

Checked?

An introduction to the logic that won't take your word for it

Vitaly Reznik

ZTL

true **false** **Z**

Z means: it's not up to you

Contents

Introduction	3
Chapter 1. The OR machine	6
Chapter 2. The NOT-AND machine	12
Chapter 3. The IF... THEN machine	17
Chapter 4. True, false and Z	21
Chapter 5. Where the trickster always wins	27
Chapter 6. The NOT machine	34
Chapter 7. Machines in a chain	38
Chapter 8. Laws	47
Chapter 9. Reasoning	54
Chapter 10. Many boxes	61
Chapter 11. Zero-trust logic	68

Introduction

A riddle to start with

There is a machine with two boxes and a light. Into each box goes a note — "yes" or "no". The machine reads both notes and lights up its answer: YES or NO.

One box is yours, the other belongs to the trickster. You want the light to show YES. The trickster wants NO. And he is cunning: he puts his note in **after** you, when he has already seen yours.

What do you put in?

Don't rush. The answer depends on the rules the machine works by. With some rules you can always beat the trickster. With others, never. This book is about telling one from the other. You will answer the riddle in the very first chapter.

Who the trickster is

The trickster in our game is not a villain. He is everything **that you don't decide**: what will be posted in your class chat, what the weather forecast will show, what a chatbot will answer, which number

will be put in place of x in a formula that must hold for every x .

We won't assume that everyone around us lies. We will honestly tell apart what you know yourself from what depends on others. And we will ask: is what you know enough for the answer to be YES, whatever the others decide?

This way of reasoning is called **ZTL** — *zero-trust logic*. ZTL says YES only if the trickster is powerless.

How to read this book

The chapters go in order, and each one builds on the ones before. At the start of each chapter it says which chapters you need.

First come three machines on which the trickster can be beaten. Then we learn to write a whole game in one line, find out where the trickster cannot be beaten, link machines into chains, test which laws of logic hold up against him, and learn to reason and to take long sentences apart. Chapters 7–9 are harder; they are good to read with an adult.

Almost every chapter has a section **"Try it with a friend"**. You don't need a machine — slips of paper and a friend willing to play the trickster are enough. Play it: that makes it clearer than reading.

At the end of each chapter there are **puzzles**. Think first, then check the answers.

No special knowledge is needed. Only attention — and a bit of competitive spirit.

About this book

ZTL — zero-trust logic — was created by **Vitaly Reznik**. He also invented the game of the honest machine and the trickster, set the order of the chapters, and read and checked each of them.

The text of the book was written by the artificial intelligence Claude (by the company Anthropic) from his design and under his direction, first in Russian and then translated into English. The answers in the book were checked against a program that computes by the rules of ZTL.

This English edition and the original Russian one, *Проверили? Введение в ZTL*, can both be downloaded as PDF from ztljudge.com/books/checked.

Chapter 1. The OR machine

You need nothing for this chapter.

The honest machine and the trickster

Picture a machine. It has two boxes and a light. Into each box goes a note — "yes" or "no". The machine reads both notes and lights up its answer: **YES** or **NO**.

The machine is **honest**. It always works by its own rules and deceives nobody.

But next to it stands the **trickster**. One box is yours, the other is his. You want the machine to show YES. The trickster wants it to show NO, and for that he is ready to play a trick: **he puts his note in after you**. He sees what you put in and puts in whatever suits him.

You want YES. The trickster wants NO. Who wins?

Can you make sure the trickster gets nowhere?

Sometimes — yes. It all depends on which machine it is.

First — the machine's rules

The machine's rules are known in advance — to you, to the trickster, to everyone. They are its **table**.

Our first machine is called **OR**. It shows YES if at least one box holds a "yes":

first note	second note	light
yes	yes	YES
yes	no	YES
no	yes	YES
no	no	NO

NO appears in only one case — when both boxes hold "no".

Let's try

First time. You put in "no". The trickster looks — and puts in a "no" too. The machine simply checks its table: no and no — **NO**. The trickster got what he wanted.

Second time. You put in "yes". The trickster looks... and thinks. If he puts in "yes", the table says YES. If he puts in "no", the table still says YES. Whatever he does, the honest machine shows **YES**. The trickster is powerless.

That is the whole secret of the OR machine:

Put in a "yes" — and the trickster can do nothing.

It doesn't matter which box is yours — the left or the right. Your "yes" in either one decides everything.

And if both boxes are the trickster's?

Then he puts "no" in both — and the machine shows **NO**. Against two of his boxes you can do nothing. You can only stop the trickster if at least one box is yours and you know what to put in it.

Try it with a friend

You don't need a machine — slips of paper will do. One of you is the trickster.

1. You write "yes" or "no" on a slip and put it down face up.
2. The trickster looks and writes his slip.
3. Look at the table: what would the honest machine show?

Try both moves — "yes" and "no" — and see for yourselves.

Where tricksters live

The trickster is everything you don't know and don't decide yourself. Life is full of such boxes.

