愛知工業大学研究報告
第28号B 平成5年

多変数同世代問題に対する問い合わせ評価法

Query Evaluation of The Same Generation Problem with Many Variables

鈴木 晋†, 茨木 俊秀††, 岸 政七†
Susumu Suzuki, Toshihide Ibaraki, Masahichi Kishi

ABSTRACT *We consider a generalization of the same generation problem, which is well known in the deductive database theory, in the sense that the arity of a recursive predicate is generally m and that each extensional database may be cyclic. For the case of $m \geq 3$, among developed methods, the magic set method or the HaNa method appears most efficient in worst-case time complexity. We propose here a modification of the HaNa method and analyze its worst-case performance. When $m \geq 3$ and some usual conditions are satisfied, the modified HaNa method is superior to the magic set method and to the HaNa method.*

1 まえがき

2つの論理式

$$s(x_1,\ldots,x_m) \ :- \ r_0(x_1,\ldots,x_m). \quad (1)$$

$$s(x_1,\ldots,x_m) \ :- \ r_1(x'_1,x_1), r_2(x'_2,x_2),\ldots, r_m(x'_m,x_m), s(x'_1,\ldots,x'_m). \quad (2)$$

に対する問い合わせ

$$s(a_1,\ldots,a_h,x_{h+1},\ldots,x_m)? \quad (3)$$

に答えることを考える．ここで，$x_1,\ldots,x_m$, $x'_1,\ldots,x'_m$は変数，$a_1,\ldots,a_h$は定数，s は IDB 述語 (intensional database predicate)，r_0および $r_1,\ldots,r_m$は EDB 述語 (extensional database predicate) である．各 EDB 述語 r_iは事実として外延データベース (extensional database) R_iに蓄えられている ($R_i = \{r_i(b,c), r_i(d,e),\ldots\}$)．述語 $s(x_1,\ldots,x_m)$ は式 (1) により初期化され，式 (2) により再帰的に定義される．$s(x_1,\ldots,x_m)$のうち，条件 $x_1 = a_1,\ldots,x_h = a_h$を満たす $(x_{h+1},\ldots,x_m)$ が問い合わせ (3) の解であり，すべての解の集合 $ANS = \{(x_{h+1},\ldots,x_m)\}$ を求めることが要求される．本問題は，$m = 2$ の場合，有名な同世代問題になり，多くの研究が行われている（この場合，式 (3) の h は $h = 1$ とする）．本論文では一般の $m(\geq 2)$ の場合を考える．また，r_iすなわち $R_i(i = 1,\ldots,m)$ は，変数 x_iあるいは x'_iのとり得る値を節点で表し，各タプル (x'_i, x_i) を節点 x'_iから節点 x_iへの枝とみなすことにより，有向グラフ $G_i = (V_i, E_i)$(V_i：節点集合，E_i：枝集合) として解釈できるが，本論文では，各 G_iが閉路を含むことを許す．

$m = 2$, かつ，G_1および G_2の両方が閉路を含む場合，HaNa 法 (HaNa method) は最悪時間量 $O(ne)$ [1, 2]，マジック集合法 (magic set method) [3] は $O(e^2)$ [4] で問い合わせを処理する．ここで，n は G_1,G_2それぞれの節点数，e は枝数 (すなわち R_iのタプル数) である (簡単のため，各 G_iの大きさは同じとしている)．その他多くの方法が提案されているが [5]-[11]，最悪時間量において HaNa 法が最も優れる．

一般の m の場合，問題自身は文献 [11, 12] にも言及されているが，問い合わせ評価法の研究はあまり行われていない．一般の m の問題は，一般の m 用に提案された方法 [4, 13] を使って解くことができるが，それ以外に，$m = 2$ に対して提案された方法 [1]-[3],[5]-[11] を一般の m に適応できるように素直に変形することにより解くこともできる．前者の場合，$G_1,\ldots,G_m$の全てが閉路を含む場合，マジック集合法は唯一適用可能と思われる方法であり，その最悪時間量は $O(e^m)$ である [4]．(また，その最悪領域量が $O(n^m)$ であることは容易に導ける．) 後者の場合，述語 $s(x_1,\ldots,x_h,x_{h+1},\ldots,x_m)$ の第 1 引き数から第 h 引き数までを一つの変数 $z_1(= (x_1,\ldots,x_h))$，残りの引き数をもう一つの変数 $z_2(= (x_{h+1},\ldots,x_m))$ として，述語 s を 2 引

† 愛知工業大学 情報通信工学科 (豊田市)
†† 京都大学 工学部 数理工学科 (京都市)

き数述語 $s(z_1, z_2)$ と見なし，また，EDB 述語の連言 $r_1 \wedge \cdots \wedge r_h$ を変数 z_1 に関する EDB 述語 $r_1'(z_1', z_1)$，$r_{h+1} \wedge \cdots \wedge r_m$ を変数 z_2 に関する EDB 述語 $r_2'(z_2', z_2)$ と見なすことにより，$m = 2$ に対して提案された方法を一般の m に適用できるように素直に変形することができる．一般の m に素直に変形された方法 [1]-[3],[5]-[11] の中で，HaNa 法は最悪時間量において最も優れていると思われ，その最悪時間量は $O(n^h(e^h + t_0) + (n^h + n^{m-h})(t_0 + e^{m-h})) = O(n^h(e^h + t_0) + n^{m-h}(t_0 + e^{m-h}))$ である [1, 2]．ここで，n^h, e^h は述語 $r_1 \wedge \cdots \wedge r_h$ が表す積グラフ $G_1 \times \cdots \times G_h$ の節点数，枝数であり，n^{m-h}, e^{m-h} は述語 $r_{h+1} \wedge \cdots \wedge r_m$ が表す積グラフ $G_{h+1} \times \cdots \times G_m$ の節点数，枝数である．また，t_0 は外延データベース R_0 のタプル数 ($t_0 = |R_0| = |\{(x_1, \ldots, x_m) \mid r_0(x_1, \ldots, x_m)\}|$) である．$n \leq e \leq n^2, t_0 \leq n^m$ と考慮すると，従来の方法の中で，$m \geq 3$ かつ h が $m/2$ と離れている場合はマジック集合法が，$m \geq 3$ かつ h が $m/2$ に近い場合は HaNa 法が最も優れていると思われる．

本論文では，R.W.Haddad および J.F.Naughton により $m = 2$ の場合に対して提案された HaNa 法 [1, 2] を $m \geq 3$ の場合に効率的になるように変形した拡張 HaNa 法を提案し，$m \geq 3$ のとき，その最悪時間量が

$$O(t_0(n^m \log n + n^{m-h}|ANS| \log n))$$

最悪領域量が

$$O(mne + |ANS|)$$

であることを解析する．ここで，$|ANS|$ は解集合のサイズ ($|ANS| \leq n^{m-h}$) である．$m \geq 3$ かつ (よく議論される応用例に見られるように) t_0 および $|ANS|$ があまり大きくない場合，拡張 HaNa 法が，マジック集合法および HaNa 法より最悪時間量において優れていることを示す．また，比較的小さな領域量 $O(mne)$ を無視すれば，拡張 HaNa 法の領域量がほぼ最適であることを示す．

