意図に基づくアプローチのもとでの SQL ビューの更新可能性
―非線形連立方程式問題への帰着―

増永 良文† 長田 悠吾‡ 石井 達夫‡

†お茶の水女子大学（名誉教授） 〒112-8610 東京都文京区大塚 2-1-1
‡SRA OSS, Inc. 日本支社 〒171-0022 東京都豊島区南池袋 2-32-8
E-mail: †yoshi.masunaga@gmail.com, ‡{nagata, ishii}@sraoss.co.jp

あらまし リレーショナルデータベースにおけるビュー更新にはこれまで強い制約が課せられてきた．その状況を打開するべく，我々は「意図に基づくアプローチ」を提案すると共に，それがリレーショナルデータモデルが立脚する「集合意味論」のみならず，SQL が立脚する「バッグ意味論」の下でも機能し,これまでのアプローチでは更新不可とされてきた直積ビューや結合ビューを状況により更新可能となしうることを示してきた．また，これらの結果に基づき，SQL の直積ビューや結合ビューの更新可能性を具体的に PostgreSQL 上で実装し検証を行ってきている．しかしながら，実装された更新可能性判定アルゴリズムは，考えられる変換候補を総当たりに一つひとつ調べるという，いわゆる「暴力法」によっていた．本稿では，バッグ意味論の下での SQL 直積ビューや結合ビューの更新判定問題が「非線形連立方程式」を解く問題に帰着できることを示す．これにより意図に基づくアプローチのより効率的な実装法の開発に新たな道を拓くことができたのではないか，と考えられる．

キーワード ビュー，ビュー更新問題，意図に基づくアプローチ，バッグ意味論，SQL，PostgreSQL，リレーショナルデータベース，非線形連立方程式

1 はじめに

ビュー更新問題はビューがリレーショナルデータモデルの発案者であるコッドにより導入されて[1]以来，長年にわたり数多くの研究が報告されてきた．それらは大別すると，構文的[2]，意味的[3]，そしてインタラクティブア[4]プローチに分類される．これらのアプローチを本報告では「従来型アプローチ」と呼ぶが，いずれも更新可能とされるビューへの制約が強く，より多くのビューを更新可能にできる解決策が模索されてきた．

この問題に対して，近年，著者らにより「意図に基づくアプローチ」が発案されて，従来不可とされてきた直積ビューや結合ビューへの更新可能性に道が拓かれた[5,6]．当初，このアプローチはリレーショナル代数で定義されるビューを対象としていた．つまり,「集合意味論」を元に理論が展開されていが，このアプローチを実際のリレーショナルデータベースの現場に適用しようとすると，国際標準リレーショナルデータベース言語 SQL は，表（table）に重複した行（row）の生起を許す「バッグ意味論」(マルチ集合意味論ともいう）に基づいているので，提案したアプローチがバッグ意味論の下でも機能するか否かが問われることとなったが，この問題は直積ビューや結合ビューに対して肯定的に解けることが示された[7,10]．

このような理論のもとに，著者らは，意図に基づくアプローチをオープンソースのリレーショナル DBMS として広く流布している PostgreSQL にプロトタイピングしてみて，それが所望の動きをすることを確認してきた[8,9,10]．

しかしながら，このプロトタイピングで実装した直積ビューへの更新要求変換アルゴリズムは，ビュー更新を実現できるかもしれないと考えられる実リレーションへのあらゆる更新を列挙してその更新可能性を検証するという，いわゆる「暴力法」（brute force，総当たり法ともいう）であったために，より効率の良い実装法が模索されることとなった．

そもそも，意図に基づくアプローチは，更新対象となったビューがその更新要求を受け付けることができるか否かを決定するために，それを一時的にマテリアライズする．本稿では，そこに着眼することにより，一般にバッグビューとそれに対する更新要求が与えられたときに，その更新可能性を判定する問題が「非線形連立方程式」が解を持つか否かという問題に帰着できることを明らかにすることに成功した．それを以下報告する．

2 ビュー更新問題と意図に基づくアプローチ

2.1 ビュー更新問題

ビュー更新問題を整理しておく．そのために，まずビ

ューの更新可能性の定義を示す[2].