Your class chat: what will be posted there in an hour is not up to you. A weather app: what it will show is not up to you. Even the letter **x** in a formula that must hold for every **x** — for example, $x \cdot 0 = 0$: anything at all may turn up in place of **x**, and the formula must work for every one of them.

We will call such information **dirty**: like water from a puddle — who knows what's in it. It is in someone else's hands, and whoever holds it can slip in what doesn't suit you.

But some things you know for sure — you saw them yourself, counted them yourself. That is your box.

An example. The OR machine: "the test is tomorrow OR today is Monday". Anything may be posted in the chat — that is the trickster's note. But you have seen the calendar yourself — that is your note.

the trickster's box (the chat):	"the test is tomorrow" — what will be posted, nobody knows
your box (the calendar):	"today is Monday"
— yes	
light:	YES

Whatever is posted in the chat, the machine shows YES: your box holds a "yes".

A logic you can't trick

There is a logic that always expects the worst: that a trickster sits in every box that isn't yours. It is called **ZTL** (from *zero-trust logic*). It answers YES only if the trickster is powerless — **whatever** he slips in. And if there are no tricksters and everything is known for sure, it is just ordinary logic.

Puzzles

1. The OR machine. Your box is the left one. What should you put in so that the trickster gets nowhere?
2. You put in "no". What does the trickster put in, and what does the machine show?
3. Both boxes are the trickster's. What does the machine show?

Think it through first, then check.

Answers. 1 — "yes": then YES whatever his note. 2 — he puts in "no", and the machine shows NO. 3 — NO: he puts "no" in both.

In the next chapter: a machine on which you stop the trickster the other way round — by putting in a "no".

Chapter 2. The NOT-AND machine

You need Chapter 1.

A new machine

Remember: you want the machine to show YES, the trickster wants NO, and he puts his note in after you. (We write the notes in small letters — "yes", "no" — and the light in capitals — YES, NO.)

In the last chapter we learned the secret of the OR machine: put in a "yes" and the trickster is powerless. Maybe "yes" always helps?

No. Here is another machine — and on it everything is the other way round.

It is called **NOT-AND**. It shows NO **only when both notes say "yes"**. In every other case it shows YES.

The table, as always, is known in advance:

first note	second note	light
yes	yes	NO
yes	no	YES
no	yes	YES
no	no	YES

NO appears in only one case — when both boxes hold "yes".

Let's try

First time. You put in "yes", as in the last chapter. The trickster looks — and puts in a "yes" too. Yes and yes — **NO**. The trickster got what he wanted.

So the familiar secret doesn't work here.

Second time. You put in "no". The trickster thinks. If he puts in "yes", the table says YES. If he puts in "no" — YES again. Whatever he does, the machine shows **YES**. The trickster is powerless.

The secret of the NOT-AND machine:

Put in a "no" — and the trickster can do nothing.

And again it doesn't matter whether your box is the left or the right one.

Why

To get NO, both notes must say "yes". Yours says "no" — so that can't happen, whatever the trickster puts in.

Compare the two machines:

OR	for NO it needs two "no"s	→	put in "yes"
NOT-AND	for NO it needs two yeses	→	put in "no"

With your note you wreck the trickster's plan.

And if both boxes are the trickster's?

He puts "yes" in both — and the machine shows **NO**. Just as with OR: against two of his boxes you can do nothing.

An example from life

The NOT-AND machine lives wherever there is a rule **"not both at once"**.

Say a movie theater has this rule for a 12+ movie: **you are turned away only when both conditions hold at once** — the viewer is under 12, and has no adult with them. In every other case you get in. That is exactly the NOT-AND machine: the answer to "let in?" is NO only when both notes say "yes".

"the viewer is under 12"	yes	yes	no	no
"has no adult with them"	yes	no	yes	no
let in?	NO	YES	YES	YES

You are 13 — that you know for sure, that is your box: "under 12" — "no". But "with an adult or without" depends on whether the cashier sees one; maybe the cashier won't even notice your mom in line. You don't fill that box.

But it doesn't matter to you what ends up there. Your box holds "no" — the two notes can't both say "yes".
You get in.

Your little brother is 10. His box holds "yes". Then everything depends on the second box: if he comes with mom, he gets in; if they decide he's alone, he doesn't.

Try it with a friend

On slips of paper, as in Chapter 1. One of you is the trickster. Try putting in "yes", then "no". See for yourself: with "no" the trickster can do nothing, with "yes" he can.

Puzzles

1. The NOT-AND machine. What should you put in your box so that the trickster is powerless?
2. You put in "yes". What does the trickster put in, and what does the machine show?

3. On which machine does "yes" help, and on which does "no"?

Think it through first, then check.

Answers. 1 — "no": then the two notes can't both say "yes". 2 — he puts in "yes", and the machine shows NO. 3 — "yes" helps on the OR machine, "no" on the NOT-AND machine.

In the next chapter: the "if... then" machine. Its two boxes are different, and that changes everything.

Chapter 3. The IF... THEN machine

You need the first two chapters.

The boxes are different

The OR and NOT-AND machines had identical boxes: it didn't matter whether yours was the left or the right one. Today's machine is not like that. Its boxes are **different**: the left one is called "if", the right one "then". And which of them is yours changes everything.

