Lecture 3

Equivalences and normal forms

Equational reasoning, Boolean algebras, normal forms

Dr Christoph Haase
University of Oxford
(with small changes by Javier Esparza)

Recap

- Syntax of formulas of propositional logic:
  1. *true* and *false* are formulas.
  2. Every propositional variable x_i is a formula.
  3. If F is a formula, then $\neg F$ is a formula.
  4. If F and G are formulas, then $(F \wedge G)$ and $(F \vee G)$ are formulas.
- Semantics of formulas via assignments, which are functions $\mathcal{A}\colon X \to \{0, 1\}$ that inductively extend to formulas
- Four notions: satisfiability, validity, entailment, equivalence.

Agenda

1. Equational reasoning
2. Boolean algebras
3. Normal forms

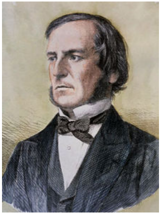
Decision problems

A decision problem is a computational problem whose output is either "yes" or "no".

- **Satisfiability**: Given a formula F, is F satisfiable?
- **Validity**: Given a formula F, is F valid?
- **Entailment**: Given formulas F and G, does $F \models G$ hold?
- **Equivalence**: Given formulas F and G, does $F \equiv G$ hold?

Equational reasoning

- Can show logical equivalence with brute force via truth tables
- **Equational reasoning** is more practical in many cases:
  - Establish basic equivalences
  - Derive new equivalences using the closure of logical equivalence under substitution



George Boole

Basic equivalences

The following is a list of basic equivalences:

$$F \wedge F \equiv F$$
$$F \vee F \equiv F$$
Idempotence

$$F \wedge G \equiv G \wedge F$$
$$F \vee G \equiv G \vee F$$
Commutativity

$$(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$$
$$(F \vee G) \vee H \equiv F \vee (G \vee H)$$
Associativity

$$F \wedge (F \vee G) \equiv F$$
$$F \vee (F \wedge G) \equiv F$$
Absorption

$$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$$
$$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$$
Distributivity

Basic equivalences

$$\neg\neg F \equiv F$$ **Double negation**

$$\neg(F \land G) \equiv (\neg F \lor \neg G)$$
$$\neg(F \lor G) \equiv (\neg F \land \neg G)$$ **De Morgan's laws**

$$F \lor \neg F \equiv \mathit{true}$$
$$F \land \neg F \equiv \mathit{false}$$ **Complementation**

$$F \lor \mathit{true} \equiv \mathit{true}$$
$$F \land \mathit{false} \equiv \mathit{false}$$ **Zero Laws**

$$F \lor \mathit{false} \equiv F$$
$$F \land \mathit{true} \equiv F$$ **Identity Laws**

Substitution

The essence of equational reasoning is the **substitution** of equals for equals.

Substitution

The essence of equational reasoning is the **substitution** of equals for equals.

- Informally, $G[F/H]$ means *"substitute F for H in G"*. E.g.:

$$(p_1 \wedge (p_2 \vee p_1))[\neg q_1/p_1] = \neg q_1 \wedge (p_2 \vee \neg q_1)$$

Substitution

The essence of equational reasoning is the **substitution** of equals for equals.

- Informally, $G[F/H]$ means *"substitute F for H in G"*. E.g.:

$$(p_1 \wedge (p_2 \vee p_1))[\neg q_1/p_1] = \neg q_1 \wedge (p_2 \vee \neg q_1)$$

- Formally, $G[F/H] := F$ if $G = H$. Whenever $G \neq H$, we proceed by induction:
  - Base cases:

$$x[F/H] := x \qquad \text{for all } x \in X$$

  - Induction steps:

$$\begin{aligned}(\neg G)[F/H] &:= \neg(G[F/H])\\ (G_1 \wedge G_2)[F/H] &:= G_1[F/H] \wedge G_2[F/H]\\ (G_1 \vee G_2)[F/H] &:= G_1[F/H] \vee G_2[F/H].\end{aligned}$$

Substitution Theorem

Theorem (Substitution Theorem)

If $F_1 \equiv F_2$ and $G_1 \equiv G_2$ then for every formula H

$$G_1[F_1/H] \equiv G_2[F_2/H].$$

Corollary

If $F_1 \equiv F_2$ then for every formula G

$$G \equiv G[F_2/F_1]$$

Proof.

