CS2205: Theory of Computation
Problem Class IV

1. For strings u and v, give a formula for computing $|uv|$, the length of string u concatenated with string v.
2. Let $\Sigma =_{\text{def}} \{a, b\}$. How many strings of length 2 does Σ^* contain? of length 3? of length n?
3. If Σ is an alphabet, let Σ^n be all strings over Σ of length n. Suppose $|\Sigma| = m$. What is $|\Sigma^n|$?
4. Hence, how many strings of length *at most* n can we form from alphabet Σ?
5. Let $L =_{\text{def}} \emptyset$. Evaluate L^*
6. Let $L =_{\text{def}} \{\epsilon\}$. Evaluate L^*
7. For each of the following languages, write a paraphrase in English that conveys what strings the languages contain.
   (a) $L_1 =_{\text{def}} \{a^n b^n \mid n \in \mathbb{N}\}$
   (b) $L_2 =_{\text{def}} \{w \in \Sigma^* \mid w = w^R\}$
8. Dr. Toc does research into the Theory of Computer Science. The good doctor is brilliant but erratic — known as much for howling errors as fabulous insights.

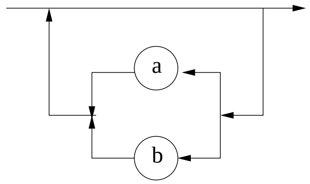
   This is what the good doctor has been saying recently: "When we concatenate two languages, L_1 and L_2, we pair every string from L_1 with every string from L_2. This is like computing the Cartesian product of two sets. From this, I conclude that $|L_1 L_2| = |L_1| \times |L_2|$." Give a counter-example to show that Dr. Toc is wrong.
9. Whilst Dr. Toc often gets things right, it seems the good doctor's intuitions about strings continue to be way off-the-mark. Mathematically sophisticated cleaning staff were appalled to see the following on a blackboard after one of Dr. Toc's lectures:

   $$L^+ = L^* \setminus \{\epsilon\}$$

   This isn't always true! Under what circumstances is it true?
10. At the end of this sheet are some syntax diagrams that define the set of URLs. The start symbol is *URL*. (Note that these are not exactly the rules for real URLs. They are a simplified set of rules. Make sure you answer this question using these syntax diagrams, not your opinions about what really constitutes a URL.)

    Which of the following are well-formed URLs according to the syntax diagrams?

    (a) http://www.cs.ucc.ie/dgb
    (b) www.cs.ucc.ie/dgb
    (c) http://www/dgb
    (d) http://www.cs.ucc.ie/dgb/toc
    (e) http://www.cs.ucc.ie/
    (f) http://www.cs.ucc.ie/dgb/mypage.html
    (g) http://w/d
    (h) http://www.cs.ucc.ie/dgb/
    (i) http://dgb.mypage.html/www/cs/ucc/ie
    (j) ϵ
11. Write down all well-formed expressions of length less than 4 that can be generated from the following syntax diagram:



12. Write a syntax diagram to define the language $\{a^n b^n \mid n \in \mathbb{N}\}$. (Hint: your diagram will be recursive.)
13. Write a BNF grammar to define the language $\{a^n b^n \mid n \in \mathbb{N}\}$.
14. Consider this BNF grammar whose start symbol is ⟨S⟩:

    ⟨S⟩ ::= a⟨X⟩
    ⟨X⟩ ::= a⟨X⟩
    ⟨X⟩ ::= b⟨X⟩
    ⟨X⟩ ::= ϵ

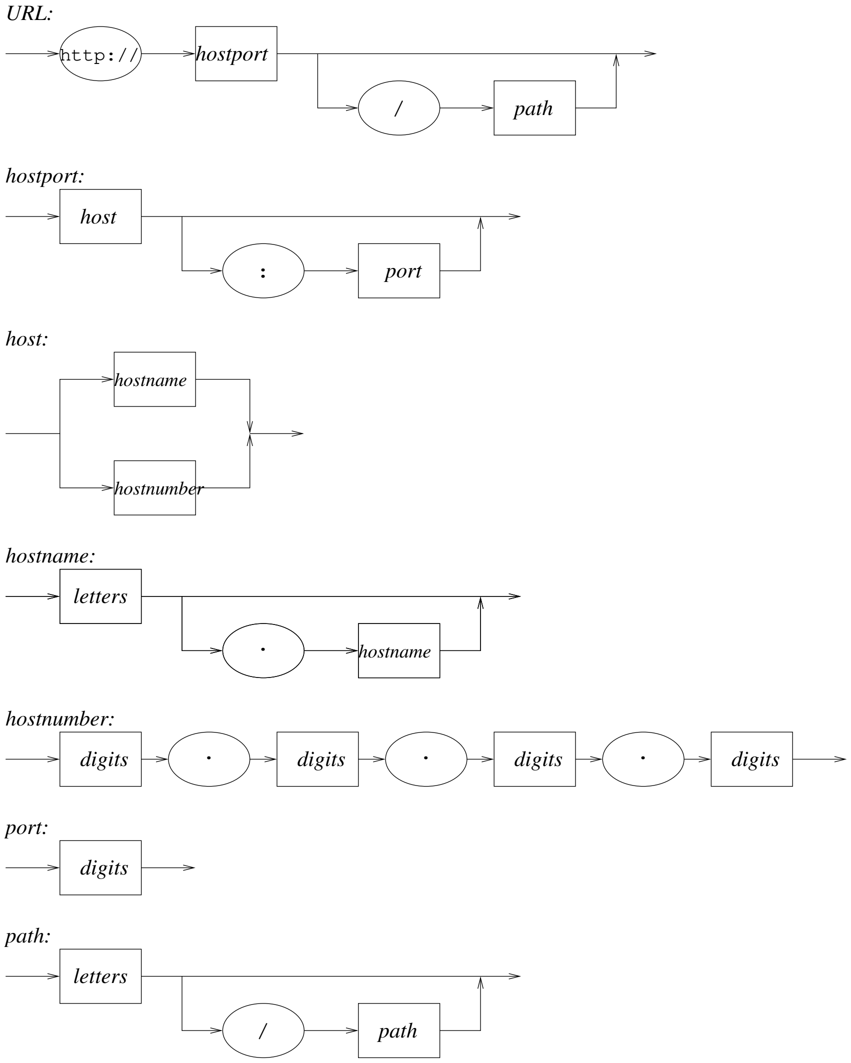
    (a) Show that *aba* is a member of the language defined by this grammar. (In other words, give a derivation.)
    (b) Draw a parse tree for the string *aabb*.
    (c) Describe in words the language defined by this grammar.
15. Consider this BNF grammar whose start symbol is ⟨S⟩:

    ⟨S⟩ ::= ⟨X⟩b⟨X⟩
    ⟨S⟩ ::= a⟨X⟩
    ⟨X⟩ ::= ⟨X⟩a
    ⟨X⟩ ::= ⟨X⟩b
    ⟨X⟩ ::= ϵ

(a) Describe in words the language defined by this grammar.

(b) Show that the grammar is ambiguous.

Figure 1: Syntax Diagrams for Question 10



letters: one or more letters (a−z, A−Z)

digits: one or more digits (0−9)