

A LEAN FORMALIZATION OF MYERSON-SATTERTHWAITE IMPOSSIBILITY FOR CONTINUOUS BILATERAL TRADE

ARTHUR FREITAS RAMOS, DAVID BARROS HULAK, AND RUY J. G. B. DE QUEIROZ

ABSTRACT. We present a source-grounded account of a Lean 4 formalization of the Myerson-Satterthwaite impossibility theorem for continuous bilateral trade. Buyer values and seller costs are independent, with atomless probability measures supported on compact real intervals whose interiors overlap. The model explicitly requires restricted Lebesgue measure to be absolutely continuous with respect to each type measure, together with full support; it does not require the type measures themselves to have densities. Under allocation and transfer measurability and integrability conditions, the registered theorem excludes the simultaneous satisfaction of almost-sure ex post efficiency, Bayesian incentive compatibility at every admissible type and report, interim individual rationality, and weak ex ante budget balance. The proof derives envelope identities from incentive inequalities, identifies the efficient allocation almost everywhere, and uses a layer-cake representation to show that the required information rents strictly exceed efficient surplus. We explain the analytic steps, their Lean declarations, and the verification evidence for the exact Palomar-registered source revision. This is an exposition of a classical result and an existing formal artifact, without a claim of mathematical novelty or formalization priority.

1. MATHEMATICAL CONTEXT AND SCOPE

Myerson and Satterthwaite's bilateral-trade theorem concerns a buyer and a seller who privately know their respective valuations for a single indivisible object. Their 1983 paper characterizes incentive-compatible, individually rational trading mechanisms and establishes an impossibility of ex post efficient trade without outside subsidies under overlapping independent type distributions [2]. The present paper explains the continuous-type impossibility result implemented in the Lean repository [4], rather than the original paper's complete characterization or its optimal-mechanism constructions.

The formal artifact uses Lean 4 [1] and Mathlib [5]. Its central declaration is `MS.Palomar.myersonSatterthwaite`. It is registered as Palomar entry PALOMAR-2026-09-26-000002, version 1 [3]. All implementation descriptions in this paper refer to the registered commit, not to an evolving default branch.

The main mathematical obstruction is an accounting inequality. Incentive compatibility forces interim utilities to accumulate information rents. Efficiency fixes the associated interim trade probabilities. When type intervals overlap, the sum of the resulting rent terms is strictly greater than the expected efficient surplus. Individual rationality prevents the mechanism from offsetting this excess by charging negative utility to its boundary types, while weak ex ante budget balance prevents financing the excess with an expected outside subsidy.

The formal proof makes explicit several conditions sometimes suppressed in an informal statement: the direction of absolute continuity between measures, sectionwise measurability

Date: October 1, 2026.

2020 Mathematics Subject Classification. Primary 91B26; Secondary 68V20.

Key words and phrases. Myerson-Satterthwaite theorem, bilateral trade, Bayesian incentive compatibility, envelope theorem, budget balance, Lean 4, Mathlib.

Author ORCID identifiers: Arthur Freitas Ramos, 0009-0003-3568-0325; David Barros Hulak, 0009-0002-8056-1774; Ruy J. G. B. de Queiroz, 0000-0003-1482-0977.

Copyright 2026 the authors. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0).

and integrability, the almost-everywhere meaning of efficiency, and the quantification of incentive constraints at all types, including endpoints. These details determine what the checked theorem says. Our contribution here is an auditable exposition of that statement and its proof architecture. We do not infer priority from the existence of a registry entry or claim that the development covers every version of bilateral-trade impossibility.

2. THE EXACT MODEL

2.1. Types and distributions. The buyer's value is $v \in I_B = [a_1, b_1]$ and the seller's cost is $c \in I_S = [a_2, b_2]$. The source assumes

$$(1) \quad a_1 < b_1, \quad a_2 < b_2, \quad a_2 < b_1, \quad a_1 < b_2.$$

Equivalently, both intervals are nondegenerate and

$$(2) \quad \ell = \max(a_1, a_2) < r = \min(b_1, b_2).$$

The type measures μ and ν are probability measures on $\mathbb{R}$. Independence is represented by the product measure $\pi = \mu \otimes \nu$; it is built into the definitions of interim expectations and not a separate correlation parameter. Both measures are atomless, and their topological supports satisfy

$$(3) \quad \text{supp } \mu \subseteq I_B, \quad \text{supp } \nu \subseteq I_S.$$

Writing λ for Lebesgue measure, the exact domination assumptions are

$$(4) \quad \lambda|_{I_B} \ll \mu, \quad \lambda|_{I_S} \ll \nu.$$

Thus a μ -null set is Lebesgue-null inside I_B , and a ν -null set is Lebesgue-null inside I_S . The direction in (4) is essential. It does not assert $\mu \ll \lambda$ or $\nu \ll \lambda$.

