

**BOLETIN
DE LA
SOCIEDAD ARGENTINA
DE
CALCULO**

AÑO 2 - 6

MAYO 1962

SUMARIO

NOTICIAS DEL EXTERIOR Pág.

Centro Internacional de Cálculo. -----	3
Noticias Varias. -----	4

NOTICIAS DEL PAIS

Jornadas de la S. A. C .-----	7
Noticias Varias -----	8
Estado Actual de las Investigaciones en el Campo de la Inteligencia Artificial -----	10
Seminario -----	18

LAS EMPRESAS

Aprovechamiento de un Sistema Electrónico orientado a Cintas Magnéticas para Operaciones con Máquinas y/o Resolución de Sistemas Lineales de Inecuaciones (Simplex), por J. Chamero, IBM Argentina -----	19
Sistemas Ejecutivos, por A. Pérez, Remington Rand Sudamericana 28	

ECCION BIBLIOGRAFICA

Comentario de Libros-----	33
---------------------------	----

LEXICO

Terminología (Cont.) -----	37
----------------------------	----

NOTICIAS DEL EXTERIOR

CENTRO INTERNACIONAL DE CÁLCULO

Ha quedado definitivamente constituido el C. I. C.

De acuerdo a la condición de la Unesco, el Centro Provisorio pasaría a tener existencia definitiva una vez que diez países ratificaran el instrumento de creación. Ello se logró, cuando la Argentina, el 28 de noviembre de 1961 depositó la ratificación de la adhesión.

Los países adheridos hasta este momento son: Japón, Bélgica, Ceylan, Italia, México, Francia, República Árabe Unida, Libia, Israel y la Argentina.

La Asamblea General se reunió del 30 de enero al 2 de febrero de 1962.

El representante argentino, Dr. Manuel Sadosky fue elegido Presidente de la Asamblea.

Participaron observadores gubernamentales de la República Federal Alemana y de Suiza, y un representante del Centro de Cálculo de Suecia.

De acuerdo a la convención, correspondió elegir los miembros del Consejo de Administración. Resultaron electos: Pierre Auger de Francia, Aldo Ghizzetti de Italia, el Dr. A. Gertz de Israel, el Profesor H. Yamashita, el Dr. H. M. Hussein de la R. A. U. y el Dr. M. Sadosky de la Argentina.

Fue elegido Presidente del Consejo el Profesor Auger y Vicepresidente el Profesor A. Ghizzetti.

Después de estudiar diversas propuestas, fue elegido director del Centro, el Dr. S. Comet de Suecia.

El Centro tiene su sede en Roma y allí será instalada la máquina ELEA 6001 que la casa Olivetti ha donado al Centro Internacional.

SEGUNDO CONGRESO INTERNACIONAL DE I. F. I. P.

Del 27 de agosto al 1 ° de septiembre de 1962 se lleva rá a cabo en Munich, Alemania, el Segundo Congreso Internacional sobre Procesamiento de la Información, como consecuencia de las resoluciones adoptadas en el Primer Congreso patrocinado por la Unesco, realizado en París en junio de 1959.

Las principales autoridades participarán en este congreso global organizado por la I. F. I. P., en el cual se tratarán todos los temas relacionados con el Procesamiento de la Información y las Computadoras Digitales.

La exhibición internacional de computadoras y otros equipos de procesamiento de la información (I. F. I. P. INTERDATA) acompañará a este Congreso.

El Dr. A. Walther de Alemania, ha sido designado Presidente del Congreso y el Dr. Niels Ivan Bech de Dinamarca, será el presidente del Comité de Programa. Miss C. Popplewell del Computing Machine Laboratory, de la Universidad de Manchester ac tuará como editora de las actas.

Las inscripciones para este Congreso se aceptan hasta el 15 de junio.

TERCERA CONFERENCIA INTERNACIONAL DE INVESTIGACION OPERATIVA

Del 1 ° al 5 de julio de 1963 se realizará en Oslo la tercera conferencia convocada por la Federación Internacional de Sociedades de Investigación Operativa. El presidente del Comité de Pro grama es el Sr. G. Kreweras, Centre de Recherches Economiques Appliquées 92, Boulevard Hausmann, París, 8.

Las comunicaciones deben ser enviadas hasta el 1^o mayo de 1962.

THE COMPUTING AND DATA PROCESSING SOCIETY OF
CANADA

La Tercera Conferencia organizada por esta Sociedad se realizará en la McGill University de Montreal del 11 al 12 de jun de 1962.

SEGUNDO CONGRESO DE A. F. C. A. L. T. I.

Se realizó en París del 17 al 20 de octubre de 1961, tratándose los siguientes temas:

Sesión 1 A. Organización de Centros de Cálculo.

1 B. Problemas no numéricos.

2 A Traducción automática.

2 B Métodos de programación.

2C Aplicaciones Industriales.

3 B Estructura y Lógica.

3 C Aplicaciones Económicas.

4 A Análisis Numérico.

4 B Documentación Automática.

- - - - - -o0o- - - - - -

Del 8 al 10 de marzo de 1962 se realizó en Houston, Texas, un symposium organizado por la Sociedad Neurológica de Houston y el Departamento de Neurología de la Baylor University, sobre tema: "Information Storage and Neural Control".

El 28 y 29 de marzo se reunió en Roma el Consejo de Administración para discutir el plan de actividades, para el año en curso, el presupuesto, el nombramiento de personal, el acuerdo con el Gobierno Italiano y la distribución de becas. La licenciada Cecilia Berdichevsky, del Instituto de Cálculo, ha obtenido una beca del C. I. C. para realizar estudios de especialización.

ALGOL

Hemos recibido una comunicación del Prof. A. van Wijngaarden, de Holanda, en la cual nos aclara que en dicho país existen varias computadoras que aceptan el lenguaje ALGOL 60 completo.

NOTICIAS DEL PAIS

JORNADAS DE LA S. A. C.

Con el propósito de dar a conocer las posibilidades de tratamiento numérico de la información en nuestro país, la Sociedad Argentina de Cálculo ha organizado unas jornadas que se realizarán del 8 al 10 de julio próximo.

La actual Comisión Directiva está interesada en establecer las bases para un programa de acción conjunta con los centros universitarios del interior del país, con ese objeto se están organizando las Jornadas de la Sociedad Argentina de Cálculo cuya finalidad inmediata es:

- a) Lograr que cada Universidad envíe miembros informantes que puedan reflejar cuál es la situación en la zona de influencia y dentro de la respectiva universidad, en cuanto a:
 - 1 - Existencia de profesionales o grupos de investigación dedicados al estudio de técnicas vinculadas al cálculo numérico y al tratamiento de la información con computadoras electrónicas.
 - 2 - Grupos de investigación que tengan necesidad de contar con los servicios de un centro de cálculo o de disponer directamente de una computadora para uso propio. Tipo y volumen del trabajo a encarar con computadora.
 - 3 - Programa de actividades futuras de la Universidad con relación a la especialidad. Necesidad de recibir colaboración, para el cumplimiento del programa trazado. Tipo de colaboración solicitada.

Las Jornadas de la Sociedad Argentina de Cálculo se realizarán en Buenos Aires del 8 al 10 de julio y, en su transcurso, se escuchará el informe de los delegados y se crearán comisiones de

trabajo que elevarán el proyecto de programa de actividades de la Sociedad para el período 1962-1963. La Sociedad Argentina de Cálculo cuenta con la colaboración de las empresas que se dedican a la comercialización de computadoras en el país, y con ellas se organizará una serie de visitas a centros de computación que funcionan en esta ciudad.

La presente invitación para participar en las Jornadas abarca a todas las Facultades de Ciencias Exactas, Ingeniería y Ciencias Económicas del país, las que podrán enviar un delegado por Facultad que tendrá todos los gastos de estada pagos, quedando por cuenta de las respectivas Facultades los correspondientes pasajes de ida y vuelta a Buenos Aires.

CURSO DE ECONOMIA MATEMATICA

Se dicta en la Facultad de Ciencias Exactas y Naturales, en el nuevo edificio de Núñez, los días lunes y miércoles de 14 a 16 horas y los viernes de 14 a 16. 30 horas.

Está a cargo del Dr. Varsavsky.

UNIVERSIDAD CATOLICA SANTA MARIA DE LOS BUENOS AIRES Instalación de una Computadora IBM 1620

La Universidad Católica Santa María de los Buenos Aires ha contratado con IBM World Trade Corporation, la instalación en la Facultad de Ingeniería y Ciencias Físico-Matemáticas de una computadora IBM 1620. La citada computadora, totalmente transistorizada, cuenta con una memoria de núcleos magnéticos con una capacidad de 40. 000 posiciones o dígitos decimales. El tiempo de acceso es de 20 microsegundos. La longitud de palabra es variable y las instrucciones constan de dos direcciones. La aritmética se lleva a cabo por medio de consulta en tablas almacenadas en la memoria. La entrada y salida de información se realizará en principio por medio de cinta de papel. La máquina imprime resultados a razón de 10 caracteres por segundo, perfora a 15 caracteres por segundo y recibe información a

razón de 150 caracteres por segundo. Los programas pueden escribirse en lenguaje absoluto, en lenguaje SPS, que es un lenguaje simbólico propio de la 1620, en el conocido lenguaje de la línea IBM, FORTRAN, y en lenguaje GOTRAN que es una variación del FORTRAN, y cuya principal característica es que la máquina opera directamente a partir de la introducción en la memoria del programa escrito en GOTRAN, es decir, no es necesario realizar el paso intermedio externo de producir una cinta con el lenguaje absoluto traducción del simbólico.

