In this sequence of exercises you are going to show an alternative proof for the alphabet reduction lemma:

**Lemma 1** (Alphabet reduction). Recall that in a CSP problem $p$, the size (i.e., number of clauses) of $p$ is denoted by $|p|$, the number of queries (i.e., the size of each clause) by $q = q(p)$, the alphabet size is denoted by $\sigma = \sigma(p)$, and the maximum fraction of satisfied clauses by $\mu = \mu(p)$.

There exists a polynomial-time function $\text{alph-red}$ and absolute constant $q_0$ such that for every $2$-query CSP $p$ we have:

- **Linear blowup** $\text{alph-red}(p)$ is a $q_0$-query CSP with alphabet $\{0, 1\}$, and size less than $C|p|$ for some $C = C(\sigma(p))$.

- **Completeness** If $\mu(p) = 1$ then $\mu(\text{alph-red}(p)) = 1$.

- **Limited loss** There’s an absolute constant $D$ (not depending on $p$ or $\sigma$) such that if $\mu(p) \leq 1 - \epsilon$ then $\mu(\text{alph-red}(p)) \leq 1 - \epsilon/D$.

**Exercise 1** (22 points). For a set $S$ define the long-code of $S$ to be the following function $\mathcal{LC} : S \rightarrow \{0, 1\}^{2|S|}$: for every $s \in S$ and a function $f : S \rightarrow \{0, 1\}$ (note that we think of $f$ also as a string of length $|S|$ and a number in $[2^{|S|}]$), the $f^{th}$ position of $\mathcal{LC}(s)$ (denoted by $\mathcal{LC}(s)_f$) is $f(s)$.

1. For every $s \in S$, one can think of the output of the long-code on $s$ as itself a function from $\{0, 1\}^{|S|}$ to $\{0, 1\}$. That is, we think of $\mathcal{LC}(s)$ as the function that maps $f : \{0, 1\}^{|S|} \rightarrow \{0, 1\}$ to $\{0, 1\}$ in the following way $\mathcal{LC}(s)(f) = f(s)$. Prove that for every $s$, $\mathcal{LC}(s)$ is a linear function.

2. Prove that for any $s$, the fraction of $f$’s such that $f(s) = 1$ is half. (Hint, this is equivalent to proving that $\Pr_f[f(s) = 1] = 1/2$ for a random function $f : S \rightarrow \{0, 1\}$).

3. Prove that $\mathcal{LC}$ is an error-correcting code with distance half. That is, for every $s \neq s' \in S$, the hamming distance of $\mathcal{LC}(s)$ and $\mathcal{LC}(s')$ is half.

4. Prove that for any $s \in S$, $\mathcal{LC}(s)$ is equal to $\mathcal{H}(e^s)$ where $\mathcal{H}$ is the Hadamard code from $\{0, 1\}^{|S|}$ to $\{0, 1\}^{2|S|}$ (i.e., $\mathcal{H}(x)_y = \langle x, y \rangle \pmod{2}$) and $e^s \in \{0, 1\}^S$ is the standard basis vector corresponding to $s$. That is, the $i^{th}$ position of $e^s$ is $0$ for $i \neq s$ and $1$ for $i = s$. 

Exercise 2 (22 points). Prove that \( \mathcal{LC} \) is self-correctible. That is, show an algorithm \( A \) and constants \( C, D \) such that given oracle access to a string \( L \) that is within fractional distance \( \epsilon \) to \( \mathcal{LC}(s) \), and a function \( f : S \rightarrow \{0, 1\} \), \( A^L(f) \) should output \( \mathcal{LC}(s)_f \) with probability \( 1 - C\epsilon \) while making at most \( D \) queries to \( L \). Note that \( A^L(f) \) should output \( \mathcal{LC}(s)_f \) with high probability even if \( L(f) \neq \mathcal{LC}(s)_f \).

Note that here (in the rest of the exercises) we don’t care about the running time of the algorithm but only that it makes at most a constant number of queries to its oracle.

Exercise 3. In this exercise you’ll prove in stages that \( \mathcal{LC} \) is locally testable.

