# Exercise sheet: Propositional Logic

## Exercise 1: CNF Conversion

(a) Prove that converting a formula into Negation Normal Form (NNF) terminates, by giving a weight function $w\colon \mathsf{formula} \to \mathbb{N}$ such that the following inequalities hold for all $F, G$:

$$\begin{aligned} w(\neg\neg F) &> w(F) \\ w(\neg(F \vee G)) &> w(\neg F \wedge \neg G) \\ w(\neg(F \wedge G)) &> w(\neg F \vee \neg G) \end{aligned}$$

$w(F)$ should be defined recursively using only addition, subtraction, multiplication, division and exponentiation.

*Note:* By "defined recursively" we mean that there are functions $f_\neg\colon \mathbb{N} \to \mathbb{N}$, $f_\vee\colon \mathbb{N}^2 \to \mathbb{N}$, $f_\wedge\colon \mathbb{N}^2 \to \mathbb{N}$ such that $w(\neg F) = f_\neg(w(F))$, $w(F \vee G) = f_\vee(w(F), w(G))$, and $w(F \wedge G) = f_\wedge(w(F), w(G))$.

(b) Prove that the second step of converting a Formula to CNF terminates, by giving a weight function $w\colon \mathsf{formula} \to \mathbb{N}$ such that the following inequalities hold for all $F, G, H$:

$$\begin{aligned} w(F \vee (G \wedge H)) &> w((F \vee G) \wedge (F \vee H)) \\ w((F \wedge G) \vee H) &> w((F \vee H) \wedge (G \vee H)) \end{aligned}$$

$w(F)$ should be defined recursively using only addition, subraction, multiplication, division and exponentiation.

(c) **Challenge:** find a weight function that fulfills the requirements of both (a) and (b).

## Solution

(a) We recursively define $w$:

$$\begin{aligned} w(p) &:= 2 \\ w(\neg F) &:= 2^{w(F)} \\ w(F \wedge G) &:= w(F) + w(G) \\ w(F \vee G) &:= w(F) + w(G) \end{aligned}$$

Then we have:

$$w(\neg\neg F) = 2^{2^{w(F)}} > w(F)$$

and since by induction $w(F) \geq 2$ and therefor $2^{w(F)} \geq 4$ for all $F$ we also have:

$$w(\neg(F \vee G)) = 2^{w(F)+w(G)} = 2^{w(F)} \cdot 2^{w(G)} > 2^{w(F)} + 2^{w(G)} = w(\neg F \wedge \neg G)$$

(and the same for the last inequality)

(b) We recursively define $w$:

$$\begin{array}{ll} w(p) & := 2 \\ w(\neg F) & := w(F) \\ w(F \wedge G) & := w(F) + w(G) \\ w(F \vee G) & := 2^{w(F)+w(G)} \end{array}$$

First note again that $w(F) \geq 2$ for all $F$. We then have

$$\begin{array}{ll} & w(F \vee (G \wedge H)) \\ = & 2^{w(F)+(w(G)+w(H))} \\ = & 2^{w(F)} \cdot 2^{w(G)} \cdot 2^{w(H)} \\ > & 2^{w(F)} \cdot \left(2^{w(G)} + 2^{w(H)}\right) \\ = & 2^{w(F)+w(G)} + 2^{w(F)+w(H)} \\ = & w((F \vee G) \wedge (F \vee H)) \end{array}$$

By symmetry the second inequality also holds.

(c) *(Solution due to Simon Kahler.)* We recursively define $w$:

$$\begin{array}{ll} w(p) & := 1 \\ w(\neg F) & := 2^{w(F)} \\ w(F \wedge G) & := 2 + w(F) + w(G) \\ w(F \vee G) & := 3 \cdot w(F) \cdot w(G) \end{array}$$

Note that $w(F) \geq 1$ for all $F$. For double negation elimination:

$$w(\neg\neg F) = 2^{2^{w(F)}} \geq 4 > w(F)$$

For De Morgan ($\neg(F \wedge G) \rightsquigarrow \neg F \vee \neg G$):

$$w(\neg(F \wedge G)) = 2^{2+w(F)+w(G)} = 4 \cdot 2^{w(F)} \cdot 2^{w(G)} > 3 \cdot 2^{w(F)} \cdot 2^{w(G)} = w(\neg F \vee \neg G)$$

For De Morgan ($\neg(F \vee G) \rightsquigarrow \neg F \wedge \neg G$):

$$w(\neg(F \vee G)) = 2^{3 \cdot w(F) \cdot w(G)} \geq 2^{w(F)+w(G)+1} = 2 \cdot 2^{w(F)} \cdot 2^{w(G)} > 2 + 2^{w(F)} + 2^{w(G)} = w(\neg F \wedge \neg G)$$

where the first inequality uses $3ab \geq a + b + 1$ for $a, b \geq 1$, and the last uses $2xy > 2 + x + y$ for $x, y \geq 2$.

