# **Condiciones Descendente**

¿Cuáles de las siguientes gramáticas se sabe con certeza que **no** son válidas para construir un Analizador Sintáctico Descendente?

1: Una gramática sin factorizar

  Esta respuesta es correcta.

Para que una gramática se pueda utilizar en un Analizador Sintáctico Descendente Predictivo tiene que estar factorizada

2: Una gramática ambigua

Explicación:

Las gramáticas ambiguas no pueden utilizarse para construir Analizadores Sintácticos

1: Todas las reglas del axioma de la gramática estarán colocadas en la fila del axioma

1: Una gramática que tenga las reglas A → a b y A → a b c

**1:** Una gramática que tenga una regla A → A b c

# **Condición LL(1)**

Dada la siguiente gramática, ¿qué afirmaciones son correctas?

S → 0 A A 0 | 1 A 1 | λ

A → B | C

B → 0 1 | 1 0

C → 0 0 | 1 1

1: La gramática no es válida para construir un Analizador Sintáctico Descendente Recursivo

1: Las reglas de C sí cumplen la condición LL(1)

  Esta respuesta es correcta.

Explicación:

El FIRST(00)={0} y el FIRST(11)={1}, por lo que la intersección de ambos conjuntos es vacía

2: La gramática no es válida para construir un Analizador Sintáctico Descendente Recursivo

Explicación:

-La gramática no es LL(1) por las reglas de A, por lo que no se puede construir un Analizador Sintáctico Descendente

-Las reglas de A no cumplen la condición LL(1) porque las dos reglas pueden derivar en cadenas que empiezan por 1 o por 0.

1: La gramática no es válida para construir un Analizador Sintáctico Descendente con tablas

  Esta respuesta es correcta.

Explicación:

La gramática no es LL(1), por lo que no se puede construir un Analizador Sintáctico Descendente con tablas

2: Las reglas de C sí cumplen la condición LL(1)

  Esta respuesta es correcta.

Explicación:

El FIRST(3)={3} y el FIRST(2B)={2}, por lo que la intersección de ambos conjuntos es vacía

3: Para aplicar la condición LL(1) a las reglas de A hay que comprobar que los First no tienen elementos en común considerando 2 de las reglas de cada vez, y también que el Follow (A) no tiene elementos en común con el First de ninguna de las dos primeras reglas

  Esta respuesta es correcta.

Explicación:

No es suficiente comprobar que no hay ningún elemento común a todas las reglas, sino que hay que comprobarlo 2 a 2 con todas las posibles parejas de reglas

**2:** La gramática no es LL(1)

3:Las reglas de A no cumplen la condición LL(1)

# **Condición LL(1) - 2**

Dada la siguiente gramática, ¿qué afirmaciones son correctas?

S → 0 A B 0 | 1 A 1 | λ

A → 0 C | 1 C

B → 0 1 | 1 0

C → 0 | 1

1: Las reglas de A sí cumplen la condición LL(1)

Explicación:

El FIRST(01)={0} y el FIRST(10)={1}, por lo que la intersección de estos conjuntos no es vacía

2: La gramática es válida para construir un Analizador Sintáctico Descendente con tablas

Explicación:

La gramática es LL(1), por lo que se puede construir un Analizador Sintáctico Descendente con tablas

1: Las reglas de B sí cumplen la condición LL(1)

2: Las reglas de S sí cumplen la condición LL(1)

1: La gramática es LL(1)

3: Las reglas de A sí cumplen la condición LL(1) porque el FIRST de una regla es {0} y el FIRST de la otra regla es {1}

**1:** Las reglas de C sí cumplen la condición LL(1)

# **Condición LL(1) - 3**

Dada la siguiente gramática, ¿cuáles de las afirmaciones son correctas?

S → A B | B C

A → 1 A | 2 A | λ

B → 3 C 4 | λ

C → 3 | 2 B

Respuestas seleccionadas:

Las reglas de A sí cumplen la condición LL(1)

1: Las reglas de C sí cumplen la condición LL(1)

Explicación:

El FIRST(3)={3} y el FIRST(2B)={2}, por lo que la intersección de ambos conjuntos es vacía

2: Para aplicar la condición LL(1) a las reglas de A hay que comprobar que los First no tienen elementos en común considerando 2 de las reglas de cada vez, y también que el Follow (A) no tiene elementos en común con el First de ninguna de las dos primeras reglas

Explicación:

No es suficiente comprobar que no hay ningún elemento común a todas las reglas, sino que hay que comprobarlo 2 a 2 con todas las posibles parejas de reglas

3: La gramática no es válida para construir un Analizador Sintáctico Descendente con tablas

  Esta respuesta es correcta.

