Middle East Technical University Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

Spring 2022-2023

Homework 4

Regulations (Please Read!)

1. The homework is due by **23:59 on May 7th, 2023**. Late submission is not allowed.
2. Submissions will be collected via ODTUClass. Do not send your homework via e-mail.
3. You can use any typesetting tool (LaTex, Word, etc.) while writing the homework. However, you must upload the homework as a **searchable pdf file**. Other formats (e.g handwriting etc.) will **not** be considered for grading.
4. Name the PDF file you submit as **HW4_yourStudentID.pdf** (e.g. HW4_1234567). Submissions violating the naming convention will be penalized.
5. Write your name and student ID number to the top of your solution sheets. A grade reduction will be applied to the solution sheets without a name and ID on them.
6. Send an e-mail to **both garipler@metu.edu.tr** and **bugra@ceng.metu.edu.tr** if you need to get in contact.
7. Please give as neat and as brief answers as possible.
8. **This is an individual homework, which means you have to answer the questions on your own. Any contrary case including but not limited to getting help from automated tools, sharing your answers with each other, extensive collaboration etc. will be considered as cheating and university regulations about cheating will be applied.**

Question 1 (40 pts)

Formally **define** (in $(K, \Sigma, \Gamma, \Delta, s, F)$ form) and **draw** a PDA for each of the following languages.

1. $\{w\#x \mid x^R \textit{ is a substring of } w \textit{ for } w, x \in \{a, b\}^*\}$
2. $\{c^n ww^R c^n | w \in \{a, b\}^* \wedge n \geq 0\}$

Note: w^R is the reversed form of the string w. To illustrate, given $w = abcd$, $w^R = dcba$.

Question 2 (30 pts)

It is known that the class of Context-free Languages is closed under Kleene Star. Give a counterexample (with clear definition of a grammar) to show that for an arbitrary CFL $L = L(G)$ where $G = (V, \Sigma, R, S)$ adding rule $S \rightarrow SS$ does not generate L^*.

Question 3 (30 pts)

Let us modify the definition of pushdown automata such that they have only one stack symbol. In the scope of this homework, we will call this special type of PDA as ***S-PDA*** and the class of languages that are recognized by S-PDAs as ***S-CFL*** (S-ContextFreeLanguages).

1. Which ones of the languages given below are S-CFLs? Briefly explain your reasoning for each.
   - $L_1 = \{a^n b^n | n \geq 0\}$
   - $L_2 = \{w | w \in \{a, b\}^* \textit{ and the number of } a's \textit{ in } w \textit{ is not equal to the number of } b's \textit{ in } w\}$
   - $L_3 = \{a^n b^{m+n} c^m | m, n \in \mathbb{N}\}$
2. Give another example of S-CFLs. Define the language with set definition and write the context-free grammar that generates the language.
3. We can roughly say that PDAs are finite automata equipped with an (infinite) stack. Note that S-PDAs are restricted in terms of computational power in comparision to the general PDAs. Then, finite automata equipped with a smaller / less powerful memory element instead of a stack is should be equivalent to S-PDAs in terms of computational power. What is that memory element? In other words, try to find the minimal memory element that must be added to finite automata so as to increase their computational power such that they will be able to recognize S-CFLs.
   Remark1: Very brief answers are expected. Write only the name or the definition of the memory element you proposed.
   Remark2: You will get (potentially full) credit even if the memory element you proposed is not the minimal possible alternative. However, is must have less capacity/power than an infinite stack.
   Hint1: Notice that grammars generating S-CFLs have a similar restriction about non-terminal symbols. While thinking, starting with this property may be helpful.
   Hint2: Thinking through the NFA construction method may also help. This may help you to find what property finite automata are lacking.
   Hint3: Make use of non-determinism similarly as non-deterministic PDAs do.
   Hint4: Given hints may or may not work for you. You may also find a valid answer with another way of thinking.

4. What is the function of the memory element you added (to finite automata)? Why finite automata must be extended with that element so as to be able to recognize S-CFLs? Explain briefly. Also roughly explain the operational semantics of the machine type you designed (i.e. how it works) in a few sentences.

5. Is the class of S-CFLs closed under complementation? If yes, give a sketch of proof. If no, give a counterexample.