

文献紹介

70-51 主項を生成するための新しいアルゴリズム

James R. Slagle, Chin-Liang Chang & Richard C. T. Lee: A New Algorithm for Generating Prime Implicants [IEEE Trans. on Computers, Apr. 1970, Vol. C-19, No. 4, pp. 304~310] Key: clauses, conjunctive (disjunctive) normal forms, consistent row sets, frequency orderings, literals, minimal sums of a Boolean function, prime implicants (implicates), semantic trees, sproutings, terminating (nonterminating) nodes

この論文は与えられたブール関数のすべての主項を生成するアルゴリズムについて述べている。このアルゴリズムは既存のそれに比べて、非常にすみやかにすべての主項を生成する場合が多い。このアルゴリズムはすべての主項を見出すことが証明されているが、主項でないものまで与えてしまうこともある。しかしながら、文字（入力変数の極性まで指定したもの）に関する frequency ordering を用いれば、そのような項はほとんど生成されないことが実験からわかっている。さらに、このアルゴリズムを用いてブール関数の最小和を求めることができる。

このアルゴリズムは、与えられたブール関数の semantic tree と呼ばれるものをつくることに帰着する。その一例はつきのごとくである。

$$F = (A + \bar{B} + C + D)(\bar{A} + B + \bar{C} + D) \cdot (A + B + C + \bar{D})(A + B + \bar{C} + D)$$

の主項を求めることをこのアルゴリズムで実行すると図1のような semantic tree ができる。

したがって、与えられたブール関数 F の主項は $\{AB, AC, AD, BC, BD, C\bar{A}\bar{D}, \bar{C}\bar{B}\bar{D}, \bar{D}\bar{A}\bar{B}\}$ である。

また、このアルゴリズムの特色は

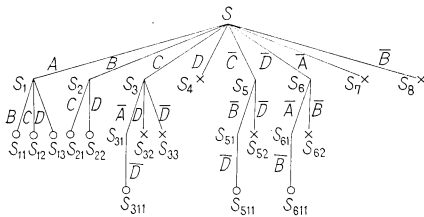


図1 F の semantic tree

1. 効率的である。ブール関数の conjunctive (または disjunctive) form が短いときにとくに効率がよい。
 2. semantic tree の成長の仕方からみて、メモリーを大量には必要としない。
 3. semantic tree を成長させるための演算が少なくすむ。
 4. このアルゴリズムでは同じ主項を重複して生成することはない。
 5. ブール関数を標準形で与える必要がない。
- である。 (宇都宮公訓)

70-52 EXCLUSIVE OR, LOGICAL EQUIVALENCE スイッチング回路の最小化

Amar Mukhopadhyay & Greg Schmitz: Minimization of EXCLUSIVE OR and LOGICAL EQUIVALENCE Switching Circuits [IEEE Trans. on Computers, Feb. 1970, Vol. C-19, No. 2, pp. 132~140] key: APL, cellular logic arrays, maximal clique of a graph, minimization algorithm, modulo 2 sum-of-products (or sums), Reed-Muller canonic forms

この論文は Reed-Muller 標準形に基づいたスイッチング回路の最小化アルゴリズムについて述べている。とくに、入力変数の極性を固定して、単出力・多出力スイッチング関数の最小 modulo 2, complement modulo 2 sum-of-products (もしくは sums) 表現を得るアルゴリズムを提案している。

ここで議論されている最小化方法の本質的思想は、Reed-Muller 展開における係数の極性関数 (polarity function) と呼ばれるものを得ることにある。入力変数の極性固定の条件で、これらの関数は、係数が“ある”か“ない”かという論理条件を与える。つぎに、最小表現を得るための最良の極性を決定する。このステップは linear nondirected graph の最大クリークを求めることに帰着するが、そのための強力なアルゴリズムが最近開発されている。

