

The CHSH inequality: Tsirelson's upper-bound and other results

Mnacho Echenim and Mehdi Mhalla and Coraline Mori

March 17, 2025

Abstract

The CHSH inequality, named after Clauser, Horne, Shimony and Holt, was used by Alain Aspect to prove experimentally that Einstein's hypothesis stating that quantum mechanics could be defined using local hidden variables was incorrect. The CHSH inequality is based on a setting in which an experiment consisting of two separate parties performing joint measurements is run several times, and a score is derived from these runs. If the local hidden variable hypothesis had been correct, this score would have been bounded by 2, but a suitable choice of observables in a quantum setting permits to violate this inequality when measuring the Bell state; this is the result that Aspect obtained experimentally. Tsirelson answered the question of how large this violation could be by proving that in the quantum setting, $2\sqrt{2}$ is the highest score that can be obtained when running this experiment. Along with elementary results on density matrices which represent quantum states in the finite dimensional setting, we formalize Tsirelson's result and summarize the main results on the CHSH score:

1. Under the local hidden variable hypothesis, this score admits 2 as an upper-bound.
2. When the density matrix under consideration is separable, the upper-bound cannot be violated.
3. When one of the parties in the experiment performs measures using commuting observables, this upper-bound remains valid.
4. Otherwise, the upper-bound of this score is $2\sqrt{2}$, regardless of the observables that are used and the quantum state that is measured, and
5. This upper-bound is reached for a suitable choice of observables when measuring the Bell state.

Contents

1 Basic algebraic results

2

2	Results in linear algebra	5
3	Results on tensor products	10
4	Preliminary results	18
4.1	Commutator and anticommutator	18
5	Maximum modulus in a spectrum	19
5.1	Definition and basic properties for Hermitian matrices	19
5.2	Eigenvector for the element with maximum modulus	23
6	The \mathcal{L}_2 operator norm	24
6.1	Definition and preliminary results	24
6.2	The \mathcal{L}_2 operator norm is equal to the maximum singular value	26
6.3	Consequences for the \mathcal{L}_2 operator norm	28
7	On density matrices	30
7.1	Density matrix characterization	30
7.2	Separable density matrices	31
7.3	Characterization of pure states	32
8	Quantum expectation values and traces	33
9	CHSH inequalities	34
9.1	Some intermediate results for particular observables	34
9.2	The CHSH operator and expectation	35
9.3	CHSH inequality for separable density matrices	40
9.4	CHSH inequality for commuting observables	41
9.5	Result summary on the CHSH inequalities	41

```

theory Tensor-Mat-Compl-Properties
  imports
    Commuting-Hermitian.Spectral-Theory-Complements
    Projective-Measurements.Projective-Measurements
begin

```

1 Basic algebraic results

```

lemma pos-sum-gt-0:
  assumes finite I
  and  $\bigwedge i. i \in I \implies (0:: 'a :: linordered-field) \leq f i$ 
  and  $0 < \text{sum } f I$ 
  shows  $\exists j \in I. 0 < f j$ 
  <proof>

```

lemma *pos-square-1-elm*:
assumes *finite I*
and $\bigwedge i. i \in I \implies (0::real) \leq f\ i$
and $\text{sum } f\ I = 1$
and $\text{sum } (\lambda x. f\ x * f\ x)\ I = 1$
shows $\exists j \in I. f\ j = 1$
 $\langle \text{proof} \rangle$

lemma *cpx-pos-square-1-elm*:
assumes *finite I*
and $\bigwedge i. i \in I \implies (0::complex) \leq f\ i$
and $\text{sum } f\ I = 1$
and $\text{sum } (\lambda x. f\ x * f\ x)\ I = 1$
shows $\exists j \in I. f\ j = 1$
 $\langle \text{proof} \rangle$

lemma *sum-eq-elm*:
assumes *finite I*
and $\bigwedge i. i \in I \implies (0::'a :: \text{linordered-field}) \leq f\ i$
and $\text{sum } f\ I = c$
and $j \in I$
and $f\ j = c$
shows $\forall k \in (I - \{j\}). f\ k = 0$
 $\langle \text{proof} \rangle$

lemma *cpx-sum-eq-elm*:
assumes *finite I*
and $\bigwedge i. i \in I \implies (0::complex) \leq f\ i$
and $\text{sum } f\ I = c$
and $j \in I$
and $f\ j = c$
shows $\forall k \in (I - \{j\}). f\ k = 0$
 $\langle \text{proof} \rangle$

lemma *sum-nat-div-mod*:
shows $\text{sum } (\lambda i. \text{sum } (\lambda j. f\ i * g\ j)\ \{..< (m::nat)\})\ \{..< (n::nat)\} =$
 $\text{sum } (\lambda k. f\ (k\ \text{div}\ m) * g\ (k\ \text{mod}\ m))\ \{..< n*m\}$
 $\langle \text{proof} \rangle$

lemma *abs-cmod-eq*:
fixes $z::complex$
shows $|z| = cmod\ z$
 $\langle \text{proof} \rangle$

lemma *real-cpx-abs-leq*:
fixes $A::complex$
assumes $A \in \text{Reals}$
and $B \in \text{Reals}$
and $|A * B| \leq 1$

shows $|Re A * Re B| \leq 1$
<proof>

lemma *cpx-real-abs-eg*:
 fixes $z::complex$ **and** $r::real$
 assumes $z \in Reals$
 and $z = r$
shows $|z| = |r|$
<proof>

lemma *cpx-real-abs-leq*:
 fixes $z::complex$ **and** $r::real$
 assumes $z \in Reals$
 and $z = r$
 and $|r| \leq k$
shows $|z| \leq (k::real)$
<proof>

lemma *cpx-abs-mult-le-1*:
 fixes $z::complex$
 assumes $|z| \leq 1$
 and $|z'| \leq 1$
shows $|z*z'| \leq 1$
<proof>

lemma *sum-abs-cpx*:
 shows $|sum K I| \leq sum (\lambda x. |(K x)::complex|) I$
<proof>

lemma *abs-mult-cpx*:
 fixes $z::complex$
 assumes $0 \leq (a::real)$
 shows $|a*z| = a * |z|$
<proof>

lemma *cpx-ge-0-real*:
 fixes $c::complex$
 assumes $0 \leq c$
 and $c \in Reals$
shows $0 \leq Re c$
<proof>

lemma *cpx-of-real-ge-0*:
 assumes $0 \leq complex-of-real a$
 shows $0 \leq a$
<proof>

lemma *set-cst-list*:

shows $(\bigwedge i. i < \text{length } l \implies l[i] = x) \implies 0 < \text{length } l \implies \text{set } l = \{x\}$
 <proof>

lemma *pos-mult-Max*:

assumes *finite F*
and $F \neq \{\}$
and $0 \leq x$
and $\forall a \in F. 0 \leq (a::\text{real})$
shows $\text{Max.F } \{x * a \mid a. a \in F\} = x * \text{Max.F } F$
 <proof>

lemma *square-Max*:

assumes *finite A*
and $A \neq \{\}$
and $\forall a \in A. 0 \leq ((f a)::\text{real})$
and $b = \text{Max.F } \{f a \mid a. a \in A\}$
shows $\text{Max.F } \{f a * f a \mid a. a \in A\} = b * b$
 <proof>

lemma *ereal-Sup-switch*:

assumes $\forall m \in P. (b::\text{real}) \leq f m$
and $\forall m \in P. f m \leq (c::\text{real})$
and $P \neq \{\}$
shows $\text{ereal } (\text{Sup } (f ` P)) = (\bigsqcup m \in P. \text{ereal } (f m))$
 <proof>

lemma *Sup-ge-real*:

assumes $a \in (A::\text{real set})$
and $\forall a \in A. a \leq c$
and $\forall a \in A. b \leq a$
shows $a \leq \text{Sup } A$
 <proof>

lemma *Sup-real-le*:

assumes $\forall a \in (A::\text{real set}). a \leq c$
and $\forall a \in A. b \leq a$
and $A \neq \{\}$
shows $\text{Sup } A \leq c$
 <proof>

2 Results in linear algebra

lemma *mat-add-eq-0-if*:

fixes $A::'a :: \text{group-add Matrix.mat}$
assumes $A \in \text{carrier-mat } n \ m$
and $B \in \text{carrier-mat } n \ m$
and $A+B = 0_m \ n \ m$
shows $B = -A$
 <proof>

lemma *trace-rank-1-proj*:

shows $\text{Complex-Matrix.trace } (\text{rank-1-proj } v) = \|v\|^2$
<proof>

lemma *trace-ch-expand*:

fixes $A::'a::\{\text{minus,comm-ring}\} \text{Matrix.mat}$
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $C \in \text{carrier-mat } n \ n$
and $D \in \text{carrier-mat } n \ n$
shows $\text{Complex-Matrix.trace } (A - B + C + D) =$
 $\text{Complex-Matrix.trace } A - \text{Complex-Matrix.trace } B +$
 $\text{Complex-Matrix.trace } C + \text{Complex-Matrix.trace } D$
<proof>

lemma *squared-A-trace*:

assumes $A \in \text{carrier-mat } n \ n$
and *unitarily-equiv* $A \ B \ U$
shows $\text{Complex-Matrix.trace } (A * A) = \text{Complex-Matrix.trace } (B * B)$
<proof>

lemma *squared-A-trace'*:

assumes $A \in \text{carrier-mat } n \ n$
and *unitary-diag* $A \ B \ U$
shows $\text{Complex-Matrix.trace } (A * A) = (\sum i \in \{0 ..< n\}. (B \$\$ (i,i) * B \$\$ (i,i)))$
<proof>

lemma *positive-square-trace*:

assumes $A \in \text{carrier-mat } n \ n$
and $\text{Complex-Matrix.trace } A = (1::\text{real})$
and $\text{Complex-Matrix.trace } (A * A) = 1$
and *real-diag-decomp* $A \ B \ U$
and $\text{Complex-Matrix.positive } A$
and $0 < n$
shows $\exists j < n. B \$\$ (j,j) = 1 \wedge (\forall i < n. i \neq j \longrightarrow B \$\$ (i,i) = 0)$
<proof>

lemma *idty-square*:

shows $((1_m \ n)::'a :: \text{semiring-1 Matrix.mat}) * (1_m \ n) = 1_m \ n$
<proof>

lemma *pos-hermitian-trace-reals*:

fixes $A::\text{complex Matrix.mat}$
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
and $\text{Complex-Matrix.positive } A$

and *hermitian* B
shows *Complex-Matrix.trace* $(B*A) \in \text{Reals}$
 <proof>

lemma *pos-hermitian-trace-reals'*:
fixes $A::\text{complex Matrix.mat}$
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
and *Complex-Matrix.positive* A
and *hermitian* B
shows *Complex-Matrix.trace* $(A*B) \in \text{Reals}$
 <proof>

lemma *hermitian-commute*:
assumes *hermitian* A
and *hermitian* B
and $A*B = B*A$
shows *hermitian* $(A*B)$
 <proof>

lemma *idty-unitary-diag*:
assumes *unitary-diag* $(1_m \ n) \ B \ U$
shows $B = 1_m \ n$
 <proof>

lemma *diag-mat-idty*:
assumes $0 < n$
shows $\text{set } (\text{diag-mat } ((1_m \ n)::'a::\{\text{one,zero}\} \text{ Matrix.mat})) = \{1\}$
 (is ?L = ?R)
 <proof>

lemma *idty-spectrum*:
assumes $0 < n$
shows $\text{spectrum } ((1_m \ n)::\text{complex Matrix.mat}) = \{1\}$
 <proof>

lemma *spectrum-ne*:
fixes $A::\text{complex Matrix.mat}$
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{spectrum } A \neq \{\}$ <proof>