The model also explicitly requires $\mu(O) > 0$ for every open set O meeting (a_1, b_1) , and similarly $\nu(O) > 0$ whenever O meets (a_2, b_2) . These full-support clauses are retained in the formal model and are used directly in the strict-positivity argument. As a mathematical observation, (4) already implies these clauses: any such open intersection has positive Lebesgue measure. Nevertheless, ordinary topological full support alone is not a substitute for (4) in the registered statement.

In particular, the theorem is not restricted to type measures possessing densities. An atomless mixture of a positive-weight uniform distribution on an interval and a singular probability measure on the same interval can satisfy these conditions without being absolutely continuous with respect to Lebesgue measure. This observation illustrates the assumptions; the repository does not supply a formal construction of that example.

2.2. Allocation, transfers, and interim quantities. The direct mechanism consists of real-valued functions $q, t_b, t_c : \mathbb{R}^2 \rightarrow \mathbb{R}$. At a report pair (v, c) , $q(v, c)$ is the probability of trade, $t_b(v, c)$ is the payment collected from the buyer, and $t_c(v, c)$ is the payment delivered to the seller. Transfers may be signed. On $I_B \times I_S$, the source requires $0 \leq q \leq 1$ pointwise. It assumes that q is almost-everywhere strongly measurable under π , and that each buyer section $q(v, \cdot)$ for $v \in I_B$ and each seller section $q(\cdot, c)$ for $c \in I_S$ is almost-everywhere strongly measurable under the respective opposing type measure.

Both transfers are integrable under π . In addition, every relevant interim section is integrable: $t_b(v, \cdot)$ under ν for each $v \in I_B$, and $t_c(\cdot, c)$ under μ for each $c \in I_S$. These section hypotheses are stronger than merely having section integrability for almost every type. They ensure that the all-type incentive and participation inequalities use genuine finite integrals. Allocation sections are integrable by their measurability, boundedness, and the probability-measure assumptions.

The four global definitions in `MS/Defs.lean` are the following Bochner integrals, with scalar values:

$$(5) \quad Q(v) = \int q(v, c) d\nu(c), \quad R(c) = \int q(v, c) d\mu(v),$$

$$(6) \quad U(v) = \int (vq(v, c) - t_b(v, c)) d\nu(c), \quad W(c) = \int (t_c(v, c) - cq(v, c)) d\mu(v).$$

Their names are `interimQ`, `interimR`, `interimU`, and `interimW`. For an implementation instance `D:MSData`, the corresponding derived definitions are `D.Q`, `D.R`, `D.U`, and `D.W`.

2.3. Efficiency, incentives, participation, and budget. Ex post efficiency is required almost surely on the two strict-order regions:

$$(7) \quad q = 1 \quad \pi|_{\{c < v\}}\text{-a.e.}, \quad q = 0 \quad \pi|_{\{v < c\}}\text{-a.e.}$$

The source does not impose pointwise efficiency at every report pair, nor does it specify a tie rule. Atomlessness makes the diagonal $\{(v, c) : v = c\}$ π -null.

Bayesian incentive compatibility (BIC) is expressed as

$$(8) \quad U(v) \geq U(v') + (v - v')Q(v') \quad (v, v' \in I_B),$$

$$(9) \quad W(c) \geq W(c') + (c' - c)R(c') \quad (c, c' \in I_S).$$

The right side of (8) is the expected utility of true type v reporting v' ; (9) has the analogous interpretation for the seller. The quantifiers range over every true type and report in the closed intervals, with no ordering restriction between them. These are unilateral-deviation inequalities, not inequalities restricted to upward or downward misreports. Dominant-strategy incentive compatibility is not assumed.

Interim individual rationality (IR) and weak ex ante budget balance are

$$(10) \quad U(v) \geq 0 \quad (v \in I_B), \quad W(c) \geq 0 \quad (c \in I_S),$$

$$(11) \quad \mathbb{E}_\pi[t_c] \leq \mathbb{E}_\pi[t_b].$$

The seller's utility is measured relative to retaining the object. Equation (11) requires only nonnegative expected net receipts; neither equality of transfers nor a pointwise no-deficit condition is part of the hypothesis.