例題を用いてこのアルゴリズムの大略を紹介するとつぎようになる。

$$(1) f(x_1, x_2, x_3) = x_1'x_2'x_3' + x_1'x_2x_3'$$

$$\begin{aligned}
 &+ x_1 x_2 x_3' + x_1 x_2 x_3 \\
 &= a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_3 x_3 \oplus a_4 x_1 x_2 \oplus a_5 x_1 x_3 \\
 &\quad \oplus a_6 x_2 x_3 \oplus a_7 x_1 x_2 x_3
 \end{aligned}$$

が最小形になるように入力変数の極性、係数 $a_i (i=0 \sim 7)$ を決定する。

ただし、 $x_i = x_i$ or x_i' \oplus は exclusive or.

(2) 極性関数は

$$\begin{aligned}
 a_0(f) &= x_1 x_2 x_3 \oplus x_1 x_2 x_3' \oplus x_1' x_2' x_3 \oplus x_1' x_2' x_3' \\
 a_1(f) &= x_2 x_3 \oplus x_2' x_3 \oplus x_2' x_3' \oplus x_2' x_3' \\
 &= x_2 x_3 \oplus x_2' x_3' \\
 a_2(f) &= x_1 x_3 \oplus x_1 x_3' \oplus x_1' x_3 \oplus x_1' x_3' = x_1' \\
 a_3(f) &= x_1 x_2 \oplus x_1 x_2' \oplus x_1' x_2 \oplus x_1' x_2' = x_1 \\
 a_4(f) &= x_3 \oplus x_3' \oplus x_3 \oplus x_3' = 1 \\
 a_5(f) &= x_2 \oplus x_2' \oplus x_2' \oplus x_2' = 1 \\
 a_6(f) &= x_1 \oplus x_1 \oplus x_1' \oplus x_1' = 0 \\
 a_7(f) &= 0
 \end{aligned}$$

(3) a_0, a_1, a_2, a_3 は値 0 または 1 をとるが、できるだけ多くの $a_i (i=0 \sim 3)$ が 0 になるように、入力変数の極性を決定する。

		x_3			
		0 0	0 1	1 1	x_2
					1 0
	0	0, 1, 2	0, 2	1, 2	2
x_1	1	1, 3	0, 3	0, 1, 3	3

これより (x_1', x_2, x_3') で

$$a_0=0, a_1=0, a_2=1, a_3=0$$

もしくは (x_1, x_2, x_3') で

$$a_0=0, a_1=0, a_2=0, a_3=1$$

が得られる。ただし、わかりやすくするために、ここでは Karnaugh Map を用いたが、大きな問題に対しては、最大クリークを求めるアルゴリズムが使用される。

(4) したがって、 f は極性 (x_1', x_2, x_3') で

$$a_0=a_1=a_2=a_6=a_7=0, a_3=a_4=a_5=1$$

すなわち

$$f = x_2 \oplus x_1' x_2 \oplus x_1' x_3'$$

もしくは、極性 (x_1, x_2, x_3') で

$$a_0=a_1=a_2=a_6=a_7=0, a_3=a_4=a_5=1$$

すなわち

$$f = x_3' \oplus x_1 x_2 \oplus x_1 x_3'$$

となる。

(宇都宮公訓)

70-53 セルラー APL 計算機のシステム設計

Kenneth J. Thurber & John W. Myrna: System Design of a Cellular APL Computer [IEEE Trans. on Computer, Apr. 1970, vol. C-19, No. 4, pp. 291 ~ 303] key: APL, array processors, cellular arrays, parallel processors, special purpose computer

この論文は、APL (K. E. Iverson が開発した言語をプログラム言語としてインプリメントしたもの) で書かれたプログラムを効率よく実行する能力を持つ計算機のシステム設計について述べている。このシステムは cellular logic array にマトリックスやベクトルを格納するという新しい計算機設計技術に基づいて組織されている。スカラー、混合、合成固定小数点演算のほとんどすべてがこの機械によりインプリメントされている。この機械の詳細な論理設計については完全には述べていないが、必要な論理、メモリーはその仕様が与えられている。また、おのおののハードウェア・セルが実行できなければならない機能を明細に述べている。ほとんどすべての APL 演算がフローチャートで示されており、その演算の実行により生じるデータの流れが詳しくわかるようになっている。

