

**CS 4124**  
**Solutions to Homework Assignment 4**  
**Collin McDevitt**

March 29, 2024

[30] 1. Let  $\Sigma = \{0, 1\}$ , the binary alphabet. Let

$$F = \{h \mid h : \Sigma^* \rightarrow \Sigma^*\},$$

the set of functions from strings to strings.

**Prove that  $F$  is uncountable.**

**HINT: Use the fact that  $\Sigma^*$  is countably infinite and employ a proof by diagonalization.**

---

My proof strategy:

I will prove that the set of functions  $F' = \{h : \Sigma^* \rightarrow \{0, 1\}\}$  is uncountable and as this set is a subset of  $F$  that would imply that  $F$  is uncountable. I will do this by constructing a bijection between  $H : F' \rightarrow \mathcal{P}(\Sigma^*)$  (the power set of  $\Sigma^*$ ) in the last homework we proved that no set can have a bijection with its power set and as  $\Sigma^*$  is countable we get that  $\mathcal{P}(\Sigma^*)$  is uncountable hence if a bijection  $H$  exists we would have that  $F'$  is uncountable.

Let  $H : F' \rightarrow \mathcal{P}(\Sigma^*)$  where  $H(f) = \{w \in \Sigma^* : f(w) = 1\}$  (informally this is the set of elements from  $\Sigma^*$  who get mapped to 1).

Now to show that this is a bijection I will come up with an inverse. We create the function

$$H^{-1} : \mathcal{P}(\Sigma^*) \rightarrow F'$$

where for  $A \in \mathcal{P}(\Sigma^*)$  we have  $H^{-1}(A) = f$  where for all  $w \in \Sigma^*$  we have  $f(w) = 1$  if and only if  $w \in A$ . The fact that  $H^{-1}$  is not a multivalued function can be proven by assuming that for some  $A \in \mathcal{P}(\Sigma^*)$  that there exists two  $f_1, f_2 \in F'$  such that  $H^{-1}(A) = f_1$  and  $H^{-1}(A) = f_2$  then we have that for all  $w \in \Sigma^*$  that  $f_1(w) = 1$  and  $f_2(w) = 1$  if and only if  $w \in A$  which directly implies that for any  $\omega \in \Sigma^*$  where  $\omega \notin A$  that  $f_1(\omega) = 0$  and  $f_2(\omega) = 0$ . As  $f_1, f_2$  are equal at every input we have that  $f_1 = f_2$ .

Now to show that let  $f \in F'$  then we have  $H(f) = \{w \in \Sigma^* : f(w) = 1\}$ . Then take  $H^{-1}(\{w \in \Sigma^* : f(w) = 1\}) = f_0$  for some  $f_0 \in F'$ . But from the definition of  $H^{-1}$  we get that  $f_0(w) = 1$  if and only if  $w \in \{w \in \Sigma^* : f(w) = 1\}$  which is the same as  $f(w) = 1$  so we have that  $f_0 = f$ . This implies that  $H^{-1} \circ H = \text{Id}_{F'}$ . Now let  $A \in \mathcal{P}(\Sigma^*)$  then we have that  $H^{-1}(A) = f$  for some  $f \in F'$ . Then taking  $H(f) = \{w \in \Sigma^* : f(w) = 1\}$  but from the definition of  $f$  we have that for all  $\omega \in \Sigma^*$  that  $f(\omega) = 1$  if and only if  $\omega \in A$ . This implies that  $H(f) = A$  so we have  $H \circ H^{-1}(A) = A$  so we have that  $H$  is the inverse of  $H^{-1}$  so we have that  $H$  is a bijection.

Then as a bijection between  $F' \subset F$  to  $\mathcal{P}(\Sigma^*)$  exists and  $\mathcal{P}(\Sigma^*)$  is uncountable we get  $F'$  is uncountable and so  $F$  is uncountable as well.

---

[30] 2. Let

$$E = \{w \in \{0, 1\}^* \mid \ell(w) \text{ is even}\},$$

**Program  $P_x$**   
**if**  $x$  halts on input  $y$   
  **then** run program  $x$  on input  $y$   
    **if**  $y$  has even length  
      **then** accept  
      **else** reject  
  **else** loop forever

the language of binary strings of even length. See Figure 2 for a sample program that decides  $E$ , just for L<sup>A</sup>T<sub>E</sub>X formatting purposes. Let

$$L = \{x \mid \text{program } x \text{ decides } E\}$$

be the language of programs that decide  $E$ .

**Prove that  $L$  is undecidable by using m-reducibility.**

I will use the M reduction  $HP \leq_M L$ . I will do this by creating the function sending  $\langle x, y \rangle \rightarrow P_x$  where  $P_x$  is the program defined above.

[40] **3.** Let

$$\text{DOUBLE} = \{ww \mid w \in \{a, b\}^*\}.$$

Let

$$W = \{x \mid \text{program } x \text{ decides DOUBLE}\}$$

be the language of programs that decide DOUBLE.

**A. Is DOUBLE decidable? Justify your answer.**

**B. Is  $W$  decidable? Prove your answer.**

A: Yes DOUBLE is decidable. We could create the program that takes in an input and checks if the input has even length and checks if the input is a palindrome. If both of these are true we accept otherwise we reject. This program would be able to decide DOUBLE as the only strings that are in DOUBLE are palindromes of even length.