

拡張された分散 k -相互排除

広島大学工学部 宮本英典 (Hidenori Miyamoto)
広島大学工学部 角川裕次 (Hirotsugu Kakugawa)
広島大学工学部 山下雅史 (Masafumi Yamashita)

1 はじめに

LAN(Local Area Networks), WAN(Wide Area Networks) などの計算機ネットワークの普及により, それらのネットワーク上に実現される分散システムに関する研究が盛んに行われるようになった. 分散システムとは, 複数のプロセスとそれらの間を結ぶ通信リンクからなるシステムである. 分散システム上での最も重要な問題の 1 つに分散相互排除問題がある. 分散相互排除問題は, 分散システムにおいて複数のプロセスによって共有される資源に対する排他的なアクセスを保証する問題である. 言い替えれば, どの時刻においても, 高々 1 つのプロセスにしか共有資源に対するアクセスを許さないことを保証する問題が, 分散相互排除問題である. この問題に対して, これまでに多くの研究がなされており, この問題を解くためのアルゴリズムが幾つか提案されている [5][6][8][11][12].

それらのアルゴリズムは, 大別するとトークン型とコーラムコンセンサス型の 2 つのタイプに分けることが出来る. トークン型のアルゴリズムは, システム内に特権トークンを循環させ, そのトークンを保持しているプロセスのみが共有資源へのアクセスを許可されることで相互排除を実現する. この考えに基づいたアルゴリズムに, 例えば [8], [12] がある. コーラムコンセンサス型のアルゴリズムでは, コータリー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, コーラムと呼ばれるプロセス集合の集まりである. 任意の 2 つのコーラムは互いに交わっている. 資源へアクセスを行なうプロセスは, いずれかのコーラムに属する全てのプロセスより許可を得るようにする. コータリーに属する任意の 2 つのコーラムは互いに交わりを持つので, どのプロセスも同時には高々 1 つの許可しか出さない事にすれば, コーラムの全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. この考えに基づいたアルゴリズムに, 例えば [5], [6], [11] がある.

共有される資源が 1 つでなく, より一般的な k 個の同一な資源の共有問題についての研究が近年なされている. k 個の共有資源への排他的なアクセスを保障するこの問題を分散 k -相互排除問題と呼ぶ. Raymond は Ricart と Agrawala の分散相互排除アルゴリズム [11] を拡張することで同時に k 個までのプロセスが資源にアクセスできるが $k+1$ 個はできないことを保障するアルゴリズムを提案した [10]. 角川らはコータリーを拡張した k -コータリーを提案し, コーラムコンセンサス型のアルゴリズムを提案している [3].

集合 U の下での k -コータリー Q とは, 以下の条件を満たす空でない集合 $Q \subseteq 2^U$ である.

1. 各 $q \in Q$ に対して, $q \neq \emptyset$.
2. Non-intersection Property:
各 $h(< k)$ に対しある $q_1, q_2, \dots, q_h \in Q$ が存在し, これらの内の任意の q_i, q_j に対し $q_i \cap q_j = \emptyset$ であれば, ある $q_{h+1} \in Q$ が存在してすべての $i (1 \leq i \leq h)$ に対し $q_{h+1} \cap q_i = \emptyset$ かつ $q_{h+1} \neq q_i$ となる.
3. Intersection Property:
任意の $q_1, q_2, \dots, q_k, q_{k+1} \in Q$ に対し, ある q_i, q_j が存在して $q_i \cap q_j \neq \emptyset$.
4. Minimality:
任意の $q_1, q_2 \in Q$ に対し $q_1 \not\subseteq q_2$.

