Tree Constraints

CLP($\mathcal{R}$) solves constraints on real numbers and *trees*.
E.g. lists (these are really tree constraints).

A *tree constructor* is a symbol beginning with lowercase letter.

A *tree* is either a constant (`emptytree`) or an ordered list of *children* (`node(5,emptytree,emptytree)`).

View any user defined data structure as a tree.

Only equality constraint works on trees: `point(X,10) = point(5,Y).`

(*variables in CLP($\mathcal{R}$) start uppercase while trees and predicates begin lowercase*)

Tree Examples

Tree examples:

```
fred
X
point(10,20)
point(X,Y)
line(point(X1,Y1),point(X2,Y2))
node(3,node(4,emptytree,emptytree),emptytree)
```

Special syntax for lists:

```
[]
[1,2,3]
[1 | [2,3]]
[a,b,[c],[d]]
```


Tree Constraints

```
1 ?- A = point(X,Y).

A = point(X, Y)
*** Yes

2 ?- point(X,X) = point(10,Y).

Y = 10
X = 10
*** Yes

3 ?- point(X,X) = point(10,20).

*** No
```


Tree Constraints (cont)

```
4 ?- [A,B,C] = [1,2,3].

C = 3
B = 2
A = 1
*** Yes

5 ?- [X|Xs] = [1,2,3].

Xs = [2, 3]
X = 1
*** Yes

6 ?- [X|Xs] = [].

*** No
```

More Tree Examples

```
flatten([],emptytree).
flatten(List,node(Data,Left,Right)) :-
   flatten(LeftList,Left),
   flatten(RightList,Right),
   append(List,LeftList,[X|RightList]).
```

Consider the goals

```
flatten([1,2,3],Tree).
```

and

```
flatten(List,node(3,node(4,emptytree,emptytree),emptytree)).
```


Modeling Circuits

```
resistor(lead(I1,V1),lead(I2,V2),Ohms) :-
   I1+I2=0,
   V2-V1=I1*Ohms.

battery(lead(I1,V1),lead(I2,V2),Volts) :-
   V1 = V2+Volts,
   I1+I2=0.

electrical_ground(lead(0,0)).

ammeter(lead(I1,V),lead(I2,V),I1) :-
   I1+I2=0.

voltmeter(lead(0,V1),lead(0,V2),Volts) :-
   V1-V2=Volts.
```


Modeling Circuits (cont)

```
connect(Leads) :-
  same_voltages(Leads),
  currents_sum(Leads,0).

same_voltages([]).
same_voltages([L]).
same_voltages([lead(I1,V),lead(I2,V)|More]) :-
   same_voltages([lead(I2,V)|More]).

currents_sum([],0).
currents_sum([lead(I1,V1)|More],I1+Sum) :-
   currents_sum(More,Sum).
```


Circuit Example 1

```
one_resistor(Volts,Ohms,Amps) :-
   battery(B1,B2,Volts),
   resistor(R1,R2,Ohms),
   ammeter(A1,A2,Amps),
   electrical_ground(G),
   connect([B2,A1]), connect([A2,R1]), connect([R2,B1,G]).
```

Circuit Example 2

```
divider(Volts,Ohms1,Ohms2,Amps,VCenter) :-
   battery(B1,B2,Volts),
   resistor(R1,R2,Ohms1),
   resistor(S1,S2,Ohms2),
   ammeter(A1,A2,Amps),
   voltmeter(V1,V2,VCenter),
   electrical_ground(G),
   connect([B2,A1]), connect([A2,R1]),
   connect([R2,S1,V1]), connect([S2,V2,B1,G]).
```


CLP($\mathcal{R}$) Summary

- Declarative.
- Solves arbitrary linear (in)equalities.
- Solves tree and nonlinear constraints via single step deductions.
- Pass by constraint.
- Searches derivation trees.
- Dynamically typed: `5 = emptytree.` $\Rightarrow$ `*** No`