演繹データベースにおける直積問題クラスとそのアルゴリズム

鈴木 晋[†] 茨木俊秀[††]

演繹データベースの新しい問題クラスである直積問題クラスを定義する．直積問題クラスは右線形問題や同世代問題を包む，datalog 問題クラスの部分クラスである．このクラスを効率的に解くために直積法を提案する．従来の方法が中間データとして基礎アトムを生成するのに対し，直積法は基礎アトムの集合を直積を利用して簡潔に表す式を生成する．これにより，直積法は生成する中間データの数を減らして，問題を効率的に解こうとする．直積問題クラス全体に適用できる従来の方法の中ではマジック集合法が最も効率的であると思われる．そこで，直積法とマジック集合法の効率を比較するために，直積問題の例として同世代問題の再帰ルールを複数にした問題および非線形にした問題を使って，計算機実験を行った．実験の結果，ファクトをランダムに生成した場合，どちらの問題に対しても問題がファクトを密に含むとき，直積法がマジック集合法より効率的であることが分かった．

The Class of Cartesian Product Problems in Deductive Databases and Its Algorithm

Susumu Suzuki[†] and Toshihide Ibaraki[††]

We define a new problem class of deductive databases, called the Cartesian product problem class (abbreviated to the CP class). The CP class is a subclass of the datalog problem class, and includes the right-linear problem, the same generation problem and others. To solve the CP class efficiently, we propose the Cartesian product method (abbreviated to the CP method). Although all the existing methods generate ground atoms as intermediate data, the CP method generates Cartesian products, each of which compactly expresses a set of ground atoms. Thus the CP method tries to reduce the number of generated intermediate data and, consequently, to solve problems efficiently. Among the existing methods applicable to the whole CP class, the magic set method seems most efficient. For performance comparisons of these two methods, we conducted experiments with two types of modified same generation problems, where one had two recursive rules and the other a non-linear recursive rule. The experimental results show that the CP method is more efficient than the magic set method when problems densely contain the facts generated at random.

1. はじめに

演繹データベースの問題を効率的に解くために，適用範囲や効率の異なる様々な方法が提案されている[3),5),7)]．たとえば，右線形問題を対象とする NRSU 法[4)]，同世代問題を対象とする計数法[1)]，一般の問題（すなわち，datalog 問題クラス[7)]）と他の一部の問題）を対象とするマジック集合法[1),2),6)]等がある．ここで，右線形問題，同世代問題は datalog 問題クラスに属する．また，“NRSU 法”，“計数法”，“マジック集合法”等は，本稿では，与えられた問題をルール集合の異なる等価な問題に書き換え，そして，その等価な問題をセミナイーブ法[7)]を使ってボトムアップに処理する方法をさすものとする．

提案された方法の多くは生成する中間データの数を減らすことにより効率化を図っており，中間データとして基礎アトム（ground atom，ga と記す）を生成するという方式をとっている．

本稿では，右線形問題や同世代問題を包含する，datalog 問題クラスの新しい部分クラス（直積問題クラスと呼ぶ）を定義し，この問題クラスを効率的に解くために，直積法を提案する．

直積法は中間データとして，ga の代わりに，ga の集合を簡潔に表す式（直積型基礎アトム集合式（ground atom set expression of Cartesian product type）と呼び，gase と記す）を生成する．1 つの gase は多数の ga をまとめて表すことができるので，中間データの数を減らせる可能性がある．1 つの gase を生成し処

† 愛知工業大学工学部
Faculty of Engineering, Aichi Institute of Technology
†† 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University

理する時間は 1 つの ga のそれに比べて大きいが，もし，生成する gase の数が大変小さくなれば，問題を解く時間を全体として減らせる可能性がある．直積法はこの考えに基づいて問題を効率的に解こうとするものである．

例 1 問題 $(sg(e,Y)?, (P,D))$ は，ルール集合

$$P:\ sg(X,X) \leftarrow A(X). \quad (1)$$
$$sg(X,Y) \leftarrow B(X',X), B(Y',Y), sg(X',Y'). \quad (2)$$

とファクト集合

$$D = \{A(a), A(b), A(c), A(d), A(e), A(f)\} \cup \{B(a,c), B(a,d), B(b,c), B(b,d), B(c,e), B(c,f), B(d,e), B(d,f)\}$$

と問合せ $sg(e,Y)?$ よりなる． □

この問題が同世代問題と呼ばれる問題であり，直積問題の 1 例である．直積法は，この問題に対し，8 個の gase $sg[\{a\}\times\{a\}], sg[\{b\}\times\{b\}], \ldots, sg[\{f\}\times\{f\}], sg[\{c,d\}\times\{c,d\}], sg[\{e,f\}\times\{e,f\}]$ を生成する．ここで，たとえば，$sg[\{e,f\}\times\{e,f\}]$ は ga 集合 $\{sg(x,y) \mid x,y \in \{e,f\}\}$ を表す．次に，$sg[\{e,f\}\times\{e,f\}]$ の中から問合せ $sg(e,Y)?$ に適合する ga を選択して解集合 $ANS = \{sg(e,e), sg(e,f)\}$ を得る．仮にこの問題から生成可能な ga をすべて生成すると 10 個の ga $sg(a,a), \ldots, sg(f,f)$ が得られるので，直積法は中間データの数を 10 から 8 に減らしている．

直積問題クラス全体に適用できる従来の方法には，一般の問題を対象とする方法がいくつか知られているが，それらの中で，マジック集合法[1),2),6)]は最も効率的な方法の 1 つと思われる．直積法とマジック集合法の効率を比較するために，本稿では，直積問題の例として同世代問題の再帰ルールを複数にした問題および非線形にした問題を考え，これらの問題を使って計算機実験を行う．問題に現れる（すなわち，ファクト集合に現れる）異なる定数の数を n と記し，EDB 述語 E のファクトの数を m' と記す．m' が n に比べて大きい $(m' \gg n)$ とき，問題は**述語 E のファクトを密に含む**という．また，その程度を表す値 m'/n を述語 E の**ファクト密度**と呼ぶ．たとえば，例 1 の問題の場合，$a, \ldots, f$ の 6 個の異なる定数があり（$n=6$），$B(a,c), \ldots, B(d,f)$ の 8 個の B ファクトがある（$m'=8$）ので，EDB 述語 B のファクト密度は 8/6 になる．実験の結果，ファクトをランダムに生成した場合，どちらの問題においても問題がファクトを密に含むとき，直積法がマジック集合法より効率的であることが分かった．

ところで，問題を解く方法はボトムアップ計算法とトップダウン計算法に分類することができ，そして，ボトムアップ計算法において従来提案されてきた効率化手法は次の 3 つに分類することができる．(a) SIPS（sideways information passing strategies）[7)]を用いて生成可能な ga を制限することにより，与えられた問題を意味（問題から生成可能な ga の集合であり，最小不動点とも呼ばれる）の小さな等価な問題に書き換える手法．(b) 問題の意味を効率的に計算する手法．(c) その他の手法．(a) の代表はマジック集合法の中で使われている問題の書き換え法（マジックルール書き換えと呼ばれる）であり，(b) の代表はマジック集合法，NRSU 法，計数法等の中で使われているセミナイーブ法である．本稿で提案する直積法はボトムアップ計算法であり，上記の手法分類では (b) に属する．ゆえに，直積法とマジック集合法の効率を比較するのではなく，直積法とセミナイーブ法の効率を比較する方が自然なようにも思われる．本稿において直積法とマジック集合法の効率を比較する理由について，実験の後，補足説明する．

本稿の以降は次のように構成される．2 章で用語を定義し，3 章で直積問題クラスを定義し，4 章でルールから gase を生成する方法を説明する．5 章で直積法を与え，6 章でその適用例を示し，7 章でその正当性を証明する．8 章で実験について述べる．9 章で直積法とマジック集合法を比較する理由を補足説明し，最後に，10 章でまとめる．