□

以上で述べた分散相互排除問題の資源の共有のモデルも、システム内の全てのプロセスが1つあるいは k 個の資源を一樣に共有する。そこで本稿では、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を扱うために従来の分散 k -相互排除問題の拡張である無名資源競合回避問題を新たに定義する。そしてこの問題に対し、コーラム集合を用いた解法について検討する。以降、問題の定式化、コーラムベースプロトコルの定義、無名資源競合回避問題を解決するためにコーラム集合が満たさなければならない十分条件、およびその条件を満たすコーラム集合の構成法について述べる。

2 問題の定式化

2.1 分散システム

分散システムは、独立に動作可能な n 個のプロセスと、各プロセス間を結ぶ通信リンクからなる。システムを集中管理するプロセスは存在しない。各プロセスは一意なプロセス識別子(プロセス ID) $u_i, 1 \leq i \leq n$ を持つ。各プロセスは自身のプロセス ID と他の全てのプロセスのプロセス ID を知っている。任意の2つのプロセス間の通信は他のプロセスを介さず1対1の通信が可能である。すなわち、ネットワークポロジは完全グラフである。プロセス間の情報のやり取りは、通信リンクを通じてのメッセージ通信のみであり、共有変数は存在しない。同じプロセスから送信された任意の2つのメッセージの順序は通信リンク内で変化しない。メッセージ通信には有限の通信遅延を伴うが、その値を予測する事はできない。最後に、全てのプロセスと通信リンクは無故障である。

2.2 無名資源競合回避問題

分散システムには複数のプロセスにより共有される k 個の資源 $r_i, 1 \leq i \leq k$ が存在する。プロセスおよび資源の全体集合をそれぞれ U, R と表すことにする。また、 α をプロセスが利用可能な資源の集合を表す写像とする。すなわち、 $\alpha: U \rightarrow 2^R$ であり、プロセス $u \in U$ がアクセスできる資源の集合は、 $\alpha(u) \subseteq R$ により表される。このとき、分散システムにおけるプロセスによる資源の共有関係を3項組 (U, R, α) により表現する。我々は (U, R, α) を共有構造と呼ぶ。

資源はそのラベル(名前) r_i を除き同じ機構を持つとする。このとき資源を要求するプロセスにとっては、資源が獲得できればそれがどれあるかは問題ではない。この意味で我々のモデルにおける資源は無名である。

分散システムにおける計算の定義より始め、無名資源競合回避問題を形式的に定義する。

定義 1 分散システムのコンフィグレーション c_i は全てのプロセスと通信リンクの状態ベクトルである。プロセスの状態は次の3つの状態間を遷移する。

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: プロセスが資源に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。

計算 π は、コンフィグレーションの列で定義する。 □

定義 2

無名資源競合回避問題 (Anonymous Resource Conflict Resolution Problem; 以下、競合回避問題) は、共有構造 (U, R, α) が与えられた時、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。 $\rho_v(c) (\subseteq \alpha(v))$ をプロセス v が分散システムのコンフィグレーション c にあるときアクセスしている資源の集合とする。

1. 計算 π の任意のコンフィグレーション c_i について,

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

を満たす.

2. 計算 π 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,

$$|\bigcup_{v \in V} \rho_v(c_t)| = |\alpha(V)|$$

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. □

従来の分散 k -相互排除問題は, 上記の問題において, $\forall u(\in U), \alpha(u) = R$ とした場合で表現される. 従って競合回避問題は, 従来の分散 k -相互排除問題をプロセスと資源の共有関係について一般化した問題である.

U 上の同値関係 $\sim$ を

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

で定義し, U の $\sim$ による同値分割を $\mathcal{G}$ とする. $\mathcal{G}$ の要素をグループと呼ぶ.

3 コーラム集合を用いた解法

競合回避問題に対するコーラム集合を用いた解法について考える.

