Sillari Refutation

Huub Vromen

```
import Mathlib.Tactic
import Mathlib.Logic.Relation
```

Formal Analysis of Sillari’s Modal Logic Approach to Lewis’s Theory of Common Knowledge

This file demonstrates that Sillari’s (2005) modal logic approach to Lewis’s theory of common knowledge has fundamental limitations. We provide machine-verified counterexamples showing that key axioms fail and the proof of Lewis’s theorem is either false or trivially true.

> **Note:** This file explains *why modal logic cannot capture Lewis’s theory*. For the correct formalization, see Vromen (2024) using justification logic.

—

Overview

Sillari proposed formalizing Lewis’s theory using Kripke semantics, defining ”reason to believe” as a modal box operator and ”indication” as a conjunction of reason to believe and material implication.

This analysis shows these definitions are structurally inadequate for Lewis’s theory.

Main Results

1. **`B3` (Lewis’s `A1`) fails** — Counterexample in `B3_fails`
2. **`C4` (Cubitt-Sugden axiom) fails** — Counterexample in `C4_fails`
3. **Lewis’s theorem fails under local assumptions** — `Lewis_fails_1i`, `Lewis_fails_2i`
4. **Lewis’s theorem is trivial under global assumptions** — `Lewis_s_2`

The Root Problem

Lewis (1969, p. 52–53) defines indication counterfactually: ”*A indicates* to someone *x* that ___ if and only if, if *x* had reason to believe that *A* held, *x* would *thereby* have reason to believe that ___.”

The word *thereby* indicates that the reason for `p` *depends on* the reason for `A`.

Sillari’s definition `Ind i φ ψ := R i φ ∧ (φ → ψ)` is merely a conjunction—it cannot capture this evidential dependence. The reason for `ψ` might be completely independent of the reason for `φ`.

Comparison with Other Approaches

Feature	Sillari (this file)	Cubitt-Sugden	Vromen
R operator	Modal □	Primitive	`∃r. rb r i φ`
Indication	`R i φ ∧ (φ → ψ)`	Primitive	`R i (A → φ)`
A1 status	**FAILS** ×	Axiom	**THEOREM**
Lewis theorem	False/Trivial ×	Provable	**PROVEN**

References

- **Lewis, D. (1969).** *Convention: A Philosophical Study.* Cambridge, MA: Harvard University Press.
- **Sillari, G. (2005).** A logical framework for convention. *Synthese, 147*, 379–400.
- **Vromen, H. (2024).** Reasoning with reasons: Lewis on common knowledge. *Economics & Philosophy, 40*(2), 397–418.

File Structure

1. Kripke frame definitions
2. Modal operators (`R`, `Rg`, `Ind`)
3. Common reason to believe (`CRB`)
4. Axiom analysis (`B1`–`B11`)
5. Critical failures (`B3`, `C4`)
6. Lewis's theorem counterexamples

```
namespace Sillari
```

—

Section 1: Kripke Frame Structure

We define multi-agent Kripke frames with accessibility relations for each agent. This is the standard semantic framework for modal logic.

Multi-Agent Frame

A Kripke frame consists of:

- `World`: A type of possible worlds
- `rel`: For each agent `i`, a binary relation on worlds (`rel i w v` means world `v` is accessible from world `w` for agent `i`)

Interpretation: `rel i w v` means ”from world `w`, agent `i` considers world `v` possible.”

