

ARXO

Canons

How We Read Rules and What Follows
From Them



Canons

How We Read Rules and What Follows From Them

CONTENTS

Prologue. Let Us Calculate	4
----------------------------	---

PART I. NOT A VERDICT

1. Four Answers to One Case	10
2. When an Exception Changes the Answer	19
3. When Does a Document Count as Evidence?	26
4. Who Gets to Decide What Is Reasonable?	35
5. A Right You Must Exercise	43

ARXO

CONTENTS

PART II. THE MODEL IS NOT THE SOURCE

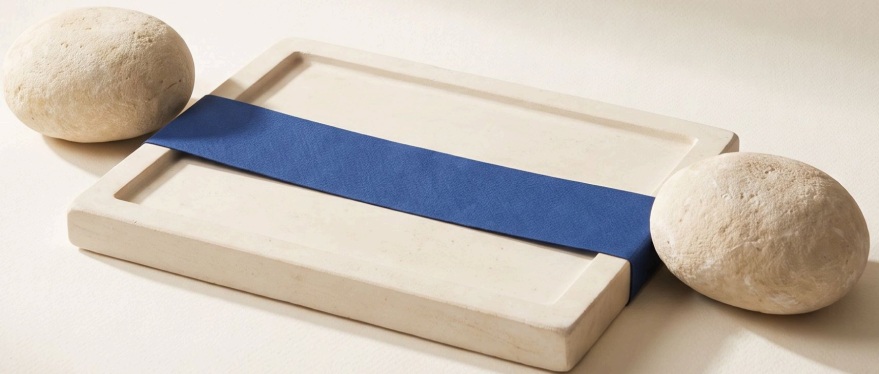
6. What the Model Leaves Out	50
7. Who Chose This Reading?	58
8. The Day the Answer Changes	66
9. One Case, Five Legal Traditions	75
10. A Counterexample Before an Amendment	84

PART III. PROOF IS NOT TRUTH

11. What the Checker Actually Proves	91
12. A Computed Obligation, an Unfinished Dispute	99
Epilogue. Calculemus, Amended	107

PROLOGUE

Let Us Calculate



...*sibi mutuo dicere: calculemus.*

G. W. Leibniz, unpublished manuscript, 1680s

Sometime in the 1680s, in a manuscript he never sent to a printer, Leibniz described a world in which arguments end quietly. When controversies arise, he wrote, two philosophers will need to dispute no more than two accountants do. It will be enough for them to take pens in hand, sit down at their counting-boards, and say to each other, with a friend called in as witness if they like: let us calculate.

The sentence is usually quoted as the daydream of a mathematician. It was not. The man who wrote it held a doctorate in law, earned at twenty, in 1666, at the small university of Altdorf, with a dissertation on perplexing cases: disputes in which the rules, honestly applied, point in two directions at once.

Here is one of the cases he examined, already ancient then. A teacher of rhetoric takes on a wealthy pupil and agrees on a fee: half now, half on the day the pupil pleads his first case in court and wins it. The pupil learns, then takes no cases. The teacher sues for the outstanding fee and addresses his pupil before the judges: you will pay either way. If I win, you pay by the judgment. If you win, you have won your first case, and you pay by the contract. The pupil reverses the argument: I pay neither way. If I win, the judgment says I owe nothing. If I lose, I have not won my first case, and the contract says I owe nothing.

Model, facts and rules

A model is a record of the chosen circumstances and rules for reasoning. Facts state what is taken as known; rules determine which conclusions follow from it. The record rests on a reading of the source.

Stop here and take a side. Does the pupil owe the money? Most people answer within a few seconds, and most answer with a reason the other side has already turned around. The judges in the story could not decide and put the case off to a distant day.

The twenty-year-old Leibniz found a way out, in one line, and the first chapter will test it. For now, keep your answer.

In the first chapter the same case will be put to a machine, and the machine will answer it four different ways, depending on what it is told to count, before it is given Leibniz's way out. Most answers people give are among the four. If yours is not, keep it anyway: the chapter ends with what is still open, and that is where it belongs.

Leibniz promised that calculation would end the dispute. Three centuries of attempts have given the promise a reputation, and the latest attempt speaks in fluent paragraphs: ask a language model about the law and it answers at once, with the assurance of a well-read colleague, whether or not the case it cites exists. This book is about a narrower machine. It is given the text of a rule, rewritten by a person into a form that can be run. Asked about a case, it returns what follows from that rewriting, with the articles it used beside the answer, and it stops where the rewriting stops. It does not understand, does not guess, does not decide.

In this book, *formalisation* means translating selected statements from a source into explicit rules and concepts that a computer can execute. Every such translation involves choices about meaning, scope and interpretation.

The question of this book is what remains in dispute after such a machine has answered.

The short version: the part that turns on people, and all of it now has a name. Which reading of the text was chosen, and by whom. Which document turned a claim into a fact. Who is to decide what the text leaves to judgment. Whether the rewriting is faithful to the source. Whether the checks that vouch for the machine prove what they seem to. The machine cannot settle any of these for you. What it can do is refuse to hide them, and

that is the deal this book offers: every chapter shows, in plain words, where the machine stops and hands the question back.

You will not be asked to take the machine's word. Every chapter puts a case to it and then changes one thing: a verdict is added, a date moves, a document is withdrawn, an argument is switched on. You will be asked to say what should happen before the machine says what did. Sometimes you will be right and the machine will be right. Sometimes an answer will depend on a premise you had not noticed, and the chapter will show where it enters and what changes when it is withdrawn. The experimental notes record the dates, model versions, assumptions and result fingerprints available for the runs discussed here. They also identify gaps in the retained records; a historical summary is not a complete execution archive. This does not establish that the conclusions are true. It gives readers identifiable premises and inference steps to inspect, reproduce and challenge.

The book has three parts, and each is a denial of a sentence that will tempt you by the end of it. *The answer is not a verdict*: what the machine returns has no legal force, and its value lies in naming what the decision turns on. *The model is not the source*: between the law and the machine stands a person who rewrote the text, and every answer carries that person's choices. *The proof is not the truth*: the machine is checked by a second machine and both by a written specification, and the third part asks what agreement establishes, what a certificate proves, and what neither can tell us about the truth of the premises.

The most tempting sentence of all is "the machine proved that under the law...". It never does. It proves what a reading yields. Whether the reading is the law is a question for people, and this book is about asking it well.

It is written for anyone who has been told "it depends" and never learned on what. You may have lost a dispute and still not know whether you lost on the rule or in a gap. You may be a journalist checking a decision, an official defending one, a developer asked to automate one. Each chapter ends by identifying what remains open to challenge. Lawyers may recognise

questions of interpretation, authority and evidence; engineers may recognise questions of modelling, verification and system boundaries. Neither perspective is required to follow the argument.

The machine has a name, Arxo, and its language has a name, Arxo Law. Both are open. The book is not a manual for either: manuals go out of date with the first correction, and the questions here do not.

How to Read a Machine Answer

Established: support was derived for the proposition, with none for its negation. *Refuted*: support was derived only for its negation. *Established and refuted*: both have support. *Not established*: neither has support. These are the four truth statuses; the last is not a negative answer.

Other labels describe other layers. *Not executable* concerns the model's coverage. *Admitted* concerns a certificate step accepted without verification. A request for *judgment* concerns a decision reserved to an authority. None is a fifth truth value.

The comparisons in this book submit the same rules, facts and question to two independently written implementations and compare their answer documents. Agreement establishes agreement; it cannot by itself establish that the model is faithful to its source. The experimental notes distinguish these historical comparisons from the supplemental checks made for this revision.

The experimental notes collect the retained records and their limits.



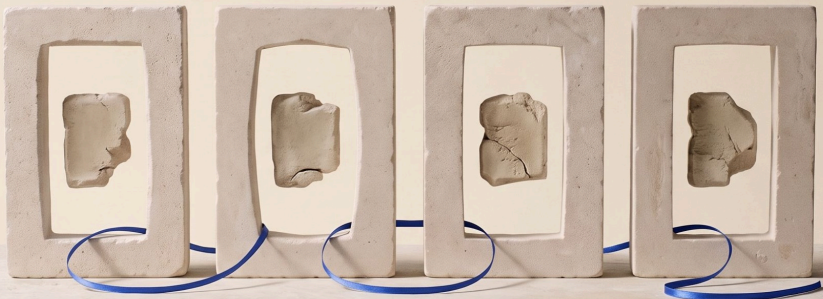
The dispute of Protagoras and Euathlus waits centuries for a decision. Leibniz proposes not to argue but to calculate.

Let us calculate, then, and keep count of what is left.

PART I · NOT A VERDICT

CHAPTER 1

Four Answers to One Case



...et utrimque pariter valet.

Aulus Gellius, *Attic Nights*, V.10.3

You took a side in the prologue. This chapter is about what happens to it.

The story comes from Aulus Gellius, a Roman of the second century who collected curiosities into a book called *Attic Nights*. He does not tell it as a lawsuit. His chapter is about a kind of argument the Greeks called *reversible*: one that can be turned against the person who made it, and, in his words, “stands equally strong on both sides”. Protagoras against his pupil Euathlus is his stock example. Sixteen short paragraphs, no date, no sum, no city, no law: the judges are nameless, and Gellius says only that they thought the matter doubtful and put it off to a very distant day.

In August 2026, a modeller translated Gellius’s sixteen paragraphs into a set of executable rules and facts. That person made choices, and the chapter will come back to them, because they are where your side of the argument either survives or does not. Two of them matter from the start. The contract, the suit and the judges became rules and facts. The two speeches did not: each is written as a *reading* that can be switched on or off, so the machine can be asked with Protagoras’s argument counted, or Euathlus’s, or both, or neither. “Neither” means the contract and court decision, with neither party’s argument enabled. And who won the suit was left to nobody but the judges.

Who won?

Start where Gellius starts: the suit is filed, the parties are before the judges, no verdict has been given. Before the money, one question lies under it: did Protagoras win? Nothing in the story says, because nobody won; the judges declined to decide.

The machine’s answer, in plain words — its own words are a status plus a request: not established; a judgment is required; the

authority that gives it is the judges; the question to put to them is whether this suit has been decided in favour of Protagoras.

Three things are in that answer, and it is worth separating them. The machine did not say no. It did not say “I don’t know”. It said that this question can be settled by a decision of named people, and it named them. No such decision has yet been given. That is the first of the author’s choices at work: there is no rule in the model by which the outcome of the suit could be computed from anything else, and there could not honestly be one, because the story gives the parties no evidence, only two speeches. The machine answered as Gellius did.

Ask about the fee now, before any verdict, with either speech counted or both, and the answer is the same every time: not established. Both speeches begin from a verdict, and there is none.

One verdict, four ways of counting

Now give the judges a decision. Say they found for Protagoras: the pupil lost. It goes into the case not as a fact like “the suit was filed” but as a decision of the court, the one kind of statement the model was written to take from a court.

Does Euathlus owe the fee? Two things before you answer. First, separate the debt under the contract from the duty to obey the judgment: in this model they are different questions, and the one asked here is the contract’s. Second, know how the author wrote the contract’s condition: a victory in a suit that precedes the one in which the fee is claimed. Whether that is the right reading is the next section’s subject; for now it is the rule in play. Then decide what you are counting. The contract and the court only? Protagoras’s speech as well? Euathlus’s? Both? Write down four answers.

Here are the machine’s.

**With the contract and court decision,
without either speech: not established.**

With Protagoras's speech: established. With Euathlus's: refuted.
With both: established and refuted at once.

Four words, so here is what each one means. Established: from the rules counted, the machine derived "the fee is owed" and nothing against it. Refuted: it derived only "the fee is not owed". Established and refuted: it derived both and had no rule to rank them. Not established: it derived neither, which is not a "no"; it is the absence of a conclusion.

FOUR ANSWER STATES

FOR AGAINST



Not established

No support either for or against

FOR AGAINST



Established

Support only for

FOR AGAINST



Refuted

Support only against

FOR AGAINST



Established and refuted

Support both for and against

A filled mark means the support was derived in the model.

If the court found for Protagoras, how can the fee be "not established"? Because the contract and court decision, without either speech, do not read a verdict as a debt under the contract. The condition you were told about requires a victory in an earlier suit, and what the court produced is a verdict in the very suit in which Protagoras is demanding the money. What the verdict does produce is a different obligation, which the machine lists beside the answer: to comply with the judgment. Pay because the court said so, not because the condition of the contract came true. The two speeches blur exactly that distinction, each in its own favour, and the author refused to let the contract blur it.

Now reverse the verdict. The judges found for Euathlus. Ask the same four questions, and the same four answers come back in the same order.

Counted	Judges for Protagoras	Judges for Euathlus
contract and court only	not established	not established
Protagoras's speech	established	established
Euathlus's speech	refuted	refuted
both speeches	established and refuted	established and refuted

Support for on the left, against on the right. A filled mark means the support is present. The columns match. Gellius said the argument stands equally strong on both sides; that was a critic’s remark about a text. Here it is a property you can see: change the verdict, and none of the four answers moves. Each cell was computed twice, by two programs written independently of each other, and their answer documents agreed to the byte.

The version in which Euathlus lost

The table did not always look like this. In the first version of the model, the right-hand column read “established” all the way down. Judges for Euathlus, the contract and court decision, without either speech: established. Euathlus’s own speech switched on: still established. Under every way of counting, Euathlus paid. The author had not intended it and did not notice it until the two columns were put side by side.

The cause was one missing condition. The contract, in that version, read “first victory” without saying in which suit. Euathlus wins this suit; this suit is his first; therefore he has won his first case; therefore the fee is due. Nothing in Gellius forbids that reading. Read that way, the contract fires on the very verdict that dismissed the claim, and the machine, quite correctly, says the fee is established. Euathlus was not losing to Protagoras. He was losing to the person who wrote the model.

The repair was to add the condition: the victory must have been won in a suit that precedes the claim. That is the condition the young Leibniz argued for in 1666, and it is a good one. But call it what it is. It is not a correction of a mistake in the machine. It is a reading of the contract, chosen by a person, and a different person could choose the other. Under the other reading the table is not symmetrical, Gellius’s “equally strong on both sides” is false of this contract, and Euathlus should pay. The model excludes that reading. It does so openly, in a line anyone can find, and that is the most the author can offer you: not that the reading is right, but that it is visible and that you can object to it.

There is a second choice of the same kind in the fourth row, and it is easier to miss. “Established and refuted” means the machine found two complete lines of reasoning, one to the debt and one away from it, and had no ground to prefer either. It has no ground because the author gave it none: in this rewriting both speeches were given the same strength. Write Protagoras’s speech as a rule that cannot be set aside and Euathlus’s as one that can, and the machine hands Protagoras the win in every cell without a word of complaint, because that is the inequality it was given. The author tried that, measured it, and rejected it, because Gellius says the two stand equally. Again: a choice, recorded, open to objection.

