Negative strategy

We show that the negative strategy is refutationally complete. You will find two proofs below, one that directly constructs the refutation proof from the semantic tree, and another that works by induction on the number of variables.

Direct construction

Assume that $\mathcal{P} = \{P_1, \ldots, P_n\}$ and let E be an unsatisfiable set of clauses. Either E contains the empty clause, in which case we are done. Otherwise, let A be the semantic tree associated with E and I the maximal interpretation (w.r.t. $\leq_{lex}$) in A such that both children of I are failure nodes. Using negative resolution, we will derive a clause that is falsified by I ; therefore, adding that clause to E will yield a smaller tree, and repetition of this procedure eventually yields the empty clause.

By I_0 we denote the right child of I. Let $\mathcal{P}' = \{P_{k_1}, \ldots, P_{k_\ell}\}$ be the set of variables having value 0 in I_0, where $k_i < k_{i'}$ if $i < i'$. If C is the clause labelling I_0, then C contains some subset of $\mathcal{P}'$ (but at least P_{k_ℓ}) as positive literals and negative literals otherwise. For $i = 1, \ldots, \ell$, we denote by J_i the interpretation that agrees with I_0 on $\{P_1, \ldots, P_{k_i - 1}\}$ and $J_i(P_{k_i}) = 1$. Because of the maximality condition on I_0, J_i must be a failure node, so let C_i be the clause labelling it. C_i contains at most $P_{k_1}, \ldots, P_{k_{i-1}}$ as positive literals and contains at least P_{k_i} as a negative literal. Note that C_1 is a negative clause.

We now resolve C_1 with each of $C_2, \ldots, C_\ell, C$ on P_{k_1} and call the resulting clauses $C_2^{(1)}, \ldots, C_\ell^{(1)}, C^{(1)}$. (If one of these resolution steps, say with C_2, is not possible because C_2 does not contain P_{k_1}, then simply take $C_2^{(1)} := C_2$.) Now P_{k_1} no longer occurs in $C_2^{(1)}, \ldots, C_\ell^{(1)}, C^{(1)}$, and $C_2^{(1)}$ is a negative clause. We now resolve $C_2^{(1)}$ with each of $C_3^{(1)}, \ldots, C_\ell^{(1)}, C^{(1)}$ on P_{k_2} etc until we obtain the clause $C^{(\ell)}$. Now, $C^{(\ell)}$ is negative and does not contain any variables from $\mathcal{P}'$, therefore I falsifies it.

Let $E' := E \cup \{C^{(\ell)}\}$ and A' its semantic tree. A' is smaller than A because I (or one of its ancestors) is labelled by $C^{(\ell)}$, moreover $C^{(\ell)}$ was derived from E by negative resolution. We thus continue the resolution proof as above, but with the set E', and repeat until we eventually reduce the semantic tree to the root.

Example

As an example, consider Figure 1, where I corresponds to the lowest branch in the tree. Applying the strategy described above, one derives the clause $\neg P_1 \vee \neg P_3$, which is falsified even by the immediate anchestor of I.

Proof by induction on the number of variables

In fact, there is a much nicer proof of the theorem, which works by induction on the number of variables : let n be the number of variables in $\mathcal{P}$ and E an unsatisfiable clause set. If $n = 0$ (the base case), then E contains the empty

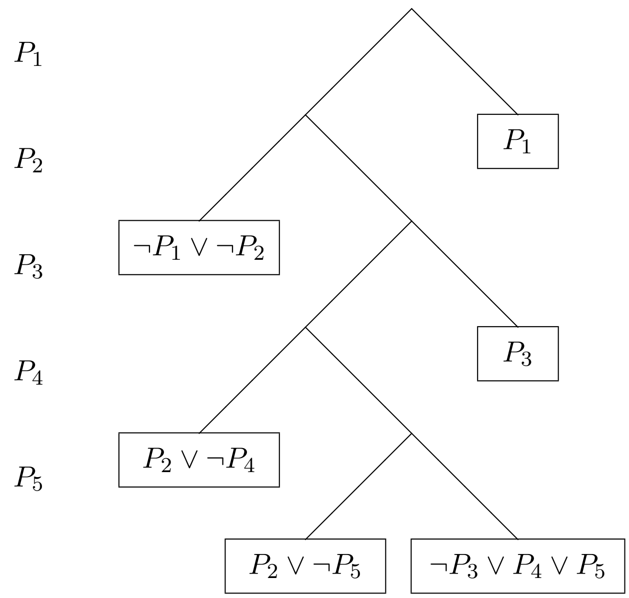

FIG. 1 – Example for negative strategy

clause, and we are immediately done. Otherwise, assume that the negative strategy is refutationally complete for sets with $n-1$ distinct variables, and let A be one of the n variables occurring in E and

Now, if E is unsatisfiable, then so are $E[A \rightarrow \perp]$ and $E[A \rightarrow \top]$, which denote the sets where all occurrences of A in E have been replaced by $\perp$ and $\top$, respectively, together with appropriate simplifications. Thus, one can obtain $E[A \rightarrow \top]$ from E by removing all clauses in which A occurs as a positive literal and removing $\neg A$ from all clauses in which it appears as a negative literal.