3.1 コーラムベースト・プロトコル

各プロセス u_i には, あるコーラム集合 $Q_i \subseteq 2^U$ を割り当てる. 全プロセスは, 以下に示す同一のコーラムベースト・プロトコル $\mathcal{P}$ に従って動作する.¹

[プロトコル $\mathcal{P}$]

(資源要求) 資源を要求するプロセス u_i は, あるコーラム $q \in Q_i$ を選び, すべてのプロセス $u_j \in q$ に対し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, 既に他の資源要求メッセージ REQ_k に対し, 許可メッセージ OK_j を送信しており, まだ資源解放メッセージ REL_k を受信していないならば REQ_i を待ち行列 $QUEUE_j$ に入れる. そうでないならば, プロセス u_j に許可メッセージ OK_j を送信する.
- 資源解放メッセージ REL_k を受信したプロセス u_j は, $QUEUE_j$ が空でなければ先頭の資源要求メッセージ REQ_i を $QUEUE_j$ より取り出しプロセス u_i に対し許可メッセージ OK_j を送信する.

(資源獲得) コーラム $q \in Q_i$ に含まれる全てのプロセスから許可メッセージを受信した時, その時に限りプロセス u_i は資源をアクセスできる.

¹プロトコル $\mathcal{P}$ では, デッドロック, 飢餓状態を回避することとはできないが, デッドロック, 飢餓状態を回避するように改良することは容易である. 例えば, [3]を参照されたい.

(資源解放) 資源アクセスを終了したプロセス u_i は、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。□

3.2 コーラム集合の構成

プロトコル P を用いて競合回避問題を解くためにコーラム集合が満たさなければならない必要および十分条件、およびその条件を満たすコーラム集合の構成法について述べる。

3.2.1 必要条件と十分条件

定理 1 $S = (U, R, \alpha)$ をあるプロセス集合 V, V' について $\alpha(V) \neq \alpha(V')$ であるような共有構造とする。この時、全てのプロセスが同じコーラム集合を用いるならば、 S についての競合回避問題をプロトコル P を用いて解くことはできない。

(略証) Q をプロトコル P を用いて S についての競合回避問題を解くようなコーラム集合であるとする。一般性を失うことなく、ある資源 r が存在して、 $r \in \alpha(V)$ かつ $r \notin \alpha(V')$ とする。このとき、 $|\alpha(V \cup V')| > |\alpha(V')|$ 。競合回避問題の条件 1 を満たすためには、コーラム集合 Q には互いに素な $(|\alpha(V')| + 1)$ 個のコーラムは存在してはいけな。しかしながら、条件 2 を満たすためには、コーラム集合 Q には互いに素な $|\alpha(V \cup V')|$ 個のコーラムが存在しなければならず、 $|\alpha(V \cup V')| > |\alpha(V')|$ によりこのふたつの要求は相反する。□

定理 1 により、競合回避問題を解くためのコーラム集合は共有構造と独立ではいけないことが分かる。

そこで、コーラム集合の割り当てについて以下の仮定を設ける。

仮定 A 任意のグループ $G_i (\in \mathcal{G})$ について、すべてのプロセス $u_i \in G_i$ には同じコーラム集合 Q_{G_i} を割り当てる。□

プロトコル P により競合回避問題を解くためにコーラム集合 Q_{G_i} に与える条件として以下のものを考える。

条件 C 任意のグループの集合 $S \subseteq \mathcal{G}$ について、 $\bigcup_{G_i \in S} Q_{G_i}$ が $|\bigcup_{G_i \in S} \alpha(G_i)|$ -コートリーである。□

この条件 C が競合回避問題を解くための十分条件であることを次に示す。

定理 2 条件 C を満たすコーラム集合の集合 Q を用いるプロトコル P は、競合回避問題を解く。

(証明) 任意のプロセス集合 $V = \{v_1, v_2, \dots, v_m\}$ について、 $S = \{G_i | v_j \in G_i \text{ for some } v_j \in V\}$ とすると、

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

また、プロセス v_j に割り当てられたコーラム集合を Q_{v_j} とすると、

$$\bigcup_{v_j \in V} Q_{v_j} = \bigcup_{G_i \in S} Q_{G_i}.$$

プロトコル P に従うと、任意のコンフィグレーション c において、 V に含まれるプロセスが使っている資源の総数 $|\bigcup_{v \in V} \rho_v(c)|$ はコーラム集合 $\bigcup_{G_i \in S} Q_{G_i}$ のロックされているコーラムの数に等しい。なぜならば、プロセス $v_j \in V$ が資源を獲得するためには、コーラム集合 Q_{v_j} の (ただ) 1 つのコーラムをロックしなければならないから。ここで、ロックとは、あるコーラムに含まれる全てのプロセスから許可メッセージを受け取ることである。