The second suit

Leibniz's resolution was not that the case is unanswerable. It was that it was brought too early. Let Protagoras lose. Now Euathlus has a won case behind him. Let Protagoras sue again.

Before the machine answers, decide what should happen, and decide it under each of the four ways of counting. Do the speeches still apply?

The case: first suit decided for Euathlus; a month later, a second suit, with the first marked as preceding it. The dates are conventional, Gellius gives none; the order is what matters. Four questions.

Established. Established. Established. Established.

The speeches no longer apply, and the reason is worth stating exactly. Each speech begins from a verdict in the suit under consideration. The second suit has no verdict. So the readings that produced "refuted" and "established and refuted" in the first suit produce nothing here, and the contract answers alone: its condition is met, the fee is due, and an obligation to pay it is in force.

No new rule was added to get here. The same model that answered "not established" and "established and refuted" answers "established" on the second suit because one thing in the case changed: the winning suit now comes before the suit that claims the fee. The condition the first version lacked is Leibniz's whole argument, written as a premise instead of a footnote. The chain, each step with its paragraph of Gellius: Euathlus spoke for himself and took no advocate, so he pleaded the first suit himself; the judges found for him, so he won it; it was his first, so his first victory has occurred; there is a contract, a second suit, and the first precedes it, so the second half is owed; the fee is owed, so the duty to hand it over arises.

That chain was also checked by a third, independent program, one that verifies derivations rather than producing them. It certified the chain to the fee step by step; it does not yet vouch for the silences or the collision, and the experimental notes says

exactly what it checked. The positive answers in this chapter are proved more strongly than the rest.

Ground of a conclusion

A fact or rule on which a conclusion rests. The chain of reasoning links the initial grounds to the answer through intermediate conclusions; each step can be examined separately.

What is left to argue about

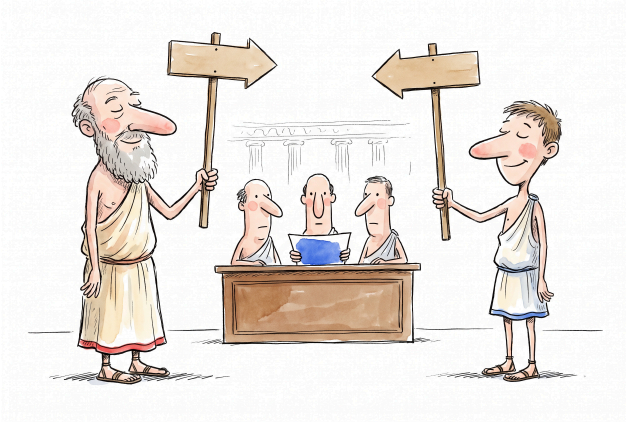
The paradox is not solved by a trick. It dissolves as soon as obligations are tied to time, and Leibniz saw that at twenty. But the machine did not solve it either. It computed, four times over, what follows from a reading, and the reading was a person's.

Here is what that leaves you, whichever side you took. If you said Euathlus owes the money, the model agrees with you only when Protagoras's speech is counted alone, or on the second suit. On the first suit, with the contract and court decision, without either speech, it neither agrees nor disagrees: without a speech counted it derives neither "owes" nor "owes nothing", and your answer stays unconfirmed, for a reason you can now name and dispute: the author decided that a verdict in the suit itself cannot be the ground of the claim. If you said he owes nothing, the model agrees with you only when his speech is counted alone, and never on the second suit. If you said the case cannot be decided, the model says so too, but only when both speeches are counted, and it says something more precise than "cannot": two lines of reasoning, both complete, and no rule to rank them. In this rewriting both speeches were given the same strength. The author could have given one of them precedence and got a different answer.

None of that is a verdict. No court decided this case, Gellius's judges least of all, and nothing proves that any court would decide it the way the model does. What the machine did was smaller and more useful: it took

“it depends” and said on what.

On a reading of the contract. On whose speech is counted. On whether the verdict came before the claim or in it. Each of those is a place where the person who wrote the model made a choice, and each is a place where you can now push back with a reason instead of a feeling.



Protagoras and Euathlus before the judges. Each argument turns against its speaker; the decision stays with the court.

The judges put the case off to a very distant day. It is still there, and it now has a map.

PART I · NOT A VERDICT

CHAPTER 2

When an Exception Changes the Answer



*...have the right to compensation by the operating air carrier in accordance with Article 7, unless:
Regulation (EC) No 261/2004, Article 5(1)(c)*

A flight inside the European Union, twelve hundred kilometres, is cancelled. The passenger had a confirmed booking and learned of the cancellation the day before departure. Airport notices often summarise passengers' compensation rights: €250 for shorter flights, with higher amounts for longer distances.

Is the passenger owed two hundred and fifty euros? Take a moment. Nearly everyone says yes, and nearly everyone is right.

The machine agrees. The model contains executable rules linked to every article of Regulation (EC) No 261/2004, drawn from the official English text. This does not mean that every possible interpretation or consequence has been formalised. In particular, the model does not incorporate the subsequent case law. Asked about this passenger, the machine says: established, two hundred and fifty euros. Alongside the answer, unasked, it lists the duties the cancellation created for the carrier at that same moment: offer a refund or re-routing, offer meals and phone calls, hand over a written notice of the passenger's rights, explain the alternative transport.

That is the easy case. The chapter is about the word that follows the right in the regulation's own sentence. *Unless*.

The carrier objects

The carrier writes back: we informed you two weeks in advance. The regulation does say that a passenger informed at least two weeks before departure gets no compensation.

Change one thing in the case. The carrier states that the passenger was informed on the twelfth of August at ten in the morning, for a departure on the twenty-sixth at ten: fourteen days to the hour. It has not yet produced proof. Is that statement enough to take the money away?

Most people say no. The machine says: established, two hundred and fifty euros.

This is not a slip, and it is the first place where you should look for the person who wrote the model. The regulation adds, in a separate paragraph, that the burden of proof concerning whether and when the passenger was informed rests with the carrier. That sentence was written into the model as a presumption: until the carrier has proved the notification, the passenger counts as not informed, and an exception that begins with “they are informed” cannot start. A date in the file is a claim.

The carrier’s proof is a separate fact, and it is not there.

Burden of proof and presumption

The burden of proof says who must confirm a circumstance. A presumption sets the position held until it is rebutted. Here the passenger counts as uninformed until the carrier proves notification.

Add it. The carrier proved the notification. Ask again.

Refuted: under the rules counted, the right does not exist. And the amount: not established, because with no right there is nothing to price. Two answers to two questions. The compensation that was owed a minute ago is, on this reading, not owed.

What “refuted” is made of

Look inside that answer: it is a different kind of thing from “not established” in the first chapter. The machine did not fail to find a right. It found two rules that both fit the facts. One is the general rule: in case of cancellation the passenger has the right to compensation. The other is the exception: not if informed at least two weeks before. Both fired, and their conclusions are

opposite, and the answer carries a record of what happened between them. The general rule was *defeated* by the exception, and the record names the ground: a rule of precedence, declared by name, which says that the exception is the more specific provision and prevails.

In the first chapter two arguments of equal standing collided, nothing in Gellius ranked them, and the machine handed back the collision. Here the text ranks them: “unless” subordinates the right to its exceptions. So one rule wins and the machine returns the winner. It does not erase the loser. The general rule is still in the answer, marked as defeated and by what, so that a reader who wants to argue that the exception should not apply knows which sentence to argue with.

Here is the second place to look for the author. That “unless” means “the exception beats the right” is a reading. It is a very natural one, and few lawyers would read it otherwise, but it was declared by a person, who wrote down which rule has precedence, and it is visible in the answer rather than buried in the order in which the rules happened to be written. Had the author declared nothing, the machine would have returned “established and refuted”, exactly as in the first chapter, and the passenger would have been told that the text itself does not settle the question. The ranking is the author’s claim that the text does settle it, and you may dispute the claim.

Exception and priority

An exception changes the general rule's effect under special conditions. Priority says which of the colliding rules prevails; the choice of that precedence itself needs a justification.

One hour

Move the notification by one hour. The twelfth of August at eleven, not ten; departure still on the twenty-sixth at ten; the carrier’s proof still in the file. Two weeks less one hour. Who gets the money now?

Established. Two hundred and fifty euros, back on the table.

The regulation says “at least two weeks before the scheduled time of departure”. The model counts hours, because the text gives a scheduled time, not a date: fourteen days are three hundred and thirty-six hours, and the carrier managed three hundred and thirty-five. A person could count calendar days instead and call the twelfth “two weeks before” the twenty-sixth. That is the third choice, and the least visible of the three: it sits in the rule that compares two moments, and nothing in the answer flags it as a choice. You have to know to look.

The case	Right to compensation
Cancelled; passenger told the day before	established, 250 EUR
Told two weeks before; carrier has not proved it	established, 250 EUR
Told two weeks before; carrier proved it	refuted; no amount
Told two weeks less one hour before; proved	established, 250 EUR



The carrier claims it warned in advance. A claim without proof does not change the answer.

Four rows, one passenger, one flight. The money appears, disappears and reappears. The output identifies the operative rules and the provisions supporting them. It does not yet expose every interpretive choice embedded in those rules, including the decision to measure this deadline in hours. Both implementations agreed on all four cases.

Where the model stops

336

HOURS

The notification-deadline condition is met.

335

HOURS

One hour short: the exception does not apply.

Comparison of two cases under the chosen model; all other circumstances equal.

Two of the carrier's other defences are not decided here. If the carrier pleads extraordinary circumstances, the machine does not decide whether a storm qualifies; it says that a court must, and keeps the right standing until the court answers. And if the flight was delayed rather than cancelled, the machine says compensation is not established, although European courts have awarded it for long delays since 2009: the model was built from the text alone, and the court decisions are not in it. That is a true statement about the model and a misleading one about the law as applied, and Part Two is about how such gaps are measured and declared.

What is left to argue about

A good lawyer keeps a list of the unlessees that could undo a claim. The machine keeps the same list, written down, ranked, and attached to every answer. That is not a small thing, but it is not a verdict either. Here is what it leaves you.

If you were the passenger in the third row, you lost, and you can see the sentence you lost to. You can also see the two choices that carried it there: that the carrier's proof, not the date, is

what counts, and that the exception outranks the right. Both are readings of the text, and both are named in the answer.

If you were the carrier in the second row, you lost with a true date in your hands, and you can see why: the model treats your notification as a claim until you prove it. That is the regulation's own allocation of the burden, read by a person into the machine. If you think the reading is wrong, you know what to say.

If you were either of them in the fourth row, you are one hour from a different answer, and the choice that put the line there is the one the answer does not flag. A calendar would have moved it. Nobody has proved that hours are the right way to read "two weeks"; the model takes it as an assumption.

A third, independent program re-checked the rule steps of both machines and, for every defeat recorded in these answers, that the declared ranking really does make the winner the stronger rule. What it does and does not vouch for is in the experimental notes; the positive answers here are proved more strongly than the defeats, and the defeats more strongly than the silences.

None of this decides the case. What it does is turn "unless" from a footnote into a rule that beats another rule, with the ranking declared and the defeat recorded, so that the passenger and the carrier argue about the same sentence and know whose reading they are arguing with.

PART I · NOT A VERDICT

CHAPTER 3

When Does a Document Count as Evidence?



Aidos works for a small company in Almaty. The office needed a keyboard. The purchase was approved, Aidos paid ten thousand tenge through Kaspi and sent the documents to Asem, the accountant. Now he is waiting to be reimbursed.

“But I sent you the receipt,” Aidos protests.

“You did,” Asem replies. “The receipt is for the keyboard. The payment confirmation is for a kettle.”

“The amount is the same, though.”

Aidos had bought the kettle for his own home. Its genuine payment confirmation had somehow ended up among the documents for the office keyboard. In the books the keyboard is purchase 42, the kettle is purchase 99. The approval order and the vendor’s receipt concern purchase 42. The bank confirmation is for the same ten thousand tenge, but for purchase 99.

Aidos, Asem, the company and its rules are invented for this example. The bank’s name helps picture the scene; the requirements on documents we set ourselves. By the terms of the problem the organisation reimburses an expense if the purchase’s approval, amount and payment are established without contradiction, and the amount does not exceed ten thousand tenge. For the calculation we take it that every document is current and the audit has confirmed their genuineness.

Is such a folder enough to reimburse purchase 42? Most people say yes — and most people would be signing off on a payment that belongs to another purchase.

The pile, as the machine reads it

The rule and the employer’s policy on documents were given to the machine together. The policy says which kinds of document, issued by whom and checked by whom, may support which of the three facts; it was written by the same person who wrote the rule, and the machine knows nothing about banks, tenge or audit desks except what that policy tells it.

Evidence admission policy

The set of conditions under which a document is admitted to confirm or rebut a given claim. The pair "document plus claim" is checked, not the document alone.

Ask whether purchase forty-two is reimbursable. Not established. And with the answer comes a report, one line for every document paired with the claim it was offered for. The order, offered for the approval: accepted — under the employer's policy, this document may support that one fact, and nothing more so far. The receipt, offered for the amount: accepted. The bank confirmation, offered for the payment: not accepted. No rule of the policy admits it, and the rule that came closest matched ten of its eleven conditions and failed on the eleventh, which the report prints: the document must concern purchase forty-two.

Read that line slowly. The machine did not call the document false. It is genuine, checked, in date, from a trusted bank. It did not say the purchase is unpaid. Ask about payment directly and the answer is not established, not refuted. It said something narrower and more useful: this document, offered for this claim, does not count, and here is the condition it fails.



The receipt is genuine. But Aidos bought a keyboard and confirmed the purchase of a kettle.

A document that does not count is not evidence of the opposite.

Now add the right confirmation: purchase forty-two, ten thousand tenge, checked by the audit desk. Established. The new document is accepted, and the report points to the step in the proof where the acceptance happened.

The amount has not changed. The rule has not changed. What changed is that the bank document now concerns the purchase you are asking to be reimbursed for.

Genuineness and evidential weight

Whether a document is genuine and whether it can confirm the needed claim are different questions. A genuine bank confirmation of another purchase does not prove payment of this one.

What a document is to the machine

Here is the chapter's one idea. In the first two chapters the facts of the case were taken as given: the suit was filed, the flight was cancelled. Here three of the facts the rule needs are declared *protected*. No statement in the case establishes them, whatever authority stands behind it. The only way in is a document, offered for a specific claim, and accepted by a rule of the policy.

So a document is not a fact and not a witness. It is one half of a pair, and the other half is the claim it is offered for. The confirmation of purchase ninety-nine is a perfectly good document and a bad support for the claim that purchase forty-two was paid; offered for a claim about purchase ninety-nine it would be accepted at once. Acceptance is decided pair by pair, and the report lists pairs, not documents.

That is why the author's choices are so visible here. Who counts as a trusted verifier, which issuer may vouch for which kind of fact, what a receipt must say: none of that is the machine's. The machine does not read the bank's paper; it reads what the folder says about each document, its kind, issuer,

amount and dates. It checks those dates against the date of the case and against the last day on which anything may still count, checks that each audit record is tied to its document by the document's fingerprint, and runs the policy's rules over the result. A different employer would write a different policy on the same frame.