The machine's rules

The table, as always, is known in advance. Don't try to match it to how people say "if... then" in everyday talk — these are simply the machine's rules, agreed on before the game:

"if" (left)	"then" (right)	light
yes	yes	YES
yes	no	NO
no	yes	YES
no	no	YES

Look carefully: there is exactly one NO — when "if" holds "yes" and "then" holds "no". Every other row says YES.

The two bottom rows may look strange: "if" holds "no", yet the light still says YES. That is how this machine is built, and that is its strength — you'll see in a moment.

Your box is "then"

Say the trickster has "if", and "then" is yours.

You put in "no". The trickster puts in "yes" to "if". The table: if yes, then no — **NO**. The trickster got what he wanted.

You put in "yes". The trickster thinks. "Yes" in "if" — if yes, then yes — YES. "No" in "if" — if no, then yes — YES. Whatever he does — **YES**.

Your box is "then" — put in "yes".

Your box is "if"

Now the other way round: "if" is yours, and the trickster has "then".

You put in "yes". The trickster puts in "no" to "then". Yes and no — **NO**.

You put in "no". The trickster thinks. "Yes" in "then" — if no, then yes — YES. "No" in "then" — if no, then no — YES. Whatever he does — **YES**.

Your box is "if" — put in "no".

That is what those two "strange" bottom rows are for: they are exactly what saves you when your box is "if".

Two moves — one idea

For NO the trickster needs two notes at once: "yes" in "if" **and** "no" in "then". One of these two boxes is yours. Put in it not what he needs — and his plan is gone:

```
your box is "if"      → put in "no"
your box is "then"    → put in "yes"
```

As in the previous chapters: look at the table to see what the machine needs for NO, and wreck that plan with your note.

And if both boxes are the trickster's?

He puts "yes" in "if" and "no" in "then" — **NO**. Against two of his boxes you can do nothing.

Try it with a friend

On slips of paper, as before, but now agree whose box is "if" and whose is "then". Play both ways round and see: in "if", "no" saves you; in "then", "yes" does.

Puzzles

1. Your box is "then". What should you put in?
2. Your box is "if". What should you put in?
3. You put "yes" in "if" (the trickster has "then").
What does the trickster put in, and what does the machine show?

Think it through first, then check.

Answers. 1 — "yes". 2 — "no". 3 — he puts in "no" to "then", and the machine shows NO.

So we have three machines on which the trickster can be stopped: OR, NOT-AND and IF... THEN. In the next chapter: new words — "true", "false" and Z.

Chapter 4. True, false and Z

You need the first three chapters.

A game in one line

A logician can write down a whole round of our game in one short line. By the end of this chapter you'll be able to do it too. For that you need two new words and one letter.

How logicians say it

So far we have written "yes" and "no" on the notes, and the light has shown YES or NO. Logicians say it differently:

```
yes   →   true
no    →   false
```

For example, a note says "Pete came". If that is how it is, it is **true**. If not, it is **false**. And the light now says not YES and NO but "true" or "false".

From this chapter on, we'll talk this way.

The trickster's box is Z

And what is in the trickster's box? We don't know: he is the one who puts the note in, after you, and he puts in what suits him.

Such a box is marked with the letter **Z**. Remember this well:

Z is always in the trickster's hands.

Z is not what is written on the note. Z tells you **whose box it is**: "here the trickster decides". That is how we say it: the value of this box is Z. We called such information dirty: the class chat, the weather forecast, the letter x in a problem. All of it is Z.

Three values in, two out

So in the logic ZTL the **inputs** — the boxes — can have three values:

true	you know for sure
false	you know for sure
Z	in the trickster's hands

And the **output** — the light — has only two: true or false. The light never shows a third one. The machine is honest: it always computes by its table. It's just

that when a box's value is Z, the trickster picks the note for it — and picks one that makes the light show "false", if he can.

So the rule of ZTL goes like this:

The light shows "true" only if the trickster cannot make it false. Otherwise — "false".

Remember right away: "false" on the light does not mean "wrong". It means **"can't be guaranteed"**. The trickster can make the sentence false — so nobody can guarantee it.

Our three machines in the new language

Let's read one line out loud. The OR machine. Your box holds "true", the other one is Z. Whatever the trickster puts in, OR with one "true" shows "true". So:

OR: yours – true, the other – Z → true

All our discoveries from the first three chapters can be written the same way:

OR. Put in "true" — the light says "true". Put in "false" — the trickster puts in "false", and the light says "false".

The other two machines as a short table: what to put in your box so that the light shows "true".

NOT-AND	put in
"false"	
IF... THEN, your box is "if"	put in
"false"	
IF... THEN, your box is "then"	put in
"true"	

Put in anything else — and the trickster gets "false".

And one rule for every machine at once: **if both boxes have the value Z, the light says "false".**

The promised line

And here is how a logician writes down a whole round. First the machine's name, then in brackets what is in the boxes, then after the "=" sign what the light shows:

OR (true, Z)	= true
OR (false, Z)	= false
NOT-AND (false, Z)	= true
OR (Z, Z)	= false

One line — one round. Now you can do it too.

