数理解析研究所講究録
第461巻 1982年 187-205

一般化された利用権制御方式について

京大　工学部　　上林　弥彦

1. まえがき

利用者が複数であるシステムでは、安全性の確保が非常に重要であり、利用者識別、利用権制御、機密性の高い通信、データの利用の監視等の諸側面より種々の研究がなされている。本論文は、データベースの利用権制御の方式に関するものである。

利用権制御機構は、対象物（ファイルやプログラム）の利用権を利用者に割当てるための方法である。これには、次の2つの分類が知られている。

{ 固定的 / 動的 }　　{ 集中的 / 非集中的 }

固定的な場合は、ファイルやプログラムが定義されるごとに、各利用者に対する利用権が設定され、途中での変更は容易ではない。しかし、ファイルやプログラム自体が、時間と共に変化していくものであるため、それに対処して、利用者の持

つ利用権も変化させる動的な方式の方が適用範囲が広いと考えられる。集中的と非集中的の違いは、これらの利用権の設定を行なう権限を有するグループの違いによる。集中的なものでは、利用権の設定変更は、ファイルやプログラムの作成者やシステムの安全性管理者またはデータベース管理者が行なう。利用者がこのような利用権の設定変更を行なえるものを非集中的であるという。動的かつ非集中的な利用権制御機構は、ファイルやプログラム（以下ではファイルのみを考える）さらに利用者集合の動的な変化にも対処しうるので、非常に有効であると考えられる。

Griffiths とWade〔GRIFW7609〕は、ある動的かつ非集中的な利用権制御機構を考察し、Fagin〔FAGI7809〕はこのアルゴリズムの誤りを訂正した。本論文では、彼らによって開発された方式をGWF機構と呼ぶことにする。GWF機構では、利用権だけでなく利用権授与権も利用者に与えられる。このような自由度は時刻印（time stamp）という概念を導入することにより可能となったが、次のような問題点もある。

(1) ある利用者から別のある利用者への利用権の授与を何度も行なうことが許されていなければアルゴリズムの正しさは保障されない〔FAGI7809〕。この結果、ある利用者が自分の与えた利用権の授与を無効とした場合の影響を計算するグラ

フは多くの枝を有することとなり、この計算が複雑となる。

(2) 時刻印は歴史を記憶する作用を持つので古い利用者ほど大きな権限を持つようになる傾向がある。たとえば、すべての利用権は最初はファイルの作成者が与えるので、作成者が自分の与えた利用権の授与をすべて無効とした場合は、作成者以外は利用権を失ってしまう。ファイルの内容は動的に変化するので、利用権を先に得たものの方が権限が大きくなるのは問題がある。

(3) ファイルやプログラムにはいくつかの安全性のレベルがある。たとえば、極秘, 秘密, 取扱い注意等である。GWF機構では、これらのレベルの扱いができず、すべてのファイルは同等に扱われる。

2節では、(1)の問題について考察し、2つの方法を与えている。ひとつの方法は、グラフの等価性のもとでの簡約化であり、他の方法は授与される利用権として新しい形のものを考えたものである。

(2)の問題は、ファイルの内容は全く自由に変化しうることから、もとのファイルの作成者の作った部分がなくなっても、権限を持ち続けるということが起こるのでこれを考えなおすべきであるという主張が動機となっている。利用者集合は、全く受身のデータと異っているため、完全に自由に変化

しうるようにはできないと考えられるが、自由度を高めることは重要といえる。ファイルの共有者という概念の導入はひとつの解決法となるであろう。

3節では、(3)の問題を扱うために、多数決利用権制御機構を導入する。この方式によれば、(2)の問題もある程度解決することができる。個々のファイルの安全性に応じて整数のしきい値が決められており、その人数以上の利用権を授与することができるものである。安全性のレベルの高いものには高いしきい値が割り当てられる。この方式の問題点は、ある利用者が与えた利用権の授与を無効とした場合の影響の計算量と考えられる。利用権の授与は、時刻印の付いた関数従属性として扱うことができ、このことから効率の良い計算法が存在することが示される。

