



FACULTAD DE INFORMÁTICA
BIBLIOTECA

SEMINARIO SOBRE PLANES DE ESTUDIOS UNIVERSITARIOS EN INFORMATICA

Participantes: J. Bondía, X. Buxán, E. García Camarero, M^a Luz González, A. Guijarro, C. Marco Peris, P. Martín Yebra, M^a T. Molina, I. Ramos, A. Ríos Galindo, M. Rodríguez Ortalejo, J. L. San Emeterio.

PLANES DE ESTUDIO EN GRAN BRETAÑA

Por X. Buxán

Introducción, a modo de disculpa

Estas notas sobre los estudios de informática y computadores en la Gran Bretaña adolecen de vigor y profundidad, debido principalmente a la complejidad de los sistemas de enseñanza británica, en parte dada por la autonomía de los estamentos docentes y dentro de éstos por la libertad de los Colleges y Universities para trazar sus planes de estudios, métodos de trabajo,...

Por tanto, todo intento de hacer esquemas de la enseñanza global, de alguna manera se cae por el propio peso, al no poder marcar unas líneas generales, tan diferenciadas entre una materia y otra, en cuanto a enseñanza, como pudiésemos hacer en nuestra actual "Universidad Napoleónica". Asimismo que en los planes de estudio, pudiera decirse en titulaciones y limitaciones del ejercicio profesional; por tanto en muchos casos las definiciones serán tan sólo, por dar idea somera de lo que intento decir, pero no definición exhaustiva.

No obstante, deberán recordarse los dos caminos de educación superior en Inglaterra, el que trata de ser más teórico, propio de las Universities y el que tiene un fin más práctico de los "Colleges of Technology" que también tienen carácter universitario.

La enseñanza de la Informática en Inglaterra

Los estudios de computación e informática tratan de formar en Inglaterra una amplia gama de profesionales en estas materias y sus afines. Se hallan integrados en los planes generales de enseñanza Británicos, pudiéndose realizarlos tanto en centros oficiales como en privados. Estos estudios son supervisados (salvo los Universitarios, que dependen de la propia Universidad) por el "Coordinating Committees activities" que es un organismo de coordinación como su nombre indica y depende del departamento de educación, del ministerio de Tecnología, del Centro nacional de computación y de la Sociedad Británica de computadores; en general supervisa también los exámenes y los títulos conferidos que son los clásicos en el mundo de los ordenadores: Analista, Programador y Operador.

Si lo vemos desde el punto de vista del estudiante que desea introducirse en el campo de la informática, tendremos que para aprender las diversas tareas le son exigidos unos conocimientos mínimos. Así, aquellos que hayan finalizado la enseñanza primaria, no podrán hacer otra cosa que prepararse para perforistas y operadores de máquinas. Si han conseguido pasar el grado O'Level, podrán prepararse para perforistas y operadores, pero también para programadores Junior, Operadores de Computadoras, etc., como puede verse en el diagrama siguiente.

CALIFICACIONES PREVIAS		ESTUDIOS QUE SE REALIZAN
NO ESTUDIOS SECUNDARIOS		PERFORISTAS OPERADORES DE MAQUINAS
BACHILLER O'LEVEL	Sin Matemáticas	PROGRAMADORES JUNIOR OPERADOR DE COMPUTADORES
	Con Matemáticas	OPERADOR DE MAQUINAS APRENDIZAJE TECNICO
BACHILLER A'LEVEL Para ingreso en la Universidad	Sin Matemáticas	VENTAS ESTUDIOS COMERCIALES APRENDIZAJE COMERCIAL
	Con Matemáticas	PROGRAMADOR DE COMPUT. EST. PRACTICOS CON COMPUT. ESTUDIOS UNIVERSITARIOS EN COMPUTADORES ESTUDIOS EN LOS INSTITU- TOS TECNOLOGICOS

Como creo que está suficientemente claro, sólo matizaré un poco en qué consisten los 3 grados que se conceden en la enseñanza primaria y secundaria en el Reino Unido.

Los niños a los 10-12 años tienen dos opciones, o bien acabar los estudios primarios, para lo cual siguen en sus Schools, o bien van a las High'Schools o Grammars Schools, donde a los 16 años pasarán el examen O'Level; este ordinary level es una especie de reválida de Bachillerato en la que tienen que aprobar al menos 5 asignaturas. Luego, si desean ir a la Universidad habrán de pasar el "A" (Advanced) level, en el que se examinarán al menos de 2 asignaturas, con lo cual pueden tener acceso a alguna universidad. Estos exámenes los realizan ya las universidades. Luego hay otro "E" level, para los que saben mucho en alguna materia, que suelen implicar la entrada en la universidad.

