

CSCE 222 Discrete Structures for Computing – Fall 2022

Hyunyoung Lee

Problem Set 8

Due dates: Electronic submission of *yourLastName-yourFirstName-hw8.tex* and *yourLastName-yourFirstName-hw8.pdf* files of this homework is due on **Monday, 12/5/2022 before 11:59 p.m.** on <https://canvas.tamu.edu>. You will see two separate links to turn in the .tex file and the .pdf file separately. Please do not archive or compress the files. **If any of the two files are missing, you will receive zero points for this homework.** Your files must contain your name and UIN in the given spaces and the electronic signature (your full name) correctly; otherwise, you may receive zero points for this homework

Name: (Blake Dejohn)**UIN:** (531002472)

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

Resources Overall (used for the whole document)

1. N/A

Problem 1

1. Section 17.1 - Languages

Problem 2

1. Section 17.1 - Languages
2. Problem Solving Exercise: Formal Languages Basics - <https://www.youtube.com/watch?v=CKGMHjzTAKg>
3. Geometric series closed-form equation - <https://www.youtube.com/watch?v=JJZ-shHiayU>

Problem 3

1. Section 17.2 - Grammars
2. Problem Solving Exercise: Grammars- <https://www.youtube.com/watch?v=IG9zfZklfxI>

Problem 4

1. Problem Solving Exercise: Grammars- <https://www.youtube.com/watch?v=IG9zfZklfxI>

Problem 5

1. Problem Solving Exercise: Grammars- <https://www.youtube.com/watch?v=IG9zfZklfxI>

Problem 6

1. Problem Solving Exercise: Grammars- <https://www.youtube.com/watch?v=IG9zfZklfxI>

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to answer this homework.

Electronic signature: (Blake Dejohn)

Total 100 + 10 (bonus) points. *Explanation will be worth a large portion of the grade for each problem.*

The intended formatting is that this first page is a cover page and each problem solved on a new page. You only need to fill in your solution between the `\begin{solution}` and `\end{solution}` environment. Please do not change this overall formatting.

Checklist:

- ✓ Did you type in your name and UIN?
- ✓ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)
- ✓ Did you sign that you followed the Aggie Honor Code?
- ✓ Did you solve all problems?
- ✓ Did you submit both the .tex and .pdf files of your homework to each correct link on Canvas?

Problem 1. (15 points) Section 17.1, Exercise 17.1. *Explain.*

Solution.

Explain whether or not they are elements of $\{0, 1\}^*$

Notation

Given a set of string of finite length over an alphabet denoted by $\{0, 1\}^*$, I will denote the alphabet that this set of strings uses as alphabet $A = \{0, 1\}$.

(i) The empty string ϵ

Every A^* (i.e. the set of strings of finite length over an alphabet A) contains the empty string because when concatenating the empty string to any element, you just get the element you started with. Therefore, following this rule, since $\{0, 1\}^*$ is also a set of strings of finite length over an alphabet given by $A = \{0, 1\}$, the empty string by itself is **contained** in the given alphabet because of the fact that it is an implicit element given in the alphabet definition.

(ii) 01110

This string is **contained** in $\{0, 1\}^*$. This is because given that this string is of finite length and only contains elements that are within the alphabet, this string obeys the conditions for whether or not it can be a string contained in $\{0, 1\}^*$ and is therefore contained in $\{0, 1\}^*$.

(iii) 01211

This is **not contained** in $\{0, 1\}^*$. This is because the element 2 is not contained in the alphabet, therefore every string that is contained in $\{0, 1\}^*$ should only have elements that are either 0 or 1.

(iv) 1111111..., the infinite string of 1s

Given the definition of $\{0, 1\}^*$ (i.e. the set of strings of finite length over the alphabet A which contains 1 and 0), we can see that any string contained in $\{0, 1\}^*$ must be of a finite length. While there might be an infinite amount of strings that are contained in $\{0, 1\}^*$, each of these strings themselves cannot have infinite length, rather they must be finite. Given this, it can be seen that this infinite strings of 1s is **not contained** in $\{0, 1\}^*$.

