CECS 329, Learning Outcome Assessment 11, May 5th, Spring 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit solutions to AT MOST 3 LO problems on separate sheets of paper.

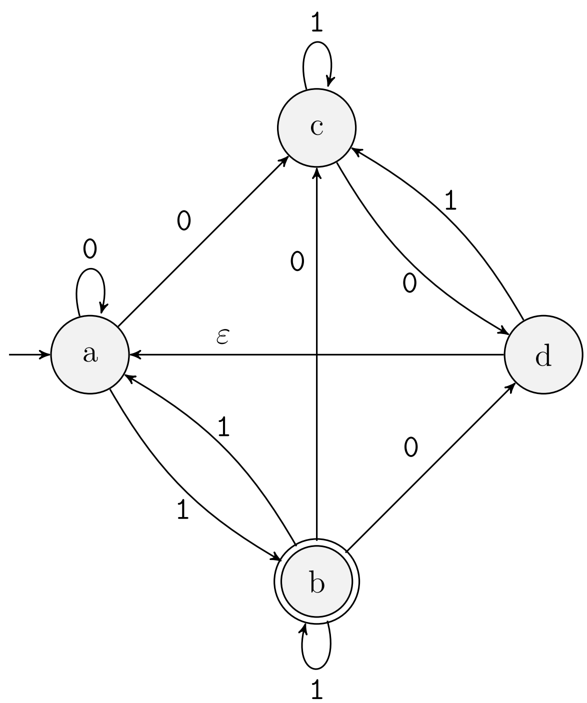
Problems

LO1. Do the following.

(a) Provide the state diagram for a DFA that accepts only those binary words that do *not* contain the subword 011.

(b) Show the computation of M on input i) $w = 10100$ and ii) $w = 00110$.

LO2. Do the following for the NFA N whose state diagram is shown below.



(a) Provide a table that represents N's δ transition function.

(b) Use the table from part a to convert N to an equivalent DFA M using the method of subset states. Draw M's state diagram.

(c) Show the computation of M on input $w = 11001$.

LO3. Provide a regular expression that represents the set of binary words w that do *not* contain the subword 011.

LO4. Do the following.

(a) Provide a context free grammar $G = (V, \Sigma, R, S)$ for which $L(G)$ is the set of words from $\{a,b\}^*$ that are palindromes of even length (i.e. words that read the same forwards as backwards). For example, abba is an even-length palindrome, but abab is not.

(b) Use G to provide a leftmost derivation of baab.

LO5. Provide a *recursive* definition for the function $f(x) = x^2 + x - 1$.

LO6. Do the following.

(a) Compute the Gödel number for program $P = J(1,2,3), S(4), Z(5), T(4,1)$. Write your answer as a sum of powers of two minus 1 (see part b).

(b) Provide the URM program P whose Gödel number equals

$$2^{15} + 2^{46} + 2^{87} + 2^{109} - 1.$$

LO7. Answer and solve the following.

(a) A universal program P_U must be able to simulate any program, regardless of how many registers that program may use. It is able to accomplish this due to the fact that (5 pts)

i. a universal program possesses a copy of each URM program's instructions and is able to look up the instructions based on the program's Gödel number x.

ii. there are primitive recursive functions that allow for a universal program to perpetuate a simulation by extracting the necessary data from both the program encoding and the current-configuration encoding.

iii. a universal program possesses an infinite number of registers that enable it to handle programs of any size.

iv. All of the above are essential properties of P_U.

(b) A universal program P_U is simulating a program that has 50 instructions and whose Gödel number is

$$x = 2^{15} + 2^{20} + 2^{26} + 2^{45} + 2^{c_5} + \cdots + 2^{c_{50}} - 1.$$

If the current configuration of the computation of P_x on some input has encoding

$$\sigma = 2^4 + 2^6 + 2^{10} + 2^{13} + 2^{18} - 1,$$

then provide the next configuration of the computation *and* its τ encoding.

LO8. Do the following.

