The Incompatibility of *SD*-Efficiency and *SD*-Strategy-Proofness

Manuel Eberl

August 16, 2018

Abstract

This formalisation contains the proof that there is no anonymous and neutral Social Decision Scheme for at least four voters and alternatives that fulfils both *SD*-Efficiency and *SD*-Strategy-Proofness. The proof is a fully structured and quasi-human-redable one. It was derived from the (unstructured) SMT proof of the case for exactly four voters and alternatives by Brandl *et al.* [1].

Their proof relies on an unverified translation of the original problem to SMT, and the proof that lifts the argument for exactly four voters and alternatives to the general case is also not machine-checked.

In this Isabelle proof, on the other hand, all of these steps are also fully proven and machine-checked. This is particularly important seeing as a previously published informal proof of a weaker statement contained a mistake in precisely this lifting step. [2]

Contents

1 Incompatibility of SD-Efficiency and SD-Strategy-Proofness 2

1.1 Preliminary Definitions 2

1.2 Definition of Preference Profiles and Fact Gathering 3

1.3 Main Proof 6

1.4 Lifting to more than 4 agents and alternatives 17

1 Incompatibility of SD-Efficiency and SD-Strategy-Proofness

theory *SDS-Impossibility*
imports
Randomised-Social-Choice.SDS-Automation
Randomised-Social-Choice.Randomised-Social-Choice
begin

1.1 Preliminary Definitions

locale *sds-impossibility* =
anonymous-sds agents alts sds +
neutral-sds agents alts sds +
sd-efficient-sds agents alts sds +
strategyproof-sds agents alts sds
for *agents* :: *'agent set* **and** *alts* :: *'alt set* **and** *sds* +
assumes *agents-ge-4*: *card agents* ≥ *4*
and *alts-ge-4*: *card alts* ≥ *4*

locale *sds-impossibility-4-4* = *sds-impossibility agents alts sds*
for *agents* :: *'agent set* **and** *alts* :: *'alt set* **and** *sds* +
fixes *A1 A2 A3 A4* :: *'agent* **and** *a b c d* :: *'alt*
assumes *distinct-agents*: *distinct* [*A1*, *A2*, *A3*, *A4*]
and *distinct-alts*: *distinct* [*a*, *b*, *c*, *d*]
and *agents*: *agents* = {*A1*, *A2*, *A3*, *A4*}
and *alts*: *alts* = {*a*, *b*, *c*, *d*}
begin

lemma *an-sds*: *an-sds agents alts sds* **by** *unfold-locales*
lemma *ex-post-efficient-sds*: *ex-post-efficient-sds agents alts sds* **by** *unfold-locales*
lemma *sd-efficient-sds*: *sd-efficient-sds agents alts sds* **by** *unfold-locales*
lemma *strategyproof-an-sds*: *strategyproof-an-sds agents alts sds* **by** *unfold-locales*

lemma *distinct-agents'* [*simp*]:
A1 ≠ *A2* *A1* ≠ *A3* *A1* ≠ *A4* *A2* ≠ *A1* *A2* ≠ *A3* *A2* ≠ *A4*
A3 ≠ *A1* *A3* ≠ *A2* *A3* ≠ *A4* *A4* ≠ *A1* *A4* ≠ *A2* *A4* ≠ *A3*
using *distinct-agents* **by** *auto*

lemma *distinct-alts'* [*simp*]:
a ≠ *b* *a* ≠ *c* *a* ≠ *d* *b* ≠ *a* *b* ≠ *c* *b* ≠ *d*
c ≠ *a* *c* ≠ *b* *c* ≠ *d* *d* ≠ *a* *d* ≠ *b* *d* ≠ *c*
using *distinct-alts* **by** *auto*

lemma *card-agents* [*simp*]: *card agents* = *4* **and** *card-alts* [*simp*]: *card alts* = *4*
using *distinct-agents distinct-alts* **by** (*simp-all add*: *agents alts*)

lemma *in-agents* [*simp*]: *A1* ∈ *agents* *A2* ∈ *agents* *A3* ∈ *agents* *A4* ∈ *agents*
by (*simp-all add*: *agents*)

lemma *in-alts* [*simp*]: $a \in alts$ $b \in alts$ $c \in alts$ $d \in alts$
  by (*simp-all add*: *alts*)

lemma *agent-iff*: $x \in agents \longleftrightarrow x \in \{A1, A2, A3, A4\}$
$(\forall x{\in}agents.\ P\ x) \longleftrightarrow P\ A1 \wedge P\ A2 \wedge P\ A3 \wedge P\ A4$
$(\exists x{\in}agents.\ P\ x) \longleftrightarrow P\ A1 \vee P\ A2 \vee P\ A3 \vee P\ A4$
  by (*auto simp add*: *agents*)

lemma *alt-iff*: $x \in alts \longleftrightarrow x \in \{a,b,c,d\}$
$(\forall x{\in}alts.\ P\ x) \longleftrightarrow P\ a \wedge P\ b \wedge P\ c \wedge P\ d$
$(\exists x{\in}alts.\ P\ x) \longleftrightarrow P\ a \vee P\ b \vee P\ c \vee P\ d$
  by (*auto simp add*: *alts*)

1.2 Definition of Preference Profiles and Fact Gathering

preference-profile
  agents: *agents*
  alts: *alts*
where *R1* = *A1*: [*c*, *d*], [*a*, *b*]   *A2*: [*b*, *d*], *a*, *c*   *A3*: *a*, *b*, [*c*, *d*]   *A4*: [*a*, *c*], [*b*, *d*]
  and *R2* = *A1*: [*a*, *c*], [*b*, *d*]   *A2*: [*c*, *d*], *a*, *b*   *A3*: [*b*, *d*], *a*, *c*   *A4*: *a*, *b*, [*c*, *d*]
  and *R3* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*c*, *d*], [*a*, *b*]   *A3*: *d*, [*a*, *b*], *c*   *A4*: *c*, *a*, [*b*, *d*]
  and *R4* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*a*, *d*], [*b*, *c*]   *A3*: *c*, [*a*, *b*], *d*   *A4*: *d*, *c*, [*a*, *b*]
  and *R5* = *A1*: [*c*, *d*], [*a*, *b*]   *A2*: [*a*, *b*], [*c*, *d*]   *A3*: [*a*, *c*], *d*, *b*   *A4*: *d*, [*a*, *b*], *c*
  and *R6* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*c*, *d*], [*a*, *b*]   *A3*: [*a*, *c*], [*b*, *d*]   *A4*: *d*, *b*, *a*, *c*
  and *R7* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*c*, *d*], [*a*, *b*]   *A3*: *a*, *c*, *d*, *b*   *A4*: *d*, [*a*, *b*], *c*
  and *R8* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*a*, *c*], [*b*, *d*]   *A3*: *d*, [*a*, *b*], *c*   *A4*: *d*, *c*, [*a*, *b*]
  and *R9* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*a*, *d*], *c*, *b*   *A3*: *d*, *c*, [*a*, *b*]   *A4*: [*a*, *b*, *c*], *d*
  and *R10* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*c*, *d*], [*a*, *b*]   *A3*: [*a*, *c*], *d*, *b*   *A4*: [*b*, *d*], *a*, *c*
  and *R11* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*c*, *d*], [*a*, *b*]   *A3*: *d*, [*a*, *b*], *c*   *A4*: *c*, *a*, *b*, *d*
  and *R12* = *A1*: [*c*, *d*], [*a*, *b*]   *A2*: [*a*, *b*], [*c*, *d*]   *A3*: [*a*, *c*], *d*, *b*   *A4*: [*a*, *b*, *d*], *c*
  and *R13* = *A1*: [*a*, *c*], [*b*, *d*]   *A2*: [*c*, *d*], *a*, *b*   *A3*: [*b*, *d*], *a*, *c*   *A4*: *a*, *b*, *d*, *c*
  and *R14* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: *d*, *c*, [*a*, *b*]   *A3*: [*a*, *b*, *c*], *d*   *A4*: *a*, *d*, *c*, *b*
  and *R15* = *A1*: [*a*, *b*], [*c*, *d*]   *A2*: [*c*, *d*], [*a*, *b*]   *A3*: [*b*, *d*], *a*, *c*   *A4*: *a*, *c*, *d*, *b*