$\bigcup_{G_i \in S} Q_{G_i}$ は条件より $|\bigcup_{G_i \in S} \alpha(G_i)| (= |\alpha(V)|)$ -コートリーであるから、 $\bigcup_{G_i \in S} Q_{G_i}$ のロック可能な、すなわち互いに素なコーラムの数は高々 $|\alpha(V)|$ である。よって、 $|\bigcup_{v \in V} \rho_v(c)| \leq |\alpha(V)|$ を達成する。

また、Non-intersection property により、 $(|\alpha(V)| - |\bigcup_{v_j \in V} \rho_{v_j}(c)|)$ 個の互いに素なコーラムが存在するので、条件 2 も満たされる。□

3.2.2 コーラム集合の構成法

条件を満たすコーラム集合の構成方法の1つを示す.

[構成法]

1. 任意のグループの集合 $S \subseteq \mathcal{G}$ について, $|\bigcup_{G_i \in S} \alpha(G_i)|$ -コートリー Q_S を構成する. 但し, 任意の $S, S' \subseteq \mathcal{G}$ について, $S \neq S'$ ならば, $\forall q \in Q_S, \forall q' \in Q_{S'}, q \cap q' = \emptyset$.
2. グループ G に含まれるプロセスに割り当てられるコーラム集合 Q_G^* を,

$$Q_G^* = \bigoplus_{S \in G} Q_S$$

とする. ここで, $Q \oplus Q' = \{q \cup q' | q \in Q, q' \in Q'\}$ である. □

以下では, 構成法により得られるコーラム集合が条件を満たすことを示す. まず, 我々はコーラム集合の直積演算 $\oplus$ に関する次の補題を得る.

補題 1 $Q_i (i = 1, 2)$ を U_i の下での k_i -コートリーとし, $U_1 \cap U_2 = \emptyset$ とする. このとき, $k = \min\{k_1, k_2\}$ とおけば, $Q_1 \oplus Q_2$ は k -コートリーである.

(証明) 一般性を失うことなく, $k = k_1 \leq k_2$ とする. $Q_1 \oplus Q_2$ が k -コートリーの3つの条件を満たすことを示す.

1) Non-intersection property: $q_1, q_2, \dots, q_h \in Q_1 \oplus Q_2$ を $h (< k)$ 個の任意の互いに素なコーラムとする. $\oplus$ の定義より, 任意の q_i についてある $a_i \in Q_1$ と $b_i \in Q_2$ が存在して, $q_i = a_i \cup b_i$. この時, Q_1 が $k_1 (= k)$ -コートリーであること, また Q_2 が $k_2 (\geq k)$ -コートリーであることより, それぞれの Non-intersection property によって, ある $a_{h+1} \in Q_1$ と $b_{h+1} \in Q_2$ が存在し,

$$a_i \cap a_{h+1} = \emptyset \wedge b_i \cap b_{h+1} = \emptyset, 1 \leq i \leq h.$$

故に, $q_{h+1} = a_{h+1} \cup b_{h+1}$ が存在し, $q_i \cap q_{h+1} = \emptyset, 1 \leq i \leq h$.

2) Intersection property: $k+1$ 個の互いに素なコーラム $q_1, q_2, \dots, q_{k+1} \in Q_1 \oplus Q_2$ が存在すると仮定する. $\oplus$ の定義より, 任意の q_i についてある $a_i \in Q_1$ と $b_i \in Q_2$ が存在して, $q_i = a_i \cup b_i$. この時, $k+1$ 個の互いに素なコーラム $a_1, a_2, \dots, a_{k+1} \in Q_1$ が存在するが, これは Q_1 が k -コートリーであることに矛盾する.

