

画像処理エキスパートシステム IPEX のための知識エディタ†

齋藤 由香梨^{††} 小島 紀子^{†††} 松尾 昭彦^{††}
 斐 東 善^{††} 秋元 晴雄^{††}

エキスパートシステムの開発において、最も重要な作業の一つは知識ベースの構築である。知識ベースの質がシステムの性能を大きく左右するからである。われわれは、画像処理エキスパートシステム IPEX (Image Processing EXpert system) の知識ベースを構築するための知識エディタを開発した。本エディタでは、専門家自身による知識ベースの構築作業を容易にするために四つの支援機能を実現している。第一に、複雑な形式の知識を扱いやすくするために図形を用いた知識の作成・編集機能を実現した。第二には、知識ベースに誤りが混入するのを防ぐために入力された知識の論理的な誤りを検出する機能を備えた。第三に、専門家に新しい知識を考える手掛かりを与えるために既存の知識から未入力知識を生成する機能を実現した。最後に、知識ベースの見通しを良くするために知識の内容を整理する機能を実現した。これらの機能を活用することにより、専門家自身による効率的な知識ベースの構築が可能となった。

1. はじめに

画像処理の研究成果として、膨大な数のアルゴリズムが開発され、蓄積されてきた。一方、画像処理の応用分野が広がるとともに、画像処理技術へのニーズが高まっている。しかし、画像処理の非専門家にはこれまで開発されてきた技術を使いこなすのは容易ではない。そこで、画像処理を知らない利用者でもシステムと対話することにより、画像処理を実行できるエキスパートシステムの開発が進められている。

エキスパートシステムの開発にあたっては、システムの知識ベースを構築することが重要な作業となっている。知識ベースの質がシステムの性能を大きく左右するからである。しかし、この作業は大変な労力を必要とする。それは、システムに取り込むべき専門家の知識が漠然としていることが原因である。すなわち、専門家は自分が問題解決に使っている知識を系統的に把握しているわけではなく、実際に問題が与えられたときに改めて思い出すからである。このため、知識ベースの作成は場当たりのものとなり、知識の内容が不完全なものになってしまう。

知識ベースの構築を支援するツールとして知識エ

ディタがある。これは、専門家の知識を知識ベースへ変換する過程を支援する。専門家やナレッジエンジニアは、知識エディタを利用して対象分野の知識やノウハウから知識ベースを構築する。知識エディタでは、単に知識をシステムの内部表現に変換する機能だけでなく、知識を整理し、誤りを検出する機能や専門家に働きかけて知識を引き出す機能が必要である^{1)~4)}。

そこで、われわれは知識ベースの構築を容易にする機能を備えた知識エディタを開発した^{5)~7)}。これは、画像処理エキスパートシステム IPEX (Image Processing EXpert system)⁸⁾ の知識ベースを構築するためのツールである。本エディタでは、以下の機能によって専門家による知識ベースの構築を支援している。

(1) 図形を用いた作成・編集機能

複雑な形式の知識を扱いやすくするために、図形を用いて知識の作成・編集を行う。

(2) 知識の誤り検出機能

知識ベースに誤りが混入するのを防ぐために、入力された知識の論理的な誤りを検出する。

(3) 未入力知識の生成機能

専門家に新しい知識を考える手掛かりを与えるために、既存の知識から未入力知識を生成し提示する。

(4) 知識ベースの整理機能

知識ベースの見通しを良くするために、知識の内容を整理する。

本論文では、まず IPEX の知識ベースについて説明し、次に知識エディタの概要と主な機能の内容につ

† A Knowledge Base Editor for the Image Processing Expert System IPEX by YUKARI SAITOH (Software Development Laboratory, Information Processing Division, Fujitsu Laboratories Ltd.), NORIKO KOJIMA (Human Interface Laboratory, Fujitsu Laboratories Ltd.), AKIHIKO MATSUO, TOHZEN HAI and HARUO AKIMOTO (Software Development Laboratory, Information Processing Division, Fujitsu Laboratories Ltd.).

