

☆



## 知識工学と原子力技術

(㈱日立製作所)元 田 浩,  
 (三菱電機㈱)荻野 敬迪, (日本原子力事業㈱)関水 浩一,  
 (日本原子力研究所)篠原 慶邦, (東北大学)北村 正晴

知識工学の応用が原子力技術の広い範囲にわたって急速に進みつつある。従来は経験的法則やノウハウによって解決が図られていた問題に対して、知識工学は新しいアプローチを提供するものであり、その潜在的可能性は極めて大である。

本稿では、この技術について様々の実際的な適用例を題材に解説し、将来の展望を行う。

☆

☆

### I. はじめに

“知識工学”という手法を応用した研究発表が本学会で初めてなされたのは1981年であるが、その後今日までの短い間に、この種の研究や技術開発の数は飛躍的な増加を示している。「昭61年会」(3月26～28日, 京大原研)では14～15件程度の発表が知識工学的手法を応用したものであった\*。

1985年9月の「原子力プラントの運転制御への計算機応用」<sup>(1)</sup>や、1986年4月の「原子力システムにおける人的因子」<sup>(2)</sup>などのANSトピカル・ミーティングでは、いずれも20～30%に達する報告がなんらかの意味で知識工学と関連を有している。応用対象の領域は、当初は異常診断や運転ガイダンスが中心であったが、最近ではこれに加えて、アラーム処理, 燃料交換計画, 機器配置, 配管ルーティング, 機器寿命予測, 保守計画など非常に広範囲の問題に対する応用が進展中である。原子力のすべての分野にわたって応用が可能といっても過言ではないほどその適用性は高い。計算機技術の進歩が原動力の1つであることはいままでもないが、従来はどうしても経験やノウハウに頼らざるを得なかった技術課題に対し、強力かつ柔軟なサポートを行う手法が初めて提供されたことが、この急速な発展の最大の理由であろう。

本「会誌」では、すでに1984年に知識工学についての基礎的な解説が掲載されているが<sup>(3)</sup>、それ以後の応用領域の一層の拡大と関連技術の進歩、さらに各分野に与えつつあるインパクトの大きさにかんがみて、再度の解説記事を企画した次第である。

以下では、まず知識工学の具体的なイメージと効用をおおまかに認識して頂くため、実際の応用の事例を先に数ケース紹介する。次いで、その基礎である知識の

表現とそれに関する推論機構について理論的背景を説明し、最後に技術的課題と将来の発展方向についての展望を述べる。

### II. 具体的応用例

#### 1. 事象関係ネットワークの記述と解析

異常診断, 確率論的安全評価(Probabilistic Safety Assessment; PSA)などの分野では、知識工学の応用が特に急速に進みつつある。事象の因果関係の記述およびそれに関する推論は、いわゆる知識工学(あるいは人工知能)向き言語であるLISP, PROLOG等を用いると極めて容易に行い得ることがその理由である。自然言語で下記のように記述される因果関係を例として考えて見る。

自然言語:

- (1) Aが生じるのはBまたはCが生じた時である。
- (2) Bが生じるのはDが生じるかEとFが同時に生じた時である。
- (3) Cが生じるのはGとHかIとJが同時に生じた時である。

入力データとして基本事象D～Jのステータスが提供された場合に事象Aが生起するか否かを判断することが課題である。この因果関係を代表的計算機言語であるFORTRAN, 知識工学向き言語PROLOGで記述すれば、それぞれは次のような表現となる。

*Knowledge Engineering and Its Applications to Nuclear Technology*: Hiroshi MOTODA, Takamichi OGINO, Ko-ichi SEKIMIZU, Yoshikuni SHINOHARA, Masaharu KITAMURA.

(1986年6月23日受理)

\* (付記)「昭61分科会」(10月16～18日, 九大)の指定テーマ①(知識情報処理)には24件発表の予定。

**FORTRAN:**

- (1) IF((G.AND.H).OR.(I.AND.J))  
C='TRUE'
- (2) IF(D.OR.(E.AND.F)) B='TRUE'
- (3) IF(B.OR.C) A='TRUE'

**PROLOG:**

- (1) A:—B;C.
- (2) B:—D;E,F.
- (3) C:—G,H;I,J.

入力データ(例えば D, H, J が生じたという情報)が与えられた時, A が生じるか否かを判定することは, 自然言語方式では混乱を招きやすく容易ではない。計算機言語は相対的に明快な記述となっているが, FORTRAN の場合には(1)~(3)の記述の順序という知識の本質とは関係ない要素が判断の流れに決定的な影響を与えてしまう。また, 知識(判断ルール)の内容を変更する必要が生じた場合にも, 全体の記述を配慮しつつ変更を行わねばならずその困難は大である。これに対して PROLOG 形式では各ルールは基本的に独立にあるので, 必要な部分だけを単独に変更することが可能である。ルールの適用, すなわち推論の結果はルールの出現順序には影響されないで, 変更, 追加などが容易に行える。因果関係がより複雑になった時は, この違いは一層際立ったものとなる。知識情報処理における道具(ツール)の占める役割の大きさがこの例から理解されよう。

以下の各節では PROLOG を利用した記述が多く現れるが, PROLOG そのものの知識はなくとも概念的に内容を追うことは可能なように工夫がなされている。これらの言語については評価の高い解説書<sup>(4)(5)</sup>も多いので, これらを一読して頂ければ本稿の内容はより明かに理解されよう。

**2. 原子力プラントの診断, (1) 基本的応用**

プラント診断機能を実現するために使用する知識は, 異常事象の因果関係に関するもの(表層知識)と診断対象の特性, 構成等の設計に関するもの(深層知識)に大別して考えることができる。実用的なプラント診断システムを構成する時は, 両方の知識を適宜使い分ける必要があると思われるが, 本節では表層知識を用いた場合について説明する。表層知識の知識源として, 運転要領書や運転員の経験等がよく知られているが, 信頼性解析の基本的な手法の1つである FMEA (Failure Modes and Effect Analysis)の結果を用いても得ることができる。ただし, FMEAの労力に比較し運転要領書から知識を得る方が容易であるため, 原子

力プラントのように運転要領書が完備している場合には, これに基づいて知識ベースを構成するのが現実的なアプローチと考えられる。簡単な例について次に考えてみる。

〔例〕原子炉に正の反応度添加, 制御棒引き抜き, 冷却材ホウ素濃度希釈等が生ずると, 原子炉冷却材平均温度  $T_{AVG}$  が上昇する。

これは, " $T_{AVG}$ "なる警報が点灯した時に, 運転員がその原因を判断するための運転要領書の一部を引用したものである。これから, 次の知識を得ることができる。

