Ejercicios tipo de aspectos sintácticos.

Preguntas sobre TRADUCTORES ASCENDENTES Y TRADUCTORES DESCENDENTES.

o Explica cuál es el fundamento de los analizadores LR.

G \in LR(k) \rightarrow para todo α w, α w' formas de frase con el mismo prefijo viable α α es un prefijo viable completo que incluye un asidero

```
k - prefijo(w) = k - prefijo(w') -> asidero(\alpha w) = asidero(\alpha w')
```

Es decir, una gramática es LR(k) si el hecho de que sean iguales un número fijo,k, de los primeros terminales de sus formas de frase derechas (con el mismo prefijo viable completo α) implica que el asidero de estas formas de frase derechas se pueda determinar de forma unívoca.

Con esta definición bastaría si sólo preguntas teoría, pero en el caso de que pregunten un caso práctico, vamos a ver cómo lo resolvemos.

Necesitamos conocer las siguientes definiciones (solo las necesitamos para comprender el ejercicio):

- En un árbol de análisis sintáctico el <u>asidero</u> es el conjunto de hojas situado más a la izquierda que constituyen el cuerpo de una producción.
- El <u>asidero</u> se corresponde con el cuerpo de la última producción aplicada en la derivación más a la derecha de la frontera del árbol.
- **Prefijos viables:** es el conjunto de prefijos de las fronteras de los árboles sintácticos que no rebasan los asideros y forman un conjunto regular.

```
E -> E + T -> E + T + T -> E + T + T * F El asidero en este caso es E + T.

Gramática: S \rightarrow aAc; A \rightarrow bAb; A \rightarrow b.
```

Para demostrar que no es ninguno de estos casos basta con encontrar dos formas de frase derechas con el mismo prefijo α completo y que tomando k=0, k=1,k=2 símbolos de preanálisis no determinen de forma unívoca su asidero.

Por ejemplo, no es LR(1):

¿es LR(0)? ¿es LR(1)? ¿es LR(2)?

```
\alpha w = abbbc con \alpha = abb y w = bc. \alpha w' = abbbbbc con \alpha = abb y w' = bbbc.
```

Como k = 1 -> pri(w) = pri(w') = b pero asidero(α w) = (3, A \rightarrow b) y asidero (α w') = (4, A \rightarrow b).

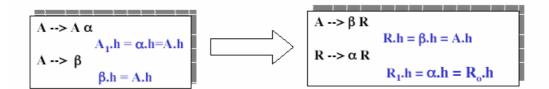
Explica qué características distinguen los analizadores SLR, LR(1), LALR.

Preguntas sobre la RECURSIÓN IZQUIERDA.

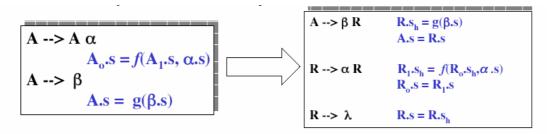
Piden lo siguiente:

- Explicar el algoritmo para eliminar la recursión a izquierdas directa con ayuda de un árbol.
- Aplicar el algoritmo anterior a un caso particular.

Esquema para eliminar la recursión izquierda (atributos heredados).



Esquema para eliminar la recursión izquierda (atributos sintetizados).



Explicación de eliminación de recursión izquierda CON ÁRBOLES.