【定義 1】(ビューの更新可能性)

V をビューとする. V が更新可能とは, いかなる時刻 τ, いかなる更新要求 $\boldsymbol{u}$ に対しても, 図 1 の可換図式が成立するときをいう. ここに, 更新とは削除, 挿入, あるいは書換のいずれかとする. ■

つまり, ビューは実バッグ群がなすデータベース状態(database state)からビュー状態(view state)への「関数」(function)であると規定することから始まる. つまり, s_τをある時刻 τ におけるデータベースの状態, V をビュー定義, $V(s_\tau)$は (その時刻 τ における)ビューの状態, $\boldsymbol{u}$ を $V(s_\tau)$に対して発行された更新操作とするとき, $\boldsymbol{u}$ が**変換可能**(translatable)であるとは, $\boldsymbol{u}$ を s_τへの更新操作に変換する副作用がなく(no side effects)かつ一意(unique)な変換(translation) T が存在するときと定義する. すなわち, 図 1 の可換図式(commutative diagram)が成立するときをいう. なお, 変換 T に副作用がないとは, $\boldsymbol{u}(V(s_\tau)) = V(T(\boldsymbol{u})(s_\tau))$ が成立することである. 変換 T に一意性を課したことは議論の余地があるところだとしながらも, その理由は T に代替案があった時, その選択基準を設けられないためである.

$$\begin{tikzcd} V(s_\tau) \arrow[r, dashed, "u"] \arrow[dr, phantom, "T\downarrow"] & u(V(s_\tau)) = V(T(u)(s_\tau)) \\ s_\tau \arrow[u, "V"] \arrow[r, dashed, "T(u)"] & T(u)(s_\tau) \arrow[u, "V"'] \end{tikzcd}$$

図 1 時刻 τ においてビュー更新要求 $\boldsymbol{u}$ が変換可能であることを示す可換図式

さて, ビュー更新問題とは, いかなる時に図 1 の可換図式が成立するのかを問う問題であるが, ここで, 可換図式の成立について補足説明をする. 上の定義から明らかなように, 可換性は本来インスタンスベースである. つまり, あるデータベースの状態, あるビュー定義, そしてある更新要求が与えられたときに, その三つ組みに対して図 1 の可換図式が成立するか否かを問うて良い. しかしながら, 従来型アプローチでは, いずれもがスキーマレベルの可換性を問うている. たとえば, 従来型アプローチでは「結合ビューは削除不可能である」であることが知られているが, その意味は, いかなるデータベースの状態でも, ビュー定義が結合ビューであり, 更新要求が削除であれば, それは受け付けないという意味である. これは可換図式が成り立たないような, あるデータベースの状態, ある結合ビューの定義, そしてある削除要求の組合せを示せるので, そうだとする (反例の提示). また, このようにスキーマレベルでビューの更新可能性を決定しておくと, リレーショナル DBMS はビューの更新可能性を管理しやすい (システムカタログで管理可能). SQL では SQL-92 で初めて SQL で定義されるビューの更新可能性が規格化され, SQL:1999 で新たな概念が示されているが, いずれの場合にも, 更新できるビューには強い制約が課せられているのは, スキーマレベルの考え方に立脚しているからである.

2.2 意図に基づくアプローチ

一方, 我々の考案した「意図に基づくアプローチ」[5,6]は図 1 の可換図式の成立を忠実にインスタンスレベルで問う (正例の提示). そのために, 更新対象となったビューを一時的にマテリアライズ (materialize, 体現化)することが要求されることとなる. そのために, パフォーマンスが劣化することが危惧されるが, この代償のもとに従来付加とされてきた直積ビューや結合ビューが状況によっては更新可能となった. その様子を表 1 に示す. さらに, 意図に基づくアプローチは, リレーショナル代数が立脚する集合意味論のみならず, SQL が立脚するバッグ意味論の下で, バッグ直積ビューやバッグ結合ビューが更新可能となる [7,10].

これまで, SQL:2008 で規格化された INSTEAD OF トリガを使ってビューを更新可能としようという動きがあったが, そのためのコアコードをどう書けばよいかの理論がなかったので, 事実上機能してこなかった. 意図に基づくアプローチはこのコードをどう書けばよいかの明快な指針を与えることとなる.

表 1 7 つの基本ビューの更新可能性の比較一覧

基本ビューの種類	更新操作	意図に基づくアプローチによる更新可能性	従来型アプローチによる更新可能性
bag-和ビュー	削除	○	○
	挿入	×*	×
	書換	○(直接書換法)	×
bag-差ビュー	削除	×*	×
	挿入	○	○
	書換	×*	×
bag-共通ビュー	削除	×*	×
	挿入	○	○
	書換	×*	×
bag-直積ビュー	削除	◎	×
	挿入	◎	×
	書換	◎(書換要求が両立)	×
bag-射影ビュー	削除	○	○
	挿入	○(主キーを含む)	○(主キーを含む)
	書換	○(直接書換法)	○(主キーを含む)
bag-選択ビュー	削除	×	×
	挿入	○	○
	書換	○	○
bag-結合ビュー	削除	◎	×
	挿入	◎	×
	書換	◎(書換要求が両立)	×

○, ×：従来型アプローチで更新可能, 不可能

◎：意図に基づくアプローチで更新可能

括弧書きは条件, ＊は本質的に不可能, を表す.

