially when link deletion is not present: The space of proofs that exist is invariant with respect to what step is chosen next. But with link deletion, some proofs that exist prior to a step no longer exist after that step. Others do, but we cannot be sure that they are not perpetually receeding over the horizon.

References

1. R. Anderson and W. Bledsoe. A linear format for resolution with merging and a new technique for establishing completeness. *JACM*, 17(3):525–534, July 1970.
2. L. Bachmair and H. Ganzinger. A theory of resolution. Technical Report MPI–I–97–2–005, Max-Planck-Institut für Informatik, Saarbrücken, 1997. To appear in: J. A. Robinson & A. Voronkov (eds.), *Handbook of Automated Reasoning*, Elsevier.
3. W. Bibel. On matrices with connections. *JACM*, 28:633–645, 1981.
4. W. Bibel and E. Eder. Decomposition of tautologies into regular formulas and strong completeness of connection graph resolution. *JACM*, 44(2):320–344, 1997.
5. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *CACM*, 5:394–397, 1962.
6. R. Hähnle, N. Murray, and E. Rosenthal. Completeness for linear regular negation normal form inference systems. In Z. W. Raś and A. Skowron, editors, *Found. of Intelligent Systems ISMIS'97, Charlotte/NC, USA*, volume 1325 of *LNCS*, pages 590–599. Springer-Verlag, 1997.
7. R. Hähnle, N. Murray, and E. Rosenthal. A remark on proving completeness. In D. Galmiche, editor, *Position Papers at Conf. on Analytic Tableaux and Related Methods, Nancy, France*, pages 41–47, 1997. Tech. Rep. 97–R–030, CRIN Nancy.
8. R. Kowalski. A proof procedure using connection graphs. *JACM*, 22(4):572–595, 1975.
9. N. Murray, A. Ramesh, and E. Rosenthal. The semi-resolution inference rule and prime implicate computations. In *Proc. Fourth Golden West International Conference on Intelligent Systems, San Fransisco, CA, USA*, pages 153–158, June 1995.
10. N. V. Murray and E. Rosenthal. Inference with path resolution and semantic graphs. *JACM*, 34(2):225–254, 1987.
11. N. V. Murray and E. Rosenthal. Dissolution: Making paths vanish. *JACM*, 40(3):504–535, 1993.
12. J. A. Robinson. The generalized resolution principle. In *Machine Intelligence*, volume 3, pages 77–93. Oliver and Boyd, Edinburgh, 1968.
13. M. E. Stickel. A nonclausal connection-graph resolution theorem-proving program. In D. Waltz, editor, *Proc. National Conference on Artificial Intelligence*, pages 229–233, Pittsburgh, PA, Aug. 1982. AAAI Press.