†† (株)富士通研究所情報処理研究部門ソフトウェア開発部

††† (株)富士通研究所ヒューマンインタフェース研究部

いて述べ、最後にこれらの機能を実際の知識ベースの構築に適用した結果について述べる。

2. IPEX の知識ベース

IPEX は、富士通画像情報システム「FIVIS/VPS」上のコンサルテーションシステムである。このシステムでは、画像処理に不慣れな利用者でもメニュー形式の質問に答えるだけで、対話的に画像処理を実行することができる。

この IPEX では、画像処理の手順に関する知識と画像の属性に関する知識を利用して推論を進めている。

画像処理の手順に関する知識は、プロダクションルールとして記述される。ルールは、それがどんな処理に使われているかによって、処理目的と呼ばれるグループに分類されている。また、ルールには条件部として入力画像の性質（属性とその属性値）が、実行部には処理目的達成のための手順の並びが記述されている。実行部に記述された各手順は、それ自体がまた別の処理目的とみなされ、さらに詳細な（具体的な）手順の並びに置き換えられる。この過程はそれ以上置き換えられない手順の並びとなるまで繰り返される。

たとえば、図 1(a)は処理目的「領域分割」を持つルールを示している。ここでは、入力画像の属性「画像の種類」の値が「濃淡画像」か「二値画像」かによ

って、適用されるルールが「領域分割1」か「領域分割2」かに決定される。仮に「領域分割1」が適用される場合、まず「領域抽出」を行い、続いて「連結成分のラベル付け」を行うことを示している。さらに「領域抽出」を行うために、処理目的「領域抽出」を持つルール(図 1(b))が調べられる。こうして処理目的「領域分割」は、より具体的な処理目的の並びに置き換えられていく。

画像の属性に関する知識は、画像の状態を表す属性とその値で表現される。この画像の状態は、ルールの条件部に記述され、利用者はシステムからの画像状態の問合せに応じて、属性値の中から現在の画像の状態を選び、入力する。

たとえば、画像の明るさは「明るさの程度」という属性で表現され、そのとりうる値として「大きい」、「小さい」などの属性値が定義される。

3. 知識エディタ

われわれの開発した知識エディタは、画像処理の専門家を対象としている。専門家は、このエディタを利用して知識ベースの構築を行う。以下では、本エディタの主な機能について述べる。

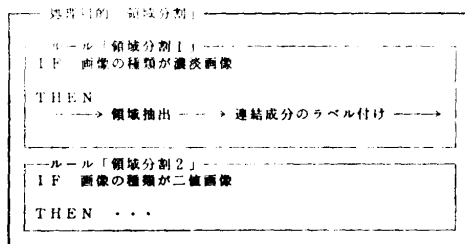
3.1 図形を用いた作成・編集機能

前章で述べたルールの実行部には処理目的の手順が記述される。この知識は、従来、専門家が図1のような処理の手順を描き、それを人手で内部表現に変換するという方法で作成していた。そのため、手間がかかり、処理の手順を内部形式に変換する際に誤りが入りやすく、内部形式に変換した後では知識の内容がわかりにくくなるという問題があった。

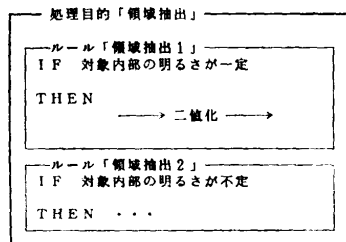
そこで、本エディタでは専門家とのインタフェースとして図形が扱えるようにした。つまり、図形表現と内部表現との双方向変換機能を実現することによって、専門家が紙に描くのと同様に処理の流れ図を作成するだけで、知識の作成や編集ができるようになった。

これによって、労力が削減でき、従来のように変換の際に誤りが混入することがなくなる。また、専門家は作成した知識の内容を、図形として直観的に把握することができる。

本エディタでは、処理目的と画像とを箱と線とで表した図形として扱う。例として、濃淡画像の方向別微分値を検出する際の知識の図形表現を図2に、その内部表現を図3に示す。図形表現中の箱は処理目的を、



(a) 「領域分割」のルール
(a) Rules for segmentation



(b) 「領域抽出」のルール
(b) Rules for region extraction

図 1 ルールの例
Fig. 1 Examples of rules.