3 意図に基づくアプローチのためのビューの更新可能性の判定法：暴力法

さて，実際にリレーショナルデータベースの現場でより強力なビューサポートを実現するためには，いかにして意図に基づくアプローチをリレーショナル DBMS に実装できるかが問われることとなった.このために,我々はオープンソースのリレーショナルDBMSとして広く使われている PostgreSQL にそれを実装して,意図に基づくアプローチが理論通り稼働することを確認してきた．しかしながら，そこで実装されたバッグビューの更新可能性判定法は考えられる更新変換をすべて列挙して一つひとつ試すという，いわゆる「暴力法」であった．その例を例題 1 に示す.

【例題 1】（暴力法による更新可能性の判定）

R(A)={1(2), 2}，S(B)={1, 2(2)}を実バッグとし，バッグ直積ビューV を次のように定義する.

$V = R\times_{bag}S=\{(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)\}$

このとき，V に対して次の挿入要求が発せられたとする.

i= INSERT INTO V(A, B) VALUES (3, 1), (3, 2)(2);

まず，表 1 に示した通り，従来型アプローチでは直積ビューへの挿入要求は受け付けられない．しかしながら，以下に示すように，意図に基づくアプローチではこの挿入要求は受け付けられるが，その更新可能性の判定法は暴力的である.

すなわち，挿入要求 ***i*** は VALUES 句に記されている通り, バッグ{(3, 1), (3, 2)(2)}を V に挿入したいという意味である．そうすると，暴力法の下では，***i*** の可能な更新変換候補は R への 4 つの挿入候補と S への 6 つの挿入候補のすべての組合せとなる.

$\boldsymbol{i}_{R1}$=no insert request
$\boldsymbol{i}_{R2}$=insert into R values (3)
$\boldsymbol{i}_{R3}$=insert into R values (3)(2)
$\boldsymbol{i}_{R4}$=insert into R values (3)(3)

一方，S に対しては次の 6 つの代替案を考えないといけない.

$\boldsymbol{i}_{S1}$=no insert request
$\boldsymbol{i}_{S2}$=insert into S values (1)
$\boldsymbol{i}_{S3}$=insert into S values (2)
$\boldsymbol{i}_{S4}$=insert into S values (2)(2)
$\boldsymbol{i}_{S5}$=insert into S values (1), (2)
$\boldsymbol{i}_{S6}$=insert into S values (1), (2)(2)

なお，ここで使える発見的手法（heuristics）は$\neg(R \ni 3)$である．そうすると，$\boldsymbol{i}_{R1}$ と $\boldsymbol{i}_{R1}$～$\boldsymbol{i}_{S6}$ の組合せは考慮する必要はなく，18（=3×6）個の組合せについて，ビューをマテリアライズして，意図に基づくアプローチのもと実現可能性を検証すればよい．その結果，$\boldsymbol{i}_{R2}$ と $\boldsymbol{i}_{S1}$ の組合せだけが副作用を引き起こすことなく,丁度 ***i*** を実現することが分かる．したがって，意図に基づくアプローチのもとで，バッグ直積ビューへの挿入要求が受理される場合のあることが示される．以上が暴力法の説明である.

4 バッグビューの更新可能性判定と非線形連立方程式

4.1 ビュー更新問題の非線形連立方程式問題への帰着

さて，暴力法は所望のビュー更新を実現してくれるかもしれない変換候補が見つかるか否かを検証するために，文字通りあらゆる代替案を列挙することが必要である．そこで，この煩雑さを回避でき，より直截にビューへの更新要求の諾否を判定できる手法がないものか，模索することとなった．その結果，意図に基づくアプローチでは，ビューをマテリアライズしてビューへの更新要求の変換可能性を判定するのであるから，そのビューを定義している実バッグの行の重複度を変数として捉えて，そのビュー定義のもとでビュー更新要求を満たし得る重複度の組は存在するのか，しないのかをダイレクトに問えばよいのではないか，という発想に至った．実際にこの着想を定式化してみると，ビューの更新可能性は行の重複度を変数とする非線形連立方程式が解を有するか否かに帰着されることを明らかにすることができた.以下,それを具体的に示す.

4.2 バッグ直積ビューの削除可能性と非線形連立方程式

意図に基づくアプローチのもと，バッグ直積ビューへの削除要求が発行されたとする．この要求が削除時異状をきたすことなく実表に変換可能なのかを次のように判定する．例で示す.

【例題 2】R(A)={1(2), 2}，S(B)={1, 2(2)}を実バッグとし，バッグ直積ビューV を次のように定義する.

```
CREATE VIEW V (A, B)
AS
SELECT R.A, S.B
FROM R, S;
```

このとき, V に対して次の削除要求 ***d1*** が発せられたとする.

d1: DELETE FROM V
WHERE A=2;

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，$V^m = R\times_{bag}S=\{(1, 1)(2), (1, 2)(4), (2, 1)(1), (2, 2)(2)\}$ を得る.