- ・原子炉に正の反応度が添加されると,  $T_{AVG}$  は高となる。……ルール 1
- ・制御棒が引き抜かれると, 正の反応度が添加される。……ルール 2
- ・冷却材ホウ素濃度が希釈されると, 正の反応度が添加される。……ルール 3

2番目と3番目の知識は手引そのままの表現となっていないが, この方が物理的には正しい記述である。知識の獲得過程において, 一般にこのような知識の吟味が必要になる。

診断機能を実現するために使用する表層知識は, その内容により3種類に分類できる。

- (1) 異常事象判定ルール: プロセス量をそのしきい値と比較して異常事象発生の有無を判定する。

〔例〕if(制御棒位置>基準値), then(反応度(正))

- (2) 異常事象関係ルール: 異常事象間の関係を示す。前述したルール1~3は, 本ルールの範疇に属する。

- (3) 機器故障判定ルール: 機器が故障(配管破損等も含む)したと判定するための条件を示す。

〔例〕if(機器入力(高) and 機器出力(低)), then(機器(故障))

機器故障判定ルールにおいては, 機器の特性モデルを用意しておき, その出力の推定値と実測値との差が許容範囲内にあるか否かにより機器故障の有無を判定するモデル規範型的方式もある<sup>(6)</sup>。この場合, 特性モデルは FORTRAN 等の数値処理言語による記述が自然であり, 診断系は記号処理と数値処理のハイブリッド型として構成される<sup>(7)</sup>。

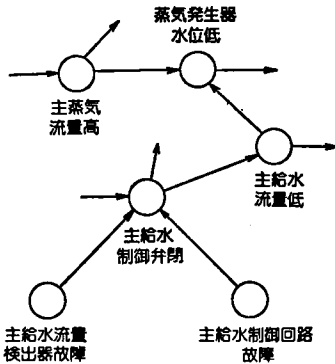
異常診断の処理は, プラント運転上の重要なパラメータがしきい値を越えた時, または運転員が異常事象の原因を問い合わせた時に開始される。以下では主給水制御系まわりの異常事象関係ルールを例にとって, 診断プロセスを説明する。なお, 説明を簡潔にするた

めに異常事象判定ルールはすでに起動され異常事象の生起の情報は得られているとし、機器故障判定ルールは省略する。

〔知識ベース〕

- rule (if (主蒸気流量(高)), then (蒸気発生器水位(低))).
- rule (if (主給水流量(低)), then (蒸気発生器水位(低))).
- rule (if (主給水制御弁(閉)), then (主給水流量(低))).
- rule (if (主給水流量検出器(故障)), then (主給水制御弁(閉))).
- rule (if (主給水制御回路(故障)), then (主給水制御弁(閉))).

以上のルールにより表現されている異常事象間の関係を第1図に示す。



第1図 異常事象間の関係

〔異常事象〕

- event (蒸気発生器水位(低)).
- event (主給水流量(低)).
- event (主給水制御弁(閉)).
- event (主給水制御回路(故障)).

ルールの形式からわかるように、条件部と結論部の(異常)事象は、測定可能なプロセス量から直接判定することができる。これは異常事象間の関係を示すルールの一般性のある特徴であり、“条件部と結論部が共に成立すれば、条件部は結論部の原因と考えられる。”という意味を持っていると解釈できる。このような考え方に基づく推論の方法は、abduction(診断の論理)と呼ばれている。この考え方に従って、縦型探索方式による推論機構は次のように記述できる。プログラム中の\*の付いた記号は変数を表わす。

〔推論機構〕

```

inference (*a):-
    event (*a),           ①
    ask (*a),             ②
    fail.                 ③
inference (*).
ask (*a):-
    rule (if (*c), then (*a)), } ④
    event (*c),             }
    ask (*c).               ⑤
ask (*a):-
    ≠+rule (if (*c), then (*a)), ⑥
    write (“故障ノード⇒”), } ⑦
    write (*a),             }
    nl.
  
```

①問合せのあった異常事象の発生の有無確認, ②異常原因の探索, ③バックトラックによるすべての異常原因の探索, ④abductionによる論理処理, ⑤因果関係を後向きに探索(再帰処理), ⑥故障を条件部に持つルールに到達し, これを異常原因と判定。なお≠+はnotの意味, ⑦結果を出力する。なお, 本PROLOGでは日本語によるルールの記述や結果の出力が可能である。

以上の条件で、蒸気発生器水位(低)なる異常事象の原因を質問し、結論(主給水制御回路(故障))に至る処理の過程を以下に示す。

```

call inference (蒸気発生器水位(低))?
call event (蒸気発生器水位(低))?
exit event (蒸気発生器水位(低))
call ask (蒸気発生器水位(低))?
call rule (if (*), then (蒸気発生器水位(低)))?
exit rule (if (主蒸気流量(高)), then (蒸気発生器水位(低)))
call event (主蒸気流量(高))?
fail event (主蒸気流量(高))
redo rule (if (主蒸気流量(高)), then (蒸気発生器水位(低)))?
exit rule (if (主給水流量(低)), then (蒸気発生器水位(低)))
call event (主給水流量(低))?
exit event (主給水流量(低))
call ask (主給水流量(低))?
↓
call ask (主給水制御回路(故障))?
  
```

```
call rule(if (*), then(主給水制御回路(故障)))?
fail rule (if (*), then (主給水制御回路(故障)))
```

↓

〔実行例〕

?— inference (蒸気発生器水位(低))。

故障ノード ⇒ 主給水制御回路(故障)

yes

処理の過程の説明の中で用いている記号について説明する：

call は手続きの最初の呼び出しを表わす

exit は手続きが成功した時の戻りを表わす

fail は手続きが失敗したことを表わす

redo はこの後の目標が失敗し、システムが後戻りして別解を求めようとすることを表わす。

PROLOG を用いたプラント診断処理のイメージを紹介したが、実用的なシステムを構築するには知識獲得において細心の注意と考察が必要である。特に、断片的な知識(ルール)の集合が全体として論理ループを持たないか、因果関係をたどってゆくと矛盾事象に到達しないか、等の検討が重要である<sup>6)</sup>。

### 3. 原子力プラントの診断, (2) 時間的変化の考慮

前節では、プラントの状態が時間的に変化しない場合のプロダクション・ルールを用いたプラント診断の例が述べられたが、本節では状態が時間的に変化する場合、特にプラント事故診断への応用の場合について説明する。

