Programming social choice in logic: some basic experimental results when profiles are restricted

Kenryo Indo

Department of Business Administration, Faculty of Economics, Kanto Gakuen University, 200 Fujiaku, Ota, Gunma 373-8515, Japan

kindo@kanto-gakuen.ac.jp

Abstract

Social choice theory argues mathematical models and their logical consequences for group decision making based on individuals' preference orderings axiomatically. The most basic results have been proved in this field are two impossibility results for group decision making. Arrow's general (im)possibility theorem for preference aggregation procedures, i.e., there is no non-dictatorial social welfare functions (SWFs), and the Gibbard-Satterthwaite theorem for the strategy-proof social choice functions (SCFs), i.e., any voting procedure which cannot be manipulated by any individual's false report on his/her own preference ordering should be dictatorial. The above two classical results are proved for unrestricted domain, i.e., any combinations of individual orderings can never be prohibited. This paper shows that, there are 18 non-dictatorial SWFs and 196 strategy-proof non-imposed SCFs by eliminating a double cyclical 6 profiles, each of these profiles are considered to be minimal and sufficient to prove the Arrow-type dictatorship for every two-individual and three alternatives. While these profiles are used in the Arrow's original proof, we may have a thorough experimentation eliminating subsets of a set of special 12 profiles which is sufficient to deduce a dictatorship. The automated proof, instead of pure mathematical proof, unveils fairly comprehensive pattern rules for SWFs in parallel with SCFs.

Keywords: Arrow's impossibility theorem, the Gibbard-Satterthwaite theorem, domain conditions, Prolog

1. Introduction

Social choice theory studies collective decision making occurs in any multi-agent situation by using axiomatic models for individual preference orderings (i.e., rankings) on which various types of collective decision rules may be defined. Computational techniques have been rarely explored for these axiomatic models except for some sorts of voting, partly because simulation methods are considered to be appropriate for qualitative models but not for qualitative ones. However, as we will demonstrate throughout this paper axiomatic modelling for the multi-agent situations can be explored meaningfully if we utilize a popular logic programming. It is not only useful for proving theorems in the literature, but also helpful leads to find new results.

Kenneth J. Arrow's general impossibility theorem is a fundamental result regarding social aggregation of individual orderings (Arrow 1951; Arrow 1963). Given a set of individuals and a set of finite alternatives, under some moderate conditions on permissible orderings of the individual and the society as a whole, a dictatorship is necessarily concluded. A social welfare function (SWF) is required to satisfy the following five axioms;

(U) unrestricted individual orderings,

(T) transitivity of aggregated orderings,

(P) weak Pareto principle, or unanimity,

(IIA) independence of irrelevant alternatives,

(D) non-dictatorship.

Arrow proved that any aggregation rule which satisfies the first four axioms should be dictatorial, and therefore it is impossible to satisfy all the five axioms.

Restricting permissible orderings for each individual, i.e., domain conditions (Gaertner 2001), may help to escape the society can from impossibilities. Many domain conditions are known. For example, value restriction, single-peakedness, single-cavedness, extreme restriction, and so on for pairwise-majority vote. Kalai and Muller (1977) proposed decomposability under an implication of decisiveness as the necessary and sufficient condition for existence of nondictatorial SWFs, paralleling the strategy-proof voting procedures, over common addmissible domains. Blau and Muller (1983) modified it for individual permissible domains. The saturated domains of Kalai, Muller, and Satterthwaite (1979) extend decomposability to economic environments. A a finer sufficient condition of dictatorial result is "free triple" proposed in the 1st edition of Arrow's book. The condition was criticized by Blau (1957) and then modified in the 2nd edition. Kelly (1994) has elaborated this condition. Recently, Ozdemir and Sanver (2007) modified it and connected it to the saturated domains (and the decomposability) to graphically interpret the linear orderings.

