

## 現実関係モデルを導入した自然語による検索キーの取り扱い

## 4G-2

## - 再利用プログラムベースの構築を目指して -

永沢勇一† 今中 武†† 上原邦昭†† 三根 久† 豊田順一††

†関西大学

††大阪大学産業科学研究所

## 1. はじめに

従来より、プログラム生産性向上のための手法として、プログラム自動合成、再利用の研究が多く行われている[1]。我々は、Prologを対象言語としたプログラム再利用の研究を行っている。特に、プログラム再利用の具体的方法として、再利用プログラムベースの構築を目指している。再利用プログラムベースとは、再利用プログラムをデータベース化し、複数のプログラマが必要なプログラムを自由に登録、検索できるようにするものである。一方、Prologプログラムでは、プログラム中によく似た構造やパターンが多く用いられ、かつプログラムのモジュール性が高いために、プログラムを再利用することにより、生産性向上を図ることが可能である[2, 3]。再利用プログラムベース構築の第1段階として、再利用プログラムの蓄積時における仕様作成を自動化するシステムIOSG/I (Input Output Specification Generator I) を作成した。IOSG/Iを用いれば、再利用プログラム登録のための仕様作成は非常に容易になるものと考えられる。本稿では、再利用プログラムベース構築の第2段階として、再利用プログラムベースに対し、柔軟に検索を行えるようにするための現実関係モデルの利用について述べる。

## 2. IOSG/Iによる入出力仕様を用いた検索

## 2.1 IOSG/Iの概要

IOSG/Iは、Prologの中心的処理であるリスト処理を対象として、プログラムのソースコードから入出力仕様を自動的に生成するシステムである。IOSG/Iでは、①あらかじめ用意した典型的なリスト処理プログラム(以降では、基本プログラムと呼ぶ)と入力されたプログラムとの差異を求め、②その差異に応じて、あらかじめ基本プログラムに与えている入出力仕様を変更することにより、入力されたプログラムの入出力仕様を生成している。IOSG/Iで用いている基本プログラムは、リスト処理プログラムでよく用いられる類似構造を抽出したものである[3]。基本プログラムの例として、入力リスト(第1引数)から条件を満たす要素をすべて削除し、出力リスト(第2引数)に出力するプログラムを図1に示す。また、IOSG/Iでは、入力されたプログラムと基本プログラムの差異を求めるために、基本プログラムに変形操作を行う変形オペレータを

定義している。変形オペレータを次々に適用し、基本プログラムから入力されたプログラムが生成可能であれば、適用した変形オペレータの列がプログラムの差異を表すことになる。プログラムの差異から入出力仕様を生成する処理は、基本プログラムの仕様に対し、各オペレータにあらかじめ対応づけた仕様変換規則を適用することにより行われる。仕様変換規則とは、個々のオペレータの適用によるプログラムの変形が仕様に対してどのような影響を与えるかを定義したものである。たとえば、プログラムのif\_recursive部分を削除すれば、「条件を満たすすべての要素に対して処理を行う」といった基本プログラムの入出力仕様が「条件を満たす最初の要素1つだけに対してのみ処理を行う」といった仕様に変更される。このような変更を規則として表現したものが、仕様変換規則である。

## 基本プログラム

```
io_attribute(predicate(input,output)).
```

```
predicate([],[]).
```

```
predicate([A|B],C):-
```

```
    predicate(B,C). ← if_recursive
```

```
predicate([A|B],[A|C]):-
```

```
    predicate(B,C). ← else_recursive
```

## 基本プログラムの入出力仕様

入力リストに対して、条件を満たす要素すべてを削除し、結果を出力引数に出力する。

図1 基本プログラムとその入出力仕様

## 2.2 再利用プログラム検索における問題点

IOSG/Iの生成した入出力仕様と再利用プログラムを組み合わせることで再利用プログラムベースに蓄積し、再利用プログラムの検索を行う場合を考える。IOSG/Iでは、あらかじめオペレータに対応づけた自然語を組み合わせることによって入出力仕様を生成するために、生成される入出力仕様は、あらかじめ準備した自然語の組み合わせで表される。しかしながら、再利用プログラムの検索者は、IOSG/Iで準備された自然語のみを用いて検索キーを記述するとは限らない。このため、検索者の必要なプログラムが再利用プログラムベースに

Method of Retrieving Reusable Programs by using Commonsense Knowledge  
Yuichi NAGASAWA, Takeshi IMANAKA, Kuniaki UEHARA, Hisashi MINE  
and Jun'ichi TOYODA

KANSAI Univ.

The Institute of Scientific and Industrial Research, OSAKA Univ.

