Von Neumann Morgenstern Utility Theorem *

Julian Parsert   Cezary Kaliszyk

December 14, 2021

Abstract

Utility functions form an essential part of game theory and economics. In order to guarantee the existence of utility functions most of the time sufficient properties are assumed in an axiomatic manner. One famous and very common set of such assumptions is that of expected utility theory. Here, the rationality, continuity, and independence of preferences is assumed. The von-Neumann-Morgenstern Utility theorem shows that these assumptions are necessary and sufficient for an expected utility function to exists. This theorem was proven by Neumann and Morgenstern in "Theory of Games and Economic Behavior" which is regarded as one of the most influential works in game theory.

We formalize these results in Isabelle/HOL. The formalization includes formal definitions of the underlying concepts including continuity and independence of preferences.

Contents

1 Composition of Probability Mass functions 2

2 Lotteries 5

3 Properties of Preferences 6
3.1 Independent Preferences 6
3.2 Continuity 9

4 System U start, as per vNM 10

5 This lemma is in called step 1 in literature. In Von Neumann and Morgenstern's book this is A:A (albeit more general) 11
5.1 Add finiteness and non emptyness of outcomes 13
5.2 Add continuity to assumptions 16

*This work is supported by the Austrian Science Fund (FWF) project P26201 and the European Research Council (ERC) grant no 714034 *SMART*.

6 Definition of vNM-utility function 20

7 Finite outcomes 21

8 Related work 23

theory *PMF-Composition*
 imports
 HOL−Probability.Probability
begin

1 Composition of Probability Mass functions

definition *mix-pmf* :: *real* ⇒ *'a pmf* ⇒ *'a pmf* ⇒ *'a pmf* **where**
 mix-pmf α *p q* = (*bernoulli-pmf* α)$\gg=$ (λX. *if X then p else q*)

lemma *pmf-mix*: *a* ∈ {*0..1*} ⟹ *pmf* (*mix-pmf a p q*) *x* = *a* ∗ *pmf p x* + (*1* − *a*) ∗ *pmf q x*
 ⟨*proof*⟩

lemma *pmf-mix-deeper*: *a* ∈ {*0..1*} ⟹ *pmf* (*mix-pmf a p q*) *x* = *a* ∗ *pmf p x* + *pmf q x* − *a* ∗ *pmf q x*
 ⟨*proof*⟩

lemma *bernoulli-pmf-0* [*simp*]: *bernoulli-pmf 0* = *return-pmf False*
 ⟨*proof*⟩

lemma *bernoulli-pmf-1* [*simp*]: *bernoulli-pmf 1* = *return-pmf True*
 ⟨*proof*⟩

lemma *pmf-mix-0* [*simp*]: *mix-pmf 0 p q* = *q*
 ⟨*proof*⟩

lemma *pmf-mix-1* [*simp*]: *mix-pmf 1 p q* = *p*
 ⟨*proof*⟩

lemma *set-pmf-mix*: *a* ∈ {*0*<..<*1*} ⟹ *set-pmf* (*mix-pmf a p q*) = *set-pmf p* ∪ *set-pmf q*
 ⟨*proof*⟩

lemma *set-pmf-mix-eq*: *a* ∈ {*0..1*} ⟹ *mix-pmf a p p* = *p*
 ⟨*proof*⟩

lemma *pmf-equiv-intro*[*intro*]:
 assumes ⋀*e*. *e* ∈ *set-pmf p* ⟹ *pmf p e* = *pmf q e*
 assumes ⋀*e*. *e* ∈ *set-pmf q* ⟹ *pmf q e* = *pmf p e*

shows $p = q$
⟨*proof*⟩

lemma *pmf-equiv-intro1*[*intro*]:
assumes $\bigwedge e.\ e \in$ *set-pmf p* $\Longrightarrow$ *pmf p e* $=$ *pmf q e*
shows $p = q$
⟨*proof*⟩

lemma *pmf-inverse-switch-eqals*:
assumes $a \in \{0..1\}$
shows *mix-pmf a p q* $=$ *mix-pmf* $(1-a)$ *q p*
⟨*proof*⟩

lemma *mix-pmf-comp-left-div*:
assumes $\alpha \in \{0..(1::real)\}$
and $\beta \in \{0..(1::real)\}$
assumes $\alpha > \beta$
shows *pmf* (*mix-pmf* (β/α) (*mix-pmf* α *p q*) *q*) *e* $= \beta *$ *pmf p e* $+$ *pmf q e* $-$
$\beta *$ *pmf q e*
⟨*proof*⟩

lemma *mix-pmf-comp-with-dif-equiv*:
assumes $\alpha \in \{0..(1::real)\}$
and $\beta \in \{0..(1::real)\}$
assumes $\alpha > \beta$
shows *mix-pmf* (β/α) (*mix-pmf* α *p q*) *q* $=$ *mix-pmf* β *p q* (**is** *?l* $=$ *?r*)
⟨*proof*⟩

lemma *product-mix-pmf-prob-distrib*:
assumes $a \in \{0..1\}$
and $b \in \{0..1\}$
shows *mix-pmf a* (*mix-pmf b p q*) *q* $=$ *mix-pmf* $(a*b)$ *p q*
⟨*proof*⟩

lemma *mix-pmf-subset-of-original*:
assumes $a \in \{0..1\}$
shows (*set-pmf* (*mix-pmf a p q*)) $\subseteq$ *set-pmf p* $\cup$ *set-pmf q*
⟨*proof*⟩

lemma *mix-pmf-preserves-finite-support*:
assumes $a \in \{0..1\}$
assumes *finite* (*set-pmf p*)
and *finite* (*set-pmf q*)
shows *finite* (*set-pmf* (*mix-pmf a p q*))
⟨*proof*⟩

lemma *ex-certain-iff-singleton-support*:
shows ($\exists x.$ *pmf p x* $= 1$) $\longleftrightarrow$ *card* (*set-pmf p*) $= 1$
⟨*proof*⟩

We thank Manuel Eberl for suggesting the following two lemmas.