- A Kripke frame with accessibility relations for multiple agents.

```
structure MultiAgentFrame (Agent : Type*) where
  World : Type*
  rel : Agent → World → World → Prop
```

—

Section 2: Modal Operators

Standard definitions for modal implication, conjunction, and validity.

```
variable {Agent : Type*} {frame : MultiAgentFrame Agent}
```

Modal Implication

`(φ →ₘ ψ) w` — ”At world w, if φ holds then ψ holds”

This is just the standard material implication lifted to the modal level.

```
def modal_imp (φ ψ : frame.World → Prop) : frame.World → Prop :=
  fun w => φ w → ψ w
infixr:90 " →ₘ " => modal_imp
```

Modal Conjunction

`(φ ∧ₘ ψ) w` — ”Both φ and ψ hold at world w”

```
def modal_conj (φ ψ : frame.World → Prop) : frame.World → Prop :=
  fun w => φ w ∧ ψ w
infixr:70 " ∧ₘ " => modal_conj
```

Validity

`⊢ φ` — ”Formula φ is valid (holds at all worlds)”

```
def valid (φ : frame.World → Prop) : Prop := ∀ w, φ w
notation "⊢ " φ => valid φ
```

—

Section 3: Sillari’s Core Definitions

These are the operators that define Sillari’s approach. The key insight is that these definitions, while natural for standard epistemic logic, are inadequate for capturing Lewis’s notion of indication.

1. Reason to Believe (`R`)

`R i φ w` — ”Agent `i` has reason to believe φ at world w”

Definition: Agent `i` has reason to believe φ at world w if φ holds at all worlds accessible to `i` from w.

This is the standard modal box operator $\Box_i$.

Formal definition:

`R i φ w := ∀ v, rel i w v → φ v`

```
def R (i : Agent) (φ : frame.World → Prop) : frame.World → Prop :=
  fun w => ∀ v, frame.rel i w v → φ v
```

2. Group Reason to Believe (`Rg`)

`Rg φ w` — ”Everyone (all agents) has reason to believe φ at world w”

Formal definition:

`Rg φ w := ∀ i, R i φ w`

```
def Rg (φ : frame.World → Prop) : frame.World → Prop :=
  fun w => ∀ i, R i φ w
```

3. Indication (`Ind`)

> **`Ind i φ ψ w`** — ”State of affairs φ indicates ψ to agent `i` at world `w`”

This definition attempts to capture Lewis's idea that if φ indicates ψ, and you have reason to believe φ, then you *thereby* have reason to believe ψ.

Formal definition:

> `Ind i φ ψ w := R i φ w ∧ (φ w → ψ w)`

The conjunction has two parts:

1. `R i φ w`: agent `i` has reason to believe φ
2. `(φ w → ψ w)`: φ materially implies ψ at this world

Critical Problem: This is merely a conjunction—it cannot capture evidential dependence. The reason for ψ might be completely independent of the reason for φ. Lewis's ”thereby” requires that the reason for ψ *depends on* the reason for φ.

```
def Ind (i : Agent) (φ ψ : frame.World → Prop) : frame.World → Prop :=
  fun w => R i φ w ∧ (φ w → ψ w)
```

—

Section 4: Transitive Closure and Common Reason to Believe

Sillari defines common reason to believe via reachability in the Kripke frame. This section develops the machinery for expressing ”all worlds reachable by following any sequence of agents' accessibility relations.”

Connected Worlds

> **`connected w1 w2`** — ”Some agent's accessibility relation links `w1` to `w2`”

Two worlds are **connected** if there exists some agent `i` such that `w2` is accessible from `w1` via `i`'s relation.

```
def connected (w1 w2 : frame.World) : Prop :=
  ∃ i, frame.rel i w1 w2
```

Transitive Closure

We define the transitive closure of a relation using an inductive type.

We use our own definition rather than Mathlib's `ReflTransGen` because we need transitive but *not* reflexive closure.

Inductive structure:

- `base`: Single step (one edge in the relation)
- `step`: Prepend a step to an existing path (inductive case)

This allows constructing paths of arbitrary finite length.

Example: If we have edges `x → y → z`, we can construct:

- `trcl r x y` via `base`
- `trcl r y z` via `base`
- `trcl r x z` via `step` (combining the above)

```
inductive trcl (r : frame.World → frame.World → Prop) : frame.World → frame.World →
 ↪  Prop
  | base {x y} : r x y → trcl r x y
  | step {x y z} : r x y → trcl r y z → trcl r x z

lemma trcl.head {r : frame.World → frame.World → Prop}
    (h : r x y) (p : trcl r y z) : trcl r x z :=
  trcl.step h p
```

Common Reason to Believe (CRB)

CRB ψ s — ”There is common reason to believe ψ at world s”

Definition: ψ holds at all worlds reachable from s via the transitive closure of all agents’ accessibility relations.

