



## エキスパート・システムの土木工学への応用について

メタデータ	言語: jpn 出版者: 室蘭工業大学 公開日: 2014-03-04 キーワード (Ja): キーワード (En): 作成者: 菅田, 紀之, 尾崎, 昶 メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/10258/732">http://hdl.handle.net/10258/732</a>

# エキスパート・システムの土木工学への応用について

その他（別言語等）のタイトル	On the Application of Expert System to Civil Engineering
著者	菅田 紀之, 尾崎 ?
雑誌名	室蘭工業大学研究報告. 理工編
巻	37
ページ	199-209
発行年	1987-11-10
URL	<a href="http://hdl.handle.net/10258/732">http://hdl.handle.net/10258/732</a>

# エキスパート・システムの土木工学への応用について

菅田紀之・尾崎 諷

## On the Application of Expert System to Civil Engineering

Noriyuki SUGATA and Shinobu OZAKI

### Abstract

Expert system is generally defined as the intellectual computer program that can solve the complex problems like a human expert. To solve the problem the system uses the knowledge obtained from the several experienced experts. To get the knowledge is one of the most important problem to create the expert system.

The system is generally developed by LISP, PROLOG or an expert shell such as the production system language OPS5. It has come into wide use rapidly in all engineering problems, mechanical engineering, civil engineering, and so on.

This paper explains the OPS5 and a computer program to choose the best foundation type as a bridge-substructure.

This example shows that the expert system can widely apply to the several design problems in the civil engineering world.

### 1. はじめに

エキスパート・システムとは、一般に困難で専門的な知識を必要とする問題を、その問題領域の専門家から獲得された知識を利用して推論を行い、専門家と同程度の能力で解決することを目標とする知的プログラムのことをいう。つまり、専門家でなければ解決困難な問題分野が、エキスパート・システムの対象である。また、エキスパート・システムの基盤は知識工学であり、知識工学の基礎は人工知能と位置づけられるので、エキスパート・システムは人工知能の具体形の一つといえる。

エキスパート・システムの研究は、1960年の中頃にスタンフォード大学で開発された有機化合物の構造式を推定する DENDRAL<sup>1)</sup> に始まる。また、同じ頃 MIT で数式処理システムの MACSYMA<sup>2)</sup> が開発されている。1970年代には、スタンフォード大学で血液中のバクテリアの診断をするシステムの MYCIN<sup>3)</sup> が開発され、1980年代になると、大学や研究所で始まった研究開発が産業界へ広まっていき、エキスパート・システム構築ツール (エキスパート・シェル) といわれるエキスパート・システムを開発するための言語が商用化されるまでになってきている。

エキスパート・システムに使用される言語には、リスト処理型言語 LISP, 論理型言語 PRO-

LOG およびエキスパート・シェルの ART, KEE, ESHELL, Knowledge Craft, OPS5, BRAIN 等がある。LISP や PROLOG でエキスパート・システムを構築する場合には、推論プログラムやその他の動作のためのプログラムを作り、知識をデータ・ベース化する必要があるが、エキスパート・シェルの使うと知識をデータ・ベース化するだけで、エキスパート・システムとして機能するという利点があり、開発効率が大幅に向上する。エキスパート・シェルでの知識の表現法にはプロダクション・システム、フレーム、意味ネットワーク、黒板モデル等があるが、ほとんどのエキスパート・シェルではプロダクション・システムとフレームを採用している。

土木工学への応用としては構造物の設計、今後重要な課題となってくる構造物の維持管理・補修および寿命予測等が考えられる。また、日本における応用研究は始まったばかりで、斜面宅地開発での斜面崩壊の研究<sup>4)</sup>、橋梁下部構造の形式選定に応用した研究<sup>5)</sup>、FEM 解析のための構造モデル化のための研究<sup>6)</sup>、岩盤力学へ応用した研究<sup>7)</sup>、RC 床版の耐用性評価のための研究<sup>8)</sup>、鉄道橋脚の健全度の診断のための研究<sup>9)</sup>、ダムゲートの寿命予測を行う研究<sup>10),11)</sup>、道路橋 RC 床版のの損傷・補修に関する研究<sup>12),13)</sup>等がある。