lemma *mix-pmf-partition*:
  fixes p :: *'a pmf*
  assumes $y \in$ *set-pmf p set-pmf p* $- \{y\} \neq \{\}$
  obtains *a q* **where** $a \in \{0<..<1\}$ *set-pmf q* = *set-pmf p* $- \{y\}$
    p = *mix-pmf a q* (*return-pmf y*)
⟨*proof*⟩

lemma *pmf-mix-induct* [*consumes 2*, *case-names degenerate mix*]:
  assumes *finite A set-pmf p* $\subseteq A$
  assumes *degenerate*: $\bigwedge x.\ x \in A \Longrightarrow P$ (*return-pmf x*)
  assumes *mix*: $\bigwedge p\ a\ y.$ *set-pmf p* $\subseteq A \Longrightarrow a \in \{0<..<1\} \Longrightarrow y \in A \Longrightarrow$
    $P\ p \Longrightarrow P$ (*mix-pmf a p* (*return-pmf y*))
  shows *P p*
⟨*proof*⟩

lemma *pmf-mix-induct'* [*consumes 2*, *case-names degenerate mix*]:
  assumes *finite A set-pmf p* $\subseteq A$
  assumes *degenerate*: $\bigwedge x.\ x \in A \Longrightarrow P$ (*return-pmf x*)
  assumes *mix*: $\bigwedge p\ q\ a.$ *set-pmf p* $\subseteq A \Longrightarrow$ *set-pmf q* $\subseteq A \Longrightarrow a \in \{0<..<1\}$
$\Longrightarrow$
    $P\ p \Longrightarrow P\ q \Longrightarrow P$ (*mix-pmf a p q*)
  shows *P p*
  ⟨*proof*⟩

lemma *finite-sum-distribute-mix-pmf*:
  assumes *finite* (*set-pmf* (*mix-pmf a p q*))
  assumes *finite* (*set-pmf p*)
  assumes *finite* (*set-pmf q*)
  shows ($\sum i \in$ *set-pmf* (*mix-pmf a p q*). *pmf* (*mix-pmf a p q*) *i*) = ($\sum i \in$*set-pmf p*. $a *$ *pmf p i*) + ($\sum i \in$*set-pmf q*. $(1-a) *$ *pmf q i*)
⟨*proof*⟩

lemma *distribute-alpha-over-sum*:
  shows ($\sum i \in$*set-pmf T*. $a *$ *pmf p i* $*$ *f i*) = $a *$ ($\sum i \in$*set-pmf T*. *pmf p i* $*$ *f i*)
  ⟨*proof*⟩

lemma *sum-over-subset-pmf-support*:
  assumes *finite T*
  assumes *set-pmf p* $\subseteq T$
  shows ($\sum i \in T$. $a *$ *pmf p i* $*$ *f i*) = ($\sum i \in$*set-pmf p*. $a *$ *pmf p i* $*$ *f i*)
⟨*proof*⟩

lemma *expected-value-mix-pmf-distrib*:
  assumes *finite* (*set-pmf p*)
    and *finite* (*set-pmf q*)
  assumes $a \in \{0<..<1\}$
  shows *measure-pmf.expectation* (*mix-pmf a p q*) *f* = $a *$ *measure-pmf.expectation p f* + $(1-a) *$ *measure-pmf.expectation q f*

⟨*proof*⟩

lemma *expected-value-mix-pmf*:
  assumes *finite* (*set-pmf p*)
    and *finite* (*set-pmf q*)
  assumes $a \in \{0..1\}$
  shows *measure-pmf.expectation* (*mix-pmf a p q*) *f* = *a* ∗ *measure-pmf.expectation p f* + (*1*−*a*) ∗ *measure-pmf.expectation q f*
⟨*proof*⟩

end

theory *Lotteries*
  imports
    PMF-Composition
    HOL−Probability.Probability
begin

2 Lotteries

definition *lotteries-on*
  where
    lotteries-on Oc = {*p* . (*set-pmf p*) ⊆ *Oc*}

lemma *lotteries-on-subset*:
  assumes *A* ⊆ *B*
  shows *lotteries-on A* ⊆ *lotteries-on B*
  ⟨*proof*⟩

lemma *support-in-outcomes*:
  ∀ *oc*. ∀ *p* ∈ *lotteries-on oc*. ∀ *a* ∈ *set-pmf p*. *a* ∈ *oc*
  ⟨*proof*⟩

lemma *lotteries-on-nonempty*:
  assumes *outcomes* ≠ {}
  shows *lotteries-on outcomes* ≠ {}
  ⟨*proof*⟩

lemma *finite-support-one-oc*:
  assumes *card outcomes* = *1*
  shows ∀ *l* ∈ *lotteries-on outcomes*. *finite* (*set-pmf l*)
  ⟨*proof*⟩

lemma *one-outcome-card-support-1*:
  assumes *card outcomes* = *1*
  shows ∀ *l* ∈ *lotteries-on outcomes*. *card* (*set-pmf l*) = *1*
⟨*proof*⟩

lemma *finite-nempty-ex-degernate-in-lotteries*:
  assumes *out* ≠ {}
  assumes *finite out*
  shows ∃ *e* ∈ *lotteries-on out*. ∃ *x* ∈ *out*. *pmf e x* = *1*
⟨*proof*⟩

lemma *card-support-1-probability-1*:
  assumes *card* (*set-pmf p*) = *1*
  shows ∀ *e* ∈ *set-pmf p*. *pmf p e* = *1*
  ⟨*proof*⟩

lemma *one-outcome-card-lotteries-1*:
  assumes *card outcomes* = *1*
  shows *card* (*lotteries-on outcomes*) = *1*
⟨*proof*⟩

lemma *return-pmf-card-equals-set*:
  shows *card* {*return-pmf x* |*x*. *x* ∈ *S*} = *card S*
⟨*proof*⟩

lemma *mix-pmf-in-lotteries*:
  assumes *p* ∈ *lotteries-on A*
    and *q* ∈ *lotteries-on A*
    and *a* ∈ {*0*<..<*1*}
  shows (*mix-pmf a p q*) ∈ *lotteries-on A*
⟨*proof*⟩

lemma *card-degen-lotteries-equals-outcomes*:
  shows *card* {*x* ∈ *lotteries-on out*. *card* (*set-pmf x*) = *1*} = *card out*
⟨*proof*⟩

end

theory *Neumann-Morgenstern-Utility-Theorem*
  imports
    HOL−Probability.Probability
    First-Welfare-Theorem.Utility-Functions
    Lotteries
begin

