

演繹能力を付加した DBMS について

大阪大学 工学部 山口高平 西岡弘明
打浪清一 手塚慶一

1 - まえがき

最近、データベース上に推論能力を付加した効率の良い質問応答システム“演繹データベースシステム”が注目されているが、これは人工知能における知識表現・論理的演繹の研究とデータベースにおける処理技術・設計技術の結合として考えられる。両者で研究されているデータモデル技術は、大別して“述語論理に基づくもの”・“ネットワークに基づくもの”・“手続きに基づくもの”に分類される。

本稿においては、記法の単純さや表現力によって、実世界をモデル化するためのスキーマとして述語論理を選び、定理証明理論とデータモデル理論（Codd の関係モデル）の結合から、演繹データベースシステムを考察する。

2 - 定理証明理論の研究状況

一階述語論理に基づく定理証明システムの実現理論として

は、1965年に F.A. Robinson が提案した導出原理が基礎となっているが、これをそのままシステムに適用したのでは、無駄や重複のある中間結果（導出形）を多数生成してしまい効率が大きな問題点になってしまう。そこで、AI用プログラミング言語（PLANNER, CONNIVER, ACTOR 等）の提案が成されたり、制限付導出法・証明戦略の研究が成されている。

一方システムの応用としては、定理証明理論が、証明図の中からプランやプログラムを抽出できるため各種の問題解決の基礎となる事を利用して、1969年に C. Green が質問応答システムを作成している。これは、定理証明理論をデータベース問題に関連づけた第一歩といえる。またデータベースの分野では、データの独立性を高めるためにスキーマ形式を採用しており、現実世界の情報を定義する概念スキーマの基本となるデータモデルの研究が現在盛んである。これらの背景を元に、定理証明理論と関係モデルを結合させて、演繹データベースシステムの研究が始ったわけであるが、このアプローチは大きく分けて、“非評価法”（J. Minker 等）と“評価法”（C.L. Chang, R. Reiter, C. Kellogg 等）とに分類する事ができる。

3 - 実用的な定理証明システム

演繹データベースシステムは、定理証明システムを応用して作成するので、効率の良い実用的な定理証明システムを用意する必要がある。そこで、効率改善という問題を Fig. 1 の様に2つの側面から捉え、各々の一般に用いられている実現方法と当研究室のアプローチ及びその結果を示す。

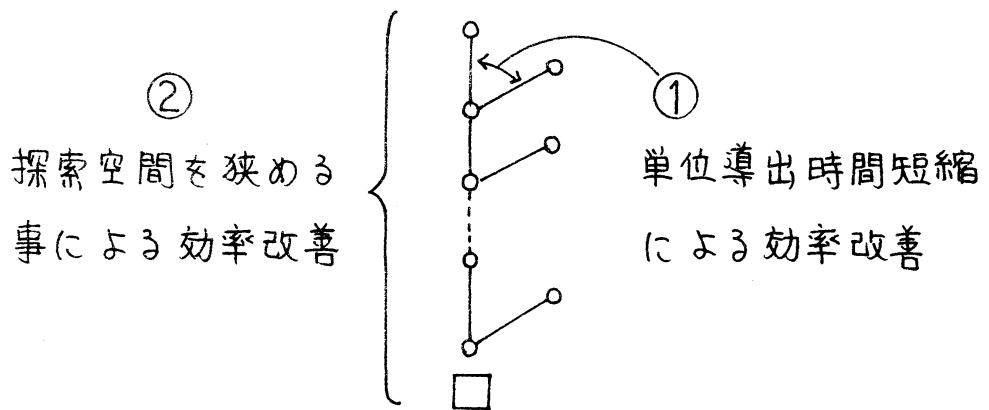


Fig. 1 効率改善のための2つの側面

TABLE 1 効率改善の実現方法

	一般手法	当研究室のアプローチ
①の実現方法	定理証明を行うのにより適したシステム(言語)の導入	定理証明システム SENRI ASSEMBLER VERSIONの作成
②の実現方法	制限付導出法の導入 戦略の導入 発見的手法の導入 補助制御言語の導入	SNL+SNLの有効戦略 SPU+SPUの有効戦略

