

Dec 31, 20 19:26

NLP.pro

Page 1/1

```

/*
  Michael E. Sparks, 11-30-20

  Demo for using DCG syntax in Prolog to implement a simple grammar.

  SAMPLE USAGE:

  ?- s(Number,ParseTree,[these, dogs, and, cats, smell, some, scrapple],[ ]).
  Number = plural,
  ParseTree = sentence(noun_phrase(determiner(these), noun(dogs, and, cats)), ve
rb_phrase(transitive_verb(smell), noun_phrase(determiner(some), noun(scrapple))
) ;
  false.

  ?- s(Number,ParseTree,[the, cat, stinks],[ ]).
  Number = singular,
  ParseTree = sentence(noun_phrase(determiner(the), noun(cat)), verb_phrase(intr
ansitive_verb(stinks))) ;
  false.

  ?- s(Number,ParseTree,[a, dog, bites, the, cat],[ ]).
  Number = singular,
  ParseTree = sentence(noun_phrase(determiner(a), noun(dog)), verb_phrase(transi
tive_verb(bites), noun_phrase(determiner(the), noun(cat)))) ;
  false.
*/

s(Num,sentence(NP,VP)) --> np(Num,NP), vp(Num,VP).
np(Num,noun_phrase(Det,Noun)) --> d(Num,Det), n(Num,Noun).
vp(Num,verb_phrase(Verb,NP)) --> transv(Num,Verb), np(_,NP).
vp(Num,verb_phrase(Verb)) --> intransv(Num,Verb).
d(singular,determiner(a)) --> [a].
d(singular,determiner(the)) --> [the].
d(plural,determiner(these)) --> [these].
d(plural,determiner(those)) --> [those].
d(unspecified,determiner(some)) --> [some].
n(singular,noun(dog)) --> [dog].
n(singular,noun(cat)) --> [cat].
n(plural,noun(dogs)) --> [dogs].
n(plural,noun(cats)) --> [cats].
n(plural,noun(dogs, and, cats)) --> [dogs, and, cats].
n(unspecified,noun(scrapple)) --> [scrapple].
transv(singular,transitive_verb(bites)) --> [bites].
transv(singular,transitive_verb(smells)) --> [smells].
transv(plural,transitive_verb(bite)) --> [bite].
transv(plural,transitive_verb(smell)) --> [smell].
intransv(plural,intransitive_verb(stink)) --> [stink].
intransv(singular,intransitive_verb(stinks)) --> [stinks].

```