Week 4 Questions - Theory of Computation

February 10, 2024

1. (1 point) The runtime of the CYK algorithm is $\mathcal{O}(n^k)$, where n is the length of the input string and k is ___**3**___.

2. (1 point) The CYK algorithm is an example of which of the following?

   A. Greedy algorithm

   B. Divide and Conquer

   C. Dynamic Programming

   D. Brute force algorithm

3. (1 point) Time complexity of the CYK algorithm with respect to the size (no. of rules) of the grammar $|G|$ is?

   A. Linear

   B. Quadratic

   C. Cubic

   D. Logarithmic

4. (1 point) If L_1 and L_2 are two context-free languages, then $L_1 \cap L_2$ is:

   A. Necessarily context-free

   B. May or may not be context-free

   C. Never context-free

5. (2 points) Context-free languages are closed under which of the following operations? Select all the correct options.

   ■ **Union**

   ■ **Kleene Closure**

   ■ **Concatenation**

   ☐ Complement

   ☐ Intersection

6. (2 points) Consider the following CFG in Chomsky Normal Form:

$$S_0 \rightarrow S \mid \varepsilon \tag{1}$$
$$S \rightarrow CD \mid CB \mid AD \mid AB \mid DC \mid DA \mid BC \mid BA \tag{2}$$
$$C \rightarrow SA \tag{3}$$
$$D \rightarrow SB \tag{4}$$
$$A \rightarrow a \tag{5}$$
$$B \rightarrow b \tag{6}$$

   Using the CYK algorithm, determine which of the following strings are generated by the above grammar. Out of the given options, select all the strings that are generated.

- ■ **abbaba**
- ■ **ababab**
- ☐ abbbb
- ☐ bbbaa

> **Solution:** The given CNF grammar is derived from the following CFG:
>
> $$S \rightarrow SaSb \mid SbSa \mid \varepsilon \tag{7}$$
>
> Which is the set of strings with an equal number of as and bs.

7. (2 points) Which of the following grammars is equivalent to the following grammar?

$$S \rightarrow AB$$
$$A \rightarrow aA \mid \varepsilon$$
$$B \rightarrow bB \mid \varepsilon$$

Note that two grammars are equivalent if they recognize the same language.

- ■ $S \rightarrow aS \mid B$
  $B \rightarrow bB \mid \varepsilon$
- ☐ $S \rightarrow aSb \mid A$
  $A \rightarrow aA \mid \varepsilon$
- ☐ $S \rightarrow aSb \mid A \mid B$
  $A \rightarrow aA \mid a$
  $B \rightarrow bB \mid b$
- ■ $S \rightarrow ASB \mid \varepsilon$
  $A \rightarrow aA \mid \varepsilon$
  $B \rightarrow bB \mid \varepsilon$

> **Solution:** The language generated by the grammar in the question is a^*b^*. The second option generates the language $a^m b^n$, where $m \geq n$. For instance, it cannot generate the string b. The third option does not generate the empty string.

8. (1 point) Which of the below options is a description of the functioning of the CYK algorithm?

   A. The algorithm determines the set of variables that can generate substrings of the input string.

   B. The algorithm generates all the strings that can be generated from the start variable, and matches the input string with the list generated.

Paragraph for Q9 - Q12

Recall the condition under which PDAs accept strings:

A PDA $M = (M, \Sigma, \Gamma, \delta, q_0, F)$ accepts an input string w if $w = w_1 w_2 \cdots w_m$, where $w_i \in \Sigma_\varepsilon$, and sequences $r_0, r_1, \cdots r_m \in Q$ and $s_0, s_1, \cdots s_m \in \Gamma^*$ exist such that:

1. $r_0 = q_0$ and $s_0 = \varepsilon$.

2. For $i = 0, 1, \cdots m - 1$ we have

$$(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$$
$$s_i = at$$
$$s_{i+1} = bt$$

for some $a, b \in \Gamma_\varepsilon$ and $t \in \Sigma^*$.

3. $r_m \in F$

We define a *Push Through Automaton* (PTA) as a PDA with a queue instead of a stack, and a Push Through Language (PTL) as any language that can be recognized by a PTA. That is, instead of pushing and popping values from the front of the stack, values are pushed from the front and popped from the back. PTAs can be formally defined using the following modified version of Condition 2 above:

2. For $i = 0, 1, \cdots m - 1$ we have

$$(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$$
$$s_i = at$$
$$s_{i+1} = \boldsymbol{tb}$$

for some $a, b \in \Gamma_\varepsilon$ and $t \in \Sigma^*$.

We also define a *2-stack PDA*, as one may assume, as a PDA with two stacks. δ now becomes

$$\delta \colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon^2 \to \mathcal{P}(Q \times \Gamma_\varepsilon^2)$$

As the automaton must now read and write one value from each stack. The definition of acceptance can also be modified appropriately.

9. (2 points) Which of the following languages are PTLs? Select all that apply.
   - ■ $\{w^2 \mid w \in \Sigma^*\}$
   - ■ $\{a^{2^n} \mid n \in \mathbb{N}\}$
   - ■ $\{w^3 \mid w \in \Sigma^*\}$
   - ■ $\{w_1 w_2^2 \mid w_1, w_2 \in \Sigma^*\}$

10. (2 points) Which of the following statements is/are correct? Select all that apply.
   - ■ **All CFLs are PTLs**
   - ☐ All PTLs are CFLs
   - ☐ There are CFLs which are not PTLs
   - ■ **There are PTLs which are not CFLs**

Solution: The first option in Q9 is an example of a language that is a PTL but not CFL. To see that all CFLs are PTLs, we can simulate a stack with a queue, by cycling through the entire queue (popping a value and pushing it back) each time we want to push/pop something.

11. (2 points) Which of the following languages can be recognized by a 2-stack PDA? Select all the correct options.
   - ■ $\{a^n b^n c^n \mid n \in \mathbb{N}\}$
   - ■ $\{a^n b^m c^m d^n \mid n, m \in \mathbb{N}\}$

- ■ $\{a^{3n}b^{4n}c^{5n} \mid n \in \mathbb{N}\}$
- ■ $\{a^n b^m c^n d^m \mid m, n \in \mathbb{N}\}$

12. (2 points) Which of the following automatons can be used to recognize a PTL? Select all the correct options.

- ■ **A 2-stack PDA**
- ■ **A 2-queue PTA (combining both definitions above)**
- ☐ A standard PDA with one stack
- ☐ A NFA