選択演算と射影演算に対応する双方向変換の関数従属性のもとでの整合性検査

M2021SC011 酒井聡太
指導教員：石原靖哲

1　はじめに

異なるデータベースを統合し，幅広く活用できるようにする，データ統合と呼ばれる技術が注目されている．データベース技術の分野において，共有先 (ターゲット) でデータが更新された場合，更新後のデータに対応するような共有元 (ソース) のデータが一意に定まらないという問題が古くから知られている．これに対し，近年，ターゲットの更新に対応するようなソースのデータを一意に定めるための戦略 (ビュー更新戦略) を定義できる，双方向変換と呼ばれる技術が注目されている [1].

BIRDS [1] と呼ばれる双方向変換フレームワークでは，関係データベースにおけるビュー更新戦略を記述するために，否定や組込み述語をもつ LVGN-Datalog [1] と呼ばれる問合せ言語が使用されている．ソースやターゲットが満たすべき制約も LVGN-Datalog で記述する．更に BIRDS では，双方向変換によって対応付けられたソースのデータとターゲットのデータのあらゆるペアが LVGN-Datalog によって記述された制約を満たす (整合性をもつ) かどうかを検査できる．

関係データベースでは，「ある属性の値が決まれば他の属性の値が一意に決まる」という意味の制約である，関数従属性がしばしば用いられる．例えば，学生に対して一意に振られている学生番号が決まれば，学生の名前が決まらなければならないという制約である．しかし，LVGN-Datalog では関数従属性を表現することができない．したがって，双方向変換によって対応付けられたソースのデータとターゲットのデータのあらゆるペアが，与えられた関数従属性を満たすかどうかを検査できないという課題がある [1].

本研究では，関係データベースにおける基本演算である，選択演算 (データベースからある条件に一致した行を取り出す演算) と射影演算 (データベースから指定した列を取り出す演算) に対応する双方向変換を対象とする．そして，それらの双方向変換によって対応付けられたソースのデータとターゲットのデータのあらゆるペアが，与えられた関数従属性を満たすための判定可能な必要十分条件を与えることを目的としている．

2　双方向変換の概要と整合性検査の必要性

双方向変換を例 1 に示す．

例 1　ある大学の学生に関するデータを全て含むソーステーブルと，そこからある部活を含むデータのみを取得するターゲットテーブルが定義されている場合を考える (図 1)．ソースの関係スキーマは S[INC] で構成される．これは S という名前のテーブルに，ID (I)，名前 (N)，部活 (C) のデータをもつことを表す．同様に，ターゲットの関係スキーマは T[INC] で構成される．ここで，テーブル S とテーブル T のどちらも属性 I の値を決めると，属性 N, C の値が一意に決まる制約である関数従属性 (I$\rightarrow$NC) を満たしているとする．テーブル T は，次の連言問合せで取得される．

$$T(\mathrm{I},\mathrm{N},\mathrm{C}) \text{ :- } S(\mathrm{I},\mathrm{N},\mathrm{C}), \mathrm{C} = \mathrm{Tennis} \tag{1}$$

これは，テーブル S から部活が Tennis である任意のデータを取り出し，テーブル T に ID，名前，部活のデータを格納することを表す．ユーザはターゲットに対する更新をどのようにソースへ反映するかを更新戦略として定義する．更新戦略は LVGN-Datalog によって定義でき，図 1 では次のように定義されている．

$$+S(\mathrm{I},\mathrm{N},\mathrm{C}) \text{ :-} T(\mathrm{I},\mathrm{N},\mathrm{C}), \neg S(\mathrm{I},\mathrm{N},\mathrm{C}) \tag{2}$$
$$-S(\mathrm{I},\mathrm{N},\mathrm{C}) \text{ :-} S(\mathrm{I},\mathrm{N},\mathrm{C}), \mathrm{C} = \mathrm{Tennis}, \neg T(\mathrm{I},\mathrm{N},\mathrm{C}) \tag{3}$$

(2) は，あるデータがテーブル T に存在しテーブル S に存在しない場合，そのデータをテーブル S に挿入することを表す．(3) は，部活が Tennis であるデータがテーブル S に存在しテーブル T に存在しない場合，そのデータをテーブル S から削除することを表す．テーブル T に ID が 4, 名前が David, 部活が Tennis であるデータを追加したとする．テーブル T の更新がテーブル S に反映され，テーブル S に ID が 3, 名前が David, 部活が Tennis であるデータが追加される．□

例 1 で双方向変換により得られた新しいソーステーブルは，関数従属性 I $\rightarrow$ NC を満たしている．しかし，新しいソーステーブルがいつも関数従属性を満たすとは限らない．

