知識継承における例外の扱いとその応用について

徳永　健伸，田中　穂積
東京工業大学

1．はじめに

筆者らはこれまで，知識表現形式ＤＣＫＲ（Definite Clause Knowledge Representaion）を提案し，自然言語の意味処理などに応用してきた［田中 86］．ＤＣＫＲでは，概念の各スロットを基本述語 sem をヘッドとするホーン節で表し，知識の継承は isaという述語を介しておこなう．ＤＣＫＲの形式で知識を表現すると推論のためのインタプリタを特別に用意しなくても，Prolog に組み込みの機能で推論機構を代用できるという利点がある．しかし，そのままでは推論過程を Prolog のインタプリタにまかせるため，ユーザが明示的に推論過程を制御できない．知識表現において下位概念は基本的に上位概念のもつ性質を継承することになるが，例外的に継承させてはならない性質が存在する．このような例外を含む意味ネットワークに関する研究として［Fahlman 81］，［Etherington 83］，［Touretzky 86］などがある．特に，Etherington は Default Logic［Reiter 80］を用いてこのような例外を許す意味ネットワークを定式化した．本稿ではこの Etherington の定式化を参考にして継承の例外が取り扱えるようにＤＣＫＲを拡張する．

2．ＤＣＫＲによる知識表現

ＤＣＫＲでは図１のように３引数の基本述語 sem を用いて知識を表現する．sem 述語の第１引数は概念名，第２引数はスロット名とスロット値がオペレータ':'で結合されたもの，第３引数は，継承の履歴を保存するためのスタックである．

```
:- op(100, yfx, ':').
sem(tom, sex:male, _).                    (2-1)
sem(tom, P, S) :-
   isa(eagle, P, [tom|S]).                (2-2)
sem(canary, P, S) :-
   isa(bird, P, [canary|S]).              (2-3)
sem(eagle, P, S) :-
   isa(bird, P, [eagle|S]).               (2-4)
sem(bird, canFly:yes, _).                 (2-5)
```

図１

知識の継承は述語 isa（定義を図２に示す）を介しておこなう．データベースからある概念（例えば tom）に関する知識を引き出すには Prolog のゴール

```
?- sem(tom, Property, _).
```

を実行すればよい．Prolog のユニフィケーションとバックトラックにより，その概念に関する知識が次々に得られる．

```
isa(Upper, P, S) :-
   P = isa:Upper ;
   sem(Upper, P, S).
```

図２　isa 述語の定義

3．継承の例外の扱い

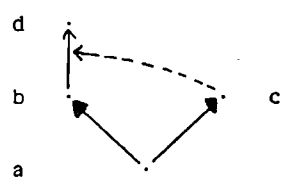
Etherington は［Etherington 83］の中で５種類のリンクを使った継承の例外を含む意味ネットワークを Default Logic を用いて定式化した．本稿では，この Etherington の定式化を参考にして，継承の例外を扱えるようにＤＣＫＲを拡張する．ただし，否定の知識に関しては従来のＤＣＫＲと同様に閉世界仮説を採用することにする．すなわち，［Etherington 83］における，

Strict IS-A　(⟶)
Default IS-A　(⟶)
Exception　(- - - - - - ⟶)

の３つのリンクだけでネットワークを記述する．

4．記述形式

継承の例外を表現するためにＤＣＫＲに exception 述語を新たに導入する．まず例をあげて exception 述語の意味を説明する．




```
sem(a, P, S) :-
   isa(b, P, [a|S]).                      (4-1)
sem(a, P, S) :-
   isa(c, P, [a|S]).                      (4-2)
sem(b, P, S) :-
   exception(b, [c], S),
   isa(d, P, [b|S]).                      (4-3)
```

図３

The handling of exceptions in knowledge inheritance and its applications
Takenobu TOKUNAGA, Hozumi TANAKA
Tokyo Institute of Technology