Los estudios universitarios se hallan divididos en 3 ciclos; el 1er. ciclo dura en general 3 años en las universidades y 4 en los colleges de tecnología, pero aquí en general es en régimen sandwich, es decir que habrán de pasar parte del tiempo trabajando en empresas o casas comerciales.

Los cursos en computadores suelen estar constituidos en el 1er. ciclo por estudios sobre matemáticas y programación fundamentalmente en los primeros años Diseño lógico, Proceso de datos,... En general, los últimos cursos suelen ya ser de especialización y dependiendo de los departamentos de la universidad o colleges, pueden ir dirigidos desde Aplicación en matemáticas hasta Aplicación de ventas y marketing, pasando por Física, Ingeniería, Economía,...

Al final de estos cursos se concede el título de "Bachelor of Science".

Los Higher Degrees son cursos para estudiantes ya degrees y van dirigidos a aquellos que quieren completar su formación con miras a la investigación, o bien los que habiendo recibido una formación en Matemáticas, Física o Ingeniería,... deseen ampliar sus conocimientos sobre Informática.

Este tipo de estudios se realizan en Colleges universitarios o en centros de computación. Al final les conceden a los estudiantes el "Master of Science" y suele ser requisito previo el haber realizado algún trabajo de investigación.

Como creo que nuestro punto de vista desde este seminario tiene en cierta forma semejanza con este planteo de la enseñanza, veremos con algún detalle cómo son estos cursos.

Además para estos cursos suelen preferir a los estudiantes con buen bagaje de conocimientos de Matemáticas y Física y los tratan de preparar para el desarrollo de una tesis doctoral, creo que aumentan el interés para nosotros de estos cursos.

Como he dicho, se desarrollan los "Master Programs" durante un curso que suele durar un año académico al menos, siendo bastante común que además vaya precedido por otro de introducción a los ordenadores, dado que muchos de los estudiantes no han hecho estudios previos en computación y además un curso, que suele ser de verano y cumple con el objetivo de seleccionar los estudiantes del año siguiente, ya que las plazas suelen ser muy limitadas.

Aunque los planes de estudio y métodos de trabajo de las diferentes universidades difieren bastante, creo que es instructivo ver el desarrollo de este curso en alguna universidad; he tomado como ejemplo al Imperial College, de Londres, debido a que es de los que más se parece al nuestro, en mi opinión.

Imperial College of London.- Programa detallado

El curso se desarrolla en el Centro de Computación y Automatización, que no depende de ningún departamento específico; en él se desarrollan dos cursos, uno de control de sistemas, que tal vez fuese interesante para los Físicos verlo con detalle, y otro más, digamos "Matemático" de ciencia de computadores.

Veamos el programa desarrollado durante el curso 67 (que no difiere mucho del 71-72), durante el cual disponía el centro de los siguientes ordenadores: un IBM 7094, IBM 1460, IBM 1401, también tenían acceso al Atlas de la Universidad de Londres, un PDP-7.

De entrada, los alumnos realizaban un curso de selección y preparación en el verano, en el que eran preferidos los que tenían conocimientos de Lógica y Matemáticas.

Luego, la primera semana de curso la dedicaron en plan intensivo al aprendizaje de un lenguaje de Alto Nivel, el Fortran IV, en 30 horas de clases.

Análogamente, la segunda semana con el régimen anterior, la dedicaron al aprendizaje de un lenguaje Assembler y a partir de esta semana comenzó un primer período de formación, en el que se dictaron las siguientes asignaturas:

MATEMATICAS. Fundamentos: 20 clases.

Teoría de conjuntos, Algebra de Boole, Simplificación de función, Mapas de Karnaught, Cálculo, Cálculo de predicados, la Notación de Lambeln, e Interpretación de Lógica Axiomática.

ORGANIZACION DE COMPUTADORES. Duración: 20 clases.

Estructura de computadores, Representación de la información, Códigos de datos e instrucciones, Circuitos de funciones básicas, AND, OR, NOT, Unidad Central de proceso y control, Aspectos avanzados de comprobación, fiabilidad y pruebas.

ESTRUCTURAS DE DATOS. Duración: 20 clases.

Representación de los datos en máquina, palabras, carácter, organización, almacenamientos, color, listas.

SISTEMAS OPERATIVOS. Duración: 10 clases.

Estructura General de D. S.

La 2ª y 3ª parte del curso comenzó a partir de las vacaciones de Navidad hasta el mes de julio y durante ella se desarrollaron las siguientes asignaturas:

COMPILADORES. Duración: 20 clases.