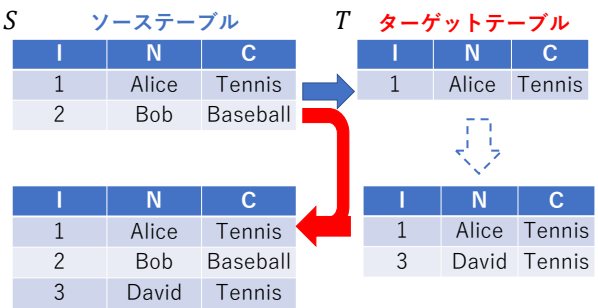

図 1　双方向変換の例

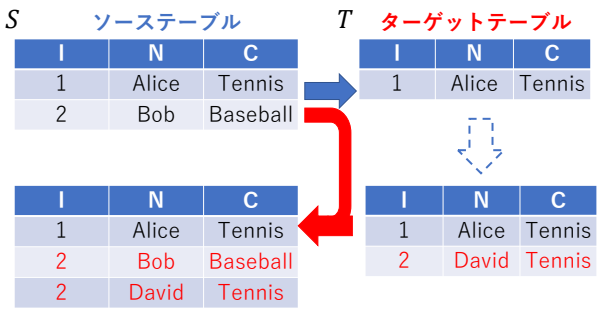

図 2　制約に違反する双方向変換の例

例 2　例 1 と同じ双方向変換を考える (図 2)．テーブル T に ID が 3, 名前が David, 部活が Tennis であるデータを追加する．テーブル T の更新がテーブル S に反映され，テーブル S に ID が 2, 名前が David, 部活が Tennis であるデータが追加されるが，図 2 の左下のような状況 (関数従属性を満たさないテーブルが得られてしまう状況) は許されない．□

例 2 のような状況はターゲットテーブルのロールバック (更新の取消処理) を引き起こす．したがって，ターゲットテーブルをどのように更新したとしても，それに対応するソーステーブルが関数従属性を必ず満たすことをあらかじめ保証できることが望ましい．そして，本研究で与える整合性検査手法を用いれば，その保証が可能になる．たとえば，上の例において，学生データベースとテニス部員データベースの間の双方向変換を決めた直後 (すなわち，実際にデータの共有を始める前) に，整合性検査によって，例 2 のような状況が発生しうるかどうかを判定すればよい．

3　諸定義

3.1　関係データベース

関係スキーマ $R[U]$ は関係データの構造を表すものであり，関係名 R と属性の有限集合 U から成る．値の可算集合 Dom を定義する．全域関数 $t : U \to \mathrm{Dom}$ を関係スキーマ $R[U]$ 上の**タプル**と呼ぶ．タプル t の定義域を $X \subset U$ に制限したタプルを $t[X]$ と書く．**関係インスタンス** (テーブル)I はタプルの有限集合である．

$R[U]$ における**原子式**は $R(x_1, \ldots, x_{|U|})$ と表せる．

定義 1 [関数従属性]　以下に関数従属性の定義を示す．

- 関係スキーマ $R[U]$ における関数従属性は $X \to Y (X, Y \subseteq U)$ の形で表される．すべてのタプル $t, t' \in I$ に対して，$t[X] = t'[X]$ ならば $t[Y] = t'[Y]$ である場合，$R[U]$ 上の関係インスタンス I は $X \to Y$ を満たすと言い，$I \models X \to Y$ と表記する．I が関数従属性の集合 Σ 中のすべての関数従属性を満たすとき，$I \models \Sigma$ と表記する．
- 関数従属性 $X \to Y$ は，Y が X の部分集合である場合，自明な関数従属性であるという．

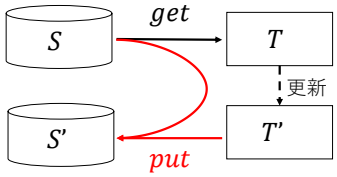

図 3　双方向変換

- Σ と Γ を関係スキーマ $R[U]$ における関数従属性の集合とする．$R[U]$ 上のすべての関係インスタンス I について「$I \models \Gamma$ ならば $I \models \Sigma$」が成立するとき，Γ は Σ を含意するといい，$\Gamma \models \Sigma$ と書く．

3.2　Datalog

Datalog プログラムは，以下の形をした規則の有限集合である．

$$H \text{ :- } L_1, \ldots, L_n \quad (H, L_1, \ldots, L_n \text{ は原子式})$$

H を**頭部**，$L_1, \ldots, L_n$ を**体部**と呼ぶ．Datalog は，体部に否定, 等式 (=), 比較 (<, >) のような組込み述語を受け入れ，否定される原子式または組込み述語に現れる各変数が，否定のない原子式にも現れなければならないという条件の下で拡張することができる．これが [1] で提案された LVGN-Datalog である．

