

格子状ネットワークにおける Rete アルゴリズムの割り当て手法の検討

A Study of Assignment Method of the Rete Algorithm in Grid Topology Network

川上 朋也† 義久 智樹† 塚本 昌彦‡
Tomoya Kawakami Tomoki Yoshihisa Masahiko Tsukamoto

1. はじめに

近年、情報機器の小型化や軽量化、低価格化に伴い、ユビキタスコンピューティング環境が実現されつつある。ユビキタスコンピューティング環境では、環境内の個々のコンピュータを制御することにより、コンピュータ群全体でユーザの行動を支援することが望まれる。現在、コンピュータ群の制御手法が数多く提案されており [1-3]、RuleCaster は各コンピュータの動作をルール形式で記述してコンピュータ群を制御する [4]。ルールベースの制御手法では、複数のコンピュータに及ぶ処理も 1 つのルールで定義でき、複数のルールを用いることで一度に多くの処理を実行できる。ユビキタスコンピューティング環境では膨大な数のコンピュータやサービスを扱う必要があり、ルールの入れ替えによって柔軟かつ適応的に動作するルールベースシステムは有効性が高い。

現在、ルールベースシステムが数多く提案されており、これら既存手法の多くは IF-THEN 形式のルールを想定している。IF-THEN 形式のルールはプロダクションシステム [5] によって扱うことができ、代表的な照合アルゴリズムに Rete アルゴリズムがある [6]。また、Rete アルゴリズムの処理を複数のコンピュータで行う手法も提案されており [7]、著者らも関連する手法を提案している [8, 9]。ユビキタスコンピューティング環境では膨大な数のコンピュータやサービスを扱う必要があり、この分散処理モデルは可用性やスケラビリティの点で有効であると考えられる。

そこで本研究では、ルールに基づくコンピュータ群の効率的な制御を目的とし、Rete アルゴリズムをコンピュータ群に割り当てる手法を検討する。本研究のシステムモデルを図 1 に示す。本研究では、コンピュータ群が格子状ネットワークを構築し、各コンピュータとその位置を $P(x_i, y_i)$ で表す。各コンピュータや管理端末は自身の位置や隣接しているコンピュータの方向を把握しており、他のコンピュータが中継することで、任意の位置のコンピュータへメッセージを送信できる。また、コンピュータには図 1 に S_i で示すセンサや A_i で示すアクチュエータが接続される。センサは温度計や照度計、人感センサなどを想定し、定期的にセンサデータを生成する。アクチュエータは電灯やエアコン、ロボットなどの駆動装置を想定し、接続しているコンピュータから操作できるものとする。また、各コンピュータや管理端末は、センサやアクチュエータが接続されているコンピュータの位置を把握しているものとする。本研究では、ルールとセンサデータに基づいてアクチュエータを適応的に動作させるため、アクチュエータへのセンサデ

†大阪大学サイバーメディアセンター, Cybermedia Center, Osaka University

‡神戸大学大学院工学研究科, Graduate School of Engineering, Kobe University

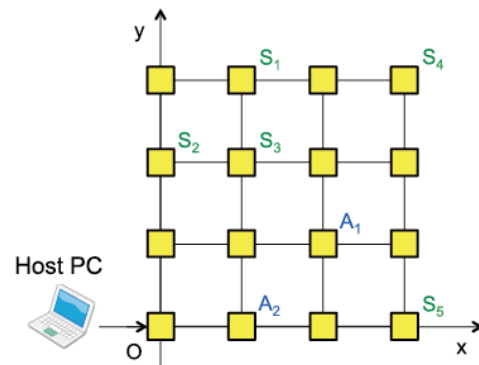


図 1: システムモデル

ータ配信やルール処理に Rete アルゴリズムを適用する。

以下、2 章で Rete アルゴリズムについて述べる。3 章で検討手法を説明し、4 章で評価を行う。最後に 5 章で本稿をまとめる。

2. Rete アルゴリズム

IF-THEN 形式のルールに基づいて推論するプロダクションシステムは、ルールを記憶するプロダクションメモリ (PM)、ワーキングメモリエレメント (WME) と呼ばれるデータを記憶するワーキングメモリ (WM)、推論エンジンと呼ばれる制御プログラムから構成される。