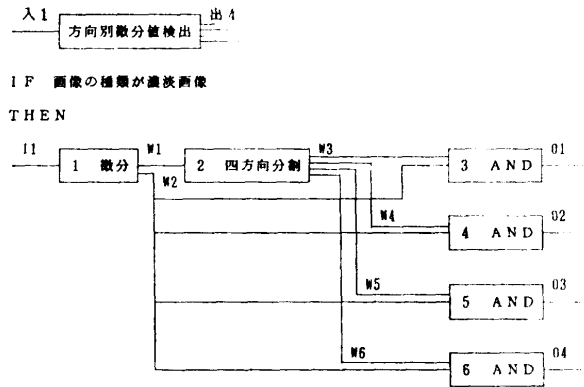


図2 ルールの図形表現の例

Fig. 2 An example of rule expression in a diagram.

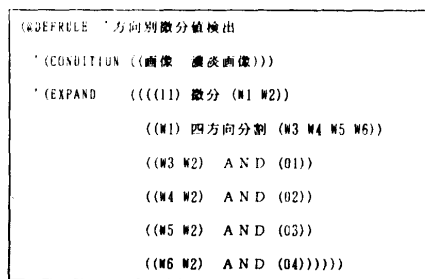


図3 ルールの内部表現の例

Fig. 3 An example of rule expression in Lisp.

箱の左線はその処理目的に対する入力画像を、右線は出力画像を表している。専門家は、箱を作る、箱を消す、線をつなぐ、線を切るの四つの操作によって図形の作成・編集を行い、知識ベースの作成を行う。

3.2 知識の誤り検出機能

知識ベースに誤りがあると、推論結果にも誤りが生ずる。しかし、初めから矛盾のない完全な知識ベースを作成することは容易ではない。そのため、いったん作成した知識ベースの内容を見直したり、テストして推論結果を確認するというデバッグ作業が必要となっている。しかし、知識ベースが大きくなってくると、誤りを人手で検出するのは手間がかかり、見落としが多くなる。そこで、知識エディタには入力された知識の誤りを検出する機能が望まれている。

本エディタでは利用者がルールを入力した段階でルールの論理的な誤り^{9),10)}を検出する機能を備えている。ここでは、ルールの論理的な誤りとして、条件部に注目した誤り検出と実行部に注目した誤り検出の二つを行っている。前者では、同じ処理目的を持つルール集合ごとに論理的な誤りを検出する。後者では、実際に推論部を起動したときに推論が途切れたり、循環する可能性がないかについて調査する。

ここでやっているチェックは、すべて利用者の注意を促すためのものであり、チェックの結果、知識を修正するか否かは利用者の判断に委ねる。

3.2.1 条件部に注目した誤り検出

条件部に注目した誤り検出では、同じ処理目的を持つルール集合中の条件部を取り出し、属性の知識を利用して、論理的な誤りが生じていないかをチェックしている。ここでは、冗長（全く同じルールが存在する）、競合（条件部が同じで実行部が異なるルールが存在する）、包含（あるルールの条件部が他のルールの条件部を包含している）、部分的に一致（あるルールの条件部が他のルールの条件部と一部分重なっている）、不完全（ルールが不足している）、属性間の関係（条件部に記述されている属性に上下関係がある）の各項目についてルールをチェックしている。

以下では、不完全と属性間の関係のチェックについて述べる。

(1) 不完全

専門家がルールを作っていく場合、ある処理目的を持つルールをすべて作成してから別の処理目的のルールの作成に移るとは限らない。そのため、専門家はいくつものルールの作成を忘れてしまうことがある。ルールが不足していると、推論実行中に適用できるルールがなくなり、推論が停止する可能性がある。本エディタではこのような状況を防ぐために、ルールの不足を検出し利用者にもそのルールの作成を促している。

たとえば、同じ処理目的を持つルール1、ルール2、ルール3があったとする。ただし、属性の知識として属性「画像の種類」は値として「濃淡画像」か「二値画像」を属性「線状の雑音」は「目立つ」か「目立たない」をとると定義されていたとする。

ルール1: IF 画像の種類=濃淡画像 かつ
線状の雑音=目立つ,

THEN...