3.3　双方向変換

双方向変換とは，2 つの情報源 (ソースとターゲット) の間で変換を介して一貫性を保つ仕組みである．図 3 のように，双方向変換はソースをターゲットに変換する**順方向変換** (get) と，ターゲットをソースに変換する**逆方向変換** (put) からなる．双方向変換における順方向変換はソース S を引数にとりターゲット T を返す get 関数で表され，逆方向変換はソース S とターゲット T を引数にとり対応するソースを返す put 関数で表される．ここで，次の式に示す特性が満たされる場合，これらの変換は well-behaved であるという．

$$put(S, get(S)) = S \quad \text{(GETPUT 特性)}$$

$$get(put(S, T')) = T' \quad \text{(PUTGET 特性)}$$

GETPUT 特性は，ターゲットにおいて更新が行われなかった場合，ソースにおいても更新が行われないことを表している．一方 PUTGET 特性は，全てのターゲットの情報を完全にソースへ変換できることを表している．

3.4　整合性の定義

Σ_S, Σ_T をそれぞれソースとターゲットの関係スキーマ上の関数従属性の集合とする．Σ_S を満たす任意のテーブル S と，get の値域に属しかつ Σ_T を満たす任意のテーブル T について，$put(S, T')$ が Σ_S を満たす時，双方向変換が Σ_S と Σ_T に対して整合性をもつという．

4 *get* が選択演算に相当する場合

get が選択演算に相当するという前提のもとで，双方向変換が整合性をもつための必要十分条件は，以下の 2 条件が成立することである．ここで，Σ_S と Σ_T をそれぞれソーススキーマとターゲットスキーマの関数従属性の集合と定義する．また，$get(S)$ は関係代数の選択演算に相当し，$put(S, T') = (S - get(S)) \cup T'$ を満たすとする．

- Σ_S が含意する任意の自明ではない関数従属性の左辺に現れる属性のみが *get* で使用される．
- Σ_T が Σ_S を含意する．

紙面の都合上，証明は省略する．

5 *get* が射影演算に相当する場合

5.1 例を用いた検討

get が射影演算に対応する双方向変換が関数従属性のもとでの整合性をもつ例を示す．

例 3 次の関係スキーマ，関数従属性，双方向変換が与えられたとする．

- $\mathrm{R}_S = \mathrm{S[ERM]}$,
- $\Sigma_S = \{\mathrm{E} \to \mathrm{RM}\}$,
- $\mathrm{R}_T = \mathrm{T[ER]}$,
- $\Sigma_T = \{\mathrm{E} \to \mathrm{R}\}$,
- *get* 定義：
  $\mathrm{T(E, R)}$:- $\mathrm{S(E, R, M)}$
- *put* 定義：
  $+\mathrm{S(E, R, M)}$:- $\mathrm{T(E, R)}, \neg\mathrm{S(E, R, \_)}, \mathrm{M} =$ ‘unknown’
  $-\mathrm{S(E, R, M)}$:- $\mathrm{S(E, R, M)}, \neg\mathrm{T(E, R)}$

あるソーステーブルと，そこから属性 E, R の任意のデータを射影したターゲットテーブルが定義されている場合を考える (図 4)．ターゲットテーブルで E = Carol, R = R303 の行が追加されると Datalog プログラムによって，新しいソーステーブルを得る．新しいソーステーブルは Σ_S を満たしている．

例として，図 4 のようなソーステーブルを挙げたが，あらゆるソーステーブルについて，Σ_T を満たすターゲットの更新がされた場合，どのような更新でも反映されたソーステーブルは Σ_S を満たす． □

get が射影演算に対応する双方向変換が関数従属性のもとでの整合性をもたない例を示す．

例 4 次の関係スキーマ，関数従属性，双方向変換が与えられたとする．

- $\mathrm{R}_S = \mathrm{S[ERM]}$,
- $\Sigma_S = \{\mathrm{E} \to \mathrm{RM}\}$,
- $\mathrm{R}_T = \mathrm{T[RM]}$,
- $\Sigma_T = \{\mathrm{R} \to \mathrm{M}\}$,
- *get* 定義：

