Resource Outline for Lecture-2: Intermediate Prolog and Language Processing

Mir Imtiaz Mostafiz

1 Reading

LPNchapter2.pdf Learn about Prolog unifications and proof search. Unification will help you writing list codes and understanding how Prolog works. Read proof search so that you can understand how a syntax checker works.

LPNchapter3.pdf Explore how recursion works. Especially have a look at different versions of descendant predicate to understand the perils of left recursion.

LPNchapter4.pdf One of the most important chapters. Read and study it thoroughly. Practice and tweak the codes.

LPNchapter5.pdf Read after page 21 to grasp new predicates about lists and how they work.

LPNchapter6.pdf One of the most important chapters. Read and study it thoroughly to learn list predicates. They will be used in the assignment.

LPNchapter7.pdf Most important chapter for the assignment. Learn about Context Free Grammer and Definite Clause Grammar. Read thoroughly about difference lists.

2 Sample CFG and DCG Code

In next subsections, I have wrote a simple grammar and developed both the CFG and DCG code. Carefully analyze and practice it. Also, see the use of unifications in append/3 predicate of listing 1.

2.1 Example Grammar

1. Sentence -> Noun Phrase + Verb Phrase
2. Noun Phrase -> Noun
3. Noun Phrase -> Determinant + Noun
4. Determinant -> a,the
5. Noun -> woman, man

6. verb phrase:- future verb + root verb

7. verb phrase:- 3rd person verb

8. verb phrase:- auxiliary verb + continuous verb

9. future verb -> will, shall

10. root verb -> kill, cry

11. 3rd person verb -> kills, cries

12. auxiliary verb:- ”is”

13. continuous verb:- killing, crying

2.2 Context Free Grammar Code

Listing 1: A sample context free grammar with Prolog

```
%List Predicates
member(X,[X|_]).
member(X,[_|T]):-member(X,T).
append([],L,L).
append(A,B,C):- [H|T] = A, [H|T1]= C, append(T,B,T1).

%Grammar rules

isDeterminant(X):- member(X,[a,the]).
isNoun(X):- member(X,[woman,man]).
isFutureVerb(X):- member(X,[will,shall]).
isRootVerb(X):- member(X,[kill,cry]).
is3rdPersonVerb(X):- member(X,[kills,cries]).
isAuxiliaryVerb(X):- member(X,["is"]).
isContinuousVerb(X):-member(X,[killing,crying]).
isNounPhrase([H]):- isNoun(H).
isNounPhrase([H1,H2]):- isDeterminant(H1), isNoun(H2).
isVerbPhrase([H,T]):- isFutureVerb(H),isRootVerb(T).
isVerbPhrase([H,T]):- isAuxiliaryVerb(H),isContinuousVerb(T).
isVerbPhrase([H]):- is3rdPersonVerb(H).
isSentence(L):- isNounPhrase(A),isVerbPhrase(B),append(A,B,L).
```

2.3 Definite Clause Grammar Code

Listing 2: A sample definite clause grammar with Prolog

```
s --> np, vp.
np --> n.
np --> det,n.
vp --> fv,rv.
vp --> tpv.
vp --> av,cv.
det --> [a];[the].
n --> [woman];[man].
fv --> [will];[shall].
rv --> [kill];[cry].
tpv --> [kills];[cries].
av --> ["is"].
cv --> [killing];[crying].

%Wrapper predicate

isSentence(X):- s(X,[]).
```

2.4 Result

In both cases, running the query $isSentence(X)$ gives us the language consisting of the sentences generated or recognized by the given grammar in table 1.

X
”[woman,will,kill]”
”[woman,will,cry]”
”[woman,shall,kill]”
”[woman,shall,cry]”
”[woman,””is””,killing]”
”[woman,””is””,crying]”
”[woman,kills]”
”[woman,cries]”
”[man,will,kill]”
”[man,will,cry]”
”[man,shall,kill]”
”[man,shall,cry]”
”[man,””is””,killing]”
”[man,””is””,crying]”
”[man,kills]”
”[man,cries]”
”[a,woman,will,kill]”
”[a,woman,will,cry]”
”[a,woman,shall,kill]”
”[a,woman,shall,cry]”
”[a,woman,””is””,killing]”
”[a,woman,””is””,crying]”
”[a,woman,kills]”
”[a,woman,cries]”
”[a,man,will,kill]”
”[a,man,will,cry]”
”[a,man,shall,kill]”
”[a,man,shall,cry]”
”[a,man,””is””,killing]”
”[a,man,””is””,crying]”
”[a,man,kills]”
”[a,man,cries]”
”[the,woman,will,kill]”
”[the,woman,will,cry]”
”[the,woman,shall,kill]”
”[the,woman,shall,cry]”
”[the,woman,””is””,killing]”
”[the,woman,””is””,crying]”
”[the,woman,kills]”
”[the,woman,cries]”
”[the,man,will,kill]”
”[the,man,will,cry]”
”[the,man,shall,kill]”
”[the,man,shall,cry]”
”[the,man,””is””,killing]”
”[the,man,””is””,crying]”
”[the,man,kills]”
”[the,man,cries]”

Table 1: Language Generated by the example grammar