

**Compiladores e Intérpretes.  
San José**

**Grupo: 40.  
Apuntes del 17 de mayo del 2017.**

Profesor: Dr. Francisco Torres Rojas.  
Apuntador: Dylan Rodríguez Barboza.  
2015057714.

I Semestre, 2017



## Contenido

Recordatorios .....	3
Quiz #10.....	4
FOLLOW() .....	5
Cálculo de FOLLOW().....	5
Argumento de FOLLOW() .....	5
FOLLOW(X) .....	5
Observaciones de FOLLOW().....	6
Ejemplo de FOLLOW() .....	8
Ejemplo de FIRST() y FOLLOW() .....	12
Obteniendo el FIRST().....	12
PREDICT().....	12
Cálculo del PREDICT() .....	14
Argumento de PREDICT() .....	14
PREDICT(X).....	14
Ejemplo de PREDICT().....	15
Otro ejemplo de PREDICT() .....	16
Construcción de la Tabla de Parsing .....	18
.....	18
.....	18
Ejemplo de Construcción de Tabla de Parsing .....	20
Otro ejemplo de Construcción de Tabla de Parsing.....	20

### Recordatorios

- El último **quiz** será el miércoles 24 de mayo, además, se debe entregar la tarea sobre las gramáticas de micro que se mencionó en clase.
- La Entrega del **proyecto 3** será el miércoles 31 de mayo.
- El **segundo examen** será el viernes 2 de junio.
- El **examen final** será el viernes 9 de mayo.
- El profe nos dará **donas**.

## Quiz #10

CALCULE EL **FIRST** DE TODOS LOS NO TERMINALES DE LAS SIGUIENTES 3 GRAMÁTICAS:

$$E \rightarrow iE'$$

$$E' \rightarrow +iE' \mid \varepsilon$$

$$A \rightarrow aAa \mid \varepsilon$$

$$D \rightarrow TV$$

$$T \rightarrow int \mid float$$

$$V \rightarrow id \mid ; \mid V \mid id$$

## FOLLOW()



### Cálculo de FOLLOW()

- Sea  $G$  una GFG-CFG.
- La función **FOLLOW**( $X$ ) regresa un conjunto de terminales, y posiblemente  $\epsilon$ , que indican todas las continuaciones factibles del No terminal  $X$  bajo la gramática  $G$ . **NO TENDRÁ A  $\epsilon$ .**

### Argumento de FOLLOW()

- En **FOLLOW**( $X$ ), el argumento  $X$  solo puede ser un no terminal.
- El cálculo de **FOLLOW** es iterativo, hasta que no haya más cambios.

### FOLLOW( $X$ )

- Si  $X$  es el símbolo inicial, agregue  $\$$  a **FOLLOW**( $X$ ).
- Si hay una regla  $A \rightarrow \alpha X \beta$ , agregue **FIRST**( $\beta$ ) –  $\{\epsilon\}$  a **FOLLOW**( $X$ ).
- Si hay una regla  $A \rightarrow \alpha X \beta$ , y  $\epsilon$  pertenece a **FIRST**( $\beta$ ), agregue **FOLLOW**( $A$ ) a **FOLLOW**( $X$ ).



Sin embargo...

.

.

.

Este último paso se considera mágico, #Misteriosossonloscaminosdelseñor



### Observaciones de FOLLOW()

- Sólo se calcula para **no terminales**.
- $\epsilon$  nunca es para de **FOLLOW()** pero \$ sí puede estar.
- Primero hay que calcular **FIRST()** de todos los **no terminales**.
- Se ignoran las reglas de la gramática que, únicamente, tengan terminales o  $\epsilon$  en su lado derecho, las demás se procesan repetidamente hasta que **NO** haya cambios.

-“Profe, me imagino que antes del FOLLOW() se debe hacer el FIRST()”.

-“Pues, se llama FOLLOW(), que sigue al FIRST() el cual significa primero.”





