

Assignment satisfies a formula

$\mu \models \varphi$ if and only if μ "evaluates" φ to T

$$\text{Atoms}(\varphi) = \{A, B, C\}$$

$$\mu(A) := \perp$$

$$\mu(B) := T$$

$$\mu(C) := T$$

$$\varphi_1 \Rightarrow (A \xrightarrow{\perp} \overline{B}^T) \vee \neg(C \wedge \overline{B}^T)$$

$$\varphi_2 \Rightarrow \neg(\overline{A}^T \rightarrow \neg \overline{B}^T) \wedge \overline{C}^T$$

$$\varphi_3 \Rightarrow (\neg A \wedge \neg B) \rightarrow (A \wedge \neg C)$$

$$\varphi_4 \Rightarrow C \vee B$$

Does $\mu \models \varphi_1$? yes

$\mu \models \varphi_2$? yes

Is φ_1 satisfiable? yes, $\mu \models \varphi_1$

Is φ_1 valid? NO, let's show it. We need to build μ^* such that $\mu^* \not\models \varphi$

We need to make both $(A \rightarrow \neg B)$ and $\neg(C \wedge B)$ false

To make $A \rightarrow \neg B$ false we need $\mu^*(A) = T$
 $\mu^*(B) = T$

To make $\neg(C \wedge B)$ false we need $(C \wedge B)$ true so:

so at tk and $\mu^* = \{+, T, T\}$

$\mu^*(C) = T$
 $\mu^*(B) = T$

A formula φ is valid if $\neg \varphi$ is unsatisfiable

If $\neg \varphi$ is satisfiable then φ is not valid

so for example: $\varphi_1 = (A \rightarrow \neg B) \vee \neg(C \wedge B)$

$$\neg \varphi_1 = \neg((A \rightarrow \neg B) \vee \neg(C \wedge B))$$

$\underbrace{\quad T \quad T \quad}_{F} \quad \underbrace{\quad T \quad T \quad}_{F}$
 $\underbrace{\quad F \quad F \quad}_{F}$
 $\underbrace{\quad F \quad}_{+}$

Equivalence and Equisatisfiability

Example:

$$\varphi_1: A \vee B$$

$$\varphi_2: (A \vee B) \wedge (C \vee \neg C)$$

$$\mu \models \varphi_1 \text{ iff } \mu(A) = \mu(B) = \perp$$

$$\mu \models \varphi_2 \text{ iff } \mu'(A) = \mu'(B) = \perp$$

$$\mu'(C) = \perp \text{ or } \mu'(C) = \top$$

There are not equivalent but equisatisfiable

Conjunctive normal form

Example:

$$(A_1 \vee \neg A_2) \wedge$$

$$(A_3 \vee A_1 \vee \neg A_2) \wedge$$

$$(A_2 \vee \neg A_4)$$

CNF with 3 clauses

$$\bigwedge_{i=1}^3 \bigvee_{j=1}^n \ell_{i,j}$$

You can think of this like a system of equations and everyone must be satisfied. While for every clause you must check only one literal because if that's true then the clause is true.

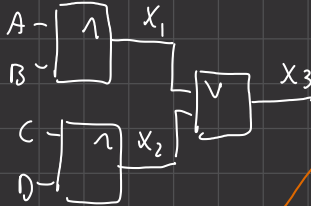
Another way of seeing CNF is seeing it like a set of literals:

$$\{\{A_1, \neg A_2\}, \{A_3, A_1, \neg A_2\}, \{A_2, \neg A_4\}\}$$

4/10/23

$$\text{Let's consider } \underbrace{(A \wedge B)}_{x_1} \vee \underbrace{(C \wedge D)}_{x_2}$$

Represented as a circuit



$$\varphi' = (x_1 \leftrightarrow A \wedge B) \wedge (x_2 \leftrightarrow C \wedge D) \wedge (x_3 \leftrightarrow x_1 \vee x_2) \wedge$$

φ' is not CNF

φ' is equisatisfiable

$$\mu(x_3) := \top$$

this must be true to be equisatisfiable

If this is true then either x_1 or x_2 must be true

$$\text{Assume } \mu'(x_1) := \top \text{ then } \mu'(A) := \top \text{ and } \mu'(B) := \top$$

$$\mu' \models \varphi'$$

$$\mu'(x_2) := \perp \text{ then either } \mu'(C) := \perp \text{ or } \mu'(D) := \perp$$

$$\text{From } \mu' \text{ we can extract: } \mu(A) := \top, \mu(B) := \top, \mu(C) := \perp, \mu(D) := \perp$$

We have found an assignment that satisfy the formula but it's still not CNF

Let's now build φ'' that will be CNF.

$$\begin{aligned}\varphi' = & (x_1 \leftrightarrow A \wedge B) \wedge \\ & (x_2 \leftrightarrow C \wedge D) \wedge \\ & (x_3 \leftrightarrow x_1 \vee x_2) \wedge \\ & x_3\end{aligned}$$

← these are constraints, not conjunctions.

