知識獲得　と　ＤＣＧ

北上　始、平川秀樹、古川康一
財）新世代コンピュータ技術開発機構

知識獲得　と　ＤＣＧ

北上　始、平川秀樹、古川康一
財）新世代コンピュータ技術開発機構（ＩＣＯＴ）

［概要］

知識獲得システムを利用して、ＤＣＧ(Definite Clause Grammar) を、合成したので報告する。知識獲得を、知識の同化および調節などによる知識モデルの作成としてとらえ、その応用として、ＤＣＧの問題を考察した。今回は、その考察の中で、ＤＣＧを合成するときに、特有のルールが存在する事を報告する。

１．はじめに

人間のもつ知識を組織的に蓄積、管理する知識ベースメカニズムの一機能として知識獲得の機能がある。知識獲得とは、以下の考察にもとづく知識の同化・調節・発見による知識の収集を意味する。

人間が外界から同化する知識は、観測事実としてのファクトの他に、ルールが挙げられる。ルールを同化する過程は、他者からの知識の伝授と捕える事ができる。

知識ベースに蓄積された知識にもとづく推論結果は、その時点で知識ベースがもっている信念であるととらえる事ができる。

人間の信念は、時として修正されるという側面をもっている。この信念の修正を実施するため、知識ベースに蓄積されている知識が修正される。修正される知識には、めったに修正される事のない知識と、流動的に変化する知識がある。

このように、知識ベースから演繹される信念が外界のモデルに一致しないとき、知識ベースの中に存在する知識が修正される。これは、知識ベースに対する調節と呼ばれている。

信念をささえる知識を修正する方法には、二つの方法が考えらる。一つは、新しい信念を説明できる仮説を生成し、妥当な仮説を既存知識とおきかえする方法であり、他の一つは既存知識を反転する方法である。知識の反転は、現在信じている知識を信じない知識に切り変えたり、信じていなかった知識を、信ずる知識に切り変えたりする操作である。

最後に、知識獲得の過程としての知識の発見には、既存知識からの有益な概念の発見が挙げられる。このレベルの発見には、特定の概念に関する学習や、類推などをともなう有益な概念の発見などがある。

本論文は、以上の議論の近似としてインプリメントされている知識獲得システム[1] を利用し、簡単なＤＣＧ（Definite Clause Grammar)の合成を行ったので、その結果について報告する。以下、知識ベースの知識は、(i) ファクト（個々の事実）、(ii)ルール（一般的な性質）、（iii)Integrity Constraints(以降、ICs と呼ぶ）とする。ICs は、ファクト及びルールに関する統合性制約である。ICs は本システムに固有のものであり、ＤＣＧ合成の効率の向上に利用される。

2．ＤＣＧ (Definite Clause Grammar)

本報告では、問題を単純化するためＤＣＧ[2,3] を合成する問題を、次の様に設定する。

（１）ＤＣＧを合成するために与える正の観測文には、単語辞書に登録されていない単語を含めない。もし、辞書にない単語を含めたいときは、辞書に登録後、使用する。

（２）単語辞書に登録されている単語を、以下の単語に限定する。

```
n ([hermia | X],X).
n ([lysander | X],X).
vi([walks  | X],X).
vt([loves  | X],X).
```

ただし、n, vi, vt は、各々、名詞、自動詞、他動詞とする。

（３）構文s(X,[]) は、単語辞書を格納している知識ベース"dcg" 内に獲得される。

（４）知識ベース“dcg ”に対するICs は、次の形をしており、知識ベース“ic”に獲得することとする。

```
inconsistent ( [insert], s( _ ,_)) :- s(X,[]), exist_variable(X).
```

このルールは、後の実行結果でもわかるが、DCG を合成する際に非常に有効なルールであり、観測事実を与える回数を極めて、少なくするのに役立っている。

このルールは、次の様な意味をもつ。

（＊）ＤＣＧの構文s(_ ,_) を“ dcg”に挿入(insert)する動作が矛盾する条件は、S(X,[]) から演繹される文章 Xの中に変数を要素とする単語が存在することである。