lemma *unitary-diag-square-spectrum*:
fixes $A::\text{complex Matrix.mat}$
assumes *hermitian* A
and $A \in \text{carrier-mat } n \ n$
and *unitary-diag* $A \ B \ U$

shows $\text{spectrum } (A * A) = \text{set } (\text{diag-mat } (B * B))$
 ⟨proof⟩

lemma *diag-mat-square-eq*:
fixes $B :: 'a :: \{\text{ring}\} \text{Matrix.mat}$
assumes *diagonal-mat* B
and $B \in \text{carrier-mat } n \ n$
shows $\text{set } (\text{diag-mat } (B * B)) = \{b * b \mid b. b \in \text{set } (\text{diag-mat } B)\}$
 ⟨proof⟩

lemma *hermitian-square-spectrum-eq*:
fixes $A :: \text{complex Matrix.mat}$
assumes *hermitian* A
and $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{spectrum } (A * A) = \{a * a \mid a. a \in \text{spectrum } A\}$
 ⟨proof⟩

lemma *adjoint-uminus*:
shows $\text{Complex-Matrix.adjoint } (-A) = - (\text{Complex-Matrix.adjoint } A)$
 ⟨proof⟩

lemma (*in fixed-carrier-mat*) *sum-mat-zero*:
assumes *finite* I
and $\bigwedge i. i \in I \implies A \ i \in \text{fc-mats}$
and $\bigwedge i. i \in I \implies f \ i = 0$
shows $\text{sum-mat } (\lambda \ i. (f \ i) \cdot_m (A \ i)) \ I = 0_m \ \text{dimR} \ \text{dimC}$ ⟨proof⟩

lemma (*in fixed-carrier-mat*) *sum-mat-zero'*:
fixes $A :: 'b \Rightarrow 'a \text{Matrix.mat}$
assumes *finite* I
and $\bigwedge i. i \in I \implies A \ i = 0_m \ \text{dimR} \ \text{dimC}$
shows $\text{sum-mat } A \ I = 0_m \ \text{dimR} \ \text{dimC}$ ⟨proof⟩

lemma (*in fixed-carrier-mat*) *sum-mat-remove*:
assumes $A \ 'I \subseteq \text{fc-mats}$
and $A: \text{finite } I$ **and** $x: x \in I$
shows $\text{sum-mat } A \ I = A \ x + \text{sum-mat } A \ (I - \{x\})$ ⟨proof⟩

lemma (*in fixed-carrier-mat*) *sum-mat-singleton*:
fixes $A :: 'b \Rightarrow 'a \text{Matrix.mat}$
assumes *finite* I
and $A \ 'I \subseteq \text{fc-mats}$
and $j \in I$
and $\forall i \in I. i \neq j \implies f \ i = 0$
shows $\text{sum-mat } (\lambda \ i. (f \ i) \cdot_m (A \ i)) \ I = f \ j \cdot_m (A \ j)$
 ⟨proof⟩

context *fixed-carrier-mat*

begin
lemma *sum-mat-disj-union*:
 assumes *finite J*
 and *finite I*
 and $I \cap J = \{\}$
 and $\forall i \in I \cup J. A\ i \in \text{fc-mats}$
shows $\text{sum-mat } A\ (I \cup J) = \text{sum-mat } A\ I + \text{sum-mat } A\ J$ *<proof>*

lemma *sum-with-reindex-cong'*:
 fixes $g :: 'c \Rightarrow 'a\ \text{Matrix.mat}$
 assumes $\forall x. g\ x \in \text{fc-mats}$
 and $\forall x. h\ x \in \text{fc-mats}$
 and *inj-on l B*
 and $\bigwedge x. x \in B \implies g\ (l\ x) = h\ x$
 shows $\text{sum-with } (+)\ (0_m\ \text{dimR}\ \text{dimC})\ g\ (l\ 'B) =$
 $\text{sum-with } (+)\ (0_m\ \text{dimR}\ \text{dimC})\ h\ B$
 <proof>

lemma *sum-mat-cong'*:
 shows $\text{finite } I \implies (\bigwedge i. i \in I \implies A\ i = B\ i) \implies$
 $(\bigwedge i. i \in I \implies A\ i \in \text{fc-mats}) \implies$
 $(\bigwedge i. i \in I \implies B\ i \in \text{fc-mats}) \implies I = J \implies \text{sum-mat } A\ I = \text{sum-mat } B\ J$
 <proof>

lemma *sum-mat-reindex-cong*:
 assumes *finite B*
 and $\bigwedge x. x \in l\ 'B \implies g\ x \in \text{fc-mats}$
 and $\bigwedge x. x \in B \implies h\ x \in \text{fc-mats}$
 and *inj-on l B*
 and $\bigwedge x. x \in B \implies g\ (l\ x) = h\ x$
 shows $\text{sum-mat } g\ (l\ 'B) = \text{sum-mat } h\ B$
 <proof>

lemma *sum-mat-mod-eq*:
 fixes $A :: \text{nat} \Rightarrow 'a\ \text{Matrix.mat}$
 assumes $\bigwedge x. x \in \{..<m\} \implies A\ x \in \text{fc-mats}$
shows $\text{sum-mat } (\lambda i. A\ (i\ \text{mod } m))\ ((\lambda i. n * m + i)\ \{..<m\}) = \text{sum-mat } A\ \{..<m\}$
 <proof>

lemma *sum-mat-singleton'*:
 assumes $A\ i \in \text{fc-mats}$
 shows $\text{sum-mat } A\ \{i\} = A\ i$
 <proof>

end

context *cpx-sq-mat*
begin

lemma *sum-mat-mod-div-ne-0*:

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and $0 < n$
and $0 < m$
and $\text{dim}R = n * m$
and $nD \neq 0$

shows $\text{sum-mat } (\lambda i. \text{sum-mat } (\lambda j. f\ i * g\ j \cdot_m ((A\ i) \otimes (B\ j))) \{.. < nD\})$
 $\{.. < nC\} =$
 $\text{sum-mat } (\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD)))) \{.. < nC * nD\}$
 $\langle \text{proof} \rangle$

lemma *sum-mat-mod-div-eq-0*:

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $0 < n$
and $nD = 0$
and $\text{dim}R = n * m$

shows $\text{sum-mat } (\lambda i. \text{sum-mat } (\lambda j. f\ i * g\ j \cdot_m ((A\ i) \otimes (B\ j))) \{.. < nD\})$
 $\{.. < nC\} =$
 $\text{sum-mat } (\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD)))) \{.. < nC * nD\}$
 $\langle \text{proof} \rangle$

lemma *sum-mat-mod-div*:

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and $0 < n$
and $0 < m$
and $\text{dim}R = n * m$

shows $\text{sum-mat } (\lambda i. \text{sum-mat } (\lambda j. f\ i * g\ j \cdot_m ((A\ i) \otimes (B\ j))) \{.. < nD\})$
 $\{.. < nC\} =$
 $\text{sum-mat } (\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD)))) \{.. < nC * nD\}$
 $\langle \text{proof} \rangle$

lemma *sum-sum-mat-expand-ne-0*:

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and $R \in \text{carrier-mat } (n * m)\ (n * m)$
and $0 < n$
and $0 < m$
and $nD \neq 0$
and $\text{dim}R = n * m$

shows $\text{sum-mat } (\lambda i. \text{sum-mat } (\lambda j. f\ i * g\ j \cdot_m ((A\ i) \otimes (B\ j)) * R) \{.. < nD\})$
 $\{.. < nC\} =$
 $\text{sum-mat } (\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD))) * R) \{.. < nC * nD\}$

<proof>

lemma *sum-sum-mat-expand-eq-0:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $R \in \text{carrier-mat } (n*m)\ (n*m)$
and $0 < n$
and $0 < m$
and $nD = 0$
and $\text{dim}R = n * m$

shows $\text{sum-mat } (\lambda i. \text{sum-mat } (\lambda j. f\ i * g\ j \cdot_m ((A\ i) \otimes (B\ j)) * R) \{.. < nD\})$
 $\{.. < nC\} =$
 $\text{sum-mat } (\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD))) * R) \{.. < nC*nD\}$
<proof>

lemma *sum-sum-mat-expand:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and $R \in \text{carrier-mat } (n*m)\ (n*m)$
and $0 < n$
and $0 < m$
and $\text{dim}R = n * m$

shows $\text{sum-mat } (\lambda i. \text{sum-mat } (\lambda j. f\ i * g\ j \cdot_m ((A\ i) \otimes (B\ j)) * R) \{.. < nD\})$
 $\{.. < nC\} =$
 $\text{sum-mat } (\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD))) * R) \{.. < nC*nD\}$
<proof>

end

3 Results on tensor products

lemma *tensor-mat-trace:*

assumes $A \in \text{carrier-mat } n\ n$
and $B \in \text{carrier-mat } m\ m$
and $0 < n$
and $0 < m$

shows $\text{Complex-Matrix.trace } (A \otimes B) = \text{Complex-Matrix.trace } A * \text{Complex-Matrix.trace } B$
<proof>

lemma *tensor-vec-inner-prod:*

assumes $u \in \text{carrier-vec } n$
and $v \in \text{carrier-vec } n$
and $a \in \text{carrier-vec } n$
and $b \in \text{carrier-vec } n$
and $0 < n$

shows $\text{Complex-Matrix.inner-prod } (\text{tensor-vec } u\ v) (\text{tensor-vec } a\ b) = \text{Complex-Matrix.inner-prod } u\ a * \text{Complex-Matrix.inner-prod } v\ b$

<proof>

lemma *tensor-mat-positive:*

assumes $A \in \text{carrier-mat } n \ n$

and $B \in \text{carrier-mat } m \ m$

and $0 < n$

and $0 < m$

and *Complex-Matrix.positive* A

and *Complex-Matrix.positive* B

shows *Complex-Matrix.positive* $(A \otimes B)$

<proof>

lemma *tensor-mat-square-idty:*

assumes $A * A = 1_m \ n$

and $B * B = 1_m \ m$

and $0 < n$

and $0 < m$

shows $(A \otimes B) * (A \otimes B) = 1_m \ (n*m)$

<proof>

lemma *tensor-mat-commute:*

assumes $A \in \text{carrier-mat } n \ n$

and $B \in \text{carrier-mat } m \ m$

and $C \in \text{carrier-mat } n \ n$

and $D \in \text{carrier-mat } m \ m$

and $0 < n$

and $0 < m$

and $A * C = C * A$

and $B * D = D * B$

shows $(A \otimes B) * (C \otimes D) = (C \otimes D) * (A \otimes B)$

<proof>

lemma *tensor-mat-mult-id:*

assumes $A \in \text{carrier-mat } n \ n$

and $B \in \text{carrier-mat } m \ m$

and $0 < n$

and $0 < m$

shows $(A \otimes 1_m \ m) * (1_m \ n \otimes B) = A \otimes B$

<proof>

lemma *tensor-mat-trace-mult-distr:*

assumes $A \in \text{carrier-mat } n \ n$

and $B \in \text{carrier-mat } m \ m$

and $C \in \text{carrier-mat } n \ n$

and $D \in \text{carrier-mat } m \ m$

and $0 < n$

and $0 < m$

shows *Complex-Matrix.trace* $((A \otimes B) * (C \otimes D)) =$

$\text{Complex-Matrix.trace } (A * C) * (\text{Complex-Matrix.trace } (B * D))$
 ⟨proof⟩

lemma *tensor-mat-diagonal*:
 assumes $A \in \text{carrier-mat } n \ n$
 and $B \in \text{carrier-mat } m \ m$
 and *diagonal-mat* A
 and *diagonal-mat* B
 shows *diagonal-mat* $(A \otimes B)$ ⟨proof⟩

lemma *tensor-mat-add-right*:
 assumes $A \in \text{carrier-mat } n \ m$
 and $B \in \text{carrier-mat } i \ j$
 and $C \in \text{carrier-mat } i \ j$
 and $0 < m$
 and $0 < j$
 shows $A \otimes (B + C) = (A \otimes B) + (A \otimes C)$
 ⟨proof⟩