All the above conditions in the literature restrict the permissible set of orderings. Especially for the case of three alternatives, they can give only a coarse prediction at most. That is, if one the 6 orderings eliminated from the common admissible domain, there is a SWF (and a SCF) except for the cases of singleton set. And in this sense, the two axioms can be considered equivalent for every common admissible domains. However, this paper proposes finer conditions which restrict the permissible set of profiles as the inputs of collective decision making procedure, and clarify interrelationships of the two set of axioms for admissible set of profiles.

Recently, computer scientists have been interested in social choice theory as the foundation of mechanism design for multiagent systems (Shoham et al 2008). Lin and Tang (2008; 2009) argued computer aided proofs of Arrow's impossibility theorem and the Gibbard-Satterthwaite theorem. Computing the SWFs can be seen as a constraint satisfaction problem (CSP) and they alleged that it can be solved for two individuals and three alternatives by using SAT solver. They also alleged that the same problem can be proved by using Prolog without showing the program. Independently from them, Indo (2007) introduced a complete Prolog program which proves Arrow's theorem and Wilson's theorem for two-individual three alternatives under linear ordering (Wilson 1972). This approach has been extended and applied to Gibbard-Satterthwaite theorem, domain conditions, simple games, pairwise majority vote (Indo 2009).

It is noteworthy that this program is also useful to investigate domain conditions in order to avoid the classical impossibilities. If we take a subset of profiles instead of the universal domain, and there may be an aggregation rule that satisfies all the five axioms except for U, unrestricted domain. We will call such an aggregation rule a SWF restricted to a subset of profiles.

Indeed, we can select double sets each of which is consists of 6 profiles:

P_1. (a > c > b, c > b > a),
P_2. (a > b > c, c > a > b),
P_3. (b > a > c, a > c > b),
P_4. (b > c > a, a > b > c),
P_5. (c > b > a, b > a > c),
P_6. (c > a > b, b > c > a),

And Q_ks are profiles (r_2, r_1) such that P_k = (r_1, r_2), k = 1, …, 6. Note that these profiles can be cyclically generated. We can get $P_{k'}$s (and $Q_{k'}$s) where k' = ((k + 1) mod 6) + 1 by sequentially reversing for a single pair for each k = 1, 2, 3, 4, 5, 6. Each set of cyclical 6 profiles is respectively minimal in that it covers all the possible binary patterns of profiles and suffices to deduce a dictatorship under the axioms even if all the other 24 profiles remain, while Arrow's impossibility theorem no longer true if both of the cyclical profiles are incomplete. Further, there are maximally 18 SWFs excluding the condition U when these 12 cyclical profiles are eliminated, as we will se in Section 4.

These profiles are used in the Arrow's original proof with a minor modification. However, in this paper a computational step is adopted instead of pure mathematics to prove the theorems. It is not an ad hoc computer simulation. Due to the nature of automated theorem proving (Robinson 1965; Chang & Lee 1973), Prolog unveils comprehensive patterns for SWFs in parallel with strategy-proof non-imposed SCFs for various elimination patterns for the 12 profiles. The programs in this paper have been tested using SWI-Prolog version 5.6.52 distributed by swi-prolog.org (http://www.swi-prolog.org).

2. Impossibility Theorems

Given a set of individuals, N = {1, 2, …, n}, and a set of finite number of alternatives, A = {x, y, z, …}. A binary relation *R* on a set is a weak ordering of A if it is complete and transitive and a linear ordering if it is also antisymmtric. *R* is complete if for all x and y, either x*R*y or y*R*x. *R* is transitive if for all x, y, and z, if x*R*y and y*R*z then x*R*z. Indifference relation, which means x*R*y and y*R*x, is denoted by x*I*y. *R* is atisymmetric if for all *x* and *y*, if *xIy* then $x = y$. Binary relation x*P*y stands for the strict part, i.e., *xRy* and *not yRx*.

In this paper, linear ordering is assumed. Let a profile $R_N = (R_1, R_2, \dots, R_n)$ be a combination of the all individuals' orderings.

