FORMAL METHODS / DECISION SYSTEMS

An exact, guarded model of the 2026 federal ordinary-income rate schedules

Source-bound Single and Married Filing Jointly illustrations in Lean, with explicit rounding and refusal boundaries

Revanth Matha · Matha Labs Research whitepaper 2.0 | Revised 16 November 2026

CENTRAL CONTRIBUTION

A source- and year-bound acceptance gate for reviewed ordinary taxable-income examples in two filing-status schedules.

The implemented output is a 2026 ordinary-income schedule illustration for reviewed Single or Married Filing Jointly facts. It is not Form 1040 liability, an IRS Tax Table or worksheet implementation, a taxable-income derivation, a filing-status determination or individualized tax advice.

Original publication: 15 November 2026 at 3:12:52 PM PST. This edition expands the paper; it does not claim a new trained model or a new clinical, municipal or financial deployment.

Contents

Abstract 3

Contributions 3

1. A small calculation with a large applicability boundary 3

2. Source hierarchy and tax-year identity 4

3. The represented schedules and omitted classifications 4

4. Reviewed facts, certificates and refusal 5

5. Deriving the positive-increment formula 6

6. Why a rate illustration is not Form 1040 liability 7

7. Worked amounts and cent boundaries 7

8. Fourteen formal statements and their limits 8

9. What the thirty fixtures actually test 9

10. Competing representations and related work 9

11. A proposed independent evaluation and drift policy 10

12. Reproduction, limitations and the next module 11

References 11

Abstract

Tax-rate arithmetic is compact enough to formalize while still exposing important problems of legal scope, source versioning and numerical representation. This paper studies a bounded implementation of the 2026 federal ordinary taxable-income schedules for Single and Married Filing Jointly. The six thresholds for each status are transcribed from Revenue Procedure 2025-32 and checked against additional IRS publications. A certificate gate requires matching source, revision, statute identifier, year and supported status, together with externally reviewed ordinary-income and scope attestations. The evaluator uses natural-number cents, accumulates an exact integer numerator in hundredths of a cent and floors once to cents under a stated research convention. Fourteen named Lean theorems establish acceptance preconditions, monotonicity, bounds, a zero-input result and a one-cent increase bound. Thirty synthetic fixtures cover schedule bases, worked values, rounding boundaries and refusal conditions. These results concern the encoded schedule illustration, not a complete tax return. The model neither derives taxable income nor determines filing-status eligibility, credits, alternative minimum tax, preferential rates or return-specific computation methods. Source-string equality also does not authenticate the publication or taxpayer facts. We explain the source hierarchy, derive the arithmetic, connect each claim to its evidence and compare alternative implementations. A proposed independent evaluation would test exact amounts and abstention across frozen source versions and reviewed cases. No tax-preparation accuracy benchmark, taxpayer dataset or filing integration is claimed.

Keywords: Lean 4, federal income tax, 2026 rate schedules, exact arithmetic, formal verification, policy versioning, research illustration

Contributions

1. A source- and year-bound acceptance gate for reviewed ordinary taxable-income examples in two filing-status schedules.

2. An exact positive-increment rate formula with an explicit single final floor to cents and no binary floating point.

3. Fourteen checked theorems about the encoded guard and arithmetic, plus thirty reproducible synthetic fixtures.

4. A source hierarchy and proposed evaluation protocol separating schedule arithmetic from complete return liability.

1. A small calculation with a large applicability boundary

A marginal-rate schedule looks like a short arithmetic function. Turning it into a useful software artifact nevertheless requires answers to several different questions: which year and filing status apply, what amount counts as taxable income, whether special computations are required, and how the numeric output should be represented. Treating those questions as one undifferentiated calculation makes a precise-looking number easy to misinterpret. This project starts with an already reviewed ordinary taxable-income amount and makes the remaining research boundary visible.

The question is whether a pure evaluator can bind a narrowly specified rule to reviewed inputs and expose properties that hold for every represented amount. The published implementation therefore separates a Boolean applicability guard from an arithmetic illustration. A rejected guard returns none rather than a number. The arithmetic remains directly callable for testing, but that lower-level function is not the guarded user-facing result. Reviewers should inspect both layers and avoid equating a total arithmetic function with a complete legal decision. [12]

This approach is intentionally useful for research before it is useful for filing. It permits exact boundary examples, universal arithmetic properties and controlled mutations of source or scope fields. It also makes omissions explicit: no wages are parsed, deductions derived, payments scheduled or returns transmitted. A future ingestion model could propose records, but it would need its own evidence and evaluation before those proposals could become the reviewed facts assumed here.