2. 準　　備

初めに，従来の用語を説明する．ほとんどは演繹データベースの慣例用語である（たとえば，文献 3), 7)）．

定数を $a, b, \ldots$，定数ベクトルを $\boldsymbol{a}, \boldsymbol{b}, \ldots$，変数を $X, Y, \ldots$，変数ベクトルを $\boldsymbol{X}, \boldsymbol{Y}, \ldots$，EDB（extensional database）述語を $A, B, \ldots$，IDB（intensional database）述語を $p, q, \ldots$ と記す．述語とそれに続く変数または定数の並びからなる式をアトム，変数を含まないアトムを基礎アトム（ga と記す），問題中に与えられる ga をファクトという．たとえば，$p(X,Y)$ はアトム，$p(a,b)$ は ga，1 章の例 1 の $B(a,c)$ はファクトである．なお，文献によっては，アトムは原子論理式と呼ばれる．

ルール集合とファクト集合と問合せよりなる 3 つ組を問題という．問題のルール集合とファクト集合より生成可能な ga の集合を問題の意味といい，IMP と記す．生成可能な，問合せに適合する ga の集合を解集合といい，ANS と記す（$ANS \subseteq IMP$）．ANS を求

めることを問題を解くという．ルールが算術述語，関数記号を含まないホーンルールである問題を datalog 問題という．

ルールにおいて ← の左部分を頭部，右部分を本体という．たとえば，例 1 のルール (2) の $sg(X,Y)$ 部分は頭部，$B(X',X), B(Y',Y), sg(X',Y')$ 部分は本体である．

一般性を失うことなくルールの頭部およびファクトの両方に現れる述語はないとする．頭部に現れる述語を IDB 述語，ファクトに現れる述語を EDB 述語という．

次に，新しく導入する用語を説明する．

本体に IDB 述語を含まないルールを**初期化ルール**，IDB 述語を含むルールを**主ルール**と呼ぶ．例 1 のルール (1) は初期化ルール，(2) は主ルールである．

$C_1, \ldots, C_h$ の各々を定数組の集合（たとえば，$\{(a,b),(a',b')\}$ や $\{c,d\}$）とするとき，角括弧を用いた式 $p[C_1 \times \cdots \times C_h]$ を**直積型基礎アトム集合式**と呼び，**gase** と記す．この式は ga 集合 $\{p(c_{11},\ldots,c_{1m_1},\ldots,c_{h1},\ldots,c_{hm_h}) \mid (c_{i1},\ldots,c_{im_i}) \in C_i, i = 1,\ldots,h\}$ を表すのに用いる．たとえば，$p[\{(a,b),(a',b')\} \times \{c,d\}]$ は gase であり，ga 集合 $\{p(a,b,c),\ p(a,b,d), p(a',b',c), p(a',b',d)\}$ を表す．一般に，ga は gase と見なすことができるが，その逆は正しくない．

直積 $C_1 \times \cdots \times C_h$ を $\boldsymbol{C}$ と記す．また，gase $p[\boldsymbol{C}]$ が表す ga 集合を $S(p[\boldsymbol{C}])$ と記す．$p(\boldsymbol{c}) \in S(p[\boldsymbol{C}])$ のとき，**ga $p(\boldsymbol{c})$ は gase $p[\boldsymbol{C}]$ に含まれる**という．たとえば，$p(a,b,c)$ は $p[\{(a,b),(a',b')\} \times \{c,d\}]$ に含まれる．U を ga 集合とする．$S(q[\boldsymbol{C'}]) \subseteq U$ のとき，**gase $q[\boldsymbol{C'}]$ は ga 集合 U に包含される**という．たとえば，$p[\{(a,b),(a',b')\} \times \{c,d\}]$ は $\{p(a,b,c),(a,b,d),(a',b',c),(a',b',d),(a',b',e)\}$ に包含される．$S(q[\boldsymbol{C'}]) \subseteq S(q[\boldsymbol{C''}])$ のとき，**gase $q[\boldsymbol{C'}]$ は gase $q[\boldsymbol{C''}]$ に包含される**という．たとえば，$p[\{(a,b),(a',b')\} \times \{c,d\}]$ は $p[\{(a,b),(a',b')\} \times \{c,d,e\}]$ に包含される．GS を gase の集合とするとき，GS に含まれる gase が表す ga 集合の和をやはり $S(GS)$ と記す．

異なる変数 $X_1,\ldots,X_m$ を持つ述語 $p(X_1,\ldots,X_m)$ に対し，その引数集合 $\{X_1,\ldots,X_m\}$ を h 個の部分集合 $\{X_{11},\ldots,X_{1m_1}\},\ldots,\{X_{h1},\ldots,X_{hm_h}\}$ に分割することにより，引数 $X_{11},\ldots,X_{1m_1}$ を持つ述語 $p_1^*(X_{11},\ldots,X_{1m_1})$，…，引数 $X_{h1},\ldots,X_{hm_h}$ を持つ述語 $p_h^*(X_{h1},\ldots,X_{hm_h})$ を作ることを，**述語を分割する**といい，$p(X_1,\ldots,X_m) \to p_1^*(X_{11},\ldots,X_{1m_1}) \wedge \cdots \wedge p_h^*(X_{h1},\ldots,X_{hm_h})$ と記す．ここで，$\{X_1,\ldots,X_m\} = \cup_{i=1,\ldots,h}\{X_{i1},\ldots,X_{im_i}\}$，かつ，$X_{ij}$（$i = 1,\ldots,h, j = 1,\ldots,m_i$）はすべて異なる．

述語 p についての述語分割 $p(X_1,\ldots,X_m) \to p_1^*(X_{11},\ldots,X_{1m_1}) \wedge \cdots \wedge p_h^*(X_{h1},\ldots,X_{hm_h})$ が与えられているとする．述語分割の左辺のアトム $p(X_1,\ldots,X_m)$ にアトム $p(X'_1,\ldots,X'_m)$（ただし，$X'_1,\ldots,X'_m$ には同じ変数が含まれてよい）を代入することにより，右辺のアトム $p_1^*(X'_{11},\ldots,X'_{1m_1}),\ldots,p_h^*(X'_{h1},\ldots,X'_{hm_h})$ を作ることを，**アトム $p(X'_1,\ldots,X'_m)$ を述語分割に従って分割する**という．たとえば，アトム $p(X,Y,Y,Z)$ を述語分割 $p(X_1,X_2,X_3,X_4) \to p_1^*(X_1,X_2) \wedge p_2^*(X_3,X_4)$ に従って分割すると，アトム $p_1^*(X,Y)$ と $p_2^*(Y,Z)$ が作られる．

3. 直積問題

問題に現れる IDB 述語すべてについて，その述語分割が与えられているとし，それら述語分割の集合を PD と記す．ここで，述語の中には実質的には分割しないものがあってもよいが，そのような述語については分割しないことを表す述語分割 $p(X_1,\ldots,X_m) \to p^*(X_1,\ldots,X_m)$ が与えられているとする．主ルール r に対し，PD を使って，次のように無向グラフ G_r を作る．G_r の節点は，r の各 IDB アトムを PD の中の述語分割に従って分割したときに得られるアトム，および，r の EDB アトムである．G_r の枝は，節点である 2 つのアトムが同じ変数を含むとき，それら 2 つの節点の間に枝を作る．この無向グラフ G_r を r のための**引数依存グラフ**と呼ぶ．

例 2 1 章の例 1 の問題において，

$$PD = \{sg(X,Y) \to sg_1^*(X) \wedge sg_2^*(Y)\} \quad (3)$$

とする．主ルール (2) の IDB アトム $sg(X,Y)$（$sg(X',Y')$）を式 (3) に従って分割するとアトム $sg_1^*(X)$ と $sg_2^*(Y)$（$sg_1^*(X')$ と $sg_2^*(Y')$）が得られるので，主ルール (2) のための引数依存グラフは**図 1** になる． □

定義 1 述語分割集合 PD に従って，問題の各主ルール r に対して作った引数依存グラフ G_r が 2 つの条件

(1) 頭部にあるアトムの分割によりできた任意の 2 つの異なる節点の間に，それらの節点を結ぶ路が存在しない，

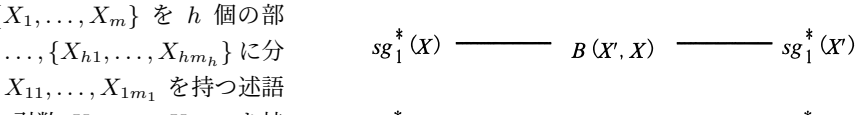

図 1 引数依存グラフ G_2
Fig. 1 Argument dependency graph G_2.