## Ejemplo de FOLLOW()

$E \rightarrow TE'$   
 $E' \rightarrow OPT E'$   
 $E' \rightarrow \varepsilon$   
 $OP \rightarrow +$   
 $OP \rightarrow -$   
 $T \rightarrow FT'$   
 $T' \rightarrow M F T'$   
 $T' \rightarrow \varepsilon$   
 $M \rightarrow *$   
 $F \rightarrow (E)$   
 $F \rightarrow \#$

- Debido a que solo se toman en cuenta las reglas que poseen no terminales, se puede pasar por recto de las reglas **3, 4, 5, 8, 9 y 11**.
- Ahora, se verá cada pasada, para llegar a los **FOLLOW()** deseados.

### Primera pasada:

NO TERMINAL	FOLLOW()		
	PASADA 1	PASADA 2	PASADA 3
<b>E</b>	{ \$ , ) }		
<b>E'</b>	{ \$ }		
<b>OP</b>	{ ( , # }		
<b>T</b>	{ + , - , \$ }		
<b>T'</b>	{ + , - , \$ }		
<b>M</b>	{ ( , # }		
<b>F</b>	{ * , + , - , \$ }		

En esta pasada, se tomará en cuenta lo siguiente:

- Al símbolo inicial se le asigna un \$.
- El **FIRST()** de E' es { + , - ,  $\varepsilon$  }, por lo que se introduce en el **FOLLOW()** de T, sin contar  $\varepsilon$ , quedando con { + , - }.

- Se tiene a  $\varepsilon$  a la derecha de  $E'$ , por lo que el **FOLLOW()** de  $E$  se debe asignar a  $E'$ , quedando con  $\{\$, \}$ .
- Debido a que  $E'$  puede ser  $\varepsilon$ , y este se encuentra a la derecha de  $T$ , se debe pasar el **FOLLOW()** de  $E$  a  $T$ , quedando con  $\{+, -, \$\}$ .
- En la segunda regla, se tiene a  $E'$  a la derecha de  $T$ , esto ya fue tomado en cuenta con la primera regla, por lo que, al pasar el **FIRST()** de  $E'$  a  $T$ , se obtiene el mismo resultado anterior:  $\{+, -, \$\}$ .
- El **FIRST()** de  $T$  es  $\{(, \#)$ , por lo que se debe pasar al **FOLLOW()** de  $OP$ , quedando con  $\{(, \#)$ .
- A la derecha de  $E'$  se tiene  $\varepsilon$ , por lo que el **FOLLOW()** de  $E'$  se pasa a  $T$ , quedando con  $\{+, -, \$\}$ .
- Debido a que  $T$  no podría ser  $\varepsilon$ , no se debe pasar el **FOLLOW()** de  $E'$  a  $OP$ .
- En la regla 6, el **FIRST()** de  $T'$  es  $\{*, \varepsilon\}$ , por lo que este debe ser pasado al **FOLLOW()** de  $F$ , quedando con  $\{*\}$ .
- A la derecha de  $T'$  hay un  $\varepsilon$ , por lo que se debe pasar el **FOLLOW()** de  $T$  a  $T'$ , quedando con  $\{+, -, \$\}$ .
- Debido a que el **FIRST()** de  $T'$  puede ser  $\varepsilon$ , se pasa el **FOLLOW()** de  $T$  a  $F$ , quedando con  $\{*, +, -, \$\}$ .
- En la regla 7, el **FIRST()** de  $T'$  es  $\{*, \varepsilon\}$ , por lo que este es pasado a  $F$ , quedando con  $\{*, +, -, \$\}$ .
- El **FIRST()** de  $F$  es  $\{(, \#)$ , por lo que se pasa al **FOLLOW()** de  $M$ , quedando con  $\{(, \#)$ .
- A la derecha de  $T'$  hay un  $\varepsilon$ , por lo que el **FOLLOW()** de  $T'$  se pasa a  $T'$ , quedando con  $\{+, -, \$\}$ .
- Debido a que el **FIRST()** de  $T'$  puede ser  $\varepsilon$ , se pasa el **FOLLOW()** de  $T'$  a  $F$ , quedando con  $\{*, +, -, \$\}$ .
- Debido a que el **FIRST()** de  $F$  no será  $\varepsilon$ , no se debe pasar el **FOLLOW()** de  $T'$  a  $M$ .
- En la regla 10, se tiene de primero un  $($ , el cual se “ignora”, para seguir con el no terminal  $E$ , acto seguido, se debe tener el **FIRST()** de  $)$ , el cual es  $)$ , se debe pasar al **FOLLOW()** de  $E$ , quedando con  $\{\$, )\}$ .