2. Source hierarchy and tax-year identity

The primary numerical source is Revenue Procedure 2025-32, section 4.01: Table 1 supplies the joint schedule and Table 3 the unmarried schedule used for Single. The implementation also records the statutory context as 26-USC-1-j. Section 1 contains historical tables and later provisions, so copying the first dollar amounts visible in the statute would be an inadequate annual-source selection procedure. The operative research bundle pairs statutory context with the identified inflation-adjusted publication and tax year. [1][3]

Publication 505 and the 2026 Form 1040-ES package provide official cross-checks for the 2026 schedules. They also place those schedules within an estimated-tax workflow that includes further computations. We use them as corroborating sources and as evidence that a rate-schedule operation occupies only one part of a larger process. The current IRS summary of 2026 marginal rates agrees with the two threshold vectors. None of these cross-checks turns our final floor-to-cents convention into an IRS filing instruction. [2][7][9]

The model revision is rev-proc-2025-32-section-4.01-taxyear-2026-v1. Its sourceId is the exact IRS PDF URL, and the gate requires facts for 2026 and a matching certificate year. These are compiled identity checks. They do not download the source, verify an agency signature or determine whether a later legal change supersedes it. This paper's November 16 source review and version 2.0 prose are therefore distinct from the pinned executable revision.

3. The represented schedules and omitted classifications

The rates are 10%, 12%, 22%, 24%, 32%, 35% and 37%. The table shows where each higher marginal rate begins; it is not a statement that all income above a threshold is taxed at one rate. The code converts the official dollar thresholds to cents. The first Single threshold is therefore 1,240,000 cents, while the first joint threshold is 2,480,000 cents. These unit declarations are part of the numerical specification, not formatting details. [1]

Single and marriedJoint are the only accepted status constructors. The unsupported constructor deliberately returns no illustration through evaluate, even though the internal thresholds function has a total fallback value. That fallback is an implementation convenience for a total function; unsupported_status_refused prevents it

from becoming an accepted result. Eligibility for Single or joint filing must be established upstream. The definition of marital status has statutory rules and exceptions that are not encoded by selecting an enumeration member. [5][12]

The input amount is taxable income rather than gross receipts, wages or adjusted gross income. Section 63 supplies substantive definitions involving deductions; this project does not implement that derivation. Applying another standard deduction inside the illustrated schedule would double-count a transformation if the supplied value already was taxable income. Equally, an unknown or negative amount cannot safely be coerced to zero simply because the Lean representation uses natural numbers. An external adapter must reject or resolve such cases explicitly. [4]

Table 1. Fixed 2026 schedule thresholds in dollars

Marginal band begins above	Single	Married Filing Jointly
First band: 10%	0	0
12%	$12,400	$24,800
22%	$50,400	$100,800
24%	$105,700	$211,400
32%	$201,775	$403,550
35%	$256,225	$512,450
37%	$640,600	$768,700

Thresholds from Revenue Procedure 2025-32, section 4.01, Tables 3 and 1 respectively. Code stores each dollar amount multiplied by 100. These are schedule constants, not return-preparation instructions. [1]