本論文では，第 2 節で問題の応用例を示す．第 3 節で HaNa 法を紹介する．第 4 節で拡張 HaNa 法を与え，第 5 節でその最悪計算量を解析し，第 6 節で解析結果について考察する．最後に第 7 節で結論を述べる．

2 応用例

$$3\_sg(x_1, x_2, x_3) : -3\_eq(x_1, x_2, x_3). \quad (4)$$

$$3\_sg(x_1, x_2, x_3) : -par(x_1', x_1), par(x_2', x_2), par(x_3', x_3), 3\_sg(x_1', x_2', x_3'). \quad (5)$$

$$3\_sg(a_1, a_2, x_3)? \quad (6)$$

を考える．ここで，3_sg は IDB 述語，$par, 3\_eq$ は EDB 述語である．EDB 述語 $3\_eq(x_1, x_2, x_3)$ は，ここでは $x_1 = x_2 = x_3$ を表すものとする．$par(x_i', x_i)$ は x_i' が x_i の親であることを示し，外延データベースとして与えられる．x の子供を $x^i (i = 1, \ldots, d)$，x^i の子供を $x^{ij} (j = 1, \ldots, d_i)$ と記す．式 (4) より $3\_sg(x, x, x)$ が成立し，$3\_sg(x, x, x)$ に式 (5) を適用すると，$3\_sg(x^{i_1}, x^{i_2}, x^{i_3}) (i_1, i_2, i_3 = 1, \ldots, d)$ を得る．$x^{i_1}, x^{i_2}, x^{i_3}$ は x より 1 世代下った子孫である．さらに，$3\_sg(x^{i_1}, x^{i_2}, x^{i_3})$ に式 (5) を適用すると，$3\_sg(x^{i_1 j_1}, x^{i_2 j_2}, x^{i_3 j_3}) (j_1 = 1, \ldots, d_{i_1}, j_2 = 1, \ldots, d_{i_2}, j_3 = 1, \ldots, d_{i_3})$ を得る．$x^{i_1 j_1}, x^{i_2 j_2}, x^{i_3 j_3}$ は x より 2 世代下った子孫である．上記の処理を繰り返すことにより，x を共通先祖とする全ての $3\_sg(x_1, x_2, x_3)$ を求めることができる．すなわち，$3\_sg(x_1, x_2, x_3)$ は，x_1, x_2, x_3 の 3 人全てがある共通先祖 x から同世代数だけ下った子孫であること，すなわち，従兄弟であることを示す．ところで，$x_i'' (i = 1, 2, 3)$ を x のある子孫とすると，一般に x_i'' と x の世代差は一意的でない．その結果，例えば，x_1'' と x の世代差が k_1 および k_2，x_2'' と x の世代差が k_1，x_3'' と x の世代差が k_2 とすると，x_1'' と x_2'' の 2 人，および，x_1'' と x_3'' の 2 人は，各々，x を共通先祖とする従兄弟であるが，x_1'', x_2'', x_3'' の 3 人は従兄弟ではない．すなわち，$3\_sg(x_1'', x_2'', x_3'')$ は成立しない．この意味で，2 人の従兄弟関係のみを求める $m = 2$ の場合の同世代問題とは異なる．問い合わせ $3\_sg(a_1, a_2, x_3)$?は，a_1, a_2 の従兄弟を全て見つけることを要求する．

3 HaNa 法の紹介

本節では，R.W.Haddad および J.F.Naughton により提案された HaNa 法 [1, 2] を紹介する．

3.1 HaNa 法の概要

$m = 2$ の場合の同世代問題を考える．問い合わせは $s(a_1, x_2)$?である．グラフ $G_i (i = 1, 2)$ において，節点 y_i から節点 x_i への 3 つの距離集合

$D_i(y_i, x_i) = \{ l \mid$ 節点 y_i から x_i へ長さ l の路が存在する (閉路を含んでも含まなくてもよい) $\}$

$CD_i(y_i, x_i) = \{ l \mid G_i$ の少なくとも一つ閉路に出会う，節点 y_i から x_i への長さ l の路が存在する $\}$

$F_i(y_i, x_i) = \{ l \mid$ 節点 y_i から x_i へ $l < n$ なる長さ l の路が存在する $\} = D_i(y_i, x_i) \cap \{0, 1, 2, \ldots, n-1\}$

を定義する．ここで，$CD_i(y_i, x_i)$ の定義における，閉路に出会う路とは，例えば，図 1 のグラフ G の路 $a^0a^5a^6a^5a^6a^7$ のように閉路 $a^5a^6a^5$ を含む路，または，単純路 $a^0a^5a^6a^7$ のように閉路 $a^5a^6a^5$ (また

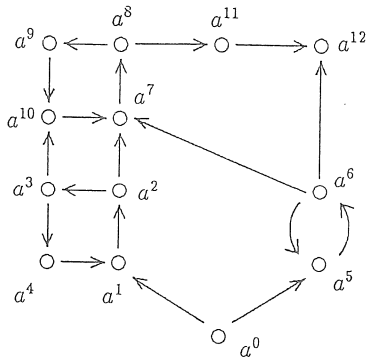

図1 外延データベースを表すグラフ G
Fig.1 A graph G representing an extensional database.

は閉路 $a^7a^8a^9a^{10}a^7$) に接している路を意味する. D_i, CD_i, F_i を使って,

$$\begin{aligned} ANS &= \{x_2 \mid \exists(y_1,y_2) \in R_0 \\ &\quad D_1(y_1,a_1) \cap D_2(y_2,x_2) \neq \phi\} \\ CANS &= \{x_2 \mid \exists(y_1,y_2) \in R_0 \\ &\quad CD_1(y_1,a_1) \cap CD_2(y_2,x_2) \neq \phi\} \\ FANS &= \{x_2 \mid \exists(y_1,y_2) \in R_0 \\ &\quad F_1(y_1,a_1) \cap F_2(y_2,x_2) \neq \phi\} \end{aligned}$$

と定義する. ANS は問い合わせ $s(a_1,x_2)$? の解集合である. CD_iは閉路を含む全ての路の長さを含むので, $CANS$ は, G_1上の閉路を含む路と G_2 上の閉路を含む路により生成される全ての解を包含する. $FANS$ は長さ $n-1$ 以下の路により生成される解の集合であり, G_1, G_2の路の少なくとも一方が閉路を含まないような全ての解を包含する. したがって,

$$ANS = CANS \cup FANS.$$

$FANS$の計算は, 有限集合 F_iを扱えばよいので容易である. 数え上げ法 [3] により $O(ne)$ 時間で計算することできる [4]. 一方, $CANS$の計算は, 無限集合 CD_iを処理しなければならず問題がある. そこで, HaNa 法 [1, 2] では, $CD_i(y_i,x_i)$ より計算の容易な簡略化距離集合 $CS_i(y_i,x_i)$ を利用して $CANS$を計算する方法を提案している.