Explicación:

La gramática no es LL(1), por lo que no se puede construir un Analizador Sintáctico Descendente con tablas

4: Esta gramática no cumple la condición LL(1) porque la regla B→3C4 se aplicaría cuando el siguiente token es “3” y la regla B→λ se aplicaría para los elemento del Follow(B) que contiene el “3”

  Esta respuesta es correcta.

**Explicación:**

Este caso incumple la condición LL(1)

# **Analizador Descendente con tablas**

En relación con el Analizador Sintáctico Descendente por Tablas (LL(1)), ¿cuáles de las siguientes respuestas son correctas?

1: Cada celda de la tabla puede contener una regla o estar vacía

  Esta respuesta es correcta.

Explicación:

Si hay una regla indica que el reconocimiento puede proseguir y si está vacía indica un error en el reconocimiento

2: Una misma regla puede aparecer únicamente en una fila de la tabla, y podría estar varias veces

  Esta respuesta es correcta.

Explicación:

Cada regla está en la fila de su no terminal y en las columnas correspondientes a su FIRST

3: La columna $ puede estar vacía

4: La tabla no puede tener una fila vacía

# **Follow 1**

Indica cuáles de los conjuntos Follow son correctos para la siguiente Gramática:

S ⟶ e A B g | C D | λ

A ⟶ A a | a

B ⟶ b B | λ

C ⟶ E F D

D ⟶ k D | k

E ⟶ E g | h | λ

F ⟶ i j F | λ

**1:** FOLLOW(B) = { g }

2: FOLLOW (E) = { i, k, g }

1: FOLLOW (S) = { $ }

2: FOLLOW (C) = { k }

3: FOLLOW (F)= { k }

**2:** FOLLOW (A) = { b, g, a }

FOLLOW (D) = { $, k }

# **Follow 2**

Indica cuáles de los conjuntos Follow son correctos para la siguiente Gramática:

S ⟶ e A B g | C B

A ⟶ A a | a

B ⟶ b B | λ

C ⟶ E F D

D ⟶ k D | λ

E ⟶ E g | h | λ

F ⟶ i j F | λ

**1:** FOLLOW (C) = { b, $ }

**2:** FOLLOW (E) = { i, k, b, $, g }

3: FOLLOW (F) = { k, b, $ }

**1:** FOLLOW (S) = { $ }

**2:** FOLLOW (A) = { b, g, a }

**3:** FOLLOW (C) = { b, $ }

**4:** FOLLOW (F) = { k, b, $ }

5: FOLLOW(B) = { g, $ }

# **Autómata LR 2**

Dada la siguiente gramática, ¿cuál de las siguientes afirmaciones relativas al Autómata Reconocedor de Prefijos Viables (método de Análisis Sintáctico Ascendente LR(1)) es correcta?

S → B A | C B

A → 1 A | 2 A | λ

B → 3 C 4 | λ

C → 3 | 2 B

Respuesta:

El estado inicial contiene al ítem A→•2A

Esta respuesta es incorrecta.

Explicación:

Al no aparecer en ningún momento la configuración “•A” en la parte derecha de un ítem, las reglas de A no entran en juego para añadir nuevos ítems al estado

La respuesta correcta era:

Respuesta:

Desde el estado inicial, hay una transición etiquetada con el símbolo “1” a un estado donde se encuentra el ítem A→•1A

Explicación:

El ítem A→•1A está presente en el estado inicial y eso produce una transición con “1” correspondiente al Goto (I0,1) en el que al calcular el cierre({A→1•A}) vuelve a salir A→•1A

**Respuesta seleccionada:** El estado inicial contiene al ítem C→•2B

**Explicación:**

Como está el ítem del axioma (S’→•S) y de ahí surge el ítem S→•CB, hay que introducir los ítems que surgen a partir de C

# **Conflictos LR**

Dado el siguiente estado perteneciente al Autómata reconocedor de Prefijos Viables de un Analizador Sintáctico LR, In={S→A F •, A→B D •, B→3 • A, B→• 3 A, C→3 • 4, B→5 A •}

¿cuál de las siguientes afirmaciones es correcta?

