# Von Neumann Morgenstern Utility Theorem * 

Julian Parsert Cezary Kaliszyk

September 13, 2023


#### 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

[^0]```
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 \(\Rightarrow{ }^{\prime} a p m f \Rightarrow{ }^{\prime} a p m f \Rightarrow{ }^{\prime} a p m f\) where mix－pmf \(\alpha\) p \(q=(\) bernoulli－pmf \(\alpha) \gg(\lambda X\) ．if \(X\) then \(p\) else \(q)\)
lemma pmf－mix：\(a \in\{0 . .1\} \Longrightarrow p m f(m i x-p m f\) a \(p q) x=a * p m f p x+(1-\) a）＊\(p m f q x\) \(\langle p r o o f\rangle\)
lemma pmf－mix－deeper：\(a \in\{0 . .1\} \Longrightarrow p m f(\) mix－pmf a \(p q) x=a * p m f p x+\) \(p m f q x-a * p m f q x\)〈proof〉
lemma bernoulli－pmf－0［simp］：bernoulli－pmf \(0=\) return－pmf False〈proof〉
lemma bernoulli－pmf－1［simp］：bernoulli－pmf \(1=\) return－pmf True \(\langle\) proof \(\rangle\)
lemma pmf－mix－0［simp］：mix－pmf \(0 p q=q\)〈proof〉
lemma pmf－mix－1［simp］：mix－pmf \(1 p q=p\) \(\langle p r o o f\rangle\)
lemma set－pmf－mix：\(a \in\{0<. .<1\} \Longrightarrow\) set－pmf（mix－pmf a p q）\(=\) set－pmf \(p \cup\) set－pmf \(q\)
\(\langle\) proof〉
lemma set－pmf－mix－eq：\(a \in\{0 . .1\} \Longrightarrow\) mix－pmf a \(p p=p\)
\(\langle p r o o f\rangle\)
lemma pmf－equiv－intro［intro］：
assumes \(\bigwedge e . e \in \operatorname{set}-p m f p \Longrightarrow p m f p e=p m f q e\)
assumes \(\bigwedge e . e \in \operatorname{set}-p m f q \Longrightarrow p m f q e=p m f p e\)
```

```
        shows p=q
        <proof>
lemma pmf-equiv-intro1[intro]:
    assumes \e.e set-pmf p\Longrightarrowpmf pe=pmf qe
    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\rangle
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 (\beta/\alpha)(mix-pmf \alpha pq)q) e= \beta*pmf pe+pmfqe-
\beta*pmf qe
\langleproof\rangle
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 (\beta/\alpha) (mix-pmf \alpha p q) q= mix-pmf \beta pq(is ?l = ?r)
<proof\rangle
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\rangle
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\rangle
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 ( }\existsx\mathrm{ . pmf p x = 1) }\longleftrightarrow card (set-pmf p)=
<proof\rangle
```

We thank Manuel Eberl for suggesting the following two lemmas.

```
lemma mix-pmf-partition:
    fixes p :: 'a pmf
    assumes }y\in\mathrm{ set-pmf p set-pmf p-{y} = {}
    obtains a q where }a\in{0<..<1} set-pmf q = set-pmf p-{y
        p= mix-pmf a q (return-pmf y)
<proof\rangle
lemma pmf-mix-induct [consumes 2, case-names degenerate mix]:
    assumes finite A set-pmf p\subseteqA
    assumes degenerate: }\x.x\inA\LongrightarrowP(return-pmf x)
    assumes mix: }\quad\bigwedgepay.set-pmf p\subseteqA\Longrightarrowa\in{0<..<1}\Longrightarrowy\inA
        P p\LongrightarrowP(mix-pmf a p (return-pmf y))
    shows P p
<proof\rangle
lemma pmf-mix-induct' [consumes 2, case-names degenerate mix]:
    assumes finite A set-pmf p\subseteqA
    assumes degenerate: \x. x \inA\LongrightarrowP(return-pmf x)
    assumes mix: }\quad\bigwedgepqa.set-pmf p\subseteqA\Longrightarrow set-pmf q\subseteqA\Longrightarrowa\in{0<..<1
#
    Pp\LongrightarrowPq\LongrightarrowP(mix-pmf a pq)
    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 (\sumi\in set-pmf (mix-pmf a p q).pmf (mix-pmf a p q) i) = (\sumi\inset-pmf
p. a * pmf pi)+(\sumi\inset-pmf q. (1-a) * pmf q i)
<proof\rangle
lemma distribute-alpha-over-sum:
    shows (\sumi\inset-pmf T. a*pmf pi*fi)=a*(\sumi\inset-pmf T.pmf pi*fi)
    <proof>
lemma sum-over-subset-pmf-support:
    assumes finite T
    assumes set-pmf p\subseteqT
    shows (\sumi\inT.a*pmf pi*fi)=(\sumi\inset-pmf p.a*pmf pi*fi)
<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
pf+(1-a)* measure-pmf.expectation qf
```