lemma *tensor-mat-zero*:
 assumes $B \in \text{carrier-mat } i \ j$
 and $0 < j$
 and $0 < m$
 shows $0_m \ n \ m \otimes B = 0_m \ (n * i) \ (m * j)$
 ⟨proof⟩

lemma *tensor-mat-zero'*:
 assumes $B \in \text{carrier-mat } i \ j$
 and $0 < j$
 and $0 < m$
 shows $B \otimes 0_m \ n \ m = 0_m \ (i * n) \ (j * m)$
 ⟨proof⟩

lemma *tensor-mat-sum-right*:
 fixes $A :: \text{complex Matrix.mat}$
 assumes *finite* I
 and $A \in \text{carrier-mat } n \ m$
 and $\bigwedge k. k \in I \implies ((B \ k) :: \text{complex Matrix.mat}) \in \text{carrier-mat } i \ j$
 and $0 < m$
 and $0 < j$
 and $\text{dim}R = n * i$
 and $\text{dim}C = m * j$
 shows $A \otimes (\text{fixed-carrier-mat.sum-mat } i \ j \ B \ I) =$
 $\text{fixed-carrier-mat.sum-mat } (n * i) \ (m * j) \ (\lambda i. A \otimes (B \ i)) \ I$
 ⟨proof⟩

lemma *tensor-mat-add-left*:
 assumes $A \in \text{carrier-mat } n \ m$

and $B \in \text{carrier-mat } n \ m$
and $C \in \text{carrier-mat } i \ j$
and $0 < m$
and $0 < j$
shows $(A + B) \otimes C = (A \otimes C) + (B \otimes C)$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-smult-left*:
assumes $A \in \text{carrier-mat } n \ m$
and $B \in \text{carrier-mat } i \ j$
and $0 < m$
and $0 < j$
shows $x \cdot_m A \otimes B = x \cdot_m (A \otimes B)$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-smult-right*:
assumes $A \in \text{carrier-mat } n \ m$
and $B \in \text{carrier-mat } i \ j$
and $0 < m$
and $0 < j$
shows $A \otimes (x \cdot_m B) = x \cdot_m (A \otimes B)$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-smult*:
assumes $A \in \text{carrier-mat } n \ m$
and $B \in \text{carrier-mat } i \ j$
and $0 < m$
and $0 < j$
shows $x \cdot_m A \otimes (y \cdot_m B) = x * y \cdot_m (A \otimes B)$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-singleton-right*:
assumes $0 < \text{dim-col } A$
and $B \in \text{carrier-mat } 1 \ 1$
shows $A \otimes B = B \$(0,0) \cdot_m A$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-singleton-left*:
assumes $0 < \text{dim-col } A$
and $B \in \text{carrier-mat } 1 \ 1$
shows $B \otimes A = B \$(0,0) \cdot_m A$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-sum-left*:
assumes *finite* I
and $B \in \text{carrier-mat } i \ j$
and $\bigwedge k. k \in I \implies A \ k \in \text{carrier-mat } n \ m$
and $0 < m$
and $0 < j$

and $\dim R = n * i$
and $\dim C = m * j$
shows $(\text{fixed-carrier-mat.sum-mat } n \ m \ A \ I) \otimes B =$
 $\text{fixed-carrier-mat.sum-mat } (n * i) \ (m * j) \ (\lambda i. (A \ i) \otimes B) \ I$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-diag-elem*:

assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } m \ m$
and $i < n * m$
and $0 < n * m$
shows $(A \otimes B) \ \$\$ (i, i) = A \ \$\$ (i \ \text{div } m, i \ \text{div } m) *$
 $B \ \$\$ (i \ \text{mod } m, i \ \text{mod } m)$
 $\langle \text{proof} \rangle$

context *cpx-sq-mat*

begin

lemma *tensor-mat-sum-mat-right*:

assumes *finite* I
and $A \in \text{carrier-mat } n \ n$
and $\bigwedge k. k \in I \implies B \ k \in \text{carrier-mat } i \ i$
and $0 < n$
and $0 < i$
and $\dim R = n * i$
shows $A \otimes (\text{fixed-carrier-mat.sum-mat } i \ i \ B \ I) = \text{sum-mat } (\lambda i. A \otimes (B \ i)) \ I$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-sum-mat-left*:

assumes *finite* I
and $B \in \text{carrier-mat } i \ i$
and $\bigwedge k. k \in I \implies A \ k \in \text{carrier-mat } n \ n$
and $0 < n$
and $0 < i$
and $\dim R = n * i$
shows $(\text{fixed-carrier-mat.sum-mat } n \ n \ A \ I) \otimes B = \text{sum-mat } (\lambda i. (A \ i) \otimes B) \ I$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-sum-nat-mod-div-ne-0*:

assumes $\bigwedge k. k < (nC::\text{nat}) \implies A \ k \in \text{carrier-mat } n \ n$
and $\bigwedge j. j < (nD::\text{nat}) \implies B \ j \in \text{carrier-mat } m \ m$
and $\text{fixed-carrier-mat.sum-mat } n \ n \ (\lambda i. f \ i \cdot_m (A \ i)) \ \{.. < nC\} = C$
and $\text{fixed-carrier-mat.sum-mat } m \ m \ (\lambda j. g \ j \cdot_m (B \ j)) \ \{.. < nD\} = D$
and $0 < n$
and $0 < m$
and $nD \neq 0$
and $\dim R = n * m$
shows $\text{sum-mat } (\lambda i. (f \ (i \ \text{div } nD) * g \ (i \ \text{mod } nD)) \cdot_m$
 $((A \ (i \ \text{div } nD)) \otimes (B \ (i \ \text{mod } nD))))$

$$\{..< nC*nD\} = C \otimes D \langle proof \rangle$$

lemma *tensor-mat-sum-nat-mod-div-eq-0:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and *fixed-carrier-mat.sum-mat* $n\ n\ (\lambda i. f\ i \cdot_m (A\ i)) \{..< nC\} = C$
and *fixed-carrier-mat.sum-mat* $m\ m\ (\lambda j. g\ j \cdot_m (B\ j)) \{..< nD\} = D$
and $0 < n$
and $0 < m$
and $nD = 0$
and $\text{dim}R = n * m$

shows *sum-mat* $(\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD))))$
 $\{..< nC*nD\} = C \otimes D$
 $\langle proof \rangle$

lemma *tensor-mat-sum-nat-mod-div:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and *fixed-carrier-mat.sum-mat* $n\ n\ (\lambda i. f\ i \cdot_m (A\ i)) \{..< nC\} = C$
and *fixed-carrier-mat.sum-mat* $m\ m\ (\lambda j. g\ j \cdot_m (B\ j)) \{..< nD\} = D$
and $0 < n$
and $0 < m$
and $\text{dim}R = n * m$

shows *sum-mat* $(\lambda i. (f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD))))$
 $\{..< nC*nD\} = C \otimes D$
 $\langle proof \rangle$

end

lemma *tensor-mat-sum-mult-trace-expand-ne-0:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and $R \in \text{carrier-mat } (n*m)\ (n*m)$
and *fixed-carrier-mat.sum-mat* $n\ n\ (\lambda i. f\ i \cdot_m (A\ i)) \{..< nC\} = C$
and *fixed-carrier-mat.sum-mat* $m\ m\ (\lambda j. g\ j \cdot_m (B\ j)) \{..< nD\} = D$
and $0 < n$
and $0 < m$
and $nD \neq 0$

shows *sum* $(\lambda i. \text{Complex-Matrix.trace } ((f\ (i\ \text{div}\ nD) * g\ (i\ \text{mod}\ nD)) \cdot_m$
 $((A\ (i\ \text{div}\ nD)) \otimes (B\ (i\ \text{mod}\ nD))) * R)) \{..< nC * nD\} =$
 $\text{Complex-Matrix.trace } ((C \otimes D) * R)$
 $\langle proof \rangle$

lemma *tensor-mat-sum-mult-trace-expand-eq-0:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $R \in \text{carrier-mat } (n*m)\ (n*m)$
and *fixed-carrier-mat.sum-mat* $n\ n\ (\lambda i. f\ i \cdot_m (A\ i)) \{..< nC\} = C$
and *fixed-carrier-mat.sum-mat* $m\ m\ (\lambda j. g\ j \cdot_m (B\ j)) \{..< nD\} = D$

and $0 < n$
and $0 < m$
and $nD = 0$
shows $\text{sum } (\lambda i. \text{Complex-Matrix.trace } ((f (i \text{ div } nD) * g (i \text{ mod } nD))) \cdot_m$
 $((A (i \text{ div } nD)) \otimes (B (i \text{ mod } nD))) * R) \{.. < nC * nD\} =$
 $\text{Complex-Matrix.trace } ((C \otimes D) * R)$
 <proof>

lemma *tensor-mat-sum-mult-trace-expand:*

assumes $\bigwedge k. k < (nC::nat) \implies A k \in \text{carrier-mat } n n$
and $\bigwedge j. j < (nD::nat) \implies B j \in \text{carrier-mat } m m$
and $R \in \text{carrier-mat } (n*m) (n*m)$
and $\text{fixed-carrier-mat.sum-mat } n n (\lambda i. f i \cdot_m (A i)) \{.. < nC\} = C$
and $\text{fixed-carrier-mat.sum-mat } m m (\lambda j. g j \cdot_m (B j)) \{.. < nD\} = D$
and $0 < n$
and $0 < m$
shows $\text{sum } (\lambda i. \text{Complex-Matrix.trace } ((f (i \text{ div } nD) * g (i \text{ mod } nD))) \cdot_m$
 $((A (i \text{ div } nD)) \otimes (B (i \text{ mod } nD))) * R) \{.. < nC * nD\} =$
 $\text{Complex-Matrix.trace } ((C \otimes D) * R)$
 <proof>

lemma *tensor-mat-sum-mult-trace-ne-0:*

assumes $\bigwedge k. k < (nC::nat) \implies A k \in \text{carrier-mat } n n$
and $\bigwedge j. j < (nD::nat) \implies B j \in \text{carrier-mat } m m$
and $R \in \text{carrier-mat } (n*m) (n*m)$
and $\text{fixed-carrier-mat.sum-mat } n n (\lambda i. f i \cdot_m (A i)) \{.. < nC\} = C$
and $\text{fixed-carrier-mat.sum-mat } m m (\lambda j. g j \cdot_m (B j)) \{.. < nD\} = D$
and $0 < n$
and $0 < m$
and $0 \neq nD$
shows $\text{sum } (\lambda i. (\text{sum } (\lambda j. \text{Complex-Matrix.trace } ((f i * g j) \cdot_m$
 $((A i) \otimes (B j)) * R) \{.. < nD\})) \{.. < nC\} =$
 $\text{Complex-Matrix.trace } ((C \otimes D) * R)$
 <proof>

lemma *tensor-mat-sum-mult-trace-eq-0:*

assumes $\bigwedge k. k < (nC::nat) \implies A k \in \text{carrier-mat } n n$
and $R \in \text{carrier-mat } (n*m) (n*m)$
and $\text{fixed-carrier-mat.sum-mat } n n (\lambda i. f i \cdot_m (A i)) \{.. < nC\} = C$
and $\text{fixed-carrier-mat.sum-mat } m m (\lambda j. g j \cdot_m (B j)) \{.. < nD\} = D$
and $0 < n$
and $0 < m$
and $0 = (nD::nat)$
shows $\text{sum } (\lambda i. (\text{sum } (\lambda j. \text{Complex-Matrix.trace } ((f i * g j) \cdot_m$
 $((A i) \otimes (B j)) * R) \{.. < nD\})) \{.. < nC\} =$
 $\text{Complex-Matrix.trace } ((C \otimes D) * R)$
 <proof>

