CS375 Homework Assignment 7 (40 points)

Due date: April 5, 2025

1. (5 points)

The language generated by the following grammar is [ ]
(2 points)

- S →abcD | abD   D → dD | Λ

This grammar is LL([ ]). (1 point)
Use left-factoring we can find an equivalent LL(k) grammar for the above grammar where k is the smallest choice for such an integer. In the following, fill out the blank in the middle portion to make the resulting grammar such an LL(k) grammar.

S → abT   [ ]   D → dD | Λ   (1 point)

What is the value of k? k = [ ] (1 point)

2. (6 points)

The language generated by the following grammar is [ ]
(2 points)

S → SaaaS | b

This grammar is left recursive, hence, it is not LL(k) for any k.
Fill out the blank below to make the resulting grammar equivalent to the above grammar but with no left recursion.

S → bT   [ ]   (2 points)

Is the resulting grammar LL(k)? [ ] Yes [ ] No   (1 point)

If your answer is YES, then what is the value of k? k = ☐ (1 point)

3. (5 points; 1 point each blank)

The following given grammar is a left recursive grammar

S → Sabcd | abc | ab

The language generated by this grammar is L = { $abc(abcd)^m,\ ab(abcd)^n \mid m, n \in N$ }.

This left recursive grammar can be transformed to a right recursive grammar as follows:

S → ☐ | ☐

T → ☐ ☐

This right recursive grammar is an LL(☐) grammar.

4. (7 points; 1 point each blank)

The following grammar is an indirect left recursive grammar

S → Babc | aa   B → Sabc | b

The language generated by this grammar is

☐ (2 points)

This indirect left recursive grammar can be transformed to a right recursive grammar as follows:

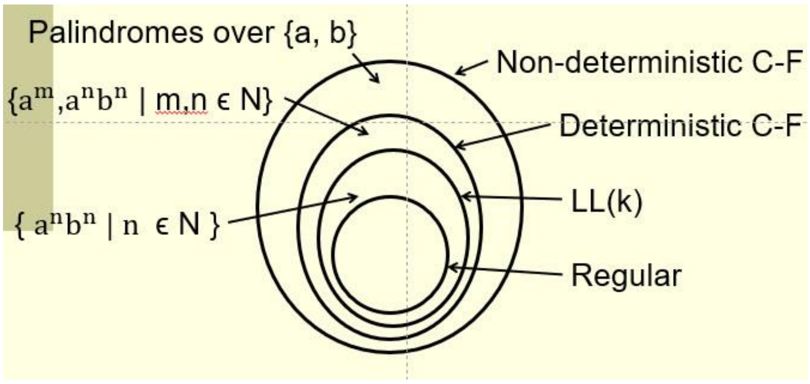
S → ☐ | ☐ (2 points)

T → ☐ ☐ (2 points)

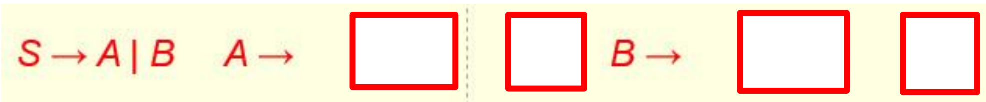
This right recursive grammar is an LL(☐) grammar. (1 point)

5. (7 points)

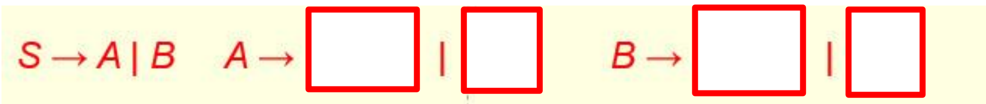
In slide 41 of the notes "Context-free Languages and Pushdown Automata IV", it is shown that the set of LL(k) languages is a proper subset of the set of deterministic C-F languages (or see the following figure). In particular, it points out that the language $\{a^m, a^n b^n \mid m, n \in N\}$ is a deterministic C-F language, but not LL(k) for any k.