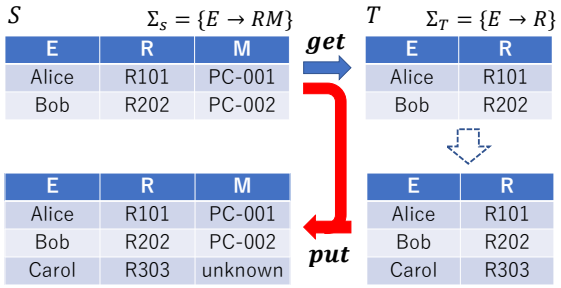

図 4 双方向変換が関数従属性のもとで整合性をもつ例

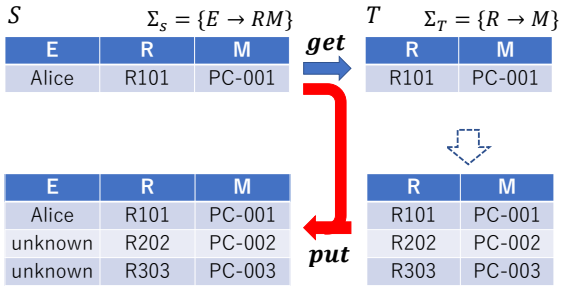

図 5 双方向変換が関数従属性のもとで整合性をもたない例

$\mathrm{T(R, M)}$:- $\mathrm{S(E, R, M)}$

- *put* 定義：
  $+\mathrm{S(E, R, M)}$:- $\mathrm{T(R, M)}, \neg\mathrm{S(\_, R, M)}$,
  $\mathrm{E} =$ ‘unknown’
  $-\mathrm{S(E, R, M)}$:- $\mathrm{S(E, R, M)}, \neg\mathrm{T(R, M)}$

あるソーステーブルと，そこから属性 R, M の任意のデータを射影したターゲットテーブルが定義されている場合を考える (図 5)．ターゲットテーブルで R = R202, M = PC-002 と R = R303, M = PC-003 の行が追加されると Datalog プログラムによって，新しいソーステーブルを得る．新しいソーステーブルは Σ_S を満たさない．よって，この双方向変換は Σ_S と Σ_T に対して整合性をもたない． □

いくつかの例を確認し，*get* が射影演算に相当するという前提のもとで，整合性をもつための必要十分条件は，以下の 2 条件が成立することであると予想した．ここで，U をソーススキーマの属性の全体集合，W を射影で使用される属性集合と定義する．

- U の任意の部分集合 X と W に属する任意の属性 A に対し，ソーススキーマが満たす関数従属性の集合が $X \to A$ を含意しかつ A が X に属さないならば，ターゲットスキーマが満たす関数従属性の集合が $X \cap W \to A$ を含意する．
- U の任意の部分集合 X と射影で使用されない属性集合 $U - W$ に属する任意の属性 A に対し，ソーススス

キーマが満たす関数従属性の集合が $X \to A$ を含意しかつ A が X に属さないならば，ターゲットスキーマが満たす関数従属性の集合が $X \cap W \to W$ を含意する．

5.2 必要十分条件であることの証明

例を用いた検討により立てた予想が正しいことを示す．紙面の都合で定理 2 の証明のみ記載する．

定理 1 Σ_S と Σ_T をそれぞれソーススキーマとターゲットスキーマの関数従属性の集合と定義する．W を射影で使用する属性集合とする．U をソーススキーマの属性の全体集合とする．$get(S)$ は関係代数の射影演算に相当し，

$$\begin{aligned} put(S,T') =& (S - \{t \in S | t[W] \notin T'\}) \cup \\ & \{t' | t'[W] \in T', (\forall t \in S, t[W] \neq t'[W]), \\ & (\forall A \in U - W, t'[A] = \text{`unknown'}\} \end{aligned}$$

を満たすとする．

以下の 2 条件が成立するならば，Σ_S を満たす任意の S と Σ_T を満たす任意の T' について $put(S,T')$ は Σ を満たす．

- 任意の $X \subseteq U$ と任意の $A \in W$ に対し「$\Sigma_S \models X \to A$ かつ $A \notin X$ ならば $\Sigma_T \models X \cap W \to A$」
- 任意の $X \subseteq U$ と任意の $A \in U - W$ に対し「$\Sigma_S \models X \to A$ かつ $A \notin X$ ならば $\Sigma_T \models X \cap W \to W$」

定理 2 Σ_S と Σ_T をそれぞれソーススキーマとターゲットスキーマの関数従属性の集合と定義する．W を射影で使用する属性とする．U をソーススキーマの属性の全体集合とする．$get(S)$ は関係代数の射影演算に相当し，

$$\begin{aligned} put(S,T') =& (S - \{t \in S | t[W] \notin T'\}) \cup \\ & \{t' | t'[W] \in T', (\forall t \in S, t[W] \neq t'[W]), \\ & (\forall A \in U - W, t'[A] = \text{`unknown'}\} \end{aligned}$$

