



## **Compiladores e Intérpretes**

Apuntes de la clase del día miércoles 24 de mayo del 2017

### **Apuntador:**

Jonathan Rodríguez Martínez

2014015749

### **Profesor:**

Dr. Francisco Torres Rojas

I Semestre 2017

## Contenido

Contenido .....	2
Quiz 11 .....	3
Repaso de la clase anterior (Rebobinando) .....	3
Mensajes de Error – Errores y LL(1) .....	3
Mensajes de error ejemplo .....	3
<b>Recuperación de error</b> .....	6
Acciones de recuperación de error .....	6
Operaciones: .....	6
Recuperación de error ejemplo .....	7
Recuperación de errores- Ejemplo 2 .....	8
Entretención: .....	10

## Quiz 11

Considere las siguientes 2 gramáticas. Haga todo lo necesario para construir la tabla de parsing correspondiente a cada una.

$$A. S \rightarrow aSdd \mid A$$

$$A \rightarrow bAc \mid bc$$

$$B. S \rightarrow 1 \mid 1S1 \mid 0S0$$

## Repaso de la clase anterior (Rebobinando)

### Mensajes de Error – Errores y LL(1)

- El algoritmo detecta hileras inválidas - entrada vacía en tabla LL(1)
- Estrategia simplista: o Que le diga: “Tiene un error”, “Error de sintaxis aquí” o No revisa el resto de la hilera.
- Mejor enfoque: o Mensajes de error: dar el mejor diagnostico posible de la causa del error o Recuperación de error: volver a sincronizar el autómata para reportar otros errores en hilera.
- Es casi un arte.
- Distintos compiladores dan mensajes diferentes para el mismo error.
- Hay muchas causas posibles.
- Es particular para cada lenguaje y situación.
- A veces se hacen de manera incremental: o Dar solo el número de error. (ERROR 17) o Ver ejemplos del error e ir mejorando la redacción.

### Mensajes de error ejemplo

- Consideremos la gramática:
  1.  $S \rightarrow a$
  2.  $S \rightarrow ( S R$
  3.  $R \rightarrow , S R$
  4.  $R \rightarrow )$

- Genera “S-expressions” o sea hileras de la forma:
  1. a
  2. (a)
  3. (a,a,a)
  4. (a,(a,a))
  5. ((a,(a,a),(a,a)),a)

Sería buena práctica que a la gramática anterior le calculáramos el First(), el Follow(), Predict() y construyéramos la tabla de parsing. Por ahora nos cae del cielo.

- Su tabla de parsing es la siguiente:

	a	,	(	)	\$
S	1		2		
R		3		4	
\$					☺

- Todos los errores (en las casillas en blanco) podrían ser de la forma:
  - “Aparece \_\_\_\_\_ cuando se esperaba \_\_\_\_\_”
- Calza bien para el caso de match
- Puede ser confuso para No-terminal y un terminal, es decir el error no es específico y podríamos “batear”.
- No –terminales no son muy claros para el programador.
- De la tabla de parsing anterior, analicemos las entradas vacías:

	a	,	(	)	\$
S	1	a	2	b	c
R	d	3	e	4	f
\$	g	h	i	j	☺

- En los errores **a** y **b** se esperaba un “S” (S-expression) y esos inicios son inválidos
- Un mensaje puede ser:
  - “Aparece “,” cuando se esperaba S-expresion”
  - “Aparece “)” cuando se esperaba S-expresion”
- El error **c** es diferente al **a** y **b**
- Las hileras que lo generan son de la forma
  - (\$
  - (a,\$
  - \$
- Un mensaje apropiado sería:

- “S-expression incompleta”
- “La expresión terminó inesperadamente”
- Los errores **d**, **e** y **f** están asociados al no terminal R
- R no tiene un significado evidente para los programadores
- En los errores no se puede mencionar R porque no nos diría nada, como el error de C “L-value” pero para los creadores del compilador de C “L-value” era significativo.
- Una estrategia es menciona las cosas que esperábamos cuando llega el token que nos ofende
  - “ ‘a’ aparece cuando se esperaba ‘,’ o ‘)’ ”
  - “ ‘(’ aparece cuando se esperaba ‘,’ o ‘)’ ”
  - “ Fin de la hilera aparece cuando se esperaba ‘,’ o ‘)’ ”
- Las hileras que típicamente causan los errores **d** y **e** son de la forma:
  - ( a **a** ) \$
  - ( ( a ) **a** ) \$
  - ( a ( a ) ) \$
  - ( ( a ) ( a ) ) \$
- Un mejor mensaje de error para estos casos sería:
  - “Falta ‘,’ ”