#### (1) 基本的な考え方

状態が時間的に変化する場合にプロダクション・ルールを適用するためには、基本的にはプロダクション・ルールに時間的変化に関する知識を組み入れればよいだけである。ただし、知識を効率的に利用するためには、時間的変化に関する知識をどのような形で取り入れるかが問題であり、いろいろと工夫を必要がある。さもないと、本質的でない知識をやたらと知識ベースに取り込むことになり、知識ベースの効率的な利用が困難になるおそれがある。

一般に、ある事象についての知識の形態は、得られた情報の性質によっていろいろな観点からの分類が可能である。例えば、

- (1) 経験的に得られたものか、理論的なものか。
- (2) 定性的な知識か、定量的な知識か。
- (3) 現象についての知識か、機構についての知識か。
- (4) あいまいさを含むか、含まないか。

などである。経験的に得られた知識は一般に現象的で

かつ定性的であることが多く、理論的に得られる知識は機構的かつ定量的であることが多いが、常にそうであるとは限らない。

時間的に変化する事象についての知識の形態も様々であり、前述のように事象について経験的あるいは定性的な情報しか得られていない場合もあれば、各変量間の動的な因果関係について詳細な動特性モデルが作られている場合もある。前者の場合には、事象についての経験的あるいは定性的な関係をそのままルールとして表現して用いることができる。後者の場合は、その動特性モデルを用いて必要に応じて計算機シミュレーションによって事象の定量的な時間的変化を求め、それをその事象の時間的変化についての具体的な知識として用いることもできる。一般に複雑なプロセスについては、これらの種々の形態の知識が入り混ざっているから、それぞれの知識の形態に適した知識表現を組み合わせ用いることが必要である。

#### (2) プラント事故に関する知識

現実のプラントにおいて発生する確率が極めて小さく、実際の経験が極めて乏しい事故の診断に知識工学の手法を応用する場合、必要な知識ベースを構築するための知識は、

(1) プラントの設計や運転経験に基づくプロセスの因果関係についての基本的な知識

(2) シミュレーションによって得られる種々の事故時のプロセスの典型的な動特性に関する知識が用いられる。

プラントの状態の時間的変化に関係する信号は、大別すると次の通りである。

(1) オン・オフ的信号(インターロック動作、スイッチ操作など)

(2) 連続的信号(温度、圧力、流量など、各種プロセス量のアナログ信号)

オン・オフ的信号については、種々の信号の発信の前後関係や発生時間差のような関係が重要であるが、信号発信の時間的前後関係だけの定性的な知識が与えられている場合や、発生時間差の情報を含んでいる定量的な知識が与えられている場合もある。

また連続的信号については、その変化率や信号の絶対的な大きさの時間的挙動が知識として重要であるが、変化率の符号だけや信号の絶対値の変化傾向だけといった定性的な知識しか与えられない場合や、それが時間関数として定量的に与えられた知識の場合もある。

## (3) プラント事故診断システム

## 知識ベース構造の例

運転中のプラントの状態が時間的に変化する場合の診断に知識工学的手法を用いた例として、原研で開発されたプラント事故診断システム DISKET<sup>900</sup>について、その基本的な考え方を説明する。

このシステムでは、知識ベースは、第1表に示すように、データ定義部とルール部から構成されている。

第1表 DISKET における知識ベース構造の概念

データ定義部	ルール部
1. 生起事象群 アナンシエータ プロセス変数 弁, ポンプ等	知識ユニット1(大局レベル)
2. 仮説群 事故分類 内部仮説 対応策	知識ユニット2(中間レベル) 知識ユニット3(詳細レベル) 知識ユニット4(対応策)

データ定義部は生起事象群と仮説群からなる。生起事象群は原子炉機器の動作状態やプロセス量のパラメータに関する定義を含んでおり、前者に対しては真偽あるいは可否といった論理変数の値で表現し、後者に対しては連続変数としての数値が用いられる。仮説群は事故を事故仮説の詳しさに応じた階層化により大, 中, 小といった幾つかのレベルに分類し, 例えば, 小分類において事故仮説が見付からなくても, 中分類あるいは大分類で事故仮説が見付かれれば, ある程度の事故対応策がとれるようにしている。対応策も仮説の一種である。また, 知識ベースに基づく効率的な推論を行うために, 推論プロセスを制御する内部仮説なるものも用いられている。

ルール部はデータ定義部で定義された生起事象(F)や仮説(H)の間の関係に関する知識を条件部と実行部の関係で定義したもので, 生起事象間の関係(FFルール), 生起事象と仮説間の関係(FHルール)および仮説間の関係(HHルール)が知識表現として与えられている。

実行部の仮説には, それに対する確信度を表わす0.0~1.0の重みがつけられている。さらに, 正負の符号によってそれぞれ肯定的および否定的な確信を表わしており, +1.0は絶対的肯定を, また-1.0は絶対的否定を表わすようにしている。

条件部にはそれぞれ複数の生起事象や仮説を用いることもでき, それらを同時に満たすAND結合, それらのうちの幾つかを満たすOR結合およびそれらの前

後関係を満たすAFTER結合を組み合わせた条件を記述することができる。

説明を簡単にするために多少の省略を行なって, 例えばFHルールの例として, 「事象Aが生起したとき, 仮説Bが成り立つ(すなわち確信度1.0で)成り立つ」という知識を

$$(A) \cdots \rightarrow H(B, 1.0)$$

と表わすとすれば, 「事象Bの後に事象Aが生起したときは, 仮説Cが確信度0.6をもって成り立つ」という知識は

$$F(A) \text{ AFTER } F(B) \cdots \rightarrow H(C, 0.6)$$

のような形で表わされる。確信度の計算にはエキスパートシステムMYCINで用いられている方法が採用されている。

DISKETの最初のもは計算機言語LISPの変形であるUTILISPを用いて書かれたが, FORTRAN版も作られている。

このプラント事故診断システムを加圧水型炉のプラント・シミュレータに適用して行われた事故診断のシミュレーションについて簡単に述べると, 知識ベースには約300種類のルールが用いられており, シミュレータから得られる170種類の信号を入力として用いた種々の事故診断のための推論において評価されたルールの数は約100~250である。より詳細については文献(9), (10)をご覧ください。

