数理解析研究所講究録
950 巻 1996 年 94-100

The Distributed Anonymous Resource Conflict Resolution Problem

朱 潔平　角川 裕次　藤田 聡　山下 雅史

(広島大学)

1 はじめに

分散相互排除問題とは, 分散システム中の複数の競合するプロセスの中から1つのプロセスを選出し, そのプロセスに使用許可を与える問題である. 分散相互排除問題は分散システムに関する研究の中で最も重要な問題のひとつであり, これまでにいくつかのアルゴリズムが提案されている [3, 4, 5, 6, 7]. これらのアルゴリズムは, 大別して, トークンに基づいた方法とコーラムコンセンサスに基づいた方法とに分けられる. 前者の方法では, システム内にトークンと呼ばれるものを1つ循環させ, トークンを持っているプロセスのみに共有資源へのアクセス許可を与えることで相互排除を実現する. 例えば [5, 7] はこの考えに基づいたアルゴリズムである. 後者では, **コータリー** (coterie) と呼ばれる構造を利用することで共有資源への排他的なアクセスを実現する. コータリーとは, **コーラム** (quorum) と呼ばれるプロセス集合の集合であり, 任意の2つのコーラムは互いに空でない交わりを持つ. 資源にアクセスしたいプロセスは, いずれかのコーラムに属す全てのプロセスから許可を得るようにし, またどのプロセスも同時には高々一つの許可しか出さないようにする. このとき, 任意の2つのコーラムは空でない交わりを持つので, あるコーラムのすべてのプロセスから許可を得ることのできるプロセスは同時には2つ以上存在しえない. よって, 排他的なアクセスが保障される. 例えば [3, 4, 6] はこの考えに基づいたアルゴリズムである.

相互排除問題を拡張し "k個" の共有資源への排他的なアクセスを保障する問題は, **分散 k-相互排除問題**と呼ばれる. 角川ら [1] と真鍋ら [2] は, この問題に対して, それぞれ別の k-コータリーを提案し, その定義に基づいたコーラムコンセンサス型のアルゴリズムの提案を行なっている.

定義 1 (角川ら [1]) プロセス集合 Uの下の k-コータリー $C = \{Q_1, Q_2, \ldots, Q_m\}$ は以下の3つの条件を満たすプロセス集合の集合である (各 Q_i ($\neq \emptyset$) をコーラムと呼ぶ).

Existence Property: 任意の $h < k$と任意の $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_h} \in C$に対して, $Q_{i_j} \cap Q_{i_l} = \emptyset$ ($1 \le j, l \le h,\ j \neq l$) ならば, ある $Q \in C$が存在し, 全ての j ($1 \le j \le h$) に対して $Q \cap Q_{i_j} = \emptyset$ である.

Intersection Property: 任意の $k+1$ 個のコーラム $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_{k+1}} \in C$に対し, ある j, l ($1 \le j, l \le k+1,\ j \neq l$) が存在して, $Q_{i_j} \cap Q_{i_l} \neq \emptyset$ である.

Minimality Property: 任意の i, j ($1 \le i, j \le m,\ i \neq j$) に対して $Q_i \not\subseteq Q_j$である. □

定義 2 (真鍋ら [2]) プロセス集合 Uの下の k-コータリー $C = \{Q_1, Q_2, \ldots, Q_m\}$ は以下の2つの条件を満たすプロセス集合の集合である (各 Q_i ($\neq \emptyset$) をコーラムと呼ぶ).

Intersection Property: 任意の $k+1$ 個のコーラム $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_{k+1}} \in C$ に対して, $\bigcap_{j=1}^{k+1} Q_{i_j} \neq \emptyset$ である.

Minimality Property: 任意の i, j ($1 \le i, j \le m,\ i \neq j$) に対して $Q_i \not\subseteq Q_j$である. □

角川らのアルゴリズムでは, 各プロセスは同時に1つの許可を出すことができる. 定義により, 互いに交わらないコーラムが k個まで存在するので, k-相互排除が実現される. 一方, 真鍋らのアルゴリズムでは, 各プロセスは同時に高々k個の許可を出すことができる. 任意の $k+1$ 個のコーラムには交わりがあり, しかもどのプロセスも $k+1$ 個以上の許可を出さないので, やはり k-相互排除が実現される.