A fin de atender todo lo referente a la computadora ha creado en la mencionada facultad, el Departamento de Computación. Su principal finalidad es la de instruir a todos los alumnos en el tema de computadoras y disciplinas afines. Se tiene el convencimiento que es preciso que surja una generación de profesionales hábiles y decididos a sacar partido de todas las posibilidades ofrecidas por las computadoras, consideradas como simples instrumentos de trabajo; que es primordial crear las condiciones necesarias para que los egresados futuros de la Facultad, sean capaces de servirse de computadoras, aún de las más perfeccionadas, tan naturalmente como el ingeniero de nuestros días utiliza su regla de cálculo. No se trata de formar especialistas en computadoras, sino más bien de profesionales abiertos a la nueva técnica, gracias a una formación integral a lo largo de su carrera.

Ing. Pedro Zadunaisky

El 10 de marzo, el Ingeniero Pedro E. Zadunaisky dictó una conferencia, organizada por la Sociedad Argentina de Cálculo, sobre el tema: "Organización del Centro de Cálculo del Smithsonian para el cálculo de órbitas de satélites artificiales".

ESTADO ACTUAL DE LAS INVESTIGACIONES EN EL CAMPO DE LA INTELIGENCIA ARTIFICIAL

A fines de 1957, H. Simon del Carnegie Institute of Technology y Allen Newell de la Rand Corporation predijeron que en el término de 10 años una computadora digital será la campeona mundial de ajedrez a menos de que no le permitan competir. Esto, sin embargo, no va acontecer en un futuro inmediato, pues ya han pasado 4 de los 10 años propuestos y todavía no se ha llegado a un programa satisfactorio en el sentido de la predicción. Los trabajos en este campo son promisorios, pero no han ido lejos en producir resultados útiles.

Los programas que prueban teoremas matemáticos no han probado algo, que no fuera ya conocido y operan solamente en «áreas restringidas de la matemática. A fin de apreciar la plausibilidad de la predicción-de Simon y Newell es necesario ver no solamente el grado presente de realizaciones, sino también el progreso experimentado en la puesta en marcha de las investigaciones, tarea realizada en un periodo relativamente corto de tiempo.

Cabe preguntarse si no hay cosas más importantes que hacer: con las computadoras, y si bien la respuesta es afirmativa muchos de los problemas a resolver son tan dificultosos que debemos comenzar por investigar cosas más simples. Nosotros no comprendemos realmente cómo los seres humanos llegan a la solución de problemas complejos, y tampoco sabemos si existen caminos mejores que los que usamos. Las computadoras proveen una herramienta más para investigar estas cuestiones.

Con toda seguridad el uso de las computadoras, conducirá por medio de la combinación de la inteligencia humana y artificial al desarrollo de mejores técnicas para uso de los seres humanos.

Algunos investigadores sostienen que hemos llegado lo bastante lejos como para poder encarar con las computadoras ciertos importantes problemas sobre ellas mismas.

No es posible describir todos los trabajos realizados en este campo, o aún hacer una mención de ellos; me limitaré a describir

brevemente los más importantes.

Uno de los mejores trabajos publicados sobre inteligencia artificial fue realizado por A. L. Samuel de IBM, titulado: "Some Studies in Machine Learning Using the Game of Checkers". Se eligió el juego de damas porque es lo bastante complicado como para ser interesante. Tiene metas y reglas definidas, comparativamente más simples que las del ajedrez, lo que hace que sea preferible a éste en las investigaciones sobre técnicas de aprendizaje.

Podríamos eventualmente ser capaces de usar computadoras en áreas donde las metas y reglas no son conocidas precisamente. Parece razonable, sin embargo, comenzar con una situación, la cual es una etapa intermedia entre un problema del cual conocemos completamente cómo resolverlo, tales como ciertos problemas matemáticos, y uno del cual tenemos solamente vagas ideas sobre cómo solucionarlo, por ejemplo la mayoría de los problemas comerciales.

Las damas es un juego intermedio, porque aunque las reglas de juego son precisas la estrategia no lo es. El programa para la IBM 704 permite aprender por memorización y generalización; el programa basado en la memorización, aunque tiene bastante éxito, posee un pequeño interés teórico (puede improvisar en la apertura y algunas partes de los finales, con la experiencia acumulada, pero muy poco en el medio juego).

La parte más interesante del programa del Dr. Samuel es la referente al aprendizaje por generalización. El programa ha sido provisto de un conjunto de rutinas usado para determinar que movidas pueden ser hechas sobre la suposición de que cada jugador debe moverlas para maximizar sus chances. Con una combinación de aprendizaje por memorización, para la apertura y algunas fases del final, y aprendizaje por generalización para el medio juego, el programa puede aprender a ganarle a un buen amateur. Es interesante de hacer notar que esta combinación de dos tipos de aprendizaje es la que usa un jugador humano al estudiar el juego.

Un número de programas ha sido escrito para hacer jugar a una máquina al ajedrez, el juego intelectual por excelencia. Enfrenta a dos intelectos en una situación tan compleja, que ninguno de

los dos puede esperar a entenderla completamente pero suficientemente sujeta a análisis como para que uno de ellos pueda esperar el hecho de tomar en cuenta más factores que su oponente y poder basar sobre ellos un esquema de juego más satisfactorio.

Si alguien pudiera inventar una máquina que jugara satisfactoriamente al ajedrez habría penetrado en el corazón de los intentos intelectuales humanos.

La historia de los programas de ajedrez es un ejemplo de los ensayos humanos para concebir y enfrentarse con mecanismos complejos. El ajedrez es mucho más complejo que las damas, por consiguiente el progreso ha sido naturalmente más lento, pero de todas maneras, impresionante.

La historia comienza con la máquina electrónica construida por el ingeniero Torres Quevedo a comienzos de este siglo, que resolvía finales de torre y rey, contra rey.

Desde el punto de vista moderno, comienza con un artículo publicado por Claude Shannon en 1949. El no presentó un programa, pero discutió muchos de los problemas básicos involucrados en el diseño de un programa para jugar al ajedrez. Las reglas del juego determinan cuándo un partido ha terminado en victoria, pérdida o empate. Por consiguiente, el ajedrez puede ser descrito completamente por medio de una bifurcación con tres posibilidades, los nudos corresponden a posiciones y las bifurcaciones corresponden a alternativas. Es claro que para un jugador que pueda apreciar las últimas con secuencias de cada alternativa el ajedrez se transforma en un juego simple. Partiendo de las posiciones finales puede razonar hacia atrás determinando en cada nudo cuál es la mejor bifurcación para él. Este procedimiento de deducción llamado minimizante en la teoría de juegos es común a todos los intentos efectuados en la realización de programas.

Shannon propuso una medida numérica formada sumando un número de factores (considerando su importancia relativa) que podrían ser computados en cada posición. Un programa para la IBM 704 realizado por Alex Bernstein de IBM juega un partido interesante aunque usa un método que no "ve" más de dos movidas adelante (dos para las negras y dos para las blancas) similar al realizado por un grupo de Los Álamos para la MANIAC 1.

El programa de Bernstein considera solamente una fracción de todos los posibles movimientos legales que podrían realizarse a continuación. Existen una serie de subrutinas llamadas generadores que proponen las movidas a considerar. El programa considera siete alternativas las cuales son obtenidas operando los generadores por orden de prioridad hasta que los siete son acumulados, entonces para cada posición se consideran siete posibles continuaciones directas que se evalúan usando la relación de dos sumas, una para las blancas y otras para las negras.

Cada una se compone de cuatro factores: balance y movilidad de las piezas, control de área y defensa del rey.

El programa de Bernstein consta de siete mil palabras, el de Los Álamos solamente tiene seiscientas, tardan ocho y doce minutos respectivamente para producir una movida. Bernstein disminuye drásticamente el número de movidas a considerar en una dada posición introduciendo un conjunto de potentes métodos heurísticos toma dos de la práctica exitosa del ajedrez.

El grupo de A. Newell y C. Shaw de Rand y H. Simon del Carnegie Institute of Technology ha realizado un programa para la JOHNNIAC de la Rand Corporation. El valor de una componente es en algunos casos un número, por ejemplo, en el balance del material se usa el valor nueve para la reina, etc.

En otros casos el valor de la componente es dicotómico simplemente señala la ausencia o presencia de alguna propiedad, por ejemplo un peón doblado, etc.

Hasta el momento no le ha ganado a nadie pero existen buenas perspectivas de que le pueda ganar a un novicio.

Las únicas metas incluidas en el presente en la evaluación de las posibles movidas son el balance y desarrollo de las piezas y el control del centro del tablero. No incluye ataques, defensa del rey, etc.

A pesar de este conjunto limitado de objetivos, el juego resultante es definitivamente interesante.

Con la inclusión de otros objetivos y con una computadora más veloz que la JOHNNIAC (20.000 ops/seg) este programa probablemente será el primero en ganarle a un jugador humano. Es interesante hacer notar que los realizadores de este programa a menudo no pueden predecir lo que el programa hará en una situación dada, hecho este que hace vacilar a los que opinan que una computadora puede sola mente hacer lo que se le ha indicado.

(Existe un programa hecho en Rusia para la BESM I del cuál no se ha suministrado ninguna información).

El mismo grupo ha desarrollado un programa para probar teoremas de lógica elemental usando también la JOHNNIAC.

Fundamentalmente, el probar teoremas y el jugar al ajedrez involucran el mismo problema: razonamientos con métodos heurísticos que seleccionan caminos provechosos de exploración en un campo donde las posibilidades crecen en forma exponencial. Ellos comienzan por dar a la máquina los axiomas y reglas de deducción para el cálculo proposicional tomadas del Principia Mathematica de Whitehead y Russell. Entonces suministran a la máquina el primer teorema y le preguntan si puede hallar la demostración, la máquina pudo hacer lo, entonces se le suministra el segundo teorema y se pregunta si puede hallar la demostración utilizando los axiomas y el primer teorema y así sucesivamente.