本論文では、本学の共同利用施設であるミニコンピュータ VAX11/750 上で実行可能なエキスパート・シェル OPS5 について述べ、その土木工学への応用として橋梁下部構造の基礎形式選定のためのエキスパート・システム開発について述べる。

## 2. OPS5

OPS5 は「プロダクション・システム」をベースとしたエキスパート・システム構築ツールであり、プロダクション・ルールという「if<sub>1</sub> then<sub>2</sub>」型の条件文の集まりで構成されていて、もし<sub>1</sub>という条件が成りたてば<sub>2</sub>という動作を実行するプログラム言語である。

OPS5 は図-1 にしめすようにプロダクション・メモリー (Production Memory)、ワーキング・メモリー

(Working Memory)、推論部 (Inference Engine) の3要素から構成されている。プロダクション・メモリーは知識と呼ばれる「if~then~」型のルール・ベースであり、ワーキング・メモリーは対象となっている事柄の状態の集合である。そして、推論部はワーキング・メモリーの状態によってプロダクション・メモリー中の実行可能なルールを選択し実行する部分で、人間の頭脳にあたる。推論部の一連の動作のことを「認知-動作サイクル」(Recognize-Act Cycle) という。

### 2.1 ワーキング・メモリー

ワーキング・メモリーは対象の状態の集まりであり、図-2のようにワーキング・メモリー要素とタイム・ダグ (Time Tag) より構成されている。ワーキング・メモリー要素のフォーマット

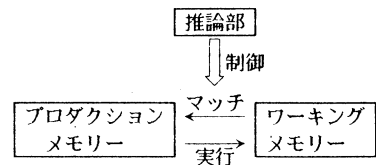


図-1 OPS5の構造

トは図-3に示すようにクラス名といくつかの属性名と値の組からなっている。クラス名には対象となる事項を、属性名と値にはその状態を与える。属性名は「<」をもって表す。

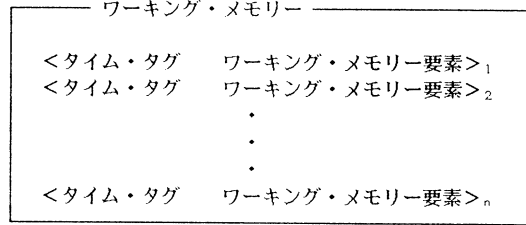


図-2 ワーキング・メモリー

すべてのワーキング・メモリー要素に対しタイム・タグという整数値が与えられている。タイム・タグとはワーキング・メモリーに時間の概念を持たせるもので、いつその要素が作られたか、あるいは修正されたかを示し、大きなタイム・タグを持つ要素ほど最近作られたか修正されたものである。これにより、推論部はワーキング・メモリー要素の新しさを認識できることになる。

(クラス名 「属性1 値1 「属性2 値2 … 「属性n 値n)

図-3 ワーキング・メモリー要素

2.2 プロダクション・メモリー

プロダクション・メモリーは「if~then~」型のプロダクション(ルール)の集合である。図-4はプロダクション・ルールのフォーマットである。条件部(Left hand side)はいくつかの条件要素(Condition Element)から構成され、ワーキング・メモリーと照合するためにワーキング・メモリー要素とほとんど同じフォーマットになっている。動作部(Right Hand Side)は条件部が満足されると実行される部分である。

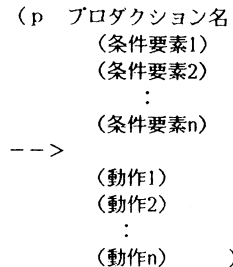


図-4 プロダクション・ルール

2.3 認識-動作サイクル

認識-動作サイクルは図-5のようなステップで実行される。

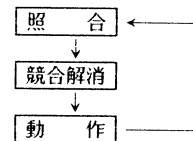


図-5 認識-動作サイクル

2.3.1 照合 (Match)