"Sake approved the purchase," Aidos insists.

"His approval may establish authorisation," Asem replies, "but it does not establish payment."

Even if Sake is entitled to approve the purchase, another question remains: which document confirms that this purchase was paid? Approval, amount and payment each need a document that satisfies the organisation's rules. A bare statement "the purchase is paid" is not enough.

Would you accept this one?

Try a few more folders. Decide each one before reading the machine's answer.

No bank document at all, but a plain statement that purchase forty-two was paid, carrying the court's mark that in the first chapter let a verdict into the case. Not established: a protected fact requires a document. Even a statement carrying the kind of judicial authority accepted in Chapter One cannot establish payment here. This model requires payment to be supported by an admissible document. Both are choices, and both are on the page.

The right confirmation, but its audit check is dated the following January, while the case is decided in March. Not established. The document existed in March; the confirmation of its genuineness did not, and a decision taken in March cannot rest on a check performed ten months later.

The right confirmation, but for a different amount: one tenge instead of ten thousand. The purchase is the same, the genuineness check is in place. Still not established.

The receipt is the anchor in this policy: it fixes what the purchase was, how much and for whom, and every other document

must agree with it. The currency matters too: a confirmation for ten thousand roubles against a ten-thousand-tenge purchase will not do either. The equality of the numbers is not enough. A separate folder, with an approval order naming a different employee, is rejected the same way, and the report names the condition: the claim concerns employee seven. In that folder payment is still established; only the approval is lost. The machine does not lose the case when one document is rejected — only the fact that document was carrying.

Two bank records that disagree: the confirmation, and a later record that the payment was reversed, offered against the claim. The policy accepts both. Here is a choice worth seeing: the author recorded the reversal as a denial of the payment, not as a later event that cancels it, and gave the later record no precedence. So the machine keeps both conclusions. Payment is established and refuted, the same third state as in the first chapter. Reimbursement needs payment established and not refuted, so it is not established. The machine did not average the two, did not prefer the later one, did not invent a fourth kind of answer called “disputed”.

The folder	Payment	Reimbursement
Approval, receipt, confirmation of purchase 99	not established	not established
The same, plus the right confirmation	established	established
Right confirmation, audit check dated next year	not established	not established
Right confirmation for one tenge	not established	not established
Right confirmation plus a reversal record	established and refuted	not established

Every question in this chapter was put to two programs written independently of each other, and their answer documents agreed byte for byte.

The folder in a different order

One more, and it is the strangest. Two documents carry the same identifier, and their contents differ: one concerns purchase forty-two, the other purchase ninety-nine. What should the machine do?

It gives no answer at all. The run stops with an error naming the collision.

The refusal preserves a basic condition of evidence: one identifier must name one document. Choosing whichever conflicting record happens to be listed last would let the order of a folder decide the answer. Rejecting the collision keeps the question visible for the person who can resolve it.

How far to trust the explanation

The chapter has leaned on one promise: that when a document is rejected, the report says why, and says it precisely enough for you to fix the folder. For the confirmation of the wrong purchase that promise holds: ten conditions met, the eleventh named. Now look at the one-tenge confirmation.

It is rejected because one tenge does not confirm payment of the required amount. The explanation must identify that failed condition and that document. A nearby record about another purchase cannot supply the reason for this refusal.

Checking a rejection therefore means checking both the status and the grounds attached to it.

A third program, built to verify derivations rather than produce them, certifies that each accepted support is bound in the proof to its policy, its document, its pairing and the rule that admitted it, and rejects seven forged certificates; it does not re-run the policy's rules or the reading of the documents, and it certifies neither "not established" nor "established and refuted". Nothing certifies the explanations of rejections at all. They are the most useful part of the answer to the person with the folder, and the least proved.

The retained run records are in the experimental notes.

What is left to argue about

A lawyer preparing a claim keeps two lists: what must be shown, and what shows it. The second list is where cases are won and lost, and it is almost never written down in a form anyone else can check. Here it is written down, pair by pair: this document, for this claim, under this policy, checked by this verifier, as of this moment of knowledge, admitted by this rule or stopped at this condition.

That leaves you three things to argue with. The policy itself: that the receipt is the anchor, that the audit desk is to be trusted, that payment is protected even from a court's word. Those are one person's choices, and nothing in the machine says they are good ones for any real employer. The moment of knowledge: the machine refused a check from next January, and you may think a check is a check whenever it was made. And the explanations: a reason must refer to the document and condition actually examined. A useful explanation remains something to check against the record, rather than a finding with authority of its own.

None of it is a verdict. It is a list of pairings and the reasons for them, and the reasons are the part you can still get wrong.

PART I · NOT A VERDICT

CHAPTER 4

Who Gets to Decide What Is Reasonable?



...what an objective, reasonable person would have understood the commercial to convey.

Leonard v. PepsiCo, Inc., 88 F. Supp. 2d 116 (S.D.N.Y. 1999)

The commercial ran in 1995. A teenager gets ready for school to a military drumroll: a Pepsi Stuff T-shirt, seventy-five points; a leather jacket; sunglasses. Then the wind rises, three boys gaze up, and a Harrier jump jet lands in front of the school. The boy climbs out, says “Sure beats the bus”, and the caption reads: HARRIER FIGHTER 7,000,000 PEPSI POINTS.

7
million

POINTS

For the Harrier jet in the Pepsi commercial. Can that figure be taken as a promise?

A twenty-one-year-old in Seattle read the caption as a price. The rules of the promotion let you buy points at ten cents each. He raised the money, sent in an order form with fifteen original points and a check for seven hundred thousand and eight dollars fifty, and asked for the jet. PepsiCo returned the check. He sued for the plane.

Does he get it? Nearly everyone says no, and nearly everyone gives the same reason: no reasonable person would think a soft-drink company was giving away a fighter aircraft. Hold on to that reason. The chapter is about who is allowed to give it.

Three reasons, not one

The case reached Judge Kimba Wood of the federal court in Manhattan, and in 1999 she dismissed it. Her opinion gives three reasons, each on its own enough to dismiss the claim. First, the commercial was merely an advertisement, not an offer. Second, its tongue-in-cheek attitude would not lead a reasonable person

to conclude that the company was giving away jets. Third, there was no signed writing between the parties, and a contract for goods above five hundred dollars needs one. The opinion calls the third an entirely separate reason, and it is.

That opinion, not the commercial, is what was rewritten for the machine. The commercial, the catalogue and the order form are known to the model only as the opinion describes them, and the appeal that affirmed her the following year is not in it.

Now the experiment. Give the machine the facts as the opinion describes them, but hold back one thing: the judge's finding about what a reasonable person would have understood. Everything else stays. Are there grounds to dismiss the claim? Established: yes.

Beside the answer stand the rules that carried it, and there are two of them, not three: the advertisement was not an offer, and there is no writing.

The second reason is missing.

Offer and reasonable perception

An offer is a proposal whose acceptance can conclude a contract. In the dispute over the commercial, whether a reasonable person could have perceived it as such an offer is established separately.

The question the machine will not answer

It is missing because the second reason turns on a question the author of the model refused to let the machine decide. Ask it directly: would an objective, reasonable person have understood the commercial as an offer of the jet?

Not established. And the answer names what is missing: a finding whether a reasonable person would have taken the commercial as an offer of the jet, and who is entitled to make it, the

court. That is not a message addressed to anyone. It is a line in the answer naming the conclusion the machine will not draw itself, and whose conclusion it needs instead.

You gave your own answer to that question a page ago, in a few seconds, and you were probably right. The machine does not say you were wrong. It says that “reasonable” is not a fact of the case and not a rule of the opinion; it is a finding someone with authority makes, and until that finding is handed to it the machine holds the question open and names them. That is the first choice of the author, and it follows the opinion’s own words: the test is what an objective, reasonable person would have understood, and the opinion says this is for the court, not a jury, because the words and actions here were so clear that reasonable people could not differ. The model carries that too: ask whether the question is for the court, and the answer is established.

Now give the machine Judge Wood’s finding: no reasonable person would have understood the commercial as an offer. Ask again whether the commercial was objectively understood as an offer. Still not established: the negative judicial finding closes the request for an external decision, but the model has no bridge deriving the negation of this queried proposition from that finding. Closing the request and deriving a negation are separate operations. Does the claim fail? Established, as before: the other two grounds were enough on their own.

Now suppose the opposite finding: a judge held that a reasonable person would have read the commercial as an offer. Enter that instead. The machine now records that a reasonable person would have understood the commercial as an offer: established. That is one condition of an offer, not the offer itself; the advertisement still fails the others. Does the claim still fail?

Yes. Established. The first and third reasons are untouched.

That is what “three separate reasons” means when it is run rather than read: reversing the judge on reasonableness moves nothing, because the other two grounds each defeat the claim alone.

Evaluative judgment and authority

The assessment of reasonableness in this case belongs to the court. The program can point to whose judgment is missing and use the judgment once given; the authority to make it does not arise from the computation.

One limit of the model belongs here. The opinion lists five circumstances that made the commercial an evident joke, from the improbable pilot to a price of seven hundred thousand dollars for a machine that cost tens of millions, and never says how many of the five would have sufficed. The model requires all five; give it four and it says neither “in jest” nor “not in jest”. That silence is a choice, and a visible one: the rule has five conditions, and you can count them.

Taking the reasons away one at a time

If reversing the judge does not win the jet, what would? Take the reasons away and see.

Following the principle illustrated by *Lefkowitz v. Great Minneapolis Surplus Store, Inc.* (1957), rewrite the commercial as a definite offer, leaving no essential terms open for negotiation. In this counterfactual version, it expressly offers the Harrier jet and specifies how the offer may be accepted. The scenario is deliberately hypothetical: it isolates the remaining contractual requirements and does not reconstruct what PepsiCo actually offered. The model ranks an exception for such advertisements above the general rule that advertisements are not offers. Now the commercial is an offer: established. Keep the favourable finding on reasonableness. Two reasons gone. Does the claim fail?

Established. The third reason stands alone: no signed writing. The opinion calls this an entirely separate reason, and the machine shows what that means: an advertisement that is an

offer, understood as one by a reasonable person, accepted by performance, still yields no enforceable contract for a seven-hundred-thousand-dollar aircraft without a signed document.

Add the document: a writing signed by the party charged, establishing the contractual relationship. Does the claim fail?

Not established. And beside the answer, for the first time, an obligation: the seller owes the claimant delivery of the item. In a world where all three reasons are cured, the jet is owed. That world never existed, and the machine does not pretend it did. The row exists for a different reason: without it, “the claim fails” would be indistinguishable from a rule that fails every claim.

The case	Does the claim fail?
Facts as the court found them	established: no offer, no writing
Judge finds no reasonable person would see an offer	established
Judge finds a reasonable person would see an offer	established: no offer, no writing
Commercial rewritten as an offer; favourable judgment	established: no writing
Signed writing added	not established; a duty to deliver arises

The original comparisons and certificate checks are recorded in the experimental notes. The two rewritten-advertisement scenarios were separately rerun with the Harrier for this revision; their inputs and outputs are retained there. Those supplemental runs must be distinguished from the original certificate checks.

The retained run records are in the experimental notes.

What is left to argue about

The title asked who decides what is reasonable. The opinion's answer is: the court itself, without a jury, because here reasonable people could not differ. The machine's answer is narrower and worth stating exactly: not the machine. It carries the court's finding, in either direction, and shows what follows from it; it never produces the finding. That is the line the author drew, and you can see the line in the answer as a request with an authority's name on it.

Two more things are yours to dispute. The five findings: the model requires all of them, and the four-finding case is left open; a lawyer might say three would do, and the model says only that the opinion did not say so. And the rewriting itself: the commercial, the catalogue and the order form are known here only through the judge's description of them, and the appellate decision that affirmed her is not in the model at all. A reader who thinks the judge described the commercial unfairly is arguing with the opinion, not the machine, and the machine has made sure that is where the argument lands.



Leonard expects a fighter jet; the company returns the cheque. Whether the promise could be taken seriously is a question for the court.

The boy in the commercial got the jet. The one in Seattle did not, and the machine can tell you that all three reasons would have to go before he could, which of them it can remove for you, and which one it is not its business to judge.

PART I · NOT A VERDICT

CHAPTER 5

A Right You Must Exercise



The game is drawn, upon a correct claim by a player having the move, when the same position for at least the third time...
FIDE Laws of Chess, Article 9.2

Imagine a game. The same position has just appeared on the board for the third time, and it is White's move. Nobody has claimed a draw. Both players know the club rule: threefold repetition is a draw.

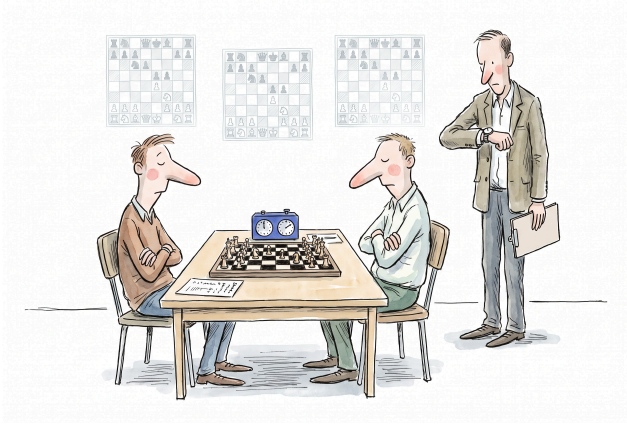
For repetition purposes, positions are identical only when the same player has the move, pieces of the same kind and colour occupy the same squares, and the possible moves are the same. Castling rights and en passant can matter. The model receives that determination from the game record rather than calculating it.

Is the repetition itself enough to end the game? Take a moment. Most people say yes.

Ask the machine, which has the 2023 Laws of Chess in a form it can run, and it says: not established.

Not because it disagrees with you about the position. Because the rule does not say what most people remember it saying. Read the epigraph again: the game is drawn *upon a correct claim by a player having the move*. The draw does not happen to the board. White has acquired a right, the right to claim the draw.

Until White claims it, the repetition ends nothing, and playing on is no breach of anything.



The position has occurred for the third time. Both hold the right to a draw, but neither has claimed it.

Only the player with the move holds that right, and only now.

Who may claim, and when

Let White claim. Drawn: established. The answer names both steps: the claim is correct, and a correct claim ends the game at once.

Now suppose Black claims and White says nothing. Drawn?

Not established. Black does not have the move, so Black holds no such right at this moment. Black's claim produces no draw: in this model nothing is triggered by it, and the machine does not even mark it as refused. What other consequences the claim may have, the answer does not say. The claim does not establish a draw under the modelled repetition rule. Whether the game ends for another reason is a separate question.

Try one more. White claims, but the position has appeared only twice. Not established: the claim is incorrect and produces no draw. From that answer you cannot conclude that the claim has no consequences at all; the model says only that it does not end the game.