3 Properties of Preferences

3.1 Independent Preferences

Independence is sometimes called substitution

Notice how r is "added" to the right of mix-pmf and the element to the left q/p changes

definition *independent-vnm*
  where
    independent-vnm C P =
    (∀ *p* ∈ *C*. ∀ *q* ∈ *C*. ∀ *r* ∈ *C*. ∀ (α::*real*) ∈ {*0*<..*1*}. *p* ⪰[*P*] *q* ⟷ *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*)

lemma *independent-vnmI1*:
  assumes (∀ *p* ∈ *C*. ∀ *q* ∈ *C*. ∀ *r* ∈ *C*. ∀ α ∈ {*0*<..*1*}. *p* ⪰[*P*] *q* ⟷ *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*)
  shows *independent-vnm C P*
  ⟨*proof*⟩

lemma *independent-vnmI2*:
  assumes ⋀*p q r* α. *p* ∈ *C* ⟹ *q* ∈ *C* ⟹ *r* ∈ *C* ⟹ α ∈ {*0*<..*1*} ⟹ *p* ⪰[*P*] *q* ⟷ *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*
  shows *independent-vnm C P*
  ⟨*proof*⟩

lemma *independent-vnm-alt-def*:
  shows *independent-vnm C P* ⟷ (∀ *p* ∈ *C*. ∀ *q* ∈ *C*. ∀ *r* ∈ *C*. ∀ α ∈ {*0*<..<*1*}.
  p ⪰[*P*] *q* ⟷ *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*) (**is** *?L* ⟷ *?R*)
⟨*proof*⟩

lemma *independece-dest-alt*:
  assumes *independent-vnm C P*
  shows (∀ *p* ∈ *C*. ∀ *q* ∈ *C*. ∀ *r* ∈ *C*. ∀ (α::*real*) ∈ {*0*<..*1*}. *p* ⪰[*P*] *q* ⟷ *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*)
⟨*proof*⟩

lemma *independent-vnmD1*:
  assumes *independent-vnm C P*
  shows (∀ *p* ∈ *C*. ∀ *q* ∈ *C*. ∀ *r* ∈ *C*. ∀ α ∈ {*0*<..*1*}. *p* ⪰[*P*] *q* ⟷ *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*)
  ⟨*proof*⟩

lemma *independent-vnmD2*:
  fixes *p q r* α
  assumes α ∈ {*0*<..*1*}
    and *p* ∈ *C*
    and *q* ∈ *C*
    and *r* ∈ *C*
  assumes *independent-vnm C P*
  assumes *p* ⪰[*P*] *q*
  shows *mix-pmf* α *p* *r* ⪰[*P*] *mix-pmf* α *q* *r*
  ⟨*proof*⟩

lemma *independent-vnmD3*:

fixes *p q r α*
assumes *α* ∈ {*0*<..*1*}
and *p* ∈ *C*
and *q* ∈ *C*
and *r* ∈ *C*
assumes *independent-vnm C P*
assumes *mix-pmf α p r* ⪰[*P*] *mix-pmf α q r*
shows *p* ⪰[*P*] *q*
⟨*proof*⟩

lemma *independent-vnmD4*:
assumes *independent-vnm C P*
assumes *refl-on C P*
assumes *p* ∈ *C*
and *q* ∈ *C*
and *r* ∈ *C*
and *α* ∈ {*0*..*1*}
and *p* ⪰[*P*] *q*
shows *mix-pmf α p r* ⪰[*P*] *mix-pmf α q r*
⟨*proof*⟩

lemma *approx-indep-ge*:
assumes *x* ≈[𝓡] *y*
assumes *α* ∈ {*0*..(*1*::*real*)}
assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) 𝓡
and *ind*: *independent-vnm* (*lotteries-on outcomes*) 𝓡
shows ∀ *r* ∈ *lotteries-on outcomes*. (*mix-pmf α y r*) ⪰[𝓡] (*mix-pmf α x r*)
⟨*proof*⟩

lemma *approx-imp-approx-ind*:
assumes *x* ≈[𝓡] *y*
assumes *α* ∈ {*0*..(*1*::*real*)}
assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) 𝓡
and *ind*: *independent-vnm* (*lotteries-on outcomes*) 𝓡
shows ∀ *r* ∈ *lotteries-on outcomes*. (*mix-pmf α y r*) ≈[𝓡] (*mix-pmf α x r*)
⟨*proof*⟩

lemma *geq-imp-mix-geq-right*:
assumes *x* ⪰[𝓡] *y*
assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) 𝓡
assumes *ind*: *independent-vnm* (*lotteries-on outcomes*) 𝓡
assumes *α* ∈ {*0*..(*1*::*real*)}
shows (*mix-pmf α x y*) ⪰[𝓡] *y*
⟨*proof*⟩

lemma *geq-imp-mix-geq-left*:
assumes *x* ⪰[𝓡] *y*
assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) 𝓡
assumes *ind*: *independent-vnm* (*lotteries-on outcomes*) 𝓡

assumes $\alpha \in \{0..(1::real)\}$
shows $(mix\text{-}pmf\ \alpha\ y\ x) \succeq[\mathcal{R}]\ y$
$\langle proof \rangle$

lemma *sg-imp-mix-sg*:
assumes $x \succ[\mathcal{R}]\ y$
assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) $\mathcal{R}$
assumes *ind*: *independent-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
assumes $\alpha \in \{0{<}..(1::real)\}$
shows $(mix\text{-}pmf\ \alpha\ x\ y) \succ[\mathcal{R}]\ y$
$\langle proof \rangle$

3.2 Continuity

Continuity is sometimes called Archimedean Axiom

definition *continuous-vnm*
where
$continuous\text{-}vnm\ C\ P = (\forall p \in C.\ \forall q \in C.\ \forall r \in C.\ p \succeq[P]\ q \wedge q \succeq[P]\ r \longrightarrow$
$(\exists \alpha \in \{0..1\}.\ (mix\text{-}pmf\ \alpha\ p\ r) \approx[P]\ q))$

lemma *continuous-vnmD*:
assumes *continuous-vnm C P*
shows $(\forall p \in C.\ \forall q \in C.\ \forall r \in C.\ p \succeq[P]\ q \wedge q \succeq[P]\ r \longrightarrow$
$(\exists \alpha \in \{0..1\}.\ (mix\text{-}pmf\ \alpha\ p\ r) \approx[P]\ q))$
$\langle proof \rangle$

lemma *continuous-vnmI*:
assumes $\bigwedge p\ q\ r.\ p \in C \Longrightarrow q \in C \Longrightarrow r \in C \Longrightarrow p \succeq[P]\ q \wedge q \succeq[P]\ r \Longrightarrow$
$\exists \alpha \in \{0..1\}.\ (mix\text{-}pmf\ \alpha\ p\ r) \approx[P]\ q$
shows *continuous-vnm C P*
$\langle proof \rangle$

lemma *mix-in-lot*:
assumes $x \in$ *lotteries-on outcomes*
and $y \in$ *lotteries-on outcomes*
and $\alpha \in \{0..1\}$
shows $(mix\text{-}pmf\ \alpha\ x\ y) \in$ *lotteries-on outcomes*
$\langle proof \rangle$

lemma *non-unique-continuous-unfolding*:
assumes *cnt*: *continuous-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
assumes *rational-preference* (*lotteries-on outcomes*) $\mathcal{R}$
assumes $p \succeq[\mathcal{R}]\ q$
and $q \succeq[\mathcal{R}]\ r$
and $p \succ[\mathcal{R}]\ r$
shows $\exists \alpha \in \{0..1\}.\ q \approx[\mathcal{R}]\ mix\text{-}pmf\ \alpha\ p\ r$
$\langle proof \rangle$