Try it with a friend — in the new words

Play as before, but write "true" and "false" on the slips. On the trickster's box write Z — that means he puts the note in, after you. Take any of the three machines and find what to put in so that the trickster is powerless.

Puzzles

1. What is the note "yes" called in the logicians' language? And "no"?
2. What is the value of a box where the trickster decides?
3. The OR machine: your box holds "false", the other one is Z. What does the light show?
4. Can the light show Z?
5. Write in one line: the NOT-AND machine, your box holds "true", the other one is Z.

Think it through first, then check.

Answers. 1 — "true" and "false". 2 — Z. 3 — "false": the trickster puts in "false". 4 — no: the light shows only "true" or "false"; only a box can be Z. 5 — NOT-AND (true, Z) = false.

In the next chapter: machines on which the trickster always wins.

Chapter 5. Where the trickster always wins

You need the first four chapters.

Is there a machine with no secret?

Each of our three machines had a secret. On OR put in "true", on NOT-AND "false", on IF... THEN it depends on which box is yours. Each time the trickster was left empty-handed.

Maybe every machine has a secret — you just have to look for it?

No. You are about to see machines on which the trickster **always** wins, whatever you put in. One box is yours, the other is his, and his box has the value Z.

The AND machine

Its light shows "true" only if **both** boxes hold "true":

first	second	light
true	true	true
true	false	false
false	true	false
false	false	false

You put in "true". The trickster puts in "false". The table: true, false — the light says **false**.

You put in "false". The trickster doesn't even need to think: with your "false" the light can no longer show "true". **False.**

In one line:

```
AND (true, Z) = false
AND (false, Z) = false
```

The AND machine needs **both** boxes to agree. And the trickster never agrees.

The NOT-OR machine

Don't mix it up: this is not the OR machine with a NOT in front of it, but **one** machine, with its own table — like NOT-AND in Chapter 2. It shows "true" only if both boxes hold "false":

first	second	light
true	true	false
true	false	false
false	true	false
false	false	true

You put in "false". The trickster puts in "true" — **false**.

You put in "true". The light can no longer show "true" — **false**.

The EITHER-OR machine

It shows "true" when the notes are **different**. If they are the same — "false".

The trickster simply **copies** your note. You put in "true" — he puts in "true". You put in "false" — he puts in "false". The notes are the same — **false**.

The SAME machine

The opposite: "true" when the notes are **the same**.

The trickster puts in **the opposite**. The notes are different — **false**.

How to tell in advance

You don't need to play to find out whether a machine has a secret. The table is enough.

Put your finger on your note — say, "true" in the first box. Two rows of the table go with it: one for when the trickster puts in "true", the other for when he puts in "false". If the light says "true" **in both** rows, that is your secret. If even one of them says "false", the trickster will pick exactly that one.

Check both of your notes this way — "true" and "false". And if the machine's boxes are different, as with IF... THEN, check each box separately too.

Check the OR machine this way: with your "true" both rows say "true". Secret found. But with the AND machine, whatever your note, at least one row says "false". No secret.

There is a secret if your note guarantees "true" — whatever the trickster puts in.

How many machines are there?

Logicians have counted every two-box machine. There are only **sixteen** different tables: a table has four rows, each row's light is "true" or "false", and that makes $2 \times 2 \times 2 \times 2 = 16$.

But six of them are not real two-box machines:

always true	flip the first
always false	copy the second
copy the first	flip the second

The first two don't read the notes at all. The others read only one box: if it is yours, you decide everything; if it is his, he does.

That leaves ten real ones. Only four of them have a secret:

OR	put in "true"
NOT-AND	put in "false"
IF... THEN	"if" – "false", "then" – "true"
IF... THEN reversed	the same secret

"IF... THEN reversed" is the same machine, only with "if" on the right and "then" on the left. "If" and "then" play different roles, so these are two different tables — and they count as two. Four tables with a secret, but three machines.

On the other six the trickster always wins: AND, NOT-OR, EITHER-OR, SAME and two more that are like AND — "the first is true and the second is false" and the other way round.

That is why the first three chapters were about exactly these three machines.

An example from life

Remember from Chapter 1: "the test is tomorrow OR today is Monday"? The chat is the trickster's box, the calendar is yours. The calendar says "Monday" — true, and OR shows "true", whatever is posted in the chat.

Let's change the machine: "the test is tomorrow **AND** today is Monday". Same calendar, "true". But now the chat decides: if they post that there's no test, the whole sentence is false. ZTL answers: **false**.

That doesn't mean there definitely is no test. It means the trickster **can** make the sentence false. And in ZTL "true" happens only when he can't.

Try it with a friend

Take the AND machine. Let your friend be the trickster. Try putting in "true" and "false" as often as you like. Do you ever get "true"?

Then play EITHER-OR, swapping roles. The trickster copies his friend's note — see whether the friend ever gets "true".

Puzzles

1. The AND machine. You put in "true". What does the trickster put in, and what does the light show?
2. The EITHER-OR machine. You put in "false". What does the trickster put in?
3. On which three machines can you stop the trickster?