図３において（4-1）と（4-2）は Strict IS-A リンクに相当し，この継承は常におこなってよいことを示している．これは従来のＤＣＫＲの表現と同じである．（4-3）では「ｂがｄの性質を継承してよいのは，これまでたどってきた経路の概念がｃの性質を継承しない場合である」ということを表現している．exception 述語の第１引数は現在注目している概念，第２引数は継承を阻害する概念（これを例外概念と呼ぶことにする）のリスト，第３引数は継承経路の履歴が保存されているスタックである．exception 述語はこれまでたどってきた経路の概念のいずれかが例外概念の性質を継承するなら失敗し，そうでなければ成功する述語である．図３の概念ａが継承できる性質について考えてみよう．まずａからｂとｃの性質は Strict IS-A リンクにより継承できる．次にｂを経由してｄの性質を継承しようとするがスタックの中にはａが存在し，ａからは例外概念であるｃの性質が継承できるのでａはｄの性質を継承することができない．したがって，ａが継承できるのはｂとｃの性質だけということになる．一方，ｂは例外概念であるｃの性質を継承できないので（4-3）より，ｄの性質を継承することができる．

５．実現

ＤＣＫＲにおける exception 述語の実現は容易である．基本となるアルゴリズムは以下の通りである．

```
exception(node, ExList, Stack) :
for all element e of ExList do
  for all element s but bottom of Stack do
    if e == s then fail fi
    if sem(s, isa:e, node) success then fail fi
  od
od
```

図４　exception 述語のアルゴリズム

外側のループでは exception 述語の第２引数に渡される例外概念をひとつずつ取り出し，内側のループではスタックの要素をひとつずつ取り出して，(1) 継承経路中に例外概念が含まれるか，(2) 継承経路に含まれる概念の中に例外概念の性質を継承するものがあるか，をチェックしている．途中 fail することなく２重ループを抜け出てくると exception 述語は成功する．したがって，exception 述語の効率はスタックの深さと例外概念の数が増えるにしたがって低下する．常識的に「例外」というのは少数であるから exception 述語はほとんどの場合成功するはずである．効率という面からは exception 述語によって例外の記述をおこなうことには問題がある．

６．応用

現在，筆者らは自然言語の意味処理に使用することを前提として概念の上位／下位関係を表すシソーラスを作成中である．そのなかでどうしても継承の例外を表現する必要性を感じている．例えば，図５において「ガソリン」も「食用油」も「油」の下位概念であるが，「食用油」からは「油」の上位概念である「燃料」の性質を継承させたくない，といった場合である．例外として扱わなくても，知識の構造を変えることによって，このような問題は解消できる場合もあるが，現在，作成中のシソーラス程度の規模（最下位の概念数で約１０００）になると，知識の構造を変えることは，他の部分との関連もあり，なかなか困難であるし，例外として扱った方が自然な表現となる場合が多い．

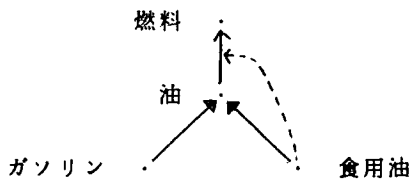


図５

７．まとめ

本稿では知識表現形式ＤＣＫＲを継承の例外が扱えるように機能を拡張した．exception 述語を用いると知識の記述が複雑になるのでより記述しやすい形式を考えることが必要である．現在，ＤＣＫＲの上位水準の言語であるＳＲＬ／０［奥村 86］に例外の記述ができるように拡張することを考えている．また，実際の応用で使用し，効率などの評価をおこなう必要がある．

参考文献

Etherington, D.W. and Reiter, R.,
"On inheritance hierarchies with exceptions", in Proc. AAAI-83, Washington, DC, pp104-108, 1983.

Fahlman, S.E., Touretzky, D.S., Roggen, W.,
"Cancellation in a Parallel Semantic Network", in Proc. IJCAI-81, pp257-263, August, 1981.

奥村学，小池康晴，田中穂積，
"意味記述用言語ＳＲＬ／０の設計とＤＣＫＲ"，情報処理学会情報学基礎研究会資料，1-2，1986.

Reiter, R.,
"A Logic for Default Reasoning", Artificial Intelligence 13, pp81-132, April, 1980.

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，Vol.3, No.4, 1986.

Touretzky, D.S.,
The Mathematics of Inheritance Systems, Morgan Kaufmann Publishers, Inc., Los Altos, California, 1986.