知識工学と人工知能 64 － 7
（１９８９．５．１８）

遅延照合アルゴリズムによるルール型計算システム実行方式

吉瀬 隆　　小泉 忍　　山野 紘一

(株) 日立製作所　システム開発研究所

ルール型計算システムではプログラムをルールの集合として記述する。ルールは条件部と実行部とから成り、条件部を満たすルールを選択して実行するという一連の処理を繰り返す。プロダクションシステムでは同じインスタンシエーション（ルールとその条件を満たすデータ組の対）を２度実行しないが、ルール型計算システムでは同じインスタンシエーションであっても条件が成立する限り何度でも繰り返し実行する。

ルール型計算システムの実行方式として、計算効率の良い遅延照合アルゴリズムを考案した。その特徴はデータ集合の検索を必要とする限定条件についてのみ計算結果を保存し、ルールの条件を３値評価することにある。

Lazy Matching Algorithm in Rule Based Computing System

Takashi Kise　　Shinobu Koizumi　　Koichi Yamano

Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji, Asao-Ku, Kawasaki-Shi, Kanagawa-Ken 215 Japan

A program is described by set of rules which have condition part and action part in a rule based computing system. This system is different from the production system in the following point. The production system doesn't apply the same instanciation (i.e. a pair of a rule and its satisfying data) twice. On the contrary, the rule based computing system can execute the same rule many times as far as its condition part is satisfying.

In this report, lazy matching algorithm is proposed. This is an efficient algorithm for a rule based system. An each condition of rules is divided into the primary conditions and the qualified conditions. The primary condition can be evaluated without searching in the database, but the qualified condition cannot. Only the results of the qualified condition are kept and re-used beyond the execution of a rule. And conditions of rule are evaluated by the 3-valued logic.

１．　まえがき

専門家の知識処理をコンピュータに代行させるエキスパートシステム作成の一手法として、プログラムをルールで記述するプロダクションシステムが用いられている。プロダクションシステムは与えられたルールについて条件照合（与えられたデータにより各ルールの条件が成立しているかを判定）し、成立するルールの１つを選択し、実行する。

この条件照合アルゴリズムの１つにＲＥＴＥ（文献［１］）がある。ＲＥＴＥは１度条件照合を行なって得た各条件を満たすデータを保存することで、条件判定回数を減らし高速化することを狙っている。しかし、ルールの実行によりデータが変更される場合に保存データの更新処理のために逆に計算量が増えてしまうという問題がある。

そこで、単独のデータで条件照合できるテストの結果について保存データを持ち、複数のデータを組み合わせた条件のテストの中間結果を保存せず、最終結果として条件を満たすデータ組を保存するというＴＲＥＡＴ［２］が提案されている。更には全くデータを保存しない方式がＲＥＴＥより性能が良いという報告［３］もある。

我々は条件が成立する限りルールを繰り返し実行するルール型計算システムにおいて、その計算効率の良い実行方式である遅延照合アルゴリズムを提案する。

２．　ルール型計算システム

２．１　動作原理

ルール型計算システムでは、図１に示すようにプログラムを条件部と実行部を持つルールの集合として記述する。条件部の成立するルールのうちの１つが選ばれ、そのルールの実行部が実行される。この一連の処理を条件部の成立するルールがある限り繰り返す。条件部の成立するルールが複数あるときには、その中のどれが実行されるかは処理系に任される。

