Problem Set 3

Jaan Tollander de Balsch

March 21, 2020

H3.1

Define a circuit scheme for computing the function $f(x,y)$, where x and y are two n-bit binary numbers, and $f(x,y)=1$ if and only if the numerical value of x is strictly greater than the numberical values of y.

First, we define the baseline circuit for x and y which are two 1-bit binary numbers.

The truth table takes form

x	y	g
0	0	0
0	1	0
1	0	1
1	1	0

Hence, the baseline circuit is

$$g(x,y) = x \wedge \neg y.$$

Given the n-bit inputs $x=(x_1,...,x_n)$ and $y=(y_1,...,y_n)$ where x_i and y_i for $i=1,...,n$ are the individual bits, the circuit scheme for computing f is defined as follows. We assume n of size $2^m, m \in \mathbb{N}$ in this analysis for simplicity.

First step

$$v := g(x_1,y_1),...,g(x_n,y_n)$$

Recursive step: Repeat until v has length 1, which will be the output of the circuit.

$$v := g(v_1,v_2), g(v_3,v_4),...,g(v_{n-1},v_n)$$

In each recursive step the size of v is halved, thus, the circuit has depth $O(\log n)$.

H3.2

We refer to Vadhan and Ullman (2010).

(i)

We have sizes $|\{0,1\}| = 2$ and $|\{0,1\}^n| = 2^n$.

The number of boolean function on n variables

$$f : \{0,1\}^n \to \{0,1\}$$

is 2^{2^n}, that is, there are 2^n inputs for each output.

(ii)

We can express a truth table in DNF form as

$$f(x) = \bigvee_{\alpha : f(\alpha)=1} \bigwedge_{i=1}^{n} (x_i = \alpha_i).$$

With n variables the formula has size of $O(n2^n)$, which means that every boolean function f can be computed by a circuit of size $O(n2^n)$, that is, number of gates.

(iii)

Since some function require circuits of size $m = O(n2^n)$ to compute, not all function cannot be computed by circuits with $m = 2^n/2n$ (or fewer) gates.

H3.3

We have boolean formula ϕ in conjunctive normal form where each clause has exactly two literals. We will prove that we can decide satisfiability of ϕ in polynomial time.

2-satisfiability

Let the variables in formula ϕ be $X = \{x_1, ..., x_n\}$. We will construct a directed graph $G_\phi = (V, E)$ as follows.

- The set of vertices V are all the $2n$ literals over X, that is, $\{x_1, \neg x_1, ..., x_n, \neg x_n\}$.
- The set of edges E consists of directed edges $\neg\alpha \rightarrow \beta$ and $\neg\beta \rightarrow \alpha$ for every clause $(\alpha \vee \beta)$ in ϕ.

The graph G_ϕ is called an *implication graph*. Deciding satisfiability for ϕ reduces to finding *strongly connected components* in the implication graph G_ϕ.

(a)

In the implication graph G_ϕ, if a variable and its negation belong to the same strongly connected component, the literals must have same values, therefore, the instance cannot be satisfied.

(b)

Runtime of finding strongly connected components is $O(|V| + |E|)$. We have

- $|V| = 2n$, and
- $|E| \leq \binom{2n}{2} \cdot 2 = O(n^2)$ – Maximum number of clauses in formula ϕ times two vertices per clause.

Therefore, runtime is polynomial in respect to the number of variables n.

References

Vadhan, S. and Ullman, J., 2010. Circuit Size Bounds Circuit Depth Bounds Boolean Formulas. [online] pp.1–4. Available at: <https://people.mpi-inf.mpg.de/%7B~%7Dnsaurabh/bfc2019/lecture2.pdf>.