Theorem 2.1 (Registered continuous bilateral-trade impossibility). *There are no intervals, probability measures, allocation, and transfers satisfying all conditions in Section 2.*

Precisely, `MS/Palomar.lean` packages those conditions as `MS.IsModel`. The declaration `MS.Palomar.myersonSatterthwaite` takes $a_1, b_1, a_2, b_2, \mu, \nu, q, t_b, t_c$ and a proof of this proposition, and returns `False`. Its proof constructs an `MSData` instance and applies `MSData.MS_impossible`.

3. FROM INCENTIVE INEQUALITIES TO ENVELOPE IDENTITIES

The analytic work begins in `MS/Envelope.lean`. It derives the envelope formulas rather than assuming differentiability of utility or allocation.

Adding the buyer's incentive inequalities for (v, v') and (v', v) gives

$$(v' - v)(Q(v') - Q(v)) \geq 0.$$

Consequently Q is nondecreasing on I_B . The analogous argument makes R nonincreasing on I_S . These are `MSData.Q_mono` and `MSData.R_anti`. The allocation bounds also give $0 \leq Q, R \leq 1$ at all admissible types.

For $v < v'$, the two buyer constraints yield

$$(12) \quad (v' - v)Q(v) \leq U(v') - U(v) \leq (v' - v)Q(v').$$

In particular, $|U(v') - U(v)| \leq |v' - v|$. The seller's constraints similarly give $|W(c') - W(c)| \leq |c' - c|$. The source proves these bounds as `MSData.U_lip` and `MSData.W_lip`, then obtains absolute continuity of the utilities on their compact intervals.

The proof uses a general, private monotone-subgradient argument. If g is nondecreasing and $f(x) \geq f(y) + (x - y)g(y)$, the difference quotient at a continuity point x of g is squeezed between nearby values of g . It follows that $f'(x) = g(x)$. A monotone function has at most countably many points of discontinuity, so this derivative equality holds Lebesgue-almost everywhere in the interior. Applying the fundamental theorem of calculus for absolutely continuous functions gives the buyer formula; the seller formula follows by using $-R$ as the nondecreasing subgradient.

Proposition 3.1 (Envelope identities). *For every $v \in I_B$ and $c \in I_S$,*

$$(13) \quad U(v) = U(a_1) + \int_{a_1}^v Q(t) dt,$$

$$(14) \quad W(c) = W(b_2) + \int_c^{b_2} R(t) dt.$$

The corresponding declarations are `MSData.U_envelope` and `MSData.W_envelope`. The endpoint constants are not normalized to zero. Their signs remain available for the final use of participation.

Define the expected rent terms

$$(15) \quad B = \int_{a_1}^{b_1} Q(t) \mu([t, \infty)) dt,$$

$$(16) \quad C = \int_{a_2}^{b_2} R(t) \nu((-\infty, t]) dt.$$

Integrating (13) and (14) against the type measures and interchanging the resulting integrals gives

$$(17) \quad \mathbb{E}_\mu[U] = U(a_1) + B, \quad \mathbb{E}_\nu[W] = W(b_2) + C.$$

These are `MSData.EU_eq` and `MSData.EW_eq`. Their formal proofs establish the relevant joint integrability before using Fubini: bounded interim allocations are integrated over finite intervals, and the Lipschitz utilities are integrable on compact supports.

4. EFFICIENCY AND THE STRICT RENT GAP

4.1. Changing the almost-everywhere measure. In `MS/Efficiency.lean`, `MSData.diag_null` proves the nullity of ties by examining singleton sections under the atomless seller measure. Combining this result with (7) gives

$$(18) \quad q(v, c) = \mathbf{1}_{\{c < v\}} \quad \pi\text{-a.e.}$$

This is `MSData.q_ae_ind`.

Product-measure section arguments first show $Q(t) = \nu((-\infty, t))$ for μ -almost every t , and $R(t) = \mu((t, \infty))$ for ν -almost every t . The rent terms, however, are integrals with respect to Lebesgue measure in t . This is exactly where (4) enters: its direction transfers these equalities to Lebesgue-almost every t on the corresponding intervals. The declarations `MSData.Q_ae` and `MSData.R_ae` state these latter equalities. Replacing (4) by the reverse absolute-continuity assumption would not justify this step.