ワーキング・メモリー要素とすべてのプロダクションの条件要素を一つずつ照合し、条件部が満足されると実行可能となり、インスタンス化(Instantiation)と呼ばれるデータが作られ競合集合(Conflict Set)に入る。インスタンス化は図-6のように実行可能なプロダクション名とそのプロダクションの条件部

プロダクション名 タグ1 タグ2 … タグn

図-6 インスタンス化

を満足したワーキング・メモリー要素のタイム・タグからなっている。また、競合集合とはインスタンス化の集合のことである。

### 2.3.2 競合解消 (Conflict Resolution)

競合集合の中から指定された戦略 (Strategy) に基づいて、一つのインスタンスエーションを選び出す。OPS5 には「LEX」と「MEA」という2つの戦略がある。

#### 1) LEX

1. 一度選択されたインスタンスエーションを競合集合から除く。
2. より新しいワーキング・メモリー要素のタイム・タグを持つインスタンスエーションを選ぶ。インスタンスエーションの比較のしかたは次のように行う。
  - a) 最新のタイム・タグを比較し新しいものを選ぶ。
  - b) 同じ場合には次に新しいタイム・タグを比較し新しいものを選ぶ。
  - c) 一つのインスタンスエーションが選ばれるまで、あるいは他方の比較するタイム・タグがなくなるまで次々にタイム・タグを比較する。しかし、同時にタイム・タグがなくなった場合には選択しない。
3. 以上で一つのインスタンスエーションが選ばれない場合は、より条件の厳しいインスタンスエーションを選ぶ。つまり、プロダクションの条件部で行われている属性値のチェックの回数を比較し回数が多いものを選ぶ。
4. 以上でも選ばれない場合は、任意のインスタンスエーションが選ばれる。

#### 2) MEA

1. 一度選択されたインスタンスエーションを競合集合から除く。
2. インスタンスエーションの先頭のタイム・タグを比較していちばん新しいものを選ぶ。つまり、プロダクションの条件部のいちばん最初に書かれている条件要素を満足したワーキング・メモリー要素のタイム・タグを比較するのである。
3. 以上で一つのインスタンスエーションが選ばれない場合は、残りのタイム・タグで、より新しいタイム・タグを持つインスタンスエーションを選ぶ。インスタンスエーションの比較のしかたは次のように行う。
  - a) 最新のタイム・タグを比較し新しいものを選ぶ。
  - b) 同じ場合には次に新しいタイム・タグを比較し新しいものを選ぶ
  - c) 一つのインスタンスエーションが選ばれるまで、あるいは他方の比較するタイム・タグがなくなるまで次々にタイム・タグを比較する。しかし、同時にタイム・タグがなくなった場合には選択しない。
4. 以上で一つのインスタンスエーションが選ばれない場合は、より条件の厳しいインスタンスエーションを選ぶ。つまり、プロダクションの条件部で行われている属性値のチェックの回数を比較し回数が多いものを選ぶ。
5. 以上でも選ばれない場合は、任意のインスタンスエーションが選ばれる。

戦略として MEA を使用すると、仕事の順序付が容易になり大規模なプログラムにおいて仕事を分ける場合に有効である。

### 2.3.3 動作 (Act)

戦略に基づいてインスタンスエーションが選ばれると、そのインスタンスエーションが示すプロダクションの動作部を順に実行する。

## 2.4 OPS5 の文法

ここでは OPS5 の文法の主要なものについて説明する。詳しい解説は文献14)および15)に示されている。

### 2.4.1 宣言文

プロダクションの中で使われるクラス名、属性名、外部ルーチン名および属性値として複数の値をもてるベクタ属性は、プログラムの先頭部分で宣言する。

#### 1) literalize

クラス名と、そのクラスに対する属性名を宣言する。

(literalize class a1 a2 ...)

#### 2) external

BASIC, FORTRAN, C 等の他言語で書かれたルーチンを OPS5 から呼び出すときに、そのルーチン名を宣言する。OPS5 では四則演算しか出来ないため、高度な数値計算が必要なときには他言語でプログラムを作り、外部ルーチンとして呼び出す。

(external routine1 routine2 ...)