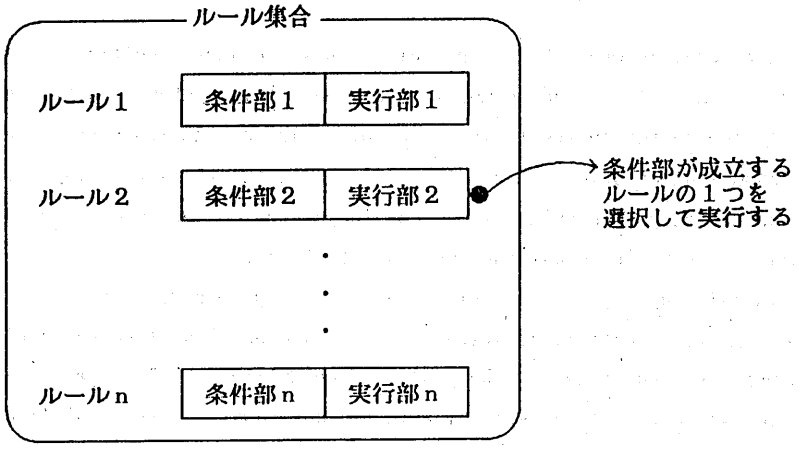

図１　ルール集合とルール型計算システム

2．2　プロダクションシステムとの相違点

プロダクションシステムとルール型計算システムの概要を、それぞれ図２、図３に示す。

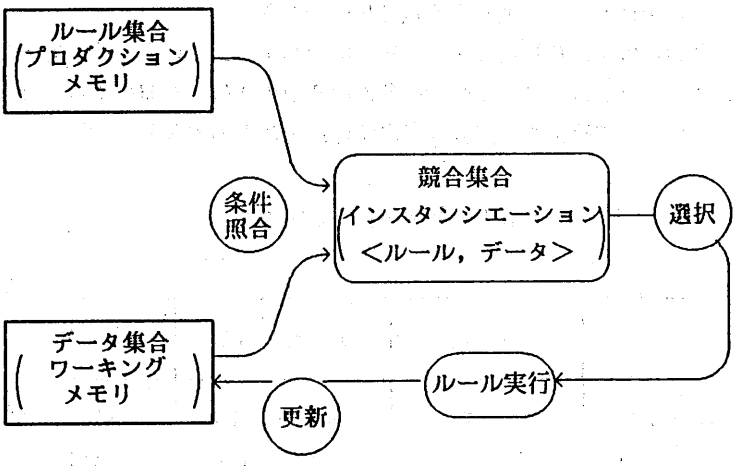

図２　プロダクションシステム

プロダクションシステム（図２）ではインスタンシエーション（ルールとその条件を満たすデータの対）を選択の対象とし、選択したインスタンシエーションに基づいてルールを実行する。また、一度選択したインスタンシエーションは、再度選択の対象とせず同じインスタンシエーションによって２度実行しない。

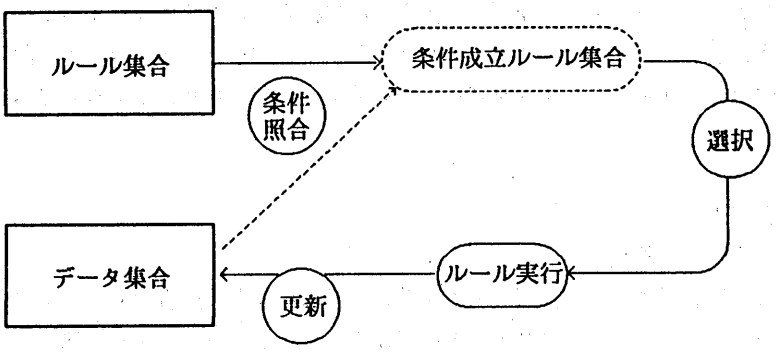

図３　ルール型計算システム

一方、ルール型計算システム（図３）では選択の対象がルールであり、ルールが成立している限りそのルールは実行可能である。そのため、同一のルール・データ対で複数回実行することも有りうる。（即ち、同じインスタンシエーションであっても何度でも繰り返し実行する。）

２．３　システムの特徴

本稿のルール型計算システムは、手続き型言語をベースにルール集合を記述する機能を付加したものである。そのため、以下の特徴を持つ。

（１）ルールの条件として、通常の変数の比較ができる。

（２）データ集合の中から特定のデータを「値のパターン」により検索できる。

従って、表１に示すように、ルールの条件は変数の比較・参照による部分（基本条件）と値パターンによって検索する部分（限定条件）とに大別できる。

表１　基本条件と限定条件

	対　象	アクセス方法	条件式
基本条件	変　数	アドレスによる 直接操作	比較、その他 論理演算
限定条件	データ集合 要素	条件による検索	限定式 Some,None,Any

基本条件については、データ検索のオーバヘッドを伴わない。また、限定条件として、ある条件（値パターン）を満たすデータをデータ集合中から検索するＳｅｌｅｃｔ限定条件、ある条件を満たすデータがデータ集合中に存在するかをチェックするＳｏｍｅ限定式、条件を満たすデータがデータ集合中に存在しないかをチェックするＮｏｎｅ限定式、データ集合中のデータすべてが条件を満たすかをチェックするＡｎｙ限定式を導入している。

３．　実行方式

３．１　高速化の着眼点

ルール型計算システムの実行で最も時間がかかるのは条件照合である。そのうちでも特に条件を満たすデータがデータ集合中に存在するかどうかを検索する処理時間が大部分を占める。そこで、データ検索を減らすために次の２点に着目した。

（１）限定条件における検索結果の保存

限定条件においては、条件を満たすデータの検索結果を保存する。ここで、保存する結果は、Ｓｅｌｅｃｔ限定条件、及び、Ｓｏｍｅ限定条件の場合は、条件を成立させるデータ、Ｎｏｎｅ、及び、Ａｎｙ限定条件では条件を成立させないデータそれぞれ１組である。