4.2. A common tail product. Define

$$(19) \quad H(t) = \nu((-\infty, t)) \mu((t, \infty)).$$

This is `MSData.H`. It is measurable and satisfies $0 \leq H \leq 1$. Atomlessness lets the proof interchange strict and weak threshold inequalities. Therefore (15) and (16) become

$$(20) \quad B = \int_{a_1}^{b_1} H(t) dt, \quad C = \int_{a_2}^{b_2} H(t) dt,$$

as proved by `MSData.B_eq` and `MSData.C_eq`.

Let

$$(21) \quad S = \mathbb{E}_\pi[(v - c)_+], \quad (x)_+ = \max(x, 0).$$

The definition is `MSData.S`. For types in the support rectangle,

$$(v - c)_+ = \int_{a_2}^{b_1} \mathbf{1}_{\{c < t < v\}} dt.$$

The formal proof of `MSData.S_layercake` applies Fubini to this bounded indicator on a finite product-measure space and uses independence to factor its inner expectation. The resulting layer-cake formula is

$$(22) \quad S = \int_{a_2}^{b_1} H(t) dt.$$

The efficient allocation also gives

$$(23) \quad \mathbb{E}_\pi[(v - c)q(v, c)] = S,$$

which is `MSData.eff_payoff`.

Proposition 4.1 (Strict rent gap). *With B, C, S as above and ℓ, r from (2),*

$$(24) \quad B + C - S = \int_{\ell}^r H(t) dt > 0.$$

Proof. Support containment gives $H(t) = 0$ for $t \leq a_2$ and for $t \geq b_1$. Thus the first integral in (20) can start at ℓ , and the second can end at r . Splitting the integrals at these endpoints shows

$$\int_{\ell}^{b_1} H + \int_{a_2}^r H - \int_{a_2}^{b_1} H = \int_{\ell}^r H.$$

For $t \in (\ell, r)$, full support assigns positive seller probability to costs strictly below t and positive buyer probability to values strictly above t . Hence $H(t) > 0$ throughout that nonempty open interval. Since H is nonnegative and integrable, its integral there is strictly positive. $\square$

The positivity claim is `MSData.H_pos`; the exported strict inequality is `MSData.BC_gt_S`. The equality in (24) is established inside the latter proof as a local intermediate fact. No density, hazard rate, or division by a density is needed in this argument.

5. THE ACCOUNTING CONTRADICTION

Proof of Theorem 2.1. Product integrability and Fubini identify the expected interim utilities with the expectations of the realized utilities in (6). Adding the buyer and seller identities, and using efficiency, gives

$$(25) \quad \begin{aligned} \mathbb{E}_\mu[U] + \mathbb{E}_\nu[W] &= \mathbb{E}_\pi[(v - c)q] - (\mathbb{E}_\pi[t_b] - \mathbb{E}_\pi[t_c]) \\ &= S - (\mathbb{E}_\pi[t_b] - \mathbb{E}_\pi[t_c]) \leq S. \end{aligned}$$

The last inequality is weak ex ante budget balance. On the other hand, (17) and endpoint IR imply

$$\mathbb{E}_\mu[U] + \mathbb{E}_\nu[W] = U(a_1) + W(b_2) + B + C \geq B + C > S,$$

contradicting (25). $\square$

This is the organization of `MSData.MS_impossible` in `MS/Main.lean`. Before the accounting step, the source derives integrability of vq, cq , the realized utilities, and the transfer difference. Compact support bounds the type coordinates, and the allocation range bounds q . The final contradiction uses only $U(a_1) \geq 0$ and $W(b_2) \geq 0$ from the stronger all-type participation hypotheses. The registered theorem nevertheless retains all-type interim IR; a separate minimized-assumption theorem is not provided.

The same calculation displays the required expected subsidy for any efficient BIC and interim-IR candidate if budget balance is dropped:

$$(26) \quad \mathbb{E}_\pi[t_c] - \mathbb{E}_\pi[t_b] = U(a_1) + W(b_2) + B + C - S \geq \int_\ell^r H(t) dt > 0.$$

This is an explanatory rearrangement of the proved identities, not a separately exported subsidy theorem or a formal construction attaining the lower bound.

