Assignment satisfies a formula

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

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

$\mu(A) := \perp$
$\mu(B) := T$
$\mu(C) := T$

φ_1 1) $(A \to B) \vee \neg(C \wedge B)$
φ_2 2) $\neg(\neg A \to \neg B) \wedge C$
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 $\mu^*(A) = T$
$\mu^*(B) = T$

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

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

$\boxed{\text{A formula } \varphi \text{ is valid if } \neg\varphi \text{ is unsatisfiable}}$

$\boxed{\text{If } \neg\varphi \text{ is satisfiable then } \varphi \text{ is not valid}}$

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

$$\neg\varphi_1 = \neg\underbrace{\left(\underbrace{\underbrace{(A \to \neg B)}_{F} \vee \underbrace{\neg(C \wedge B)}_{F}}_{F}\right)}_{T}$$

Equivalence and Equisatisfiability

Example:

$\varphi_1 : A \vee B$

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

$\mu \not\models \varphi_1$ 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:

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

$\bigwedge_{i=1}^{3} \bigvee_{j=1}^{n_i} l_{ij}$ $n_1 = 2$ $l_{11} = A_1$

$n_2 = 3$ $l_{22} = A_2$

$n_3 = 2$ $l_{31} = A_2$

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

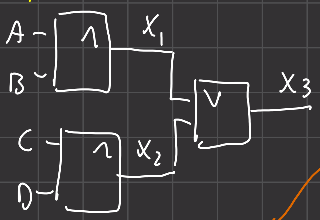
Another way of seeing CNF is seeing it like a set of literals:

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

4/10/23

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

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 $\mu(B) := \top$ $\mu' \models \varphi'$

$\mu'(x_2) := \bot$ 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.

$$\varphi' = (x_1 \leftrightarrow A \land B) \land \\ (x_2 \leftrightarrow C \land D) \land \\ (x_3 \leftrightarrow x_1 \lor x_2) \land \\ x_3$$

these are constraints, not conjunctions.

φ' is equi-satisfiable w.r.t φ

Let's begin to build φ'':

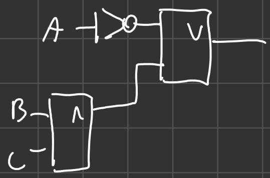
$(x_1 \to (A \land B)) \land$	$(\neg x_1 \lor (A \land B)) \land$	$\neg x_1 \lor A \quad \land$ $\neg x_1 \lor B \quad \land$
$((A \land B) \to x_1) \land$	$(\neg (A \land B) \lor x_1) \land$	$\neg A \land \neg B \lor x_1 \quad \land$
$(x_2 \to (C \land D)) \land$	$(\neg x_2 \lor (C \land D)) \land$	$\neg x_2 \lor C \quad \land$ $\neg x_2 \lor D \quad \land$
$((C \land D) \to x_2) \land$	$(\neg (C \land D) \lor x_2) \land$	$\neg C \lor \neg D \lor x_2 \quad \land$
$(x_3 \to (x_1 \lor x_2)) \land$	$(\neg x_3 \lor (x_1 \lor x_2)) \land$	$\neg x_3 \lor x_1 \lor x_2 \quad \land$
$((x_1 \lor x_2) \to x_3) \land$	$(\neg (x_1 \lor x_2) \lor x_3) \land$	$\neg x_1 \lor x_3 \quad \land$ $\neg x_2 \lor x_3 \quad \land$
x_3	x_3	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

$$A \to (B \land C)$$
$$\Big\downarrow \text{NNF}$$
$$\underbrace{\neg A \lor \underbrace{(B \land C)}_{x_1}}_{x_2}$$



$$\begin{aligned} &x_1 \leftrightarrow B \land C \\ &x_2 \leftrightarrow \neg A \lor x_1 \\ &x_2 \end{aligned} \implies \begin{aligned} &\neg x_1 \lor B \\ &\neg x_1 \lor C \\ &\neg B \lor \neg C \lor x_1 \\ \\ &\neg x_2 \lor \neg A \lor x_1 \\ &A \lor x_2 \\ &\neg x_1 \lor x_2 \\ \\ &x_2 \end{aligned}$$

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

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

$\mu'(x_2) := T$ then either: $\mu'(A) := \bot$ or $\mu'(x_2) := T$

???

$\Downarrow$

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

EXAMPLE FOR subsumed rule:

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

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

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

EXAMPLE

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

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

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

$\varphi_v^+ \quad cl \in \varphi \quad v \in cl$

$Assign(v, \varphi)$ remove cl φ_v^+

Replace $cl \in \varphi_v^-$ $cl / \{\neg v\}$

05/10/23

Resolution example

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

$\varphi': \{\{B, C\}, \cancel{\{B, C, \neg C\}}\}$ Tautology

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

Example 2:

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

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

Resolve away

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

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

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

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

All possible combination

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

No simplification possible so resolve B

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

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

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

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

φ'' contains only an empty clause φ'' is unsat e 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

$() \wedge A$

$\bot$

unsat

Assign$(\neg B, \varphi)$

$(\neg A \vee C)$

| pure literal on A

empty formula (SAT)

$\mu(B) := \bot \quad \mu(A) := \bot \quad \mu(C) := \text{undef}$

DPLL builds a satisfying assignment.

