

3. 質問回答過程とその機械化

大阪大学 大型計算機センター 磯本 征雄

はじめに

質問回答過程の機械化の試みは、これまでも様々のものであるが、その手法としては決定論的なものが多い¹⁾。一方、我々の経験の中には、確率論的取扱いの適している事柄も決りして少ない。ここでは、グラフ理論²⁾及び確率論的³⁾に整理された定式に従って質問回答をおこなうシステムについて述べる。システムを取扱う事象は、擬順序集合²⁾ V の要素であり、事象 $u, v \in V$ 間には、2項関係 R_{ij} 及び関係の強さ R_{ij} が確率論的に与えられている。システムは、データの具体的意味とは独立に、データの構造のみを解析する。したがって、その振舞いは一般的であり、応用範囲は広いと思う。

以下、システム開発の動機からその応用に至る過程を具体的に述べる。ただし、このためにシステムの特徴を説明する上で、その一般性を損う恐れはないであろう。

1. 質問回答システム開発の具体的動機

大型計算機センターには、プログラム・デバッグ手法の助言のために、プログラム相談員が居る。未熟なプログラマーが、デバッグ中のプログラムを持ってプログラム相談員の所へ相談に行けば、相談員はプログラマーにデバッグについて、具体的に適切な指示を与える。相談員が指示を与えるに至るまでには、相談員はプログラム誤動作の症状から、プログラムのバグの所在を模索、類推する。そして、その間に、相談員とプログラマーの間で多様な、しかし形式的な質問回答の手続きが繰り返される。

質問回答システム開発の最初の動機は、これら相談員の行なう模索・類推・思考と質問回答の過程を計算機により実現することであった。プログラム相談員の類推・思考の特徴は、次のとおりである。

- ① 相談員の類推・思考の手順は、ほとんど一定している。相談員は、具体的対象となっているプログラムの構造を吟味し、過去の経験をふまえたうえで、明瞭に確認できる事象を先頭に、それに続くできる限り多くの可能な事象を類推する。そして、可能性の強弱を考えながら、相談員はプログラマーと質的向回答を繰り返してバグの可能性の範囲を狭めてゆく。この過程は、推移確率を持つ樹状構造を主体とした有向グラフで表現できる。
- ② 相談員の類推・思考に参与する事象の範囲は、一般向問題解決プログラムの例に比較して狭い。特に、プログラム・デバッキングに限定するならば、質的向回答の際に使われる言葉の範囲は狭い。
- ③ 類推・思考の範囲が限られてはいるが、一人の人間の記憶にたよるには、その可能性は余りに多い。

以上の状況から判断して、プログラム相談員の役割りを計算機で実現する試みは、思考や質的向回答過程の機械化のための一試みとして適当であると考えた。

2. プログラム相談システムの設計目標

確率論的意味を持つ有向グラフ、特にそれがマルコフ連鎖になっている場合、単に頂点の一つ一つをたどることだけでなく、任意のステップ数の推移行列を用いることによって、ある程度巨視的に状況の把握が可能であり、その推移のステップ数の増減によって、視野を狭めたり広めたりすることができる。さらに、推移の極限状態の分布（定常分布）が存在する場合には、それを求めることによって、そのグラフ全体のあらゆる種の推移の傾向が把握できる。以上、述べた様な性質を、プログラム相談における向問題解決の手段に使おうとするのが、当面の目的であった。

マルコフ連鎖を成す有向グラフの様々な性質、吸収点、消滅点、周期性などは、グラフの適用対象によって意味づけられるものである。向問題を明確にするために、我々がここでとりあげているプログラム・デバッキングの質的向回答システムの場合について、以下に概念的な意味づけを行ない、次節の定式化の基礎とする。

例えば、FORTRAN のプログラムを実行してみても予期した結果が得られない場合、FORTRAN のプロセッサから直接エラーの原因を指摘してくれることはまれであって、多くの場合、個々のプログラマーが表面的に現れた現象から、誤りの可能性の範囲を遡りつめていて誤りの源をつまみつけて訂正するという手段をとる。まず、第1段階のエラー状態の分類は、次のように分けられる。

- ① 実行時のエラー・メッセージが出ている (特に使教又は I/O 等)。
- ② memory protection violation に引っかかっている。
- ③ 明らかに CPU の流れが暴走している。
- ④ 計算結果は出ているが、予期しているものと全く異なる。