Definition. A social welfare function (SWF) is defined as a function of the set of all the permissible profiles to the set of social orderings. Ordering of the society as a whole, R_s, = f(R_N), aggregated by a SWF should satisfy the following five conditions.

(U) The SWF is defined for every profile (i.e., unrestricted domain, or universal domain).

(T) The social ranking R_S should be transitive.

(P) For any pair of alternatives, x and y, if xR_iy for every individual i then xR_Sy.

(IIA) For any (x, y) a pair of alternatives if every individual keeps the ranking for this pair after her or his profile changes to R_N' from R_N then xR_Sy iff $xR_S'y$.

An individual i is called a dictator if for any pair of alternatives, x and y, if xR_iy then xR_S y.

(D) There is no dictator.

A SWF called resolute if the social ordering is linear for every profile. We will abuse the notion of social welfare function when its domain is restricted to a subset of profiles, and therefore dropping the conditions U and D.

Theorem (Arrow's Impossibility Theorem). If there are one or more individuals, and more than two alternatives, then any social welfare function that satisfies U, T, P, and I is dictatorial.

Definition. A (resolute) social choice function (SCF) is a function that selects a single alternative from each non-empty subset of alternatives (i.e., the agenda) for every permissible profile.

A SCF is manipulable if there is an individual who can report a false ordering and thereby lead to more preferable outcome for herself/himself.

(U') A SCF is defined for every profile and the agenda is restricted to A.

(S) A SCF is not manipulable (strategy-proofness, or non-manipulability)

(CS) There is no alternative x such that x is never selected (as a single winner) for every profile and agenda. (citizens' sovereignty, or non-imposition)

A single individual such that her/his top rank alternative is always selected as a winner is called a dictator. A SCF is dictatorial if there is a dictator.

Allan Gibbard and Mark Satterthwaite independently argued the closed relationship between non-manipulable voting rules to Arrow's social welfare function using a framework of non-cooperative games.

Theorem (The Gibbard-Satterthwaite Theorem). If there are one or more individuals, and more than two alternatives, then any (resolute) social choice function that satisfies U', S, and C is dictatorial.

Although the mutual derivation of the two impossibility results is well known, ambiguity about the notion of equivalence has been posed critically. As for the proof of these theorems the reader should refer Gibbard (1973), Satterthwaite (1975). Barbera (2001) and Taylor (2005) are also helpful.

In the remaining part of this paper, we will focus on the case of two individuals and three alternatives. It is well known that the general case of these two impossibility theorems can be proved inductively based on this basic case.

3. Logic Programs

This section outlines some applications using the social choice logic programming (SCLP) a computational version of the axiomatic social choice theory (Indo 2007; Indo 2009). For two-individual three-alternative case, all the possible strategy-proof social choice functions as well as social welfare functions given a subset of profiles easily generated by using a simple recursion. Thereby we can verify their interrelationships.

A SWF (Social Welfare Function) is defined as a function from any permissible combination of individuals' orderings, i.e., profiles, to an ordering of the society as a whole. The Prolog version of SWF can be coded as a simple recursion.

```
swf([ ], [ ]).
swf([P-S|F], [P|L]):-
     swf(F, L),
     r(S),
     pareto(P-S),
```

```
iia(P-S, F).
```

Because of the axiom T, predicate `r(S)` represents a transitive ordering, while all the results explained in the next section assume linear valued. Its transitivity can be easily checked as follows:

```
?- r(R), r([X,Z],R),r([Z,Y],R),
\+ r([X,Y],R).
false.
```

The above line after a prompt `?-` is a query such that "Is there any term simultaneously substitutable for variables `R`, `X`, `Y`, and `Z` such that these consist a violation of the transitive relation? Please show me examples, if possible." The system's answer `false` means that there is no such violation.