TABLE 2 実用的な定理証明システム SENRI の効率

システムと 証明問題	SENRI ASSEMBLER VERSION	
	SNL + 有効戦略	SPL + 有効戦略
TPU EX-1	805	2838
TPU EX-2	TIME OVER	23037
TPU EX-3	16983	24803
TPU EX-4	1478	4857
TPU EX-5	37	941
TPU EX-6	87	4278
TPU EX-7	188	250
TPU EX-8	5520	2572
(TPU EX-9)	478	REJECTION

TIME OVER

30秒経過しても証明
結果が導かれな()

REJECTION

TPU EX-9は, renaming
しても Horn Set にならな
() ので, SPL では Rejec
tion された.

単位時間

msec

使用計算機

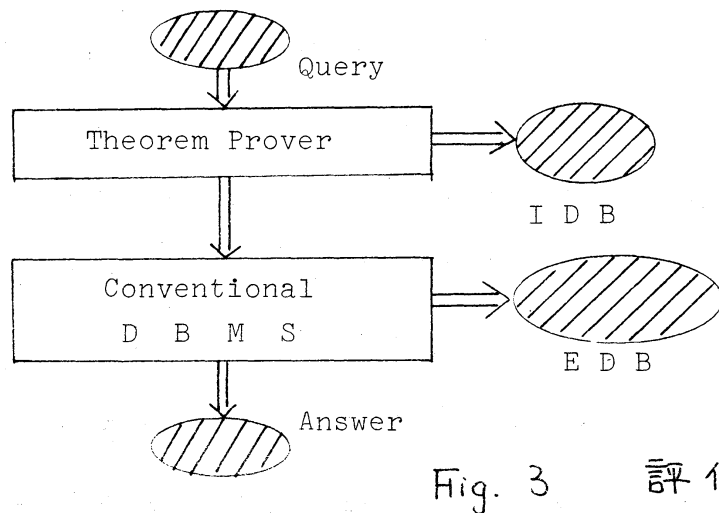
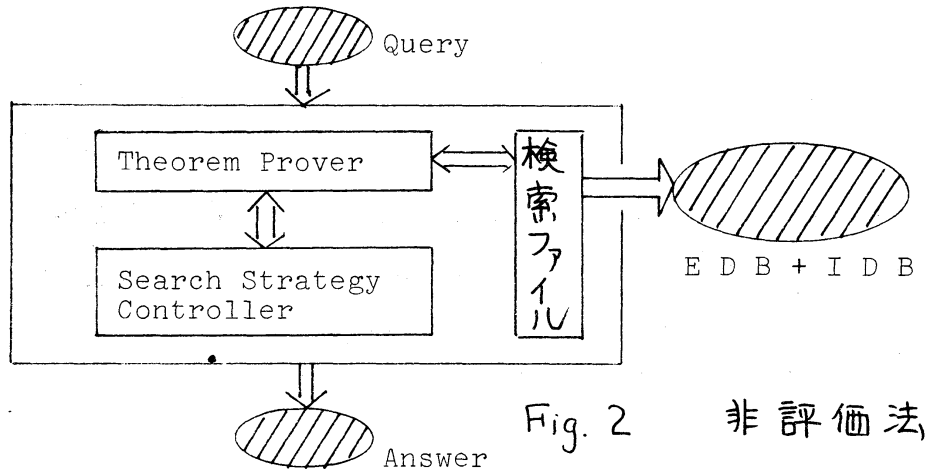
ACOS-900

4-非評価法と評価法

非評価法は, Green 等の質問応答システムの研究を改善して確立するものであり, 外延データと内包データを共に公理と考え *query clause* を初期節点とする top-down goal tree (一つの証明過程) を形成して答を得る方法であり, Search Strategy Controller と検索ファイルによって効率化を図っている。しかしながら, 実効的な効率改善までには達せず, 大規模データベース上での実現には困難があると考えられる。

(Fig. 2 参照)