Segunda pasada:

No TERMINAL	FOLLOW()		
	PASADA 1	PASADA 2	PASADA 3
<b>E</b>	{ \$ , ) }	{ \$ , ) }	{ \$ , ) }
<b>E'</b>	{ \$ }	{ \$ , ) }	{ \$ , ) }
<b>OP</b>	{ ( , # }	{ ( , # }	{ ( , # }
<b>T</b>	{ + , - , \$ }	{ + , - , \$ , ) }	{ + , - , \$ , ) }
<b>T'</b>	{ + , - , \$ }	{ + , - , \$ , ) }	{ + , - , \$ , ) }
<b>M</b>	{ ( , # }	{ ( , # }	{ ( , # }
<b>F</b>	{ * , + , - , \$ }	{ * , + , - , \$ , ) }	{ * , + , - , \$ , ) }

En esta pasada, se tomará en cuenta lo siguiente:

- El **FIRST()** de E' es { + , - ,  $\varepsilon$  }, por lo que se introduce al **FOLLOW()** de T, quedando con { + , - , \$ }.
- A la derecha de E' se encuentra  $\varepsilon$ , por lo que el **FOLLOW()** de E se introduce a E', quedando con { \$ , ) }.
- Debido a que el **FIRST()** de E' podría ser  $\varepsilon$ , se introduce el **FOLLOW()** de E a T, quedando con { + , - , \$ , ) }.
- En la regla 2, el **FIRST()** de E' es { + , - ,  $\varepsilon$  }, por lo que se introduce en el **FOLLOW()** de T, quedando con { + , - , \$ , ) }.
- El **FIRST()** de T es { ( , # }, por lo que se introduce al **FOLLOW()** de OP, quedando con { ( , # }.
- A la derecha de E' se encuentra  $\varepsilon$ , por lo que se introduce el **FOLLOW()** de E' a E', quedando con { \$ , ) }.
- Debido a que el **FIRST()** de E' podría ser  $\varepsilon$ , se introduce el **FOLLOW()** de E' a T, quedando con { + , - , \$ , ) }.
- Debido a que el **FIRST()** de T no podrá ser  $\varepsilon$ , no se introduce el **FOLLOW()** de E' a OP.
- En la regla 6, el **FIRST()** de T' es { \* ,  $\varepsilon$  }, por lo que se introduce al **FOLLOW()** de F, quedando con { \* , + , - }.
- Debido a que a la derecha de T' está  $\varepsilon$ , se introduce el **FOLLOW()** de T al **FOLLOW()** de T', quedando con { + , - , \$ , ) }.
- Debido a que el **FIRST()** de T' puede ser  $\varepsilon$ , se introduce el **FOLLOW()** de T al **FOLLOW()** de F, quedando con { \* , + , - , \$ , ) }

- En la regla 7, el **FIRST()** de  $T'$  es  $\{^*, \varepsilon\}$ , por lo que se introduce al **FOLLOW()** de  $F$ , quedando con  $\{^*, +, -, \$, )\}$ .
- El **FIRST()** de  $F$  es  $\{(, \#)$ , por lo que se introduce al **FOLLOW()** de  $M$ , quedando con  $\{(, \#)$ .
- Debido a que a la derecha de  $T'$  está  $\varepsilon$ , se introduce el **FOLLOW()** de  $T'$  a  $T'$ , quedando con  $\{+, -, \$, )\}$ .
- Debido a que el **FIRST()** de  $T'$  puede ser  $\varepsilon$ , se introduce el **FOLLOW()** de  $T'$  a  $F$ , quedando con  $\{^*, +, -, \$, )\}$ .
- Debido a que  $F$  no será  $\varepsilon$ , no hace falta introducir el **FOLLOW()** de  $T'$  a  $M$ .
- En la regla 10, no se toma en cuenta el  $($ , se sigue con el no terminal  $E$ .
- Debido a que el **FIRST()** de  $)$  es  $)$ , se introduce al **FOLLOW()** de  $E$ , quedando con  $\{ \$, ) \}$ .
- En la siguiente pasada que se quiera hacer, será solo para comprobación, debido a que los **FOLLOW()** darán el mismo resultado.

