# Seminar on Sublinear Time Algorithms – Handout 4

Robert Krauthgamer

April 14, 2010 (Revised May 2, 2010)

## 1 Today's topics

- Testing homomorphism of a function (deferred from last week)
- Testing a Dense Graph for Bipartiteness

## 2 Open problems

It is interesting to compare different "types" of testers, e.g. non-adaptive vs. adaptive, one-sided vs. two-sided error, query complexity vs. running time, and polynomial vs. larger dependence on $\varepsilon$. Several results about these are known.

Here is a specific open problem directly related to today's material: The adaptive query complexity of bipartiteness in the dense model is not completely known: the lower bound is $\Omega(1/\varepsilon^{3/2})$, and an upper bound $\tilde{O}(1/\varepsilon^2)$ follows from non-adaptive case.

## 3 Homework

1. Let $s \in \Sigma^n$ be a fixed string. We say that a string $x \in \Sigma^n$ is <u>shift-equivalent</u> to $s$ if it can be derived from $s$ using cyclic shift operations. Give a tester for shift-equivalent to $s$, i.e. a fast algorithm that determines ~~for testing~~ whether an input string $x$ is shift-equivalent to $s$ or $\varepsilon$-far from it.

   (a) Analyze the number of queries the algorithm makes into $x$, i.e. the algorithm can access $s$ (as well as other operations) "for free".

   (b) Analyze the tester running time, by designing a preprocessing stage that depends only on $s$. (Hint: Let the preprocessing build a polynomial-sized table, which the tester will then use). Clarification: The preprocessing may be randomized; moreover, the success of the query procedure might depend on the coin tosses made by the preprocessing stage. In principle, the query procedure could even be determinstic.

   Hint: The complexity should be roughly logarithmic in $n$ (and some dependence on $\varepsilon$).

   Remark: A straightforward adaptation to having several such strings $s$, is essentially testing for membership in a cyclic code.