評価法は, *query clause* を定理証明システムにより, 内包データを用いて外延データのみから成る形に変形し, 通常の DBMS によって答を得る方法であり, 定理証明システムは IDB だけの処理を行い, 非評価法に比べて, 大量のデータを含む EDB は通常の関係 DBMS が処理を行うため, より実際的なアプローチであると考えられる。(Fig. 3 参照)



5 - 述語論理に基づくデータモデル

本稿においては、記法の単純さや表現力によって、実世界をモデル化するためのスキーマとして述語論理を用いるが、以下にこのモデルの長所と短所について述べる。

[述語論理モデルの長所]

- (1) モデル化される世界を記述するすべての事実に対して、単純で統一された表現を与える。
- (2) 述語論理で表現されたモデルは、事実がモデルに対して最小の副作用でもって、モデルに付加されたりモデルから削除されたりするという点で、*modular* である。
- (3) 述語論理は、演繹推論や一貫性検査のために、強力な定理証明手続きを提供する。

[述語論理モデルの短所]

- (1) モデルの手続き的な側面を表現する primitive が欠如している。
- (2) 表記法としての述語論理は、理解しにくい。

短所(1)に対しては、自然な手続き的解釈が成立する Horn 節による述語論理プログラミングが対策として考えられ、短所(2)に対しては、例題型・自然言語型等の関係言語族を手助けにすれば、理解が容易になると考えられる。

一階述語論理は、各変数の sort (変数の値がとりうる領域) を外延的に仕様化する必要があるのに対して、一階多類論理では sort の集合が定義され、各変数は特別な sort において値をとる。関係モデルでは、属性が値をとる領域を各関係に対して仕様化する必要があるため、一階多類論理に基づくデータモデルの使用は好ましいと考えられ、述語論理モデルのうち特にこのモデルを本稿では採用する。

[一階多類論理モデルの特徴]

- (1) 一階多類論理は、一階述語論理と表現能力が同等で、人間の常識から考えてナンセンスな文を生成しない様にする事が容易である。
- (2) 変数 x が記述対象のどの部分集合に属するかという定義は、sort の集合の形式で与えられるためそれだけ述語の数が減り、また述語の引数がどの定義域に属するかを考慮する事によって煩雑サーチの効率化が図れる。

6 - 演繹データベースシステムの構成に用いた定理証明理論

6-1 結合グラフ (connection graph)

定理証明においては、リテラル間の潜在的に補となる可能性を表すグラフー結合グラフーを反駁を行うために利用した

が、演繹データベースシステムではEDBのアクセスプランを抽出するために用いる。従って、基本関係(計算機に明らかに蓄えられている関係)を表すリテラルと仮想関係を表す(公理によって定義された関係)リテラルとが区別される。(Fig. 4に節集合 S と S から形成される結合グラフの例を示す。p.10参照)

6-2 結合グラフの探索空間狭化

効率改善に関して探索空間を狭める制限付導出法及び戦略の研究が定理証明の分野で行われている事は既に述べたが、この理論をそのまま結合グラフの探索空間狭化に利用する事は困難である。なぜなら、定理証明においては空節をひとつ導出する事が目的であるのに対して、演繹データベースシステムでは問合せの条件を満足するすべての解を見つけ出す事が目的であるから、定理証明に対応させた場合その証明過程が重要となる。しかしながら、事実検索等の様に解がひとつの場合は、探索空間狭化のための制限付導出法・戦略の研究が結合グラフに適用できる。

節集合をHorn節集合に限定した場合多くの完全性を持つ制限付導出法が提案されているが、なかでもL.F. HenschenのSemantic Resolution for Horn Set⁽²⁾はnon-unit導出を制限

