CODASYL DML に対する非手続的グラフ問合せ言語の設計と実現†

滝沢 誠†† 野口正一†††

本論文では，CODASYL モデルを提供する CODASYL DBS を概念，論理，仮想，物理の4階層に分け，論理層と仮想層の論理構造と，層間写像とについて考察する．論理層の論理データ構造は，集合と二つの集合間の部分関数とで表現され，論理操作言語は述語論理形式の CODASYL 問合せ言語 CQL とこれと等価な CODASYL 問合せグラフ (CQG) とで定義される．CQG は論理操作演算を見やすい形式で表せるので利用者言語として優れている．仮想層では，論理データ構造に新たに順序関係が導入されて，仮想データ構造が定まる．この上の仮想操作演算 VOP が従来の COBOL DML に対応したものとして与えられる．本論文では，この意味を仮想的な CODASYL 機械 (VCM) として，多テープチューリング機械を用いて表し，DML の意味を明らかにする．概念層の任意の CQG を VCM 上の演算，すなわち COBOL DML プログラムで表せることを示す．さらに，各 CQG に対して，中間ファイルを不要とし，ファイル操作として順次書込みだけで済む DML が存在することを明らかにし，その構成方法を示す．本論文は，従来の CODASYL DBS 上に非手続的な利用者インタフェースを設計するための論理的基礎を与えている．

1. はじめに

CODASYL モデル[1),2)]を提供するデータベースシステム (DBS) を CODASYL DBS とする．CODASYL DBS は，COBOL DML[2)] 等の手続的操作言語を用いて，高性能性が求められる大型の定型業務に広く用いられてきていた．しかし，最近，一般利用者がデータベースを非定型的に利用するための非手続的インタフェースが求められてきている．このような非手続的インタフェースを設計するためには，まず従来の CODASYL モデルの論理的な構造を明確にする必要がある．しかし，現在まで，CODASYL モデルに対する十分な論理的考察はほとんどなされていない．わずかに，DML の意味について，文献 13) が状態遷移により，14) がデータ抽象化の公理的方法により意味記述を試みている程度である．また，述語論理形式の非手続的操作演算を手続的 DML に変換する問題は，限定された形式の検索演算について文献 5)～7)，12) が，更新演算については文献 8) が論じている．

本論文では，①CODASYL モデルの論理的階層構造を明確に与え，②このもとで，DML の意味を明らかにし，③自由な形式の非手続的検索演算を満足する DML プログラムの構成方法を与える．②として，従来の COBOL DML[2)] の意味を多テープチューリング機械の動作として与えることで，DML の明確な意味を与える．③として，まず，従来の CODASYL モデルを抽象化して得られる集合と部分関数とで与えられるデータ構造に，述語論理形式の検索言語 CODASYL 問合せ言語 CQL と，そのグラフ表現 CQG (CODASYL 問合せグラフ) とを与える．次に，CQG の条件を満足する DML プログラムが存在することと，目標集合を格納するための出力ファイル数を最小化する DML プログラムを構成できることを示す．本論文の方法は，文献 5)～7)，12) に対して，①完備した検索演算を備えた CQG を利用者に提供でき，②CQG の条件を満足する目標集合を，一つの物理的な出力ファイルだけを用いた DML プログラムで求められる特徴をもつ．

以上を検討するうえで，従来の CODASYL モデルとして，文献 15) 等も考えられるが，その論理的に本質的問題を検討するには，文献 1) と 2) で十分と考える．

2章では，従来の CODASYL モデルの論理階層を明確にする．3章と4章ではおのおの，論理階層のなかの CQL を提供する概念層と DML を提供する仮想層とを定式化する．5章では，CQL と DML との関係を論じ，6章では，LC モデル[8)] と DML との対応を示す．

2. 従来の CODASYL モデルの論理階層

従来の CODASYL モデル[1),2)] は，物理的側面と論理的側面の議論が不十分であり，専門家向けの COB-

† Design and Implementation of Non-procedural Graph Query Language for CODASYL DML's by MAKOTO TAKIZAWA (Japan Information Processing Development Center) and SHOICHI NOGUCHI (Research Institute of Electrical Communication, Tohoku University).

†† (財)日本情報処理開発協会
††† 東北大学電気通信研究所

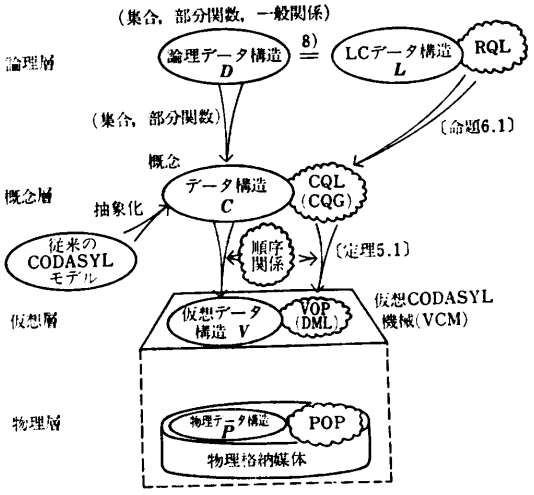


図 1 CODASYL DBS の階層構造
Fig. 1 Hierarchy of the CODASYL DBS.

OL DML[2]等の手続的操作言語を提供している．しかし，一般利用者には，論理的なデータ構造を非手続的に操作できることが望ましい．このためには，物理と論理的側面を分離し，DML の操作層と非手続的な言語の操作層との関係を明らかにする必要がある．

本論文では，CODASYL モデルは，論理，概念，仮想，物理の4層の階層構造でモデル化できることを示す（**図1**）．まず，従来の CODASYL データ構造[1]の論理構造を抽象化すると，集合（レコード(R)型）と集合間の部分関数（セット(S)型）とで与えられるデータ構造が導ける．このデータ構造上に，述語論理形式の非手続的操作言語を定めることができる．このデータ構造と言語で定まるモデルを，概念CODASYLモデルとする．このモデルは，論理的データ構造と非手続的操作演算を利用者に提供するもので，3章で定式化する．さらに，概念 CODASYL データ構造を一般化すると集合間の一般関係（リンク(L)型）が定まる*．集合，部分関数に加えてこの一般関係が定められたデータ構造を，概念 CODASYL データ構造上の論理 CODASYL データ構造とする．このデータ構造は，文献8) で定義された関係型のローカル概念 (LC) データ構造[8]との間に一対一の対応関係が与えられ，詳細は文献8) で示されている．

次に，概念 CODASYL データ構造の集合と関係の元に，ある規準によって定まる順序関係を与える．これにより，このデータ構造の操作として，与えられた順序関係に基づいて各演算が逐次行われることになる．この演算が従来の COBOL DML[2]に対応している．このデータ構造と操作演算で定まるモデルを，概念 CODASYL モデルの一つ論理的下位の仮想 CODASYL モデルとし，4章で定式化する．

次に，仮想 CODASYL モデルを物理格納装置上に実現するモデルとして，物理 CODASYL モデルが定義される．

以上の論理階層を設けることによって，従来の CODASYL モデルの論理構造が明確となり，これをもとにしてCODASYL DBS 上の非手続的利用者インタフェース設計の論理的基礎とすることができる．