SAT builds a partial assignment

11/20/23

f function symbol 1-place

not ground -> $f(x)$ x is a variable

ground -> $f(s)$ s is a constant   0-ary funtion

not ground -> $f(f(x))$

P predicate symbol   1-place

Q predicate symbol   2-place

$P(x)$; $P(f(x))$, $P(s)$

$Q(x,s)$ $Q(f(x),s)$, $Q(s,s)$

all atomic

Examples of WFFs

Atoms are predicates and arguments of predicates are terms.

for every ... then

$\forall x. (P(x) \to Q(x,s))$

there exist

$\exists x. Q(x, f(x))$

closed

$Q(x,x) \wedge P(f(x))$ not closed

binds

$\forall x. \forall y. (P(x,y) \to \exists y. Q(y))$

binds   binds

12/20/23   Ex slide 32 (1)

1) $\varphi := \exists x. \exists y. P(f(x,y), 2)$

$\models_I \varphi$ iff $\models_I \exists y\, P(f(d_1,y),2)$ for some $d_1 \in D\,(d_1 \in \mathbb{N})$

iff $\models_I P(f(d_1,d_2),2)$ for some $d_1,d_2 \in D\,(d_1,d_2 \in \mathbb{N})$

iff $(g(f)(d_1,d_2), g(2)) \in g(P)$

$(\cdot(d_1,d_2), 2) \in \leq$

2) This one it's not true in any case because I can find a countermodel.

φ': $\forall x. \forall y. (P(x,y) \rightarrow P(y,x))$ symmetry

$\models_I \neg\varphi'$ $\exists x. \exists y. (P(x,y) \wedge \neg P(y,x))$

also we can write: $\not\models_I \varphi'$

$\models_I \varphi'$ iff for any two $d_1, d_2 \in \mathbb{N}$

I have that $d_1 \leq d_2$ implies $d_2 \leq d_1$ but if I choose $d_1 = 0$ and $d_2 = 1$ I see that $0 \leq 1$ but $1 \not\leq 0$ $\Rightarrow$ counter model.

3)

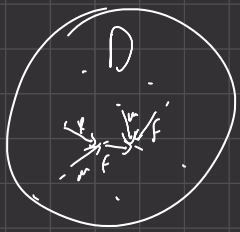
$\forall x. \forall y. \forall z. (((P(x,y) \wedge P(y,z)) \rightarrow P(x,z))$ transitivity

We can try to check if $d_1, d_2, d_3 \in \mathbb{N}$ such that $d_1 \leq d_2$, $d_2 \leq d_3$ but $d_1 \leq d_3$? No, therefore $\models_I \varphi$

the interpretation satisfies φ.

3g)

Let's think this as a finite Domain



I is such that $\models_I \forall x.\, Person(x)$

D is finite

$y(Person) \equiv D$

If the domain is finite then at a certain point there would be no way of choosing father or mother of some elements.

18/10/23

Skolem normal form

Example

$\forall x_1. \forall x_2. \exists y. \forall x_3. \forall x_4.\ \underbrace{\varphi(x_1, x_2, y, x_3, x_4)}$

form without quantifiers with $x_1, x_2, \varphi, x_3, x_4$ free variables

A model for φ is one where:

- given any choice of x_1
- given any choice of x_2
- there exists a choice for y such that given any choice for x_3, x_4 the formula is true

So now the formula transformed in Skolem would be:

$$\forall x_1. \forall x_2. \forall x_3. \forall x_4\, \varphi(x_1, x_2, f(x_1, x_2), x_3, x_4)$$

The order is important so your choice for y depends on your choice of x_1, x_2. so y becomes a function of x_1 and x_2.

Exercise:

$$\forall x. (P(x) \rightarrow \neg \forall z. \exists y. (R(x,z) \wedge Q(x,y)))$$

NNF

$$\forall x. (\neg P(x) \vee \exists z. \forall y (\neg R(x,z) \vee \neg Q(x,y)))$$

PNF

$$\forall x. \exists z. \forall y. (\underbrace{\neg P(x) \vee \neg R(x,z) \vee \neg Q(x,y)}_{\text{CNF with one clause or DNF with 3 terms.}})$$

SNF

$$\forall x. \forall y. (\neg P(x) \vee \neg R(x, \underbrace{f(x)}_{\text{Skolem function for } z}) \vee \neg Q(x,y))$$

The advantage of SNF is that has only universally quantifiers.

Herbrand

$$D_H: \{ a, g(f(a)), g(f(g(f)(a)) \ldots \}$$ -> This is the domain for the previous formula.

↑ infinite domain.