(2) 本体にある同じ IDB アトムの分割によりできた任意の 2 つの異なる節点の間に，それらの節点を結ぶ路が存在しない，

を満たすとき，問題はその PD に関して**直積分割可能である**という．□

定義 2 datalog 問題は，少なくとも 1 つの IDB 述語について述語を実質的に分割する（すなわち，2 つ以上の部分に分ける）ような直積分割可能な PD を持つとき，**直積問題**であるという．□

例 3 図 1 の引数依存グラフ G_2 は，節点 $sg_1^*(X)$ から $sg_2^*(Y)$ への路を持たないので定義 1 の条件 (1) を満たし，かつ，節点 $sg_1^*(X')$ から $sg_2^*(Y')$ への路を持たないので条件 (2) を満たす．ゆえに，例 1 の問題は式 (3) の PD に関し直積分割可能である．PD は sg を実質的に分割するので，この問題は直積問題である．□

例 4 例 1 の問題に主ルール

$$sg(X_1, X_6) \leftarrow B(X_1, X_2), sg(X_2, X_3), B(X_3, X_4), sg(X_4, X_5), B(X_5, X_6).$$

を加えても直積問題である．□

例 5 例 1 の問題に主ルール

$$sg(X, Y) \leftarrow B(X', X, Z), B(Y', Y, Z), sg(X', Y').$$

を加えると，直積問題でなくなる．□

定義 1 の条件 (1) が成立するとき，問題の意味 IMP を，生成した gase が表す ga 集合の和として表すことができ（7 章で証明する），ゆえに，gase を生成して問題を解くことができる．

定義 1 の条件 (2) が成立するとき，主ルールから gase を生成する際に，主ルールの本体に gase を直接代入することができる．すなわち，gase をそれが表す ga に展開しなくてよい．ゆえに，gase を効率的に生成することができる．

実質的な述語分割が与えられた述語については，gase の生成は ga の生成に比べ中間データ数の削減になる可能性があるが，一方，そうでない述語については，gase たとえば $p[\{(a,b),(a',b')\}]$ の生成は，そこに含まれる個々の ga $\{p(a,b), p(a',b')\}$ の生成と同じなので，中間データ数の削減にならない．定義 2 は，少なくとも 1 つの述語について，中間データ数を削減できる可能性のある問題を直積問題と定めている．ところで，通常，引数の数の大きな述語は多数の ga が生成されるが，一方，そうでない述語は少数の ga しか生成されない．ゆえに，中間データ数を大幅に減らせる可能性があるのは，すなわち，直積問題が特に意味を持つのは，引数の数の大きなすべての述語について実質的な述語分割が与えられるときである．

4. ルールからの gase の生成法

初めに，初期化ルールより gase を生成する方法を説明する．この生成では，初期化ルールから生成可能な各 ga に対し，ga の引数を述語分割に従って複数の部分に分割しながら，その ga を表す gase を生成する．たとえば，初期化ルールより ga $p(a, a')$ が生成可能で，述語分割が

$$p(X, Y) \rightarrow p_1^*(X) \wedge p_2^*(Y) \quad (4)$$

であるとき，gase $p[\{a\}\times\{a'\}]$ を生成する．

次に，主ルールの本体に gase を代入して頭部のための gase を生成する方法を例を使って説明する．主ルールを

$$p(X_1, X_6) \leftarrow B(X_1, X_2), p(X_2, X_3), B(X_3, X_4), p(X_4, X_5), B(X_5, X_6). \quad (5)$$

述語分割を式 (4)，B ファクト集合を $\{B(a,a), B(b,b), B(b,c)\}$，主ルールの本体の 1（2）番目の IDB アトム $p(X_2, X_3)$（$p(X_4, X_5)$）に代入する gase を $p[\{a,c\}\times\{b,c\}]$（$p[\{c,d\}\times\{a,b\}]$）とする．

主ルール (5) を述語分割 (4) に従って分割すると

$$p_1^*(X_1) \leftarrow B(X_1, X_2), p_1^*(X_2). \quad (6)$$

$$p_2^*(X_3) \wedge B(X_3, X_4) \wedge p_1^*(X_4), \quad (7)$$

$$p_2^*(X_6) \leftarrow p_2^*(X_5), B(X_5, X_6). \quad (8)$$

を得る．ここで，式 (6)（(8)）は，頭部アトム $p(X_1, X_6)$ を分割して得られるアトム $p_1^*(X_1)$（$p_2^*(X_6)$）を含む，引数依存グラフの中の連結成分であり，式 (7) はそうでない連結成分である．式 (6)，(8) を**部分ルール**，式 (7) を**部分条件式**と呼ぶ．部分ルール (6) について考える．1 番目の gase $p[\{a,c\}\times\{b,c\}]$ の 1 番目の定数集合 $\{a,c\}$ の中の定数を部分ルール (6) の変数 X_2 に代入し，かつ，B ファクトをアトム $B(X_1, X_2)$ に代入して述語 p_1^* の生成可能な ga をすべて生成する．$p_1^*(a)$ と $p_1^*(b)$ が生成されるので，その引数部分を取り出して定数集合 $\{a,b\}$ を作る．部分ルール (8) についても 2 番目の gase の 2 番目の定数集合を使って同様に処理し，定数集合 $\{a,b,c\}$ を作る．部分条件式 (7) について考える．1 番目の gase $p[\{a,c\}\times\{b,c\}]$ の 2 番目の定数集合 $\{b,c\}$ の中の定数を変数 X_3 に，2 番目の gase $p[\{c,d\}\times\{a,b\}]$ の 1 番目の定数集合 $\{c,d\}$ の中の定数を変数 X_4 に，B ファクトをアトム $B(X_3, X_4)$ に代入したときに部分条件式 (7) が真となるような代入があるか調べる．この場合，代入 $\{X_3 = b, X_4 = c\}$ が部分条件式 (7) を真とする．このように，各部分ルールから定数集合が生成でき，かつ，各部分条件式を真にできるときかつそのときに限

り，部分ルールから計算した定数集合 $\{a,b\},\{a,b,c\}$ を持つ gase $p[\{a,b\}\times\{a,b,c\}]$ を，主ルール (5) の頭部のための gase として，生成する．

以後，上のように，主ルールの本体に gase $p[\{a,c\}\times\{b,c\}],p[\{c,d\}\times\{a,b\}]$ を代入し，さらに，EDB アトムがある場合は代入可能なファクトをすべて代入して頭部のための gase $p[\{a,b\}\times\{a,b,c\}]$ を生成することを，簡単のため，主ルールの本体に gase 組 $(p[\{a,c\}\times\{b,c\}],p[\{c,d\}\times\{a,b\}])$ を代入して gase $p[\{a,b\}\times\{a,b,c\}]$ を生成する，という．

5. 直 積 法

PD は与えられているとする．**図 2** に**直積法**を示す．生成した gase は 2 つの gase 集合 NGS と OGS に分けて管理する．NGS はまだどの主ルールの本体にも代入したことがない gase の集合であり，OGS は，その中の gase だけからなるいかなる gase 組もすでに代入済みの gase の集合である．

(I) **step1**

4 章の方法を使って，初期化ルールから gase を生成し，それらを NGS にセットする（文 2）．

(II) **step2**

各主ルールの本体に step1 または step2 で生成した gase よりなる gase 組を代入して頭部のための gase を生成していく．この操作を $NGS=\phi$ になるまで繰り返す．以下，各文を説明する．