Problem 2. (15 points) Section 17.1, Exercise 17.2. *Explain.* [Hint: To find a compact (closed form) expression of the summation, write out the summation and use the method we used to find the closed form for the geometric series.]

Solution.

How many words does $A_{\leq n}$ contain letting A be an alphabet containing k symbols with $k \geq 2$ and $A_{\leq n}$ denote the subset of A^* that contains all words of length n or less.

To illustrate the general solution to this problem, an example problem with fixed values will be solved.

Given a binary alphabet A that contains $\{0, 1\}$ (which means $k = 2$) and a word length, or n , of 3, we can see that according to the multiplication principle, there are 2^3 or 8 different possible binary strings of size 4 that can be made. This is because for each place in the string, there are only 2 possible elements that place in the string can hold, either 0 or 1. Then, when each of these possibilities are multiplied to each other by the number of places in the string length, you get the total amount of different, possible strings given the conditions. After this, n can be decreased by one and the same procedure can be done until n equals zero to get the total number of strings that are either of length n or less. After this, the summation principle can be used to add up the number of possible strings since because of their differing length, these strings are fundamentally different and therefore only need to be added in order to find the total amount of differing strings. So, in this case, the total amount of binary strings that are of length 3 or less is denoted by $2^3 + 2^2 + 2^1$ or 13.

Applying this to a general formula where n and k are not some fixed value can be found by writing out the sum a general sum, found from the specific case, and finding the closed form of the geometric series that is found through

the use of algebra.

$$\begin{aligned} A_{\leq n} &= \sum_{i=0}^n k^{n-i} \\ &= k^n + k^{n-1} + k^{n-2} + k^{n-3} + \dots + k + 1 \text{ (since } k^{n-n} = 1 \text{ or the empty string)} \end{aligned}$$

$$\begin{aligned} \frac{1}{k} * A_{\leq n} &= k^{n-1} + k^{n-2} + k^{n-3} + \dots + 1 + \frac{1}{k} \\ A_{\leq n} - \frac{1}{k} * A_{\leq n} &= k^n - \frac{1}{k} \\ A_{\leq n} \left(1 - \frac{1}{k}\right) &= k^n - \frac{1}{k} \\ A_{\leq n} &= \frac{k^n - \frac{1}{k}}{1 - \frac{1}{k}} \\ A_{\leq n} &= \frac{\frac{k^{n+1}}{k} - \frac{1}{k}}{\frac{k}{k} - \frac{1}{k}} \\ A_{\leq n} &= \frac{\frac{k^{n+1}-1}{k}}{\frac{k-1}{k}} \\ A_{\leq n} &= \frac{k^{n+1}-1}{k} * \frac{k}{k-1} \\ A_{\leq n} &= \frac{k^{n+1}-1}{k-1} \end{aligned}$$

Finding the closed form of the geometric series gotten from the above rationale and using algebra to simplify the equation, it can be seen that from k number of symbols within an alphabet and words with length n or less, $A_{\leq n}$ or the number of words of length n or less is given by

$$A_{\leq n} = \frac{k^{n+1} - 1}{k - 1}$$

Problem 3. (20 points) Section 17.2, Exercise 17.6. *Explain.*

Solution.

Let $G = (N, T, P, S)$ be a grammar with $T = \{0, 1\}$, $N = \{S, A\}$, and

$$P = \{S \rightarrow 0S0, S \rightarrow 0A0, A \rightarrow 1A, A \rightarrow 1\}$$

Determine $L(G)$.