3. 概念 CODASYL モデル

本章では，概念 CODASYL モデルのデータ構造と操作演算とを定義する．

3.1 概念データ構造

概念データ構造 $\boldsymbol{C}$ は，時間不変な論理構造を与える概念スキーマ C と，これに実際のデータを与えた（時間可変な）概念データベース C とから成る．C はレコード(R)型とセット(S)型とから成る．R型 R は項目集合 $\Omega_R=\{@R, t_1, \cdots, t_m\}$ $(m\geq 0)$ とインテグリティ条件 Γ_R とから成る $(R=(\Omega_R,\Gamma_R))$．各項目 $\tau\in\Omega_R$ のとりうる値の集合(定義域)を $\mathrm{dom}(\tau)$ とする．$@R$ は db キーで，$\mathrm{dom}(@R)$ は R の識別子集合である．各 $t_i\in\Omega_R$ はデータ項目である．Γ_R は，Ω_R がとりうる値の組（レコード(R)実現値）を定める述語で，この組の集合をレコード実現値 (RO) 集合 $\mathrm{R}=\{o \mid o\in \mathrm{dom}(\Omega_R)\wedge\Gamma_R(\tau)\}$（ここで，$\mathrm{dom}(\Omega_R)=\mathrm{dom}(@R)\overset{m}{\underset{i=1}{\times}}\mathrm{dom}(t_i)$）とする．各 $o\in$R の意味のある部分項目集合 I $(\subseteq\Omega_R)$ の値を $I(o)$ とする．各 $o\in$R の db キー $@R$ の値は，R のみならず他の RO 集合内でも一意であるので，$@R$ は R の主キーと考えられ，$@R(o)$ を $\delta(o)$ と表す．また利用者には db キー情報が知らされないので，R 型を $R(t_1, \cdots, t_m)$（Γ_R は省略）とも表す．各 R について，次のR述語 P_R: $\mathrm{dom}(\Omega_R)\rightarrow$ {true, false} を定める．

$$\mathcal{P}_R(o)=\text{true } \underline{\text{if}}\ o\in\mathrm{R},\ \text{false } \underline{\text{otherwise}} \quad (1)$$

$\boldsymbol{C}$ では，二つのR型間の関係として，セット(S)型が定義される．すなわち，部分関数 f_S: $\mathrm{dom}(\Omega_{R_2})\rightarrow \mathrm{dom}(\Omega_{R_1})$ が定義されているとき，S型 $S=(R_1, R_2, \Gamma_S)$ と表す．R_1 と R_2 は，おのおの f_S の値域と定義域を与えるR型であり，親と子R型とする．Γ_S は

* 概念 CODASYL データ構造のなかで，複数の R 型 $R_1, \cdots, R_n$ と，共通の子 R をもつS型 $S_1, \cdots, S_n$ が存在するとき，新たに R と $S_1, \cdots, S_n$ をまとめて，$R_1, \cdots, R_n$ 間の一般関係を与えるリンク(L)型を定義する．

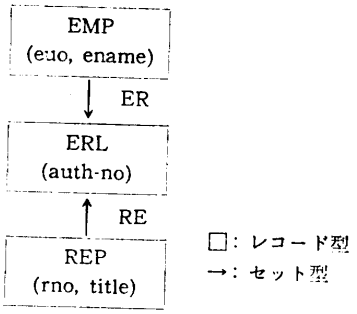


図 2 論理スキーマ例
Fig. 2 Logical CODASYL schema.

Ω_{R_1} と Ω_{R_2} の関係より定まる述語で，これにより S に実際に与えられるR実現値対（セット(S)実現値）が定まる．そしてこの集合をセット実現値 (SO) 集合 $S=\{u \mid u\in \overset{2}{\underset{i=1}{\times}} \mathrm{dom}(\Omega_{R_1}) \wedge \Gamma_S(u)\}$ とする．各 $u=(o_1, o_2)\in S$ で，$o_1\in R_1, o_2\in R_2$ でなければならず，おのおのを親と子R実現値，また o_1 と o_2 は親子関係にあるという．S は f_S により与えられるので，各 $o_1\in R_1$ は $h(\geq 0)$ 個の子 $o_2\in R_2$ を，また各 $o_2\in R_2$ はたかだか一つの親 $o_1\in R_1$ をもてる．Γ_S の一つとして，あるデータ項目 $t\in\Omega_{R_i}$ ついて，各 $o_1\in R_1$ のすべての子 $o_2\in R_2$ が互いに異なった t 値をもたねばならない条件がある．この条件を満たす t を S の DNA (duplicates are not allowed) 項目という．$@R_i$ は R_i の主キーであり，識別子であるので，各 $(o_1, o_2)\in S$ をdbキーの対 $(\delta(o_1), \delta(o_2))$ によって表すこともある．本論文では，以下 $S=\{u \mid u=(\delta(o_1), \delta(o_2))\wedge u'=(o_1, o_2)\in \overset{2}{\underset{i=1}{\times}} \mathrm{dom}(\Omega_{R_i})\wedge\Gamma_S(u')\}$ とする．S は次のS述語 $\sigma_S: \overset{2}{\underset{i=1}{\times}} \mathrm{dom}(\Omega_{R_i})\rightarrow\{\mathrm{true}, \mathrm{false}\}$ と同じ意味をもつ．

$$\sigma_S(o_1, o_2)=\mathrm{true}\ \underline{\mathrm{if}}\ (\delta(o_1), \delta(o_2))\in S,\quad \mathrm{false}\ \underline{\mathrm{otherwise}} \qquad (2)$$

以上の RO 集合と SO 集合の集合を C の概念データベースCとする．**図2**は職員と論文の情報を表す C の例である．箱はR型，矢印は親から子R型へのS型を表す．

3.2 概念操作演算

概念操作演算としては，検索と更新演算がある．更新演算の基本的問題は，文献8)で論じているので，本論文では検索演算について述べる．

3.2.1 CODASYL 問合せ言語 (CQL)

概念データベースC上の検索演算は，①目標スキーム設定と，②検索条件設定との二つの手順によって表される．検索条件 Ψ は，次の3種の基本式，(a) $r.\alpha\theta v$，(b) $r.\alpha\theta r'.\beta$，(c) $\sigma_S(r, r')$ から成る論理式で表される．ここで，r はある RO 集合 R 内のR実現値をとる O 変数，$r.\alpha$ は r のとるR実現値の項目 α の値とする．θ は比較演算子，v は定数である．r' は RO 集合 R′ の O 変数で，β は R′ の項目である．(c)は(2)式で与えられるS述語である．このとき，Ψ は次のように正規化される．

$$\Psi=\sigma\ \wedge\ \rho\ \wedge\ \pi \qquad (3)$$

ここで，σ はS述語 σ_S の積，ρ は各 O 変数 r ごとの条件式 ρ_r の積で，ρ_r は $r.\alpha\theta v$ の積・和の形で表現される．π は O 変数 r と r' 間の条件式 $\pi_{rr'}$ の積で，$\pi_{rr'}$ は $r.\alpha\theta r'.\beta$ で表現される．Ψ 内の O 変数を $r_1, \cdots, r_l (l\geq 1)$ とすると，$\Psi: \overset{l}{\underset{i=1}{\times}} \mathrm{dom}(\Omega_{R_i})\rightarrow\{\mathrm{true}, \mathrm{false}\}$ である（r_i は RO 集合 R_i の O 変数とする）．