以上で述べた分散相互排除問題と分散 k-相互排除問題では, システム内の全てのプロセスが1つあるいはk個の資源を一様に共有すると仮定していた. これを拡張し, システム内の各プロセスが利用可能な資源の集合が必ずしも同一ではない問題を考えた場合, コータリーの定義をさらに拡張する必要がある. 宮本らは,

各プロセスが必ずしも同一の利用可能な資源集合を持たない場合の資源に対する競合解消問題を**無名資源競合回避問題**として定義し，この問題を解くための，角川らの k-coterie の定義に基づいたコーラム集合の構成法を提案した [8]．また我々は前回の夏の LA で，真鍋らの k-coterie の定義に基づいたコーラム集合の構成法を提案した [9]．本稿ではそれらの議論を一般化し，コーラムコンセンサスに基づいた方法で無名資源競合回避問題が解けるための十分条件を，基本とする k-coterie の定義によらない一般的な形で与え，さらに，その条件を満足するコーラム集合の構成法をひとつ示す．

2 無名資源競合回避問題

独立に動作可能な n 個の計算機 (プロセス) とその間を結ぶ通信リンクとからなる分散システムを考える．システムを集中管理するプロセスは存在しないものとする．またプロセス間の情報のやりとりは通信リンクを介してのメッセージ通信のみでなされ，共有変数は存在しないものとする．

分散システムにおけるプロセスと資源の共有関係は共有構造と呼ばれる 3 項組 (U, R, α) により表現される．U, Rはそれぞれプロセス，資源の全体集合であり，αは各プロセスが利用可能な資源の集合を表す写像 $(\alpha : U \rightarrow 2^R)$ である．プロセス $u \in U$が使用権を持つ資源の集合を$\alpha(u)$ と記す．

我々のモデルにおける資源は “無名”である．無名とは以下の意味である．各資源はそのラベルを除き同じ機能を持つものとする．また資源を共有する各プロセスにとっては，各プロセスは自分がどの資源を利用するかということを知る必要はなく，資源を獲得できさえすればよいものとする．このとき，各プロセスにとって資源は名前を持たない，つまり無名であるのと同じことである．

問題を定義する前に，いくつかの定義をしておく．

定義 3 コンフィグレーション c_iとは，全てのプロセスと通信リンクの状態ベクトルである．ただし各プロセスは次の 3 つの状態間を遷移するものと仮定する．

Request 状態: プロセスが資源を要求している状態．要求が認められれば Access 状態へ遷移する．

Access 状態: プロセスが資源をアクセスしている状態．資源へのアクセスが終了すれば Normal 状態へ遷移する．

Normal 状態: Request 状態でも Access 状態でもない状態．資源を要求すると Request 状態へ遷移する．

また計算πは，コンフィグレーションの列である．□

プロセス vがコンフィグレーション c においてアクセスしている資源の集合を$\rho_v(c)$ $(\subseteq \alpha(v))$ で表す．

無名資源競合回避問題は以下のように定義される [8].

定義 4 (無名資源競合回避問題) 無名資源競合回避問題は，共有構造 (U, R, α) が与えられたとき，任意の計算πが次の 2 つの条件を満足することを保証する問題である:

1. 計算$\pi = c_0, c_1, c_2, \ldots$ の任意のコンフィグレーション c_iについて

$$\forall V\ (\subseteq U) :\ \left|\bigcup_{v \in V} \rho_v(c_i)\right| \ \leq\ |\alpha(V)|$$

を満たす．

2. 計算$\pi = c_0, c_1, c_2, \ldots$ の任意のコンフィグレーション c_iについて，c_iから到達可能なあるコンフィグレーション c_tが存在して

$$\forall V (\subseteq U) :\ \left|\bigcup_{v \in V} \rho_v(c_t)\right| \ =\ |\alpha(V)|$$

を満たす．

ただし$\alpha(V)$ はαの定義を

$$\forall V\ (\subseteq U) :\ \alpha(V)\ =\ \bigcup_{v \in V} \alpha(v)$$

によって$\alpha : 2^U \rightarrow 2^R$に拡張したものである．□

定義 5 U上の同値関係$\approx$を

$$u \approx v \iff \alpha(u)\ =\ \alpha(v)$$

と定義する．Uの$\approx$による同値分割を $\mathcal{G}$ であらわす．$\mathcal{G}$ の各要素を**グループ**と呼ぶ．□

以上の定義を用いると k-相互排除問題は以下のように表現できる．

定義 6 (k-相互排除問題) k-相互排除問題は，ある k個の共有資源を持つ分散システムが与えられるとき，任意の計算が次の 2 つの条件を満足することを保証する問題である．

1. 計算$\pi = c_0, c_1, c_2, \ldots$ の任意のコンフィグレーションc_iについて
$$\forall V\ (\subseteq U): \left|\bigcup_{v \in V} \rho_v(c_i)\right| \leq k$$
を満たす.
2. 計算$\pi = c_0, c_1, c_2, \ldots$ の任意のコンフィグレーションc_iについて, c_iから到達可能なあるコンフィグレーションc_tが存在して
$$\forall V\ (\subseteq U): \left|\bigcup_{v \in V} \rho_v(c_t)\right| = k$$
を満たす. □

簡便のためグループ集合に対応するグラフを定義する. 共有構造$\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を$\mathcal{G}$とする. 各$S \subseteq \mathcal{G}$に対し, 各グループを点に, 2つのグループの間で共有される資源を点間を結ぶ枝にそれぞれ対応させて得られたグラフを**グループ集合** S**に対応するグラフ**と呼ぶ. (ただし2つの点の間に複数の辺がある場合, 1つにする.) 任意の$S \subseteq \mathcal{G}$について, Sに対応するグラフは$\mathcal{G}$に対応するグラフの部分グラフである. 任意の$S \subseteq \mathcal{G}$について, Sに対応するグラフの連結関係による分割$T = \{S_1, S_2, \ldots, S_m\}$ をS**の分割**と呼ぶ. すなわち各$S_i \in T$について, S_iに対応するグラフは連結であり, 各$S_i, S_j\ (i \neq j)$ について, $S_i \bigcup S_j$に対応するグラフは非連結である.