So the draw by repetition is a right in a precise sense. The condition, three occurrences, is a fact of the game. The draw is

not. Between the fact and the draw stands an act that a particular person must perform at a particular moment, and a machine that reads the rule as written must refuse to supply the act on the player's behalf. Most players, and most summaries of the rules, quietly supply it. This one does not.

Here the model has to stop and say what it is. It is not a chess engine. It does not know where the pieces stand; it does not count repetitions or moves; it does not decide whether a move is legal or whether a king can still be mated. All of that arrives as facts of the record: who has the move, how many times the position has appeared, how many moves without a pawn move or capture, how many illegal moves each player has completed, whether the position is one in which mate is still possible. The person who wrote the model chose to take the count from the arbiter's record rather than derive it from the moves, and the choice is not cosmetic: a model that counted for itself would draw its facts from another source, and the author chose the record.

Having a right and exercising it

A right gives the ability to demand a result. Using it may require an act: for example, claiming a draw. That grounds for the claim exist does not mean it has been made.

A draw that needs no claim

Raise the count. The same position for the fifth time, and nobody says a word. Drawn?

Established. No claim, no player named. The Laws have a second rule, added in 2014, under which five occurrences end the game by themselves, and the machine applies it without asking who had the move. Three occurrences give a player a right; five take the decision away from both players.

The record	Drawn?
Third occurrence; no claim	not established
Third occurrence; White, having the move, claims	established
Third occurrence; Black, not having the move, claims	not established
Second occurrence; White claims	not established
Fifth occurrence; no claim	established

The Laws hold a second pair of the same kind, the fifty-move and seventy-five-move rules; both are in the experimental notes, and they change nothing in the picture.

Completeness assumption

The condition that all material records of a given kind are on file. Only on the example's accepted assumption of completeness does the absence of a record of unmateability allow mate to count as possible.

Things that happen to you

Not every consequence in the Laws waits for a claim. Some fall on a player whether anyone asks or not.

A player completes an illegal move. Under the FIDE Laws, the first completed illegal move results in two additional minutes being awarded to the opponent. The model does not calculate clock adjustments; it addresses the consequences for the game's result, and this first move does not establish a loss. The same player completes a second illegal move. Lost: established, and the game ended at that moment. Now change one fact: the opponent has nothing left to mate with. Lost?

Not established. Drawn: established. The article that declares the loss adds “however, the game is drawn if the position is such that the opponent cannot checkmate”, and the model reads both halves. What it takes as a fact, again, is the mate-possibility itself; that is analysis, not rule, and it comes from the record.

Notice one more choice hidden in that article. It says the arbiter “shall declare the game lost”. The model does not wait for the declaration; it treats the loss as following from the count, and the arbiter’s declaration as the announcement of a result already settled. That is a reading. A different author might have made the declaration itself the event, as the claim is for a draw, and then a second illegal move with no arbiter at hand would leave the game undecided. The model does not, and you may think it should.

So the chapter’s rules come in two kinds, and the machine treats the two kinds differently. Some consequences follow from the record by themselves: five repetitions, a second illegal move. Some follow only from an act that a named person must perform at a named moment: the claim of a draw. A player who knows only the first kind loses draws they were entitled to. A machine that modelled only the first kind would hand out draws nobody claimed. The third kind, findings that only an authority can make, was the fourth chapter’s subject; the Laws have their own, the arbiter’s judgment whether a piece was touched with intent, and it is in the experimental notes.

Every row above, and the rest of the model’s cases, were computed by two programs written independently of each other, and their answers agreed to the byte. A third program checked the derivations, and proves less here than in earlier chapters, because most of these rules compare counts; the experimental notes says what it vouched for.

What is left to argue about

The Laws of Chess are an unusually candid source: they say in their own preface that they cannot cover every case and that an

arbiter's judgment fills the gaps. The model takes them at their word, and that is where the disputes begin.

You may dispute the boundary of the record. The model asks the record for repetition counts and mate-possibility because deriving them would make it a chess engine and a different thing; a player who believes the arbiter's count is wrong is arguing with the record, and the machine has made sure the argument lands there and not on the rule.

You may dispute the declaration. The loss for a second illegal move is automatic in the model and pronounced by the arbiter in the text; if you think the pronouncement is the event, the row for the second illegal move should read "not established until declared", and the author chose otherwise.

And you may dispute what the model leaves out. A claim made before the move is played, by writing the move down and declaring the intention to make it, is a second form of the threefold claim in the Laws; the model

knows only the claim on a position that has just appeared.

Clocks, and with them the first illegal move's penalty, are outside the articles it covers.

None of this is a verdict on any game. What the chapter shows is narrower: that a right can exist on paper, have its condition satisfied on the board, and still do nothing, because nobody with the power to exercise it did so in time, and that a machine which reads the rule honestly will tell you that rather than the draw you expected.

PART II · THE MODEL IS NOT THE SOURCE

CHAPTER 6

What the Model Leaves Out



A source preserved as text is not yet a rule that can yield an answer.

Principle of the method

Two questions to the same machine, about the same law.

The law is the Social Code of Kazakhstan, the 2023 code that gathers pensions, disability benefits, family allowances and social assistance into two hundred and seventy-two articles. All of them are in the machine, in the sense that their text is there, pinned byte for byte. Whether the machine can answer a question about them is another matter, and that is the chapter.

First question. A woman has twenty full years in the pension system and asks what her basic state pension is in 2026. The code says: seventy per cent of the subsistence minimum for ten years or less, two per cent more for every year beyond ten, up to a ceiling. The subsistence minimum for 2026 is fixed by the budget law at fifty thousand eight hundred and fifty-one tenge. Give the machine the twenty years and that figure; the figure is yours to supply, and the machine does not check it here. What pension does the formula give?

Forty-five thousand seven hundred and sixty-five tenge ninety tiyn: established, with article 206 beside the answer and the rule that produced it, seventy per cent plus twenty. Ask about a different figure, forty-five thousand seven hundred and sixty-five even, and the machine says not established; it is not splitting the difference, it is computing.

Second question. A family has a member with a severe disability, cannot cope, and asks whether it qualifies for the special social services the same code promises in its articles one hundred and thirty-one to one hundred and forty-one. Ask the machine whether the family qualifies.

Nothing. Not “not established”: nothing. There is no question to ask. Search the model for “special social services” and it returns the text of the articles, eleven of them, and not a single rule. The machine can show you the law. It cannot apply it,

because nobody has yet written those articles into a form it can run.

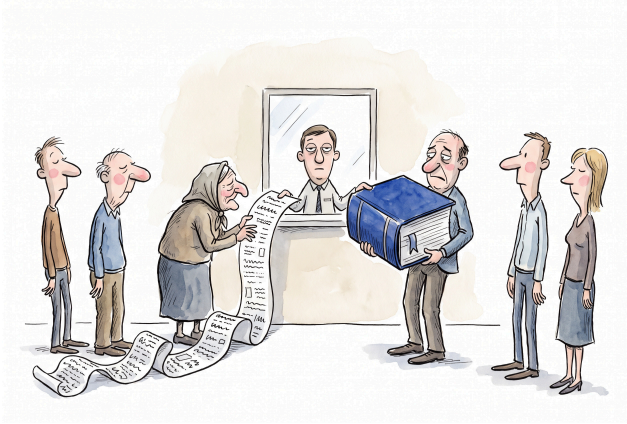
Two silences

From outside, the two answers can look alike.

Both are the absence of a yes. Inside the machine they are as different as a court that heard your case and found nothing for you and a court that has not opened the file.

“Not established” in the first sense is the answer of the earlier chapters: rules ran, and none reached the conclusion. That answer comes with a list of what was missing, and it is a statement about your case under the model.

The second silence is not about your case at all. It is about the model. The article’s text is in the model; no rule applies it; and a machine that let you confuse the two would be lying by omission every time it said “not established”. So the machine does something the earlier chapters did not need: before it answers anything, it tells you how deep the model goes. Its own catalogue line for this code reads, in its own words: *Social Code, executable 180*. At least one executable rule is linked to each of these 180 articles.



One code, one program. For one question there is a calculation, for the other only the text of the law.

The remaining 92 articles have no linked executable rules. This measures formalisation coverage, not a guarantee that every query involving a covered article will produce a conclusion. These counts describe the model snapshot recorded for this experiment.

That number is not a claim the author makes. It is computed from the model itself, article by article, by counting what refers to each one.

Executable rule and model coverage

An executable rule allows a recorded condition to be applied to a case. Coverage shows which provisions received such rules. It certifies neither the completeness of each article's rendering nor the correctness of its reading.

The measure

Each article in a pinned source receives a mechanically derived coverage classification.

Executable: at least one executable rule refers to the article; whether it produces a conclusion depends on the case and question. Article 206 above. This says nothing about whether the whole article was carried over;

180

ARTICLES

At least part of the article is carried by executable rules.

92

ARTICLES

Carried in the model as text only.

Of the 272 articles in the pinned edition. Model state recorded on 5 September 2026; a measure of coverage, not of correctness.

one rule tied to one sentence is enough for this coverage classification. The Regulation on air passengers had all of its articles at this level, which is why the second chapter never met the second silence.

Interpreted: a rule exists, but only under a reading that must be switched on; on an ordinary case the article yields nothing. Gellius's chapter has one such article: the paragraph with the two speeches.

Anchored: something refers to the article, a fact or a closed list, but no rule; the article can be pointed at, and nothing is derived from it. Judge Wood's statement of the facts in the fourth chapter is anchored: the machine can cite it, and nothing follows from it.

Source only: the text is pinned and nothing refers to it. The Social Code's ninety-two are all of this kind, and for all ninety-two nobody has written down why. That is the honest debt of the model, and the measure prints it as a list of ninety-two article headings rather than a percentage.

Excluded: the article is deliberately left out of this edition of the model, and a reason is recorded: repealed, or not yet in force.

And a dash: the article was never brought into the model at all, not even as text. The Laws of Chess in the fifth chapter cover six articles and leave six outside.

The measure is not a grade. A dash is not worse than "excluded", and "anchored" is not a failure. It is a map of where the machine can speak, and the point of it is the sentence the specification forbids: a model must never say it is complete while part of the text is text only. These numbers may change as the model is extended or revised. The requirement to report them remains.

Provenance of inputs

The path from the source to the text, fact or number used: where they were taken from, which edition or date they belong to, and how they were obtained. Checking that path complements checking the calculation itself.

The bytes underneath

There is a second question the machine answers before it answers yours: what text is this model built on, and how was that text obtained?

The Social Code in the model is the Russian consolidated edition as it stood on the twenty-seventh of August 2026, taken from the left column of the official legal portal's comparison page, with the page's hash pinned.

The text is official.

The bytes are not the bytes of the official publication; they are an extraction from a web page, and the model says so, in a field with a name: pinned unofficial copy.

That is the commonest kind in the corpus; the Latin of Gellius, the Leonard opinion and the Laws of Chess are all copies of one kind or another, and the experimental notes has the census. None of this makes an answer wrong. Each kind changes what an answer is evidence of. A model built on official bytes lets you argue that the machine misread the law; a model built on an unofficial copy lets you argue, first, that the extraction departed from the publication, and the machine has handed you the fingerprint to check.

What the author chose

Go back to the subsistence minimum you supplied. The same corpus holds the 2026 budget law, pinned, with that figure in it, and the two are not wired together for this rule: the rule for article 206 takes the amount as a fact of the case, “applicable to

the person”. The author chose to have you supply the minimum rather than have the model fetch it, and a different author might have wired the budget law in. The pension you were quoted is right if the figure you supplied is; the machine did not check that.

Which chapters of the code are executable is a choice too. The author formalised the addressed social assistance, the disability and care benefits, the special professional payment, the basic pension and the contribution rates, and left the rest as text. The package says this in its own header and calls it “movement towards a full act, not a claim that all two hundred and seventy-two articles run”. The measure holds the author to that: an article cannot slip from executable to text only without the change being refused, and cannot be declared out without a reason.

Every figure in this chapter was computed by two programs written independently of each other, and their answers agreed to the byte; the measure itself is counted by a script, not proved by anything, and the experimental notes says how it is guarded.

What is left to argue about

The second silence has an address now. If you are the family in the second question, the machine has told you, before you asked, that articles one hundred and thirty-one to one hundred and forty-one are text only, that nobody has declared them out, and that nobody has said why. You cannot argue with the model about your eligibility, because the model has no opinion, and bringing more documents will not help, because nothing reads them; you can argue with the author about the ninety-two, and the measure has given you the list.

If you are the woman with twenty years, you have a number, an article, a rule, and a hash. What you do not have is the machine's word that the subsistence minimum you gave it was right, or that the extraction from the portal matches the official publication, or that the one hundred and eighty executable articles were formalised correctly. The measure counts derivation, not correctness. That is the eighth chapter's subject, and the eleventh's.

For now the lesson is the one the specification wrote into its own rules: the model may be thin, and it may be built on a copy, but it may not hide either. "Not in the model" and "the law is silent" are different answers, and the first must never be dressed as the second.

PART II · THE MODEL IS NOT THE SOURCE

CHAPTER 7

Who Chose This Reading?



...reimbursement within seven days, by the means provided for in Article 7(3), of the full cost of the ticket...

Regulation (EC) No 261/2004, Article 8(1)(a)

Back to the cancelled flight of the second chapter. The passenger chose a refund rather than re-routing on the first of August. The carrier paid on the eighth. The regulation says reimbursement is due within seven days.

Was the carrier in time? Count it out before reading on. Seven days from the first is the eighth, so most people say yes, and a few say the seven days ran from the cancellation, which was earlier, so no.

The machine says: not established. Not because it counted differently. Because it did not count at all. In the answer's own record of what it applied there is a line for readings, and the line is empty.

Seven days from what

The regulation does not say when the seven days begin. From the cancellation? From the passenger's claim? From the moment the passenger chose reimbursement over re-routing? The text is silent, and at least three starting points can be argued for.

The person who wrote the model did not pick one silently. The rule that counts the seven days from the passenger's choice exists in the model, but it is wrapped in a declaration the model calls an *interpretation*, marked *disputed*, with a label that says, in the machine's own words: the seven-day period has no starting point in the text; it may run from the cancellation, from the claim or from the choice; this formalisation reads it from the choice, because before the choice no duty to reimburse has arisen. The rule is written, but it applies only when that reading is selected. Without the selection the machine does not check the deadline at all.

Select it. Paid on the eighth: established, within seven days. Paid on the ninth: not established; with this reading in force,

a payment on the ninth does not establish that the deadline was met. Deselect it: on either date the deadline is simply not checked. The right to reimbursement itself stands in every one of these answers; only the deadline question moves. And in every answer the record of applied readings names the reading in force, or names none.

So a reading can be in three states. Selected: its assumption is in force for this calculation. Not selected: its rule is not applied, and no other reading is chosen in its place; the question stays open. And baked in: the assumption is always in force, and the only way to change it is to change the model.