4. Write in one line: the SAME machine, your box holds "true", the other one is Z.
5. The machine "the first is true and the second is false": its light shows "true" only in that case. Your box is the first one. Does it have a secret? Draw its table and check with your finger.

Think it through first, then check.

Answers. 1 — "false"; the light says "false". 2 — "false" too: the notes are the same, and the light says "false". 3 — OR, NOT-AND, IF... THEN. 4 — SAME (true, Z) = false. 5 — no: if you put in "true", the trickster puts "true" in the second box, and the light says "false"; if you put in "false", the light says "false" straight away.

On "Try it with a friend". On the AND machine — never: the trickster puts in "false". On EITHER-OR — never either: a copied note makes them the same.

In the next chapter: a machine with just one box — NOT.

Chapter 6. The NOT machine

You need the first five chapters.

A machine with one box

All our machines so far had two boxes. This one has **one**.

It is called **NOT**. Its job is to flip: whatever goes in, the light shows the opposite.

box	light
true	false
false	true

The table is tiny, and as always it was agreed on before the game.

If the box is yours

Then there is nobody to play against. You put in "true" — the light says "false". You put in "false" — the light says "true". The trickster has nowhere to put a note: there is one box, and it is yours.

NOT (true)	= false
NOT (false)	= true

If the box is the trickster's

Now the only box is his. The box's value is Z. You have nothing to put in.

The trickster looks at the table. He needs "false" on the light. So the box needs "true". That is what he puts in.

NOT (Z) = false

In short: **if the box is the trickster's, NOT always shows "false"**.

"It will" and "it won't" — neither is true

Here is what that means. The class chat is the trickster's box: what will be posted about tomorrow's test is not up to you. Imagine a debate with a friend.

Your friend says: **"There will be a test tomorrow."**

Can ZTL call that true? Remember the rule from Chapter 4: "true" only if the trickster cannot make the sentence false. But here he decides everything. So — not "true".

You answer: **"There will NOT be a test tomorrow."**

That is the NOT machine over the same box: NOT (Z) = false. Not "true" either.

The argument is a draw: ZTL guarantees neither your friend's sentence nor yours.

Careful, a trap! You may want to reason like this: "The sentence 'there will be no test' is false — so there will be a test."

No! As in Chapter 5: "false" here doesn't mean "wrong", it means "the trickster **can** make this sentence false".

And he can make your friend's sentence false too. So neither of the two sentences is true.

You really can't guarantee anything — and ZTL agrees with you.

NOT — not a secret, but a flip

OR, NOT-AND and IF... THEN had a secret: your note beat the trickster. NOT has no second box, so it has no secret either. If the box is yours, you know everything; if it is his, he decides everything.

Try it with a friend

One of you is the trickster and holds the only box. The other one tells him: "Put in 'false'!" Can the trickster obey and still get "false" on the light? Try

every option — there are only two. (Remember: the trickster wins if the light shows "false".)

Puzzles

1. NOT (false) — what does the light show?
2. The box is the trickster's. What does he put in, and what does NOT show?
3. The sentence "it will rain tomorrow", and the forecast is not up to you. Can ZTL call it true? And the sentence "it will NOT rain tomorrow"?

Think it through first, then check.

Answers. 1 — true. 2 — he puts in "true", and the light says "false": NOT (Z) = false. 3 — no and no: the first is the trickster's box, the second is NOT (Z) = false.

On "Try it with a friend". No. If he puts in "false", NOT shows "true", and he loses. So he always puts in "true".

And what if you put into NOT's box not a note but the light of another machine? That is the next chapter: machines in a chain.

Chapter 7. Machines in a chain

You need the first six chapters.

A light into a box

So far the boxes have held notes. But a box can also take the **light** of another machine: whatever it showed is what the next machine reads.

That is how machines are joined into a **chain**:

```
"true" (yours)  Z
      ↓         ↓
    [  OR  ]
      ↓ light
    [  NOT ]
      ↓
    light
```

Logicians write a chain in one line — one machine inside the brackets of another:

```
NOT ( OR (true, Z) )
```

You read it **from the inside out**. First the inner machine works — OR. It has your "true", so it shows

"true". That "true" goes into the box of NOT. NOT flips it:

```
NOT ( OR (true, Z) ) = false
```

The main rule of a chain

Remember: the light shows only "true" or "false". Z never appears on a light.

So when a light goes into a box, it is **not the trickster's box**. An honest machine lit it. It can't be swapped, replaced or changed.

A light that was lit at one step is a fixed note at the next.

Then who is the trickster in a chain? Let's agree before the game:

Every machine has its own trickster. All he cares about is the light of his machine.

He is in charge of his machine's boxes that have the value Z — and only there. What happens further down the chain is none of his business: his job is to make **his own** light show "false". The rule from

Chapter 4 works for each machine separately: its light is "true" only if **its** trickster is powerless.

And one more thing: if Z appears in two places in a chain, those are **two different boxes**, one for each machine. Each trickster puts whatever he likes in his own. There is only one chat, but each machine asks it separately — and the trickster can answer each one differently. That's what makes him a trickster.