Recursive construction in Prolog language is familiar to AI programmers in order to write a problem solving system. However, in order to generate SWFs, we can definitely take the set of all profiles on which the SWF is defined in the second argument of two-arity predicate `swf` /2.

```
swf(F):- all_profiles(L), swf(F, L).
```

Note that the above predicate `swf` /1 is intended to satisfy all the Arrow's axioms except for (D). We use Prolog's backtracking to generate all the possible aggregation rules. Further, we will think that the experimentally observed fact that every possible rule is dictatorial to be considered a proof of the impossibility theorem. Then, a proof of Arrow's theorem by Prolog is given as follows:

```
?- swf( F), ¥+ dict_swf( F).
false.
```

As long as both that Arrow's theorem is right and that programs model the axioms appropriately, the theorem would be proved computationally after a time. This task completed in 4.0 seconds on my PC (Windows XP SP3, Celeron 1GHz), even if no special ideas of fast computing other than recursion. Indeed, it depends on sequences of the profile list `L` and fairly improves if the cyclic 6 profiles explained in the first section are at the last position of `L`.

Note that weak ordering consists of 6 intermediate (composite) orderings regarding the same above sequence and a total indifferent ordering as well as 6 these linear ones. These 12 profiles are minimal again for transitivity-valued SWF in accordance with the original definition.

Similarly, a program in order to prove the Gibbard-Satterthwaite theorem may be written as follows:

```
scf([P-A | F], [P | L]):-
    scf(F, L),
    x(A),
    ¥+ manipulable(P-S, F, _).
scf(F):-
    all_profiles(L),
    scf(F, L),
    non_imposed(F).
```

Self-contained Prolog source code that can prove all the experimental results, which will be explained in the next section, are shown in Appendix.

4. Experimental Results

The experimental results are summarized in the following three tables. Table 1 shows the number of restricted domains when the cyclical 12 profiles {P_1,…, P_6,

$Q_1,\ldots, Q_6\}$ explained in Section 1 eliminated from the two-individual three-alternative unrestricted domain. Columns of Table 1 represent numbers of eliminated profiles and rows numbers of possible linear-valued SWFs. There are $2^{12} = 4096$ patterns of elimination to be verified. Each cell designates the number of such restricted domains.

Table 1 aroud here

Similarly, Table 2 shows the strategy-proof and non-imposed SCFs. Table 3 represents the cross analysis between SWFs and SCFs. There are 18 (or 51 if transitive-valued) non-dictatorial SWFs, and 196 strategy-proof non-imposed SCFs, if both cyclical profiles have been eliminated completely. It is noteworthy that the number is maximal for SWFs, without regard to selection of all the other 24 profiles, but not for SCFs.

Table 2 aroud here

In the light of these data, the 12 cyclical profiles are considered minimal both for SWFs and SCFs in the following sense.

- (Fact 1) The impossibility no longer occurs if more than half of the 12 profiles have been eliminated. (See the left half of Table1 and Table 2.)
- (Fact 2) The possibility may occur if either of two cyclical profiles $\{P_1,\ldots, P_6\}$ or $\{Q_1,\ldots, Q_6\}$ is complete. (See the row 3 column 10 of Table1 and Table 2.)

Indeed, these are 18 adjacent and double profile pairs in the cyclical ordering (See test1 in Appendix).

Mention should also be made of another cross analysis by transitive-valued SWFs similar to Table 3. Especially, their two columns 2 and 3 are of the same numbers. But the details are left to the leader. Intuitively, we find a somewhat obscure correlation in Table 3.

Table 3 aroud here

- (Fact 3) Arrow's axioms and the G-S axioms, except for U and U', to be considered equivalent in that a dictatorial result can be proved under the former if and only if so does under the later. Table 3 shows that this is the case for 169 domains where both are true and 3897 domains both are false. However, there are 30 domains where there is no SWF but a non-dictatorial non-imposed SCF exist.

Therefore, the two set of axioms have similar semantics in that each domain where the impossibility occurs is interpreted as a possible world (its accessibility relation may be defined in different ways, for example unilateral deviations from each profile, or reversions of one of the individuals orderings), but not quite equivalent. Additionally, if we substitute strategy-proofness by monotonicity, then the experimental data pattern obtained is the same as Table 1. Therefore, as long as we eliminate a subset of the 12 cyclical profiles, SWF and monotone non-imposed SCF are equivalent regarding number of possibilities for every restricted domain as a subset of profiles. However, even the equivalence in this sense is not true if the elimination beyond the cyclical profiles.

