An overview of Knowledge Compilation
Exercise Sheet 1, *December 04, 2017.*

Exercise 1. Efficient queries on DNNF.

1. Give a polynomial time algorithm that given DNNF D on variable X and an assignment τ of the variables $Y \subseteq X$, decide if one can extend τ to a satisfying assignment of D.

2. Let X be a set of variables. For each variable $x \in X$, we are given two weights $w_{x,1} \in \mathbb{Q}$ and $w_{x,0} \in \mathbb{Q}$. The weight of an assignment $\tau : X \to \{0,1\}$ is defined as $w(\tau) = \prod_{x \in X} w_{x,\tau(x)}$. Give a polynomial time algorithm that given a DNNF D on variables X outputs a satisfying assignment of D of maximal weight.

3. ($\star$) Given an algorithm that outputs all satisfying assignments of a given DNNF D. Each satisfying assignment should be output exactly once and the time spent between two consecutive outputs should be polynomial in the size of D.

   Hint: *Use the result of the first question as a black box.*

Exercise 2. Hard queries on DNNF.

1. Show that if you can count the number of satisfying assignments of a DNNF D in time polynomial in the size of D, then you can solve #SAT in polynomial time.

2. Let X be a set of variables. For each variable $x \in X$, we are given a probability $p_x \in [0,1]$. Given a DNNF D on X and $(p_x)_{x \in X}$, the probability of satisfying D is the probability of D being satisfied if one picks an assignment τ as follows: $\tau(x) = 1$ with probability p_x and $\tau(x) = 0$ otherwise. Show that if you can compute the probability of D being satisfied in polynomial time, then you can solve #SAT in polynomial time.

3. ($\star$) Let X be a set of variables. For every pair of variable $x, y \in X$, we are given a function $f_{x,y} : \{0,1\}^2 \to \mathbb{Q}$. The weight of an assignment τ is defined by $\prod_{x,y \in X} f_{x,y}(\tau(x), \tau(y))$. Show that the problem of finding a maximal weight satisfying assignment of a given a DNNF D with functions $f_{x,y}$ is NP-hard.

Exercise 3. Some relations between languages.

1. Show that you can transform any FBDD F into a decision DNNF D of size linear in the size of F.

2. Show that if every DNNF can be simulated by a polysize deterministic DNNF then $\mathsf{NP} \subseteq \mathsf{P/poly}$.

3. ($\star\star$) Show that any decision DNNF D of size N can be computed by an FBDD of size $N^{\log(N)+1}$.