

LCTG—ASSIGNMENT 1
Number Agreement in a DCG
due Oct 12th 2007, in class

1 Introduction

Number agreement in English is a simple example of a more general phenomenon whereby functors like verbs have to agree with arguments like noun-phrases on one or more features, such as gender and case, as well as number. It describes the property that makes the following ill- or well- formed:

- (1) a. The boys like the girl.
- b. *The boys likes the girl.
- c. The girl likes the boys.
- d. *The girl like the boys.

Agreement was argued in the early days of TG to call for a transformational interpretation. But in fact it can be handled locally, using feature unification in a DCG or ATN. The homework asks you to show how to do this.

2 Syntactic structure-building DCG in Prolog DCG notation

Here is the DCG that was discussed in class—it is also downloadable from the class page. It fails to obey number agreement.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%                                                                    %%  
%%   Syntactic structure-building DCG in DCG notation                %%  
%%                                                                    %%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%%%% Stuff to overcome Sictus' ever-so-helpful abbreviation of list terms  
  
portray(Term) :- is_list(Term), write(Term).  
  
is_list([]).
```

```

is_list([_|_]).

%%% End of stuff

s([s,[NP, VP]]) --> np(NP), vp(VP).

np([np,[PN]]) --> pn(PN).

np([np,[Det, N, Rel]]) -->
    det(Det), n(N), optrel(Rel).

vp([vp,[TV, NP]]) -->
    tv(TV),
    np(NP).

vp([vp,[IV]]) -->
    iv(IV).

optrel([rel,[epsilon]]) --> [].

optrel([rel,[that, VP]]) -->
    [that], vp(VP).

%pn(harry) --> [harry].
%% etc. -- very tedious, therefore ....

pn(Word) --> [Word], {pn(Word)}.
pn(harry).
pn(barry).
pn(they).

iv(Word) --> [Word], {iv(Word)}.
iv(walks).
iv(walk).

tv(Word) --> [Word], {tv(Word)}.
tv(sees).

```

tv(see).

det(Word) --> [Word], {det(Word)}.

det(a).

det(some).

n(Word) --> [Word], {n(Word)}.

n(frog).

n(frogs).

%| ?- ['hw1.p'].

%% consulting /amd/nfs/pegasus/disk/ptn053/steedman/lectures/tl/hw1.p...

%% consulted /amd/nfs/pegasus/disk/ptn053/steedman/lectures/tl/hw1.p in module

%yes

%| ?- s(T, [harry, walks], []).

%

%T = [s, [[np, [harry]], [vp, [walks]]]] ?

%

%yes

%| ?- s(T, [harry, walk], []).

%

%T = [s, [[np, [harry]], [vp, [walk]]]] ? % Note the overgeneralisation !

%

%yes

%| ?- s(T, [harry, sees, a, frog, that, walks], []).

%

%T = [s, [[np, [harry]], [vp, [sees, [np, [a, frog, [rel, [that, [vp, [walks]]]]]]]]]] ?

%

%yes

%| ?- s(T, [harry, sees, a, frog, that, walk], []).

%

%T = [s, [[np, [harry]], [vp, [sees, [np, [a, frog, [rel, [that, [vp, [walk]]]]]]]]]] ?

%

%yes

%| ?-

%%%Running s in (-,+,+) mode:

```

%| ?- s([s, [[np, [harry]], [vp, [sees, [np, [a, frog, [rel, [that, [vp, [walks]]]]]]]]]]],
%L = [harry, sees, a, frog, that, walks] ?
%yes
%| ?- s([s, [[np, [a, frog, [rel, [epsilon]]]], [vp, [sees, [np, [harry]]]]]]], L, []).
%L = [a, frog, sees, harry] ?
%yes
%| ?-

```

3 What you have to do

Change the DCG (including the lexicon) so that it only accepts those sentences in the fragment that obey number agreement, including the following:

```

harry walks
*harry walk
some frog walks
*some frog walk
some frogs walk
*some frogs walks
some salmon walk
some salmon walks
some frog that shows harry some salmon walks
*some frog that show harry some salmon walk

```

(Don't eliminate the tree-building apparatus. It will help you to see what is going on.) *Hint:* If the noun *frog* can be represented by a list structure `[n, [frog]]`, then the fact that it is a *singular* noun can be captured in a similar structure `[n(singular), [frog]]`. Prolog variables and the technique of building *Partial Execution* into your DCG rules via judicious use of variables will do the rest.

Hand in a listing of all of the code, including comments and results. Make sure you show whether or not your parses are the only parse, or whether there are duplicates, by failing answers with the “;” prompt. Late homeworks *must* be arranged in advance with a definite date, and be handed in to the TA.