4 System U start, as per vNM

These are the first two assumptions which we use to derive the first results. We assume rationality and independence. In this system U the von-Neumann-Morgenstern Utility Theorem is proven.

context
  fixes *outcomes* :: *'a set*
  fixes $\mathcal{R}$
  assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) $\mathcal{R}$
  assumes *ind*: *independent-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
begin

abbreviation $\mathcal{P} \equiv$ *lotteries-on outcomes*

lemma *relation-in-carrier*:
  $x \succeq[\mathcal{R}]\ y \Longrightarrow x \in \mathcal{P} \wedge y \in \mathcal{P}$
  ⟨*proof*⟩

lemma *mix-pmf-preferred-independence*:
  assumes $r \in \mathcal{P}$
    and $\alpha \in \{0..1\}$
  assumes $p \succeq[\mathcal{R}]\ q$
  shows *mix-pmf* $\alpha\ p\ r \succeq[\mathcal{R}]$ *mix-pmf* $\alpha\ q\ r$
  ⟨*proof*⟩

lemma *mix-pmf-strict-preferred-independence*:
  assumes $r \in \mathcal{P}$
    and $\alpha \in \{0{<}..1\}$
  assumes $p \succ[\mathcal{R}]\ q$
  shows *mix-pmf* $\alpha\ p\ r \succ[\mathcal{R}]$ *mix-pmf* $\alpha\ q\ r$
  ⟨*proof*⟩

lemma *mix-pmf-preferred-independence-rev*:
  assumes $p \in \mathcal{P}$
    and $q \in \mathcal{P}$
    and $r \in \mathcal{P}$
    and $\alpha \in \{0{<}..1\}$
  assumes *mix-pmf* $\alpha\ p\ r \succeq[\mathcal{R}]$ *mix-pmf* $\alpha\ q\ r$
  shows $p \succeq[\mathcal{R}]\ q$
⟨*proof*⟩

lemma *x-sg-y-sg-mpmf-right*:
  assumes $x \succ[\mathcal{R}]\ y$
  assumes $b \in \{0{<}..(1::real)\}$
  shows $x \succ[\mathcal{R}]$ *mix-pmf* $b\ y\ x$
⟨*proof*⟩

lemma *neumann-3B-b*:

assumes $u \succ[\mathcal{R}]\ v$
assumes $\alpha \in \{0<..<1\}$
shows $u \succ[\mathcal{R}]\ \textit{mix-pmf}\ \alpha\ u\ v$
$\langle \textit{proof} \rangle$

lemma *neumann-3B-b-non-strict*:
assumes $u \succeq[\mathcal{R}]\ v$
assumes $\alpha \in \{0..1\}$
shows $u \succeq[\mathcal{R}]\ \textit{mix-pmf}\ \alpha\ u\ v$
$\langle \textit{proof} \rangle$

lemma *greater-mix-pmf-greater-step-1-aux*:
assumes $v \succ[\mathcal{R}]\ u$
assumes $\alpha \in \{0<..<(1::real)\}$
and $\beta \in \{0<..<(1::real)\}$
assumes $\beta > \alpha$
shows $(\textit{mix-pmf}\ \beta\ v\ u) \succ[\mathcal{R}]\ (\textit{mix-pmf}\ \alpha\ v\ u)$
$\langle \textit{proof} \rangle$

5 This lemma is in called step 1 in literature. In Von Neumann and Morgenstern's book this is A:A (albeit more general)

lemma *step-1-most-general*:
assumes $x \succ[\mathcal{R}]\ y$
assumes $\alpha \in \{0..(1::real)\}$
and $\beta \in \{0..(1::real)\}$
assumes $\alpha > \beta$
shows $(\textit{mix-pmf}\ \alpha\ x\ y) \succ[\mathcal{R}]\ (\textit{mix-pmf}\ \beta\ x\ y)$
$\langle \textit{proof} \rangle$

Kreps refers to this lemma as 5.6 c. The lemma after that is also significant.

lemma *approx-remains-after-same-comp*:
assumes $p \approx[\mathcal{R}]\ q$
and $r \in \mathcal{P}$
and $\alpha \in \{0..1\}$
shows $\textit{mix-pmf}\ \alpha\ p\ r \approx[\mathcal{R}]\ \textit{mix-pmf}\ \alpha\ q\ r$
$\langle \textit{proof} \rangle$

This lemma is the symmetric version of the previous lemma. This lemma is never mentioned in literature anywhere. Even though it looks trivial now, due to the asymmetric nature of the independence axiom, it is not so trivial, and definitely worth mentioning.

lemma *approx-remains-after-same-comp-left*:
assumes $p \approx[\mathcal{R}]\ q$
and $r \in \mathcal{P}$
and $\alpha \in \{0..1\}$

shows *mix-pmf* α *r p* $\approx[\mathcal{R}]$ *mix-pmf* α *r q*
⟨*proof*⟩

lemma *mix-of-preferred-is-preferred*:
assumes $p \succeq[\mathcal{R}]\ w$
assumes $q \succeq[\mathcal{R}]\ w$
assumes $\alpha \in \{0..1\}$
shows *mix-pmf* α *p q* $\succeq[\mathcal{R}]\ w$
⟨*proof*⟩

lemma *mix-of-not-preferred-is-not-preferred*:
assumes $w \succeq[\mathcal{R}]\ p$
assumes $w \succeq[\mathcal{R}]\ q$
assumes $\alpha \in \{0..1\}$
shows $w \succeq[\mathcal{R}]$ *mix-pmf* α *p q*
⟨*proof*⟩ **definition** *degenerate-lotteries* **where**
degenerate-lotteries = $\{x \in \mathcal{P}.\ card\ (set\text{-}pmf\ x) = 1\}$

private definition *best* **where**
best = $\{x \in \mathcal{P}.\ (\forall y \in \mathcal{P}.\ x \succeq[\mathcal{R}]\ y)\}$

private definition *worst* **where**
worst = $\{x \in \mathcal{P}.\ (\forall y \in \mathcal{P}.\ y \succeq[\mathcal{R}]\ x)\}$

lemma *degenerate-total*:
$\forall e \in degenerate\text{-}lotteries.\ \forall m \in \mathcal{P}.\ e \succeq[\mathcal{R}]\ m \lor m \succeq[\mathcal{R}]\ e$
⟨*proof*⟩

