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 $w\colon \textsf{formula} \to \mathbb{N}$ such that the following inequalities hold for all F, G:

$$w(\neg\neg F) > w(F)$$
$$w(\neg(F \vee G)) > w(\neg F \wedge \neg G)$$
$$w(\neg(F \wedge G)) > w(\neg F \vee \neg G)$$

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

(b) Prove that the result of converting a formula into NNF is unique: Let $F \rightsquigarrow G$ denote that G is obtained from F by using double negation elimination (DNE), or one of De Morgan's laws. Prove that $\rightsquigarrow$ has the Church-Rosser property: if $F \rightsquigarrow G_1$ and $F \rightsquigarrow G_2$ then there exists a formula H such that $G_1 \rightsquigarrow^* H$ and $G_2 \rightsquigarrow^* H$.

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

$$w(F \vee (G \wedge H)) > w((F \wedge G) \vee (F \wedge H))$$
$$w((F \wedge G) \vee H) > w((F \wedge H) \vee (G \wedge H))$$

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

(d) **Challenge:** find a weight function that fulfills the requirements of both (a) and (c).
Note: We do not have a solution for this.

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) The only interesting case is the following:

$$\begin{array}{ccc} \neg\neg(F \vee G) & \rightsquigarrow & F \vee G \\ \downarrow\wr & & \downarrow\wr\, * \\ \neg(\neg F \wedge \neg G) & \overset{*}{\rightsquigarrow} & H \end{array}$$

We can choose $H = F \vee G$. Clearly $F \vee G \rightsquigarrow^0 F \vee G$. and $\neg(\neg F \wedge \neg G) \rightsquigarrow \neg\neg(F) \vee \neg\neg G \rightsquigarrow^2 F \vee G$

(c) 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.

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

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