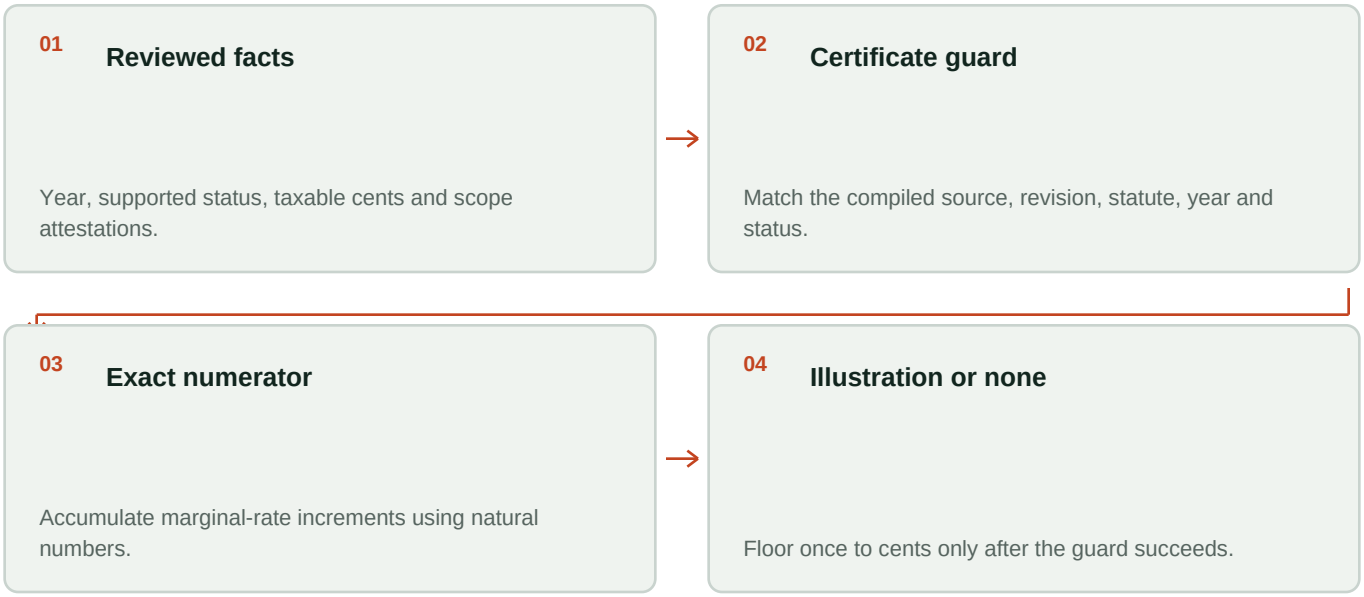
4. Reviewed facts, certificates and refusal

ReviewedFacts contains taxYear, status, ordinaryTaxableIncomeCents, reviewed, ordinaryOnly, hasCredits, requiresAMT and specialComputation. Certificate contains sourceId, revision, statuteId, taxYear and status. Conditions requires exact source, revision and statute agreement; facts for 2026; matching certificate year and status; a supported status; reviewed and ordinaryOnly set true; and the three exclusion flags set false. verify decides this finite conjunction, and evaluate calls the arithmetic only after success. [12]

A false review flag, a certificate for another year or an unsupported status produces none. This release supplies no structured rejection reason, so an interface cannot truthfully claim that evaluate itself explains which condition failed. A caller can inspect the predicates or build a separate explanation layer, but that layer would need its own consistency checks. The difference from the permitting model's named review reasons is deliberate evidence about what each repository actually implements.

The exclusion flags represent external judgments, not automatic detection. For example, section 55 establishes an AMT computation distinct from the ordinary schedule, but the program merely refuses a case marked requiresAMT. It neither computes the minimum tax nor proves that a false flag is appropriate. Similarly, ordinaryOnly does not inspect transaction categories or establish the absence of qualified dividends, capital gains, foreign-income issues or another special computation. [2][6]

Figure 1. The implemented guarded calculation



The actual implementation contains the guard and arithmetic stages. Determining taxable income and reviewing scope precede this boundary; complete return preparation is unimplemented. [12]

5. Deriving the positive-increment formula

Let x be taxable income in cents, and let t1 through t6 be the chosen schedule's thresholds in cents. Define each excess as natural-number subtraction, which is zero when income is below the threshold. The base term charges ten percentage points on x. Additional terms introduce only the change in rate: two, ten, two, eight, three and two percentage points. Their cumulative rates are the seven published marginal rates. For the ordered official thresholds, this is algebraically equivalent to summing the occupied bands.

The numerator N is measured in hundredths of a cent, because a percentage-point coefficient multiplying cents has denominator one hundred. The formula retains that denominator throughout accumulation. illustrationCents then performs natural-number division by 100 once. This produces a final floor to cents. It avoids binary floating-point error and avoids repeated rounding of band contributions, but it is a chosen research convention rather than a claim about the correct rounding or table method for a filed return.