Organización, análisis lexical, recorrimientos de stands, análisis de expresión, código objeto, diseño y generación de cargadores y conectadores, paquetes recursivos, compiladores automáticos.

TEORIA DE COMPUTABILIDAD.

Máquinas de Turing, funciones calculables por una máquina, funciones recursivas, Numeración de Gödel, Predicados computables y decidibles, Teorema de Kleene. Otros modelos.

APLICACIONES NUMERICAS. 10 clases.

Evaluación e Interpretación, cuadraturas, curvas finitas, aproximación de funciones, Sistemas algebraicos, Ecuaciones diferenciales.

APLICACIONES INDUSTRIALES Y COMERCIALES.

Principios, sistemas de análisis: diseño y aplicaciones de instalaciones y terminales de equipos.

Aplicaciones a: Camino crítico, simulación, negocios, teleproceso, juegos, control de producción, sistemas de información.

GRAFOS. 10 clases.

Teoría de Grafos, aplicación a la arquitectura y astilleros.

APLICACIONES A HUMANIDADES

Proceso de textos, lingüística, análisis sintáctico, proceso de textos históricos, proceso de textos musicales, análisis y composición.

TECNICAS DE SIMULACION

Tipos de simulación, estructura de tiempo, decisión discreta.

AVANZADOS LENGUAJES ASSEMBLER: 20 clases

LENGUAJES DE ALTO NIVEL: 30 clases

SINTAXIS Y SEMANTICA DE LENGUAJES: 20 clases

SISTEMAS EN TIEMPO REAL: 30 clases

AVANCES EN DISEÑO DE COMPUTADORES

Seminario sobre Inteligencia Artificial

APENDICE

Del B.C.S. Annual educational Review, 1967 ("Working Party") está tomado el siguiente programa detallado para "ayudar y estimular" el establecimiento de "Ciencias de Computadores en Colegios y Universidades". Algunos son propios para undergraduates y otros para postgraduates, pero en conjunto espero que será muy útil con los fines de este seminario.

INTRODUCCION A LA CIENCIA DE LAS COMPUTADORAS

Tiempo y prerrequisitos: De 20 a 40 horas, de las cuales el 50% deben ser de prácticas.

Para la primera mitad del curso los programas deben ser generados con fines de probar el sistema; también al principio es aconsejable que trabajen individualmente y luego formen grupos con vistas a integrarse en algún tipo de estudios.

Programa

Introducción a los lenguajes de programación y al ordenador (Debe permitirse a los estudiantes hacer sus propios programas mostrándoles las facilidades de entrada, salida, operaciones aritméticas, almacenamiento, control de programas). El lenguaje debe ser apropiado a la enseñanza, si fuese necesario, debe ser un lenguaje simple que dé facilidades para el reconocimiento de sus propios errores. El lenguaje debe ser introducido lentamente y matizadas las restricciones de entrada-salida.

Uso de diagramas de flujo. Descripción de problemas matemáticos con diagramas de flujo. Uso de los Símbolos Standard In- gleses. Descripción del sistema operativo local para probar programas. Flujo de datos a través del sistema local. Incorporación de instrucciones de modificación y de matrices de almace- namiento, Consideraciones económicas.

Documentación de un programa completo; esto debe hacerse si- guiendo las siguientes líneas:

1. título
2. indexación
3. descripción del programa
4. métodos de uso, detalle de la especificación de preparación de datos, operatividad de las instrucciones procederas para resultados en formatos y errores
5. análisis matemático y análisis de procedimiento
6. diagramas de flujo
7. programación
8. pruebas.

Problemas no numéricos; descripciones de códigos binarios, ope- raciones con números y caracteres, operaciones lógicas, métodos de ordenación. Descripción y uso de unidades periféricas. Dise- ño de formatos de entrada-salida, consideración sobre errores en los datos. Uso de subrutinas, procederes y métodos de progra- mación más sofisticados. Descripción de procederes operativos en uso en el sistema local para pasar programas.

Referencias:

- British Standards 4058. Flowchards 1966 y 3527, vocabulary 1962.
 Gruenberger F. y Jaffray G.- Problems for computer solution.
 Wiley 1965.
 Merchant and Pegg.- Digital Computers. A practical Approach.
 Blackie 1967.
 Nicol. Elementary Programming and ALGOL. McGraw Hill.

PROGRAMACION BASICA DE COMPUTADORAS

Prerrequisitos y duración: El curso debe durar 200 horas, de las cuales el 70% deben ser prácticas. Cuando se comienza el aprendizaje de lenguajes debe dársele a los estudiantes facilidades para que preparen y corrijan sus propios programas. Los estudiantes ya deben estar familiarizados con las técnicas básicas y debe durar un año simple el curso.