する一方、関連のある正単位節と導出を行うもので興味深く、この制限付導出法による結合グラフの探索空間狭化を考える。

— 定理 1 —

S を充足不可能な Horn 節集合とし、 I を S に対する解釈とする。この時、反駁における各導出が以下の 2 つの条件のうちひとつを満足する。

(1) 親節のひとつは、解釈 I において偽となる正単位節である。

(2) 親節のひとつとその導出形は、解釈 I によって共に偽である。

定理 1 を結合グラフに応用するためには、解釈 I の各要素と結合グラフの節点間で I に関して偽となる可能性があるものを枝で結合し、偽となる可能性がある節から、偽の可能性が保持される様にサーチを展開する事によって探索空間を狭める。Fig. 5 にその例を示すが、 \Rightarrow の方向にはサーチする必要がなく、探索空間が狭められている。

6-3 Resolution plan

clause plan は Fig. 6 に示す様に、節 (clause segment) の

$$S = \{\neg P(x) \vee Q(y), P(a), \neg Q(b)\}$$

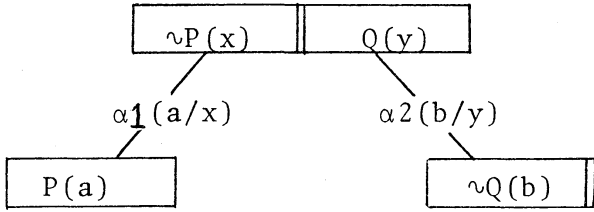
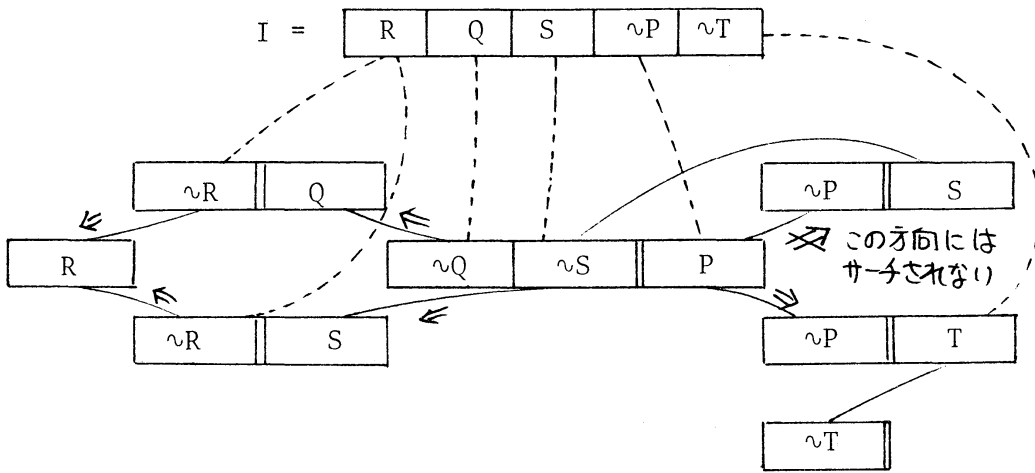


Fig. 4 結合グラフ



- : 潜在的に補となる可能性を示すアーク
- - - : 解釈Iに関して偽となる可能性を示すアーク
- ⇒ : サーチされる方向

Fig. 5 結合グラフの探索空間狭化

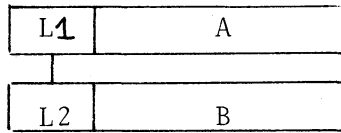


Fig. 6
 結合グラフから得られる clause plan
 [A] ∨ [B] (L1, L2)
 ← clause segment —*— plan —>
 segment

後に一連のリテラルペア (plan segment) が続く形式で表現される。今2つの clause plan $S = CP, T = DQ$ (C, D : clause segment, P, Q : plan segment) を考え、 C のリテラル $C' = \{L_1, \dots, L_m\}$ と D のリテラル $D' = \{M_1, \dots, M_n\}$ が各々結合グラフ (潜在的に補となる可能性を示すアーク) で結合されている時、

binary resolution plan

$$\Rightarrow (C - C') \vee (D - D') (L_1, M_1) \dots (L_m, M_n) PQ$$

で定義される。

6-4 Partial plan

query clause の仮想関係を表すリテラルより Resolution plan を生成し、すべて基本関係に置き換えられた時の Resolution plan を Partial plan と定義する。