lemma *degen-outcome-cardinalities*:
card degenerate-lotteries = *card outcomes*
⟨*proof*⟩

lemma *degenerate-lots-subset-all*: *degenerate-lotteries* $\subseteq \mathcal{P}$
⟨*proof*⟩

lemma *alt-definition-of-degenerate-lotteries*[*iff*]:
$\{return\text{-}pmf\ x\ |x.\ x\in\ outcomes\}$ = *degenerate-lotteries*
⟨*proof*⟩

lemma *best-indifferent*:
$\forall x \in best.\ \forall y \in best.\ x \approx[\mathcal{R}]\ y$
⟨*proof*⟩

lemma *worst-indifferent*:
$\forall x \in worst.\ \forall y \in worst.\ x \approx[\mathcal{R}]\ y$
⟨*proof*⟩

lemma *best-worst-indiff-all-indiff*:
assumes $b \in best$

and $w \in$ *worst*
and $b \approx[\mathcal{R}]\ w$
shows $\forall\, e \in \mathcal{P}.\ e \approx[\mathcal{R}]\ w\ \forall\, e \in \mathcal{P}.\ e \approx[\mathcal{R}]\ b$
⟨*proof*⟩

Like Step 1 most general but with IFF.

lemma *mix-pmf-pref-iff-more-likely* [*iff*]:
assumes $b \succ[\mathcal{R}]\ w$
assumes $\alpha \in \{0..1\}$
and $\beta \in \{0..1\}$
shows $\alpha > \beta \longleftrightarrow$ *mix-pmf* $\alpha\ b\ w \succ[\mathcal{R}]$ *mix-pmf* $\beta\ b\ w$ (**is** *?L* $\longleftrightarrow$ *?R*)
⟨*proof*⟩

lemma *better-worse-good-mix-preferred*[*iff*]:
assumes $b \succeq[\mathcal{R}]\ w$
assumes $\alpha \in \{0..1\}$
and $\beta \in \{0..1\}$
assumes $\alpha \geq \beta$
shows *mix-pmf* $\alpha\ b\ w \succeq[\mathcal{R}]$ *mix-pmf* $\beta\ b\ w$
⟨*proof*⟩

5.1 Add finiteness and non emptyness of outcomes

context
assumes *fnt*: *finite outcomes*
assumes *nempty*: *outcomes* $\neq \{\}$
begin

lemma *finite-degenerate-lotteries*:
finite degenerate-lotteries
⟨*proof*⟩

lemma *degenerate-has-max-preferred*:
$\{x \in$ *degenerate-lotteries*. $(\forall\, y \in$ *degenerate-lotteries*. $x \succeq[\mathcal{R}]\ y)\} \neq \{\}$ (**is** *?l* $\neq \{\}$)
⟨*proof*⟩

lemma *degenerate-has-min-preferred*:
$\{x \in$ *degenerate-lotteries*. $(\forall\, y \in$ *degenerate-lotteries*. $y \succeq[\mathcal{R}]\ x)\} \neq \{\}$ (**is** *?l* $\neq \{\}$)
⟨*proof*⟩

lemma *exists-best-degenerate*:
$\exists\, x \in$ *degenerate-lotteries*. $\forall\, y \in$ *degenerate-lotteries*. $x \succeq[\mathcal{R}]\ y$
⟨*proof*⟩

lemma *exists-worst-degenerate*:
$\exists\, x \in$ *degenerate-lotteries*. $\forall\, y \in$ *degenerate-lotteries*. $y \succeq[\mathcal{R}]\ x$
⟨*proof*⟩

lemma *best-degenerate-in-best-overall*:
  $\exists x \in$ *degenerate-lotteries*. $\forall y \in \mathcal{P}$. $x \succeq[\mathcal{R}]$ y
⟨*proof*⟩

lemma *worst-degenerate-in-worst-overall*:
  $\exists x \in$ *degenerate-lotteries*. $\forall y \in \mathcal{P}$. $y \succeq[\mathcal{R}]$ x
⟨*proof*⟩

lemma *overall-best-nonempty*:
  best $\neq \{\}$
  ⟨*proof*⟩

lemma *overall-worst-nonempty*:
  worst $\neq \{\}$
  ⟨*proof*⟩

lemma *trans-approx*:
  assumes $x \approx[\mathcal{R}]$ y
    and $y \approx[\mathcal{R}]$ z
  shows $x \approx[\mathcal{R}]$ z
  ⟨*proof*⟩

First EXPLICIT use of the axiom of choice

private definition *some-best* **where**
  some-best = (*SOME x*. $x \in$ *degenerate-lotteries* $\wedge$ $x \in$ *best*)

private definition *some-worst* **where**
  some-worst = (*SOME x*. $x \in$ *degenerate-lotteries* $\wedge$ $x \in$ *worst*)

private definition *my-U* :: *'a pmf* $\Rightarrow$ *real*
  where
    my-U p = (*SOME* α. $\alpha \in \{0..1\}$ $\wedge$ $p \approx[\mathcal{R}]$ *mix-pmf* α *some-best some-worst*)

lemma *exists-best-and-degenerate*: *degenerate-lotteries* $\cap$ *best* $\neq \{\}$
  ⟨*proof*⟩

lemma *exists-worst-and-degenerate*: *degenerate-lotteries* $\cap$ *worst* $\neq \{\}$
  ⟨*proof*⟩

lemma *some-best-in-best*: *some-best* $\in$ *best*
  ⟨*proof*⟩

lemma *some-worst-in-worst*: *some-worst* $\in$ *worst*
  ⟨*proof*⟩

lemma *best-always-at-least-as-good-mix*:
  assumes $\alpha \in \{0..1\}$
    and $p \in \mathcal{P}$
  shows *mix-pmf* α *some-best* $p \succeq[\mathcal{R}]$ p
  ⟨*proof*⟩

lemma *geq-mix-imp-weak-pref*:
  assumes $\alpha \in \{0..1\}$
    and $\beta \in \{0..1\}$
  assumes $\alpha \geq \beta$
  shows *mix-pmf* α *some-best some-worst* $\succeq[\mathcal{R}]$ *mix-pmf* β *some-best some-worst*
  ⟨*proof*⟩

lemma *gamma-inverse*:
  assumes $\alpha \in \{0<..<1\}$
    and $\beta \in \{0<..<1\}$
  shows $(1::real) - (\alpha - \beta) \ / \ (1 - \beta) = (1 - \alpha) \ / \ (1 - \beta)$
⟨*proof*⟩