Formal definition:

```
CRB ψ s := ∀ w, trcl connected s w → ψ w
```

Interpretation: No matter how you follow the agents’ accessibility relations (mixing agents arbitrarily), you always reach worlds where ψ holds.

```
def CRB (ψ : frame.World → Prop) (s : frame.World) : Prop :=
  ∀ w, trcl (connected : frame.World → frame.World → Prop) s w → ψ w
```

Helper Lemmas for Transitive Closure

- Transitive closure is transitive

```
lemma trcl_trans {r : frame.World → frame.World → Prop} {x y z : frame.World} :
    trcl r x y → trcl r y z → trcl r x z := by
  intro hxy hyz
  induction hxy with
  | base hxy => exact trcl.step hxy hyz
  | step hxy _ ih => exact trcl.step hxy (ih hyz)
```

- Single step inclusion

```
lemma trcl_single {r : frame.World → frame.World → Prop} {x y : frame.World}
    (h : r x y) : trcl r x y :=
  trcl.base h
```

—

Section 5: Helper Definitions for Counterexamples

To construct our counterexamples, we need to specify simple frame structures. These definitions help us describe frames with a fixed number of worlds and agents.

- Two distinct worlds

```
def two_worlds : frame.World → frame.World → Prop :=
  fun w1 w2 => w2 ≠ w1
```

- Three distinct worlds

```
def three_worlds : frame.World → frame.World → frame.World → Prop :=
  fun w1 w2 w3 => w2 ≠ w1 ∧ w3 ≠ w2 ∧ w3 ≠ w1
```

- Two distinct agents

```
def two_agents : Agent → Agent → Prop :=
  fun i1 i2 => i1 ≠ i2 ∧ ∀ i, i = i1 ∨ i = i2
```

—

Section 6: Axiom Analysis

Sillari presents eleven axioms (`B1-B8`, `B10-B11`) in a proof-theoretic style, combined with semantic definitions of `R` and `Ind` via Kripke frames.

We investigate which axioms follow from the semantic definitions and which fail.

Results Summary

- `B1`, `B2`, `B4`, `B5`, `B6`: **PROVABLE** as lemmas from the definitions
- × **`B3`: FAILS** (we provide counterexample) — **CRITICAL PROBLEM**
- ? `B7`, `B8`: Not provable without special frame properties (transitivity, seriality)
- `B9`: Definition of indication
- `B10`, `B11`: **PROVABLE**

The failure of `B3` is a fundamental flaw because `B3` is Lewis's axiom `A1`, which is essential to Lewis's entire argument.

```
variable {φ ψ γ : frame.World → Prop}
```

Axiom B1: Modal Modus Ponens (K axiom)

If `i` has reason to believe `φ`, and `i` has reason to believe that `φ` implies `ψ`, then `i` has reason to believe `ψ`.

> **Formula:** `R i φ w → R i (φ →m ψ) w → R i ψ w`

Status PROVABLE

```
lemma B1 : ∀ w, R i φ w → R i (φ →m ψ) w → R i ψ w := by
  intros v h1 h2 u h3
  aesop
```

Axiom B2: Formation of Indication

If `i` has reason to believe `φ`, and `φ` materially implies `ψ` at this world, then `φ` indicates `ψ` to `i`.

> **Formula:** `R i φ w → (φ →m ψ) w → Ind i φ ψ w`

Status PROVABLE (follows directly from the definition of `Ind`)

```
lemma B2 : ∀ w, R i φ w → (φ →m ψ) w → Ind i φ ψ w := by
  intro w h1 h2
  exact ⟨h1, h2⟩
```

—

Critical Failure: Axiom B3 (Lewis's A1)

This is the **most important negative result**. B3 states that if i has reason to believe φ, and φ indicates ψ to i, then i has reason to believe ψ.