無論、これらは排他的ではなく、複数個のものが同時に現れることもある。この中で、記憶保護領域の侵犯が起っている時は、まず配列要素への書き込み時の添字式の値を疑うべきであり、それを調べ明瞭な誤りが見つからなければ、次には引数の対応関係を調べてみる必要がある。この様に、エラーの発生の可能性の範囲を狭めてゆく過程は、多くのプログラムを作成している経験者には自然に身に付くものである。しかし、これはあくまでも人間の類推・思考・直観に依存するもので、定式化することは困難である。

我々の試作した質問回答システムでは、このようなプログラム記述上の、あるいは実行過程での様々の現象の一般的な因果関係を、確率的重み付きの有向グラフとして持っている。これがシステム側の知識として働き、個々のプログラマーに助言を与える形式で質問回答過程が進められる。プログラマーの側が自分の抱えている症状を自然言語で計算機に伝えることは、それだけで大きな問題であるので、ここではこの点は回避し、後に示すように実際上の試作システムでは、計算機の側から助言を出して、プログラマーが簡単な回答を返すという手段をとっている。

さて、システムは、プログラマーの直面している現象を知りた上での助言を出すためにグラフの探索を行なう。この探索法には、いろいろのものが考えられるが、代表的な例を3種あげる。

① 現在の現象に関連のある総ての現象をプログラマーに提示し、その

総てを調べさせる。この探索はデバグの初期段階ではあまり意味がないが、ある程度範囲を追い込んだあとでは有効である。

②ゴールに相当する極めて具体的なエラー事象が複数個ある中で、現在の事象から最も到達の可能性の大きいものを探す。途中の関連事象を無視して、そのプログラマーに妥当なゴールが見つければ、この探索は最も直接的である。

③直接にゴールを見つけることが困難であれば、直接に因果関係のある隣接事象を順にたぐって、プログラマーに提示して応答を待つ。当然重みの大きいものから順に提示していくことになるが、この重みは一般的傾向に過ぎず、個々のプログラムの状況においては、ある事象が現われていなければ、次の重みの事象を選ぶことによって順路をかえて、そのプログラマーのおかれた状況の範囲を狭めていく。

本システムでは、様々の状況を考えた上で、オ3番目の方式をとることにした。

このシステムは、個々のプログラムの認識を行なうものではない。一般的傾向を確率モデルとして持っているだけであるから、特定のプログラマーについてはその一般的なグラフ・モデルの中で、現実に現われている事象を手掛りにして部分グラフを順に見つけていって、個々の場合に当てはまるゴールを見つけるのが探索の要点である。したがってゴールの意味も、個々のプログラムについてのエラーを直接指示することは不可能であって、それ以上関連事象をたぐっていても原因を追求することにならないもの、例えば、FORTRAN 文の横記述子の書き誤り、DIMENSION 文の書き誤り、あるいは配列の大きさの引数による受け渡しに正常でないことなどが、ゴールの事象として扱われる。

3. モデルの定式化

事象間の2項関係 $v_i R v_j$ に対して、 v_i から v_j への有向辺をえがけば、我々は有向グラフ G を得る。ここで考えるシステムは、樹状構造を主体とし、部分的に強連結な部分グラフを持つものとする(図3.1参照)。有向グラフ G 上で矢印にそって進む時、これを推測と呼ぶものとする。

$v_i, v_j \in V$ において、任意の v_i に対して $v_i R v_j$ を満す v_j の集合を $\Gamma(v_i)$ ²⁾とし、任意の v_j に対して $v_i R v_j$ を満す v_i の集合を $\Gamma^{-1}(v_j)$ ²⁾とする。
 また、 $\cup \subset V$ に対して $\Gamma(\cup)$ と $\Gamma^{-1}(\cup)$ を次式で定義する。

$$\Gamma(\cup) = \bigcup_{v_i \in \cup} \Gamma(v_i), \quad \Gamma^{-1}(\cup) = \bigcup_{v_i \in \cup} \Gamma^{-1}(v_i).$$

各事象間の推移の強さは推移確率行列 P により与えられ³⁾、グラフ G はマルコフ連鎖グラフとなる。

事象 v_s より、グラフの有向辺にそって到達可能な事象集合 $\hat{V}(v_s)$ を、事象 v_s の到達可能集合 V と呼び、 v_s を V の初期事象と呼ぶものとする。