lemma *all-mix-pmf-indiff-indiff-best-worst*:
  assumes $l \in \mathcal{P}$
  assumes $b \in best$
  assumes $w \in worst$
  assumes $b \approx[\mathcal{R}]$ w
  shows $\forall \alpha \in \{0..1\}.\ l \approx[\mathcal{R}]$ *mix-pmf* α b w
  ⟨*proof*⟩

lemma *indiff-imp-same-utility-value*:
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  assumes $\alpha \in \{0..1\}$
  assumes $\beta \in \{0..1\}$
  assumes *mix-pmf* β *some-best some-worst* $\approx[\mathcal{R}]$ *mix-pmf* α *some-best some-worst*
  shows $\beta = \alpha$
  ⟨*proof*⟩

lemma *leq-mix-imp-weak-inferior*:
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  assumes $\alpha \in \{0..1\}$
    and $\beta \in \{0..1\}$
  assumes *mix-pmf* β *some-best some-worst* $\succeq[\mathcal{R}]$ *mix-pmf* α *some-best some-worst*
  shows $\beta \geq \alpha$
⟨*proof*⟩

lemma *ge-mix-pmf-preferred*:
  assumes $x \succ[\mathcal{R}]$ y
  assumes $\alpha \in \{0..1\}$
    and $\beta \in \{0..1\}$
  assumes $\alpha \geq \beta$
  shows (*mix-pmf* α x y) $\succeq[\mathcal{R}]$ (*mix-pmf* β x y)

⟨*proof*⟩

5.2 Add continuity to assumptions

context
assumes *cnt*: *continuous-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
begin

In Literature this is referred to as step 2.

lemma *step-2-unique-continuous-unfolding*:
assumes $p \succeq[\mathcal{R}]\ q$
and $q \succeq[\mathcal{R}]\ r$
and $p \succ[\mathcal{R}]\ r$
shows $\exists!\alpha \in \{0..1\}.\ q \approx[\mathcal{R}]$ *mix-pmf* $\alpha\ p\ r$
⟨*proof*⟩

These folowing two lemmas are referred to sometimes called step 2.

lemma *create-unique-indiff-using-distinct-best-worst*:
assumes $l \in \mathcal{P}$
assumes $b \in$ *best*
assumes $w \in$ *worst*
assumes $b \succ[\mathcal{R}]\ w$
shows $\exists!\alpha \in\{0..1\}.\ l \approx[\mathcal{R}]$ *mix-pmf* $\alpha\ b\ w$
⟨*proof*⟩

lemma *exists-element-bw-mix-is-approx*:
assumes $l \in \mathcal{P}$
assumes $b \in$ *best*
assumes $w \in$ *worst*
shows $\exists\,\alpha \in\{0..1\}.\ l \approx[\mathcal{R}]$ *mix-pmf* $\alpha\ b\ w$
⟨*proof*⟩

lemma *my-U-is-defined*:
assumes $p \in \mathcal{P}$
shows *my-U p* $\in \{0..1\}\ p \approx[\mathcal{R}]$ *mix-pmf* (*my-U p*) *some-best some-worst*
⟨*proof*⟩

lemma *weak-pref-mix-with-my-U-weak-pref*:
assumes $p \succeq[\mathcal{R}]\ q$
shows *mix-pmf* (*my-U p*) *some-best some-worst* $\succeq[\mathcal{R}]$ *mix-pmf* (*my-U q*) *some-best some-worst*
⟨*proof*⟩

lemma *preferred-greater-my-U*:
assumes $p \in \mathcal{P}$
and $q \in \mathcal{P}$
assumes *mix-pmf* (*my-U p*) *some-best some-worst* $\succ[\mathcal{R}]$ *mix-pmf* (*my-U q*) *some-best some-worst*
shows *my-U p* $>$ *my-U q*

⟨*proof*⟩

lemma *geq-my-U-imp-weak-preference*:
  assumes $p \in \mathcal{P}$
    and $q \in \mathcal{P}$
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  assumes *my-U* $p \geq$ *my-U* q
  shows $p \succeq[\mathcal{R}]\ q$
⟨*proof*⟩

lemma *my-U-represents-pref*:
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  assumes $p \in \mathcal{P}$
    and $q \in \mathcal{P}$
  shows $p \succeq[\mathcal{R}]\ q \longleftrightarrow$ *my-U* $p \geq$ *my-U* q (**is** *?L* $\longleftrightarrow$ *?R*)
⟨*proof*⟩

lemma *first-iff-u-greater-strict-preff*:
  assumes $p \in \mathcal{P}$
    and $q \in \mathcal{P}$
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  shows *my-U* $p >$ *my-U* $q \longleftrightarrow$ *mix-pmf* (*my-U* p) *some-best some-worst* $\succ[\mathcal{R}]$ *mix-pmf* (*my-U* q) *some-best some-worst*
⟨*proof*⟩

lemma *second-iff-calib-mix-pref-strict-pref*:
  assumes $p \in \mathcal{P}$
    and $q \in \mathcal{P}$
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  shows *mix-pmf* (*my-U* p) *some-best some-worst* $\succ[\mathcal{R}]$ *mix-pmf* (*my-U* q) *some-best some-worst* $\longleftrightarrow p \succ[\mathcal{R}]\ q$
⟨*proof*⟩

lemma *my-U-is-linear-function*:
  assumes $p \in \mathcal{P}$
    and $q \in \mathcal{P}$
    and $\alpha \in \{0..1\}$
  assumes *some-best* $\succ[\mathcal{R}]$ *some-worst*
  shows *my-U* (*mix-pmf* α p q) $= \alpha *$ *my-U* $p + (1 - \alpha) *$ *my-U* q
⟨*proof*⟩

Now we define a more general Utility function that also takes the degenerate case into account

private definition *general-U*
  where
    general-U p = (*if some-best* $\approx[\mathcal{R}]$ *some-worst then 1 else my-U* p)

lemma *general-U-is-linear-function*:
  assumes $p \in \mathcal{P}$

and $q \in \mathcal{P}$
and $\alpha \in \{0..1\}$
shows *general-U* (*mix-pmf* α *p q*) $= \alpha * ($*general-U p*$) + (1 - \alpha) * ($*general-U q*$)$
⟨*proof*⟩

lemma *general-U-ordinal-Utility*:
shows *ordinal-utility* $\mathcal{P}$ $\mathcal{R}$ *general-U*
⟨*proof*⟩

Proof of the linearity of general-U. If we consider the definition of expected utility functions from Maschler, Solan, Zamir we are done.

theorem *is-linear*:
assumes $p \in \mathcal{P}$
and $q \in \mathcal{P}$
and $\alpha \in \{0..1\}$
shows $\exists u.\ u$ (*mix-pmf* α *p q*) $= \alpha * (u\ p) + (1-\alpha) * (u\ q)$
⟨*proof*⟩