6. FORMAL ORGANIZATION AND SEMANTIC BOUNDARIES

The development separates mathematical data from its public registry interface. `MSData` is a structure containing the endpoints, measures, rules, and hypotheses. The internal lemmas can therefore use a single argument D without repeatedly unpacking the entire model. The public proposition `MS.IsModel` expresses the same data as a conjunction, while `MS.Overlap` isolates the four interval inequalities. The public impossibility theorem is a wrapper around the internal contradiction.

The implementation has five substantive modules, totaling 1,441 lines at the registered revision. `MS/Defs.lean` supplies the model and interim definitions. `MS/Envelope.lean` derives monotonicity, Lipschitz continuity, envelope formulas, and expected rents. `MS/Efficiency.lean` supplies the null-diagonal argument, efficient interim allocations, layer-cake formula, and strict rent gap. `MS/Main.lean` performs the final accounting, and `MS/Palomar.lean` exposes the selected theorem. Appendix A gives a concise declaration map.

6.1. Why integrability is explicit. Mathlib’s Bochner integral is a total operation, so an expression involving an integral alone does not establish the analytic conditions expected in economic notation. The product and section hypotheses prevent the model from relying on the totalized integral’s value for a nonintegrable transfer. Bounded measurable allocation sections provide finite trade probabilities; compact support then controls the value and cost terms. The source repeatedly supplies integrability proofs when applying linearity or Fubini, rather than treating these operations as unrestricted algebraic rewrites.

Another important boundary is the coexistence of two notions of almost-everywhere equality. Efficiency is expressed under π , whereas the envelope and rent formulas use Lebesgue interval integrals. The domination assumptions connect these notions only after taking sections, as explained in Section 4. Full support supplies strict positivity, while atomlessness removes ties and the distinction between open and closed threshold sets. These are distinct roles, even though some distribution hypotheses logically imply others.

6.2. What the theorem does not formalize. The registered result concerns independent atomless types on finite, overlapping intervals and real quasi-linear utilities. It does not formalize correlated types, atomic or finite-discrete type models, unbounded supports, risk aversion, multiple objects, or multiple traders. It also does not establish the revelation principle, characterize all implementable allocations, construct an optimal inefficient mechanism, or prove attainability of the subsidy bound (26).

The impossibility applies to randomized allocations because q may take any value in $[0, 1]$ before efficiency is imposed. Its efficiency requirement is almost sure, its incentive and participation constraints are interim and all-type, and its budget constraint is ex ante and weak. Pointwise efficient mechanisms or pointwise balanced mechanisms satisfying the remaining hypotheses are included as stronger special cases; the development does not independently formalize those implications.

7. ARTIFACT PROVENANCE AND VERIFICATION EVIDENCE

7.1. The registered revision. The Palomar record [3] identifies the repository `Arthur742Ramos/myerson-satterthwaite-lean`, with commit

`f4dae9744e26a499679e4b8a7aeb9d82dd3f11ee`.

It records Lean v4.35.0-rc2 and Mathlib revision

065356127b1dc0016f66b7283ce0ce2c4055aa55.

The committed `lake-manifest.json` pins the dependencies. The source repository and its metadata attribute the formalization to Arthur Freitas Ramos and license the code under BSD-3-Clause. This manuscript is distributed under CC BY 4.0; the two licenses govern different artifacts.

`Challenge.lean` declares the selected statement and six definitions using only Mathlib imports. It contains one deliberate statement `sorry`, used as the target specification for comparison. `Solution.lean` imports the implementation, whose five substantive modules contain no proof holes. The permitted axioms listed in the registered comparator configuration are `propext`, `Classical.choice`, and `Quot.sound`. The deliberate challenge hole is not an implementation proof and is not presented as one.

7.2. Historical checks and present inspection. The registry’s archived mechanical report has status `pass`, stage `complete`, and verification timestamp 2026-09-26 02:28:25 UTC. It records a successful solution build and Comparator acceptance by the Lean default kernel, `nanoda`, and `con-ron`. The associated workflow run is linked in the record. These are historical verification results for the pinned source, not newly executed checks performed for this paper.

For the present exposition, the pinned source files and archived report were inspected. The downloaded challenge and solution SHA-256 values match those in the registered record; the archived report’s SHA-256 value also matches its recorded digest. A fresh Lean compilation or Comparator run was not performed during manuscript preparation. The present audit supports the account of the source and its archived evidence; it does not turn that archival evidence into a claim of fresh execution.

The record’s automated review has outcome `neutral` and specifically warns that the `MS.IsModel` docstring mentions positive densities although the definition has the domination conditions (4). This paper follows the actual definition. The source’s README and formalization metadata also say that no density requirement is imposed. Registration, mechanical checking, and review are distinct evidence; none alone proves that an informal paraphrase faithfully states the formal theorem.