Now, $E[A \rightarrow \perp]$ and $E[A \rightarrow \top]$ have $n-1$ variables, and if E is unsatisfiable, then so are these two. Thus, by induction, we can derive a refutation proof for $E[A \rightarrow \top]$ using a negative strategy. We take this proof and put $\neg A$ back into all the clauses where we removed it from, thus obtaining a derivation of $\neg A$. (Notice that the resulting proof still follows the negative strategy.)

Now, we resolve $\neg A$ with all the clauses in which A occurs as a positive literal. These steps are allowed by the negative strategy, and they give us exactly the clauses that form $E[A \rightarrow \perp]$. And according to the induction hypothesis, we can derive a negative refutation proof from $E[A \rightarrow \perp]$.

Notes

- Try it out on the example from above, and you will find that it constructs exactly the same refutation as from the first proof.
- A tiny modification proves that the "positive" strategy is also refutationally complete, i.e. the strategy where every resolution step contains an entirely positive premise.

Input strategy for Horn clauses

We show that the input strategy is refutationally complete for Horn clauses. Let $\mathcal{P}$ be the set of propositional variables and E be an unsatisfiable set of clauses.

Preliminaries

Recall that any Horn clause is one of tree types. We say that a clause

- with a single positive literal, e.g., P_1, is type (I) ;
- with a single positive literal and at least one negative literal, e.g., $\neg P_1 \vee P_2$ (equivalently, $P_1 \to P_2$), is type (II) ;
- without any positive literal, e.g., $\neg P_1 \vee \neg P_2$, (equivalently, $P_1 \wedge P_2 \to \perp$), is type (III).

Let us recall the "marking" algorithm from the previous exercise. In this algorithm, a variable was marked if it was necessarily 1 in any satisfying interpretation.

> Initially, mark every variable occurring in a type (I) clause. If all negative variables of a type (II) clause are marked, mark its positive variable. If all negative variables of a type (III) clause are marked, terminate saying that E is unsatisfiable. Otherwise, if no more variables can be marked, then E is satisfiable.

Constructing a refutation

We now show that the empty clause can be derived using the input strategy. Let us assume that E does not contain the empty clause, otherwise the proof is trivial.

The idea is to construct the proof "backwards" from the marking algorithm, i.e., we start with the type (III) clause that causes termination, and then use resolution steps to replace each variable with those that caused it to become marked. For this, we extend the marking algorithm by recording (partial) functions $\xi\colon \mathcal{P} \rightsquigarrow E$ and $h\colon \mathcal{P} \rightsquigarrow \mathbb{N}$ like this :

- for a type (I) clause $C = P_i$, we set $\xi(P_i) = C$ and $h(P_i) = 1$;
- if P_i is marked because of a type (II) rule $C = P_1 \wedge \cdots \wedge P_n \to P_i$, we set $\xi(P_i) = C$ and $h(P_i) = 1 + \sum_{j=1}^{n} h(P_j)$;

For any type (III) clause $C = P_1 \wedge \cdots \wedge P_n \to \perp$, define $g(C) := \sum_{j=1}^{n} h(P_j)$. Now we construct an input strategy proof as follows :

1. If E is found to be unsatisfiable because of a type (III) rule $C = P_1 \wedge \cdots \wedge P_n \to \perp$, set $D := C$ (notice that $D \in E$).
2. If D is the empty clause, we are done.
3. Otherwise, $D = D' \vee \neg P$ for some variable P. Perform a resolution step between D and $\xi(P)$ (notice that $\xi(P) \in E$, so this is legal) and call the result D_{new}. Either $\xi(P)$ has the form $D'' \to P$, then $D_{new} = D' \vee D''$ and $g(D_{new}) = g(D') + g(D'') = (g(D) - h(P)) + (h(P) - 1)$. Or $\xi(P)$ has the form P, then $D_{new} = D'$ and $g(D_{new}) = g(D') = g(D) - h(P) = g(D) - 1$. In both cases, D_{new} is again type (III) and $g(D_{new}) < g(D)$.

4. Set $D := D_{new}$ and go to step 2.

Thus, every resolution step is between a negative clause and a clause from E, and $g(D)$ strictly decreases in each step ; eventually, $g(D)$ has reached 0, which implies that D is the empty clause.

Example

Let $E = \{P_1, P_2, P_3, P_1 \land P_2 \to P_4, P_3 \to P_5, P_4 \land P_5 \to \bot\}$.