lemma *tensor-mat-sum-mult-trace:*

assumes $\bigwedge k. k < (nC::nat) \implies A\ k \in \text{carrier-mat } n\ n$
and $\bigwedge j. j < (nD::nat) \implies B\ j \in \text{carrier-mat } m\ m$
and $R \in \text{carrier-mat } (n*m)\ (n*m)$
and $\text{fixed-carrier-mat.sum-mat } n\ n\ (\lambda i. f\ i\ \cdot_m\ (A\ i))\ \{..< nC\} = C$
and $\text{fixed-carrier-mat.sum-mat } m\ m\ (\lambda j. g\ j\ \cdot_m\ (B\ j))\ \{..< nD\} = D$
and $0 < n$
and $0 < m$
shows $\text{sum } (\lambda i. (\text{sum } (\lambda j. \text{Complex-Matrix.trace } ((f\ i * g\ j)\cdot_m$
 $((A\ i) \otimes (B\ j)) * R))\ \{..< nD\}))\ \{..< nC\} =$
 $\text{Complex-Matrix.trace } ((C \otimes D) * R)$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-make-pm-mult-trace:*

assumes $A \in \text{carrier-mat } n\ n$
and *hermitian* A
and $B \in \text{carrier-mat } m\ m$
and *hermitian* B
and $R \in \text{carrier-mat } (n*m)\ (n*m)$
and $(nA, M) = \text{cpx-sq-mat.make-pm } n\ n\ A$
and $(nB, N) = \text{cpx-sq-mat.make-pm } m\ m\ B$
and $0 < n$
and $0 < m$
shows $\text{sum } (\lambda i. (\text{sum } (\lambda j. \text{Complex-Matrix.trace}$
 $((\text{complex-of-real } (\text{meas-outcome-val } (M\ i)) *$
 $\text{complex-of-real } (\text{meas-outcome-val } (N\ j))))\cdot_m$
 $((\text{meas-outcome-prj } (M\ i)) \otimes (\text{meas-outcome-prj } (N\ j))) * R))\ \{..< nB\}))$
 $\{..< nA\} =$
 $\text{Complex-Matrix.trace } ((A \otimes B) * R)$
 $\langle \text{proof} \rangle$

lemma *tensor-mat-mat-conj:*

assumes $A \in \text{carrier-mat } n\ n$
and $B \in \text{carrier-mat } n\ n$
and $U \in \text{carrier-mat } n\ n$
and $C \in \text{carrier-mat } m\ m$
and $D \in \text{carrier-mat } m\ m$
and $V \in \text{carrier-mat } m\ m$
and $0 < n$
and $0 < m$
and $A = \text{mat-conj } U\ B$
and $C = \text{mat-conj } V\ D$
shows $A \otimes C = \text{mat-conj } (U \otimes V)\ (B \otimes D)$
 $\langle \text{proof} \rangle$

lemma *unitarily-equiv-mat-conj[simp]:*

assumes *unitarily-equiv* $A\ B\ U$
shows $A = \text{mat-conj } U\ B$ $\langle \text{proof} \rangle$

lemma *hermitian-tensor-mat-decomp:*

```

assumes  $A \in \text{carrier-mat } n \ n$ 
and  $C \in \text{carrier-mat } m \ m$ 
and unitary-diag  $A \ B \ U$ 
and unitary-diag  $C \ D \ V$ 
and  $0 < n$ 
and  $0 < m$ 
shows unitary-diag  $(A \otimes C) \ (B \otimes D) \ (U \otimes V)$ 
 $\langle \text{proof} \rangle$ 

end

```

```

theory Matrix-L2-Operator-Norm
imports
  Tensor-Mat-Compl-Properties
begin

```

We formalize the \mathcal{L}_2 operator norm on matrices on nonempty vector spaces. This norm can be defined on a matrix A by $\|A\|_2 = \sup\{\|A \cdot v\|_2 \mid \|v\|_2 = 1\}$, and it is equal to the maximum singular value of A .

4 Preliminary results

4.1 Commutator and anticommutator

We define the notions of commutator and anticommutator of two matrices. When these matrices commute, their commutator is the zero matrix.

```

definition commutator :: complex Matrix.mat  $\Rightarrow$  complex Matrix.mat  $\Rightarrow$ 
  complex Matrix.mat where
  commutator  $A \ B = A * B - B * A$ 

```

```

definition anticommutator where
  anticommutator  $A \ B = A * B + B * A$ 

```

```

lemma commutator-dim:
  assumes  $A \in \text{carrier-mat } n \ n$ 
  and  $B \in \text{carrier-mat } n \ n$ 
shows commutator  $A \ B \in \text{carrier-mat } n \ n$   $\langle \text{proof} \rangle$ 

```

```

lemma anticommutator-dim:
  assumes  $A \in \text{carrier-mat } n \ n$ 
  and  $B \in \text{carrier-mat } n \ n$ 
shows anticommutator  $A \ B \in \text{carrier-mat } n \ n$   $\langle \text{proof} \rangle$ 

```

```

lemma commutator-zero-iff:
  assumes  $A \in \text{carrier-mat } n \ n$ 
  and  $B \in \text{carrier-mat } n \ n$ 
shows commutator  $A \ B = 0_m \ n \ n \iff A * B = B * A$ 

```

<proof>

lemma *anticommutator-zero-iff:*

fixes $A::'a ::ring Matrix.mat$

assumes $A \in carrier\text{-}mat\ n\ n$

and $B \in carrier\text{-}mat\ n\ n$

shows $anticommutator\ A\ B = 0_m\ n\ n \longleftrightarrow B * A = -(A * B)$

<proof>

lemma *commutator-mult-expand:*

assumes $A \in carrier\text{-}mat\ n\ n$

and $B \in carrier\text{-}mat\ n\ n$

and $C \in carrier\text{-}mat\ n\ n$

and $D \in carrier\text{-}mat\ n\ n$

shows $commutator\ A\ B * commutator\ C\ D =$

$A * B * (C * D) - A * B * (D * C) - B * A * (C * D) + B * A * (D * C)$

<proof>

5 Maximum modulus in a spectrum

We prove some basic results on the maximum modulus of elements in a matrix A , and focus on the case where A is a Hermitian matrix.

5.1 Definition and basic properties for Hermitian matrices

definition *spmax:: complex Matrix.mat \Rightarrow real where*

spmax $A = Max.F\ \{cmod\ a\ | a. a \in spectrum\ A\}$

lemma *spmax-mem:*

assumes $A \in carrier\text{-}mat\ n\ n$

and $0 < n$

shows $spmax\ A \in \{cmod\ a\ | a. a \in spectrum\ A\}$

<proof>

lemma *spmax-geq-0:*

assumes $A \in carrier\text{-}mat\ n\ n$

and $0 < n$

shows $0 \leq spmax\ A$

<proof>

lemma *Re-inner-mult-diag-le:*

fixes $B::complex\ Matrix.mat$

assumes *diagonal-mat* B

and $B \in carrier\text{-}mat\ n\ n$

and $0 < n$

and $M = Max.F\ \{Re\ (conjugate\ a)\ | a. a \in diag\text{-}elems\ B\}$

shows $\forall v \in carrier\text{-}vec\ n. Re\ (inner\text{-}prod\ (B *_{\nu}\ v)\ v) \leq$

$M * Re\ ((inner\text{-}prod\ v\ v))$

<proof>

lemma *Re-inner-mult-diag-le'*:

fixes $B::\text{complex Matrix.mat}$

assumes *diagonal-mat* B

and $B \in \text{carrier-mat } n \ n$

and $0 < n$

and $(M::\text{real}) = \text{Max.F } \{cmod \ a \mid a. a \in \text{diag-elems } B\}$

and $v \in \text{carrier-vec } n$

shows $cmod (\text{inner-prod } v (B *_v v)) \leq M * \text{inner-prod } v \ v$

<proof>

lemma *hermitian-mult-inner-prod-le*:

fixes $A::\text{complex Matrix.mat}$

assumes $A \in \text{carrier-mat } n \ n$

and $0 < n$

and *hermitian* A

and $v \in \text{carrier-vec } n$

shows $cmod (\text{inner-prod } v (A *_v v)) \leq (\text{spmax } A) * (\text{inner-prod } v \ v)$

<proof>

lemma *hermitian-trace-rank-le*:

assumes $A \in \text{carrier-mat } n \ n$

and *hermitian* A

and $v \in \text{carrier-vec } n$

and $0 < n$

shows $cmod (\text{Complex-Matrix.trace } (A * (\text{rank-1-proj } v))) \leq$

$(\text{spmax } A) * (\text{inner-prod } v \ v)$

<proof>

lemma *hermitian-pos-decomp-cmod-le*:

assumes $A \in \text{carrier-mat } n \ n$

and $C \in \text{carrier-mat } n \ n$

and $0 < n$

and *hermitian* C

and *Complex-Matrix.positive* A

shows $cmod (\text{Complex-Matrix.trace } (C * A)) \leq$

$\text{Re } (\text{Complex-Matrix.trace } A) * (\text{spmax } C)$

<proof>

lemma *hermitian-density-cmod-le*:

fixes $R::\text{complex Matrix.mat}$

assumes $R \in \text{carrier-mat } n \ n$

and $A \in \text{carrier-mat } n \ n$

and $0 < n$

and *hermitian* A

and *density-operator* R

shows $cmod (\text{Complex-Matrix.trace } (A * R)) \leq (\text{spmax } A)$

<proof>

lemma *tensor-mat-hermitian-positive-le:*
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } m \ m$
and $C \in \text{carrier-mat } n \ n$
and $D \in \text{carrier-mat } m \ m$
and $0 < n$
and $0 < m$
and *hermitian* A
and *hermitian* B
and *Complex-Matrix.positive* C
and *Complex-Matrix.positive* D
shows $\text{cmod } (\text{Complex-Matrix.trace } ((A \otimes B) * (C \otimes D))) \leq$
 $\text{Re } (\text{Complex-Matrix.trace } C) * \text{Re } (\text{Complex-Matrix.trace } D) *$
 $\text{spmax } A * \text{spmax } B$
<proof>

lemma *tensor-mat-hermitian-density-le:*
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } m \ m$
and $C \in \text{carrier-mat } n \ n$
and $D \in \text{carrier-mat } m \ m$
and $0 < n$
and $0 < m$
and *hermitian* A
and *hermitian* B
and *density-operator* C
and *density-operator* D
shows $\text{cmod } (\text{Complex-Matrix.trace } ((A \otimes B) * (C \otimes D))) \leq$
 $\text{spmax } A * \text{spmax } B$
<proof>

lemma *idty-spmax:*
assumes $0 < n$
shows $\text{spmax } (1_m \ n) = 1$ *<proof>*

lemma *spmax-uminus:*
fixes $A :: \text{complex Matrix.mat}$
assumes *hermitian* A
and $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{spmax } (-A) = \text{spmax } A$
<proof>

lemma *spmax-smult:*
fixes $A :: \text{complex Matrix.mat}$

assumes *hermitian* A
and $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{spmax } (x \cdot_m A) = \text{cmod } x * \text{spmax } A$
<proof>

lemma *spmax-smult-pos*:
fixes $A :: \text{complex Matrix.mat}$
assumes *hermitian* A
and $A \in \text{carrier-mat } n \ n$
and $0 < n$
and $0 \leq x$
shows $\text{spmax } (x \cdot_m A) = x * \text{spmax } A$
<proof>

lemma *hermitian-square-spmax*:
fixes $A :: \text{complex Matrix.mat}$
assumes *hermitian* A
and $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{spmax } (A*A) = \text{spmax } A * \text{spmax } A$
<proof>

lemma *hermitian-square-idty-spmax*:
assumes $0 < n$
and $A \in \text{carrier-mat } n \ n$
and *hermitian* A
and $A*A = 1_m \ n$
shows $\text{spmax } A = 1$
<proof>

lemma *hermitian-mult-density-trace*:
assumes $A \in \text{carrier-mat } n \ n$
and $R \in \text{carrier-mat } n \ n$
and $0 < n$
and *hermitian* A
and $A * A = 1_m \ n$
and *density-operator* R
shows $|\text{Complex-Matrix.trace } (A*R)| \leq 1$
<proof>

lemma *tensor-mat-hermitian-density-spmax-le*:
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } m \ m$
and $C \in \text{carrier-mat } n \ n$
and $D \in \text{carrier-mat } m \ m$
and $0 < n$
and $0 < m$
and *hermitian* A

and *hermitian* B
and $A * A = 1_m \ n$
and $B * B = 1_m \ m$
and *density-operator* C
and *density-operator* D
shows $cmod \ (Complex-Matrix.trace \ ((A \otimes B) * (C \otimes D))) \leq 1$
 $\langle proof \rangle$