本稿では**コーラムベース・アプローチ (QBA)** を用いた無名資源競合問題の解法について考える. QBAは以下のように一般的に記述される:

定義 7 (QBA) プロセス集合Vの各プロセスに コーラム集合$Q \subseteq 2^U$を割り当てる. 各プロセスにはそれぞれ固定された数のトークンを与えておく.

- (資源要求) 資源を要求するプロセス$u_i \in V$はQ_i中のあるコーラムqを選び, qに属する全てのプロセス$u_j\ (\in q)$ に対して資源要求メッセージREQ_iを送信する.
- (許可) プロセスu_iからの資源要求メッセージREQ_iを受信したプロセスu_jはREQ_iを待ち行列$QUEUE_j$に入れる, u_jは今持っているトークン数が0ではなくしかもREQ_iが$QUEUE_j$の先頭であるとき, プロセスu_iに対して許可メッセージOK_jを送信して, トークン数を1つ減らす.
- (資源獲得) プロセスu_iはコーラム$q \in Q_i$に含まれる全てのプロセスu_jから許可メッセージOK_jを受信した時に資源をアクセスできる.
- (資源解放) 資源アクセスを終了したプロセスu_iは資源要求時に選ばれたコーラムqに含まれる全てのプロセスに対し資源解放メッセージREL_iを送信する. REL_iを受信したプロセスはそのトークン数を1つ増やす. □

3 十分条件

無名資源競合回避問題では, システム内の各プロセスが利用可能な資源集合は必ずしも同一ではない. コーラム集合のプロセスへの割り当てに関して, 以下の定理が成り立つ.

定理 1 共有構造$\mathcal{C} = (U, R, \alpha)$ に関する無名資源競合回避問題は, $\alpha(u_1) \neq \alpha(u_2)$ であるような2つのプロセス$u_1, u_2\ (\in U)$ に対してu_1とu_2が同一のコーラム集合を用いるようなQBAによって解くことはできない.

証明) 背理法で証明する. $u_i\ (i = 1, 2)$ が用いるコーラム集合をQ_iとし, $Q_1 = Q_2$とする. $\alpha(u_1) \neq \alpha(u_2)$より, 一般性を失うことなく, $\alpha(u_1) - \alpha(u_2) \neq \emptyset$ と仮定することができる. ある時刻においてu_2が$|\alpha(u_2)|$個の資源を獲得しているとする. $Q_1 = Q_2$であるので, Q_1, Q_2ともにもうこれ以上の資源へのアクセスはできない. しかしu_1は$\alpha(u_1) - \alpha(u_2)\ (\neq \emptyset)$ 中の任意の資源を使うことができるはずである. よって矛盾. □

第1節で述べたk-coterieの定義の中には, トークン数に対する設定は含まれていない. しかしQBAによるk-相互排除問題の解法では, 各プロセスに対してどのようにトークン数を割り当てるかが重要となる. 例えば真鍋らのk-coterieでは, 各プロセスに割り当てられるトークン数がkのときは正しくk-相互排除問題を解くことができるが, 1のときは正しく解くことができない. 一方, 角川らのk-coterieでは, 各プロセスに割り当てられるトークン数が1のときは正しく問題を解くことができるが, kのときは正しく解くことができない. 以上のことから, ここではまず, k-相互排除問題を解くことのできるコーラム集合とトークン数の割り当ての対を, あらたな概念として定義する.

定義 8 (k-crowd) k-crowd $\mathcal{Q}$は, k-相互排除問題を解くことのできるコーラム集合Qと 各プロセスへのトークン数の割り当て$\tau:\ V \rightarrow N$の2項組(Q, τ) で定義される. ここでVはQを構成するプロセスの集合であり, Nは自然数集合である. プロセスuに割り当てられるトークン数を$\tau(u)$で表す. □

k-crowd $\mathcal{Q} = (Q, \tau)$ に対し, $\mathcal{Q}' = (Q', \tau)$ かつ $Q' \subseteq Q$ のとき, $\mathcal{Q}' \subseteq \mathcal{Q}$ と記す.

次に k-crowd に関するいくつかの定義をする.