To find $L(G)$, it will help to first find some example strings that can be contained in the language based on the grammar rules. For instance, some strings that could be in $L(G)$ include:

$$L(G) = \{010, 0110, 01110, 00100, 001100, 0001000, \dots\}$$

Below, the process these sample strings were gotten will be modeled with a few examples. First, to make notation easier however, the grammar rules in the following processes will be numbered 1 through 4. (i.e. grammar rule 1 = $S \rightarrow 0S0$, rule 2 = $S \rightarrow 0A0$, rule 3 = $A \rightarrow 1A$, and rule 4 = $A \rightarrow 1$)

For 010:

$$\begin{aligned} S &\rightarrow 0A0 \quad \text{by rule 2} \\ 0A0 &\rightarrow 010 \quad \text{by rule 4} \end{aligned}$$

For 0110:

$$\begin{aligned} S &\rightarrow 0A0 \quad \text{by rule 2} \\ 0A0 &\rightarrow 01A0 \quad \text{by rule 3} \\ 01A0 &\rightarrow 0110 \quad \text{by rule 4} \end{aligned}$$

For 00100:

$$\begin{aligned} S &\rightarrow 0S0 \quad \text{by rule 1} \\ 0S0 &\rightarrow 00A00 \quad \text{by rule 2} \\ 00A00 &\rightarrow 00100 \quad \text{by rule 4} \end{aligned}$$

Starting from the smallest possible string, which is 010, it can be seen that within the strings contained in this language, there is always at least one 1 between at least one couple of 0's. However, looking at the example strings, it can be seen that while there is always the same number of zeros between each side of at least one 1, there is not always a 2-to-1 ratio between zeros and ones in the string (i.e. for every 2 zeros outside, there is one 1 inside). Therefore, it can be reasoned that these terminals symbols, which are 1 and 0, are not increased by the same number/do not have the same powers. Rather, they are

influenced separately which should denote that they have different powers in the set notation. Given this rationale, the set notation that best describes this language includes

$$L(G) = \{0^b 1^a 0^b \mid a \geq 1, b \geq 1\}$$

Problem 4. (20 points) Section 17.2, Exercise 17.8. *Explain.*

Solution.

Give a grammar G such that $L(G) = \{0^n 1^n \mid n \geq 1\}$

Similar to finding languages using grammar rules, it will be useful to look at sample strings that belong to the language.

Given the rules of the language,

$$L(G) = \{01, 0011, 000111, 00001111, \dots\}$$

From here, it can be seen that there are always an amount of zeros (that start from 1 0) that are followed by an equal number of 1s. Another observation that is important in this language is the fact that the empty string is not included, which became the case since $L(G)$ only includes n 's that are greater or equal to 1. From this observation, it should be noted that at least 2 non-terminal symbols will be needed for the grammar that describes this language, since the starting symbol S cannot transition to being the empty string itself. This is important because the empty string is also needed in the grammar for this language. Given that there is always an equal amount of 0's and 1's that come in that order, some grammar rules which make a terminal symbol be able to transition to itself between 0 and 1 while also being able to become the empty string ensures the ability for the grammar to be recursive and have the same number of 0's and 1's, but also be able to stop at some finite length thanks to the non-terminal symbol becoming the empty string at some point. These grammar rules can be denoted with something like $A \rightarrow 0A1$ and $A \rightarrow \epsilon$. From here, it can be seen that as A transitions to $0A1$ multiple times, there is always an amount of zeros followed by an equal amount of 1's, which is what is required. Then, after some length is obtained, A can become the empty string to "eliminate" itself and leave the desired string behind.

Given this rationale, the grammar that best describes this language is given by:

$G = (N, T, P, S)$ where $N = \{S, A\}$, $T = 0, 1$, S is the starting symbol, and $P\{S \rightarrow 0A1, A \rightarrow 0A1, A \rightarrow \epsilon\}$

Problem 5. (20 points) Section 17.3, Exercise 17.13. *Explain.*

Solution.