## 4. 定性推論と診断知識の獲得

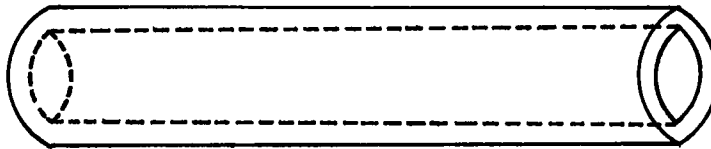
診断のための知識を獲得することは, 知識工学的手法を用いると否とにかかわらず診断の成否を握る重要な作業である。エキスパートシステムの診断システム構築の際には経験的知識が利用されるのが普通であるが, 対象の特性をある程度定量的に事前評価しうる原子力プラントでは, 第II-3節でも述べられたように, シミュレーション計算が積極的に利用されて良い。ただし, ここで次の2点に注意を払う必要がある。

- (1) 対象の動特性は, 設計データと運転条件が指定されれば明確に定まるはずである。換言すれば, これらの知識を基礎として, 余計な中間的処理(数式モデルの作成, 数値計算アルゴリズムの選定, アルゴリズムパラメータの設定, コーディング等)を極力少なくした計算法が, 効率, 人的過誤の低減などの観点から望ましい。
- (2) 熟練運転員は現象を巨視的に認識して診断を行なっている。ある変数の変動幅が5%であるか6.5%であるかといった詳細な数値にはあまり情報量は含まれていない。このことに着眼す

れば、シミュレーションに際して、精度は最重要問題ではなく、むしろ適当な粗視化を行なって全体的バランスの良い結果を得ることが重要である。

(1)の観点からプラントデータベースに蓄積されたデータに基づいて中間的なモデルを経由せずに直接に推論を実行しプラント挙動を解析する方式、(2)の観点から挙動をおおまかに把握できる定性シミュレーション方式を併用した試みがなされている<sup>(11)</sup>。これについてその要点を紹介する。

プラントの設計情報を、構造に関する情報と構成物質の性質に関する情報に分離して整理、格納したデータベースを作成する。構造データファイルは階層的に構築され、上位は機器、配管等の結合関係がネットワーク的に、下位は機器の形態、内部の構造および状態を直接表現方式を用いて記述している。直接表現方式は第2図に概念的に示されるように、プラント構成要素の幾可学的形態を2次元のセルと呼ぶ配列で近似的に表現したものであり、各セルにはその領域の物質組成や物理的状态のデータが格納されている。



PIPE (X5, X6)

*	air(t=295,p=10)	air(t=295,p=10)	air(t=295,p=10)	*
X5.1	metal(t=640,tb=640,ttm=50,ttw=50,p=640,pb=640,pt=50)	metal(t=640,tb=640,ttm=50,ttw=50,p=638,pb=638,pt=50)	metal(t=640,tb=640,ttm=50,ttw=50,p=636,pb=636,pt=50)	X6.1
X5.2	water(t=640,tb=640,ttw=50,ttm=50,p=640,pb=640,pt=50,w=100,wb=100,wt=50,s=2)	water(t=640,tb=640,ttw=50,ttm=50,p=638,pb=638,pt=50,w=100,wb=100,wt=50,s=2)	water(t=640,tb=640,ttw=50,ttm=50,p=636,pb=636,pt=50,w=100,wb=100,wt=50,s=2)	X6.2
X5.3	metal(t=640,tb=640,ttm=50,ttw=50,p=640,pb=640,pt=50)	metal(t=640,tb=640,ttm=50,ttw=50,p=638,pb=638,pt=50)	metal(t=640,tb=640,ttm=50,ttw=50,p=636,pb=636,pt=50)	X6.3
*	air(t=295,p=10)	air(t=295,p=10)	air(t=295,p=10)	*
*	*	*	*	*

第2図 プラント構成要素の構造データの直接表現例

このような構造情報に対し、各物質の物理的ステータス(例えば液体か気体か)という物性情報は独立に定まると考えてよい。また、互いに接する2領域間の相互作用による状態遷移のルールも、基本的には構造と独立に規定することができる。これらの情報をまとめて物性データファイルに格納する。プラント挙動のシミュレーションは、構造データファイルの各セル内に格納されたプラント状態データに対して物性データに格納された状態遷移ルール中の適合するものを空間的に同期させつつ適用することで行われる。

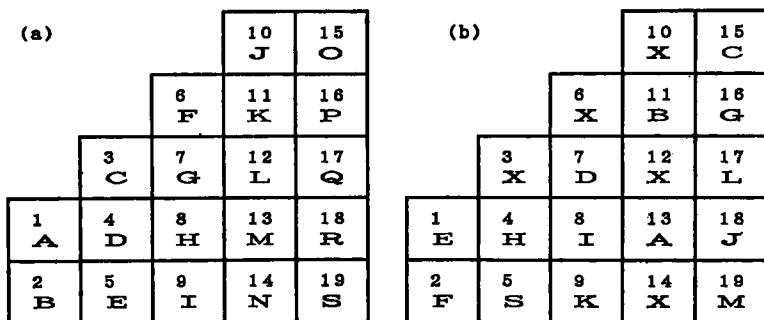
この方式は体系に粗い空間的離散化を施し、かつ微分方程式などを用いない簡略化された状態遷移法則を

利用した、大幅な近似に基づく定性シミュレーションといえる。しかしそれでもなお、通常の意味での定性推論<sup>(12)</sup>よりは近似精度が良く、様々な過渡現象に対して、定量的手法によるシミュレーションコードと合致度のよい結果を与えることが確認されている<sup>(11)</sup>。人工的建造物である原子力プラントの動的挙動は、細菌感染に対する人間の反応などに比較すればはるかに明確に規定されうるはずである。この観点に立つならば、シミュレーションによる知識獲得は、さらに積極的に進められてよいが、その際に、上記(1),(2)のような合目的性からの配慮がなされることが望ましいと考える次第である。

## 5. 燃料移動計画の立案

原子力発電所の定検時に実施される燃料移動作業は、定検工程の1/3近くを占める。燃料移動に要する時間は、燃料集合体を燃料交換機で把み上げたり所定位置に挿入するに要する時間(すなわち上下方向操作時間)と、燃料の空間的移動に要する時間とから成る。前者の方が支配的であるため、燃料の移動回数を最小にすることが最も重要になる。これに際し、考慮すべき制約条件として次のものがある。