図 1 は実際の APL セル形計算機システムを図示したものである。その各成分の機能は、つぎのようである。

(1) matrix logic-in-memory unit (MLIM)

以下に述べる 16 個の memory array のおのおのと大きさの等しい 32×32 の基本セルの array でありその中に格納されている情報に対して演算をほどこすことができる。おのおのの基本セル i, j は 4 つのシフト・レジスタ $A_{i,j}, B_{i,j}, C_{i,j}, T_{i,j}$ を持ち、ある種の算術演算、論理演算を行なうことができる。

(2) memory array (MA1, MA2, ..., MA16)

この計算機のメモリーは 16 個のメモリー・アレーからできている。各メモリー・アレーは 32×32 のメモリー・セルからできている。おのおののセルは 1 語をたくわえることができる。

(3) instruction memory unit (IMU)

この計算機で用いられるすべての命令を格納している read-only memory である。すべてのマイクロプログラムがこの中に格納されている。

(4) routing logic (RL)

マトリックスやベクトルを移送する間にマトリックスやベクトルの generalized indexing を行なうこと

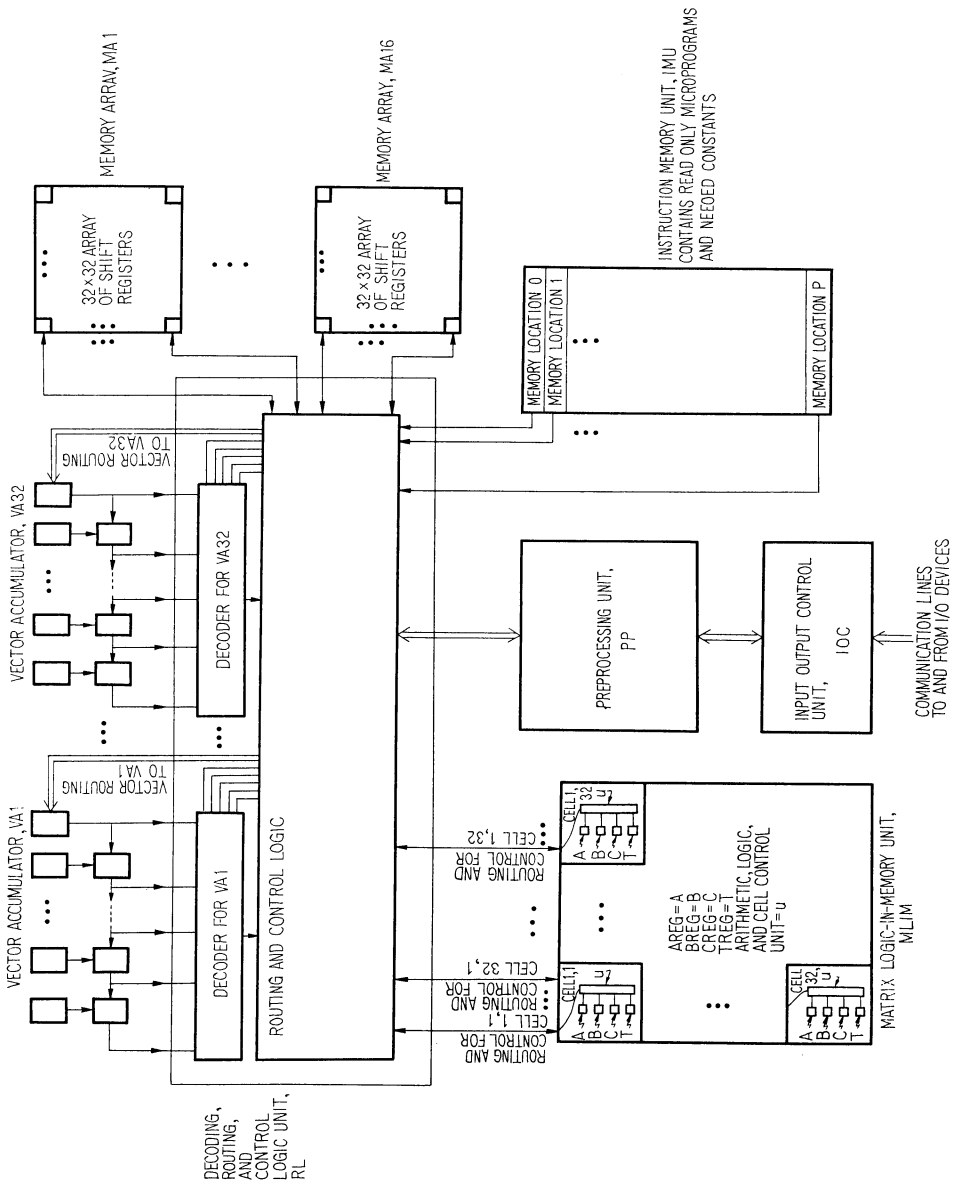


図 1 セルラー APL 計算機システム

ができる。また、メモリー・アレー全体、もしくはその一部を MLIM の 1 つの plane 上に移送する機能、その他、重要な機能をいくつか持っている。

(5) vector accumulator (VA1, VA2, ..., VA32)
32 個の vector accumulator があり、そのおののはすべての APL reduction 演算を行なうことができる。

