CSE 341 — Assignment 8
December 7, 1998 — Solution

1. Define the append and sum relations as follows:

```
append([],L,L).
append([H|T],L,[H|U]) :- append(T,L,U).

sum([],0).
sum([X|Xs],X+S) :- sum(Xs,S).
```

Try `append` on the following. In each case reject the answers to see what happens when CLP(R) backtracks.

- `append([a,b,c],[w,x,y,z],L).`

```
1 ?- append([a,b,c],[w,x,y,z],L).

L = [a, b, c, w, x, y, z]

*** Yes
```

- `append([a,b],Y,[a,b,c,d]).`

```
2 ?- append([a,b],Y,[a,b,c,d]).

Y = [c, d]

*** Yes
```

- `append([a,c],Y,[a,b,c,d]).`

```
3 ?- append([a,c],Y,[a,b,c,d]).

*** No
```

- `append(X,Y,[a,b,c,d]).`

```
4 ?- append(X,Y,[a,b,c,d]).

Y = [a, b, c, d]
X = []

*** Retry? y

Y = [b, c, d]
X = [a]
```

```
*** Retry? y

Y = [c, d]
X = [a, b]

*** Retry? y

Y = [d]
X = [a, b, c]

*** Retry? y

Y = []
X = [a, b, c, d]

*** Retry? y

*** No
```

- `append(X,Y,Z).`

  ```
  5 ?- append(X,Y,Z).

  Z = Y
  X = []

  *** Retry? y

  Z = [_h1 | Y]
  X = [_h1]

  *** Retry? y

  Z = [_h1, _h5 | Y]
  X = [_h1, _h5]

  *** Retry? y

  Z = [_h1, _h5, _h9 | Y]
  X = [_h1, _h5, _h9]

  *** Retry? n
  ```

Now try `sum` on the following, again backtracking when possible.

- `sum([5,10,20],N).`

```
6 ?- sum([5,10,20],N).

N = 35

*** Yes
```

- `sum([5,10,20],8).`

```
7 ?- sum([5,10,20],8).

*** No
```

- `sum([5,X],100).`

```
8 ?-  sum([5,X],100).

X = 95

*** Yes
```

- `sum([5,X,Y],100).`

```
9 ?- sum([5,X,Y],100).

X = -Y + 95

*** Yes
```

- `sum(A,100).`

```
10 ?- sum(A,100).

A = [100]

*** Retry? y

A = [_t3, -_t3 + 100]

*** Retry? y

A = [_t3, _t5, -_t5 - _t3 + 100]

*** Retry? n
```

2. Exercise P4.3 from Marriott and Stuckey (define the `abs` predicate). Show your predicate working with various combinations of constants and variables as arguments, including both arguments constants, both variables, and one constant and one variable. Backtrack to find multiple answers.

   P4.3. Write rules defining a predicate `abs(X,A)` which holds if A is the absolute value of X.

   Solution:

```
abs(X,-X) :- X < 0.
abs(X,X) :- X >= 0.
```

   Output:

```
3 ?- abs(5,2).

*** No

4 ?- abs(5,-5).

*** No

5 ?- abs(5,5).

*** Yes

6 ?- abs(-5,5).

*** Retry? y

*** No

7 ?- abs(X,5).

X = -5

*** Retry? y

X = 5

*** Yes

8 ?- abs(5,A).

A = 5
```

```
*** Yes

9 ?- abs(-5,A).

A = 5

*** Retry? y

*** No
```

3. Exercise P6.3 from Marriott and Stuckey (finding temperature values).

   P6.3. Consider a metal plate modelled using a 9 by 9 grid. Suppose we know that the temperature at the center of the plate is $50C$ and at the point midway between the center and the north side, as well as the point midway between the center and the east side is $90C$. Suppose we also know that exterior points on the same side have the same temperature. Use the program in Example 6.4 to fine values for N, S, E, and W where these are the temperature of the exterior points on the north, south, east, and west sides respectively. Nost that the temperature of the corner points is irrelevent.