#### 3) vector-attribute

属性値に複数の値を持たせる場合に、ベクタ属性の宣言をする必要がある。このベクタ属性には、1つのクラスに対して1つしか持てないという制限がある。

(vector-attribute vector)

(literalize class a1 a2 ... vector)

### 2.4.2 LHS

プロダクションの条件要素の属性値には定数の他に変数、述語、選言命題、連言命題を書くことが出来る。また、否定条件要素、要素変数というものもある。

#### 1) 定数 (Constant)

定数には整数と文字列がある。

#### 2) 変数 (Variable)

<var>のように < > でかこまれた文字列であり、一つのプロダクション内でのみ有効で他のプロダクションへの受渡しには使用できない。条件部に2回以上同じ変数が使われている場合は、最初の条件要素でワーキング・メモリー要素の属性値が代入され、2つ目以降ではその代入され

た値とワーキング・メモリー要素の属性値が照合される。

### 3) 述語 (Predicate)

定数と変数に前置される述語には = (等しい), <=> (同じ型), <> (等しくない), < (~より小さい), <= (~以下), > (~より大きい), >= (~以上) がある。<=> は整数であるか文字列であるかを照合する。

### 4) 選言命題 (Disjunction)

<< >> 内にいくつかの値をならべたもので、その中の値のどれか一つがワーキング・メモリー要素の属性値と適合すればよい。値には定数のみをとる。

### 5) 連言命題 (Conjunction)

{ } 内にいくつかの値をならべたもので、その中の全てがワーキング・メモリー要素の属性値と適合すればよい。値には定数の他に変数、述語が前置された定数および変数、選言命題を含んでもよい。

### 6) 否定条件要素

条件要素の前に - を付けると否定条件要素となる。この条件要素と適合するワーキング・メモリー要素が存在しないときに条件が満足されたことになる。否定条件要素はプロダクションの最初の条件要素としてはならない。つまり、プロダクションの最初の条件要素は肯定条件要素でなければならない。

- (condition element)

### 7) 要素変数 (Element Variable)

{ } を用いて変数と肯定条件要素をかこむと、変数は条件要素と適合しているワーキング・メモリー要素と結合し、プロダクションの動作部においてワーキング・メモリー要素を指定するときに用いられる。

{(condition element) <wm>}

or

{<wm> (condition element) }

## 2.4.3 RHS

プロダクションの動作部には make, modify, remove, write, bind, cbind, build, call, halt 等の動作と、その引数を表すものとして要素指示子、定数、変数、関数がある。

### 1) 要素指示子 (Element Designator)

要素指示子はワーキング・メモリー要素を指定するもので、数値または要素変数である。数値 K が用いられた場合には、条件部における K 番目の肯定条件要素と結合しているワーキング・メモリー要素を指す。また、要素変数のときは条件部で結合した要素を指す。

### 2) 関数 (Function)



関数は ( ) でかこんで使われる。

i) accept

accept はユーザーからの入力を読み込む関数である。

ii) compute

compute は数値演算をする関数で、演算には四則演算 +, -, \*, // および余り % がある。また、演算の優先順位はなく右から左へ演算を行うが、( ) を使うことによって順番を変えることが出来る。

(compute ( <a> \* <b> ) - 4 \* <c> // <d> + 1)

iii) crlf

crlf write のときに使われる関数で、改行を行う。

この他にも OPS5 で使われ関数として acceptline, genatom, litval, rjust, substr, tabto 等がある。

3) 動作 (Action)

i) make

make はワーキング・メモリーに新しい要素を加える。引数には1つのクラス名といくつかの属性名と値の対をとる。

(make class ^al v1 ^a2 v2 ...)

ii) modify

modify はワーキング・メモリー要素の属性値を変更する。引数には要素指示子とその要素に含まれる属性名と変更する値の対をとる。

(modify ed ^al v1 ^a2 v2 ...)

iii) remove

remove はワーキング・メモリーから要素を削除する。引数には要素指示子をとる。

(remove ed1 ed2 ...)