ある非負整数 c および自然数の有限集合 Pに対し, 集合 CDが

$$CD = \{c + \sum_{p\in P} \lambda_p p \mid \lambda_p : \text{非負整数}\}$$

と表されるとき, 集合 CDは線形 (linear) であると言う. グラフ G 上の距離集合 $CD(y,x)$ は有限個の線形集合の和として表すことができる [1, 2]. 例えば, 図1のグラフ Gにおいて, 単純路 $a^0a^5a^6a^7$ に単純閉路 $a^5a^6a^5$と単純閉路 $a^7a^8a^9a^{10}a^7$を任意個加えてできる a^0から a^7への路が存在するので, $CD(a^0,a^7)$ は, 集合 $\{3+2\lambda_2+4\lambda_4 \mid \lambda_2,\lambda_4 :$ 非負整数 $\}$ を含む. 同様に, 単純路 $a^0a^1a^2a^7$, $a^0a^1a^2a^3a^{10}a^7$とそれらの各々に出会う閉路を考えることにより,

$$\begin{aligned} CD(a^0,a^7) &= \{3+2\lambda_2+4\lambda_4 \mid \lambda_2,\lambda_4 : \text{非負整数}\} \\ &\quad \cup \{3+4\lambda_4 \mid \lambda_4 : \text{非負整数}\} \\ &\quad \cup \{5+4\lambda_4 \mid \lambda_4 : \text{非負整数}\} \end{aligned}$$

と表すことができる.

c, p'を非負整数, Pを自然数の有限集合とする. 記号 $L(c;p')$ および $L(c;P)$ を

$$\begin{aligned} L(c;p') &= \{c+\lambda p' \mid \lambda : \text{非負整数}\} \\ L(c;P) &= \{c+\sum_{p\in P}\lambda_p p \mid \lambda_p : \text{非負整数}\} \end{aligned} \quad (7)$$

と定義する. (文献 [1, 2] では, Cを非負整数の有限集合とし, $L(C;p')$ および $L(C;P)$ が

$$\begin{aligned} L(C;p') &= \{c+\lambda p' \mid c\in C, \lambda : \text{非負整数}\} \\ L(C;P) &= \{c+\sum_{p\in P}\lambda_p p \mid c\in C, \lambda_p : \text{非負整数}\} \end{aligned}$$

として定義されている. 本論文では, 例えば, $L(C;p')$ は

$$L(C;p') = \bigcup_{c\in C} L(c;p')$$

として扱うことができるので, 説明を簡単にするため, L の第1引き数は集合 Cではなく非負整数 c を表すものとする.) また, Pの全ての要素の最大公約数 gを $g=\gcd(P)$ と記す. $P_i(i=1,\ldots,d)$ を自然数のある有限集合とすると, 先に述べたように, 集合 $CD(y,x)$ は

$$CD(y,x) = \bigcup_{i=1}^{d} L(c_i;P_i) \quad (8)$$

と表されるので, これより, 新しい集合

$$\begin{aligned} B(CD(y,x)) &= B(\bigcup_{i=1}^{d} L(c_i;P_i)) \\ &= \bigcup_{i=1}^{d} L(c_i \bmod g_i; g_i) \end{aligned} \quad (9)$$

を定義する. ただし, $g_i=\gcd(P_i)(i=1,\ldots,d)$. この集合 $B(CD(y,x))$ を $CD(y,x)$ の簡略化距離集合と呼び, $CS(y,x)$ と記す. 上記の例では,

$$\begin{aligned} CS(a^0,a^7) &= B(CD(a^0,a^7)) \\ &= L(1;2) \cup L(3;4) \cup L(1;4) \\ &= \{1+2\lambda \mid \lambda : \text{非負整数}\} \\ &\quad \cup \{3+4\lambda \mid \lambda : \text{非負整数}\} \\ &\quad \cup \{1+4\lambda \mid \lambda : \text{非負整数}\} \end{aligned} \quad (10)$$

$CD(y,x)$ および $CS(y,x)$ の定義より,

$$CD(y,x) \subset CS(y,x),$$
$$|CS(y,x) - CD(y,x)| < \infty$$

を示すことができる [1, 2]. したがって,

$$\begin{aligned} & CD_1(y_1,x_1)\cap CD_2(y_2,x_2)\neq\phi \\ \Leftrightarrow\ & |CD_1(y_1,x_1)\cap CD_2(y_2,x_2)|=\infty \\ \Leftrightarrow\ & |CS_1(y_1,x_1)\cap CS_2(y_2,x_2)|=\infty \\ & (\Leftarrow: |CS_i(y_i,x_i)-CD_i(y_i,x_i)|<\infty \text{ より}. \\ & \Rightarrow: CD_i(y_i,x_i)\subset CS_i(y_i,x_i) \text{ より}) \\ \Leftrightarrow\ & CS_1(y_1,x_1)\cap CS_2(y_2,x_2)\neq\phi \end{aligned}$$

が成立するので, $CANS$を簡略化距離集合 CS_iを使って

$$\begin{aligned} CANS &= \{x_2 \mid \exists(y_1,y_2)\in R_0 \\ & \qquad CS_1(y_1,a_1)\cap CS_2(y_2,x_2)\neq\phi\} \end{aligned}$$

によって計算することができる.

3.2 簡略化距離集合の性質

簡略化距離集合 $CS(y,x)(=\bigcup_{i=1}^{d} L(c_i;p_i))$ は, 具体的には, それを表す集合 $L(c_i,p_i)$ の組を計算することにより求められる. 同じ $CS(y,x)$ を表す $L(c_i,p_i)$ の組は複数存在し得るが, HaNa 法はその一つを以下のように効率的に定める. (この性質は拡張 HaNa 法の効率化に利用される.)

ケース 1) 節点 x が G のある閉路に含まれる場合:
式 (8),(9) の記号を用いる. $g_1,\ldots,g_d$の最小公倍数 $p(=lcm(g_1,\ldots,g_d))$ および

$$C = \{c \bmod p \mid c\in\bigcup_{i=1}^{d} L(c_i \bmod g_i;g_i)\}$$

を計算する. このとき, $CS(y,x)$ は

$$CS(y,x) = \bigcup_{c\in C} L(c;p)$$

と表される. $CS(y,x)$ のこの表現$\bigcup_{c\in C} L(c;p)$ を $M(CS(y,x))$ と記す. 例えば, 式 (10) は, この表現により

$$CS(a^0,a^7)=L(1;4)\cup L(3;4)$$

と簡略化される. (文献 [1, 2] では, G の極大強連結成分$\tilde{G}$に含まれる全ての閉路の長さの最大公約数を$\tilde{G}$の周期と定義しているが,) 本論文では, この $p(=lcm(g_1,\ldots,g_d))$ を $CS(y,x)$ の周期と呼ぶ. p は以下の性質を持つ.

(i) x を含む G の極大強連結成分を$\tilde{G}=(\tilde{V},\tilde{E})$($\tilde{V}$: 節点集合,$\tilde{E}$: 枝集合) とする. 節点 x' が$\tilde{G}$に含まれるならば, $(CD(y,x)=\bigcup_i L(c_i;P_i)$ の P_iと $CD(y,x')=\bigcup_i L(c'_i;P_i)$ の P_iは全て等しいので,) $CS(y,x)$ の周期 p と $CS(y,x')$ の周期 p'は等しい.