5.2 Eigenvector for the element with maximum modulus

definition *spmax-wit* **where**

$spmax-wit \ A = (SOME \ k. \ eigenvalue \ A \ k \wedge \ spmax \ A = cmod \ k)$

lemma *spmax-wit-eigenvalue*:

assumes $A \in carrier-mat \ n \ n$

and $0 < n$

shows $eigenvalue \ A \ (spmax-wit \ A) \wedge \ spmax \ A = cmod \ (spmax-wit \ A)$

$\langle proof \rangle$

lemma *find-eigen-spmax-neq-0*:

assumes $A \in carrier-mat \ n \ n$

and $0 < n$

shows $find-eigenvector \ A \ (spmax-wit \ A) \neq 0_v \ n \ \langle proof \rangle$

lemma *find-eigen-spmax-dim*:

assumes $A \in carrier-mat \ n \ n$

and $0 < n$

shows $dim-vec \ (vec-normalize \ (find-eigenvector \ A \ (spmax-wit \ A))) = n$

$\langle proof \rangle$

lemma *nrm-spmax-eigenvector-eq*:

assumes $v = vec-normalize \ (find-eigenvector \ A \ (spmax-wit \ A))$

and $A \in carrier-mat \ n \ n$

and $0 < n$

shows $cmod \ (inner-prod \ v \ (A * _v \ v)) = spmax \ A$

$\langle proof \rangle$

6 The \mathcal{L}_2 operator norm

6.1 Definition and preliminary results

definition *rvec-norm* **where**

$rvec-norm \ v = Re \ (vec-norm \ v)$

definition *L2-op-nrm* **where**

$L2-op-nrm \ A =$

$Sup \ \{rvec-norm \ (A * _v \ v) \ | \ v. \ dim-vec \ v = dim-col \ A \wedge \ rvec-norm \ v = 1\}$

lemma *mat-mult-inner-prod-le*:
fixes $A::\text{complex Matrix.mat}$
assumes $0 < \text{dim-col } A$
and $v \in \text{carrier-vec } (\text{dim-col } A)$
shows $\text{cmod } (\text{inner-prod } (A *_v v) (A *_v v)) \leq$
 $\text{spmax } ((\text{Complex-Matrix.adjoint } A) * A) * (\text{inner-prod } v v)$
 $\langle \text{proof} \rangle$

lemma *normalized-rvec-norm*:
assumes $v \neq 0_v (\text{dim-vec } v)$
shows $\text{rvec-norm } (\text{vec-normalize } v) = 1$
 $\langle \text{proof} \rangle$

lemma *vec-norm-smult*:
shows $\text{vec-norm } (c \cdot_v v) = (\text{cmod } c) * (\text{vec-norm } v)$
 $\langle \text{proof} \rangle$

lemma *rvec-norm-smult*:
shows $\text{rvec-norm } (c \cdot_v v) = (\text{cmod } c) * (\text{rvec-norm } v)$
 $\langle \text{proof} \rangle$

lemma *mult-mat-zero-vec*:
assumes $A \in \text{carrier-mat } n m$
and $v = 0_v m$
shows $A *_v v = 0_v n$
 $\langle \text{proof} \rangle$

lemma *mat-mult-vec-normalize*:
assumes $\text{dim-col } A = \text{dim-vec } v$
shows $A *_v v = \text{vec-norm } v \cdot_v (A *_v (\text{vec-normalize } v))$
 $\langle \text{proof} \rangle$

lemma *vec-norm-real*:
shows $\text{vec-norm } v \in \text{Reals}$
 $\langle \text{proof} \rangle$

lemma *rvec-norm-geq-0*:
shows $0 \leq \text{rvec-norm } v \langle \text{proof} \rangle$

lemma *rvec-norm-triangle*:
assumes $\text{dim-vec } u = \text{dim-vec } v$
shows $\text{rvec-norm } (u + v) \leq \text{rvec-norm } u + \text{rvec-norm } v$
 $\langle \text{proof} \rangle$

lemma *cmod-vec-norm*:
shows $\text{cmod } (\text{vec-norm } v) = \text{vec-norm } v$
 $\langle \text{proof} \rangle$

lemma *cmod-rvec-norm*:

shows $cmod (rvec\text{-}norm\ v) = rvec\text{-}norm\ v$
 ⟨proof⟩

lemma *inner-prod-rvec-norm-pow2*:

shows $(rvec\text{-}norm\ v)^2 = v \cdot c\ v$
 ⟨proof⟩

lemma *rvec-norm-mat-mult-le*:

assumes $v \in carrier\text{-}vec\ (dim\text{-}col\ A)$
and $0 < dim\text{-}col\ A$
shows $cmod\ (inner\text{-}prod\ (A *_{v}\ v)\ (A *_{v}\ v))$
 $\leq\ spmax\ (Complex\text{-}Matrix.adjoint\ A * A) * (rvec\text{-}norm\ v)^2$
 ⟨proof⟩

lemma *square-leg*:

assumes $a^2 \leq b * c^2$
and $0 \leq c$
shows $a \leq (sqrt\ b) * c$
 ⟨proof⟩

lemma *rvec-set-ne*:

assumes $0 < dim\text{-}col\ A$
shows $\{rvec\text{-}norm\ (A *_{v}\ v) \mid v.\ dim\text{-}vec\ v = dim\text{-}col\ A \wedge rvec\text{-}norm\ v = 1\} \neq \{\}$
 ⟨proof⟩

lemma *unitary-col-vec-norm*:

assumes $U \in carrier\text{-}mat\ n\ n$
and *unitary* U
and $i < n$
shows $vec\text{-}norm\ (Matrix.col\ U\ i) = 1$ ⟨proof⟩

lemma *unitary-col-rvec-norm*:

assumes $U \in carrier\text{-}mat\ n\ n$
and *unitary* U
and $i < n$
shows $rvec\text{-}norm\ (Matrix.col\ U\ i) = 1$ ⟨proof⟩

lemma *Cauchy-Schwarz-complex-rvec-norm*:

assumes $dim\text{-}vec\ x = dim\text{-}vec\ y$
shows $cmod\ (inner\text{-}prod\ x\ y) \leq rvec\text{-}norm\ x * rvec\text{-}norm\ y$
 ⟨proof⟩

6.2 The \mathcal{L}_2 operator norm is equal to the maximum singular value

definition *max-sgval* **where**

$max\text{-}sgval\ A = sqrt\ (spmax\ (Complex\text{-}Matrix.adjoint\ A * A))$

lemma *max-sgval-geq-0*:

assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $0 \leq \text{max-sgval } A$
 ⟨proof⟩

lemma *max-sgval-uminus*:
shows $\text{max-sgval } (-A) = \text{max-sgval } A$
 ⟨proof⟩

lemma *rvec-leq-sg-spmx*:
assumes $0 < \text{dim-col } A$
and $v \in \text{carrier-vec } (\text{dim-col } A)$
shows $\text{rvec-norm } (A *_v v) \leq (\text{max-sgval } A) * \text{rvec-norm } v$
 ⟨proof⟩

lemma *max-sgval-smult*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{max-sgval } (a_m A) = \text{cmod } a * \text{max-sgval } A$
 ⟨proof⟩

lemma *L2-op-nrm-le-max-sgval*:
assumes $0 < \text{dim-col } A$
shows $L2\text{-op-nrm } A \leq \text{max-sgval } A$ ⟨proof⟩

lemma *max-sgval-eigen*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
and $C = \text{Complex-Matrix.adjoint } A * A$
and $v = \text{vec-normalize } (\text{find-eigenvector } C \ (\text{spmax-wit } C))$
shows $\text{rvec-norm } (A *_v v) = \text{max-sgval } A$
 ⟨proof⟩

lemma *rvec-normalize-leq-L2-op-nrm*:
assumes $\text{rvec-norm } v = 1$
and $\text{dim-col } A = \text{dim-vec } v$
and $0 < \text{dim-col } A$
shows $\text{rvec-norm } (A *_v v) \leq L2\text{-op-nrm } A$
 ⟨proof⟩

lemma *max-sgval-le-L2-op-nrm*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{max-sgval } A \leq L2\text{-op-nrm } A$
 ⟨proof⟩

lemma *vec-norm-leq-L2-op-nrm*:
assumes $A \in \text{carrier-mat } n \ n$
and $v \in \text{carrier-vec } n$
and $0 < n$
and $\text{vec-norm } v = 1$
shows $\text{vec-norm } (A *_v v) \leq \text{L2-op-nrm } A$
 $\langle \text{proof} \rangle$

lemma *rvec-norm-leq-L2-op-nrm*:
assumes $A \in \text{carrier-mat } n \ n$
and $v \in \text{carrier-vec } n$
and $0 < n$
and $\text{rvec-norm } v = 1$
shows $\text{rvec-norm } (A *_v v) \leq \text{L2-op-nrm } A$ $\langle \text{proof} \rangle$

lemma *cmo-d-trace-rank-le-L2-op-nrm*:
assumes $A \in \text{carrier-mat } n \ n$
and $v \in \text{carrier-vec } n$
and $0 < n$
and $\text{rvec-norm } v = 1$
shows $\text{cmo-d } (\text{Complex-Matrix.trace } (A * \text{rank-1-proj } v)) \leq \text{L2-op-nrm } A$
 $\langle \text{proof} \rangle$

lemma *expect-val-L2-op-nrm*:
fixes $A :: \text{complex Matrix.mat}$
assumes $A \in \text{carrier-mat } n \ n$
and $R \in \text{carrier-mat } n \ n$
and $0 < n$
and $\text{density-operator } R$
shows $\text{cmo-d } (\text{Complex-Matrix.trace } (A * R)) \leq \text{L2-op-nrm } A$
 $\langle \text{proof} \rangle$