NOT NOT

Let's pass the trickster's box through NOT twice:

```

  Z
  ↓
[ NOT ]
  ↓ light
[ NOT ]
  ↓
light
```

```
NOT ( NOT (Z) )
```

Step 1. The trickster of the first NOT puts in "true" — its light says "false". We know that from Chapter 6.

Step 2. The box of the second NOT holds not Z but a light that shows "false". Its trickster has nowhere to put anything. NOT flips it: **true**.

```
NOT ( NOT ( Z ) ) = true
```

Careful, a trap! At school they say two NOTs cancel out. "It's not true that there won't be a test" means "there will be a test". In ZTL you **can't** do that. The sentence "there will be a test" itself is still not true: the trickster holds the chat.

The second NOT is not about the chat. It is about what the **first light** showed. Each NOT is its own step.

No "**both true and not true**"

The sentence: "it is not true that there will **and** won't be a test".

```
NOT ( AND ( Z, NOT ( Z ) ) )
```

There are two Z boxes here — one for the AND machine and one for the inner NOT. Step by step, from the inside out:

```
NOT ( Z )           = false
                     (the trickster puts in "true")
AND ( Z, false )    = false
                     (the box already holds "false")
NOT ( false )       = true
```

The result is **true**, even though we know nothing about the chat itself.

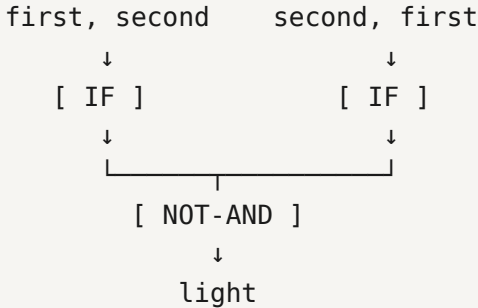
But this light has to be read as carefully as in "NOT NOT". The outer NOT is not about the chat but about the light of the AND machine. And that one showed "false": ZTL does not guarantee "there will and won't be a test at once".

The result "true" reads like this: ZTL guarantees the sentence "**it is not true that it is both true and not true**". But it will never guarantee "both true and not true" itself.

EITHER-OR from three machines

Remember EITHER-OR from Chapter 5? "True" when the notes are different. The trickster always beat it: he just copied your note.

It can be **built** from our machines. Take two boxes: the first and the second. The left IF reads "if the first, then the second", the right one the other way round, "if the second, then the first". Their lights go into NOT-AND:



First without tricksters. If the notes are the same, both IFs show "true", and NOT-AND on two "true"s gives "false". If the notes are different, one of the IFs shows "false", and NOT-AND gives "true". Exactly the table of EITHER-OR!

Now with tricksters. The first box is yours and holds "true". The second is Z. Both IFs need it, so each IF has its own copy and its own trickster.

```

left IF (true, Z)           = false
    (the trickster puts "false" in "then")
right IF (Z, true)          = true
    ("then" holds "true" – he is powerless)
NOT-AND (false, true)       = true
  
```

The chain showed **true**. But the single EITHER-OR machine on the same boxes would have shown **false**.

The trickster of the left IF got what he wanted — its light says "false". But for NOT-AND one "false" in a

box is exactly what you need. His win helped you at the next machine.

In ZTL it matters not only which table you end up with, but how the machine is built. Without tricksters, the built EITHER-OR and the single one are the same. With tricksters, they are not.

And if one trickster ran the whole chain?

You could invent a different game: one trickster sees the whole chain at once and cares only about the last light. Then he would give way at the first step to win at the end: in NOT (NOT (Z)) he would put in "false" — the first light "true", the second "false". That would be a **different** logic.

ZTL plays in steps. Each machine gives its own honest answer, and each light is lit once and for all.

Try it with a friend

Draw the chain NOT (NOT (Z)): two machines, the light of the first going into the box of the second. You are the trickster of the **first** NOT; your friend watches the tables.

What do you want from your light? What do you put in? What does the second light show? Now think:

what would a trickster put in if he ran the whole chain alone?

Puzzles

1. NOT (OR (false, Z)) — what does each light show?
2. Can the trickster put a note into a box that has received another machine's light?
3. NOT (NOT (Z)) = true. Does that mean the trickster's box itself is true?
4. IF (Z, true) — what does the light show?

Think it through first, then check.

Answers. 1 — OR: the trickster puts in "false", the light says "false"; NOT flips it — "true". 2 — no: it holds a fixed note, not Z. 3 — no: the second NOT is about the first light, not about the box; ZTL still does not guarantee the box itself. 4 — true: "then" holds "true", and the trickster is powerless.

On "Try it with a friend". You are the trickster of the first NOT and want "false" on its light: you put in "true". The first light says "false", the second "true". A trickster running the whole chain alone would put in "false": the first light "true", the second "false".

In the next chapter: laws — which rules of ordinary logic ZTL keeps, and which it doesn't.

Chapter 8. Laws

You need the first seven chapters.

