

Constraint Handling Rules (CHR)

Exercise 1. Compare the following CHR programs, which consist of *one* of the given rules by posing the given queries. Check your answers with the system's answers. Make sure, you understand why seemingly innocuous rules produce different answers.

c1 @ c(X), c(X) <=> q(X,X).	Queries:
c2 @ c(X), c(Y) <=> r(X,Y).	a) c(X), c(X)
c3 @ c(X), c(X) ==> q(X,X).	b) c(X), c(Y)
c4 @ c(X), c(Y) ==> r(X,Y).	c) c(X), c(Y), X=Y

More variants:

q1 @ p(X,Z), q(Z,Y) <=> q(X,Y).	
q2 @ q(Z,Y), p(X,Z) <=> q(X,Y).	Queries:
q3 @ p(X,Z), q(Z,Y) ==> q(X,Y).	d) p(A,B), q(B,C)
q4 @ q(Z,Y), p(X,Z) ==> q(X,Y).	e) p(A,B), q(B,C), p(D,A)
q5 @ p(X,Z) \ q(Z,Y) <=> q(X,Y).	
q6 @ q(Z,Y) \ p(X,Z) <=> q(X,Y).	

Comment on the system's answers for queries a) to e).

Comment on the system's answers for the rule q5 and the following two queries.

- p(X,C), p(Y,C), q(C,A) und
- p(Y,C), p(X,C), q(C,A).

Exercise 2. Implement the constraints `less/2` (encoding $<$) und `leq/2` (encoding \leq) and their mutual relations/interactions in CHR. You may find the lecture's CHR program for the \leq constraint helpful.

For an example query, take your last name as a sequence of variables with \leq constraints between succeeding characters.

The name *Fruehwirth* translates to the query

F leq R, R leq U, U leq E, E leq H, H leq W, W leq I, I leq R, R leq T, T leq H
with answer F leq E, H=E, I=E, R=E, T=E, U=E, W=E.

Tests should include (at least) three more queries consisting of combined `less` and `leq` constraints.

Exercise 3. Use the CHR Constraint `leq/2` from the previous exercise for a “constrain-and-generate” version of the sorting example from assignment #1-2. To this end, replace the `clpq`-Constraint `=<` by the CHR-constraint `leq`.

Your tests should (at least) include the following queries

```
?- permsortCHR([1,A,3],[1,3,7]).  
?- permsortCHR([2,A],X).  
?- permsortCHR([A,B,A],X).  
?- permsortCHR(List,[1,X,3]).  
?- permsortCHR([1,X,Y],[X,1,Y]), permsortCHR([4,5,10],[Z,Y,W]).
```