4節では、この方法の一般化について考えている。これには、利用権授与方式の一般化(利用者に対する重み付け、Yes, Noの他にPassを許す等)、利用権の授与を無効にする部分に対する多数決方式の導入、一部の利用権授与の自動化(あるファイル集合の利用権を持てばあるファイル集合に対する利用権が自動的に与えられる)、抽象的な利用権の設定(実際に存在しないで計算によって得られるファイルの利用権や共助的利用権〔MINS8109〕)等を考えている。

2. GWF機構とその一般化

IBMのSystem Rのための高水準利用権制御方式として考察されたGWF機構は次のようなものである。

(1) どの利用者でもファイルを作ることができる。

(2) 利用者が自分のファイルを他の利用者と共有しようと思うときは、GRANT命令によって他の利用者にREAD, INSERT, DELETE, UPDATE, DROPという諸権限の一部または全部を与えることができる。

(3) さらに、利用権授与権付の利用権を与えることもでき、この権限を有していると、他の利用者に自分の持つ権限(利用権授与権も含む)を与えることができる。

(4) 与えた権限は、REVOKE命令を発することによりいつでも取りやめることができる。

1つの重要な問題は、REVOKE命令によって影響される部分を計算する点にある。利用者xの利用権がなくなると、xより利用権を得たすべての利用者は利用権を失う可能性がある。この計算のためGWFグラフを導入する。

定義1：GWFグラフ $G_{GWF} = (V, E, v_1)$ は次のようなものである。$V = \{v_1, v_2, \cdots, v_m\}$ は節点集合，$E = \{e_{ij}(t)\}$ は枝集合で、$e_{ij}(t)$ は v_i より v_j へ向かう重みtの枝を示す。tは整数である。v_1 は初期節点で，$v_1 \in V$ である。

簡単のため、1つのファイルと1つの権限に対して1つのGWFグラフが定義されていると仮定する。実際にはいくつかのGWFグラフを併合して扱うことが可能である。節点は利用権ないしは利用権授与権を有する利用者に対応している。ファイル作成者はv_1である。時刻tに利用者v_iが利用者v_jに利用権を与えると$e_{ij}(t)$という枝がグラフに付加される。利用権のみの授与の扱いは単純なので、以下では利用権授与権付の利用権のみを考える。時刻印tはシステムで共通に用いられ、必要のある場合のみ1だけ増加する。時刻tで権限を得た利用者は、時刻t'($\geq t+1$)にこの権限を他の利用者に与えることができる。利用者v_jに対して与えた利用権を無効にした場合、v_jに向かうすべての枝が取り去られる。さらにこの結果として、つぎの条件を満足する経路$e_{k_1k_2}(t_1)\ e_{k_2k_3}(t)\cdots e_{k_hp}(t_h)$のない枝$e_{pq}(t)$も除かれる。

$v_{k_1}=v_1$.

$t_1<t_2<\cdots<t_h<t$.

Faginは、ある利用権を無効にした場合に無効になる利用権集合を正しく計算するためには、v_iよりv_jへ向かう枝が複数本存在することを許さなければならないことを示した。図1にFaginの例が示されている。この図で利用者2が3に与えた利用権を取り消すと太線で示した利用権授与のみが有効

となる。しかし、利用者3より4に向かう2本の枝のうちどちらかを消すと、このような正しい結果が得られない。

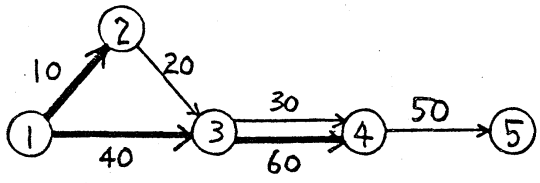

図1. Fagin の例

この性質はGWFグラフにいくらでも枝を許すこととなるので、計算を非常に複雑化させる可能性がある。次の命題は冗長性の条件を示したものである。

命題1：枝$e_{ij}(t)$は次の条件AかBが成立すれば冗長である。

A：(1) $j \neq 1$であり

(2) $t_1 < t$を満足する$e_{ij}(t_1)$があり、

(3) $t_1 \leqq t_2 < t$を満足する枝$e_{hi}(t_2)$がない。

B：$j = 1$

(証明)利用者i($\neq 1$)が$e_{ij}(t_1)$を出してから新たに権限を受けていなければ$e_{ij}(t)$はどのような無効命令による権限保持者集合も変化させないのでAが言える。1は常に権限を有しているのでBが言える。