このルールは、一般の文構成要素 Y(_ ,_) に容易に拡張が可能であり、全ての文法獲得時に適用される。

述語“ exist_variable”は、次の様に定義されている。

```
exist_variable([ X | Y]):- var(X),!.
exist_variable([ X | Y]):- !, exist_variable(Y).
exist_variable([]):- fail.
```

（５）ＤＣＧを合成する際に使用される観測事実、ICs 、組込み述語には、誤りがないと仮定する。また観測事実とICs の間には、矛盾がないものと仮定する。

３．実行結果

ＤＣＧを合成する命令は、

```
create(Knowledge-base-name,ICs-name,Concept)
```

で与えられる。createと命名したのは、ＤＣＧの構文を、知識ベース内に新たに作成する意味をもたせるための配慮である＊。
付録の（Example1）は、ICs-nameに相当する部分が［］になっているが、これは、ICs を使用しないでＤＣＧを合成することを意味する。
この方式では、観測文として、以下の７つの事実を与えなければならないことがわかる。

```
(1) s([hermia,walks],[]),true
(2) s([x,y],[]),false
(3) s(X,walks],[]),false
(4) s([hermia,loves,lysander],[]),true
(5) s([x,y,z],[]),false
(6) s([x,y,lysander],[]),false
(7) s([x,loves,lysander],[]),false
```

しかしながら、ＩＣｓを事前に定義して、ＤＣＧを合成する問題を解くと、付録の（Example 2）の実行結果が得られる。これによると、観測文として、以下の２つの事実（正の事実）をあたえるだけで、ＤＣＧが合成されることがわかる。

```
(1) s([hermia,walks],[]),true
(2) s([hermia,loves,lysander],[]),true
```

いずれの例も、最後に retrieve 命令が実行されているが、その命令形式は、次のようになる。これは、獲得された知識にたいする検証に利用される。

```
retrieve(Knowledge-base-name,Goals,Return-list).
```

＊少しでも合成対象の知識が知識ベース内に存在するときは、命令“accommodate ”を利用する。この命令は、既存知識の修正に、利用される。

４．おわりに

知識獲得システムを使った、ＤＣＧの簡単な合成を行ってみたところ、ICs が観測事実をあたえる回数を低減するのに非常に強力に働く事を確認した。また、ICs が、直感的にわかりやすい形式であたえられる事も示した。本研究の意義は、ICs と一般的な仮説生成スキーマを利用して、ＤＣＧを合成した事にある[4] 。

５．今後の課題

本報告では、ＤＣＧの Bottom Up による合成には、良い見通しをあたえているが、Top Downによる合成には、何の解決策もあたえていない。ユーザ定義の述語が、不完全な形をしているときの合成アプローチは、今後の課題である。また、単なるＣＦＧの合成ではなく、引数を含むより実際的な文法の合成も今後の課題である。

６．謝辞

本研究の機会をあたえて下さったＩＣＯＴの研究所長・淵一博氏、および有益な討論に参加していただいたＩＣＯＴ第２研究室・国藤進、宮地泰造、両氏に深謝致します。

7．参考文献

［１］ Kitakami,H.,Kunifuji,S.,Miyachi,T.and Furukawa,K.:
A Methodology for Implementation of A Knowledge Acquisition System,
ICOT Technical Memo. TM-0024 (1983).

［２］長尾真：言語工学，昭晃堂　人工知能シリーズ２　(1983).

［３］中島秀之：Ｐｒｏｌｏｇ，産業図書　コンピュータサイエンス・ライブラリー
(1983).

［４］E.Y.Shapiro:Algorithmic Program Debugging,The MIT press (1982).