This is Lewis's "principle of detachment," essential to his entire argument.

> **Formula:** R i φ w → Ind i φ ψ w → R i ψ w

Counterexample: We construct a frame with worlds s and t, where i relates s to t only.

Visual representation:

```
Frame Structure:
┌───┐   rel i    ┌───┐
│ s │  ───────>  │ t │
└───┘            └───┘

Proposition: φ = "always true"
┌─────┬─────┬─────┐
│ w   │  s  │  t  │
├─────┼─────┼─────┤
│ φ w │  ✓  │  ✓  │
└─────┴─────┴─────┘

Target: ψ = "at world s"
┌─────┬─────┬─────┐
│ w   │  s  │  t  │
├─────┼─────┼─────┤
│ ψ w │  ✓  │  ✗  │
└─────┴─────┴─────┘
```

What holds:

1. R i φ s — Agent i has reason to believe φ at world s * Because φ is true at all accessible worlds (only world t, where φ t is true)
2. Ind i φ ψ s — φ indicates ψ to i at s *R i φ s (shown above)* φ s → ψ s (both are true at s)

What fails:

1. R i ψ s × — Agent i does NOT have reason to believe ψ * Because ψ t is false

Conclusion B3 fails. We have R i φ s ∧ Ind i φ ψ s but NOT R i ψ s.

Why this matters: Without B3, you cannot derive common reason to believe from reflexive common indicators. Lewis's entire proof collapses.

```
lemma B3_fails
    (h2a : ¬ frame.rel i s s)
    (h2b : frame.rel i s t) :
    ∃ (w : frame.World) (φ ψ : frame.World → Prop), R i φ w ∧ Ind i φ ψ w ∧ ¬ R i ψ w
     ↪ := by
  let w := s
  let ψ := fun w => w ≠ t
  let φ := fun w => w ≠ s
  have h4 : R i φ s := by intro v hv; aesop
  have h5 : Ind i φ ψ s := by rw [Ind]; aesop
  have h6 : ¬ R i ψ s := by aesop
  exact ⟨s, φ, ψ, h4, h5, h6⟩
```

Axiom B4: Transitivity of indication

Formula: `Ind i φ γ w → Ind i γ ψ w → Ind i φ ψ w`

Status PROVABLE

```
lemma B4 : ∀ w, Ind i φ γ w → Ind i γ ψ w → Ind i φ ψ w := by
  intro w h1 h2
  constructor
  { exact h1.1 }
  { have h4 : φ w → γ w := h1.2
    have h5 : γ w → ψ w := h2.2
    intro hw
    exact h5 (h4 hw) }
```

Axiom B5: Modus ponens for valid formulas

Formula: `(⊢ φ) → (⊢ (φ →ₘ ψ)) → (⊢ ψ)`

Status PROVABLE

```
lemma B5 : (⊢ φ) → (⊢ (φ →ₘ ψ)) → (⊢ ψ) := by
  intro h1 h2 w
  exact h2 w (h1 w)
```

Axiom B6: Necessitation rule

If φ indicates ψ to `i`, and ψ indicates γ to `i`, then φ indicates γ to `i`.

Formula: `(⊢ φ) → (⊢ R i φ)`

Status PROVABLE but implies logical omniscience

```
lemma B6 : (⊢ φ) → (⊢ R i φ) := by
  intro h1 u v _
  exact h1 v
```

axiom B7 and axiom B8: Not provable without special frame properties**

- B7 (Positive introspection): `R i φ → R i (R i φ)` Requires transitive accessibility relations
- B8 (Negative introspection): `¬ R i φ → R i (¬ R i φ)` Requires euclidean accessibility relations

Sillari states these axioms but does not use them in his main argument. We don't discuss them further.

