

Tarea con LA
 (id + id) * id + id
 Agustin Luzzillo Juan Peraldo

- ① $E \rightarrow E + T$
- ② $\rightarrow T$
- ③ $T \rightarrow T * F$
- ④ $\rightarrow F$
- ⑤ $F \rightarrow (E)$
- ⑥ $\rightarrow \text{id}$

Pila	B	Acción - GOTO
0	(id + id) * id + id	D-4
0, L-4	(id + id) * id + id	D-5
0, L-4, id-5	+ id) * id + id	B6-3
0, L-4, id-5,	+ id) * id + id	B4-2
0, L-4, F-3,	+ id) * id + id	B2-8
0, L-4, T-2,	+ id) * id + id	D-6
0, L-4, E-8) * id + id	D-5
0, L-4, E-8, +-6) * id + id	B6-3
0, L-4, E-8, +-6, id-5) * id + id	B4-9
0, L-4, E-8, +-6, F-3) * id + id	B1-8
0, L-4, E-8, +-6, T-9	* id + id	D-1-1
0, L-4, E-8	* id + id	B5-3
0, L-4, E-8,)-11	* id + id	B4-2
0, F-3	* id + id	D-7
0, T-2	id + id	D-5
0, T-2, *-2	+ id	B6-10
0, T-2, *-2, id-5	+ id	B3-2
0, T-2, *-2, F-10	+ id	B2-1
0, T-2	+ id	D-6
0, E-1	id	D-5
0, E-1, +-6	\$	B6-3
0, E-1, +-6, id-5	\$	B4-9
0, E-1, +-6, F-3	\$	B1-1
0, E-1, +-6, T-9	\$	Acceptar
0, E-1	\$	

Tarea 2L
(id + id) * id + id

- ① $E \rightarrow E + T$
- ② $E \rightarrow T$
- ③ $T \rightarrow T * F$
- ④ $T \rightarrow F$
- ⑤ $F \rightarrow (E)$
- ⑥ id

Buscando en la tabla
del libro con acciones derivadas

Pila de Símbolos	Cadena de Entrada	Acción
\$ E	(id + id) * id + id \$	4: $M[E, (] = E \rightarrow TE'$
\$ E' T	(id + id) * id + id \$	4: $M[T, (] = T \rightarrow FT'$
\$ E' T' F	(id + id) * id + id \$	4: $M[F, (] = F \rightarrow (E)$
\$ E' T') E ((id + id) * id + id \$	2: $(= ($
\$ E' T') E	(id + id) * id + id \$	4: $M[E, id] = E \rightarrow TE'$
\$ E' T') E' T	(id + id) * id + id \$	4: $M[T, id] = T \rightarrow FT'$
\$ E' T') E' T' E	(id + id) * id + id \$	4: $M[F, id] = F \rightarrow id$
\$ E' T') E' T' id	(id + id) * id + id \$	2: $id = id$
\$ E' T') E' T'	(id + id) * id + id \$	4: $M[T', +] = T' \rightarrow E$
\$ E' T') E'	(id + id) * id + id \$	$M[E', +] = E' \rightarrow +TE'$
\$ E' T') E' T +	(id + id) * id + id \$	$+ = +$
\$ E' T') E' T	(id + id) * id + id \$	$M[T, id] = FT'$
\$ E' T') E' T' F	(id + id) * id + id \$	$M[F, id] = F \rightarrow id$
\$ E' T') E' T' id	(id + id) * id + id \$	$id = id$
\$ E' T') E' T'	(id + id) * id + id \$	$M[T',)] = T' \rightarrow E$
\$ E' T') E'	(id + id) * id + id \$	$M[E',)] = E' \rightarrow E$
\$ E' T')	(id + id) * id + id \$	$) =)$
\$ E' T'	(id + id) * id + id \$	$M[T', *] = T' \rightarrow *FT'$
\$ E' T' F *	(id + id) * id + id \$	$* = *$
\$ E' T' F	(id + id) * id + id \$	$M[F, id] = F \rightarrow id$
\$ E' T' id	(id + id) * id + id \$	$id = id$
\$ E' T'	(id + id) * id + id \$	$M[T', +] = T' \rightarrow E$
\$ E'	(id + id) * id + id \$	$M[E', +] = E' \rightarrow +TE'$
\$ E' T +	(id + id) * id + id \$	$+ = +$
\$ E' T	(id + id) * id + id \$	$M[T, id] = T \rightarrow FT'$
\$ E' T' F	(id + id) * id + id \$	$M[F, id] = F \rightarrow id$
\$ E' T' id	(id + id) * id + id \$	$id = id$
\$ E' T'	(id + id) * id + id \$	$M[T', $] = T' \rightarrow E$
\$ E'	(id + id) * id + id \$	$M[E', $] = E' \rightarrow E$
\$	(id + id) * id + id \$	$$ = $$
		4: aceptar