(ii) $\tilde{G}$に含まれる全ての単純閉路の長さを $\dot{p}_1,\ldots,\dot{p}_f$とする. $CD(y,x)=\bigcup_{i=1}^{d} L(c_i;P_i)$ において, $\dot{p}_1,\ldots,\dot{p}_f$は全ての $P_i(i=1,\ldots,d)$ に含まれるので,

$$\begin{aligned} p &= lcm(\gcd(P_1),\ldots,\gcd(P_d)) \\ &\leq \gcd(\{\dot{p}_1,\ldots,\dot{p}_f\})\leq|\tilde{V}|. \end{aligned} \qquad (11)$$

ケース 2) 節点 x が閉路に含まれない場合:
節点 yから x へのある路 $y\cdots z_i v_{i_1} v_{i_2}\cdots v_{i_c}x$ において, 節点 z_iはある閉路に含まれ, 節点 $v_{i_1},v_{i_2},\ldots,v_{i_c}$は全てどの閉路にも含まれないとする. この条件を満足する全ての z_iを $z_1,\ldots,z_e$ とし, 各 z_iの簡略化距離集合を (ケース 1 の節点であるので)

$$CS(y,z_i)=\bigcup_{c\in C_i} L(c;p_i) \qquad i=1,\ldots,e$$

とする. HaNa 法は

$$\begin{aligned} & C'_i = \\ & \quad \{(c+\text{路 } z_i v_{i_1}\cdots v_{i_c}x \text{ の長さ}) \bmod p_i \mid c\in C_i\} \end{aligned}$$

を計算する. このとき, $CS(y,x)$ は

$$CS(y,x) = \bigcup_{i=1}^{e}\bigcup_{c\in C'_i} L(c;p_i)$$

と表される. 例えば, 図 1 の例では, $y=a^0, x=a^{12}$に対し, $z_1=a^6, z_2=a^8$となり, $CS(a^0,a^6)=L(0;2)$, $CS(a^0,a^8)=L(0;4)\cup L(2;4)$ より

$$CS(a^0,a^{12})=L(1;2)\cup L(2;4)\cup L(0;4).$$

図 1 の G において, $y=a^0$から全ての x に対する $CS(a^0,x)$ を求めると,

$$\begin{aligned} CS(a^0,a^0) &= \phi, \\ CS(a^0,a^1) &= L(1;4), \\ CS(a^0,a^2) &= L(2;4), \\ CS(a^0,a^3) &= L(3;4), \\ CS(a^0,a^4) &= L(0;4), \\ CS(a^0,a^5) &= L(1;2), \\ CS(a^0,a^6) &= L(0;2), \\ CS(a^0,a^7) &= L(1;4)\cup L(3;4), \\ CS(a^0,a^8) &= L(0;4)\cup L(2;4), \\ CS(a^0,a^9) &= L(1;4)\cup L(3;4), \\ CS(a^0,a^{10}) &= L(0;4)\cup L(2;4), \\ CS(a^0,a^{11}) &= L(1;4)\cup L(3;4), \\ CS(a^0,a^{12}) &= L(1;2)\cup L(0;4)\cup L(2;4) \end{aligned}$$

となる. HaNa 法は, グラフ G 上の一つの節点 y と yから到達可能な全ての節点 x との間の簡略化距離集合 $CS(y,x)$ を最悪時間量 $O(ne)$ で計算す

る [1, 2].

4 拡張 HaNa 法

本節では，一般の $m(\geq 2)$ の場合の同世代問題 (1節) の解法として，拡張 HaNa 法を与える．HaNa 法は 1 節で述べたやり方により一般の m に適用することができるが，h が $m/2$ と離れている場合，マジック集合法より劣る．そこで，4.1 節では，1 節とは異なるやり方で HaNa 法を一般の m に素直に拡張する．次に，その方法の一般の m には適用できない部分，および，非効率である部分を各々4.2 節，4.3 節で変更し，その結果得られるアルゴリズム (拡張 HaNa 法) を 4.4 節で与える．例題を 4.5 節で扱う．

4.1 HaNa 法との関係

$D_i(y_i, x_i), CD_i(y_i, x_i), F_i(y_i, x_i), CS_i(y_i, x_i)(i = 1, \ldots, m)$ を 3 節と同様に定義する．また，

$$
\begin{aligned}
ANS = \{&(x_{h+1}, \ldots, x_m) \mid \\
&\exists (y_1, \ldots, y_m) \in R_0 \\
&(\bigcap_{i=1}^{h} D_i(y_i, a_i)) \cap (\bigcap_{i=h+1}^{m} D_i(y_i, x_i)) \neq \phi\} \\
CANS = \{&(x_{h+1}, \ldots, x_m) \mid \\
&\exists (y_1, \ldots, y_m) \in R_0 \\
&(\bigcap_{i=1}^{h} CD_i(y_i, a_i)) \cap (\bigcap_{i=h+1}^{m} CD_i(y_i, x_i)) \neq \phi\} \\
FANS = \{&(x_{h+1}, \ldots, x_m) \mid \\
&\exists (y_1, \ldots, y_m) \in R_0 \\
&(\bigcap_{i=1}^{h} F_i(y_i, a_i)) \cap (\bigcap_{i=h+1}^{m} F_i(y_i, x_i)) \neq \phi\}
\end{aligned}
$$

と定義する．ANS は解集合であり，3 節と同じ議論により，

$$
\begin{aligned}
&ANS = CANS \cup FANS, \\
CANS = \{&(x_{h+1}, \ldots, x_m) \mid \\
&\exists (y_1, \ldots, y_m) \in R_0 \\
&(\bigcap_{i=1}^{h} CS_i(y_i, a_i)) \cap (\bigcap_{i=h+1}^{m} CS_i(y_i, x_i)) \neq \phi\}
\end{aligned}
$$

が成立する．

この結果にしたがって，1 節とは異なるやり方で一般の m に素直に拡張した HaNa 法を図 2 に示す．ただし，

$$
\begin{aligned}
\prod_{i=h+1}^{m} V_i = \{&(x_{h+1}, \ldots, x_m) \mid \\
&x_{h+1} \in V_{h+1}, \ldots, x_m \in V_m\},
\end{aligned}
$$

```
begin
 step1: FANSを計算する;
 step2: { CANSの計算 }
 CANS := φ;
 for ∀(y_1,...,y_m) ∈ R_0 do { ループ 1 の初め }
 begin
  step2.1: HaNa 法のアルゴリズムにより,
  簡略化距離集合 CS_i(y_i, x_i)(i = 1,...,m, x_i ∈ V_i)
  を計算する;
  step2.2: {(y_1,...,y_m) に対する CANSの計算 }
  for ∀(x_{h+1},...,x_m) ∈ ∏_{i=h+1}^{m} V_i
  do { ループ 2 の初め }
   if (x_{h+1},...,x_m) ∉ CANS then
   begin
    for ∀(L(c_1;p_1),···,L(c_m;p_m)) ∈ ∏_{i=1}^{h} CS_i(y_i, a_i)
        × ∏_{i=h+1}^{m} CS_i(y_i, x_i) do { ループ 3 の初め }
     if L(c_1;p_1) ∩ ... ∩ L(c_m;p_m) ≠ φ then
     begin
      CANS := CANS ∪ {(x_{h+1},...,x_m)};
      goto exitloop
     end; { ループ 3 の終わり }
    exitloop: { ループ 3 を抜けるための空文 }
   end { ループ 2 の終わり }
 end; { ループ 1 の終わり }
 step3: ANS := FANS ∪ CANS
end.
```

