Problem 1 Write a DCG that accepts strings of the form $u2v$ where $u$ and $v$ are strings over the alphabet $\{0, 1\}$ such that $u$ is $v$ in reverse. For example,

?- s([0,1,1,2|L],[]).
L = [1,1,0] ;
false

Problem 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

$[h(Col1,Nat1,Pet1), h(Col2,Nat2,Pet2), h(Col3,Nat3,Pet3)]$

satisfying (*), where the nationalities are

english, spanish, japanese

and the pets are

jaguar, snail, zebra.

To avoid confusion with the first problem, use different binary predicates for the difference lists, and, in particular, nbd/2 for the 3 houses. For example,

?- nbd([h(red,english,snail), h(blue,japanese,jaguar),
        h(green,spanish,Z)], []).
Z = zebra ;
false.

Problem 3 The regular expression

$(0 + 1)^*1(0 + 1)(0 + 1)$

denotes the set

$L_3 := \{s \in \{0, 1\}^* \mid s \text{ has length } \geq 3 \text{ and its third to the last bit is } 1\}$
of bitstrings that end with one of the four strings 100, 101, 110, 111 from $1(0 + 1)(0 + 1)$. Recall from lecture that the predicate accept/1 defined below is true of strings accepted by a finite automaton with transitions given by tran/3, final states given by final/1 and initial state q0.
accept(L) :- steps(q0,L,F), final(F).
steps(Q,[],Q).
steps(Q,[H|T],Q2) :- tran(Q,H,Qn), steps(Qn,T,Q2).

Define the predicates \textbf{tran} and \textbf{final} to accept precisely the strings in $L_3$ so that, for example,

?- accept([0,0,Z,0,0]).
Z = 1 ;
false.

Turn your transitions into a dgc for $L_3$ so that, for example,

?- q0([0,0,Z,0,0],[]).
Z = 1 ;
false.

Finally, define a predicate $l_3(String,\text{Numeral})$ that holds if $String$ belongs to $L_3$ and has length $\text{Numeral}$ and \texttt{numeral(Numeral)}, where

\begin{verbatim}
    numeral(0).
    numeral(succ(X)) :- numeral(X).
\end{verbatim}

For example,

?- l3(String, succ(0)).
false.

?- l3(String, succ(succ(succ(succ(0))))).
String = [0, 1, 0, 0] ;
String = [0, 1, 0, 1] ;
String = [0, 1, 1, 0] ;
String = [0, 1, 1, 1] ;
String = [1, 1, 0, 0] ;
String = [1, 1, 0, 1] ;
String = [1, 1, 1, 0] ;
String = [1, 1, 1, 1] ;
false.