プロダクションシステムでは条件照合の高速化に関するさまざまな研究が行われており、Forgy によって提案された Rete アルゴリズムは代表的な条件照合アルゴリズムである [3]。Rete アルゴリズムは、Rete ネットワークと呼ばれるグラフ構造をルールに基づいて作成する。Rete ネットワークでは、入力データをルールの条件によってフィルタリングする節 (ノード) を α ノード、 β ノードと呼ぶ。 α ノードは、1 つの親からの入力データを条件によってフィルタリングする。一方、 β ノードは、2 つの親からの入力データを結合し、結合データを条件によってフィルタリングする。この β ノードにおける処理は “join” と呼ばれる。これら α ノード、 β ノードで条件を満たしたデータは下流のノードへ送られる。Rete アルゴリズムは α ノードや β ノードでの条件照合において、条件を満たすデータあるいはデータの組合せを、それぞれ α メモリ、 β メモリとして記憶しておき、データが追加や削除された場合に必要となる条件照合処理を削減している。

Rete アルゴリズムについて、図 1 に示したルール例を用いて説明する。図 1 は「人が存在すれば、電灯を点ける」というルールである。図 1 のルールから生成される Rete ネットワークを図 2 に示す。ルールを満たすコンテンツの

```

for actuator a, sensor_data s
if a.type = "light" and
  a.status = "OFF" and
  s.type = "human detection" and
  s.value = true
then
  turn a on

```

図 2：擬似コードによるルールの例

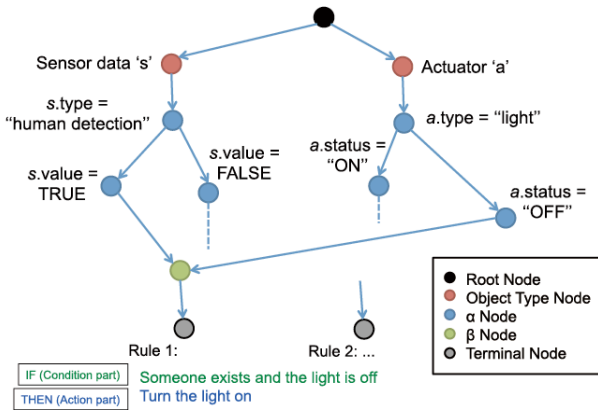


図 3：ルールから生成される Rete ネットワーク

抽出は、図 2 の最上流に“Root Node”で示したノードからアクチュエータやセンサのデータを流すことで行われる。データはまず種類に応じて、下流の Object Type ノードと呼ばれる“sensor data s”，“actuator a”へと振り分けて流される。同様に各ノードで示した条件を満たすデータのみをさらに下流のノードへ流すことで、最終的にルールを満たすデータの組みが図 2 の最下流に“Terminal Nodes”で示したノードに流される。Terminal Node はルールごとに存在し、ルールを満たすデータは対応する Terminal Node を確認することで取得できる。図 2 の Terminal Node では、左側に「人が存在し、点いていない電灯」のデータが流される。このとき、 α ノード、 β ノードには、処理結果がそれぞれ記憶されている。例えば、「人が存在しない」という条件を含むルールを新たに追加する場合、“s.value = FALSE”で示した α ノードの下流にノードを追加することでこの α ノードがもつ α メモリの内容を取得でき、共通する条件に対する処理を抑制できる。

Rete アルゴリズムでは、データ更新が頻繁に発生する環境を想定し、 α メモリや β メモリを保持しない TREAT アルゴリズム[10]や、条件照合の計算コストが高いルールに対処するための手法[11]が提案されている。また、条件照合を高速化するためのワーキングメモリのデータ構造も提案されている[12]。さらに、Rete アルゴリズムを分散環境の Publish/Subscribe (Pub/Sub) システムへ適用した手法として、DHT (Distributed Hash Table) の一種である Chord[13]を用いて α メモリを共有する手法が提案されており[7]、著者らは α メモリだけでなく β メモリも共有する手法を提案している[8, 9]。

3. 検討手法

本研究では、Rete アルゴリズムを格子状ネットワークのコンピュータ群へ割り当てる方法を検討する。

3.1 アイデア

本研究の想定環境では、センサやアクチュエータは特定のコンピュータに接続される。本研究では、ルールやセンサデータに基づいてアクチュエータを動作させるため、ルール処理に必要なデータをアクチュエータの接続しているコンピュータが他のコンピュータから収集する方式を考える。このとき、ルール処理に必要なセンサからすべてのデータを収集すると、ルールの条件を満たさないデータが含まれるため、不要なトラフィックがネットワークに生じる。そこで、Rete アルゴリズムに基づいて条件を満たすデータのみ配信するため、Rete アルゴリズムにおける α ノードと β ノードの役割をコンピュータ群に割り当てる。

