Rule-based Programming

Assignment #2

Excercise 1 (Color Mixing)**.** Represent colors as propositional CHR constraints `red, blue,...`. Write simplification rules that describe the result of mixing two primary colors, e.g.

```
  red, yellow <=> orange.
```

Observe what happens if you have all three primary colors, in different orders, in the query. How do you ensure that the answer is always the same, say `brown`?

Excercise 2. In an example from geometry, assume that lines are given by variables (or constants) and that CHR constraints express the relationships between two lines, `parallel` and `orthogonal`. Write propagation rules that derive further such relationships from the given relationships, e.g.

```
  parallel(L1,L2), parallel(L2,L3) ==> parallel(L1,L3).
```

Ensure termination.

Excercise 3. Compute the factorial of a number `n`, given `fact(1),...,fact(n)`.

Excercise 4. Compute the factorial of a number `n`, given only `fact(n)`, i.e. add rules to generate candidates.

Excercise 5. Implement the Sieve of Erathostenes:

```
  gen @ upto(N) <=> N>1 | M is N-1, upto(M), prime(N).
  sift @ prime(X) \ prime(Y) <=> Y mod X =:= 0 | true.
```

Test the program, then modify it such that it factorizes natural numbers.