6G－8

オブジェクト指向データベースプログラムにおける型検査のアクセス制御への応用

泉野 博明　石原 靖哲　関 浩之　伊藤実
奈良先端科学技術大学院大学 情報科学研究科

1 はじめに

アクセス制御法はデータベースのセキュリティ保全のために必須の技術である．オブジェクト指向データベース(OODB)では，一定のアクセス操作列をメソッドとしてカプセル化可能であるので，OODBのアクセス制御を行なう際には，組み込み演算の他にメソッド単位のアクセス権をも考慮する必要がある．[2]

本稿ではHullらの提案したOODBの操作モデル[3]を用いて，OODBにおけるアクセス制御の形式化を試みる．Hullらのモデルは，メソッドの引数が一つ（モナディック）である手続き型モデルであり，データベースの更新は属性への代入文として簡潔に表現することができる．また，従来の研究では，ユーザプログラムの正当性（アクセス権違反を行なわないこと）はプログラムの実行時にチェックすることを前提としていたが，本稿では，与えられたユーザプログラムの正当性を静的に（コンパイル時に）解析する方法について考察する．

2 問題の定式化

2.1 Hull-田中-吉川のモデル

スキーマとは4字組 $S=(C,A,M,Impl)$ である．ここで C はクラス名の半順序有限集合，A は属性宣言の有限集合，M はメソッド名の有限集合，$Impl$ はメソッド定義の有限集合である．ここで，$A(c,a)=\{c_1,c_2,...,c_n\}$ のとき，「クラス c のオブジェクトの属性 a は，クラス $c_1,c_2,...,c_n$ のオブジェクトである」こと，$A(c,a)=\{B\}$ は，「クラス c のオブジェクトの属性 a は，基本値である」ことを表す．また，メソッド定義とは次の7種の文の系列である．

$y := x$,
$y := self$,
$y := self.a$（self属性 a の参照），
$y := m(x)$（メソッドの呼出し），
$y := \theta(y_1,y_2,...,y_m)$（基本値に対する組込み演算），
$self.a := x$（self属性 a の更新），
$return(x)$.

ただし，$x,y,y_1,y_2,...,y_m$ は変数，a は属性，$self$ はselfオブジェクト，すなわちメソッドが呼び出されたオブジェクト，θ は基本値に対する組込み演算を表す．詳細は省略するが，クラス c に継承されたメソッド m の定義を，$Impl^*(m,c)$ と表す．

例えば，構成員という属性値を読み込み，メソッド $count$ によって人数を算出し，人数数という属性にその値を書き込んでその値を返すメソッドの例を下に示す．

$x := self.$構成員
$y := count(x)$
$self.$人数数$:= y$
$return(y)$

スキーマ S に対するインスタンスとは，S における各クラスへのオブジェクト集合の割当てと，各オブジェクトの属性値（オブジェクトまたは基本値）の指定である．

さらにHullらは以下のような3つのスキーマのクラスを定義している．

スキーマのクラス	実行文の種類	実行文の依存関係
simple retrieval schema	・self, self属性の参照 ・メソッドの呼出 ・基本値に対する組込み演算	self, self属性に対してメソッドの呼出を高々1回行なった後，基本演算を高々1回行なう
retrieval schema	同上	任意
update schema	同上 ＋ self属性の更新	任意

本稿ではsimple retrieval schemaを単純検索スキーマ，retrieval schemaを検索スキーマ，update schemaを更新スキーマと訳す．

2.2 正当性判定問題

スキーマ S の正当性，すなわち，S がアクセス権違反を行なわないことを次のように定義する．まず，アクセス権を，有限集合 $AP=\{(u_1,m_1,c_1),...,(u_n,m_n,c_n)\}$（$u_i$ はユーザ，m_i は S の中のメソッド，c_i は S の中のクラス）で表す．$(u,m,c)\in AP$ は，u が c のオブジェクトに対して m を呼び出す権利を持つという意味である．

S の任意のインスタンス I に対して，次の条件が成り立つとき，(S,u) は AP に対して正当性をもつ，という．

An Application of Type-Checking to Access Control in Object-Oriented Database Programs
Hiroaki Izuno, Yasunori Ishihara, Hiroyuki Seki, and Minoru Ito
Nara Institute of Science and Technology

条件　Iのもとで, Sで定義されたメソッドを実行中に, クラスcのオブジェクトに対して, メソッドmが呼び出されるならば,

$$(u, m, c) \in AP$$

3　正当性判定アルゴリズム

スキーマS, ユーザu, アクセス権APが与えられたとき, (S, u)がAPに対して正当性をもつかどうかを調べるためには, S中の文$y := m(x)$において, どのクラスのオブジェクトがxに代入されるかを調べればよい（型検査）. 既にHullらによって, 単純検索スキーマに対する型検査問題は決定可能, 更新スキーマに対しては決定不能であることが示されている[3]. また, メソッドスキーマがモナディック（引数がたかだか一つ）の場合は, 型検査問題は多項式時間可解[4]であることが示されている. ここでは, その手法を用いることによって検索スキーマの正当性問題を多項式時間で判定するアルゴリズムを示す.