本研究では、IF-THEN ルールの条件部 (Condition part) を構成する要素を条件要素と呼び、 E_i で表す。条件要素は図 1 の「s.type = “human detection”」などで、それぞれ特定のセンサデータに関連する。IF-THEN ルールの動作部 (Action part) となるアクチュエータは、処理するルールの条件要素に関連するセンサからデータを配信される。配信経路は配信元のセンサと配信先のアクチュエータの間で構築され、ホップ数はマンハッタン距離 $d(P(x_1, y_1), P(x_2, y_2))$ で表される。このとき、 β ノードにおける join は複数の条件要素の結果から行われるため、複数のセンサデータの配信経路が重なる場合、重なった位置のコンピュータが部分的なルールを処理し、条件を満たすデータのみ中継することで、不要なトラフィックの発生を抑制できる。例えば、2つのセンサ S_1, S_2 の時間的なデータ量をそれぞれ n_1, n_2 で表し、それぞれのセンサデータが関係する条件要素 E_1, E_2 が条件を満たす時間的な適合率 (Precision) を p_1, p_2 ($0 \leq p_1 \leq 1, 0 \leq p_2 \leq 1$) とする。2つのセンサデータを条件要素とするルールがあり、それぞれの条件要素が確率的に独立とすると、最終的にアクチュエータへ配信されるデータ量は以下に削減される。

$$p_1 p_2 (n_1 + n_2)$$

また、配信経路上のコンピュータが処理結果を α メモリや β メモリとして保持し、同様の条件要素や処理結果を別の宛て先へ配信する場合に利用することで、配信元のセンサからのトラフィックの発生を抑制できる。

3.2 配信経路の決定

ルールの動作部となるアクチュエータ A_1 の位置を $P(x_0, y_0)$ とする。ルールの条件部の条件要素が 1 つの場合、配信経路はアクチュエータとセンサ S_1 の位置 $P(x_1, y_1)$ の最短距離 $d(P(x_0, y_0), P(x_1, y_1))$ となる経路のうちから選択する。一方、ルールの条件部の条件要素が 2 つで、関連するセンサからの配信経路も 2 つの場合、2 つの配信経路を結合する位置までの距離が短いほど、それ以降の中継による多くのトラフィックを削減できる。条件要素が 3 つ以上の場合も 2 つずつの配信経路の結合を再帰的に行う。

2 つの条件要素 E_i, E_j が関連するセンサをそれぞれ S_i, S_j 、位置を $P(x_i, y_i), P(x_j, y_j)$ とする。配信経路の結合には、アクチュエータ A_k の位置 $P(x_k, y_k)$ の $x = x_k, y = y_k$ の 2 直線を基準にし、アクチュエータと各センサの位置によって得られる領域 $R(P(x_k, y_k), P(x_i, y_i)), R(P(x_k, y_k), P(x_j, y_j))$ について、以下の 4 つの場合で考える。