$NGS\neq\phi$ の場合（文 3），NGS の中の gase $p[\boldsymbol{C}]$ を 1 つ選ぶ（文 4）．

本体に述語 p を含む主ルール r を次々に選び（文 5），各 r に対し，r の本体に代入するための gase 組を次々に作る（文 6）．

作った gase 組を r の本体に代入し，4 章の方法を使って，頭部のための gase の生成を試みる（文 7）．gase が生成される場合，それを $p_0[\boldsymbol{C}_0]$ と記す．

gase $p_0[\boldsymbol{C}_0]$ が生成された場合，$p_0[\boldsymbol{C}_0]$ が NGS, OGS の中の述語 p_0 の gase が表す ga 集合の和に包含されているか検査する（文 8）．これを**被覆テスト**と呼ぶ．包含されている場合，$p_0[\boldsymbol{C}_0]$ は新しい ga を含まないので，その gase は捨てる（何もしない）．一方，包含されていない場合，$p_0[\boldsymbol{C}_0]$ は新しい ga を含むので，それを NGS に保存する（文 10）．また，保存に先立ち，効率化のため，OGS（NGS）の中の gase で $p_0[\boldsymbol{C}_0]$ に包含されるものを OGS（NGS）の中から消去する（文 9）．

被覆テストは次のように行う．例として，$S(p[\{a,b,c\}\times\{a',b',c'\}])\subseteq S(p[\{a,b\}\times\{a',b'\}])\cup S(p[\{a,b\}\times\{b',c'\}])\cup S(p[\{b,c\}\times\{a',b',c'\}])$ の判定を考える．この判定のためには，$\boldsymbol{C}_0=\{a,b,c\}\times\{a',b',c'\}$, $\boldsymbol{C}'_1=\{a,b\}\times\{a',b'\}$, $\boldsymbol{C}'_2=\{a,b\}\times\{b',c'\}$, $\boldsymbol{C}'_3=\{b,c\}\times\{a',b',c'\}$ として，$\boldsymbol{C}_0-\boldsymbol{C}'_1-\boldsymbol{C}'_2-\boldsymbol{C}'_3=\phi$ を判定すればよい．$\boldsymbol{C}_0-\boldsymbol{C}'_1$ の結果は 2 つの直積 $\boldsymbol{D}_1=\{a,b\}\times\{c'\}$ と $\boldsymbol{D}_2=\{c\}\times\{a',b',c'\}$ の和 $\boldsymbol{D}_1\cup\boldsymbol{D}_2$ として表せる．ここで，たとえば，$\boldsymbol{D}_1$ の 1（2）番目の集合は $\boldsymbol{C}_0$ の 1（2）番目の集合と $\boldsymbol{C}'_1$ の 1（2）番目の集合の共通部分（差）として計算される．得

step1: /* 初期化ルールからの gase の生成 */
1: $OGS=\phi$;
2: $NGS=$ (初期化ルールより生成される gase の集合);
step2: /* 主ルールからの gase の生成 */
3: **while**$(NGS\neq\phi)$ {
4: NGS の中の gase $p[\boldsymbol{C}]$ を 1 つ選ぶ;
5: **for** (本体に述語 p を含む各主ルール r) {
/* r の頭部述語を p_0，本体の IDB 述語を $p_1,\ldots,p_n$ と記す．ここで，$p_1,\ldots,p_n$ の少なくとも 1 つは p である．*/
6: **for** $(p_i[\boldsymbol{C}_i]\in\{p[\boldsymbol{C}]\}\cup OGS, i=1,\ldots,n,$ かつ，少なくとも 1 つの $p_i[\boldsymbol{C}_i]$ が $p[\boldsymbol{C}]$ である各 gase 組 $(p_1[\boldsymbol{C}_1],\ldots,p_n[\boldsymbol{C}_n])$) {
7: r の本体に $(p_1[\boldsymbol{C}_1],\ldots,p_n[\boldsymbol{C}_n])$ を代入して頭部述語のための gase の生成を試みる（gase が生成される場合，それを $p_0[\boldsymbol{C}_0]$ と記す）;
8: **if** ($p_0[\boldsymbol{C}_0]$ が生成され，かつ，$p_0[\boldsymbol{C}_0]$ が NGS,OGS の中の述語 p_0 の gase が表す ga 集合の和に包含されない) {
9: OGS（NGS）の中の gase で $p_0[\boldsymbol{C}_0]$ に包含されるものを OGS（NGS）の中から消去する;
10: $p_0[\boldsymbol{C}_0]$ を NGS に加える;
11: } } }
12: $p[\boldsymbol{C}]$ が消去されずに NGS の中に残っているならば，$p[\boldsymbol{C}]$ を NGS から OGS へ移す;
13: }
step3: /* 解集合 ANS の作成 */
14: 問合せを $q(\boldsymbol{X})$? と記す．OGS の中の述語 q の各 gase（$q[\boldsymbol{C}'']$ と記す）に対し，$q[\boldsymbol{C}'']$ に含まれる ga で，かつ，問合せ $q(\boldsymbol{X})$? に適合する ga の集合を生成する．そして，それら ga 集合の和を解集合 ANS とする;

図 2 直積法
Fig. 2 The Cartesian product method.

られた $\boldsymbol{D}_1$ から $\boldsymbol{C}'_2$ を引くと空になり，また，$\boldsymbol{D}_2$ から $\boldsymbol{C}'_3$ を引くと空となるので，$\boldsymbol{C}_0-\boldsymbol{C}'_1-\boldsymbol{C}'_2-\boldsymbol{C}'_3=\phi$ が分かる．効率化のため，被覆テストは，このように，gase を ga 集合に展開することなく行う．

文 7 で生成され，かつ，文 8 で新しい ga を含むことが分かった gase $p_0[\boldsymbol{C}_0]$ の中に，文 4 で選択された $p[\boldsymbol{C}]$ を包含するものがあるとき，$p[\boldsymbol{C}]$ は文 9 で NGS の中から消去される．ここで，述語名 p_0 と p は同じとする．たとえば，$p[\boldsymbol{C}]$ が $p[\{a,b\}\times\{c,d\}]$，文 8 で新しい ga を含むことが分かった $p_0[\boldsymbol{C}_0]$ が $p[\{a,b\}\times\{c,d,e\}]$ であるとき，$p[\{a,b\}\times\{c,d\}]$ は $p[\{a,b\}\times\{c,d,e\}]$ に包含されるので，$p[\{a,b\}\times\{c,d\}]$ は文 9 で NGS の中から消去される（ただし，$p[\boldsymbol{C}]$ が NGS の中から消去されても，$p[\boldsymbol{C}]$ に関する文 5 から文 11 の処理は継続される）．このように，文 4 で選択された $p[\boldsymbol{C}]$ は文 9 で消去されることもある．$p[\boldsymbol{C}]$ が消去されない場合，$p[\boldsymbol{C}]$ と OGS の中の gase のみを使ってできる gase 組はすべて主ルールに代入し終わっているので，$p[\boldsymbol{C}]$ を NGS から OGS へ移す（文 12）．

なお，文 10 では，生成した gase を NGS の先頭に加え，文 4 では，NGS の中の先頭の gase を $p[\boldsymbol{C}]$ として選ぶようにしている．これは次の理由による．上のように，gase の保存，選択を行うと，NGS の中から，新しく生成された gase が優先して選ばれることになる．ファクト密度が高い場合，このように gase を選ぶと，おおむね，NGS の中から大きな gase が優先して選ばれるようになる．その結果，実行の早い時期に大きな gase が生成されて，少数のそれら大きな gase の和として問題の意味 IMP が求められる．直積法はこのような場合に効率的になるからである．

(III) **step3**

OGS の中の gase より解集合 ANS を求める．たとえば，$OGS=\{p[\{a\}\times\{a'\}],p[\{b,c\}\times\{b',c'\}],p[\{c,d\}\times\{b',d'\}]\}$，問合せを $p(c,Y)$? とする．OGS の中の gase のうち 1 番目の定数集合が問合せ中の定数 c を含むものは 2 番目と 3 番目の gase である．1 番目の引数が問合せ中の c，2 番目の引数が 2 番目の gase または 3 番目の gase の中の 2 番目の定数集合の中の定数であるような ga $p(c,b'),p(c,c'),p(c,d')$ を生成し，それらを解集合 ANS とする（$ANS=\{p(c,b'),p(c,c'),p(c,d')\}$）．

