SAT Solvers

Sérgio Campos

Introduction

The Boolean Satisfiability Problem (SAT) appears in many contexts in the field of computer aided design such as

- Automatic Test Pattern Generation (ATPG)
- Timing analysis
- Delay fault testing
- Logic validation
- ...

Though well researched and widely investigated, it remains the focus of continuing interest because efficient techniques for its solution can have great impact.

Introduction

Over the years many algorithmic solutions have been proposed for SAT.
The most well known solutions are variants of the Davis-Putnam procedure.

1. Chooses one variable to assign a value; chooses one value
2. Determines the implications of this assignment in *unit clauses*
   - Deciding $a = 0$ with a clause $a \vee b$ implies $b = 1$
3. If there is a conflict, undo assignment, otherwise repeat.

SAT Solvers

Many SAT solvers exist:

- SATO
- Prover
- Grasp
- zChaff
- MiniSAT
- ...

We will see *Grasp*, an efficient SAT solver.

- And show how it differs from most of the other ones...

João Silva and Karen Sakallah. GRASP — A New Search Algorithm for Satisfiability. University of Michigan, Report CSE-TR-292-96, april 1996.

Definitions

- A *literal* is the occurrence of a variable or its complement, e.g., $x, \neg x$.
- A *clause* ω is the disjunction of one or more literals, e.g., $\omega = \neg x_2 \vee \neg x_3 \vee x_4$
- A *conjunctive normal form formula (CNF)* φ on n binary variables is the conjunction of m clauses $\omega_1, \ldots, \omega_m$. A formula φ denotes a unique n-variable boolean function $f(x_1, \ldots, x_n)$, e.g., $\varphi = f(x_1, x_2, x_3) = (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3)$
- The *satisfiability problem* is to determine an assignment of values to the variables x_1 to x_n that makes the f true, or to prove that f is equal to 0.

Definitions

- A variable is *assigned* it its value has already been determined.
- Otherwise the variable is *unassigned*
- A *truth assignment* is a set of assigned variables and their corresponding values, e.g., $A = \{(x_1, 0), (x_7, 1), (x_13, 0)\}$
- An assignment A is *complete* if $\mid A \mid = n$ otherwise it is *partial*.

Definitions

Evaluating a formula φ for a given truth assignment A yields three possible outcomes:

- $\varphi|_A = 1$: φ is satisfied, A is a satisfying assignment.
- $\varphi|_A = 0$: φ is unsatisfied, A is an unsatisfying assignment.
- $\varphi|_A = X$: The value of φ cannot yet be determined.

An assignment partitions the clauses of φ into three sets:

- Satisfied clauses which evaluate to 1;
- Unsatisfied clauses which evaluate to 0;
- Unresolved clauses, which evaluate to X;

The unassigned literals of a clause are referred to as its *free literals*. A clause is a *unit clause* if the number of its free literals is one.

The Search Process

1. Choose a variable to assign a value, and a value to assign to it.
   - Grasp chooses the variable that will satisfy the largest number of clauses.
2. Determine the implications of the current assignment. This is called *boolean constraint propagation*.
   - The additional assignments derived at this step are called *implications*.
3. If an implication makes the formula unsatisfiable, create a *conflict*.
   - Undo the current assignment, i.e., *backtrack*. Try a new one.
   - If there are no new assignment, *fail*.
4. If there is no conflict:
   - If the formula is unresolved, repeat.
   - If the formula is satisfied, *success*.

Example — The Clauses

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

In this example we will always choose the variables in the order a, b, c, d, e, f, g and always choose *false* first.

Assign $a \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

$a@1 = 0$

Assign $b \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

$a@1 = 0$

$b@2 = 0$

Assign $c \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

Assign $f \leftarrow true$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

ω_3 is a unit clause.

Decisions:

$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

Assign $f \leftarrow true$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

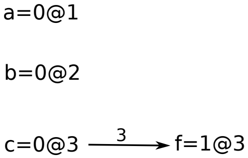
Decisions:

$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

Implications:



Assign $d \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

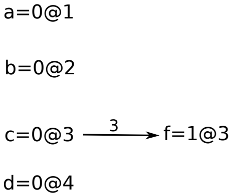
$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$

Implications:



Assign $e \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$

ω_2 is a unit clause

Assign $e \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

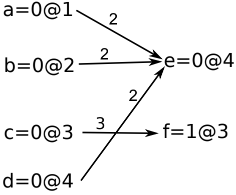
$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$

Implications:



Assign $e \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$

Decisions:

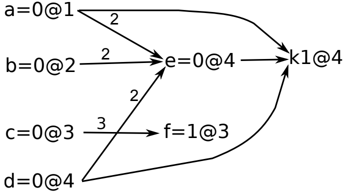
$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$

Implications:



Oops, conflict! Caused by $a = 0$, $b = 0$, and $d = 0$, i.e., $\neg a \wedge \neg b \wedge \neg d$.