を満たすとする．Σ_S を満たす任意の S と Σ_T を満たす任意の T' について $put(S,T')$ が Σ_S を満たすならば，以下の 2 条件が成立する．

- 任意の $X \subseteq U$ と任意の $A \in W$ に対し「$\Sigma_S \models X \to A$ かつ $A \notin X$ ならば $\Sigma_T \models X \cap W \to A$」
- 任意の $X \subseteq U$ と任意の $A \in U - W$ に対し「$\Sigma_S \models X \to A$ かつ $A \notin X$ ならば $\Sigma_T \models X \cap W \to W$」

証明 対偶を示す．以下, 仮定条件に応じて場合分けする．

- $\Sigma_S \models X \to A$ かつ $A \notin X$ かつ $\Sigma_T \not\models X \cap W \to A$ である $X \subseteq U$ と $A \in W$ が存在すると仮定する．$\{u, u'\} \models \Sigma_T$ かつ $u[X \cap W] = u'[X \cap W]$ かつ $u[A] \neq u'[A]$ を満たすターゲットスキーマ W 上のタプル $\{u, u'\}$ が存在する．$t[W] = u$, $t'[W] = u'$, $\forall D \in U - W, t[D] = t'[D] =$ 'unknown' であるようなタプル t, t' を考える．$S = \phi$, $T' = \{u, u'\}$ とおくと $put(S,T') = \{t, t'\}$ である．$\{t, t'\} \not\models X \to A$ より，$put(S,T')$ は Σ_S を満たさない (図 6).

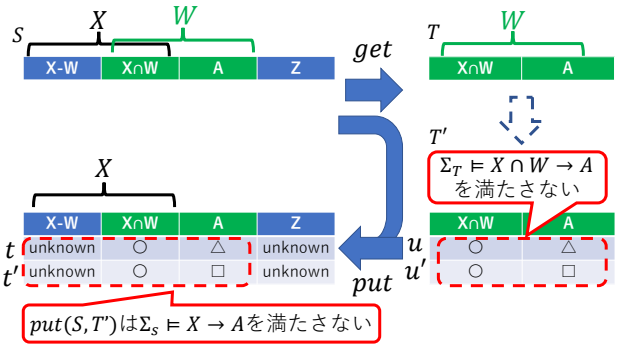

図 6 定理 2 の前半の証明

- $\Sigma_S \models X \to A$ かつ $A \notin X$ かつ $\Sigma_T \not\models X \cap W \to W$ のような $X \subseteq U$ と $A \in U - W$ が存在すると仮定する．$\{u, u'\} \models \Sigma_T$ かつ $u[X \cap W] = u'[X \cap W]$ かつ $u \neq u'$ を満たすターゲットスキーマ W 上のタプル $\{u, u'\}$ が存在する．$t[W] = u$, $t'[W] = u'$ かつ $t[A] \neq t'[A]$, $\forall D \in U - W A, t[D] = t'[D] =$ 'unknown' であるようなタプル t, t' を考える．$S = \{t\}$, $T' = \{u, u'\}$ とおくと $put(S,T') = \{t, t'\}$ である．$\{t, t'\} \not\models X \to A$ より，$put(S,T')$ は Σ_S を満たさない (図 6 と類似のため図は省略).

□

6 おわりに

本研究では，get が選択演算および射影演算に対応する前提のもとで，双方向変換が関数従属性のもとでの整合性をもつための必要十分条件を与えた．これらの必要十分条件は判定可能であることから，選択演算と射影演算に対応する双方向変換によって対応付けられたあらゆるソースとターゲットのペアが, 与えられた関数従属性を満たすかを検査できることを示した．

今後の課題は，LVGN-Datalog で記述されたあらゆる双方向変換が関数従属性のもとで整合性をもつための必要十分条件を示すことである．そのためのアプローチとして，tuple generating dependencies (tgd) と呼ばれる変換が関数従属性に対して整合性をもつかの判定法を提案した [2] の手法を拡張することが有望であると予想している．

参考文献

[1] Van-Dang Tran, Hiroyuki Kato, and Zhenjiang Hu. Programmable view update strategies on relations. *Proceedings of the VLDB Endowment*, Vol. 13, No. 5, pp. 726–739, 2020.

[2] Yasunori Ishihara, Takashi Hayata, and Toru Fujiwara. The absolute consistency problem for relational schema mappings with functional dependencies. *IEICE Transactions on Information and Systems*, Vol. E103-D, No. 11, pp. 2278–2288, 2020.