```
\langleproof\rangle
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 pq)f=a* measure-pmf.expectation
pf+(1-a)* measure-pmf.expectation qf
<proof\rangle
end
```

```
theory Lotteries
    imports
        PMF-Composition
        HOL-Probability.Probability
begin
```


## 2 Lotteries

definition lotteries-on
where
lotteries-on $O c=\{p \cdot($ set-pmf $p) \subseteq O c\}$
lemma lotteries-on-subset:
assumes $A \subseteq B$
shows lotteries-on $A \subseteq$ lotteries-on $B$
$\langle p r o o f\rangle$
lemma support-in-outcomes:
$\forall o c . \forall p \in$ lotteries-on oc. $\forall a \in$ set-pmf $p . a \in o c$
$\langle p r o o f\rangle$
lemma lotteries-on-nonempty:
assumes outcomes $\neq\{ \}$
shows lotteries-on outcomes $\neq\{ \}$
〈proof〉
lemma finite-support-one-oc:
assumes card outcomes $=1$
shows $\forall l \in$ lotteries-on outcomes. finite (set-pmf l)
$\langle p r o o f\rangle$
lemma one-outcome-card-support-1:
assumes card outcomes $=1$
shows $\forall l \in$ lotteries-on outcomes. card $($ set-pmf $l)=1$
$\langle p r o o f\rangle$

```
lemma finite-nempty-ex-degernate-in-lotteries:
    assumes out }\not={
    assumes finite out
    shows \existse\in lotteries-on out. \existsx\in out.pmf e x=1
<proof\rangle
lemma card-support-1-probability-1:
    assumes card (set-pmf p)=1
    shows }\foralle\in\mathrm{ set-pmf p.pmf pe=1
    \langleproof\rangle
lemma one-outcome-card-lotteries-1:
    assumes card outcomes =1
    shows card (lotteries-on outcomes) = 1
\langleproof\rangle
lemma return-pmf-card-equals-set:
    shows card {return-pmf x |x. x \inS} = card S
\langleproof\rangle
lemma mix-pmf-in-lotteries:
    assumes p\inlotteries-on A
        and q\in lotteries-on A
    and }a\in{0<..<1
    shows (mix-pmf a p q) \in lotteries-on A
\langleproof\rangle
lemma card-degen-lotteries-equals-outcomes:
    shows card {x\inlotteries-on out. card (set-pmf x)=1}=card out
\langleproof\rangle
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 $\mathrm{q} / \mathrm{p}$ changes

```
definition independent-vnm
    where
    independent-vnm C \(P=\)
    \((\forall p \in C . \forall q \in C . \forall r \in C . \forall(\alpha::\) real \() \in\{0<. .1\} . p \succeq[P] q \longleftrightarrow\) mix- \(p m f \alpha p\)
\(r \succeq[P]\) mix-pmf \(\alpha q r\) )
lemma independent-vnmI1:
    assumes \((\forall p \in C . \forall q \in C . \forall r \in C . \forall \alpha \in\{0<. .1\} . p \succeq[P] q \longleftrightarrow\) mix-pmf \(\alpha\)
\(p r \succeq[P]\) mix-pmf \(\alpha q r\) )
    shows independent-vnm C P
    〈proof〉
```