目標スキームを $\tilde{\Lambda}_C=\{a_1, \cdots, a_k\} (k\geq 1)$ とし，a_i を目標項目という．いま $R_1, \cdots, R_l$ を C 内のR型とすると，$\tilde{\Lambda}_C$ は $\tilde{\lambda}(\Omega_{R_1}, \cdots, \Omega_{R_l})=\tilde{\Lambda}_C$ なる関係 $\tilde{\lambda}$ で定義される*．$\tilde{\lambda}$ から誘導される関数 $\lambda: \overset{l}{\underset{i=1}{\times}} \mathrm{dom}(\Omega_{R_i})\rightarrow \mathrm{dom}(\tilde{\Lambda}_C)$ を目標関数とする．

検索演算は，(λ, Ψ) と表せて，次の意味をもつ．

$$G=\{t \mid t=\lambda(o)\wedge\Psi(o)\wedge o=(o_1, \cdots, o_l)\ \overset{l}{\underset{i=1}{\wedge}}\ \mathcal{P}_{R_i}(o_i)\} \qquad (4)$$

以上をもとに，検索演算は CODASYL 問合せ言語 CQL で次のように表せる．

$$\underline{\mathrm{var}}\ (r_1, R_1)\cdots(r_l, R_l); \qquad (5)$$

$$\underline{\mathrm{get}}\ \underline{\mathrm{into}}\ G\ (\Lambda)\ \underline{\mathrm{where}}\ \Psi; \qquad (6)$$

(5)式は，(1)式の $\mathcal{P}_{R_i}(r_i)$ を表し，RO 集合 R_i に O 変数 r_i を定める．Λ は目標関数 λ によって定まるもので，$\{\beta_1, \cdots \beta_k\}$ で与えられる．β_i は，目標項目 a_i の値を与える関数 $\varepsilon_i: \overset{l}{\underset{j=1}{\times}} \mathrm{dom}(\Omega_{R_j})\rightarrow \mathrm{dom}(a_i)$ である．β_i は，$a_i=\varepsilon_i$ と与えられる．

各 ε_i 内の項目が出力項目である．Ψ は(3)式で与えられる条件式である．また $\Lambda=(\beta_1, \cdots, \beta_k)$ を目標リストという．

[定義 3.1] 検索演算の CQL 表現を CODASYL 問合せと定義する． □

3.2.2 CODASYL 問合せグラフ (CQG)

CODASYL 問合せのグラフ表現としての CODASYL 問合せグラフ (CQG)Q は次のようにして与えられる．

[CQG の構成方法]

* $\tilde{\lambda}$ は目標項目として，項目集合 Ω_{R_i} からどのような項目を選ぶかを与える関係である．

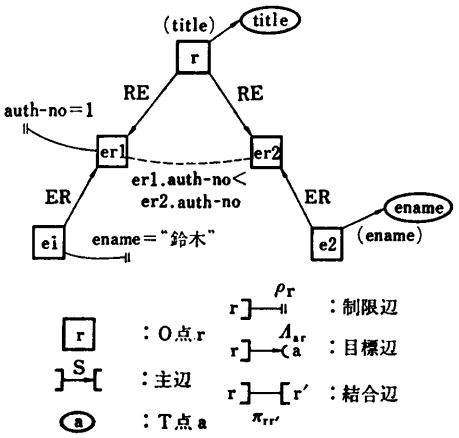


図 3 CODASYL 問合せグラフ (CQG)
Fig. 3 CODASYL Query Graph (CQG).

①CODASYL 問合せの各 O 変数 r に対して O 点 r を設ける（以下 O を省略し点 r とする）．②Ψ の各 $\sigma_S(r, r')$ に対して，この値が真ならば点 r から r' に有向な主辺 S を設ける．③各 $\pi_{rr'}$ に対して，点 r と r' 間に結合辺 $\pi_{rr'}$ を設ける．④各 ρ_r に対して，点 r に接する目標辺 ρ_r を設ける．⑤Λ 内の各 $a=\varepsilon$ に対して，T点 a を設け，⑥ε 内のT点に対応する点 r からT点 a に有向な目標辺 Λ_{ra} (ε 内の r の出力項目集合) を設ける． □

[定義 3.2] CQG Q で，目標辺に接続する点を出力点とする．Qが主辺について連結なときのみQを連結とする．連結なQ内の任意の2点 r と r' 間の路とは，接続する主辺とその上の点を逐次たどることによって得られる r と r' 間の点と主辺の系列である． □

[命題 3.1] 任意の CODASYL 問合せと CODASYL 問合せグラフ (CQG) とは一対一対応する．

[証明] CQG の構成方法より明らかである． ■

命題 3.1 よりすべての問合せは CQG で表現され，この表現法は利用者インタフェースとして非常に見やすい形になっている．

[例 3.1] 図2に対する検索演算「鈴木を第一著者とする共著論文名とその共著者を求めよ」は，CQL で (7) のように書ける．

var (e1, EMP) (e2, EMP) (er1, ERL) (er2, ERL) (r, REP); get into G (r. title, e2. ename) where e1. ename= "鈴木" and ER (e1, er1) and RE (r, er1) and RE (r, er2) and ER (e2, er2) and er1. auth-no =1 and er2. auth-no>er1. auth-no; (7)

(7) に対応した CQG を**図 3** に示す．箱は点，点間の矢と点線はおのおの，主辺と結合辺を，また点に接する接地記号は制限辺を表す．楕円はT点を，点からT点への有向辺は目標辺を表す． □

4. CODASYL DBS の仮想層

本章では，概念層と物理層の中間に位置づけられ，従来の DML[2] の操作層に対応した仮想層の定義を行う．

4.1 仮想データ構造

概念データ構造 $\boldsymbol{C}=(C, \mathrm{C})$ の仮想層での表現を仮想データ構造 $\boldsymbol{V}=(V, \mathrm{V})$ とする．V とVをおのおの，仮想スキーマと仮想データベースとする．C 内の各R型 $R=(\Omega_R, \Gamma_R)$ に対して，新たに仮想 R(VR) 型 $X=(\Omega_X, \Gamma_X, \prec_X)$ が定義できる．ここで $\Omega_X=\Omega_R, \Gamma_X=\Gamma_R$ である．$\prec_X$ は db キーによる昇順の順序関係である．以上のことから，X に実際にデータを与えた仮想 RO (VRO) 集合Xは，RO集合Rを $\prec_X$ で全順序づけたものである．各R実現値 $o\in$R に対応した $v\in$X を，o の仮想 R(VR) 実現値とする．