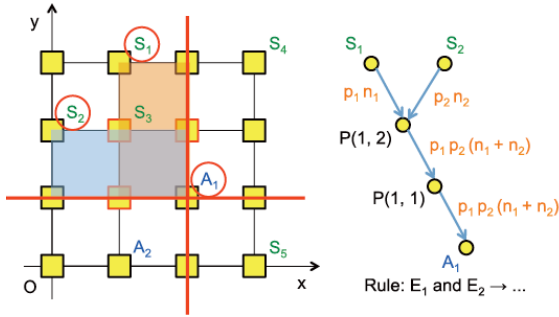


図4：条件部が E_1 かつ E_2 における例

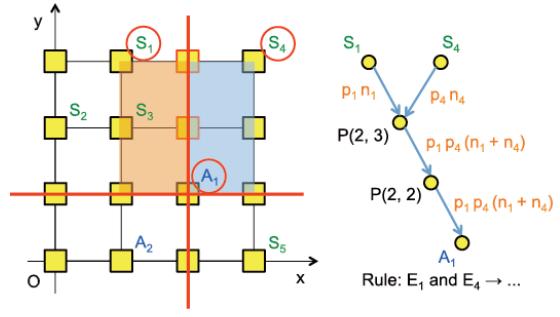


図6：条件部が E_1 かつ E_4 における例

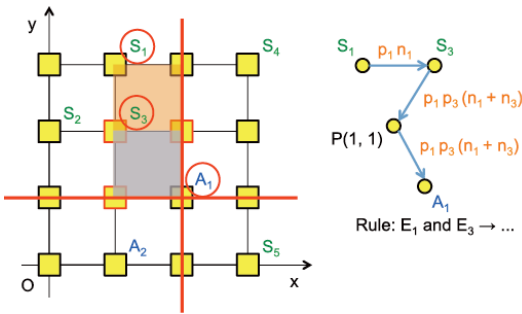


図5：条件部が E_1 かつ E_3 における例

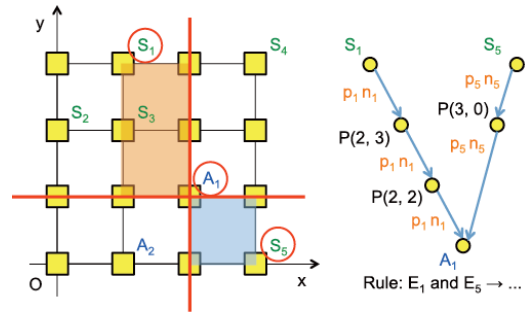


図7：条件部が E_1 かつ E_5 における例

- ・ 交差する場合
- ・ 包含する場合
- ・ 隣接する場合
- ・ 点で接する場合

交差する場合

図1のセンサ S_1, S_2 の場合を図4に示す。図4の場合、アクチュエータ A_1 と S_1 の領域 $R(P(2, 1), P(1, 3))$, S_2 の領域 $R(P(2, 1), P(0, 2))$ は交差する。このとき、 $P(1, 1), P(1, 2), P(2, 2)$ が最短経路に含まれるが、最も配信経路を結合するまでの距離が短いのは $P(1, 2)$ である。

包含する場合

同様に、センサ S_1, S_3 の場合を図5に示す。図5の場合、アクチュエータ A_1 と S_1 の領域 $R(P(2, 1), P(1, 3))$ は S_3 の領域 $R(P(2, 1), P(1, 2))$ を包含する。このとき、 $P(1, 1), P(1, 2), P(2, 2)$ が最短経路に含まれるが、最も配信経路を結合するまでの距離が短いのは $P(1, 2)$ である。検討手法による効果は最も高い。

隣接する場合

センサ S_1, S_4 の場合を図6に示す。図6の場合、アクチュエータ A_1 と S_1 の領域 $R(P(2, 1), P(1, 3))$ と S_4 の領域 $R(P(2, 1), P(3, 3))$ は隣接する。このとき、 $P(2, 2), P(2, 3)$ が最短経路に含まれるが、最も配信経路を結合するまでの距離が短いのは $P(2, 3)$ である。

点で接する場合

センサ S_1, S_5 の場合を図7に示す。図7の場合、アクチュエータ A_1 と S_1 の領域 $R(P(2, 1), P(1, 3))$ と S_5 の領域 $R(P(2, 1), P(3, 0))$ は A_1 の位置 $P(2, 1)$ でのみ接する。検討手法による効果は最も低い。

条件要素が2つ以上の場合は、上記の方針に基づき、近接性が高い領域から順に配信経路を結合する。また、複数

のアクチュエータが処理するルールで条件要素が共通する場合、配信経路上のコンピュータが配信経路を分岐する。アクチュエータ A_1, A_2 , センサ S_1 の場合を図8に示す。図8の場合、 S_1 と A_1 の領域 $R(P(2, 1), P(1, 3))$ と A_2 の領域 $R(P(1, 0), P(1, 3))$ は $P(1, 1)$ で交差し、そのコンピュータが分岐して配信する。配信経路を分岐する場合は、2つの配信経路を結合する場合と逆に、配信先までの距離が短い位置ほどトラフィックを削減できる。図1の例においてアクチュエータ A_1, A_2 がそれぞれ以下のルールを処理する場合、配信経路とトラフィックを図7に示す。

- ・ $A_1: E_1 \text{ and } E_2 \text{ and } E_3 \text{ and } E_4 \text{ and } E_5 \rightarrow \dots$
- ・ $A_2: E_1 \text{ and } E_2 \text{ and } E_5 \rightarrow \dots$

4. 評価

本研究では、3章の提案手法をシミュレーションにより評価した。

4.1 シミュレーション環境

シミュレーション環境を表1に示す。本シミュレーションでは図9のモデルでシミュレーションを行い、各センサのデータ量 n_i はすべて同一とした。また、用いたルールは2種類で、ルールの条件要素 E_i の適合率 p_i はすべて同一とし、0.1~1.0で変化させた。

