Proof for 2-SAT

Burton Rosenberg

May 17, 2004

Let $\mathcal{C} = \{ c_i \}$ be a finite set of clauses, each c_i being of the form $(\gamma_1 \vee \gamma_2 \vee \ldots \vee \gamma_k)$, with each γ_j being either v or $\neg v$ for some variable v in the set of variables V. The form of 2-SAT is when each clause has at most two variables or negation of variables. The satisfaction of a set of clauses as an assignment of true values to each variable such that every clause yields true.

Lemma 1 *For a SAT instance $\mathcal{C}, V$ and a satisfying assignment, for any $\mathcal{C}' \subset \mathcal{C}$, the assignment also satisfies $\mathcal{C}'$.*

The basic procedure is to chose a variable among those appearing in $\mathcal{C}$ and assigning it a value, then simplify the set of clauses by the consequence of this assignment.

1. If the variable v is set true, remove all clauses which contain v. If v is set false, remove all clauses which contain $\neg v$.
2. If the variable v is set true and a clause contains $\neg v$, remove the variable from the clause. Likewise, if v is set false and a clause contains v, remove the variable from the clause.
3. Inspect the resulting set of clauses for obvious contradictions: a pair of clauses of the form (x) and $(\neg x)$. If such a contradiction occurs the basic procedure *rejects*.
4. Else repeatedly remove clauses with a single variable by assigning the appropriate value to the variable and doing the above simplifications. If the variable appears in the affirmative, set it true; if the variable appears negated, set it false.
5. Repeat this until either the procedure rejects; or the procedure leaves no clauses, in which case the procedure *succeeds*; or no clause has a single variable.

In the case of 2-SAT this procedure makes terrific headway, and is the basic algorithm to find a satisfying assignment.

Notation: Let $\mathcal{A}$ be a set of variable, value assignments. The basic procedure begins by inserting $v = T$ or $v = F$ into $\mathcal{A}$, and additional assignments are made. The result of the procedure is a set of clauses $\mathcal{C}' \subset \mathcal{C}$ after assignment and simplification, which is denoted $\mathcal{C}|_{\mathcal{A}}$.

Lemma 2 *Let $\mathcal{C}, V$ be an instance of 2-SAT, and $\mathcal{A}$ the result of the basic procedure. One of three cases holds:*

1. *If the basic procedure rejects starting from both $\mathcal{A} = \{ v = T \}$ and $\mathcal{A} = \{ v = F \}$. Then $\mathcal{C}$ is not satisfiable.*
2. *If the basic procedure succeed, that is $\mathcal{C}|_\mathcal{A}$ is empty. Then $\mathcal{C}$ is satisfiable and $\mathcal{A}$ is a satisfying assignment.*
3. *If the procedure ends with a reduced set of clauses $\mathcal{C}' \subset \mathcal{C}$, then $\mathcal{C}$ is satisfiable if and only if $\mathcal{C}'$ is satisfiable. Furthermore, if $\mathcal{A}'$ is an satisfying assignment for $\mathcal{C}'$ then $\mathcal{A} \cup \mathcal{A}'$ is a satisfying assignment for $\mathcal{C}$.*

Proof: All fairly obvious. The key point in the third case is that $\mathcal{C}'$ contains no variables assigned in $\mathcal{C}$, so that you are free to splice together the assignment satisfying $\mathcal{C}'$ with the already determined values listed in $\mathcal{A}$. If there is any satisfying assignment for $\mathcal{C}$, that assignment restricted to the variables found in $\mathcal{C}'$ is a sufficient $\mathcal{A}'$. On the other hand, if $\mathcal{C}'$ is not satisfiable, then $\mathcal{C}$ is not satisfiable regardless of the procedure by which we selected the subset $\mathcal{C}'$.

Problem 6.3.3 in Lewis and Papadimitriou suggest the following exercise. Encode 2-SAT as a directed graph. The vertex set is a pair of nodes for each variable, one to represent the variable in the affirmative, the other for the negation. For each clause $x \vee y$ (since logically $\neg x \Rightarrow y$ and $\neg y \Rightarrow x$) direct an edge from $\neg x$ to y and another from $\neg y$ to x.

Lemma 3 (Problem 6.3.3., L. and P.) *The instance $\mathcal{C}, V$ of 2-SAT is not satisfiable if and only if there exists an $v \in V$ such that in the directed graph there is a path from v to $\neg v$ and from $\neg v$ to v.*

Proof: (Only if) We follow the basic procedure until we discover $\mathcal{C}$ is not satisfiable. Hence the basic procedure beginning from $v = T$ rejects, as it does beginning with $v = F$. Suppose for $v = T$ it rejects because both x and $\neg x$ are resulting clauses. Following the procedure and interpreting in the graph, these eliminations demonstrate paths,

$$v \Rightarrow \ldots \Rightarrow x$$
$$v \Rightarrow \ldots \Rightarrow \neg x$$

The contrapositive paths are also available. In particular there is a path,

$$v \Rightarrow \ldots \Rightarrow x \Rightarrow \ldots \Rightarrow \neg v.$$

Likewise, for $v = F$ there is a variable y for which,

$$\neg v \Rightarrow \ldots \Rightarrow y \Rightarrow \ldots \Rightarrow v.$$

Hence we have the required pair of paths.