定義 9 k-crowd $\mathcal{Q}$ に対し, 以下の条件を満たす $\mathcal{Q}' \subseteq \mathcal{Q}$ を $\mathcal{Q}$ の**共存部分 crowd** と呼ぶ.

1. $\mathcal{Q}'$は k-crowd である.
2. $\mathcal{Q}$ を使って k_1 $(\leq k)$ 個資源を獲得していたとき, $\mathcal{Q}'$を使ってさらに $k - k_1$個資源を獲得できる. □

例えば真鍋らの k-coterie の定義に基づいた k-crowd $\mathcal{Q}$ では, 任意の $\mathcal{S} \subseteq \mathcal{Q}$ は $\mathcal{Q}$ の共存部分 crowd である. また角川らの k-coterie の定義に基づいた k-crowd $\mathcal{Q}$ では, $\mathcal{Q}$ の共存部分 crowd は $\mathcal{Q}$ 自身のみである.

定義 10 (一貫性) k_1-crowd $\mathcal{Q}_1 = (Q_1, \tau_1)$ と k_2-crowd $\mathcal{Q}_2 = (Q_2, \tau_2)$ を構成するプロセスの集合をそれぞれ V_1, V_2 とする.

$$\forall u \in V_1 \cap V_2: \ [\tau_1(u) = \tau_2(u)]$$

ならば, $\mathcal{Q}_1, \mathcal{Q}_2$は**一貫性**をもつと言う. □

定義 11 一貫性をもつ crowd $\mathcal{Q}_1, \mathcal{Q}_2$ に対し, QBA を用いて, $\mathcal{Q}_1$, $\mathcal{Q}_2$をそれぞれ単独に使う場合とそれらを同時に使う場合に獲得できる資源数が同一であるならば, $\mathcal{Q}_1$と $\mathcal{Q}_2$は**相互独立**であるという. □

定義 12 コーラム集合 Q_1, Q_2に対し,

$$Q_1 \oplus Q_2 \ = \ \{q \cup q' \mid q \in Q_1, q' \in Q_2\}$$

と定義する. □

一貫性がある 2 つの crowd $\mathcal{Q}_1, \mathcal{Q}_2$に対し, コーラム集合の演算$\oplus$は以下のように拡張できる.

定義 13 一貫性をもつ 2 つの crowd $\mathcal{Q}_1 = (Q_1, \tau_1)$, $\mathcal{Q}_2 = (Q_2, \tau_2)$ に対し,

$$\mathcal{Q}_1 \oplus \mathcal{Q}_2 \ = \ (Q_1 \oplus Q_2, \tau_1 \cup \tau_2)$$

と定義する. □

定義 14 一貫性をもつ 2 つの crowd $\mathcal{Q}_1 = (Q_1, \tau_1)$, $\mathcal{Q}_2 = (Q_2, \tau_2)$ に対し,

$$\forall q \in Q_2: \ [\exists q' \in Q_1, q' \subseteq q]$$

ならば, $\mathcal{Q}_1$ は $\mathcal{Q}_2$を**支配** (dominate) すると言う. □

定義 15 $\mathcal{Q}$ と $\mathcal{P} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_m\}$ の各要素は, 互いに一貫性を持つ crowd であるとする. $\mathcal{Q}$ が任意の $\mathcal{Q}_i$ $(\in \mathcal{P})$ を支配するとき, $\mathcal{Q}$ は $\mathcal{P}$を支配すると言う. □

以下の補題が成り立つ.

補題 1 $\mathcal{S} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_m\}$ を互いに一貫性をもつ *crowd* の集合とする. もし $\mathcal{S}$を支配する ***k-crowd*** $\mathcal{Q}$ が存在するならば, $\bigcup_{\mathcal{Q}_j \in \mathcal{S}} \mathcal{Q}_j$を使って高々$k$個の資源しか獲得できない.

証明) 各 i について $\mathcal{Q}_i = (Q_i, \tau_i)$ とし, $\mathcal{Q} = (Q, \tau)$ とする. 支配の定義により, 任意の $q \in \bigcup_{\mathcal{Q}_j \in \mathcal{S}} Q_j$ について, qの全てのプロセスから許可を獲得するためには, ある $q'(\in Q)$ の全てのプロセスから許可を獲得しなければならない. $\mathcal{Q}$ は k-crowd であるから, $\mathcal{Q}$ を使って高々k個の資源しか獲得できない. よって$\bigcup_{\mathcal{Q}_j \in \mathcal{S}} \mathcal{Q}_j$を使って高々$k$個の資源しか獲得できない. □

本稿で提案する十分条件 SC は以下のようにあらわされる.

十分条件 SC: 共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を $\mathcal{G}$ とする.

(1) 各グループ $G_i(\in \mathcal{G})$ には, 互いに一貫性を持つ $|\alpha(G_i)|$-crowd $\mathcal{Q}_i$が割り当られる.