6. 適 用 例

問題を 1 章の例 1 の問題，その述語分割集合を 3 章の例 2 の式 (3) として，直積法を適用する．問題の主ルール (2) を式 (3) に従って分割すると，2 つの部分ルール

$$sg_1^*(X)\leftarrow B(X',X),sg_1^*(X'). \tag{9}$$

$$sg_2^*(Y)\leftarrow B(Y',Y),sg_2^*(Y'). \tag{10}$$

が作られる（部分条件式は作られない）ので，主ルール (2) からの gase の生成にはこれらの部分ルールを使用する．

文 1，文 2 の実行後次式を得る．

$$NGS=\{sg[\{a\}\times\{a\}],\ldots,sg[\{f\}\times\{f\}]\}, \tag{11}$$

$$OGS=\phi. \tag{12}$$

$p[\boldsymbol{C}]$ として gase $sg[\{a\}\times\{a\}]$ を選び（文 4），主ルール (2) に対し（文 5），gase 組 ($sg[\{a\}\times\{a\}]$) を作る（文 6）．この gase 組を主ルール (2) の本体に代入して，gase の生成を試みる．部分ルール (9) から ga $sg_1^*(c)$ と $sg_1^*(d)$ が生成されるので，それらの引数の集合 $\{c,d\}$ が得られる．同様に，部分ルール (10) から集合 $\{c,d\}$ が得られる．ゆえに，これら 2 つの定数集合を持つ gase $sg[\{c,d\}\times\{c,d\}]$ を，主ルール (2) の頭部のための gase として，生成する（文 7）．gase が生成されたので被覆テストを行う（文 8）．その結果，この gase は NGS（11）および OGS（12）の中の gase が表す ga 集合の和に包含されていないことが分かる．したがって，この gase に包含される NGS の中の gase $sg[\{c\}\times\{c\}]$ と $sg[\{d\}\times\{d\}]$ を消去した（文 9）後，この gase を NGS に加える（文 10）．文 4 で選択した gase $sg[\{a\}\times\{a\}]$ を NGS から OGS へ移す（文 12）．その結果，

$$NGS=\{sg[\{c,d\}\times\{c,d\}],sg[\{b\}\times\{b\}],sg[\{e\}\times\{e\}],sg[\{f\}\times\{f\}]\},$$

$$OGS=\{sg[\{a\}\times\{a\}]\}.$$

step2 の初めに戻り，$p[\boldsymbol{C}]$ として $sg[\{c,d\}\times\{c,d\}]$ を選び，上と同様の処理を行う．$sg[\{e,f\}\times\{e,f\}]$ が生成され，文 12 の実行後次式を得る．

$$NGS=\{sg[\{e,f\}\times\{e,f\}],sg[\{b\}\times\{b\}]\},$$

$$OGS=\{sg[\{c,d\}\times\{c,d\}],sg[\{a\}\times\{a\}]\}.$$

step2 の初めに戻り，$p[\boldsymbol{C}]$ として $sg[\{e,f\}\times\{e,f\}]$ を選び，主ルール (2) からの gase の生成を試みるが，部分ルール (9) の変数 X' に $sg[\{e,f\}\times\{e,f\}]$ の中の 1 番目の定数集合 $\{e,f\}$ の中のどの定数を代入しようともアトム $B(X',X)$ に代入可能な B ファクトがないため，部分ルール (9) から定数集合を生成することができず，ゆえに，gase は生成されない．文 12 の実行後次式を得る．

$$NGS=\{sg[\{b\}\times\{b\}]\}, \tag{13}$$

$$OGS=\{sg[\{e,f\}\times\{e,f\}],sg[\{c,d\}\times\{c,d\}],sg[\{a\}\times\{a\}]\}. \tag{14}$$

step2 の初めに戻り，$p[\boldsymbol{C}]$ として $sg[\{b\}\times\{b\}]$ を選ぶ．$sg[\{c,d\}\times\{c,d\}]$ を生成するが，この gase は NGS (13) および OGS (14) の中の gase に包含されているので，捨てる．文 12 の実行後次式を得る．

$$NGS = \phi,$$
$$OGS = \{sg[\{b\}\times\{b\}], sg[\{e,f\}\times\{e,f\}], sg[\{c,d\}\times\{c,d\}], sg[\{a\}\times\{a\}]\}.$$

$NGS = \phi$ となったので，step2 を終了し，step3 を実行して，次の解集合 ANS を作る．

$$ANS = \{sg(e,e), sg(e,f)\}.$$

7. 直積法の正当性

補題 1 直積法は有限時間内に停止する．

証明 (a) 文 10 が実行されるたびに ga 集合 $S(NGS \cup OGS)$ は真に単調に増加する．一方，$S(NGS \cup OGS)$ の中の ga は，問題が datalog であるので，問題に現れる述語と定数のみからなり，ゆえに，(b) $S(NGS \cup OGS)$ は有界である．性質 (a) と (b) より文 10 は有限回しか実行されず，したがって，補題が証明される． □

補題 2 $r, p_0[\boldsymbol{C}_0], \ldots, p_n[\boldsymbol{C}_n]$ を直積法の文 7 のように定義する．また，r の本体に任意の IDB ga の組 $(p_1(\boldsymbol{c}_1), \ldots, p_n(\boldsymbol{c}_n))$（ただし，$p_i(\boldsymbol{c}_i) \in S(p_i[\boldsymbol{C}_i])$, $i = 1, \ldots, n$）を代入して生成できる ga の集合を I と記す．ここで，r に EDB アトムがある場合は，それらに，代入可能なファクトをすべて代入するものとする．gase $p_0[\boldsymbol{C}_0]$ が生成される場合，$S(p_0[\boldsymbol{C}_0]) \subseteq I$ が成立する．

証明 例を使って証明する．一般の場合も同様の手順で証明できる．主ルールを 4 章の式 (5)，述語分割を式 (4)，頭部のための gase を $p[C_{01}\times C_{02}]$，本体のための gase を $p[C_{i1}\times C_{i2}], i = 1, 2$, とする．ここで，部分ルールは式 (6)，(8)，部分条件式は式 (7) になる．任意の $p(c_1, c_6) \in S(p[C_{01}\times C_{02}])$ に対し，$p(c_1, c_6) \in I$ であることを示す．

$p(c_1, c_6) \in S(p[C_{01} \times C_{02}])$ より，(a) 部分ルール (6) に対しては，ある代入 $\{X_1 = c_1, X_2 = c_2\}$ が存在して，$c_2 \in C_{11}$，かつ，$B(c_1, c_2)$ が問題に存在し，(b) 部分条件式 (7) に対しては，ある代入 $\{X_3 = c_3, X_4 = c_4\}$ が存在して，$c_3 \in C_{12}$，かつ，$c_4 \in C_{21}$，かつ，$B(c_3, c_4)$ が存在し，(c) 部分ルール (8) に対しては，ある代入 $\{X_5 = c_5, X_6 = c_6\}$ が存在し，$c_5 \in C_{22}$，かつ，$B(c_5, c_6)$ が存在する．また，述語分割 (4) は 3 章の定義 1 の条件 (1) を満たしているので，(d) (a)，(b)，(c) における代入どうしが同じ変数を含むことはない．

(d) より，主ルール (5) への代入に，(a)，(b)，(c) で用いた代入 $\{X_1 = c_1, \ldots, X_6 = c_6\}$ を使うことができ，これにより得られる式

$$p(c_1, c_6) \leftarrow B(c_1, c_2), p(c_2, c_3), B(c_3, c_4), p(c_4, c_5), B(c_5, c_6).$$

では，条件 (a)，(b)，(c) が成立する．ゆえに，$p[C_{11}\times C_{12}]$（$p[C_{21}\times C_{22}]$）の中から $p(c_2, c_3)$（$p(c_4, c_5)$）を選び，また，ファクトは $B(c_1, c_2)$，$B(c_3, c_4)$，$B(c_5, c_6)$ を選んで主ルール (5) に代入することにより，$p(c_1, c_6)$ を生成することができる．すなわち，$p(c_1, c_6) \in I$.