Programa

Los estudiantes deben aprender sucesivamente 3 lenguajes. El primero debe ser como Fortran o Algol, luego otro tipo COBOL y al final un lenguaje assembler. Deben pasar programas y familiarizarse con el sistema operativo, que deben ir completamente documentados, para lo que es aconsejable que sigan las leyes de las siguientes líneas:

1. título y descripción del programa
2. indexación
3. métodos de uso:
 - a) preparación de datos
 - b) instrucciones de operación
 - c) descripción de formatos de entrada/salida
 - d) error de los procedures y técnicas de restauración
4. análisis matemáticamente
5. análisis de procedures
6. diagramas de flujo
7. programación
8. pruebas

Debe también matizar el uso de diagramas de flujo, y el uso económico del computador en almacenamiento y otros aspectos, tales como tiempo de programación, etc., uso y apreciaciones sobre los periféricos y almacenaje medio asociado con el computador local.

Estructura de Datos. Palabras, discos, ficheros, matrices, listas árboles, códigos de caracteres, procesos de traducción, pruebas de validez, direccionamiento indirecto, direccionamiento simbólico, métodos de ordenación interna, ordenación radial, selectiva por cambio. Métodos de ordenación externa, por inserción, por mezclaje, en varias fases, en cascada. Uso de subrutinas, procedures y macros, también de lugar y división de los programas. Técnicas características y diseño de ensambladores, interpretadores y compiladores.

Referencias:

Arden B.W.- Introducción a la computación digital. Addison-Wesley. 1963.

Galler B.A.- Lenguajes de los Computadores, McGraw Hill. 1962.
 Iverson.- A Programming language. Wiley. 1962.
 Randel B. and Russell L.- Algol 60. Implementation. Ac. Press
 Wegner P.- Introduction to symbolic Programming. Ac. Press.

ORGANIZACION DE COMPUTADORES Y PROGRAMACION DE SISTEMAS

Prerrequisito: Curso básico en Ciencia de la Informática.
 Duración: Sobre 40 horas.

Programa: Organización del Hardware de un computador, propiedad de los sistemas de almacenamiento, flujo de información a través del sistema, métodos de control periférico, propiedades de las unidades periféricas.

Organización del software de un computador: técnicas de traducción, notación polaca. Métodos de optimización, localización de los almacenamientos, estructura de bloques, rutinas, funciones, compiladores; intérpretes, sintaxis directa de compiladores; generadores, aplicación de expresiones, Lenguajes intermedios, técnicas de trapping.

Tiempo compartido en un programa simple, entrada/salida autónoma operada por instrucciones básicas, uso del control de interrupciones, solapamientos de entrada, paso de programas, paso a bajo control del sistema monitor, uso de reloj y del contador de instrucciones, solapamiento en salida y computación de diferentes programas, previsión de facilidades de Sistemas multiprogramados, compartición del procesador central entre varios programas, métodos de conmutación entre programas, espacio de localización, colas de trabajo y tipos por lotes de almacenamiento, métodos de transferencia a los almacenamientos por lotes, sistemas de interrupción.

Prueba de programación Hardware: Fallos en máquinas simples, pruebas de sistemas de multiprogramación on line, recuperación después de una condición de fallo. Prueba de programación en software, interface con hardware, volcadores de memoria, alteración y trazado de localización de valores y secuencia de control. Traps lógicos y fallos en programas.

Sistema de accesos on line: Aspectos de programación de acceso directo, control directo desde la consola, sistemas de acceso directo multifase, sistemas de multicomputador.

Referencias:

Corbato F.J.- The multics System Fall Join Computer conference. 1965.
 Fisher and Swindle G.- Computer Programming Systems. Holt-Rinehat and Wilson. 1964.
 Heistand R.- An executive System. ACM 1964.
 Wegner P. Introduction to System Programming. Academic Press. 1964.
 Bucholz W.- Planning a computer System. McGraw Hill. 1962.

ESTRUCTURAS DE INFORMACION Y PROCESAMIENTO

Tiempo: 20-40 clases.

Prerrequisitos: 1 curso de computación básica.

Programa: Información sobre el almacenamiento en máquina: Sistemas de almacenamiento de acceso inmediato, tambores, discos, cinta magnética y tarjetas.

Estructuras de información lineal, escalas numéricas, representación en punto fijo y flotante, palabras y matrices, almacenamiento no numérico, registro de longitud fija y variable.