(2) 任意の $\mathcal{S} \subseteq \mathcal{G}$について, $\bigcup_{G_i \in \mathcal{S}} \{\mathcal{Q}_i\}$ を支配する任意の x-crowd は

$$x \ \geq \ \left| \bigcup_{G_i \in \mathcal{S}} \alpha(G_j) \right|$$

を満たす. また $\mathcal{S}$に対応するグラフが連結のとき, $\bigcup_{G_i \in \mathcal{S}} \{\mathcal{Q}_i\}$ を支配する $|\bigcup_{G_i \in \mathcal{S}} \alpha(G_i)|$-crowd が存在する.

(3) 任意のグループ G_i $(\in \mathcal{G})$ について, G_iに割り当てられる crowd $\mathcal{Q}_i$は

$$\bigoplus_{\mathcal{Q} \in \mathcal{P}_i} \mathcal{Q}$$

の共存部分 crowd である. ここで $\mathcal{P}_i$は $\mathcal{Q}_i$を支配する全ての互いに相互独立な crowd の集合である. また $\mathcal{P}_i$の各要素 $\mathcal{Q}$ は, (1) で割り当てられた crowd のうち, $\mathcal{Q}$ に支配されない任意の crowd と相互独立である. □

以下の定理が成り立つ.

定理 2 コーラム集合の集合とトークン数の割り当てが十分条件 SC を満たすならば, QBA によって無名資源競合回避問題を解くことができる.

証明) 共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を $\mathcal{G}$とする. 任意のプロセス集合 $V = \{v_1, v_2, \ldots, v_m\}$ ($\subseteq U$) を考える. $S = \{G_i \mid v_j \in G_i$ for some $v_j \in V\}$ とすると,

$$\alpha(V) = \bigcup_{G_i \in S} \alpha(G_i) \qquad (1)$$

が成り立つ. 以下では, 無名資源競合回避問題の定義の 2 つの条件を満たすことを示す.

1. Sの分割を $T = \{S_1, S_2, \ldots, S_n\}$ とする. このとき以下の式が成り立つ.

$$\left| \bigcup_{G_i \in S} \alpha(G_i) \right| = \sum_{S_t \in T} \left| \bigcup_{G_j \in S_t} \alpha(G_j) \right| \qquad (2)$$

任意の S_i ($\in T$) に対応するグラフは連結なので, 十分条件 SC-(2) より, $\bigcup_{G_j \in S}\{\mathcal{Q}_j\}$ を支配する $|\bigcup_{G_i \in S} \alpha(G_i)|$-crowd が存在する. また補題 1 より, $\bigcup_{G_j \in S_i} \mathcal{Q}_j$を使って, 高々 $|\bigcup_{G_j \in S_i} \alpha(G_j)|$ 個の資源しか獲得できない. よって式 (1),(2) により, $|\bigcup_{v \in V} \rho_v(c)| \leq |\alpha(V)|$ を達成する.

2. 背理法で示す: あるコンフィグレーション c_iで S 中のプロセスが獲得している資源の数を

$$\left| \bigcup_{v_j \in V} \rho_{v_j}(c_i) \right| < |\alpha(V)|$$

とし, これ以上の資源を獲得できないと仮定する.

仮定より, まだ獲得されていない資源 r ($\in \alpha(V)$) が少なくともひとつ存在する. $r \in \alpha(G^*)$ であるようなグループ G^*を考える. G^*の属している連結成分を S_iとする. 十分条件SC-(1) より, グループ G^*には $|\alpha(G^*)|$-crowd $\mathcal{Q}^* = (Q^*, \tau^*)$ が割り当てられている.

$\mathcal{Q}^*$を支配する全ての互いに相互独立な crowd の集合を $\mathcal{P}^*$とする. 十分条件 SC-(3) により, $\mathcal{Q}^*$は $\bigoplus_{\mathcal{Q} \in \mathcal{P}^*} \mathcal{Q}$ の共存部分 crowd である.

$\forall \hat{\mathcal{Q}} \in \mathcal{P}^*$に対し, 各グループに割り当てられた crowd の中で$\hat{\mathcal{Q}}$によって支配されるものの集合を $\hat{\mathcal{P}}$とし, $\hat{S} = \bigcup_{\mathcal{Q}_i \in \hat{\mathcal{P}}} G_i$と定義する. 十分条件SC-(2) より, $\hat{\mathcal{Q}}$は

$$x \geq \left| \bigcup_{G_i \in \hat{S}} \alpha(G_i) \right|$$

であるような x-crowd である. $G^* \in \hat{S}$であるので, $\hat{S}$中のプロセスが獲得していた資源数は $|\bigcup_{G_i \in \hat{S}} \alpha(G_i)|$ より少ない. さらに, 十分条件 SC-(3) により, $\hat{\mathcal{Q}}$は各グループに割り当てられた crowd の中で$\hat{\mathcal{P}}$に属していない任意の crowd と相互独立であるから, ある $q \in \hat{Q}$が存在し, qの全てのプロセスから許可を獲得できる. よって, ある $q' \in \bigoplus_{\mathcal{Q} \in \mathcal{P}^*} Q$ が存在し, q'の全てのプロセスから許可を獲得できる. 最後に $\mathcal{Q}^*$は $\bigoplus_{\mathcal{Q} \in \mathcal{P}^*} \mathcal{Q}$ の共存部分 crowd であるから, 必ずある $q'' \in Q^*$が存在し, q''の全てのプロセスから許可を獲得できる. これは仮定と矛盾を生じる. したがって題意が成り立つ.