and $R16$ = $A1$: [a, b], [c, d] $A2$: [c, d], [a, b] $A3$: a, c, d, b $A4$: [a, b, d], c
and $R17$ = $A1$: [a, b], [c, d] $A2$: [c, d], [a, b] $A3$: [a, c], [b, d] $A4$: d, [a, b], c
and $R18$ = $A1$: [a, b], [c, d] $A2$: [a, d], [b, c] $A3$: [a, b, c], d $A4$: d, c, [a, b]
and $R19$ = $A1$: [a, b], [c, d] $A2$: [c, d], [a, b] $A3$: [b, d], a, c $A4$: [a, c], [b, d]
and $R20$ = $A1$: [b, d], a, c $A2$: b, a, [c, d] $A3$: a, c, [b, d] $A4$: d, c, [a, b]
and $R21$ = $A1$: [a, d], c, b $A2$: d, c, [a, b] $A3$: c, [a, b], d $A4$: a, b, [c, d]
and $R22$ = $A1$: [a, c], d, b $A2$: d, c, [a, b] $A3$: d, [a, b], c $A4$: a, b, [c, d]
and $R23$ = $A1$: [a, b], [c, d] $A2$: [c, d], [a, b] $A3$: [a, c], [b, d] $A4$: [a, b, d], c
and $R24$ = $A1$: [c, d], [a, b] $A2$: d, b, a, c $A3$: c, a, [b, d] $A4$: b, a, [c, d]
and $R25$ = $A1$: [c, d], [a, b] $A2$: [b, d], a, c $A3$: a, b, [c, d] $A4$: a, c, [b, d]
and $R26$ = $A1$: [b, d], [a, c] $A2$: [c, d], [a, b] $A3$: a, b, [c, d] $A4$: a, c, [b, d]
and $R27$ = $A1$: [a, b], [c, d] $A2$: [b, d], a, c $A3$: [a, c], [b, d] $A4$: [c, d], a, b
and $R28$ = $A1$: [c, d], a, b $A2$: [b, d], a, c $A3$: a, b, [c, d] $A4$: a, c, [b, d]
and $R29$ = $A1$: [a, c], d, b $A2$: [b, d], a, c $A3$: a, b, [c, d] $A4$: d, c, [a, b]
and $R30$ = $A1$: [a, d], c, b $A2$: d, c, [a, b] $A3$: c, [a, b], d $A4$: [a, b], d, c
and $R31$ = $A1$: [b, d], a, c $A2$: [a, c], d, b $A3$: c, d, [a, b] $A4$: [a, b], c, d
and $R32$ = $A1$: [a, c], d, b $A2$: d, c, [a, b] $A3$: d, [a, b], c $A4$: [a, b], d, c
and $R33$ = $A1$: [c, d], [a, b] $A2$: [a, c], d, b $A3$: a, b, [c, d] $A4$: d, [a, b], c
and $R34$ = $A1$: [a, b], [c, d] $A2$: a, c, d, b $A3$: b, [a, d], c $A4$: c, d, [a, b]
and $R35$ = $A1$: [a, d], c, b $A2$: a, b, [c, d] $A3$: [a, b, c], d $A4$: d, c, [a, b]
and $R36$ = $A1$: [c, d], [a, b] $A2$: [a, c], d, b $A3$: [b, d], a, c $A4$: a, b, [c, d]
and $R37$ = $A1$: [a, c], [b, d] $A2$: [b, d], [a, c] $A3$: a, b, [c, d] $A4$: c, d, [a, b]
and $R38$ = $A1$: [c, d], a, b $A2$: [b, d], a, c $A3$: a, b, [c, d] $A4$: [a, c], b, d
and $R39$ = $A1$: [a, c], d, b $A2$: [b, d], a, c $A3$: a, b, [c, d] $A4$: [c, d], a, b
and $R40$ = $A1$: [a, d], c, b $A2$: [a, b], c, d $A3$: [a, b, c], d $A4$: d,

c, [*a*, *b*]
and *R41* = *A1*: [*a*, *d*], *c*, *b* *A2*: [*a*, *b*], *d*, *c* *A3*: [*a*, *b*, *c*], *d* *A4*: *d*, *c*, [*a*, *b*]
and *R42* = *A1*: [*c*, *d*], [*a*, *b*] *A2*: [*a*, *b*], [*c*, *d*] *A3*: *d*, *b*, *a*, *c* *A4*: *c*, *a*, [*b*, *d*]
and *R43* = *A1*: [*a*, *b*], [*c*, *d*] *A2*: [*c*, *d*], [*a*, *b*] *A3*: *d*, [*a*, *b*], *c* *A4*: *a*, [*c*, *d*], *b*
and *R44* = *A1*: [*c*, *d*], [*a*, *b*] *A2*: [*a*, *c*], *d*, *b* *A3*: [*a*, *b*], *d*, *c* *A4*: [*a*, *b*, *d*], *c*
and *R45* = *A1*: [*a*, *c*], *d*, *b* *A2*: [*b*, *d*], *a*, *c* *A3*: [*a*, *b*], *c*, *d* *A4*: [*c*, *d*], *b*, *a*
and *R46* = *A1*: [*b*, *d*], *a*, *c* *A2*: *d*, *c*, [*a*, *b*] *A3*: [*a*, *c*], [*b*, *d*] *A4*: *b*, *a*, [*c*, *d*]
and *R47* = *A1*: [*a*, *b*], [*c*, *d*] *A2*: [*a*, *d*], *c*, *b* *A3*: *d*, *c*, [*a*, *b*] *A4*: *c*, [*a*, *b*], *d*
by (*simp-all add*: *agents alts*)

derive-orbit-equations (*an-sds*)
R10 R26 R27 R28 R29 R43 R45
by *simp-all*

prove-inefficient-supports (*ex-post-efficient-sds sd-efficient-sds*)
R3 [*b*] **and** *R4* [*b*] **and** *R5* [*b*] **and** *R7* [*b*] **and** *R8* [*b*] **and**
R9 [*b*] **and** *R11* [*b*] **and** *R12* [*b*] **and** *R14* [*b*] **and** *R16* [*b*] **and**
R17 [*b*] **and** *R18* [*b*] **and** *R21* [*b*] **and** *R22* [*b*] **and** *R23* [*b*] **and**
R30 [*b*] **and** *R32* [*b*] **and** *R33* [*b*] **and** *R35* [*b*] **and** *R40* [*b*] **and**
R41 [*b*] **and** *R43* [*b*] **and** *R44* [*b*] **and** *R47* [*b*] **and**
R10 [*c*, *b*] *witness*: [*a*: *1* / *2*, *b*: *0*, *c*: *0*, *d*: *1* / *2*] **and**
R15 [*c*, *b*] *witness*: [*a*: *1* / *2*, *b*: *0*, *c*: *0*, *d*: *1* / *2*] **and**
R19 [*c*, *b*] *witness*: [*a*: *1* / *2*, *b*: *0*, *c*: *0*, *d*: *1* / *2*] **and**
R25 [*b*, *c*] *witness*: [*c*: *0*, *d*: *1* / *2*, *a*: *1* / *2*, *b*: *0*] **and**
R26 [*c*, *b*] *witness*: [*b*: *0*, *d*: *1* / *2*, *a*: *1* / *2*, *c*: *0*] **and**
R27 [*c*, *b*] *witness*: [*a*: *1* / *2*, *b*: *0*, *c*: *0*, *d*: *1* / *2*] **and**
R28 [*b*, *c*] *witness*: [*c*: *0*, *d*: *1* / *2*, *a*: *1* / *2*, *b*: *0*] **and**
R29 [*b*, *c*] *witness*: [*a*: *1* / *2*, *c*: *0*, *d*: *1* / *2*, *b*: *0*] **and**
R39 [*b*, *c*] *witness*: [*a*: *1* / *2*, *c*: *0*, *d*: *1* / *2*, *b*: *0*]
by (*simp-all add*: *agent-iff alt-iff*)