Find a type 3 (or regular) grammar $G = (N, T, P, S)$ with $T = \{0, 1\}$ that generates the language

$$L(G) = \{\omega \in T^* \mid \omega \text{ is of even length}\}$$

In order for a string to be of even length in this context, it either needs to be the empty string ϵ or have a length that is a multiple of 2. Also, since the terminal characters within this language have no restrictions (i.e. every place in the string should have the chance to be a zero or one), every non-terminal character used in the grammar for this language will have to have production rules that allow it to be transformed to a 0 or 1, no matter the position. Finally, a good approach to this problem includes having 2 non-terminal characters that denote whether the current string is even or odd. What this means is that, if the string has a non-terminal character of say D within it still, then that could denote that the string is odd (since it was just changed from the other non-terminal character which denotes evenness) and therefore needs to change back to the other non-terminal character so that it can become even again and possibly finish the string since the non-terminal character that was just transitioned into should have the ability to become the empty string. Overall, this non-terminal character should have this ability because there should be no way for the string to be finished if it has an odd number of elements. Therefore, only the non-terminal symbol that denotes evenness should have the ability to finish the string (i.e. turn into the empty string). This evenness will be set up by counting both the non-terminal and terminal symbols in the current string. With these rules in mind, a grammar could be set up like this:

$G = (N, T, P, S)$ where $N = \{S, A\}$, $T = 0, 1$, S is the starting symbol, and

$$P\{S \rightarrow \epsilon, S \rightarrow 1A, S \rightarrow 0A, A \rightarrow 0S, A \rightarrow 1S\}$$

Also, confirming this is a type 3 grammar means comparing the production rules with the acceptable type 3 rules. Overall, a type 3 grammar is defined as the following.

If all the production rules in P are of one of the following forms, then the grammar is type 3.

1. $A \rightarrow aB$ for terminal symbols A and B in N and a in T
2. $A \rightarrow a$ for terminal symbol A in N and a in T
3. $A \rightarrow \epsilon$ for terminal symbol A in N

Given this, it can be seen that $S \rightarrow 1A$, $S \rightarrow 0A$, $A \rightarrow 0S$, and $A \rightarrow 1S$ follow the type 3 rule 1 and $S \rightarrow \epsilon$ follows type 3 rule 3.

Problem 6. (20 points) Section 17.4, Exercise 17.17. *Explain.*

Solution.

Give a type 3 grammar for the formal language given in Example 17.17

A type 3 grammar is defined as the following.

If all the production rules in P are of one of the following forms, then the grammar is type 3.

1. $A \rightarrow aB$ for terminal symbols A and B in N and a in T
2. $A \rightarrow a$ for terminal symbol A in N and a in T
3. $A \rightarrow \epsilon$ for terminal symbol A in N

From Example 17.17, it can see that the formal language gotten from the state diagram is $L(G) = \{10^n \mid n \geq 0\}$. From this, a grammar can be formed.

Again, it is useful to start from example strings that are contained within the language.

$$L(G) = \{1, 10, 100, 1000, 10000, \dots\}$$

Since there is only one 1 within each string in the language, we can have the starting symbol S take care of the 1 by giving it a production rule that includes a 1 within it. Then, after seeing that all the strings have some number of 0's (starting from zero 0's) within them, another production rule can be thought of were a terminal symbol transitions to itself with an extra zero attached to it. This handles the recursive aspect of this language. However, since each string in this language must have a finite length, making this terminal symbol be able to become the empty string through a production rule will also be needed. But, this non-terminal symbol cannot be the starting symbol S since that would mean that the empty string would be included in the language, which it is not. So, another non-terminal symbol other than S will be needed in the grammar. Keeping all this in mind, it can seen that the grammar for this language can be characterized as:

$G = (N, T, P, S)$ where $N = \{S, A\}$, $T = 0, 1$, S is the starting symbol, and $P\{S \rightarrow 1A, A \rightarrow 0A, A \rightarrow \epsilon\}$

Finally, checking that this grammar is of type 3, we can see that grammar rule $S \rightarrow 1A$ corresponds to type 3 rule 1, grammar rule $A \rightarrow 0A$ obeys type 3 rule 1 also, and grammar rule $A \rightarrow \epsilon$ is in agreement with type 3 rule 3.