Special case of the theorem with $G_1 := G$, $G_2 := G$, $H := F_1$. □

Substitution Theorem

We prove the Substitution Theorem for the special case $H = x$.

We need a semantic counterpart of substitution:

Given an assignment $\mathcal{A}$, a propositional variable x, and a truth value $b \in \{0, 1\}$, define the assignment $\mathcal{A}_{[x \mapsto b]}$ by

$$\mathcal{A}_{[x \mapsto b]}(y) := \begin{cases} b & \text{if } y = x \\ \mathcal{A}(y) & \text{if } y \neq x \end{cases}$$

for each propositional variable $y \in X$.

Substitution Theorem

Lemma (Translation Lemma)

Given formulas F, G and a propositional variable x, we have

$$\mathcal{A}(G[F/x]) = \mathcal{A}_{[x \mapsto \mathcal{A}(F)]}(G).$$

Substitution Theorem

Lemma (Translation Lemma)

Given formulas F, G and a propositional variable x, we have

$$\mathcal{A}(G[F/x]) = \mathcal{A}_{[x \mapsto \mathcal{A}(F)]}(G).$$

Proof.

By structural induction on G.
If $G = x$ then $\mathcal{A}(x[F/x]) = \mathcal{A}(F) = \mathcal{A}_{[x \mapsto \mathcal{A}(F)]}(x)$.
If $G = y$ for $y \neq x$ then $\mathcal{A}(y[F/x]) = \mathcal{A}(y) = \mathcal{A}_{[x \mapsto \mathcal{A}(F)]}(y)$.
If $G = G_1 \wedge G_2$ then

$$\begin{aligned}
\mathcal{A} \models (G_1 \wedge G_2)[F/x] \quad & \text{iff} \quad \mathcal{A} \models G_1[F/x] \wedge G_2[F/x] \\
& \text{iff} \quad \mathcal{A} \models G_1[F/x] \text{ and } \mathcal{A} \models G_2[F/x] \\
& \text{iff} \quad \mathcal{A}_{[x \mapsto \mathcal{A}(F)]} \models G_1 \text{ and } \mathcal{A}_{[x \mapsto \mathcal{A}(F)]} \models G_2 \quad \text{(i.h.)} \\
& \text{iff} \quad \mathcal{A}_{[x \mapsto \mathcal{A}(F)]} \models G_1 \wedge G_2
\end{aligned}$$

The induction cases for disjunction and negation are similar and are omitted. □

Substitution Theorem

Theorem (Substitution Theorem for $H = x$)

If $F_1 \equiv F_2$ and $G_1 \equiv G_2$ then

$$G_1[F_1/x] \equiv G_2[F_2/x].$$

Proof.

The proof is a direct application of Lemma 3.

$$\begin{aligned}
\mathcal{A}(G_1[F_1/x]) &= \mathcal{A}_{[x \mapsto \mathcal{A}(F_1)]}(G_1) && \text{by Lemma 3} \\
&= \mathcal{A}_{[x \mapsto \mathcal{A}(F_1)]}(G_2) && \text{since } G_1 \equiv G_2 \\
&= \mathcal{A}_{[x \mapsto \mathcal{A}(F_2)]}(G_2) && \text{since } F_1 \equiv F_2 \\
&= \mathcal{A}(G_2[F_2/x]) && \text{by Lemma 3}
\end{aligned}$$

□

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \lor (Q \lor R) \land (R \lor \neg P)) \equiv R \lor (\neg P \land Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \lor (Q \lor R)) \land (R \lor \neg P) &\equiv ((P \lor Q) \lor R) \land (R \lor \neg P) \\
&\equiv (R \lor (P \lor Q)) \land (R \lor \neg P) \\
&\equiv R \lor ((P \lor Q) \land \neg P) \\
&\equiv R \lor (\neg P \land (P \lor Q)) \\
&\equiv R \lor ((\neg P \land P) \lor (\neg P \land Q)) \\
&\equiv R \lor (\mathit{false} \lor (\neg P \land Q)) \\
&\equiv R \lor (\neg P \land Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv ({\color{red}R} \vee (P \vee Q)) \wedge ({\color{red}R} \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv ({\color{red}R} \vee (P \vee Q)) \wedge ({\color{red}R} \vee \neg P) \\
&\equiv {\color{red}R} \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \vee (Q \vee R) \wedge (R \vee \neg P)) \equiv R \vee (\neg P \wedge Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \vee (Q \vee R)) \wedge (R \vee \neg P) &\equiv ((P \vee Q) \vee R) \wedge (R \vee \neg P) \\
&\equiv (R \vee (P \vee Q)) \wedge (R \vee \neg P) \\
&\equiv R \vee ((P \vee Q) \wedge \neg P) \\
&\equiv R \vee (\neg P \wedge (P \vee Q)) \\
&\equiv R \vee ((\neg P \wedge P) \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\mathit{false} \vee (\neg P \wedge Q)) \\
&\equiv R \vee (\neg P \wedge Q).
\end{aligned}$$

