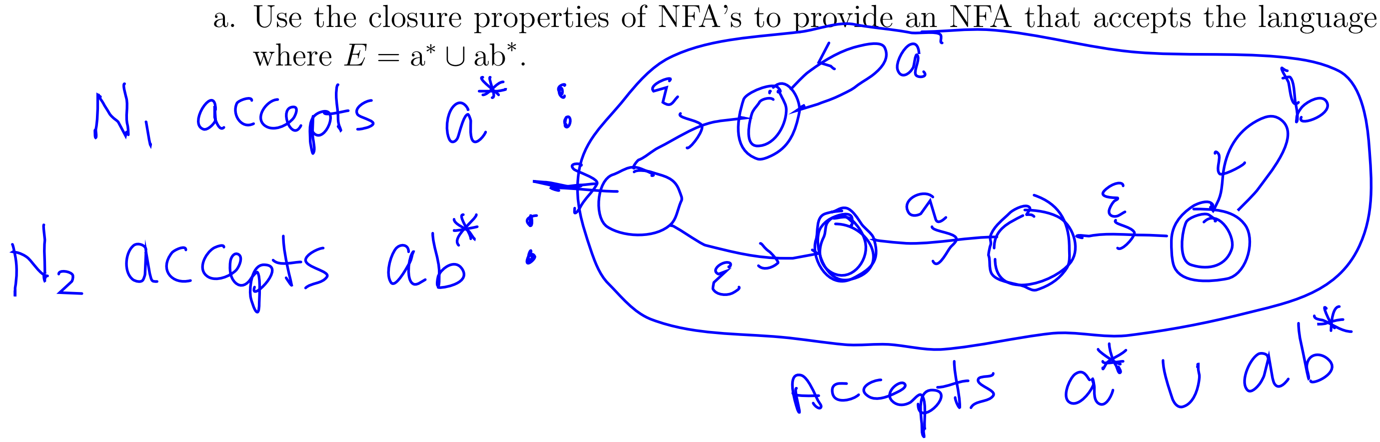
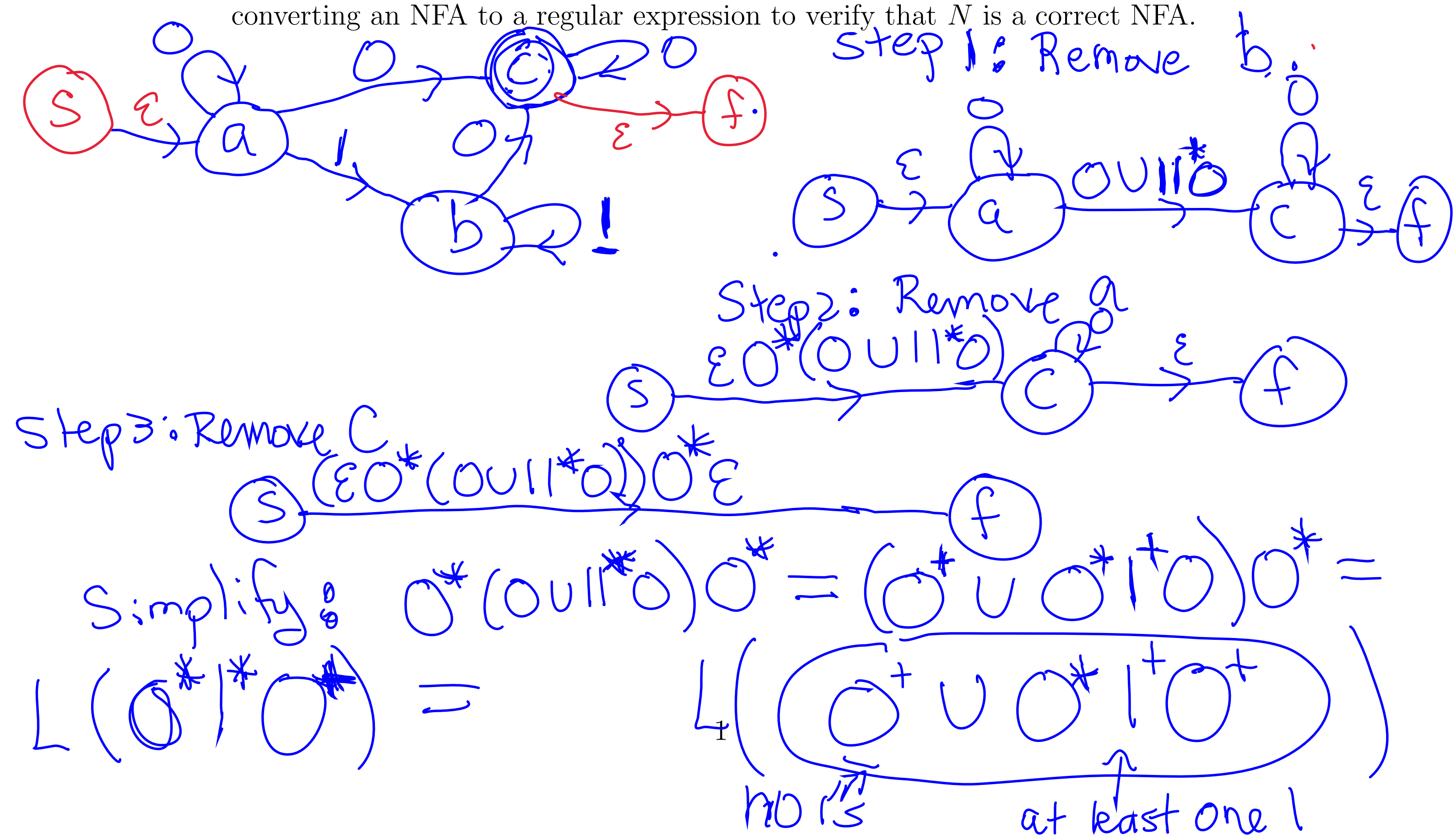
CECS 329, Review Problems for Exam 2, Spring 2023, Dr. Ebert