命題2：$t_1 < t_2$で$e_{ij}(t_1)$と$e_{ij}(t_2)$という2つの枝が存在したとする。

(1) $i=1$ ならば $e_{ij}(t_2)$ は冗長である。

(2) $i \neq 1$ で $t_1 < t_3 \leq t_2$ を満足する $e_{jk}(t_3)$ がどのような k に対しても存在しなければ、$e_{ij}(t_1)$ は冗長である。

(証明) (1) は自明である。GWF機構では、枝を無効にする場合に、そとまの指定しかできず時刻印の指定はできない。このため、$e_{ij}(t_2)$ のみを除くことはできず、1つの無効命令で両方とも除かれてしまう。この両方が存在するときはいつも $e_{ij}(t_1)$ が冗長であるため、この枝を除いておくことができる。

上記の2つの命題を満足するような冗長枝が生じないようにGWFグラフを作っていっても、サイクルが生じる場合には枝の数はいくらでも増加する可能性がある。

定義2：GWFグラフにおいて、次の条件を満足する経路 $e_{k_1k_2}(t_1)\, e_{k_2k_3}(t_2) \cdots e_{k_{p-1}k_p}(t_{p-1})$ は有効であると言われる。

(1) $k_1 = 1$

(2) $t_1 < t_2 < \cdots < t_{p-1}$

(3) k_i より k_{i+1} ($i=1, \cdots, p-1$) に複数の枝があると $e_{k_{i+1}k_{i+2}}(t_{i+1})$ は、$t_i < t_{i+1}$ を満足するうちで最小の t_{i+1} となっている。

(4) すべての $i \neq j$ に対し、$k_i \neq k_j$

命題3：どのような有効結路にも含まれない枝 $e_{ij}(t)$ は冗長

である。

（証明）どの有効経路にも含まれない枝はその時点で冗長である。これがグラフの変化により有効経路に含まれるようなことが起こらないことを言えば良い。消去命令を考えなければ、有効経路に一度含まれないと決まった枝は永久に含まれるようにならない。$v \neq u$で、$e_{ij}(t')$が有効経路に含まれているとする。この経路内の枝を消去した場合、$e_{ij}(t')$は有効経路に含まれなくなる可能性があるが、$e_{ij}(t)$が含まれるようにはならない。このようなvがない場合、$e_{ij}(t)$がでよりまへ向かう唯一の枝となる。これが有効経路に含まれないのは、vへ至る有効経路がすべてuを通っている場合で、消去命令によっても$e_{ij}(t)$が有効経路に含まれることは起こりえない。

命題3は命題1や2より強力である。すべての有効経路を求めるには、有限オートマトンの正規集合を求めるアルゴリズムを若干変更すればよい。有効経路に含まれる枝を除くことはできないため、命題3はGUF機構に対する最善の冗長条件となる。

命題4：n人の利用者がいる場合、有効閉路に含まれる枝の数の上限は$\sum_{k=1}^{n-1}\frac{(n-1)!}{k!}$であり、任意の$n \geq 1$に対しその条件を満足するグラフが存在する。

（証明）有効閉路はv_1を始点としてその長さは$n-1$以下である。v_1よりは$n-1$個の節点に直接到達でき　それぞれの節点ではそれを始点とする長さ$n-2$の有効閉路を考えることとなる。$f(n)$を枝数とすると、

$$f(n)=(n-1)+(n-1)f(n-1)$$

$$f(2)=1 \qquad f(1)=0$$

この$f(n)$は上の値となる。次にこの枝に適当にラベルを付けるとすべての枝が有効経路に含まれることを言う。一般的な証明の代わりに、$n=4$の場合を図2に示す。