- (1) 炉停止余裕の制限条件
- (2) DBG(Double Blade Guide)の挿入



第3図 燃料装荷の初期(a)および目標(b)パターン

第3図のような模擬的炉心での燃料の移動操作を分析すると、次のような手順に要約できる。

- (i) ←N(14)←X
- (ii) ←O(15)←C(3)←X
- (iii) ←P(16)←G(7)←D(4)←H(8)←I(9)  
←K(11)←B(2)←F(6)←X
- (iv) ←Q(17)←L(12)←X
- (v) ←R(19)←J(10)←X
- (vi) (A(1))←E(5)←S(19)←M(13)←A(1)

ここで( )内の数字は燃料集合体のとりうる位置、A～Sは炉心内の燃料集合体識別タグ、Xは新燃料を表わす。手順(i)は、位置14に存在する集合体Nを炉外に取り出し、その空いた位置に新燃料を装荷することを意味する。(ii)～(v)も同様に炉外に燃料を取り出す操作に始まり新燃料の装荷に終る燃料移動を表わしている。このような一連の燃料移動をパスと呼ぶ。(vi)では燃料移動が循環している。このような移動をループと呼ぶ。ループ中の燃料を移動させるためにはこのうちの1本を一度炉外に取り出す必要があるが、これを実行すればループはパスになる。

燃料交換中に実現する燃料装荷パターンはパスのリストとして表現できる。例えば、初期の装荷パターン

(1)は、燃料移動中に過渡的にはあっても1本の制御棒のまわりに反応度の高い燃料が集中した状態を防ぐものであり、この制御棒が抜け落ちる事態が生じても炉心が臨界になることを妨げている。(2)のDBGとは、制御棒を囲む燃料集合体のうち何本かを取り除いた場合に制御棒が支えられずに倒れてしまうことを防ぐものであり、制御棒を囲む4燃料集合体の配置パターン(1本またはそれ以上が抜けた状態)が限定される。

これらの制約を考慮に入れて燃料移動計画を立案するわけであるが、制約条件および燃料移動の表現に適した定式化が必要である。

では燃料移動は全く実行されていないので、このパターンが(0, 0, 0, 0, 0)で表わされ、目標の装荷パターンでは上記(i)～(vi)の操作が完了しているので各パスの長さのリスト(2, 3, 9, 3, 3, 5)が対応する。燃料交換途中での装荷パターンがパスリストPATで表わされていたとしよう。このパターンから、どのパスに属する燃料を移動すべきかという問題が、次のPROLOGで表現できる。

```
state (FINAL PAT, PAT, (PATH;SOL)) (1)
:-select path(FINAL PAT, PAT,
PATH, NEW PAT), (2)
eval pattern(NEW PAT, PATH), (3)
state(FINAL PAT, NEW PAT, SOL). (4)
```

FINAL PATは目標とする燃料装荷パターン(2, 3, 9, 3, 3, 5)であり、NEW PATは燃料配置PATからPATH番目のパスに対応する燃料を移動した結果実現される燃料装荷パターンである。SOLはNEW PATからFINAL PATに至る燃料移動を示すパスのリストである。(2)式では、与えられた燃料配置から、移動可能な燃料に対応するパスPATHを選択し、新しい燃料配置NEW PATを求める。(3)式では、NEW PAT

が炉停止余裕およびDBGの制約を満足しているかどうかを評価する。制約が満足されない場合にはバックトラックして(2)式で新たなPATHが選択される。制限条件が満足された場合には(4)式が評価される。実際には再び(1)式でNEW PATをPAT,SOLを(PATH;SOL)に置きかえて評価される。初期の燃料装荷パターン(0, 0, 0, 0, 0)から目標のパターン(2, 3, 9, 3, 3, 5)に至る制約条件を満足する解X(パスのリスト)は

?-state((2, 3, 9, 3, 3, 5.), (0, 0, 0, 0, 0, 0), X).

により得られる。これは燃料の移動回数を最小にするものである。

### 6. その他の応用

前節までに紹介したほかにも、知識工学手法を用いた様々な技術開発が進行中である。プラント事故時に多数のアラームが短い時間の間に発生して運転員の負担が大になる事態に際して判断支援を行うアラームハンドリングシステム<sup>(18)</sup>、空間的制約が厳しい条件下でのプラント配管のルート決定法<sup>(14)</sup>、大規模な計算コードの膨大な入力作成および出力整理支援システム<sup>(19)</sup>など、実用性の高いシステムがすでに開発されている。

アラーム処理は必ずしも知識工学的手法を用いなくても可能と考えられようが、知識工学のツールの利用は作業効率を飛躍的に向上させる。第II-1節でも述べたように、事象因果関係を知識工学的手法で表現すると、在来型計算機言語に比較して扱いが容易になり人的過誤も減少する。処理対象アラーム数を80から200に増加させた時にプログラムの修正は2-man-dayで終了したと報告されているが、このようなことは在来技術では難しいであろう。配管ルート決定は、本質的に在来法では対処が極めて難しく、熟練者のノウハウと試行錯誤的問題解決以外に手段がなかった問題領域である。この種の問題を効果的に処理できる手法が開発できたことは、知識工学のポテンシャルの高さを示すものといえる。計算コードの入出力支援も普通に想像される以上に重要な技術である。多くの入力データのうちのたった1数字のミスタイプが原因で、高価な計算を実行した結果が無意味なものとなってしまった事態は、すべての計算機利用者に共通の体験であろう。計算機の計算能力そのものは極めて高いものになりつつある反面、プログラム開発、実装、検証、運用等の支援技術が未発達で、人間サイドの負担が限界に近づきつつある現在、このような方面への知識工学の応用はもっと積極的に推進されるべきと考える。

ここまで引用した例は、完全に包括的なものでは

なく、引用できなかったものの中にも優れた応用は多く残されている。しかしながら、知識工学が従来は対処が非常に困難であった問題に対する新しい接近法を提供するものであることだけは例証できたのではないかと考える。

## III. 知識表現と推論機構

前章にて幾つかの応用例が紹介されたが、本章では知識表現と推論機構に関して理論的な背景を要約してみる。

### 1. 知識とは

一口に知識といっても対象や概念に関する知識、事象、行為や状態に関する知識、手順に関する知識、知識の使い方に関する知識(メタ知識\*)など、表現しようとする内容により多くのものに分類される。また、経験的な知識、教科書的な知識というように質的な差異で分類することもある。さらに、宣言的知識、手続き的知識など、知識の表現形式によって分類することもある。一般には、1つの表現形式によっても種々の内容の知識が表現可能であるので、以下、表現形式について代表的なものを紹介し、推論機構などの特徴を述べる<sup>(16)(17)</sup>。なお、微分方程式などの数式で記述されるものも知識の1つではあるが、これらは以下に示す知識表現に埋め込むことができる。