iv) write

write は情報を出力する。引数には数値、文字列および変数をとる。文字列をそのまま書くと、小文字でも大文字として出力されるが、| | でかこむと小文字の出力もできる。

(write (crlf) <var> | This is a test of write|)

v) bind

bind は変数に値を代入する。引数には変数と値をとる。値には accept, compute 等の関数も使用できる。

(bind <var2> (compute <var1> + 1))

vi) halt

halt はプログラムを終了する。

vii) call

call は他言語で書かれたルーチンを呼ぶ。引数にはルーチン名と受渡す値をとる。

(call external v1 v2 …)

#### 2.4.4 STARTUP 文

STARTUP 文は、プログラムが実行されるときに、最初に一度だけ無条件に実行される。プログラムを実行するための初期設定をするのが目的で make, watch, run 等がある。make は RHS の make と同じでワーキング・メモリーに要素を加え、watch は実行中のワーキング・メモリーと競合集合の変化、選ばれるプロダクションの情報の出力を制御する。run を指定すると startup 文を実行したあとに続けてプログラムを実行するが、指定しないときはプログラムは止まりコマンド入力待ち状態になる。

### 3. 橋梁下部構造基礎形式選定システム

橋梁等の構造物におけるコンピュータの利用法としては、設計段階で構造計算を高速に効率的に行うことが主であり、構造形式の選定というような計画段階におけるコンピュータの利用は、ほとんど行われていなく、専門家の判断に依存しているのが現状である。そこで、橋梁下部構造の基礎形式選定にエキスパート・システムを応用し、プロダクション・システムをベースとする OPS5 によって「橋梁下部構造基礎形式選定システム」を開発した。

基礎形式は図-7 に示す11形式をこのシステムの選定対象とした。基礎形式の選定条件には①地盤条件、②施工条件、③上部構造からの条件、④環境条件、⑤工費・工期がある。①の条件としては支持層の状態、中間層の性質、地下水の状態等がある。②の条件としては施工深度、作業空間、水深等がある。③の条件としては荷重の大きさ、上部構造の力学的性質等がある。④の条件としては騒音、振動、有害ガス等がある。⑤の条件は①～④の条件に適合する形式に対して考慮される。実際の選定では、①～④の条件によって基礎形式を複数個選定し、さらに⑤の条件により比較検討を行い最終決定にいたるが、このシステム

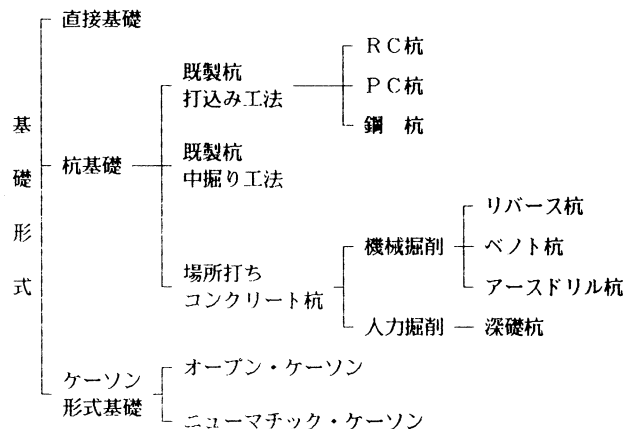


図-7 基礎構造の分類

では①～④の条件のみで基礎形式の選定を行う。表-1にこのシステムで用いた条件を示す。○が適合，△が検討が必要，×が不適を表している。

プログラムは図-8のように大きく4つの部分からできている。初期設定部では構造というクラスとデータというクラスを宣言し、構造には属性値として基礎形式や各種条件による判断などが入り、データには判断の材料となるデータが入る。また、プログラムを実行するために必要な要素をワーキング・メモリーに作っておく。入力部は必要なデータの入力を行う