8．付録　ユーザによる実行トレース。

```
<< Example 1 >>


% ?- list_all(dcg).

n([hermia|_0],_0)
n([lysander|_0],_0)
vi([walks|_0],_0)
vt([loves|_0],_0)

yes
% ?- list_all(ic).

yes
% ?- create(dcg,[],s(X,Y)).

* Initialize parameter(y/n)? y.

* The Knowledge_base "solutions" was created on Core_Space.

* The Knowledge_base "rejected" was created on Core_Space.

Next fact(sentence,true/false) or end ? s([hermia,walks],[]),true.
Checking fact(s)...
Error: missing solution s([hermia,walks],[]). diagnosing...
Error diagnosed: s([hermia,walks],[]) is uncovered.

Searching for a cover to s([hermia,walks],[])...
Refining: (s(X,Y):-true)
Checking: (s(X,[]):-true)
Found clause: (s(X,[]):-true)
    after searching 4 clauses.
Listing of s(X,Y):
    s(X,[]).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? s([x,y],[]),false.
Checking fact(s)...
Error: wrong solution s([x,y],[]). diagnosing...
Error diagnosed: (s([x,y],[]):-true) is false.
Listing of s(X,Y):

Checking fact(s)...
Error: missing solution s([hermia,walks],[]). diagnosing...
Error diagnosed: s([hermia,walks],[]) is uncovered.

Searching for a cover to s([hermia,walks],[])...
Refining: (s(X,Y):-true)
Checking: (s(X,[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s([X|Y],Z):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s(X,[]):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s(X,Y):-n(X,U))
Refining: (s([X,Y|Z],U):-true)
```

```
Checking: (s([X,Y|Z],Z):-true)
Refuted: (s([x,y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s([X|Y],Z):-vi(Y,V))
Checking: (s([X|Y],Z):-vi(Y,Z))
Found clause: (s([X|Y],Z):-vi(Y,Z))
    after searching 16 clauses.
Listing of s(X,Y):
    (s([X|Y],Z):-vi(Y,Z)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? s([x,walks],[]),false.
Checking fact(s)...
Error: wrong solution s([x,walks],[]). diagnosing...
Error diagnosed: (s([x,walks],[]):-vi([walks],[])) is false.
Listing of s(X,Y):

Checking fact(s)...
Error: missing solution s([hermia,walks],[]). diagnosing...
Error diagnosed: s([hermia,walks],[]) is uncovered.

Searching for a cover to s([hermia,walks],[])...
Refining: (s([X|Y],Z):-vi(Y,V))
Checking: (s([X|Y],Z):-vi(Y,Z))
Refuted: (s([x,y],[]):-vi([y],[]))
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s(X,Y):-n(X,U),n(X,W))
Refining: (s(X,Y):-n(X,U),vi(U,W))
Checking: (s(X,Y):-n(X,U),vi(U,Y))
Found clause: (s(X,Y):-n(X,U),vi(U,Y))
    after searching 10 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ?
                                s([hermia,loves,lysander],[]),true.
Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...

Query: vi([loves,lysander],[])? n.
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s(X,Y):-true)
Checking: (s(X,[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s([X|Y],Z):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s(X,[]):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s(X,Y):-n(X,U))
Refining: (s([X,Y|Z],U):-true)
Checking: (s([X,Y|Z],[]):-true)
```

```
Refuted: (s([x,y],[]):-true)
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s([X|Y],Z):-vt(Y,V))
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([x,y],[]):-true)
Refining: (s(X,Y):-n(X,U),n(X,W))
Refining: (s(X,Y):-n(X,U),vt(U,W))
Refining: (s([X,Y,Z|U],V):-true)
Checking: (s([X,Y,Z|U],U):-true)
Found clause: (s([X,Y,Z|U],U):-true)