高速化のために一度条件照合を行なった時の結果を保存するのは、条件判定に要する時

間が大きくデータ変更に伴う保存結果の更新のオーバヘッドが小さい時に有効である。

限定条件については、通常データ集合が大きく、検索自体の手間が結果の更新のオーバヘッドを上回ると考えられる。

ただし、限定条件が複数データの組で定義されているときに、データ検索は多重ループの処理となる。データ組の１つ１つに対して検索済みという情報を持つと、データ数のＮ乗に比例して、記憶容量が必要となる。このため本方式では検索でループ最外となるデータについてのみ検索済みの情報を持つ。

一方、基本条件については、通常単純な比較式が多いと考えられるため、結果を保存せず、毎回評価する方式とした。

（２）３値論理値を用いた遅延照合

ルールの条件部の論理式の演算の組み合わせによって全条件を計算しなくてもその条件部の真偽が求まることがある。また、ルールの実行によりデータが変更されると結果を保存している限定条件について再計算が必要となる場合がある。

そこで限定条件については、遅延照合を行う。即ち、あるルールの条件部において、その条件部を構成する条件のうち限定条件についてはルール条件部の真偽判定で必要とされるまで、その限定条件の真偽判定を保留しておく。（これにより、オーバヘッドの大きいデータ検索の回数が減ることを期待している。）

そのため２値論理の「真」「偽」の値のほかに、真偽を判定していないことを示す「不明」という値を入れた３値論理値を考える。

なお、上記を行うため、ルールの条件部の評価には「副作用が伴わない」ことを前提とする。

以上の２つの特徴を表２に示す。

表２　遅延評価アルゴリズムの特徴

	保存情報	条件式の評価
基本条件	な　し	毎回計算
限定条件	・３値論理値 ・条件成立データ１組 ・最外ループのデータ 検索済範囲	・２値評価の時のみ データ集合を検索

３．２　遅延照合アルゴリズム

ルール型計算システムにおける遅延照合アルゴリズムの概要を図４に示す。

ステップ０：限定条件の３値論理値、ルールの優先度などの情報を初期化する。

ステップ１：全ルールの条件部を３値評価する。

ステップ２：「偽」でないルールが存在する限り２．１～２．６の処理を繰り返す。

２．１：優先度最大のルールを選択し、条件部の２値評価を行なう。（この時に初めて、必要な限定条件の真偽を判定する。）

２．２：選択したルールが「偽」であるかぎり、優先度の高い順にルールを選択し、条件部の２値評価を繰り返す。

２．３：選択したルールが「真」であればそのルールを実行する。

２．４：限定条件に関する保存情報として、２．３のルール実行により変更・削除されたデータを参照しているもの、或いはデータの追加によって論理値が変わる可能性のあるものについて、その限定条件の３値論理値を「不明」にする。

２．５：ルールの優先度の更新を行う。

２．６：全ルールの条件部を３値評価する。

ステップ３：全てのルールの条件部が「偽」であれば、処理を終了する。

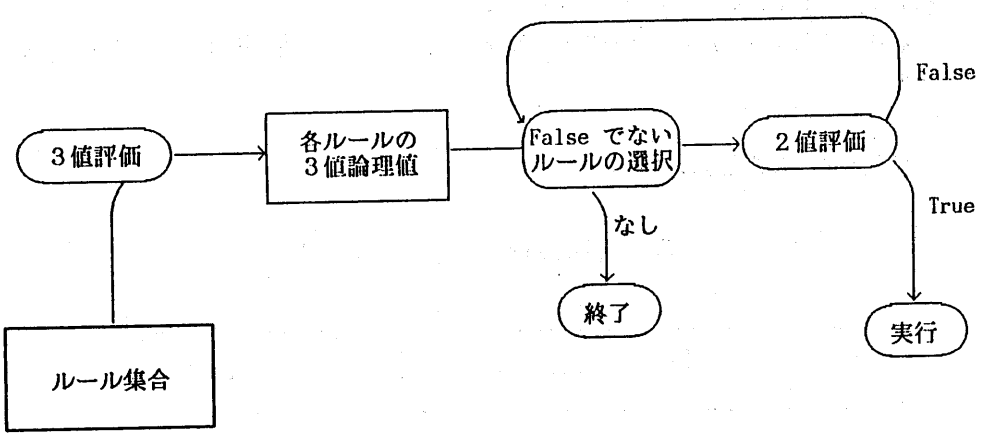

図４　遅延評価アルゴリズムの概要

4．　例題によるアルゴリズムの説明と評価

4．1　例題

表に数字が書かれたカードが裏にして置いてある。一人で”神経衰弱”を行なうプログラムを考える。すなわち２枚表にして同じ数であれば取り除き、異なる数であれば裏に戻す。