Procesos de estructuras de información lineales, Notación de Iverson, editing, sorting, merging. Consideraciones máquina, accesos secuenciales y en paralelo, control de error.

Estructuras de información compleja, listas, árboles, estructuras de listas y su almacenamiento en máquina.

Procesamiento de información compleja; lenguajes de procesamiento de listas LISP, IPLV, SLIP, con estudio especial y uso práctico de al menos un lenguaje, Manipulación de cadenas en lenguajes especiales COMIT, SNOBOL, TRAC con estudio especial de uno al menos y su uso práctico.

Estructuras transformadoras de información y procesos deductivos.

El almacenamiento y recuperación de material de librería, manipulación algebraica, formal diferenciación e integración, Introducción a la programación heurística.

Referencias:

- Iverson.- A programming Language. John Wiley & Sons.
 Bucholz.- File organization and addressing. IBM Journal . June 1963.
 McCarthy.- LISP 1.5 Programmers Manual. MIT Press. 1962.
 Newell.- Information Processing Language. Prentice Hall. 1964.
 Weizensbem J.- Symetric list processor. A.C.M. 1963.
 Yngue V.- COMIT Programers Manual. MIT Press. 1963.
 Faber D.S. Criswold and Polonsky.- The SNOBOL 3. Bell System Journal. 1966.
 Moorers C.W.- TRAC - a procedure describing language for the seacture tipewriter. ACM, 9,3. Marzo 1966.
 Bobrow D. G. and Raphael B.- A comparison of list processing computer languages. ACM, 7,4. April 1969.
 Browl and Iverson.- Automation Data processing. John Wiley & Sons. 1963.
 Slage.- Experiments with a deductive question answering program. ACM 8, 12. Dec. 1965.
 Bobrow J. D.- A question answering system for high School algebra word problem. Fall Joint Conference 1964.

COMPILADORES, Implementación y diseño

Prerrequisito: Un curso básico de Computadores.
Tiempo: unas 80 clases.

Programa: Teoría de Gramáticas: Chomsky tipo 0, 1, 2, 3. Sintaxis específica por BNS y variantes. Gramáticas admitiendo un algoritmo especial de Parsing tal como análisis de procedencia (4 clases).

Estructura de lenguaje y contenido: Lexicografía y sintaxis, rutinas, bloques, sentencias componentes, declaraciones, tipos de parámetros formales, condiciones. expresiones, tipos de datos primitivos, estructuras de los tipos de datos, paralelismo. Mayores facilidades en definición, macros, Entrada/Salida y operaciones con las unidades periféricas. Supervisión de operaciones, trapping por fallo (22 clases).

Estructura de un traductor

1. Análisis lexicográfico y edición de las unidades sintácticas elementales, tales como identificadores, ctes. ...

2. Métodos de análisis sintáctico, Análisis sintáctico controlado: métodos basados en la precedencia de operadores y otros casos especiales de los que Floyd llama basado en el contexto sintáctico, Definiciones gramaticales para simplificar esos métodos.

3. Procesamientos de los resultados del análisis. Esto puede ser algún tipo de la notación libre de paréntesis, que puede ser interpretada directamente o bien convertida en código máquina. Compilador-Compilador, FSL, COGENT. Tablas auxiliares: Identificador y listas propiamente dichas, Métodos de combinación.

4. Tratamiento de macros. Estructura del programa objeto: el lugar de almacenaje. Bloques y rutinas administrativos, Transferidores de control, Aritmética elemental, ciclos, Matrices, I/O y sus rutinas, operaciones en periféricos, Trap por fallos. Lenguajes de propósito especiales. (Lenguajes de recuperación de información. Lenguajes on line). (22 clases).

Referencias:

- Cheatham.- Theory and Construction of Compilers. Computer Associates. 1966.
- Bolliet L.- Compiler Written techniques. Institutes des Mathématiques Appliquées. 1966.
- Godman.- Varios artículos en Annual Review in Automatic Programming.
- Wegner P.- Introduction to System of Programming Academic Press. 1964.
- Theichow and Lubin.- Computer simulation. ACM 9, 10. Octubre 1966.
- Cotin A. S. Multiple variate Counter. University of London. Inst. Computer Science.
- Griffiths and Petrick.- On the relative Efficiencies of Context free Grammar Recognitions. ACM. May 1965.

TEORIA DE INFORMACION

Requisitos: Algunos conocimientos de Algebra abstracta.
Duración: 40 clases.

1. Teoría de Códigos, canales de información, tipos de clases teóricas.

Detección de errores y corrección, máximo rendimiento del código.

Registros binarios, utilidad para la producción de secuencias binarias o pseudo aleatorias y para su decodificación.