5. Conclusions

In this paper, patterns for SWFs in parallel with strategy-proof and non-imposed SCFs for various elimination patterns for the cyclical 12 profiles to be considered minimal in order to prove the Arrow’s theorem for the 2-individual 3-alternative linear ordering case. This also clarifies the notion of equivalence between the two set of axioms of Arrow’s impossibility theorem and the Gibbard-Satterthwaite theorem when the permissible set of profiles are restricted.

References

Arrow KJ (1951) Social choice and individual values, 1st edn.Wiley, New York.
Arrow KJ (1963) Social choice and individual values, 2nd edn. Wiley, New York.
Available via WWW http://cowles.econ.yale.edu/P/cm/m12-2/m12-2-pre.pdf
Barbera S (2001) An introduction to strategy-proof social choice functions. Social Choice and Welfare 18(4): 619–653
Blair D, Muller E (1983) Essential aggregation procedure on restricted domains of preferences. Journal of Economic Theory 30: 34–53
Chang C-L, Lee RC-T (1973) Symbolic logic and mechanical theorem proving. Academic Press, Orlando
Gaertner W (2001) Domain conditions in social choice theory. Cambridge University Press, Cambridge
Gibbard A (1973) Manipulation of voting schemes: a general result. *Econometrica* 41: 587–602
Kelly JS (1994) The free triple assumption. Social Choice and Welfare 11(2): 97–101
Indo K (2007) Proving Arrow's theorem by Prolog. Computational Economics 30(1): 57–63. doi: 10.1007/s10614-007-9086-2
Indo K (2009) Modeling a small agent society based on the social choice logic programming. In T. Terano et al (eds), Agent-based approaches in economic and social complex systems V: post-proceedings of the AESCS international workshop 2007, Springer Verlag, Tokyo, pp. 93–104
Kalai E, Muller E (1977) Characterization of domains admitting nondictatorial social welfare functions and nonmanipulable voting procedures. J Econ Theory 16: 457–469
Kalai E, Muller E, Satterthwaite MA (1979) Social welfare functions when preferences are convex, strictly monotonic and continuous. Public Choice 34: 87–97
Lin F, Tang P (2008) Computer-aided proofs of Arrow's and other impossibility theorems. In Proceedings of AAAI-08, pp. 114–118
Lin F, Tang P (2009) Computer-aided proofs of Arrow's and other impossibility theorems. Artificial Intelligence 173(11): 1041–1053
Ozdemir U, Sanver MR (2007) Dictatorial domains in preference aggregation. Social Choice and Welfare 28(1): 61–76
Robinson A (1965) A machine-oriented logic based on the resolution principle. JACM 12(1): 23–41
Satterthwaite MA (1975) Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. J Econ Theory 1: 187–217
Shoham Y, Leyton-Brown K (2008) Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge
Taylor AD (2005) Social choice and the mathematics of manipulation. Cambridge University Press, New York.
Wilson R (1972) Social choice theory without the Pareto principle. J Econ Theory 5: 478–486

