

# 自然言語に近い知識表現を可能にする 概念化学プログラミングの実現\*

友部 博教<sup>†</sup> 木本 協司<sup>‡</sup> 伊庭 斉志<sup>†</sup> 石塚 満<sup>†</sup>

<sup>†</sup> 東京大学工学部電子情報工学科 <sup>‡</sup> ミレーヌコーポレーション

## 1 はじめに

従来、知識を表現する言語として論理が多く用いられている。しかし、論理表現は当初は人間の思考のツールである自然言語表現との密接な関係から生まれたが、近代で数学分野として発展したことから数理的側面が強くなり、形式が重視された表現となっていると言える。具体的には、述語論理で何を述語記号にして表現するかで任意性が生じ、一つの自然言語文章表現が一意な論理表現にはならないという問題がある。今後、自然言語で表された情報や知識の役割は増大するので、これを直接的に知識表現形に変換して操作（推論処理）できる枠組みが求められると思われる。知識共有の観点からも自然言語表現との対応がよい知識表現が必要になる。「概念化学プログラミング」は(有)ミレーヌコーポレーションの木本協司氏が提案した知識表現であり、発明・発見の概念過程は化学分野の架橋反応に類似しているという着想から、概念縮合という推論法が考案されている [1]。これは自然言語表現との対応がよい表現となっている。我々はこの「概念化学プログラミング」の実用化の第一歩として、その処理系の実装（現バージョンは LISP による）を行ったので報告する。

## 2 知識表現法

概念化学プログラミングにおいて、知識は次のような概念関係式によって表現される。

\* Conceptual Chemical Programming enabling a Knowledge Representation Scheme close to Natural Language

Hironori Tomobe<sup>†</sup>, Kyouji Kimoto<sup>‡</sup>, Hitoshi Iba<sup>†</sup>, Mitsuru Ishizuka<sup>†</sup>

<sup>†</sup> University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan  
e-mail: tomobe@miv.t.u-tokyo.ac.jp

「人間は泳ぐ」⇒ 人間 [DO] 泳ぐ

「ソクラテスは人間である」  
⇒ ソクラテス [BG] 人間

「カインはアベルを殺した」  
⇒ カイン [DO\*P] 殺すーアベル

ここでは概念化学プログラミングによる知識表現の詳細を説明する。

### 概念関係式

「概念化学プログラミング」は、2つの概念要素を概念関係で結合した概念関係式（または概念分子と呼ぶ）を基本として知識を表現する。概念関係式は、一般に以下で定義される。

概念要素 [概念関係、真理値] 概念要素

真理値は0（偽）から1（真）の連続値をとるものとし、省略時は1とする。

### 概念要素

概念要素には次のようなものがある。

原子 車 彼 太郎 赤い 5 \$X など

関数 会員 (年齢) 姓名 (姓、名) など

関数 会員 (年齢) 姓名 (姓、名) など

演算 5+3 X\*Y など

概念原子において、\$X などのように先頭に\$のつくものは変数とする。一つの概念関係式内に複数の変数を用いることも可能であり、同じ名前の変数は同じものとみなすが、異なる概念関係式に同じ名前の変数があっても、これらは同じものとはみなさない。

### 概念関係

概念要素同士を結びつける概念関係には次のようなものがある。

[EQ]	恒等関係	[GT] [LT]	大小関係
[BE]	連結関係	[SM]	類似関係
[CD]	対応関係	[BG] [CT]	帰属関係
[CP] [HV]	構成関係	[CS] [RF]	因果関係
[CH] [PF]	生成関係		
[IP]	入力関係	[OP]	出力関係
[AT]	引力関係	[RL]	斥力関係
=	数値置換	[RP]	概念置換
[FG]	画像置換	[DO]	動詞代表

このうち、引数可能なものも存在する。たとえば [GT]、[LT] では差を引数として使用できる。[SM] では、どのような点で類似しているかを引数とすることができる。また、[GT] と [LT]、[BG] と [CT]、[CP] と [HV] などは逆行の関係にあたり、これらを用いることで概念要素の並び替えを行うことができる。

a [BG] b  $\Rightarrow$  b [CT] a

### 論理記号

論理記号として <AD> (積)、<OR> (和)、<IF>~<TH> (仮定)、限定作用素として <SM> ( $\exists$ )、<AL> ( $\forall$ ) も用意されている。なお、否定は真理値を 0 とすることによって記述する。<IF>~<TH> の仮定と \$X などの変数を用いることによって、ルールを記述することもできる。その例を下に挙げる。

<IF>(\$X [BG] 鳥)<AD>(\$X [HV] 長い足)

<TH> \$X [BG] ダチョウ

## 3 推論法

ここでは、「概念化学プログラミング」の推論法について説明する。まず、同一の概念要素を持つ概念分子 (概念関係式) と概念分子を結合させることにより概念縮合分子を生成する。元の概念分子各々が概念関係を含んでいるため、これらから生成された概念縮合分子は複数の概念関係を含むことになる。そこで、概念関係式同士の意味的な積として定義された概念関係積というものをを用い、新しい概念関係を導き出すことによって推論を行う。この仮定を概念縮合と呼ぶ。以下は概念関係積の例である。

[BG]  $\times$  [DO] = [DO]

[LT]  $\times$  [LT] = [LT]

[CP]  $\times$  [CT] = [CP]

[BG]  $\times$  [CT] = [SM]

以下に、知識ベースに「人間は死ぬ」、「ソクラテスは人間である」という知識があり、そこから「ソ

クラテスは死ぬ」という結論を導き出す際の推論の例を挙げる。

概念分子：人間 [DO] 死ぬ、ソクラテス [BG] 人間

概念縮合分子：ソクラテス [BG] 人間 [DO] 死ぬ

概念関係積：[BG]  $\times$  [DO]  $\Rightarrow$  [DO]

結論：ソクラテス [DO] 死ぬ

## 4 LISP による実装

実用化の第一歩として LISP による処理系の実装を行った (CCP Ver.1.0)。詳細は省略するが、論理的なゴールに基づく後向き推論を実行できる。以下には実際のプログラム表現を例示する。

RULE <IF>(\$X [EQ] \$Y の・父)

<AD>(\$Z [EQ] \$Y の・母)

<TH> \$X [EQ] \$Z の・夫;

FACT 太郎 [EQ] 次郎の・父;

FACT 花子 [EQ] 次郎の・母;

QUES \$X [EQ] 花子の・夫;

Ans. \$X = 太郎

## 5 概念化学プログラミングの意義

概念化学プログラミングによる知識表現は、述語論理において述語の数を有限個に制限し、概念関係積と言う形で自然言語に関する意味的知識を内在化させ、引数の数を 2 個に限定し語順を変えた知識表現と考えることができる。この場合、概念関係が述語記号に、2 つの概念要素が引数に相当する。よって論理の完全性を継承し、また表現の述語を概念関係に固定することで、人間にとって理解が比較的容易になるともいえる。その結果複数のユーザによる情報共有や発想支援等が行いやすくなる。問題点として、知識ベース内の情報が肥大化した際の処理が挙げられる。概念化学プログラミングは複数のユーザからの知識入力を想定しており、膨大な知識が記憶されることになる。このようにして得た知識を組み合わせ概念縮合することにより有益な結論を得るため、効果的かつ高速な推論エンジンが必要となる。

## 参考文献

- [1] 木本 協司: “研究開発における発想とその支援ツール、概念化学プログラミング—構造化中間言語—”. 研究開発マネジメント, 1998 年 9 月.
- [2] 石塚 満: “知識の表現と高速推論”. 丸善, 1996.