(Sillari 2005, p. 388)

axiom B9 is just the definition of indication

Axiom `B10`: CRB is a fixed-point

Formula: `⊢ CRB φ →ₘ Rg (φ ∧ₘ CRB φ)`

Status PROVABLE

```
lemma B10 : ⊢ CRB φ →ₘ Rg (φ ∧ₘ CRB φ) := by
  intro s hCR i t hst
  constructor
  · exact hCR t (trcl_single ⟨i, hst⟩)
  · intro w htw
    exact hCR w (trcl_trans (trcl_single ⟨i, hst⟩) htw)
```

Axiom B11: CRB is regulated by rule

Formula: `(⊢ φ →ₘ Rg (ψ ∧ₘ φ)) → ⊢ (φ →ₘ CRB ψ)`

Status PROVABLE

```
lemma B11 : (⊢ φ →ₘ Rg (ψ ∧ₘ φ)) → ⊢ (φ →ₘ CRB ψ) := by
  intro hvalid s hφs
  have propagate : ∀ {x y}, φ x →
      trcl (connected : frame.World → frame.World → Prop) x y → (ψ y ∧ φ y) := by
    intro x y hφx hxy
    induction hxy with
    | base hconn =>
        rename_i x y
        rcases hconn with ⟨j, hj⟩
        have hRg : Rg (ψ ∧ₘ φ) x := (hvalid x) hφx
        exact hRg j y hj
    | step hconn hrest ih =>
        rename_i x y z
        rcases hconn with ⟨j, hj⟩
        have hRg : Rg (ψ ∧ₘ φ) x := (hvalid x) hφx
        have hy : (ψ ∧ₘ φ) y := hRg j y hj
        exact ih hy.2
  intro w hsw
  exact (propagate hφs hsw).1
```

—

Additional Failure: Condition `C4` (Cubitt-Sugden)

Cubitt and Sugden's condition `C4` states that if `A` indicates `u` to `i`, then `i` has reason to believe that `A` indicates `u` to `j`.

This formalizes shared reasoning standards—agents know they reason alike.

Formula: `Ind A i u → R i (Ind A j u)`

Counterexample: We construct a frame with worlds `s` and `t`, where:

- At `s`: φ indicates ψ to `i`
- Agent `i` cannot "see" that φ indicates ψ to `j`

Visual representation:

```
Frame Structure:
┌───┐  rel i       ┌───┐
│ s │  ──────>      │ t │
└───┘               └───┘

 ↺ rel j      (no j-edge from t)

Proposition: φ = "always true"
┌─────┬─────┬─────┐
│ w   │  s  │  t  │
├─────┼─────┼─────┤
│ φ w │  ✓  │  ✓  │
└─────┴─────┴─────┘

Target: ψ = "at world s"
┌─────┬─────┬─────┐
│ w   │  s  │  t  │
├─────┼─────┼─────┤
│ ψ w │  ✓  │  ✗  │
└─────┴─────┴─────┘
```

What holds:

- `Ind i φ ψ s` — φ indicates ψ to i at s *R i φ s (true at all accessible worlds)* `φ s → ψ s` (both true at s)

What fails:

- `R i (Ind j φ ψ) s` × — i does NOT have reason to believe that φ indicates ψ to j *At world t (accessible to i from s): Ind j φ ψ t fails* Because `φ t → ψ t` means `True → False`, which is `False` * So i cannot have reason to believe `Ind j φ ψ` (it fails at accessible world t)

Conclusion `C4` fails in this Kripke frame.

```
lemma C4_fails
    (h2a : ¬ frame.rel i s s)
    (h2b : frame.rel i s t):
    ∃ (w : frame.World) (φ ψ : frame.World → Prop), (Ind i φ ψ w ∧ ¬ R i (Ind j φ ψ)
     ↪  w) := by
  let φ := fun _ : frame.World => True
  let ψ := fun w : frame.World => w = s
  have h3 : Ind i φ ψ s := by
    constructor
    { intro w _; aesop }  -- R i True s (trivial)
    { aesop }
  have h3a : ¬ R i (Ind j φ ψ) s := by
    rw [R]; push_neg; use t
    constructor
    · exact h2b
    · intro hn; have hp : ψ t := hn.2 trivial; aesop
  exact ⟨s, φ, ψ, h3, h3a⟩
```

—

Section 7: Lewis's Theorem — Two Problematic Interpretations

Sillari's (2005, p. 391) Proposition 4.1 claims to prove Lewis's theorem. However, under local assumptions it's false; under global assumptions it's trivially true.