-“¿Hay forma de ir calculando el **FIRST()** y el **FOLLOW()** al mismo tiempo?”.

-“Sigán la receta y no busquen atajos o les va a ir **M U Y M A L**”.

-“Bison sí usa atajos”.

-“Pero Bison fue creado siguiendo la receta”.



## Ejemplo de FIRST() y FOLLOW()

$$S \rightarrow ( S ) S$$

$$S \rightarrow \varepsilon$$

### Obteniendo el FIRST()

NO TERMINAL	FIRST()		
	PASADA 1	PASADA 2	PASADA 3
S	$\emptyset$	$\{ \varepsilon , ( \}$	$\{ \varepsilon , ( \}$

Se debe tomar en cuenta que:

- ( no posee a  $\varepsilon$ , por lo que se coloca en el **FIRST()** de S y se continúa con la siguiente regla.
- $\varepsilon$  se coloca en el **FIRST()** de S, quedando con el conjunto  $\{ \varepsilon , ( \}$
- En las siguientes pasadas, al ( no poseer  $\varepsilon$ , no se pasa al no terminal S, por lo que las pasadas quedan con el mismo **FIRST()**.

## PREDICT()

-Mucho cuidado, Acuario. Ahora vamos contigo Piscis....



-B11 (Beyoncé)

- “En nivel de dificultad, comparado con el *FIRST()* y *FOLLOW()* ¿Qué tal es?”.

-“Facilísimo”.

-“No lo sé Rick, parece falso”.



### Cálculo del PREDICT()

- Sea  $G$  una ~~GFG~~-CFG.
- La función **PREDICT**( $X$ ) regresa un conjunto de terminales y, posiblemente  $\epsilon$  que predican cuando hay que usar la regla  $X$ .

### Argumento de PREDICT()

- En **PREDICT**( $X$ ), el argumento  $X$  solo puede ser una **regla de la gramática**.
- Antes, se deben calcular los **FIRST**() y los **FOLLOW**().

### PREDICT( $X$ )

- $X$  es una regla de la forma  $A \rightarrow \alpha$ .
- If(**FIRST**( $\alpha$ ) incluye a  $\epsilon$ )-> **PREDICT**( $X$ ) = (**FIRST**( $\alpha$ ) -  $\{\epsilon\}$ )  $\cup$  **FOLLOW**( $A$ )).
- Else-> **PREDICT**( $X$ ) = **FIRST**( $\alpha$ ).

## Ejemplo de PREDICT()

$$S \rightarrow (S) S$$

$$S \rightarrow \varepsilon$$

Nos dan los **FIRST()** y **FOLLOW()** de S:

NO TERMINAL	FIRST()	FOLLOW()
S	{( , $\varepsilon$ }	{\$ , )}

Para obtener el PREDICT(), utilizamos la siguiente tabla:

REGLA	FIRST( $\alpha$ )	FOLLOW(A)	PREDICT()
<b><math>S \rightarrow (S) S</math></b>	{(}	{\$ , )}	{(}
<b><math>S \rightarrow \varepsilon</math></b>	{ $\varepsilon$ }	{\$ , )}	{\$ , )}

Se debe considerar:

- En la regla 1, el **FIRST( $\alpha$ )** no incluye a  $\varepsilon$ , por lo que el **PREDICT()** para esta regla será el **FIRST( $\alpha$ )**, es decir, {(}
- En la regla 2, el **FIRST( $\alpha$ )** incluye a  $\varepsilon$ , por lo que el **PREDICT()** para esta regla será **(FIRST( $\alpha$ ) – { $\varepsilon$ })  $\cup$  FOLLOW(A)**, es decir, (({ $\varepsilon$ } - { $\varepsilon$ })  $\cup$  {\$ , })), dando como **PREDICT()** {\$ , }).



