CS 456/656 Answers to Study Guide for Examination November 20, 2024

1. True or False.

   Answer the True/False questions given in the file "Tests/tfstd.pdf."

2. Prove that $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$ for every positive integer n. (Hint: The binomial theorem states that $(a+b)^2 = a^2 + 2ab + b^2$, and $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$.)

   Write $F(n) = \frac{n^2(n+1)^2}{4}$. The proof is by induction on n. The formula cleary works for $n = 1$, since $\sum_{i=1}^{1} i^3$ and $F(1) = \frac{1^2(1+1)^2}{4}$ are both equal to 1. Suppose the formula holds for some n. We need to prove that it holds for $n+1$.

   Proof:

$$\begin{aligned}
\sum_{i=1}^{n+1} i^3 &= \sum_{i=1}^{n} i^3 + (n+1)^3 \\
&= F(n) + (n+1)^3 \text{ by the inductive hypothesis} \\
&= \frac{n^2(n+1)^2}{4} + (n+1)^3 \\
&= \frac{n^4 + 2n^3 + n^2}{4} + n^3 + 3n^2 + 3n + 1 \\
&= \frac{n^4 + 2n^3 + n^2}{4} + \frac{4n^3 + 12n^2 + 12n + 4}{4} \\
&= \frac{n^4 + 6n^3 + 13n^2 + 12n + 4}{4} \\
&= \frac{(n^2 + 2n + 1)(n^2 + 4n + 4)}{4} \\
&= \frac{(n+1)^2(n+2)^2}{4} \\
&= F(n+1)
\end{aligned}$$

   ∎

3. State the pumping lemma for context-free languages.

   For any context-free language L
   There exists a positive number p such that
   For any $w \in L$ of length at least p
   There exist strings u, v, x, y, z such that the following hold:
   1. $w = uvxyz$
   2. $|vxy| \leq p$
   3. v and y are not both empty
   4. For any $i \geq 0$, $uv^ixy^iz \in L$

4. Give an example of a CFL that is not regular.

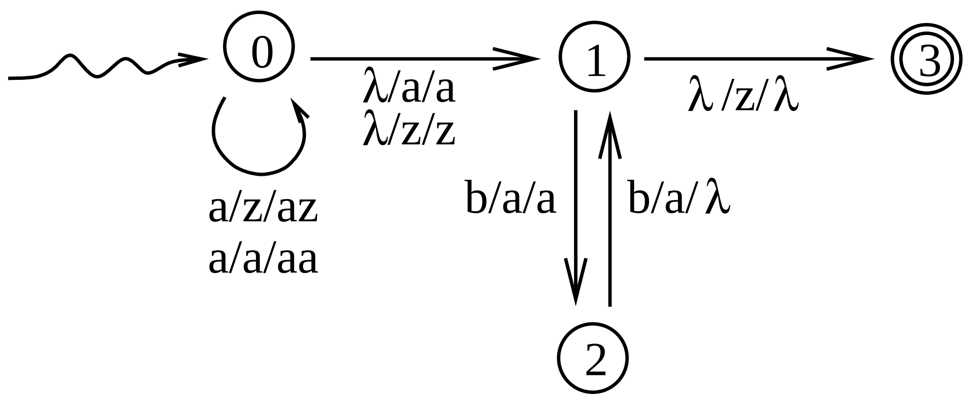
There are many examples. The simplest example is $\{a^n b^n : n \geq 0\}$.

5. Give an example of a CFL that is not a DCFL.

   The simplest example is the palindrome language generated by the CFG
   $S \to aSa$
   $S \to bSb$
   $S \to \lambda$

6. Sketch the diagram of a DPDA which accepts the language of all strings over $\{a, b\}$ which have twice as many a's as b's.

   This problem will not be on the exam; I'll show you the answer after the exam.. Here is the diagram of a DPDA for a simpler context-free language $\left\{a^n b^{2n} : n \geq 0\right\}$, namely twice as many b's as a's, but in alphabetical order.



7. Read and understand `Handout/complexityIII.pdf`

8. Prove that every language L which is enumerated in canonical order by some machine is decidable.

   The following program decides L. Let $w_1, w_2, \ldots$ be the canonical order of L, which we can use as an input to our program.