登録されているにも拘らず、登録されている仕様と検索キーの表現が異なるために検索できない可能性がある。また、検索者に対して IOSGI で生成する仕様に準じた検索キーによる記述を求めることは困難である。

たとえば、再利用プログラムベースからリストの要素が整理されているかどうかを確認するプログラムを検索する場合を考える(図2)。このプログラムは1引数からなり、入力されたリストの要素が [6, 4, 3, 1] のように降順に並んでいる場合に成功するが、[6, 3, 4, 1] のように整理されていないリストが入力されると失敗する。利用者が、「入力リストに対して、すべての連続する2つの要素の値が大きい順であるか調査するプログラム」といった検索キーを用いた場合は、このプログラムを直ちに検索できる。しかしながら、同じ内容であっても、「入力リストに対して、降順にソートされているか調査するプログラム」という検索キーによって検索した場合、検索キーと登録されている仕様では、用いられている自然語が異なるために検索できない。

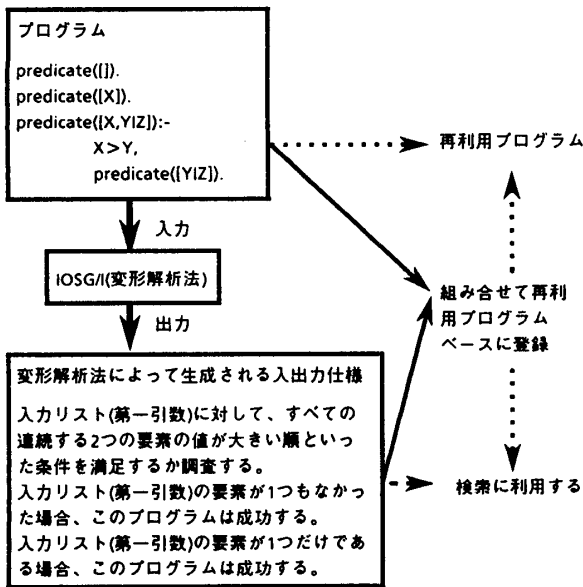


図2 再利用プログラムの登録

### 3. 現実関係モデルを導入した再利用プログラムの検索

#### 3.1 現実関係モデル

我々は、多様な表現で与えられる検索キーを利用可能にする方法として、現実関係モデルを提案する。現実関係モデルとは、現実世界の知識をネットワークによって表現したモデルである[2]。現実関係モデルでは、ノードが事物を表し、アークはその両端の関係を表している。

現実関係モデルで表される知識は、数学的な知見により準備できるもので、プログラムと独立してあらかじめ定義できる。図3は、リスト要素がソート(整列)されている状態を数学的に定義し、ネットワーク表現したものである。

#### 3.2 現実関係モデルを用いた検索

現実関係モデルを用いて、プログラムを検索する手順を示す。まず、検索者が入力した検索キーを用いて検索を行う。このとき、適切なプログラムが検索できれば、検索は終了する。しかしながら、検索できなかった場合は、現実関係モデルを用いて検索キーの表現を変換し、プログラムの再検索を行う。たとえば、図3の現実関係モデルを用いた場合、「ソートする」といった検索キーから、「入力リストに対して、すべての連続する2つの要素の値が大きい順であるか調査する」といった表現の異なる仕様に変換可能であり、「ソートする」といった要求から図2のプログラムの検索が行えるようになる。また、再検索を行っても適切なプログラムが検索できなかった場合は、再び検索キーを変換し、検索処理を繰り返す。現実関係モデルの記述は有限であるために、検索を繰り返し行っているうちに検索キーの変換が不可能になる場合、プログラムの検索処理は断念される。

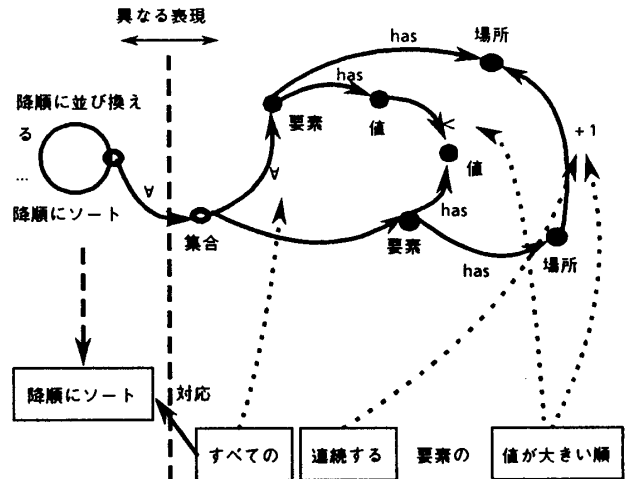


図3 現実関係モデル

#### 4. まとめ

多様な表現で入力される検索キーに対して柔軟な検索を可能とするために、現実関係モデルの導入を提案した。今後は、入出力仕様だけでなくアルゴリズムなどで記述される内部仕様をどのように扱うのが問題になる。内部仕様が扱えるようになれば、再利用プログラム検索時にプログラムの実行効率なども参照可能になる。

#### [参考文献]

- [1] 原田, 篠原: 部品合成によるプログラム自動合成システム ARIES/I, 情処学論, Vol. 27, No4, pp. 417-424 (1986).
- [2] 今中, 上原, 豊田: 類推、帰納の概念を導入したプログラミング知識の学習メカニズム, 情処研報, AI-59-13, Vol. 88, No. 44, pp.113-122 (1988).
- [3] 今中, 上原, 豊田: プログラムの類似性定義のためのネットワーク表現, 第38回情報全大, 7L-3, pp. 1201-1202 (1989).