Now I define a Utility function that assigns a utility to all outcomes. These are only finitely many

private definition *ocU*
where
ocU p = *general-U* (*return-pmf p*)

lemma *geral-U-is-expected-value-of-ocU*:
assumes *set-pmf p* $\subseteq$ *outcomes*
shows *general-U p* = *measure-pmf.expectation p ocU*
⟨*proof*⟩

lemma *ordinal-utility-expected-value*:
ordinal-utility $\mathcal{P}$ $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x ocU*)
⟨*proof*⟩

lemma *ordinal-utility-expected-value'*:
$\exists u.$ *ordinal-utility* $\mathcal{P}$ $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x u*)
⟨*proof*⟩

lemma *ocU-is-expected-utility-bernoulli*:
shows $\forall x \in \mathcal{P}.\ \forall y \in \mathcal{P}.\ x \succeq[\mathcal{R}]\ y \longleftrightarrow$
measure-pmf.expectation x ocU $\geq$ *measure-pmf.expectation y ocU*
⟨*proof*⟩

end

end

end

lemma *expected-value-is-utility-function*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq$ {}
assumes $x \in$ *lotteries-on outcomes* **and** $y \in$ *lotteries-on outcomes*
assumes *ordinal-utility* (*lotteries-on outcomes*) $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x u*)
shows *measure-pmf.expectation x u* $\geq$ *measure-pmf.expectation y u* $\longleftrightarrow$ $x \succeq[\mathcal{R}]$ y (**is** *?L* $\longleftrightarrow$ *?R*)
⟨*proof*⟩

lemma *system-U-implies-vNM-utility*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq$ {}
assumes *rpr*: *rational-preference* (*lotteries-on outcomes*) $\mathcal{R}$
assumes *ind*: *independent-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
assumes *cnt*: *continuous-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
shows $\exists$ *u*. *ordinal-utility* (*lotteries-on outcomes*) $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x u*)
⟨*proof*⟩

lemma *vNM-utility-implies-rationality*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq$ {}
assumes $\exists$ *u*. *ordinal-utility* (*lotteries-on outcomes*) $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x u*)
shows *rational-preference* (*lotteries-on outcomes*) $\mathcal{R}$
⟨*proof*⟩

theorem *vNM-utility-implies-independence*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq$ {}
assumes $\exists$ *u*. *ordinal-utility* (*lotteries-on outcomes*) $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x u*)
shows *independent-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
⟨*proof*⟩

lemma *exists-weight-for-equality*:
assumes $a > c$ **and** $a \geq b$ **and** $b \geq c$
shows $\exists (e::real) \in \{0..1\}.\ (1-e) * a + e * c = b$
⟨*proof*⟩

lemma *vNM-utilty-implies-continuity*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq$ {}
assumes $\exists$ *u*. *ordinal-utility* (*lotteries-on outcomes*) $\mathcal{R}$ ($\lambda x.$ *measure-pmf.expectation x u*)
shows *continuous-vnm* (*lotteries-on outcomes*) $\mathcal{R}$
⟨*proof*⟩

theorem *Von-Neumann-Morgenstern-Utility-Theorem*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq$ {}

shows *rational-preference* (*lotteries-on outcomes*) $\mathcal{R}$ ∧
independent-vnm (*lotteries-on outcomes*) $\mathcal{R}$ ∧
continuous-vnm (*lotteries-on outcomes*) $\mathcal{R}$ ⟷
(∃ *u*. *ordinal-utility* (*lotteries-on outcomes*) $\mathcal{R}$ (λ*x*. *measure-pmf.expectation x u*))
⟨*proof*⟩

end

theory *Expected-Utility*
imports
Neumann-Morgenstern-Utility-Theorem
begin

6 Definition of vNM-utility function

We define a version of the vNM Utility function using the locale mechanism. Currently this definition and system U have no proven relation yet.

Important: u is actually not the von Neuman Utility Function, but a Bernoulli Utility Function. The Expected value p given u is the von Neumann Utility Function.

locale *vNM-utility* =
fixes *outcomes* :: *'a set*
fixes *relation* :: *'a pmf relation*
fixes *u* :: *'a* ⇒ *real*
assumes *relation* ⊆ (*lotteries-on outcomes* × *lotteries-on outcomes*)
assumes ⋀*p q*. *p* ∈ *lotteries-on outcomes* ⟹
q ∈ *lotteries-on outcomes* ⟹
p ≿[*relation*] *q* ⟷ *measure-pmf.expectation p u* ≥ *measure-pmf.expectation q u*
begin

lemma *vNM-utilityD*:
shows *relation* ⊆ (*lotteries-on outcomes* × *lotteries-on outcomes*)
and *p* ∈ *lotteries-on outcomes* ⟹ *q* ∈ *lotteries-on outcomes* ⟹
p ≿[*relation*] *q* ⟷ *measure-pmf.expectation p u* ≥ *measure-pmf.expectation q u*
⟨*proof*⟩

lemma *not-outside*:
assumes *p* ≿[*relation*] *q*
shows *p* ∈ *lotteries-on outcomes*
and *q* ∈ *lotteries-on outcomes*
⟨*proof*⟩

lemma *utility-ge*:
  assumes $p \succeq$[*relation*] q
  shows *measure-pmf.expectation p u* $\geq$ *measure-pmf.expectation q u*
  ⟨*proof*⟩

end

sublocale *vNM-utility* $\subseteq$ *ordinal-utility* (*lotteries-on outcomes*) *relation* (λp. *measure-pmf.expectation p u*)
⟨*proof*⟩

context *vNM-utility*
begin

lemma *strict-preference-iff-strict-utility*:
 assumes $p \in$ *lotteries-on outcomes*
 assumes $q \in$ *lotteries-on outcomes*
 shows $p \succ$[*relation*] $q \longleftrightarrow$ *measure-pmf.expectation p u* $>$ *measure-pmf.expectation q u*
  ⟨*proof*⟩

lemma *pos-distrib-left*:
  assumes $c > 0$
  shows ($\sum$ $z \in$*outcomes*. *pmf q z* $*$ (c $*$ *u z*)) = c $*$ ($\sum$ $z \in$*outcomes*. *pmf q z* $*$ (*u z*))
⟨*proof*⟩

lemma *sum-pmf-util-commute*:
  ($\sum$ $a \in$*outcomes*. *pmf p a* $*$ *u a*) = ($\sum$ $a \in$*outcomes*. *u a* $*$ *pmf p a*)
  ⟨*proof*⟩