② ***d1*** を変形して $\boldsymbol{d1^m}$ とし，それを実行する.
$\boldsymbol{d1^m}$: DELETE FROM V^m

WHERE A=2;

その結果，***d1^m*** は V^m から{(2, 1)(1), (2, 2)(2)}を削除したいという要求であることが分かる．そして，$W=V^m-_{bag}$\{(2, 1), (2, 2)(2)\}= \{(1, 1)(2), (1, 2)(4)\}とする．

③ R(A)=\{1(2), 2\}，S(B)=\{1, 2(2)\}であるので，行の重複度を表す変数（＝未知数）x, y, u, v を導入して，バッグ R^d=\{1(x), 2(y)\}, S^d=\{1(u), 2(v)\}を定義する*．ここに，変数 x, y, u, v は次を満たす．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v$

④ $R^d \times_{bag} S^d$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

$R^d \times_{bag} S^d$=\{(1, 1)(x×u), (1, 2)(x×v), (2, 1)(y×u), (2, 2)(y×v)\}

⑤ もし，V に対する ***d1*** が受理されるのであれば，W= $R^d \times_{bag} S^d$ が成立する x, y, u, v が存在するはずである．そのためには x, y, u, v は次の連立方程式を満たさなければならない．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v$・・・(0)
x×u=2　・・・(1)
x×v=4　・・・(2)
y×u=0　・・・(3)
y×v=0　・・・(4)

⑥ すると，(1)と(2)から，x, u, v≧ 1 でなければならないことが分かる．したがって，このことと，(3)あるいは(4)から y=0 であることが分かる．

⑦ また，x, u, v≧ 1 であることと(1)から，次の二つのケースがありうることが分かる．
(ア) x=1, u=2 である場合
(イ) x=2, u=1 である場合

⑧ まず，(ア)の場合を検証する．
この場合，x=1, y=0, u=2 であるので，この結果と(2)から，v=4 であることが分かる．これが何を意味するかを見てみると次のようである．

R=\{1(2), 2\}であり，x=1, y=0 なので，R^d=\{1(1)\}を実現するには，R から 1(2)のうちの一本の 1 だけを削除しなければならない．しかしながら，それは「バッグの重複元の識別不可能性」によりできない．しかしながら，（強引に）一旦 R から\{1(2)\}を削除し，続けて R に\{1(1)\}を挿入すれば R^d=\{1(1)\}を実現できる．次に，S^d=\{1(2), 2(4)\}を実現するには，S=\{1, 2(2)\}に\{1(1), 2(2)\}を挿入すればよい．この挿入は問題なくできる．そうすると，$R^d \times_{bag} S^d$=\{(1, 1)(2), (1, 2)(4)\}となり，所望の削除が実現できている．

⑨ 次に，(イ)の場合を検証する．
この場合，x=2, y=0, u=1 であるので，この結果と(2)から，v=2 であることが分かる．これが何を意味するかを見てみると次のようである．

R=\{1(2), 2\}, S=\{1, 2(2)\}なので，R から 2 を削除し，S についてはそのままとすると，所望の削除が実現できている．■

つまり，この例題で示したことは，非線形連立方程式の解が一意ではない場合（この例では 2 つの解が見つかった），それは上記の⑧と⑨に見るがごとく，発行された削除要求 ***d1*** を実現できる二つの代替案があることを意味するので，変換の一意性が満たされず，このビュー更新は不可と結論せざるを得ない．

さて，勿論，例題 2 の場合と異なり，ビューへの更新要求を実現する変換が一つも存在しないので，受理されない V への削除要求もある．そのような例を次に示す．

【例題 3】実バッグ R(A)と S(B)，およびバッグ直積ビューV は例題 2 で与えた通りとする．このとき，V に対して次の削除要求 ***d2*** が発せられたとする．

d2: DELETE FROM V
WHERE A=2 AND B=2;

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，$V^m = R \times_{bag} S$=\{(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)\}を得る．

② ***d2*** を変形して ***d2^m*** とし，それを実行する．
d2^m: DELETE FROM V^m
WHERE A=2 AND B=2;
その結果，***d2^m*** は V^m から\{(2, 2)(2)\}を削除したいという要求であることが分かる．
そして，$W=V^m-_{bag}$\{(2, 2)(2)\}= \{(1, 1)(2), (1, 2)(4), (2, 1)\}とする．

③ R(A)=\{1(2), 2\}，S(B)=\{1, 2(2)\}であるので，行の重複度を表す変数 x, y, u, v を導入して，バッグ R^d=\{1(x), 2(y)\}, S^d=\{1(u), 2(v)\}を定義する．ここに，変数 x, y, u, v は次を満たす．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v$

