MODELOS DE COMPUTACION I

RELACION DE PROBLEMAS 4.

1. Determinar si la siguiente gramática es ambigua y si el lenguaje generado es inherentemente ambiguo:

   $S \to A_1, A_2$

   $A_1 \to aA_1b, aA_1, \epsilon$

   $A_2 \to aA_2b, A_2b, \epsilon$

2. Sea la gramática

$$S \to aSA, \quad S \to \epsilon, \quad A \to bA, \; z \;\; A \to \epsilon$$

   a) Demostrar que es ambigua

   b) Dar una expresión regular para el lenguaje generado.

   c) Construir una gramática no ambigua que genere el mismo lenguaje

3. Describe el lenguaje generado por la siguiente gramática $G = (\{S, A\}, \{a, b\}, P, S)$, con

$$P = \{S \to aAa, S \to bAa, A \to aAa, A \to bAa, A \to \epsilon\}$$

   - Demuestra que el lenguaje generado por la gramática no es regular, pero si independiente del contexto,
   - Normaliza la gramática G en la Forma Normal de Greibach, y determina todas la derivaciones más a la izquierda para la cadena ab^2a^5.

4. Obtener la forma normal de Greibach para la siguiente gramática:

$$< \{S_1, S_2, S_3\}, \{a, b, c, d, e\}, S_1, P >$$

   donde

$$P = \{S_1 \to S_1S_2c, S_3, S_3bS_3; S_2 \to S_1S_1, d; S_3 \to S_2e\}$$

5. Considera la gramática $G = (V, T, S, P)$ donde

$$V = \{< expresion >, < identificador >\}, T = \{a, b, c, d, -\}, S =< expresion >$$

   y P contiene las producciones:

$$< expresion > \to < identificador >$$

$$< expresion > \to < identificador > - < expresion >$$

$$< expresion > \to < expresion > - < identificador >$$

$$< identificador > \to a, b, c, d$$

- demuestra que esta gramática no puede ser empleada para describir un posible lenguaje de programación, teniendo en cuenta que que la sustración no es una operación conmutativa, y que $(a-b)-d \neq a-(b-d)$,
- ¿es ambígua la gramática G? >es la ambiguedad inherente al lenguaje generado por G ? Justifica adecuadamente la respuesta.
- ¿es posible modificar G de manera que la nueva gramática pueda ser usada para generar el lenguaje de las expresiones aritméticas correctas con el operador de resta

6. Dada la gramática

$$S \to A, \quad S \to B, \quad A \to aaA, \quad A \to \epsilon$$

$$B \to aaaB, \quad B \to \epsilon$$

- Demostrar que es ambigua
- Construir un autómata finito determinístico que acepte el mismo lenguaje
- Construir una gramática lineal pgb la derecha, a partir del autómata determinístico, que genere el mismo lenguaje,
- Demostrar que la gramática resultante no es ambigua.

7. Dar una gramática libre de contexto no ambigua que genere el lenguaje $L = \{a^i b^j a^k b^l \ : \ (i=j) \vee (k=l)\}$.

8. Determinar cuales de las siguientes gramáticas son ambiguas y, en su caso, comprobar si los lenguajes generados son inherentemente ambiguos:

   a) $S \to aSb|Sb|aS|a$

   b) $S \to aaS|aaaS|a$

   c) $S \to aS|aSb|X$
   $X \to Xa|a$

9. Dar gramáticas libres de contexto o regulares (cuando sea posible) para los siguientes lenguajes sobre el alfabeto $A = \{a, b, c\}$:

   a) $L_1 = \{a^i b^j c^k \ : \ i \neq j \vee j \neq k\}$

   b) $L_2 = \{(ab)^i (bc)^j \ : \ i, j \geq 0\}$

   c) $L_3 = \{a^i b^{i+j} c^j \ : \ i, j \geq 0\}$

   d) L_4 definido como el conjunto de palabras que comienzan por aab y terminan por bbc y tales que estas dos subcadenas no aparecen nunca en el interior de la palabra (sólo están al principio y al final).

10. Pasar a forma normal de Greibach la gramática

$$
\begin{array}{ll}
S \to AAA, & S \to B \\
A \to aA, & A \to B \\
B \to \epsilon &
\end{array}
$$

11. Dada la gramática:

$$S \to 01S, \quad S \to 010S, \quad S \to 101S, \quad S \to \epsilon,$$

determinar si es ambigua.

Construir un autómata finito determinista asociado y calcular la gramática lineal por la derecha que se obiene a partir del autómata. ¿Es ambigua la gramática resultante?

12. Demostrar que la gramática: $S \to A1B, \quad A \to 0A|\epsilon, \quad B \to 0B|1B|\epsilon$ no es ambigua.

Encontrar una gramática para el mismo lenguaje que sea ambigua y demostrar su ambigüedad.