Procediendo de esta manera el programa produjo demostraciones para la mayoría de los teoremas del segundo capítulo (de cincuenta y dos teoremas demostró treinta y ocho, la mayoría de las fallas fueron por limitaciones de tiempo o espacio).

En un caso la computadora encontró una demostración que requería bastantes menos pasos que la solución dada por el texto.

En algunos casos como en el ajedrez puede no existir un algoritmo práctico, porque para evaluar todas las posibles movidas en el juego llevaría milenios aún con la computadora más veloz que se pueda imaginar. En otros casos puede no existir un algoritmo por la naturaleza del problema.

En algunos casos se recurre a métodos heurísticos

(0029) que permiten resolver un problema dado, pero "a priori" no ofrecen garantías de resolverlo. La ventaja de estos -métodos reside en el hecho, de que permiten encontrar una solución en forma mucho más: rápida, que por medio de un algoritmo.

Estos métodos sólo indican el camino que puede llevar a la solución del problema de modo que solamente los caminos promisorios son explorados.

En la prueba de teoremas las indicaciones son, en efecto, generadas por el programa mismo. La máquina trabaja hacia atrás desde la conclusión del teorema buscando las posibles líneas de razonamiento que conduzcan a las principales premisas de partida.

Si uno de estos caminos conduce a información no suministrada, el camino es descartado y se intenta otro camino más promisorio.

Este método general de trabajo es el corazón de muchos de los programas para resolver problemas que se están estudiando.

Herb Gelernter de IBM ha desarrollado un programa para probar teoremas en geometría plana. Opera en las mismas líneas del razonamiento heurístico antes descrito. Los pasos en una aplicación típica de el método heurístico a la demostración de teoremas son los siguientes:

- 1) Calcular el carácter del teorema.
- 2) Usando el carácter del teorema, calcular los métodos aplicables y estimar el mérito de cada uno.
- 3) Seleccionar el método más apropiado.
- 4) Probarlo.
- 5) En caso de fracasar, abandonar este método y volver al paso 3).

6) En caso de éxito imprimir la demostración y parar.

El carácter de un teorema (o de un problema dado) es en esencia la representación de la máquina del teorema (o del problema). El objetivo de Gelernter es encontrar nuevas aplicaciones para las computadoras.

Hao Wang de la Universidad de Oxford ha desarrollado el llamado "inferential analysis" como disciplina hermana del análisis numérico.

Ha realizado 3 programas para la IBM 704 en el año 1958. Con el primero logró demostrar más de 200 teoremas del libro de Whitehead y Russell en unos 37 minutos que descontando el tiempo de entrada y salida, quedan reducidos a menos de 3 minutos efectivos de máquina. Los 52 teoremas tomados por el grupo de Newell-Shaw-Simon fueron probados en menos de medio minuto (descontando nuevamente el tiempo de input-output). Los otros dos programas se refieren a problemas no considerados por Newell-Shaw-Simon en sus trabajos. El segundo programa hace que la máquina forme proposiciones del cálculo proposicional a partir de símbolos básicos y algunos teoremas no triviales. El tercer programa complementa el trabajo ya iniciado con el primero; continúa con un total de más de 150 teoremas de los siguientes cinco capítulos del libro de Whitehead y Russell. Realizó las demostraciones y la impresión para el 85% de los teoremas en alrededor de una hora. En un caso la máquina demostró un teorema por un método más elemental. Más recientemente logró probar todos los teoremas (alrededor de 350) de los primeros 10 capítulos en 9 minutos.

En resultados, el trabajo de Wang va por supuesto más lejos que el de Newell, Shaw y Simon pero no es posible una comparación directa pues están trabajando en distintas direcciones. Wang espera que en un futuro no lejano podemos esperar una ayuda real de las máquinas en la investigación matemática. El contempla la posibilidad de que una computadora permita verificar y formalizar esbozos de demostraciones y en una etapa posterior vencer o dejar atrás la ayuda del matemático y probar muchos teoremas interesantes. Por ejemplo espera encontrar la demostración (o la refutación) del último teorema de Fermat, esto marcaría el fin de las discusiones sobre si una máquina puede o no pensar. Por supuesto ninguna máquina tiene la facultad

tad creadora y la originalidad de un Newton o un Mozart, pero cabe hacerse la pregunta de cuántos seres humanos la tienen. El mayor problema es que los psicólogos tienen muy poco que decir acerca del pensamiento independientemente del pensador. A. M. Turing propuso un juego con el cual reemplaza la inevitablemente ambigua pregunta referente a si una máquina puede pensar. En este juego un examinador presenta preguntas a un ser humano y a una computadora usando una máquina de escribir. Después de jugado el juego el examinador decide cuál fue el ser humano y cuál la máquina. Excepto por las sutilezas del lenguaje una máquina puede competir exitosamente con un ser humano:

Al parecer los rusos le han dado gran importancia a la investigación en este campo. Ellos utilizan el término cibernética para abarcar este campo, tanto como el de la automatización, computadoras y control automático.

Hugo D. Scolnik

BIBLIOGRAFIA

- 1) C. E. Shannon: "Programming a Computer for Playing Chess. Phil. Mag." 41, 256 (March 1950).
- 2) A. Bernstein and M. Roberts: "Computer vs. Chess-Player, Scient. Amer." 198, 6 (June 1958).
- 3) "Some Studies in Machine Learning, Using the Game of Checkers" A. L. Samuel IBM J. of Res. and Devel. 3.210 (July 1959).
- 4) A. Bernstein: "A Chess Playing Program for the IBM 704." Proceeding of the 1958 Western Joint Computer Conference. May 1958.
- 5) A. Newell, J. C. Shaw and H. A. Simon: "Chess Playing Programs and the Problem of Complexity". IBM J. of Res. and Devel. 2. 321 (October 1958).
- 6) A. Newell and H. A. Simon: "The Logic Theory Machine". Transactions on Information Theory. IT-2, N° 3, September 1956

- 7) H. Gelernter: "Theorem Proving by Machine" Ibid, pág. 305.
- 8) H. Wang: "A Variant to Turing's Theory of Computing Machines!". Journal ACM, 4, 88-92. (January 1957).
- 9) H. Wang: "Toward Mechanical Mathematics". IBM Journal of Research and Development. Vol. 4, N° I, pág. 2, (January 1960).
- 10) M. Minsky: "Steps Toward Artificial Intelligence". Proceedings of the Ire. (January 1961). (Contiene abundante Bibliografía).

----- 000 -----

Seminario:

"RESOLUCION NUMERICA DE ECUACIONES DIFERENCIALES"

Prof. P. Zadunaisky

En el Instituto de Cálculo de la F. C. E. y N. se está desarrollando un Seminario sobre "Resolución Numérica de Ecuaciones Diferenciales".

El libro, "Discrete Variable Methods in Ordinary Differential Equations", junto con el de Milne "Numerical Solution of Differential Equations" se emplearán en el Seminario de Resolución Numérica de Ecuaciones Diferenciales.

La idea de este Seminario a cargo del Prof. P. Zadunaisky es trabajar en el Tema con un grupo reducido de jóvenes entusiastas.

Cada uno de estos jóvenes estudiará uno o varios métodos, lo expondrá y lo experimentará en la computadora Mercury que posee el instituto. Para ello contará con la guía y el consejo del Prof. P. Zadunaisky en los problemas teóricos y de V. Pereyra en el aspecto práctico del cálculo.

El material resumido en los distintos experimentos será de gran utilidad para la comparación de los distintos métodos y ayudará seguramente en la investigación que se realiza para poner a punto, métodos eficientes de integración numérica de órbitas.

EN LAS EMPRESAS

IBM WORLD TRADE CORPORATION

APROVECHAMIENTO DE UN SISTEMA ELECTRONICO ORIENTADO A CINTAS MAGNETICAS PARA OPERACIONES CON MATRICES Y/O RESOLUCION DE SISTEMAS LINEALES DE INE- CUACIONES (SIMPLEX).

Se realiza un estudio simplificado de tres organigramas clásicos en operaciones matriciales: multiplicación, inversión y método simplex para Programación Lineal. Si bien se realiza para un sistema de configuración mixta, Técnica-Comercial (IBM 1401), los organigramas no pierden generalidad. Se realiza además un cálculo de tiempos operativos y órdenes de matrices procesables.

Multiplicación de Matrices

Se trata de multiplicar dos matrices A, B almacenadas en cinta para obtener la matriz resultado C que quedará almacenada en una tercera cinta. Si adoptamos el criterio de almacenar A a modo de fila, o sea cada fila ocupando un registro de cinta, tendremos que almacenar B a modo de columna o sea un registro por columna. Si A es de orden $m \times n$ y B de orden $n \times k$ y adoptamos el esquema anterior necesitaremos rebobinar la cinta 2 (que almacena a B) n veces. En efecto por cada fila leída de cinta 1 (que almacena a A), necesitamos leer R columnas de cinta 2 para computar una fila de la matriz C que se almacenará a modo de fila en la cinta 3.

Este proceso de cálculo esquematizado en el organigrama tiene aparentemente como única parte operativa la subrutina del producto interno de dos vectores repetida m veces. En cambio si se desea crear un programa que sea de utilidad para realizar álgebra de matrices la complejidad crece de tal modo que la importancia de la subrutina del producto se desvanece. En efecto, normalmente el programa imprime y/o perfora la matriz C. Si se decide perforar tendrá que ser en un formato compacto a los efectos de disminuir el tiempo de salida y los datos estarán a punto decimal flotante. La entrada de datos en cambio podrá estar a punto decimal fijo o flotante y en por lo menos dos formatos distintos, a un coeficiente por tarje-

las cintas. Finalizada la carga se leerá de la cinta el subprograma de cómputo continuándose el proceso hasta el final, momento en el que se ordenará leer el subprograma de salida como última instrucción del subprograma de cómputo. Todos estos tres subprogramas se almacenarán en las mismas 2.000 posiciones que constituyen la zona disponible.