④ $R^d \times_{bag} S^d$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

* これらのバッグはもし V に対する ***d1*** が受理されるのであれば，それを実現するであろう R と S の結果を表していることになる．

$R^d \times_{bag} S^d$={(1, 1)(x×u), (1, 2)(x×v), (2, 1)(y×u), (2, 2)(y×v)}

⑤ もし，V に対する ***d2*** が受理されるのであれば，W= $R^d \times_{bag} S^d$ が成立する x, y, u, v が存在するはずである．そのためには x, y, u, v は次の連立方程式を満たさなければならない．

0≦x, 0≦y, 0≦u, 0≦v・・・(0)
x×u=2　・・・(1)
x×v=4　・・・(2)
y×u=1　・・・(3)
y×v=0　・・・(4)

⑥ すると，(1)，(2)，(3)から，x, y, u, v≧1 でなければならないことが分かる．しかし，これは(4)式と矛盾する．したがって，この連立方程式に解はない．つまり，V に対する削除要求 ***d2*** は受理できない．■

そうすると，最後に検証しておかねばならない問題は，変換の単調性に抵触することなく，唯一に変換可能な直積ビューへの削除要求はないのか，ということである．

【例題 4】実バッグ R(A)と S(B)，およびバッグ直積ビューV は例題 2 で与えた通りとする．このとき，V に対して次の削除要求 ***d3*** が発せられたとする．

d3: DELETE FROM V
WHERE A=1 OR (A=2 AND B=2);

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，V^m = R $\times_{bag}$S={(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)}を得る．

② ***d3*** を変形して ***d3*** とし，それを実行する．

d3^m: DELETE FROM V^m
WHERE A=1 OR (A=2 AND B=2);

その結果，***d3^m*** は V^m から{(1, 1)(2), (1, 2)(4), (2, 2)(2)}を削除したいという要求であることが分かる．

そして，W=$V^m -_{bag}$ {(1, 1)(2), (1, 2)(4), (2, 2)(2)}= {(2, 1)(1)}とする．

③ R(A)={1(2), 2}，S(B)={1, 2(2)}であるので，行の重複度を表す変数 x, y, u, v を導入して，バッグ R^d={1(x), 2(y)}, S^d={1(u), 2(v)}を定義する．ここに，変数 x, y, u, v は次を満たす．

0≦x, 0≦y, 0≦u, 0≦v

④ $R^d \times_{bag} S^d$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

$R^d \times_{bag} S^d$={(1, 1)(x×u), (1, 2)(x×v), (2, 1)(y×u), (2, 2)(y×v)}

⑤ もし，V に対する ***d3*** が受理されるのであれば，W= $R^d \times_{bag} S^d$ が成立する x, y, u, v が存在するはずである．そのためには x, y, u, v は次の連立方程式を満たさなければならない．

0≦x, 0≦y, 0≦u, 0≦v・・・(0)
x×u=0　・・・(1)
x×v=0　・・・(2)
y×u=1　・・・(3)
y×v=0　・・・(4)

すると，x=0, y=1, u=1, v=0 でなければならないことが分かる．つまり，***d3*** を次に示す R と S に対する削除要求に変換すれば所望の更新を実現できることが保証される．

d3^R：DELETE FROM R
WHERE A=1;
d3^S：DELETE FROM S
WHERE B=2;　■

以上の議論から，次を示すことができた．

【命題 1】意図に基づくアプローチのもとで，バッグ直積ビューは削除可能である．

（証明）例題 2，3，4 の議論より明らか．■

4.3 バッグ直積ビューの挿入可能性と非線形連立方程式

意図に基づくアプローチのもと，バッグ直積ビューの挿入が許される場合，許されない場合があることを例示する．

【例題 5】実バッグ R(A)と S(B)，およびバッグ直積ビューV は例題 2 で与えた通りとする．このとき，V に対して次の挿入要求が発せられたとする．

i1：INSERT INTO V(A, B)
VALUES (3, 1) , (3, 2)(2);

この書式は SQL 固有のものではないが，V に行のバッグ{(3, 1) , (3, 2)(2)}を挿入したいことを表している．

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，V^m = R $\times_{bag}$S={(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)}を得る．

② ***i1*** を変形して ***i1^m*** とし，それを実行する．

i1^m: INSERT INTO V^m(A, B)
VALUES (3, 1) , (3, 2)(2);

その結果，***i1^m*** は V^m を W=$V^m \cup_{bag}$ {((3, 1) , (3, 2)(2)}= {(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2) , (3, 1) , (3, 2)(2)}にしたいことが分かる．

③ R(A)={1(2), 2}，S(B)={1, 2(2)}であるが R には 3 を表す行がないのでそれを考慮する（発見的手法の介入）．行の重複度を表す変数 x, y, z, u, v を導入して，バッグ R^i={1(x), 2(y), 3(z)},