    after searching 26 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    s([X,Y,Z|U],U).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? s([x,y,z],[]),false.
Checking fact(s)...
Error: wrong solution s([x,y,z],[]). diagnosing...
Error diagnosed: (s([x,y,z],[]):-true) is false.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).

Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s([X,Y,Z|U],V):-true)
Checking: (s([X,Y,Z|U],U):-true)
Refuted: (s([x,y,z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([x,y,z],[]):-true)
Refining: (s([X,Y|Z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([x,y,z],[]):-true)
Refining: (s([X,Y|Z],U):-n(Z,W))
Checking: (s([X,Y|Z],U):-n(Z,U))
Found clause: (s([X,Y|Z],U):-n(Z,U))
    after searching 7 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s([X,Y|Z],U):-n(Z,U)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? s([x,y,lysander],[]),false.
Checking fact(s)...
Error: wrong solution s([x,y,lysander],[]). diagnosing...
Error diagnosed: (s([x,y,lysander],[]):-n([lysander],[])) is false.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).

Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s([X,Y|Z],U):-n(Z,W))
Checking: (s([X,Y|Z],U):-n(Z,U))
```

```
Refuted: (s([x,y],[]):-n([],[]))
Refining: (s([X,Y|Z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([x,y,z],[]):-true)
Refining: (s([X|Y],Z):-vt(Y,V),n(V,X1))
Checking: (s([X|Y],Z):-vt(Y,V),n(V,Z))
Found clause: (s([X|Y],Z):-vt(Y,V),n(V,Z))
    after searching 7 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s([X|Y],Z):-vt(Y,V),n(V,Z)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? s([x,loves,lysander],[]),false.
Checking fact(s)...
Error: wrong solution s([x,loves,lysander],[]). diagnosing...
Error diagnosed:
  (s([x,loves,lysander],[]):-
   vt([loves,lysander],[lysander]),n([lysander],[])) is false.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).

Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s([X|Y],Z):-vt(Y,V),n(V,X1))
Checking: (s([X|Y],Z):-vt(Y,V),n(V,Z))
Refuted: (s([x,y],[]):-vt([y],X),n(X,[]))
Refining: (s([X|Y],Z):-vt(Y,V),vt(Y,X1))
Refining: (s([X,Y|Z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([x,y,z],[]):-true)
Refining: (s(X,Y):-n(X,U),n(X,W),n(X,Y1))
Refining: (s(X,Y):-n(X,U),n(X,W),vt(U,Y1))
Refining: (s(X,Y):-n(X,U),n(X,W),vt(W,Y1))
Refining: (s(X,Y):-n(X,U),vt(U,W),n(X,Y1))
Refining: (s(X,Y):-n(X,U),vt(U,W),n(W,Y1))
Checking: (s(X,Y):-n(X,U),vt(U,W),n(W,Y))
Found clause: (s(X,Y):-n(X,U),vt(U,W),n(W,Y))
    after searching 28 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s(X,Y):-n(X,U),vt(U,W),n(W,Y)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? end.

* Purge Facts(y/n) ? y.

* The Knowledge_base "solutions" was dropped from Core_Space.

* Purge refuted theory(y/n) ? y.

* The Knowledge_base "refuted_hypothesis" was dropped from Core_Space.

* Purge refinement status(y/n) ? y.

* Eliminate Redundancy ? (y/n) n.

* Construction of the Knowledge "s"
```

```
 has been finished.

yes
% ?- list_all(dcg).

n([hermia|_0],_0)
n([lysander|_0],_0)
vi([walks|_0],_0)
vt([loves|_0],_0)
(s(_0,_1):-n(_0,_2),vi(_2,_1))
(s(_0,_1):-n(_0,_2),vt(_2,_3),n(_3,_1))

yes
% ?- retrieve(dcg,s(X,[])).

[[hermia,loves,hermia]]
[[hermia,loves,lysander]]
[[hermia,walks]]
[[lysander,loves,hermia]]
[[lysander,loves,lysander]]
[[lysander,walks]]

yes
```