The formula also makes some proof obligations simple. Every excess term is monotone in x, and every coefficient is nonnegative. Each truncated excess is at most x, so the whole numerator is at most 37 times x.

These structural facts hold even for arbitrary Thresholds records; the separate use of published, ordered constants gives the expression its intended schedule interpretation. A monotonicity proof alone would still be true for many incorrectly transcribed thresholds, which is why source and example checks remain necessary. [12]

```
N(x) = 10x + 2(x-t1)+ + 10(x-t2)+ + 2(x-t3)+ + 8(x-t4)+ + 3(x-t5)+ + 2(x-t6)+; illustration(x) = floor(N(x)/100)
```

x and every threshold are cents. (z)+ means max(z,0), implemented by natural-number subtraction. N is hundredths of a cent. The final floor is the declared research convention.

6. Why a rate illustration is not Form 1040 liability

Publication 505's expected-tax discussion places the schedule calculation before additional items such as AMT and credits, and identifies situations requiring other worksheets. Form 1040-ES similarly supplies a larger estimated-tax worksheet. Our function models neither the entire estimated-tax workflow nor an amount to remit. A number from the ordinary schedule can be arithmetically correct while being insufficient for the decision that a taxpayer actually needs to make. [2][7]

The currently published 2025 Form 1040 instructions provide another concrete distinction: their line 16 discussion separates the Tax Table below $100,000 from the Tax Computation Worksheet at or above that amount, subject to exceptions. We cite that older-year instruction only to explain why filing methods matter, not as the source for 2026 rates or as a representation of final 2026 return instructions. The implementation contains no IRS Tax Table intervals and no return-specific worksheet selection. [8]

Consequently, the output type and documentation call the result an illustration. It is not a refund, balance due, withholding recommendation or individual filing answer. The guard is conservative about several omitted computations, but false exclusion flags are not a legal completeness certificate. A future filing tool would need to resolve income classification, deductions, status, applicable methods and other taxes before presenting an authoritative return amount. Formal precision in one module does not eliminate the dependencies of the surrounding workflow.

7. Worked amounts and cent boundaries

For Single taxable income of $100,000, the first $12,400 contributes $1,240. The next $38,000 contributes $4,560. The remaining $49,600 contributes $10,912. The sum is $16,712, and the exact numerator is 167,120,000 hundredths of a cent. With the base reviewed facts and matching certificate, evaluate returns some 1,671,200 cents. No deduction is applied again because the input already names taxable income. This is one of the executable fixtures. [13]

For Married Filing Jointly at the same taxable income, the first $24,800 contributes $2,480 and the remaining $75,200 contributes $9,024. The illustration is $11,504. At a Single income of $1,000,000, the result is $325,957.25 under this schedule. These examples illuminate schedule arithmetic only; they do not assert which complete tax method or other adjustments apply to a person's return.

The first Single boundary also exposes the floor convention. At $12,399.99, the exact contribution is $1,239.999, which floors to $1,239.99. At $12,400.01, the extra cent contributes $0.0012 and the illustration remains $1,240.00. At $12,400.09 it becomes $1,240.01. One cent of taxable income produces a sub-cent contribution and therefore a zero-cent illustration. The tests state those outcomes explicitly so a later developer cannot silently substitute another rounding rule.

Table 2. Implemented examples and refusal boundaries

Reviewed input or mutation	Guarded illustration	Interpretation
Single; $100,000 taxable income	some 1,671,200 cents	$16,712.00 schedule illustration.
Married Filing Jointly; $100,000	some 1,150,400 cents	$11,504.00 under the joint schedule.
Single; $12,400.01	some 124,000 cents	Extra fraction is floored once at the end.
Single; $12,400.09	some 124,001 cents	Shows the declared final-cent boundary.
Same facts; reviewed = false	none	Review attestation is required.
Same facts; requiresAMT = true	none	AMT is not approximated by this model.
Matching certificate and facts for 2025	none	Only the pinned 2026 bundle is accepted.