(6) input-output controllers (IOC)

ユーザーが利用する入出力装置を制御する装置であ

り、広い融通性を持っていなければならない。

(7) preprocessor (PP)

基本的役割は、ハードウェアの APL meta-interpretor であることである。与えられた APL ステートメントやプログラムはこの meta-interpretor により解釈され、一連の並列機械動作が行なわれる。

(杉本迪雄)

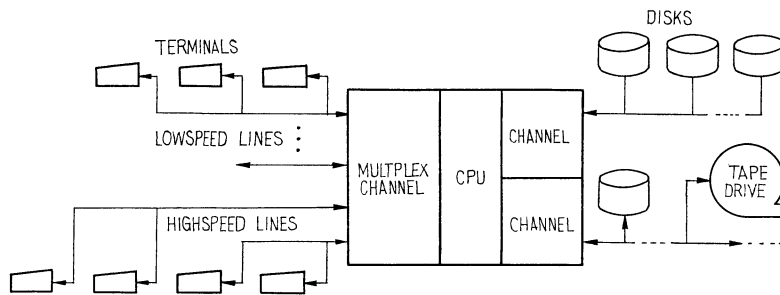


Fig. 6 Real-time, terminal oriented system

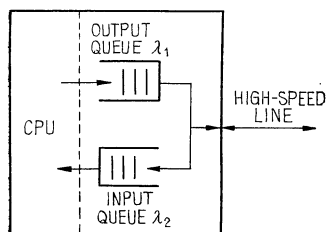


Fig. 7 Queuing model of communication line

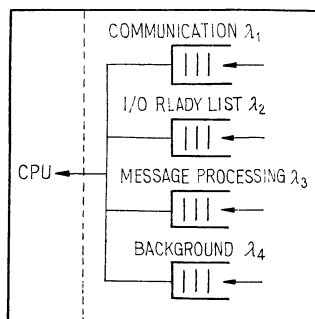


Fig. 8 CPU queuing model

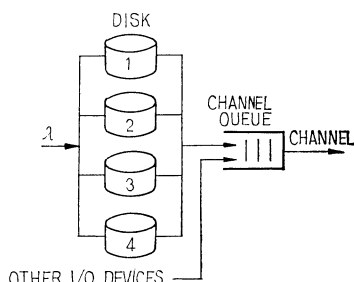


Fig. 9 Model of a disk system

70-54 コンピューティングシステムにおける単一サーバの待合せの扱いについて

W. Chang: Single-server queuing process in Computing Systems (IBM. SYST. J., Vol. 9, No. 1, 1970 pp. 36~71). key: basic concepts, arrival pattern, service pattern, imbedded Markov chains, waiting-time, response-time, queue-size, busy-period distribution

