```lua
local _,the = require"lib", require"the"
local fmt, inc,slots = _.fmt, _.inc, _.slots
local class,OBJ      = _.class, _.OBJ

local ABCD = class("ABCD",OBJ)

function ABCD:new(data,rx)
  self.data, self.rx = data or "", rx or ""
  self.yes, self.no  = 0,0
  self.known, self.a, self.b, self.c, self.d = {},{},{},{},{} end

function ABCD:exists(x,   new)
  new = not self.known[x]
  inc(self.known,x)
  if new then
    self.a[x]=self.yes + self.no; self.b[x]=0; self.c[x]=0; self.d[x]=0 end end

function ABCD:report(    p,out,a,b,c,d,pd,pf,pn,f,acc,g,prec)
  p = function (z) return math.floor(100*z + 0.5) end
  out= {}
  for x,xx in pairs( self.known ) do
    pd,pf,pn,prec,g,f,acc = 0,0,0,0,0,0,0
    a= (self.a[x] or 0); b= (self.b[x] or 0);
    c= (self.c[x] or 0); d= (self.d[x] or 0);
    if b+d > 0     then pd   = d     / (b+d)         end
    if a+c > 0     then pf   = c     / (a+c)         end
    if a+c > 0     then pn   = (b+d) / (a+c)         end
    if c+d > 0     then prec = d     / (c+d)         end
    if 1-pf+pd > 0 then g=2*(1-pf) * pd / (1-pf+pd) end
    if prec+pd > 0 then f=2*prec*pd / (prec + pd)   end
    if self.yes + self.no > 0 then
       acc= self.yes /(self.yes + self.no) end
    out[x] = {data=self.data,rx=self.rx,num=self.yes+self.no,
              a=a,b=b,c=c,d=d,acc=p(acc),
              prec=p(prec), pd=p(pd), pf=p(pf),f=p(f), g=p(g), class=x} end
  return out end

function ABCD:pretty(t)
  print""
  local s1  = "%10s|%10s|%4s|%4s|%4s|%4s"
  local s2  = "|%3s|%3s|%3s|%4s|%3s|%3s|"
  local d,s = "---", (s1 .. s2)
  print(fmt(s,"db","rx","a","b","c","d","acc","pd","pf","prec","f","g"))
  print(fmt(s,d,d,d,d,d,d,d,d,d,d,d,d))
  for key,x in pairs(slots(t)) do
    local u = t[x]
    print(fmt(s.."%s", u.data,u.rx,u.a, u.b, u.c, u.d,
                       u.acc, u.pd, u.pf, u.prec, u.f, u.g, x)) end end

function ABCD:adds(gotwants, show)
  for key,one in pairs(gotwants) do
    self:exists(one.want)
    self:exists(one.got)
    if one.want == one.got then self.yes=self.yes+1 else self.no=self.no+1 end
    for x,xx in pairs(self.known) do
      if   one.want == x
      then inc(one.want == one.got and self.d or self.b, x)
      else inc(one.got  == x       and self.c or self.a, x) end end end
  return show and self:pretty(self:report()) or self:report() end

return ABCD
```