各S型 $S=(R_1, R_2, \Gamma_S)$ に対して，仮想 S(VS) 型 $Y=(X_1, X_2, \Gamma_Y, \prec_Y)$ が定義される．X_i は R_i の VR 型で $(i=1,2)$, $\Gamma_Y=\Gamma_S$, $\prec_Y$ は順序関係である．X_1 と X_2 をおのおの，Y の親と子 VR 型とする．各R実現値 $o_1\in$R$_1$ の VR 実現値 $v_1\in X_1$ に対して，$m_Y(v_1)$ をSの子 $o_2\in$R$_2$ の VR 実現値 $v_2\in X_2$ の集合とする．Y に実際にデータを与えた仮想 SO (VSO) 集合を，$Y=\{(v_1, m_Y(v_1)) \mid v_1\in X_1 \wedge m_Y(v_1)\neq\phi\}$ とする．各 $(v_1, m_Y(v_1))\in Y$ を仮想 S(VS) 実現値とする．$m_Y(v_1)$ は $\prec_Y$ で全順序づけられている．$\prec_Y$ としては，①あるデータ項目 $t\in\Omega_{X_2}$ についての昇順と②降順，③任意の順との3種がある．①と②の t をYの整列項目とする．

VRO 集合と VSO 集合の集合を仮想データベースVとする．

[性質 4.1] 仮想データ構造 $\boldsymbol{V}$ から，順序関係を除くと概念データ構造 $\boldsymbol{C}$ になる． □

4.2 仮想 CODASYL 機械による仮想操作演算の表現

従来の COBOL DML[2] の論理的意味を明らかにするために，多テープチューリング機械として，仮想 CODASYL 機械 (VCM) を考える．VCM は有限制御 $\mathcal{F}$ と任意の有限本のテープ集合 $\mathcal{T}$ とから成る (**図 4**)．各テープTは任意の有限個の単位セルから成る．各単位セルは，識別子とデータをおのおの記憶する ID セルと D セルから成る．$\mathcal{F}$ はTごとにヘッド H_T をもち，時刻ごとに一つの単位セルを見ることができる．$\mathcal{T}$ は仮想データベースVを記憶する DB テープ，仮想

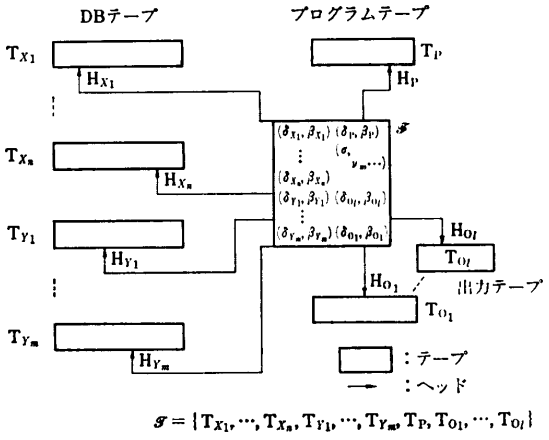


図 4 VCM の構成
Fig. 4 Virtual CODASYL Machine (VCM).

操作演算 (VOP) を記憶する P テープ，出力用の O テープの 3 種から成る．VR 型 $X_i(i=1,\cdots,n)$ と VS 型 $Y_j=(X_{j1}, X_{j2})(j=1,\cdots,m)$ から成る仮想 CODASYL スキーマ V に対して，DB テープは各 X_i と Y_j を記憶する n 本の VR テープ T_{X_i} と m 本の VS テープ T_{Y_j} とから成る．DB テープの各単位セルには，与えられた順序関係に従って VR 実現値が連続して書かれ，ID セルには db キー，D セルにはデータ項目値が書かれる．それぞれ VRO 集合 $X_i=\{v_1,\cdots,v_p\}$ と VSO 集合 $Y_j=\{(a_j, b_{j1},\cdots, b_{jl_j})\mid a_j\in X_{j1}\wedge b_{jh}\in m_{Y_j}(a_j)\}$ のテープ表示は**図 5** となる．左右端にはおのおの ⊢ と ⊣ が，他の単位セルには も が書かれている．図 5 (b) で ; は VS 実現値の区切りを表す．P テープ T_P は読取り専用で，各単位セルの D セルには左から VOP (**表 1**) が，ID セルにはラベルが書かれ，VCM は逐次これらを実行する．各 O テープ T_{O_l} の各単位セルには，検索の結果求まった目標集合 G_i の組が連続して書かれる．

有限制御 $\mathcal{F}$ は，次状態決定関数 ν，制御状態集合 F，内部記憶状態集合 M で定まる．M は VCM 全体のレジスタ σ および $v_1,\cdots$ と，各テープ T に対して与えられているレジスタ δ_T, β_T によって表される．δ_T には H_T の現在いる単位セルの ID の情報が記憶され，β_T には，この単位セルの D の情報または D に書き込むべき情報が記憶される．σ には，VOP の目的が達成されないとき (たとえば，VR テープ X の H_X が next (X) で情報 ⊣ をもつ単位セル上にきたとき) F (失敗)，達成されたとき S (成功) が記憶される．v_i は VOP の作業用レジスタである．以下，各レジスタ α 内に記憶されている情報を (α) と表す．

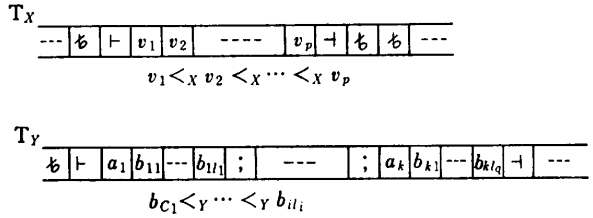


図 5 テープ表示
Fig. 5 Tape description.

VCM は，T_P から VOP (表 1) を一つ読み，DB と O テープを動作させ，ν によって次の制御状態と内部記憶状態とを定める．たとえば，next (X) は VR テープ T_X の H_X を一単位セル右に動かす．get (X) は H_X のいる単位セルの D の情報を β_X に記憶する．

VCM は，$\mathcal{F}$ の制御状態 $f\in F$，各テープ T の H_T が現在いる単位セルの情報，内部記憶状態 $\mu\in M$ によって，次の制御状態，内部記憶状態を ν で定め，H_T を動かし単位セルの情報を変える．H_T の動作としては以下がある．①N：H_T を動かさない．②R(L)：H_T を一単位セル右 (左) に動かす．③H(T)：H_T を ⊢ (⊣) に動かす．④RP：H_T のいる単位セルの D を (β_T) で置換．⑤AP：H_T のいる単位セルとその右側の全単位セルを一つ右に移し，H_T のいる単位セルの D に (β_T) を書く．⑥DL：H_T のいる単位セルの右側の全単位セルを一つ左に移す (H_T 下の単位セルの消去)．⑦DA：(δ_T) を ID とする単位セルの位置に H_T を移動．⑧RD：H_T のいる単位セルの D の情報を β_T に書く．⑨AC：あるデータ項目 t について，(β_T) の t の値を D 内にもつ単位セルの位置に H_T を移動．ここで，$m=\{N, R, L, H, T, DA, RP, AP, DL, RD, AC\}$ とおくと，VCM は次のように定式化できる．

[定義 4.1] VCM は六字組 $(F, M, \boldsymbol{T}, \nu, f_0, f_F)$ である．$f_0\in F$ は初期制御状態，$f_F\in F$ は最終制御状態である．$\boldsymbol{T}$ は各テープ T の情報集合の直積である．次状態決定関数 $\nu: F\times M\times \boldsymbol{T}\to F\times M\times \boldsymbol{T}\times m^{n+m+l+1}$ である．VCM の各時刻の様相は $(f\in F, \boldsymbol{t}\in \boldsymbol{T}, \mu\in M, p)$ で，$\boldsymbol{t}$ は各時刻ごとに各 T に書かれている情報，μ は各レジスタ α の (α) の組，p は各 H_T の位置の組である．□

表 1 に VOP ごとの VCM の動作と，対応する COBOL DML を示す*．これより次の命題が成り立つ．

[命題 4.1] VOP と COBOL DML とは一対一対

* 従来の CODASYL モデルのセット型のメンバシップクラスとしては，ここでは Optional/Manual のみを考えることにする．

表 1 VCM 演算 (VOP) と COBOL DML
Table 1 VCM Operations and COBOL DML's.

