Assignment satisfies a formula

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

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

$\mu(A) := \bot$
$\mu(B) := \top$
$\mu(C) := \top$

φ_1 1) $(\overset{\bot}{A} \to \overset{\top}{B}) \vee \neg(\overset{\top}{C} \wedge \overset{\bot}{B})$
φ_2 2) $\neg(\overset{\bot}{A} \to \neg\overset{\top}{B}) \wedge \overset{\top}{C}$
φ_3 3) $(\neg A \wedge \neg B) \to (A \wedge \neg C)$
φ_4 4) $C \vee B$

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 $(A \to \neg B)$ and $\neg(C \wedge B)$ false
To make $A \to \neg B$ false we need $\mu^*(A) := \top$
$\mu^*(B) = \top$

To make $\neg(C \wedge B)$ false we need $(C \wedge B)$ true so:
$\mu^*(C) = \top$
$\mu^*(B) = \top$

so at the end $\mu^* = \{\bot, \top, \top\}$

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

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

So for example: $\varphi_1 := (A \to \neg B) \vee \neg(C \wedge B)$

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

T T T T T
F F
F
T

Equivalence and Equisatisfiability

Example:

φ_1 : $A \vee B$

φ_2 : $(A \vee B) \wedge (C \vee \neg C)$

$\mu \not\models \varphi_1$ iff $\mu(A) = \mu(B) = \bot$

$\mu \not\models \varphi_2$ iff $\mu'(A) = \mu'(B) = \bot$

$\mu'(C) = \bot$ or $\mu'(C) = \top$

These are not equivalent but equisatisfiable

Conjunctive normal form

Example:

$$\left.\begin{array}{l}(A_1 \vee \neg A_2) \wedge \\ (A_3 \vee A_1 \vee \neg A_2) \wedge \\ (A_2 \vee \neg A_4)\end{array}\right\} \text{CNF with 3 clauses}$$

$\bigwedge_{i=1}^{3} \bigvee_{j=1}^{n_i} \ell_{ij}$

$n_1 = 2$, $\ell_{11} = A_1$

$n_2 = 3$, $\ell_{22} = A_2$

$n_3 = 2$, $\ell_{31} = A_2$

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

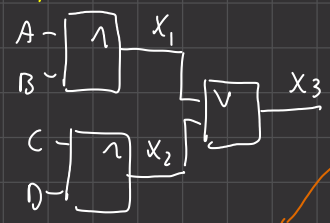
Another way of seeing CNF is seeing it like a set of literals:

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

4/10/23

Let's consider $\underbrace{\underbrace{(A \wedge B)}_{x_1} \vee \underbrace{(C \wedge D)}_{x_2}}_{x_3}$

Represented as a circuit



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

x_3 is the final output

φ' is not CNF

φ' is equisatisfiable

$\mu(x_3) := \top$ this must be true to be equisatisfiable

either $\mu(x_1) := \top$ or $\mu(x_2) := \top$ If this is true then either x_1 or x_2 must be true

Assume $\mu'(x_1) := \top$ then $\mu'(A) := \top$ and $\mu(B) := \top$ $\quad \mu' \models \varphi'$

$\mu'(x_2) := \bot$ then either $\mu'(C) := \bot$ or $\mu'(D) = \bot$

From μ' we can extract: $\mu(A) := \top$, $\mu(B) := \top$, $\mu(C) := \bot$, $\mu(D) := \bot$

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

Let's now build φ'' that will be CNF.

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

← there are constraints, not of conjunctions.

