日本語向け論理型意味表現言語

7N-10

林　隆史　　田中　英彦　　元岡　達

（東京大学　工学部）

1．はじめに

日本語は省略や語順の変化が多く、その意味を論理式で表現しようとする場合、従来の述語論理を用いるのでは不適当な場合が多い。そこで、述語の引数の個数と順序を不定として引数に名前を付けて扱うこととした［1］。このような述語を扱うには、引数を集合と考える必要がある。そこで、集合を項として扱うことが可能な拡張Prolog 作成した。リストとの可換性をもたせるために集合は集合化リストとして表現される。

本稿では、集合化リストを扱うためのProlog の拡張手法と集合化リストを用いたプログラミング手法を紹介する。

2．Prolog の拡張

本稿で述べる拡張Prolog とは、一般のProlog の単一化機能に集合を取扱うための以下の機能を追加したものである。

a）集合化リストの単一化
b）項形式の定義
c）項記述機能［2］

以上の機能の内、集合化リスト導入のために最小限必要な機能は a）だけであり、 b）は集合化リストを利用する上で必要となる記法上の便法を提供するものであり、 c）は述語の実行のタイミングを制御するためのものである。以下、各機能について解説する。

なお、集合化リスト導入によるProlog システムの正当性については、本稿では論じない。

3．集合化リストと非決定的単一化

本拡張Prolog では、Xをリストとするとき｛X｝を集合化リストと呼ぶ。たとえば、リスト形式の項［a,b,c ］を集合として扱いたい場合には、集合化リストの項として、｛［a,b,c ］｝と記述する。

集合化リストの単一化は、その各要素の単一化をすべての組み合せついて行なわなければならない。したがって、集合化リストの単一化を行なう場合には、次のような特異な動作を行なうことになる。つまり、単一化が非決定的となり、単一化に際してもバックトラックを行なわなければならない。

例として、｛［a,b,c ］｝と｛［X｜L］｝の単一化を行なう場合について考えると、変数XとLの値の取り方は、次のように3通りの解が存在することになる。

X = a	X = b	X = c
L = [b,c]	L = [a,c]	L = [a,b]

このように、節の選択だけではなく、単一化においてもバックトラックを行なうという特徴を生かしてプログラムを記述することができる。以下にその例を示す。

```
sort({[X]},[X]).
sort({[A,B|R]},[A,B|R1]) :-
        A=<B,
        sort({[B|R]},[B|R1]).
```

このプログラムは与えられた集合化リストをソーティングしてリストとして返すものである。ただし計算量は指数オーダであり、実用的なプログラムではない。

4．項形式の定義

集合化リストの各要素にタグを付加して用いることにより、C，Pascal 等の構造データのような使い方も可能である。このときには、集合化リストが長くなり、1つの節のなかで何回か同一のものを使うときには、プログラムが非常にわかりにくいものとなってしまう。そこで、そのような項を局所変数と単一化する機能を導入した。この機能はX：＝Tと記述することによって利用される。

実際には、X：＝TとYの単一化を行なう場合、まずXとTの単一化を行なったのち、XとYについて単一化を実行するようになっている。

5．項記述機能

項記述とは項に対して成立すべき制限条件を述語によって記述するものである。ここでは、項Tの制限条件がCであるとき、T：－C　と表現する。

本システムでの項記述の実行には単一化駆動方式

を採用しており、項記述を用いることにより述語の実行タイミングを制御することが可能である。したがって、項記述を用いることにより、集合化リストの単一化における無駄な非決定性を制限し、実行効率を改善することが可能である。

6．状況意味論［３］への応用

本システムは自然言語理解システム作成用に開発されたものである。そこで、状況意味論を用いた簡単な自然言語理解システムに応用してみた。

状況意味論における複合不定項を記述するための言語としてはＣＩＬ［４］があるが、既に述べた機能を用いて、本システムでも複合不定項を記述することができる。さらに、状況を集合化リストを用いて事実の集合として自然に表現することが可能である。例として、［４］で用いられている簡単な発話と対話の状況の記述とその意味解析を行なうプログラムを、本システム用に書き直したものを右に示した。

7．おわりに

集合の概念を集合化リストの形でProlog の項として扱う手法を示した。集合化リストは、集合や構造データの記述を可能とする点で有用であると考えている。

さらに、集合化リストによる関係データベースとのインタフェース、非決定的単一化からの並列性の抽出、オブジェクト主導型言語に見られるプログラムの継承等を実現する方式についても検討を進めている。

また、本システムはＣ－Prolog 上にインタープリタが実装されている。

【参考文献】

［１］林他「知識獲得機能を有する日本語理解システムに関する考察」第30回情処全大５Ｋ-1、(1985)

［２］戸村「ＴＤProlog ：項記述可能なProlog 処理系」Proc. of the Logic Programming Conf.'85 (1985)

［３］Barwise and Perry: Situations and Attitudes, ＭＩＴ Press, (1983)

［４］Mukai : Unification over Complex Indeterminates in Prolog, Proc. of the Logic Programming Conf.'85 (1985)


```
ds(({[speaker(I),hearer(Y),loc(L),exp(E)]},
    {[(L,(speaking,I),1),
      (L,(addressing,Y),1),
      (L,(utter,E),1)]})).

mOpt((Ci:=({[ds(({[exp(E)|_]},DS))]},MO)))
        :- sentence(Ci,E,[]).

sentence(({[ds(DS)|_]},MO),A,B) :-
        noun(({[ds(DS)|_]},Ag),A,A1),
        verb(({[ds(DS),
                agent(Ag),
                object(Obj),
                pol(1)|_]},
              MO),A1,A2),
        noun(({[ds(DS)|_]},Obj),A2,B).

noun(({[ds(({[speaker(I)|_]},DS))]},I),[i|A],A).

noun(({[ds(({[hearer(Y)|_]},DS))]},Y),[you|A],A).

verb(({[ds(({[loc(L)|_]},DS)),
        agent(Ag),object(Obj),pol(Pol)|_]},
      {[(L,(love,Ag,Obj),Pol)|_]}),[love|Z],Z).

dConstr([X],M) :- mOpt(({[ds(X)|_]},M)).
dConstr([X,Y|Z],M) :-
        chRole(X,Y),
        mOpt(({[ds(X)|_]},M)),
        dConstr([Y|Z],M).

chRole(({[speaker(I),hearer(Y),loc(L)|_]},CDS),
        ((Ci:=({[speaker(Y),
                 hearer(I),
                 loc(next(L))|_]},NDS))
         :- ds(Ci))).

INPUT:
?- dConstr(
        [((Ci1:=({[exp([i,love,you]),
                   speaker(jack),
                   hearer(betty),
                   loc(loc01)|_]},DS1))
                  :- ds(Ci1)),
         ((Ci2:=({[exp([i,love,you])|_]},DS2))
                  :- ds(Ci2))],
        MO)),
   print(MO),nl.

OUTPUT:
{[(loc01,(love,jack,betty),1),
  (next(loc01),(love,betty,jack),1)|_4572]}
```