Understanding how to attack DES via SAT

William Jones - 635773

December 11, 2013

Contents

1 SAT

What is SAT? (brief introduction to the topic(defining definitions and how to build a framework to solve a SAT problem)only to be completed when the main body is finished).

1.1 Sets

A set is a collection of objects. To show that an object x is in the set X we write it is as $x \in X$. Sets can be elements of other sets which are referred to as subsets. To show that all subsets of a set X we write it as $\mathcal{P}(X)$.

Two sets are equivalent if they have the same exact elements in both of the sets. It should be noted that although the two sets have the same elements, the sets themselves can mean different things. Set X is in set Y if all elements of X are an element of Y.
Therefore we can say for the sets X and Y:

$$X = Y \text{ if and only if } Y \subseteq X \text{ and } X \subseteq Y$$

A set can have nothing within inside it (no elements) which is referred as the empty set. The empty set is denoted as the symbol $\emptyset$ or $\{\}$.

There are four operations that can be applied to sets that can produce new sets.

Intersection

The Intersection operation is taking two sets and finding the element that is in both set X and Y, $X \cap Y$. Therefore intersection is: $\{x : x \in Y \text{ and } x \in X\}$.

Union

The Union operation of two sets is finding an element that is in either both of the two sets or belongs in the set X or Y, $X \cup Y$. Therefore the Union of sets X and Y is: $\{x : x \in Y \text{ or } x \in X\}$.

Difference

The difference operation is finding an element that is in set X but not Y, $X \ / \ Y$. Therefore the difference of sets X and Y is: $\{x : x \in Y \text{ and } x \notin X\}$.

The sets we have been talking so far up to this point have elements within them that are not dependant on the order they are in. For example the set $\{x, y\}$ is the same as the set $\{y, x\}$ as they both have the same elements. However we can have ordered sets were the position of a element

within the set is important and makes the set unique. To define a set that is ordered we use <> , i.e., $<x, y>$.
The Cartesian product of set X and Y is $\{<x,y> : x \in X \text{ and } y \in Y\}$, so that $X \times Y$.
A relation is a subset of the Cartesian product of sets which is written as $R \subseteq X \times X$ [1].

A function relates an input to a output. Functions are denoted by a small "f" followed by its input. A simple example of this would be f(x) = x^2. When x is 4, f(4) = 16.

1.2 Syntax and semantics for propositional logic

A propositional variable is the starting point of propositional logic and is a alphabetic symbol that represents a object or number which is subject to change or is not known. We will refer to a set of variables as $\boldsymbol{VA}$. Let a stand for all variables, $a \in \boldsymbol{VA}$.
The symbol $\top$ will be used to show that a formula is always true and $\bot$ for when a formula is always false. The symbol $\neg$ will be used to show the negation of an object. The binary symbols $\wedge$ (conjunction), $\vee$ (disjunction), $\Rightarrow$ (implies) and $\equiv$ (equivalence) will be used throughout this paper and are known as *functors*[1].
A formula is defined as a set of strings over the set *Var* such that $\{Form : a, \neg, \top, \bot, \wedge, \vee, \Rightarrow, \equiv \in Form\}$.

A literal is a variable that is either positive or negative (denoted by the negation symbol) which will be referred to as $\boldsymbol{LIT} = \boldsymbol{VA} \cup \{ \bar{v} : v \in \boldsymbol{VA}\}$, for a set of literals.

A clause is a formula in the form $l_1 \vee ... \vee l_k$, where each l_j, $1 \leq j \leq k$ is a *literal* [1]. We will refer to a set of clauses as $\boldsymbol{CL} := \{\mathrm{C} \subseteq \boldsymbol{LIT}\ \mathrm{C} \cap \bar{C} = \emptyset\}$. A set of clause-set's contains $\boldsymbol{CL}$ which we will refer as $\boldsymbol{CLS} := \{F \subseteq \boldsymbol{CL}\}$

Partial Assignment

A partial assignment ($\boldsymbol{PASS}$) creates a mathematical object which can be instansiated by applying an instance of it to a clause set. Within the partial assignment there will be some undefinable variables. $\boldsymbol{PASS}$ is the set of all partial assignments. A total assignment ($\boldsymbol{TASS}$) is an assignment to all literals such that $\{\varphi \in \boldsymbol{TASS} : \varphi \in \mathrm{Var} \wedge \varphi \in \{0,1\}\}$. The task is to define $\varphi \times \mathrm{F}$ for partial assignment φ and clause set F. It should be noted that if a partial assignment has a mapping such that all variables have a assigned value $\varphi \in \{0,1\}$, then $\boldsymbol{PASS} = \boldsymbol{TASS}$. The relation of a partial assignment and a $\boldsymbol{CSL}$ can be denoted as:

$$* : \boldsymbol{PASS} \times \boldsymbol{CSL} \rightarrow \boldsymbol{CSL}$$

This will give us $\top := \emptyset$, so now we can have $\top \in \boldsymbol{CSL}$. Therefore:

$$\varphi * F = \top$$

φ is a satisfying assignment for F if $\varphi * = \top$. A clause set is satisfiable if there exists a partial assignment φ which satisfies F, i.e., $\varphi * = \top$.

Example of *PASS*

$<> \in \boldsymbol{PASS}$ is an example of the empty partial assignment.

$<v \rightarrow \epsilon>$ where $v \in Var \wedge v \in \{0,1\}$. v is assigned to ϵ which is within the ***PASS*** where v is also found as a member in the Var and is either 0 or 1.

1.3 Total Assignment

1.4 Partial Assignment

1.5 propositional Formula

1.6 Satisfiability

2 References

[1] Into to maths and Satisfiability.