3.1　準備

アルゴリズムでは, 以下の変数を用いる.

$T(m, c, s_i)$: $Impl^*(m, c)$ 中の第i番目の実行文をs_i : $y := \alpha$とするとき, 左辺yに代入されるオブジェクトの属するクラスを記憶する.

$T(m, c)$: $Impl^*(m, c)$の返り値となるオブジェクトの属するクラスを記憶する.

3.2　アルゴリズム

文$s_i : y := x, y := m(x), return(x)$では, 変数$x$の値を参照している. その値を$x$に代入している文を$d(s_i)$で表す.

文$s_i : y := \theta(x_1, x_2, ..., x_m)$に対しても同様に, s_iで参照している$x_j (1 \leq j \leq m)$の値を代入している文を$d_j(s_i)$で表す.

アルゴリズム V^*
入力
S:検索スキーマ, u:ユーザ, AP:アクセス権
repeat
for each $m \in M$ **and** $c \in C$
such that $(u, m, c) \in AP$かつ $Impl^*(m, c)$が定義されている **do**
$Impl^*(m, c) = s_1, s_2, ..., s_n$とする
for $i := 1$ **to** n **do**
switch
s_iが $y := self$: $T(m, c, s_i) := \{c\}$;
s_iが $y := self.a$: $T(m, c, s_i) := A(c, a)$;
s_iが $y := x$: $T(m, c, s_i) := T(m, c, d(s_i))$;
s_iが $return(x)$: $T(m, c) := T(m, c, d(s_i))$;
s_iが $y := \theta(x_1, x_2, ..., x_m)$
for $j := 1$ **to** m **do**
if $T(m, c, d_j(s_i))$が（B以外の）クラスを含む
then
"(m, c, s_i)でタイプエラー"と表示
else $T(m, c, s_i) := \{B\}$;
s_iが $y := m'(x)$:
for each $c' \in T(m, c, d(s_i))$ **do**
if $Impl^*(m', c')$が未定義 **then**
"(m, c, s_i)で未定義メソッドの呼出し"と表示
else if $(u, m', c') \notin AP$ **then**
"(m, c, s_i)でアクセス権違反"と表示
else $T(m, c, s_i)$に$T(m', c')$を追加;
until $T(m, c)$が一つも変化しない;
アクセス権違反が検出されなければ,
"(S, u)はAPに対して正当"と表示

定理　アルゴリズムV^*は, 与えられた検索スキーマS, ユーザu, アクセス権APに対して, (S, u)がAPに対して正当性をもつかどうかを, 入力の記述長の多項式時間で判定する. □

4　おわりに

更新スキーマにおいては, 一般に, オブジェクトoに対して実行中のメソッドmからメソッドm'を呼出したときに, m'の実行中にo自身の属性値が更新される可能性があり, 型検査は困難となる（一般には決定不能）.

このような更新が起こらないために, 同一の変数xに対して$y := m(x), z := m'(x)$のように複数のメソッドを呼び出さないという制限を設ける. さらに, インスタンスはサイクルなしであると仮定する. ここで, サイクルなしのインスタンスとは, どのオブジェクトoからも, 属性値を1回以上たどってo自身に到達できないようなインスタンスである. このような「制限された更新を許す更新スキーマの部分クラス」においては型検査問題は多項式時間可解であると予想され, 現在証明中である.

また, 3.2のアルゴリズムを一部変更することにより, 一般の更新スキーマに対する正当性問題の十分条件の判定アルゴリズムとしても利用可能である.

参考文献

[1] S.Abiteboul, et al. :"Method Schemas", Proc. 9th ACM PODS, 16-27(1990).

[2] R.Ahad, et al. : "Supporting Access Control in an Object-Oriented Database Language", Proc. 3rd Intl. Conf. Extending Database Technology, LNCS 580, 184-200(1992).

[3] R.Hull, K.Tanaka, M.Yoshikawa : "Behavior Analysis of Object-Oriented Databases : Method Structure, Execution Trees, and Reachability", Proc. 3rd Intl. Conf. on Foundations of Data Organization and Algorithms, 372-388 (June 1989).

[4] 百々, 石原, 関 : "メソッドスキーマにおける型整合性の解析アルゴリズム", 情処学研報, 94-PRG-15-5(1994-01).