lemma independent-vnmI2:
assumes $\bigwedge p q r \alpha . p \in C \Longrightarrow q \in C \Longrightarrow r \in C \Longrightarrow \alpha \in\{0<. .1\} \Longrightarrow p \succeq[P]$
$q \longleftrightarrow$ mix-pmf $\alpha$ pr $\succeq[P]$ mix-pmf $\alpha q r$
shows independent-vnm C P
〈proof〉
lemma independent-vnm-alt-def:
shows independent-vnm $C P \longleftrightarrow(\forall p \in C . \forall q \in C . \forall r \in C . \forall \alpha \in\{0<. .<1\}$.
$p \succeq[P] q \longleftrightarrow$ mix-pmf $\alpha$ pr $\succeq[P]$ mix-pmf $\alpha q r$ ) (is ? $L \longleftrightarrow ? R$ )
$\langle p r o o f\rangle$
lemma independece-dest-alt:
assumes independent-vnm $C P$
shows $(\forall p \in C . \forall q \in C . \forall r \in C . \forall(\alpha::$ real $) \in\{0<. .1\} . p \succeq[P] q \longleftrightarrow$ mix-pmf
$\alpha p r \succeq[P]$ mix-pmf $\alpha q r$ )
$\langle p r o o f\rangle$
lemma independent-vnmD1:
assumes independent-vnm $C P$
shows $(\forall p \in C . \forall q \in C . \forall r \in C . \forall \alpha \in\{0<. .1\} . p \succeq[P] q \longleftrightarrow$ mix-pmf $\alpha p$
$r \succeq[P]$ mix-pmf $\alpha q r$ )
$\langle p r o o f\rangle$
lemma independent-vnmD2:
fixes $p q r \alpha$
assumes $\alpha \in\{0<. .1\}$
and $p \in C$
and $q \in C$
and $r \in C$
assumes independent-vnm $C P$
assumes $p \succeq[P] q$
shows mix-pmf $\alpha p r \succeq[P]$ mix-pmf $\alpha q r$
$\langle p r o o f\rangle$
lemma independent-vnmD3:

```
fixes p qr \alpha
assumes }\alpha\in{0<..1
    and p\inC
    and}q\in
    and}r\in
assumes independent-vnm C P
assumes mix-pmf \alpha pr\succeq[P] mix-pmf \alphaqr
shows p\succeq[P] q
<proof\rangle
lemma independent-vnmD4:
    assumes independent-vnm C P
    assumes refl-on C P
    assumes p\inC
    and q\inC
    and}r\in
    and \alpha}\in{0..1
    and p\succeq[P]q
shows mix-pmf \alpha pr\succeq[P] mix-pmf \alpha qr
<proof\rangle
lemma approx-indep-ge:
    assumes }x\approx[\mathcal{R}]
    assumes }\alpha\in{0..(1::real)
    assumes rpr: rational-preference (lotteries-on outcomes) \mathcal{R}
    and ind: independent-vnm (lotteries-on outcomes) }\mathcal{R
    shows }\forallr\inlotteries-on outcomes.(mix-pmf \alpha y r)\succeq[\mathcal{R}](mix-pmf \alpha x r)
<proof\rangle
lemma approx-imp-approx-ind:
    assumes }x\approx[\mathcal{R}]
    assumes }\alpha\in{0..(1::real)
    assumes rpr: rational-preference (lotteries-on outcomes) \mathcal{R}
    and ind: independent-vnm (lotteries-on outcomes) \mathcal{R}
    shows }\forallr\inlotteries-on outcomes. (mix-pmf \alpha yr)\approx[\mathcal{R}](mix-pmf \alphaxr)
    <proof\rangle
lemma geq-imp-mix-geq-right:
    assumes }x\succeq[\mathcal{R}]
    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-pmf \alpha x y)\succeq[\mathcal{R}] y
<proof\rangle
lemma geq-imp-mix-geq-left:
assumes \(x \succeq[\mathcal{R}] y\)
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-pmf \alpha y x)\succeq[\mathcal{R}] y
<proof>
lemma sg-imp-mix-sg:
    assumes }x\succ[\mathcal{R}]
    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-pmf \alpha x y)\succ[\mathcal{R}] y
<proof\rangle
```