Tiempos operativos

Los tiempos son muy dependientes del tipo de entrada/salida así es que supondremos un trabajo tipo con entrada por tarjetas y 30% de coeficientes no nulos (que son los únicos que se perforarán) y salida por impresión de la solución.

Una fórmula aproximada del tiempo estará dada por:

$$(6 \times n \times m \times k) \text{ m seg.}$$

lo que para $n = m = k = 100$ arrojará un tiempo de cómputo de 100 minutos.

Considerando que los elementos entran a 4 por tarjeta, tendremos aproximadamente más de 800 tarjetas, lo que a una velocidad promedio de 780 tarjetas por minuto arroja 1, 03 minutos para la entrada de los datos.

La impresión final a una velocidad promedio de 500, líneas por minuto y a 5 coeficientes por línea arroja un tiempo de 4 minutos para los 10. 000 coeficientes de la solución C. Tenemos pues un tiempo total de 2 h. 52 minutos para multiplicar dos matrices A, B con impresión final de la matriz resultado C.

Inversión de matrices

Para la inversión de matrices adoptaremos el siguiente proceso:

Supongamos tener una matriz A en cinta a modo de fila:

ta o a varios. Lógicamente por razones de uniformidad el formato de entrada a punto decimal flotante tendrá que ser el de salida normal. Además el programa tiene que poder aceptar entrada directa de cinta (grabadas en procesos anteriores). Evidentemente el programa debe preveer el procesamiento de matrices perforadas: grabadas a modo de fila o de columna. El deseo de lograr un programa que cumpla con estos requisitos y además nos procese las matrices de máximo orden permisible por nuestra memoria operativa, dificulta la tarea a realizar.

Distribución de las zonas de trabajo

Disponemos de una unidad central de proceso IBM 1401, con una capacidad de 8. 000 dígitos decimales de memoria de núcleos magnéticos. Necesitamos además tres zonas de trabajo, L, P, T que almacenarán respectivamente a las filas de A, las columnas de B y las filas de C. Suponiendo que procesamos matrices cuadradas o que tengan como máximo un orden $(m, n, \text{ ó } k) \leq 200$, necesitaremos en total 6.000 posiciones de memoria (a 10 dígitos decimales para la representación de un número en flotante).

El resto, 2.000 posiciones, las ocuparemos en el programa, zonas de entrada/salida y subrutinas de error en cinta.

En el organigrama, (I) indicará el punto donde se nuclea la parte de cómputo en si. En este punto se realizará $m \times n \times k$ veces el algoritmo de producto y suma. En (II) se rebobinará m veces la cinta 2, para obtener las m filas de C.

Evidentemente si se desea obtener un programa que prevea todas las posibilidades de entrada-salida acotadas arriba, las 2. 000 posiciones de memoria son insuficientes. Solamente el análisis y conversión de las posibilidades de modo de entrada,

<u>MATRIZ A</u>	<u>MATRIZ B</u>	
F	F	donde con F se designa
F	C	fila y con C se designa
C	F	columna.
C	C	

Conducente a las dos posibilidades $A \times B$ ó $B \times A$, demandará gran parte de la memoria utilizable para programa.

Analicemos las distintas transformaciones:

- F F deberemos, previo almacenamiento en cinta 1 y 3 respectivamente, entrar a la subrutina de conversión entre cinta 3 - cinta 2.
- C C deberemos, previo almacenamiento en cintas 2 y 3 respectivamente, entrar a la subrutina de conversión entre cinta 3 - cinta 1.
- C F deberemos almacenar la matriz A en cinta 3 y entrar a la subrutina de conversión entre cinta 3 - cinta 1, repitiendo el mismo proceso para la matriz B.
- F C es el modo normal de entrada.

Se nota que el organigrama se encuentra dividido en bloques por las marcas $-f$, $-ff$, constituyendo cada uno de ellos partes perfectamente superponibles en la memoria. En efecto, el primero de estos bloques constituirá el subprograma involucrado en la entrada del sistema en todas sus variantes. El segundo, el subprograma de cómputo en sí y el tercero el subprograma de salida. Una de las distribuciones posibles del programa completo sería en orden físico (suponemos el programa y los datos en tarjetas):

- 1 - Segundo bloque
- 2 - Primer bloque
- 3 - DATOS (si los hubiera en tarjetas)
- 4 - Tercer bloque

Una vez entrado el primer lote de tarjetas en la máquina (segundo bloque) se almacenará en cinta y se procederá a continuar la carga del segundo lote (subprograma de entrada). Finalizado esto se ordenará la ejecución del mismo que entrará y armará los datos en

a_{11}	a_{12}	...	a_{1n}	Registro 1
a_{21}	a_{22}	...	a_{2n}	Registro 2
a_{31}	a_{32}	...	a_{3n}	Registro 3
...
...
a_{n1}	a_{n2}	...	a_{nn}	Registro n

y supongamos además que tenemos en la memoria operativa a la primera de estas filas en una zona que denominaremos T.

$$a_{11} \quad a_{12} \quad \dots \quad a_{1n} \quad \text{ZONA T}$$

Eligiendo * a_{11} como pivote de la transformación y agregando un 1 a continuación de a_{1n} se obtiene a partir del elemento a_{12} ,

$$b_{11} \quad b_{12} \quad \dots \quad b_{1n} \quad 1/a_{11} \quad \text{definiendo } b_{ij} = a_{1j}/a_{11}$$

que pasará a ser el vector transformante y estará ubicado en la zona L.

Este proceso está descrito en el organigrama en (I) En (II) se describe el proceso algorítmico de Gauss adaptado al procesamiento vector a vector. Se lee el vector fila en la zona P y si la cabeza de fila $a_{i1} = 0$ se procede a la transformación de la fila. En caso contrario la fila no sufre transformación.

* No hacemos en este trabajo consideraciones relativas a conmutación de columnas debido a elección del pivote como el elemento de máximo valor absoluto y supondremos además que $a_{11} \neq 0$.

Para la transformación se coloca un 0 como elemento (n+1) ésimo de la zona P y se procede a transformar los elementos a partir del elemento a_{i2} . El elemento 0 se transforma como $-a_{i1} / a_{11}$.

Al terminar de leerse la matriz en un paso $k < n$ de una de las cintas, hay grabada en la otra n-1 filas correspondientes al paso $k + 1 \leq n$.

La fila enésima del paso k+1 será la fila transformante tal como está ubicada en la zona L. (IV en el organigrama).

Como la primera fila del paso k+1 constituirá la base del vector transformante en el paso k+2 es necesario almacenarlo en la zona T. Evidentemente como la primera fila está en la zona T y en el primer registro de la cinta que se termina de grabar, necesitamos saltarlo para el próximo ciclo.

Por rotación de cintas se entiende el cambio de 1 a 2 ó de 2 a 1 en todas las instrucciones que involucran lectura, grabación de cinta o control de cinta.

La transformación matricial está esquematizada del siguiente modo:

$$\begin{array}{ccc}
 \begin{array}{ccc}
 k & k & k \\
 a_{11} & a_{12} & a_{13} \\
 \\
 k & k & k \\
 a_{21} & a_{22} & a_{23} \\
 \\
 k & k & k \\
 a_{31} & a_{32} & a_{33}
 \end{array}
 &
 \begin{array}{c}
 \\
 \\
 \text{pasa} \\
 \text{a} \\
 \text{convertirse} \\
 \text{en}
 \end{array}
 &
 \begin{array}{ccc}
 k+1 & k+1 & k & k \\
 a_{22} & a_{23} & - a_{21} & / a_{11} \\
 \\
 k+1 & k+1 & k & k \\
 a_{32} & a_{33} & - a_{31} & / a_{11} \\
 \\
 k & k & k \\
 b_{12} & b_{13} & 1 / a_{11}
 \end{array}
 \end{array}$$

Dado el lugar geométrico que ocupan los coeficientes de esta matriz y por similitud con la matriz equivalente a partir de la cual fue generada, es mejor representarla así:

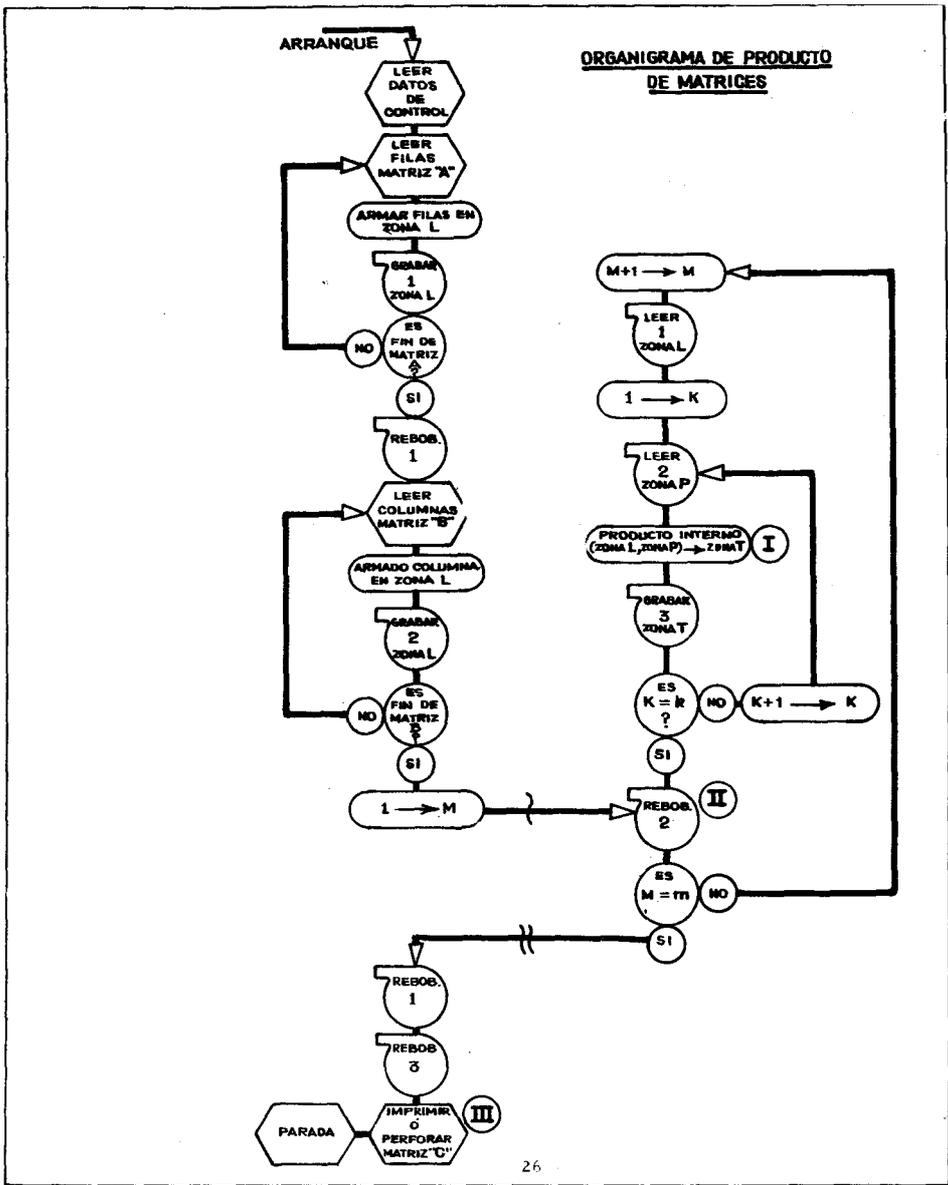
$$\begin{array}{ccc}
 k+1 & k+1 & k+1 \\
 a_{11} & a_{12} & a_{13} \\
 \\
 k+1 & k+1 & k+1 \\
 a_{21} & a_{22} & a_{23} \\
 \\
 k+1 & k+1 & k+1 \\
 a_{31} & a_{32} & a_{33}
 \end{array}$$

En este esquema se quiere indicar que el contenido del primer campo destinado a la ubicación del primer coeficiente será en orden sucesivo,

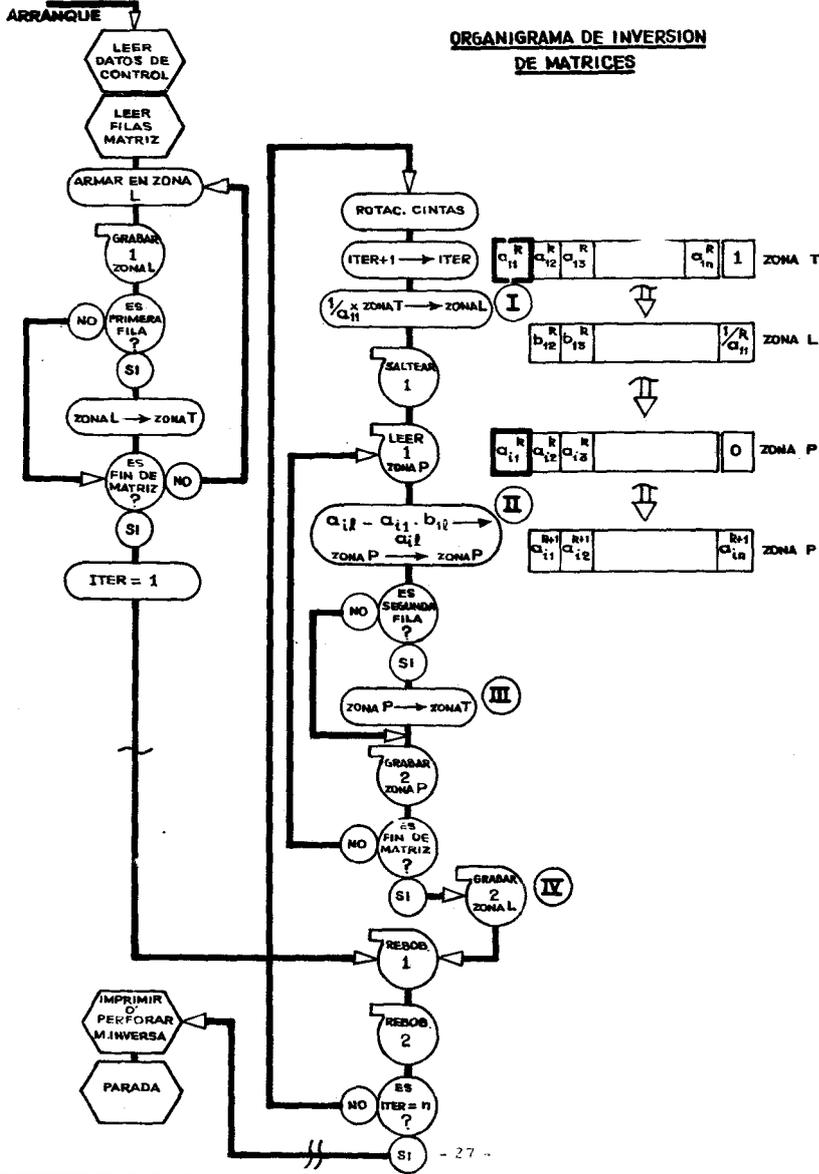
$$a_{11}^1, a_{22}^2, a_{33}^3, \dots, a_{nn}^n$$

Continuaremos en el próximo número con el estudio del organigrama del método Simplex, para Programación Lineal adaptado a Investigación Paramétrica.

Ingeniero J. Chamero
Departamento de Ingeniería de Sistemas
IBM Argentina



**ORGANIGRAMA DE INVERSION
DE MATRICES**



REMINGTON RAND SUDAMERICANA

SISTEMAS EJECUTIVOS

1) Introducción Histórica

En las primeras etapas de la computación, al realizar un programa debían prepararse todas y cada una de las instrucciones que lo componían, en código absoluto o de máquina. Dado que las aplicaciones iniciales fueron de carácter científico, se vió que determinados conjuntos de instrucciones se repetían en uno y otro de los programas, por ejemplo, calculo de funciones trigonométricas. Esto llevó a la idea, hoy tan natural, de codificar estos conjuntos de una vez para siempre e incorporarlos a cada programa que los necesitase. Así nacieron las rutinas matemáticas.

Posteriormente, al comenzar con el tratamiento masivo de datos, propio de la mayoría de las aplicaciones comerciales y algunas científicas, las necesidades se trasladaron al campo del control de las unidades de Entrada/Salida, dando origen a las rutinas respectivas. Estas rutinas gobiernan y mantienen un flujo constante de información hacia y desde el computador, a través de las unidades respectivas.

En este punto el programador solamente debe ocuparse del traslado y almacenamiento de datos dentro del computador, ya que posee rutinas preescritas de control de Entrada/Salida, de Clasificación e Intercalación, etc., y rutinas de diversas operaciones matemáticas.

El objetivo se dirige ahora a facilitar aún más esta operación de manejo interno de datos, reemplazando los códigos absolutos por códigos mnemotécnicos y las direcciones absolutas por direcciones relativas, haciendo que el computador realice la transformación inversa.

Se crean diversos tipos de lenguaje que, combinados con las rutinas de traducción respectivas, reciben el nombre de sistemas compiladores. Un sistema compilador tipo asigna código (y direcciones absolutas, de tal forma que a cada línea de codificación corresponde una instrucción en lenguaje absoluto. El compilador-traductor realizará la comprobación de la memoria ya ocupada y la que va asignando en el transcurso de la traducción, además de calcular locaciones en tiempo mínimo, si ello correspondiere, y realizar el listado de ambos programas, el original y el objeto.

Los sistemas compiladores están, en general, orientados hacia un computador específico y sus instrucciones particulares, lo que impide la utilización de un mismo programa fuente en diversos computadores. Aparece, por lo tanto, la necesidad de contar con lenguajes en un nivel tal que puedan ser utilizados por todos los computadores. Tales lenguajes están en pleno desarrollo principalmente los denominados ALGOL y COBOL.

La tarea de programación queda reducida al análisis del problema y la escritura de las instrucciones básicas en lenguaje corriente. El computador realizará en forma automática la conversión de cada una de estas instrucciones en trozos completos de codificación absoluta e intercalará rutinas y subrutinas de la biblioteca. Mediante el uso de rutinas de análisis se realiza la depuración (DEBUGGING) del programa y finalmente se lo incorpora a la Cinta Maestra de Instrucciones, listo para su operación.

2) Necesidad de Controlar la Operación

A medida que los computadores incrementan su velocidad y la variedad de medios de Entrada/Salida se recarga la tarea de los operadores. Ya no sólo deben tomar a cargo la tarea de cambiar carretes de cinta magnética, proveer de tarjetas a las lectoras y de papel a las impresoras, sino que deben tomar a su cargo las paradas y tratar de reducir a un mínimo el tiempo perdido en ellas.

Al finalizar un programa, éste reduce sus necesidades de memoria a un mínimo, para resultar limitado por la salida.

En ese caso, se desea seleccionar el próximo programa a ejecutar y cargarlo en el computador aprovechando las facilidades disponibles.

Por otro lado, una instalación desearía correr simultáneamente en el computador un programa limitado por Entrada/Salida y otro limitado por el computador. La tarea de coordinar ambos programas quedará en las manos de los programadores y ambos programas resultarán unidos en todas sus utilizaciones.