derive-strategyproofness-conditions (*strategyproof-an-sds*)
distance: *2*
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16 R17 R18 R19 R20
R21 R22 R23 R24 R25 R26 R27 R28 R29 R30 R31 R32 R33 R34 R35 R36 R37 R38 R39 R40
R41 R42 R43 R44 R45 R46 R47
by (*simp-all add*: *agent-iff alt-iff*)

lemma *lottery-conditions*:
assumes *is-pref-profile R*

shows *pmf (sds R) a ≥ 0 pmf (sds R) b ≥ 0 pmf (sds R) c ≥ 0 pmf (sds R) d ≥ 0*
pmf (sds R) a + pmf (sds R) b + pmf (sds R) c + pmf (sds R) d = 1
using *lottery-prob-alts[OF sds-wf[OF assms]]*
by (*simp-all add*: *alts pmf-nonneg measure-measure-pmf-finite*)

1.3 Main Proof

lemma *R45* [*simp*]: *pmf (sds R45) a = 1/4 pmf (sds R45) b = 1/4*
pmf (sds R45) c = 1/4 pmf (sds R45) d = 1/4
using *R45.orbits lottery-conditions[OF R45.wf]* **by** *simp-all*

lemma *R10-bc* [*simp*]: *pmf (sds R10) b = 0 pmf (sds R10) c = 0*
using *R10.support R10.orbits* **by** *auto*

lemma *R10-ad* [*simp*]: *pmf (sds R10) a = 1/2 pmf (sds R10) d = 1/2*
using *lottery-conditions[OF R10.wf] R10-bc R10.orbits* **by** *simp-all*

lemma *R26-bc* [*simp*]: *pmf (sds R26) b = 0 pmf (sds R26) c = 0*
using *R26.support R26.orbits* **by** *auto*

lemma *R26-d* [*simp*]: *pmf (sds R26) d = 1 − pmf (sds R26) a*
using *lottery-conditions[OF R26.wf] R26-bc* **by** *simp*

lemma *R27-bc* [*simp*]: *pmf (sds R27) b = 0 pmf (sds R27) c = 0*
using *R27.support R27.orbits* **by** *auto*

lemma *R27-d* [*simp*]: *pmf (sds R27) d = 1 − pmf (sds R27) a*
using *lottery-conditions[OF R27.wf] R27-bc* **by** *simp*

lemma *R28-bc* [*simp*]: *pmf (sds R28) b = 0 pmf (sds R28) c = 0*
using *R28.support R28.orbits* **by** *auto*

lemma *R28-d* [*simp*]: *pmf (sds R28) d = 1 − pmf (sds R28) a*
using *lottery-conditions[OF R28.wf] R28-bc* **by** *simp*

lemma *R29-bc* [*simp*]: *pmf (sds R29) b = 0 pmf (sds R29) c = 0*
using *R29.support R29.orbits* **by** *auto*

lemma *R29-ac* [*simp*]: *pmf (sds R29) a = 1/2 pmf (sds R29) d = 1/2*
using *lottery-conditions[OF R29.wf] R29-bc R29.orbits* **by** *simp-all*

lemmas *R43-bc* [*simp*] = *R43.support*

lemma *R43-ad* [*simp*]: *pmf* (*sds R43*) *a* = *1/2 pmf* (*sds R43*) *d* = *1/2*
  using *lottery-conditions*[*OF R43.wf*] *R43-bc R43.orbits* **by** *simp-all*

lemma *R39-b* [*simp*]: *pmf* (*sds R39*) *b* = *0*
proof −
  {
    assume [*simp*]: *pmf* (*sds R39*) *c* = *0*
    with *R29-R39.strategyproofness*(*1*)
      have *pmf* (*sds R39*) *d* ≤ *1/2* **by** *auto*
    with *R39-R29.strategyproofness*(*1*) *lottery-conditions*[*OF R39.wf*]
      have *pmf* (*sds R39*) *b* = *0* **by** *auto*
  }
  with *R39.support* **show** *?thesis* **by** *blast*
qed

lemma *R36-a* [*simp*]: *pmf* (*sds R36*) *a* = *1/2* **and** *R36-b* [*simp*]: *pmf* (*sds R36*) *b* = *0*
proof −
  from *R10-R36.strategyproofness*(*1*) *lottery-conditions*[*OF R36.wf*]
    have *pmf* (*sds R36*) *a* + *pmf* (*sds R36*) *b* ≤ *1/2* **by** *auto*
  with *R36-R10.strategyproofness*(*1*) *lottery-conditions*[*OF R36.wf*]
    show *pmf* (*sds R36*) *a* = *1/2 pmf* (*sds R36*) *b* = *0* **by** *auto*
qed

lemma *R36-d* [*simp*]: *pmf* (*sds R36*) *d* = *1/2* − *pmf* (*sds R36*) *c*
  using *lottery-conditions*[*OF R36.wf*] **by** *simp*

lemma *R39-a* [*simp*]: *pmf* (*sds R39*) *a* = *1/2*
proof −
  from *R36-R39.strategyproofness*(*1*) *lottery-conditions*[*OF R39.wf*]
    have *pmf* (*sds R39*) *a* ≥ *1/2* **by** *auto*
  with *R39-R36.strategyproofness*(*1*) *lottery-conditions*[*OF R39.wf*]
    show *?thesis* **by** *auto*
qed

lemma *R39-d* [*simp*]: *pmf* (*sds R39*) *d* = *1/2* − *pmf* (*sds R39*) *c*
  using *lottery-conditions*[*OF R39.wf*] **by** *simp*

lemmas *R12-b* [*simp*] = *R12.support*

lemma *R12-c* [*simp*]: *pmf* (*sds R12*) *c* = *0*
  using *R12-R10.strategyproofness*(*1*) *lottery-conditions*[*OF R12.wf*]
  by (*auto simp del*: *pmf-nonneg*)

lemma *R12-d* [*simp*]: *pmf* (*sds R12*) *d* = *1* − *pmf* (*sds R12*) *a*

using *lottery-conditions*[*OF R12.wf*] **by** *simp*

lemma *R12-a-ge-one-half*: *pmf* (*sds R12*) *a* ≥ *1/2*
using *R10-R12.strategyproofness*(*1*) *lottery-conditions*[*OF R12.wf*]
by *auto*

lemma *R44* [*simp*]:
pmf (*sds R44*) *a* = *pmf* (*sds R12*) *a pmf* (*sds R44*) *d* = *1* − *pmf* (*sds R12*) *a*
pmf (*sds R44*) *b* = *0 pmf* (*sds R44*) *c* = *0*
proof −
from *R12-R44.strategyproofness*(*1*) *R44.support* **have** *pmf* (*sds R44*) *a* ≤ *pmf* (*sds R12*) *a* **by** *simp*
with *R44-R12.strategyproofness*(*1*) *R44.support lottery-conditions*[*OF R44.wf*]
show *pmf* (*sds R44*) *a* = *pmf* (*sds R12*) *a pmf* (*sds R44*) *c* = *0*
pmf (*sds R44*) *d* = *1* − *pmf* (*sds R12*) *a* **by** (*auto simp del*: *pmf-nonneg*)
qed (*insert R44.support*, *simp-all*)