Compare the first chapter. There the two speeches were readings of the same kind, switched on and off to produce four answers. But the repair that made the table symmetrical, the condition that the victory must come from an earlier suit, was not written as a switchable reading. It was written into the base, as if it were the text. It was a reading too, and the first chapter said so.

The difference between the two is not in the text.

It is a decision of the author about which of their readings to expose as switches and which to bake in, and that decision is the first thing in this chapter you may dispute. Every baked-in reading is a switch the author decided you did not need.

Explicit and built-in assumptions

An explicit assumption stands out as a separate choice. A built-in one is already enclosed in a definition, a rule's condition or a link between rules. That the first can be changed does not make the second indisputable.

The words the machine uses

There is a second layer of authorship between the text and the answer, and it is easier to miss because it looks like mere translation.

Every rule in a model carries the words it will be described with. When the machine says “the passenger has the right to compensation”, that sentence is not generated; it was written by the author as the rule’s label, in one or more languages, and the machine’s explanations are built from those labels. Read an answer and you are reading the author’s paraphrase of the law, one rule at a time.

Imagine it the other way round: you asked one short question and were sent three pages of explanation. You read them once, then again.

— Түсінбедім. I didn’t understand.

The explanation has been presented. But the reader still cannot say why this answer came out, or what exactly to argue with. The program’s explanations use wordings prepared by people. They name concepts, describe the conditions of rules and explain why an exception applies. Those words may render the calculation exactly and the meaning of the source badly, so an explanation needs its own examination. The author of a rule, the author of its wording and the translator may be different people; a correct calculation certifies none of their sentences.

Kazakhstan’s 2026 Constitution, adopted on 15 March 2026 and in force from 1 July 2026 in the pinned source edition, shows how far this goes. Article 14(1) provides that a citizen may not be extradited to a foreign state unless an international treaty provides otherwise. Its rules carry labels in Russian and Kazakh, the two authentic texts, and further languages are added from outside as translations, each marked as a working rendering by the formaliser and not an official text. Here is one rule, Article 14, paragraph 1, in the authentic Russian and in the English rendering:

*Статья 14 пункт 1. Гражданин Республики Казахстан
не может быть выдан
иностранному государству*

*Article 14, paragraph 1. A citizen of the Republic of Kaza-
khstan may not
be extradited to a foreign state*

These are labels of one rule, not the full text of the article. In the pinned Russian source the ban is followed by a proviso: unless otherwise provided by international treaties of the Republic of Kazakhstan. The model expresses that proviso as a separate rule with priority. Read the first label alone and you can mistake a general rule for an unconditional ban and miss the exception.

So what needs checking is not only the translation against the Russian label, but the explanation as a whole — against the source and the shape of the model. A translation may render the label faithfully while the label, without its neighbouring rule, gives a false impression.

Selecting a reading changes the calculation.



Seven days is the same period. But from which event should it be counted?

Selecting a language changes only the words of the explanation, and cannot change an answer, because the translations are kept out of the part of the model that computes. But an unchanged calculation does not prove that those words render the law faithfully. If the English is wrong, the rule is not wrong, the label is, and the current checker does not establish whether a rule's natural-language label faithfully represents the source text. A translation makes the model checkable by more people. It certifies nothing.

Explanation and source

An explanation describes the rule applied and the conclusion reached. The source is the text the rule was made from. Faithful rendering of the calculation does not yet guarantee faithful rendering of the source's meaning.

How to object

Suppose you have read the label, switched the readings, looked at the articles, and you still think the author got it wrong. What then?

The first thing to notice is that the model has already told you what kind of wrong. A wrong reading is a switch you can flip or a baked-in condition you can point to. A wrong label is a sentence you can quote against the text. A missing rule is a measured gap of the kind the sixth chapter described. Each has an address, and an objection without an address is an opinion.

The public server carries a tool for exactly this. A report names the model and the rule or article, says what you expected and what you observed, and is written to a journal together with the fingerprint of the model it concerns, so that whoever reads it a week later knows which version was being argued with. What the report does not do is change anything. Nothing in the model moves because a report was filed; “not established” does not become “established” because someone objected. A person reads the journal.

That person has two ways to close a report. Either a rule is wrong and gets fixed, with a test behind the fix. Or the rules are right and the documentation was silent, and a note is added beside the rule so that the next reader with the same objection finds the answer already there. Take the objection this chapter has been circling: that the seven days should run from the claim, not from the choice. Filed, it would land with the person who keeps this model, and that person would have to decide, in writing, whether the choice is a reading to keep as a switch, to replace, or to bake in. The machine cannot make that decision. It can only make sure the objection arrives with an address.

There is a third channel, and it runs the other way. A lawyer can mark one rule’s wording, in one language, as checked against the text. That mark is a necessary condition for calling a fragment verified and not a sufficient one, and it lapses when the rule’s meaning or its wording changes. Who may dispute such a mark, and what happens to “verified” when they do, is a question the project has written down as open.

The retained run records are in the experimental notes.

What is left to argue about

Three things, and this time all three are the author's.

The line between readings exposed as switches and readings baked into the base. The seven-day start is a switch; the earlier-suit condition of the first chapter is not; nothing in either text draws that line, and the answer shows only the switches.

The words. Every explanation the machine gives is a paraphrase written by a person, and in most of the Constitution's languages the paraphrase is a working translation that nobody has approved.

An unchanged calculation tells you the program did not change when the language did.

It does not tell you the translation is faithful.

The institution. A channel for objections exists, keeps its records, and changes nothing by itself; a person decides, and the project's own record says that how often reports are confirmed is not yet measured and that the mechanics of disputing a lawyer's mark are not yet built. The machine can make the switchable choices of its author visible; the baked-in ones it shows only to a reader who knows where to look. It cannot make the author answer. The current checker does not establish whether a rule's natural-language label faithfully represents the source text. That work is yours, and the model has been arranged so that you can do it.

PART II · THE MODEL IS NOT THE SOURCE

CHAPTER 8

The Day the Answer Changes



The statement of debt is drawn up as of the calculation date. The day of the calculation date is included in the period over which the penalty accrues. Supply contract No. 7/2026, clause 5.1 (a model contract written for this example; translation)

A supplier wants to know what it is owed. Here is the case, as far as stated so far. One delivery, one million tenge, due on the tenth of June. On the twenty-first of June the buyer paid four hundred thousand. The contract charges a penalty of one tenth of a percent of the unpaid principal for every calendar day of delay, and caps the penalty at ten percent of the original sum.

How much does the buyer owe?

Most people answer six hundred thousand plus a bit. The bit is the penalty, and they start counting days. A few stop before counting and ask the question the supplier forgot to ask: owed as of when?

They are right to stop. This chapter is about that question, and about how much of an answer lives in a date the question never mentioned.

One caution before the numbers. The contract does not exist. Neither do the two companies named in it. The text was written for this example as a model of a supply contract, and it says so in its own first paragraph. What is real is the law it leans on, the Civil Code of Kazakhstan, and the calendar it counts by. Everything below is a test case, and every figure in it comes from the machine's own record.

Three answers to the same facts

The machine does not answer “how much is owed”. It answers “how much is owed as of a date”, and the contract makes it do so: the statement of debt is drawn up as of the calculation date. So I gave it three dates and did not change anything else.

As of the twentieth of June the contract's formula gives one million ten thousand. The payment has not arrived yet, so the

principal is still a million, and ten days of delay at a tenth of a percent is ten thousand.

As of the thirtieth of June the formula gives six hundred and sixteen thousand: six hundred thousand of principal and sixteen thousand of penalty. The payment came in on the twenty-first and the principal dropped to six hundred thousand. The penalty is now two pieces: ten days on a million, ten thousand, then ten days on six hundred thousand, six thousand.

As of the thirty-first of July the formula gives six hundred and thirty-four thousand six hundred. Same principal; the smaller base has now run forty-one days, from the twenty-first of June to the thirty-first of July.

As of	Principal (KZT)	Penalty (KZT)	Total (KZT)
20 June	1,000,000	10,000	1,010,000
30 June	600,000	16,000	616,000
31 July	600,000	34,600	634,600

These figures are contractual calculations, not predictions of a court award. This example contract separately reserves the recoverable penalty to a court; the end of the chapter explains that modelling boundary.

“Қазіп” (*qazir*) means “right now”, though in everyday conversation it can also signal a promise whose exact timing remains uncertain. The supplier has heard it before. As well as “already left” from a man still looking for his keys. So now he looks at the time on the bank statement. A promise to transfer the money and its crediting are different events.

Three answers, one set of facts. None of them is wrong, and none of them is the answer.

Each carries its date in the first line of the record, and the machine will not let the date fall off.

The supplier who writes “the buyer owes 616,000” in a letter has quietly dropped the part of the answer that made it true.

Look at how the penalty is written. Not as one number but as periods. For the eleventh to the twentieth of June, the base is a million. From the twenty-first, the day the money landed, that day included, the base is six hundred thousand. Each period lists its days, its base, its rate and what it produced. That is not decoration. It is where a dispute would go: the buyer who says the money was credited on the nineteenth is arguing about the boundary between two rows, and can point at it.

Calculation date and event history

The calculation date sets the moment for which the answer is needed. The event history shows when its grounds changed. Debt and penalty must therefore be viewed by periods, not only by the final balance.

The day the answer stops changing

Push the date out to the end of the year. As of the thirty-first of December the penalty on the smaller base has run for a hundred and ninety-four days and comes to a hundred and sixteen thousand four hundred. With the first ten thousand that is a hundred and twenty-six thousand four hundred. The contract caps the penalty at ten percent of the original million. The record says: cap reached, penalty a hundred thousand, total seven hundred thousand.

By that date the cap has been reached, and from whatever day it was reached, the calculation date stops mattering to the

penalty. The debt still stands and the delay still runs, but the penalty no longer grows. The record shows the cap as its own row, before and after, so a reader can see that it was applied and where.

I want to be exact about what the machine decided here and what it did not. It did not decide that ten percent is a fair cap. The contract said so, and the model carries that clause under its address. It did not decide that half a tiyn rounds up: that comes from clause 3.5, whereas the cap comes from clause 3.4. The rounding rule does not affect the example totals, but becomes relevant when the daily rate produces fractions of the smallest currency unit. The machine's contribution is that these choices were applied to these facts in this order and nowhere else, and that two independent implementations produced the same bytes when they did it.

What the machine will not assume

Now the part I found most instructive.

The contract has a clause that most readers skip. The statement is drawn up on the full payment history, and the party drawing it up confirms in writing that the history is complete as of the calculation date. Without that confirmation, the contract says, the final penalty and the final debt are not determined.

The person who wrote the model took that clause seriously, and this was a choice. The confirmation is a fact the machine expects to be told. Tell it “not confirmed” and the record changes character. The periods are still there, ten thousand and six thousand, computed and shown. The totals are blank. The status line says the data is insufficient, and below it the machine names what is missing: the confirmation of completeness, not established. It does not guess that the payments it was given are all the payments there were.

The question of completeness must be answered, one way or the other. “Cannot confirm” leaves the calculation provisional, as above. No answer at all, and the program does not accept the calculation: it stops before any rule runs, and the reason it

gives is the clause. That is a separate choice of the author's, and worth naming: the program demands an explicit answer even from a person who cannot confirm completeness. Silence, on this reading, is not the same as "no other payments", and the difference is shown at the door, not buried in a total that looks finished.

That is the whole contrast. Not that the machine is cleverer than a calculator, but that it was built to refuse a particular shortcut and to say which shortcut it refused.

Completeness of the payment history

Confirmation that all payments material to the calculation as of the chosen date are recorded. In the study contract this is a separate premise: without it, the known accruals do not determine the final debt.

The payment that did not count

One more variation, and this one caught me.

Back to the thirtieth of June and the same payment, four hundred thousand on the twenty-first. But the payment document does not say which obligation the money is for. There is only one obligation in the contract, so surely it is for that one.

The formula now gives one million twenty thousand. The principal is still a million. Twenty days of penalty on a million, no second period, no drop. And a warning at the bottom: a payment was not allocated to any obligation and did not reduce the principal.

The contract is explicit. The buyer states in the payment document which obligations the money goes to, and a payment allocated to none reduces the principal of none. The model's author could have added "unless there is only one". They did not, and they said why: the rule of allocation belongs to the buyer, and a calculator that fills it in for the buyer has changed who decides.

Whether that is the right reading of the contract is arguable. What is not arguable is that the record shows the choice. The

payment is listed, dated, and marked as unallocated in a column of its own. The buyer who disagrees knows exactly which line to fight about, and the fight is about a clause, not about arithmetic.

Computed sum and sum awarded

A computed sum follows from the given formula. The sum awarded also depends on legal conditions and judicial assessment. An arithmetical result alone does not prejudice the court's decision.

The number that is not the answer

Every one of these records ends the same way, and it is the boundary that matters most.

The model calculates the penalty specified by the contract. If the dispute reaches court, Article 297 of Kazakhstan's Civil Code allows the court, at the debtor's request, to reduce a penalty that is excessive in relation to the creditor's losses. The contractual calculation and the amount ultimately awarded must therefore be distinguished. In this specially written example contract, clause 6.3 reserves the recoverable amount to the court. That clause, rather than a general requirement that every penalty be judicially fixed, explains why this model records a question addressed to a court. The computed penalty and the awarded penalty have different names, and no rule turns one into the other.

This is the answer to the question a reader of this chapter should be asking by now. Is the number the machine produced the number you will get? No. It is the number the contract's formula produces from the facts you gave, as of the date you gave, on the reading of the contract the model's author chose.

The court's number is not in the model.

The model knows it is not, and says so on every run.

A day that moves on its own

Money changes with the date. Some dates move on their own, with the calendar, and one short contrast on the same machine shows the difference.

The Civil Code counts a period in days from the day after the event, and when the last day falls on a non-working day, the period ends on the next working day. Five days from Monday the sixteenth of March end on Saturday the twenty-first. In 2026 the twenty-first, twenty-second and twenty-third of March are the Nauryz holidays; the twenty-fourth and twenty-fifth are days of rest which the law on holidays moves to sit next to them because the holiday fell on a weekend; the twenty-sixth is the first working day. The machine walks forward through all of that and returns Thursday the twenty-sixth, marking that the day was moved.

The basis matters as much as the date. If a holiday law itself moves a day of rest, citing a separate decree would give the date the wrong explanation even if the calculation stayed unchanged. Checking that an answer follows from a model does not check that the model follows from the law. The source of each calendar rule therefore belongs beside the result, where a reader can examine it.

The retained run records are in the experimental notes.

What is left to argue about

The record of the penalty has five places where an argument can stand, and they are all visible.

The date. It is the first line, and changing it changes the answer.



The feast is over, the promise remains. The calculation needs the dates and amounts of the payments received.

Anyone who quotes the total without it has misquoted.