□

補題 3 記号を補題 2 と同じように定義する．$I \neq \phi$ の場合，$p_0[\boldsymbol{C}_0]$ が生成されて，$I \subseteq S(p_0[\boldsymbol{C}_0])$ が成立する．

証明 補題 2 の証明の議論を逆方向に行えばよい．

□

補題 4 直積法が生成する任意の gase $p[\boldsymbol{C}]$ について，$S(p[\boldsymbol{C}]) \subseteq IMP$ が成立する．

証明 $p[\boldsymbol{C}]$ を生成する次のような木を考える．節点は gase であり，根は $p[\boldsymbol{C}]$，葉は初期化ルールから生成される gase，内部節点は主ルールから生成される gase である．枝については，主ルールの本体に gase 組 $(p_1[\boldsymbol{C}_1], \ldots, p_n[\boldsymbol{C}_n])$ を代入して gase $p_0[\boldsymbol{C}_0]$ を生成しているとき，親節点 $p_0[\boldsymbol{C}_0]$ から子節点 $p_i[\boldsymbol{C}_i], i = 1, \ldots, n$, へ枝がある．この生成木の葉から根に向かって各節点を順次調べていく．補題 2 より，各節点 $p''[\boldsymbol{C}'']$ について $S(p''[\boldsymbol{C}'']) \subseteq IMP$ を示せる．ゆえに補題が証明される． □

補題 5 直積法の while ループの終了時，任意の $p(\boldsymbol{c}) \in IMP$ について，$p(\boldsymbol{c}) \in S(OGS)$ が成立する．

証明 $p(\boldsymbol{c})$ を生成する次のような木を考える．節点は IDB ga であり，根は $p(\boldsymbol{c})$，葉は初期化ルールから生成される ga，内部節点は主ルールから生成される ga である．枝については，主ルール r の本体に IDB ga の組 $(p_1(\boldsymbol{c}_1), \ldots, p_n(\boldsymbol{c}_n))$ を代入して（ただし，EDB アトムがある場合はファクトも代入する）ga $p_0(\boldsymbol{c}_0)$ を生成しているとき，親節点 $p_0(\boldsymbol{c}_0)$ から子節点 $p_i(\boldsymbol{c}_i), i = 1, \ldots, n$, へ枝がある．

まず，(a) 各節点 $p'(\boldsymbol{c}')$ に対し，直積法が $p'(\boldsymbol{c}') \in S(p'[\boldsymbol{C}'])$，かつ，最後まで消去されない gase $p'[\boldsymbol{C}']$ を少なくとも 1 つは生成することを示す．$r, p_0(\boldsymbol{c}_0), \ldots, p_n(\boldsymbol{c}_n)$ を上のように定義し，各 $p_i(\boldsymbol{c}_i), i = 1, \ldots, n$, に対し，直積法が $p_i(\boldsymbol{c}_i) \in S(p_i[\boldsymbol{C}_i])$，かつ，最後まで消去されない gase $p_i[\boldsymbol{C}_i]$ を少なくとも 1 つは生成する，と仮定する．$p_i[\boldsymbol{C}_i], i = 1, \ldots, n$, は最後まで消去されないので，直積法は，実行のある時点で

gase 組 $(p_1[\boldsymbol{C}_1], \ldots, p_n[\boldsymbol{C}_n])$ を r に代入し，その結果，補題 3 より，$p_0(\boldsymbol{c}_0) \in S(p_0[\boldsymbol{C}_0''])$ を満たす gase $p_0[\boldsymbol{C}_0'']$ を生成する．この $p_0[\boldsymbol{C}_0'']$ は文 9 で消去されることもあるが，そのような場合には，それを包含する gase が生成されて保存されるので，ゆえに，直積法は $p_0(\boldsymbol{c}_0) \in S(p_0[\boldsymbol{C}_0])$，かつ，最後まで消去されない gase $p_0[\boldsymbol{C}_0]$ を少なくとも 1 つは生成する．$p'(\boldsymbol{c}')$ が葉である場合に (a) が成立することを示せるので，上の議論を生成木の葉から根に向かって順次繰り返すことにより，(a) が示される．

$p(\boldsymbol{c})$ について (a) が成立しているので，補題が証明される． □

定理 1 直積法は有限時間で停止し，正しく解集合を計算する．

証明 直積法は，補題 1 より有限時間で停止し，かつ，補題 4 と補題 5 より停止前には $S(OGS) = IMP$ が成立しているので，正しく解集合を計算する． □

8. 実　　験

8.1 節の 2 つの問題を解くための直積法とマジック集合法のプログラムを prolog 言語を使って作成し，それらを Pentium II 300 MHz CPU のパソコン上で実行して，直積法とマジック集合法の効率を調べる．

8.1 実験に使用する問題

問題 1 ルール集合は

$$s(X_1, X_2, X_3) \leftarrow A(X_1, X_2, X_3).$$
$$s(X_1, X_2, X_3) \leftarrow B_1(X_1', X_1), B_2(X_2', X_2), B_3(X_3', X_3), s(X_1', X_2', X_3').$$
$$s(X_1, X_2, X_3) \leftarrow C_1(X_1', X_1), C_2(X_2', X_2), C_3(X_3', X_3), s(X_1', X_2', X_3').$$

問合せは $s(1, 1, X_3)$? である．述語分割集合は $\{s(X_1, X_2, X_3) \rightarrow s_1^*(X_1) \wedge s_2^*(X_2) \wedge s_3^*(X_3)\}$ を使う． □

問題 2 ルール集合は

$$s(X_1, X_2) \leftarrow E(X_1, X_2).$$
$$s(X_1, X_6) \leftarrow F_1(X_1, X_2), s(X_2, X_3), F_2(X_3, X_4), s(X_4, X_5), F_3(X_5, X_6).$$

問合せは $s(1, X_2)$? である．述語分割集合は $\{s(X_1, X_2) \rightarrow s_1^*(X_1) \wedge s_2^*(X_2)\}$ を使う． □

問題 1 は，同世代問題の再帰ルールを複数にし，IDB 述語の引数の数を 2 から 3 に増やした問題である．問題 2 は再帰ルールを非線形にした問題である．どちらの問題も，計数法[1]等の同世代問題を対象とした方法では解くことはできない．

8.2 実験に使用するマジック集合法

マジック集合法にはいくつかの変形があるが，ここでは，問題 1 と問題 2 のどちらに対しても最も効率的である一般化補助マジック集合法[2]を使用する．一般化補助マジック集合法のルール書き換えを使って問題 1（問題 2）のルール集合をマジックルール集合に書き換え，そして，ファクト集合とマジックルール集合よりなる問題をセミナイーブ法を使ってボトムアップに処理する．

問題 1 に対するマジックルール集合を次に示す．

$$ms(1, 1) \leftarrow .$$
$$msp_1(X_1', X_2) \leftarrow ms(X_1, X_2), B_1(X_1', X_1).$$
$$ms(X_1', X_2') \leftarrow msp_1(X_1', X_2), B_2(X_2', X_2).$$
$$msp_2(X_1', X_2) \leftarrow ms(X_1, X_2), C_1(X_1', X_1).$$
$$ms(X_1', X_2') \leftarrow msp_2(X_1', X_2), C_2(X_2', X_2).$$
$$s(X_1, X_2, X_3) \leftarrow ms(X_1, X_2), A(X_1, X_2, X_3).$$
$$sp_1(X_1, X_2, X_3') \leftarrow ms(X_1, X_2), B_1(X_1', X_1), B_2(X_2', X_2), s(X_1', X_2', X_3').$$
$$s(X_1, X_2, X_3) \leftarrow sp_1(X_1, X_2, X_3'), B_3(X_3', X_3).$$
$$sp_2(X_1, X_2, X_3') \leftarrow ms(X_1, X_2), C_1(X_1', X_1), C_2(X_2', X_2), s(X_1', X_2', X_3').$$
$$s(X_1, X_2, X_3) \leftarrow sp_2(X_1, X_2, X_3'), C_3(X_3', X_3).$$

