1. Write a DCG that accepts strings of the form \( u2v \) where \( u \) and \( v \) are strings over the alphabet \( \{0,1\} \) such that the number of 0’s in \( u \) is the same as the number of 1’s in \( v \). For example,

\[
| \leftarrow s([0,1,1,2,0,0,1,0],L).
L = []; \\
L = [0]; \\
no
\]

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

\((*)\) 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

\[[\text{Col}_1,\text{Nat}_1,\text{Pet}_1, \text{Col}_2,\text{Nat}_2,\text{Pet}_2, \text{Col}_3,\text{Nat}_3,\text{Pet}_3]\]
satisfying \((*)\), where the nationalities are \text{english}, \text{spanish}, \text{japanese} and the pets are \text{jaguar}, \text{snail}, \text{zebra}. For example,

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

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,

\[
| \leftarrow s(3,L,[]). \\
L = [3]; \\
L = [2,1]; \\
L = [1,2]; \\
L = [1,1,1]; \\
no
\]

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

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

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

---

\(^1\)Submit to Blackboard by Tuesday, Dec 5. For any extensions beyond that date, email your demonstrator, David Woods, dwoods@tcd.ie.