6.3 Consequences for the \mathcal{L}_2 operator norm

lemma *L2-op-nrm-geq-0*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $0 \leq \text{L2-op-nrm } A$
 $\langle \text{proof} \rangle$

lemma *L2-op-nrm-max-sgval-eq*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{L2-op-nrm } A = \text{max-sgval } A$
 $\langle \text{proof} \rangle$

lemma *rvec-leq-L2-op-nrm*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
and $v \in \text{carrier-vec } n$
shows $\text{rvec-norm } (A *_v v) \leq (\text{L2-op-nrm } A) * \text{rvec-norm } v$
 $\langle \text{proof} \rangle$

lemma *L2-op-nrm-mult-le*:
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{L2-op-nrm } (A*B) \leq \text{L2-op-nrm } A * \text{L2-op-nrm } B$
 $\langle \text{proof} \rangle$

lemma *L2-op-nrm-smult*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{L2-op-nrm } (c \cdot_m A) = \text{cmod } c * \text{L2-op-nrm } A$
 $\langle \text{proof} \rangle$

lemma *L2-op-nrm-uminus*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{L2-op-nrm } (-A) = \text{L2-op-nrm } A$
 $\langle \text{proof} \rangle$

lemma *L2-op-nrm-triangle*:
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{L2-op-nrm } (A+B) \leq \text{L2-op-nrm } A + \text{L2-op-nrm } B$
 $\langle \text{proof} \rangle$

lemma *L2-op-nrm-triangle'*:
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
shows $\text{L2-op-nrm } (A-B) \leq \text{L2-op-nrm } A + \text{L2-op-nrm } B$
 $\langle \text{proof} \rangle$

lemma *hermitian-max-sgval-eq*:
fixes $A::\text{complex Matrix.mat}$
assumes *hermitian* A
and $0 < \text{dim-row } A$
shows $\text{max-sgval } A = \text{spmax } A$
 $\langle \text{proof} \rangle$

lemma *hermitian-L2-op-nrm-spmax-eq*:
fixes $A::\text{complex Matrix.mat}$

assumes *hermitian A*
and $0 < \text{dim-row } A$
shows $L2\text{-op-nrm } A = \text{spmax } A$
 $\langle \text{proof} \rangle$

lemma *hermitian-L2-op-nrm-sqrt*:
fixes *A::complex Matrix.mat*
assumes *hermitian A*
and $0 < \text{dim-row } A$
shows $L2\text{-op-nrm } A = \text{sqrt } (L2\text{-op-nrm } (A*A))$
 $\langle \text{proof} \rangle$

lemma *idty-L2-op-nrm*:
assumes $0 < n$
shows $L2\text{-op-nrm } (1_m \ n) = 1$
 $\langle \text{proof} \rangle$

lemma *commutator-L2-op-nrm-le*:
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
shows $L2\text{-op-nrm } (\text{commutator } A \ B) \leq 2 * L2\text{-op-nrm } A * L2\text{-op-nrm } B$
 $\langle \text{proof} \rangle$

lemma *herm-sq-id-L2-op-nrm*:
assumes $0 < n$
and $A \in \text{carrier-mat } n \ n$
and *hermitian A*
and $A*A = 1_m \ n$
shows $L2\text{-op-nrm } A = 1$
 $\langle \text{proof} \rangle$

lemma *comm-L2-op-nrm-le*:
assumes $A \in \text{carrier-mat } n \ n$
and $B \in \text{carrier-mat } n \ n$
and $0 < n$
and $A*A = 1_m \ n$
and $B*B = 1_m \ n$
and *hermitian A*
and *hermitian B*
shows $L2\text{-op-nrm } (\text{commutator } A \ B) \leq 2$
 $\langle \text{proof} \rangle$

lemma *idty-smult-nat-L2-op-nrm*:
assumes $0 < n$
shows $L2\text{-op-nrm } ((m::\text{nat}) \cdot_m (1_m \ n)) = m$
 $\langle \text{proof} \rangle$
end

```

theory Density-Matrix-Basics
  imports
    Matrix-L2-Operator-Norm
begin

```

7 On density matrices

7.1 Density matrix characterization

Density matrices are defined as positive operators with trace 1, we prove in this section that they are exactly the convex combinations of pure states.

```

lemma (in cpx-sq-mat) mixed-state-density-operator:
  assumes  $\bigwedge i. i \in \{..< (n::nat)\} \implies 0 \leq p\ i$ 
  and  $\text{sum } p\ \{..< n\} = 1$ 
  and  $\bigwedge i. i \in \{..< n\} \implies \text{dim-vec } (v\ i) = \text{dim}R$ 
  and  $\bigwedge i. i \in \{..< n\} \implies \|v\ i\| = 1$ 
shows density-operator ( $\text{sum-mat } (\lambda\ i. (p\ i)\ \cdot_m\ (\text{rank-1-proj } (v\ i)))\ \{..< n\}$ )
   $\langle \text{proof} \rangle$ 

```

```

lemma (in cpx-sq-mat) density-operator-mixed-state:
  assumes  $R \in \text{fc-mats}$ 
  and density-operator  $R$ 
shows  $\exists p\ v\ (n::nat). (\forall i \in \{..< n\}. 0 \leq p\ i) \wedge$ 
   $(\forall i \in \{..< n\}. \text{dim-vec } (v\ i) = \text{dim}R) \wedge$ 
   $(\forall i \in \{..< n\}. \|v\ i\| = 1) \wedge (\text{sum } p\ \{..< n\} = 1) \wedge$ 
   $(R = \text{sum-mat } (\lambda\ i. (p\ i)\ \cdot_m\ (\text{rank-1-proj } (v\ i)))\ \{..< n\})$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma (in cpx-sq-mat) density-operator-iff-mixed-state:
  assumes  $R \in \text{fc-mats}$ 
  shows density-operator  $R \longleftrightarrow$ 
   $(\exists p\ v\ (n::nat). (\forall i \in \{..< n\}. 0 \leq p\ i) \wedge$ 
   $(\forall i \in \{..< n\}. \text{dim-vec } (v\ i) = \text{dim}R) \wedge$ 
   $(\forall i \in \{..< n\}. \|v\ i\| = 1) \wedge (\text{sum } p\ \{..< n\} = 1) \wedge$ 
   $(R = \text{sum-mat } (\lambda\ i. (p\ i)\ \cdot_m\ (\text{rank-1-proj } (v\ i)))\ \{..< n\}))$  (is  $?L \longleftrightarrow ?R$ )
   $\langle \text{proof} \rangle$ 

```

7.2 Separable density matrices

We define the notion of a separable density matrix: this is a matrix of the form $\sum_{i=1}^n p_i \rho_A^i \otimes \rho_B^i$, where the p_i s are positive and sum up to 1.

```

definition separately-decomposes where
  separately-decomposes  $R\ (n::nat)\ nA\ nB\ K\ F\ S \equiv$ 
   $(\forall a < n. (0::\text{complex}) \leq (\text{complex-of-real } (K\ a))) \wedge$ 
   $F\ a \in \text{carrier-mat } nA\ nA \wedge S\ a \in \text{carrier-mat } nB\ nB \wedge$ 
  density-operator  $(F\ a) \wedge \text{density-operator } (S\ a) \wedge 0 < nA * nB \wedge$ 

```

$$\text{sum } K \{..< n\} = 1 \wedge R = \text{fixed-carrier-mat.sum-mat } (nA * nB) (nA * nB) \\ (\lambda a. K a \cdot_m ((F a) \otimes (S a))) \{..< n\}$$

definition *separable-density* **where**

separable-density nA nB $R \equiv$

$\exists (n::\text{nat}) K F S. \text{separately-decomposes } R \ n \ nA \ nB \ K \ F \ S$

lemma *separately-decomposes-carrier*:

assumes *separately-decomposes* $R \ (n::\text{nat}) \ nA \ nB \ K \ F \ S$

and $0 < nA$

and $0 < nB$

shows $R \in \text{carrier-mat } (nA*nB) (nA*nB)$

<proof>

lemma *separately-decomposes-carrier-pos*:

assumes *separately-decomposes* $R \ n \ nA \ nB \ K \ F \ S$

shows $0 < nA \ 0 < nB$

<proof>

lemma *separable-density-carrier*:

assumes *separable-density* $nA \ nB \ R$

and $0 < nA$

and $0 < nB$

shows $R \in \text{carrier-mat } (nA*nB) (nA*nB)$

<proof>

lemma *separately-decomposes-trace*:

assumes *separately-decomposes* $R \ n \ nA \ nB \ K \ F \ S$

shows *Complex-Matrix.trace* $R = 1$

<proof>

lemma *separately-decomposes-positive*:

assumes *separately-decomposes* $R \ n \ nA \ nB \ K \ F \ S$

and $0 < nA$

and $0 < nB$

shows *Complex-Matrix.positive* R

<proof>

A separable density matrix is indeed a density matrix:

lemma *separable-density-operator*:

assumes *separable-density* $nA \ nB \ R$

and $0 < nA$

and $0 < nB$

shows *density-operator* R *<proof>*

7.3 Characterization of pure states

A density matrix represents a pure state if it is the rank 1 projection of a single vector. These can be characterized either as the density matrices with

a square of trace 1, or as the density matrices that are projectors.

definition *pure-density-operator* **where**
pure-density-operator $R \equiv (\exists v. R = \text{rank-1-proj } v)$

lemma *density-pure-single-diag*:
assumes $A \in \text{carrier-mat } n \ n$
and $\text{Complex-Matrix.trace } A = (1::\text{real})$
and $\text{Complex-Matrix.trace } (A*A) = (1::\text{real})$
and *unitary-diag* $A \ B \ U$
and $I = \{0 \ .. < n\}$
and $\forall i \in I. A \ \$\$ \ (i,i) \geq 0$
and $\forall i \in I. B \ \$\$ \ (i,i) \geq 0$
shows $\exists j \in I. B \ \$\$ \ (j,j) = 1 \wedge (\forall i \in I - \{j\}. B \ \$\$ \ (i,i) = 0)$
 $\langle \text{proof} \rangle$

lemma *rank-1-proj-square-trace*:
fixes $v::\text{complex Matrix.vec}$
assumes $A = \text{rank-1-proj } v$
shows $\text{Complex-Matrix.trace } (A*A) = \|v\|^2 * \text{Complex-Matrix.trace } A$
 $\langle \text{proof} \rangle$

lemma *rank-1-proj-trace'*:
assumes $\text{Complex-Matrix.trace } (\text{rank-1-proj } v) = 1$
shows $\|v\| = 1$
 $\langle \text{proof} \rangle$

lemma *density-square-pure*:
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
and *density-operator* A
and $\text{Complex-Matrix.trace } (A*A) = 1$
shows *pure-density-operator* A
 $\langle \text{proof} \rangle$

lemma *density-square-pure'*:
assumes *density-operator* A
and $A = \text{rank-1-proj } v$
shows $\text{Complex-Matrix.trace } (A*A) = 1$
 $\langle \text{proof} \rangle$

lemma
assumes $A \in \text{carrier-mat } n \ n$
and $0 < n$
and *density-operator* A
shows *pure-density-charact*:
 $(\text{pure-density-operator } A) \longleftrightarrow (\text{Complex-Matrix.trace } (A*A) = 1)$
and *pure-density-charact'*:
 $(\text{pure-density-operator } A) \longleftrightarrow (A*A = A)$
 $\langle \text{proof} \rangle$

8 Quantum expectation values and traces

The expectation value of a projective measurement is the average outcome value of the measurement, where each outcome value is weighted by the probability that it occurs. We show that the expectation value of a density matrix ρ for an observable represented by the Hermitian matrix A is $\text{Tr}(A \cdot \rho)$.

definition (in *cpx-sq-mat*) *expect-value* **where**

expect-value $R \ p \ M =$

$\text{sum } (\lambda i. \text{meas-outcome-prob } R \ M \ i * (\text{meas-outcome-val } (M \ i))) \ \{.. < p\}$

definition (in *cpx-sq-mat*) *obs-expect-value* **where**

obs-expect-value $R \ A =$

$\text{expect-value } R \ (\text{proj-meas-size } (\text{make-pm } A)) \ (\text{proj-meas-outcomes } (\text{make-pm } A))$

lemma (in *cpx-sq-mat*) *expect-value-trace*:

assumes *proj-measurement* $p \ M$

and $R \in \text{fc-mats}$

shows *expect-value* $R \ p \ M =$

$\text{Complex-Matrix.trace } (\text{sum-mat}$

$(\lambda i. \text{meas-outcome-val } (M \ i) \cdot_m (\text{meas-outcome-prj } (M \ i))) \ \{.. < p\} * R)$

<proof>

lemma (in *cpx-sq-mat*) *expect-value-hermitian*:

assumes $A \in \text{fc-mats}$

and *hermitian* A

and $\text{make-pm } A = (p, M)$

and $R \in \text{fc-mats}$

shows *expect-value* $R \ p \ M = \text{Complex-Matrix.trace } (A * R)$

<proof>

lemma *obs-expect-value*:

assumes $A \in \text{carrier-mat } n \ n$

and *hermitian* A

and $R \in \text{carrier-mat } n \ n$

and $0 < n$

shows *cpx-sq-mat.obs-expect-value* $n \ n \ R \ A = \text{Complex-Matrix.trace } (A * R)$

<proof>

end

theory *Tsirelson*

imports

Projective-Measurements.CHSI-Inequality

Matrix-L2-Operator-Norm Density-Matrix-Basics

begin

This part contains a formalization of the CHSH operator and the CHSH

quantum expectation, along with Tsirelson's proof that this quantum expectation cannot be greater than $2 \cdot \sqrt{2}$. The development of this proof permits to extract the additional result that when one of the parties involved in the CHSH experiment makes measurements on commuting observables, the quantum expectation cannot be greater than 2. This is the same upper-bound as in the case where a local hidden variable hypothesis is made.

