Languages and Group Theory, CSC-161, Dr. Ostheimer

For this group work, we will let $X = \{a, b, A, B\}$.

(1) Let us first consider the language $L_1 = X^*$.
   (a) **True or False:**
      (i) $X \subseteq L_1$;
      (ii) $L_1 \subseteq X$;
      (iii) L_1 is closed under concatenation. (A language L over alphabet X is *closed under concatenation* if for all $u, v \in L$, the concatenation uv is also an element of L.)
   (b) Enumerate all the words in L_1 of length 2 or less.
   (c) How many words are in L_1?

(2) We will now define certain "rewriting rules" :

$$\begin{aligned} aA &\rightarrow \Lambda \\ Aa &\rightarrow \Lambda \\ bB &\rightarrow \Lambda \\ Bb &\rightarrow \Lambda \end{aligned}$$

Let w_1 and w_2 be words in L_1. We say that w_1 and w_2 are *equivalent* if w_1 can be obtained from w_2 by using the rewriting rules (as many times as you want) in either direction. We call the set of words which are equivalent to w_1 the *equivalence class* of w_1. For example $baABa$ is equivalent to $aAbBa$ since

$$baABa \rightarrow bBa \rightarrow a \rightarrow bBa \rightarrow aAbBa.$$

Furthermore, $baABa$, bBa, a, bBa and $aAbBa$ are all in the equivalence call of $baABa$.

Let L_2 be the set of words in L_1 that are equivalent to Λ.

   (a) Find 5 words in L_2. Try to pick words that illustrate the range of different kinds of words that are in L_2.
   (b) **True or False:**
      (i) $X \subseteq L_2$;
      (ii) $L_1 \subseteq L_2$;
      (iii) $L_2 \subseteq L_1$;
      (iv) L_2 is closed under concatenation.
   (c) Describe an algorithm for deciding whether a given word in X^* is in L_2.
   (d) Analyze the time complexity of your algorithm. In other words, find a big-O estimate for a function $f(n)$ that is the maximum number of steps needed to complete your algorithm,

1. Optional Challenge Problems

For these problems, it would probably be very beneficial to team up with another student.

(1) This is a theoretical problem from discrete mathematics about *equivalence*.
   (a) Write down the definition of *equivalence relation* from your CS24 text. Be sure to also write down the definitions of any terms used therein.
   (b) Convince yourself that in the example on the previous page, the use of the term *equivalent* conforms to the definition above.

(2) This is a problem from my own area of research which is on the boundary of theoretical computer science and group theory, a branch of theoretical computer science which is introduced in Math 145. We consider the more general case in which we are given a finite set of rewriting rules that *includes* the rewriting rules given above, but which *may not be* limited to those.
   (a) Consider three examples of rewriting rules:
      - the example considered above;
      - the example obtained by adding the relation $ab \to ba$;
      - the example obtained by adding, furthermore, the relations $a^3 \to \Lambda$ and $b^2 \to \Lambda$.

      For each example, answer the following questions:
      (i) How many equivalence classes are there?
      (ii) Describe a set of lexicographically shortest representatives for each equivalence class.
      (iii) Consider the language of words which are equivalent to Λ. Describe an algorithm for deciding whether a given word in X^* is in this language.
      (iv) Analyze the time complexity of your algorithm.