ルール2: IF 画像の種類=濃淡画像 かつ
線状の雑音=目立たない,

THEN...

ルール3: IF 画像の種類=二値画像 かつ
線状の雑音=目立つ,

THEN...

エディタは、「画像の種類」が「二値画像」でありかつ「線状の雑音」が「目立たない」の場合に適用できるルールが存在しないので、そのような条件を持つルー

ルが不足していることを警告する。これによって、利用者は不足しているルールを知ることができる。

(2) 属性間の関係

これは、同じ処理目的を持つルールの集合の条件部の記述を調べ、属性間に上下関係があるか、またはルールの記述の見直しが必要かどうかについて調べる。これは、ルールの条件部の組合せがすべて尽くされているルール集合についてチェックする。

たとえば、同じ処理目的を持つルール4, ルール5, ルール6があったとする。ただし、属性の知識として属性「明るさの変化」は「大きい」か「小さい」かをとると定義されていたとする。

```
ルール4 : IF 画像の種類=濃淡画像 かつ
           明るさの変化=大きい,
           THEN...
```

```
ルール5 : IF 画像の種類=濃淡画像 かつ
           明るさの変化=小さい,
           THEN...
```

```
ルール6 : IF 画像の種類=二値画像,
           THEN...
```

このとき、これらのルールの集合は属性「画像の種類」と属性「明るさの変化」の値の組合せすべてについて条件部の成り立つルールが必ず一つ存在する。しかし、ルールの条件部を調べると、「画像の種類」の値が「濃淡画像」のときだけ「明るさの変化」について記述されている。もし、「明るさの変化」が「濃淡画像」のときだけに意味のある属性ならば、「明るさの変化」は「濃淡画像」の下位の属性であるといえる。そうでなければ、「画像の種類」が「二値画像」のときに「明るさの変化」が「大きい」か「小さい」かによって実行部の内容が異なるかどうかについて考える必要がある。

そこで、本エディタでは以上の事実を検出して、利用者に“属性「明るさの変化」は、属性「画像の種類」の値が「濃淡画像」のときだけ意味を持つか？”と尋ねる。もし、利用者がその関係の成立を認めるならば、エディタはその属性間の関係を記録する。そのような関係がないならば、“ルール6の記述は考え直す必要がないか？”というアドバイスを利用者に与える。

このチェックによって検出された属性間の関係は、属性の知識に追加項目として記録される。その後の属性間の関係のチェックでは同じ関係が検出されたらそれが既知かどうかを調べ、既知ならば利用者に関係の確認はしない。

このようにして、属性間の上下関係を引き出したリ、ルールの記述の見直しを促すことができる。

3.2.2 実行部に注目した誤り検出

実行部に注目した誤り検出では、推論部を起動した際にルールの連鎖に論理的な誤りが生じないかについてチェックしている。ここでは、論理的な誤りとしてルールの呼び出しが途切れたり、循環する可能性がないかについてチェックしている。

IPEX では、ルールの呼び出しは展開ルールの実行部に記述される処理目的を持つルールをさらに呼び出すことで行われる。エディタは、その実行部に記述される処理目的を順に辿ることで、ルールの呼び出しの連鎖を調べる。

(1) 途切れ

推論の途中で次に実行されるルールが存在しないとき、推論が途切れる。これは、ルールの実行部の記述に誤りがあるか、呼び出されるルールがまだ作成されていない場合に起こる。この誤りは、推論の停止を起こす可能性がある。

(2) 循環

推論の途中でルールの呼び出しが再帰的に起こると、循環を起こす可能性がある。これは、ルールの実行部の記述に誤りがあり、画像の属性が変化しない場合に起こる。循環が生ずると、推論が停止しなくなる可能性がある。