9 CHSH inequalities

The CHSH operator is used to represent the experiment in which two parties each perform measurements using two observables, respectively A_1, A_2 and B_1, B_2 . Given the resource R , in general a density matrix representing an entangled state, the CHSH expectation represents the quantum expectation of performing simultaneous measurements on R . The CHSH setting also assumes that along with being Hermitian matrices, all the squared observables are equal to the identity and commute with the observables of the other party.

9.1 Some intermediate results for particular observables

lemma *chsh-complex:*

fixes $A0::\text{complex}$

assumes $A0 \in \text{Reals}$

and $B0 \in \text{Reals}$

and $A1 \in \text{Reals}$

and $B1 \in \text{Reals}$

and $|A0 * B1| \leq 1$

and $|A0 * B0| \leq 1$

and $|A1 * B0| \leq 1$

and $|A1 * B1| \leq 1$

shows $|A0 * B1 - A0 * B0 + A1 * B0 + A1 * B1| \leq 2$

<proof>

lemma (in *bin-cpx*) *Z-XpZ-rho-trace:*

shows $\text{Complex-Matrix.trace } (Z-I * I-XpZ * \text{rho-psim}) = 1/\text{sqrt } 2$

<proof>

lemma (in *bin-cpx*) *X-XpZ-rho-trace:*

shows $\text{Complex-Matrix.trace } (X-I * I-XpZ * \text{rho-psim}) = 1/\text{sqrt } 2$

<proof>

lemma (in *bin-cpx*) *X-ZmX-rho-trace:*

shows $\text{Complex-Matrix.trace } (X-I * I-ZmX * \text{rho-psim}) = 1/\text{sqrt } 2$

<proof>

lemma (in *bin-cpx*) *Z-ZmX-rho-trace*:
 shows *Complex-Matrix.trace* ($Z \cdot I * I \cdot ZmX * rho-psim$) = $-1 / \sqrt{2}$
 ⟨*proof*⟩

9.2 The CHSH operator and expectation

definition *CHSH-op* :: 'a::conjugatable-field *Matrix.mat* ⇒ 'a *Matrix.mat* ⇒
 'a *Matrix.mat* ⇒ 'a *Matrix.mat* ⇒ 'a *Matrix.mat*

where

CHSH-op *A0 A1 B0 B1* = $A0 * B1 - A0 * B0 + A1 * B0 + A1 * B1$

definition *CHSH-expect* :: 'a::conjugatable-field *Matrix.mat* ⇒ 'a *Matrix.mat* ⇒
 'a *Matrix.mat* ⇒ 'a *Matrix.mat* ⇒ 'a *Matrix.mat* ⇒ 'a

where

CHSH-expect *A0 A1 B0 B1 R* = *Complex-Matrix.trace* ((*CHSH-op* *A0 A1 B0 B1*)
 * *R*)

definition *CHSH-cond* :: nat ⇒ 'a::conjugatable-field *Matrix.mat* ⇒
 'a::conjugatable-field *Matrix.mat* ⇒

'a::conjugatable-field *Matrix.mat* ⇒ 'a::conjugatable-field *Matrix.mat* ⇒ bool

where

CHSH-cond *n A0 A1 B0 B1* =

(*A0* ∈ *carrier-mat* *n n* ∧

$A0 * A0 = 1_m \ n$ ∧

A1 ∈ *carrier-mat* *n n* ∧

$A1 * A1 = 1_m \ n$ ∧

B0 ∈ *carrier-mat* *n n* ∧

$B0 * B0 = 1_m \ n$ ∧

B1 ∈ *carrier-mat* *n n* ∧

$B1 * B1 = 1_m \ n$ ∧

$A0 * B1 = B1 * A0$ ∧

$A0 * B0 = B0 * A0$ ∧

$A1 * B0 = B0 * A1$ ∧

$A1 * B1 = B1 * A1$)

definition *CHSH-cond-hermit* **where**

CHSH-cond-hermit *n A0 A1 B0 B1* ⇔ *CHSH-cond* *n A0 A1 B0 B1* ∧ *hermitian*
A0 ∧

hermitian *A1* ∧ *hermitian* *B0* ∧ *hermitian* *B1*

lemma *CHSH-op-dim*:

assumes *A0* ∈ *carrier-mat* *n m*

and *A1* ∈ *carrier-mat* *n m*

and *B0* ∈ *carrier-mat* *m p*

and *B1* ∈ *carrier-mat* *m p*

shows *CHSH-op* *A0 A1 B0 B1* ∈ *carrier-mat* *n p* ⟨*proof*⟩

lemma *CHSH-op-hermitian*:

assumes *hermitian* *A0*

and *hermitian* $B0$
and *hermitian* $A1$
and *hermitian* $B1$
and $A0 * B0 = B0 * A0$
and $A1 * B0 = B0 * A1$
and $A0 * B1 = B1 * A0$
and $A1 * B1 = B1 * A1$
shows *hermitian* ($CHSH\text{-op}$ $A0$ $A1$ $B0$ $B1$)
 ⟨*proof*⟩

lemma *CHSH-cond-hermit-expect-eq*:
assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $R \in \text{carrier-mat } n \ n$
and $0 < n$
shows *CHSH-expect* $A0$ $A1$ $B0$ $B1$ $R =$
 $\text{cpx-sq-mat.obs-expect-value } n \ n \ R \ (CHSH\text{-op } A0 \ A1 \ B0 \ B1)$
 ⟨*proof*⟩

lemma *CHSH-op-expand-right*:
fixes $A0 :: 'a :: \text{conjugatable-field Matrix.mat}$
assumes $A0 \in \text{carrier-mat } n \ m$
and $A1 \in \text{carrier-mat } n \ m$
and $B0 \in \text{carrier-mat } m \ p$
and $B1 \in \text{carrier-mat } m \ p$
and $R \in \text{carrier-mat } p \ p'$
shows $(CHSH\text{-op } A0 \ A1 \ B0 \ B1) * R =$
 $A0 * B1 * R - A0 * B0 * R + A1 * B0 * R + A1 * B1 * R$
 ⟨*proof*⟩

lemma *CHSH-op-expand-left*:
fixes $A0 :: 'a :: \text{conjugatable-field Matrix.mat}$
assumes $A0 \in \text{carrier-mat } n \ m$
and $A1 \in \text{carrier-mat } n \ m$
and $B0 \in \text{carrier-mat } m \ p$
and $B1 \in \text{carrier-mat } m \ p$
and $R \in \text{carrier-mat } p \ n$
shows $R * (CHSH\text{-op } A0 \ A1 \ B0 \ B1) =$
 $R * (A0 * B1) - R * (A0 * B0) + R * (A1 * B0) + R * (A1 * B1)$
 ⟨*proof*⟩

lemma *CHSH-expect-expand*:
assumes $A0 \in \text{carrier-mat } n \ m$
and $A1 \in \text{carrier-mat } n \ m$
and $B0 \in \text{carrier-mat } m \ p$
and $B1 \in \text{carrier-mat } m \ p$
and $R \in \text{carrier-mat } p \ n$
shows *CHSH-expect* $A0$ $A1$ $B0$ $B1$ $R =$
 $\text{Complex-Matrix.trace } (A0 * B1 * R) -$
 $\text{Complex-Matrix.trace } (A0 * B0 * R) +$

$\text{Complex-Matrix.trace } (A1 * B0 * R) +$
 $\text{Complex-Matrix.trace } (A1 * B1 * R)$
 ⟨proof⟩

lemma *CHSH-condD*:

assumes *CHSH-cond* n $A0$ $A1$ $B0$ $B1$
shows $A0 \in \text{carrier-mat } n$
 $A0 * A0 = 1_m$ n
 $A1 \in \text{carrier-mat } n$
 $A1 * A1 = 1_m$ n
 $B0 \in \text{carrier-mat } n$
 $B0 * B0 = 1_m$ n
 $B1 \in \text{carrier-mat } n$
 $B1 * B1 = 1_m$ n
 $A0 * B1 = B1 * A0$
 $A0 * B0 = B0 * A0$
 $A1 * B0 = B0 * A1$
 $A1 * B1 = B1 * A1$ ⟨proof⟩

lemma *CHSH-cond-simps*[*simp*]:

assumes *CHSH-cond* n $A0$ $A1$ $B0$ $B1$
shows $A1 * B1 * (A0 * B1) = A1 * A0$
 $A1 * B1 * (A1 * B0) = B1 * B0$
 $A1 * B1 * (A1 * B1) = 1_m$ n
 $A1 * B1 * (A0 * B0) = A1 * A0 * (B1 * B0)$
 $A1 * B0 * (A0 * B1) = A1 * A0 * (B0 * B1)$
 $A1 * B0 * (A0 * B0) = A1 * A0$
 $A1 * B0 * (A1 * B0) = 1_m$ n
 $A1 * B0 * (A1 * B1) = B0 * B1$
 $A0 * B0 * (A0 * B1) = B0 * B1$
 $A0 * B0 * (A0 * B0) = 1_m$ n
 $A0 * B0 * (A1 * B0) = A0 * A1$
 $A0 * B0 * (A1 * B1) = A0 * A1 * (B0 * B1)$
 $A0 * B1 * (A0 * B1) = 1_m$ n
 $A0 * B1 * (A0 * B0) = B1 * B0$
 $A0 * B1 * (A1 * B0) = A0 * A1 * (B1 * B0)$
 $A0 * B1 * (A1 * B1) = A0 * A1$
 ⟨proof⟩

lemma *CHSH-op-square*:

assumes *CHSH-cond* n $A0$ $A1$ $B0$ $B1$
shows $(\text{CHSH-op } A0$ $A1$ $B0$ $B1) * (\text{CHSH-op } A0$ $A1$ $B0$ $B1) =$
 $(4::\text{nat}) \cdot_m (1_m$ $n) - (\text{commutator } A0$ $A1) * (\text{commutator } B0$ $B1)$
 ⟨proof⟩

lemma *CHSH-cond-hermitD*:

assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
shows *CHSH-cond* n $A0$ $A1$ $B0$ $B1$
hermitian $A0$

hermitian A1
hermitian B0
hermitian B1
 ⟨proof⟩

lemma *CHSH-cond-hermit-unitary*:
assumes *CHSH-cond-hermit n A0 A1 B0 B1*
shows *unitary A0 unitary A1 unitary B0 unitary B1*
 ⟨proof⟩

lemma *CHSH-expect-add*:
assumes *A0 ∈ carrier-mat n n*
and *A1 ∈ carrier-mat n n*
and *B0 ∈ carrier-mat n n*
and *B1 ∈ carrier-mat n n*
and *R0 ∈ carrier-mat n n*
and *R1 ∈ carrier-mat n n*
shows *CHSH-expect A0 A1 B0 B1 (R0 + R1) =*
CHSH-expect A0 A1 B0 B1 R0 +
CHSH-expect A0 A1 B0 B1 R1
 ⟨proof⟩

lemma *CHSH-expect-zero*:
assumes *A0 ∈ carrier-mat n n*
and *A1 ∈ carrier-mat n n*
and *B0 ∈ carrier-mat n n*
and *B1 ∈ carrier-mat n n*
shows *CHSH-expect A0 A1 B0 B1 (0_m n n) = 0*
 ⟨proof⟩

lemma (in *cpx-sq-mat*) *CHSH-expect-sum*:
assumes *finite S*
and *A0 ∈ fc-mats*
and *A1 ∈ fc-mats*
and *B0 ∈ fc-mats*
and *B1 ∈ fc-mats*
and $\bigwedge i. i \in S \implies R\ i \in \text{fc-mats}$
shows *CHSH-expect A0 A1 B0 B1 (sum-mat R S) =*
sum (λi. CHSH-expect A0 A1 B0 B1 (R i)) S ⟨proof⟩

lemma *CHSH-expect-smult*:
assumes *A0 ∈ carrier-mat n n*
and *A1 ∈ carrier-mat n n*
and *B0 ∈ carrier-mat n n*
and *B1 ∈ carrier-mat n n*
and *R0 ∈ carrier-mat n n*
shows *CHSH-expect A0 A1 B0 B1 (a ·_m R0) =*
*a * CHSH-expect A0 A1 B0 B1 R0*