比較手法として、ルール処理に必要なセンサデータを各センサがアクチュエータへ個別に配信する場合を用いた。比較手法では、各センサは対象となるアクチュエータに対して、適合率 p_i で条件を満たすセンサデータのみを個別に配信する。対象となるアクチュエータが複数の場合でもそれぞれ個別に配信し、配信経路は共有しない。

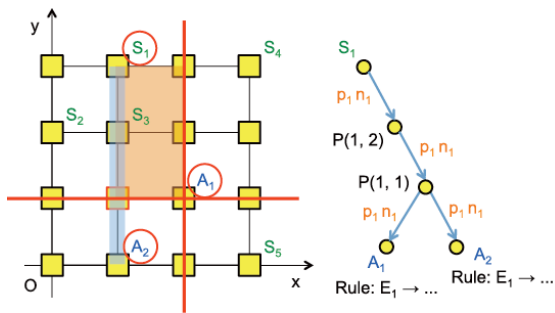


図 8：動作部が A_1 および A_2 における例

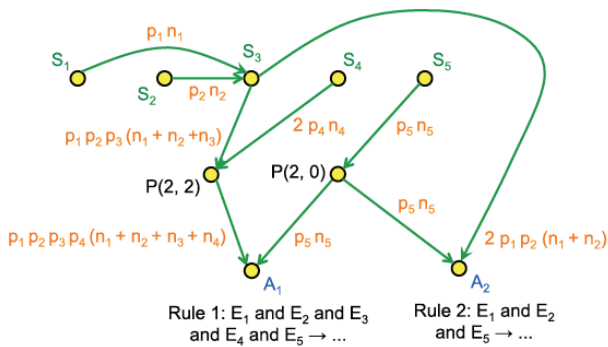


図 9：図 1 における配信経路の例

4.2 トラフィック量および削減率

本シミュレーションでは、図 7 の例でコンピュータ間のトラフィック量を測定した。トラフィック量の結果を図 10 に、比較手法から提案手法への削減率を図 11 に示す。

図 10、図 11 より、提案手法では適合率が低いほど比較手法からの削減率は高いが、トラフィック量は適合率が 0.5~0.8 で特に大きく抑えられている。これは、適合率が低い場合は配信データ量が小さくなり、提案手法によって抑えられる不要なデータ配信による効果も小さくなるためである。一方、適合率が高い場合は条件を満たさないデータの割合が低くなり、提案手法によって抑えられる不要なデータ配信が少なくなる。

表 1：シミュレーション環境

コンピュータ群	4×4
センサデータ	S_1, \dots, S_5
アクチュエータ	A_1, A_2
条件要素および関係するセンサデータ	$E_1: S_1$... $E_5: S_5$
ルールの条件部	$A_1: E_1 \text{ and } E_2 \text{ and } E_3$ and $E_4 \text{ and } E_5 \rightarrow \dots$ $A_2: E_1 \text{ and } E_2 \text{ and } E_5 \rightarrow \dots$
各センサのデータ量	$n_1 = n_2 = \dots = n_5$
各条件要素の適合率	$p_1 = p_2 = \dots = p_5$ 0.1 ~ 1
各オブジェクトの配置	図 1 を参照
データ配信経路	図 9 を参照