3.3 未入力知識の生成機能

専門家は自分の知識を常に意識しているわけではない。専門家が知識エディタを使って知識ベースを構築する場合、知識エディタが従来のエディタのように常に受け身で利用者の入力を待っているだけでは、専門家が意識していない知識を引き出すことはできない。エディタの方から専門家に知識を引き出す刺激となるような情報を与えることが必要である。

専門家にどのような情報を与えるかを検討するために、エディタに蓄えられた知識の内容を調査した。その結果、ルールの条件部に記述される属性の値と実行部に記述される処理目的とが対応しているものがみられた。たとえば、条件部に「処理対象の画像の種類が濃淡画像である」と記述されているルールでは、実行部に「二値化」という処理目的が高い頻度で現れることがわかった。

そこで、われわれはそれを利用してエディタがルールの生成を行う機能を検討した。それは、利用者があるルールを入力すると、エディタがそのルールと既存

の知識ベースから抽出した対応関係とを組み合わせ、新しいルールを生成する機能である。利用者は、エディタが生成した知識の妥当性を考えることにより、新しい知識を作成する手掛かりとすることができる。

3.3.1 生成方法

未入力ルールの生成は、以下の手順で行う。

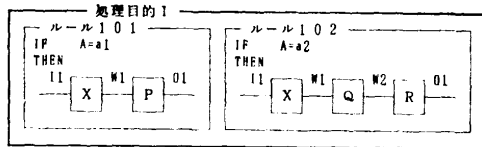
① 対応関係の抽出

エディタは、同じ処理目的を持つルールの中から条件部に記述される属性値と実行部に記述される処理目的との対応関係を抽出する。

たとえば、図4(a)に示すような処理目的Iを持つルールの集合があったとする。すべてのルール中に共通の処理目的Xが存在するか、または全く存在しない場合に、属性値a1と処理目的P、属性値a2と処理目的QとRのように、属性値と処理目的が一対一に対応するとき、それぞれの属性値と処理目的の間には対応関係があるという。エディタは、これらの対応関係(図4(b))を処理目的ごとに抽出する。

② 新しいルールの生成

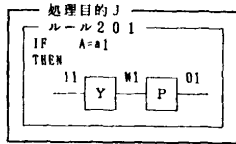
①で抽出された対応関係を利用して、エディタは新



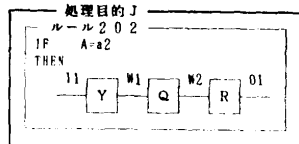
(a) 既存のルール
(a) Existing rules

属性	属性値	対応する処理
属性 A	a 1	X → P
	a 2	X → Q → R

(b) 既存のルールから得られた対応関係
(b) Extracted relations from existing rules



(c) 次に入力されたルール
(c) Next input rule

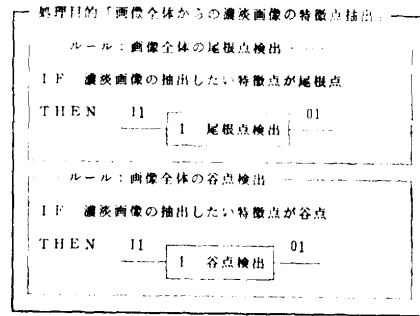


(d) 生成されたルール
(d) Generated rule

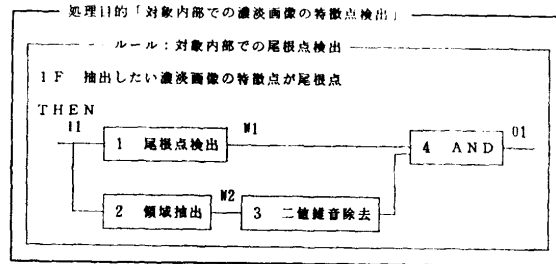
図4 未入力のルールの生成
Fig. 4 Generating new rules.