Amounts and refusal conditions are reproduced from the arithmetic and Main.lean fixtures; the two cent-level guarded rows follow the same accepted base facts with changed income. These are synthetic illustrations, not individualized tax calculations. [12][13]

8. Fourteen formal statements and their limits

verification_iff connects Boolean success to Conditions. accepted_source_matches, accepted_year_is_2026 and accepted_status_matches expose identity dependencies. accepted_requires_review and accepted_is_ordinary_only expose the two positive scope attestations. unsupported_status_refused and rejected_has_no_illustration establish two refusal properties. These eight theorems state properties of the supplied record and encoded guard; none authenticates taxpayer evidence or proves filing-status eligibility. [12]

numerator_monotone and illustration_monotone prove that increasing represented income cannot decrease the exact numerator or final illustration. numerator_bounded_by_top_rate proves N(x) is at most 37x. illustration_le_income establishes the looser output bound by income, and zero_income_zero_illustration establishes the zero case. one_cent_increase_bounded proves that one additional cent of input raises the illustrated amount by at most one cent. It is a discrete increment bound, not a proof about every real-valued tax function or about after-tax benefits.

These arithmetic properties quantify over all natural-number inputs and threshold structures. They do not constitute a formal equivalence proof between the published IRS prose/tables and the encoded constants. The concrete bases are checked separately by examples. Lean's kernel checks the proof terms relative to its environment and assumptions; a trusted source translation and an appropriate mathematical statement are still

needed. The distinction is essential because an implementation can be monotone, bounded and consistently wrong about a particular statutory threshold. [11]

9. What the thirty fixtures actually test

Main.lean reports thirty named Boolean fixtures and raises an error if any fails. Positive examples cover supported status acceptance, zero income, the two $100,000 amounts, a higher Single amount, the Single bracket bases, the final joint base and a list checking all joint bases. Boundary examples exercise one cent below and above the first Single threshold, the final-floor crossing and a sub-cent result. The same fixture list is also evaluated at compile time, with an additional compile-time check of the Single base list. [13]

Negative examples mutate source, revision, statute identifier, facts year, certificate year, status agreement, supported status, reviewed, ordinaryOnly, hasCredits, requiresAMT and specialComputation. These are meaningful acceptance-boundary tests because they hold most of the base case constant while breaking one required assumption. They are not a sample of taxpayer returns, and the successful count cannot be interpreted as tax-preparation accuracy, coverage of the Code or evidence that upstream review is reliable.

The fixture suite complements the proofs in a different way. Source-derived amounts can detect a transcription error that a generic monotonicity theorem would not notice. Universal bounds can detect a weakened arithmetic implementation beyond the finite sample. Neither replaces the other, and both can share an incorrect intended specification. A stronger assessment should therefore use an independent reference and source review rather than merely generating more inputs from the same implementation under test.

10. Competing representations and related work

A conventional piecewise function, spreadsheet or decimal-money library could represent these two schedules. A band-summing implementation is especially attractive as an independent comparator because its structure differs from the positive-increment formula. A direct encoding of the published base-plus-excess rows provides a second useful oracle. Agreement across differently structured implementations is evidence about transcription and arithmetic, although it still depends on choosing the right tax year and legal scope.

Catala's work on computational law emphasizes a language designed to express legal structure and exceptions, including tax-law examples. Our project instead studies a small Lean function with explicit propositions about acceptance and arithmetic. A larger tax formalization might benefit from combining domain-readable rule representations with independently checked properties. We make no performance comparison, reuse no Catala results and do not claim that fourteen theorems establish a superior tax implementation. [10]

Lexifina supplies a related architectural idea: interpretations can carry evidence that a separate checker tests. In a tax-record pipeline, a model could propose a source year or income classification while a deterministic layer refuses mismatched or unsupported inputs. This repository implements only the reviewed-input and rule layer. It contains no model training, record extraction, calibrated confidence threshold or demonstrated factual-certificate mechanism over taxpayer documents. Those would be separate contributions requiring their own empirical evidence. [14]