A estos problemas se dan varias soluciones. El computador UNIVAC LARC posee una o dos unidades de cálculo aritmético y un procesador independiente para el manejo y control de las Entradas y Salidas de informaciones, cada uno de los cuales sigue su propio programa. En el Gamma 60 se extendió esta idea multiplicando el número de unidades de control, o sea de registros decodificadores (X).

Este tipo de solución recarga los circuitos del equipo y esto se traduce en un encarecimiento del mismo.

Otro tipo de solución se dirigió hacia el conjunto de ru tinas que son provistas con un equipo por la Casa constructora.

Esto significó la creación de una rutina madre o maestra que proveyera control sobre el esquema diario de corridas del computador, asignación de memoria y unidades externas, gobierno de las operaciones de Entrada/Salida, que permitiera el procesado múltiple y concurrente y controlara las interrupciones en la operación.

(X) El número de registros decodificadores indica en un computador la cantidad de instrucciones que se ejecutan simultáneamente. Consideramos necesario una revisión del concepto de simultaneidad, tanto en la ejecución de instrucciones como de programas, y la definición precisa del mismo

3) Definición de un Sistema Ejecutivo

Un sistema ejecutivo es un conjunto de rutinas y subrutinas que realiza las operaciones mencionadas en el último párrafo del punto 2).

El objetivo principal del sistema ejecutivo (similar al del ejecutivo en, una empresa) es permitir una operación 100% del procesador central, incluyendo toda la memoria y tantos dispositivos de Entrada/Salida como fuera necesario para la operación simultánea (time-sharing) de varias aplicaciones, algunas de las cuales estarán limitadas por el computador y otras por E/S.

Si este objetivo fuera imposible de alcanzar, debido a los problemas particulares que van a ser corridos, entonces se busca un objetivo secundario consistente en mantener la operación de todas las unidades de E/S a un nivel de 100%.

Veamos un ejemplo para aclarar esta situación: consideremos que se desea correr dos problemas: uno limitado por el computador y el otro por E/S. Más aún, consideremos que el problema previo en el computador nos ha dejado una cierta cantidad de información que debe ser impresa y que el próximo problema requiere Entrada a partir de tarjetas perforadas.

El computador posee memoria principal en núcleos magnéticos, memoria auxiliar (maliva) en tambores magnéticos y como medios de E/S se tienen cintas magnéticas, tarjetas perforadas e impresión. El ejemplo' puede resumirse en el siguiente cuadro:

<u>Tipo de Problema</u>	<u>Facilidades</u>	<u>Tiempo (1)</u>
Cómputo: Solución de una matriz	Núcleo, tambor	96 minutos
E/S: Telemetría (reducción de datos)	Cinta, núcleo	149 minutos
Salida: Impresión de resultados	Tambor, núcleo, impresora	150 minutos

<u>Tipo de Problema</u>	<u>Facilidades</u>	<u>Tiempo (1)</u>
Entrada: Datos a tambor	Lectora de tarjetas, núcleo, tambor	150 minutos

(1) - Este tiempo de ejecución es el de cada problema operando en forma aislada y sucesiva.

El tiempo total de ejecución bajo el control del sistema ejecutivo es de sólo 151 minutos.

Resumiendo, podemos decir que los sistemas ejecutivos facilitan la operación del computador y ahorran tiempo al permitir la realización simultánea de varios programas.

En la segunda parte de estas notas analizaremos las diversas secciones que se distinguen en tales sistemas, las características necesarias del computador y finalmente las comunicaciones entre los distintos programas y el sistema ejecutivo.

Alfredo Pérez
Remington Rand, Sudamericana

SECCION BIBLIOGRAFICA

Mikusifiski, Jan
"OPERATIONAL CALCULUS".
Pergamon Press, 1959.
Quinta Edición.
485 páginas. .

Es éste un verdadero tratado sobre el tema, que expone metódicamente el Cálculo Operacional según la teoría del profesor Mikusifiski; como es sabido, se basa en el espacio F de cocientes de convolución. De este modo, a diferencia de otros tratados de Cálculo Operacional, basados en la integral de Laplace, evita problemas de analiticidad. Sumamente claro, posee abundantes ejemplos y plantea problemas al final de cada capítulo, adjuntando su solución

El libro es autosuficiente, gracias al Apéndice, en que se detallan y demuestran todos los principios y teoremas necesarios para una cabal comprensión del texto, y que aparece en esta edición por primera vez,

En el prólogo, el autor confiesa que ha escrito su libro teniendo en cuenta tanto al matemático puro como al ingeniero; prueba de ello son los capítulos dedicados a la solución de problemas

planteados por la técnica. El correspondiente a estática de vigas fue escrito en colaboración con el Profesor S. Drobot.

El volumen está dividido en siete partes:

- I) Álgebra operacional.
- II) Sucesiones y series de operadores.
- III) Cálculo diferencial operacional.
- IV) Esbozo de la teoría general de ecuaciones diferenciales lineales con coeficientes constantes.
- V) Cálculo integral operacional.
- VI) Apéndice (además de los prerequisites antes señalados, contiene los últimos resultados de la teoría),
- VII) Fórmulas y Tablas.

Beatriz Margolis

Durand, E.
"SOLUTIONS NUMERIQUES DES
EQUATIONS ALGEBRIQUES".
Equations du type $F(x) = 0$
Racines d'un polynôme.
Tomo I.

Es este un tratado moderno que cubre una gran parte de lo que se conoce sobre los aspectos teórico-prácticos del tema.

En ese sentido es un libro de matemática real, que analiza cada método con miras a su utilización efectiva.

Los métodos están descriptos con claridad y las demostraciones se han elegido entre las que usan argumentos más elementales.

Pero lo que lo hace real mente valioso y actual, es la experiencia del autor en calculadoras electrónicas, que se refleja en la gran cantidad de tablas numéricas con resultados de la aplicación de los distintos métodos y sobre todo en los comentarios, reflexiones y comparaciones que pueden ser una guía valiosa para el que se inicia.

También estas discusiones y ejemplos representan un ahorro de tiempo para aquellos que desean desarrollar programas eficientes para máquinas

nuevas y proporcionan un material muy apto para nuevas experiencias..

Quizá sea necesario tomar con cautela algunas comparaciones ya que el autor ha experimentado con una máquina particular (IBM 650) y puede suceder que el método más eficiente para ésta, no lo sea para otra con características diferentes (memoria, velocidad, equipo auxiliar).

Victor Pereyra

Vajda, S.
"MATHEMATICAL PROGRAMMING".
London, Addison-Wesley, 1961.
IX + 310 páginas.

El presente libro de reciente aparición está escrito con vista a servir como texto de estudio de programación lineal a nivel universitario. Los primeros capítulos los dedica a fijar rigurosamente los conceptos básicos de programación lineal al estudiar Álgebra de las desigualdades lineales (Cap. 2), Álgebra de dualidad (Cap. 3) y la Teoría de los grafos y teoría combinatoria (Cap. 4). Los dos capítulos siguientes los dedica al estudio

detallado del método simplex y a los esquemas particulares del problema de transportes o del método húngaro. Los capítulos 7 y 8 tratan algunas aplicaciones. Termina dando una exposición del estado de la Programación lineal paramétrica (Cap.9), Programación lineal discreta (Cap. 10) Programación lineal estocástica (Cap. 11), Programación no lineal (Cap. 12) y Programación dinámica (Cap. 13).

Todos los capítulos contienen numerosos ejemplos y ejercicios, dedica 45 páginas a su solución y da 128 referencias bibliográficas lo que aumenta la utilidad práctica y el carácter didáctico del libro.

E. García Camarero

Henrici, Peter
"DISCRETE VARIABLE METHODS
IN ORDINARY DIFFERENTIAL
EQUATIONS".
Extracto del prefacio.

Este libro tiene la doble misión de servir como texto en la licenciatura y de presentar nuevos y significativos resultados teóricos.

Reflejando mi actitud hacia la enseñanza del análisis numérico, la

parte elemental del libro pone énfasis en los conceptos básicos y las cuestiones de interés intrínseco más que en la enumeración de técnicas especializadas. Mucho material de la parte no elemental, en especial lo referente a propagación de errores, es nuevo.

El libro tiene como origen las notas de un curso titulado "Métodos numéricos en ecuaciones diferenciales" que he dictado repetidas veces a matemáticos y a físicos e ingenieros con inclinaciones matemáticas.

El tema de este curso es equivalente al contenido en los Cap. 1 y 2 y la 1ª mitad del 5º Cap. de este libro.

Los conocimientos previos para este curso son-un primer curso de ecuaciones diferenciales, algo de análisis matemático y un barniz de cálculo en computadoras. (En la Universidad de California, Los Ángeles, esto último puede adquirirse tomando un "one-unit lower division", curso de codificación).

El equivalente en U. B. es un curso de AUTOCODE para Mercury o de FORTRAN en el Sistema IBM. (Pero no-imprescindible para el seminario).

Los problemas que se encuentran al final de cada capítulo son principalmente teóricos.

Es claro que un estudiante que desee comprender completamente los algoritmos desarrollados en el libro debe aplicarlos a problemas prácticos completos.

Sin embargo, la magnitud de un problema en el cual un estudiante puede razonablemente trabajar depende grandemente del equipo de cálculo que tenga a su disposición. Por esta razón la selección de problemas numéricos ha sido dejada, con algunas excepciones, al instructor.

INDICE

Introducción

PARTE I:

Métodos de paso simple para problemas de valores iniciales.

Cap. 1. - Método de Euler para una ecuación de primer orden.

Cap. 2. - Métodos generales de paso simple para una ecuación de primer orden.

Cap. 3. - Métodos generales de paso simple para sistemas de ecuaciones de primer orden.

Cap. 4. - Métodos de paso simple para sistemas de ecuaciones de orden mayor.