### 3.2 Continuity

Continuity is sometimes called Archimedean Axiom

```
definition continuous-vnm
where
    continuous-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-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-pmf \(\alpha\) pr) \(\approx[P] q))\)
\(\langle\) proof〉
lemma continuous-vnmI:
```

```
assumes \bigwedgepqr.p\inC\Longrightarrowq\inC\Longrightarrowr\inC\Longrightarrowp\succeq[P] q^q\succeq[P]r\Longrightarrow
```

assumes \bigwedgepqr.p\inC\Longrightarrowq\inC\Longrightarrowr\inC\Longrightarrowp\succeq[P] q^q\succeq[P]r\Longrightarrow
\exists\alpha\in{0..1}.(mix-pmf \alphapr)\approx[P]q
\exists\alpha\in{0..1}.(mix-pmf \alphapr)\approx[P]q
shows continuous-vnm C P
shows continuous-vnm C P
<proof\rangle
<proof\rangle
lemma mix-in-lot:
lemma mix-in-lot:
assumes x l lotteries-on outcomes
assumes x l lotteries-on outcomes
and y}\inlotteries-on outcome
and y}\inlotteries-on outcome
and \alpha}\in{0..1
and \alpha}\in{0..1
shows (mix-pmf \alpha x y) \in lotteries-on outcomes
shows (mix-pmf \alpha x y) \in lotteries-on outcomes
<proof\rangle

```
<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-pmf $\alpha p r$
$\langle p r o o f\rangle$

## 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\Longrightarrowx\in\mathcal{P}\wedge y\in\mathcal{P}
    \langleproof\rangle
lemma mix-pmf-preferred-independence:
    assumes r f\mathcal{P}
        and \alpha}\in{0..1
    assumes p\succeq[\mathcal{R}]q
    shows mix-pmf \alpha pr\succeq[\mathcal{R}] mix-pmf \alphaqr
    <proof>
lemma mix-pmf-strict-preferred-independence:
    assumes r\in\mathcal{P}
        and }\alpha\in{0<..1
    assumes p}\succ[\mathcal{R}]
    shows mix-pmf \alpha pr\succ[\mathcal{R}] mix-pmf \alpha qr
    <proof\rangle
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 pr\succeq[\mathcal{R}] mix-pmf \alpha qr
    shows p}\succeq[\mathcal{R}]
<proof\rangle
lemma x-sg-y-sg-mpmf-right:
    assumes }x\succ[\mathcal{R}]
    assumes }b\in{0<..(1::\mathrm{ real ) }
    shows }x\succ[\mathcal{R}] mix-pmf b y x
<proof>
lemma neumann-3B-b:
```

```
    assumes }u\succ[\mathcal{R}]
    assumes }\alpha\in{0<..<1
    shows }u\succ[\mathcal{R}]mix-pmf \alphau
<proof>
lemma neumann-3B-b-non-strict:
    assumes }u\succeq[\mathcal{R}]
    assumes \alpha\in{0..1}
    shows u\succeq[\mathcal{R}] mix-pmf \alphauv
<proof\rangle
lemma greater-mix-pmf-greater-step-1-aux:
    assumes v}\succ[\mathcal{R}]
    assumes }\alpha\in{0<..<(1::\mathrm{ real ) }
    and }\beta\in{0<..<(1::\mathrm{ real ) }
assumes \beta>\alpha
shows (mix-pmf \beta}vu)\succ[\mathcal{R}](mix-pmf \alphavu
<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 \((\) mix-pmf \(\alpha x y) \succ[\mathcal{R}](\) mix-pmf \(\beta x y)\)
\(\langle p r o o f\rangle\)
```