ここで，ms はマジック述語，msp_1，msp_2，sp_1，sp_2 は補助マジック述語である．

問題 2 に対するマジックルール集合を次に示す．

$$ms(1) \leftarrow .$$
$$sp_1(X_1, X_2) \leftarrow ms(X_1), F_1(X_1, X_2).$$
$$sp_2(X_1, X_3) \leftarrow sp_1(X_1, X_2), s(X_2, X_3).$$
$$sp_3(X_1, X_4) \leftarrow sp_2(X_1, X_3), F_2(X_3, X_4).$$
$$sp_4(X_1, X_5) \leftarrow sp_3(X_1, X_4), s(X_4, X_5).$$
$$ms(X_2) \leftarrow sp_1(X_1, X_2).$$
$$ms(X_4) \leftarrow sp_3(X_1, X_4).$$
$$s(X_1, X_2) \leftarrow ms(X_1), E(X_1, X_2).$$
$$s(X_1, X_6) \leftarrow sp_4(X_1, X_5), F_3(X_5, X_6).$$

ここで，ms はマジック述語，sp_1，sp_2，sp_3，sp_4 は補助マジック述語である．

8.3 問題 1 についての実験

問題例は，問題に現れる（すなわち，ファクト集合に現れる）異なる定数の数 n とファクト密度 d を用いて次のように作る．A ファクトは $A(i, i, i), i = 1, \ldots, n$，とする．$B_1$ ファクトは全部で nd 個作り，各 $B_1(i, j)$ ファクトは $\{1, \ldots, n\}$ の中からランダムに i，j を選んで作る．B_2，B_3，C_1，C_2，C_3 ファクトも B_1 ファクトと同様に作る．

n を $n = 50$ に固定し，d を $d = 0.25, 0.5, 0.75, \ldots, 4.75, 5$ と変化させ，そして，各 d に対し $k = 5$ 個の問題例（合計 $1 \times 20 \times 5$ 個）を作り，これらに直積法とマジック集合法を適用した．

実験結果を**図 3**，**図 4**，**図 5** に示す．いずれのグラ

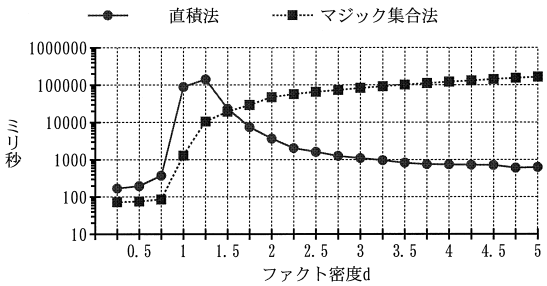

図 3 問題 1 に対する平均 CPU 時間
Fig. 3 Average CPU time for Problem 1.

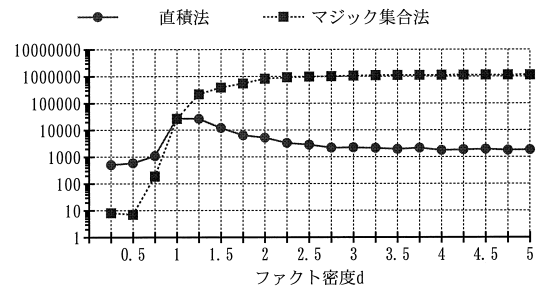

図 4 問題 1 に対する平均領域量
Fig. 4 The average size of memory for Problem 1.

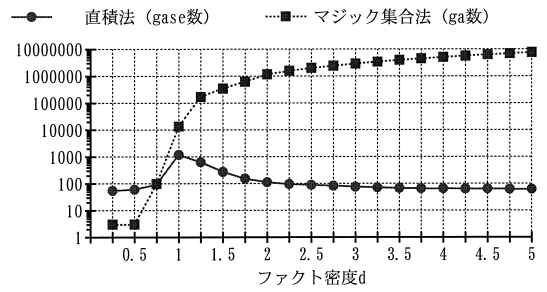

図 5 問題 1 に対する平均 gase 数および平均 ga 数
Fig. 5 The average number of gase's and that of ga's for Problem 1.

フも，横軸はファクト密度 d，縦軸は $k = 5$ 個の問題例に対する対象数値の平均値であり，対数目盛りである．

図 3 は平均 CPU 時間である．直積法の平均 CPU 時間は $d = 1.25$ 付近でピークを持っているが，d が大きいところでは小さい．d が大きい（$d \geq 1.5$）とき，直積法の平均 CPU 時間はマジック集合法のそれより小さい．

図 4 は平均領域量である．直積法の領域量は，OGS と NGS に gase を記憶するための領域量と，被覆テスト等の作業を行うための領域量の和である．1 つの gase，たとえば，$s[\{a,b,c\}\times\{b\}\times\{a,d\}]$ のための領域量は $|\{a,b,c\}|+|\{b\}|+|\{a,d\}| = 6$ である．マジック集合法の領域量は ga を記憶するための領域量であり，1 つの ga，たとえば，$s(a,b,c)$ のための領域量は 3 である．直積法の平均領域量は $d = 1$ 付近で少し大きいが，他のところでは小さい．d が大きい（$d \geq 1$）とき，直積法の平均領域量はマジック集合法のそれより小さい．

図 5 は，直積法が生成した gase の数の平均値と，マジック集合法が生成した ga の数の平均値である．ここで，重複して生成したものも数に含む．直積法の生成 gase 数は $d = 1$ 付近で少し大きいが，他のところでは小さい．d が大きい（$d \geq 1$）とき，直積法の生成 gase 数はマジック集合法の生成 ga 数より小さい．

直積法の各計算量が，上のように，d が大きいところで小さくなるのは次の理由によると考えられる．

生成 gase 数について述べる．d が大きい（$d \gg 1$）とき，直積法の実行の早い時期に，大きな gase が生成される．大きな gase がいくつか生成されると，(a) その後生成される gase の多くは，それら大きな gase に被覆されるので，NGS に保存されることなく捨てられる．また，(b) NGS に保存されていた小さな gase（たとえば初期化ルールより生成された gase）の多くは，大きな gase に包含されるため，消去される．その結果，NGS の中には大きな gase が少しだけ残るが，(c) それらの gase も，gase の生成に使用された後で，NGS から OGS へ移される．(a)，(b)，(c) の結果，NGS は早い時期に空になり，ゆえに，少数の gase だけが生成される．

領域量が，d が大きいところで小さいのは，生成 gase 数が少ないからである．

CPU 時間について述べる．直積法の主な時間量は被覆テストにかかる時間量である．被覆テスト 1 回あたりの時間量を考える．d が大きい場合，NGS と OGS には大きな gase が少しだけ保存されているので，被覆を判定するための式 $\boldsymbol{C}_0 - \boldsymbol{C}'_1 - \cdots - \boldsymbol{C}'_h$ において，各直積 $\boldsymbol{C}'_i$ は大きく，かつ，数 h は小さい．その結果，式を計算する過程で作られる直積の数が少なくなり，ゆえに，被覆テスト 1 回あたりの時間量は小さくなる．被覆テストの回数は生成 gase 数に等しく，d が大きいところでは小さい．以上，d が大きいところでは被覆テスト 1 回あたりの時間量と被覆テストの回数の両方が小さくなるため，被覆テストの全時間量も小さくなり，その結果，CPU 時間も小さくなる．

実験では n のより大きな問題例に対しても，タイムアウト時間を 360000 ミリ秒（1 時間）として，デー

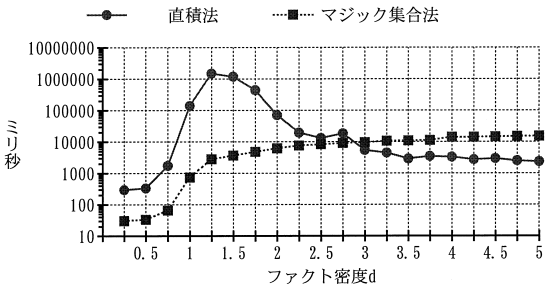

図 6 問題 2 に対する平均 CPU 時間
Fig. 6 Average CPU time for Problem 2.