This section presents both interpretations and shows the problems with each.

```
section LewisTheoremCounterexamples
```

Option 1: Local Assumptions — FALSE

If conditions C1–C3 hold only at the starting world s, Lewis's theorem fails. We provide explicit counterexamples with both one agent and two agents.

```
section LewisTheoremOption1
```

Counterexample with One Agent We construct a linear chain of three worlds where the target proposition fails at the final world.

Visual representation:

```
Frame Structure (Linear Chain):

┌───┐  rel i   ┌───┐  rel i   ┌───┐
│ s │ ───────> │ u │ ───────> │ v │
└───┘          └───┘          └───┘

Proposition: φ = "not at s"
┌─────┬─────┬─────┬─────┐
│ w   │  s  │  u  │  v  │
├─────┼─────┼─────┼─────┤
│ φ w │  ✗  │  ✓  │  ✓  │
└─────┴─────┴─────┴─────┘

Target: ψ = "at u"
┌─────┬─────┬─────┬─────┐
│ w   │  s  │  u  │  v  │
├─────┼─────┼─────┼─────┤
│ ψ w │  ✗  │  ✓  │  ✗  │
└─────┴─────┴─────┴─────┘

Path to Contradiction:
1. At s: Conditions C1-C3 hold for ψ
2. Path exists: s →ⁱ u →ⁱ v  (2 steps)
3. At v: ψ v fails (v ≠ u)
4. Therefore: CRB ψ s fails
```

Conclusion: Conditions hold at s but CRB fails because v is reachable and ψ fails there.

```
lemma Lewis_fails_1i {i1 : Agent}
    (h3w : three_worlds s u v)
    (hrel : frame.rel = fun (_ : Agent) (w1 w2 : frame.World) =>
        (w1 = s ∧ w2 = u) ∨ (w1 = u ∧ w2 = v)) :
    ∃ φ, R i1 φ s ∧
      Ind i1 (R i1 φ) (R i1 (R i1 φ)) s ∧
      Ind i1 (R i1 φ) (R i1 (fun w => w = u)) s ∧
      ¬ CRB (fun w => w = u) s := by
  obtain ⟨hsu, hvu, hvs⟩ : u ≠ s ∧ v ≠ u ∧ v ≠ s := by simpa [three_worlds] using h3w
  let φ : frame.World → Prop := fun w => w ≠ s
  let ψ : frame.World → Prop := fun w => w = u

  have rel_s_u : frame.rel i1 s u := by aesop
  have rel_u_v : frame.rel i1 u v := by aesop
  have succ_s_eq_u : ∀ w, frame.rel i1 s w → w = u := by intro w hw; aesop
  have succ_u_eq_v : ∀ x, frame.rel i1 u x → x = v := by
    intro x hx
    have hx' : (u = s ∧ x = u) ∨ (u = u ∧ x = v) := by simpa [hrel] using hx
    cases hx' with
    | inl h => exact (hsu h.1).elim
    | inr h => exact h.2

  have hRphi_s : R i1 φ s := by
    intro w hw
    have : w = u := succ_s_eq_u w hw
    simpa [φ, this] using hsu

  have hR_Rphi_s : R i1 (R i1 φ) s := by
    intro w hw x hx
    have hw_u : w = u := by aesop
    have hx_v : x = v := by aesop
    simpa [φ, hx_v] using hvs

  have hInd1 : Ind i1 (R i1 φ) (R i1 (R i1 φ)) s := ⟨hR_Rphi_s, fun _ => hR_Rphi_s⟩
  have hRpsi_s : R i1 ψ s := by intro w hw; aesop
  have hInd2 : Ind i1 (R i1 φ) (R i1 ψ) s := ⟨hR_Rphi_s, fun _ => hRpsi_s⟩

  have hNotCR : ¬ CRB (fun w => w = u) s := by
    intro hCR
    have hsu_path : trcl connected s u := trcl_single ⟨i1, rel_s_u⟩
    have huv_path : trcl connected u v := trcl_single ⟨i1, rel_u_v⟩
    have hsv : trcl connected s v := trcl_trans hsu_path huv_path
    have : v = u := hCR v hsv
    exact hvu this

  exact ⟨φ, hRphi_s, hInd1, hInd2, hNotCR⟩
```