	a	,	(	)	\$
S	1	a	2	b	c
R	d	3	e	4	f
\$	g	h	i	j	☺

- Conociendo las hileras que típicamente causan el error **f** son de la forma:
  - ( a , a **\$** )
  - ( ( a ) , a **\$** )
  - ( a **\$** )
- Un mejor mensaje de error para este caso sería:
  - “Falta ‘)’ ”
- Los errores **g**, **h**, **i** y **j** se dan cuando el autómata cree que la hilera debe terminar y aún quedan símbolos:
  - ( a ) **a** \$
  - ( ( a ) , a ) **a** \$
  - ( a ) ( a ) **\$**
- Un mejor mensaje de error apropiado para todos estos casos sería:
  - “ \_\_\_ aparece después de fin de S-expression”
- De todos los mensajes anteriores se puede construir una tabla de mensajes de error para cada situación

Error	Mensaje
a	Aparece ',' cuando se esperaba S-expression
b	Aparece ')' cuando se esperaba S-expression
c	S-expression incompleta
d	Falta ','
e	Falta ','
f	Falta ')'
g	'a' aparece después de fin de S-expression
h	',' aparece después de fin de S-expression
i	('' aparece después de fin de S-expression
j	)' aparece después de fin de S-expression



## Recuperación de error

- Error Recovery
- Cascada de errores es causada por mala recuperación.
- Restablecer la pila y avanzar la entrada para poder seguir analizando
- No es 100% preciso – heurísticas
- Caso extremo:
  - Dar un solo mensaje de error
  - Baja productividad
  - ¿Cuánto salto?
- Riesgo
  - Casada de errores
  - Dar mensajes equivocados
  - Programador se concentra en los 2 o 3 primeros

## Acciones de recuperación de error

- Se asocian acciones de recuperación a cada situación
- Usar tablas de errores
- Específico a cada gramática y lenguaje

## Operaciones:

- POP: Saca al no terminal de la fila
- PUSH: inserta los elementos indicados en el TOP de la pila
- REPLACE: reemplaza el TOP de la pila por los elementos indicados
- ADVANCE: Avanza en la entrada

- RETAIN: permanece en el mismo punto de la entrada
- EXIT: terminar el parsing

Estas las vamos a asociar a la tabla de mensajes de error.

### • Posibles acciones

Error	Mensaje	Acción
a	Aparece ',' cuando se esperaba S-expression	POP –RETAIN
b	Aparece ')' cuando se esperaba S-expression	POP- RETAIN
c	S-expression incompleta	EXIT
d	Falta ','	REPLACE( R S)- RETAIN
e	Falta ','	REPLACE(R S) – RETAIN
f	Falta ')'	EXIT
g	'a' aparece después de fin de S-expression	PUSH(R S) – RETAIN
h	',' aparece después de fin de S-expression	PUSH(R) – RETAIN
i	('' aparece después de fin de S-expression	PUSH (R S) – RETAIN
j	')' aparece después de fin de S-expression	ADVANCE

### Recuperación de error ejemplo

Vamos a parsear la hilera: ( a , , a \$

1.  $S \rightarrow a$
2.  $S \rightarrow ( S R$
3.  $R \rightarrow , S R$
4.  $R \rightarrow )$

	a	,	(	)	\$
S	1	a	2	b	c
R	d	3	e	4	f
\$	g	h	i	j	☺

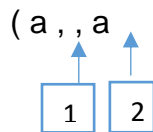
Error	Mensaje	Acción
a	Aparece ',' cuando se esperaba S-expression	POP –RETAIN
b	Aparece ')' cuando se esperaba S-expression	POP- RETAIN
c	S-expression incompleta	EXIT
d	Falta ','	REPLACE( R S)- RETAIN
e	Falta ','	REPLACE(R S) – RETAIN
f	Falta ')'	EXIT
g	'a' aparece después de fin de S-expression	PUSH(R S) – RETAIN
h	',' aparece después de fin de S-expression	PUSH(R) – RETAIN
i	('' aparece después de fin de S-expression	PUSH (R S) – RETAIN
j	')' aparece después de fin de S-expression	ADVANCE

Parseando: Chop chop!