Códigos especiales, códigos de bloques, distancia de Haming, grupos de códigos lineales, teoría de Slepian, Estructura de grupos y códigos cíclicos.

Secuencias binarias, Generación de funciones, Secuencias por división. Identidades fundamentales de Selmer's, relaciones de recurrencia. Método clásico. Períodos y propiedades. Secuencias pseudoaleatorias.

Códigos especiales, códigos útiles de corrección de errores, códigos de Haming, Reed-Muller, Iterados, etc.

Referencias:

- Colomb S. W.- Digital communications with space Applications. Prentice Hall. 1964.
Peterson W.- Error correcting codes. MIT Press. 1961.
Selner E.S.- Linear recurrence over finite fields. University Bergen. 1966.

2. Teoría de Conmutación

Idea de máquina secuencial como procesador de tiempo o espacio de sucesiones de dígitos. Requerimientos para la combinación, secuenciación y almacenamiento de los Patters de dígitos.

Combinaciones lógicas y medidas de complejidad para redes combinaciones y contadores con puertas. Métodos de descripción, sintetización y simplificación, funciones combinacionales. Azar, máquinas de May, máquinas de Moore. Número de estados en un circuito secuencia, proceso de estados. Diagramas de estados. Producción en la máquina reducida a forma canónica. Asignación de estados Flip Hops.

Acomodación de señales no ideales. Sincronismo, asincronismo y rapidez independiente del circuito. Teoría de Hoffman de circuitos asincrónicos. Teoría de Muller's de independencia de la rapidez. Post álgebra y Eichelberger's ternaria, descripción lógica.

Bibliografía:

- Mc Clusberg.- Introduction to the Theory Switching Circuits. McGraw Hill. 1961.
 Meay.- A method for syntacting sequential circuits. Bell System Journal 34.
 Curtis.- Design of Switching Circuits. Van Nostrand. 1962.

LOGICO DISEÑO Y TEORIA DE CIRCUITOS

Duración: 40 horas.

Requisitos: conocimientos matemáticos y algo de electricidad o electrónica.

Programa: Algebra de Boole, postulados, teoremas, formas canónicas, representación de funciones y simplificación por métodos y gráficos tabulares, realización lógica. Tipos de puertas y métodos de sistemas apropiados. Hazards, circuitos, formas especiales, Elementos biestables, descripción lógica y aplicación a contables y registros. Operadores aritméticos, secuencias, estaciones, métodos de multiplicación, máquinas secuenciales, Modelos de Moore y Mealy. Síntesis de procedimientos para máquinas síncronas y asíncronas.

Referencias:

- Chu.- Digital Computer Design Fundamentals. McGraw Hill. 1962.
 Fleg H.G.- Boolean Algebra Blakie. 1964.
 Miller.- Switching theory. Wiley. 1965.

DISEÑO LOGICO Y TEORIA DE CONMUTACION

Duración: 15 a 20 clases.

A continuación del curso I.

Programa: Funciones especiales de Boole y sus implicaciones en la ingeniería, clasificación de acuerdo con el circuito. Lineales y retroalimentados, registros. Refinamientos en la aritmética. Máquinas secuenciales, teoremas de estado mínimo, especificación incompleta, problema de asignación de estados, expresiones regulares. Diseño para rehabilitación, redundancia, Nivel de operatividad de circuitos.

COMPUTACION ANALOGICA E HIBRIDA

Duración: 60 clases con un trabajo de prácticas.

Prerrequisito: conocimientos generales de Ingeniería o Matemáticas, incluyendo lenguajes de programación digital.

Programa: Elementos lineales, amplificadores, potenciómetros, resolución de ecuaciones diferenciales lineales con coeficientes constantes. Control de Entrada/Salida a través de periféricos; modos de control y circuitos, análisis lineal y el problema de las escalas: tiempos, amplitud de escala y frecuencia de aproximación; análisis de los operadores lineales.

Elementos no lineales, Multiplicadores, diodos, redes, técnicas para engendrar funciones generadoras..., Ecuaciones diferenciales no lineales.

Análisis de los errores, prueba estática y dinámica, estabilidad, ruido, características de evaluación, técnicas de reducción de errores.

Técnicas analógicas avanzadas, generaciones repetitivas, simulación directa, optimización y control de sistemas.

Sistemas de computadores híbridos. Computadores analógicos con modelos lógicos, con o sin dependencia de computadores digitales. Sistemas mezclados con tiempo compartido. Conversión analógica digital y digital-analógica, interfaces híbridos.

