

Clase #21 de 27

Lenguajes Formales & Autómata de Pila

Octubre 3, Lunes

Agenda para esta clase

- Analizador sintáctico simple
- Sintaxis de las sentencias de C
- Ambigüed de la sintaxis de C

Analizador Sintáctico Simple

Analizador Sintáctico Simple

- Basado en ejercicio [K&R 1-24]
- Escriba un programa que verifique si un programa C posee errores sintácticos simples. Este programa es dificultoso si se lo diseña realmente genérico. Reglas a validar:
 - Paréntesis, corchetes y llaves desbalanceados
 - Apóstrofos y comillas, secuencias de escape
 - Comentarios
- Restricciones
 - Resolver con máquina de estados
 - ¿Es utilizable el módulo Stack visto en la calculadora de notación polaca inversa?
- Utilizar el símbolo \$ para la pila vacío
- Implementar con switch y goto
- Leer Autómata de Pila de [MUHC2012V2c2]
- Entrega
 - Definición formal del autómata de pila
 - Diagrama de transiciones
 - Parser.c
 - Stack.h
 - Stack.c

Sentencias ANSI C

Sentencia

sentencia

sentencia-expresión

sentencia-compuesta

sentencia-de-selección

sentencia-de-iteración

sentencia-etiquetada

sentencia-de-salto

sentencia-expresión

expresión? ;

Sentencia compuesta (¿C11?)

sentencia-compuesta

{ *lista-de-declaraciones?* *lista-de-sentencias?* }

lista-de-declaraciones

declaración

lista-de-declaraciones *declaración*

lista-de-sentencias

sentencia

lista-de-sentencias *sentencia*

Sentencia de selección

sentencia-de-selección

if (*expresión*) *sentencia*

if (*expresión*) *sentencia* **else** *sentencia*

switch (*expresión*) *sentencia*

Sentencia de iteración (¿C11?)

sentencia-de-iteración

while (*expresión*) *sentencia*

do *sentencia* **while** (*expresión*) ;

for (*expresión?* ; *expresión?* ; *expresión?*) *sentencia*

Sentencia etiquetada y de salto

sentencia-etiquetada

case *expresión-constante* : *sentencia*

default : *sentencia*

identificador : *sentencia*

sentencia-de-salto

continue ;

break ;

return *expresión?* ;

goto *identificador* ;

Ambigüedad en la definición sintáctica

Ejercicio de S&S: Sentencia if

6.8.4 Selection statements

Syntax

selection-statement:

```
if ( expression ) statement  
if ( expression ) statement else statement  
switch ( expression ) statement
```

Semantics

A selection statement selects among a set of statements depending on the value of a controlling expression.

Ejercicio de S&S: Sentencia if (cont.)

- S&S
 - Sintaxis
 - Restricciones para que tenga semántica
 - Semántica
- Dos ejemplos

```
if ( a )
f();
```



```
if ( b );
g();
else
h();
```

Sintaxis Concreta y Abstracta

- Sintaxis concreta
 - Específica de un LP
 - Tokens
 - Forma
 - Derivación según BNF
- Sintaxis abstracta
 - General
 - Conceptual
 - Necesaria para el análisis semántico
- Árboles de sintaxis concreta y abstracta del siguiente código

```
if ( a )
    if( b )
        g();
else
    h();
```

¿Conclusión?

- Dos derivaciones posibles
- Dos semánticas posibles
- Ambigüedad
- ¿Solución?
- Corrección de la S&S de la sentencia if
- Pragmática

```
if ( a )
    if( b )
        g();
else
    h();
```

```
if ( a ) {
    if( b )
        g();
} else
    h();
```

Ejercicio de S&S: Sentencia if – Resolución final

6.8.4.1 The **if** statement

Constraints

The controlling expression of an **if** statement shall have scalar type.

Semantics

In both forms, the first substatement is executed if the expression compares unequal to 0. In the **else** form, the second substatement is executed if the expression compares equal to 0. If the first substatement is reached via a label, the second substatement is not executed.

An **else** is associated with the lexically nearest preceding **if** that is allowed by the syntax.

Términos de la clase #21

Definir cada término con la bibliografía

- Parte I
 - Alfabeto de pila
- Parte II
 - Sentencia
 - Sentencia compuesta
- Parte III
 - Ambigüedad

Tareas para la próxima clase

1. Leer de [K&R1988] Sentencias.



¿Consultas?



Fin de la clase

Clase #22 de 27

Examen Diagnóstico y Caso Real de Expresiones

Octubre 17, Lunes

Agenda para esta clase

- Examen Diagnóstico
- Caso Real de Expresiones

Apellido y Nombre _____ Legajo _____ Equipo _____

- No se responden consultas durante la evaluación; por lo tanto, de ser necesario, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Resuelva en esta misma hoja; no se aceptan otras.