VOP	意味	COBOL DML
(1) first (X);	Hx を Tx の先頭に移動 (H)	find first X within A.*
(2) last (X);	Hx を Tx の最後に移動 (T)	find last X within A.
(3) next (X);	Hx を一つ右に移動 (R)	find next X within A.
(4) prior (X);	Hx を一つ左に移動 (L)	find prior X within A.
(5) CALC (X);	βx 内の t の値をDにもつ単位セルに Hx を移動 (AC)	find any X.
(6) first (Y); $Y=(X_1, X_2)$ とする	H$_{X_1}$ 下の単位セルを親とする先頭の子に H$_Y$ および H$_{X_2}$ を動かす	find first X_2 within Y.
(7) last (Y);	H$_{X_1}$ 下の単位セルを親とする最後の子に H$_Y$ と H$_{X_2}$ を動かす	find last X_2 within Y.
(8) next (Y);	H$_Y$ を一つ右へ動かす (H$_Y$=R). H$_{X_2}$ も H$_Y$ 下のR実現値に動かす	find next X_2 within Y.
(9) prior (Y);	H$_Y$ を一つ左に動かす (H$_Y$=L). H$_{X_2}$ も H$_Y$ 下のR実現値に動かす	find prior X_2 within Y.
(10) owner (Y);	H$_Y$ 下の単位セルの親に H$_{X_1}$ を動かす	find owner within Y.
(11) get (X);	mx=RD	get X.
(12) if $\sigma=F$. go to L;	$(\sigma)=F$ ならば, ID=L の単位セルに H$_P$ を動かす	if $\sigma=F$, go to L.
(13) open (T$_i$);	HT$_i$ を TT$_i$ の先頭に移動 (H)	open T$_i$.
(14) close (T$_i$);	HT$_i$ を TT$_i$ の最後に移動 (T)	close T$_i$.
(15) write (T$_i$);	HT$_i$ の単位セルに βT$_i$ を書き出す (AP)	write T$_i$.
(16) accept (X, v);	v に (δx) を記憶する	accept v from X current.
(17) accept (Y, v);	v に (δ_Y) を記憶する	accept v from Y current.
(18) find (X, v);	Hx を, (v) を ID (db キー) としてもつ単位セルに動かす	find X; db-key is v.
(19) stop;	VCM の停止	stop.
(20) store (X);	(βx) を Lx を満たすように Tx に加える (AP)	store X.
(21) erase (X);	Hx 下の VR 実現値を含むすべての単位セルをテープから消す (DL)	erase X.
(22) modify (X);	Hx のいる単位セルのDを (βx) とする (RP)	modify X.
(23) connect (Y);	H$_{X_1}$, H$_{X_2}$ 下の単位セルを親子として, T$_Y$ に L$_Y$ を保つように加える	connect X_2 to Y.
(24) disconnect (Y);	DL を行う (H$_Y$ のいる単位セルの消去)	disconnect X_2 from Y.
(25) modify (Y);	H$_{X_2}$ 下の単位セルの親を H$_{X_1}$ 下の単位セルにする	modify X_2 only Y.
(26) go to L;	ID=L の単位セルに H$_P$ を動かす	go to L.
(27) $p(v_1, \cdots, v_e)$	$v_1, \cdots, v_e$ 内の値について述語Pを評価する	$p(v_1, \cdots, v_e)$
(28) $v_i \leftarrow v_j$;	(v_j) を v_i に記憶する	move v_j to v_i.
(29) $\Lambda(v_1, \cdots, v_e)$	$v_1, \cdots, v_e$ 内の値について関数 Λ を評価する	$\Lambda(v_1, \cdots, v_e)$
(30) read (T_i)	HT$_i$ のいるDの情報を βT$_i$ に読み出す	read T_i

* Aは, X の属するエリア [CODAS 73] である.
以下で X と $Y=(X_1, X_2)$ はおのおの VR 型と VS 型とする.

応し，その動作は表1で表される．

以上より，仮想 CODASYL データベース V 上の COBOL DML の意味が多テープチューリング機械の動作として正確に記述され，CODASYL DBS アーキテクチャ設計の基礎を与えることができる．これにより，CQG (3.2.2項) の動作が明確に与えられ，以下に述べるシステム評価の考察の基礎を与える．

5. CODASYL 問合せの VOP 表現

概念データ構造 C 上の CODASYL 問合せグラフ (CQG) の条件を満たす操作演算が，仮想 CODASYL データ構造上の仮想操作演算 VOP (すなわち，COBOL DML) としてどのように表現されるかについて論じる．

5.1 CQG の VOP 表現可能性

[定義5.1] VOP (DML) の系列を VOP(DML) プログラムとする． □

[定義5.2] CQG Q の目標リスト Λ と検索条件式 Ψ とを満たす目標集合 G を導出する VOP (および DML) プログラム P が存在するとき，P は Q の条件を満たすという． □

[定理5.1] 任意の CQG Q の条件を満足する VOP (および DML) プログラム P が存在する．

[証明] 次の二つの基本操作ができれば定理は明らかである．①Qの点 r_i, r_j とその主辺Sに与えられている条件をすべてチェックし，それを満たす目標集合 G_S を選ぶ操作と，②これらの G_S を1本または複数本のテープに出力し，その後与えられた順序で出力する操作．①の操作は，各点 r_i, r_j, (r_i から r_j) の主辺Sの条件 $\rho_i, \rho_j, \sigma_S, \pi_{ij}$ を満足する R 実現値対 ($o_i \in R_i, o_j \in R_j$) に対応した VR 実現値対 ($v_i \in X_i, v_j \in X_j$) (X_i と X_j はおのおの，R_i と R_j に対応した VRO 集合) を見いだすことで行える．それは次のようにして求められる．(a) X_i を表1の VOP(1)と(2)を用いて ρ_i を満たす v_i を見つけ，(b)その子のなかから(6)と(8)を用いて操作し，ρ_j と π_{ij} を満たす v_j を見つけ，(c)Oテープ T_{o_s} に (v_i, v_j) を(15)で書く．よって，①を VOP で行える．また②を VOP で行えるのは明らかである． ■

