Student ID: ______________________

Collaborators: ______________________

CS181 Winter 2018 – Problem Set 4

Due Wednesday, February 28, 11:59 PM

- Please write your student ID **and the names of anyone you collaborated with** in the spaces provided and attach this sheet to the front of your solutions. **Please do not include your name anywhere since the homework will be blind graded.**
- An extra credit of **5%** will be granted to solutions written using LaTeX. Here is one place where you can create LaTeXdocuments for free: `https://www.sharelatex.com/`. The link also has tutorials to get you started. There are several other editors you can use.
- If you are writing solutions by hand, please write your answers in a neat and readable hand-writing.
- Always explain your answers. When a proof is requested, you should provide a rigorous proof.
- If you don't know the answer, write "I don't know" along with a clear explanation of what you tried. For example: "I couldn't figure this out. I think the following is a start, that is correct, but I couldn't figure out what to do next. [[Write down a start to the answer that you are sure makes sense.]] Also, I had the following vague idea, but I couldn't figure out how to make it work. [[Write down vague ideas.]]" At least 20% will be given for such an answer.

  Note that if you write things that do not make any sense, no points will be given.
- The homework is expected to take anywhere between 6 to 10 hours. You are advised to start early.
- Submit your homework online on the course webpage on CCLE. You can also hand it in at the end of any class before the deadline.

1.
FRTMs are very similar to regular Turing Machines(TM). As a result, we can keep most of the functionality of the original TMs when converting a TM to a FRTM. We can keep the original set of states and all the R-transitions (transitions that move the tape head to the right).

We have to change three things. First, we need to change both our stack and input alphabet. We need to have marked counterparts of the symbols that are already in the alphabet. For example if this is the original alphabet:

$$\Sigma = \{a, b\} \tag{1}$$

$$\Gamma = \{\sqcup, a, b\} \tag{2}$$

Then the new alphabet would be like this:

$$\Sigma = \{a, b, a', b'\} \tag{3}$$

$$\Gamma = \{\sqcup, a, b, a', b'\} \tag{4}$$

The marked versions of the symbols will serve to mark where the first/left-most cell on the tape is located.

The second thing we need to change is the symbol at the beginning of the tape. We simply replace it with the marked version of the symbol, so we know that this is the first cell in the tape.

The last thing we have to change is the way we deal with L-transitions. We can keep the original L-transitions. However, we must add more L-transitions after the initial L-transition since we must move the head of the tape all the way to the left. To achieve this, we can add a loop that keeps moving the head left until it reaches the beginning of the tape. Once the head is at the first cell, it will read the marked symbol and leave the loop via another transition. To demonstrate this, I will use an example. Let's say the input "a" causes a state transition from q_1 to q_2 and moves the head to the left one space like so:

$$a'bbbq_1ab \tag{5}$$

$$a'bbq_2bab \tag{6}$$

Since there is a loop at q_2 that will continually move the head left on any unmarked input, we keep moving the head left like so:

$$a'bq_2bbab \tag{7}$$

$$a'q_2bbbab \tag{8}$$

$$q_2a'bbbab \tag{9}$$

Once we reach the left-most cell, the machine will take another transition which accepts a marked symbol as input. With this mechanism, we have accomplished what the FIRST transition does in the FRTM using a TM.

2a.

Theorem 2a. The set of three-dimensional coordinates {(x,y,z) | x,y,z $\in \mathbb{Z}$} has size equal to $\mathbb{N}$.

Proof. To prove this theorem, I will first map each individual component in (x,y,z) to $\mathbb{N}$ using the bijection we discussed in class, so I don't have to deal with negative numbers in my later calculations. After that, I will take the result and map the entire coordinate to a natural number $\in \mathbb{N}$ using an equation and the Fundamental Theorem of Arithmetic.

To begin, I will map each of x,y,z in (x,y,z) to a natural number. Since x,y,z $\in \mathbb{Z}$, I can use the bijection used in class to map $\mathbb{Z}$ to $\mathbb{N}$. For reference, here is the bijection:

$$\begin{cases} x = 0 \to 1 \\ x = a \to 2a \\ x = -a \to 2a + 1 \end{cases}$$

We do this for both the y and z components of the coordinate as well. Let's call the resulting coordinate (x',y',z'). Now that we know x',y',z' $\in \mathbb{N}$, we can apply the second part of the plan to complete our bijection. We insert x',y',z' into the following equation to map the coordinate to a unique natural number:

$$n = 2^{x'}3^{y'}5^{z'} \tag{10}$$

The following is the equation represented in a mapping:

$$\begin{cases} x = (x', y', z') \to 2^{x'}3^{y'}5^{z'} \end{cases}$$

This equation is guaranteed to map each unique coordinate to a unique natural number due to the Fundamental Theorem of Arithmetic. The theorem states that there is exactly one way to represent any positive integer using a product of prime powers. The equation above is indeed a product of prime powers so it must generate a unique natural number for each coordinate. To summarize, here is an example of the entire process:

$$(0, -2, 3) \to (1, 5, 6) \tag{11}$$

$$(1, 5, 6) \to 2^1 3^5 5^6 \to 2 * 243 * 15625 = 7593750 \tag{12}$$

And here's another example using a similar coordinate:

$$(0, 3, -2) \to (1, 6, 5) \tag{13}$$

$$(1, 6, 5) \to 2^1 3^6 5^5 \to 2 * 729 * 3125 = 4556250 \tag{14}$$

The injection going the opposite direction is trivial to prove. Using the Fundamental Theorem of Arithmetic, we can say that for each natural number n, there is exactly one triplet that satisfies the equation n = $2^{x'}3^{y'}5^{z'}$. Thus our bijection is complete.