本論文は、コンピューティングシステムにおける単一サーバの待合せ問題についての基本事項を集大成したもので、約 35 ページの容量に整理されている(計算例含)。著者はこの分野では著名な W. Chang 氏である。

論文の構成は、各種の基本的な待合せ問題に関連する諸概念の解説、コンピューティングシステムをサブシステムに分割し、それらへの待合せモデルの導入、その解析結果、数値計算例などから成っている。また巻末には約 40 編のこの分野で有名な古くからの論文がリストアップされており、本文中の基本的な数式のいくつかについては Appendix で、それらを導出している。

最近の、とくにリアルタイム性を要求される TSS などにおいては、複数サーバが次第に主流となりつつあるが、基本はやはり単一サーバの特性であると考えられる。その意味で、本論文はシステムの設計初期などに手近かに置いて簡便に参照するのに有益であろう。

本論文の単一サーバモデルの条件、およびおもな結果を以下にまとめる。

- (1) この論文では、ランダム入力、一般サービスタイム、優先権を持つ単一サーバの待合せを解析する (M/G/1 モデル)。
- (2) 待時間、応答時間、および繁忙期間 (busy-period) の各分布はラプラス変換の形式で導出している。
- (3) 待行列長の分布は、待行列長の母関数より導出可能なように与えられている。

Fig. 6~9 に、論文に記載されている適用例のシステム構成、およびモデルを示す。Fig. 6 がシステム全体の構成図で、Fig. 7~8 が各サブシステムに分解し、おのおのに適用したモデルである。これらについて詳細な計算の手順とその計算例が示されている。

(花田収悦)

70-55 自然言語を用いて計算機と質問 応答するためのシステム

R. F. Simmons: Natural Language Question-Answering Systems: 1969 [Comm. ACM, Jan., 1970 Vol. 13, No. 1, pp. 15~30] key: question-answering system, natural language, artificial intelligence, language processing, fact retrieval, semantics.

これは、計算機プログラムとして作成された、自然言語による質問応答システムを集め、比較検討し、それらの中で用いられた syntax, semantics, logical analysis などに関する確立されつつある方法を示したものであり、同じ著者による 1965 年の survey に続くものである。ここでは、この著者は次のように考えている。

これらの質問応答システムは、Chomsky の生成文法の考えたと、コンパイラ概念に基づいており、また、COMIT, LISP などの言語と TSS の普及によりシステム作成が容易になり、方々に活発に研究が行なわれた。

ほとんどのシステムが、どの一つをとってみても、構文解析、意味分析、推論過程、出力英文作成の各側面すべてについて完全であるとはいえないが、各側面それぞれのための方法はそのどれかにおいては開発されている。

構文解析については、基本的な方法はかなり明らかにされたといえる。

意味分析は、希望は持てるが、また十分にできてい

るとはいえない。‘pattern-operation rule’ の使用が鍵になると考えられる。意味の内容の表現方法としては、Chomsky のいう深層構造は自然言語の基礎的研究には適しているが、計算機に処理させるためのデータ構造としては適当でないので、述語論理の形式による表現法などの代案がいくつか用いられている。意味内容の各項目を探し出すには、LISP, SNOBOL, SLIP の機能を用いた、パターンによる引き出し法が用いられているが、将来連想記憶装置が使えるようになればより好都合である。また、現在の諸システムでは、記憶できる単語の数がせいぜい数百語程度であるが、実用に供するためには数万語くらい必要であり、このためには 10^6 words 程度の主記憶容量を要する。しかし、この容量が確保でき単語の数が増えると、新たに量的な問題が生じるであろう。