本定理は，CQG の条件と VOP との対応を考えるうえの基本として必要である．

5.2 最適化—出力テープ数の最小化

CQG の条件を満たす VOP プログラムは，定理5.1で示した以外にも存在する．そのなかで，ある目標値を最適にするプログラムを求める問題がある．目標値として，①Oテープ数の最小化と，②応答時間の最小化との二点[5]がある．定理5.1の方法では，検索のために主辺数+1本のOテープが必要になる．Oテープは，通常の物理的ファイルに対応するので，テープ数の増加はファイル空間と操作量の増加をもたらし，好ましくない．以下の考察では，Oテープ数の最小化問題を考える．②については，文献9) で論じる．

5.2.1 巡航木

[定義5.3] VOP (および DML) プログラム P が次の条件を満たすとき，P は巡航的であるという．①1本のOテープ T_o のみを用い，②T_o の open と close は1回のみで，他に T_o 上の write のみを用い，③close したとき，T_o 内に目標集合 G が定められた順序に従って書き出されている (冗長な組を含んでもよい)． □

すなわち，VCM が，P に基づいて，DB テープを動作させながら，T_o に順編成ファイルに対応した結果を順次書いていく場合である．

[定義5.4] CQG Q が主辺について連結で，閉路をもたず，ある点を根とするとき，Q を木と定義する．根と葉間の路を街道とし，各主辺Sに接続する2点 r_i と r_j のなかで，根に近い点 r_i を上点，他の r_j を下点とする．下点 r_j がS型 S の子の O 変数を表すときM型，親のときO型とする．この主辺Sについて，R 実現値 $o_i \in R_i, o_j \in R_j$ が親子または子親関係にあるときのみ真となる主述語 $\beta_S(o_i, o_j)$ を定義する． □

[定義5.5] 木Qが次の条件を満たすとき，Qを巡航木という．①各主辺で，上点に対し下点が順序づけられ，②全出力点はQの最右の街道内にあり，③任意の結合辺は同一街道内にある．ここで，葉に最も近い出力点と根との間の路を幹とする．各点 r について，r から根までの路内のすべての点 r' と r 間の条件式 $\pi_{rr'}$ の積を上方結合式 Π_r とし，下方結合式 Δ_r を Π_r 以外の r の結合式の積とする．根 r_0 から r_i までの路を $r_0, S_1, \cdots, S_i, r_i$ とすると，r_i の上方条件式 ϕ_i を次のように定義する．o_j を O 変数 r_j がとるR実現値とすると ($j=0, 1, \cdots, i$)

$$\phi_i(o_0, \cdots, o_i) \triangleq \begin{cases} \text{true if } \rho_i(o_i) \wedge \beta_{S_i}(o_{i-1}, o_i) \wedge \Pi_i(o_0, \cdots, o_i) \\ \text{false otherwise} \end{cases} \quad (8)$$ □

図3の CQG は根を r とする木であるが，結合辺が同一街道内にないので巡航木ではない．しかし，根を

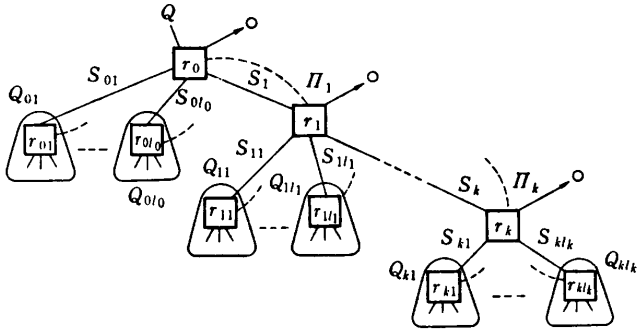


図 6 巡航木
Fig. 6 Navigational tree.

er1 とすると巡航木になる.

図 6 の巡航木Qを考える. Qの根を r_0, 幹を $r_0, S_1, r_1, \cdots, S_k, r_k$ とする. 各点 r_i は主辺 S_{ij} で連結する幹外の下点 $r_{ij}(j=1,\cdots,l_i(\geq 0))$ をもつ. Q_{ij} を r_{ij} を根とする部分木とする. 幹点 r_h のとるR実現値を $o_h(h=0,1,\cdots,i)$ とする. $o_0,\cdots,o_i$ に対して, Q_{ij} 内の全点で(1)式の上方条件式 ϕ が真となるときのみ真となる Φ_{ij} を Q_{ij} の部分木述語とする.

以上より巡航木のQの意味 g_Q は次のようになる.

$$g_Q=\{t \mid t=\Lambda(o)\wedge o=(o_0,o_1,\cdots,o_k)\bigwedge_{i=0}^{k}\mathcal{P}_{R_i}(o_i)$$

$$\bigwedge_{i=0}^{k}\phi_i(o_0,\cdots,o_i)\quad\bigwedge_{i=0}^{k}\bigwedge_{j=1}^{l_i}\Phi_{ij}(o_0,\cdots,o_i)\}\qquad(9)$$

すなわち, 各幹点 r_i のとるR実現値 o_i の組 $o=(o_0,\cdots,o_k)$ に対して, r_i のとる o_i は, $\rho_i(o_i)\wedge\beta S_i(o_{i-1},o_i)\wedge\Pi_i(o_0,\cdots,o_i)$ で, かつ全部分木条件 $\Phi_{ij}(o_0,\cdots,o_i)$ を満たすとき, $t=\Lambda(o)$ が目標集合 G の組となる.

[性質5.1] (8)式の論理演算を行う VOP が存在する.

[略証] 表1の(6)(8)(10)(11)(12)の VOP を用いて, (8)式の論理演算を行えることは明らかである. ∎

[定理5.2] 任意の巡航木Qが与えられたとき, 一本のOテープを用いて, Qの条件を満たす目標集合を一定順序で求める巡航的 VOP プログラムが存在する.

[証明] Qとして図6を考える. Qの各点 r で, (8)式の ϕ_r の論理演算が真か偽を VOP で調べられることは, 性質5.1より明らかである. 各部分木 Q_{ij} が条件 Φ_{ij} を満たすかどうかを調べる VOP が同様に存在する.

次に出力が巡航的であることを以下の帰納法で示す. 各幹点 r_i に対応して, 出力項目値を記憶するためのレジスタ λ_i を与える. まず, 幹点が一つのとき, 巡航的であることは明らかである. 次に一般の場合を考える. いま, 根 r_0 と幹点 r_{i-1} 間のすべての点 r_p に対して, 全条件 ϕ_p, Φ_{pj} を満足するVR実現値 v_p があれば, v_p に対応した単位セルにヘッド Hy_p と Hx_p がいて, 各 λ_p には v_p の出力項目値が記憶されている. このとき, 点 r_i について, 目標組を一定順序でOテープに出力する方法を以下に示す. ①点 r_{i-1} の表すVR実現値を v^h_{i-1} とする. ここで, h はヘッド Hx_{i-1} の位置を表す. その先頭の子 (r_i はM型) または親 (r_i はO型) v_i を求める. ②v_i について全条件 ϕ_i, Φ_{ij} を調べ, ③すべて満足すれば, v_i の出力項目値を λ_i に書く. ④②の条件の一つでも満足しなければ, Hy_i を一つ右に進めて②の操作を行う. ⑤進められない ($\sigma=F$ または, r_i はO型) ならば, r_{i-1} について次のVR実現値 v^{h+1}_{i-1} を選び①を行う. 以下, 最下の幹点まで以上の操作を行う. r_i が最下の幹点 (r_k) のときは, 必要とする全条件を満足することになり, $\lambda_0,\cdots,\lambda_k$ より目標組 t を求め, Oテープに出力する. このとき, 出力系列は, システムの定めた順になっていることは明らかである. ∎

[定義5.6] 木Qが次の条件を満足するとき, Qを準巡航木という. ①出力点は最右の街道内か, または出力点から根までの全点がO型の路内に存在し, ②各主辺で上点に対し下点が順序づけられ, ③各結合辺は同一路内にある. □

[定理5.3] 任意の準巡航木Qの条件を満足する巡航的 VOP (および DML) プログラムが存在する.

