

LangLABにおける
高水準辞書記述言語SRL/O

奥村学, 田中穂積
(東京工業大学 工学部)

1 はじめに

本研究室で開発されたDCKR(Definite Clause Knowledge Representation)[小山 85, 田中 86]は、ホーン節形式をベースにした知識表現形式であり、Structured Object (以後これを単に Objectとよぶ。)を構成する各スロットを、述語 sem をヘッドとする1つのホーン節で表現する。従って、1つの object は、第1引数が等しい sem述語をヘッドとするホーン節(スロット)の集合とみなすことができる。このことから、DCKRで記述した知識に関する推論を行なうプログラムのほぼ全てを、Prologに組み込みの機能で代用できる。

自然言語の意味処理に、このDCKRを応用すると、2で述べるように、辞書項目の柔軟な意味記述が可能になるとともに、それを利用した意味処理用プログラムの中核が、Prologに組み込みの機能で代用可能になる[田中 85, 奥村 86]。

今回、DCKRを辞書記述の機械語レベルの形式として考え、ユーザが辞書項目を記述する辞書記述用の高水準言語を設計した。3では、この辞書記述用言語の第0版SRL/O[奥村 86]について述べる。

最後に、4では、まとめと今後の課題について述べる。

2 DCKRを応用した意味解析

フレームは、意味解析を行なうための辞書記述の形式としてよく用いられてきた表現である。フレームは、スロットの集合から構成されている。一般に、スロットは、スロット名、制約条件、アクションの3つ組から構成される。スロット名は、フィラーが文中で満たすべき統語制約条件を表わす。制約条件には、そのスロットを満たすことができるフレーム(これをフィラーとよぶ。)の意味的性質を記述する。これを以下、意味制約条件とよぶ。

辞書記述としてフレームを用いる意味解析の手順は、およそ次のようである。

- (i) 選択したスロットの統語ならびに意味制約条件をフィラーが満たせば、アクションを起動して終了、さもなければ(ii)へ。
- (ii) 次のスロットがあれば、それを選択して(i)へ、さもなければ(iii)へ。
- (iii) 上位の概念があれば、そのフレームを取り出し(i)へ、さもなければ意味処理失敗として終了。

(i)から(iii)の意味解析の手順を観察すると、

[a] (i)の意味制約条件は論理式で表現することが多く、これはDCKRで容易に表わすことができる

[b] (ii)のスロットの選択には、Prologに組み込みのバックトラック機構が利用できる。

これは、DCKRでは、スロットを1つのホーン節として表現しているからである

```
sem( open , 1 ~ subj : N ~ In ~ Out ) :- nonvar( N ) ,
( sem( N , isa : event ) ;
  sem( N , isa : thingOpen ) ) ,
  addProp( object : N , In , Out ) .
sem( open , 2 ~ subj : N ~ In ~ Out ) :- nonvar( N ) ,
( sem( N , isa : instrument ) ,
  addProp( instrument : N , In , Out ) ;
  sem( N , isa : wind ) ,
  addProp( reason : N , In , Out ) ) .
sem( open , 2 ~ obj : N ~ In ~ Out ) :- nonvar( N ) ,
  sem( N , isa : thingOpen ) ,
  addProp( object : N , In , Out ) .
sem( open , 3 ~ subj : N ~ In ~ Out ) :- nonvar( N ) ,
  sem( N , isa : human ) ,
  addProp( agent : N , In , Out ) .
sem( open , 3 ~ obj : N ~ In ~ Out ) :- nonvar( N ) ,
( sem( N , isa : event ) ;
  sem( N , isa : thingOpen ) ) ,
  addProp( object : N , In , Out ) .
sem( open , 3 ~ with : N ~ In ~ Out ) :- nonvar( N ) ,
( member( sem( V , object : X ) , In ) ->
( sem( X , isa : thingOpen ) ,
  nonvar( N ) , sem( N , isa : instrument ) ,
  addProp( instrument : N , In , Out ) ) ;
  addProp( demon( sem( V , object : X ) ~
    sem( V , 3 ~ with : N ~ TIn ~ TOut ) ) , In , Out ) ) .
```

図2.1 DCKRによる open の辞書記述

"User Level Dictionary Representation Language SRL/O in LangLAB"

Manabu OKUMURA and Hozumi TANAKA

Tokyo Institute of Technology

[c] (iii)の知識の継承は、DCKRのもつ知識継承機構で容易に実現できることがわかる。

以上のことから、意味解析の中核となるプログラムを、Prologの基本計算機構で代用可能なことが読み取れる。図2.1にその実例を示す。

このように、DCKRを用いると、意味解析に必要なプログラムの大部分を、Prologに組み込みの基本計算機構で代用することができる。また、計算効率も、フレームをリスト構造で表現する場合に比べ、改善されることが期待できる。

3 辞書記述用言語SRL/O

図2.1の記述から分かるように、DCKRは、ユーザが記述するにはその記述性において問題があり、ユーザがそのままDCKRで辞書記述するのは困難である。そこで、辞書記述用の高水準言語SRL/Oを設計した。図2.2に open についてSRL/Oで記述した例を示す。この記述がトランスレータにより図2.1の形のDCD節に変換されることになる。このSRL/Oトランスレータは、現在、LangLAB上に試作中である。

4 おわりに

以下に本研究の結論および今後の課題を述べる。

(1) DCKRを機械語レベルの形式として考え、ユーザが辞書項目を記述する辞書記述用の高水準言語SRL/Oを設計した。

今後は、DCKRで記述された知識のうち、まだSRL/Oで記述を試みていない、例外に関するものや、DCKRの[小山85]以後の拡張点[小山86]についても記述できるように、SRL/Oをバージョンアップしていく予定である。

(2) また、このSRL/Oを含めて、辞書の開発環境を整備していく必要があり、その中でも特に、辞書項目の修正、追加に伴う、辞書の保守・管理を行なうプログラムを開発する必要がある。

参考文献

- [小山85]: 小山晴生, 田中穂積, "Definite Clause Knowledge Representation—Prologによる structured object の表現形式と推論—", Proceedings of the Logic Programming Conference '85, 4.3, 1985.

```
open ::
[ subj $ isa:event ;
  isa:thingOpen => object ]
::
[ subj $ isa:instrument => instrument ;
  isa:wind => reason ]
[ obj $ isa:thingOpen => object ]
::
[ subj $ isa:human => agent ]
[ obj $ isa:thingOpen ;
  isa:event => object ]
( with $ isa:instrument
  where
  obj!caller isa:thingOpen => instrument )
( with $ isa:animal => coagent )
::
((at ;
  in) $ isa:place => location ).
```

図2.2 SRL/Oによる open の辞書記述

- [田中86]: 田中穂積, 小山晴生, 奥村学, "知識表現形式DCKRとその応用", コンピュータソフトウェア 論理と自然言語特集号(予定), 1986.
- [田中85]: 田中穂積, 池田光生, 奥村学, "Definite Clause Dictionary — Prologによる辞書項目記述と意味処理", Proceedings of the Logic Programming Conference '85, 12.1, 1985.
- [奥村86]: 奥村学, 小池康晴, 田中穂積, 意味記述用言語SRL/Oの設計とDCKR, 情報処理学会自然言語処理研究会資料, 54.5, 1986.
- [小山86]: 小山晴生, Definite Clause Knowledge Representation—Prologによる structured object の表現形式と推論—, 東京工業大学大学院修士論文, 1986.