**NOMBRE: ORTEGA BUSTAMANTE ANTONIO**

**GRUPO: ISC -7 AV**

**MATERIA: AUTOMATAS II**

**PROFESOR: PARRA HERNANDEZ ABIEL TOMAS**

**ESCUELA: INSTITUTO TECNOLOGICO DE IZTAPALAPA**

**TRABAJO: EXAMEN PRIMER PARCIAL**

****

**Ejercicio 11 cap. 2**

el algoritmo de minimización DFA dado en figura 2.9 está formulado para enumerar todos los elementos de P y todos los caracteres en cada iteración del bucle while

1. **refundir el algoritmo de modo que utilice un pool de trabajo para contener los conjuntos que todavía deben ser examinados**
2. **b) refundir la función de división para que particione el conjunto alrededor de todos los caracteres en E**

|  |  |
| --- | --- |
| T ← {DA, { D − DA} };  P ← ∅  while (P 6= T) do  P ← T;  T ← ∅;  for each set p ∈ P do  T ← T ∪ Split(p);  end;  end;  Split(S) {  for each c ∈ 6 do  if c splits S into s1 and s2  then return {s1  ,s2  };  end;  return S; | T ← {DA, { D − DA} };  P ← ∅  while (P 6= T) do  P ← T;  T ← ∅;  for each set p ∈ P do  T ← T ∪ Split(p);  end;  **while (estado 1=3)**  **{**  **Simbolo= get Simbolo ();**  **Estado= T [estado] [Simbolo]**  **}**  Split(S) {  for each c ∈ 6 do  if c splits S into s1 and s2  then return {s1  ,s2  };  end;  return S; |

1. **¿cómo la complejidad del caso esperado c lenguaje construye algoritmo original**

|  |
| --- |
| **T🡪 DA (D/ DA){**  **P 🡪 0**  **While (P 6=DA)**  **P 🡪 0**  **T =/ P**  **For ( p + T; t+ p; p++)**  **T= P;**  **}** |

**ejercicio 14 cap. 3**

Considere la tarea de crear un analizador para el lenguaje de programación

Esquema.

Contraste el esfuerzo necesario para un descenso recursivo de arriba hacia abajo

analizador con el necesario para un analizador Lr (1) controlado por tabla.

(Supongamos que ya tiene un generador de tablas Lr (1).)

|  |
| --- |
| Descenso de LR1 |
| LR 🡪 L+R  LR 🡪 L\*R  LR 🡪 L-R  LR 🡪 L+L  LR 🡪 L+L\*R  LR 🡪 L+L\*R-L  LR 🡪 R\*R  LR 🡪 R-R\*L  LR 🡪 R-R\*L-L |