Conjunctive Normal Form and Disjunctive Normal Form

Javier Escalada Gómez

1 Previous concepts

A formula F is a **conjunction** if it follows the following structure:

$$F_1 \wedge F_2 \wedge \ldots \wedge F_n$$

Similarly, it is a **disjunction** if it follows this other structure:

$$F_1 \vee F_2 \vee \ldots \vee F_n$$

A **literal** is either a proposition (p, q, r, ...) or its negation ($\neg p$, $\neg q$, $\neg r$, ...).

2 Conjunctive Normal Form (CNF)

A formula F is in **Conjunctive Normal Form** (CNF) if it is a **conjunction**

$$F_1 \wedge F_2 \wedge \ldots \wedge F_n$$

where each F_i is a **disjunction** of **literals**. Also, each literal can only appear once in each disjunction. Therefore, the same literal cannot appear in a disjunction together with its negation.

2.1 Example

The following formulas are in CNF:

1. $\neg p \wedge (q \vee r)$
2. $(p \vee q) \wedge (\neg q \vee r \vee \neg s) \wedge (s \vee \neg t)$
3. $p \wedge q$
4. p
5. $\neg q$

And, the following ones are **not** in CNF:

1. $\neg(q \vee r)$
2. $(p \wedge q) \vee r$
3. $p \wedge (q \vee (s \wedge t))$
4. $(p \vee \neg p) \wedge \neg q$

> **Exercise:** Why are they not in CNF?

but they can be transformed to CNF:

1. $\neg q \wedge \neg r$
2. $(p \vee r) \wedge (q \vee r)$
3. $p \wedge (q \vee s) \wedge (q \vee t)$

4. $\neg q$

Exercise: What transformations have been applied in each formula?

3 Disjunctive Normal Form (DNF)

A formula F is in **Disjunctive Normal For** (DNF) if it is a **disjunction**

$$F_1 \vee F_2 \vee \ldots \vee F_n$$

where each F_i is a **conjunction** of **literals**. As in CNF, each literal can only appear once in each conjunction.

3.1 Example

The following formulas are in DNF:

1. $p \wedge q$
2. p
3. $(p \wedge q) \vee r$
4. $(p \wedge \neg q \wedge \neg r) \vee (\neg s \wedge t \wedge u)$

However, the following ones are **not** in DNF:

1. $\neg(p \vee q)$
2. $p \vee (q \wedge (r \vee s))$

Exercise: Why are they not in DNF?

Exercise: Transform them into DNF

4 An example to simplify a formula into CNF

We start from the expression:

$$\neg(p \wedge \neg q) \wedge (r \vee s) \wedge (r \vee \neg s)$$

We apply De Morgan to the first conjunction:

$$(\neg p \vee q) \wedge (r \vee s) \wedge (r \vee \neg s)$$

Although it is a conjunction of disjunctions, it can be simplified further. So, we group the last 2 disjunctions:

$$(\neg p \vee q) \wedge ((r \vee s) \wedge (r \vee \neg s))$$

We apply the distributive law:

$$(\neg p \vee q) \wedge (r \vee (s \wedge \neg s))$$

We apply the law of complements:

$$(\neg p \vee q) \wedge (r \vee F)$$

And we apply the law of identity:

$$(\neg p \vee q) \wedge r$$

5 How to check the exercises?

There are different pages on the Internet that allow you to simplify Boolean expressions:

- https://www.dcode.fr/boolean-expressions-calculator
- https://srexamen.com/mathlogic

Exercise: Test these web applications with the previous example

```
not (p and not q) and (r or s) and (r or not s)
```

6 Why CNF and DNF are helpful?

When we make programs, we constantly use Boolean expressions. For example with conditionals:

```
IF a certain condition happens:
    this code is executed
ELSE:
    this other code is executed
```

Or with loops:

```
WHILE a certain condition happens:
    this code is executed iteratively
```

In these scenarios, it is handy to simplify the Boolean expressions to understand them better and to be able to detect errors. As you can see from the example above, it is much easier to understand

$$(\neg p \lor q) \land r$$

than the original expression

$$\neg(p \land \neg q) \land (r \lor s) \land (r \lor \neg s)$$

In addition, the resulting code is much simpler

```
(not p or q) and r
```

than the original one

```
not (p and not q) and (r or s) and (r or not s)
```