lemma *R9-a* [*simp*]: *pmf* (*sds R9*) *a* = *pmf* (*sds R35*) *a*
proof −
from *R9-R35.strategyproofness*(*1*) *R35.support R9.support*
have *pmf* (*sds R35*) *a* ≤ *pmf* (*sds R9*) *a* **by** *simp*
with *R35-R9.strategyproofness*(*1*) *R9.support R35.support* **show** *?thesis* **by** *simp*
qed

lemma *R18-c* [*simp*]: *pmf* (*sds R18*) *c* = *pmf* (*sds R9*) *c*
proof −
from *R18-R9.strategyproofness*(*1*) *R18.support R9.support*
have *pmf* (*sds R18*) *d* + *pmf* (*sds R18*) *a* ≥ *pmf* (*sds R9*) *d* + *pmf* (*sds R9*) *a* **by** *auto*
with *R9-R18.strategyproofness*(*1*) *R18.support R9.support*
lottery-conditions[*OF R9.wf*] *lottery-conditions*[*OF R18.wf*]
show *?thesis* **by** *auto*
qed

lemma *R5-d-ge-one-half*: *pmf* (*sds R5*) *d* ≥ *1/2*
using *R5-R10.strategyproofness*(*1*) *R5.support lottery-conditions*[*OF R5.wf*] **by** *auto*

lemma *R7* [*simp*]: *pmf* (*sds R7*) *a* = *1/2 pmf* (*sds R7*) *b* = *0 pmf* (*sds R7*) *c* = *0 pmf* (*sds R7*) *d* = *1/2*
proof −
from *R5-d-ge-one-half* **have** *1/2* ≤ *pmf* (*sds R5*) *d* **by** *simp*
also from *R5-R17.strategyproofness*(*1*) *R17.support lottery-conditions*[*OF R5.wf*] *lottery-conditions*[*OF R17.wf*]
have … ≤ *pmf* (*sds R17*) *d* **by** (*auto simp del*: *pmf-nonneg*)
also from *R17-R7.strategyproofness*(*1*) *lottery-conditions*[*OF R7.wf*] *lottery-conditions*[*OF R17.wf*] *R7.support*

have *pmf* (*sds R17*) *d* ≤ *pmf* (*sds R7*) *d* **by** (*auto simp del*: *pmf-nonneg*)
finally have *pmf* (*sds R7*) *d* ≥ *1/2* **.**
with *R7-R43.strategyproofness*(*1*) *lottery-conditions*[*OF R7.wf*] *R7.support*
show *pmf* (*sds R7*) *a* = *1/2* *pmf* (*sds R7*) *b* = *0* *pmf* (*sds R7*) *c* = *0* *pmf* (*sds R7*) *d* = *1/2*
by *auto*
qed

lemma *R5* [*simp*]: *pmf* (*sds R5*) *a* = *1/2* *pmf* (*sds R5*) *b* = *0* *pmf* (*sds R5*) *c* = *0* *pmf* (*sds R5*) *d* = *1/2*
proof −
from *R5-R7.strategyproofness*(*1*) *lottery-conditions*[*OF R5.wf*] *R5.support*
have *pmf* (*sds R5*) *d* ≤ *1/2* **by** *auto*
with *R5-d-ge-one-half* **show** *d*: *pmf* (*sds R5*) *d* = *1* / *2* **by** *simp*
with *R5-R10.strategyproofness*(*1*) *lottery-conditions*[*OF R5.wf*] *R5.support*
show *pmf* (*sds R5*) *c* = *0* *pmf* (*sds R5*) *a* = *1/2* **by** *simp-all*
qed (*simp-all add*: *R5.support*)

lemma *R15* [*simp*]: *pmf* (*sds R15*) *a* = *1/2* *pmf* (*sds R15*) *b* = *0* *pmf* (*sds R15*) *c* = *0* *pmf* (*sds R15*) *d* = *1/2*
proof −
{
assume *pmf* (*sds R15*) *b* = *0*
with *R10-R15.strategyproofness*(*1*) *lottery-conditions*[*OF R15.wf*]
have *pmf* (*sds R15*) *a* + *pmf* (*sds R15*) *c* ≤ *1/2* **by** *auto*
with *R15-R10.strategyproofness*(*1*) *lottery-conditions*[*OF R15.wf*]
have *pmf* (*sds R15*) *c* = *0* **by** *auto*
}
with *R15.support* **show** [*simp*]: *pmf* (*sds R15*) *c* = *0* **by** *blast*
with *R15-R5.strategyproofness*(*1*) *lottery-conditions*[*OF R15.wf*]
have *pmf* (*sds R15*) *a* ≥ *1/2* **by** *auto*
moreover from *R15-R7.strategyproofness*(*1*) *lottery-conditions*[*OF R15.wf*]
have *pmf* (*sds R15*) *b* + *pmf* (*sds R15*) *d* ≥ *1/2* **by** *auto*
ultimately show *pmf* (*sds R15*) *a* = *1/2* **using** *lottery-conditions*[*OF R15.wf*] **by** *auto*
with *R15-R5.strategyproofness*(*1*) *lottery-conditions*[*OF R15.wf*]
show *pmf* (*sds R15*) *d* = *1/2* *pmf* (*sds R15*) *b* = *0* **by** *auto*
qed

lemma *R13-aux*: *pmf* (*sds R13*) *b* = *0* *pmf* (*sds R13*) *c* = *0* *pmf* (*sds R13*) *d* = *1* − *pmf* (*sds R13*) *a*
and *R27-R13* [*simp*]: *pmf* (*sds R27*) *a* = *pmf* (*sds R13*) *a*
using *R27-R13.strategyproofness*(*1*) *R13-R27.strategyproofness*(*1*) *lottery-conditions*[*OF R13.wf*] **by** *auto*

lemma *R13* [*simp*]: *pmf* (*sds R13*) *a* = *1/2* *pmf* (*sds R13*) *b* = *0* *pmf* (*sds R13*) *c* = *0* *pmf* (*sds R13*) *d* = *1/2*
using *R15-R13.strategyproofness*(*1*) *R13-R15.strategyproofness*(*1*) *R13-aux* **by** *simp-all*

lemma *R27* [*simp*]: *pmf* (*sds R27*) *a* = *1/2* *pmf* (*sds R27*) *b* = *0* *pmf* (*sds R27*) *c* = *0* *pmf* (*sds R27*) *d* = *1/2*
  by *simp-all*