推論過程については、演繹論理を行なうものもあるし、帰納論理を行なうものもできはじめている。

出力英文作成については、十分なものがあまりないが、ここでも“pattern-operation rule”の採用が鍵となるだろう。単文だけでなく文間の結合や参照の問題もまだ残されている。

以上総合すると、1965 年以來かなりの進歩が、どの側面に対する方法にもあったといえる。しかし、意味分析には本質的な問題が未解決のまま残っている。そして扱える単語の数の増大に伴って新たな問題が生じてくることに注意しなければならない。

(亀田 寿夫)

 ニ ュ ー ス

IBM 社システム/370 を発売

7月1日 IBM は新機種を2モデル発表した。システム/360 の基本構造を拡張して開発され、モデルに165と155とある。中央演算処理装置の基本マシン・サイクル・タイムは、それぞれ80+1秒と115+1秒。標準命令セットとして、システム/360 汎用命令セットの他に、新しい汎用命令6種類、拡張精度浮動小数点命令、制御用命令が加わっている。また命令ユニットと実行ユニットが独立であるので、命令の取出し、解読、オペランドの取出しと、命令の実行が並行して行なわれる。マイクロ・プログラムのための記憶機構として、Read Only Storage-ROS の他に、Writable Control Storage-WCS が用いられ、マイクロ・プログラムの変更が容易である。記憶装置は、主記憶装置と緩衝記憶装置 (Buffer storage) の2段階になっており、緩衝記憶装置は、8K バイトで16K バイトまで拡張でき、サイクル・タイムは80+1秒である。主記憶装置は、3,072K バイトと2,048K バイトまでの容量がある。IBM は同時に新しい周辺装置も発表した。1分間2,000行の印刷装置と容量8億字のディスク装置である。新しい機能はOSによりサポートされ、7,000 シリーズ Emulator が OS のもとの稼動する。

通研で高速、低電力の LSI を開発

日本電信電話公社電気通信研究所では、このほど高速の情報処理装置に適した大規模集積回路 LSI として、1個のシリコン・チップ上に高速論理回路を105ゲート集積し、ゲートあたりの遅延時間約2ns、消費電力4mW という高速・低電圧のモノシリック形 LSI を設計し、日立製作所がこの試作に成功した。

従来、低速の MOS トランジスタによる LSI は、多くつくられてきたが、高速バイポーラトランジスタを使用し、これだけの集積度と高速性を同時に、しかも低電力で実現した LSI は世界的にもこれが初めてである。

この試作 LSI は、4ビット構成で、加減算および16種の論理演算の機能が集積された。いままでの IC でこれと同じ機能を持たせるには IC 約50個、消

費電力約5Wを必要とするので、この試作品を使用すると、実装容積1/5~1/10で、消費電力約1/10ですむことになる。したがって、装置の容積や発熱が問題となる大型計算機において、高速化・経済化に大きく貢献できるものと、関係者は期待している。

試作品の特性概要は以下のとおりである。

機 能	4ビット加算、減算、論理、キャリアスキップ
ゲート数	105個
演算時間	加算 20ns
端子数	40ピン
電源電圧	1.6~2.0V
チップサイズ	5mm×5.2mm
全消費電力	400mW (電源安定化回路含む)
外形寸法	24mm×27mm フラット形ケース

バブルドメイン素子用材料の量産技術の開発

日本電気はバブルドメイン (あわ磁区) 素子用のオルソフェライト単結晶を赤外線加熱型フローティングゾーン単結晶製造装置を用いて量産することに成功した。バブルドメイン素子の実用化の鍵は、大型の単結晶が安価に得られるかどうかにあった。このたび、量産技術によって、棒状 (直径8mm、長さ50mm) で結晶中に素子の動作上欠陥のないイットリウムオルソフェライト単結晶を8時間で製造できたので、バブルドメイン素子の実用化が風飛的に前進したといえる。