3) Minimality: $\oplus$ の定義より, 任意の $q_i \in Q_1 \oplus Q_2$ についてある $a_i \in Q_1$ と $b_i \in Q_2$ が存在して, $q_i = a_i \cup b_i$. この時, Q_1 が Minimality を保持していることにより $a_i \subseteq a_j$ であるような2つのコーラム $a_i, a_j \in Q_1$ は存在しない. また, Q_2 が Minimality を保持していることにより $b_i \subseteq b_j$ であるような2つのコーラム $b_i, b_j \in Q_2$ は存在しない. 故に, $q_i \subseteq q_j$ であるような2つのコーラム $q_i = a_i \cup b_i, q_j = a_j \cup b_j$ は $Q_1 \oplus Q_2$ には含まれない. □

補題 1 より, 構成法により得られるコーラム集合に関する次の系が得られる.

系 1 任意のグループ $G \in \mathcal{G}$ について, Q_G^* は $|\alpha(G)|$ -コートリーである.

(証明) $S_G = \{S_i | G \in S_i\}$ とする. $k = \min\{k_i : S_i \in S_G\}$ とし, $Q_{\{G\}}$ が k -コートリーであるとする. この時, $|\alpha(G)| = k$ である. なぜなら, グループの任意の集合 $S_i \in S_G$ について,

$$|\alpha(G)| \leq \left| \bigcup_{G_j \in S_i} \alpha(G_j) \right|.$$

故に, 補題 1 により Q_G^* は $|\alpha(G)|$ -コートリーである. □

定理 3 任意のグループの集合 $S \subseteq \mathcal{G}$ について, $\bigcup_{G_i \in S} Q_{G_i}^*$ は $|\bigcup_{G_i \in S} \alpha(G_i)|$ -コートリーである.

(証明) $Q^*(S) = \bigcup_{G_i \in S} Q_{G_i}^*$ とする. $Q^*(S)$ が $|\bigcup_{G_i \in S} \alpha(G_i)|$ -コートリーの 3 つの条件を満たすことを示す.

1) Minimality: 任意の 2 つのコラム $q_1, q_2 \in Q^*(S)$ について, ある 2 つのグループが存在し, $q_1 \in Q_{G_1}^*, q_2 \in Q_{G_2}^*$ とする. つまり, $q_1 \in \bigoplus_{G_i \in S} Q_{G_i}, q_2 \in \bigoplus_{G_i \in S} Q_{G_i}$ である.

$q_1 \subseteq q_2$ かつ $Q_{G_1}^* = Q_{G_2}^*$ とすると, $Q_{G_1}^*$ が Minimality を保持していることに反する.

$q_1 \subseteq q_2$ かつ $Q_{G_1}^* \neq Q_{G_2}^*$ と仮定する. この時, ある $q \in Q_{\{G_1\}}$ が存在して, $q \subseteq q_1$. しかしながら, $q' \in Q_{\{G_1\}}$ であるような $q' \subseteq q_2$ は存在しない. 故に, $q_1 \subseteq q_2$ であるような q_2 は存在せず, 矛盾.

2) Intersection property: $\sigma = |\bigcup_{G_i \in S} \alpha(G_i)|$ とする.

$\sigma + 1$ 個の互いに素なコラム $q_1, \dots, q_{\sigma+1} \in Q^*(S)$ が存在すると仮定する. このとき, $q_j \in Q_{G_i}^*, G_i \in S$ である.

$Q_{G_i}^* = \bigoplus_{G_j \in \mathcal{H}} Q_{G_j}$ であるから, 特に, $\exists p_i \in Q_S, p_i \subseteq q_i$ である. すなわち, $\sigma + 1$ 個の互いに素なコラム $p_i \in Q_S$ が存在することになるが, これは Q_S が σ -コートリーであることに矛盾する.