lemma *R19* [*simp*]: *pmf* (*sds R19*) *a* = *1/2* *pmf* (*sds R19*) *b* = *0* *pmf* (*sds R19*) *c* = *0* *pmf* (*sds R19*) *d* = *1/2*
proof −
  have *pmf* (*sds R19*) *a* = *1/2* ∧ *pmf* (*sds R19*) *b* = *0* ∧ *pmf* (*sds R19*) *c* = *0* ∧ *pmf* (*sds R19*) *d* = *1/2*
  proof (*rule disjE*[*OF R19.support*]; *safe*)
    assume [*simp*]: *pmf* (*sds R19*) *b* = *0*
    from *R10-R19.strategyproofness*(*1*) *lottery-conditions*[*OF R19.wf*]
      have *pmf* (*sds R19*) *a* + *pmf* (*sds R19*) *c* ≤ *1/2* **by** *auto*
    moreover from *R19-R10.strategyproofness*(*1*)
      have *pmf* (*sds R19*) *a* + *pmf* (*sds R19*) *c* ≥ *1/2* **by** *simp*
   ultimately show *pmf* (*sds R19*) *d* = *1/2* **using** *lottery-conditions*[*OF R19.wf*] **by** *simp*
    with *R27-R19.strategyproofness*(*1*) *lottery-conditions*[*OF R19.wf*]
      show *pmf* (*sds R19*) *a* = *1/2* *pmf* (*sds R19*) *c* = *0* **by** *auto*
  next
    assume [*simp*]: *pmf* (*sds R19*) *c* = *0*
    from *R19-R10.strategyproofness*(*1*) **have** *pmf* (*sds R19*) *a* ≥ *1/2* **by** *auto*
    moreover from *R19-R27.strategyproofness*(*1*) **have** *pmf* (*sds R19*) *d* ≥ *1/2* **by** *auto*
    ultimately show *pmf* (*sds R19*) *a* = *1/2* *pmf* (*sds R19*) *d* = *1/2* *pmf* (*sds R19*) *b* = *0*
      using *lottery-conditions*[*OF R19.wf*] **by** (*auto simp del*: *pmf-nonneg*)
  qed
  thus *pmf* (*sds R19*) *a* = *1/2* *pmf* (*sds R19*) *b* = *0* *pmf* (*sds R19*) *c* = *0* *pmf* (*sds R19*) *d* = *1/2*
    by *blast*+
qed

lemma *R1* [*simp*]: *pmf* (*sds R1*) *a* = *1/2* *pmf* (*sds R1*) *b* = *0*
proof −
  from *R19-R1.strategyproofness*(*1*) *lottery-conditions*[*OF R1.wf*]
    have *pmf* (*sds R1*) *a* + *pmf* (*sds R1*) *b* ≤ *1/2* **by** *simp*
  with *R1-R19.strategyproofness*(*1*) *lottery-conditions*[*OF R1.wf*]
    show *pmf* (*sds R1*) *a* = *1/2* *pmf* (*sds R1*) *b* = *0* **by** *auto*
qed

lemma *R22* [*simp*]: *pmf* (*sds R22*) *a* = *1/2* *pmf* (*sds R22*) *b* = *0* *pmf* (*sds R22*) *c* = *0* *pmf* (*sds R22*) *d* = *1/2*
proof −
  from *R33-R5.strategyproofness*(*1*) *R33.support*
    have *1/2* ≤ *pmf* (*sds R33*) *a* **by** *auto*
  also from *R33-R22.strategyproofness*(*1*) *R22.support R33.support*
    lottery-conditions[*OF R22.wf*] *lottery-conditions*[*OF R33.wf*]

have . . . ≤ *pmf* (*sds R22*) *a* **by** *simp*
finally show *pmf* (*sds R22*) *a* = *1/2 pmf* (*sds R22*) *b* = *0 pmf* (*sds R22*) *c* = *0 pmf* (*sds R22*) *d* = *1/2*
using *R22-R29.strategyproofness*(*1*) *lottery-conditions*[*OF R22.wf*] **by** (*auto simp del*: *pmf-nonneg*)
qed

lemma *R28* [*simp*]: *pmf* (*sds R28*) *a* = *1/2 pmf* (*sds R28*) *b* = *0 pmf* (*sds R28*) *c* = *0 pmf* (*sds R28*) *d* = *1/2*
proof −
have *pmf* (*sds R28*) *a* ≤ *pmf* (*sds R32*) *d*
using *R32-R28.strategyproofness*(*1*) *lottery-conditions*[*OF R32.wf*] **by** *auto*
hence *R32-d*: *pmf* (*sds R32*) *d* = *pmf* (*sds R28*) *a*
using *R28-R32.strategyproofness*(*1*) *lottery-conditions*[*OF R32.wf*] **by** *auto*

from *R22-R32.strategyproofness*(*1*) *lottery-conditions*[*OF R32.wf*] *R32.support*

have *pmf* (*sds R32*) *a* ≤ *1/2* **by** *auto*
with *R32-R22.strategyproofness*(*1*) *lottery-conditions*[*OF R32.wf*] *R32.support*
show *pmf* (*sds R28*) *a* = *1/2 pmf* (*sds R28*) *b* = *0 pmf* (*sds R28*) *c* = *0 pmf* (*sds R28*) *d* = *1/2*
by (*auto simp*: *R32-d simp del*: *pmf-nonneg*)
qed

lemma *R39* [*simp*]: *pmf* (*sds R39*) *a* = *1/2 pmf* (*sds R39*) *b* = *0 pmf* (*sds R39*) *c* = *0 pmf* (*sds R39*) *d* = *1/2*
proof −
from *R28-R39.strategyproofness*(*1*) **show** *pmf* (*sds R39*) *c* = *0* **by** *simp*
thus *pmf* (*sds R39*) *a* = *1/2 pmf* (*sds R39*) *b* = *0 pmf* (*sds R39*) *d* = *1/2*
by *simp-all*
qed

lemma *R2* [*simp*]: *pmf* (*sds R2*) *a* = *1/2 pmf* (*sds R2*) *b* = *0 pmf* (*sds R2*) *c* = *0 pmf* (*sds R2*) *d* = *1/2*
proof −
from *R1-R2.strategyproofness*(*1*) *R2-R1.strategyproofness*(*1*) *lottery-conditions*[*OF R2.wf*] *lottery-conditions*[*OF R1.wf*]
have *pmf* (*sds R2*) *a* = *1/2 pmf* (*sds R2*) *c* + *pmf* (*sds R2*) *d* = *1/2*
by (*auto simp*: *algebra-simps simp del*: *pmf-nonneg*)
with *R39-R2.strategyproofness*(*1*) *lottery-conditions*[*OF R2.wf*]
show *pmf* (*sds R2*) *a* = *1/2 pmf* (*sds R2*) *b* = *0 pmf* (*sds R2*) *c* = *0 pmf* (*sds R2*) *d* = *1/2*
by *auto*
qed

lemma *R42* [*simp*]: *pmf* (*sds R42*) *a* = *0 pmf* (*sds R42*) *b* = *0 pmf* (*sds R42*) *c* = *1/2 pmf* (*sds R42*) *d* = *1/2*
proof −
from *R17-R5.strategyproofness*(*1*) *lottery-conditions*[*OF R17.wf*] *R17.support*

have *pmf* (*sds R17*) *d* ≤ *1/2* **by** *auto*
moreover from *R5-R17.strategyproofness*(*1*) *R17.support lottery-conditions*[*OF R17.wf*]
have *pmf* (*sds R17*) *d* ≥ *1/2* **by** *auto*
ultimately have *R17-d*: *pmf* (*sds R17*) *d* = *1/2* **by** *simp*

from *R6-R42.strategyproofness*(*1*)
have *pmf* (*sds R42*) *a* + *pmf* (*sds R42*) *c* ≤ *pmf* (*sds R6*) *a* + *pmf* (*sds R6*) *c* **by** *simp*
also from *R6-R19.strategyproofness*(*1*) *lottery-conditions*[*OF R6.wf*]
have *pmf* (*sds R6*) *a* + *pmf* (*sds R6*) *c* ≤ *1/2* **by** (*auto simp del*: *pmf-nonneg*)
finally have *pmf* (*sds R42*) *a* + *pmf* (*sds R42*) *c* ≤ *1* / *2* **.**
moreover from *R17-R11.strategyproofness*(*1*) *R11.support R17.support lottery-conditions*[*OF R11.wf*] *lottery-conditions*[*OF R17.wf*]
have *pmf* (*sds R11*) *d* ≥ *1/2* **by** (*auto simp*: *R17-d*)
ultimately have *pmf* (*sds R42*) *a* + *pmf* (*sds R42*) *c* ≤ *pmf* (*sds R11*) *d* **by** *simp*
with *R42-R11.strategyproofness*(*1*) *R11.support*
have *E*: *pmf* (*sds R11*) *d* ≤ *pmf* (*sds R42*) *c* **by** *auto*
with ‹*pmf* (*sds R11*) *d* ≥ *1/2*› **have** *pmf* (*sds R42*) *c* ≥ *1/2* **by** *simp*
moreover from *R17-R3.strategyproofness*(*1*) *R3.support R17.support lottery-conditions*[*OF R17.wf*] *lottery-conditions*[*OF R3.wf*]
have *pmf* (*sds R3*) *d* ≥ *1/2* **by** (*auto simp*: *R17-d*)
ultimately show *pmf* (*sds R42*) *a* = *0 pmf* (*sds R42*) *b* = *0 pmf* (*sds R42*) *c* = *1/2 pmf* (*sds R42*) *d* = *1/2*
using *R42-R3.strategyproofness*(*1*) *lottery-conditions*[*OF R3.wf*] *lottery-conditions*[*OF R42.wf*]
by *linarith*+
qed

