1. (a) Eliminando símbolos inútiles, y luego reglas ϵ se obtiene la gramática

$$\begin{array}{lcl} S & \rightarrow & B \mid C \\ B & \rightarrow & a \mid Bb \mid Ca \\ C & \rightarrow & b \mid Ba \end{array}$$

que presenta recursión indirecta (B, C).

La elección del orden de auxiliares S, C, B y la aplicación del algoritmo de eliminación de recursión por la izquierda permite obtener

$$\begin{array}{lcl} S & \rightarrow & B \mid C \\ B & \rightarrow & aB' \mid baB' \\ B' & \rightarrow & bB' \mid aaB' \mid \epsilon \\ C & \rightarrow & b \mid Ba \end{array}$$

que ya no presenta recursión por la izquierda.

(b) Esta gramática no es $LL(1)$, puesto que al menos en la casilla de cruce de C con b aparecerán las reglas $C \rightarrow b$ y $C \rightarrow Ba$, puesto que $b \in PRIMEROS(b)$

(c) De la gramática obtenida puede deducirse que a partir de B' puede generarse única y exclusivamente $\alpha = (b \mid aa)^*$, y por lo tanto, desde B se podrá generar exactamente $(a \mid ba)\alpha$, y desde C, $b \mid (a \mid ba)\alpha$. Por lo tanto el lenguaje generado es $a\alpha(a \mid \epsilon) \mid ba\alpha(a \mid \epsilon)$ A partir de ahi puede calcularse el RFD.

También puede obtenerse un RFN directamente a partir de la gramática equivlente regular por la derecha que se obtiene eliminando reglas simples para S.

El autómata mínimo resultante es

A	B	C
(B)	C	B
(C)	B	D
D	D	D

2. La única derivación más a la derecha posible (para las tres formas del enunciado) es

$$S \Rightarrow AB \Rightarrow ABa \Rightarrow Aba \Rightarrow AbBba \Rightarrow AbBaba \Rightarrow Abbaba \Rightarrow abbaba$$

Por lo tanto, los pivotes son, en el primer caso, la regla $B \rightarrow Ba$ con el consecuente en la segunda posición, en el segundo la regla $B \rightarrow b$ para la b de la tercera posición y la regla $A \rightarrow a$ para la a inicial en el tercer caso.

3. Basta hacer una máquina de Turing que haga lo siguiente

   si la cadena empieza por un $\vert$, lo borre y vaya a la derecha, pasando sobre los $\vert$, hasta reescribirlo en sustitución del signo $+$

   si la cadena empieza por $+$, sencillamente lo borre

   en ambos casos, continúe hacia la derecha hasta encontrar el símbolo $=$, para borrarlo

   y finalmente comprueba que no hay nada más detrás.

   Por ejemplo

		$+$	$=$	b
q_1	$q2, b, \to$	$q_3, b, \to$		
q_2	$q_2 \vert, \to$	$q_3, \vert, \to$		
q_3	$q_3, \vert, \to$		$q_4, b, \to$	
q_4				$q_5, b, \to$
(q_5)				

4. El lenguaje es independiente de contexto no regular. Es generable por la gramática $S \to 1S1 \mid R, R \to +P, P \to 1P1 \mid =$ y su intersección con $1^* + 1 = 1^*$ es $\{1^p + 1 = 1^{p+1}\}$ que claramente no verifica el lema de bombeo de regulares. Por lo tanto el algoritmo más adecuado es un autómata a pila.