S^i={1(u), 2(v)}を定義する．このバッグはもしVに対する ***i1*** が受理されるのであれば，それを実現するであろうRとSの結果を表している．このとき次が成り立っている．

$0 \leqq x, 0 \leqq y, 0 \leqq z, 0 \leqq u, 0 \leqq v$

④ $R^i \times_{bag} S^i$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

$R^i \times_{bag} S^i$={(1, 1)(x×u), (1, 2)(x×v), (2, 1)(y×u), (2, 2)(y×v), (3, 1)(z×u), (3, 2)(z×v)}

⑤ もし，Vに対する ***i1*** が受理されるのであれば，W= $R^i \times_{bag} S^i$ が成立する x, y, z, u, v が存在するはずである．そのためには x, y, z, u, v は次の連立方程式を満たさなければならない．

$0 \leqq x, 0 \leqq y, 0 \leqq z, 0 \leqq u, 0 \leqq v$ ・・・(0)

$x \times u=2$ ・・・(1)

$x \times v=4$ ・・・(2)

$y \times u=1$ ・・・(3)

$y \times v=2$ ・・・(4)

$z \times u=1$ ・・・(5)

$z \times v=2$ ・・・(6)

⑥ すると，(3)と(5)から，y=u=z=1 でなければならないことが分かる．したがって，このことと(1)から x=2 であることが分かる．また．このことと(2)から v=2 であることが分かる．つまり，この連立方程式は一意に解けて，答えは x=2, y=1, z=1, u=1, v=2 であることが分かった．

⑦ 従って，R^i={1(2), 2(1), 3(1)}と S^i={1(1), 2(2)}のバッグ直積が所望の挿入要求を実現する R と S の更新結果であることが分かる．

⑧ そこで，これらを元の R と S と比較してみると，

$R^i=R \cup_{bag}\{3(1)\}$

$S^i=S$

なので，***i1*** を次に示す挿入要求に変換すれば所望の更新を実現できることが保証される．

i^R：INSERT INTO R(A)
VALUES 3;　■

なお，この結果は，同じ問題設定ものと，3.3 節で示した暴力法により得られた結果と一致している．

勿論，受理されない V への挿入要求もある．

【例題 6】実バッグ R(A)と S(B)，およびバッグ直積ビューV は例題 2 で与えた通りとする．このとき，V に対して次の挿入要求が発せられたとする．

i2： INSERT INTO V(A, B)
VALUES (1, 1)(2);

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，$V^m = R \times_{bag} S$={(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)}を得る．

② ***i2*** を変形して ***i2^m*** とし，それを実行する．

i2^m: INSERT INTO V^m(A, B)
VALUES (1, 1)(2);

その結果，***i2^m*** は V^m を $W=V^m \cup_{bag}$ {(1, 1)(2)}={(1, 1)(4), (1, 2)(4), (2, 1), (2, 2)(2) }としたいことが分かる．

③ R(A)={1(2), 2}，S(B)={1, 2(2)}であるので，行の重複度を表す変数 x, y, u, v を導入して，バッグ R^i={1(x), 2(y)}, S^i={1(u), 2(v)}を定義する．このバッグはもし V に対する ***i2*** が受理されるのであれば，それを実現するであろう R と S の結果を表しているので，次が成り立つ．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v$

④ $R^i \times_{bag} S^i$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

$R^i \times_{bag} S^i$={(1, 1)(x×u), (1, 2)(x×v), (2, 1)(y×u), (2, 2)(y×v)}

⑤ もし，Vに対する ***i2*** が受理されるのであれば，W= $R^i \times_{bag} S^i$ が成立する x, y, u, v が存在するはずである．そのためには x, y, u, v は次の連立方程式を満たさなければならない．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v$ ・・・(0)

$x \times u=4$ ・・・(1)

$x \times v=4$ ・・・(2)

$y \times u=1$ ・・・(3)

$y \times v=2$ ・・・(4)

⑥ すると，(0)と(3)から，y=u=1 でなければならないことが分かる．したがって，このことと(4)から v=2 であることが分かる．

⑦ しかし，u=1 であることと(1)から x=4 でないといけないが，一方で，v=2 であることと(2)から x=2 でないといけない．これは矛盾であって，この非線形連立方程式に解はない．

⑧ 従って，***i2*** を受理できない．　■

以上の議論から，次を示すことができた．

【命題 2】意図に基づくアプローチのもとで，バッグ直積ビューは挿入可能である．

（証明）例題 5 と 6 の議論より明らか．■

4.4 バッグ直積ビューの書換可能性と非線形連立方程式

意図に基づくアプローチのもと，バッグ直積ビューの書換が許される場合，許されない場合があることを例示する．

【例題 7】実バッグ R(A)と S(B)，およびバッグ直積ビューV は例題 2 で与えた通りとする．このとき，V に対して次の書換要求が発せられたとする．

```
r1 : UPDATE V
     SET A=3
     WHERE (A=2 AND B=1) OR (A=2 AND B=2);
```