APPENDIX

```
/* preference ordering */
```

```
lx([a,b,c]).
x(X):- lx(A), member(X,A).
% linear ordering
rc(1, [a, c, b]).
rc(2, [a, b, c]).
rc(3, [b, a, c]).
rc(4, [b, c, a]).
rc(5, [c, b, a]).
rc(6, [c, a, b]).
% (weak) ordering additionally
rc(K, A, []) :- rc(K, A).
rc(7, [a, c, b], [b, c]).
rc(8, [a, b, c], [b, a]).
rc(9, [b, a, c], [c, a]).
rc(10, [b, c, a], [c, b]).
rc(11, [c, b, a], [a, b]).
rc(12, [c, a, b], [a, c]).
rc(13, [a, b, c], [c, b, a]).
% an intransitive relation
rc(0, [a, b, c], [c, a]).
% numbering
r(K):- rc(K,_,_), K > 0.
p(K):- r(K), K < 7.
% binary relation
precede(X,Y,R):-
  append(_,[X|Z],R), member(Y,Z).
r([X,Y],R):-
  rc(R,A,B),(precede(X,Y,A);precede(X,Y,B)).
r([X, X], R):- r(R), x(X).
i([X, Y], R):- r([X, Y], R), r([Y, X], R).
p(XY, R):- r(XY, R), \+ i(XY, R).

/* profile of ordering */

pp([R, Q]):- p(R), p(Q).
agree( p, B, [R, Q]):- p(B,R), p(B,Q).
agree( r, B, [R, Q]):- r(B,R), r(B,Q).
agree( _, B, P):- \+ (member( R, P), r(B,R)).
all_profiles(L):- findall( P, pp(P), L).

/* SWF */

swf(F):- all_profiles(L), swf(F, L).
swf([], []).
swf([P-S|F], [P|L]):-
  swf(F, L), p(S), % originally, r(S).
  pareto(P-S), iia(P-S, F).
pareto(P-S):-
  ¥+ (agree(p, XY, P), \+ p(XY,S)).
iia([P,Q]-S, F):- \+ (
  x(X), x(Y), member([U,V] - T, F),
  agree(r, [X,Y], [P,U]),
```

```
  agree(r, [X,Y], [Q,V]),
  \+ agree(r, [X,Y], [S,T])).
dict_swf(J,F):-
  nth1(J,[P,Q],R),
  \+ (member([P,Q]-S,F), p(B,R), \+ p(B,S)).

/* SCF */

scf(F):- all_profiles(L), scf(F, L), cs(F).
scf([], []).
scf([P-Z|F], [P|L]):- scf(F, L), x(Z),
  \+ manipulable(P-Z, F).
cs(F):- \+ (x(X), \+ member(_-X, F)).
non_imposed(F):- cs(F).
manipulable([R,Q]-S, F):- member([P,Q]-T,F),
  (p([T,S],R); p([S,T],P)).
manipulable([R,Q]-S, F):- member([R,W]-T,F),
  (p([T,S],Q); p([S,T],W)).
best(X,Q):- x(X), \+ (x(Y), \+ r([X,Y],Q)).
dict_scf(J,F):- nth1(J,[P,Q],R),
  \+ (member([P,Q]-X,F),\+ best(X,R)).

/* tables */

fig(_):- nl,tab(12),p(K),write(K),fail.
fig(F):-
  p(J), rc(J,P,_), nl, write(J:P),
  tab(1), p(K), fig_cell([J,K],F),fail.
fig(_):- nl,write('--').
fig_cell(P,F):- member(P-S, F),!, write(S).
fig_cell(_,_):- nl,write('-').

/* restricted domain */

:- dynamic r_admit/1.
restricted_domain(L,N):- T=[1,2,3,4,5,6],
  nth1(N,T,_),select_n(L,T,N),
  abolish(r_admit/1),assert(r_admit(L)).
select_n([],[],0).
select_n([R|Q],[R|S],A):-
  select_n(Q,S,B), A is B + 1.
select_n(Q,[_|S],B):- select_n(Q,S,B).

/* the 12 cyclical profiles */

ppc([J,K]):-  rc(K, _,_), K < 7, J is (K + 3) mod 13 + 1.
ppc6(C):- findall(P,ppc(P),C).
ppc6r(C):- findall([Q,P],ppc([P,Q]),C).
ppc12(C):- ppc6(A),ppc6r(B),append(A,B,C).

/* verification */
```

```
test1:- all_profiles(L),ppc6(C),ppc6r(D),
   nth1(K,C,V),nl,write([K]),tab(1),
   nth1(J,D,W),subtract(L,[V,W],H),
   swf(F,H),\+ dict_swf(_,F), write(J; ' '),