φ' is equi-satisfiable w.r.t φ

Let's begin to build φ'':

$(x_1 \rightarrow (A \wedge B)) \wedge$
$((A \wedge B) \rightarrow x_1) \wedge$
$(x_2 \rightarrow (C \wedge D)) \wedge$
$((C \wedge D) \rightarrow x_2) \wedge$
$(x_3 \rightarrow (x_1 \vee x_2)) \wedge$
$((x_1 \vee x_2) \rightarrow x_3) \wedge$
x_3

$(\neg x_1 \vee (A \wedge B)) \wedge$
$(\neg (A \wedge B) \vee x_1) \wedge$
$(\neg x_2 \vee (C \wedge D)) \wedge$
$(\neg (C \wedge D) \vee x_2) \wedge$
$(\neg x_3 \vee (x_1 \vee x_2)) \wedge$
$(\neg (x_1 \vee x_2) \vee x_3) \wedge$
x_3

$\neg x_1 \vee A \quad \wedge$
$\neg x_1 \vee B \quad \wedge$
$\neg A \wedge \neg B \vee x_1 \wedge$
$\neg x_2 \vee C \quad \wedge$
$\neg x_2 \vee D \quad \wedge$
$\neg C \vee \neg D \vee x_2 \wedge$
$\neg x_3 \vee x_1 \vee x_2 \wedge$
$\neg x_1 \vee x_3 \quad \wedge$
$\neg x_2 \vee x_3 \quad \wedge$
x_3

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

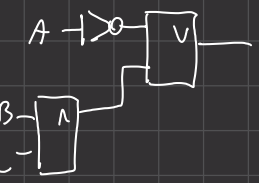
$A \rightarrow (B \wedge C)$

↓ NNF

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

x_1

x_2



$x_1 \leftrightarrow B \wedge C$

$x_2 \leftrightarrow \neg A \vee x_1$

x_2

$\Rightarrow$

$\neg x_1 \vee B$
$\neg x_1 \vee C$
$\neg B \vee \neg C \vee x_1$

$\neg x_2 \vee \neg A \vee x_1$
$A \vee x_2$
$\neg x_2 \vee x_2$

x_2

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

$\mu' \models \varphi'$ is such that

$\mu'(x_2) := \top$ then either: $\mu'(A) := \bot$ or $\mu'(x_2) := \top$

↪ ???

$\Downarrow$

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

EXAMPLE FOR subsumed rule:

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

because $A \wedge B \models A \vee B \vee C$

$A \vee B$ subsumes $A \vee B \vee C$

EXAMPLE

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

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

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

φ_V^+ $\quad CL \in \varphi \quad V \in CL$

$Assign(V, \varphi)$ remove CL φ_V^+

Remove $CL \in \varphi_V^-$ $\quad CL / \{\neg V\}$

05/10/23

Resolution example

φ: $\{\{\neg A, B, C\}, \{\neg B, C\}, \{A, \neg C\}\}$ $\quad$ φ': $\{\{B, C\}, \{B, \neg C\}\}$ Tautology

V

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

Example 2:

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

Resolve away

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

$$\frac{\neg A \vee B \vee \neg C \qquad A \vee B \vee \neg D}{B \vee \neg C \vee \neg D} \quad (1,4)$$

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

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

Tautology

All pairs tautology

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

$\varphi' := \{\{B, \neg C, D\},$
$\{B, \neg C, \neg D\},$
$\{\neg B, C, D\}\}$

No simplification possible to resolve B.

$\dfrac{B \vee \neg C \vee D \quad \neg B \vee C \vee D}{\neg C \vee C \vee D}$ → tautology

$\dfrac{B \vee \neg C \vee \neg D \quad \neg B \vee C \vee D}{\neg C \vee C \vee D \vee \neg D}$ → tautology

$\varphi'' = \{\{\underbrace{\neg C \vee C \vee D}_{\text{taut}}\}, \{\underbrace{\neg C \vee C \vee D \vee \neg D}_{\text{taut}}\}\} \Rightarrow$ satisfiable

---

Example of algorithm.

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

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

Unit literal propagation on B

$\varphi'' = \{\square\}$

φ'' contains (only) an empty clause φ'' 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)

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

↓ unit clause

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

↓ literal prop

$C \wedge A$

$\top$

unsat

Assign $(\neg B, \varphi)$

$(\neg A \vee C)$

| pure literal on A

empty formula (SAT)

$\mu(B) := \bot \quad \mu(A) := \top \quad \mu(C) :=$ anything

DPLL builds a satisfying assignment.

SAT builds a partial assignment