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，$V^m = R \times_{bag} S$={(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)}を得る．

② ***r1*** を変形して ***r1^m*** とし，それを実行する．

```
r1m : UPDATE Vm
      SET A=3
      WHERE (A=2 AND B=1) OR (A=2 AND B=2);
```

その結果，***r1^m*** は V^m を $W=V^m -_{bag}$ {((2, 1) , (2, 2)(2)} $\cup_{bag}$ {((3, 1) , (3, 2)(2)}= {(1, 1)(2), (1, 2)(4), (3, 1), (3, 2)(2) }としたいことが分かる．

③ R(A)={1(2), 2}，S(B)={1, 2(2)}であるが R には 3 を表す行がないので，（発見的手法により）それを考慮して，行の重複度を表す変数 x, y, z, u, v を導入して，バッグ R^r={1(x), 2(y), 3(z)}, S^r={1(u), 2(v)}を定義する．このバッグはもし V に対する ***r1*** が受理されるのであれば，それを実現するであろう R と S の結果を表しているので，次が成り立つ．

$0 \leqq x, 0 \leqq y, 0 \leqq z, 0 \leqq u, 0 \leqq v$

④ $R^r \times_{bag} S^r$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

$R^r \times_{bag} S^r$={(1, 1)(x×u), (1, 2)(x×v), (2, 1)(y×u), (2, 2)(y×v), (3, 1)(z×u), (3, 2)(z×v)}

⑤ もし, V に対する ***r1*** が受理されるのであれば，$W= R^r \times_{bag} S^r$ が成立する x, y, z, u, v が存在するはずである．そのためには x, y, z, u, v は次の連立方程式を満たさなければならない．

$0 \leqq x, 0 \leqq y, 0 \leqq z, 0 \leqq u, 0 \leqq v$ ・・・(0)

$x \times u=2$ ・・・(1)

$x \times v=4$ ・・・(2)

$y \times u=0$ ・・・(3)

$y \times v=0$ ・・・(4)

$z \times u=1$ ・・・(5)

$z \times v=2$ ・・・(6)

⑥ すると，(0)と(3)と(5)から，y=0, 0＜u でなければならないことが分かる．したがって，このことと(5)から u=z=1 であることが分かる．そうすると．このことと(6)から v=2 であることが分かる．

⑦ 残るは，x を同定しないといけないので，(1)あるいは(2)に注目すると，x=2 を得る．

⑧ つまり，この連立方程式は一意に解けて，答えは x=2, y=0, z=1, u=1, v=2 であることが分かった．

⑨ つまり，R^r={1(2), 3(1)}と S^r={1(1), 2(2)}のバッグ直積が所望の書換要求を実現する R と S の更新結果であることが分かる．

⑩ そこで，これらを元の R と S と比較してみると，

$R^r=R -_{bag}$ {2(1)} $\cup_{bag}$ {3(1)}

$S^r=S$

なので，***r1*** を次に示す書換要求に変換すれば所望の更新を実現できることが保証される．

```
rR : DELETE FROM R
     WHERE A=2;
     INSERT INTO R(A)
     VALUES 3;
```

あるいは，

```
rR : UPDATE R
     SET A=3
     WHERE A=2;   ■
```

勿論，受理されない V への書換要求もある．

【例題 8】実バッグ R(A)と S(B)，およびバッグ直積ビューV は例題 2 で与えた通りとする．このとき，V に対して次の書換要求が発せられたとする．

```
r2 : UPDATE V
     SET B=3
     WHERE (A=2 AND B=1) OR (A=2 AND B=2);
```

●判定のための非線形連立方程式

① V をマテリアライズする．その結果，$V^m = R \times_{bag} S$={(1, 1)(2), (1, 2)(4), (2, 1), (2, 2)(2)}を得る．

② ***r2*** を変形して ***r2^m*** とし，それを実行する．

```
r2m : UPDATE Vm
      SET B=3
      WHERE (A=2 AND B=1) OR (A=2 AND B=2);
```

③ その結果，***r2^m*** は V^m を $W=V^m -_{bag}$ {((2, 1) , (2, 2)(2)} $\cup_{bag}$ {((2, 3) , (2, 3)(2)}= {(1, 1)(2), (1, 2)(4), (2, 3)(3) }としたいことが分かる．

④ R(A)={1(2), 2}，S(B)={1, 2(2)}であるが S には 3 を表す行がないので，（発見的手法により）それを考慮して，行の重複度を表す変数

x, y, u, v, w を導入して, バッグ R^r={1(x), 2(y)}, S^r={1(u), 2(v), 3(w)}を定義する．このバッグはもし V に対する ***r2*** が受理されるのであれば，それを実現するであろう R と S の結果を表しているので，次が成り立つ．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v, 0 \leqq w$

