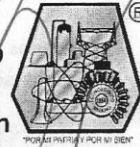




Instituto Tecnológico de Ciudad Madero

Departamento de Sistemas y Computación



Lenguajes y Automatas I

Unidad 1

M.S.I. Armando Becerra del Angel

Nombre: Luis Ricardo Reyes Villar

No. Control: 21076373

1.- Indica la respuesta en el paréntesis según corresponda:

- 1.1.- Esta fase del proceso de compilación, que se encarga de chequear que los componentes correspondan en tipo (c)
a) Analizador Léxico b) Analizador Sintáctico c) Analizador Semántico
- 1.2.- Fase del proceso de compilación que se encarga de chequear cada uno de los componentes se escriban correctamente (b)
a) Analizador Léxico b) Analizador Sintáctico c) Analizador Semántico
- 1.3.- Fase del proceso de compilación que genera una representación mnemotécnica, similar al lenguaje ensamblador (a)
a) Generación de código b) Optimización de código c) Generación de código intermedio
- 1.4.- Es la fase del proceso de compilación que utiliza variables temporales o auxiliares sin perder la esencia del programa fuente (c)
a) Generación de código b) Optimización de código c) Generación de código intermedio
- 1.5.- En esta fase del proceso de compilación se reduce el número de líneas de código sin perder la esencia del programa original (b)
a) Generación de código b) Optimización de código c) Generación de código intermedio

2.- Subraya la respuesta correcta según corresponda:

- 2.1.- Se encarga de convertir un programa fuente en un programa objeto equivalente.
a) Tabla de errores b) Tabla de símbolos c) Compilador
- 2.2.- Es una estructura de datos que contiene un registro para cada identificador.
a) Tabla de errores b) Tabla de símbolos c) Compilador
- 2.3.- Etapa del proceso de compilación que está íntimamente relacionada con el programa objeto.
a) Etapa inicial b) Etapa final c) Etapa intermedia
- 2.4.- Etapa del proceso de compilación que está íntimamente relacionada con el programa fuente.
a) Etapa final b) Etapa intermedia c) Etapa inicial
- 2.5.- Es la representación del conjunto de todas las cadenas posibles sin incluir la cadena vacía hasta infinito.
a) Cerradura de kleene b) Cerradura Mixta c) Cerradura Positiva

3.- Dados los siguientes alfabetos $\Sigma_1 = \{x, y\}$ y $\Sigma_2 = \{w, p\}$ indica la Representación de cada una de sus operaciones.

a) $\Sigma_1 \cup \Sigma_2 =$ b) $\Sigma_1 \Sigma_2 =$ c) $\Sigma_1^* \cup \Sigma_2 =$ d) $(\Sigma_2 \Sigma_1)^* =$ e) $(\Sigma_1^* \Sigma_2^+)^* =$

4.- Dados los siguientes lenguajes $L = \{a, b\}$ y $M = \{c, d\}$ indica la representación de cada una de sus operaciones.

a) $(LUM)^* =$
b) $LM^+ =$
c) $L^* \cup M =$
d) $(M^+)L =$
e) $((ML)^+)^* =$

4.- Indica como son tratados los siguientes enunciados por las fases del proceso Compilación.

a) $\text{Amor} = (\text{honestidad} + \text{honradez} + \text{lealtad}) / 2;$

b) $F = m * a + v / e;$

Buena Suerte!!!

3. $\Sigma_1 = \{x, y\}$, $\Sigma_2 = \{w, p\}$
- a) $\Sigma_1 \cup \Sigma_2 = \{x, y, w, p\}$
 - b) $\Sigma_1 \Sigma_2 = \{xw, xp, yw, yp\}$
 - c) $\Sigma_1^* \cup \Sigma_2 = \{\epsilon, w, p, x, y, \dots\}$
 - d) $(\Sigma_2 \Sigma_1)^* = \{wx, wy, px, py\}^* = \{\epsilon, wx, wy, px, py, \dots\}$

4. $L = \{a, b\}$, $M = \{c, d\}$
- a) $(L \cup M)^* = \{a, b, c, d\}^* = \{\epsilon, a, b, c, d, \dots\}$
 - b) $LM^+ = \{a, b\}^+ \{c, d, cc, cd, \dots\} = \{ac, ad, arc, ard, \dots, bc, bd, bcc, bcd, \dots\}$
 - c) $L^* \cup M = \{\epsilon, a, b, \dots\} \cup \{c, d\} = \{\epsilon, a, b, \dots, c, d\}$
 - d) $(M^+)L = \{c, d, \infty\} \{a, b\} = \{ca, da, \dots, cb, db, \dots\}$
 - e) $((ML)^+)^* = \{ca, cb, da, db, \infty\}$

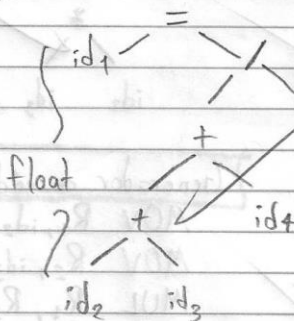
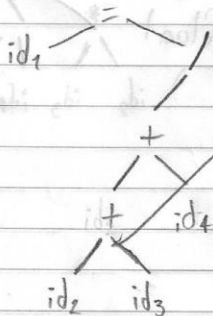
a) $\text{Amar} = (\text{honestidad} + \text{honradez} + \text{lealtad}) / 2;$

Análizador léxico:

$id_1 = (id_2 + id_3 + id_4) / 2;$

Análizador sintáctico:

Análizador semántico:



Generador de código intermedio

```

t1 = id2;
t2 = id3;
t3 = t1 + t2;
t4 = id4;
t5 = t3 + t4;
t6 = 2;
t7 = t5 / t6;
id1 = t7;
    
```

Optimizador

```

t1 = id2 + id3 + id4;
t2 = t1 / 2;
id1 = t2;
    
```

Generador de código objeto

```

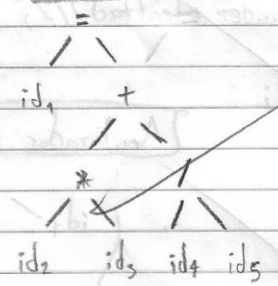
MOV R1, id2;
MOV R2, id3;
ADD R1, R2;
MOV R3, id4;
ADD R1, R3;
MOV R4, 2;
DIV R4, R1;
MOV id1, R4;
    
```

b) $F = m * a + v / e;$

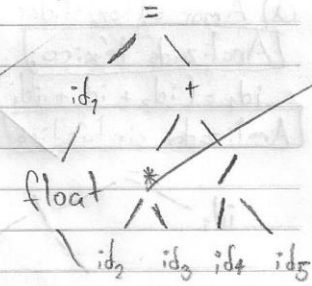
Analizador léxico

$id1 = id2 * id3 + id4 / id5;$

Analizador sintáctico



Analizador semántico



Generador de código intermedio

```

t1 = id2;
t2 = id3;
t3 = t1 * t2;
t4 = id4;
t5 = id5;
t6 = t4 / t5;
t7 = t3 + t6;
id1 = t7;
    
```

Generador de código objeto

```

MOV R1, id2;
MOV R2, id3;
MUL R1, R2;
MOV R3, id4;
MOV R4, id5;
DIV R4, R3;
ADD R1, R4;
MOV id1, R1;
    
```

Optimizador

```

t1 = id2 * id3;
t2 = id4 / id5;
t3 = t1 + t2;
id1 = t3;
    
```