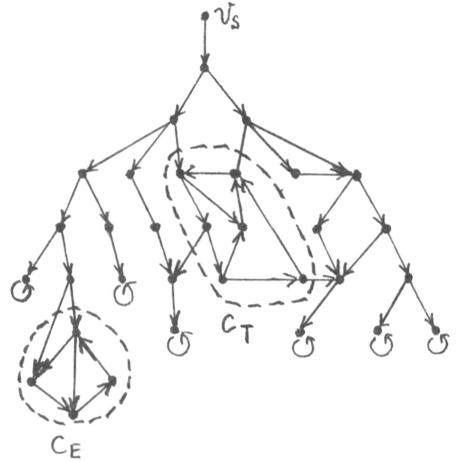


図3.1 本システムで取り扱う有向グラフの代表的な例。

グラフ G の内、強連結部分グラフを成す事象集合をクラスターと名づける。

$\Gamma(v_e) = \{v_e\}$ を満す事象及び $\Gamma(\cup) = C$ を満すクラスターより成る集合は、推移の末端を形成し、端末事象集合 V_E と名づける。また、 $V_T = V - V_E$ とする。 V_T 及び V_E に対応して、行列 P を次のように分割する。

$$P = \begin{bmatrix} P_T & P_E \\ 0 & I \end{bmatrix}, \quad V = V_T + V_E \quad (3.1)$$

但し、 0 はゼロ行列であり、 I は単位行列である。

3.1. マルコフ連鎖グラフの生成

本システムでは、有向グラフの要素となる情報、事象 v_i 、推移 $P(v_i)$ 、推移確率 P_{ij} ($v_j \in \Gamma(v_i)$)が、各事象ごとにファイルの中に記録されている。初期事象 v_s の入力直後に、ファイルより $v_s, P(v_s), P_{ij}$ ($v_j \in \Gamma(v_s)$)が読み出される。同時に、次の集合が定義される。

$$\Lambda_1(v_s) = \{v_s\} \cup \Gamma(v_s), \quad \lambda_1(v_s) = \{v_s\}^c \cap \Gamma(v_s). \quad (3.2)$$

以下、次の漸化式に従って、集合 $\Lambda_n(v_s)$ を逐次計算する。

$$\lambda_n(v_s) = \Gamma(\lambda_{n-1}(v_s)) \cap \Lambda_{n-1}(v_s),$$

$$\Lambda_n(v_s) = \lambda_n(v_s) \cup \Lambda_{n-1}(v_s) \quad (3.3)$$

$\lambda_n(v_s)$ は、 v_s に連結されるべき事象の内、 i 番目にファイルより主記憶上へ取り出される事象集合である。以下、この操作をくり返して、 $\lambda_N(v_s) = \phi$ なる状態に達した時、 v_s の到達可能集合 V が得られる。これら集合は、ビット形式の変数により表現され、各要素をビットに対応させて記憶されている。また、事象 $v_i \in V$ 及び $\Gamma(v_i)$, P_{ij} 等は総て配列で与えられ、それらは、主記憶上に記憶される順序に従って、 i 番目の配列要素に記憶される。初期事象は、 $i=1$ 番目の要素となる。これらの主記憶上での記憶の書式は、例えば次のとおりである。

$$v_i : 0428 \quad (\text{コード番号}),$$

$$\Gamma(v_i) : 001001000 \dots,$$

$$P_{i3} : 0.6 \quad P_{i6} : 0.4$$

添字 i は、 i 番目の配列要素に記憶されていることを示す。 v_i は事象につけられたコード番号である。 v_i に関するメッセージも別に記憶ある。 $\Gamma(v_i)$ の $i=3$ と 6 ビットに 1 があることより、 v_3 と v_6 への推移が可能であることがわかる。 $P_{i3}=0.6$ 及び $P_{i6}=0.4$ より、それらの推移確率が与えられる。

3.2. 初期事象 v_s より $v_i \in V_E$ への到達確率

v_s より i 段階推移直後、事象 v_i が推移を受ける確率は、次のベクトルの要素 $\bar{g}_i(n)$ で与えられる。

$$\bar{g}_i(n) = [1, 0, 0, \dots, 0] \cdot P^n \quad (3.4)$$

無限回推移の全過程において受ける推移確率の総和は、式(3.4)右辺を用いて、次式により計算される行列 B_T, B_E の要素として与えられる³⁾

$$v_i \in V_T \text{ に対して } B_T = [I - P_T]^{-1}, \quad (3.5)$$

$$v_i \in V_E \text{ に対して } B_E = [I - P_T]^{-1} \cdot P_E. \quad (3.6)$$

3.3. クラスタ

事象集合 V において、 $v_i \in V$ から推移にそって到達可能な総ての事象の集合を $\hat{P}(v_i)$ とし、また $v_i \in V$ へ到達可能な事象の集合を $\tilde{P}(v_i)$ とする。この時、クラスタ C は、次式により与えられる。