しいルールを生成する。

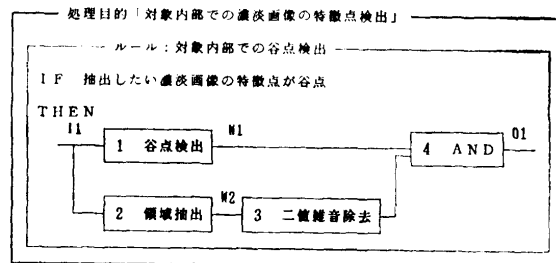
図4の例では、図4(b)の対応関係を抽出した後、利用者が図4(c)に示す処理目的Jを持つ全く新しいルール201を入力した。このルールの条件部の属性Aの値はa1であり、実行部に処理目的Pが記述されている。これは、処理目的Iから得た対応関係と同じである。そこで、エディタは図4(b)の情報を利用して、図4(d)に示すようなルール202を生成する。ルール202は、ルール201の条件部の属性Aの値をa2に変え、ルール201の実行部の処理目的Pに相当する部分を処理目的Qと処理目的Rとで置き換えることによって生成する。利用者は、エディタの生成したルールが妥当であればそれを新しいルールとしてそのまま登録することができる。また、違っていればその内容を修正したり、棄却することもできる。



(a) 既存のルールの例
(a) An example of existing rules



(b) 次に入力されたルール
(b) Next input rule



(c) 生成されたルール
(c) Generated rule

図5 未入力のルールの生成の例
Fig. 5 An example of generating new rules.

図5に「画像全体からの濃淡画像の特徴点検出」という処理目的を持つルールから得た情報から「対象内部での濃淡画像の特徴点検出」という処理目的を持つルールを生成した例を示す。

3.4 知識の整理機能

前節で述べた未入力知識の生成機能は、ある処理目的を持つルールから得られた対応関係を他の処理目的を持つルールに適用することで実現されている。すなわち、そこで適用された対応関係は、重複して知識ベース内に存在することになる。また、この重複した対応関係は、異なる処理目的を持つルール間に存在する陽に記述されていない潜在的な知識と考えることができる。

ルールを入力していく過程では、生成機能を活用して似ている知識を入力していく方が効率的である。しかし、知識ベースを作成し終わった段階では、知識ベースに重複した対応関係が多く存在すると、知識ベースの見通しが悪くなる。

そこで、本エディタでは知識ベースを整理する機能を実現した。知識ベースの整理は、重複する部分を新たな知識として登録し、元の知識を書き換えることで行う。

3.4.1 整理の方法

① 既存の知識ベース中の対応関係の抽出

3.3.1 項で述べた方法と同様に、今までのルールから属性値と処理目的との対応関係を抽出する。ただし、この場合はどの処理目的を持つルールからその対応関係の情報が得られたかをその対応関係の根拠としてエディタが記録する。

たとえば、処理目的Kと処理目的Lを持つルール(図6(a))が存在するとき、これらから図6(b)に示すような属性値と処理目的との対応関係が得られる。

② 対応関係のルール化

①で抽出された対応関係について、根拠となる処理目的が複数記述されている対応関係からルールを生成する。これは、対応関係のデータに記述されている属性値を条件部とし、それに対応する処理目的を実行部とするルールを作ることによって行う。

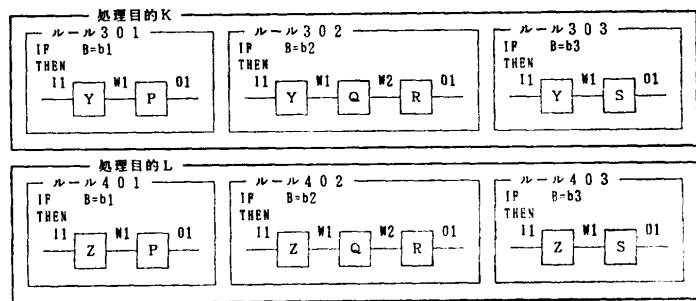
たとえば、図6の例では、図6(b)に示

す属性Bと属性値 b_1, b_2, b_3 をそれぞれ条件部とし、それに対応する処理目的 P, Q と R, S をそれぞれ実行部とするルールを新たに生成する。このルールの内容を利用者に示して、ルールの名前と処理目的Mの名前を付けてもらう。

