

# CS 511, Fall 2020, Lecture Slides 05

## Propositional Logic:

Conjunctive Normal Forms,  
Disjunctive Normal Forms,  
Horn Formulas,  
and other special forms

Assaf Kfoury

08 September 2020

## *conjunctive normal form & disjunctive normal form*

### **CNF**

$L ::= p \mid \neg p$	literal
$D ::= L \mid L \vee D$	disjunction of literals
$C ::= D \mid D \wedge C$	conjunction of disjunctions

### **DNF**

$L ::= p \mid \neg p$	literal
$C ::= L \mid L \wedge C$	conjunction of literals
$D ::= C \mid C \vee D$	disjunction of conjunctions

# Why CNF?

- ▶ A disjunction of literals  $L_1 \vee \dots \vee L_m$  is **valid** (or a **tautology**) iff there are  $1 \leq i, j \leq m$  with  $i \neq j$  such that  $L_i$  is  $\neg L_j$ .
- ▶ A conjunction of disjunctions  $D_1 \wedge \dots \wedge D_n$  is **valid** (or a **tautology**) iff for every  $1 \leq i \leq n$  it is the case that  $D_i$  is valid.
- ▶ **CNF** allows for a fast and easy syntactic test of **validity**.
- ▶ Unfortunately, conversion into **CNF** may lead to exponential blow-up:

$(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \dots \vee (x_n \wedge y_n)$  becomes

$$(x_1 \vee \dots \vee x_{n-1} \vee x_n) \wedge (x_1 \vee \dots \vee x_{n-1} \vee y_n) \wedge \dots \wedge (y_1 \vee \dots \vee y_{n-1} \vee y_n)$$

*i.e.*, the initial WFF of size  $\mathcal{O}(n)$  becomes an equivalent WFF of size  $\mathcal{O}(2^n)$ , because each clause in the latter contains either  $x_i$  or  $y_i$  for every  $i$ .

- ▶ Converting a WFF into an equivalent WFF in **CNF**, preserving **validity**, is NP-hard!

(However, converting a WFF into another WFF, not necessarily equivalent, preserving **satisfiability** can be carried out in linear time – more in a later handout.)

# Why DNF?

- ▶ A conjunction of literals  $L_1 \wedge \cdots \wedge L_m$  is **satisfiable** iff  $\{L_1, \dots, L_m\}$  does not include both a propositional atom  $P$  and its negation  $\neg P$ .
- ▶ A disjunction of conjunctions  $C_1 \vee \cdots \vee C_n$  is **satisfiable** iff there is some  $1 \leq i \leq n$  such that  $C_i$  is satisfiable.
- ▶ **DNF** allows for a fast and easy syntactic test of **satisfiability**.
- ▶ Unfortunately, conversion into **DNF** may lead to exponential blow-up:

$(x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge \cdots \wedge (x_n \vee y_n)$  becomes

$$(x_1 \wedge \cdots \wedge x_{n-1} \wedge x_n) \vee (x_1 \wedge \cdots \wedge x_{n-1} \wedge y_n) \vee \cdots \vee (y_1 \wedge \cdots \wedge y_{n-1} \wedge y_n)$$

*i.e.*, the initial WFF of size  $\mathcal{O}(n)$  becomes an equivalent WFF of size  $\mathcal{O}(2^n)$ , because each clause in the latter contains either  $x_i$  or  $y_i$  for every  $i$ .

- ▶ Converting a WFF into an equivalent WFF in **DNF**, preserving **satisfiability**, is NP-hard!

## further comments on CNF and DNF, summing up:

- ▶ propositional WFF's can be partitioned into three disjoint subsets:
  1. tautologies, or **unfalsifiable** WFF's
  2. contradictions, or **unsatisfiable** WFF's
  3. WFF's that are both **satisfiable** and **falsifiable**
- ▶ satisfiability of:
  - ▶ **CNF** is in NP
  - ▶ **DNF** is in P
- ▶ tautology of:
  - ▶ **CNF** is in P
  - ▶ **DNF** is in co-NP
- ▶ falsifiability of:
  - ▶ **CNF** is in P
  - ▶ **DNF** is in NP

## other special forms of propositional WFF's:

- ▶ One such form is that of the WFF's in **negation normal form (NNF)**: the negation operator ( $\neg$ ) is only applied to variables, and the only logical operators are conjunction ( $\wedge$ ) and disjunction ( $\vee$ ).
- ▶ More formally:

$$L ::= p \mid \neg p$$

$$\varphi ::= L \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

## other special forms of propositional WFF's:

- ▶ **Fact:** Every WFF in **CNF** or in **DNF** is also in **NNF**, but the converse is not true in general. See next slide for an example.

- ▶ **Fact:** **NNF** is **not** a canonical form, in contrast to **CNF** and **DNF**.<sup>1</sup>

**Example:**  $x \wedge (y \vee \neg z)$  and  $(x \wedge y) \vee (x \wedge \neg z)$  are equivalent and both in **NNF**.

- ▶ **Fact:** Every propositional WFF  $\varphi$  can be translated in linear time into an equivalent propositional WFF  $\psi$  in **NNF** such that  $|\psi| < (3/2) \cdot |\varphi|$ .

**Proof.** Left to you.

---

<sup>1</sup> Strictly speaking, not quite yet, because *canonical* implies *unique in its form*. However, a **CNF** (resp. a **DNF**) can be written *uniquely* as a conjunction of *maxterms* (resp. disjunction of *minterms*). Look up definitions of *maxterms* and *minterms* on the Web.





## another special form of propositional WFFs: *Decomposable Negation Normal Form (DNNF)*

A propositional WFF  $\varphi$  is a **decomposable negation normal form (DNNF)** if it is a **NNF** satisfying the **decomposability property**:

for every conjunction  $\psi = \psi_1 \wedge \cdots \wedge \psi_n$  which is a sub-WFF of  $\varphi$ , no propositional variable/atom is shared by any two distinct conjuncts of  $\psi$ :

$$\text{Var}(\psi_i) \cap \text{Var}(\psi_j) = \emptyset \quad \text{for every } i \neq j$$

**Example:** The **NNF** shown on page 8 is in fact a **DNNF**.

**Fact:** Satisfiability of WFF in **DNNF** is decidable in linear time.

## an important restricted class: *Horn formulas*

$$P ::= \perp \quad | \quad \top \quad | \quad p$$

$$A ::= P \quad | \quad P \wedge A$$

$$C ::= A \rightarrow P \quad \text{Horn clause}$$

$$H ::= C \quad | \quad C \wedge H \quad \text{Horn formula}$$

**Fact:** Satisfiability of Horn clauses is decidable in linear time.

**Proof:** To see this, rewrite a Horn clause into an equivalent disjunction of literals:

$$L_1 \wedge \cdots \wedge L_n \rightarrow L \quad \equiv \quad \neg L_1 \vee \cdots \vee \neg L_n \vee L.$$

**Fact:** Satisfiability of Horn formulas is decidable in linear time.

**Exercise** Search the Web to identify one or two applications, or areas of computer science, where each of the following forms are encountered:

1. Propositional WFF's in **NNF**.
2. Propositional WFF's in **DNNF**.
3. Propositional WFF's that are **Horn** formulas.

(THIS PAGE INTENTIONALLY LEFT BLANK)