¿Cómo se maneja el generador de errores en el análisis sintáctico?

Para el proceso de compilación, el programador no siempre está todo lo correcto que debe estar, por lo que debe estar preparado y tener listo, los casos o tipos de errores que se puedan presentar.

Considerar estos errores desde el principio ayuda a facilitar y simplificar la estructura del compilador.

El manejo de errores sintácticos es el más complicado dentro del diseño del compilador, puesto que se desea que al encontrar un error la compilación no se cancele, sino que siga encontrando errores y los recupere (Mora Mata y Gálvez Rojas, 2005).

Tomando en cuenta esto, tenemos los objetivos siguientes:

- Indicar el error de forma clara y precisa
- Recuperarse del error para seguir el análisis
- No reutilizar la compilación

Para manejarlos correctamente, Mora y Gálvez (2005) mencionan las siguientes estrategias:

- Ignorar el problema: Descartando los tokens a partir del error hasta el siguiente "token de seguridad".
- Recuperación a nivel frase: Intentar corregir el error encontrado.
- Reglas de producción adicionales: Añadir reglas de producción para los errores más comunes.
- Corrección Global: Recibe todos los tokens e intenta devolver un árbol lo más parecido a la cadena de entrada pero sin errores.

¿Qué es la aplicación que tienen los tableros de símbolos en el análisis sintáctico?

La tabla de símbolos es una estructura que almacena toda la información referente a los identificadores dentro del programa a analizar, ya sean nombres de variables o nombres de módulos del usuario.

En la fase del análisis sintáctico, no se usa tan impactantemente, pues el proceso en realidad solamente recupera o investiga en la tabla al momento de encontrar un identificador, para ver si este ya fue declarado o no para ser agregado a la tabla. Así mismo, en el momento o sección de declaración de las variables debe comprobar que la sentencia de declaración sea correcta, y que la tabla registre los datos de dicha declaración.

En sí, esta tabla sirve, durante el análisis sintáctico, para que las estructuras de las sentencias de declaración sean correctas y preparar todo para la fase de análisis semántico.

~~Tabla de los análisis~~

¿Cuál de los análisis ascendentes se adaptaría mejor a su ~~programa~~ lenguaje de programación y por qué?

Considero que de los ascendentes quedaría mejor el Ascendente de gramática LR(1), por la facilidad y velocidad que tiene para hacer el análisis, pese a el proceso que llevaría tener la tabla para dicho análisis, no tenemos gramáticas tan complejas (a mi consideración actual) para que el proceso de diseño sea muy pesado.

¿Cuál de los análisis descendentes se adaptaría mejor a su lenguaje de programación y por qué?

El análisis descendente con Funciones Recursivas porque el flujo con el que avanza el proceso de análisis, así como los diagramas para plantear dicho análisis, me parecen una manera clara de representar y llevar a cabo todo el proceso, además considero que encaja bien las gramáticas que puede generar nuestro lenguaje.

Referencias:

Mora Mata, M., y Gálvez Rojas, S. (2005). Java a Topic: Traductores y compiladores con Lex/Yacc, JFlex/Cup y Javacc. [Electronico]. Sergio Álvarez Rojas. https://eve.topic-tecnm.mx/pluginfile.php/85682/mod_resource/content/1/Libro%20de%20compiladores.pdf.