What a law is

Logic has **laws** — sentences that are always true, whatever you put into them. For example: "there will be a test or there won't". Whether there is one or not doesn't matter: one of the two is sure to happen, and the sentence is true.

In the language of machines, a law is a chain whose light shows "true" whatever notes are in the boxes.

There are also laws of another kind: **two different chains always show the same thing**. We'll test those too.

But we have tricksters. And a new question arises:

Does a law stay a law when the trickster holds the boxes?

Without tricksters — everything holds

The good news first. Remember from Chapter 1: if there are no tricksters and everything is known for

sure, ZTL is just ordinary logic.

So it is. If every box holds a fixed "true" or "false", every law of ordinary logic works in ZTL. Every single one.

The interesting part starts when a box holds Z.

A law that survived

"Both true and not true — never happens." We checked it in Chapter 7:

```
NOT ( AND ( Z, NOT (Z) ) ) = true
```

The tricksters hold every box, and the light still says "true". The law survived: ZTL guarantees "it is not true that it is both true and not true". Just remember how to read it (Chapter 7): it will never guarantee "both true and not true" itself, and the law tells you nothing about the chat itself.

A law of the second kind held too — **"order doesn't matter"**: $\text{OR}(\text{true}, Z)$ and $\text{OR}(Z, \text{true})$ both show "true". The same goes for OR, AND and NOT-AND: swap the boxes and the light doesn't change. (For IF... THEN order does matter: "if" and "then" are different boxes.)

The law "it's one or the other"

Let's take that very example: "there will be a test **or** there won't".

```
OR ( Z, NOT (Z) )
```

Steps:

```
NOT (Z)           = false
  (the trickster puts in "true")
OR (Z, false)     = false
  (the trickster puts in "false")
```

The light says **false**. The law "it's one or the other" broke!

Remember the argument from Chapter 6: ZTL guaranteed neither "there will be" nor "there won't". So it doesn't guarantee "there will or there won't" either. You don't have a single note of your own: the second box of OR holds the light "false", and the trickster holds the first.

The law did not become untrue. Tomorrow everything will be clear, and one of the two will happen. It's just that today ZTL won't guarantee it: the chat is not up to you.

The law "if it is, then it is"

The most obvious law in the world: "if there will be a test, then there will be a test".

```
IF (Z, Z)
```

Both boxes of the IF machine are the trickster's. It doesn't matter that the same sentence goes in both: there are two boxes, and he fills each one himself. He puts "true" in "if" and "false" in "then". The light says **false**.

Remember the rule from Chapters 3 and 4? **Against two of the trickster's boxes you can do nothing.** Even if the same thing is written on both.

The law "not not"

In ordinary logic "it is not true that it isn't" is the same as "it is": if one is guaranteed, so is the other. In Chapter 7 we saw:

```
NOT ( NOT (Z) ) = true
```

ZTL guarantees "it is not true that it isn't". But not "it is" itself — the trickster's box. They no longer mean the same thing. The law broke.

De Morgan's law

There is a law with a name — the British mathematician Augustus De Morgan wrote it down almost two hundred years ago. It says: "it is not true that (a and b)" is the same as "not a or not b".

In the language of machines these are two chains:

```
left:    NOT ( AND (a, b) )
right:   OR ( NOT (a), NOT (b) )
```

Without tricksters they always show the same thing. Now let a be your box, "true", and b be Z.

Left:

```
AND (true, Z)      = false
NOT (false)        = true
```

Right:

```
NOT (true)         = false
NOT (Z)            = false
OR (false, false)  = false
```

Left "true", right "false". Two chains that are the same in ordinary logic have parted ways in ZTL — like EITHER-OR in Chapter 7.

What this means

The laws of ordinary logic are rules for a world where everything is known. ZTL doesn't cancel them: without tricksters they all hold.

But when the trickster holds a box, some laws hold and some break. You can't always tell just by looking at a sentence. One thing is reliable: take the chain and work it out step by step.

ZTL doesn't argue with ordinary logic. It just won't guarantee what isn't up to you.

A law is tested not by how the sentence looks, but by what the light shows when the tricksters hold the boxes.

Try it with a friend

Play the law "if it is, then it is" — IF (Z, Z). One of you is the trickster and gets both boxes. The other checks: can the trickster get "false" if he must put **the same thing** in both boxes? And if he doesn't have to?

Puzzles

1. Does the law "it's one or the other" break if the box is yours and holds "true"?
2. Why does IF (Z, Z) show "false", even though "if it is, then it is" is obvious?
3. Which laws in this chapter held when the tricksters held the boxes?
4. Work it out step by step: OR (false, NOT (Z)).

Think it through first, then check.

Answers. 1 — no: OR (true, NOT (true)) = OR (true, false) = true. 2 — both boxes are the trickster's, and he fills each one himself: "true" in "if", "false" in "then". 3 — "both true and not true — never happens" (and "order doesn't matter"). 4 — NOT (Z) = false; OR (false, false) = false.

On "Try it with a friend". If the trickster must put the same thing in both boxes, he is powerless: IF (true, true) and IF (false, false) both show "true". But that was not the deal in our game: there are two boxes, and he fills each one himself. That is why the law breaks.