For distributivity $(F \vee (G \wedge H) \rightsquigarrow (F \vee G) \wedge (F \vee H))$:

$$\begin{array}{ll} & w(F \vee (G \wedge H)) \\ = & 3 \cdot w(F) \cdot (2 + w(G) + w(H)) \\ = & 6 \cdot w(F) + 3 \cdot w(F) \cdot w(G) + 3 \cdot w(F) \cdot w(H) \\ > & 2 + 3 \cdot w(F) \cdot w(G) + 3 \cdot w(F) \cdot w(H) \\ = & w((F \vee G) \wedge (F \vee H)) \end{array}$$

since $6 \cdot w(F) \geq 6 > 2$. By symmetry the other distributivity inequality also holds.

### Exercise 2: Large disjunctive normal form

1. Write down a **DNF**-formula equivalent to $(a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \cdots \wedge (a_n \vee b_n)$.
2. Prove that any **DNF**-formula equivalent to the above formula must have at least $2^n$ clauses.

### Solution

1. An equivalent **DNF** formula is $\displaystyle\bigvee_{S \subseteq \{1,\ldots,n\}} \left( \bigwedge_{i \in S} a_i \wedge \bigwedge_{i \notin S} b_i \right)$.
2. Let $F$ be a **DNF** formula that is equivalent to the formula in part (1). We show that $F$ has at least $2^n$ clauses.

   We say that an assignment is *minimal* if it maps exactly one of $a_i$ and $b_i$ to 1 for $i = 1, \ldots, n$. There are $2^n$ minimal assignments and each one satisfies the formula in part (1). It follows that each minimal assignment must satisfy some clause of the DNF formula $F$. We claim that no two minimal assignments satisfy the same clause of $F$. From this it follows that $F$ has at least $2^n$ clauses.

   Let $\mathcal{A}$ and $\mathcal{B}$ be two distinct minimal assignments and let $i \in \{1, \ldots, n\}$ be such that $\mathcal{A}(a_i) \neq \mathcal{B}(a_i)$. Define a new assignment $\min(\mathcal{A}, \mathcal{B})$ pointwise by $\min(\mathcal{A}, \mathcal{B})(a) := \min(\mathcal{A}(a), \mathcal{B}(a))$ for each propositional variable $a$. Clearly $\min(\mathcal{A}, \mathcal{B}) \not\models a_i \vee b_i$ and hence $\min(\mathcal{A}, \mathcal{B}) \not\models F$. On the other hand, if $\mathcal{A}$ and $\mathcal{B}$ both satisfy the same clause $G$ of $F$ then, since $G$ is a conjunction of literals, we have $\min(\mathcal{A}, \mathcal{B}) \models G$ and hence $\min(\mathcal{A}, \mathcal{B}) \models F$, which is a contradiction.

### Exercise 3: Perfect matching

A **perfect matching** in an undirected graph $G = (V, E)$ is a subset of the edges $M \subseteq E$ such that every vertex $v \in V$ is an endpoint of exactly one edge in $M$. Given a finite graph $G$, describe how to obtain a propositional formula $F_G$ such that $F_G$ is satisfiable if and only if $G$ has a perfect matching. The formula $F_G$ should be computable from $G$ in time polynomial in $|V|$.

**Solution**

Introduce a propositional variable $x_e$ for each edge $e \in E$. For each vertex $v$, let $E(v)$ be the set of edges with $v$ as an endpoint. Then the formula is

$$F_G := \bigwedge_{v \in V} \left( \bigvee_{e \in E(v)} x_e \wedge \bigwedge_{\substack{e,e' \in E(v) \\ e \neq e'}} \neg x_e \vee \neg x_{e'} \right)$$