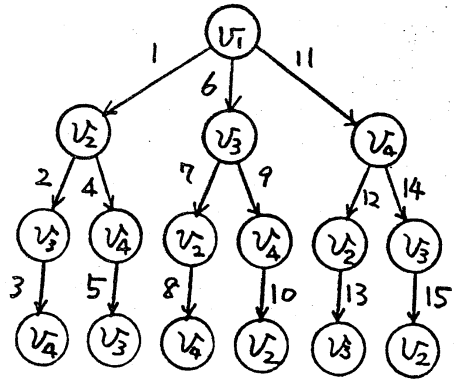

図2．最大数の枝を含む有効経路集合(n=4)

このため、GWF機構に対し枝数が増加しないように一般化する方法を考えなくてはならない。

最初の方法はラベルの変更による方法である。

(1) v_i より v_j へむかう枝が複数本あれば最も時刻印の大きな枝のみを残し他は消し去る。

(2) 上の操作で消される有効経路は、ラベルを変更して有効とさせる。この場合定義2を変更しても $t_1 \leqq t_2 \leqq \cdots$ とする必要がある。

図3(a)において枝 $e_{23(10)}$ を取り去ると図3(b)のようになる。

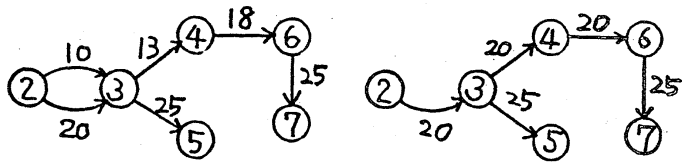

図3. ラベルの変更

この方法の1つの問題は、有効経路集合を常に調べておかなければいけない点にある。

もう1つの方法は、新しいタイプの枝を導入する方法である。利用者ごとに複数回利用権を与えるのは、それが自分が利用権を有しているときは必ずしも利用権を有するということを希望していると考え、枝に＊というラベルを導入する。$e_{ij}(t)$ に＊が付けられていると、これはすべての $t' \geqq t$ に対し $e_{ij}(t')$ が存在するのと等価となる。この枝の表示は $e_{ij}(-t)$ で表すものとする。このようにして、枝数を n^2 のオーダに下

げることができる。この場合、消去命令の影響は次のように計算できる。

(1) $e_{kg}(-t)$が消去命令の対象となった場合は、この枝を除く。

(2) 時刻tで利用者eが権限を失なうと、eより出ている枝$e_{ig}(-t_i)$は$e_{ig}(-t)$に置き換えられる。

3. 多数決利用権制御機構

機密性のレベルを扱うため本節では多数決利用権制御機構を導入する。利用権付与の操作を増加させないため、前節の2番目の方法(＊印を付ける)を採用する。

定義3：多数決利用権制御機構は次のように定義される。

$$A_T(F, P) = (U, G, U_0, Q)$$

これはファイル集合Fと権限集合Pに対して定義されるもので、$U = \{u_1, \ldots, u_n\}$は利用者集合、$G = \{g_1, \ldots, g_n\}$は権限付与操作の集合で各g_iは$U_i \to u\{t, w\}$で示される。ここで、$U_i (\subseteq U)$は利用者集合、uは利用者、tは時刻印、wはr, g, r*, g*のいずれかである。rは権限付与、gは委譲権付権限付与である。＊は前節で示したようにt以後のすべての時刻印に対応する権限付与操作の集合とみなすことができるものである。U_0はFの共有者の集合である。$Q = \{g_r, g_g\}$はしきい値集合である。ここで、g_rは権限付与のた