lemma *R37* [*simp*]: *pmf* (*sds R37*) *a* = *1/2 pmf* (*sds R37*) *b* = *0 pmf* (*sds R37*) *c* = *1/2 pmf* (*sds R37*) *d* = *0*
proof −
from *R37-R42.strategyproofness*(*1*) *lottery-conditions*[*OF R37.wf*]
have *pmf* (*sds R37*) *a* = *1/2* ∨ *pmf* (*sds R37*) *a* + *pmf* (*sds R37*) *b* > *1/2*
by (*auto simp del*: *pmf-nonneg*)
moreover from *R37-R42.strategyproofness*(*2*) *lottery-conditions*[*OF R37.wf*]
have *pmf* (*sds R37*) *c* = *1/2* ∨ *pmf* (*sds R37*) *c* + *pmf* (*sds R37*) *d* > *1/2*
by (*auto simp del*: *pmf-nonneg*)
ultimately show *pmf* (*sds R37*) *a* = *1/2 pmf* (*sds R37*) *b* = *0 pmf* (*sds R37*) *c* = *1/2 pmf* (*sds R37*) *d* = *0*
using *lottery-conditions*[*OF R37.wf*] **by** (*auto simp del*: *pmf-nonneg*)
qed

lemma *R24* [*simp*]: *pmf* (*sds R24*) *a* = *0 pmf* (*sds R24*) *b* = *0 pmf* (*sds R24*) *d* = *1* − *pmf* (*sds R24*) *c*
using *R42-R24.strategyproofness*(*1*) *lottery-conditions*[*OF R24.wf*] **by** (*auto simp del*: *pmf-nonneg*)

lemma *R34* [*simp*]:
pmf (*sds R34*) *a* = *1* − *pmf* (*sds R24*) *c pmf* (*sds R34*) *b* = *pmf* (*sds R24*) *c*
pmf (*sds R34*) *c* = *0 pmf* (*sds R34*) *d* = *0*
proof −
from *R24-R34.strategyproofness*(*1*) *lottery-conditions*[*OF R34.wf*]
have *pmf* (*sds R34*) *b* ≤ *pmf* (*sds R24*) *c* **by** (*auto simp del*: *pmf-nonneg*)
moreover from *R34-R24.strategyproofness*(*1*) *lottery-conditions*[*OF R34.wf*]
have *pmf* (*sds R34*) *b* ≥ *pmf* (*sds R24*) *c* **by** *auto*
ultimately show *bc*: *pmf* (*sds R34*) *b* = *pmf* (*sds R24*) *c* **by** *simp*
from *R34-R24.strategyproofness*(*1*) *bc lottery-conditions*[*OF R34.wf*]
show *pmf* (*sds R34*) *c* = *0* **by** *auto*
moreover from *R24-R34.strategyproofness*(*1*) *bc* **show** *pmf* (*sds R34*) *d* = *0*
by *simp*
ultimately show *pmf* (*sds R34*) *a* = *1* − *pmf* (*sds R24*) *c*
using *bc lottery-conditions*[*OF R34.wf*] **by** *auto*
qed

lemma *R14* [*simp*]: *pmf* (*sds R14*) *b* = *0 pmf* (*sds R14*) *d* = *0 pmf* (*sds R14*) *c*
= *1* − *pmf* (*sds R14*) *a*
using *R14-R34.strategyproofness*(*1*) *R14.support lottery-conditions*[*OF R14.wf*]

by (*auto simp del*: *pmf-nonneg*)

lemma *R46* [*simp*]: *pmf* (*sds R46*) *a* = *0 pmf* (*sds R46*) *c* = *0 pmf* (*sds R46*) *d*
= *1* − *pmf* (*sds R46*) *b*
using *R46-R37.strategyproofness*(*1*) *lottery-conditions*[*OF R46.wf*] **by** *auto*

lemma *R20* [*simp*]: *pmf* (*sds R20*) *a* = *0 pmf* (*sds R20*) *c* = *0 pmf* (*sds R20*) *d*
= *1* − *pmf* (*sds R20*) *b*
using *R46-R20.strategyproofness*(*1*) *lottery-conditions*[*OF R20.wf*] **by** (*auto simp del*: *pmf-nonneg*)

lemma *R21* [*simp*]: *pmf* (*sds R21*) *d* = *1* − *pmf* (*sds R21*) *a pmf* (*sds R21*) *b*
= *0 pmf* (*sds R21*) *c* = *0*
using *R20-R21.strategyproofness*(*1*) *lottery-conditions*[*OF R21.wf*] **by** *auto*

lemma *R16-R12*: *pmf* (*sds R16*) *c* + *pmf* (*sds R16*) *a* ≤ *pmf* (*sds R12*) *a*
using *R12-R16.strategyproofness*(*1*) *R16.support lottery-conditions*[*OF R16.wf*]
by *auto*

lemma *R16* [*simp*]: *pmf* (*sds R16*) *b* = *0 pmf* (*sds R16*) *c* = *0 pmf* (*sds R16*) *d*
= *1* − *pmf* (*sds R16*) *a*
proof −
from *R16-R12* **have** *pmf* (*sds R16*) *c* + *pmf* (*sds R16*) *a* ≤ *pmf* (*sds R12*) *a*
by *simp*
also from *R44-R40.strategyproofness*(*1*) *lottery-conditions*[*OF R40.wf*] *R40.support*
have *pmf* (*sds R12*) *a* ≤ *pmf* (*sds R40*) *a* **by** *auto*
also from *R9-R40.strategyproofness*(*1*) *R9.support R40.support*

have *pmf* (*sds R40*) *a* ≤ *pmf* (*sds R9*) *a* **by** *auto*
finally have *pmf* (*sds R16*) *c* + *pmf* (*sds R16*) *a* ≤ *pmf* (*sds R9*) *a* **by** *simp*
moreover from *R14-R16.strategyproofness*(*1*) *R16.support lottery-conditions*[*OF R16.wf*]
have *pmf* (*sds R16*) *a* ≥ *pmf* (*sds R14*) *a* **by** *auto*
ultimately have *pmf* (*sds R16*) *c* ≤ *pmf* (*sds R9*) *a* − *pmf* (*sds R14*) *a* **by** *simp*
also from *R14-R9.strategyproofness*(*1*) *R9.support lottery-conditions*[*OF R9.wf*]
have *pmf* (*sds R9*) *a* − *pmf* (*sds R14*) *a* ≤ *0* **by** (*auto simp del*: *pmf-nonneg*)
finally show *pmf* (*sds R16*) *b* = *0* *pmf* (*sds R16*) *c* = *0* *pmf* (*sds R16*) *d* = *1* − *pmf* (*sds R16*) *a*
using *lottery-conditions*[*OF R16.wf*] *R16.support* **by** *auto*
qed