## Otro ejemplo de PREDICT()

$E \rightarrow TE'$

$E' \rightarrow OPT E'$

$E' \rightarrow \varepsilon$

$OP \rightarrow +$

$OP \rightarrow -$

$T \rightarrow FT'$

$T' \rightarrow MFT'$

$T' \rightarrow \varepsilon$

$M \rightarrow *$

$F \rightarrow (E)$

$F \rightarrow \#$

	FIRST()	FOLLOW()
<b>E</b>	{( , #}	{\$ , )}
<b>T</b>	{( , #}	{\$ , ) , + , -}
<b>E'</b>	{+ , - , $\varepsilon$ }	{\$ , )}
<b>OP</b>	{+ , -}	{( , #}
<b>F</b>	{( , #}	{\$ , ) , + , - , *}
<b>T'</b>	{* , $\varepsilon$ }	{\$ , ) , + , -}
<b>M</b>	{*}	{( , #}

Obtendremos la siguiente tabla, donde obtenemos el **PREDICT()** de cada regla:

REGLA	<b>FIRST(<math>\alpha</math>)</b>	<b>PREDICT()</b>
<b><math>E \rightarrow TE'</math></b>	$\{( , \#)$	$\{( , \#)$
<b><math>E' \rightarrow OP T E'</math></b>	$\{+ , -\}$	$\{+ , -\}$
<b><math>E' \rightarrow \varepsilon</math></b>	$\{\varepsilon\}$	$\{\$ , )\}$
<b><math>OP \rightarrow +</math></b>	$\{+\}$	$\{+\}$
<b><math>OP \rightarrow -</math></b>	$\{-\}$	$\{-\}$
<b><math>T \rightarrow FT'</math></b>	$\{( , \#)$	$\{( , \#)$
<b><math>T' \rightarrow M F T'</math></b>	$\{*\}$	$\{*\}$
<b><math>T' \rightarrow \varepsilon</math></b>	$\{\varepsilon\}$	$\{\$ , ) , + , -\}$
<b><math>M \rightarrow *</math></b>	$\{*\}$	$\{*\}$
<b><math>F \rightarrow (E)</math></b>	$\{( \}$	$\{( \}$
<b><math>F \rightarrow \#</math></b>	$\{\#\}$	$\{\#\}$

Se debe tomar en cuenta lo siguiente:

- La columna **FIRST( $\alpha$ )** está conformada del **FIRST()** del primer terminal o no terminal (más a la izquierda) de la regla (a partir de la flecha).
- En la regla 1, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será **FIRST( $\alpha$ )**, es decir,  $\{( , \#)$ .
- En la regla 2, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será **FIRST( $\alpha$ )**, es decir,  $\{+ , -\}$ .
- En la regla 3, el **FIRST( $\alpha$ )** contiene a  $\varepsilon$ , por lo que el **PREDICT()** será **(FIRST( $\alpha$ ) -  $\{\varepsilon\}$ )  $\cup$  FOLLOW(A))**, es decir,  $(\{\varepsilon\} - \{\varepsilon\}) \cup (\{\$ , )\}$ , dando como **PREDICT()**  $\{\$ , )\}$ .
- En la regla 4, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{+\}$ .
- En la regla 5, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{-\}$ .
- En la regla 6, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{( , \#)$ .
- En la regla 7, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{*\}$ .
- En la regla 8, el **FIRST( $\alpha$ )** contiene a  $\varepsilon$ , por lo que el **PREDICT()** será **(FIRST( $\alpha$ ) -  $\{\varepsilon\}$ )  $\cup$  FOLLOW(A))**, es decir  $(\{\varepsilon\} - \{\varepsilon\}) \cup (\{\$ , ) , + , -\})$ , dando como **PREDICT()**,  $\{\$ , ) , + , -\}$ .
- En la regla 9, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{*\}$ .
- En la regla 10, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{( \}$ .
- En la regla 11, el **FIRST( $\alpha$ )** no contiene a  $\varepsilon$ , por lo que el **PREDICT()** será el **FIRST( $\alpha$ )**, es decir,  $\{\#\}$ .