図 2 一般の m に素直に拡張された HaNa 法
Fig.2 The HaNa method generalized for the case of general m in a straight forward manner.

$$
\begin{aligned}
&\prod_{i=1}^{h} CS_i(y_i, a_i) \times \prod_{i=h+1}^{m} CS_i(y_i, x_i) = \\
&\quad \{(L(c_1; p_1), \ldots, L(c_m; p_m)) \mid \\
&\quad L(c_i; p_i) \subset CS_i(y_i, a_i), i = 1, \ldots, h, \\
&\quad L(c_i; p_i) \subset CS_i(y_i, x_i), i = h+1, \ldots, m\}.
\end{aligned}
$$

の記法を用いている．

しかし，この方法には以下のような問題点がある．

1) step1 の $FANS$の計算を，$m = 2$ の場合，HaNa 法は数え上げ法により最悪計算時間 $O(ne)$ で行う．一般の m の場合 (1 節の式 (1) - (3)) に数え上げ法を適用するには，1 節で述べたやり方で行えばよい．しかし，その最悪時間量は $O(n(e^h + e^{m-h} + t_0))$ となり，h が 1 または m に近いとき，マジック集合法の最悪時間量 $O(e^m)$ とほぼ等しく，非行率である．(少なくとも一つの G_iが閉路を含まないので，$FANS$の計算に必要な，積グラフ $G_1 \times \cdots \times G_h$ および $G_{h+1} \times \cdots \times G_m$の路の最大長は nである．)

2) step2 の if 文の L 式判定テスト:

$$
L(c_1; p_1) \cap \ldots \cap L(c_m; p_m) \neq \phi \qquad (12)
$$

を，$m=2$ の場合，HaNa 法は

$$L(c_1;p_1)\cap L(c_2;p_2)\neq\phi \Leftrightarrow \exists K(\text{整数})(c_1-c_2)/\gcd(p_1,p_2)=K \quad (13)$$

により行うことができるが [11]，この式は一般の m には適用できない．

3) step2 の主な計算時間は L 式判定テストを行う時間である．L 式判定テストは 3 重の for ループ；すなわち，ループ 1，ループ 2，ループ 3 により繰り返し実行される．ループ 1 の繰り返し回数は t_0，ループ 2 の繰り返し回数は n^{m-h}である．また，$CS_i(y_i,x_i)$ に含まれる異なる $L(c;p)$ の数を $\parallel CS_i(y_i,x_i)\parallel$ と記すと，(3.2 節 (ii) の式 (11) から示すことができるように) $\parallel CS_i(y_i,x_i)\parallel\leq n$ であるので [1, 2]，ループ 3 の繰り返し回数は

$$\prod_{i=1}^{h}\parallel CS_i(y_i,a_i)\parallel\times\prod_{i=h+1}^{m}\parallel CS_i(y_i,x_i)\parallel=n^m$$

である．したがって，全ての L 式判定テストを行う最悪計算時間は $O(t_0n^{2m-h}T_{test})$ となる．ここで，T_{test}は 1 回の L 式判定テストを行う時間である．$h(\leq m)$ があまり大きくない場合，図 2 の方法はマジック集合法より劣る．

これらの問題点のうち，第 1 に関しては，逆数え上げ法 [3] により，最悪時間量 $O(t_0nme+t_0n|FANS|log|FANS|)$，最悪領域量 $O(nm)$ で $FANS$を計算することができる．第 2 と第 3 の問題点については，4.2 節と 4.3 節で説明する．

4.2 L 式判定テスト

はじめに，$L(c;p)\cap L(c';p')\neq\phi(0\leq c<p, 0\leq c'<p')$ の場合，

$$L(c'';p'')=L(c;p)\cap L(c';p')$$

を満たす $c''(\leq p''), p''$ を求める方法を図 3 に示す．この計算時間は，ユークリッド互除法にかかる $O(\log\max\{p,p'\})$ [14, 15] である．

次に，図 3 の方法を使って，1 回の L 式判定テスト (式 (12)) を $O(m\log m\log n)$ 時間で行う方法を図 4 に示す．図 4 の方法の計算時間は，step3，step4 におけるユークリッド互除法の計算時間 $\log(\max\{p_{2i-1}^k,p_{2i}^k\})$ [14, 15] の和である．$p_i^k\leq n^{2^{k-1}}$であるので，

$$\sum_{k=1}^{\log m}\sum_{i=1}^{m/2^k}O(\log(\max\{p_{2i-1}^k,p_{2i}^k\})) = \sum_{k=1}^{\log m}\sum_{i=1}^{m/2^k}O(\log n^{2^{k-1}}) = O(m\log m\log n).$$

step1: $\gcd(p,p')$ をユークリッド互除法により求める．よく知られているように，$\gcd(p,p')$ はある整数δ,γ を用いて

$$\gcd(p,p')=\delta p+\gamma p' \quad (F1)$$

と表される (このδ,γも求まる)．

step2: $L(c;p)\cap L(c';p')\neq\phi$および式 (13) より

$$\exists K(\text{整数}) \quad (c-c')/\gcd(p,p')=K \quad (F2)$$

が成立するので，式 (F1),(F2) より $\gcd(p,p')$ を消去して，

$$c-K\delta p=c'+K\gamma p'$$

を得る．これより，$L(c'';p'')$ は

$$p'' = lcm(p,p')=pp'/\gcd(p,p'),$$
$$c'' = (c-K\delta p) \bmod p''.$$

図 3 $L(c'';p'')=L(c;p)\cap L(c';p')$ を満たす $L(c'';p'')$ の計算法

Fig.3 The algorithm to compute $L(c'';p'')$ satisfying $L(c'';p'')=L(c;p)\cap L(c';p')$.

step1: { 初期化 } c_i,p_iを各々c_i^1,p_i^1と記す $(i=1,\ldots,m)$．$k:=1$;

step2: $k=\log m+1$(すなわち，$L(c_1;p_1)\cap\ldots\cap L(c_m;p_m)\neq\phi$を計算済み) ならば，式 (12) は成立として終了．

step3: 式 (13) を使って，$L(c_{2i-1}^k;p_{2i-1}^k)\cap L(c_{2i}^k;p_{2i}^k)\neq\phi$の判定を行う $(i=1,\ldots,m/2^k)$．少なくとも一つの i に対し $L(c_{2i-1}^k;p_{2i-1}^k)\cap L(c_{2i}^k;p_{2i}^k)=\phi$ であれば，式 (12) は不成立として終了．

step4: $L(c_i^{k+1};p_i^{k+1})=L(c_{2i-1}^k;p_{2i-1}^k)\cap L(c_{2i}^k;p_{2i}^k)$ を満足する c_i^{k+1},p_i^{k+1}を図 3 の方法により求める $(i=1,\ldots,m/2^k)$．$k:=k+1$; として step2 へ．

図 4 L 式判定テスト

Fig.4 The algorithm to examine an L expression.