PARTE II:

Métodos de paso múltiple para problemas de valores iniciales.

Cap. 5. - Métodos de paso múltiple para ecuaciones de primer orden.

Cap. 6. - Métodos de paso múltiple para ecuaciones especiales de Segundo orden.

PARTE III:

Problemas de contorno.

Cap. 7. - Métodos directos para una clase de problemas de contorno no lineales de segundo orden.

Peter Henrici

Los Ángeles, California.
1961 den.

LEXICO

TERMINOLOGIA MULTILINGUE DEL TRATAMIENTO AUTOMATICO DE LA INFORMACION

De acuerdo a lo anunciado en los números anteriores de este boletín, a partir del presente ejemplar continuamos publicando una primera versión provisoria de la terminología mencionada en el epígrafe.

La comisión encargada de la aprobación de los términos que se han de emplear en español, que integran los Ings. Criado, Molina, Panich, Pollitzer y el Lic. Vella reiteran su solicitud de colaboración a todas aquellas personas interesadas en las cuestiones cu ya terminología quede encuadrada dentro de los límites de la compilación que se está realizando.

Las criticas y comentarios a la terminología adoptada provisionalmente deberán ser enviados a la sede de la Sociedad Argentina de Cálculo, a fin de que una discusión posterior de la comisión permita decidir si hay lugar a alguna corrección, a fin de incorporar ésta a la versión definitiva que se remitirá al IFIP.

40 DISEÑO LOGICO

4001

Lógica.

En la técnica de (0003). El esquema sistemático sobre el cual están basadas las interrelaciones del proceso en el diseño preliminar de un sistema (0003).

4002

Diseño Lógico.

La especificación práctica, derivada de la lógica (4001), o de las relaciones de trabajo entre las partes de un equipo.

4003

Elemento Lógico.

Dispositivo que a partir de un número específico de señales de entradas produce una o más salidas, en relación con las entradas y sirve, con otros elementos lógicos, para un diseño lógico particular (4002). La relación es usualmente simple y puede ser especificada breve y convenientemente. Ejemplos: compuerta de coincidencia (4006), sumador (4014).

4004

Símbolo Lógico.

Representación gráfica de un elemento lógico (4003).

4005

Diagrama Lógico.

Una representación gráfica del diseño lógico (4002), usando símbolos lógicos (4004).

4006

Compuerta de Coincidencia o Y.

Un elemento lógico (4003) que tiene al menos dos señales de entradas binarias y una sola señal de salida binaria, las relaciones entre las señales de entrada y salida son definidas por la (2610) y operación. Más simplemente, un circuito cuya señal de salida representa un 1 solo cuando todas las señales de entrada representan 1.

4007

Compuerta "0".

Un elemento lógico (4003) cuya acción cuando las dos condiciones de cada señal de entrada corresponden a un 1 y a un 0, representa el correctivo de Boole "0" (2607). Más simplemente, un circuito cuya señal de salida representa un 1 cuando una o más de sus señales de entrada representan un 1.

4008

Inversor.

Un elemento lógico (4003) que tiene sólo una entrada, la salida relacionada a la entrada por el operador de Boole "no" (2605). Más simplemente, un circuito cuya señal de salida representa un 1 cuando su entrada representa 0 y viceversa.

4009

Elemento "o" Exclusivo.

Un elemento lógico (4003) cuya acción representa el operador de Boole "o exclusivo" (2607). Más simplemente, un circuito de dos entradas cuya salida representa 1 cuando sus entradas difieren una de otra.

4010

Elemento Coincidente.

Un elemento lógico (4003) cuya acción representa el operador de Boole "equivalente" (2607). Más simplemente, un circuito de dos entradas cuya salida representa 1 cuando sus entradas son idénticas.

4011

Elemento de Retardo de un Dígito.

Un elemento lógico (4003) que retarda su señal de salida en un periodo de un dígito (4700).

1012

Medio Sumador.

Un elemento lógico (4003) con dos entradas y dos salidas a las cuales puede aplicárseles señales que representen un dígito de un número y un sólo sumando (2637) o un dígito de transporte (2625). Una señal de salida representa un dígito de la suma (2639), la otra representa un dígito de transporte hacia adelante.

NOTA 1: El nombre se debe al hecho que dos de tales elementos pueden ser combinados para formar un sumador. Ver Sumador, Nota 2.

NOTA 2: En una operación serial, el transporte debe ser retardado un período de un dígito (4700) de modo que tenga un significado correcto cuando es aplicado a la entrada. El término medio sumador es algunas veces aplicado a un elemento que comprende a un medio sumador como fue definido antes junto con un elemento de retardo (4011) para el transporte y una compuerta "0" (4007) para combinar el retardo del transporte y la entrada del sumando.

4013

Medio Sumador Binario.

(4012) un medio sumador operando en base 2, es decir, un elemento lógico (4003) que tiene dos señales de salida K y S (transporte (2625) y suma (2639)) y dos señales de entrada A y B tales que las señales de salida están relacionadas a las señales de entrada de acuerdo a la siguiente tabla:

Entrada	(A0011 {B0101
Salida	(K0001 (S0110

4014

Sumador de Tres Entradas.

En una operación digital, un elemento lógico (4003) con dos salidas y tres entradas a las cuales puede ser aplicada señales que represen

ten dígitos del sumando (2637), aumento (2638) y transporte (2625). Las señales de salida representan un dígito de la suma (2639) y un dígito a ser transportado hacia adelante.

NOTA: En operación en serie, el transporte de salida debe ser retardado un período de un dígito (4700) de modo que tendrá su significado correcto cuando se aplique a la entrada. El término "sumador de tres entradas" es algunas veces usado para un elemento que incorpora el retardado del transporte y que por consecuencia tiene sólo dos entradas para las señales generadas externamente, y una salida que representa el dígito de la suma.

4015

Medio Sustractor.

Un elemento lógico (4003) con dos salidas y dos entradas a las cuales se le puede agregar señales que representan un dígito de un número y un sólo sustrayendo (2650) o un dígito... (2626). Una señal de salida representa un dígito de la diferencia (2616), y el otro representa un dígito... a ser usado en la posición del dígito que tiene la próxima Significación mayor (1604).

NOTA 1: El nombre se debe al hecho de que dos de tales elementos pueden ser combinados para formar un sustraedor completo (4016).

NOTA 2: En una operación en serie (1212), la señal de salida... debe ser retardada un periodo de un dígito (4700) de modo que tendrá su significado correcto cuando sea aplicado a la entrada. El término medio sustraedor es algunas veces aplicado a un elemento que comprende un medio sustraedor como fue definido antes junto con un elemento de retardo de un dígito (4001) para el... y una compuerta "0" (4007) para combinar el retardo ... y ... señales de entrada.

4016

Sustraedor Completo.

Un elemento lógico (4003) con dos salidas y tres entradas a las cuales puede ser aplicada (5008) señales que representan dígitos (1202) del minuendo (2640), (2650) y (2626). Las señales de salida que representan un dígito de la diferencia (2641) y un dígito... para ser usado en la posición del dígito (1603) que tiene el próximo mayor significado (1604).

NOTA 1: En la operación en serie (1212), la señal de salida... debe ser retardada un período de un dígito (4700) de modo que tendrá su significado correcto (1604), cuando sea aplicado a la entrada. El término incorpora el elemento de retardo de un dígito (4001) para el... y por consecuencia tiene sólo dos entradas para señales generadas externamente, y una señal de salida la cual representa el dígito de suma.

NOTA 2: Dos medios sustractores pueden ser combinados para formar un sustractor (4016) aplicándole al primero señales que representan al minuendo y al.... Las entradas al segundo son la diferencia de salida del primero y la señal que representa al sustraendo. Como alternativa se usa un medio sumador (4012) para formar la suma (2639) del... y sustraer do, seguido por un medio sustractor para sustraer esta suma del minuendo.

4017

Sumador Sustractor.

Un elemento lógico (4003) que puede ser hecho para actuar como un sumador (4014) o como un sustractor (4016) en respuesta a una señal de control.

4018

Celda.

1. Unidad elemental de almacenaje; por ejemplo en memoria a núcleos magnético, un sólo núcleo (7004). 2. Almacenaje (4301) para una unidad de dato (0001), usualmente un bit (1614) o un dígito (1202).

4020

Registro.

Una memoria (4301), usualmente de una capacidad de una palabra (1205) y destinado para un propósito especial o propósitos especiales en el computador (0007).

4021

Registro de Línea de Retardo.

Un registro (4020) incorporado a una línea de retardo (5026), medio para la regeneración de señal (5017), y un canal de re-alimentación, de modo tal que el almacenamiento (4301) de datos en (1212) representado en serie puede ser logrado por circulación continua.

4022

Registro de Desplazamiento.

Un registro (4020) adaptado* para realizar desplazamientos (2621) por ejemplo un registro de línea de retardo (4021) cuyo tiempo de circulación puede ser aumentado o retardado a voluntad de modo de desplazar el contenido; o un registro compuesto de celdas binarias (4018) en las cuales los bits (1614) son transferidos desde una celda a la próxima por la aplicación de un pulso (4708) común a todas las celdas.

4023

Contador.

Un elemento lógico (4003), un registro (4020) o una locación de almacenaje (4303) para almacenar números, permiten a estos números ser aumentados o disminuidos en una unidad o por un número arbitrario, y frecuentemente capaz de ser puesto a cero. Ejemplo: un dispositivo en n estados estables que pueden ser disparado (conmutado) de un estado el próximo, o un almacenaje cíclico (5402) asociado con un medio sumador (4012).-

4024

Estatizador.

Un elemento lógico (4003) que transforma una serie temporal de estados, representando dígitos (1202) en una distribución espacial de esta dos simultáneos correspondientes.