Si se añade el ítem A→3 •, habría un conflicto de Reducción-Reducción independientemente de cual fuera el Follow (A)

Explicación:

En todo caso habría dos reducciones posibles para cualquier elemento del Follow (A), por A→B D y por A→3

Si el Follow(A) contiene el terminal “4”, hay un conflicto de Reducción-Desplazamiento

Explicación:

En este estado, se debería desplazar con el token “4” (por C→3 • 4), pero también reducir (por A→B D •), por lo que hay un conflicto

Si el Follow(A) contiene el terminal “3”, hay un conflicto de Reducción-Desplazamiento

El estado inicial contiene el ítem B -> · 3C4  
Explicación:

Como está el ítem axioma(S’ -> ·S) y de ahí surge el ítem S -> ·BC, hay que introducir los ítems que surgen a partir de B.

Si se añade el ítem A→3 •, habría un conflicto de Reducción-Reducción independientemente de cual fuera el Follow (A)

## **First - 6**

¿Cuáles de los siguientes conjuntos son correctos, dada la siguiente gramática?

P → D P | S P | λ

D → var T id ; D | λ | F ; D

F → function id T ( id : T L ) begin S end

L → ; id : T L | λ

T → integer | boolean

S → if E do S | return E | id := E ; S

E → id ( K )

K → E R

R → λ | ; K R

**1:** First (P) = {var, function, λ, if, return, id}

  Esta respuesta es correcta.

**Explicación:**

P puede ser lambda o empezar por lo mismo que empieza D o S

**2:** First (D) = {var, λ, function}

  Esta respuesta es correcta.

**Explicación:**

D puede ser lambda o empezar por var o por lo que empiece F

**3:** First (S) = {if, return, id}

  Esta respuesta es correcta.

**4:** First (K) = {id}

Esta respuesta es correcta.

**Explicación:**

K empezará por lo mismo por lo que empieza E

**5:** First (E) = {id}

  Esta respuesta es correcta.

**Explicación:**

E siempre empezará con un id

**6:** First (:=) = {:=}

  Esta respuesta es correcta.

**Explicación:**

El First de un terminal es el propio terminal

7. First (R) = {;, λ}

8. First (L) = {;, λ}

9. First (boolean) = {boolean}

10. First (F) = {function}

# **Gramática aumentada**

Gramática aumentada -> Descendente:

¿Cuáles de las siguientes afirmaciones son correctas en relación con la Gramática aumentada?

Respuestas Seleccionadas:

1: Si S es el axioma de la gramática original, la gramática aumentada tendrá la regla S'→S, siendo S' un nuevo axioma

  Esta respuesta es correcta.

Explicación:

Así es como se crea la gramática aumentada

2: La gramática aumentada se obtiene a partir de la gramática original, añadiendo un nuevo axioma y una nueva regla que deriva el nuevo axioma en el axioma de la gramática original

  Esta respuesta es correcta.

Explicación:

De esta manera, se asegura que al reducir por esta regla, se aceptará la cadena de entrada

3: La gramática aumentada es necesaria para que un Analizador Sintáctico Ascendente LR sepa cuándo se debe utilizar la acción de Aceptar

  Esta respuesta es correcta.

Explicación:

Se necesita para poder identificar la acción de Aceptar en un analizador ascendente

**2:** La gramática aumentada es necesaria para construir un Analizador Sintáctico Ascendente LR

  Esta respuesta es correcta.

**Explicación:**

Se necesita para poder identificar la acción de Aceptar

**Además, te ha faltado por seleccionar:**

**Respuesta:**

La nueva regla del axioma introducida en la gramática aumentada dará lugar a un ítem en el estado inicial del autómata de prefijos viables que tendrá el punto inmediatamente antes del axioma de la gramática original

**Explicación:**

Este ítem indica que aún no ha comenzado el análisis de la cadena de entrada

**1:** Si S es el axioma de la gramática original y S' el nuevo axioma de la gramática aumentada, el estado inicial del autómata del LR tendrá el ítem S'→·S

  Esta respuesta es correcta.

**Explicación:**

Este ítem indica que aún no ha comenzado el análisis de la cadena de entrada

**Respuesta:**

La gramática aumentada es necesaria para asegurarse que el axioma nunca aparecerá en el lado derecho de ninguna regla

**Explicación:**

De esta manera, al reducir por la nueva regla del nuevo axioma, se sabrá que se debe aceptar la cadena

# **goto**

Para la siguiente gramática, indica cuáles de los siguientes cálculos de los conjuntos cierre o goto son correctos, cuando se quiere construir un Analizador LR:

S → A B

A → B C | k C

B → C D | λ

C → + A

D → k B

Respuestas seleccionadas:

I0=cierre({S'→·S})={S'→·S, S→·AB, A→·BC, A→·kC, B→·CD, B→·, C→·+A}

1: I2=goto(I0, A)={S→A·B, B→·CD, B→·, C→·+A}

  Esta respuesta es correcta.