□

4 crowd 集合の構成法

前節では, 無名資源競合回避問題が QBA によって解けるための十分条件 SC を述べた. 以下では, この条件を満たす crowd 集合の具体的な構成法を与える.

より一般的な共有構造について考える前に, 集中構造という特別な共有構造を定義する.

共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を $\mathcal{G}$とする. $\alpha(G_1) \cap \alpha(G_2) \neq \emptyset$ であるような任意の G_1, G_2 ($\in \mathcal{G}$) に対し, $\alpha(G_3) = \alpha(G_1) \cup \alpha(G_2)$ であるような G_3 ($\in \mathcal{G}$) が存在するならば, 共有構造 $\mathcal{C}$は**集中構造**であると言う.

$\mathcal{C} = (U, R, \alpha)$ が集中構造のとき, 対応するグラフが連結であるような任意の $S \subseteq \mathcal{G}$に対し, $\bigcup_{G_i \in S} \alpha(G_i) = \alpha(G_m)$ を満たすグループ G_m ($\in \mathcal{G}$) が存在する.

全体資源集合 Rに対して, 可能な全てのグループ集合の数は $2^{|R|} - 1$ である. 共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合がそれら可能な全てのグループを包含するとき, 共有構造$\mathcal{C}$は**完全構造**であると言う. 明らかに完全構造は集中構造である.

任意の共有構造に対し, 以下の定理が成り立つ.

定理 3 共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を $\mathcal{G}$とする. このとき, 対応するグループ集合 $\mathcal{G}'$が $\mathcal{G}' \supseteq \mathcal{G}$を満たすような集中構造 $\mathcal{C}' = (U', R, \alpha')$ が存在する.

証明) 共有構造 $\mathcal{C} = (U, R, \alpha)$ の全体資源集合 Rに対する完全構造を $\mathcal{C}' = (U', R, \alpha')$ とし, $\mathcal{C}'$に対応するグ

ループ集合を $\mathcal{G}'$ とすると, $\mathcal{G}' \supseteq \mathcal{G}$ である. 完全構造は集中構造であるので, よって定理が成り立つ. □

任意の共有構造に対し, 以下の定理が成り立つ.

定理 4 共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を $\mathcal{G}$ とし, 十分条件 *SC* を満たす *crowd* 集合を $\bigcup_{G_i \in \mathcal{G}} \mathcal{Q}_i$ とする. 任意の $S \subseteq \mathcal{G}$ に対して, $\bigcup_{G_i \in S} \mathcal{Q}_i$ は十分条件 *SC* を満たす.

証明) $\bigcup_{G_i \in \mathcal{G}} \mathcal{Q}_i \supseteq \bigcup_{G_i \in S} \mathcal{Q}_i$ であるので, 十分条件 SC-(1),(3) は明らかに満たされる. また十分条件 SC-(2) の主張は, 任意の S $(\subseteq \mathcal{G})$ に対して成り立っており, しかも $\forall S' \subseteq S$, $S' \subseteq \mathcal{G}$ であるので, 十分条件 SC-(2) が成り立つ. □

定理 3,4 より, 共有構造を包含する集中構造に対応する十分条件 SC を満たす構造の集合を見つけることができれば, 共有構造に対して十分条件 SC を満たす構造の集合を見つけることができることがわかる.

任意の共有構造に対し, 十分条件 SC を満たす crowd 集合の集合の構成法を以下に示す.

構成法 1: 共有構造 $\mathcal{C} = (U, R, \alpha)$ に対応するグループ集合を $\mathcal{G}$ とする.

step1: 対応するグループ集合 $\mathcal{G}'$ が $\mathcal{G}' \supseteq \mathcal{G}$ であるような集中構造 $\mathcal{C}' = (U', R, \alpha')$ を見つける.

step2: $\mathcal{C}'$ の各グループ G_i に対する $|\alpha(G_i)|$-crowd $\mathcal{Q}'_i = (Q'_i, \tau_i)$ を, 全体プロセス集合 U の上に構成する. ここで, 任意の $\mathcal{Q}'_1, \mathcal{Q}'_2 \in \bigcup_{G_i \in \mathcal{G}'} \{\mathcal{Q}'_i\}$, $q \in Q_1$, $q' \in Q_2$ に対し, $q' \cap q = \emptyset$ である.

step3: 各グループ G_i $(\in \mathcal{G})$ に以下のような crowd $\mathcal{Q}_i$ を割り当てる.

$$\mathcal{Q}_i = \bigoplus_{\alpha(G_j) \supseteq \alpha(G_i)} \mathcal{Q}'_j,$$

□

$\oplus$ 演算について, 以下の補題が成り立つ:

補題 2 一貫性を持つ相互独立な k_1-*crowd* $\mathcal{Q}_1$ と k_2-*crowd* $\mathcal{Q}_2$ に対し, $\mathcal{Q}_1 \oplus \mathcal{Q}_2$ は $\min\{k_1, k_2\}$-*crowd* である.

