

PROGRAMACIONPrograma2do. cuatrimestre 19721.- CONCEPTOS DE ALGEBRA

Conjuntos. Operaciones con conjuntos. Funciones y relaciones. Relaciones de orden y equivalencia. Conjuntos ordenados, Infinito y supremo. Pedes. Anillos. Anillos de Boole. Algebras de Boole. Funciones Booleanas.- Minimización. Diseño lógico.

2.- GRAFOS

Definición. Matrices definidas sobre anillos booleanos. Representación de relaciones por medio de matrices booleanas. Operaciones con relaciones: intersección, unión, composición, clausura transitiva. Componentes fuertemente conexas de un grafo. Relación de orden inducida. Aplicaciones.

3.- SISTEMAS FORMALES

Algoritmos y procedimientos. Funciones recursivas. Conjuntos recursivos y enumerablemente recursivos. Generadores, Aceptores. Traductores. Autómatas finitos, autómatas "Push Down" y máquinas de Turing.

4.- CONSTRUCCION DE MAQUINAS DE TURING

Definición de algoritmos. Representación de algoritmos mediante máquinas de Turing.

Técnicas de construcción. Submáquinas. Métodos gráficos. Ejemplos mostrando conceptos elementales de programación. Sistema "Turingia".

5.- DESCRIPCION DE "ALPHA"

Arquitectura. Codificación de la información. Estructura de las instrucciones.

Descripción de las instrucciones. Funcionamiento. Ejemplos.

6.- CODIFICACION DE LA INFORMACION

Números naturales. Representación de números naturales. Codificación.

Códigos ponderados. Bases de numeración. Eficiencia. Base binaria. Operaciones aritméticas.

Transformación de base.

Otros códigos: decimal codificado en binario, exceso de tres; Operaciones aritméticas.

Ejemplos de circuitos lógicos.

Información "no numérica": códigos standard.

Representación de información numérica en "punto flotante": normalización, pérdida de significación, rango de definición.

7.- CODIGOS AUTOCORRECTORES

Problema de errores. Códigos sistemáticos. Distancias de Hamming.

Códigos de paridad. Códigos cíclicos. Ventajas. Códigos ECF.

8.- DESCRIPCION FORMAL DE UNA COMPUTADORA

Instrucciones. Su estructura.

Distintos tipos de instrucciones: de transferencia de información, de control de secuencia de entradas y salidas, de proceso. Formalización del modelo de una computadora. El lenguaje "LEDESCOM".

9.- DIRECCIONAMIENTO

Modificación de la dirección. Direccionamiento directo, indirecto, implícito e inmediato.

10.- ORGANIZACION DE LA INFORMACION EN MEMORIA CENTRAL

Vectores, Matrices. Listas. Pilas. Stacks. Colas. Descriptores de información.

11.- ESTRUCTURA DE PROGRAMAS

Concepto de bloque. Puntos de entrada al bloque. Comunicación entre bloques.

Efectos colaterales. Zonas de memoria locales y globales.
Inicialización, cuerpo y finalización.
Estructura de programa en bloques. Direccionamiento relativo.
Registro base. Protección de memoria.
Bloques reentrantes. Linealmente reentrantes y no reentrantes.

12.- SUBRUTINAS

Utilidad de las subrutinas.
Diferentes tipos. Subrutinas abiertas y cerradas. Funciones.
Transferencia de argumentos y resultados: por dirección, por nombre y por valor.
Secuencia de llamada. Almacenamiento de la información de estado. Retorno.
Recursividad. Programación de rutinas recursivas. Eficiencia de la Programación: Consumo de tiempo y memoria. Efectos colaterales.
Reentrabilidad de la programación de subrutinas. Bloques de inicialización.

13.- LENGUAJES DE PROGRAMACION

Noción de gramática. Lenguaje generado por una gramática.
Distintos tipos de gramáticas.
El problema del parsing. Gramáticas ambiguas.
Semántica.
Compiladores e interpretadores.

14.- LENGUAJES SIMBOLICOS

Estructura de un lenguaje simbólico. Sentencias declarativas y operativas. Comandos al compilador.
El compilador: distintas fases, asignación de memoria, errores.

15.- PREPROCESAMIENTO DE PROGRAMAS SIMBOLICOS

El lenguaje de Macrodefinición:
definición de una macro,
definición de parámetros formales,

asignación de valores en tiempo de procesamiento, sentencias de control de preprocesamiento.

La tabla de Macrodefiniciones:

Código generado por el lenguaje de macrodefinición, Biblioteca de macros.

El preprocesador de macrodefinición: esquema de funcionamiento preparador. Posibilidades:

inclusión de macros en una macrodefinición o como parámetro, macrodefiniciones dentro de una macrodefinición, macrodefinición recursiva.

16.- LENGUAJES SUPERIORES

Lenguajes funcionales y algorítmicos. Definición. Historia. Formas usuales de formalización: sintaxis y semántica. Elementos básicos: identificadores, constantes, símbolos reservados. El concepto de atributo.

Lenguajes funcionales. Evaluación de funciones por sustitución. El cálculo " λ ".

Máquinas . LISP. Máquinas LISP. Procesamiento de listas.

17.- LENGUAJES ALGORITMICOS

Definición.

Atributos de un identificador:

representación de la información, estructuración de la información, asignación de memoria, acceso externo e interno, representación de información externa.

Nombre de subprogramas

Funciones básicas

declaración de atributos, manejo de estructuras, evaluación de expresiones simples, función asignación, alteración de la secuencia de ejecución, función de decisión, función de entrada/salida, función inicialización.

Estructura de los programas:

Conveniencia de la estructura en bloques
funciones de la estructuración en bloques, una idea general
sobre la generación de códigos y los problemas en tiempo
de ejecución.

Subprogramas

Puntos de entrada, rasaje de segmentos, devolución de valores
al punto de llamada, llamada recursiva.

Preprocesamiento de programas.

Ing. Esteban G. di Tada

Ing. Luis Trabb Pardo