The confirmation. The supplier who signed that the history is complete has vouched for the record and answers for that confirmation. If a payment turns up later, the statement was not wrong in its arithmetic; it was wrong in a premise that has a name.

The allocation. Whether an unallocated payment reduces the only obligation is a question about the contract, and the model's author answered it one way and showed the answer.

The cap. Ten percent, applied after the periods and before the rounding, in a row of its own.

The court. Whatever the total, it is not what a court will award, and the record says so beside the total on every run.

The supplier asked how much it was owed. The honest answer is a sentence, not a number: as of the thirtieth of June, on a payment history you have confirmed complete, on the reading that an unallocated payment counts for nothing, the contract's formula gives six hundred and sixteen thousand, and a court will decide the penalty part. The machine wrote that sentence. It just wrote it as a table.

PART II · THE MODEL IS NOT THE SOURCE

CHAPTER 9

One Case, Five Legal Traditions



En fait de meubles, la possession vaut titre.
Code civil des Français, 1804, article 2279

Picture this. Nurlan is going away and leaves his dombyra with an acquaintance for safekeeping. The acquaintance sells it to a man who knows nothing of the arrangement. Coming back, Nurlan finds the instrument and demands it back.

“This is my dombyra.”

“And I paid for it.”

The acquaintance, who could explain the situation to both, is staying silent for now.

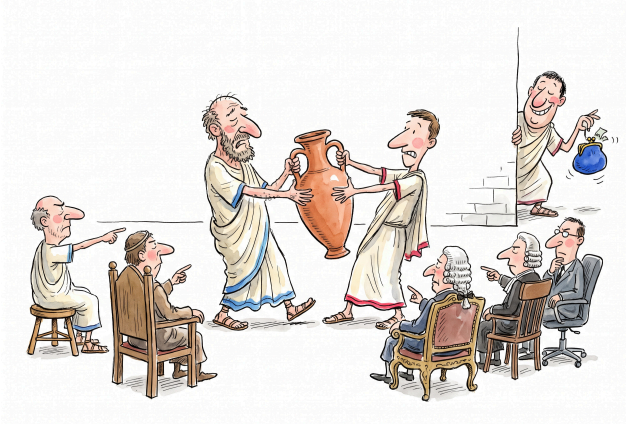
Behind this invented exchange stands an old question: whose claim should win, when the owner never agreed to the sale and the buyer knew nothing of another’s deceit? Now the legal case, as we will compare it across sources.

Titius owns a thing. He leaves it with Seius for safekeeping. Seius sells it to Maevius, who pays the price and has no idea that Seius is not the owner. Titius finds out, finds Maevius, and demands the thing back.

Who keeps it?

Titius has a plain argument: he is the owner and never meant to sell. Maevius has his own: he is an honest buyer, he paid, and he should not answer for another’s deceit. One of them risks walking away with nothing, though both were deceived by Seius.

Try to pick a side before reading on.



Titius and Maevius pull the thing each toward himself, Seius walks away with the money. Judges of different traditions point in different directions.

Both groups have the law on their side. That is the whole chapter. The same case, in the same words, was put to six bodies of law from five legal traditions, Rome counted twice at two dates four centuries apart, and they split three against three. The split is not old against new: the Ottoman code of the 1870s sides with Gaius, and the French code of 1804 sides against him.

The names are Roman because the case is. Titius, Seius and Maevius are the placeholder names of the Roman jurists, and the comparison keeps them because every one of the six bodies of law has a place for a case told in these terms.

"Effective" right

A working term for comparison: the model gives the claimant a remedy and does not take it away. It is a shared question to the chosen models, not a substitute for each legal tradition's concepts.

The same words

An argument over “Almaty or Astana?” can run for hours. One talks about the weather, another about work, a third about the city where his friends live.

“Let us first agree what ‘better’ means.”

“You just ruined the argument.”

For comparing legal traditions this step is unavoidable: we must agree on the exact question we expect answered.

Here is what makes the comparison possible, and it was a choice.

Each legal order has its own word for the thing a person holds when the law will back them. The Roman jurist has an *actio*. The Ottoman Mejlle has a *da'wa*. The French code has an *action*, the German an *Anspruch*. The Russian-language version of Kazakhstan's Civil Code uses *isk* (иск); Kazakh terminology uses *talap* (талап) and *talap qoyu* (талап қою), depending on context. These terms belong to different theories of what a claim or remedy means; putting their answers side by side requires an explicit comparison.

The model's authors built one shared vocabulary and made every code speak into it. The vocabulary is Roman in shape: a right is called effective when the legal order gives a remedy and does not take it away, and every code is asked the same question, whether Titius's right against Maevius is effective. The other outcomes the vocabulary knows will be explained when they appear.

These mappings serve this particular comparative question. They do not assert complete semantic equivalence between the legal traditions' terms.

The vocabulary itself decides nothing. That is not a slogan; it is measured. Every case is also run against the vocabulary alone, with no code attached, and the expected answer there is always "not established". If someone ever added a rule to the shared vocabulary that derived an answer from the bare facts, the vocabulary would have started judging by itself, and that empty run would turn colour. It has not.

Three yeses, three noes

Now the six answers to the case of the entrusted thing, in the order of their texts' dates, which is not the order of their answers, each with the sentence it rests on.

Gaius, writing around the year 161. The owner may demand the thing back from whoever holds it. Good faith in the buyer is not something book four of his Institutes knows as a defence, and a possession of zero years cannot ripen into ownership. Titius wins.

Justinian, 533. Delivery transfers ownership when it comes from the owner; “a thing delivered by its owner is alienated”, and Seius was not the owner. Nothing has been held long enough to matter. Titius wins.

The Mejelle, the Ottoman civil code of 1869 to 1876, read as of 1900. A sale of what belongs to another is suspended until the owner approves it. Without that approval the buyer holds another’s property without lawful cause and must return it. The buyer’s honesty is not a category the code uses here. Titius wins.

The Code civil of 1804, read as of 1810. “In the matter of movables, possession is as good as title.” The honest possessor who got the thing from a non-owner is protected, and the owner may reclaim it only if it was lost or stolen, and then for three years. Titius handed it over himself. Maevius keeps it.

The German civil code of 1896, in force from 1900. The acquirer becomes owner even if the thing was not the seller’s, unless he was in bad faith at the time, and unless the thing had gone astray from the owner: lost, stolen, or otherwise out of his hands against his will. Entrusted is not against his will. Maevius keeps it.

The Civil Code of Kazakhstan, read on the second of September 2026. The owner may reclaim from a buyer who paid and did not know of the owner’s right only when the property was lost, stolen, or left the owner’s possession against his will. It left with Titius’s consent. Maevius keeps it.

Look at what the three modern ones have in common. Not good faith on its own. The hinge in each is whether Titius parted with the thing willingly. Change one fact, make Seius a thief instead of a custodian, and the model’s fifth case gives the answer: all six bodies of law hand the thing back to Titius, the French and the Kazakh included. Eighteen centuries agree about

theft. What they disagree about is a much narrower thing: who bears the loss when an owner chooses badly whom to trust.



One has the owner's argument, the other a payment confirmation. The seller left them to settle it between themselves.

That sentence is not the machine's. The machine produced six cells and six article numbers. The pattern across them is mine, and a reader who sees a different one has the same cells to argue from.

Loss of right and loss of remedy

Acquisition of a thing by another through lapse of time may end the former ownership. Limitation of claim protection may operate while the right survives. Years of possession and the time for defending a claim must therefore be told apart.

Two kinds of no

Time is the second thing the registry varies, and here the vocabulary earned its keep in a way I did not expect. What follows is a separate case, not a continuation of the entrusted thing; its facts are set afresh.

Maevius has held a movable thing honestly, on a lawful basis, for one full year, and only then does the former owner, Titius again, come for it. Under Gaius a year of possession of a movable makes the possessor owner; the Twelve Tables said so. Under Justinian it takes three years. Under the German code ten, under

the Kazakh five. The *Mejelle* has no such thing as ownership by lapse of time at all; what it has is a rule that a claim is not heard after fifteen years. And the French code needs no year, because possession was title from the first day.

So after one year, under Justinian, the German code, the *Mejelle* and the Kazakh code, Titius can still demand the thing back; under the French code he cannot. In Rome under Gaius the machine says something else: not established.

Not “refuted”, and the difference is the point, though it takes a sentence to state without the vocabulary. The model derives no claim for the former owner, because by its rules ownership has already passed to the possessor. That is a different situation from the one in which the owner keeps his ownership but cannot obtain the return. To the person asking “do I get it back?” both are a no; the grounds differ. The vocabulary keeps them apart by a choice of its authors: “refuted” is reserved for a remedy that was given and then taken away, and the absence of any claim is recorded as “not established”. A convention to know: the same word also appears when facts are simply missing, so the cell’s note, not the word alone, tells you which it is.

Another separate case: land, held for thirty years. Now Gaius, Justinian, the French and the Kazakh codes all say the same “not established”: the land has changed hands by the passage of time and the old owner has no claim. The *Mejelle* and the German code say “refuted”, and mean the other thing by it. The old owner is still the owner. His right is intact. But the claim has become one the court will not hear, or one the possessor may refuse to satisfy. A right with no remedy; the vocabulary calls it naked.

Whether that distinction matters depends on what happens next. The model has a case for that too. If the debtor pays a debt whose time has run, can he get the money back? Every modern code in the chain says no, each in its own article: what was paid on a naked right stays paid. The right was real; only the remedy was gone. A reader who thought “prescribed” meant “extinguished” has learned, from a table, that four legal orders disagree with him and say so in writing.

How the table was made honest

A table like this is easy to fake. You decide what each code should say, then adjust the model until it says it. The methodological rule against that is simple and was followed: the expected answer in every cell was written from the code's text, with the article it rests on, before the run. A cell that comes out wrong is a subject for analysis, not a reason to change the expectation.

A possession rule needs the conditions under which possession counts as title. A model that records the protection but omits the qualifying purchase can give a different answer from the source. The expected answer should stay tied to the code while the missing condition is examined. Otherwise a table can appear consistent simply because its expectations were adjusted to match it.

The machine answers from the model. Keeping the source, the chosen reading and the expectation distinct makes the table open to scrutiny.

Two more things the table keeps in view rather than hiding: some cells are marked skipped, with a reason, rather than answered, and the published table carries the date of the models it was made from. The experimental notes have both.

What the machine did not do

It did not compare. It ran six models on one set of facts and wrote six records. Nothing in the machine knows that the six are related; the table is a reader's arrangement of six independent answers, each with its own proof.

It did not read the codes whole. Each model covers eight blocks chosen by the modeller: persons, things, possession, ownership, obligations, delicts, intestate succession, remedies. The Code civil has over two thousand articles and the model carries a few dozen. That is declared, block by block, as a boundary, and the boundary is part of the record.

It did not read the current French or German law. The models are the first editions, 1804 and 1896, because the chain was built

as history, and that was a choice with a name on it. Anyone who wants the modern articles has to build them as later editions of the same acts, and the machine will then show what changed and when.

It did not translate the *Mejelle* from the Arabic or the *Institutes* from the Latin for you. Every model is built from the pinned original text, and the notes in the cells are the modeller's reading of it, marked as such.

Where would you rather be Titius

Back to the thing on the table.

Where would you rather be Titius? Under Gaius, or under any Ottoman judge applying the *Mejelle*. Where would you rather be Maevius? Under the Napoleonic code, where possession was title before he had finished counting out the price.

The machine says nothing about which is right.

It says: here are six answers, each with its article; here is the one fact that flips three of them, whether Titius handed the thing over himself; and here is what none of the six will say, because none of the six models contains it: a reason. The reason is in the history of trade, in the eighteenth century's need for goods to change hands without a title search, and it lives in books the machine has not read. What changed in 1804 was not who counts as honest. It was whose carelessness the law decided to charge.

The table shows the change.

The reason belongs to a history book, and this chapter was never going to be one.

PART II · THE MODEL IS NOT THE SOURCE

CHAPTER 10

A Counterexample Before an Amendment



“No, brother, you won’t fool me this time. Give me the roots, and take the tops yourself.”

“Tops and Roots”, Afanasyev, No. 24 (translation)

A peasant and a bear sow a field of turnips together. In the model used here, the peasant chooses what to plant: the bargain specifies how the harvest is divided and places no constraint on his choice of crop. When the crop is in, the peasant proposes the split: roots to me, tops to you. The bear agrees, takes a cartload of leaves, and discovers over dinner what a turnip is. He has been had, and he says so.

Next spring they sow again. The bear has learned. Before the sowing he states his terms: this time the roots are mine. The peasant agrees. Stop here. The peasant chooses what to sow. Which crop lets him leave the bear the worthless part again?

You probably have it. The field is wheat. The bear gets the roots of the wheat, which are worth nothing, and the peasant takes the grain.

The Grimms tell a version with a devil instead of a bear. What I want from you is not the moral but the drafting. Imagine that you have seen how the bear’s second bargain will end. You are now sent back to the spring before he signs it, as his lawyer. What should you advise him to change? Some say “the bear picks the crop”, others “split by value, not by part”. Hold your answer. This chapter is about what a machine can and cannot tell the bear before he signs.

The model treats the bargain, not the tale, as its subject. There is no court in the forest and no legislature; there is an agreement between two parties, in two editions, taken from the version of the tale in Afanasyev’s collection. Two conditions of the model should be said out loud. Only the turnip root and the wheat grain count as valuable; every other part is worthless. And the crop is chosen by whoever sows it, which the agreement does not constrain. Everything below is a run on that agreement, and the lesson about amendments would hold for a contract with real money in it.

Two editions, two crops, four runs

The model has two editions of the agreement. In the first, the bear is allotted the tops. In the second, after his amendment, the roots. Each edition has a season. The crop in each season is a fact the parties supply.

That gives four combinations, and the modeller ran all four, though the tale tells only two.

	Turnips	Wheat
First edition: tops to the bear	worthless part to the bear (as in the tale)	valuable part to the bear
Second edition: roots to the bear	valuable part to the bear	worthless part to the bear (as in the tale)

Read the square and the tale’s usual moral falls apart. It is not true that the bear loses whatever he does. Under each edition there is a crop that would have made him the winner. He lost both years because a second variable, which crop to sow, is bound by nothing in the agreement and stayed with the peasant.

The amendment moved the variable the bear could see and left the one that decided.

Two of the four runs are counterfactual, and the machine does not mind. It has no notion of what happened; it has an edition, a season’s facts and a question. That indifference is the point. The bear could have asked the second-edition-turnip question before the second spring, and the answer would have told him that “roots to me” wins only if the peasant sows turnips again, which the peasant was never going to do.

Counterexample

An admissible case in which the stated guarantee fails. To rebut the promise "always so", one such case is enough; proving constant failure is not required.

Not a single breach

Now the part that makes the tale a loophole and not just a bad deal.

Ask the model about the duty in each historical year: did the peasant deliver to the bear the part the agreement allotted him? Both years, the answer is that the duty was satisfied. Not "not yet breached". Satisfied. The tops were delivered in year one and the roots in year two, to the

letter. Zero violated obligations, and the bear's purpose defeated twice.

