

# Mayank Kataruka - 1217200413

## Program 1 - DCG Sentence

```
sentence --> noun_phrase,verb_phrase.
noun_phrase --> article,noun.
article --> [a].
article --> [the].
noun --> [girl].
noun --> [dog].
verb_phrase --> verb,noun_phrase.
verb --> [sees].
verb --> [pets].

?- sentence([the,girl,pets,a,dog],[ ]).
true.
?- sentence([the,girl,pets,the,dog],[ ]).
true.
?- sentence([the,girl,pets,an,dog],[ ]).
false.
```

## Program 2 - DCG Sentence Parse Tree

```
sentence_tree(sentence(X,Y)) --> noun_phrase_tree(X),verb_phrase_tree(Y).
noun_phrase_tree(noun_phrase(X,Y)) --> article_tree(X),noun_tree(Y).
article_tree(article(a)) --> [a].
article_tree(article(the)) --> [the].
noun_tree(noun(girl)) --> [girl].
noun_tree(noun(dog)) --> [dog].
verb_phrase_tree(verb_phrase(X,Y)) --> verb_tree(X),noun_phrase_tree(Y).
verb_tree(verb(sees)) --> [sees].
verb_tree(verb(pets)) --> [pets].

?- sentence_tree(Y,[a, girl, pets, the, dog],[ ]).
Y = sentence(noun_phrase(article(a), noun(girl)), verb_phrase(verb(pets),
noun_phrase(article(the), noun(dog)))).

?- sentence_tree(Y,[the, girl, pets, the, dog],[ ]).
Y = sentence(noun_phrase(article(the), noun(girl)), verb_phrase(verb(pets),
noun_phrase(article(the), noun(dog)))).
```

## Program 3 - DCG Expression

```
%% Not following associative rule because only plus and multiplication are given in the grammar
which is commutative, but does follow precedence because brackets will be evaluated first then
multiplication and then addition.

expression --> t, [+], expression | t.
t --> n, [*], t | n.
n --> ['('],expression,[')'] | number.
number --> digit,number | digit.
digit --> [1].
digit --> [2].
digit --> [3].
digit --> [4].
```

```
digit --> [5].  
digit --> [6].  
digit --> [7].  
digit --> [8].  
digit --> [9].  
digit --> [0].
```

```
?- expression(['(',2,+,3,')'],[]).  
true ;  
?- expression(['(', '(',2,+,3,')',+,'(',2,*,3,')',')'],[]).  
true .  
?- expression(['(',2,1,+,3,1,')'],[]).  
true ;
```