### 2. 代表的な知識表現とそれを用いた推論機構

#### (1) 述語論理表現

論理はすべての知識表現の基礎である。述語論理は定数(特定の対象を表現)、変数(特定されていない対象を表現)、関数(対象から対象への写像を表現)、述語(対象やその属性間の関係を表現)、論理結合子(∧: 連言(かつ), ∨: 選言(または), ∼: 否定, →: 含意), 限定子(∀: 全称限定子(すべての), ∃: 存在限定子(ある))を構成要素として関係(真, 偽)を表現するものであり、宣言的知識に分類される。通常使われるものは述語に変数を許さないもので、これを一階述語論理と呼ぶ。述語論理は意味の違いを正確に表現できる。例えば、

\* 例えば、“首相の自宅の電話番号を知っているか”と聞かれた場合、われわれは直ちに“知らない”と答える。それは“そんな有名な人の電話番号を知っていれば、自分が知っているということを自分自身が知っているはずである”と考えて直ちに結論を出すからであって、知っていることを1つ1つ思い出(検索)してその中に入っていないから“知らない”と答えているのではない。このような知識に関する知識をメタ知識といい、それを用いた推論をメタ推論という。



$\forall x \exists y (\text{Loves } x \ y)$  と  $\exists y \forall x (\text{Loves } x \ y)$  では意味が違う(前者では  $y$  は  $x$  に依存するが, 後者では依存しない)。またモジュール性が強く, 推論課程の完全性が保証されている点に特徴がある。しかし, 真, 偽が明確でない知識の表現に難があり, 表現形式が一律で平坦であるため, 大規模な知識を扱おうとすると推論の効率が問題になってくる。

述語論理における推論の基礎は導出原理に基づいている。いま, 論理式を節表現したものを(5),(6)式のように書く((5)と(6)式は  $\wedge$  で結ばれているが, 通常このように分離して書く)。

$$RV_1V_2V_3 \cdots V_m \quad (5)$$

$$\sim RV_1V_2V_3 \cdots V_n \quad (6)$$

$R$ (リテラルという)が真なら(6)式から  $\sim R$  が削除可能,  $R$  が偽なら(5)式から  $R$  が削除可能であるので,  $R$  の真偽にかかわらず(5),(6)式から(7)式を得る。

$$P_1V_1V_2 \cdots V_mV_1V_2V_3 \cdots V_n \quad (7)$$

この課程を導出(Resolution)という。直観的には2つの節に  $R$  と  $R$  の否定( $\sim R$ )がそれぞれあればそれらを消して, 1つの節にまとめることであるが, 一般には  $P, Q, R$  には変数が含まれるので, 変数に適当な値を代入して2つのリテラルが等しくなった場合(この操作をユニフィケーションという)に, このような導出が可能になる。

簡単な例を示す((5'),(6')式から(7')式を導出)。

$$\sim AT(\text{Taro}, x) \vee AT(\text{Pochi}, x) \quad (5')$$

$$AT(\text{Taro}, \text{School}) \quad (6')$$

$$AT(\text{Pochi}, \text{School}) \quad (7')$$

このように推論とは導出の繰り返しである。通常は求めたいものの否定を節集合に加えて, 導出により空節を導き, 矛盾を証明することによって推論を終了する(この課程を反駁導出という)。各時点において, 導出可能なリテラルの組が2つ以上あるときには競合が生じ, このための戦略が必要になる。

PROLOG は一階述語論理のサブセットである肯定リテラルが1つしかないホーン節に表現形式を限定し, 競合解消の戦略として縦型線形探索を採用したものである<sup>(18)</sup>。したがって, PROLOG で表現されるものは(8)式の形のものであり, これは(9)式と等価である。

$$QV \sim P_1V \sim P_2V \cdots V \sim P_m \quad (8)$$

$$P_1 \wedge P_2 \wedge \cdots \wedge P_m \rightarrow Q \quad (9)$$

(8),(9)式が等価なことは  $\sim(P_1 \wedge P_2) = \sim P_1 \vee \sim P_2$  であって,  $P \rightarrow Q$  と  $\sim P \vee Q$  の真理値が  $P, Q$  の真, 偽

のすべての組合せに対して等しいことから理解できる。直観的には  $PV \sim P$  は恒真であり  $P \rightarrow Q$  であるから,  $QV \sim P$  が成立すると考えてもよい。したがって, (5')式を

$$\forall x (AT(\text{Taro}, x) \rightarrow AT(\text{Pochi}, x)) \quad (5'')$$

(ポチは太郎の行くところならどこへでもついて行く。)

と解釈すると(7')式の結論は自明であろう。

先に, 述語論理は宣言的表現と述べたが, これは(8)式を宣言的表現と解釈したものであり, これと等価な(9)式を手続き的表現と解釈してもかまわない。第II-2, -5節の応用例では知識表現言語としてPROLOGを使用し, PROLOG に内蔵されたユニフィケーションとバックトラックの機構を推論に利用している。

## (2) ルール表現

知識をルールの形で表現するのは自然である。古くは, 人間の問題解決課程をシミュレートする機構として Newell(1967)によって提案されたプロダクション・システムに端を発したもので, 現在では, もっともよく使われている知識表現である。ルールはよく「if-then ルール」といわれるように条件部(前提, if 部)と結論部(帰結, then 部)からなり, 推論は三段論法的に実施される。最初的前提からルールの連鎖を探索する方式を前向き推論(データ駆動型推論, ボトムアップ型推論), 仮定した帰結から逆向きに探索する方式を後向き推論(目標駆動型推論, トップダウン型推論)という。ルールに書く内容は厳密な意味での因果関係である必要はない。例えば, MYCIN などの診断ルールは症状が条件部に病気の原因を結論部に書くが, これは因果関係の逆である(原因があってはじめて結果が出る)。推論の方向と因果の方向とは全く別のものであることに注意する必要がある。

ルールは手順を書くのに適している。特に, 各知識が断片的で互いに独立である場合には, ルールの更新や追加も容易である。さらに特定の前提からある結論がなぜ導けるのかを, それらの間の連鎖を表示することにより明確に示すことができる(説明機能)。しかし, 手順でない静的な知識を表現するには適していない。例えば, 「if( $i$ =電流)&(R=抵抗), then(電圧= $i \cdot R$ )」は「電圧=電流 \* 抵抗」という静的な知識の持つ情報の内の電圧を求める際の手順だけを記述したに過ぎない。また述語論理と同じく, 表現が一律であるため構造化の工夫をしないと, 大規模な問題に対しては組合せ的爆発の問題が生ずる。