1	Un compilador sube el nivel de abstracción _____	V F
2	El preprocesador es parte de la compilación . _____	V F
3	Hay sentencias for que son semánticamente equivalentes a sentencias while _____	V F
4	Las transiciones de Mealy están formados sólo un estado origen y uno destino _____	V F
5	Las variables automáticas se inicializan automáticamente _____	V F
6	Un programa puede ser a la vez correcto semánticamente pero incorrecto léxicamente _____	V F
7	Las pilas agregan velocidad a los Autómata de Pila _____	V F
8	Los compiladores tienen dos ends: Front y Back _____	V F
9	Las ER son útiles para especificar la sintaxis de las declaraciones _____	V F
10	El Lenguaje C resuelve la ambigüedad de la sentencia if e if-else _____	V F

Caso Real: Analizar Error

```
asignarStringFix(str, leng); //invocación
```

```
private static String asignarString(String stringComp, int longitud){  
    try{  
        if (stringComp != null )  
            return stringComp.trim().substring(0, Math.min(stringComp.length(), longitud)).toUpperCase();  
        else  
            return "";  
    }catch(Exception e){  
        LoggerUtils.logOpMsg(log, Level.ERROR, e.getMessage());  
    }  
    return "";
```

- Analizar invocación
- Analizar definición
 - Analizar expresión del primer return
 - Precedencia
 - Orden de Evaluación
 - Efecto de lado
 - Árbol de expresiones.
 - Proponer por lo menos dos resoluciones
 - Identificar la causa raíz del problema.

Términos de la clase #22

Definir cada término con la bibliografía

- Parte II
 - Java
 - Función miembro
 - Paquete
 - Try
 - Catch



¿Consultas?



Fin de la clase

Clase #23 de 27

Resolución de Examen Final

Octubre 24, Lunes

Resolución de un Examen Final

Apellido y Nombre _____

Legajo _____

- El examen debe resolverse en tinta y en esta hoja; No se aceptan hojas adicionales.
- Durante el examen no se responde consultas; de ser necesario, escriba hipótesis de trabajo, las cuales también se evalúan.

Parte 1 - Ejercicios

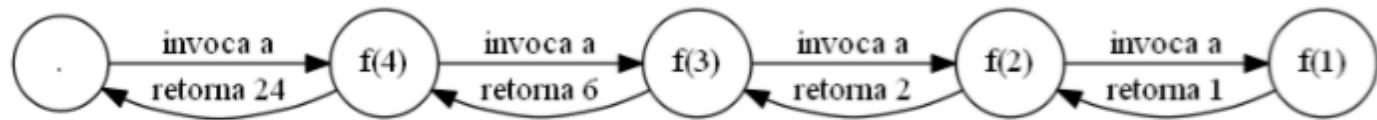
- 1 Sea la función `int f(int n){if(n==1) return 1;return n*f(n-1);}` Justifique:
 - el valor de la expresión **f(4)**,
 - cuántas invocaciones se realizan para obtener ese valor, contando **f(4)**,
 - y si las invocaciones tienen o no efecto de lado.
- 2 Dibuje un AF No Determinístico para la siguiente ER Extendida o metaER: **[0-2]{0,2}***
- 3 Indique cuales lexemas requieren invocar a **ungetc** durante el análisis léxico de ANSI C. Si alguna cadena no es lexema ANSI C, indíquelo con una equis (X).

Lexema	Requiere invocar a ungetc: Sí/No, ó X si no es lexema ANSI C
/	
procedure	
{	
42L	
<	

Parte 2 - Afirmaciones

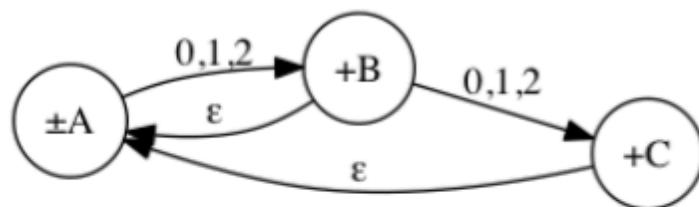
- 1 Sea **double b[2];** la expresión **sizeof b** se evalúa en tiempo de ejecución..... V F
 - 2 Los AFD reconocen la misma clase de LF que representan las metaER..... V F
 - 3 El operador **&&** tiene el orden de evaluación de sus operandos definido..... V F
 - 4 Sea **int a[]={3,4},i=0;** el efecto de lado de **a[i]=++i** está definido por ANSI C..... V F
 - 5 ANSI C define más de un árbol de derivación para **if(a)if(b)f();else g();**..... V F
 - 6 La sentencia **{int a(void); char a; a(a);}** es sintácticamente correcta..... V F
 - 7 Los prototipos tienen la misma sintaxis que las invocaciones..... V F
-

Una posible resolución del ejercicio 1
La función implementa factorial.



La función no utiliza variables externas ni tiene parámetros tipo puntero.
 $f(4): 24$.
Invocaciones: 4.
Efectos de lado: ninguno.

Una posible resolución del ejercicio 2



Una posible resolución del ejercicio 3

Lexema	Requiere invocar a ungetc: Sí/No, ó X si no es lexema ANSI C
I	Sí
/	Sí
procedure	Sí
{	No
42L	Sí
<	Sí

Afirmaciones

- 1 V
- 2 V
- 3 V
- 4 F
- 5 F
- 6 V
- 7 F



Fin de la clase