φ' is equi-satisfiable w.r.t φ

Let's begin to build φ'' :

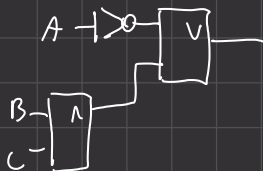
$$\begin{aligned}(x_1 \rightarrow (A \wedge B)) \wedge & (\neg x_1 \vee (A \wedge B)) \wedge \begin{cases} \neg x_1 \vee A & \wedge \\ \neg x_1 \vee B & \wedge \end{cases} \\ (A \wedge B \rightarrow x_1) \wedge & (\neg(A \wedge B) \vee x_1) \wedge \begin{cases} \neg A \vee \neg B \vee x_1 & \wedge \end{cases} \\ (x_2 \rightarrow (C \wedge D)) \wedge & (\neg x_2 \vee (C \wedge D)) \wedge \begin{cases} \neg x_2 \vee C & \wedge \\ \neg x_2 \vee D & \wedge \end{cases} \\ ((C \wedge D) \rightarrow x_2) \wedge & (\neg(C \wedge D) \vee x_2) \wedge \begin{cases} \neg C \vee \neg D \vee x_2 & \wedge \end{cases} \\ (x_3 \rightarrow (x_1 \vee x_2)) \wedge & (\neg x_3 \vee (x_1 \vee x_2)) \wedge \begin{cases} \neg x_3 \vee x_1 \vee x_2 & \wedge \end{cases} \\ ((x_1 \vee x_2) \rightarrow x_3) \wedge & (\neg(x_1 \vee x_2) \vee x_3) \wedge \begin{cases} \neg x_1 \vee \neg x_2 \vee x_3 & \wedge \\ \neg x_2 \vee x_3 & \wedge \end{cases} \\ x_3 & x_3 \end{aligned}$$

φ'' is equi-satisfiable w.r.t φ

The formula does not blow up because you add a variable for the new clauses. And you keep equi-satisfiability by maintaining equivalence between the transformations.

" φ is in CNF"

$$\begin{aligned}A \rightarrow (B \wedge C) \\ \downarrow \text{CNF} \\ \neg A \vee \underbrace{(B \wedge C)}_{x_1} \\ \underbrace{}_{x_2}\end{aligned}$$



$$\begin{aligned}x_1 \leftrightarrow B \wedge C & \Rightarrow \neg x_1 \vee B \\ x_2 \leftrightarrow \neg A \vee x_1 & \Rightarrow \neg x_1 \vee C \\ & \neg B \vee \neg C \vee x_1 \\ & \neg x_2 \vee \neg A \vee x_1 \\ & A \vee x_2 \\ & \neg x_2 \vee x_2 \\ & x_2\end{aligned}$$

$\mu \models \varphi$ is such that:
either $\mu(A) := \perp$
or $\mu(A) := \top, \mu(B) := \top, \mu(C) := \top$

$\mu' \models \varphi'$ is such that
 $\mu'(x_2) := \top$ then either: $\mu'(x_1) := \perp$ or $\mu'(x_2) := \top$

> ???