素子中の磁区が安定に存在すること、磁区の発生・移動が可能で、磁区間の相互作用があることから、磁区の有無を2進情報に対応させると、この素子は記憶機能と論理機能を合わせ持つこととなり、情報処理や通信の分野に広い用途を有している。

たとえば、記憶機能に着目すると、バブルドメイン素子を順次アクセスメモリとして使用することができる。量産に成功したイットリウムオルソフェライト単結晶から切り出された薄板で、数千ビットのシフトレジスタを構成できる。この装置は磁気ひずみ遅延線や LSI と競合関係にあるが、情報の不揮発性、動作時の低消費電力および低価格であることからかなりの優位性を持つ。

さらに高い記憶密度を有するオルソフェライトやガ

ーネット型フェライトの製造にも量産技術が適用できるので、困体化された外部メモリも、近い将来に実現できるものと思われる。

電気試験所が電子技術総合研究所に所名変更

通商産業省工業技術院電気試験所は、このたびの機構改革により、7月1日付けをもって所名が電子技術総合研究所（略称は電総研）に変更された。「電気試験所」という名前は明治24年（1891年）以来の伝統をもつものであるが、誤解を生じやすい名前であることは事実であり、「名は体を表わす」ように近づける動きのまえに、ついに別れを告げることになった。また、今回の機構改革では、時代の流れを反映して情報処理部門も拡充され、従来の電子計算機部と制御部とに加えてパターン情報部とソフトウェア部が新設された。

国際会議案内

1970年8月24—28日

IEIP World Conference on Computer Education, Amsterdam. Sponsor: IEIP Technical Committee for Education and Administrative Data Processing Group. Chm: A. A. M. Veenhuis, Sec. Gen., IEIP Conf. Cptr. Educ. 1970, 6 Stadhouderskade, Amsterdam 13, Netherland. US Reps: W. F. Atchison, Mrs. S. Charp, D. Teichrow.

1970年9月1—3日

ACM NATIONAL CONFERENCE, New York Hilton, New York City. Conf. Chm: Sam Matsa, IBM Corp., 410 East 62 St., New York, NY 10021. Prog. Chm: Robert E. Bemer, General Electric Co., 13430 North Black Canyon Highway, Phoenix, AR 85029.

1970年9月7—11日

Sixth International Congress on Cybernetics, Namur, Belgium. Contact: Secretariat, Association Internationale de Cybernetique, Palais des Expositions. Place Andre Rijckmans, Namur, Belgium. (✂切: 2月1日)

1970年9月9—18日

International Federation for Documentation

(FID) Annual Conference and International Congress on Scientific Information, Buenos Aires. Contact: USNCFID Secretariat, Nat'l Academy of Sciences, 2101 Constitution Ave., N. W., Washington, DC 20418.

1970年9月14—18日

7th International Symposium on Mathematical Programming, The Hague, Holland. Sponsors: ACM SIGMAP, in conjunction with many other societies. Contact: G. Zoutendijk, c/o Computing Center. University of Leiden, Stations-plein 20, Leiden, The Netherlands.

1970年10月4—9日

American Society for Information Science, 33rd Annual Meeting, Bellevue Stratford Hotel, Philadelphia. Sponsor: ASIS. Contact: Kenneth H. Zubriskje, Jr., Biosciences Information Services of Biological Abstract. 2100 Arch. St., Philadelphia, PA 19103.