7.3. AI assistance. The repository’s `formalization.yaml` reports AI assistance by `gpt-6-1 una` through the Codex CLI for historical proof engineering. OpenAI GPT-6.1 assisted with drafting this manuscript, inspecting the registered source, organizing the proof exposition, and editorial and consistency checks. AI assistance is not authorship, and the manuscript authors are responsible for its claims. The registered mechanical verification evidence concerns the formal artifact; it does not certify this prose or imply that the authors independently checked every proof step by hand.

8. CONCLUSION

The registered formalization proves a continuous bilateral-trade impossibility under an explicit measure-theoretic model. Its proof connects incentive inequalities to absolutely continuous utility envelopes, then connects almost-sure efficiency to Lebesgue rent integrals through the domination of Lebesgue measure by the type measures. A positive overlap integral measures the excess of the forced information rents over efficient surplus, yielding a contradiction with participation and expected budget balance. The source and its archived verification evidence make this argument inspectable at a fixed revision, while the distinction between the formal statement and its informal presentation remains part of the verification boundary.

APPENDIX A. GUIDE TO THE CHECKED DECLARATIONS

All names below have prefix `MSData`, unless another prefix is shown. The links point to files at the registered commit.

Model and interface.: `MSData` stores all hypotheses. `MS.IsModel` is the public conjunction; `MS.Palomar.myersonSatterthwaite` returns `False` from it.

Envelope analysis.: `Q_mono`, `R_anti`, `U_lip`, `W_lip` derive monotonicity and Lipschitz bounds. `U_envelope`, `W_envelope`, `EU_eq`, and `EW_eq` give (13)–(17).

Efficiency and surplus.: `diag_null`, `q_ae_ind`, `Q_ae`, `R_ae` identify efficient allocations and interim probabilities. `B_eq`, `C_eq`, `S_layercake`, and `eff_payoff` give (20)–(23).

Strictness and contradiction.: `H_pos`, `BC_gt_S` establish the positive rent gap. `MS_impossible` combines it with expected-utility accounting and endpoint IR.

APPENDIX B. A UNIFORM-DISTRIBUTION ILLUSTRATION

For independent uniform types on $[0, 1]$, one has $H(t) = t(1-t)$ on that interval. Therefore

$$S = \int_0^1 t(1-t) dt = \frac{1}{6}, \quad B = C = \frac{1}{6}, \quad B + C - S = \frac{1}{6}.$$

The expected rent requirements are twice the efficient surplus before adding nonnegative endpoint utilities. Any efficient BIC and interim-IR candidate under these assumptions must therefore have expected subsidy at least $1/6$ by (26). This calculation illustrates the general proof; it is not an additional Lean theorem in the repository, and it does not assert the construction of an attaining mechanism.

REFERENCES

1. Leonardo de Moura and Sebastian Ullrich, *The Lean 4 theorem prover and programming language*, Automated Deduction—CADE 28, Lecture Notes in Computer Science, vol. 12699, Springer, 2021, https://doi.org/10.1007/978-3-030-79876-5_37, pp. 625–635.
2. Roger B. Myerson and Mark A. Satterthwaite, *Efficient mechanisms for bilateral trading*, Journal of Economic Theory **29** (1983), no. 2, 265–281, [https://doi.org/10.1016/0022-0531\(83\)90048-0](https://doi.org/10.1016/0022-0531(83)90048-0).
3. Palomar Registry, *Myerson-Satterthwaite impossibility theorem (continuous bilateral-trade version)*, Registered formalization, PALOMAR-2026-09-26-000002, version 1, 2026, Registered September 26, 2026. <https://palomar-registry.org/entry?id=PALOMAR-2026-09-26-000002&version=1>.
4. Arthur Freitas Ramos, *Myerson-Satterthwaite theorem (Lean 4 formalization)*, Source repository, registered revision, 2026, Commit `f4dae9744e26a499679e4b8a7aeb9d82dd3f11ee`. <https://github.com/Arthur742Ramos/myerson-satterthwaite-lean/tree/f4dae9744e26a499679e4b8a7aeb9d82dd3f11ee>.
5. The mathlib Community, *The Lean mathematical library*, Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, ACM, 2020, <https://doi.org/10.1145/3372885.3373824>, pp. 367–381.