Counterexample with Two Agents This shows the problem persists with multiple agents and richer frame properties.

Visual representation:

```
Frame Structure (Two Agents):

     i₁         i₁
s ─────→ u ─────→ u  (i₁ transitive at u)
|         |
| i₂      | i₂
↓         ↓
s         v ←───── v  (reflexive at v for both)

Simplified view of reachability:
┌───┐  i₁  ┌───┐  i₂  ┌───┐
│ s │ ───> │ u │ ───> │ v │
└───┘      └───┘      └───┘
  ↺ i₂                ↺ i₁,i₂

Proposition: ψ = "not at v"
┌─────┬─────┬─────┬─────┐
│ w   │  s  │  u  │  v  │
├─────┼─────┼─────┼─────┤
│ ψ w │  ✓  │  ✓  │  ✗  │
└─────┴─────┴─────┴─────┘

Mixed-Agent Path:
s →^i₁ u →^i₂ v

What fails: ψ v = (v ≠ v) is false
```

Conclusion: Even with multiple agents, conditions at `s` don't guarantee `CRB` when different agents' relations can be chained to reach counterexample worlds.

```
lemma Lewis_fails_2i {i1 i2 : Agent}
    (h1 : three_worlds s u v)
    (h2 : two_agents i1 i2)
    (h3 : frame.rel = fun i w1 w2 =>
      (i = i1 ∧ w1 = s ∧ w2 = u) ∨
      (i = i1 ∧ w1 = u ∧ w2 = u) ∨
      (w1 = v ∧ w2 = v) ∨
      (i = i2 ∧ w1 = s ∧ w2 = s) ∨
      (i = i2 ∧ w1 = u ∧ w2 = v)) :
    ¬ ∀ (i : Agent) (φ ψ : frame.World → Prop),
        Rg φ s → Ind i (Rg φ) (Rg (Rg φ)) s → Ind i (Rg φ) (Rg ψ) s → CRB ψ s := by
  rw [two_agents] at h2
  rw [three_worlds] at h1
  let φ : frame.World → Prop := fun _ => True
  let ψ : frame.World → Prop := fun w => w ≠ v

  push_neg
  use i1, φ, ψ

  have h41 : Rg φ s := by intro i w; aesop
  have h42a : Rg (Rg φ) s := by intro i x _ w hw; aesop
  have h42 : Ind i1 (Rg φ) (Rg (Rg φ)) s := ⟨h42a i1, fun _ => h42a⟩
  have h43 : Ind i1 (Rg φ) (Rg ψ) s := ⟨h42a i1, fun _ i w => by aesop⟩

  have h44 : ¬ CRB ψ s := by
    rw [CRB]; push_neg; use v
    constructor
    · have h5a : connected s u := ⟨i1, by aesop⟩
      have h5b : connected u v := ⟨i2, by aesop⟩
      exact trcl.head h5a (trcl_single h5b)
    · aesop

  aesop

end LewisTheoremOption1
```

Option 2: Global Assumptions — TRIVIALLY TRUE

If conditions C1-C3 hold at *all* worlds (not just at the starting world), Lewis's theorem becomes provable but philosophically empty.