That combination is the machine's working definition of a loophole: a path to a goal along which no obligation is ever violated. And the loophole lives in a rule the agreement does not contain, which is why no amount of reading the rules it does contain would have found it.

The bear's response in the tale is not a lawsuit. It is "don't come to my forest for firewood". The model has that too, and the way it has it is instructive. Whether the bear was deceived is not derived from any fact; it is a judgment, and the authority who makes it is the aggrieved party himself. Ask whether deceit is established and the machine returns a request for judgment addressed to the bear. Supply the bear's judgment and the sanction follows, with zero violated norms still on the record. The modeller's note on this is one line: "the victim as his own judge is what self-help looks like when you write it down." A legal order that lacks a court is not a gap in the model. It is a property of the forest.

Three tools, two honest failures

Three different questions can be put to the machine about change, and they should be kept apart before the answers arrive. First, compare the two editions of the agreement as they stand: what did the bear's amendment actually change? Second, let the machine alter the rules on the bear's behalf: is there a small amendment that would have won him the crop? Third, keep the rules fixed and search the peasant's possible actions: is there a way for the peasant to reach the valuable part without breaking any rule? Note whose goal each one serves. The second looks for the bear's win; the third looks for the peasant's. I ran all three. One succeeded. Two failed, and both failures are the kind you want.

What the amendment changed. The machine compares the two editions across the four combinations and reports that the outcome changed in all four. That surprised me, because in two of the four the short answer, whether the bear gets the valuable part, was “not established” both before and after. What changed was not the short answer but the list of obligations in force. The duty to deliver the allotted part exists in exactly one edition for each year of the tale: in the first-edition turnip year it is there before the amendment and gone after, and in the second-edition wheat year the reverse. The amendment did not just move the harvest; it moved which obligations exist at all on a given set of facts. A comparison that looked only at the short answers would miss those changes. “Unchanged” means unchanged in the properties actually compared; the obligations in force must be among them.

The minimal amendment. Ask the machine for the smallest change to the agreement's rules under which the bear gets the valuable part in the second-edition wheat year, keeping the other three combinations as they are. The answer is that no such change exists among the changes the search is allowed to make, and it tried all of them. This search cannot add conditions. It can only remove rules, reorder them, or drop a date. The amendment the bear needed was an addition: bind the crop to joint consent,

or split by value, or name the crop in the deal. Among the changes allowed to it, the needed amendment was not there, and the machine says so by failing completely rather than by offering something plausible.

The loophole search. Ask the machine to find a sequence of actions that gets the peasant the valuable part in year two with no obligation violated. It did not find one, and it failed twice, for two reasons that are worth more than a success. The first time, the action that actually performs the division had been left out of the set of actions the search was allowed to try, and the machine said so: its verdict was not “not found” but “the search was cut short before it began”. The second time, with every action allowed, the sequences it tried were too short: the division in the tale takes several steps beyond what the search started from, and it searched two. A negative result from a search is a claim about the space that was searched, and this one now says exactly which space. The loophole itself is shown by the four squares, not by the search.

Completeness of case coverage

For a general conclusion, all admissible variants material to the claim must be covered. If only some were checked, the absence of a counterexample found does not yet confirm the guarantee.

What the counterexample buys

The bear amended without a counterexample. He asked himself “what went wrong” and answered “I got the tops”, so he asked for the roots. The question he did not ask was “under my new terms, is there a crop where I still lose?”



The bai checks the roots against the agreement. Aldar carries off the grain: the condition is met, and the bai's purpose stays outside the contract.

That question has a mechanical answer, and the answer was yes before the ink dried.

That is the chapter's whole claim, and it is modest. A machine that holds the agreement as rules can be asked about the world you have not seen. Its amendment search cannot add conditions, and among the changes allowed to it the clause the bear needed was not there. Its loophole search did not find this loophole, first because the deciding action was left out of the search and then because the sequences it tried were too short. It cannot say whether anyone was deceived; it hands that to the bear.

What it can do is run the four squares before the bear signs the second one, and one of those squares is the counterexample.

“Roots to me” was the right sentence in the wrong contract. The machine would not have written a better sentence. It would have shown the bear, a spring early, that the sentence he had written was not the one that mattered.

PART III · PROOF IS NOT TRUTH

CHAPTER 11

What the Checker Actually Proves



The report separates “proved” from “admitted by assumption” on purpose: a checker that silently accepts what it has not checked is worse than none.

Header of the checker’s main module (translation)

Every answer in this book came with a proof attached. The proof is a graph: leaves that point at facts, steps that name a rule and the values substituted into it, a root that states the answer. I have shown you those graphs as tables and told you what they contain. I have not yet told you who reads them.

Here is the question the chapter turns on. Suppose someone hands you a proof graph and says the machine produced it. What would it take for you to believe it, without trusting the machine?

Most people’s first answer is “run the machine again and compare”. An independent implementation makes that comparison stronger, but agreement is not proof: two readings can share a mistaken premise or the same omission. This is a principle of the method, even when every comparison agrees. The second answer is the subject of the rest of the chapter. You believe the proof if something that is not the machine, and knows nothing about how the machine works, checks every step against the rules and accepts.

That something exists. It is a third program, written in a language built for proofs, and the interesting thing about it is not that it accepts. It is how carefully it says what it did not check.

The second witness

Two separately written programs can be asked the same question under the same rules. Compare each answer with the other and with an expected answer written from the rules before either program runs. A disagreement then points to a question that needs investigation: did one program misapply a rule, did the expected answer misread it, or was the rule itself ambiguous?

Independence matters. Making one program copy the other removes the second witness. The expectation needs its own

grounds too: it should come from the source and the chosen reading, rather than from whichever answer happens to be convenient.

Consider the remainder when minus seven is divided by three. Under a Euclidean convention it is two; a convention that truncates the quotient toward zero gives minus one. A comparison becomes meaningful only after the intended convention has been stated. Agreement checks the application of that convention, not the wisdom of choosing it.

Agreement means two independent readings of the same text landed in the same place. That is useful evidence, and it is not proof. Shared premises remain shared premises however many programs repeat them. For proof there is a third witness.

Derivation certificate and checker

A certificate is a document presenting the grounds and steps of a derivation. The checker verifies them against the rules and the case record. The report shows what was confirmed, admitted as an assumption, or left outside the check.

The third witness

The checker is small. Small enough to read in an afternoon, and written in a language made for checking proofs rather than for running programs. It imports nothing outside itself: no mathematics library, not even the proof assistant's own standard tools for reading data, because pulling those in would make the trusted base as large as the tool. The specification asks for a small deterministic checker, and this one is.

It does not run the law. It reads three things: the rules, the case, and the answer with its proof graph. Then, for each node in the graph, it asks one narrow question. For a leaf: does it point at a fact that is really in the case or in the rules, word for word? For a step: does the named rule exist, and does the step's conclusion follow from its premises by exactly that rule? For the whole graph: does every reference land on a node that exists, and does nothing rest on itself?

Those questions are answered by a function, and the formal core comes with a theorem checked by the proof assistant. Under its CertOK condition, conclusions follow from the formal rules and supplied base facts. The operational checker can also accept steps by explicit admission: those steps remain assumptions, and accepting a certificate does not discharge them. The theorem’s proof has no sorry gaps; this does not mean that every accepted certificate is assumption-free. The experimental notes record the theorem’s premises and its logical dependencies.

Read that carefully, because it is easy to hear more than it says. The theorem does not say the answer is right. For the steps it verifies, it checks inference against the specification’s definition of “follows”; admitted steps retain their assumptions.

If the rules are a bad model of the statute, the certificate is a valid proof of a bad model.

If a fact was accepted that should not have been, the proof stands on it. The checker certifies inference, and only inference.

What it says it did not check

Every line of the checker’s report says one of two things about a node. The checker checked this step. Or the checker admitted this step without checking it, and here is why. The commonest reason is that the step is of a kind the checker cannot check yet. That is a reason for admitting, not a third kind of result, and the reader should hold on to the distinction:

an admitted step is not wrong, it is unexamined.

I ran the checker on two certificates on the day of this chapter, in its verbose mode, where it prints every admission it was forced to make.

The first was the simplest case in the collection: a fact asserted, and the question whether it is established. Two nodes. One leaf checked, one query checked, no admissions. The checker said it had verified the certificate by its proven function, and it had.

The second was a remainder case, minus seven modulo three. Three nodes. One leaf checked. Then a line I want you to see: the rule step is admitted, not checked, because its conclusion contains an arithmetic term, and the checker's current version does not do that arithmetic. And a second line: the query's "established" rests on an unchecked step. The checker still said OK, but the OK now carries two named admissions. Successful validation is not necessarily an assumption-free proof: readers must inspect both the verified steps and the admissions.

On the evidence case of chapter three the report is longer, and one line in it deserves a plain translation. For each document that became a fact, the checker says that it confirmed which document the fact refers to, and that whether the fact was rightly read out of the document is not something it checks. The reading of documents is outside the core, and the checker says so eight times, once per document.

That is the design, and the sentence at the head of this chapter is its reason. A checker that prints "OK" and nothing else is a checker whose OK cannot be read. This one prints "checked" and "admitted" in separate columns, and on real acts the second column is not small: about a third of the rule steps in the corpus's certificates are admitted rather than checked. Most of those are rules marked as defeasible. Applying such a rule produces a candidate, not a conclusion; the candidate may still be defeated. The checker reads the rule's strength and refuses to count the step as checked. The distinction matters: a formula for a health-insurance contribution can be arithmetically right while the obligation it computes remains presumptive, subject to an exemption the arithmetic does not establish.

Forging a certificate

A checker that never says no is decoration. So the method includes a set of forged certificates, a premise removed, a fact altered, a conclusion swapped, a reference into nowhere, and the check that runs the checker over the whole corpus also runs the forgeries and requires each to be rejected.

I made one of my own by hand, on a certificate from the corpus with a genuine rule step in it, to see a refusal rather than read about it. The certificate has four nodes: a fact, a rule step, a query, and one node outside the core. Genuine, it passes. I deleted one premise from the step and ran the checker again. Refused: the step's conclusion no longer follows from what is left.

The refusal is the product. A proof graph that a third program will reject when one premise is deleted is a proof graph you can hand to an opponent.

A graph must contain its grounds

Take a document that computes a total. Its root cites intermediate steps as premises. If those steps are absent, the certificate has not supplied the grounds it asks us to accept. Two programs could emit that same incomplete graph and agree exactly. The comparison would establish their agreement; it would not supply the missing steps.

A checker asks a different question. Every premise reference must lead to a node in the certificate, and the chain must not rest on itself. Closure and freedom from cycles are separate requirements: a graph can have no cycle and still point into nowhere. Both must be checked.

Likewise, a list of supporting facts must have the form the certificate contract requires. If that contract treats the list as a set, duplicates or a noncanonical order are grounds for refusal. A refusal tells us which requirement was not met; agreement between producers cannot override it.

What it does not know

The boundaries, since the checker names them itself and I would be misrepresenting it to leave them out.

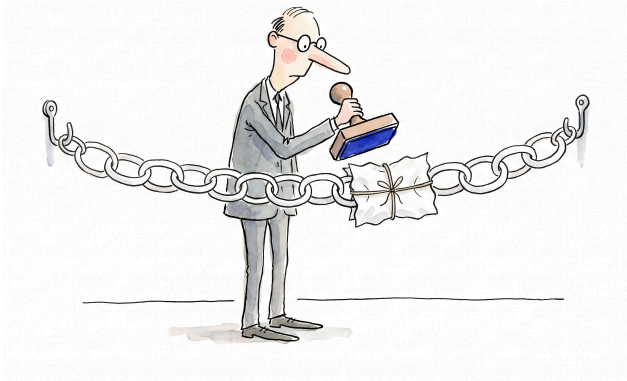
It does not do division, or sums over collections. It does do multiplication, rounding and comparison of exact decimals, with arithmetic of its own, so that a computed contribution is recomputed rather than trusted.

It does not prove that a defeasible conclusion survives. It recomputes the set of defeats and compares it with the document, rejecting a missing defeat and an extra one alike, but a surviving candidate is still a candidate. Conditional priorities between rules it does not attempt.

The checker validates individual derivations but does not independently establish that the engine has found every derivable conclusion. Certifying the absence of support would require an additional completeness argument. “Refuted” reports support for the opposite proposition; checking that support is different from proving that no positive support was missed.

It does not prove completeness. That every derivable fact has a certificate is a property of the engine; the checker’s job is to be strict, not to be generous.

And it does not read the statute. Nothing in the third witness knows what a turnip is, or a tenge, or a good-faith purchaser.



The checker certifies the chain of reasoning. A link taken on trust it does not unwrap.

It knows rules, substitutions and what follows from them, and it tells you, one line per node, where its knowledge stopped.

Soundness of inference and truth of premises

A sound derivation follows from the given premises by the rules of reasoning. That does not yet prove that the initial records are trustworthy and the rules render the source faithfully.

The proof is not the truth

This is the sentence Part Three is named for, and the checker is where it stops being a slogan.

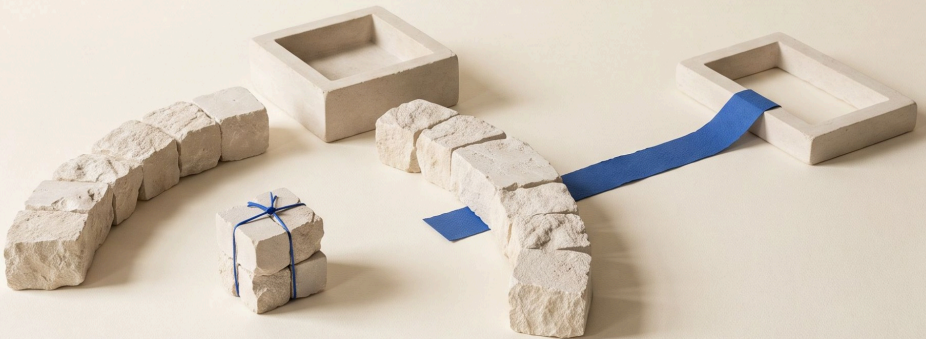
A certificate that passes the checker establishes exactly this: for every step the checker checked, the conclusion follows from these rules and these facts by the specification's definition of following; and for the steps it admitted, it has told you it admitted them, and why. That is a great deal more than "the machine says so". It is a great deal less than "this is the law".

Between the two lies everything the earlier chapters were about: whether the rules are the statute, whether the facts were admitted rightly, which reading was chosen, what the model leaves out, what the court will decide. The checker cannot see any of it, and its honesty consists in never pretending to. When you hold a certificate that passed, you hold a proof of the steps it checked and a list of the steps it took on trust. Whether you hold the truth is the question the proof was never asked.

