Resolution Completeness Notes

September 5, 2019

Theorem 1 *Let f be an unsatisfiable CNF formula. Then f has a resolution refutation.*

The proof that we gave was based on the notion of a decision tree.

Definition 2 *A decision tree, T, for $f(x_1, \ldots, x_n)$ is a rooted tree of depth at most n. The nodes of T are labelled with variables x_i. If a nonleaf node is labelled by x_i, its left and right outedges are labelled by $x_i = 0$ and $x_i = 1$, respectively. A path labelling corresponds to a partial truth assignment. A node v is a leaf if and only if f is set to either 0 or to 1 under the partial assignment σ_v labelling the path to node v.*

A decision tree for f is just a way to exhaustively enumerate over all truth assignments, but where we can truncate early whenever a partial assignment to some of the variables already sets the formula to true or false. Thus the following claim follows from the definition.

Claim 3 *f is unsatisfiable if and only if for any decision tree T for f, all leaves of T are labelled by 0.*

We now want to show that given a decision tree for f where all of the leaves are labelled by 0, the tree can be converted into a resolution refutation for f.

In order to see this, we will label the leaves of the tree, T, with not only the value 0 (indicating that the partial assignment to this leaf falsifies some clause), but also with the name of some clause that is actually made false by the partial assignment.

This gives us an annotated decision tree, where all leaf nodes are labelled by clauses. We now show how to label the intermediate vertices as well such that the entire tree forms a resolution refutation of f.

Going from the leaves up to the root, we label the intermediate vertices by the clause that is obtained by applying the resolution rule to the children. For example, if $(x \vee y \vee z)$ and $(\neg x \vee w)$ label two leaves of T, and if these two leaves are children of a parent vertex labelled by x, then we label the parent vertex by $(y \vee z \vee w)$.

In order to show that this yields a resolution refutation, we first argue that for each intermediate vertex v, if v is labelled by the variable x, with left child v_l labelled by clause C_l and right child v_r labelled by the clause C_r, then it is possible to apply the resolution rule to C_r and C_l resolving upon x in order to obtain a new clause C that will be the label for v. To see this, we just need to show that x occurs in both C_r and C_l, occurring positively in C_l and negatively in C_r.

This is true whenever both v_l and v_r are leaf vertices. To see this, first note that if x occurs in C_l it must occur positively since on the left branch we set $x = 0$ and if it occurred negatively then C_l would be set to 1 by the partial assignment, and thus f would be satisfiable. Thus if x occurs in C_l, it must occur positively. Similarly if x occurs in C_r it must occur negatively. It is left to argue that x is in C_l and $\neg x$ is in C_r whenever both v_l, v_r are leaf vertices. If it does not then since we know that the partial assignment to v_l falsifies C_l, it follows that the partial assignment to v already falsifies C_l and therefore we should have truncated at v. Thus we have argued that x occurs positively in C_l and similarly that x occurs negatively in C_r, and therefore we can apply the resolution rule to obtain the clause C from C_l and C_r.

However if v_l and v_r are not both leaf nodes, then it is possible that one of C_l or C_r to not contain x. For example consider the formula $f = (y \vee q)(y \vee \neg q)(w)(\neg w \vee \neg y \vee x)(\neg y \vee \neg w \vee z)(\neg y \vee \neg w \vee \neg z)$. Suppose that at the root we query y. On the left child we query q, and on the right child we query w. After querying w on the right side we then query x followed by z. (Need a picture here!) In this example, x is queried at level 2 from the root (call this vertex v), but v_l is labelled by $(\neg w \vee \neg y \vee x)$ and v_r is labelled by $(\neg y \vee \neg w)$. Thus, x does not occur in both C_l and C_r. If it occurs in both, then we can resolve on x as usual. Otherwise, it does not occur on at least one side – for sake of argument let's suppose that x does not occur in C_r. Then we can just label v with C_r. That is, the entire subproof rooted at v_l was unnecessary and can be removed. After this surgery (essentially replacing the subtree rooted at v with the subtree rooted at C_r) we can continue.

In all cases, we have shown that we can extract a Resolution refutation of f from the decision tree, and the size of the Resolution tree will be no greater than the size of the decision tree.

Finally, we want to argue that the root of the tree is labelled by the empty clause, and thus the entire annotated tree is indeed a resolution refutation. To see this, we will argue that for any vertex v, the set of variables occurring in C_v (the clause assocated with v) is a subset of the set of variables labelling the edges to v (from the root). Clearly this is true for the leaf nodes, since they were labelled by a clause that was falsified by the partial assignment to the leaf. For the inductive step, assume that vertex v has children v_l and v_r where v_l is labelelled by C_l and v_r is labelled by C_r. By induction, the variables occurring in C_l are a subset of the variables labelling the edges from the root to v_l, and similarly for C_r. Since we obtain C, the clause associated with vertex v by resolving on the literal queried at v, it follows that the clause at vertex v does not contain the literal queried at v, and thus the variables in C are also a subset of the variables on the path from the root to v.

This completes the argument.