図５にプログラムの概要を示す。一度表になったカードの数字はデータ集合の要素として覚えておく。

変数
　表になっているカードの枚数

データ集合
＜カードの位置、表裏、数字（判らないときは０）＞
　　の組を要素とするデータ集合。

ルール集合

ルール１：表になっているカードが０枚で、（裏になっているカードで）
　　同じ数であると判っている２枚のカードがある。
　⇒　そのうちの１枚を表返す。

ルール２：表になっているカードが０枚で、（裏になっているカードで）
　　同じ数であると判っている２枚のカードがない。
　　数の判らないカードｃがある。
　⇒　ｃを表返す。

ルール３：表になっているカードが１枚で、
　　表のカードと同じ数であると判っている裏のカードがある。
　⇒　そのカードを表返す。

ルール４：表になっているカードが１枚で、
　　表のカードと同じ数であると判っている裏のカードがない。
　　数の判らないカードｃがある。
　⇒　ｃを表返す。

ルール５：表になっているカードが２枚である。
　⇒　２枚の数が同じであれば、その２枚を取り除く。
　　数が同じでなければ、裏に返す。
　　表のカード枚数を０とする。

手続き

カードを表に返す＝　表のカード枚数を１増やし、データ集合中のカードに対応する要素において、状態を表にして、判った数字を設定する。

カードを取り除く＝　データ集合からカードに対応する要素を削除する。

裏に返す＝　データ集合中のカードに対応する要素において、状態を裏にする。

図５　例題プログラム（神経衰弱）の概要

４．２　遅延照合アルゴリズムの動作の説明

例題のルール集合から基本条件、限定条件、ルールの条件部を抜き出したものが図６である。表３は各条件の論理値が実行とともに変化する様子を示したものである。表中のＴ，Ｆ，Ｕはそれぞれ「真」「偽」「不明」の論理値を示す。

基本条件

Ｐ０＝「表のカードが０枚である。」
Ｐ１＝「表のカードが１枚である。」
Ｐ２＝「表のカードが２枚である。」

限定条件

Ｑ１＝「同じ数であると判っている２枚のカードがある。」
Ｑ２＝「数の判らないカードがある。」
Ｑ３＝「表のカードと同じ数であると判っている裏のカードがある。」
Ｑ４＝「２枚の表のカードがある。」

ルールの条件部

Ｒ１＝　Ｐ０　and　Ｑ１
Ｒ２＝　Ｐ０　and　（not　Ｑ１）　and　Ｑ２
Ｒ３＝　Ｐ１　and　Ｑ３
Ｒ４＝　Ｐ１　and　（not　Ｑ３）　and　Ｑ２
Ｒ５＝　Ｐ２　and　Ｑ４

図６　例題におけるルール条件部の構成

初めは基本条件のＰ０がＴでＰ１，Ｐ２がＦである。限定条件は全てＵである。従って、ルールの条件部はＲ１，Ｒ２がＵで他はＦである。

論理値がＦ以外のルールの中からＲ１が選ばれたとする。２値評価で限定条件Ｑ１が計算されＦとなり、Ｒ１がＦと判明した。

次にＲ２が選ばれて２値評価される。Ｑ２が計算されＴとなり、Ｒ２がＴとなった。

ルールＲ２が実行された後、データ集合の要素が変更されてＱ１，Ｑ２がＵになる。Ｐ１がＴでＰ０，Ｐ２はＦとなった。従ってＲ３，Ｒ４がＵで他はＦとなる。

論理値がＦ以外のルールの中からＲ３が選ばれたとする。２値評価で限定条件Ｑ３が計算されＦとなり、Ｒ３がＦと判明した。

次にＲ４が選ばれて２値評価される。Ｑ２が計算されＴとなり、Ｒ４がＴとなった。

ルールＲ４が実行された後、データ集合の要素が変更されてＱ２，Ｑ３がＵになる。Ｐ２がＴでＰ０，Ｐ１はＦとなった。従ってＲ５がＵで他はＦとなる。

このようにして、全てのルールの条件がＦになるまで処理を繰り返す。

表３　アルゴリズム動作の様子

	P0	P1	P2		Q1	Q2	Q3	Q4		R1	R2	R3	R4	R5
初期	T	F	F		U	U	U	U		U	U	F	F	F
R１					F					F				
R２						T					T			
Act R2	F	T	F		U	U				F	F	U	U	F
R３							F					F		
R４						T							T	
Act R4	F	F	T			U	U			F	F	F	F	U
R５								T						T
Act R5	T	F	F					U		U	U	F	F	F
R１					F					F				
R２						T					T			
Act R2	F	T	F		U	U				F	F	U	U	F
R３							T					T		
Act R3	F	F	T				U			F	F	F	F	U
R５								T						T
Act R5	T	F	F					U		U	U	F	F	F