(a) In one or more complete sentences, describe what is asserted by the Church-Turing Thesis.

(b) Consider the function

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

In other words, $g(x) = 1$ iff the program having Gödel number x halts when its own Gödel number x serves as input, and it outputs 0 otherwise. We want to prove that $g(x)$ is undecidable, meaning there is no URM program that computes g. To do this, let's assume that $g(x)$ is computable by some URM program G. Then define the function $f(x)$ as follows.

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

Provide an informal description of the steps you would take to compute $f(x)$ using pencil and paper. Why does this imply that $f(x)$ is URM computable?

(c) Let e denote the Gödel number of the URM program F that computes $f(x)$ from part b. In other words $F = P_e$. Show that a contradiction arises when we try to compute $f(e)$ using F. Hint: consider the two cases when i) $f(e) = 1$ and ii) $f(e) =\uparrow$, meaning that $f(e)$ is undefined.

LO9. An instance of the decision problem `Is Predicate Function` is a Gödel number x, and the problem is to decide if function ϕ_x is a predicate function, meaning that, ϕ_x is a total function that always outputs either 0 or 1. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x \text{ is a predicate function} \\ 0 & \text{otherwise} \end{cases}$$

(a) Evaluate $g(x)$ for each of the following Gödel number's x. Note: 2 out of 3 correct is considered passing. **Justify your answers**.

i. $x = e_1$, where e_1 is the Gödel number of the program that computes the function $\phi_{e_1}(y) = y$.

ii. $x = e_2$, where e_2 is the Gödel number of the program that computes the function $\phi_{e_2}(y) = \text{sgn}(y)$.

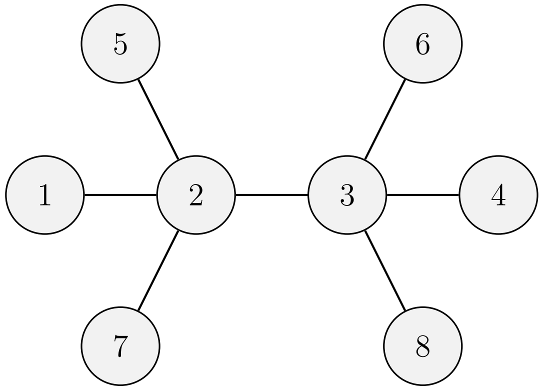
iii. $x = e_3$, where e_3 is the Gödel number of the program that computes $g(x)$ (assuming that $g(x)$ is URM computable).

(b) Prove that $g(x)$ is not URM computable. In other words, there is no URM program that, on input x, always halts and either outputs 1 or 0, depending on whether or not ϕ_x is a predicate function. Do this by writing a program P that uses g and makes use of the `self` programming concept. Then show how P creates a contradiction.

LO10. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction from decision problem A to decision problem B.

(b) For the mapping reduction f : `Vertex Cover` $\rightarrow$ `Half Vertex Cover`, draw $f(G, k)$ for the `Vertex Cover` instance whose graph is shown below, and for which $k = 2$.



(c) Verify that both (G, k) and $f(G, k)$ are either both positive instances, or are both negative ones. Explain and show work.

LO11. Recall the mapping reduction $f(F) = \mathcal{C}$, where f maps an instance of `SAT` to an instance of the `3SAT` decision problem. Given `SAT` instance

$$F(x_1, x_2, x_3) = \overline{x}_1 \vee (x_2 \wedge \overline{x}_3),$$

show each of the following steps towards computing $f(F)$.

(a) Draw F's parse tree, label its internal nodes with y-variables, and provide the initial Boolean formula that asserts that F is satisfiable.

(b) Convert the formula from part a to an equivalent one that uses only AND, OR, and NEGATION.

(c) Use De Morgan's rule and the distributive rule to convert your formula from part b to one that is an "AND or OR's".

(d) Convert the formula from part c to a `3SAT` instance by using `3SAT` notation, and duplicating literals whenever necessary in order to ensure that each clause has three literals.