$$C = \hat{P}(v_i) \cap \tilde{P}(v_i) \quad (3.7)$$

$\hat{P}(v_i)$ は、漸化式(3.2), (3.3)と同じ論法で得られる。 $\tilde{P}(v_i)$ は、漸

化式(3.2), (3.3)の P も P' に変えた式により, 同様に計算される。

$\Gamma(C) = C$ の時, V において C を1つの事象 v_i で置きかえ, C に向う有向辺をすべて v_i に向わせ, 改めて v_i より v_i へ向う有向辺を付けて, 推移確率 $p_{cc} = 1$ とする。これにより, 式(3.6)の計算が容易になる。

クラスター中では, 推移が巡回するため, 式(3.5), (3.6)で得られた確率は重複して数えられたものである。確率の定義上, クラスター以外の事象とのつり合いをとるため, 以下の補正を行なう。 $\Gamma(C_{(e)}) = C_{(e)}$ なるクラスター $C_{(e)}$ の推移確率行列を $P_{C_{(e)}}$ とする。 $C_{(e)}$ は V_E の部分集合であり, 推移の過程において他の事象へ影響せず孤立系を成す。

そして, $C_{(e)}$ の各事象に対する定常分布は, マルコフ過程に対する理論的帰結³⁾より, 次式を満たすベクトルの各要素の値で与えられる。

$$\vec{b}_{C_{(e)}} = \vec{b}_{C_{(e)}} P_{C_{(e)}} \quad (3.8)$$

$P_{C_{(e)}}$ の最大固有値は1であり⁴⁾, $\vec{b}_{C_{(e)}}$ は $P_{C_{(e)}}$ の最大固有値に対する固有ベクトルである。 $\vec{b}_{C_{(e)}}$ は, クラスター $C_{(e)}$ 内の事象の相対的重要度の判定に用いる。 $\Gamma(C) \neq C$ なるクラスターは, 他事象集合との間に強い結びつきをもつ。したがって, 式(3.8)では取り扱えない。ここでは, 式(3.5)の帰結において, クラスター C 内で巡回したために重複して数えられた確率を取去ることにより補正する。

$$b_{si} = \beta \cdot b_{si}, \quad v_i \in V$$

$$\beta = 1 - \left(\sum_{v_i, v_j \in C} b_{si} p_{ij} \right) / \left(\sum_{v_j \in C} b_{sj} \right)$$

b_{si} は, 式(3.5)により計算される行列 B の s 行 i 列要素である。

β は, 上記の意味での補正係数である。

4. ファイル-主記憶間のデータの転送

プログラム相談に必要な全事象及び事象間の推移は, ファイルの中に記憶されている。初期事象 v_s が入力された後, システムはファイルの中より推移に従って到達可能集合を主記憶上に取り出す。到達可能集合よりなるグラフは, 一般に充分大きいと考えられる。したがって, グラフ全体を一時に主記憶上に移すことは, 主記憶の容量の限界から, 不可能な場合が起こり得る。この点に対する処置として, 以下のように

な手法を取る。

初期事象 v_0 からの到達可能集合より成るグラフを、3つの部分に分けて考える。

I. オ1領域

初期事象に最も近い部分で、質向回答の手順からいえば、すでに確率1であることが確定した事象より成る部分である。

II. オ2領域

質向回答過程において、未だ探索の可能性を残し、選択の余地のある事象より成る部分である。

III. オ3領域

事象個々の存在の可能性は予測されているが、未だ具体的検索の対象にな

っていない事象より成る部分である。プログラム相談システムにおいては、 v_0 より到達可能な事象の内、未だファイルの中に残されている部分である。質向回答の進行に共ってオ2領域が縮小した時点で、やがてオ2領域へ移される。質向回答過程において、オ3領域の有向グラフが発芽すると考えることもできるであろう。

このように、グラフを3領域に分割することにより、システムの大きさに適した方法で、質向回答過程を進めることができる。このような考えは、一般的質向回答システムの実現に対して計算機の大きさに対する依存性が少くなるので、重要であろう。ただし、グラフを分割しても、オ3節の定式を変える必要はない。