４．３　評価

限定条件の計算はデータ集合中のデータを検索するため、データ数が多く、そのうちで条件を満たすデータが少ないような場合には、限定条件の計算により時間がかかるため、それに比べて基本条件の計算時間は無視できる。

そこで、限定条件の計算回数で遅延照合アルゴリズムの性能評価を行なう。

この例題ではルールの実行によりＰ０，Ｐ１，Ｐ２の順で繰り返して基本条件がＴになる。Ｐ０＝Ｔの時には、最善でＱ１のみ、最悪でＱ１，Ｑ２が計算される。Ｐ１＝Ｔの時には、最善でＱ３のみ、最悪でＱ３，Ｑ２が計算される。Ｐ２＝Ｔの時には、Ｑ４が計算される。各段階でＱ１～Ｑ４の全てを計算するのに比べて、限定条件の計算回数は、最善で２５％、最悪で４２％となる。

さらにルール数が多く、基本条件の評価だけで計算すべき限定条件の絞り込みの度合いが高いような問題であれば、遅延照合アルゴリズムの効果はより大きくなる。

5．　他方式との比較

ＲＥＴＥ［1］では条件照合の途中の計算結果を全て保存する。毎回同じ条件照合を繰り返すのに比べて条件照合回数を減らしているが、ルールの実行によりデータが変更されてしまうと中間結果を削除するためのオーバヘッドがかかり、性能が悪くなる。

ＴＲＥＡＴ［2］では中間結果のうち、複数のデータを組合せた条件のテストの中間結果を保存しないが、やはり最終結果として条件を満たすデータ組の全てを計算し保存するため、データ変更時の削除のオーバヘッドが大きい。

本稿の遅延照合アルゴリズムでは、ルールの条件を真偽判定するために必要最小限の限定条件だけについてデータの検索を行なうため、条件照合回数が最適である。また、データベースへの操作があった時に、限定条件の種類とデータベースへの操作の組み合わせで、その最終結果の有効性をチェックでき、最終結果のデータ組も高々１組しか持たないので削除のオーバヘッドも少ない。例えば条件を満たすデータが存在する状態でデータが削除された時、条件を満たすデータが存在しない状態でデータが追加された時に論理値を「不明」とすれば良い。

本方式では必要最小限のルールの評価しか行なわないため、結果保存が全く役に立たない問題でも、結果を保存しない方式と同等の性能を持つ。

その他、必要最小限のデータの評価を行なう要求駆動型マッチングアルゴリズム［5］が提案されている。

本稿のアルゴリズムではルールの条件を基本条件と限定条件に分けている。データベース検索を必要とせず、変数の形で直接参照できる基本条件については結果を保存せず毎回評価を行なう。実用上、基本条件の大部分を占める変数の比較等では、データ変更による保存情報の更新のオーバヘッドの方が大きいので、条件を直接再評価する方が効率が良いと思われる。また、ルールの条件部の論理式は副作用を持たないという前提で、演算の第１項の値によって演算結果が判る場合に第２項の評価を行なわない、所謂「短絡評価」を行なうので条件照合回数がさらに少ない。

6．むすび

ルール型計算システムの実行方式として、データ集合の検索を必要とする限定条件についてのみ結果を保存し、ルールの条件を３値評価することを特徴とする、計算効率の良い遅延照合アルゴリズムを考案した。

［参考文献］

［1］ Forgy, C.L.: Rete: A Fast Algorithm for the Many Pattern／Many Object Pattern Match Problem: Artificial Intelligence, Vol.19, No.1, pp.17-37 (1982)

［2］ Miranker, D.P.: TREAT: A Better Match Algorithm for AI Production Systems: Proc. of AAAI-87, pp.42-47 (1987)

［3］ Nuutila, E., et al.: XC - A Language for Embedded Rule Based System: ACM SIGPLAN Notices, Vol.22, No.9, pp.23-32 (1987)

［4］ 石田、桑原：プロダクションシステムの高速化技術：情報処理, Vol.29, No.5, pp.467-477 (1988)

［5］ 藤田 他： 前向きプロダクションシステムのための要求駆動型マッチングアルゴリズム：情処研報, Vol.89, No.30 (89-ARC-75) (1989)