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 mix-pmf \(\alpha\) pr \(\approx[\mathcal{R}]\) mix-pmf \(\alpha q r\)
    \(\langle p r o o f\rangle\)
```

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 \alpha r p\approx[\mathcal{R}] mix-pmf \alpha rq
<proof\rangle
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 \alpha p q\succeq[\mathcal{R}]w
\langleproof\rangle
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 \alphapq
<proof\rangle definition degenerate-lotteries where
    degenerate-lotteries ={x\in\mathcal{P}.card (set-pmf x)=1}
private definition best where
    best }={x\in\mathcal{P}.(\forally\in\mathcal{P}.x\succeq[\mathcal{R}]y)
private definition worst where
    worst }={x\in\mathcal{P}.(\forally\in\mathcal{P}.y\succeq[\mathcal{R}]x)
lemma degenerate-total:
    \foralle\indegenerate-lotteries.}\forallm\in\mathcal{P}.e\succeq[\mathcal{R}]m\veem\succeq[\mathcal{R}]
    <proof\rangle
lemma degen-outcome-cardinalities:
    card degenerate-lotteries = card outcomes
    <proof\rangle
lemma degenerate-lots-subset-all: degenerate-lotteries }\subseteq\mathcal{P
    <proof\rangle
lemma alt-definition-of-degenerate-lotteries[iff]:
    {return-pmf x |x. x\in outcomes } = degenerate-lotteries
<proof\rangle
lemma best-indifferent:
    \forallx\in best. }\forally\in\mathrm{ best. }x\approx[\mathcal{R}]
    \langleproof\rangle
lemma worst-indifferent:
    \forallx\in worst. }\forally\in\mathrm{ worst. }x\approx[\mathcal{R}]
    <proof>
lemma best-worst-indiff-all-indiff:
    assumes b}\in\mathrm{ best
```

```
    and w\in worst
    and b\approx[\mathcal{R}]w
    shows }\foralle\in\mathcal{P}.e\approx[\mathcal{R}]w\foralle\in\mathcal{P}.e\approx[\mathcal{R}]
<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 \(\beta b w\)
\(\langle p r o o f\rangle\)
```


### 5.1 Add finiteness and non emptyness of outcomes

```
context
    assumes fnt: finite outcomes
    assumes nempty:outcomes }\not={{
begin
lemma finite-degenerate-lotteries:
    finite degenerate-lotteries
    <proof>
lemma degenerate-has-max-preferred:
    {x\in degenerate-lotteries. (\forally\indegenerate-lotteries. }x\succeq[\mathcal{R}]y)}\not={}(is ?l l
{})
<proof\rangle
lemma degenerate-has-min-preferred:
    {x\in degenerate-lotteries. ( }\forally\in\mathrm{ degenerate-lotteries. y }\succeq[\mathcal{R}]x)}\not={} (is ?l l=
{})
<proof>
lemma exists-best-degenerate:
    \existsx\in degenerate-lotteries. }\forally\in\mathrm{ degenerate-lotteries. }x\succeq[\mathcal{R}]
    <proof\rangle
lemma exists-worst-degenerate:
    \existsx\indegenerate-lotteries.}\forally\in\mathrm{ degenerate-lotteries. }y\succeq[\mathcal{R}]
    <proof\rangle
```

```
lemma best-degenerate-in-best-overall:
    \existsx\in degenerate-lotteries. }\forally\in\mathcal{P}.x\succeq[\mathcal{R}]
<proof\rangle
lemma worst-degenerate-in-worst-overall:
    \existsx\in degenerate-lotteries..}\forally\in\mathcal{P}.y\succeq[\mathcal{R}]
<proof\rangle
lemma overall-best-nonempty:
    best }={{
    <proof\rangle
lemma overall-worst-nonempty:
    worst }\not={
    <proof\rangle
lemma trans-approx:
    assumes }x\approx[\mathcal{R}]
        and }y\approx[\mathcal{R}]
shows }x\approx[\mathcal{R}]
<proof>
First EXPLICIT use of the axiom of choice
```

```
private definition some-best where
```

private definition some-best where
some-best =(SOME x. x degenerate-lotteries }\wedgex\in\mathrm{ best )
private definition some-worst where
some-worst =(SOME x. x degenerate-lotteries ^ x\in worst)
private definition my-U :: 'a pmf => real
where
my-U p=(SOME \alpha. \alpha\in{0..1}\wedge p\approx[\mathcal{R}] mix-pmf \alpha some-best some-worst)
lemma exists-best-and-degenerate: degenerate-lotteries \cap best }\not={
<proof\rangle
lemma exists-worst-and-degenerate: degenerate-lotteries }\cap\mathrm{ worst }\not={
<proof\rangle
lemma some-best-in-best: some-best \in best
<proof\rangle
lemma some-worst-in-worst: some-worst \in worst
\langleproof\rangle

```
lemma best－always－at－least－as－good－mix：
assumes \(\alpha \in\{0 . .1\}\)
and \(p \in \mathcal{P}\)
shows mix－pmf \(\alpha\) 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 \(\alpha\) some－best some－worst \(\succeq[\mathcal{R}]\) mix－pmf \(\beta\) some－best some－worst〈proof〉
lemma gamma－inverse：
assumes \(\alpha \in\{0<. .<1\}\) and \(\beta \in\{0<. .<1\}\)
shows \((1::\) real \()-(\alpha-\beta) /(1-\beta)=(1-\alpha) /(1-\beta)\)
\(\langle p r o o f\rangle\)
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 \(\alpha b w\)
\(\langle p r o o f\rangle\)
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 \(\beta\) some－best some－worst \(\approx[\mathcal{R}]\) mix－pmf \(\alpha\) some－best some－worst
shows \(\beta=\alpha\)
\(\langle p r o o f\rangle\)
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 \(\beta\) some－best some－worst \(\succeq[\mathcal{R}]\) mix－pmf \(\alpha\) some－best some－worst shows \(\beta \geq \alpha\)
\(\langle p r o o f\rangle\)
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 \(\alpha x y) \succeq[\mathcal{R}](\) mix－pmf \(\beta x y)\)
```