<proof>

lemma *CHSH-expect-real:*

assumes $0 < n$
and *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $R \in$ *carrier-mat* n n
and *Complex-Matrix.positive* R
shows *CHSH-expect* $A0$ $A1$ $B0$ $B1$ $R \in$ *Reals*

<proof>

lemma *CHSH-op-square-L2-op-nrm-le:*

assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $0 < n$
shows *L2-op-nrm* $((\text{CHSH-op } A0 \ A1 \ B0 \ B1) * (\text{CHSH-op } A0 \ A1 \ B0 \ B1)) \leq 8$

<proof>

lemma *CHSH-op-square-spmx-le:*

assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $0 < n$
shows *spm* $((\text{CHSH-op } A0 \ A1 \ B0 \ B1) * (\text{CHSH-op } A0 \ A1 \ B0 \ B1)) \leq 8$

<proof>

lemma *CHSH-op-L2-op-nrm-le:*

assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $0 < n$
shows *L2-op-nrm* $(\text{CHSH-op } A0 \ A1 \ B0 \ B1) \leq 2 * \text{sqrt } 2$

<proof>

lemma (**in** *cpx-sq-mat*) *CHSH-cond-hermit-lhv-upper:*

assumes *CHSH-cond-hermit* $\text{dim}R$ $A0$ $A1$ $B0$ $B1$
and *lhv* M $A0$ $B1$ R $U0$ $V1$
and *lhv* M $A0$ $B0$ R $U0$ $V0$
and *lhv* M $A1$ $B0$ R $U1$ $V0$
and *lhv* M $A1$ $B1$ R $U1$ $V1$
and $0 < n$
shows $|(LINT \ w|M. \ \text{qt-expect } A0 \ U0 \ w * \ \text{qt-expect } B1 \ V1 \ w) -$
 $(LINT \ w|M. \ \text{qt-expect } A0 \ U0 \ w * \ \text{qt-expect } B0 \ V0 \ w) +$
 $(LINT \ w|M. \ \text{qt-expect } A1 \ U1 \ w * \ \text{qt-expect } B0 \ V0 \ w) +$
 $(LINT \ w|M. \ \text{qt-expect } A1 \ U1 \ w * \ \text{qt-expect } B1 \ V1 \ w)|$
 ≤ 2

<proof>

lemma (**in** *cpx-sq-mat*) *CHSH-expect-lhv-lint-eq:*

assumes $R \in$ *fc-mats*
and *Complex-Matrix.positive* R
and *CHSH-cond-hermit* $\text{dim}R$ $A0$ $A1$ $B0$ $B1$
and *lhv* M $A0$ $B1$ R $U0$ $V1$
and *lhv* M $A0$ $B0$ R $U0$ $V0$
and *lhv* M $A1$ $B0$ R $U1$ $V0$

and $lhv\ M\ A1\ B1\ R\ U1\ V1$
shows $(LINT\ w|M.\ qt-expect\ A0\ U0\ w * qt-expect\ B1\ V1\ w) -$
 $(LINT\ w|M.\ qt-expect\ A0\ U0\ w * qt-expect\ B0\ V0\ w) +$
 $(LINT\ w|M.\ qt-expect\ A1\ U1\ w * qt-expect\ B0\ V0\ w) +$
 $(LINT\ w|M.\ qt-expect\ A1\ U1\ w * qt-expect\ B1\ V1\ w) =$
 $CHSH-expect\ A0\ A1\ B0\ B1\ R\ (is\ ?L = ?R)$
 $\langle proof \rangle$

9.3 CHSH inequality for separable density matrices

definition *CHSH-cond-local where*

$CHSH-cond-local\ n\ m\ A0\ A1\ B0\ B1 \equiv$
 $A0 \in carrier-mat\ n\ n \wedge A1 \in carrier-mat\ n\ n \wedge$
 $B0 \in carrier-mat\ m\ m \wedge B1 \in carrier-mat\ m\ m \wedge$
 $hermitian\ A0 \wedge hermitian\ A1 \wedge hermitian\ B0 \wedge hermitian\ B1 \wedge$
 $A0 * A0 = 1_m\ n \wedge A1 * A1 = 1_m\ n \wedge B0 * B0 = 1_m\ m \wedge B1 * B1 = 1_m\ m$

lemma *CHSH-cond-local-imp-cond-hermit:*

assumes $CHSH-cond-local\ n\ m\ A0\ A1\ B0\ B1$

and $0 < n$

and $0 < m$

shows $CHSH-cond-hermit\ (n*m)\ (A0 \otimes 1_m\ m)\ (A1 \otimes 1_m\ m)$
 $(1_m\ n \otimes B0)\ (1_m\ n \otimes B1)$

$\langle proof \rangle$

lemma *limit-CHSH-cond:*

shows $CHSH-cond-hermit\ 4\ Z-I\ X-I\ I-ZmX\ I-XpZ$

$\langle proof \rangle$

lemma *CHSH-expect-separable-expand:*

assumes $separately-decomposes\ R\ n\ nA\ nB\ K\ F\ S$

and $A0 \in carrier-mat\ nA\ nA$

and $A1 \in carrier-mat\ nA\ nA$

and $B0 \in carrier-mat\ nB\ nB$

and $B1 \in carrier-mat\ nB\ nB$

shows $CHSH-expect\ (A0 \otimes 1_m\ nB)\ (A1 \otimes 1_m\ nB)\ (1_m\ nA \otimes B0)\ (1_m\ nA \otimes B1)$
 $R =$

$sum\ (\lambda a.\ K\ a * CHSH-expect\ (A0 \otimes 1_m\ nB)\ (A1 \otimes 1_m\ nB)\ (1_m\ nA \otimes B0)\ (1_m$
 $nA \otimes B1)$
 $((F\ a) \otimes (S\ a))\ \{.. < n\}$

$\langle proof \rangle$

lemma *CHSH-expect-tensor-leq:*

assumes $CHSH-cond-local\ nA\ nB\ A0\ A1\ B0\ B1$

and $RA \in carrier-mat\ nA\ nA$

and $density-operator\ RA$

and $RB \in carrier-mat\ nB\ nB$

and $density-operator\ RB$

and $0 < nA$

and $0 < nB$
shows $|CHSH-expect (A0 \otimes 1_m nB) (A1 \otimes 1_m nB) (1_m nA \otimes B0) (1_m nA \otimes B1) (RA \otimes RB)| \leq 2$
 (proof)

9.4 CHSH inequality for commuting observables

lemma *CHSH-op-square-commute-L2-op-nrm-eq*:
assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $0 < n$
and *commutator* $A0$ $A1 = 0_m$ n $n \vee$ *commutator* $B0$ $B1 = 0_m$ n n
shows $L2-op-nrm ((CHSH-op$ $A0$ $A1$ $B0$ $B1) * (CHSH-op$ $A0$ $A1$ $B0$ $B1)) = 4$
 (proof)

lemma *CHSH-op-square-commute-spmx-eq*:
assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $0 < n$
and *commutator* $A0$ $A1 = 0_m$ n $n \vee$ *commutator* $B0$ $B1 = 0_m$ n n
shows $spmax ((CHSH-op$ $A0$ $A1$ $B0$ $B1) * (CHSH-op$ $A0$ $A1$ $B0$ $B1)) = 4$
 (proof)

lemma *CHSH-op-commute-L2-op-nrm-eq*:
assumes *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and $0 < n$
and *commutator* $A0$ $A1 = 0_m$ n $n \vee$ *commutator* $B0$ $B1 = 0_m$ n n
shows $L2-op-nrm (CHSH-op$ $A0$ $A1$ $B0$ $B1) = 2$
 (proof)

9.5 Result summary on the CHSH inequalities

Under the local hidden variable hypothesis, this value is bounded by 2.

lemma *CHSH-expect-lhv-leq*:
assumes $R \in$ *carrier-mat* n n
and $0 < n$
and *Complex-Matrix.positive* R
and *CHSH-cond-hermit* n $A0$ $A1$ $B0$ $B1$
and *cpx-sq-mat.lhv* n n M $A0$ $B1$ R $U0$ $V1$
and *cpx-sq-mat.lhv* n n M $A0$ $B0$ R $U0$ $V0$
and *cpx-sq-mat.lhv* n n M $A1$ $B0$ R $U1$ $V0$
and *cpx-sq-mat.lhv* n n M $A1$ $B1$ R $U1$ $V1$
shows $|CHSH-expect$ $A0$ $A1$ $B0$ $B1$ $R| \leq 2$
 (proof)

When the considered density operator is separable, this value is still bounded by 2.

lemma *CHSH-expect-separable-leq*:
assumes *CHSH-cond-local* nA nB $A0$ $A1$ $B0$ $B1$
and *separable-density* nA nB R
and $A0 \in$ *carrier-mat* nA nA

and $A1 \in \text{carrier-mat } nA \ nA$
and $B0 \in \text{carrier-mat } nB \ nB$
and $B1 \in \text{carrier-mat } nB \ nB$
shows $|CHSH\text{-expect } (A0 \otimes 1_m \ nB) (A1 \otimes 1_m \ nB) (1_m \ nA \otimes B0) (1_m \ nA \otimes B1) R|$
 ≤ 2
<proof>

When any of the pairs of observables used in the measurements commutes, this value remains bounded by 2.

lemma *CHSH-expect-commute-leq*:
assumes *CHSH-cond-hermit* $n \ A0 \ A1 \ B0 \ B1$
and $R \in \text{carrier-mat } n \ n$
and *density-operator* R
and $0 < n$
and *commutator* $A0 \ A1 = 0_m \ n \ n \vee$ *commutator* $B0 \ B1 = 0_m \ n \ n$
shows $|CHSH\text{-expect } A0 \ A1 \ B0 \ B1 \ R| \leq 2$
<proof>

In the general case, this value is bounded by $2 \cdot \sqrt{2}$.

lemma *CHSH-expect-gen-leq*:
assumes *CHSH-cond-hermit* $n \ A0 \ A1 \ B0 \ B1$
and $R \in \text{carrier-mat } n \ n$
and *density-operator* R
and $0 < n$
shows $|CHSH\text{-expect } A0 \ A1 \ B0 \ B1 \ R| \leq (2 * \text{sqrt } 2)$
<proof>

The bound $2 \cdot \sqrt{2}$ can be reached by a suitable choice of observables, when the Bell state is measured.

lemma *CHSH-expect-limit*:
shows $|CHSH\text{-expect } Z\text{-I } X\text{-I } I\text{-ZmX } I\text{-XpZ } \text{rho-psim}| = 2 * \text{sqrt } 2$
<proof>

end