ところで，各 $(y_1,\ldots,y_m)(\in R_0)$ に対する複数回の L 式判定テスト ..., $L(c_1;p_1)\cap\ldots\cap L(c_{m-1};p_{m-1})\cap L(c_m;p_m)\neq\phi$, $L(c_1;p_1)\cap\ldots\cap L(c_{m-1};p_{m-1})\cap L(c'_m;p'_m)\neq\phi$, ... において，前回の L 式判定テスト $L(c_1;p_1)\cap\ldots\cap L(c_m;p_m)\neq\phi$ の中間データを記憶し，かつ，今回の L 式判定テスト $L(c_1;p_1)\cap\ldots\cap L(c'_m;p'_m)\neq\phi$ が前回の L 式判定テストと一つの $L(c;p)$ しか替わらないようにすると ($L(c'_m;p'_m)$ のみ $L(c_m;p_m)$ と異なる)，図 4 の step3, step4 は一つの i に対してのみ計算すればよい．したがって，各 $(y_1,\ldots,y_m)(\in R_0)$ に対する 2 回目以降の L 式判定テストにおいて，1 回の計算時間を$\sum_{k=1}^{\log m}O(\log n^{2^{k-1}})=O(m\log n)$ に減らすことができる．

4.3 L式判定テストの回数の減少

[補題] グラフ $G=(V,E)$ において節点 y を一つ固定する．y と G の全ての節点との間の簡略化距離集合の和を

$$SS(y) = \bigcup_{x\in V} CS(y,x) = \bigcup_{i} L(c_i;p_i)$$

と記し，その中に含まれる異なる式 $L(c_i;p_i)$ の数を $\| SS(y) \|$ と記す．そのとき，

$$\| SS(y) \| \leq |V|(=n). \qquad (14)$$

(証明) (i)$L(c_i;p_i)$ に現れる異なる p_i を，一般性を失うことなく，$p_1,p_2,\ldots,p_{d''}$ とする．はじめに，

$$p_1+p_2+\cdots+p_{d''} \leq n \qquad (15)$$

を示す．G の閉路を含む極大強連結成分を $\tilde{G}^i=(\tilde{V}^i,\tilde{E}^i)(i=1,\ldots,f)$ とすると，

$$SS(y) = \Big(\bigcup_{x\in\tilde{V}^1\cup\cdots\cup\tilde{V}^f} CS(y,x)\Big) \cup \Big(\bigcup_{x\in V-(\tilde{V}^1\cup\cdots\cup\tilde{V}^f)} CS(y,x)\Big)$$

と表せる．ここで，第1項 $\bigcup_{x\in\tilde{V}^1\cup\cdots\cup\tilde{V}^f} CS(y,x)$ は閉路に含まれる全ての節点の簡略化距離集合の和であり，第2項 $\bigcup_{x\in V-(\tilde{V}^1\cup\cdots\cup\tilde{V}^f)} CS(y,x)$ は閉路に含まれない全ての節点の簡略化距離集合の和である．3.2節のケース2で紹介したように，第2項の中に現れる任意の $L(c';p)$ の p は第1項の中のある $L(c;p)$ の p と等しい．また，同じく3.2節のケース1で紹介したように，第1項の各 $\bigcup_{x\in\tilde{V}^i} CS(y,x)$ の中に現れる $L(c;p)$ の p は全て等しく(それを p'_i とおく)，$p'_i \leq |\tilde{V}^i|$ である．したがって，

$$\begin{aligned} &p_1+p_2+\cdots+p_{d''} \leq p'_1+p'_2+\cdots+p'_f \\ &(i\neq j\text{に対し}\ p'_i=p'_j\text{の場合も有り得る}) \\ &\leq |\tilde{V}^1|+\cdots+|\tilde{V}^f| \leq |V| = n. \end{aligned}$$

(ii)次に，任意の $L(c;p)$ において，$0\leq c\leq p-1$ であるので，

(同じ p_i をもつ異なる $L(c_i;p_i)$ の数) $\leq p_i$. (16)

式(15),(16)より式(14)が証明された．□

補題を利用してL式判定テストの回数を減らすことができる．図2のstep2の3重のforループのうち，ループ2では，各 $(x_{h+1},\ldots,x_m)(\in\prod_{i=h+1}^m V_i)$ ごとにL式判定テスト(式(12))を行う．このため，同じL式判定テストが異なる $(x_{h+1},\ldots,x_m)$ に対して重複して行われる場合がある．拡張HaNa法は，この重複を防ぐために，各 $(y_1,\ldots,y_m)(\in R_0)$ に対し $SS_i(y_i)(i=h+1,\ldots,m)$ を作り，$SS_i(y_i)$ の中の式 $L(c_i,p_i)$ の重複を取り除いた後，L式判定テストを行う．同じ $L(c_1,p_1),\ldots,L(c_m,p_m)$ の組に対する重複テストを除くようにすれば，補題より，L式判定テストの全回数を，

$$|R_0|\times\prod_{i=1}^{h}\| CS_i(y_i,a_i)\| \times \prod_{i=h+1}^{m}\| SS_i(y_i)\| = O(t_0 n^m)$$

回に減らすことができる．計算時間は $O(t_0 n^m T_{test})$ である．

```
begin
 step1: 逆数え上げ法を使って FANS を計算する;
 step2: { CANSの計算 }
 CANS := φ;
 for ∀(y_1,...,y_m) ∈ R_0 do { ループ1の初め }
 begin
  step2.1: HaNa法のアルゴリズムにより,
  簡略化距離集合 CS_i(y_i,x_i)(i=1,...,m, x_i ∈ V_i)
  を計算する;
  step2.2: 各グラフ G_i(i=h+1,...,m) において,
  SS_i(y_i) および NN_i(L(c_i;p_i),y_i)
  (L(c_i;p_i) ⊂ SS_i(y_i)) を計算する;
  step2.3: {(y_1,...,y_m) に対する CANSの計算 }
  for ∀(L(c_{h+1};p_{h+1}),...,L(c_m;p_m))
   ∈ ∏_{i=h+1}^m SS_i(y_i) do { ループ2の初め }
  begin
   for ∀(L(c_1;p_1),...,L(c_h;p_h)) ∈ ∏_{i=1}^h CS_i(y_i,a_i)
   do { ループ3の初め }
    if L(c_1;p_1) ∩ ··· ∩ L(c_m;p_m) ≠ φ then
    begin
     CANS := CANS ∪ ∏_{i=h+1}^m NN_i(L(c_i;p_i),y_i);
     goto exitloop
    end; { ループ3の終わり }
   exitloop: { ループ3を抜けるための空文 }
  end { ループ2の終わり }
 end; { ループ1の終わり }
 step3: ANS := FANS ∪ CANS
end.
```

図5 拡張HaNa法
Fig.5 The modified HaNa method.