<< Example 2 >>

```
% ?- list_all(dcg).

n([hermia|_0],_0)
n([lysander|_0],_0)
vi([walks|_0],_0)
vt([loves|_0],_0)

yes
% ?- insert_knowledge(ic,
% ?- (inconsistent([insert],s(_,_)):- s(X,[]),exist_variable(X)),_).

yes
% ?- list_all(ic).

(inconsistent([insert],s(_0,_1)):- s(_2,[]),exist_variable(_2)).

yes
% ?- create(dcg,ic,s(_,_)).

* Initialize parameter(y/n)? n.

* The Knowledge_base "solutions" was created on Core_Space.

* The Knowledge_base "rejected" was created on Core_Space.

Next fact(sentence,true/false) or end ? s([hermia,walks],[]),true.
Checking fact(s)...
Error: missing solution s([hermia,walks],[]). diagnosing...
Error diagnosed: s([hermia,walks],[]) is uncovered.

Searching for a cover to s([hermia,walks],[])...
Refining: (s(X,Y):-true)
Checking: (s(X,[]):-true)
Found clause: (s(X,[]):-true)
    after searching 4 clauses.
Listing of s(X,Y):
    s(X,[]).

* The Knowledge "(s(_0,[]):-true)"
 is inconsistent with the Integrity_Constraints
 "(inconsistent([insert],s(_0,_1)):- s(_2,[]),exist_variable(_2))".

Checking fact(s)...
Error: missing solution s([hermia,walks],[]). diagnosing...
Error diagnosed: s([hermia,walks],[]) is uncovered.

Searching for a cover to s([hermia,walks],[])...
Refining: (s(X,Y):-true)
Checking: (s(X,[]):-true)
Refuted: (s(X,[]):-true)
Refining: (s([X|Y],Z):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([X|Y],[]):-true)
Refining: (s(X,[]):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([X|Y],[]):-true)
Refining: (s(X,Y):-n(X,U))
Refining: (s([X,Y|Z],U):-true)
Checking: (s([X,Y|Z],Z):-true)
Refuted: (s([X,Y|Z],Z):-true)
```

```
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([X,Y|Z],[]):-true)
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([X,Y|Z],[]):-true)
Refining: (s([X|Y],Z):-vi(Y,V))
Checking: (s([X|Y],Z):-vi(Y,Z))
Found clause: (s([X|Y],Z):-vi(Y,Z))
    after searching 16 clauses.
Listing of s(X,Y):
    (s([X|Y],Z):-vi(Y,Z)).

* The Knowledge "(s([_0|_1],_2):-vi(_1,_2))"
 is inconsistent with the Integrity_Constraints
 "(inconsistent([insert],s(_0,_1)):- s(_2,[]),exist_variable(_2))".

Checking fact(s)...
Error: missing solution s([hermia,walks],[]). diagnosing...
Error diagnosed: s([hermia,walks],[]) is uncovered.

Searching for a cover to s([hermia,walks],[])...
Refining: (s([X|Y],Z):-vi(Y,V))
Checking: (s([X|Y],Z):-vi(Y,Z))
Refuted: (s([X|Y],Z):-vi(Y,Z))
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([X,Y|Z],[]):-true)
Refining: (s(X,Y):-n(X,U),n(X,W))
Refining: (s(X,Y):-n(X,U),vi(U,W))
Checking: (s(X,Y):-n(X,U),vi(U,Y))
Found clause: (s(X,Y):-n(X,U),vi(U,Y))
    after searching 10 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ?
                                    s([hermia,loves,lysander],[]),true.
Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...

Query: vi([loves,lysander],[])? n.
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s(X,Y):-true)
Checking: (s(X,[]):-true)
Refuted: (s(X,[]):-true)
Refining: (s([X|Y],Z):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([X|Y],[]):-true)
Refining: (s(X,[]):-true)
Checking: (s([X|Y],[]):-true)
Refuted: (s([X|Y],[]):-true)
Refining: (s(X,Y):-n(X,U))
Refining: (s([X,Y|Z],U):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([X,Y|Z],[]):-true)
Refining: (s([X|Y],[]):-true)
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([X,Y|Z],[]):-true)
Refining: (s([X|Y],Z):-vt(Y,V))
Refining: (s([X|Y],[]):-true)