[証明] 略. ∎

5.2.2 CQG の巡航木表現

[定義5.7] 任意の二つの CQG Q と Q′ が, 同一の目標リストと同値な検索条件式を表すとき, 両者は等価であるという. □

[定理5.4] 任意の CQG Q と等価な巡航木 N が存在する.

[証明] 任意のグラフの点と辺を複数回たどることを許すと, すべての点と辺とを含む路が存在することは明らかである. 路は巡航木なので, 定理は成り立つ. ∎

[定理5.5] 任意の CQG の条件を満足する巡航的 VOP (および DML) プログラムが存在する.

[証明] 定理5.2と5.4より明らかである. ∎

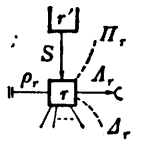
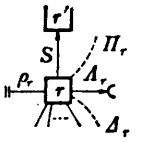
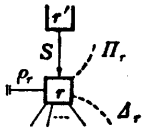
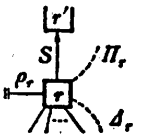
5.2.3 巡航木の条件を満たすVOPプログラムの構成

巡航木Nから，4.2節で与えたVCM上の巡航的VOPプログラムの構成手順VOPGを以下に与える．

［VOPG手順］ Nの根をr_0，各主辺Sの上点をr'，下点をrとする．まず，r_0に対して表2(a)よりVOPをPテープT_Pに書く．次に，点r'までVOPが構成されているとする．r'から与えられた順序に従って下点rを求め，表2(b)～(e)より条件に従ってVOPを求めT_Pに書く．同様のことを，以下の全点について行う．すなわち，depth-firstの木探索を行う． □

表2内で代表的な(b)を説明する．主辺Sの上点をr'，下点rとする．いま，r'が表すVR実現値v'に対応した単位セルに$H_{X'}$がいるとする．rにH_Xのいる単位セル内のVR実現値のdbキー，目標項目値，下方結合式Δ_r内のrの項目値のおのおのを，必要に応じて記憶する．おのおのの記憶を$\gamma_r, \lambda_r, \alpha_r$とする（おのおのは，VCM内の作業用レジスタである）．以下にrでの操作を示す．ⓐr'のv'の先頭の子をT_Yより求める(②)．ⓑ存在すれば(⑤)，T_XよりDの情報をβ_Xに読出し(⑥)，条件式ρ_r, Π_rを調べる(⑦)．ⓒ条件を満たせば，β_Xより必要な情報をα_rとλ_rに記憶する．rの一つ下位の点について以下同様に行う．ⓓ条件を満たさねばH_Yを一つ右に進め(④)，ⓐ～ⓓを行う．ⓔ子がよくなったとき，v'の次の順序のVR実現値v''をv'としてⓐ～ⓔを行う(⑤)．rが最右の葉のときは，全条件を満たせば，幹内の各出力点r_Pの(λ_{rP})よ

表2 CQGとVOP
Table 2 CQG and VOP.

CQG	VOP
(a) 	open (T_0); $\beta_0' \leftarrow \phi$; first (X_0); go to L_0; K_0: ⟦find (X_0, τ_0);⟧ M_0: next (X_0); L_0: if $\sigma = F$ {close (T_0); stop;}; get (X_0); if $\sim\rho_0(\beta_{X_0})$ go to M_0; accept (X_0, τ_0); $\alpha_0 \leftarrow \beta_{X_0}.\tau_0$; $\lambda_0 \leftarrow \beta_{X_0}.\Lambda_0$; […]* [go to M_0;]*
(b) 幹内のM型点 (r) 	S_r: ⟦find $(X_{r'}, \tau_{r'})$;⟧ ① first (Y); go to L_r; ② K_r: ⟦find (X_r, τ_r);⟧ ③ M_r: next (Y); ④ L_r: if $\sigma = F$ go to K_l; ⑤ get (X_r); accept (X_r, τ_r); ⑥ if $\sim(\rho_r(\beta_{X_r}) \wedge \Pi_r(\alpha_0, \cdots, \alpha_{r'}, \beta_{X_r}))$ go to M_r; ⑦ $\alpha_r \leftarrow \beta_X.\tau_r$; $\lambda_r \leftarrow \beta_X.\Lambda_r$; ⑧ […]* ⑨ [go to M_r;]* ⑩
(c) 幹内のO型点 (r) 	S_r: ⟦find $(X_{r'}, \tau_{r'})$;⟧ owner (Y); if $\sigma = F$ go to K_l; get (X_r); accept (X_r, τ_r); if $\sim(\rho_r(\beta_{X_r}) \wedge \Pi_r(\alpha_0, \cdots, \alpha_{r'}, \beta_{X_r}))$ go to K_l; $\alpha_r \leftarrow \beta_{X_r}.\tau_r$; $\lambda_r \leftarrow \beta_{X_{r'}}.\Lambda_r$; […]* [go to K_l;]*
(d) 幹外のM型点 r 	S_r: ⟦find $(X_{r'}, \tau_{r'})$;⟧ first (Y); go to L_r; K_r: ⟦find (X_r, τ_r);⟧ M_r: next (Y); L_r: if $\sigma = F$ go to K_l; get (X_r); accept (X_r, τ_r); if $\sim(\rho_r(\beta_{X_r}) \wedge \Pi_r(\alpha_0, \cdots, \alpha_{r'}, \beta_{X_r}))$ go to M_r; $\alpha_r \leftarrow \beta_{X_r}.\tau_r$; […]* [go to K_l;]*
(e) 幹外のO型点 r 	S_r: ⟦find $(X_{r'}, \tau_{r'})$;⟧ owner (Y); if $\sigma = F$ go to K_l; get (X_r); accept (X_r, τ_r); if $\sim(\rho_r(\beta_{X_r}) \wedge \Pi_r(\alpha_0, \cdots, \alpha_{r'}, \beta_{X_r}))$ go to K_l; $\alpha_r \leftarrow \beta_{X_r}.\tau_r$; […]* [go to K_l;]*

ここで，
$[\cdots]^* = [\beta_0 \leftarrow \Lambda(\lambda_0, \lambda_1, \cdots, \lambda_k);$
if $\beta_0 \neq \beta_0'$ {write (T_0); $\beta_0' \leftarrow \beta$;} ;]*
$l = r$から根r_0までの街道内で，rに最も近いM型点，なければr_0
$k = {}_{\sqcup}r'$を親とし，rの右隣りの点．なければl
$\beta_{X_r}.\Lambda_r$＝レジスタβ_{X_r}内の目標項目集合Λ_rの値
$\beta_{X_r}.\tau_r$＝レジスタβ_{X_r}内のΔ_rの目項集合τ_rの値
[…]*は，点rが最右の葉点のとき必要なVOP

り目標組を求めOテープに出力する(⑨)*. 表2の他も同様である.