Equational reasoning

The equivalence

$$(P \lor (Q \lor R) \land (R \lor \neg P)) \equiv R \lor (\neg P \land Q)\,.$$

has the following equational proof:

$$\begin{aligned}
(P \lor (Q \lor R)) \land (R \lor \neg P) &\equiv ((P \lor Q) \lor R) \land (R \lor \neg P) \\
&\equiv (R \lor (P \lor Q)) \land (R \lor \neg P) \\
&\equiv R \lor ((P \lor Q) \land \neg P) \\
&\equiv R \lor (\neg P \land (P \lor Q)) \\
&\equiv R \lor ((\neg P \land P) \lor (\neg P \land Q)) \\
&\equiv R \lor (\mathit{false} \lor (\neg P \land Q)) \\
&\equiv R \lor (\neg P \land Q).
\end{aligned}$$

Boolean algebras

A **Boolean algebra** is a set A together with two elements *true*, *false* $\in A$, one unary operation $\neg\colon A \to A$, and two binary operations $\wedge, \vee\colon A \times A \to A$ satisfying the Boolean algebra axioms.

Boolean algebras

A **Boolean algebra** is a set A together with two elements *true*, *false* $\in A$, one unary operation $\neg\colon A \to A$, and two binary operations $\wedge, \vee\colon A \times A \to A$ satisfying the Boolean algebra axioms.

Here are two other examples of Boolean algebras:

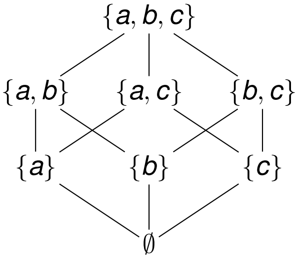
- $A = \{0, 1\}$, *true* $= 1$, *false* $= 0$, $\wedge = \min$, $\vee = \max$, $\neg x = 1 - x$.

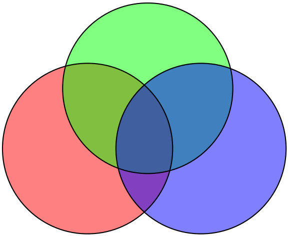
Boolean algebras

A **Boolean algebra** is a set A together with two elements *true*, *false* $\in A$, one unary operation $\neg\colon A \to A$, and two binary operations $\wedge, \vee\colon A \times A \to A$ satisfying the Boolean algebra axioms.

Here are two other examples of Boolean algebras:

- $A = \{0, 1\}$, $\mathit{true} = 1$, $\mathit{false} = 0$, $\wedge = \min$, $\vee = \max$, $\neg x = 1 - x$.
- For any set X, take $A = 2^X$ with $\mathit{true} = X$, $\mathit{false} = \emptyset$, $\wedge = \cap$, $\vee = \cup$, $\neg S = X \setminus S$.
  In fact, any finite Boolean algebra is of the form 2^X.





Boolean algebras and Boolean rings

- A **Boolean ring** is a ring A with 1 in which every element satisfies $a^2 = a$.

Boolean algebras and Boolean rings

- A **Boolean ring** is a ring A with 1 in which every element satisfies $a^2 = a$.
- Any Boolean ring A gives a Boolean algebra by

$$
\begin{aligned}
a \wedge b &:= ab \\
a \vee b &:= a + b + ab \\
\neg a &:= 1 + a \\
\mathit{false} &:= 0 \\
\mathit{true} &:= 1
\end{aligned}
$$

Boolean algebras and Boolean rings

- A **Boolean ring** is a ring A with 1 in which every element satisfies $a^2 = a$.