6-5 Total plan

query clause の各リテラルに対して生成された Partial plan 間で代入の一貫性が保たれるものの組み合わせとして Total plan は定義される。すなわち、Total plan の clause segment は代入の一貫性が保たれる Partial plan の clause segment の論理和であり、Total plan の plan segment は

それらの plan segment の合成である。

7- 演繹データベースシステムの構成

Fig. 7 に示す様に、まず Connection Grapher によって、IDB から生成された結合グラフと *query* を結合する。次に Partial Planner によってリテラルがすべて基本関係に置き換えられた Partial plan を生成し、Consistent Checker によって代入の一貫性が保たれるものだけを Total plan として登録する。Semantic Checker によって、各定義域間の関係を示す semantic graph を参照して、代入の時意味的に矛盾が生じないかどうかを検査して、満足し得るものだけをアクセスプランとする。最後に Subproblem Selector によって定数の引数をより多く持つリテラルからアクセスする様にして、DBMS にアクセスプランを引き渡す。

8- システムのアクセスプラン抽出例

祖先関係を例にとり、「 $\exists EUS$ を祖先とする人は誰ですか？」という *query* を考えると、Fig. 8 に示す様に、まずその *query* と祖先関係を表す公理群との結合グラフが作成され、次にこれを用いて6つの Partial plan が作成される。Consistent Checker は、*query clause* はただひとつのリテラ