$\mu(x_2) := \top$ iff $\mu(B) := \top, \mu(C) := \top$

EXAMPLE FOR subsumed rule:

$$\varphi: \begin{array}{l} (A \vee B) \wedge \\ (\neg B \vee C) \wedge \\ (A \vee B \vee C) \end{array} \rightarrow \varphi' := \begin{array}{l} (A \vee B) \wedge \\ (\neg B \vee C) \end{array}$$

↓
because $A \vee B \models A \vee B \vee C$
 $A \vee B$ subsumes $A \vee B \vee C$

EXAMPLE

$$\varphi = \{ \{A, B, C\}, \{ \neg A, B \}, \{B, \neg C\} \}$$

$$\text{Assign}(A, \varphi) = \{ \{B\}, \{B, \neg C\} \} \rightarrow \text{If I assume that } A \text{ is true the first constraint can be removed because is or}$$

$$\text{Assign}(\neg A, \varphi) = \{ \{B, C\}, \{B, \neg C\} \}$$

$$\varphi_v^+ \quad \text{cl} \in \varphi \quad \forall v \in \text{cl}$$

$$\text{Assign}(V, \varphi)$$

$$\begin{array}{l} \text{remove} \quad \text{cl} \quad \varphi_v^+ \\ \text{insert} \quad \text{cl} \in \varphi_v^- \quad \text{cl} / \{ \neg V \} \end{array}$$

08/10/23

Resolution example

$$\varphi: \begin{array}{l} \{ \neg A, B, C \} \\ \{ \neg B, C \} \\ \{ A, \neg C \} \end{array} \quad \varphi': \begin{array}{l} \{ \{B, C\} \} \\ \{ \{B, C, \neg C\} \} \end{array}$$

V → ~~$\{B, C, \neg C\}$~~ *Tautology*

$$\frac{\neg A \vee B \vee C \quad A \vee \neg C}{B \vee C \vee \neg C}$$

Example 2:

$$\varphi = \begin{array}{l} \{ \neg A, B, \neg C \} \\ \{ A, D \} \\ \{ \neg A, \neg B, C \} \\ \{ A, B, \neg D \} \end{array}$$

Resolve away

$$\frac{\neg A \vee B \vee \neg C \quad A \vee D}{B \vee C \vee D} \quad (1,2)$$

$$\frac{\neg A \vee B \vee \neg C \quad A \vee B \vee \neg D}{B \vee \neg C \vee \neg D} \quad (2,4)$$

$$\frac{\neg A \vee \neg B \vee C \quad A \vee D}{\neg B \vee C \vee D} \quad (3,2)$$

$$\frac{\neg A \vee \neg B \vee C \quad A \vee B \vee \neg D}{\neg B \vee B \vee C \vee \neg D} \quad (3,4)$$

$$\neg B \vee B \vee C \vee \neg D$$

Tautology

- No unit clauses
- No pure literals
- No unit clauses

All possible combinations

$$\varphi^1 := \left\{ \left\{ B, \neg C, D \right\}, \left\{ B, \neg C, \neg D \right\}, \left\{ \neg B, C, D \right\} \right\}$$

No simplification possible so
resolve B

$$\frac{B \vee \neg C \vee D \quad \neg B \vee C \vee D}{\neg C \vee C \vee D \rightarrow \text{tautology}}$$

$$\frac{B \vee \neg C \vee \neg D \quad \neg B \vee C \vee D}{\neg C \vee C \vee D \vee \neg D \rightarrow \text{tautology}}$$

$$\varphi^2 = \left\{ \underbrace{\left\{ \neg C \vee C \vee D \right\}}_{\text{true}}, \underbrace{\left\{ \neg C \vee C \vee D \vee \neg D \right\}}_{\text{true}} \right\} \Rightarrow \text{Satisfiable}$$

Example of algorithm.

$$\varphi = \{ \{A, B\}, \{A, \neg B\}, \{ \neg A, B \}, \{ \neg A, \neg B \} \}$$

$$\varphi^1 = \{ \{B\}, \{ \neg B \} \}$$

Unit literal propagation on B

$$\varphi^2 = \{ \{ \} \}$$

φ^2 contains ^{only} an empty clause φ^2 is unsat so is φ .

DPLL example

$$\varphi := (\neg A \vee B \vee C) \wedge (\neg B \vee C) \wedge (\neg B \vee \neg C) \wedge (A \vee \neg B \vee \neg C)$$

Assign(B, φ)

$$C \wedge \neg C \wedge (A \vee \neg C)$$

\downarrow unit clause

$$\neg C \wedge (A \vee \neg C)$$

\downarrow literal prop

$$\frac{() \wedge A}{\text{unsat}}$$

unsat

Assign($\neg B, \varphi$)

$$(\neg A \vee C)$$

\downarrow pure literal on A

empty formula (SAT)

$$\mu(B) := \perp \quad \mu(A) := \perp \quad \mu(C) := \text{unsat}$$

DPLL builds a satisfying assignment.

SAT builds a partial assignment