```
c \succ d ==
  LET NumberPreferring(a, b) ==
    (*************************************************************************)
    (* The number of voters who prefer candidate `a' to candidate `b'.       *)
    (*************************************************************************)
    Cardinality({v \in 1..V : RankBy(a, v) < RankBy(b, v)})
  IN  NumberPreferring(c, d) > NumberPreferring(d, c)

CondorcetRanking ==
  LET IsDominatingSet(D, C) ==
    (*************************************************************************)
    (* True iff D is a dominating set in the election for the set C of       *)
    (* candidates.                                                           *)
    (*************************************************************************)
    /\ D # {}
    /\ \A d \in D : \A e \in C \ D : d \succ e

  CWinners(C) ==
    (*************************************************************************)
    (* The set of Condorcet winners in the election for the set C of         *)
    (* candidates.                                                           *)
    (*************************************************************************)
    CHOOSE D \in SUBSET C :
      /\ IsDominatingSet(D, C)
      /\ \A E \in SUBSET C : IsDominatingSet(E,C) => (D \subseteq E)

  RECURSIVE CRanking(_)
  CRanking(C) == IF C = {} THEN <<>>
                           ELSE LET CW == CWinners(C)
                                IN  <<CW>> \o CRanking(C \ CW)
  IN  CRanking(Cand)

(***************************************************************************)
(* In this definition of CondorcetRanking, the LET definition of           *)
(* IsDominatingSet(D, C) uses the `dominates' relation \succ that is       *)
(* defined in terms of the votes in the election for all candidates,       *)
(* rather than the votes in an election only for candidates in C. Explain  *)
(* why this doesn't matter.                                                *)
(***************************************************************************)
```

CLOSE