- Any Boolean ring A gives a Boolean algebra by

$$\begin{aligned} a \wedge b &:= ab \\ a \vee b &:= a + b + ab \\ \neg a &:= 1 + a \\ \mathit{false} &:= 0 \\ \mathit{true} &:= 1 \end{aligned}$$

- Any Boolean algebra A gives a Boolean ring by

$$\begin{aligned} ab &:= a \wedge b \\ a + b &:= (a \wedge \neg b) \vee (\neg a \wedge b) \end{aligned}$$

- So **Boolean algebras = Boolean rings**

Normal forms

- A **literal** is a propositional variable or the negation of a propositional variable:

$$x \text{ or } \neg x$$

- A formula F is in **conjunctive normal form** (**CNF**) if it is a conjunction of disjunctions of literals $L_{i,j}$:

$$F = \bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m_i} L_{i,j})$$

  Each conjunct is called a **clause** or (less often) a **maxterm.**

Normal forms

- A **literal** is a propositional variable or the negation of a propositional variable:

$$x \text{ or } \neg x$$

- A formula F is in **conjunctive normal form** (**CNF**) if it is a conjunction of disjunctions of literals $L_{i,j}$:

$$F = \bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m_i} L_{i,j})$$

  Each conjunct is called a **clause** or (less often) a **maxterm.**
- A formula F is in **disjunctive normal form** (**DNF**) if it is a disjunction of conjunctions of literals $L_{i,j}$:

$$F = \bigvee_{i=1}^{n} (\bigwedge_{j=1}^{m_i} L_{i,j})$$

  Each conjunct is called a **minterm**.
- Convention: *true* is CNF with no clauses, *false* is CNF with a single clause without literals

Theorem (Normalisation Theorem)

For every formula there is an equivalent formula in CNF and an equivalent formula in DNF.

Proof by truth table

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Each row with value 1 gives a clause in the DNF formula
- For each propositional variable x, the clause contains the literal x if 1 appears in column x, and $\neg x$ otherwise

Proof by truth table

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Each row with value 1 gives a clause in the DNF formula
- For each propositional variable x, the clause contains the literal x if 1 appears in column x, and $\neg x$ otherwise

Proof by truth table

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Each row with value 1 gives a clause in the DNF formula
- For each propositional variable x, the clause contains the literal x if 1 appears in column x, and $\neg x$ otherwise

- Each row with value 0 gives a clause in the CNF formula
- For each propositional variable x, the clause contains the literal x if 0 appears in column x, and $\neg x$ otherwise.

Proof by truth table

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Each row with value 1 gives a clause in the DNF formula
- For each propositional variable x, the clause contains the literal x if 1 appears in column x, and $\neg x$ otherwise

- Each row with value 0 gives a clause in the CNF formula
- For each propositional variable x, the clause contains the literal x if 0 appears in column x, and $\neg x$ otherwise.

Equational transformation to CNF

Is this efficient?

Equational transformation to CNF

Is this efficient? No! Better way:

1. Use **double negation** and **De Morgan's laws** to substitute in F every occurrence of a subformula of the form

$$\begin{array}{rcl} \neg\neg G & \text{by} & G \\ \neg(G \wedge H) & \text{by} & (\neg G \vee \neg H) \\ \neg(G \vee H) & \text{by} & (\neg G \wedge \neg H) \\ \neg \mathit{true} & \text{by} & \mathit{false} \\ \neg \mathit{false} & \text{by} & \mathit{true} \end{array}$$

until no such formulas occur (i.e., push all negations inward until negation is only applied to propositional variables), yielding the **negation normal form**

Equational transformation to CNF

2. Use **distributivity** to substitute in F every occurrence of a subformula of the form

$$
\begin{array}{rcl}
G \vee (H \wedge R) & \text{by} & (G \vee H) \wedge (G \vee R) \\
(H \wedge R) \vee G & \text{by} & (H \vee G) \wedge (H \vee R) \\
G \vee \mathit{true} & \text{by} & \mathit{true} \\
\mathit{true} \vee G & \text{by} & \mathit{true}
\end{array}
$$

until no such formulas occur (i.e., push all disjunctions inward until no conjunction occurs under a disjunction).

Equational transformation to CNF

2. Use **distributivity** to substitute in F every occurrence of a subformula of the form

$$\begin{array}{rcl} G \vee (H \wedge R) & \text{by} & (G \vee H) \wedge (G \vee R) \\ (H \wedge R) \vee G & \text{by} & (H \vee G) \wedge (H \vee R) \\ G \vee \mathit{true} & \text{by} & \mathit{true} \\ \mathit{true} \vee G & \text{by} & \mathit{true} \end{array}$$

   until no such formulas occur (i.e., push all disjunctions inward until no conjunction occurs under a disjunction).