<proof\rangle

```

\subsection*{5.2 Add continuity to assumptions}

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

\section*{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 $\alpha p r$
$\langle p r o o f\rangle$

```

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 $\alpha b w$
$\langle p r o o f\rangle$
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 $\alpha b w$
$\langle p r o o f\rangle$
lemma my-U-is-defined:
assumes $p \in \mathcal{P}$
shows my-U $p \in\{0 . .1\} p \approx[\mathcal{R}]$ mix-pmf ( $m y$ - $U p$ ) some-best some-worst
$\langle p r o o f\rangle$
lemma weak-pref-mix-with-my-U-weak-pref:
assumes $p \succeq[\mathcal{R}] q$
shows mix-pmf ( $m y$ - $U p$ ) 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 $(m y-U p)$ some-best some-worst $\succ[\mathcal{R}]$ mix-pmf (my-U q)
some-best some-worst
shows $m y-U p>m y-U q$

```
```

\langleproof\rangle
lemma geq-my-U-imp-weak-preference:
assumes p\in\mathcal{P}
and q\in\mathcal{P}
assumes some-best }\succ[\mathcal{R}] some-wors
assumes my-U p\geqmy-Uq
shows p\succeq[\mathcal{R}]q
\langleproof\rangle
lemma my-U-represents-pref:
assumes some-best }\succ[\mathcal{R}] some-wors
assumes p\in\mathcal{P}
and q\in\mathcal{P}
shows }p\succeq[\mathcal{R}]q\longleftrightarrowmy-U p\geqmy-U q(is ?L\longleftrightarrow \longleftrightarrowR
<proof\rangle
lemma first-iff-u-greater-strict-preff:
assumes p\in\mathcal{P}
and q\in\mathcal{P}
assumes some-best }\succ[\mathcal{R}] some-wors
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-wors
shows mix-pmf (my-U P) some-best some-worst }\succ[\mathcal{R}] mix-pmf (my-U q) some-bes
some-worst \longleftrightarrowp\succ[\mathcal{R}]q
<proof\rangle
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-wors
shows my-U (mix-pmf \alphapq) = \alpha*my-Up+(1-\alpha)*my-Uq
<proof\rangle
Now we define a more general Utility function that also takes the degenerate case into account

```
```

private definition general- }

```
private definition general- }
    where
    where
        general-U p =(if some-best }\approx[\mathcal{R}]\mathrm{ some-worst then 1 else my-U p)
        general-U p =(if some-best }\approx[\mathcal{R}]\mathrm{ some-worst then 1 else my-U p)
lemma general-U-is-linear-function:
lemma general-U-is-linear-function:
    assumes p}\in\mathcal{P
```

    assumes p}\in\mathcal{P
    ```
```

    and q\in\mathcal{P}
    and \alpha\in{0..1}
    shows general-U (mix-pmf \alpha pq)=\alpha*(general-U p)+(1-\alpha)*(general-U
    q)
<proof\rangle