Técnicas de aplicaciones híbridas. Métodos iterativos generales, contornos y valores iniciales. Ecuaciones diferenciales. Técnicas de control y optimización.

Referencias (con comentarios):

Tifer S. Analogue Computing.. Mc-Graw-Hill. 1961. Referencia Standard.

Tomovic and Karplus.- Computers Analog High Speed. Wiley. 1962. (Excelente en análisis de errores).

Rogers and Connelly.- Analog computation in Engineering Design. Stewart and Atkinson.- Basic Analogue Computer Technical.

McGraw-Hill. 1967. (Buen libro y barato).

LOGICA MATEMATICA

Tiempo: 40 clases.

Prerrequisito: Unos conocimientos matemáticos a nivel de una graduación estandar en Ingeniería.

Programa: Lógica Matemática: Introducción, álgebra de conjuntos, álgebra de Boole, Algebra proposicional. Notación de Hilbert y Lukasiewics, conexión, dualidad y formas normales.

Cálculo proposicional. Concepto de fórmula, deducción de Teoremas, fórmula igualverídica, consistencia y completitud del cálculo proposicional.

Lógica de predicados: predicado, cuantificadores, axiomas en forma normal, restricción de cálculo de predicados.

Aritmética recursiva y autómatas finitos, primitiva y fórmula general recursiva, Máquinas de Turing, computación de fórmulas recursivas por máquinas de Turing, conjuntos recursivos y predicados, λ -notación y cálculo de Church, Autómatas finitos y tablas de estados, circuitos síncronos secuenciales, formas reducidas.

Referencias:

- Church A.- Mathematical Logic. Princeton. 1965.
 Kneebone G.- Mathematical Logic and Foundations of Maths. Van Nostrand. 1963.
 Gill A.- Introduction to theory of finite state Machines. Mc Graw-Hill. 1962.
 Mendelson E.- Introduction to Mathematical Logic. Van Nostrand. 1966.

METODOS NUMERICOS

Duración: 40 clases.

Prerrequisitos: Curso en computación.

Programa: Algebra Lineal, matrices, métodos directos e iterativos para resolución de ecuaciones e inversión de matrices. Teoría de autovalores y vectores propios.

Métodos de Aproximación: Aproximación de polinomios por el método de las diferencias finitas de Lagrange, aplicación a la interpolación, diferenciación numérica, integración, polinomios ortogonales, método de los mínimos cuadrados.

Soluciones de ecuaciones no lineales, resolución sucesiva, regla de la falsa posición, método de Newton-Raphson, de Baristow's y Muller, procesos δ^2 de Aiken, análisis de convergencia y métodos iterativos.

Ecuaciones diferenciales ordinarias, métodos analíticos, métodos de Picard y series de Taylor, métodos simples de varios pasos, método de la corrección de diferencias. Conjuntos de ecuaciones simultáneas y ecuaciones de mayor orden. Métodos para ecuaciones de 2° orden. Estabilidad y problemas de contorno. Soluciones aproximadas usando el método de las diferencias finitas, correcciones.

Introducción a las ecuaciones en derivadas parciales. Ecuaciones cuasilineales de 2° orden y discusión en funciones de las condiciones de contorno, pares simultáneos de ecuaciones de 1er orden.

Aproximación por diferencias finitas. Resolución de ecuaciones Tipo elíptico. Integración paso a paso para las ecuaciones parabólicas. Aplicación elemental para obtención de soluciones por el método de las características para ecuaciones hiperbólicas.

Referencias:

- Hildebrand.- Introduction to Numerical Analysis. Mc Graw Hill. 1956.
 Ralston.- A first course in Analysis. Mc Graw Hill. 1965.
 Noble B.- Numerical Methods. Oliver & Boyd.
 Bull G.- Computational Method with Algol. Harrap. 1966.

ANALISIS NUMERICO

Análisis de error en álgebra lineal, normas, vectores y matrices.

Matrices condicionadas, vectores propios y autovalores. Métodos computacionales para obtención de autovalores.

Ecuaciones diferenciales ordinarias. Propagación de errores por métodos de variables discretas. Estabilidad y convergencia, estudio teórico y propagación de errores de redondeo.

Ecuaciones en derivadas parciales y tratamiento de ecuaciones elíptico. Problemas de convergencia y estabilidad de ecuaciones parabólicas. Elaboración del método de las características. Introducción al problema del choque de ondas. Valores propios. Tratamiento de singularidades.

Ecuaciones Integrales: métodos para la resolución de ecuaciones integrales.