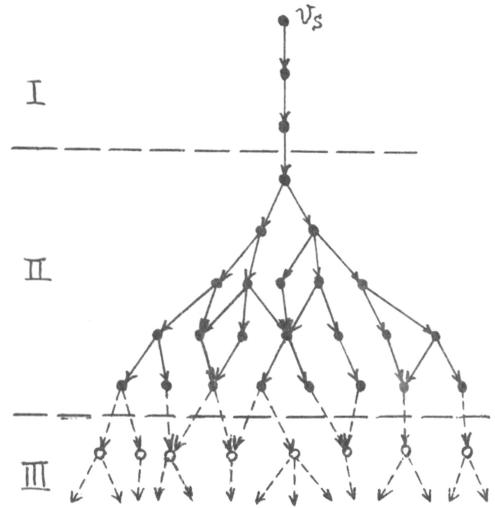


図4.1. 到達可能集合の3領域への分割の例

5. 質向回答過程

有向グラフの書き換えは、システム対プログラマーの質向回答を通し

て行なわれる。これに関しては、実際のプログラム相談システムでの
 質向回答過程の具体例を通して説明する。システムはTime Sharing
 systemにより実現されている。図5.1の有向グラフをもとにして行
 なった質向回答の1例を、図5.2に示す。図5.1の有向グラフのフ
 イルより主記憶上への転送は、オ4節のとおりであるが、質向回答の過
 程では表面に現れない。式(3.2), (3.3)による有向グラフの主記憶
 上での生成は、人間における類推・連想に対比して考えることができる
 。以下、図5.2を説明する。

図5.2において、各行の左端の数字は説明
 の便宜につけた番号であり、その左
 側の*印は、プログラマーにより入
 力されたものであることを示す。

各行ごとの意味は次の通りである。
 ①, ②: 質向回答の開始である。
 コード番号とは、各事象に
 付けられた番号であり、事
 象の内容を示す説明文と共
 に、質向コード番号表とし
 て別に準備されている。

- ③: コード番号(0001)が入力された。
- ④~⑦: システムが結論を出力するまでの時間、システム内の実行状
 況を、参考までに出力したものである。
- ⑧~⑭: 式(3.5), (3.6)で得られる、(0001)より各事象への到達確率の大き
 な事象を、順次先頭よりつないで出力した結果である。カ
 ソコ内2桁の整数は、主記憶上で各事象に便宜的につけられた
 番号である。2番目にコード番号が書かれ、右端にメッセージ
 が書かれている。左にカソコ付き2桁整数の無い事象は、
 到達確率が1で、検索の余地の無い事象である。
- ⑮, ⑯: “判断”に対する入力書式がわからなければ、*を入力する。
 この時、入力書式についての説明が出力される。わかっている
 場合は、例えば次のように入力する。

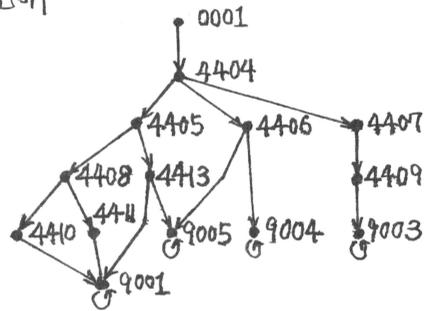


図5.1 テストに使われた有
 向グラフ。图中的数字は
 、事象につけられたコード
 番号である。

Y02; (02)番を Yes (到達確率1) とすることをシステムに示す。
N05; (05)番を No (到達確率1) とすることをシステムに示す。

⑰: この段階で、システムは(02)4405の事象が確率1であることを認識し、この事象の含まれない順路をグラフ上から除去する。同様にして、N05より、(05)4413をグラフ上から除去する。

⑱: 説明省略。

⑲~⑳: ㉑~㉒に同様。

㉑, ㉒: N02; (02)番が No (到達確率0) となることをシステムに示す。

㉓: この段階で、グラフ上より(02)4411を除去する。

㉔~㉗: 以上の質向回答の過程で、順路はただ一つに確定した。したがって、カッコ付き番号は出力されず、この段階での出力がシステムよりの最終回答となる。

(コード番号に対する各メッセージは、現在整備中であり、ここでは単に出力書式の説明上の参考にとどめておいて下さい。)

図5.2は、システム対プログラマーの質向回答の一具体例にすぎないが、質向回答過程における有向グラフの書き替え手続き、及び最終結論に到達するまでの手順については、充分に示せたものと思う。

6. 今後の課題

本システムは、様々の水準のプログラマーにより利用されるであろう。したがって、データ・ファイルのみで、プログラマーの知的水準に合った柔軟なシステムの応答が望まれる。これについては、残された課題として、現在いくつかの解決方法を検討中である。

