Word Equations and SLPs: Exercise Sheet 0

Task 1 Show that the problem of satisfiability of a system of word equations can be reduced to the problem of satisfiability of a single word equation, when we are allowed to add letters to the alphabet. Show the same result also when adding letters is not allowed, but $|\Sigma| \geq 2$.
Hint: Do this first for two equations. What goes wrong when we simply concatenate the left-hand sides and right-hand sides? We may need to use each side more than once.

Task 2 (Long in statement, but not difficult) Consider a mapping from $\Sigma = \{a, b\}$ to 2×2 matrices over $\mathbb{N}$, defined as

$$\varphi(a) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } \varphi(b) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} .$$

Extend this to Σ^* as a homomorphism.

Show that for any $w \in \Sigma^*$ its image is a a matrix with a determinant one.

Show that this mapping is injective; to do this, consider, what are the rows of a matrix $\varphi(a) \cdot \begin{bmatrix} n & n' \\ m & m' \end{bmatrix}$ and what are the rows of $\varphi(b) \cdot \begin{bmatrix} n & n' \\ m & m' \end{bmatrix}$. Deduce from this that looking at the matrix $M_w = \varphi(w)$ we can determine the left-most letter of w by looking at rows of M_w.

Show that if a 2×2 matrix M with determinant 1 and all natural entries can be represented as either $\varphi(a) \cdot M'$ or $\varphi(b) \cdot M'$, where M' has a determinant 1 and all natural entries. Again: compare the rows.

Deduce from this that φ is an isomorphism between Σ^* and 2×2 matrices with determinant 1 and all entries natural.

Deduce from this that satisfiability of word equation over $\Sigma = \{a, b\}$ reduces to the satisfiability of equations over natural numbers; to do this, represent a $\varphi(X)$ as a matrix of variables representing natural numbers.

Word Equations and SLPs: Exercise Sheet 1

Task 3 Show that the composition system of size n can be turned into an SLP of polynomial size. How small you can make the degree of the polynomial?

Task 4 Suppose that S is a length-minimal solution of an equation $U = V$ over an alphabet Σ and UV contains at least one letter from Σ.

- Suppose that ab, where $a \neq b$ and $a, b \in \Sigma$, occurs in $S(U)$. Show that there is an occurrence of ab in $S(U)$ or in $S(V)$ that has a cut between those letters.
- Suppose that a^k is a maximal block in $S(U)$. Show that there is an occurrence of a maximal block a^k such that it touches a cut, i.e. the cut is between the letters of the block or right next to it.

Task 5 Show that we can uncross and compress all blocks of all letters in parallel, i.e. as one procedure that pops at most one prefix and one suffix per occurrence of variable.

Task 6 Show that we can uncross and compress a set of pairs $\{a_i b_i\}_{i \in I}$ in parallel, assuming that $a_i \in \Sigma_1$ and $b_i \in \Sigma_2$ for each $i \in I$, where $\Sigma_1 \cap \Sigma_2 = \emptyset$.

Task 7 A *partition* of an alphabet Σ is a pair (Σ_1, Σ_2) such that $\Sigma_1 \cup \Sigma_2 = \Sigma$ and $\Sigma_1 \cap \Sigma_2 = \emptyset$.

Consider a word $w \in \Sigma^*$ such that none of its two consecutive letters are the same. An occurrence of a pair ab in w is *covered* by a partition (Σ_1, Σ_2) if $a \in \Sigma_1$ and $b \in \Sigma_2$. Show that there is a partition of Σ such that it covers at least $\frac{|w|-1}{2}$ letters in w. Show that it can be computed in linear time.

Generalise this observation to a word equation .

Hint: Reduce this problem to calculation of a maximal (weighted) cut in a graph. It has a simple randomised solution which can be derandomised using expected value approach. It is described in Michael Mitzenmacher, Eli Upfal *Probability and computing* book as well as in Vijay Vazirani *Approximation algorithm* book.

Task 8 Using tasks 5–7 devise an algorithm for word equation that keeps a linear-size equation; the algorithm can use more memory when processing the equations, moreover, at some point it will have to store blocks a^{cn}, but we treat them as size-1. (The latter is a cheat, but we will learn how to deal with this later on).

Task 9 In this task we consider word equations with regular constraints. Formally, we are given a function $\rho : \Sigma \to \mathbb{M}_q$, where $\mathbb{M}_q$ are the set of Boolean matrices of size $q \times q$. We extend it to words as a homomorphism, i.e. by $\rho(a_1 \cdots a_k) = \rho(a_1) \cdots \rho(a_k)$, where the multiplication on the right is the Boolean matrix multiplication. (If you are not familiar with this setting: think that this is a transition matrix of a finite automaton, with 1 at position i, j in $\rho(a)$ meaning that there is a transition from i to j labelled with a).

A constraints for X is given as $\rho_X \in \mathbb{M}_q$ and the solution should satisfy $\rho(S(X)) = \rho(X)$.

Show that given a word equation over the alphabet Σ with regular constraints given by ρ we consider this equation over the alphabet

$$\Sigma \cup \{a_\tau \,:\, \tau \in N \text{ and there is a word } w \in \Sigma^* \text{ such that } \rho(w) = \tau\} \ ,$$

where $\rho(a_P) = P$. Show the equisatisfiability of the satisfiability problem over the original alphabet and over such an extended alphabet.