CS 278 - Homework 3

Joshua Turcotti

April 18, 2021

1 SAT Algorithms for ACC^0

(a) For any set $S \subseteq [n]$, let a_S be corresponding coefficient in P. For any set $S \subseteq [n]$ let $a_S^{(0)}$ and $a_S^{(1)}$ be the corresponding coefficients in P_0 and P_1, respectively. Now:

$$a_S^{(0)} = a_S$$
$$a_S^{(1)} = a_{S \cup \{n\}}$$

(b) For all $x \in \{0,1\}^n$ with $x_n = 0$, $P(x) = P_0(x)$, so no computation is needed, just a copy, time $O(1)$. For all $x \in \{0,1\}^n$ with $x_n = 1$, $P(x) = P_0(x) + P_1(x)$. We assume to have been given all of the latter such values, so only an $O(1)$ addition must be performed for each of the 2^{n-1} such x with $x_n = 1$. Thus for all $x \in \{0,1\}^n$, an $O(1)$ operation suffices to compute $P(x)$ given all values of P_0 and P_1, yielding $O(2^n)$ overall time to compute all values of P over $x \in \{0,1\}^n$.

(c) Since each round is accumulated time $O(2^n)$ across the branches, and the entire computation is n rounds deep, the algorithm takes time $O(2^n n)$ (and can be done in-place, i.e. $O(1)$ space complexity).

(d) Once we have a list of all 2^n values that P takes on $x \in \{0,1\}^n$, we simply check Ψ on each of those values, and iff we encounter some $1 \leq w_0 \leq w$ that occurs as in the range of P such that $\Psi(w_0) = 1$, we conclude that C' is satisfiable.

2 SAT Algorithms - Faster than 2^n

(a) We can phrase satisfiability for C as:

$$\exists x \in \{0,1\}^n : C(x) = 1$$

Equivalently:

$$\exists x \in \{0,1\}^{n-t} : \exists z \in \{0,1\}^t : C(z,x) = 1$$

Equivalently:

$$\exists x \in \{0,1\}^{n-t} : \bigvee_{z \in \{0,1\}^t} C(z,x) = 1$$

Equivalently:

$$\exists x \in \{0,1\}^{n-t} : C^*(x) = 1$$

But this last predicate is exactly satisfiability for C^*, so we are done.

(b) The circuit size of C^* is $O(2^t s)$; bounding by one copy of C per $z \in \{0,1\}^t$. It depends on $n-t$ input bits, and has depth $d+1 = O(1)$.

(c) To apply the SAT algorithm from part 1, we need to show that the circuit size, $s' = O(2^t s)$ is $poly(n-t)$. Since $t = O(\log n)$, $poly(n-t) = poly(n)$. Now $s = poly(n)$, which is given, and $2^t = poly(n)$, which follows from the definition of $t = 101\log(n)$, are sufficent to conclude that s' is $poly(n-t)$, so we may apply the SAT algorithm and conclude that satisfiability for C^* can be determined in time $O(2^{n-t}(n-t)) = O(\frac{2^n}{n^{101}}(n-t)) = O(\frac{2^n}{n^{100}})$.

3 Paralllel Repetition of AM

We will show the lifting of perfect completeness and strengthened error for soundness to the parallel version of AM. For completeness, assume $x \in L$, and note that if Merlin uses k independent copies of Π to respond to the k messages r_i received, then Arthur will always compute $A(x, y_i, r_i) = 1$ for all i by the perfect completeness of Π. For exponentially strengthened soundness, let $x \notin L$, and assume that Merlin manages to respond with a scheme that is independent between messages (we assume cheating-free), yet Arthur accepts with probability strictly greater than $(1/3)^k$. By independence, the probability over r that Arthurt accepts is equal to the product of the probabilities over r that $A(x, y_i, r_i) = 1$ for each each i. But now for some i, this probability must be greater than $1/3$, which contradicts the assumptions of Π. We can conclude that for any scheme under which Merlin answers, the probability that Arthurt accepts is at most $(1/3)^k$.