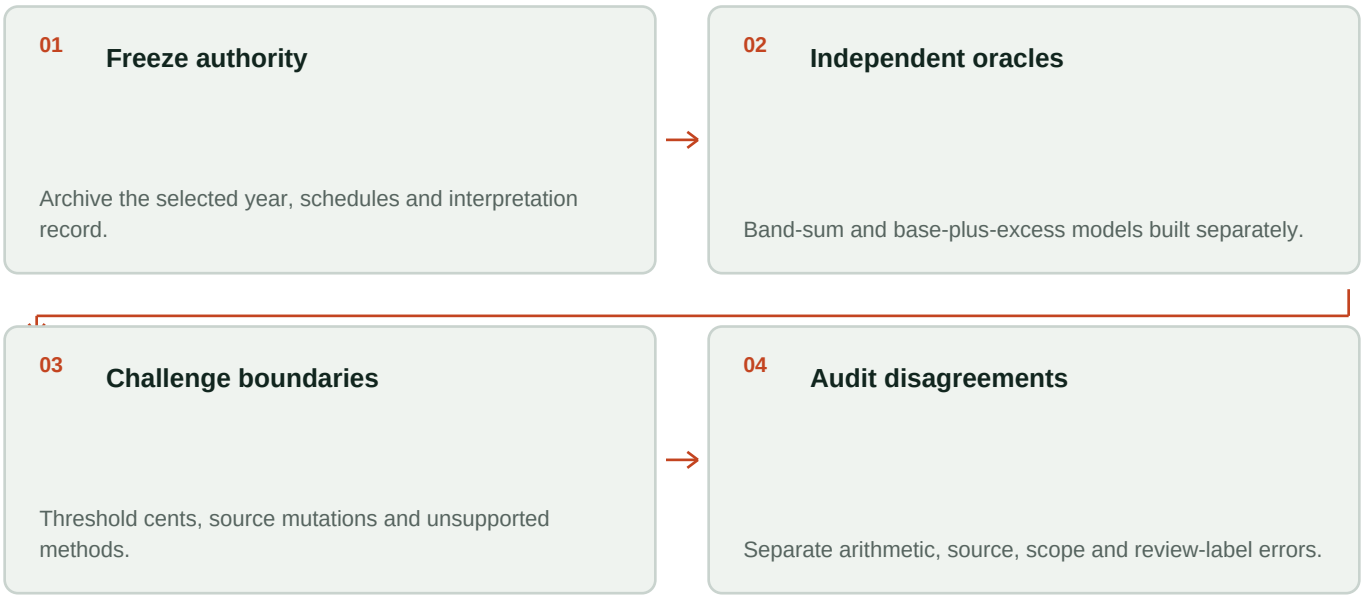
11. A proposed independent evaluation and drift policy

A future arithmetic evaluation should freeze official source snapshots and compare three implementations: the Lean positive-increment function, a separately authored occupied-band sum and a separately authored base-plus-excess schedule. Include every threshold at the boundary and adjacent cents, values across all bands, very small amounts and large natural numbers. Compare exact rational numerators before comparing a shared, explicitly selected output-rounding convention. This protocol is proposed; no new independent benchmark is claimed by this paper revision.

A separate applicability evaluation should use reviewed records with independently labeled year, status and omitted-computation flags. Measure false numeric outputs for unsupported cases, incorrect refusal and the fraction retained in scope. Keep records from one taxpayer or scenario family in the same partition, adjudicate reviewer disagreements and report denominators rather than only a total pass count. Extraction errors, wrong-year selection, unsupported-method errors and arithmetic errors should remain distinct categories.

Version handling should preserve publication identity, retrieval date, code commit, source interpretation and the facts used for each result. A new tax year must receive a new bundle and new checks; replacing a URL's content should not silently redefine an old replay. The present gate verifies only fixed strings and year values, so a live-source monitoring or legal-change service would be an additional system. Before expansion, the source-translation process and its review criteria should be published alongside proposed acceptance tests.

Figure 2. Proposed amount and applicability evaluation



A proposed future study, not a completed benchmark. The current release provides the Lean model and authored fixtures, without taxpayer data or measured filing accuracy.

12. Reproduction, limitations and the next module

This paper binds its executable claims to lean-irs-tax commit 41bd42fe19dd3b183348504aa4f5a426d6c48502. The repository pins leanprover/lean4:v4.35.0, imports Std and requires no additional Lean package. lake build checks the declarations and compile-time examples; lake exe tax_demo executes the thirty named fixtures; python3 scripts/check_proofs.py scans authored Lean source for unchecked project declarations. That source scan is a supplementary guard, not evidence that the formalization covers the intended law. [12][13]

Important omissions remain: taxable-income derivation, status eligibility, other statuses, IRS Tax Tables, capital-gain and qualified-dividend methods, credits, AMT, special worksheets, other federal taxes, payments, refunds and state taxes. A natural-number amount also cannot represent a loss or missing value, so any future data boundary must distinguish those cases before evaluation. The reviewed and ordinary-only flags summarize decisions whose correctness this program does not establish.