③ 元のルールの内容の変更

②の新しいルールの生成に伴い、①で根拠となる処理目的に記録された処理目的を持つルールの削除や修正を行う。

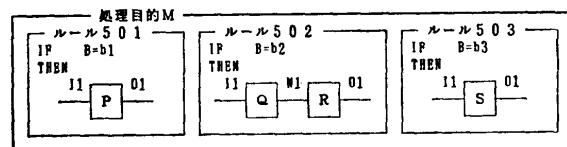
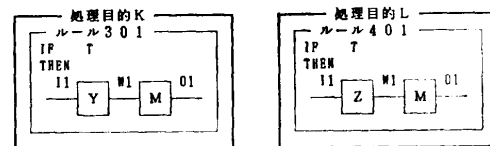
図6の例では、根拠となる処理目的である処理目的Kと処理目的Lを持つすべてのルールの条件部から、属性Bに関する条件を削除する。この場合、ルールの条件がなくなるのでT(常に成り立つ)となる。次に、条件部が全く同じになったルールは、一つを残して他は削除する。残ったルールの実行部について、新



(a) 元のルール
(a) Original rules

属性	属性値	対応する処理	根拠となる処理目的
属性B	b_1	P	処理目的K 処理目的L
	b_2	Q, R	
	b_3	S	

(b) 既存のルールから得られた対応関係
(b) Extracted relations from existing rules



(c) 整理後のルール
(c) Rules after refining

図6 ルールの整理

Fig. 6 Refining rules.

しく生成したルールの処理目的に相当する P, Q と R, S の部分を新しい処理目的 M で書き換える。こうしてルールの整理を終了する (図 6 (c) 参照)。

4. 各機能の効果

本エディタの図形による作成・編集機能, 知識の誤り検出機能, 未入力知識の生成機能, 知識の整理機能を実際の知識の作成・編集に用いた。

図形を用いた作成・編集機能によって, 知識の作成が容易になった。専門家は, 紙の上に描いた流れ図と同様の図をディスプレイに描くだけで知識を作成していくことができる。また, 図形上で知識を編集すればよいので, 内部表現を知らない専門家にも容易に知識の作成ができた。図形を用いることで, 内部表現では理解しづらい知識の内容も直観的に理解できるようになった。

知識の誤り検出機能は, ルールを入力した時点で誤りを検出できるので, 知識のデバッグを早い段階で行うことができる。また, ルールを一括してチェックできるので, 誤りの見落としがなくなった。チェック項目では, 不完全のチェックと属性間の関係のチェックの二つが特に有効であった。不完全のチェックによって, 不足しているルールを検出することができ, ルールの入力忘れを防ぐことができた。属性間の関係のチェックは, 利用者が無意識に使った属性間の関係を引き出すことができ, またルールの記述の見直しに役立った。

未入力知識の生成機能をルールが 350 個程度作成された段階で試した結果, 58 個のルールが生成できた。エディタが生成した 58 個のルールは, 利用者によってすべて妥当であると確認できた。

この機能で重要なのは, エディタの方から専門家に新しい知識を示すことである。専門家は自分が意識している知識の入力は容易であるので, エディタはそれを受け入れるだけでよい。しかし, 専門家が新たな知識を思いつかなくなった場合, エディタが受身のままでは知識ベースの構築が進まなくなる。そのとき, エディタは専門家に知的な刺激を与え, 知識を引き出すような働きかけを行わなければならない。本エディタの知識の生成機能は単純で適用範囲もまだ狭い方法で行っているが, 専門家に新たな知識を考えさせるのに役立つ重要な機能である。

生成機能によって妥当なルールが生成できる知識は, 知識の整理機能を利用して整理することができ

る。この機能によって, 知識ベースに含まれる陽に記述されていない潜在的な知識を顕在化し, 知識ベースの見通しを良くすることができた。

5. おわりに

本論文では, 画像処理エキスパートシステム IPEX の知識ベースを構築するための知識エディタの主な四つの機能とその効果について述べた。これらの機能を利用することで, 画像処理の専門家による知識ベースの構築を効率良く行うことができるようになった。