Since the composition of two bijections is a bijection, there exists a bijection between (x,y,z) and $\mathbb{N}$. Since there exists a bijection between (x,y,z) and $\mathbb{N}$, that means |(x,y,z)| and $|\mathbb{N}|$ are equal.

$\square$

2b.

Theorem 2b. |Sym| and |ℕ| where Sym is the set of all languages decidable by a Symmetric NFA.

Proof. An SNFA accepts all regular languages. This is due to the fact that an SNFA can always be converted to a NFA by removing the extra δ transitions. Using this fact, we can come up with a bijection that maps all languages decidable by a SNFA to the set of all natural numbers.

Since an SNFA can accept all regular languages, we can use the fact that any language that has a deterministic length is a regular language. An example of a deterministic length language is 1221, which has a length of 4. An example of a non-deterministic length language is something like $1^n 2^n$ because it accepts strings of varying lengths.

Using the facts already stated, we can proceed to map each language decidable by a SNFA to a natural number. To do this, allow me to demonstrate with an example that features a language L such that L $\in$ {1,2}*.

$$\begin{cases} x = 1 \to 1 \\ x = 2 \to 2 \\ x = 11 \to 3 \\ x = 12 \to 4 \\ x = 21 \to 5 \\ x = 22 \to 6 \\ x = 111 \to 7 \\ ... \\ x = x_1 x_2 ... x_{\mathrm{p}} \to 2^{p-1}x_1 + 2^{p-2}x_2 + ... + 2^0 x_{\mathrm{p}} \end{cases}$$

An explanation of the above example is as follows. Since these languages feature two characters in their alphabet, I can map each language using the conversion equation in the last case. Now you might ask what happens if a language has non-deterministic length and is regular, such as 1*2*. How would it be represented in the above mapping? The answer is that it is in fact represented in the relation. Using the example language 1*2*, we can see that the mapping handles 1, 11, 12, 111, etc.

The next question that could be raised is "How do we handle the language L such that L $\in$ {0,1}*?" Let's try the previous mapping:

$$\begin{cases} x = 0 \rightarrow 1 \\ x = 1 \rightarrow 2 \\ x = 10 \rightarrow 3 \\ x = 11 \rightarrow 4 \\ x = 100 \rightarrow 5 \\ x = 101 \rightarrow 6 \\ x = 110 \rightarrow 7 \\ ... \\ x = x_1 x_2 ... x_{\mathrm{p}} \rightarrow 2^{p-1}x_1 + 2^{p-2}x_2 + ... + 2^0 x_{\mathrm{p}} + 1 \end{cases}$$

On the surface, this mapping seems right. However, it will fail to map 0* because there is no difference in value whatsoever between 0, 00, 000, etc. so the equation will fail to generate a unique natural number for these strings. We can get around this issue by mapping {0,1} to something like {1,2}. From there, we can use the mapping from above.

Another question that can be raised is "How do we deal with languages with alphabets that do not have 2 symbols?". We can modify the equation to deal with this. Let's say there is a language L ∈ {1,2,3}*. We can use this mapping:

$$\begin{cases} x = 1 \rightarrow 1 \\ x = 2 \rightarrow 2 \\ x = 3 \rightarrow 3 \\ x = 11 \rightarrow 4 \\ x = 12 \rightarrow 5 \\ x = 13 \rightarrow 6 \\ x = 21 \rightarrow 7 \\ ... \\ x = x_1 x_2 ... x_{\mathrm{p}} \rightarrow 3^{p-1}x_1 + 3^{p-2}x_2 + ... + 3^0 x_{\mathrm{p}} \end{cases}$$

Notice the change in the coefficients of the equation from 2 to 3. This reflects the change in the size of the alphabet. Using this information, we can generalize our equation and mapping:

$$\begin{cases} x = x_1 x_2 ... x_{\mathrm{p}} \rightarrow m^{p-1}x_1 + m^{p-2}x_2 + ... + m^0 x_{\mathrm{p}} \,\text{such that}\, m = \|\Sigma\|, 0 \notin \Sigma \end{cases}$$

Now that we have a mapping that accommodates for all alphabet sizes, we are almost done. However one issue still remains. What if the alphabet contains symbols that are not numbers and thus cannot be plugged into the equation? For example, what if our alphabet Σ = {a,b,c}? In this case, we can use a mapping that maps each non-number symbol to a natural number. For example, we could map {a,b,c} to {1,2,3} and use the equation and mapping.

The injection going the opposite direction is trivial to prove. Notice that given a natural number n, there is exactly one string that satisfies the equation n = $m^{p-1}x_1 + m^{p-2}x_2 + ... + m^0 x_{\mathrm{p}}$ assuming that $\|\Sigma\|$ = m. Now our bijection is complete.

With our generalized equation and bijection, we have proven that there exists a bijection between languages that are decidable by a SNFA and the set of all natural numbers. Thus, $|\mathrm{Sym}| = |\mathbb{N}|$. $\square$