Explicación:

El cálculo es correcto

1: I1=goto(I0, S)={S'→S·} Es

 ta respuesta es correcta.

Explicación:

El cálculo es correcto

2: I3=goto(I0, B)={A→B·C, C→·+A}

  Esta respuesta es correcta.

Explicación:

El cálculo es correcto

**3.**  I5=goto(I0, C)={B→C·D, D→·kB}

**Explicación:**

El cálculo es correcto

**4.**  I6=goto(I0, +)={C→+·A, A→·BC, A→·kC, B→·CD, B→·, C→·+A}

**Explicación:**

El cálculo es correcto

5. I4=goto(I0, k)={A→k·C, C→·+A}

1: I7=goto(I0, D)={}   Esta respuesta es correcta. Explicación: El cálculo es correcto

Resultados Autómata LR

Dada la siguiente gramática, ¿cuál de las siguientes afirmaciones relativas al Autómata Reconocedor de Prefijos Viables (método de Análisis Sintáctico Ascendente LR(1)) es correcta?

S → A B | B C

A → 1 A | 2 A | λ

B → 3 C 4 | λ

C → 3 | 2 B

Respuesta seleccionada: El ítem C→•3 pertenece al estado inicial del autómata

  La respuesta seleccionada fue incorrecta

Explicación:

Al no aparecer en ningún momento la configuración “•C” en la parte derecha de un ítem, las reglas de C no entran en juego para añadir nuevos ítems al estado

La respuesta correcta era:

  El estado inicial contiene al ítem A→•2A

Dada la siguiente gramática, ¿cuál de las siguientes afirmaciones relativas al Autómata Reconocedor de Prefijos Viables (método de Análisis Sintáctico Ascendente LR(1)) es correcta?

S → A B | B C

A → 1 A | 2 A | λ

B → 3 C 4 | λ

C → 3 | 2 B

**Respuesta seleccionada:** El estado inicial contiene al ítem B→•3C4

  La respuesta seleccionada fue correcta

**Explicación:**

Como está el ítem del axioma (S’→•S) y de ahí surge el ítem S→•BC, hay que introducir los ítems que surgen a partir de B

# **First - 8**

¿Cuáles de los siguientes conjuntos son correctos, dada la siguiente gramática?

P → A B C | d

A → a | λ

B → a | b | λ

C → e A D R f | λ

D → B S E

R → int | bool | P

S → if ( E ) { S } else { S } ; S | id = E ; S | λ

E → ( E ) | g

**Respuestas seleccionadas:**

**1:** First (A) = {a, λ}

  Esta respuesta es correcta.

**Explicación:**

A empieza por a o por nada

**2:** First (C) = {e, λ}

  Esta respuesta es correcta.

**Explicación:**

C empieza por e o por nada

**3:** First (D) = {a, b, if, id, (, g}

  Esta respuesta es correcta.

**Explicación:**

D empieza por lo que empieza B , S o

**4:** First (S) = {if, id, λ}

  Esta respuesta es correcta.

**Explicación:**

Las reglas de S empiezan por esos terminales o son anulables

**5**. First (E) = { (, g }

FIRST 7

¿Cuáles de los siguientes conjuntos, correspondientes a lados derechos de reglas, son correctos, dada la siguiente gramática?

P → D P | S P | λ

D → var T id ; D | λ | F ; D

F → function id T ( id : T L ) begin S end

L → ; id : T L | λ

T → integer | boolean

S → if E do S | return E | id := E ; S

E → id ( K )

K → E R

R → λ | ; K R

**Respuestas Seleccionadas:**

**1:** First (S P) = {if, return, id}

  Esta respuesta es correcta.

**Explicación:**

S siempre empieza por una de estas tres palabras, con lo que no tenemos que preocuparnos de P

**2:** First (var T id ; D) = {var}

  Esta respuesta es correcta.

**Explicación:**

Esta cadena de símbolos gramaticales siempre empieza por var

**3:** First (E R) = {id}

  Esta respuesta es correcta.

**Explicación:**

E siempre empieza por identificador

**4:** First (D P) = {var, function, λ, if, return, id}

  Esta respuesta es correcta.

**Explicación:**

D empieza por var y F, y puede ser lambda, por lo que tenemos que ver por qué empieza P, que también puede ser lamda

**2:**First (F ; D) = {function}

  Esta respuesta es correcta.