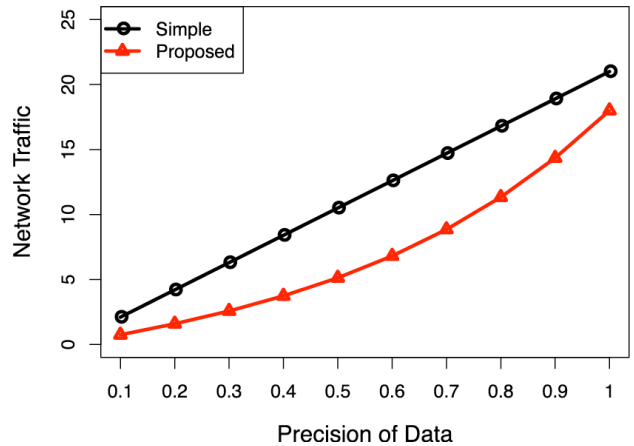


図 10：データ適合率によるトラフィック量

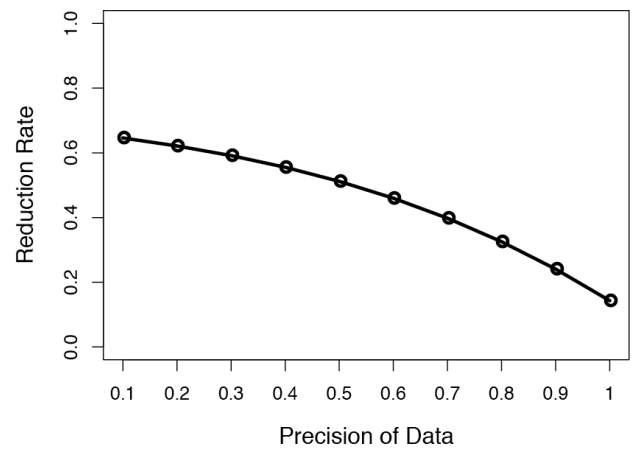


図 11：データ適合率による削減率

5. まとめ

本研究では、ルールに基づくコンピュータ群の効率的な制御を目的とし、Rete アルゴリズムをコンピュータ群に割り当てる手法を検討した。検討手法はシミュレーションにより評価し、ルール処理に必要なセンサデータや処理結果の配信経路を結合することで、トラフィックが抑制できることを確認した。

今後の課題としては、コンピュータやセンサ、アクチュエータ、ルールの膨大な環境での評価が挙げられる。

謝辞

本研究の一部は、科学研究費補助金（基盤研究 A）「ユビキタス環境のためのトポロジコーディングによる全体プログラミング」（課題番号：23240010）、および、独立行政法人科学技術振興機構、京都地域スーパークラスタープログラム「クリーン・低環境負荷社会を実現する高効率エネルギー利用システムネットワークの構築」、科学研究費補助金（若手研究 A）「次世代オンデマンド型視聴形態のためのコンテンツ配信方式」（課題番号：23680007）による成果である。

参考文献

- [1] R. Gummadi, O. Gnawali, R. Govindan: Macro-Programming Wireless Sensor Networks Using Kairos, Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2005), pp. 126-140 (2005).
- [2] R. Newton, G. Morrisett, M. Welsh: The Regiment Macroprogramming System, Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN 2007), pp. 489-498 (2007).
- [3] 國本慎太郎, 藤田直生, 佐野渉二, 寺田 努, 塚本昌彦: モバイルエージェントを用いた格子状ネットワークを構成するユビキタスコンピュータ群の制御, 情報処理学会論文誌, Vol. 14, No. 2, pp. 1697-1708 (2013).
- [4] U. Bischoff and G. Kortuem: Rulecaster: A Macroprogramming System for Sensor Networks, Proceedings of the 21th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2006) Workshop on Building Software for Sensor Networks (2006).
- [5] 小林重信: プロダクションシステム, 情報処理, Vol. 26, No. 12, pp. 1487-1496 (1985)
- [6] C. L. Forgy: Rete: A Fast Algorithm for the Many Patterns/Many Objects Match, Artificial Intelligence, Vol. 19, No. 1, pp. 17-37 (1982).
- [7] Y. Shvartzshnaider, M. Ott, and D. Levy: Publish/Subscribe on Top of DHT Using RETE Algorithm, Proceedings of the 3rd Future Internet Symposium (FIS 2010), pp.20-29 (2010).
- [8] T. Kawakami, T. Yoshihisa, N. Fujita, and M. Tsukamoto: A Rule-Based Home Energy Management System Using the Rete Algorithm, Proceedings of the 2nd IEEE Global Conference on Consumer Electronics (GCCE 2013), pp. 162-163 (2013).
- [9] T. Kawakami, T. Yoshihisa, N. Fujita, and M. Tsukamoto: An Evaluation and Implementation of Rule-Based Home Energy Management System Using the Rete Algorithm, The Scientific World Journal, Vol. 2014, No. 591478, pp. 1-8 (2014).
- [10] D. P. Miranker: TREAT: A Better Match Algorithm for AI Production Systems, Proceedings of the 6th National Conference on Artificial Intelligence (AAAI'87), pp. 42-47 (1987).
- [11] 木村春彦, 住吉一之, 小林真也, 武部 幹: プロダクションシステムの直接条件照合アルゴリズム, 電子情報通信学会論文誌 D-II, Vol. J77-D-2, No. 2, pp. 370-379 (1994).
- [12] 南保英孝, 木村春彦, 広瀬貞樹: プロダクションシステムにおけるジョイン演算の順序に関する一考察, 電子情報通信学会論文誌 D-II, Vol. J80-D-2, No. 10, pp. 2790-2799 (1997).
- [13] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications, IEEE/ACM Trans. on Networking, Vol. 11, No. 1, pp. 17-32 (2003)