   Example 6.4:

```
rows([_,_]).
rows([R1,R2,R3|Rs]) :- cols(R1,R2,R3), rows([R2,R3|Rs]).

cols([_,_],[_,_],[_,_]).
cols([TL, T, TR | Ts],[ML, M, MR | Ms],[BL, B, BR | Bs]) :-
        M = (T + ML + MR + B)/4,
        cols([T,TR|Ts],[M,MR|Ms],[B,BR|Bs]).
```

   Solution 1:

```
listindex ([X|Xs],0,X).
listindex ([_|Xs],I,X) :- I > 0, listindex(Xs,I-1,X).

matrixindex (M,I,J,X) :- listindex(M,I,Row), listindex(Row,J,X).

matrixrow(M,I,Row) :- listindex(M,I,Row).

matrixcol([],I,[]).
matrixcol([Row|Rest],I,[X|Xs]) :- listindex(Row,I,X), matrixcol(Rest,I,Xs).
```

```
listall([],X).
listall([X|Xs],X) :- listall(Xs,X).

length([],0).
length([_|Xs],1+L) :- L >= 0, length(Xs,L).

plate(P,N,S,E,W) :- length(P,9),
        listall(Nrow,N), length(Nrow,9), matrixrow(P,0,Nrow),
        listall(Srow,S), length(Srow,9), matrixrow(P,8,Srow),
        listall(Ecol,E),append([N|Ecol],[S],Eside), matrixcol(P,8,Eside),
        listall(Wcol,W),append([N|Wcol],[S],Wside), matrixcol(P,0,Wside).

exper(N,S,E,W) :- plate(P,N,S,E,W), matrixindex(P,4,4,50),
        matrixindex(P,2,4,90), matrixindex(P,4,6,90), rows(P).
```

Solution 2:

```
exper2(N,S,E,W) :-
        P = [[_,  N,  N,  N,  N,  N,  N,  N,  _],
             [W,  _,  _,  _,  _,  _,  _,  _,  E],
             [W,  _,  _,  _, 90,  _,  _,  _,  E],
             [W,  _,  _,  _,  _,  _,  _,  _,  E],
             [W,  _,  _,  _, 50,  _, 90,  _,  E],
             [W,  _,  _,  _,  _,  _,  _,  _,  E],
             [W,  _,  _,  _,  _,  _,  _,  _,  E],
             [W,  _,  _,  _,  _,  _,  _,  _,  E],
             [_,  S,  S,  S,  S,  S,  S,  S,  _]],
        rows(P).
```

Output:

```
10 ?- exper(N,S,E,W).

E = 0.248541*W + 151.127
S = -W - 81.8983
N = -0.248541*W + 130.772

*** Retry? n

11 ?- exper2(N,S,E,W).

E = 0.248541*W + 151.127
S = -W - 81.8983
N = -0.248541*W + 130.772

*** Retry? n
```

4. Define a rule `listmax` that finds the maximum element of a list of numbers. For example:

```
listmax([10,5,8],X)   succeeds with X=10
listmax([A,5,8],100)  succeeds with A=100
listmax([A,5,8],5)    fails
```

Try your rule on the above cases, and also `listmax([A,B,C],100)` and `listmax(A,100)`. (Backtrack to find all of the answers, or if there are an infinite number, the first several answers that are produced.)

Solution:

```
listmax ([A],A).
listmax ([A|Xs],A) :- listmax(Xs,B), B <= A.
listmax ([X|Xs],A) :- listmax(Xs,A), A > X.
```

Output:

- `listmax([10,5,8],X).`

```
14 ?- listmax([10,5,8],X).

X = 10

*** Retry? y

*** No
```

- `listmax([A,5,8],100).`

```
15 ?- listmax([A,5,8],100).

A = 100

*** Retry? y

*** No
```

- `listmax([A,5,8],5).`

```
16 ?- listmax([A,5,8],5).

*** No

13 ?- listmax([A,B,C],100).

A = 100
B <= 100
C <= B
```

```
*** Retry? y

A = 100
B < C
C <= 100

*** Retry? y

B = 100
C <= 100
A < 100

*** Retry? y

C = 100
B < 100
A < 100

*** Retry? y

*** No
```

- `listmax(A,100).`

```
18 ?- listmax(A,100).

A = [100]

*** Retry? y

A = [100, _t2]
_t2 <= 100

*** Retry? y

A = [100, _t3, _t2]
_t3 <= 100
_t2 <= _t3

*** Retry? y

A = [100, _t4, _t3, _t2]
_t3 <= _t4
_t4 <= 100
_t2 <= _t3
```

```
*** Retry? n
```