The next useful module should be chosen for a specific reviewed workflow rather than for headline breadth. One option is an independently specified IRS table/worksheet selection layer; another is a typed derivation of a narrowly defined taxable-income input. Either should preserve clear source-year boundaries, explicit refusal and separate arithmetic tests. The current result is a reproducible schedule illustration with inspectable assumptions, which is a smaller and more testable foundation than an unsupported claim to have formalized the complete tax system.

References

[1] IRS, Revenue Procedure 2025-32. Section 4.01, Tables 1 and 3, PDF pages 10-11: 2026 joint and unmarried rate thresholds and cumulative bases; section 1 notes the publication/change context. Accessed 2026-11-16.

[2] IRS, Publication 505 (2026): Tax Withholding and Estimated Tax. Chapter 2, Expected Taxes and Credits, lines 4-11c; 2026 Tax Rate Schedules; Worksheet 2-1 and its special-computation cautions. Used for cross-checking and scope. Accessed 2026-11-16.

[3] Office of the Law Revision Counsel, 26 U.S.C. section 1: Tax imposed. Subsections (j), (f) and (h) provide annual-rate and preferential-rate context. Current online text observed through Public Law 119-142; historical opening tables are not the selected 2026 constants. Accessed 2026-11-16.

[4] Office of the Law Revision Counsel, 26 U.S.C. section 63: Taxable income defined. Subsections (a)-(c): taxable-income and deduction definitions; identifies upstream computations the model does not perform. Accessed 2026-11-16.

[5] Office of the Law Revision Counsel, 26 U.S.C. section 7703: Determination of marital status. Subsections (a)-(b): marital-status determination and special conditions; demonstrates why a status label requires separate reviewed facts. Accessed 2026-11-16.

[6] Office of the Law Revision Counsel, 26 U.S.C. section 55: Alternative minimum tax imposed. Subsections (a)-(b): a separate minimum-tax computation; the release flags AMT cases for refusal rather than implementing these provisions. Accessed 2026-11-16.

[7] IRS, 2026 Form 1040-ES: Estimated Tax for Individuals. How To Figure Your Estimated Tax; 2026 Estimated Tax Worksheet, line 4 and surrounding lines; 2026 Tax Rate Schedules. Estimated-tax context is distinct from this illustration. Accessed 2026-11-16.

[8] IRS, 2025 Instructions for Form 1040. Line 16, Tax: less-than-$100,000 Tax Table versus Tax Computation Worksheet and listed exceptions. Cited only to explain method distinctions, not as 2026 rate authority. Accessed 2026-11-16.

[9] IRS, Working Families Tax Cuts: Individuals and workers. Tax inflation adjustments, Marginal rates for tax year 2026: official current summary cross-check of both supported threshold vectors. Accessed 2026-11-16.

[10] Merigoux, Chataing and Protzenko, Catala: A Programming Language for the Law. 2021 paper, revised July 2, 2021; abstract and computational-law language motivation, including tax examples. No benchmark or implementation result is reused. Accessed 2026-11-16.

[11] Lean project, Theorem Proving in Lean 4: Axioms and Computation. Chapter on proof checking, foundational assumptions and computation; bounds the interpretation of machine-checked theorem claims. Accessed 2026-11-16.

[12] Matha Labs, lean-irs-tax: model and proofs. Pinned threshold vectors, Conditions, numerator, illustrationCents, evaluate and fourteen named theorem statements. Accessed 2026-11-16.

[13] Matha Labs, lean-irs-tax: executable fixtures. singleFacts, singleBases, jointBases and thirty fixtures, with compile-time examples; source of the reported regression inventory. Accessed 2026-11-16.

[14] Lexifina, Formalizing all US case law in Lean. November 13, 2026; Section 2, Nothing is taken on trust, motivates evidence-bearing interpretation with a separate formal checker; no legal corpus or vendor measurement reused. Accessed 2026-11-16.

Reproducibility record

Source: https://github.com/RevanthM/lean-irs-tax | Evaluated implementation: 41bd42fe19dd3b183348504aa4f5a426d6c48502