Sort:

Net as callback function. Single call single return per instane. Call with :new sort(a,{}) and keep the reference for the return value.

import de.renew.unify.*; import de.renew.net.*; NetInstance net: List v,x,y,z; int a,b,c,d; -[{},x]-:new(v,x) [v,x] :ret(x) [v,x] [v,x]guard(v!={}) [v,v,x] $[v,{},x]$ $[{a:y},v,x]$ net:ret(a,b) net:new swap(a,b) -[y,y,{a:x}] ▶ -[v,y,{b:x}] [v,y,x,net] -[v,y,x,net] [v,y,x,a,b][v,y,x]-[v,y,x,a,b] -[v,{a:y},{b:x}] guard(y!={})