PART III · PROOF IS NOT TRUTH

CHAPTER 12

A Computed Obligation, an Unfinished Dispute



10. *Withdrawal of all foreign armed formations, military equipment, as well as mercenaries from the territory of Ukraine under monitoring of the OSCE. Disarmament of all illegal groups. Package of Measures for the Implementation of the Minsk Agreements, 12 February 2015 (translation from the Russian text)*

The last case in this book is a document of thirteen points, signed in Minsk on the twelfth of February 2015 by the members of a contact group and attached, five days later, to a resolution of the United Nations Security Council. It is short. You can read it in ten minutes. Its second point is arithmetic, its tenth point raises a consequential question of interpretation, and the distance between the two is where this chapter lives.

One rule before anything else. Nothing in this chapter says what happened in eastern Ukraine after February 2015. The model holds the text of the document and nothing about the world; the two courses of events I will run through it were written by the modeller as constructions, not as history, and the model keeps every disputed fact outside itself. That is a choice, made in the package's own header, and I am repeating it here so that no sentence below can be read as a claim about the ground.

A distance you can compute

Point two orders both sides to pull heavy weapons back to equal distances, to create a security zone: fifty kilometres wide for artillery of caliber one hundred millimetres and more, seventy for multiple rocket launchers, a hundred and forty for four named long-range systems.

Here is the reader's question. A howitzer of exactly one hundred millimetres. Which zone? And a gun of ninety-nine?

The machine says fifty for the first and, for the second, nothing. Not zero kilometres. No zone, no duty to withdraw it, no answer, because the document's definition of heavy weapons begins at a hundred and the gun is not in it. The modeller could

have written a fourth row for “everything else” with a distance of zero, and refused, with a reason: the document names no such distance, and a zero would be a fact invented to make the table look complete.

What the table does contain is four rows, and the machine proves two things about them before it uses them: that no two rows can fire on the same weapon, and that together they cover every weapon the document calls heavy. Not “the author checked”. Proved, with an error code for each way it could fail. The distances then come out of the rows as conclusions: fifty, seventy, a hundred and forty, a hundred and forty. Each is a computed obligation in the strict sense. Give the machine a weapon’s class and it gives you the number the document assigns, with the row it came from.

The dates are computed too. Withdrawal must begin no later than the second day after the ceasefire, and the document sets the ceasefire at midnight Kyiv time on the fifteenth, so the second day is the seventeenth. The record checks both sides of that edge: the seventeenth is in time, the eighteenth is not. The other deadlines in the document, fourteen days to finish the withdrawal, five days to the exchange of prisoners, thirty days to a parliamentary resolution, the end of 2015 for the constitutional reform, were computed and checked the same way, because a one-day error in a deadline is invisible in prose and glaring in a table.

One boundary the machine draws here is worth pausing on. A window that has closed does not, by itself, make a violation. Ask on the second of January 2016 whether the reform obligation was violated, and with no fact about performance either way the answer is “undetermined”. Add the fact that it was not performed, and the answer is “violated”. The machine will not infer a breach from the calendar alone. That is the rule from chapter five, and it holds for treaties as it held for chess.

A sequence nobody wrote down

Now the tenth point.

The document has an order. It says so itself, in three phrases: “from the first day of withdrawal”, “on the first day after withdrawal”, “on the first day after local elections”. From those phrases the modeller laid the document out as a sequence of seven stages, and the machine checked the layout: six transitions, no structural defect, and one honest note that the last stage has no exit, because the document never says what its end is.

Point nine puts the restoration of Ukraine’s control over the border under three conditions: local elections, the constitutional reform of point eleven, agreement with the representatives of the districts. Point ten, the withdrawal of foreign armed formations and the disarmament of illegal groups, is put under no condition and given no deadline. And the document never says whether point ten must be done before the elections or runs beside them.

That question has been argued for years by people with much more at stake than a reader of this book, and I am not going to tell you the answer, because the machine does not have one and was built not to.

What the model does instead is write both readings down. Under the first, security comes first: elections held while foreign armed formations remain on the territory cannot meet the standards the document itself demands in point twelve, so point ten is a condition of the elections. Under the second, the document knows how to state a condition and does so explicitly in points four and nine, so its silence in point ten is a statement, and point ten stands alone. Each reading is marked disputed. Neither is marked right. The package’s own note says that the positions of the parties live in the readings and nowhere else, and that the file exists separately so that the choice cannot be made quietly by adding a clause to some neighbouring rule.

Dependence on reading

With identical facts, the answer may change because of the chosen link between provisions of the document. The comparison shows the consequences of the reading; its justification remains a separate question.

Where the readings part

Two constructed courses of events, then, both written by the modeller as journals. In journal A, everything in the thirteen points is done and done on time, except point ten. Journal B is the same, with two facts added on the day of the elections: the foreign formations withdrawn, the illegal groups disarmed. Point ten done.

Run journal A with no reading chosen. The journal records elections on a date, and the machine says: not established, whether they count as the elections of points nine and twelve. The duty to restore control of the border: not established. Nor is its absence established. The process stops at the stage of dialogue, and the run records three attempted steps that had no legal effect and a finding that the case is stuck short of the end. That is not the model failing. The document does not say when elections may be held, and the model does not decide for it.

Now the reader's turn. Keep every entry of journal A and change only the reading. Under the literal reading, point ten stands alone and is a condition of nothing. Will the machine now establish the duty to restore control of the border?

It will. The elections count, the duty is established, the process runs to its last stage. Under the security-first reading, with the same journal, the elections do not count, the duty is not established, and the process stops at dialogue, as with no reading at all.

Then journal B, where point ten is done. Under both readings the elections count, the duty is established, the process reaches the end. Without a reading, the duty is again not established: the model still does not know whether these are the elections of point nine, and no journal entry can tell it.

The border-control question fits in six cells.

Synthetic scenarios only. Neither journal represents a finding about events in Ukraine.

Reading	Journal A: point ten not done	Journal B: point ten done
none chosen	not established	not established
security first	not established	established
literal	established	established

Each of the six journal-and-reading combinations was evaluated against two questions: whether the elections count and whether border control is due. That produces twelve query results. The table reports border control; both implementations agreed on all twelve results. Laid out this way the dispute has a shape it did not have on paper. The two readings diverge in exactly one cell: point ten unperformed. Everywhere else, the zones, the deadlines, the first four stages, the whole of journal B, they say the same thing.

The disagreement is not about the whole document.

It is about what the document’s silence on the order of point ten means, and the machine has drawn a line around that silence and shown you what stands on each side of it.

What the machine did and did not do

It did not resolve the dispute. It could not: the text supports both readings, and the model’s author declined to decide, and said so. It did not pretend the dispute was everywhere. Before this run I would have said that a disagreement about sequencing infects the whole document. It does not. It infects two questions and one stage.

It did not accept an order it was not given. Without a chosen reading it answered “not established” and stopped, rather than picking the reading the modeller found more natural. The modeller’s note explains why the machine was not made to demand a reading before answering anything: the package’s hand-written tests cannot pass one, and insisting would have silenced them. That is a limitation of a tool, named as such.

It computed what could be computed and stopped where computation stops. The zones and the dates are the document’s arithmetic, and the machine did the arithmetic in public, on both sides of every threshold. The order of points nine and ten is the document’s silence, and the machine kept the silence, in two declared forms.

And it kept the world out. No fact in the model says what any party did. The journals are the modeller’s constructions, labelled as such, and the one thing the model asserts about the real document is its text, pinned byte for byte from the United Nations copy.

The last line

Every chapter of this book has ended on a boundary, and this is the last one.

A computed obligation is a sentence of the document rewritten as a number with its row attached.

An unfinished dispute is a sentence of the document that supports two readings, rewritten as two declared readings with the cases where they part. The machine can do the first and can do the second, and the second is the more useful of the two, because a dispute with a drawn boundary is a dispute you can have without pretending it is about everything.

The machine drew that boundary through the middle of one short document, and it runs between point nine and point ten. What happens on either side of it is not the machine's to say. It never was.

EPILOGUE

Calculemus, Amended



...*sibi mutuo dicere: calculemus.*

G. W. Leibniz, unpublished manuscript, 1680s

The prologue left Leibniz at his counting-board, promising that two philosophers in dispute would one day need to argue no more than two accountants do. Twelve chapters later, here is what was left of the promise when someone built the board.

It did not mean that disputes end. The teacher and the pupil of the first chapter are exactly where Gellius left them, except that their argument now has four written answers instead of two shouted ones, and you can see which of the four you gave. Titius still wants his thing back from Maeivius, and six bodies of law still split three against three, except that the one fact that flips three of them has a name. The bear still gets the roots. Points nine and ten of a document signed in Minsk still do not say which comes first, and the machine, asked, said so twice and stopped.

What the promise meant was narrower and, I think, better. Let us calculate does not mean let us settle it by calculation. It means: we wrote down the rules, the facts and the disputed readings, and the machine showed which answers follow from them and which do not. A dispute run through this machine comes back sorted: here is the part that follows from the rules and facts accepted for this calculation, whose acceptance remains open to challenge; here is the part that follows only if you read the text this way; here is the part where the rules are silent; here is the part that someone with authority must decide; here is the part the model never contained. The philosophers still have to argue. They argue about less, and they can see what they are arguing about.

Three negations, once more

The book was built on three sentences, each with a “not” in it, and each chapter tried to earn one.

The answer is not a verdict. It comes in four kinds, and “not established” is not “refuted”; it comes with a date, and changing

the date changes it; it comes with a list of what was assumed; and when the rules hand the question to a court, or to the bear, the machine hands it on and says so, rather than guessing. A reader who receives an answer from a machine of this kind and reads only its first word has not read the answer.

The model is not the source. What answers is a reading of the statute built by someone, to some depth, with some things left out, and the reading is inspected, measured, and open to objection through channels that change nothing by themselves. A calculation can return the right date while citing the wrong basis. Checking whether an answer follows from a model does not establish whether the model follows from the law. That question is a person's.

Grounds of knowledge and contestability

Experience, evidence and accepted propositions on which what we take as known rests. Contestability makes these supports open to examination and objection; it does not promise infallibility.

The proof is not the truth. Two programs agreeing byte for byte proves they agree. A certificate accepted by the checker proves that the conclusion follows from these rules and these facts for the steps the checker confirmed. The remaining steps it took as assumptions, and it named them one per line. A comparison can agree even when a premise is missing; the certificate must be checked against its own requirements. Green is a claim about what the check checks. It is worth having. It is not truth.

What the machine refuses, and where

Six questions have emerged throughout the book. Here they are together, with the boundaries each one reveals.

Does the machine ever refuse to answer? Yes, in three ways that look different: when the rules assign the question to a judge, it returns a request addressed to that judge; when the answer depends on a reading that has not been chosen, it returns “not established” with the empty line for readings showing; when a

fact the rules require has not been supplied, it says which one, and will not fill it in. The bear's question, the seven days, the one-tenge confirmation.

What happens when the law is silent? "Not established", which is a different answer from "refuted", and the machine keeps them apart even where a human reader would merge them. Gaius after a year of possession; the closed list of the Minsk note.

Who chose the reading? A named person, in a named place, and the choice can be switched by the reader without touching the rules. The seven-day clock; the sequencing of Minsk.

What does the model leave out? The system records which parts of a source have executable coverage and which remain outside it. Where exclusions are deliberate, their reasons can be recorded. Where no explanation has been supplied, that absence must remain visible. The Social Code; the eight blocks of the comparative chain.

If every check passes, is the answer right? No. The checks measure agreement between the machine and its own record of what it checks. The independent comparison; the certificate whose references must all lead to stated grounds.

What does the proof prove? Inference, and only inference, from these rules and these facts, with the assumptions listed one per line. The checker's verbose report; the forged certificate refused.

The author's choices

Every case in this book was chosen, and I have tried to say so where it mattered. I chose old and foreign cases where I could, because a reader who has no stake in Protagoras or Titius can watch the machine without watching their own money. I chose a model contract and a fairy tale for the chapters about amendment and calculation because they let us change a condition without confusing an example with an actual dispute. I chose to run every case on the day I wrote about it and to print what came back, including the limits on what each answer establishes, because a book about a machine that claims to be checkable has

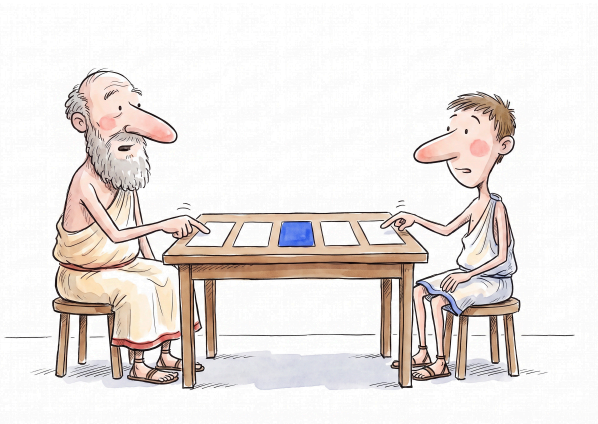
to be checkable itself. The examples state their conditions so that a reader can challenge a premise, a reading or a step rather than merely trust the conclusion.

I did not choose the machine's answers. Where I expected one and got another, the other is what is printed: the year that Gaius's answer was "not established" rather than "refuted"; the difference between a checked inference and a step admitted as an assumption.

Calculus, amended

Leibniz imagined the accountants sitting down together and agreeing on the sum. The picture is right and the moral was too large. Two accountants agree on sums because sums are the kind of thing one agrees on; the philosophers' dispute was never a sum, and no board will make it one.

What a board can do, it turns out, is tell the philosophers which of their sentences were sums all along, and which were readings, and which were silences, and which were somebody's to decide. That is less than Leibniz hoped, and it comes with a condition he did not write down, which this book has tried to write for him.



The same dispute, but now it can be conducted point by point: which victory, the same case, whose argument is stronger.

Let us calculate. And
let us write down,
beside every answer,
what we did not.

Colophon

Canons. How We Read Rules and What Follows From Them

First English edition · Edition 2026.10

© 2026 Rifat Jumagulov

Published by Arxo

This book is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0), <https://creativecommons.org/licenses/by/4.0/>. You may copy, share, translate and adapt it, including for commercial purposes, provided you give appropriate credit, link to the licence and indicate whether changes were made.

The licence covers the text of this edition and, to the extent they are protected, its illustrations. It grants no rights to the Arxo name or logo. Quoted passages and other third-party material remain under their own terms; their sources are named where they appear.

Translations and adaptations made by others are not official editions. Please mark them as unofficial and link to the original.

The chapter illustrations and cover background were generated with Google's Nano Banana Pro image model via Replicate and selected and edited for this edition. They may carry an invisible SynthID watermark. The English cover typography was composed separately for this edition.

Set in Charis SIL, IBM Plex Sans and IBM Plex Mono, all under the SIL Open Font License 1.1.

The current edition, free in PDF, EPUB and on the web: <https://books.arxo.io/canons/>

Experimental records and their limitations: <https://books.arxo.io/canons/experimental-notes/>