DCG (Definite Clause Grammar - 1/2)

Reminder about Context-Free Grammars

A context-free grammar is a tuple (Vn, Vt, R, A) where:

- Vn is a finite set of nonterminals
- Vt is a finite set of terminals
- R is a set of rewriting rules X -> Y where X is an element of Vn and Y is a sequence of elements of Vn and Vt
- A is the axiom of the grammar, it is an element of Vn

For example, the grammar G below can be used to generate the senteces made up of n times the symbol a followed by n times the symbol b (n >= 1):

```
S -> a b
S -> a S b
```

Here we have:

```
Vn = {S}
Vt = {a, b}
R = the rules above
A = S
```

Using this grammar we can generate some sentences. Here are three examples:

```
a b
a a b b
a a a b b b
```

DCG (Definite Clause Grammar)

As a first approximation, a DCG is a context free grammar written in Prolog. A DCG is defined by a set of rules. A DCG rule has the form:

```
Head --> Body.
```

where Head is a nonterminal and Body is a sequence of terminals or nonterminals. Body may be empty, it will be noted []. As an example of a DCG, consider the context free grammar above. We can write it as a DCG in this way:

```
s --> [a], [b].
s --> [a], s, [b].
```

So terminals are just lists containing one term and nonterminals are just literals.
Take care, we use --> instead of :-

Then we can ask if some sentences (list of words) can be generated from the axiom of the grammar using this grammar (that is, if the sentence is syntactically correct) or we can ask for all possible sentences that can be generated by the DCG. To do so we must use the built-in predicate phrase/2:

```
?- phrase(s,[a, a, a, b, b, b]).
true

?- phrase(s,[a, a, b, b, b]).
false

?- phrase(s,L).
L = [a, b]
L = [a, a, b, b]
L = [a, a, a, b, b, b]
...
```

Exercise 1

Write the following DCG:

```
s --> [a, b].
s --> [a], s, [b].
```

load it to your Prolog interpreter (using consult) and run the following goals:

```
?- phrase(s,[a, a, a, b, b, b]).
?- phrase(s,[a, a, b, b, b]).
?- phrase(s,L).
```

Exercise 2

Consider the following context free grammar:

```
sentence -> nounphrase verbalphrase
nounphrase -> determiner possibleadjective noun
```

```
possibleadjective -> ø
possibleadjective -> adjective
verbalphrase -> verb, nounphrase
determiner -> a
determiner -> the
noun -> cat
noun -> dog
adjective -> little
adjective -> big
verb -> eats
```

here we have:

```
Vn = {sentence, nounphrase, possbleadjective, verbalphrase, determiner, noun, adjective, verb}
Vt  {a, the, cat, dog, little, big, eats}
R = the rules above
A = sentence
```

1. Convert this grammar to a Prolog DCG.
2. Using the built-in predicate phrase/2:
    - prove that [a, little, cat, eats, the, dog] is a syntacticaly correct sentence
    - prove that [a, little, cat, hurts, the, dog] is a syntacticaly incorrect sentence
    - prove that [a, little, cat, eats, dog] is a syntacticaly incorrect sentence
    - generate all the possible sentences that are syntacticaly correct

Exercise 3

Consider the following context free grammar of the arithmetic expressions on integers 0, 1, 2 and 3:

```
exp -> exp1 '+' exp
exp -> exp1 '-' exp
exp -> exp1
exp1 -> exp2 '*' exp1
exp1 -> exp2 '/' exp1
exp1 -> exp2
exp2 -> '(' exp ')'
exp2 -> '0'
exp2 -> '1'
exp2 -> '2'
exp2 -> '3'
```

here we have

```
Vn = {exp, exp1, exp2, exp3}
Vt = {'+', '-', '*', '/', '(', ')', '0', '1', '2', '3'}
R = the rules above
A = exp
```

1. Convert this grammar to a Prolog DCG.
2. Using the built-in predicate phrase/2:
    - prove that ['1', '*', '(', '3', '+', '2', ')'] is syntacticaly correct
    - prove that ['1', '*', '3', '+', '2', ')'] is not syntacticaly correct
    - try to generate some syntacticaly correct sentences. What is the problem?