4.4 拡張HaNa法

以上の議論をまとめ，拡張HaNa法を図5に与える．ただし，

$$NN_i(L(c_i;p_i),y_i) = \{x_i \mid L(c_i;p_i)\subset CS_i(y_i,x_i)\} \quad i=h+1,\ldots,m,$$

$$\prod_{i=h+1}^{m} NN_i(L(c_i;p_i),y_i) = \{(x_{h+1},\ldots,x_m) \mid$$

$$x_i \in NN_i(L(c_i;p_i),y_i), i=h+1,\ldots,m\},$$

$$\prod_{i=h+1}^{m} SS_i(y_i) = \{(L(c_{h+1};p_{h+1}),\ldots,L(c_m;p_m)) \mid L(c_i;p_i) \subset SS_i(y_i), i=h+1,\ldots,m\}$$

とおく.

4.5 例題

$m=3, h=2$, 問い合わせを $s(a^5,a^7,x)$? とする. また, $R_0=\{(a^0,a^0,a^0)\}$ とし, $G_i(i=1,2,3)$ は全て図 1 の G に等しいとする.

はじめに $CANS$ を求める (図 5 の step2). $|R_0| = \{(a^0,a^0,a^0)\}$ であるので, ループ 1 すなわち step2.1 から step2.3 は, $(y_1,y_2,y_3)=(a^0,a^0,a^0)$ に対し 1 回だけ実行される. step2.1 で計算される簡略化距離集合 $CS_1(a^0,a^j), CS_2(a^0,a^j), CS_3(a^0,a^j)(j=0,\ldots,12)$ は全て, 3.2 節で求めた $CS(a^0,a^j)$ に等しい. したがって, step2.2 において,

$$\begin{aligned} SS_3(a^0) &= \bigcup_{j=0}^{12} CS_3(a^0,a^j) \\ &= L(0;2)\cup L(1;2)\cup L(0;4) \\ &\quad \cup L(1;4)\cup L(2;4)\cup L(3;4). \end{aligned}$$

また, 例えば, $L(1;2)$ は $CS(a^0,a^5)$ と $CS(a^0,a^{12})$ に含まれるので,

$$NN_3(L(1;2),a^0) = \{a^5,a^{12}\}.$$

他も同様に計算すると,

$$\begin{aligned} NN_3(L(0;2),a^0) &= \{a^6\}, \\ NN_3(L(0;4),a^0) &= \{a^4,a^8,a^{10},a^{12}\}, \\ NN_3(L(1;4),a^0) &= \{a^1,a^7,a^9,a^{11}\}, \\ NN_3(L(2;4),a^0) &= \{a^2,a^8,a^{10},a^{12}\}, \\ NN_3(L(3;4),a^0) &= \{a^3,a^7,a^9,a^{11}\}. \end{aligned}$$

step2.3 のループ 2 の $SS_3(y_3)$ は $SS_3(a^0)$, また, ループ 3 の $CS_1(y_1,a_1)$ と $CS_2(y_2,a_2)$ は各々, $CS_1(a^0,a^5)$, $CS_2(a^0,a^7)$ である. したがって, 例えば, $L(1;2)(\in CS_1(a^0,a^5))$, $L(1;4)(\in CS_2(a^0,a^7))$, $L(1;2)(\in SS_3(a^0))$ に対し, L 式判定条件:

$$L(1;2)\cap L(1;4)\cap L(1;2) \neq \phi$$

が成立する. 同様に, $L(1;4), L(3;4)(\in SS_3(a^0))$ に対しても, L 式判定条件が成立し,

$$\begin{aligned} CANS &= NN_3(L(1;2),a^0)\cup NN_3(L(1;4),a^0) \\ &\quad \cup NN_3(L(3;4),a^0) \\ &= \{a^1,a^3,a^5,a^7,a^9,a^{11},a^{12}\} \end{aligned}$$

を得る.

本例では, $FANS$の計算結果は $CANS$と等しく, $ANS=CANS$となる.

5 最悪計算量の解析

(1) 図 5 の step1: $FANS$を計算する逆数え上げ法 [3] の最悪時間量は $O(t_0nme + t_0n|FANS|\log|FANS|)$, 最悪領域量 $O(nm)$ である.

(2) 図 5 の step2.1: i および y_iを固定したときの $CS_i(y_i,x_i)(\forall x_i \in V_i)$ を計算する最悪時間量は $O(ne)$ であるので [1, 2], step2.1 の最悪時間量は $O(t_0mne)$, 最悪領域量は $O(mne)$ である.

(3) 図 5 の step2.2: i および y_iを固定し, 全ての $x_i(\in V_i)$ について $CS_i(y_i,x_i)$ を調べながら, 一つの $SS_i(y_i)$ を計算する時間は $\sum_{x_i\in V_i} \| CS_i(y_i,x_i) \| \log \| SS_i(y_i) \|$ である. また, 文献 [1, 2] および 4.3 節の補題より $\| CS_i(y_i,x_i) \| \leq \| SS_i(y_i) \| \leq n$ である. したがって, t_0個の $(y_1,\ldots,y_m)\in R_0$ に対して $SS_{h+1}(y_{h+1}),\ldots,SS_m(y_m)$ を計算する時間 $Time1$ は,

$$\begin{aligned} Time1 &= \\ &\quad t_0\Big(\sum_{i=h+1}^{m}\sum_{x_i\in V_i} \| CS_i(y_i,x_i) \| \log \| SS_i(y_i) \|\Big) \\ &= O(t_0(m-h)n^2\log n). \end{aligned}$$

また, そのとき必要となる領域量 $Space1$ は

$$Space1 = \sum_{i=h+1}^{m} \| SS_i(y_i) \| = O((m-h)n).$$

各 $NN_i(L(c_i;p_i),y_i)$ の計算は, $CS_i(y_i,x_i)$ より $SS_i(y_i)$ を作る際に $L(c_i;p_i)$ から x_iへのポインタをはることにより, 計算時間を増やすことなく, $SS_i(y_i)$ の計算と同時に進めることができる. そのとき必要となる領域量 $Space2$ は, $NN_i(L(c_i;p_i),y_i) \leq |V_i| = n$ より,

$$\begin{aligned} Space2 &= \sum_{i=h+1}^{m}\sum_{L(c_i;p_i)\subset SS_i} |NN_i(L(c_i;p_i),y_i)| \\ &= O((m-h)n^2). \end{aligned}$$

以上より, step2.2 の最悪時間量は $O(t_0(m-h)n^2\log n)$, 最悪領域量は $Space1$ と $Space2$ より $O((m-h)n^2)$ である.