1. Given an oracle to a function \( L : \{0, 1\}^{|S|} \rightarrow \{0, 1\} \), consider the following test: choose \( f \) at random from \( \{0, 1\}^{|S|} \) and if \( L(f) = 1 \) accept. Otherwise, (if \( L(f) = 0 \)), choose \( g \) to be a random subset of \( f \). That is, for every \( s \) such that \( f(s) = 0 \) choose \( g(s) = 0 \) and for every \( s \) with \( f(s) = 1 \) choose \( g(s) = 1 \) with probability \( 1/2 \) (otherwise choose \( g(s) = 0 \). Accept iff \( L(g) = 0 \). Prove that if \( L \) is a longcode codeword (i.e., \( L = \mathcal{LC}(s) \) for some \( s \)) then it passes this test with probability 1.

2. Prove that if \( L \) is a long-code codeword, then for every \( f : \{0, 1\}^{|S|} \), \( L(f) \neq L(\overline{f}) \) where \( \overline{f} \) is the negation of \( f \) (i.e., \( \overline{f}(s) = 1 - f(s) \) for every \( s \in S \)).

3. Let \( L : \{0, 1\}^{|S|} \rightarrow \{0, 1\} \) be a non-zero linear function. That is, there exists some non-zero string \( \ell \in \{0, 1\}^{|S|} \) such that for every \( f \in \{0, 1\}^{|S|} \), \( L(f) = (\ell, f) \mod 2 \). We say that \( L \) is a longcode codeword if \( L = \mathcal{LC}(s) \) for some \( s \in S \), or equivalently, \( \ell = e^s \) for some \( s \). Prove that if \( L \) is not a longcode code word then it will fail the test from 1 with probability at least 1/100.

4. Prove that \( \mathcal{LC} \) is locally testable. That is, show that there exist constants \( C, D \) and an algorithm \( T \) such that for any \( \epsilon \geq 0 \) given oracle access to an oracle \( L \) that of distance at least \( \epsilon \) from \( \mathcal{LC}(s) \) for every \( s \), \( T^L \) will reject with probability at least \( \epsilon/C \) and will make at most \( D \) queries. The test should be complete in the sense that \( T^L \) should accept with probability one for every \( L \) that is a longcode codeword. You can use without proof the result stated in class on linearity testing.

5. Show that this implies that there is such an algorithm with \( C = 1/100 \).

Exercise 4 (22 points). Let \( c : S \times S \rightarrow \{0, 1\} \) be some function. Show an algorithm \( T \) that given oracle access to \( L_1, L_2, L_3 \) where \( L_1, L_2 \) are functions from \( \{0, 1\}^{|S|} \rightarrow \{0, 1\} \) and \( L_3 \) is a function from \( \{0, 1\}^{|S|^2} \rightarrow \{0, 1\} \) makes at most a constant number of queries to its oracles and satisfies the following properties:

1. If \( L_1 = \mathcal{LC}(s) \), \( L_2 = \mathcal{LC}(s') \), and \( L_3 = \mathcal{LC}(s \circ s') \) for \( s, s' \) that satisfy \( c(s, s') = 1 \) then \( T \) will accept with probability 1.

2. If \( L_1 = \mathcal{LC}(s) \), \( L_2 = \mathcal{LC}(s') \) and \( L_3 = \mathcal{LC}(s'') \) with \( s'' \neq s \circ s' \) then \( T \) will reject with probability at least 0.99.

3. If \( L_1 = \mathcal{LC}(s) \), \( L_2 = \mathcal{LC}(s') \), and \( L_3 = \mathcal{LC}(s \circ s') \) for \( s, s' \) that satisfy \( c(s, s') = 0 \) then \( T \) will reject with probability at least 0.99.

Exercise 5 (22 points). Prove Lemma 1 using the above exercises. See footnote for hint\(^1\)

\(^1\)Hint: if we let \( S \) denote the alphabet of the original problem \( P \) then in the new problems we’ll have \( n_2^{|S|} \) new Boolean variables that are supposed to be longcode encodings of each variable in the original formula and \( n_2^{|S|^2} \) new Boolean variables that for every 2-query constraint \( c(x_i, x_j) \) are supposed to be longcode encoding of \( x_i \circ x_j \).