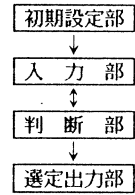


図-8 プログラム構成

う部分で、指示に従って入力をする。判断部には表-1の条件の知識データ・ベースがあり、入力されたデータを基にデータ・ベースと比較して、適・不適の判断を行う。そして、選定出力部では判断結果から基礎形式を選定し出力を行う。条件に修正がある場合には判断部のデータ・ベースを修正し、追加された場合には入力部にデータ入力のためのプロダクションを増やし、判断部にデータ・ベースを追加することによりシステムは対応できる。

```

(p 地下水流速による判定・・・3m/min・検討
(データ 条件 地下水流速 は > 2)
{(構造 は << 中掘り オープン >>
  検討 <val> 地下水流速 nil) <wm>}
--->
(bind <vall> (compute <val> + 1))
(modify <wm> 検討 <vall> 地下水流速 終了))
    
```

図-9 データ・ベースの例

図-9は判断部のデータ・ベースの1つである。もし地下水の流速が3m/minよりも大きく構造が中掘りまたはオープンケーソンであれば、検討の値に1加えるというプロダクションである。検討の値を条件部から変数として得、動作部で1加えてワーキング・メモリーの値を変更し、判断終了という値も与えている。

図-10は実行例である。アンダー・ラインのある部分が入力データで、画面の指示に従って入力を行うと最後に基礎形式が選定されて画面に出力される。

\*\*\* 橋梁下部構の基礎形式選定エキスパート・システム \*\*\*

```

支持方式      完全支持      : 1
               不完全支持    : 2
               摩擦支持      : 3 ... 1
支持層の傾斜角度 (°) ... 20
支持層の凹凸  大:1   小:0   ... 1
               .
               .
               .
    
```

```

*** 選定基礎形式 ***
・ベノト杭      検討項目が0個あります。
・リバース杭     検討項目が0個あります。
    
```

\*\*\* 終了 \*\*\*

図-10 システム実行例

#### 4. おわりに

本論文では、エキスパート・システムの土木工学への応用として、エキスパート・シェル

表-1 基礎形式の選定条件

			直	打 込 み			中	場所打ち			深	ケーソン	
				接	R C	P C		鋼	掘 り	リ バ ー ス		ベ ノ ト	ア ー ス
地	支持層の ・状 支態	完全支持 完全支持 完全支持 (30°以上) 大	○	○	○	○	○	○	○	○	○	○	○
			○	○	○	△	○	○	○	○	○	○	○
盤	中間層の 状態	粘性土 N値 ~4 4~10 10~20 砂 N値 ~15 15~30 30~ 礫層 層厚(cm) ~10 10~30 30~ 液状化する	○	○	○	○	○	○	○	○	○	○	○
			○	△	○	○	○	○	○	○	○	○	○
			○	×	△	○	○	○	○	○	○	○	○
			○	○	○	○	○	○	○	○	○	○	○
			○	△	○	△	○	○	○	△	○	△	○
			○	×	×	△	○	○	△	○	○	○	○
件	地下 水	地下水位 フェック下面以上 フェック下面以下 かつ杭先端以上 地下水の流速 3m/min~ 被圧地下水 地表より0~2m 2m~	△	○	○	○	○	○	○	○	×	○	○
			○	○	○	○	○	○	○	○	△	○	○
			×	○	○	○	△	×	×	×	×	△	○
			△	○	○	○	△	△	△	△	×	△	○
施	施工 条件	施工深度(m) 0~5 5~10 10~20 20~30 30~60 作業空間が狭い 水上施工 水深(m) ~5 5~	○	○	△	△	△	×	×	×	△	△	×
			△	○	○	△	○	△	△	△	○	○	○
			×	△	○	○	○	○	○	○	○	○	○
			×	△	△	○	△	○	○	△	△	○	△
			×	×	×	○	×	○	△	×	×	×	×
			○	△	△	△	△	○	△	△	○	△	△
上	よ	部	荷 重 (t)	○	○	○	○	○	△	△	△	△	△
				○	○	○	○	○	○	○	○	○	△
				○	△	△	△	△	○	○	○	○	△
				○	×	×	△	×	△	△	△	△	○
環	境	条	有害ガスが発生する 騒音・振動の規制がある 影響を受ける隣接構造物がある	△	○	○	○	○	○	○	×	○	×
				○	×	×	×	○	○	○	○	△	○