4025

Dinamizador.

Un elemento lógico (4003) que convierte una distribución espacial de estados simultáneos representando dígitos (1202) en una serie temporal correspondiente.

4026

Circuito de Disparo Simple.

Un circuito que provee desde una entrada al azar una señal simple (5008) en forma y tiempo aceptable para una máquina particular.

4027

Nivel Habilitante.

Una serial (5008) que permite que una operación (2601) tenga lugar.

43 MEMORIA

4301

Memoria (almacenaje).

La retención de datos (0001) para su posterior referencia.

4302

Memoria.

Dispositivo en el cual los datos (0001) pueden ser introducidos y conservados por el tiempo que sea necesario para ser obtenidos cuando se lo desee.

NOTA: En muchos tipos de memoria los datos pueden ser obtenidos sin perder su registración.

4303

Locación.

Una posición en la memoria (4302) que contiene una palabra (1205) o parte de una palabra. Un registro (4020) es también una locación.

4305

Rango (alcance capacidad).

1. de una memoria (4302). El número de dígitos (1202), o palabras que pueden ser almacenadas. 2. (...). El límite superior o inferior de los números que pueden ser procesados en un registro del computador (4020). (Ver range (1008)).

4306

Rango o capacidad del Registro.

El rango (4305) de un registro (4020).

NOTA: Los términos Largo, Medio y Corto son algunas veces usados para distinguir imprecisamente entre los diferentes rangos de un registro.

4307

Lectura Destructiva.

Proceso de lectura que destruye la registración de los datos (0001) que han sido leídos (2003), por ejemplo, en ciertas memorias de núcleos (5809) la lectura reduce el estado de los núcleos a un estado normalizado (normalmente (4712)...).

4309

Reescribir.

En un sistema de lectura destructiva. Para retornar a la memoria (4302) los datos (0001) inmediatamente después que han sido leídos (2003).

4310

Lectura no Destructiva.

Un método para determinar el estado de la memoria de una celda (4018) sin cambiar el estado.

4311

Almacenamiento Externo (almacenamiento periférico).

Una memoria (4302) no permanentemente disponible para el computador (0007) pero que mantiene los datos (0001) en una forma aceptable para él, por ejemplo, memoria de cinta magnética (5509).

NOTA: La memoria interna (4312) y externa (4311) son términos tomados en sentido estricto sólo con referencia a una dada instalación.

4312

Almacenamiento.

Una memoria (4302) construida en un computador (0007) y directamente controlada por él.

4313

Sección de Entrada.

Parte de una memoria (4302) usada para la recepción de datos (0001) de entrada (2001). Un block del almacenaje usado para recibir la entrada (2001).

4314

Sección de Salida.

Parte de una memoria (4302) usada para la liberación de datos (00019 de salida (2002).
Un block del almacenaje usado para liberar la da (2002).

4315

Memoria de Trabajo.

Una memoria (4302) en un computador (0007) que comprende aquellas locaciones (4303) designadas por las locaciones (3006) en instrucciones aritméticas.

4316

Memoria de Acceso Libre.

Una memoria (4302) que consiste de una o más locaciones (4303) donde el tiempo de espera (4723) es despreciable en comparación con otros tiempos de operación (2601).

4317

Memoria de Acceso al Azar.

Una memoria (4302) destinada a disminuir el efecto de variación del tiempo de acceso (4724) en una secuencia arbitraria de dirección (3006).

4318

Memoria Rápida.

Término impreciso que se refiere a una memoria (4302) cuyo tiempo de acceso es relativamente corto.

4319

Memoria Auxiliar.

Una memoria (4302) de mucho mayor capacidad (4305) que la memoria de trabajo (4315), pero con mayor tiempo de acceso (4724). Los datos (0001) pueden ser transferidos en bloques (1210) entre dos memorias.

NOTA: Con las técnicas corrientes, el aumento del tiempo de acceso resulta menor costo para una dada capacidad.

4320

Memoria Compensadora (almacenamiento compensador).

Memoria (4302) utilizada para compensar una diferencia en la velocidad (o flujo) de datos (0001), o los tiempos de ocurrencia de..., cuando se transmiten los datos de un dispositivo a otro. Puede abreviarse a "buffer" (4320) cuando el texto hace su significado claro.

45 UNIDADES DEL EQUIPO

4501

Unidad de Control.

La porción de un dispositivo automático (0003) de procesamiento de datos que dirige la secuencia de operación (2601), interpreta el código de instrucción (3201), y actúa sobre los circuitos para la ejecución de la instrucción.

4502

Registro de Instrucciones.

Un registro (4020) en la unidad de control (4501) que almacena la instrucción en curso (3201) del programa (3001) de modo que pueda ser interpretada por la unidad de control.

4503

Registro de Próxima Instrucción.

Un registro (4020) de modo que a partir de su contenido (3007) se deriva la dirección (3006) de la próxima instrucción (3201).

4504

Registro Índice.

En un computador (0007) con facilidades para modificación automática (4504). Un registro (4020) que mantiene el modificador (3228).

4505

Indicador.

Dispositivo que puede ser puesto en uno, dos o más estados de acuerdo a los resultados de un proceso previo, y que, en consecuencia puede ser usado para determinar la selección de un proceso alternativo. Por ejemplo, un indicador de "overflow" es uno que es puesto cuando ocurre "overflow" (2633).

Dispositivo que puede ser puesto en un estado prescripto de acuerdo a resultados de un proceso previo, y que en consecuencia puede ser usado por la unidad de control (4501) para determinar la selección en un proceso alternativo. Por ejemplo, un indicador de "overflow" es un indicador que es puesto para determinar cuando ocurre "overflow" (2633). El estado de un indicador puede o no ser displayed.

4506

Unidad Aritmética.

Sección de un computador (0007) donde son realizadas las operaciones aritméticas (2603), lógicas (2604) de saltos (2621).

4507

Acumulador.

Un registro (4020) en un computador digital (0014) donde los resultados de una operación aritmética común son almacenados. Puede adicionalmente tener corridas lógicas o propiedades modificatorias.

1. Dispositivo que almacena el aumento (2638) y que recibe los sumandos (2637) de la suma de dos números y almacena la suma. 2. Más simplemente, un registro en una unidad aritmética que mantiene un operando, como medio para realizar varias operaciones involucrando este operando y otro; el resultado no necesariamente permanece en el acumulador.

4508

Panel de Control.

Conjunto de controles manuales y displays, etc., para utilizar el operador, de un computador (0007).

4509

6814

Panel de Control.

Componente de algunas máquinas de procesamiento de datos (0002) similar en principio al cuadro de distribución eléctrica de un teléfono manual. Las terminales de entrada y salida de las unidades en la máquina son permanentemente conectadas a las (4520) en el panel, y las interconexiones entre las unidades, prescriptas por el problema o el trabajo, son hechas por los (4521) o por los (4519).

NOTA: En algunos equipos de tarjetas perforadas (6001) el término panel de control (6814) es usado para este componente.

4510

Un panel de control (4509).

4512

Generador de Sincronismo.

Unidad que genera señales (5008) de reloj (4707).

4513

Pulso-reloj de Sincronismo.

1. Dispositivo de regulación usado para recordar automáticamente y controlar el uso del equipo de (0003). 2. Dispositivo regulador usado para regular la operación de un equipo de acuerdo con una escala de tiempo definida. Por ejemplo, cuando, un computador digital (0014) es usado como (4516), el pulso sirve para iniciar la operación en el intervalo de tiempo requerido. 3. Ver también sedal de pulso (4707).

4514

Comparador.

1. Dispositivo para comparar dos transcripciones diferentes del mismo dato (0001) para verificar (2802) la exactitud (1019) de la transcripción. 2. Dispositivo para comparar dos señales (5008) y dar una salida dependiendo de alguna relación entre ellos, por ejemplo, cantidades numéricas, cuando una es mayor que, igual a, o menor que, el otro.

4515

Convertidor.

Unidad que cambia la representación de dato (0001) de una forma a otra de manera de hacerla disponible o aceptable para otra máquina, por ejemplo, unidad que cambia los datos perforados en tarjetas (6001) en datos recordados en cinta magnética. Un convertidor puede también editar el dato.

4516

(Registrador).

Dispositivo que registra procesos físicos y eventos automáticamente usualmente con respecto al tiempo.

4517

En procesamiento de datos (0002). Dispositivo que automáticamente presenta el estado de varios procesos, condiciones o estados físicos y transfiere las cantidades obtenidas a un dispositivo registrador o de control.

4518

Consola.

Pupitre que comprende un panel de control (4508).

4519

Conector (m).

Dispositivo usualmente conectado a los conductores en un lugar flexible y usado para hacer conexión a un (452.0) conector (h).

NOTA: El uso de este término para significar un (4521) es usado en máquinas de tarjetas perforadas (6814)....

4520

Conector (hembra).

Dispositivo usado generalmente para terminar la conexión permanente de un circuito, el acceso al cual es obtenido por la inserción en él de un (4519).

4521

Cable de Conexiones.

Cuerda flexible terminada en ambos extremos por conectores machos (4519) y usado para interconectar los conectores hembras (4520) de un tablero (4509). Los conductores de la cuerda flexible son aislados y pueden incluir elementos eléctricos pasivos.

4522

Decodificador.

1. Dispositivo capaz de decodificar (2203) un grupo de señales (5008) y generar otras señales que pueden poner en marcha una operación (2601). 2. En equipo de procesamiento de datos (0002). Dispositivo con un número de líneas de entrada y salida en las cuales una combinación específica de señales de entrada (5008) origina una línea particular de salida para dar una señal.

4523

Encodificador.

El equipo de procesamiento de dato (0002). Dispositivo con varias entradas de las cuales sólo una entrada es excitada por vez y cada entrada produce una combinación específica de salida.