プロダクション・システムではルールはプロダクシ

ン・メモリー(PM)に、データはワーキング・メモリー(WM)に格納される。推論はインタプリタにより認知-行動サイクル(recognition-action cycle)と呼ばれる手順で一般には前向きに実施される。各ルールの条件部とWMの要素の照合(パタンマッチングという、ユニフィケーションと違いルールの条件部には変数があってもWMの要素には変数は含まれない)によって適用可能なルール集合を見出し、その中から特定の戦略によって1つを選択し(競合解消)、選択されたルールの結論部を実行し、WMの一部を書き換えることを繰り返す。推論の高速化手法として、一度に変化するWMの要素の数は少ないという前提のもとにルールの条件部をあらかじめネットワークに展開しておく(ルールをコンパイルするという)Rete アルゴリズム<sup>(10)</sup>が有名である。

第Ⅱ-2節の応用例ではPROLOGを用いているが、知識はルールの形をしている。ルールはMYCINとは違い正確に因果関係を記述している。推論方式は後向きであるが、通常の後向き推論とは違う。通常の後向き推論では結論を仮定しそれを証明しようとするが、ここでは条件部、結論部ともに成立しており、一番元になる条件部を求めるといって「診断論理」を用いて問題を解いている。第Ⅱ-3節はインタプリタをLISP<sup>(20)</sup>で記述したルールベース・システムである。ルール集合の構造化により推論の効率をあげている。

### (3) フレーム表現

フレーム表現は人間の記憶や認知の課程をモデル化するための枠組みとしてMinsky(1974)によって提案された知識表現方式であり、構造的な知識を表現するのに適している。フレームはある対象を表現し、属性として幾つかのスロットを持つ宣言的知識である。スロットには値や手続きを書くことができる。必要に応じてファセットと呼ばれるサブスロットを持つこともできる。サブスロットとして、if-needed, if-added, if-removed ファセットを設け、その値に手続きを書くことにより、そのスロットが参照されるたびに自動的に指定された手続きが実行される。このような機能をデモンと呼んでいる。また、フレームの特徴の1つは上位フレームの性質を下位フレームに継承させることができる点である。これにより記憶容量が節約される。

フレーム表現における推論の基本は、注目しているスロットの値を確定する(スロットの充足という)ことである。フレーム自身には上述した性質の継承を利用した推論や、スロットに書かれている付加手続きを利

用した推論機能しか内蔵されていないが、プロダクション・システムと組み合わせるなど問題に応じた推論機構を組み込むことができる。

第Ⅱ-4節の応用例ではフレームを知識表現の基本とし、特定のスロットの値にルールを記述し、インタプリタがそれを解釈するような使い方をしている。

### (4) 意味ネットワーク

人間の記憶課程のモデルとしてQuillian(1968)によって提案された意味ネットワークがある。これは知識をグラフのノード(節)とリンク(有向枝)の2項関係で表現するもので、ノードには対象、概念、事象、行為などを、リンクにはノード間の関係を対応させる。関係には部分-全体(part-of)、上位-下位(is-a)の関係があり、これらには推移的な関係がある(例えば、xがyでありyがzであれば、xはzである)。また、上位-下位の間にはフレームと同様性質の継承が行われる。意味ネットワークにおける推論では、リンクをたどる連想的な探索が最も単純なものであるが、多くのシステムで用いられる方法は2つのネットワーク構造間の照合(マッチング)を利用したものである。

意味ネットワークはフレームの基になったものであり、分類的な階層構造を持つ知識を表現するには適している。しかし、論理的な結合関係や時間的に変化する状態の記述には適していない。第Ⅱ-4節の応用例では最上位の知識表現に意味ネットワークを用いている。

### 3. 支援ツール

以上、代表的な知識表現とそれを用いた推論機構とを概説したが、知識工学的手法の特徴は知識と推論が分離している点であるので、ユーザーが1から作らなくても知識を与えればシステムが簡単に構築できるような汎用の知識表現言語(支援ツール)が提供されている。1970年代後半から米国の大学を中心に開発されたもの(EMYCIN, EXPERT, OPS5 など)は、知識表現と推論機構のおのおのに対して単一の方式しかサポートしておらず、これらは第一世代のツールと呼ばれている。一般には1つの問題解決に複数の知識表現が必要になる(自然に表現しようとするれば)ので、これらの要望に応じて複数の知識表現、推論機構をサポートするものができている。1980年前半に出てきたこれらのツール(ART, KEE など)は第二世代のツールと呼ばれている。最近では、国内各社からも種々の支援ツールが提供され始めている。第二世代のツールの中には知識の整合性保持機構を持ったものもある。

これらの支援ツールは使いこなせば便利なものであ

るが、問題のどの部分にどの知識表現を利用し、どの推論機構を使用すべきかについては何も支援してはくれないので、何らかの設計指針が必要である。今後、支援ツールは第三世代、第四世代と進化して行くであろうが、知識獲得、知識検証、より深い推論、大規模知識ベース、高速推論などへの対応へ向けられるであろう。それには具体的なエキスパート・システムを開発して行く上で得られる知見のフィードバックが不可欠である。

#### IV. 知識工学の今後の課題

以上、知識工学の応用例とその背後にある知識表現および推論機構について紹介した。本章ではこの技術を一層実用性の高いものにするために解決されるべき技術課題について述べる。これらの解決は知識工学者のみによっては困難であり、具体的問題に挑戦している各応用分野の専門家の貢献が強く望まれるものである。

##### 1. 理解の深層化

現在の知識工学システムは表層的な知識をつないで問題を解いているに過ぎず、問題理解の程度が浅いといえる。理解を深めるためには、対象とする問題領域の世界モデルを構築し、それを推論に反映させる必要がある。このような観点から第II-4節にも述べたように、「定性的推論」あるいは「常識推論」の試みがなされている。このような深層レベルの知識表現が実現されれば、直接的ルールのような表層知識はコンパイルにより生成できることになり、その意義は大きい。

##### 2. 推論の柔軟性向上

現在の技術では常識が扱えないため、意味的には同じである表現上の違いを理解できず、また知識ベース中の知識から少しでも離れた問題に対しては無力に近い。人間は不完全な知識からでもそれなりに妥当な判断を下すことができる。加えて、新しい情報が与えられた条件下では、一度下した結論を覆す(信念の翻意)こともできる。原子炉診断への応用などを考えれば、これらの機能、いわゆる不完全条件下の推論が扱える技術開発の必要性は明らかであろう。