**Explicación:**

F siempre empieza por function

**3:**First (var T id ; D) = {var}

  Esta respuesta es correcta.

**1:** First (id := E ; S) = {id}

  Esta respuesta es correcta.

**Explicación:**

Esta cadena de símbolos gramaticales siempre empieza por id

## **First - 8**

Has terminado la actividad:

¿Cuáles de los siguientes conjuntos son correctos, dada la siguiente gramática?

P → A B C | d

A → a | λ

B → a | b | λ

C → e A D R f | λ

D → B S E

R → int | bool | P

S → if ( E ) { S } else { S } ; S | id = E ; S | λ

E → ( E ) | g

**Respuestas Seleccionadas:**

**1:**First (R) = {int, bool, a, b, e, λ, d}

  Esta respuesta es correcta.

**Explicación:**

R empieza por un tipo o por lo que empieza P

**2:**First (C) = {e, λ}

  Esta respuesta es correcta.

**Explicación:**

C empieza por e o por nada

**3:**First (P) = {a, b, e, λ, d}

  Esta respuesta es correcta.

**Explicación:**

P empieza siempre por d o por lo que empieza A, por lo que empieza B o por lo que empieza C

## **Follow - 3**

Has terminado la actividad:

¿Cuáles de los siguientes conjuntos son correctos, dada la siguiente gramática?

P → D P | S P | λ

D → var T id ; D | λ | F ; D

F → function id T ( id : T L ) begin S end

L → ; id : T L | λ

T → integer | boolean

S → if E do S | return E | id := E ; S

E → id ( K )

K → E R

R → λ | ; K R

**Respuestas Seleccionadas:**

**1:** Follow (S) = {var, function, $, if, return, id, end}

  Esta respuesta es correcta.

**Explicación:**

Después de S solo puede venir P.

**2:** Follow (T) = {id, (, ), ;}

  Esta respuesta es correcta.

**Explicación:**

Después de T solo puede venir L o lo que venga tras L

**3:** Follow (R) = {), ;}

  Esta respuesta es correcta.

**Explicación:**

Tras R vendrá lo mismo que lo que puede estar tras K

**4:** Follow (E) = {do, var, function, $, if, return, id, end, ;, )}

  Esta respuesta es correcta.

**Explicación:**

E puede ir seguido de do, del punto y coma, de R y de lo que venga tras S

**5:** Follow (F) = {;}

  Esta respuesta es correcta.

**Explicación:**

Después de F no puede venir nada más

**6:** Follow (K) = {), ;}

  Esta respuesta es correcta.

**Explicación:**

Tras K solo puede venir el paréntesis o R

**5:** Follow (P) = {$}

  Esta respuesta es correcta.

**Explicación:**

Por definición, el dólar está después del axioma

**Además, te ha faltado por seleccionar:**

**Respuesta:**

Follow (D) = {var, function, $, if, return, id}

**Explicación:**

Detrás de D viene P (que puede ser anulable)

Follow(L) = { ) }

## **Follow - 4**

¿Cuáles de los siguientes conjuntos son correctos, dada la siguiente gramática?

P → A B C | d

A → a | λ

B → a | b | λ

C → e A D R f | λ

D → B S E

R → int | bool | P

S → if ( E ) { S } else { S } ; S | id = E ; S | λ

E → ( E ) | g

**Respuestas Seleccionadas:**

**1:** Follow (C) = {$, f}

  Esta respuesta es correcta.

**Explicación:**

Tras C puede venir lo que hay tras P y tras R

**2:** Follow (A) = {a, b, e, $, f, if, id, (, g}

  Esta respuesta es correcta.

**Explicación:**

Hay que ver por qué empieza B C y D R f y añadir el dólar

**3:** Follow (R) = {f}

  Esta respuesta es correcta.

**Explicación:**

Tras R es lo único que hay

Follow(P) = {$, f}

Follow(E) = {), ;, int, bool, a, b, e, d, f}

**1:**Follow (S) = {}, (, g}

  Esta respuesta es correcta.

**Explicación:**

Tras S puede venir la llave o por lo que empiece E

**2:**Follow (D) = {int, bool, a, b, e, d, f}

  Esta respuesta es correcta.

**Explicación:**

Tras D viene todo lo que se puede derivar de R

**3:**Follow (C) = {$, f}

  Esta respuesta es correcta.

**Explicación:**

Tras C puede venir lo que hay tras P y tras R

**1:** Follow (B) = {e, $, f, if id, (, g}