



# **CS 228 : Logic in Computer Science**

Krishna. S

# The DPLL Algorithm

---

Input : CNF formula  $F$ .

1. Initialise  $\alpha$  as the empty assignment
2. While there is a unit clause  $L$  in  $F|_{\alpha}$ , add  $L = 1$  to  $\alpha$  (unit propagation)
3. If  $F|_{\alpha}$  contains no clauses, then stop and output  $\alpha$
4. If  $F|_{\alpha}$  contains the empty clause, then apply the learning procedure to add a new clause  $C$  to  $F$ . If it is the empty clause, output UNSAT. Otherwise, backtrack to the highest level at which  $C$  is a unit clause, go to line 2.
5. Decide on a new assignment  $p = b$  to be added to  $\alpha$ , goto line 2.

# DPLL Example

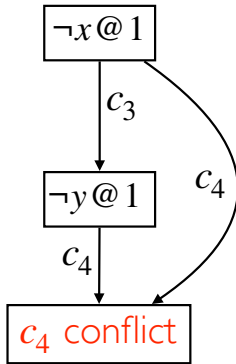
---

$$c_1 = \neg x \vee \neg y$$

$$c_2 = \neg x \vee y$$

$$c_3 = x \vee \neg y$$

$$c_4 = x \vee y$$



Clause learnt :  $x$  (Resolve  $c_4$  with  $c_3$ )

# DPLL Example

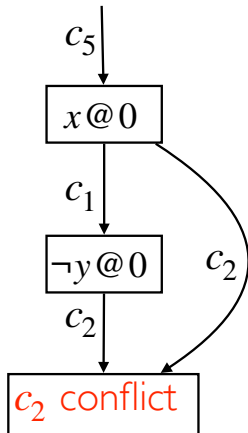
$$c_1 = \neg x \vee \neg y$$

$$c_2 = \neg x \vee y$$

$$c_3 = x \vee \neg y$$

$$c_4 = x \vee y$$

$$c_5 = x$$



Clause learnt : Resolve  $c_2$  with  $c_1, c_5$ . Empty clause.

# DPLL Correctness

---

## Termination

A sequence of decisions which lead to a conflict cannot be repeated : the variables in the learned clause are all decision variables. In a future assignment, if all but one of these are set to false, the remaining one will not be a decision variable.

## Correctness

Correctness is straightforward :  $F \vdash$  the learned clause. Thus, if the empty clause is learnt, then  $F$  is unsat. Otherwise, if DPLL terminates with a satisfying assignment  $\alpha$ , then the input formula is also satisfied by  $\alpha$ .

# Modern SAT Solvers

---

Numerous enhancements/heuristics

- ▶ Decision heuristics to choose decision variables
- ▶ Random restarts

## First Order Logic

# FOL

---

Extends propositional logic

- ▶ Propositional logic : atomic formulas have no internal structure
- ▶ FOL : atomic formulas are predicates that assert a relationship between certain elements
- ▶ Quantification in FOL : ability to assert that a certain property holds for all elements or only for some element.
- ▶ Formulae in FOL are over some signature.



# Signatures

---

- ▶ A **vocabulary** or **signature**  $\tau$  is a set consisting of
  - ▶ constants  $c_1, c_2, \dots$
  - ▶ Relation symbols  $R_1, R_2, \dots$ , each with some arity  $k$ , denoted  $R_i^k$
  - ▶ Function symbols  $f_1, \dots$  each with some arity  $k$ , denoted  $f_i^k$
- ▶ We look at finite signatures
- ▶  $\tau = (E^2, F^3, f^1)$  is a finite signature with two relations,  $E$  with arity 2 and  $F$  with arity 3, and a function  $f$  with arity 1

# Symbols in First Order Logic

---

Formulae of FO, over signature  $\tau$ , are sequences of symbols, where each symbol is one of the following:

- ▶ The symbols  $\perp, \top$  called **false, true**
- ▶ An element of the infinite set  $\mathcal{V} = \{x_1, x_2, \dots\}$  called **variables**
- ▶ Constants, relations and functions from  $\tau$
- ▶ The symbols  $\rightarrow, \neg, \wedge, \vee$
- ▶ The symbol  $\forall$  called the **universal quantifier**
- ▶ The symbol  $\exists$  called the **existential quantifier**
- ▶ The symbols  $($  and  $)$  called **paranthesis**

# Terms

---

Given a signature  $\tau$ , the set of  $\tau$ -terms are defined inductively as follows.

- ▶ Each variable is a term
- ▶ Each constant symbol is a term
- ▶ If  $t_1, \dots, t_k$  are terms and  $f$  is a  $k$ -ary function, then  $f(t_1, \dots, t_k)$  is a term
- ▶ Ground Terms : Terms without variables. For instance  $f(c_1, \dots, c_k)$  for constants  $c_1, \dots, c_k$ .

# Well formed Formulae

---

A well-formed formula (wff) over a signature  $\tau$  is inductively defined as follows:

- ▶  $\perp, \top$  are wff
- ▶ If  $t_1, t_2$  are either variables or constants in  $\tau$ , then  $t_1 = t_2$  is a wff
- ▶ If  $t_i$  is a term, for  $1 \leq i \leq k$  and  $R$  is a  $k$ -ary relation symbol in  $\tau$ , then  $R(t_1, \dots, t_k)$  is a wff
- ▶ If  $\varphi$  and  $\psi$  are wff, then  $\varphi \rightarrow \psi, \varphi \wedge \psi, \varphi \vee \psi, \neg\psi$  are all wff
- ▶ If  $\varphi$  is a wff and  $x$  is a variable, then  $(\forall x)\varphi$  and  $(\exists x)\varphi$  are wff
- ▶ The second and third are **atomic** formulae.
- ▶ If a formula  $F$  occurs as part of another formula  $G$ , then  $F$  is called a **sub formula** of  $G$ .

# Logical Abbreviations : Boolean Connectives

---

- ▶  $\neg\varphi = \varphi \rightarrow \perp$
- ▶  $\top = \neg\perp$
- ▶  $\varphi \vee \psi = \neg\varphi \rightarrow \psi$
- ▶  $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$
- ▶  $\exists x.\varphi = \neg(\forall x.\neg\varphi)$
- ▶ Precedence of operators : Quantifiers and negation highest, followed by  $\vee$ ,  $\wedge$ , followed by  $\rightarrow$ .
  - ▶  $\forall x P(x) \wedge R(x)$  is  $[\forall x.[P(x)]] \wedge R(x)$

# An Example

---

Consider the signature  $\tau = \{R\}$  where  $R$  is a binary relation. The following are FO formulae over this signature.

- ▶  $\forall x R(x, x)$  Reflexivity
- ▶  $\forall x (R(x, x) \rightarrow \perp)$  Irreflexivity
- ▶  $\forall x \forall y (R(x, y) \rightarrow R(y, x))$  Symmetry
- ▶  $\forall x \forall y \forall z (R(x, y) \rightarrow (R(y, z) \rightarrow R(x, z)))$  Transitivity