lemma *R12-R14*: *pmf* (*sds R14*) *a* ≤ *pmf* (*sds R12*) *a*
using *R14-R16.strategyproofness*(*1*) *R16-R12* **by** *auto*

lemma *R12-a* [*simp*]: *pmf* (*sds R12*) *a* = *pmf* (*sds R9*) *a*
proof −
from *R44-R40.strategyproofness*(*1*) *R40.support lottery-conditions*[*OF R40.wf*]

have *pmf* (*sds R12*) *a* ≤ *pmf* (*sds R40*) *a* **by** *auto*
also from *R9-R40.strategyproofness*(*1*) *R9.support R40.support*
have *pmf* (*sds R40*) *a* ≤ *pmf* (*sds R9*) *a* **by** *auto*
finally have *B*: *pmf* (*sds R12*) *a* ≤ *pmf* (*sds R9*) *a* **by** *simp*
moreover from *R14-R9.strategyproofness*(*1*) *lottery-conditions*[*OF R9.wf*] *R9.support*

have *pmf* (*sds R9*) *a* ≤ *pmf* (*sds R14*) *a* **by** (*auto simp del*: *pmf-nonneg*)
with *R12-R14* **have** *pmf* (*sds R9*) *a* ≤ *pmf* (*sds R12*) *a* **by** *simp*
ultimately show *pmf* (*sds R12*) *a* = *pmf* (*sds R9*) *a* **by** *simp*
qed

lemma *R9* [*simp*]: *pmf* (*sds R9*) *b* = *0* *pmf* (*sds R9*) *d* = *0* *pmf* (*sds R14*) *a* = *pmf* (*sds R35*) *a* *pmf* (*sds R9*) *c* = *1* − *pmf* (*sds R35*) *a*
using *R12-R14* *R14-R9.strategyproofness*(*1*) *lottery-conditions*[*OF R9.wf*] *R9.support*
by *auto*

lemma *R23* [*simp*]: *pmf* (*sds R23*) *b* = *0* *pmf* (*sds R23*) *c* = *0* *pmf* (*sds R23*) *d* = *1* − *pmf* (*sds R23*) *a*
using *R23-R19.strategyproofness*(*1*) *lottery-conditions*[*OF R23.wf*] *R23.support*

by (*auto simp del*: *pmf-nonneg*)

lemma *R35* [*simp*]: *pmf* (*sds R35*) *a* = *pmf* (*sds R21*) *a* *pmf* (*sds R35*) *b* = *0* *pmf* (*sds R35*) *c* = *0* *pmf* (*sds R35*) *d* = *1* − *pmf* (*sds R21*) *a*
proof −
from *R35-R21.strategyproofness*(*1*) *R35.support*
have *pmf* (*sds R21*) *a* ≤ *pmf* (*sds R35*) *a* + *pmf* (*sds R35*) *c* **by** *auto*
with *R21-R35.strategyproofness*(*1*) *R35.support lottery-conditions*[*OF R35.wf*]

show *pmf* (*sds R35*) *a* = *pmf* (*sds R21*) *a* *pmf* (*sds R35*) *b* = *0*
pmf (*sds R35*) *c* = *0* *pmf* (*sds R35*) *d* = *1* − *pmf* (*sds R21*) *a* **by** *simp-all*
qed

lemma *R18* [*simp*]: *pmf* (*sds R18*) *a* = *pmf* (*sds R14*) *a* *pmf* (*sds R18*) *b* = *0*
pmf (*sds R18*) *d* = *0* *pmf* (*sds R18*) *c* = *1* − *pmf* (*sds R14*) *a*
proof −
from *R23-R12.strategyproofness*(*1*)
have *R21-R23*: *pmf* (*sds R21*) *a* ≤ *pmf* (*sds R23*) *a* **by** *simp*

from *R23-R18.strategyproofness*(*1*)
have *pmf* (*sds R18*) *d* ≤ *pmf* (*sds R21*) *a* − *pmf* (*sds R23*) *a* **by** *simp*
also from *R21-R23* **have** . . . ≤ *0* **by** *simp*
finally show *pmf* (*sds R18*) *d* = *0* **by** *simp*
with *lottery-conditions*[*OF R18.wf*] *R18.support*
show *pmf* (*sds R18*) *a* = *pmf* (*sds R14*) *a*
pmf (*sds R18*) *c* = *1* − *pmf* (*sds R14*) *a* **by** *auto*
qed (*insert R18.support*, *simp-all*)

lemma *R4* [*simp*]: *pmf* (*sds R4*) *a* = *pmf* (*sds R21*) *a* *pmf* (*sds R4*) *b* = *0*
pmf (*sds R4*) *c* = *1* − *pmf* (*sds R4*) *a* *pmf* (*sds R4*) *d* = *0*
proof −
from *R30-R21.strategyproofness*(*1*) *R30.support* *lottery-conditions*[*OF R30.wf*]

have *pmf* (*sds R4*) *c* + *pmf* (*sds R21*) *a* ≤ *pmf* (*sds R4*) *c* + *pmf* (*sds R30*)
a **by** *auto*
also {
have *pmf* (*sds R30*) *a* ≤ *pmf* (*sds R47*) *a*
using *R47-R30.strategyproofness*(*1*) *R30.support* *R47.support*
lottery-conditions[*OF R4.wf*] *lottery-conditions*[*OF R47.wf*] **by** *auto*
moreover from *R4-R47.strategyproofness*(*1*) *R4.support* *R47.support*
lottery-conditions[*OF R4.wf*] *lottery-conditions*[*OF R47.wf*]
have *pmf* (*sds R4*) *c* ≤ *pmf* (*sds R47*) *c* **by** *simp*
ultimately have *pmf* (*sds R4*) *c* + *pmf* (*sds R30*) *a* ≤ *1* − *pmf* (*sds R47*)
d
using *lottery-conditions*[*OF R47.wf*] *R47.support* **by** *simp*
}
finally have *pmf* (*sds R4*) *c* + *pmf* (*sds R14*) *a* ≤ *1*
using *lottery-conditions*[*OF R47.wf*] **by** (*auto simp del*: *pmf-nonneg*)
with *R4-R18.strategyproofness*(*1*) *lottery-conditions*[*OF R4.wf*] *R4.support*
show *pmf* (*sds R4*) *a* = *pmf* (*sds R21*) *a* *pmf* (*sds R4*) *b* = *0*
pmf (*sds R4*) *c* = *1* − *pmf* (*sds R4*) *a* *pmf* (*sds R4*) *d* = *0* **by** *auto*
qed

lemma *R8-d* [*simp*]: *pmf* (*sds R8*) *d* = *1* − *pmf* (*sds R8*) *a*
and *R8-c* [*simp*]: *pmf* (*sds R8*) *c* = *0*
and *R26-a* [*simp*]: *pmf* (*sds R26*) *a* = *1* − *pmf* (*sds R8*) *a*
proof −
from *R8-R26.strategyproofness*(*2*) *R8.support* *lottery-conditions*[*OF R8.wf*]

have *pmf* (*sds R26*) *a* ≤ *pmf* (*sds R8*) *d* **by** *auto*
with *R26-R8.strategyproofness*(*2*) *R8.support lottery-conditions*[*OF R8.wf*]
have *pmf* (*sds R26*) *a* = *pmf* (*sds R8*) *d* **by** *auto*
with *R8-R26.strategyproofness*(*2*) *R8.support lottery-conditions*[*OF R8.wf*]
show *pmf* (*sds R8*) *c* = *0* *pmf* (*sds R8*) *d* = *1* − *pmf* (*sds R8*) *a*
pmf (*sds R26*) *a* = *1* − *pmf* (*sds R8*) *a* **by** (*auto simp del*: *pmf-nonneg*)
qed