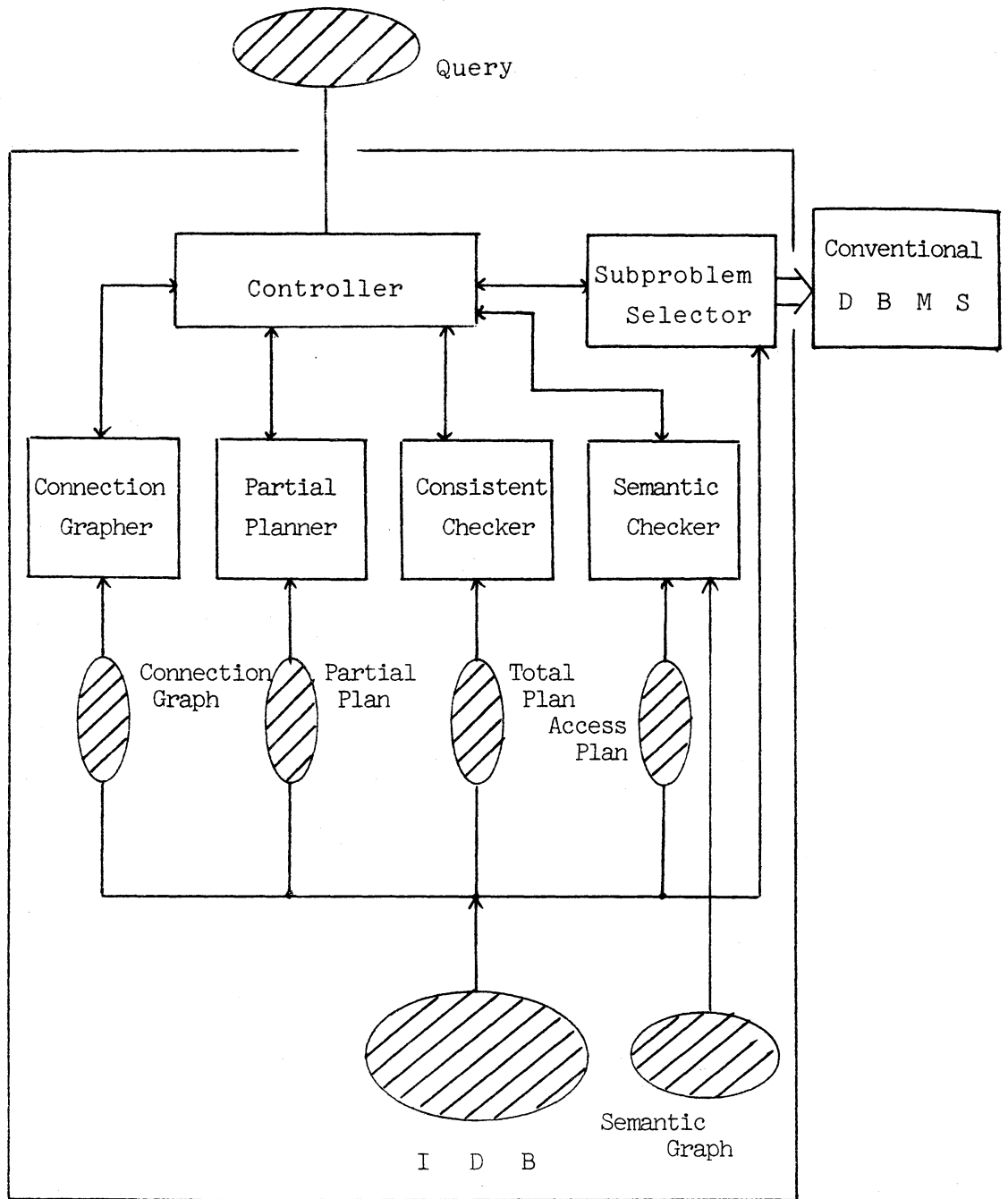


Fig. 7 演繹データベースシステムの構成

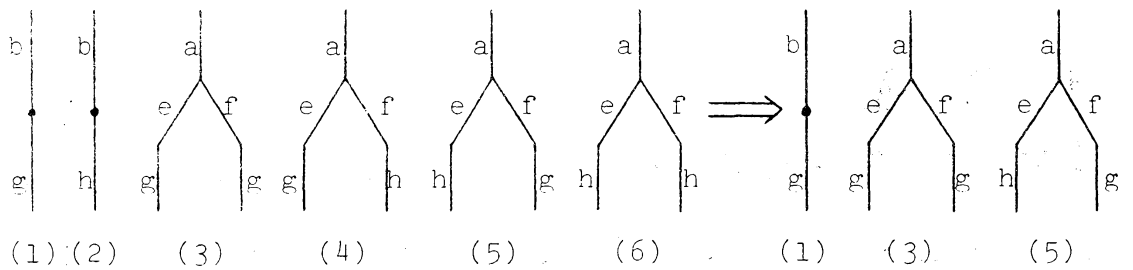
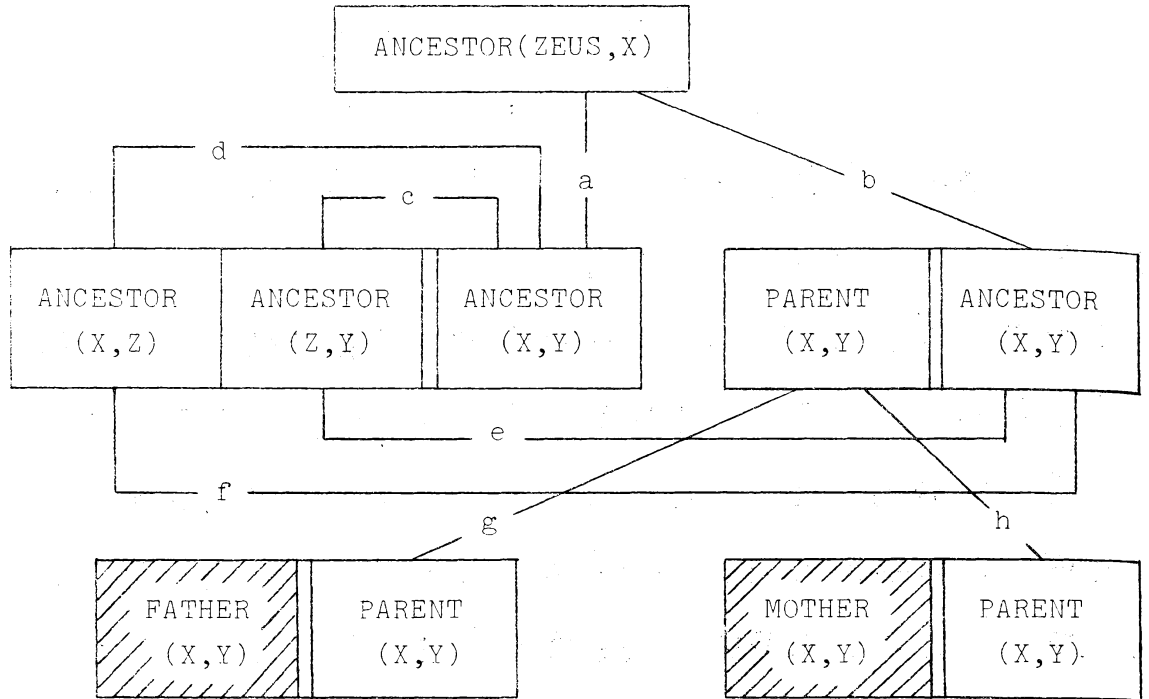
ルから成るため、代入の一貫性を検査する必要がなく、Partial plan をそのまま Total plan とする。しかしながら、Semantic Checker は、ZEUS/male が $(\forall x/\text{female})(\forall y/\text{human})(\text{MOTHER}(x,y))$ の引数 x に代入される (2)・(4)・(6) を意味的に矛盾があるとして除去し、残りの (1)・(3)・(5) をアクセスプランとする。