[命題5.1] 任意の巡航木に対して, これを満足する巡航的 VOP プログラムを, 表2と VOPG 手順より構成できる.

[証明] 任意の巡航木の各主辺Sと下点rについて考えると, rは幹内か幹外の点で, かつM型かO型である. したがって, Sとrは, 表2の(b)～(e)のどれかのタイプであり, 木の根は(a)のタイプである. VOPG により構成された VOP プログラムの出力が巡航的であることは, 定理5.2と表2の VOP の意味より明らかである. ∎

[定理5.6] 概念 CODASYL データ構造上の任意の CQL 問合せの条件を満たす巡航的 VOP プログラムを, 表2と VOPG 手順より構成できる.

[証明] 命題3.1より, CQL 問合せと CQG とは一対一対応する. さらに, 定理5.4と命題5.1が成り立つので, 本定理は明らかである. ∎

すなわち, CQL で表せる任意の検索演算から, この条件を満たす巡航的 VOP プログラムを, 表2と VOPG 手順とによって構成できる.

6. 非手続的インタフェース

文献8)では, ローカル概念 (LC) モデルと LC 操作言語 RQL とを定め, LC と論理 CODASYL データ構造の一対一対応性を示した. データ構造の一対一対応性より, 次の命題が成り立つことは明らかである.

[命題6.1] 任意の RQL 検索演算と CQL 検索演算 (および CQG) とは一対一対応する. ∎

[定理6.1] 任意の RQL 検索演算の条件を満たす巡航的 VOP (DML) プログラムが存在する.

[証明] 命題6.1と定理5.5より明らかである. ∎

[定理6.2] 任意の RQL 演算条件を満たす VOP (DML) プログラムが存在する.

[証明] RQL 演算は (ρ, Λ, Ψ) と表される. 基本操作演算 ρ を VOP で表せることは文献8)より明らかである. 目標リスト Λ と検索条件 Ψ を VOP で表せることは, 定理6.1より明らかなので, 定理は成り立つ. ∎

よって, LC データ構造上の任意の RQL 演算の表す条件を COBOL DML プログラムで表せることがわかり, CODASYL DBS 上に非手続的インタフェース設計の基礎が明らかとなった.

* COBOL DML では, 同一R型に対応した複数の点が存在するときには, ヘッド位置を正しく保つために表2の[…]の VOP が必要である.

7. おわりに

本論文では, CODASYL DBS を論理, 概念, 仮想, 物理の4層から成るとし, このなかで, 概念層と仮想層, および層間写像を論理的に明らかにした. 概念層では, 集合と部分関数から成る概念データ構造 $\boldsymbol{C}$ と, 述語論理形式の CODASYL 問合せ言語 (CQL) および等価な CODASYL 問合せグラフ (CQG) との定義を与えた. CQG は見やすい形になっていることから, 利用者インタフェースとして優れている. 仮想層では, $\boldsymbol{C}$ に順序関係を導入した仮想データ構造 $\boldsymbol{V}$ を与え, 従来の COBOL DML と一対一対応する仮想操作演算 (VOP) を実行できる仮想 CODASYL 機械 VCM を多テープチューリング機械で定義し, 従来の DML の意味を明らかにした. 層間写像として, すべての CQG を, 一本の出力テープとこれへの逐次書込みだけで目標集合を導出できる巡航的 VOP プログラムによって正しく表せることを論理的に明らかにし, その具体的方法を示した. 本方法は, 文献 5), 6) に対して, 完全な逐次出力で目標集合を導出でき, 文献 12) に対しては, さらに任意の結合辺と, 主辺について木型以外の検索演算を許している点で優れている.

CQG の条件を満たす VOP を考えるとき, 出力テープ数と操作量の最小化に加えて, 応答時間と出力量の最小化が目標となる. これについては文献 9) で論じる.

以上より, 本論文は, 従来の CODASYL DBS 上に非手続的利用者インタフェース設計の論理的基礎を与えている. また, 本論文で述べた概念に基づいて, CODASYL DBS 上の非手続的インタフェースシステムローカルデータベースプロセッサ (LDP-V2)[3)~5),7),8)] が Jipdec の AIM (M-170F), ADBS (Acos-700) 上ですでに実働している.

謝辞 LDP-V2 の開発を共同して行った(財)日本情報処理開発協会の横塚実氏に感謝する.

参考文献

1) CODASYL DDL Committee: CODASYL Data Description Language, *J. Dev.* (1973).
2) Olle, T. W.: *The Codasyl Approach to Data Base Management*, John Wiley & Sons, New York (1978).
3) Takizawa, M., et al.: Resource Integration

and Data Sharing on Heterogeneous Resource Sharing System, Proc. ICCC '78, pp. 253-258 (1978).
4) Takizawa, M. et al.: The Four-Schema Concept as the Gross Architecture of Distributed Databases and Heterogeneity Problems, *JIP (IPSJ)*, Vol. 2, No. 3, pp. 134-142 (1979).
5) Takizawa, M. et al.: Query Translation in Distributed Databases, Proc. IFIP '80, pp. 451-456 (1980).
6) Takizawa, M.: Distribution Problems in Distributed Databases, *JIP (IPSJ)*, Vol. 5, No. 3, pp. 139-147 (1982).
7) 滝沢，横塚他：CODASYL データベースシステムに対する関係インタフェースシステム (LDP-V 1.5)の設計と実現，情報処理学会論文誌, Vol. 23, No. 3, pp. 665-675 (1982).
8) 滝沢，野口：CODASYL データベースシステムに対する非手続的更新インタフェースの基本概念，情報処理学会論文誌，Vol. 24, No. 6, pp. 857-866 (1983).
9) 滝沢，野口：CODASYL 問合せの最適化と評価について，準備中.
10) Held, G. et al.: INGRES—A Relational Data Base System, AFIPS Conf. Proc., pp. 409-416 (1976).
11) Codd, E. F.: A Relational Model of Data for Large Shared Data Bank, *CACM*, Vol. 13, No. 6, pp. 337-387 (1980).
12) Dayal, U. et al.: Query Optimization for CODASYL Database Systems, Proc. ACM SIGMOD, pp. 138-150 (1980).
13) Biller, H. et al.: On the Semantics of Data Bases: The Semantics of Data Manipulation Language, Proc. IFIP TC-2, pp. 239-267 (1976).
14) Gangopadhyay, D. et al.: Semantics of Network Data Manipulation Languages: An Object-Oriented Approach, Proc. the 8th VLDB, pp. 357-369 (1982).
15) CODASYL DDL Committee: Report of the CODASYL Data Description Language Committee, *Inf. Syst.*, Vol. 3, No. 4, pp. 247-320 (1978).

(昭和58年9月21日受付)
(昭和59年3月 6 日採録)