め、g_gは委譲権付権限付与のためのしきい値である。

たとえば、g_1：$\{u_1, u_2\} \rightarrow u_3[t, g^*]$は、$u_1$と$u_2$が共同で時刻$t$に$u_3$に権限Pとその権限委譲権を与えることを示している。左辺の集合の数はQで決められる。Qの要素に対し、$g_t \leqq g_g$が成立しなければならない。機密性のレベルが高いものには大きなg_tとg_gが定められる（利用者集合がこの値より小さいときは、全利用者が共同するとよいという形で一般化することもできる）。

例1：$U=\{u_1, u_2, u_3, u_4, u_5\}$，$G=\{g_1, g_2, g_3\}$，$U_0=\{u_1, u_2\}$，$Q=\{2, 2\}$とし、Gの要素は次の形で与えられるとする。

g_1：$u_1\ u_2 \rightarrow u_3[10, g]$

g_2：$u_1\ u_2 \rightarrow u_4[10, t]$

g_3：$u_2\ u_3 \rightarrow u_4[20, g]$

委譲権付権限を持つ利用者集合は次のとおりである。

$t=5$	u_1, u_2
$t=10$	u_1, u_2, u_3
$t=20$	u_1, u_2, u_3, u_4

左辺を構成する利用者のうちどれか1人が無効と考えれば、その権限委譲を無効にできる。時刻30で、u_2がg_1を無効にしたとすると、u_3は権限を失い、その結果g_3も無効となってu_4

も権限を失う。ある消去命令の影響は次のアルゴリズムによって計算できる。

<u>アルゴリズム1</u>：多数決利用権制御機構における消去命令の影響の計算

(1) t_{ri}とt_{gi}をu_iが最初に権限および委譲権付権限を得た時刻とする。そのような時刻のない場合は非常に大きな値とする。

(2) $g_j : U_j \rightarrow u_{gi}[t_g, w]$に対し消去命令が発せられたとする。$g_j$をGより除く。

(3) U_{go}は、t_{gi}の値がt_gより小さい利用者の集合とする。これらの利用者はg_jの消去命令の影響を受けない。

(4) U_{go}に対し次の条件を満足するU_{gk}を加える。$g_k : U_k \rightarrow U_{gk}[t_k, g]$（または$[t_k, g^*]$）が存在し、$U_k \leq U_{go}$; $\forall u_{ki} \in U_k$, $t_{gki} < t_k$（g^*の場合はこの条件は不要），この場合、t_{ggk}はt_kとなる（g^*の場合$\max(t_k, t_{gki}$ for $u_{ki} \in U_k)$）。

(5) U_{go}が増加しなくなるまで(4)を繰り返す。U_{go}は委譲権付利用権を持つ利用者集合である。

(6) 権限のみ持つ利用者集合U_{ro}もU_{go}を用いて同様に計算される。

(7) (4)や(6)で使われなかったGの要素を除く。

Gの要素をtの大きさの順に並べておけば計算の効率を上げることができる。

4. 多数決利用権制御機構の一般化

本節では、次の2つの立場より多数決利用権制御機構の一般化について考察する。

(1) 権限授与関数の一般化

(2) 多数決機構の他の方面への利用

4.1. 権限授与関数の一般化

個々の利用者u_iに対し3値変数x_iを割り当てる。利用者の意見yes, no, passに対応してx_iの値は1, 0, *となる。ある利用者u_gに利用権を与えるかどうかを$f(x_1, \ldots, x_n)$の値で決め、利用権を奪うかどうかを$g(x_1, \ldots, x_n)$の値で決めることとする。$f=1$となれば、u_gは利用権を得、$g=1$となれば利用権を失う。GUF機構ではfもgも1つの入力変数の値のみが1で他が*で出力が1となるような関数となっている。多数決利用権制御機構ではfは単純多数決関数であり、gは上記と同じ関数である。さらにgを単純多数決として一般化することもできる。

(1) $f \cdot g = 0$

$f=1$かつ$g=1$となるとu_gの利用権が決定できない。

(2) fやgは単調関数である。

$1 > * > 0$として、$X_1 \geq X_2$はすべての第i要素について$x_{1i} \geq x_{2i}$が成立することだとする。fが単調であるとは、任意