- |             |              |
|-------------|--------------|
| 1. \$ S     | ( a , , a \$ |
| 2. \$ R S ( | ( a , , a \$ |
| 3. \$ R S   | a , , a \$   |
| 4. \$ R a   | a , , a \$   |
| 5. \$ R     | , , a \$     |

6. \$ R S , , a \$
7. \$ R S , a \$
8. → Aparece ' , ' cuando se esperaba S-expression
9. POP – RETAIN
10. \$ R , a \$
11. \$ R S , , a \$
12. \$ R S a \$
13. \$ R a a \$
14. \$ R \$
15. → Falta ' ) '
16. EXIT

Después de parsear la hilera:



- 1: Aparece ' , ' cuando se esperaba S-expression
- 2: Falta ' ) '

Recuperación de errores- Ejemplo 2

Parsearemos la hilera ( ( a ) a ) a a

5. $S \rightarrow a$
6. $S \rightarrow ( S R$
7. $R \rightarrow , S R$
8. $R \rightarrow )$

	a	,	(	)	\$
S	1	a	2	b	c
R	d	3	e	4	f
\$	g	h	i	j	😊

Error	Mensaje	Acción
a	Aparece ' , ' cuando se esperaba S-expression	POP – RETAIN
b	Aparece ' ) ' cuando se esperaba S-expression	POP- RETAIN
c	S-expression incompleta	EXIT
d	Falta ' , '	REPLACE( R S)- RETAIN
e	Falta ' , '	REPLACE(R S) – RETAIN
f	Falta ' ) '	EXIT
g	'a' aparece después de fin de S-expression	PUSH(R S) – RETAIN
h	' , ' aparece después de fin de S-expression	PUSH(R) – RETAIN
i	'(' aparece después de fin de S-expression	PUSH (R S) – RETAIN
j	' ) ' aparece después de fin de S-expression	ADVANCE

1. \$ S
  2. \$ R S (
  3. \$ R S
  4. \$ R R S (
  5. \$ R R S
- (( a ) a ) a a \$
- (( a ) a ) a a \$
- ( a ) a ) a a \$
- ( a ) a ) a a \$
- a ) a ) a a \$



6. \$ R R a a ) a ) a a \$  
 7. \$ R R ) a ) a a \$  
 8. \$ R ) a ) a a \$  
 9. \$ R a ) a a \$  
 10. → Falta ','  
 11. REPLACE (R S) – RETAIN  
 12. \$ R S a ) a a \$  
 13. \$ R a a ) a a \$  
 14. \$ R ) a a \$  
 15. \$ ) a a \$  
 16. \$ a a \$  
 17. → 'a' aparece después de fin de S-expression  
 18. PUSH (R S) – RETAIN  
 19. \$ R S a a \$  
 20. \$ R a a a \$  
 21. \$ R a \$  
 22. → Falta ','  
 23. REPLACE (R S) – RETAIN  
 24. \$ R S a \$  
 25. \$ R a a \$  
 26. \$ R \$  
 27. → Falta ')'  
 28. EXIT

**Después de parsear la hilera:**

((a) a) a a  
 ↑ ↑ ↑ ↑  
 1 2 3 4

1. Falta ','
2. 'a' aparece después de fin de S-expression
3. Falta ','
4. Falta ')'

## Entretenimiento:

En esta parte el profe nos cuenta otra historia:

-Trabajé para Calderón, pero me pagaban.

En las elecciones de los 90's la Unidad estaba haciendo sus estadísticas, y querían compactar la información de 3 millones de personas en 20 megas, datos como: el nombre, apellidos, cédula, entre otros.

El profe parecía "sospechoso", pelo largo, carro con banderas verdes (de los Celtics) como cualquier persona decente.

Los guardas lo reportaron: "Ahí va un mae, pelo largo, todo raro, con banderas de Liberación, viene todos días, y las madrugadas también"

El profe hizo el programa que iniciaba con sonido, "el futuro es Calderón" en colaboración de Chepe. "Tututuuuu tutututuuuu"



El chiste Principal es que el profe puso su número en el programa, el día de las elecciones ocurría lo siguiente:

Aló! Es que tenemos problemas en la mesa 100.



Aaah! Yo no soy del partido



Aló! Es que una  
señora perdió su cédula...



¡Que no soy del partido!



Aló! Es que hay una seño...



¡Que no soy del partido!



Por cierto le quedaron debiendo al profe.