3) Non-intersection property:

$\rho (< \sigma)$ 個の $q_i \in Q^*(S)$ について考える. ある単射な関数 $f: \{q_i, 1 \leq \rho\} \rightarrow \bigcup_{G_i \in S} \alpha(G_i)$ が存在して, $R^* = \{r_i | r_i = f(q_i)\}$ とする. $\rho < \sigma$ であるので, ある資源 $r \in \bigcup_{G_i \in S} \alpha(G_i) - R^*$ が存在する. さらに, $r \in \bigcup_{G_i \in S} \alpha(G_i)$ であるので, $r \in \alpha(G)$ であるようなグループ G が存在する.

任意の $\mathcal{H} \supseteq \{G\}$ について, $\Phi_{\mathcal{H}} = \{q_{\mathcal{H}} | \exists q_i, q_{\mathcal{H}} \subseteq q_i\}$, $\rho_{\mathcal{H}} = |\Phi_{\mathcal{H}}|$ とすると,

$$\rho_{\mathcal{H}} \leq |\bigcup_{G_i \in \mathcal{H}} \alpha(G_i) - r| < |\bigcup_{G_i \in \mathcal{H}} \alpha(G_i)|.$$

従って, 任意の $\mathcal{H} \supseteq \{G\}$ について, ある $q_{\mathcal{H}} \in Q_{\mathcal{H}}$ が存在し,

$$q_{\mathcal{H}} \cap (\bigcup_{i=1}^{\rho} q_i) = \emptyset.$$

よって, $q \in \bigoplus_{\{G\} \subseteq \mathcal{H}} q_{\mathcal{H}}$ が存在し,

$$q \cap (\bigcup_{i=1}^{\rho} q_i) = \emptyset.$$

□

4 おわりに

本稿で我々は, これまで多く研究されてきた分散 k -相互排除問題のプロセスと資源の共有関係に関する拡張である無名資源競合回避問題を定義した. 我々はこの問題に対し, コラム集合を用いるアプローチを行った. コラムベースプロトコルを定義し, そのプロトコルを用いて競合回避問題を解くためのコラム集合に与える十分条件の 1 つを示した. さらに, その条件を満たすコラム集合の構成方法の 1 つを示した. 残された問題として, より単純なコラム集合の構成方法についての考察があげられる.

参考文献

- [1] H. Garcia-Morina, D. Barbara, "How to Assign Votes in a Distributed System," *Journal of the Association for Computing Machinery*, Vol.32, No.4, pp.841-860, Oct. 1985.
- [2] H. Kakugawa, S. Fujita, M. Yamashita, T. Ae, "Availability of k -Coterie," *IEEE Transactions on Computers*, Vol.42, No.5, pp.553-558, May 1993.
- [3] H. Kakugawa, S. Fujita, M. Yamashita, T. Ae, "A Distributed k -Mutual Exclusion Algorithm using k -Coterie," *Information Processing Letters*(to appear).
- [4] 角川裕次, 山下雅史, "分散システムにおける資源割当てアルゴリズム," 1994 年冬の LA シンポジウム講究録, 1994.

- [5] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, Vol.21, No.7, pp.558-565, July 1978.
- [6] 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.
- [7] 真部義文, 青柳滋己, "分散 k -相互排除問題について," 電子情報通信学会コンピュータ研究会, Vol.COMP 91-13, pp.11-18, May 1993.
- [8] 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.
- [9] 宮本英典, 角川裕次, 山下雅史, "拡張された分散 k -相互排除," 情報処理学会アルゴリズム研究会, 34-2, Aug. 1993
- [10] K. Raymond, "A Distributed Algorithm for Multiple Entries to a Critical Section," *Information Processing Letters*, Vol.30, pp.189-193, Feb. 1989.
- [11] G. Ricart, A. K. Agrawala, "An Optimal Algorithm for Mutual Exclusion in Computer Networks," *Communications of the ACM*, Vol.24, No.1, pp.9-17, Jan. 1981.
- [12] I. Suzuki, T. Kasami, "A Distributed Mutual Exclusion Algorithms," *ACM Transactions on Computer Systems*, Vol.3, No.4, pp.344-349, Nov. 1985.