```

```
Checking: (s([X,Y|Z],[]):-true)
Refuted: (s([X,Y|Z],[]):-true)
Refining: (s(X,Y):-n(X,U),n(X,W))
Refining: (s(X,Y):-n(X,U),vt(U,W))
Refining: (s([X,Y,Z|U],V):-true)
Checking: (s([X,Y,Z|U],U):-true)
Found clause: (s([X,Y,Z|U],U):-true)
    after searching 26 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    s([X,Y,Z|U],U).

* The Knowledge "(s([_0,_1,_2|_3],_3):-true)"
 is inconsistent with the Integrity_Constraints
 "(inconsistent([insert],s(_0,_1)):- s(_2,[]),exist_variable(_2))".

Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s([X,Y,Z|U],V):-true)
Checking: (s([X,Y,Z|U],U):-true)
Refuted: (s([X,Y,Z|U],U):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([X,Y,Z|U],[]):-true)
Refining: (s([X,Y|Z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([X,Y,Z|U],[]):-true)
Refining: (s([X,Y|Z],U):-n(Z,W))
Checking: (s([X,Y|Z],U):-n(Z,U))
Found clause: (s([X,Y|Z],U):-n(Z,U))
    after searching 7 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s([X,Y|Z],U):-n(Z,U)).

* The Knowledge "(s([_0,_1|_2],_3):-n(_2,_3))"
 is inconsistent with the Integrity_Constraints
 "(inconsistent([insert],s(_0,_1)):- s(_2,[]),exist_variable(_2))".

Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s([X,Y|Z],U):-n(Z,W))
Checking: (s([X,Y|Z],U):-n(Z,U))
Refuted: (s([X,Y|Z],U):-n(Z,U))
Refining: (s([X,Y|Z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([X,Y,Z|U],[]):-true)
Refining: (s([X|Y],Z):-vt(Y,V),n(V,X1))
Checking: (s([X|Y],Z):-vt(Y,V),n(V,Z))
Found clause: (s([X|Y],Z):-vt(Y,V),n(V,Z))
    after searching 7 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s([X|Y],Z):-vt(Y,V),n(V,Z)).

* The Knowledge "(s([_0|_1],_2):-vt(_1,_3),n(_3,_2))"
 is inconsistent with the Integrity_Constraints
 "(inconsistent([insert],s(_0,_1)):- s(_2,[]),exist_variable(_2))".
```

```
Checking fact(s)...
Error: missing solution s([hermia,loves,lysander],[]). diagnosing...
Error diagnosed: s([hermia,loves,lysander],[]) is uncovered.

Searching for a cover to s([hermia,loves,lysander],[])...
Refining: (s([X|Y],Z):-vt(Y,V),n(V,X1))
Checking: (s([X|Y],Z):-vt(Y,V),n(V,Z))
Refuted: (s([X|Y],Z):-vt(Y,V),n(V,Z))
Refining: (s([X|Y],Z):-vt(Y,V),vt(Y,X1))
Refining: (s([X,Y|Z],[]):-true)
Checking: (s([X,Y,Z|U],[]):-true)
Refuted: (s([X,Y,Z|U],[]):-true)
Refining: (s(X,Y):-n(X,U),n(X,W),n(X,Y1))
Refining: (s(X,Y):-n(X,U),n(X,W),vt(U,Y1))
Refining: (s(X,Y):-n(X,U),n(X,W),vt(W,Y1))
Refining: (s(X,Y):-n(X,U),vt(U,W),n(X,Y1))
Refining: (s(X,Y):-n(X,U),vt(U,W),n(W,Y1))
Checking: (s(X,Y):-n(X,U),vt(U,W),n(W,Y))
Found clause: (s(X,Y):-n(X,U),vt(U,W),n(W,Y))
    after searching 28 clauses.
Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s(X,Y):-n(X,U),vt(U,W),n(W,Y)).

Checking fact(s)...no error found.

Next fact(sentence,true/false) or end ? end.

* Purge Facts(y/n) ? y.

* The Knowledge_base "solutions" was dropped from Core_Space.

* Purge refuted theory(y/n) ? y.

* The Knowledge_base "refuted_hypothesis" was dropped from Core_Space.

* Purge refinement status(y/n) ? y.

* Eliminate Redundancy ? (y/n) n.

* Construction of the Knowledge "s"
 has been finished.

yes
% ?- list(dcg,s(_,_)).

Listing of s(X,Y):
    (s(X,Y):-n(X,U),vi(U,Y)).
    (s(X,Y):-n(X,U),vt(U,W),n(W,Y)).

yes
% ?- retrieve(dcg,s(X,[]),[X]).

[[hermia,loves,hermia]]
[[hermia,loves,lysander]]
[[hermia,walks]]
[[lysander,loves,hermia]]
[[lysander,loves,lysander]]
[[lysander,walks]]

yes
```