証明) $\mathcal{Q}_1 = (Q_1, \tau_1), \mathcal{Q}_2 = (Q_2, \tau_1)$ とする. ここで一般性を失うことなく, $k_1 \leq k_2$ と仮定する. $\mathcal{Q}$ をつかって $\min\{k_1, k_2\}$-相互排除問題が解けることを証明する.

1. 定義より, 任意の $q \in Q_1 \oplus Q_2$ に対して, $q = q' \cup q''$ であるような $q' \in Q_1, q'' \in Q_2$ が存在する. q の全てのプロセスから許可を獲得するためには, q', q'' の全てのプロセスの許可を獲得しなければならない. よって $\mathcal{Q}_1 \oplus \mathcal{Q}_2$ からは高々 $\min\{k_1, k_2\}$ 個の資源しか獲得できない.
2. $\mathcal{Q}_1 \oplus \mathcal{Q}_2$ から $h < \min\{k_1, k_2\}$ 個の資源しか獲得できないと仮定する. $\mathcal{Q}_1$ と $\mathcal{Q}_2$ は相互独立な k_1-crowd と k_2-crowd であるので, ある q' $(\in Q_1)$, q'' $(\in Q_2)$ が存在し, q' と q'' に属する全てのプロセスから許可を獲得できる. 演算 $\oplus$ の定義により, $q = q' \cup q''$ であるような $q \in Q_1 \oplus Q_2$ が必ず存在するから, q の全てのプロセスの許可を獲得できる, よって矛盾.

以上により, $\mathcal{Q}_1 \oplus \mathcal{Q}_2$ は $\min\{k_1, k_2\}$-crowd である. □

明らかに, $\mathcal{Q}_i$ $(i = 1, 2)$ は $\mathcal{Q}_1 \oplus \mathcal{Q}_2$ を支配する.

以下で構造方法 1 で構成する crowd の集合は十分条件 SC を満たすことを証明する.

定理 5 構成法 *1* によって構成された構造の集合は十分条件 *SC* を満たす.

証明) 与えられる共有構造を $\mathcal{C} = (U, R, \alpha)$ とし, それに対応するグループ集合を $\mathcal{G}$ とする. 対応するグループ集合 $\mathcal{G}'$ が $\mathcal{G}' \supseteq \mathcal{G}$ であるような集中構造 $\mathcal{C}' = (U', R, \alpha')$ を考える.

以下では, 構成法 1 によって構成された crowd の集合が十分条件 SC の (1),(2),(3) をそれぞれ満たすことを示す.

1. 構成法 1 によって

$$\mathcal{Q}_i = \bigoplus_{\alpha(G_j) \supseteq \alpha(G_i)} \mathcal{Q}'_j$$

ここで, $\mathcal{Q}'_j$ は step1 でつくられた $|\alpha(G_j)|$-crowd である. 各 $\mathcal{Q}'_h, \mathcal{Q}'_k$, $q \in Q'_h, q' \in Q'_k$ に対し, $q' \cap q = \emptyset$ であるので, $\mathcal{Q}'_h$ と $\mathcal{Q}'_k$ は相互独立である. さらに,

$$|\alpha(G_i)| = \min_{\alpha(G_j) \supseteq \alpha(G_i)} \{|\alpha(G_j)|\}$$

であるので, 補題 2 により, $\mathcal{Q}_i$ は $|\alpha(G_i)|$-crowd である.

2. 任意の S $(\subseteq \mathcal{G}')$ に対し, crowd の集合 $\mathcal{P}'$ が存在し,

$$\mathcal{P}' = \{\mathcal{Q}'_j \mid \alpha(G_j) \supseteq \bigcup_{G_i \in S} \alpha(G_i)\}$$

が成り立つ. 構成法1により, $\bigcup_{G_i \in S}\{\mathcal{Q}_i\}$ を支配する任意の crowd $\mathcal{Q}''$について

$$\exists \mathcal{P}'' \subseteq \mathcal{P}',\ \mathcal{Q}'' = \bigoplus_{\mathcal{Q}'_j \in \mathcal{P}''} \mathcal{Q}'_j$$

が成り立つ. step2 より任意の $\mathcal{Q}'_j$は $|\alpha(G_j)|$-crowd である. $|\alpha(G_j)| \geq |\bigcup_{G_i \in S} \alpha(G_i)|$ である. $\mathcal{Q}'_i$と $\mathcal{Q}'_j$は相互独立であるので, 補題 2 により, $\mathcal{Q}''$は $x \geq |\bigcup_{G_i \in S} \alpha(G_i)|$ であるような x-crowd である.