Why it's trivial: If Rg ψ holds at all worlds, then of course ψ holds at all reachable worlds. But this assumption is much stronger than what Lewis needs, and it makes the theorem vacuous.

Note Condition C2 is completely unused in this proof!

```
section LewisTheoremOption2
```

Lewis's Theorem (Trivial Global Version)

> If Rg φ holds at all worlds, and Ind i (Rg φ) (Rg ψ) holds at all worlds, then CRB ψ s holds.

Status TRUE but ⊠ vacuous
Observation The proof never uses condition C2. This suggests the theorem

statement is not correctly capturing Lewis's requirements.

```
lemma Lewis_s_2
    (C1 : ∀ w, Rg φ w)
    (C3 : ∀ w, Ind i (Rg φ) (Rg ψ) w) :
    CRB ψ s := by
  have hRgψ_all : ∀ w, Rg ψ w := fun w => (C3 w).2 (C1 w)
  intro v hv
  induction hv with
  | base h_edge =>
      rcases h_edge with ⟨j, hj⟩
      exact (hRgψ_all _) j _ hj
  | step _ _ ih => exact ih

end LewisTheoremOption2

end LewisTheoremCounterexamples

end Sillari
```

—

Section 8: Summary and Assessment

What We Proved

This file demonstrates four fundamental problems with Sillari's approach:

1. **`B3 (A1)` fails** — Lewis's principle of detachment does not hold
2. **`C4` fails** — Shared standards axiom does not hold
3. **Lewis's theorem fails locally** — Counterexamples with 1 and 2 agents
4. **Lewis's theorem is trivial globally** — Proof doesn't use `C2`, reduces to tautology

The Root Cause

Modal logic tracks **that** agents believe propositions, not **how** beliefs are justified. Lewis's notion of indication requires evidential dependence—the reason for the conclusion must depend on the reason for the premise.

Sillari's definition:

```
Ind i φ ψ := R i φ ∧ (φ → ψ)
```

is a conjunction that cannot capture this dependence. The reason for ψ might come from a completely different source than the reason for φ.

Philosophical Lessons

1. **Framework choice matters** — Modal logic is the wrong tool for Lewis's theory
2. **Formal verification catches subtle errors** — The proof gap in Proposition 4.1 is now explicit
3. **Definitions encode commitments** — Small definitional choices have large consequences
4. **Negative results are valuable** — They clarify the problem space and motivate better solutions

The Problem with Modal Box

The modal operator $\Box_i$ (our `R i`) treats "reason to believe" as a black box:

- `R i φ` means "φ holds at all worlds `i` considers possible"
- But it doesn't tell us **why** `i` has this reason
- It doesn't track the **evidential structure** of reasoning

When we try to compose reasons (`A` gives reason for `B`, `B` gives reason for `C`), modal logic cannot verify that the reason for `C` actually depends on the reason for `A` through `B`.

The Solution: Justification Logic

Vromen (2024) uses justification logic with explicit reason terms:

```
R i φ := ∃r. rb r i φ
```

(There exists an explicit reason `r` such that `r` is a reason for `i` to believe `φ`)

```
Ind A i φ := R i (A → φ)
```

(`i` has reason to believe the implication from `A` to `φ`)

This captures ”thereby”: when combined with a reason for `A` via the application rule `t:(A → φ) → s:A → (t·s):φ`, the resulting reason for `φ` explicitly contains the reason for `A`.

Key insight The compositional structure of reason terms enforces evidential

dependence at the syntactic level.

Achievements of This File

- Machine-verified counterexamples to Sillari’s key axioms
- Explicit demonstration of where modal logic fails
- Clear explanation of the ”thereby” problem
- Motivation for justification logic approach

Recommendation

- For what **doesn’t work** ⟶ `this file`
- For the **structure** of Lewis’s argument ⟶ `Cubitt_Sugden_baseline.lean`
- For the correct **foundations** ⟶ `Vromen_justification_logic.lean`