   fail.

:- dynamic test2_data/4.

test2:- all_profiles(L),
   ppc12(B), nth0(N,[_|B],_), M is N -1,
   test2_stat(swf,M),
   select_n(C,B,N), subtract(L,C,U),
   findall(1,swf(_,U),H), length(H,I),
   findall(1,(scf(_,U),cs(F)),G),length(G,J),
   assert( test2_data(N,C,I,J) ),
   fail.
test2.

test2_stat(F,K):-
   ppc12(B),nth0(K,[_|B],_),
   member(F, [swf,scf]),
   nl,write([elim:K]),tab(1),
   test2_stat(F,K,I,D),
   bagof(1,D,W),length(W,S),
   write(I-S;' '),
   fail.
test2_stat(_,_).

test2_stat(swf,K,I,C^J^test2_data(K,C,I,J)).
test2_stat(scf,K,J,C^I^test2_data(K,C,I,J)).

test2_cross:-
    setof(J,K^C^I^test2_data(K,C,I,J),L),
    member(J,L),nl,write([J]),tab(1),
    bagof(1,C^K^test2_data(K,C,I,J),W),
    length(W,S),write(I-S;' '),
    fail.
test2_cross.

/* demo */

?- test1.
[1] 1; 2; 6;
[2] 1; 2; 3;
[3] 2; 3; 4;
[4] 3; 4; 5;
[5] 4; 5; 6;
[6] 1; 5; 6;
false.

?- test2.
[0] 2-1;
```

```
[1] 2-12;
[2] 2-48; 3-18;
[3] 2-76; 3-108; 4-36;
[4] 2-48; 3-156; 4-225; 5-60; 6-6;
[5] 2-12; 3-60; 4-228; 5-348; 6-120; 7-24;
[6] 2-2; 4-54; 5-170; 6-390; 7-252; 8-50; 9-6;
[7] 5-12; 6-60; 7-228; 8-348; 9-120; 10-24;
[8] 8-48; 9-156; 10-225; 11-60; 12-6;
[9] 11-76; 12-108; 13-36;
[10] 14-48; 15-18;
[11] 17-12;
[12] 20-1;

?- test3_stat(scf,_).

[0] 2-1;
[1] 2-12;
[2] 2-48;  3-18;
[3] 2-64;  3-120;  4-36;
[4] 2-30;  3-114;  4-255;  5-90;  6-6;
[5] 2-12;  4-144;  5-300;  6-252;  7-72;  8-12;
[6] 2-2;  5-62;  6-150;  7-294;  8-242;  9-78;  10-72;  12-18;  13-6;
[7] 6-12;  8-120;  9-132;  10-192;  11-48;  12-108;  13-48;  14-72;  15-12;  16-24;
19-24;
[8] 10-18;  11-36;  12-57;  13-30;  14-36;  15-36;  16-69;  17-36;  18-72;  19-12;
20-12;  22-36;  25-30;  28-3;  29-6;  30-6;
[9] 14-4;  17-12;  20-36;  21-12;  22-36;  23-12;  26-12;  28-24;  31-24;  34-12;  35-
12;  38-12;  40-12;
[10] 37-12;  38-6;  40-6;  41-12;  46-6;  48-12;  50-6;  74-6;
[11] 88-12;
[12] 196-1;
true.

?- test3_cross.

[2] 2-169;
[3] 2-24;  3-228;
[4] 2-6;  3-84;  4-345;
[5] 3-6;  4-144;  5-302;
[6] 3-24;  4-24;  5-168;  6-204;
[7] 5-36;  6-192;  7-138;
[8] 4-24;  5-36;  6-78;  7-168;  8-68;
[9] 5-12;  6-12;  7-60;  8-96;  9-30;
[10] 5-36;  6-36;  7-18;  8-120;  9-60;  10-12;
[11] 7-24;  8-12;  9-24;  10-24;
[12] 6-30;  7-36;  8-30;  9-24;  10-48;  11-12;  12-3;
[13] 4-6;  7-24;  8-24;  10-12;  11-18;
[14] 7-24;  8-48;  10-30;  11-10;
[15] 8-12;  9-24;  11-12;
[16] 6-12;  8-18;  9-48;  10-12;  12-3;
[17] 9-12;  10-24;  13-12;
[18] 10-60;  11-12;
```

```
[19] 6-12;  7-12;  10-12;
[20] 9-6;  10-6;  11-24;  12-12;
[21] 12-12;
[22] 8-12;  9-24;  12-24;  13-12;
[23] 13-12;
[25] 9-24;  10-6;
[26] 12-12;
[28] 10-3;  11-24;
[29] 9-6;
[30] 8-6;
[31] 12-24;
[34] 12-12;
[35] 12-12;
[37] 14-12;
[38] 11-12;  15-6;
[40] 11-12;  14-6;
[41] 15-12;
[46] 14-6;
[48] 14-12;
[50] 14-6;
[74] 14-6;
[88] 17-12;
[196] 20-1;
true.
```