To show the language is not LL(k) for any k, note that a grammar for this language is



or

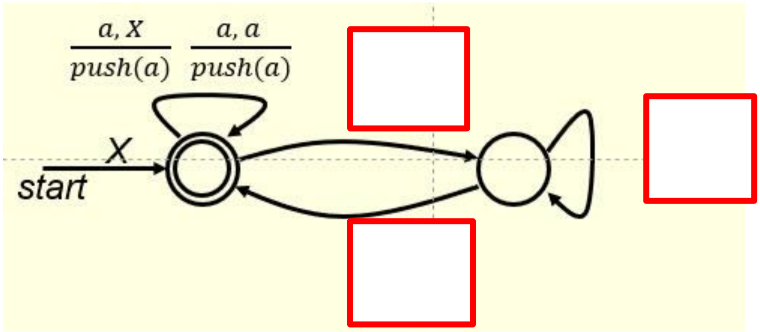


(4 points)

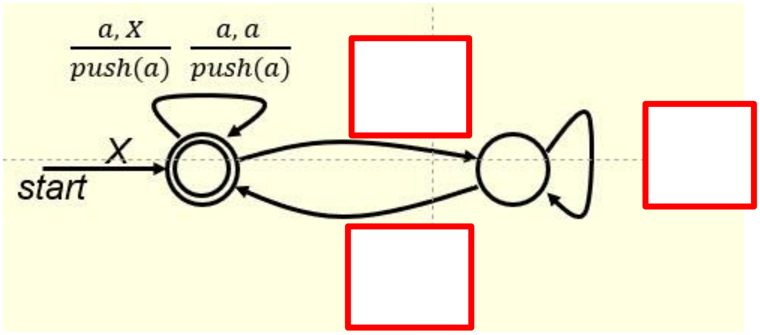
(you only need to answer one case here, either one). The language contains Λ as an element. Now consider the case k = 1 and consider the input string ab. When the first symbol is scanned, we get an 'a'. This information alone is not enough for us to make a proper choice. So we don't even know what to do with the first step in the parsing process.

For k = 2, if we consider the input string aabb, we face the same problem. For any k > 2, the input string $a^k b^k$ would cause exactly the same problem. So this grammar is not LL(k) for any k.

On the other hand, by putting proper instructions into the blanks of the following figure, we get a deterministic final-state PDA that accepts the language $\{a^m, a^n b^n \mid m, n \in N\}$.



or



(3 points)

(again, you only need to answer one case here, either one). Hence, this language is indeed deterministic C-F, but not LL(k) for any k.

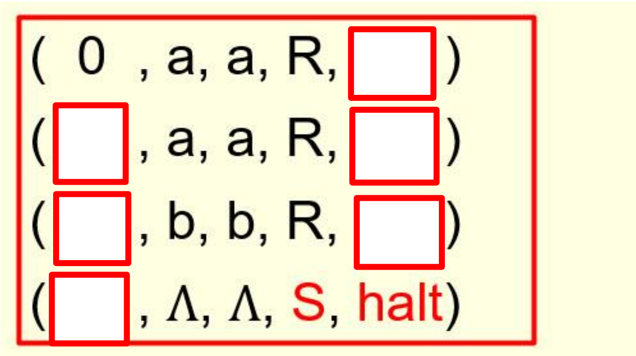
6. (4 points)

Fill out the following blanks for the instructions of a Turing machine that accepts the language $\{ab^n \mid n \in N\}$. Use smallest possible non-negative integers to represent the states of the TM.

(0, a, a, R, ___)
(___, b, b, R, ___)
(___, Λ, Λ, S, halt)

7. (6 points)

Fill out the following blanks for the instructions of a Turing machine that accepts the language $\{aab^n \mid n \in N\}$. Use smallest possible non-negative integers to represent the states of the TM.



- Solutions must be typed (word processed) and submitted both as a pdf file and a word file to Canvas before 23:59 on 04/05/2025.
- Don’t forget to name your files as
  CS375_2025s_HW7_LastName.docx / CS375_2025s_HW7_LastName.pdf