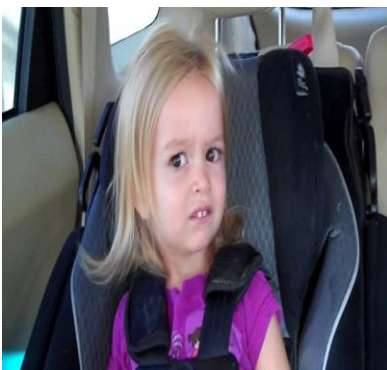
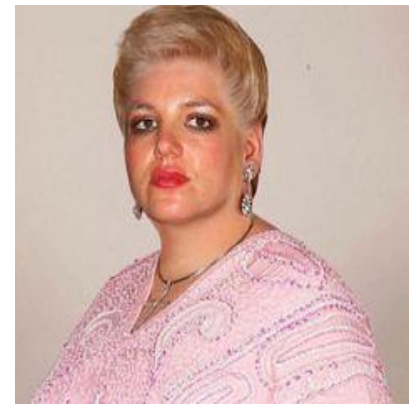
## Construcción de la Tabla de Parsing

Tendremos:

- Tabla **M**: una fila por cada no terminal y una columna por cada terminal y \$.
- Se calculan los **FIRST()** y los **FOLLOW()**.
- Se calcula el **PREDICT()** de cada regla.
- Para toda regla  $A \rightarrow \alpha$ : Para cada elemento de  $a$  en **PREDICT**( $A \rightarrow \alpha$ ) agregue regla  $A \rightarrow \alpha$  en  $M[A][a]$ .

Así de fácil es ☺

Todos:



Y bueno, todo comenzó a tener sentido...

*“Semana 14 y todo empieza a tener sentido”.*



## Ejemplo de Construcción de Tabla de Parsing

Nos dan las Reglas y **PREDICT()** de cada una:

REGLA	PREDICT()
$S \rightarrow ( S ) S$	{(}
$S \rightarrow \varepsilon$	{), \$}

La tabla quedará:

	(	)	\$
S	1	2	2

Se debe tomar en cuenta que:

- Los números representan las reglas con las que coinciden con el **PREDICT()**, se muestra, por ejemplo, que el '(' se encuentra en el conjunto del **PREDICT()** de la Regla 1, siendo este representado en la tabla, lo mismo ocurre con ')' y '\$', los cuales pertenecen al **PREDICT()** de la regla 2.

## Otro ejemplo de Construcción de Tabla de Parsing

REGLA	PREDICT()
$E \rightarrow TE'$	{(, #}
$E' \rightarrow OP T E'$	{+, -}
$E' \rightarrow \varepsilon$	{\$, )}
$OP \rightarrow +$	{+}
$OP \rightarrow -$	{-}
$T \rightarrow FT'$	{(, #}
$T' \rightarrow M F T'$	{*}
$T' \rightarrow \varepsilon$	{\$, ), +, -, }
$M \rightarrow *$	{*}
$F \rightarrow (E)$	{(}
$F \rightarrow \#$	{#}

Dando como resultado la tabla de parsing:

	(	#	)	+	-	*	\$
<b>E</b>	1	1					
<b>E'</b>			3	2	2		3
<b>OP</b>			4	5			
<b>T</b>	6	6					
<b>T'</b>			8	8	8	7	8
<b>M</b>						9	
<b>F</b>	10	11					

Se debe tomar en cuenta que:

- Para la regla 1 (E), el **PREDICT()** contiene {( , #}.
- Para la regla 2 (E') el **PREDICT()** contiene {+ , -}.
- Para la regla 3 (E') el **PREDICT()** contiene {\$ , )}.
- Para la regla 4 (OP) el **PREDICT()** contiene {+}.
- Para la regla 5 (OP) el **PREDICT()** contiene {-}.
- Para la regla 6 (T) el **PREDICT()** contiene {( , #}.
- Para la regla 7 (T') el **PREDICT()** contiene {\*}.
- Para la regla 8 (T') el **PREDICT()** contiene {\$ , ) , + , -}.
- Para la regla 9 (M) el **PREDICT()** contiene {\*}.
- Para la regla 10 (F) el **PREDICT()** contiene {}.
- Para la regla 11 (F) el **PREDICT()** contiene {#}.

-“¿Qué pasa si dos reglas quieren caer en un cuadro de la tabla? Por ejemplo, si en T con # cae 6,7”.

-“Sería algo malo, muy malo, se pueden considerar diferentes formas de suicidio”.



**ESTA FUE LA MATERIA VISTA EL 17 DE MAYO, RECUERDEN EMPEZAR EL PROYECTO 3 Y ESTUDIAR PARA EL QUIZ Y EXAMEN, AH SÍ, Y LA TAREA 😊.**

**BYE.**