swf	0	1	2	3	4	5	6	7	8	9	10	11	12	total
2							2	12	48	76	48	12	1	199
3								60	156	108	18			342
4							54	228	225	36				543
5						12	170	348	60					590
6						60	390	120	6					576
7						228	252	24						504
8					48	348	50							446
9					156	120	6							282
10					225	24								249
11				76	60									136
12				108	6									114
13				36										36
14			48											48
15			18											18
17		12												12
20	1													1
total	1	12	66	220	495	792	924	792	495	220	66	12	1	4096

Table 1: Number of restricted domains by the number of SWFs (row) and the number of remaining profiles in P_ks and Q_ks (column).

sp	0	1	2	3	4	5	6	7	8	9	10	11	12	total
2							2	12	30	64	48	12	1	169
3									114	120	18			252
4								144	255	36				435
5							62	300	90					452
6						12	150	252	6					420
7							294	72						366
8						120	242	12						374
9						132	78							210
10					18	192	72							282
11					36	48								84
12					57	108	18							183
13					30	48	6							84
14				4	36	72								112
15					36	12								48
16					69	24								93
17				12	36									48
18					72									72
19					12	24								36
20				36	12									48
21				12										12
22				36	36									72
23				12										12
25					30									30
26				12										12
28				24	3									27
29					6									6
30					6									6
31				24										24
34				12										12
35				12										12
37			12											12
38			6	12										18
40			6	12										18
41			12											12
46			6											6
48			12											12
50			6											6
74			6											6
88		12												12
196	1													1
total	1	12	66	220	495	792	924	792	495	220	66	12	1	4096

Table 2: Number of restricted domains by the number of strategy-proof non-imposed SCFs (row) and the number of remaining profiles in P_ks and Q_ks (column).

	swf																
scf	2	3	4	5	6	7	8	9	10	11	12	13	14	15	17	20	total
2	169																169
3	24	228															252
4	6	84	345														435
5		6	144	302													452
6		24	24	168	204												420
7				36	192	138											366
8			24	36	78	168	68										374
9				12	12	60	96	30									210
10				36	36	18	120	60	12								282
11						24	12	24	24								84
12					30	36	30	24	48	12	3						183
13			6			24	24		12	18							84
14						24	48		30	10							112
15							12	24		12							48
16					12		18	48	12		3						93
17								12	24			12					48
18									60	12							72
19					12	12			12								36
20								6	6	24	12						48
21											12						12
22							12	24			24	12					72
23												12					12
25								24	6								30
26											12						12
28									3	24							27
29								6									6
30							6										6
31											24						24
34											12						12
35											12						12
37													12				12
38										12				6			18
40										12			6				18
41														12			12
46													6				6
48													12				12
50													6				6
74													6				6
88															12		12
196																1	1
total	199	342	543	590	576	504	446	282	249	136	114	36	48	18	12	1	4096

Table 3: Cross analysis between SWFs and strategy-proof non-imposed SCFs.