参考文献

- 1) G.W.Ernst and A.Newell: GPS: A Case Study in Generality and Problem Solving, Academic Press (New York, London), (1969).
- 2) 尾崎弘, 白川功: グラフとネットワークの理論, コロナ社, (1973)
- 3) 北川敏男: 情報科学講座A5.1, マルコフ過程, 共立出版, (1961)
- 4) R.S.Varga: Matrix Iteration Analysis, (Prentice-Hall, Inc. Maruzen. Co. LTD), (1964)

図5.2 プログラム相談システムの質問回答の実行例。

図中で、デバッキング手法に関する説明(⑩~⑭, ⑲~㉔, ⑳~㉗)は、

(い) ひ マシセージ”

の順序で出かされてゐる。(説明詳細は本文にあり)

- ① ※※ CODE バックウオカ 4-クタノセクタヲ ニユウヨク シテクダサイ。 ※※
- ② (CODE バックウオノカクタイ バックハ 1000 オ ニユウヨク シテクダサイ。)
- *③ ※0001
- ④ ## ニユウヨク シテクダサイ CODE バックウオカ ヲカマシマシ。 ##
- ⑤ ※クタノシマシニカケイノカクオオクタカトシマシ。
- ⑥ ※※ コクタシマシヨカシマシコトオシマシニトシマシマシ ※※
- ⑦ クタノシマシニカケイノカクオオクタカトシマシ。
- ⑧ ヲカニシマシヲシマシハクタノシマシニシマシカケオニシマシイマシ。
- ⑨ (カクオカシマシニシマシヲシマシシマシシマシシマシ)
- ⑩ 1 TYPE OF FILE NN IS NEITHER SEQUENTIAL NOR DIRECT.
- ⑪ 4404 READ シマシニカケイノカクオオクタカトシマシイマシ。
- ⑫ (2)4405 READ シマシニカケイノカクオオクタカトシマシイマシ。
- ⑬ (5)4413 READ シマシニカケイノカクオオクタカトシマシイマシ。
- ⑭ (11)9001 ヲカノシマシニシマシ。
- ⑮ ※※ クタノシマシニシマシシマシシマシ。 ※
- ⑯ ※Y02, N05

⑰ ----- クタカオシマシシマシ -----

18) アタノシツペンニカガイノアルゴトニツイソシラノカガイオシラハマス。

19) ツキニシマスブツヨウハアタノシツペンニサイズカイトウニナツイマスカ。
 (ウエカラシムニテキトウセツクシオツツエソテクダサイ)
 20) 1 TYPE OF FILE NN IS NEITHER SEQUENTIAL NOR DIRECT.
 21) READ ブンニカシテFILEノツカイタガマチガツイマス。
 22) 4404 READ ブンニカシテヒツヨウナFILE TABLEオPROGRAMニケツコウシテクダサイ。
 23) 4405 READ ブンニカシテヒツヨウナFILE TABLEオPROGRAMニケツコウシテクダサイ。
 24) 4408 READ ブンニカシテWORK CARD NO TSUKAIKATA
 25) (2)4411 READ ブンニカシテWORK CARD NO PARAMETER
 26) 9001 リヨウノテビキオミヨ。
 27) ※※ アタノハンダシオニユウリヨクシテクダサイ。 ※※
 28) ※N02

29) ----- ケンサクオツツクマス -----

30) ツキニシマスブツヨウハアタノシツペンニサイズカイトウニナツイマスカ。
 (ウエカラシムニテキトウセツクシオツツエソテクダサイ)
 31) 1 TYPE OF FILE NN IS NEITHER SEQUENTIAL NOR DIRECT.
 32) READ ブンニカシテFILEノツカイタガマチガツイマス。
 33) 4404 READ ブンニカシテヒツヨウナFILE TABLEオPROGRAMニケツコウシテクダサイ。
 34) 4405 READ ブンニカシテヒツヨウナFILE TABLEオPROGRAMニケツコウシテクダサイ。
 35) 4408 READ ブンニカシテWORK CARD NO TSUKAIKATA
 36) 4410 READ ブンニカシテWORK CARD O SOUNYU SURU KOTO
 37) 9001 リヨウノテビキオミヨ。

38) ----- オツカルサマデシタ。 -----
 39) サイゴニシマシタ。 カイトウブツヨオモツアタノシツペンノカイトウトイタマス。
 40) -----
 41) -----

本 PDF ファイルは 1976 年発行の「第 17 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場（＝情報処理学会電子図書館）で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>