- The algorithm first marks P_1, P_2, P_3, setting $\xi(P_i) = P_i$ and $h(P_i) = 1$ for $i = 1, 2, 3$.
- Then, P_4 is marked, where $\xi(P_4) = P_1 \land P_2 \to P_4$ and $h(P_4) = 3$.
- Then, P_5 is marked, where $\xi(P_5) = P_3 \to P_5$ and $h(P_5) = 2$.
- Finally, we find unsatisfiability because of $P_4 \land P_5 \to \bot$.

The resulting proof is shown below, where in each line, D is shown on the left, and a clause from E appears on the right.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P_4 \land P_5 \to \bot \quad P_1 \land P_2 \to P_4}{P_1 \land P_2 \land P_5 \to \bot} \quad P_1}{P_2 \land P_5 \to \bot} \quad P_2}{P_5 \to \bot} \quad P_3 \to P_5}{P_3 \to \bot} \quad P_3}{\bot}$$

Notes

- The proof strategy shown above not only obeys the input strategy, but also the negative strategy and the linear strategy. (The linear strategy says that the first resolution step is arbitrary, and in every resolution step after that, one of the premises is the clause derived in the previous step. The linear strategy is refutationally complete for general formulae.)
- The marking algorithm corresponds to a proof strategy for Horn clauses that uses positive resolution.

Linear strategy

We prove that the linear strategy is refutationally complete. To begin with, we call an unsatisfiable set *minimal* iff each of its proper subsets is satisfiable. Clearly, each unsatisfiable set contains at least one minimal unsatisfiable subset (in general, there might be multiple, for instance $\{P, \neg P, Q, \neg Q\}$ has two minimal subsets). Note also that we will use L to denote a literal and $\overline{L}$ its complement, so $\overline{P} = \neg P$ and $\overline{\neg P} = P$.

In order to show refutational completeness, we show the following stronger property :

> Let E be a minimal unsatisfiable set and $F \in E$ one of its clauses. There exists a linear resolution proof where F is used in the first resolution step.

Let n be the number of propositional variables in E. The proof is by induction on n.

If $n = 0$ (the base case), then E (being minimal) contains only the empty clause, and we are immediately done.

Otherwise, assume that the linear strategy is refutationally complete for minimal sets with $n - 1$ distinct variables. We now make a case distinction :

Either F consists of a single literal L. Let E' be some minimal subset of $E[L \to \top]$. Realise that $E[L \to \top]$ consists of two subsets :

- clauses from E that mention neither L nor $\overline{L}$;
- clauses F' such that $F' \vee \overline{L}$ is in E.

If E' contained only clauses of the first type, it would be a proper subset of E and therefore satisfiable. Thus, E' contains a clause F' such that $F' \vee \overline{L} \in E$. Since E' mentions fewer variables than E and is minimal, we can construct a linear refutation proof π on E' for the empty clause that starts with F'.

Now we modify π as follows :

- Except for the use of F' in the first step of π, we replace every instance of a clause F'', where $F'' \vee \overline{L} \in E$, with $F'' \vee \overline{L}$. Call the result π'.
- Add a first step to π', where $F = L$ and $F' \vee \overline{L}$ are resolved to yield F'. Call the result π''.

We see that π'' is a valid linear chain of resolution steps within E. Either π'' already proves the empty clause, then we are done. Or π'' proves $\overline{L}$, in which case we add a final resolution step between $\overline{L}$ and L, which is in E. In either case, we obtain a linear resolution proof of the empty clause.

If F is not a single literal, then it is of the form $F = L \vee F'$. Let E' be a minimal unsatisfying subset of $E[L \to \bot]$. Since E is minimal, there must be some interpretation I that falsifies F and satisfies all other elements of E. Thus, I falsifies both L and F'. Now, $E[L \to \bot]$ contains two subsets :

- clauses from E that mention neither L nor $\overline{L}$;
- clauses F'' such that $F'' \vee L$ is in E.

From the above it follows that I satisfies all clauses of the first part and all clauses $F'' \neq F'$ of the second part. Therefore, if E' is unsatisfiable, it must contain F'.

By induction assumption, we can obtain a linear proof π for the empty clause in E'. Similar to above, we modify π and replace every clause F'' where $F'' \vee L \in E$ by $F'' \vee L$. The result is a linear proof of L in E starting with F, which we call π'.

Consider the set $E'' := E \setminus \{F\} \cup \{L\}$, which is unsatisfiable because any satisfying interpretation of E'' would also satisfy E. Moreover, the interpretation I (from above) falsifies L and satisfies all other clauses in E'', so any minimal subset of E'' surely contains L. Then, using the first case from above, we can construct a linear proof π'' of the empty clause in E'' that starts with L.

Finally, we concatenate π' and π'' to obtain a linear proof of the empty clause starting with F.