Referencias:

- Fox.- Numerical solution of ordinary and Partial Differential Equations. Pergamon Press. 1962.
 Varga R.- Matrix Iterative Analysis. Prentice Hall. 1962.
 Wendroff B.- Theoretical Numerical Analysis. Academic Press. 1966.
 Wals Joan.- Numerical Analysis.

PROCESO DE DATOS ESTADISTICOS

Duración: 40 horas y 20 de trabajos prácticos.

Prerrequisito: un curso de estadística.

Programa: Análisis de datos: muestreo, colecciones de datos, codificación y pruebas. Técnicas descriptivas: Gráficas, tablas y otras salidas. Análisis multivariante, aspectos computacionales, formación de la matriz de covarianza, matrices de correlación, análisis de la componente principal, análisis del discri-

minante. Análisis de la varianza. Diseños simples, cuadros latinos. Falta de observación por el uso de la matriz de diseño. Análisis de regresión, regresión multilíneal con inclusión y exclusión de variables, regresión no lineal. Análisis de series de tiempo. Análisis espectral. Promedio de movimientos de variaciones cíclicas.

Referencias:

- Yates.- Sampling Methods for Censuses and Surveys. Griffin and Co. 1959.
 Kendall.- A course in Multivariate Analysis. Griffin and Co. 1957.
 Davies.- Statistical Methods in Research and Production. Oliver and Boyd. 1957.
 Smillie.- An Introduction to Regression and Correlation. Ac. Press.

PROCESO DE DATOS

40 clases con trabajos prácticos.
 Previo un curso de introducción a los computadores.

Sumario:

Ficheros, paquetes y discos. Ficheros de entrada. Fichero maestro y Fichero de transacciones. Diagramas de flujo. Introducción al Cobol, ficheros de tarjeta perforada, Perforación de datos a procesar, Perforación de llaves y verificadores...

PROCESO DE DATOS APLICADO AL COMERCIO

Un curso de 30-40 horas.
 Prerrequisito: Un curso de proceso de datos y otro de programación Cobol.

Sumario:

Sistemas de negocios, estudio de los diversos tipos, concepto de sistema integrado. Sistemas de análisis, diseño, documentación, pruebas, management, científicos. Planificación y producción. Programación lineal; implementación del simplex, métodos del camino crítico y programación heurística.

Referencias:

- Brandon.- Management Stands for Data Processing. Van Nostrand.
 Burton and Mills.- Electronic Computer and theirs Business Applications. Ben. 1960.

INVESTIGACION OPERATIVA

30-40 clases y/o un trabajo de 1/2 a 1 año de duración.
Requisito: curso de estadística.

Programa: Debe ser un curso introductor a la investigación operativa y a los métodos matemáticos básicos. Programación matemática, análisis de redes, caminos críticos, colas e inventarios, y métodos de simulación.

MÉTODOS DE OPTIMIZACIÓN MATEMÁTICA

Este curso debe durar sobre 1 año y los estudiantes deben realizar algún tipo de trabajo práctico.

Programa: (Los algoritmos deben ser dados de tal forma que los estudiantes los pasen por ordenador).

Descripción del problema de optimización, local y global.

Minimización de una función de variable simple, interpolación.

Métodos clásicos de optimización, funciones cuadráticas, generalización del método Newton-Raphson. Métodos usando sólo la 1ª derivada, Métodos de variables métricas, Davidson y otros. Construcción del mínimo, reducción a la forma inconstruible. Multiplicadores de Lagrange y otros.

Programación Lineal. El problema de programación lineal, formas canónicas, espacios de vectores y conjuntos convexos. Método del simplex, variables artificiales, degeneración, ciclaje, perturbación. Problemas de dualidad. Método del simplex revisado.

Problema del transporte y programación entera. Programación cuadrática y funciones convexas. Teoría de Kunh-Tucker.

Programación dinámica, principios de optimización, ecuaciones funcionales y procedures computables. Localización de procesos unidimensionales y multidimensionales. Multiplicadores de Lagrange.

Referencias:

Bralakrishnan and Neustad.- Computers Methods in Optimization problems.

Dantzing.- Linear Programing and Extension.

Gass S.- Linear Programing. Interciencia. Mexico.

Menhauser.- Introduction to Pinamic Programing.

PROGRAMACION EN TIEMPO REAL

Tiempo: 30 clases.

Prerrequisitos: Conocimientos básicos en Computación.

Sumario:

Aplicación de la computación a sistemas en tiempo real. Sistemas on-line y multiaccesos trabajando problemas de aplicaciones comerciales, industriales y transportes.

Bibliografía:

Martín.- Programing Real-time Computer Systems. Prentice Hall. 1965.

Desmonde W. Real time Data Processing Systems. Introductory concepts. Prentice Hall.