```
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 $\exists u . u(m i x-p m f \alpha p q)=\alpha *(u p)+(1-\alpha) *(u q)$
$\langle p r o o f\rangle$

```

Now I define a Utility function that assigns a utility to all outcomes．These are only finitely many
private definition oc \(U\)
where
```

ocU p = general-U (return-pmf p)

```
lemma geral- \(U\)-is-expected-value-of-oc \(U\) :
    assumes set-pmf \(p \subseteq\) outcomes
    shows general- \(U p=\) measure-pmf.expectation \(p\) oc \(U\)
    〈proof〉
lemma ordinal-utility-expected-value:
    ordinal-utility \(\mathcal{P} \mathcal{R}(\lambda x\). measure-pmf.expectation \(x\) oc \(U)\)
\(\langle p r o o f\rangle\)
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 \(x\) oc \(U \geq\) measure-pmf.expectation \(y\) oc \(U\)
    〈proof〉
end
end
end
```

lemma expected-value-is-utility-function:
assumes fnt: finite outcomes and outcomes }\not={
assumes }x\inlotteries-on outcomes and y\in lotteries-on outcomes
assumes ordinal-utility (lotteries-on outcomes)}\mathcal{R}(\lambdax\mathrm{ . measure-pmf.expectation
x u)
shows measure-pmf.expectation x }u\geq\mathrm{ measure-pmf.expectation y }u\longleftrightarrowx\succeq[\mathcal{R}
y(is ?L \longleftrightarrow ?R)
<proof\rangle
lemma system-U-implies-vNM-utility:
assumes fnt: finite outcomes and outcomes }\not={
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 }\existsu\mathrm{ . ordinal-utility (lotteries-on outcomes) }\mathcal{R}(\lambdax\mathrm{ . measure-pmf.expectation
x u)
<proof>
lemma vNM-utility-implies-rationality:
assumes fnt: finite outcomes and outcomes }\not={
assumes \existsu. ordinal-utility (lotteries-on outcomes)}\mathcal{R}(\lambdax\mathrm{ . measure-pmf.expectation
x u)
shows rational-preference (lotteries-on outcomes) \mathcal{R}
<proof\rangle
theorem vNM-utility-implies-independence:
assumes fnt: finite outcomes and outcomes }\not={
assumes \existsu. ordinal-utility (lotteries-on outcomes) \mathcal{R}}\mathrm{ ( }\lambdax\mathrm{ . measure-pmf.expectation
x u)
shows independent-vnm (lotteries-on outcomes) }\mathcal{R
<proof\rangle
lemma exists-weight-for-equality:
assumes }a>c\mathrm{ and }a\geqb\mathrm{ and b \c
shows }\exists(e::real)\in{0..1}.(1-e)*a+e*c=
<proof\rangle
lemma vNM-utilty-implies-continuity:
assumes fnt: finite outcomes and outcomes }\not={
assumes \existsu. ordinal-utility (lotteries-on outcomes) \mathcal{R}}\mathrm{ ( }\lambdax\mathrm{ . 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 }\not={

```
```

shows rational-preference (lotteries-on outcomes) \mathcal{R}}
independent-vnm (lotteries-on outcomes) }\mathcal{R}
continuous-vnm (lotteries-on outcomes) \mathcal{R}\longleftrightarrow
( }\exists\mathrm{ u. ordinal-utility (lotteries-on outcomes) }\mathcal{R}(\lambdax.measure-pmf.expectation x
u))
<proof\rangle

```
end
```

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

```

\section*{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 $v N M$-utility $=$
fixes outcomes :: 'a set
fixes relation :: 'a pmf relation
fixes $u::{ }^{\prime} a \Rightarrow$ real
assumes relation $\subseteq$ (lotteries-on outcomes $\times$ lotteries-on outcomes)
assumes $\bigwedge p q . \quad p \in$ lotteries-on outcomes $\Longrightarrow$
$q \in$ lotteries-on outcomes $\Longrightarrow$

```
        \(p \succeq[\) relation \(] q \longleftrightarrow\) measure-pmf.expectation \(p u \geq\) measure-pmf.expectation
qu
begin
lemma vNM-utilityD:
    shows relation \(\subseteq\) (lotteries-on outcomes \(\times\) lotteries-on outcomes \()\)
        and \(p \in\) lotteries-on outcomes \(\Longrightarrow q \in\) lotteries-on outcomes \(\Longrightarrow\)
        \(p \succeq[\) relation \(] q \longleftrightarrow\) measure-pmf.expectation \(p u \geq\) measure-pmf.expectation \(q\)
\(u\)
    \(\langle p r o o f\rangle\)
lemma not-outside:
    assumes \(p \succeq[\) relation \(] ~ q\)
    shows \(p \in\) lotteries-on outcomes
        and \(q \in\) lotteries-on outcomes
\(\langle p r o o f\rangle\)
```

lemma utility-ge:
assumes p\succeq[relation] q
shows measure-pmf.expectation p u \geq measure-pmf.expectation qu
<proof>
end