タをとった．その結果，直積法は n の大きな問題例も，ファクト密度 d が大きいとき，効率的に解くことができた（マジック集合法は，d が大きいとき，上記の時間内では解けなかった）．直積法によると，たとえば $n = 500$ の場合，$d = 2$ の問題例はタイムアウトしたが，$d = 2.5$（5）の問題例は平均 CPU 時間 43980（10774）ミリ秒で解くことができた．

8.4 問題 2 についての実験

各問題例を，E，F_1，F_2，F_3 の各ファクトを実験 1 の B_1 ファクトと同じようにランダムに nd 個作ることにより，作る．$n = 100$，d と k は問題 1 の実験と同じとして，合計 $1 \times 20 \times 5$ 個の問題例を作り，これらに直積法とマジック集合法を適用した．平均 CPU 時間を**図 6** に示す．ファクト密度 d が大きい（$d \geq 3$）とき，直積法の平均 CPU 時間はマジック集合法のそれより小さい．平均領域量についても，d が大きいとき，直積法の方が小さかった．

n のより大きな問題例に対しても，タイムアウト時間を 360000 ミリ秒として，データをとった．直積法は n の大きな問題例も，ファクト密度 d が大きいとき，効率的に解くことができた（マジック集合法は，d が大きいとき，解けなかった）．直積法によると，たとえば $n = 1000$ の場合，$d = 3.5$ の問題例はタイムアウトしたが，$d = 4$（5）の問題例は平均 CPU 時間 153000（82556）ミリ秒で解くことができた．

8.5 実験のまとめ

以上，問題 1，2 のどちらにおいても，ファクトをランダムに生成した場合，ファクト密度 d が大きいとき，直積法がマジック集合法より効率的であることが分かった．

9. 直積法とマジック集合法を比較する理由

1 章に述べたように，直積法はボトムアップ計算法であり，使われている効率化手法は，従来の分類では，問題の意味の効率的計算手法（1 章の (b)）に分類される．したがって，問題を意味の小さな等価な問題に書き換える手法（1 章の (a)）であるマジックルール書き換えを直積法と組み合わせて使用することもできる．すなわち，マジックルール書き換えによりマジックルール集合を作り，マジックルール集合とファクト集合よりなる問題をセミナイーブ法の代わりに直積法を使って処理することもできる．

しかし，マジックルール書き換えとセミナイーブ法が独立した効率化手法であって，それらの組合せがセミナイーブ法単独よりも効率的になるのに対し，一方，マジックルール書き換えと直積法の組合せは直積法単独より必ずしも効率的にならない．すなわち，マジックルール集合とファクト集合よりなる問題を直積法を使って処理すると，元の問題をそのまま直積法を使って処理するよりも，かえって非効率になる場合がある．

たとえば，8 章の問題 1 において，ルール

$$s(X_1, X_2, X_3) \leftarrow B_1(X_1', X_1), B_2(X_2', X_2), B_3(X_3', X_3), s(X_1', X_2', X_3'). \quad (15)$$

をそのまま直積法を使って処理する場合と，そのマジックルール

$$s(X_1, X_2, X_3) \leftarrow ms(X_1, X_2), B_1(X_1', X_1), B_2(X_2', X_2), B_3(X_3', X_3), s(X_1', X_2', X_3'). \quad (16)$$

を直積法を使って処理する場合を考える．ここで，マジック集合法のマジックルール（8.2 節）では，効率化のために補助マジック述語も使用するが，直積法では，補助マジック述語を導入しても効率化されないので，ルール (16) では補助マジック述語は使用していない．ルール (15) の本体に gase $s[\boldsymbol{C}']$ を代入すると，頭部のための gase $s[\boldsymbol{C}]$ が 1 つ生成される．一方，ルール (16) の本体に gase $s[\boldsymbol{C}']$ を代入すると，通常，複数の ms gase があるため，$s[\boldsymbol{C}]$ の代わりに複数の小さな s gase が生成される．このことが直積法の効率を低下させる．マジックルール書き換え（マジック述語の使用）は，問題の意味を小さくするという意味では直積法を使う場合にも有効であるが，しかし，その反面，小さな gase を多数生成させるという意味で直積法の効率を低下させる．そして，その総合的な結果として，マジックルール書き換えと直積法の組合せが直積法単独より非効率になることがある．実際，我々が実験した限りでは，そのようなことが多くあった．

直積法で使われている効率化手法は問題の意味の効率的計算であるので，1 章に述べたように，直積法とマジック集合法の効率を比較するより，直積法とセミナイーブ法の効率を比較する方が自然なようにも思われる．しかし，仮に，直積法とセミナイーブ法を比較

して，直積法がセミナイーブ法より効率的であることがいえたとしても，直積法が，セミナイーブ法より効率的であるマジック集合法（マジックルール書き換えとセミナイーブ法の組合せ）より効率的であるとはいえないし，また，上に述べたように，マジックルール書き換えと直積法の組合せは直積法単独より非効率になることがあるので，マジックルール書き換えと直積法の組合せがマジック集合法より効率的になるともいえない．そのため，本稿では，直積法とセミナイーブ法の効率ではなく，直積法とマジック集合法の効率を比較した．

10. むすび

datalog 問題クラスの新しい部分クラスである直積問題クラスを定義し，この問題クラスを効率的に解くための方法として直積法を提案した．直積法とマジック集合法の効率を比較するため，同世代問題の再帰ルールを複数にした問題，および，非線形にした問題を考え，これらの問題を使って計算機実験を行った．実験の結果，ファクトをランダムに生成した場合，どちらの問題においても，ファクト密度が大きいとき，直積法がマジック集合法より効率的であることが分かった．

謝辞 注意深く読んでいただき，多くの貴重なご助言をいただきました査読者の方々に深謝いたします．

参考文献

1) Bancilhon, F., Maier, D., Sagiv, Y. and Ullman, J.: Magic sets and other strange ways to implement logic programs, *Proc. 5th ACM SIGMOD-SIGACT Symp. on PODS*, pp.1–15 (1986).
2) Beeri, C. and Ramakrishnan, R.: On the power of magic, *Proc. 6th ACM SIGACT-SIGMOD-SIGART Symp. on PODS*, pp.269–283 (1987).
3) 宮崎収兄，世木博久：演繹データベースの問合せ処理，情報処理，Vol.31，No.2，pp.216–224 (1990).
4) Naughton, J.F., Ramakrishnan, R., Sagiv, Y. and Ullman, J.D.: Efficient evaluation of right-, left-, and multi-linear rules, *ACM-SIGMOD Conf.*, pp.235–242 (1989).
5) 西尾章治郎，楠見雄規：演繹データベースにおける再帰的な問合せ評価法，情報処理，Vol.29, No.3, pp.240–255 (1988).
6) Ramakrishnan, R.: Magic templates: A spell-binding approach to logic programs, *Proc. 5th Int. Conf. and Symp. on Logic Programming*, pp.140–159 (1988).
7) Ullman, J.D.: *Principles of database and knowledge-base systems, Vol.I, II*, Computer Science Press (1988,1989).

(平成 11 年 11 月 15 日受付)
(平成 12 年 11 月 2 日採録)

鈴木 晋（正会員）

昭和 55 年京都大学工学部卒業．昭和 57 年同大学院修士課程修了．同年日本電気（株）入社．昭和 63 年愛知工業大学工学部情報通信工学科講師，平成 11 年同助教授，現在に至る．アルゴリズム，計算の複雑さ，演繹データベースの研究に従事．工学博士．電子情報通信学会会員．

茨木 俊秀（正会員）

昭和 38 年京都大学工学部卒業．昭和 40 年同大学院修士課程修了．昭和 44 年京都大学工学部数理工学科助手，昭和 48 年同助教授，昭和 58 年豊橋技術科学大学教授，昭和 60 年京都大学大学院情報学研究科数理工学専攻教授，現在に至る．この間，Illinois 大学，Waterloo 大学，Simon Fraser 大学，Rutgers 大学等客員．工学博士．組合せ最適化，アルゴリズム，計算の複雑さ等の研究に従事．著書「C によるアルゴリズムとデータ構造」（昭晃堂），「最適化プログラミング」（岩波），「最適化の手法」（共立），「Enumerative approaches to combinatorial optimization」（Balzer），「Resource allocation problems: algorithmic approaches」（MIT Press）等．IEEE，ACM，Mathematical Programming Society，日本 OR 学会，電子情報通信学会，システム情報制御学会，人工知能学会，応用数理学会，日本数理科学協会各会員．