3. Use the **identity** and **zero laws** to remove *false* from any clause and to delete all clauses containing *true*.

Tseytin's transformation

We give an algorithm, known as **Tseytin's transformation**, that

- takes as input a formula F of length n, and
- outputs a formula G in CNF of length $O(n)$ such that F is satisfiable iff G is satisfiable.

Observe that G cannot be the result of putting F in CNF, because in the worst case G can have length $\Theta(2^n)$.

Trade-off:

- G is only a small constant factor larger than F.
- F and G will **not** be equivalent in general.
  In fact, G will even contain auxiliary variables that do not apear in F.

Application: SAT-solving algorithms can assume that the input formula is in CNF.

There is **not** a similar algorithm for DNF.

Tseytin's transformation

- For every subformula G of F, let q_G be a fresh auxiliary variable. (Unless G is an atom, then we define $q_G := G$.)
- For every subformula G, define the formula C_G as follows:
  - If $G = x$ then $C_G := x$.
  - If $G = \neg H$, then let C_G be a formula in CNF equivalent to $(q_G \leftrightarrow \neg q_H)$.
  - if $G = (H \text{ op } H')$ for op $\in \{\wedge, \vee, \rightarrow, \leftrightarrow, \ldots\}$, then let C_G be a formula in CNF equivalent to $C_G \equiv (q_G \leftrightarrow (q_H \text{ op } q_{H'}))$.
- Define

$$E_F := q_F \wedge \bigwedge_{G \in sub(F)} C_G$$

Observe that E_F is in CNF.

Tseytin's transformation

Theorem

For every formula F: E_F is satisfiable iff F is satisfiable.

Proof Sketch.

For every assignment $\mathcal{A}$, let $\mathcal{A}'$ be an assignment satisfying $\mathcal{A}'(q_G) = \mathcal{A}(G)$ for every subformula G of F.

We have:

- $\mathcal{A}(F) = \mathcal{A}'(F) = \mathcal{A}'(q_F)$.
- $\mathcal{A}'(C_G) = 1$ for every subformula G.

It follows $\mathcal{A}'(E_F) = 1$ iff $\mathcal{A}(F) = 1$, and we are done. □

Tseytin's transformation: Example

- Let $F = (p_1 \wedge p_2) \vee (p_3 \wedge p_4)$.
- Introducing auxiliary variables yields:
  $q_0 \wedge (q_0 \leftrightarrow (q_1 \vee q_2)) \wedge (q_1 \leftrightarrow (p_1 \wedge p_2)) \wedge (q_2 \leftrightarrow (p_3 \wedge p_4))$
- For CNF we put each of the formulas $(q_0 \leftrightarrow (q_1 \vee q_2))$, $(q_1 \leftrightarrow (p_1 \wedge p_2))$ und $(q_2 \leftrightarrow (p_3 \wedge p_4))$ in CNF.
- Each formula has 3 variables, and so the CNF can always be chosen with at most 4 clauses.
- We obtain:

$$\begin{array}{rcl} E_F := q_0 & \wedge & (\neg q_0 \vee q_1 \vee q_2) \wedge (q_0 \vee \neg q_1) \wedge (q_0 \vee \neg q_2) \\ & \wedge & (q_1 \vee \neg p_1 \vee \neg p_2) \wedge (\neg q_1 \vee p_1) \wedge (\neg q_1 \vee p_2) \\ & \wedge & (q_2 \vee \neg p_3 \vee \neg p_4) \wedge (\neg q_2 \vee p_3) \wedge (\neg q_2 \vee p_4) \end{array}$$

Summary

- Equational reasoning is often more practical than using truth tables
- It allows reduction to normal forms
- CNF and DNF formulas are equally expressive as the class of all formulas

Summary

- Equational reasoning is often more practical than using truth tables
- It allows reduction to normal forms
- CNF and DNF formulas are equally expressive as the class of all formulas

Note:

- CNF can be exponentially shorter than DNF, see Sheet 1
- SAT is trivial for DNF formulas
- Later: SAT for CNF formulas = SAT for any formula