1. Solve each of the following.

   a. Use the closure properties of NFA's to provide an NFA that accepts the language $L(E)$, where $E = \mathrm{a}^* \cup \mathrm{ab}^*$.



   b. Provide a three-state NFA N that accepts the language $0^*1^*0^+$ and use the procedure for converting an NFA to a regular expression to verify that N is a correct NFA.



2. Do the following.

   a. Provide the instructions for a URM that computes the function $f(x, y) = |x - y|$.

   b. For each register used in your program, use one or more sentences to describe its purpose.

3. Show that the greatest common divisor function $\gcd(x, y)$ is primitive recursive. You may use any PR functions from the General Models of Computation lecture examples (but not exercises).

4. Solve the following.

   a. Let LPP(x, i) denote the function that returns the largest prime power of p_i that divides x. Prove that LPP(x, i) is primitive recursive. Show that the decoding functions $\pi_1(w)$, $\pi_2(w)$, $\xi_1(w)$, $\xi_2(w)$, and $\xi_3(w)$ are all primitive recursive. For example, if $\xi(x, y, z) = w$, then $\xi_1(w) = x$, $\xi_2(w) = y$, and $\xi_3(w) = z$.

   b. Use the functions from part a to show that MaxRegIns(x) is primitive recursive, where MaxRegIns(x) denotes the maximum register used by the instruction whose β-encoding equals x.

The following is an outline of a diagnonalization argument that proves that the following function is not URM-computable:

$$g(x) = \begin{cases} 1 & \text{if } P_x \text{ is total} \\ 0 & \text{otherwise} \end{cases}$$

In other words, $g(x) = 1$ iff the program P_x having Gödel number x halts on every possible input y, and $g(x)$ outputs 0 otherwise. We assume that g is a total URM-computable function and show a contradiction. Note: correctly solving this problem counts for passing LO8.

a. Define function $f(x)$ as

Abstract URM Program

$$f(x) = \begin{cases} \phi_x(x) + 1 & \text{if } g(x) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Suppose URM program G computes $g(x)$, has 500 instructions, uses registers R_1 through R_{75}. Suppose also that universal URM program U computes universal function $\Psi_U(x, y)$, has 1000 instructions and uses registers R_1 through R_{200}. Use these programs along with other instructions to design a URM program F that correctly computes $f(x)$. Explain why this works.

1. T(1, 201) // Copy x to a safe register
2. G[1:500]
3. J(1, 202, 2000) // Jump out of program if g(x)=0
4. Z(1), Z(2), ..., Z(75)
5. T(201, 1)
6. T(201, 2)
7. U[1:1000]
8. S(1) // $\phi_x(x) + 1$

b. Let e denote the Gödel number of program F from part a. In other words $F = P_e$. Show that a contradiction arises when we try to compute $F(e) = P_e(e) = \phi_e(e)$.

$F(e) = f(e) = \phi_e(e) + 1$ (Since F is total) $= F(e) + 1$ Contradiction