Computational Methods for the Satisfiability Problem

Any set Γ of formulas is equivalent to a set of clauses—the set of all conjunctive terms of the CNFs of the elements of Γ. Consequently, the general satisfiability problem can be reduced to the satisfiability problem for sets of clauses.

A *unit* clause is a clause that consists of a single literal. If a set of clauses contains a unit clause then it can be simplified using the fact that $\{F, F \vee G\}$ is equivalent to $\{F\}$, and $\{F, \neg F \vee G\}$ is equivalent to $\{F, G\}$. A simplification step like this may create a new unit clause, and that can make further simplifications possible. This process is called *unit clause propagation.*

In the unit clause propagation procedure shown in Figure 1, Γ is a set of clauses, and U is a consistent set of literals such that, for any literal $L \in U$, neither L nor $\overline{L}$ occurs in any clause in Γ. During the execution of the procedure, Γ is simplified and U grows bigger, while the union $\Gamma \cup U$ remains equivalent to what it was originally. Upon termination, there are no unit clauses in Γ. To apply unit clause propagation to a given set Γ_0 of literals, UNIT-PROPAGATE is invoked with $\Gamma = \Gamma_0$ and $U = \emptyset$. Then, after every execution of the while loop, $\Gamma \cup U$ is equivalent to Γ_0.

Example 1. To apply unit clause propagation to the set

$$\{p, \neg p \vee \neg q, \neg q \vee r\} \tag{1}$$

we invoke UNIT-PROPAGATE with (1) as Γ and with the empty U. After the first execution of the body of the while loop,

$$\Gamma = \{\neg q, \neg q \vee r\} \text{ and } U = \{p\},$$

and after the second iteration

$$\Gamma = \emptyset \text{ and } U = \{p, \neg q\}.$$

This computation shows that (1) is equivalent to $\{p, \neg q\}$.

There are two cases when the process of unit clause propagation alone is sufficient for solving the satisfiability problem for Γ_0. Consider the values of

UNIT-PROPAGATE(Γ, U)
  while there is a unit clause $\{L\}$ in Γ
    $U := U \cup \{L\}$;
    for every clause $C \in \Gamma$
      if $L \in C$ **then** $\Gamma := \Gamma \setminus \{C\}$
      else $\Gamma := \Gamma \setminus \{C\} \cup \{C \setminus \{\overline{L}\}\}$
    end for
  end while

Figure 1: Unit clause propagation

Γ and U upon the termination of UNIT-PROPAGATE. *Case 1:* Γ includes the empty clause. Then Γ is unsatisfiable, and so is Γ_0. *Case 2:* $\Gamma = \emptyset$. Then Γ_0 is equivalent to U, which is a consistent set of literals; a model of Γ_0 can be easily extracted from U. For instance, the computation from Example 1 shows that any interpretation which maps p to t and q to f is a model of (1).

Next we discuss the *Davis-Putnam procedure* (Figure 2)—an extension of the unit clause propagation method that makes it capable of soving the satisfiability problem in full generality.

For any set Γ of formulas and any formula F, the set of models of Γ is the union of the set of models of $\Gamma \cup \{F\}$ and the set of models of $\Gamma \cup \{\neg F\}$. The Davis-Putnam procedure uses this fact to apply unit clause propagation even when Γ does not contain unit clauses.

In this recursive procedure, Γ is again a set of clauses, and U is a consistent set of literals such that, for any literal $L \in U$, neither L nor $\overline{L}$ occurs in any clause in Γ. The return in the first if statement of DP indicates that $\Gamma \cup U$ is unsatisfiable. The exit in the second if statement produces a model of $\Gamma \cup U$. Initially, DP is invoked with $\Gamma = \Gamma_0$ and $U = \emptyset$.

Example 2. Consider the application of the Davis-Putnam procedure to the set

$$\{\neg p \vee q, \neg p \vee r, q \vee r, \neg q \vee \neg r\}. \tag{2}$$

First DP is called with (2) as Γ and the empty U (Call 1). After the call to UNIT-PROPAGATE, the values of Γ and U remain the same. Assume that the literal selected as L is p. Now DP is called recursively with

$$\{\neg p \vee q, \neg p \vee r, q \vee r, \neg q \vee \neg r, p\}$$

DP(Γ, U)

Unit-propagate(Γ, U);

if $\emptyset \in \Gamma$ **then** return;

if $\Gamma = \emptyset$ **then** exit with a model of U;

$L :=$ a literal such that L or $\overline{L}$ occurs in Γ;

DP($\Gamma \cup \{L\}, U$);

DP($\Gamma \cup \{\overline{L}\}, U$)

Figure 2: The Davis-Putnam procedure

as Γ and $U = \emptyset$ (Call 2). After the call to Unit-propagate, $\emptyset \in \Gamma$. Next DP is called with

$$\{\neg p \vee q, \neg p \vee r, q \vee r, \neg q \vee \neg r, \neg p\}$$

as Γ and $U = \emptyset$ (Call 3). After the call to Unit-propagate, Γ is

$$\{q \vee r, \neg q \vee \neg r\}$$

and $U = \{\neg p\}$. Assume that the literal selected as L is q. Then DP is called with

$$\{q \vee r, \neg q \vee \neg r, q\}$$

as Γ and $U = \{\neg p\}$ (Call 4). After the call to Unit-propagate, $\Gamma = \emptyset$ and $U = \{\neg p, q, \neg r\}$. The computation produces the model of (2) that assigns the value t to q, and the value f to both p and r.