S ($\subseteq \mathcal{G}'$) に対応するグラフが連結であるとき, 集中構造であるので, $\bigcup_{G_i \in S} \alpha(G_i) = \alpha(G_m)$ であるようなグループ $G_m \in \mathcal{G}'$が存在する. よって step3 より, $|\bigcup_{G_i \in S} \alpha(G_i)|$-crowd $\mathcal{Q}'_m$は $\bigcup_{G_i \in S}\{\mathcal{Q}_i\}$ を支配する.

3. 構成法 1 により, $\forall G_i$, $\mathcal{Q}_i$を支配する全て互いに相互独立の crowd 集合 $\mathcal{P}$は$\bigcup_{\alpha(G_j) \supseteq \alpha(G_i)}\{\mathcal{Q}'_j\}$ である. $\mathcal{Q}_i$は$\bigoplus_{\alpha(G_j) \supseteq \alpha(G_i)} \mathcal{Q}'_j$ であるから, $\bigoplus_{\alpha(G_j) \supseteq \alpha(G_i)} \mathcal{Q}'_j$の共存部分 crowd でもある.

   構成法 1 により,$\forall \mathcal{Q}'_j (\in \mathcal{P})$ に支配される crowd 集合 $\mathcal{P}_j = \bigcup_{\alpha(G_h) \subseteq \alpha(G_j)}\{\mathcal{Q}_h\}$ である. $\mathcal{Q}'_j$ は任意の $\mathcal{Q}_n \in \bigcup_{G_h \in \mathcal{G}'}\{\mathcal{Q}_h\} - \mathcal{P}_j$ と相互独立である.

定理 4 により, $\bigcup_{G_i \in \mathcal{G}} \mathcal{Q}_i$は十分条件 SC を満たす. □

構成法 1 の step2 では, 各グループ G_iに対する $|\alpha(G_i)|$-crowd を作るために, どの k-coterie の定義を使っても構わない. さらに, 各グループにおいてそれぞれ異なる k-coterie の定義を使うこともできる.

各グループ G_iに対し, 真鍋らの k-coterie を使って, $|\alpha(G_j)|$-crowd $\mathcal{Q}_i$を作るとすると, $2^{|R|} - 1$ 個のプロセスがあれば, 構成法 1 によって, 無名資源競合回避問題が解ける crowd の集合を必ず構成できる.

5 おわりに

本稿では, QBA を用いて分散無名資源競合回避問題を解くための十分条件 SC を示した. またその条件を満たす crowd 集合の構成方法を示した. 今後の研究の課題として, 十分条件 SC の必要性の検討と構成方法の条件についての考察があげられる.

参考文献

[1] H.Kakugawa, S.Fujita, M.Yamashita, T.Ae, "A Distributed k-Mutual Exclusion Algorithm using k-Coterie," *Information Processing Letters*,1994.

[2] Y.Manabe, S.Aoyagi, "A Distributed k-Mutual Exclusion Algorithm using k-Coterie," *IEICE Japan,SIG Computation Record* COMP91-13,pp.11-18,May 1993.

[3] L.Lamport, "Time,clocks,and the ordering of events in a distributed system," *Communications of the ACM,* Vol.21,No.7,pp.558-565,Jul.1978.

[4] M.Maekawa, "A $\sqrt{N}$ Algorithm for Mutual Exclusion in Decentralized Systems," *ACM Transactions on Computer Systems,* Vol.3, No.2, pp.145-159, May.1985.

[5] M.Mizuno, M.L.Neilsen, R.Rao, "A Token Based Distributed Mutual Exclusion Algorithm based on Quorum Agreements," Proc. of 11th International Conference on Distributed Computing Systems, pp.361-368,May 1991.

[6] G.Ricart, A.K.Agrawala, "A Optimal algorithm for Mutual Exclusion in Computer Networks," *Communications of the ACM,* Vol.24,No.1,pp.9-17,Jan.1981.

[7] I.Suzuki, T.Kasami, "A Distributed Mutual Exclusion Algorithms," *ACM Transactions on computer Systems,* Vol.3,No.4,pp.344-349,Nov.1985.

[8] H.Miyamoto, "A Study on Quorun Based Approach for Solving the Anonymous Resource Conflict Resolution Problem," 広島大学修士論文,1994.

[9] 朱 潔平, 角川 裕次, 藤田 聡, 山下 雅史, "分散システムにおける無名資源競合回避問題" 夏の LA シンポジウム,1995.