```
sublocale \(v N M\)-utility \(\subseteq\) ordinal-utility (lotteries-on outcomes) relation ( \(\lambda\). mea-
sure-pmf.expectation \(p u\) )
\(\langle p r o o f\rangle\)
context \(v N M\)-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
qu
    \(\langle p r o o f\rangle\)
lemma pos-distrib-left:
    assumes \(c>0\)
    shows \(\left(\sum z \in\right.\) outcomes. pmf \(\left.q z *(c * u z)\right)=c *\left(\sum z \in\right.\) outcomes. pmf \(q z *(u\)
z))
\(\langle p r o o f\rangle\)
lemma sum-pmf-util-commute:
    ( \(\sum a \in\) outcomes. pmf \(\left.p a * u a\right)=\left(\sum a \in\right.\) outcomes. \(\left.u a * p m f p a\right)\)
    \(\langle\) proof〉

\section*{7 Finite outcomes}

\section*{context}
assumes fnt: finite outcomes
begin
lemma sum-equals-pmf-expectation:
assumes \(p \in\) lotteries-on outcomes
\(\operatorname{shows}\left(\sum z \in\right.\) outcomes. \((\) pmf \(\left.p z) *(u z)\right)=\) measure-pmf.expectation \(p u\)
\(\langle\) proof \(\rangle\)
lemma expected-utility-weak-preference:
assumes \(p \in\) lotteries-on outcomes
and \(q \in\) lotteries-on outcomes
shows \(p \succeq[\) relation \(] q \longleftrightarrow\left(\sum z \in\right.\) outcomes. \((\) pmf \(\left.p z) *(u z)\right) \geq\left(\sum z \in\right.\) outcomes.
\((p m f q z) *(u z))\) \(\langle p r o o f\rangle\)
```

lemma diff-leq-zero-weak-preference:
assumes p\in lotteries-on outcomes
and q\in lotteries-on outcomes
shows }p\succeqq\longleftrightarrow((\suma\inoutcomes.pmf qa*ua)-(\suma\inoutcomes. pmf p

* ua) \leq0)
\langleproof\rangle
lemma expected-utility-strict-preference:
assumes p\in lotteries-on outcomes
and q\in lotteries-on outcomes
shows }p\succ[\mathrm{ relation ] q}\longleftrightarrow measure-pmf.expectation p u> measure-pmf.expectation
qu
<proof>
lemma scale-pos-left:
assumes c>0
shows vNM-utility outcomes relation ( }\lambdax.c*ux
\langleproof\rangle
lemma strict-alt-def:
assumes p\in lotteries-on outcomes
and q\in lotteries-on outcomes
shows }p\succ[\mathrm{ relation] }q
(\sumz\inoutcomes. (pmf pz)*(uz))>(\sumz\inoutcomes. (pmf qz)* (uz))
<proof\rangle
lemma strict-alt-def-utility-g:
assumes p}\succ[\mathrm{ relation] q

```

```

<proof>
end
end
lemma vnm-utility-is-ordinal-utility:
assumes vNM-utility outcomes relation u
shows ordinal-utility (lotteries-on outcomes) relation ( }\lambda\mathrm{ p. measure-pmf.expectation
p u)
\langleproof\rangle
lemma vnm-utility-imp-reational-prefs:
assumes vNM-utility outcomes relation u
shows rational-preference (lotteries-on outcomes) relation
<proof\rangle
theorem expected-utilty-theorem-form-vnm-utility:
assumes fnt: finite outcomes and outcomes }\not={
shows rational-preference (lotteries-on outcomes) \mathcal{R}}

```
```

    independent-vnm (lotteries-on outcomes) \mathcal{R}^
    continuous-vnm(lotteries-on outcomes)}\mathcal{R}
    ( \existsu.vNM-utility outcomes \mathcal{R u})
    <proof\rangle
end

```

\section*{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.

\section*{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 \(L N C S\), 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.```


[^0]:    *This work is supported by the Austrian Science Fund (FWF) project P26201 and the European Research Council (ERC) grant no 714034 SMART.