Assign $e \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$
- $\omega_{k1} = a \vee b \vee d$

Decisions:

$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$

Conflict caused by $\neg a \wedge \neg b \wedge \neg d$. To avoid this conflict in the future, we add $a \vee b \vee d$ as a *conflict clause*.

Assign $d \leftarrow true$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$
- $\omega_{k1} = a \vee b \vee d$

Decisions:

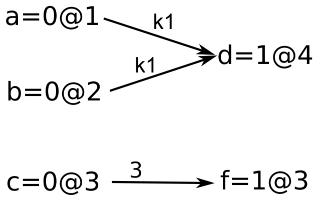
$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$ $d@4 = 1$

Implications:



ω_{k1} forces d to be true.

Assign $g \leftarrow true$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$
- $\omega_{k1} = a \vee b \vee d$

Decisions:

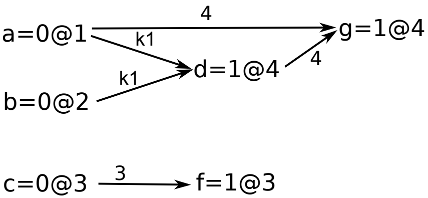
$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$ $d@4 = 1$

Implications:



ω_4 is a unit clause

Assign $g \leftarrow true$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$
- $\omega_{k1} = a \vee b \vee d$

Decisions:

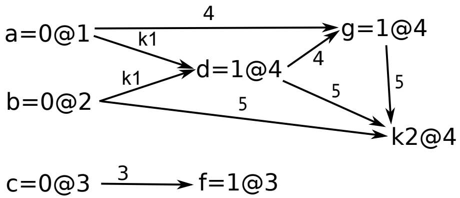
$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$ $\quad d@4 = 1$

Implications:



ω_5 causes conflict! Implication graph shows that decisions $a = 0$ and $b = 0$ are the cause.

Assign $g \leftarrow true$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$
- $\omega_{k1} = a \vee b \vee d$
- $\omega_{k2} = a \vee b$

Decisions:

$a@1 = 0$

$b@2 = 0$

$c@3 = 0$

$d@4 = 0$ $d@4 = 1$

- Because of the conflict we need to backtrack.
  - We have to undo assignment to d, but we tried 0 and 1.
  - So we should backtrack to $c \leftarrow 1$.
  - But the assignment to c has nothing to do with the conflict...
  - So we backtrack to the assignment to b

Grasp introduces the concept of *non-chronological* backtrack.

Undo the Assignment to b

- $\omega_1 = a \lor b \lor e \lor d$
- $\omega_2 = a \lor b \lor \neg e \lor d$
- $\omega_3 = c \lor f$
- $\omega_4 = \neg d \lor g \lor a$
- $\omega_5 = \neg d \lor \neg g \lor b$
- $\omega_{k1} = a \lor b \lor d$
- $\omega_{k2} = a \lor b$

Decisions:

$a@1 = 0$

$b@2 = 0$ $b@2 = 1$

$c@3 = 0$

$d@4 = 0$ $d@4 = 1$

Implications:

a=0@1

b=1@2

Assign $c \leftarrow$ *false* and $f \leftarrow$ *true*

- $\omega_1 = a \lor b \lor e \lor d$
- $\omega_2 = a \lor b \lor \neg e \lor d$
- $\omega_3 = c \lor f$
- $\omega_4 = \neg d \lor g \lor a$
- $\omega_5 = \neg d \lor \neg g \lor b$
- $\omega_{k1} = a \lor b \lor d$
- $\omega_{k2} = a \lor b$

Decisions:

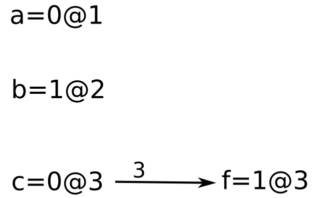
$a@1 = 0$

$b@2 = 0$ $b@2 = 1$

$c@3 = 0$

$d@4 = 0$ $d@4 = 1$

Implications:



Assign $d \leftarrow false$

- $\omega_1 = a \vee b \vee e \vee d$
- $\omega_2 = a \vee b \vee \neg e \vee d$
- $\omega_3 = c \vee f$
- $\omega_4 = \neg d \vee g \vee a$
- $\omega_5 = \neg d \vee \neg g \vee b$
- $\omega_{k1} = a \vee b \vee d$
- $\omega_{k2} = a \vee b$

Decisions:

$a@1 = 0$

$b@2 = 0 \quad b@2 = 1$

$c@3 = 0$

$d@4 = 0 \quad d@4 = 1$

Implications:

a=0@1

b=1@2

c=0@3 $\xrightarrow{3}$ f=1@3

No More Assignments!

All clauses are satisfied.
We are done! Can you believe it ? I thought it would never end...