7 Finite outcomes

context
  assumes *fnt*: *finite outcomes*
begin

lemma *sum-equals-pmf-expectation*:
  assumes $p \in$ *lotteries-on outcomes*
  shows($\sum$ $z \in$*outcomes*. (*pmf p z*) $*$ (*u z*)) = *measure-pmf.expectation p u*
⟨*proof*⟩

lemma *expected-utility-weak-preference*:
  assumes $p \in$ *lotteries-on outcomes*
    and $q \in$ *lotteries-on outcomes*
 shows $p \succeq$[*relation*] $q \longleftrightarrow$ ($\sum$ $z \in$*outcomes*. (*pmf p z*) $*$ (*u z*)) $\geq$ ($\sum$ $z \in$*outcomes*. (*pmf q z*) $*$ (*u z*))
  ⟨*proof*⟩

lemma *diff-leq-zero-weak-preference*:
assumes $p \in$ *lotteries-on outcomes*
and $q \in$ *lotteries-on outcomes*
shows $p \succeq q \longleftrightarrow ((\sum a{\in}outcomes.\ pmf\ q\ a * u\ a) - (\sum a{\in}outcomes.\ pmf\ p\ a * u\ a) \leq 0)$
$\langle proof \rangle$

lemma *expected-utility-strict-preference*:
assumes $p \in$ *lotteries-on outcomes*
and $q \in$ *lotteries-on outcomes*
shows $p \succ[relation]\ q \longleftrightarrow$ *measure-pmf.expectation* $p\ u >$ *measure-pmf.expectation* $q\ u$
$\langle proof \rangle$

lemma *scale-pos-left*:
assumes $c > 0$
shows *vNM-utility outcomes relation* $(\lambda x.\ c * u\ x)$
$\langle proof \rangle$

lemma *strict-alt-def*:
assumes $p \in$ *lotteries-on outcomes*
and $q \in$ *lotteries-on outcomes*
shows $p \succ[relation]\ q \longleftrightarrow$
$(\sum z{\in}outcomes.\ (pmf\ p\ z) * (u\ z)) > (\sum z{\in}outcomes.\ (pmf\ q\ z) * (u\ z))$
$\langle proof \rangle$

lemma *strict-alt-def-utility-g*:
assumes $p \succ[relation]\ q$
shows $(\sum z{\in}outcomes.\ (pmf\ p\ z) * (u\ z)) > (\sum z{\in}outcomes.\ (pmf\ q\ z) * (u\ z))$
$\langle proof \rangle$

end

end

lemma *vnm-utility-is-ordinal-utility*:
assumes *vNM-utility outcomes relation u*
shows *ordinal-utility* (*lotteries-on outcomes*) *relation* ($\lambda p.$ *measure-pmf.expectation* $p\ u$)
$\langle proof \rangle$

lemma *vnm-utility-imp-reational-prefs*:
assumes *vNM-utility outcomes relation u*
shows *rational-preference* (*lotteries-on outcomes*) *relation*
$\langle proof \rangle$

theorem *expected-utilty-theorem-form-vnm-utility*:
assumes *fnt*: *finite outcomes* **and** *outcomes* $\neq \{\}$
shows *rational-preference* (*lotteries-on outcomes*) $\mathcal{R} \wedge$

independent-vnm (*lotteries-on outcomes*) $\mathcal{R}$ $\wedge$
continuous-vnm (*lotteries-on outcomes*) $\mathcal{R}$ $\longleftrightarrow$
($\exists$ *u*. *vNM-utility outcomes* $\mathcal{R}$ *u*)
⟨*proof*⟩

end

8 Related work

Formalizations in Social choice theory has been formalized by Wiedijk [13], Nipkow [7], and Gammie [4, 5]. Vestergaard [12], Le Roux, Martin-Dorel, and Soloviev [10, 11] provide formalizations of results in game theory. A library for algorithmic game theory in Coq is described in[1].

Related work in economics includes the verification of financial systems [9], binomial pricing models [3], and VCG-Auctions [6]. In microeconomics we discussed a formalization of two economic models and the First Welfare Theorem [8].

To our knowledge the only work that uses expected utility theory is that of Eberl [2]. Since we focus on the underlying theory of expected utility, we found that there is only little overlap.

References

[1] A. Bagnall, S. Merten, and G. Stewart. A library for algorithmic game theory in ssreflect/coq. *Journal of Formalized Reasoning*, 10(1):67–95, 2017.

[2] M. Eberl. Randomised social choice theory. *Archive of Formal Proofs*, May 2016. http://isa-afp.org/entries/Randomised_Social_Choice.shtml, Formal proof development.

[3] M. Echenim and N. Peltier. The binomial pricing model in finance: A formalization in isabelle. In L. de Moura, editor, *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6-11, 2017, Proceedings*, volume 10395 of *LNCS*, pages 546–562. Springer, 2017.

[4] P. Gammie. Some classical results in social choice theory. *Archive of Formal Proofs*, Nov. 2008. http://isa-afp.org/entries/SenSocialChoice.html, Formal proof development.

[5] P. Gammie. Stable matching. *Archive of Formal Proofs*, Oct. 2016. http://isa-afp.org/entries/Stable_Matching.html, Formal proof development.

[6] M. Kerber, C. Lange, C. Rowat, and W. Windsteiger. Developing an auction theory toolbox. *AISB 2013*, pages 1–4, 2013.

[7] T. Nipkow. Arrow and Gibbard-Satterthwaite. *Archive of Formal Proofs*, 2008.

[8] J. Parsert and C. Kaliszyk. Formal Microeconomic Foundations and the First Welfare Theorem. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2018, pages 91–101. ACM, 2018.

[9] G. O. Passmore and D. Ignatovich. Formal verification of financial algorithms. In L. de Moura, editor, *Automated Deduction – CADE 26*, pages 26–41. Springer, 2017.

[10] S. L. Roux. Acyclic Preferences and Existence of Sequential Nash Equilibria: A formal and constructive equivalence. In S. Berghofer, T. Nipkow, C. Urban, and M. Wenzel, editors, *Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings*, volume 5674 of *LNCS*, pages 293–309. Springer, 2009.

[11] S. L. Roux, É. Martin-Dorel, and J. Smaus. An existence theorem of Nash Equilibrium in Coq and Isabelle. In P. Bouyer, A. Orlandini, and P. S. Pietro, editors, *Proceedings Eighth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2017, Roma, Italy, 20-22 September 2017.*, volume 256 of *EPTCS*, pages 46–60, 2017.

[12] R. Vestergaard. A constructive approach to sequential nash equilibria. *Inf. Process. Lett.*, 97(2):46–51, 2006.

[13] F. Wiedijk. Formalizing Arrow's theorem. *Sadhana*, 34(1):193–220, Feb 2009.