OPS5 による橋梁下部構造の基礎形式選定システムの開発について述べた。OPS5 は FORTRAN や BASIC といった言語とは全く違う言語であり、主に知識をデータ・ベース化するという作業だけでシステムが完成し、プログラムの流れ、アルゴリズムはあまり考える必要はない。しかし、複雑な数値計算はできなく不便な点であるが、FORTRAN 等によってプログラムを作り、それをサブルーチンまたは関数として呼び出すことによって解決することができる。

今回開発したシステムは完全なものとは言えないが、ある程度使用に供するものであり、知識の修正、追加を行うことにより実用的なシステムに近づくものと思われる。

システム開発にあたっては、本学の共同利用施設である VAX-11/750 を利用した。

また、VAX-11/750 の利用にあたり、産業機械工学科、久保洋教授から多くの助言をいただきましたことを記し、ここに謝意を表します。

#### 参考文献

- 1) Buchanan, B. G. and Feigenbaum, E. A.: DENDRAL and Meta-DENDAL: their application dimension, Journal of Artificial Intelligence, Vol.11, 5-24, 1978.
- 2) Martin, W. A. and Fateman, R. J.: The MACSYMA system, Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, pp.59-75, March 1971.
- 3) Shortliffe, E. H.: MYCIN: Computer-based Medical Consultations, American Elsevier, NY, 1976.
- 4) 松尾義弘, 枝村俊郎, 宮垣直也: 斜面宅地開発コンサルテーションの試み, 土木学会第40回年次学術講演会講演概要集第IV部, pp.183-184, 1985.
- 5) 白石成人, 松本勝, 谷川浩司: 新しいコンピュータ言語 Prolog の橋梁形式選定システムへの適用, 橋梁, pp.2-10, 1985. 5.
- 6) 瀬口靖幸, 多田幸生, 田中正夫, 菅沼真一郎: FEM モデル化過程のエキスパート・システム構築の一つの試み, 情報処理学会第31回全国大会, 2M-2, pp.925-926, 1986.
- 7) 大阪一, 土山茂希, 市川康明: エキスパート・システムの岩盤力学への適用について, 土木学会第41回年次学術講演会講演概要集第III部, pp.601-602, 1986.
- 8) 白石成人, 吉田均, 川上宏一郎: RC 床版の耐用性評価のためのエキスパート・システムの開発に関する基礎的研究, 土木学会第41回年次学術講演会講演概要集第I部, pp.601-602, 1987.
- 9) 八賀 明, 長田弘康, 松岡彰彦, 河田博之, 鳥取誠一: 鉄道土木構造物診断システム (その1), 情報処理学会第31回全国大会, 1L-5, 1986.
- 10) 中村秀治, 松浦真一, 松井正一, 寺野隆雄: 知識工学的手法に基づく水力鋼構造物の寿命予測, 土木学会論文集, No.368/I-5, pp. 301-310, 1986.
- 11) 中村秀治, 松浦真一, 寺野隆雄, 篠原靖志: 水力鋼構造物の寿命予測エキスパート・システムとその適用, 土木学会論文集, No.374/I-6, pp. 513-521, 1986.
- 12) 三上市蔵, 江澤義典, 森澤敬文, 田中成典, 朝倉隆文: RC 床版の点検・補修に関するエキスパート・システム, 第11回電算機利用に関するシンポジウム講演集, 土木学会, pp.159-166, 1986.
- 13) 三上市蔵, 江澤義典, 田中成典, 朝倉隆文: 道路橋鉄筋コンクリート床版の損傷度判定と補修工法選定のための知識ベースエキスパートシステム, 構造工学論文集, Vol. 33A, pp. 317-326, 1987.
- 14) 日本 DEC: VAX-11 OPS5 ユーザーズマニュアル, 1984.
- 15) 日本 DEC: VAX-11 OPS5 ユーザーズガイド, 1984.