この例では、ANCESTOR が再帰的に定義されている公理があるため、このままでは無限の Partial plan が生成される。この様な再帰的公理の処理には、(i) リテラル数制限等による定理証明戦略の応用、(ii) 通常の DBMS からフィードバックする部分を設け、再帰的公理を使う意味を考慮する事により EDB から制御情報を抽出し Partial plan の生成を制限する。

9 - 結 論

本稿で述べた英訳データベースシステムは、Partial Planner と Consistent Checker 及び Semantic Checker に特色を持ち、また限られた問題に対しては結合グラフの探索空間を狭化する事ができ、より適切なアクセスプランだけを効率良く抽出できる。今後の課題は、(i) 一般的な結合グラフの探索空間狭化法、(ii) 概念階層構造を semantic graph に利用する (iii) SENRI サブルーチン群による英訳 DBS の実働化、等である。

Connection Graph



Partial Plan

Access Plan

(Total Plan)

Fig. 8 アクセスプラン抽出例

— 参考文献 —

- (1) C.L.Chang, R.C.Lee : " Symbolic Logic and Mechanical Theorem Proving " Academic Press. (1973)
- (2) L.J.Henschen : " Semantic Resolution for Horn Sets " Proc, 4th IJCAI, pp.46-52 (1975)
- (3) S.Sickel : " Variable Range Restrictions in Resolution " Machine Intelligence, Vol.8 pp.73-85 (1977)
- (4) Harry,K.T., Wong, et al. : " Two views of data semantics : a survey of data models in artificial intelligence and data base management ", INFOR, 15, 3 (Oct. 1977)
- (5) H.Gallaire, et al. : " An Overview and Introduction to Logic and Data Bases " in Logic and Data Bases (1978)
- (6) J.Minker : " An Experimental Relational Data Base System based on Logic " in Logic and Data Bases (1978)
- (7) Kellogg,C. et al. : " Deductive Capability for Data Management " in Systems for Large Data Bases (1976)
- (8) C.L.Chang : " Resolution Plans in Theorem Proving " Proc, 6th IJCAI, pp.143-148 (1979)
- (9) 古川 : " 評価法に基づく演繹データベースの実現 " 信学会全国大会予稿, 59-8 (1980)
- (10) W.Kim, 植村俊亮訳 : " 関係モデルデータベースシステム " in コンピュータ・サイエンス acm computing surveys' 79
- (11) 山口,西岡,打浪,手塚 : " 定理証明システム ENRI とその応用 " 信学会, EC 80-26 (1980)
- (12) 山口,西岡,打浪,手塚 : " 仮想関係を包含する質問文を関係DB M 上で評価する手法について " 関西支部連合大会 GR-29 (1980)