lemma *R21-R47*: *pmf* (*sds R21*) *d* ≤ *pmf* (*sds R47*) *c*
using *R4-R47.strategyproofness*(*1*) *R4.support R47.support*
lottery-conditions[*OF R4.wf*] *lottery-conditions*[*OF R47.wf*]
by *auto*

lemma *R30* [*simp*]: *pmf* (*sds R30*) *a* = *pmf* (*sds R47*) *a* *pmf* (*sds R30*) *b* = *0*
pmf (*sds R30*) *c* = *0* *pmf* (*sds R30*) *d* = *1* − *pmf* (*sds R47*) *a*
proof −
have *A*: *pmf* (*sds R30*) *a* ≤ *pmf* (*sds R47*) *a*
using *R47-R30.strategyproofness*(*1*) *R30.support R47.support*
lottery-conditions[*OF R4.wf*] *lottery-conditions*[*OF R47.wf*] **by** *auto*
with *R21-R47 R30-R21.strategyproofness*(*1*)
lottery-conditions[*OF R30.wf*] *lottery-conditions*[*OF R47.wf*]
show *pmf* (*sds R30*) *a* = *pmf* (*sds R47*) *a* *pmf* (*sds R30*) *b* = *0*
pmf (*sds R30*) *c* = *0* *pmf* (*sds R30*) *d* = *1* − *pmf* (*sds R47*) *a*
by (*auto simp*: *R30.support R47.support simp del*: *pmf-nonneg*)
qed

lemma *R31-c-ge-one-half*: *pmf* (*sds R31*) *c* ≥ *1/2*
proof −
from *R25.support* **have** *pmf* (*sds R25*) *a* ≥ *1/2*
proof
assume *pmf* (*sds R25*) *c* = *0*
with *R25-R36.strategyproofness*(*1*) *lottery-conditions*[*OF R36.wf*]
show *pmf* (*sds R25*) *a* ≥ *1/2* **by** (*auto simp del*: *pmf-nonneg*)
next
assume [*simp*]: *pmf* (*sds R25*) *b* = *0*
from *R36-R25.strategyproofness*(*1*) *lottery-conditions*[*OF R25.wf*]
have *pmf* (*sds R25*) *c* + *pmf* (*sds R25*) *a* ≤ *pmf* (*sds R36*) *c* + *1* / *2* **by** *auto*
with *R25-R36.strategyproofness*(*1*) **show** *pmf* (*sds R25*) *a* ≥ *1/2* **by** *auto*
qed
hence *pmf* (*sds R26*) *a* ≥ *1/2*
using *R25-R26.strategyproofness*(*1*) *lottery-conditions*[*OF R25.wf*] **by** (*auto simp del*: *pmf-nonneg*)
with *lottery-conditions*[*OF R47.wf*]
have *1/2* ≤ *pmf* (*sds R26*) *a* + *pmf* (*sds R47*) *d* **by** (*simp del*: *pmf-nonneg*)
also have . . . = *1* − *pmf* (*sds R8*) *a* + *pmf* (*sds R47*) *d* **by** *simp*
also from *R4-R8.strategyproofness*(*1*)
have *1* − *pmf* (*sds R8*) *a* ≤ *pmf* (*sds R21*) *d* **by** *auto*
also note *R21-R47*

also from *R30-R41.strategyproofness(1) R41.support lottery-conditions[OF R41.wf] lottery-conditions[OF R47.wf]*
have *pmf (sds R47) c + pmf (sds R47) d* $\leq$ *pmf (sds R41) d* **by** *(auto simp del: pmf-nonneg)*
also from *R41-R31.strategyproofness(1) R41.support lottery-conditions[OF R31.wf]*

lottery-conditions[OF R41.wf]
have *pmf (sds R41) d* $\leq$ *pmf (sds R31) c* **by** *auto*
finally show *pmf (sds R31) c* $\geq$ *1/2* **by** *simp*
qed

lemma *R31*: *pmf (sds R31) a = 0 pmf (sds R31) c = 1/2 pmf (sds R31) b + pmf (sds R31) d = 1/2*
proof –
from *R2-R38.strategyproofness(1) lottery-conditions[OF R38.wf]*
have *A*: *pmf (sds R38) b + pmf (sds R38) d* $\geq$ *1/2* **by** *auto*
with *R31-c-ge-one-half R31-R38.strategyproofness(1) lottery-conditions[OF R31.wf] lottery-conditions[OF R38.wf]*
have *pmf (sds R38) b + pmf (sds R38) d = pmf (sds R31) d + pmf (sds R31) b* **by** *auto*
with *R31-c-ge-one-half A lottery-conditions[OF R31.wf] lottery-conditions[OF R38.wf]*
show *pmf (sds R31) a = 0 pmf (sds R31) c = 1/2 pmf (sds R31) b + pmf (sds R31) d = 1/2*
by *linarith+*
qed

lemma *absurd*: *False*
using *R31 R45-R31.strategyproofness(2)* **by** *simp*

end

1.4 Lifting to more than 4 agents and alternatives

lemma *finite-list'*:
assumes *finite A*
obtains *xs* **where** *A = set xs distinct xs length xs = card A*
proof –
from *assms* **obtain** *xs* **where** *set xs = A* **using** *finite-list* **by** *blast*
thus *?thesis* **using** *distinct-card[of remdups xs]*
by *(intro that[of remdups xs]) simp-all*
qed

lemma *finite-list-subset*:

assumes *finite A card A* $\geq$ *n*
obtains *xs* **where** *set xs* $\subseteq$ *A distinct xs length xs* = *n*
proof –
obtain *xs* **where** *A* = *set xs distinct xs length xs* = *card A*
using *finite-list'*[*OF assms*(*1*)] **by** *blast*
with *assms* **show** *?thesis*
by (*intro that*[*of take n xs*]) (*simp-all add*: *set-take-subset*)
qed

lemma *card-ge-4E*:
assumes *finite A card A* $\geq$ *4*
obtains *a b c d* **where** *distinct* [*a*,*b*,*c*,*d*] {*a*,*b*,*c*,*d*} $\subseteq$ *A*
proof –
from *finite-list-subset*[*OF assms*] **guess** *xs* **.**
moreover then obtain *a b c d* **where** *xs* = [*a*, *b*, *c*, *d*]
by (*auto simp*: *eval-nat-numeral length-Suc-conv*)
ultimately show *?thesis* **by** (*intro that*[*of a b c d*]) *simp-all*
qed

context *sds-impossibility*
begin

lemma *absurd*: *False*
proof –
from *card-ge-4E*[*OF finite-agents agents-ge-4*] **guess** *A1 A2 A3 A4* **.**
note *agents* = *this*
from *card-ge-4E*[*OF finite-alts alts-ge-4*] **guess** *a b c d* **.**
note *alts* = *this*
define *agents' alts'* **where** *agents'* = {*A1*,*A2*,*A3*,*A4*} **and** *alts'* = {*a*,*b*,*c*,*d*}
from *agents alts*
interpret *sds-lowering-anonymous-neutral-sdeff-stratproof agents alts sds agents' alts'*
unfolding *agents'-def alts'-def* **by** *unfold-locales simp-all*
from *agents alts*
interpret *sds-impossibility-4-4 agents' alts' lowered A1 A2 A3 A4 a b c d*
by *unfold-locales* (*simp-all add*: *agents'-def alts'-def*)
from *absurd* **show** *False* **.**
qed

end

end

References

[1] F. Brandl, F. Brandt, and C. Geist. Proving the incompatibility of Efficiency and Strategyproofness via SMT solving. *Proceedings of the*

25th International Joint Conference on Artificial Intelligence (IJCAI), 2016. Forthcoming.

[2] F. Brandl, F. Brandt, and W. Suksompong. The impossibility of extending Random Dictatorship to weak preferences. *Economics Letters*, 141:pp. 44 – 47, 2016.