##### 3. 自己成長能力の実現

人間には経験から学んで知識を増やすとともに、それを体系化する自己組織化能力が備わっている。このようなことは計算機では全く期待できず、それ以前に知識を系統的に収集する手法すら確立していないのが実情である。第1段階としてこのような手法を開発することは必要であるが、知識の源を人間エキスパート

に全面的に依存する方式では、最終的に人間より劣った機能しか実現できない。知識の獲得と組織化の方向への技術開発が望まれる。

##### 4. 記号処理と数値処理のハイブリッド化

知識情報処理といえは記号処理といわれるように、大多数の知識工学応用例では記号処理が主要道具となっている。工学的問題では数値計算が知識工学応用問題解決の主要道具としてもっと利用されることが自然である。このためには低コストで高性能な数値計算専用プロセッサ(いわゆる数値処理エンジン)の開発と、これを記号処理システムと有機的に結合して利用できる計算機環境の実現が必要である。この課題はそう遠くない将来に達成されようが、その折には一層きめ細かい工学的知識情報処理が可能になると考える。

##### 5. 処理速度の向上

記号処理による推論は基本的にはリストのマッチング(ユニフィケーション)であるため処理速度が遅い。プロダクションシステムでルールを前処理するなどの工夫をしても、アルゴリズム的な記述方式に比較すればどうしても時間がかかることになる。この解決には、記号処理系の処理速度向上と並列化が必要であろうが、問題解決手続きの内の可能な部分はアルゴリズム化していくことも第IV-4節と関連して重要な課題であろう。

その他、一般的な課題として、ハードウェア、ソフトウェア両面での信頼性向上、計算機環境のユーザーフレンドリー性向上、特に自然言語理解などが挙げられる。しかし、もちろんこれらの課題が解決されなければ実用的な知識工学応用ができないということではない。現に本稿で引用した応用例中には、実用に供されているものも含まれている。積極的な応用を通じて着実な知見を蓄積し、技術を深めていくことが応用サイドの課題である。

#### V. おわりに

原子力分野への応用が急速に進んでいる知識工学の手法につき、いくつかの応用例を中心に解説した。現在のところまだまだ初期の発展段階にあるにもかかわらず、極めて多種多様な問題に適用されていることから知識工学の持つ可能性の大きさがうかがわれよう。第IV章で論じたような将来的課題の内には短期間での解決は困難なものもあるが、これらの克服を通じて知識工学の有効性は更に増大しよう。従来は経験的知識によってしか対処できなかった面も多い原子力

ステムの設計、構築、運用に新しい強力な手段を提供する知識工学への読者の関心に、本稿が若干でも応えることができたのであれば幸いである。

—参考文献—

- (1) *Proc. Int. Topical Mtg. on Computer Applications for Nuclear Power Plant Operation and Control, Pasco, Washington*, (1985).
- (2) *Proc. Int. Topical Mtg. on Advances in Human Factors in Nuclear Power Systems, Knoxville, Tennessee*, (1986).
- (3) 元田 浩: 原子力誌, 26[4], 271 (1984).
- (4) CLOCKSIN, W.F., MELLISH, C.S.: "Programming in Prolog", (1981), Springer-Verlag.
- (5) 中島秀之: "Prolog", (1983), 産業図書.
- (6) 荻野敬迪, 他: 原子力誌, 25[10], 822~834 (1983).
- (7) 芥川哲雄, 他: 三菱電機技報, 58[6], 5 (1984).
- (8) 峯尾佳幸, 他: 計測自動制御学会第2回知識工学シンポジウム, 131 (1984).
- (9) 横林正雄, 他: 知識工学を用いた原子炉事故診断システムの推論機構; IERIAS, JAERI-M 84-205, (1984).
- (10) YOSHIDA, K., et al.: Development and verification of an accident diagnostic system for nuclear power plant by using a simulator, Paper presented at ANS Mtg. on Computer Applications for Nuclear Power Plant Operation and Control, 1985.
- (11) 鷲尾 隆, 他: 第4回知識工学シンポジウム講演論文集, 75~80, (1986).
- (12) KUIPERS, B.: *Artificial Intelligence*, 24, 169~203 (1984).
- (13) CORSEBERG, D.R.: *Proc. 6th Power Plant Dynamics, Control and Testing Symp., Knoxville*, Vol. 1, 59.01~59.16, (1986).
- (14) 小林康弘, 他: 情報処理学会「知識情報処理シンポジウム」講演予稿集, 47~56 (1985).
- (15) FROGNER, B.: Private communication, (1986).
- (16) NILSSON, N.J.: "Principle of Artificial Intelligence", (1980), Tioga Publ.
- (17) 白井良明, 辻井潤一: "人工知能", 岩波講座 情報科学22, (1982), 岩波書店.
- (18) CLOCKSIN, W.F., MELLISH, C.S.: "Programming in Prolog", (1981), Springer Verlag.
- (19) FORGY, C.L.: Rete; A fast algorithm for the many pattern/many object pattern match problem, *Artificial Intelligence*, 19, 17~33 (1982).
- (20) WINSTON, P.H., HORN, B.K.P.: "Lisp", (1981), Addison Wesley.

## 核融合炉材料

井形直弘編 A5・352頁 定価5500円

トカマク型炉を例に、炉の設計条件から生ずる材料の諸特性から、核融合炉構造材料の合金設計までを系統的に解説。極限条件下の材料設計という観点より、原子炉・高速増殖炉、宇宙用材料等まで応用可能である。

## センサハンドブック

片岡・柴田・高橋・山崎編 B5・1200頁 定価58000円

センサの開発、応用に携わる研究者・技術者およびユーザーのために、その原理・構造の基礎知識と最新動向から、科学技術および産業界における応用システムまで、重要事項をすべて網羅したセンサの決定版。

## NTT技術陣による 情報/先端メディア

### 1. 基礎技術

北原安定監修/小林 登編 B6・208頁 定価1200円

### 2. 基本システム

北原安定監修/小林 登編 B6・224頁 定価1200円

## 素励起による凝縮系物理学

眞隅泰三著 A5・328頁 定価4200円

半導体、イオン結晶・光物性、磁性、超伝導、超流動などの物性を、素励起の概念でとらえようとする「素励起で見る固体物理学」の入門的解説書。式のもつ物理的意味あいや実験データに主眼をおき平明に解説。

〒102 東京都千代田区九段南 4-3-12

培風館

TEL (03)262-5256 振替東京 4-44725