1. Write a DCG that accepts strings of the form \(uv\) where \(u\) and \(v\) are strings over the alphabet \(\{0, 1\}\) such that the number of 0's in \(u\) is twice the number of 1's in \(v\). For example,

\[
| \text{?- } s([0,1,0,1,2,0,0,1,0], L).
\]
\[
L = [];
\]
\[
L = [0];
\]
\[
\text{no}
\]

2. Exercise 6.6 in Learn Prolog Now describes a street with

\(\ast\) three neighbouring houses that all have a different colour, namely red, blue, and green. People of different nationalities live in the different houses and they all have a different pet.

Leaving out all the other constraints mentioned in that exercise, write a DCG that outputs strings

\[
[\text{Col1, Nat1, Pet1, Col2, Nat2, Pet2, Col3, Nat3, Pet3}]
\]
satisfying \(\ast\), where the nationalities are english, spanish, japanese and the pets are jaguar, snail, zebra. For example,

\[
| \text{?- } s([\text{red, english, snail, blue, japanese, jaguar, green, spanish, Z}], []).
\]
\[
Z = \text{zebra};
\]
\[
\text{no}
\]

3. Write a DCG that given a non-negative integer \(\text{Sum}\), accepts lists of integers \(\geq 1\) that add up to \(\text{Sum}\). For example,

\[
| \text{?- } s(3, L, []).
\]
\[
L = [3];
\]
\[
L = [2, 1];
\]
\[
L = [1, 2];
\]
\[
L = [1, 1, 1];
\]
\[
\text{no}
\]

It may be useful to write a predicate \texttt{mkList}(\(+\text{Num}, \text{?List}\)) that returns a list \texttt{List} of integers from \texttt{Num} down to 1. For example,

\[
| \text{?- } \text{mkList}(3, L).
\]
\[
L = [3, 2, 1];
\]
\[
\text{no}
\]

Be sure you understand how the DCG clauses translate to ordinary Prolog clauses with difference lists.

---

\(^1\)Due Nov 20 (Tuesday): submit to Blackboard and, if possible, demonstrate during lab (Mon 4-5, Tues 2-3). For any extensions beyond Nov 20, email your demonstrator, David Woods (dwoods@tcd.ie).