今後は, 画像処理の意味内容まで考慮した誤り検出や生成機能について検討することにより, より知的で使い良いエディタとして発展させていくことが必要である。

謝辞 日頃ご指導いただく棚橋部門長, 三宅部長, 手塚課長, 有益な助言をいただいた吉田部長に感謝します。また, 熱心に討論をしてくださった鳥生・岩瀬両氏に感謝します。

参考文献

- 1) Waterman, D. A.: *A Guide to Expert Systems*, p. 419, Addison-Wesley, Massachusetts (1986).
- 2) Davis, R.: *Interactive Transfer of Expertise: Acquisition of New Inference Rules*, *Artif. Intell.*, Vol. 12, pp. 121-157 (1979).
- 3) Boose, J. H.: *Personal Construct Theory and the Transfer of Human Expertise*, *Proc. of AAAI-84*, pp. 27-33 (1984).
- 4) Kahn, G., Nowlan, S. and McDermott, J.: *MORE: An Intelligent Knowledge Acquisition Tool*, *Proc. of 9th IJCAI*, pp. 581-584 (1985).
- 5) 小島, 斎藤, 豊, 秋元: 画像処理エキスパートシステム IPEX のルールエディタの開発, 第 34 回情報処理学会全国大会論文集, pp. 1675-1676 (1987).
- 6) 斎藤, 小島, 松尾, 豊, 秋元: 画像処理エキスパートシステムールールエディタ, 第 35 回情報処理学会全国大会論文集, pp. 2095-2096 (1987).
- 7) 松尾, 斎藤, 豊, 秋元: 画像処理エキスパートシステム用知識エディタ, 情報処理学会コンピュータビジョン研究会資料, 52-5 (1988).
- 8) 鳥生, 岩瀬, 後藤, 吉田: コンサルテーション型画像処理用エキスパートシステム, 情報処理学会論文誌, Vol. 29, No. 2, pp. 150-159 (1988).
- 9) Suwa, M., Scott, A. C. and Shortliffe, E. H.: *An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System*, *AI Magazine*, Vol. 3, No. 4, pp. 16-21 (1982).

- 10) Nguyen, T. A., Perkins, W. A., Laffey, T. J. and Pecora, D.: Checking an Expert Systems Knowledge Base for Consistency and Completeness, *Proc. of 9th IJCAI*, pp. 375-378 (1985).

(昭和 63 年 6 月 29 日受付)

(平成 元年 1 月 17 日採録)



斎藤由香梨 (正会員)

昭和 37 年生。昭和 60 年慶應義塾大学文学部文学科卒業。同年富士通(株)入社。現在、富士通研究所ソフトウェア開発部勤務。エキスパートシステムの開発に従事。



小島 紀子 (正会員)

昭和 37 年生。昭和 60 年東海大学理学部情報数理学科卒業。同年富士通(株)入社。現在、富士通研究所ヒューマンインタフェース研究部にて、コンピュータグラフィックスに

関する研究に従事。



松尾 昭彦 (正会員)

昭和 39 年生。昭和 62 年東京理科大学理学部物理学科卒業。同年富士通(株)入社。現在、富士通研究所ソフトウェア開発部においてエキスパートシステムおよび知識エディタ

の開発に従事。



藪 東善

昭和 29 年生。昭和 53 年早稲田大学理工学部電気工学科卒業。昭和 55 年名古屋大学工学部情報工学科修士課程修了。同年(株)富士通研究所入社。パターン認識の研究を経て、現在エキスパートシステムの開発に従事。電子情報通信学会会員。



秋元 晴雄 (正会員)

1948 年生。1972 年東京工業大学工学部電気工学科卒業。1974 年同大学院修士課程修了。同年(株)富士通研究所入社。以来、手書文字認識、記号処理マシンの研究を経て、エキスパートシステム、図形を用いたユーザインタフェースの研究に従事。現在、同研究所ソフトウェア開発部主任研究員。電子情報通信学会、人工知能学会、日本認知科学会、IEEE、AAAI 各会員。