   Read w
   For all w_i in canonical order
   If $w = w_i$ halt and accept
   If $w < w_i$ (in the canonical order) halt and reject

9. Prove that every decidable language L is enumerated in canonical order by some machine.

   Let Σ be the alphabet of L. Let $w_1, w_2, \ldots$ be the canonical order enumeration of Σ^*. The following program enumerates L in canonical order.

   For i from 1 to ∞
   if $w_i \in L$ write w_i

10. Prove that every recursively enumerable language L is accepted by some machine.

Let $w_1, w_2 \ldots$ be an enumeration of L written by some machine. The following program accepts L.

Readw For i from 1 to ∞
If $w = w_i$ halt and accept

11. Prove that every language L accepted by a machine M is recursively enumerable.

    Let Σ be the alphabet of L. Let $w_1, w_2, \ldots$ be the canonical order enumeration of Σ^*. The following program enumerates L.

    For integers t from 1 to ∞
    For integers i from 1 to t
    If M accepts w_i within t steps, write w_i.

12. Write a (pseudo-code) program which accepts HALT. (Hint: You can write it in no more the four lines.)

    Read $\langle M \rangle w$ (an instance of the halting problem)
    Emulate M with input w. If it halts, accept $\langle M \rangle w$.

13. Prove that HALT is undecidable.

    By constradiction. Assume HALT is decidable. Let P be the following program

    Read a machine description $\langle M \rangle$.
    If M halts with input $\langle M \rangle$, enter an infinite loop.
    Else halt.

    The input to P could be $\langle P \rangle$. If so, does it halt?

    If P halts with input $\langle P \rangle$, then, with input $\langle P \rangle$, the code will cause P to enter an infinite loop, that is, not halt, contradiction.

    Alternatively, if P does not halt with input $\langle P \rangle$, then, with intput $\langle P \rangle$, the code will cause P to halt, contradiction.

    In either case, we have a contradiction. We conclude that HALT is not decidable.

14. The LALR handout.

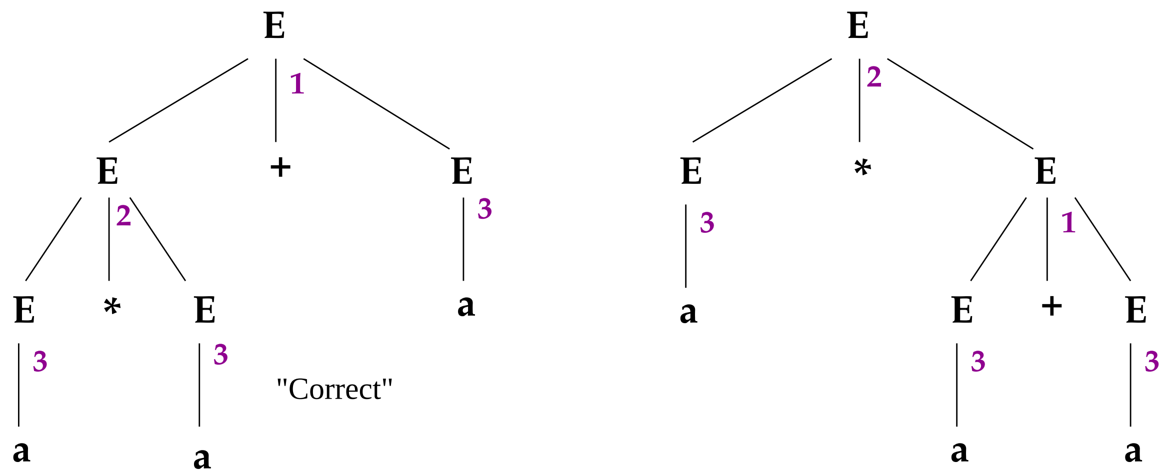
    (a) Read Handout/lalrhandout1.

    (b) That handout contains 8 questions. Answers to questions 1, 2, 3, and 6. are given in the handout. Understand those questions and answers.

    (c) Work questions 4, 5, 7, and 8.

    The answers to the questions are found in Handouts/lalrhandout1ans.pdf.

    (d) Using the grammar on page 1 of the handout, give two *different* parse trees of the string `a*a+a`, showing that the grammar is ambiguous. Which one of those parse trees is "correct," *i.e.*, respects the standard precedence of operators?


E
1
E + E
2
3
E * E
a
3
3
a a
"Correct"
E
2
E * E
3
1
a
E + E
3
3
a a