の$X_1 > X_2$に対し、$f(X_1) \geq f(X_2)$が成立することである。これは、ある人の意見がno→pass→yesと変わるにつれて、fは0から1に変わる可能性はあるがその逆はあり得ないことに対応する。

このような一般的な関数fやgを用いると、特定の利用者（集団）による拒否権を認めたり、特定の利用者の意見の重みを増すことができる。このような関数を利用者ごとに重みを変化させた多数決関数で表現できることもある。

4．2．多数決機構の利用

多数決機構は、利用権制御の他に次のような面にも利用できる。

(1) 多数決利用権制御機構におけるしきい値の変更

ファイルの内容の変化に伴って機密性のレベルが変化する可能性があり、対応するしきい値も動的に変化させる必要が生じる。しきい値の変更も多数決で行うのがよい。しきい値を変更する場合、変更前と変更後のしきい値より変更に要する利用者数の多いことが望ましい。

(2) 利用権授与の自動化

ファイルf_1を使用できる利用者は、f_1より作られた各種ファイル，たとえば逆ファイルf_2や統計処理によって得られたファイルf_3を、計算効率を上げるために利用することができ

る。しかし、f_3を使用できる利用者がf_1を使えるとはかぎらない。また、f_4とf_5を使用できる利用者は、f_6（たとえばf_4とf_5より作られている）を利用できる等ということがある。これは、

$f_1 \rightarrow f_2 f_3$

$f_4 f_5 \rightarrow f_6$

という形式で記述でき、あるファイル集合の使えることの判っている利用者の使用可能なファイル全体の集合は、上記の表現を関係データベースの関数従属性とみたときの閉包の計算により求めることができる。

(3) 仮想的なファイルの利用権制御

利用権制御を実ファイル名ではなく、仮想的なファイル名に対して行なうことも考えられる。この方式によると、(a) 利用権制御上1つのまとまりとなるファイル集合に1つの名前を使って制御できる。(b) 実際には存在せず、他のファイルの内容より計算されるようなファイルについても名前をつけて制御できる。

(b) の場合、あるファイル集合$f_1 f_2 \cdots f_k$から仮想的なファイルf_kが構成されるとき、$f_1 f_2 \cdots f_k \rightarrow f_k$と書けば、関数従属的な扱いが可能である。

(4) 間接的な利用権制御

さらに、利用者に与えられる利用権をファイル名とはせずにある種の利用権を抽象的に示す名前であるとする。権限Aを有するものは、ファイルf_1, f_2が使えるとか、権限A, Bの共方を有するものはファイルf_3が使えるという形で利用権を制御できる。これらもやはり関数従属的に次のように表現できる。

$$A \rightarrow f_1 f_2$$

$$AB \rightarrow f_3$$

なお、権限AとBの共方を有するものは権限Cを含むという関係はAB→Cとなり、利用者の有する権限と使えるファイルの集合の関係は、やはり関数従属性の問題に帰着できる。なおこの方法は、Minski〔MINS8109〕の導入した共助的権限を形式的に計算する方式を与えることになる。

謝辞　本論文に対し御検討をいただいた、矢島脩三教授をはじめとする研究室の諸氏に深謝する。なお、本研究の一部は文部省科学研究費によるものである。

文献

[BEERB7903] Beeri,C.and Bernstein,P.A., "Computational problems related to the design of normal form relation schemes," ACM TODS, Vol.4, No.2, March 1979, pp.30-59

[FAGI7809] Fagin,R. "On an authorization mechanism," ACM TODS, Vol.3, No.3, Sept.1978, pp.310-319

[GRIF7609] Griffiths,P.P.and Wade,B.W., "An authorization mechanism for a relational database system," ACM TODS, Vol.1, No.3, Sep.1976, pp.242-255

[MINS8109] Minsky,N., "Synergistic Authorization in Database Systems," VLDB-7, Sept.1981, pp.543-552