1970年10月28—30日

Eleventh Annual Symposium on Switching and Automata Theory, Santa Monica, California. Sponsors: IEEE Computer Group and Department of Systems Science of the University of California, Los Angeles. Contact: papers, Peter Weiner, Department of Computer Science, Yale University, New Haven, CT 06520; arrangements, J. W. Carlyle, Department of System Science, University of California, Los Angeles, CA 90024. (✂切: 5月15日)

1970年11月17—19日

1970 Fall Joint Computer Conference, Astrohall, Houston, Texas. Sponsors: AFIPS. Gen. Chm. Robert Sibley, Jr., Dept. of Computer Science, University of Houston. Cullen Blvd., Houston, TX 77004. Program chm: Larry Axsom, IBM Scientific Center, 6900 Fannin St., Houston, TX 77025. (✂切: 4月10日)

1971年8月3—5日

ACM National Conference……Vol. 11, No. 3 を参照のこと。

1971年8月23—28日

IFIP Congress 71……Vol. 11, No. 3を参照のこと
(〆切: 11月30日)

1971年1月19—21日

Mexico 1971 International IEEE Conference on Systems, Networks and Computers, Oaxtepec, Mor., Mexico. Sponsor: Region 9 and Mexico

Section of IEEE. Theme: Systems Engineering, Systems Science, Management Science, Operations Research, Network Theory, Integrated Circuits, Information Processing, Computer Science, Automatic Control, Communication Systems, Transportation Systems, Power Systems Conference co-chm: M. A. Murray-Lasso, National University of Mexico Robert W. Newcomb, University of Maryland. (〆切: 8月31日)

連絡先: 川崎市下沼部 1753

日本電気株式会社 中央研究所内 渡部 和

雑 報

AICA 国際シンポジウム論文募集のお知らせ

主 催 アナログ技術研究会 (会長 沢井善三郎)

後 援 情報処理学会, AICA (国際アナログ計算連盟), 米国シミュレーション学会, 日本学術会議, 電気通信学会, 電気学会, 日本機械学会, 計測自動制御学会, 日本ME学会, 日本自動制御協会, 日本科学技術連盟

場 所 東京

開催時期 1971年9月3日(金)～7日(火)

テ ー マ Simulation of Complex Systems
アナログ・ハイブリッドまたはデジタルシミュレーションの新技术およびその工学, 物理・生物・経済・社会などのシステムへの応用

申 込 み 1～2頁の論文梗概 (double-spaced,

typewritten)で, 1970年1月15日申込〆切。(受理の可否は1971年3月中旬までに通知いたします。)

論文送付先および使用言語

原則として英語にかぎる。

東京都渋谷区千駄ヶ谷5の10の11

日本科学技術連盟内 アナログ技術研究会 電話 東京 (03)341-5221(代)

第13回標準化全国大会

(財)日本規格協会主催の「標準化全国大会」が **World Standards Day** として, 来る45年10月13日～16日に下記要領で行なわれますので, 同協会内全国大会事務局へ詳細お問合わせのうえ, ご参加ください。申込〆切9月12日。電話 東京(03)583-8001(代)。

- 10月13日(火) 標準化文献賞贈呈・特別講演・討論会。
- 10月14日(水), 15日(木) 研究・体験発表
- 10月16日(金) 工場見学会。

昭和45年度役員

会 長 高橋秀俊

副 会 長 大泉充郎, 緒方研二

常務理事 大野 豊, 金田 弘, 関口良雅
野田克彦

理 事 浦 昭二, 尾関雅則, 後藤英一
高橋 茂, 高柳 晃, 中原啓一

監 事 藤井 純

編集幹事会

担当 常務理事 大野 豊, **理事** 浦 昭二

幹 事 石田晴久, 伊藤雅信, 井上誠一, 遠藤 誠,
大須賀節雄, 草鹿庸次郎, 末包良大, 近谷英昭,
筑後道夫, 塚田啓一, 戸川隼人, 林 達也,
淵 一博, 穂鷹良介, 真子ユリ子, 矢島敬二, 吉沢 正, 渡辺一郎