(4) 図 5 の step2.3: 4.2 節より 1 回の L 式判定テストに必要な時間は $O(m\log n)$, また, 4.3 節より L 式判定テストの全回数は $O(t_0n^m)$ 回である. したがって, 全ての L 式判定テストに必要な時間 $Time3$ は,

$$Time3 = O(t_0n^m m\log n).$$

次に，式

$$CANS := CANS \cup \prod_{i=h+1}^{m} NN_i(L(c_i; p_i), y_i) \quad (17)$$

を計算するのに必要な全時間 $Time4$ について考える．式 (17) は，ループ 3 の繰り返しでは高々1 回しか実行されない．したがって，式 (17) の全実行回数は，$|R_0| \prod_{i=h+1}^{m} \| SS_i(y_i) \|$ である．また，1 回の実行に必要な時間は，$\prod_{i=h+1}^{m} |NN_i(L(c_i; p_i), y_i)| (\leq |CANS|)$ 個の解 $(x_{h+1}, \ldots, x_m)$ を集合 $CANS$に重複を除きながら格納する時間であり，一つの解を $CANS$に格納する時間は，$CANS$の要素 (すなわち解) を整列しておけば $O(\log |CANS|)$ である．したがって，

$$\begin{aligned} Time4 &= |R_0| \prod_{i=h+1}^{m} \| SS_i(y_i) \| \\ &\quad \prod_{i=h+1}^{m} |NN_i(L(c_i; p_i), y_i)| \log |CANS| \\ &= O(t_0 n^{m-h} |CANS| \log |CANS|). \end{aligned}$$

$Time3$ と $Time4$ を合わせ，step2.3 の最悪時間量は $O(t_0 n^m m \log n + t_0 n^{m-h} |CANS| \log |CANS|)$ である．また，step2.3 において必要となる領域量は $CANS$を記憶するためのものであり，$O(|CANS|)$ である．

(5) 図 5 の step3: 最悪時間量は $O(|ANS| \log |ANS|)$，最悪領域量は $O(|ANS|)$ である．

(6) 以上，(1) から (5) をまとめると，拡張 HaNa 法の最悪時間量は

$$O(t_0(mne + n^m m \log n + n^{m-h} |ANS| \log |ANS|))$$

最悪領域量は

$$O(mne + |ANS|).$$

ここで m を定数と見なし，m についての多項式項を消去すると，$m \geq 3$ のとき最悪時間量は

$$O(t_0(n^m \log n + n^{m-h} |ANS| \log n)).$$

6 最悪計算量の比較

$m \geq 3$ の場合に拡張 HaNa 法がマジック集合法および HaNa法より効率的であることを示す．($m = 2$ の場合，拡張 HaNa 法は HaNa 法より劣る.)

(1) $m \geq 3$ かつ $|ANS| \leq n^h$の場合の最悪時間量:

$h \geq m/2$ の場合，$|ANS| \leq n^h$であり，また，$h < m/2$であっても解集合のサイズが小さければ，この条件は成立する．このとき，拡張 HaNa 法の最悪時間量は $O(t_0 n^m \log n)$ となる．(m の多項式項は消去．) よく議論される応用例では，外延データベース R_0 は $R_0 = \{(y_1, \ldots, y_m) \mid y_1 = \cdots = y_m\}$ として定義され，$t_0 = O(n)$ 程度の大きさである．一方，マジック集合法の最悪時間量は $O(e^m)$，HaNa法の最悪時間量は $O(n^h(e^h + t_0) + n^{m-h}(t_0 + e^{m-h}))$ であるので，$O(n) \leq O(e) \leq O(n^2)$ を考慮すると，$O(e) > O(n)$ の場合，拡張 HaNa法は最悪時間量においてマジック集合法および HaNa 法より優れる．

(2)$m \geq 3$ の場合の最悪領域量:

いかなるアルゴリズムも，結果を貯えるために $|ANS|$ の領域量を必要とするので，比較的小さな領域量 $O(mne)$ を無視すれば，拡張 HaNa 法の領域量はほぼ最適である．($O(mne)$ はマジック集合法の最悪領域量は $O(n^m)$ と比べ小さい．)

7 むすび

演繹データベースで扱われる有名な問題である同世代問題を，再帰述語のもつ変数の数を 2 から m に一般化した問題に対し，$m = 2$ の場合に知られている HaNa法を一般の m に変形した拡張 HaNa 法を提案し，その最悪計算量を解析した．さらに，従来の方法の中で効率的と思われるマジック集合法および HaNa 法と比較して，拡張 HaNa 法が，$m \geq 3$ かつ通常よく生じるデータの場合，最悪時間量において優れていることを示した．また，拡張 HaNa 法の最悪領域量がほぼ最適であることを示した．平均計算量の評価が今後の課題である．

文献

[1] R.W.Haddad and J.F.Naughton: "Counting method for cyclic relations", Proc. 7th ACM Symp. on PODS, pp.333-340 (1988).

[2] R.W.Haddad and J.F.Naughton: "A counting algorithm for a cyclic binary query", J. Comput. & Syst. Sci., 43, 1, pp.145-169 (1991).

[3] F.Bancilhon, D.Maier, Y.Sagiv and J.Ullman: "Magic sets and other strange ways to implement logic programs", Proc. 5th ACM SIGMOD-SIGACT Symp. on PODS, pp.1-15 (1986).

[4] A.Marchetti-Spaccamela and D.Sacca: "Worst-case complexity analysis of methods for logic query implementation", Proc. 6th ACM SIGACT-SIGMOD-SIGART Symp. on PODS, pp.294-301 (1987).

[5] L.J.Henschen and S.A.Naqvi: "On compiling queries in recursive first order databases ", JACM, **31**, 1, pp.47-85 (1984).

[6] F.Bancilhon and R.Ramakrishnan: "An amateur's introduction to recursive query processing strategies", Proc. ACM-SIGMOD Conf., pp.16-52 (1986).

[7] G. Grahne, S. Sippu and E.Soisalon-Soininen: "Efficient evaluation for a subset of recursive queries", Proc. ACM SIGACT-SIGMOD-SIGART Symp. on PODS, pp.284-293 (1987).

[8] J.Han and L.J.Henschen: "Handling redundancy in the processing of recursive database queries", Proc. ACM SIGMOD Conf., pp.73-81 (1987).

[9] D. Sacca and C. Zaniolo: "Magic Counting Methods", Proc. ACM SIGMOD Conf., pp.49-59 (1987).

[10] H. Aly and Z. M. Ozsoyoglu: "Synchronized counting method", Proc. 5th Conf. of Data Engineering, pp.366-373 (1989).

[11] J.Han and L.J.Henschen: "The level-cycle merging method", Proc. 1st Conf. on Deductive and Object-Oriented Databases, pp.113-129 (1989).

[12] J.Han: "Multi-way counting Method", Information Systems, **14**, 3, pp.219-229(1989).

[13] D. Sacca and C. Zaniolo: "On the Implementation of a Simple Class of Logic Queries for Databases", Proc. 5th ACM SIGACT-SIGMOD Symp. on PODS, pp.16-23 (1986).

[14] 茨木俊秀: "アルゴリズムとデータ構造", pp.5-16, 昭晃堂 (1989).

[15] D.E.Knuth: "The Art of Computer Programming: Volume 2(2nd ed.)", Addison-Wesley (1981) (中川圭介訳: "準数値算法/算術演算", サイエンス社 (1986)).

(受理 平成 5 年 3 月 20 日)