⑤ $R^r \times_{bag} S^r$ を計算する．その結果，バッグ直積演算結果の行の重複度計算の定義により次を得る．

$R^r \times_{bag} S^r$={(1, 1)(x×u), (1, 2)(x×v), (1, 3)(x×w), (2, 1)(y×u), (2, 2)(y×v), (2, 3)(y×z)}

⑥ もし, V に対する ***r2*** が受理されるのであれば, W= $R^r \times_{bag} S^r$ が成立する x, y, u, v, w が存在するはずである．そのためには x, y, u, v, w は次の連立方程式を満たさなければならない．

$0 \leqq x, 0 \leqq y, 0 \leqq u, 0 \leqq v, 0 \leqq w$ ・・・(0)

$x \times u=2$ ・・・(1)

$x \times v=4$ ・・・(2)

$x \times w=0$ ・・・(3)

$y \times v=0$ ・・・(4)

$y \times u=0$ ・・・(5)

$y \times w=3$ ・・・(6)

⑦ すると，(1)，(2)，(6)から，x, u, v, y, w は何れも≠0，即ち 1 以上である．

⑨ しかし，このことは，(3)，あるいは(4)，あるいは(5)と矛盾する．したがって，この非線形連立方程式は解を有しない．

⑩ 従って，***r2*** を受理する訳にはいかない． ■

以上の議論から，次を示すことができた．

【命題 3】意図に基づくアプローチのもとで，バッグ直積ビューは書換可能である．

（証明）例題 7 と 8 の議論より明らか．■

5 おわりに

本稿では，意図に基づくアプローチの下で，バッグ意味論における直積ビューと結合ビューの更新問題が「非線形連立方程式が解を有するか否か」という問題に帰着できることを示した．

今後の課題として，非線形連立方程式の解法は一般には難解とされ，近似解法が幾つか存在するが，ビュー更新問題に特化した場合の厳密解法を明らかにすると共に，発見的手法の導入や最適化法の開発により[11]，より効率の良いビュー更新アルゴリズムを開発してそれを PostgreSQL 上に実装し，これまで更新不可とされてきた結合演算を含むビューへの更新可能性をできうる限り向上させて，リレーショナルデータベースのユーザに資することに注力する．

【謝辞】本研究は JSPS 科研費 JP16K00152 の助成を受けたものである．

参　考　文　献

[1] E. F. Codd, “Recent Investigations in a Relational Database System,” Information Processing 74, pp. 1017-1021, North-Holland, 1974.

[2] U. Dayal and P. Bernstein, “On the Updatability of Relational Views,” Proc. 4th VLDB, pp.368-377, 1978.

[3] Y. Masunaga, “A Relational Database View Update Translation Mechanism,” Proc. 10th VLDB, pp.309-320, 1984.

[4] A. Sheth, J. Larson and E. Watkins, “TAILOR, A Tool for Updating Views,” LNCS, Vol. 303, pp.190-213, Springer, 1988.

[5] 増永良文，“更新意図の外形的推測に基づくリレーショナルデータベースビューの更新可能性,” 8p., WebDB Forum 2015 会議録，2015 年 11 月．

[6] Y. Masunaga, “An Intention-based Approach to the Updatability of Views in Relational Databases,” Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication (ACM IMCOM 2017), P5-8, Beppu, Japan, January 5-7, 2017.

[7] 増永 良文，長田 悠吾，石井 達夫，“バッグ意味論のもとでのビュー更新問題の検討－更新意図の外形的推測に基づくアプローチの適用可能性－,” DEIM Forum 2017 会議録, 3-5, 2017 年 3 月．

[8] Y. Nagata and Y. Masunaga, “Extending View Updatability by a Novel Theory -Prototype Implementation on PostgreSQL-,” PGCon 2017-The PostgreSQL Conference, Ottawa, Canada, May 25-26, 2017.

[9] 長田悠吾，石井達夫，増永良文，“意図に基づくアプローチによる更新可能ビューの拡張― PostgreSQL におけるプロトタイピング ―,” WebDB Forum 2017 会議録，DBS-165, No.14, 2017 年 9 月．

[10] Y. Masunaga, Y. Nagata and T. Ishii, “Extending the View Updatability of Relational Databases from Set Semantics to Bag Semantics and Its Implementation on PostgreSQL,” Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication (ACM IMCOM 2018), 8-3, Langkawi, Malaysia, January 5-7, 2018.

[11] 長田悠吾，増永良文，石井達夫，“意図に基づくアプローチのもとでの SQL ビューの更新可能性―ビュー更新判定問題の定式化と実問題への適用に向けた計算量削減の試み―,” DEIM Forum 2018 会議録, G1-4, 2018 年 3 月．