Chapter 9. Reasoning

You need the first eight chapters.

Can we reason at all?

In the last chapter even the law "if it is, then it is" broke. That's worrying: maybe you can't reason with tricksters around at all? Maybe nothing can be deduced?

It can. This chapter has two rules the trickster cannot break.

What "deducing" means

Reasoning is when you get something new from what you already know.

In the language of machines: something is already definitely "true" — your note or the light of another machine. You look at it and find out what else has to be "true".

We'll use the letters a , b , c for sentences. A sentence can be your note, the light of another machine or the trickster's box. Writing " $a = \text{true}$ " means: a is your note "true", or a light that shows "true". You can't say that about the trickster's box: its value is Z .

The bridge rule

Say you know two things:

```
a                = true
IF (a, b)        = true
```

What can you say about b?

Remember the table of IF... THEN from Chapter 3 (back then we wrote "yes" and "no", now "true" and "false"). Take only the rows where "if" holds "true". There are two. The light says "true" in only one of them: when "then" holds "true" too.

But what if b is the trickster's box? Then the trickster of this IF would put "false" in "then", and the light would say "false". But it says "true". So b is not the trickster's box but a fixed "true".

The bridge rule: if $a = \text{true}$ and $IF(a, b) = \text{true}$, then $b = \text{true}$ as well.

IF (a, b) is like a bridge from a to b. You stand on a, the bridge holds — you cross over to b:

```
a ==bridge==> b
```

A bridge isn't built from nothing

Notice what the bridge rule does **not** do. It doesn't make true what the trickster holds.

If b is the trickster's box, then $\text{IF}(\text{true}, b)$ will never show "true": the trickster will put in "false". There is no bridge — and nothing to cross.

Reasoning passes "true" along, but it doesn't make it out of thin air.

The chain of bridges rule

Now you know two other things:

```
IF (a, b)      = true
IF (b, c)      = true
```

A bridge from a to b and a bridge from b to c :

$$a \implies b \implies c$$

So you can go from a straight to c :

```
IF (a, c)      = true
```

Let's check all three cases for a .

$a = \text{false}$. Then $\text{IF}(a, c) = \text{true}$ by the table, whatever c holds.

a = true. By the bridge rule $b = \text{true}$. By the bridge rule again $c = \text{true}$. And $\text{IF}(\text{true}, \text{true}) = \text{true}$.

a is the trickster's box. Every machine has its own trickster and its own copy of a . The trickster of the machine $\text{IF}(a, b)$ would put "true" in "if" and win — unless b were "true". Since its light says "true", $b = \text{true}$. By the bridge rule $c = \text{true}$. And the machine $\text{IF}(a, c)$ has "true" in "then" — its trickster is powerless.

The chain of bridges rule: from $\text{IF}(a, b)$ and $\text{IF}(b, c)$ follows $\text{IF}(a, c)$.

What about "if it is, then it is"?

Strange: the law "if it is, then it is" broke, but the bridge rule stands. Why?

$\text{IF}(Z, Z)$ is a machine where you have **not a single note of your own**. Both boxes are the trickster's, and he gets "false".

The bridge rule starts from what you **already have**: $a = \text{true}$ and bridge = true. That is what b is deduced from. In $\text{IF}(Z, Z)$ nothing at the input is yours — so the trickster puts "true" in "if" and "false" in "then".

For the curious: the rule as a law

The bridge rule can be written as a chain, like the laws in Chapter 8. Notice: *a* appears in it twice — those are two boxes.

```
IF ( AND (a, IF (a, b)), b )
```

Let the tricksters hold every box. Step by step:

```
IF (Z, Z)           = false
AND (Z, false)      = false
IF (false, Z)       = true
```

If you go through all the cases — with your notes and with tricksters — the light always says "true". The same goes for the chain of bridges.

But to be honest: when the tricksters hold the boxes, the law held because the inner IF showed "false". The law tells us nothing about the boxes themselves. It confirms something else: **this way of reasoning is reliable**.

Try it with a friend

This time without a trickster. The IF machine; your friend puts notes in both boxes however they like, and you write down what the light showed.

Find all the ways of filling the boxes where "if" holds "true" and the light showed "true". What is in "then" every time? That is the bridge rule.

Puzzles

1. $a = \text{true}$ and $\text{IF}(a, b) = \text{true}$. What can you say about b ?
2. $\text{IF}(a, b) = \text{true}$ and $\text{IF}(b, c) = \text{true}$. What can you say about $\text{IF}(a, c)$?
3. b is the trickster's box, $a = \text{true}$. What does $\text{IF}(a, b)$ show? Can you deduce b by the bridge rule?
4. Why did "if it is, then it is" break while the bridge rule didn't?

Think it through first, then check.

Answers. 1 — $b = \text{true}$. 2 — $\text{IF}(a, c) = \text{true}$. 3 — "false": the trickster puts "false" in "then". No, you can't: there is no bridge. 4 — in $\text{IF}(Z, Z)$ you don't have a single note of your own, and the trickster wins; the bridge rule starts from what is already "true".

On "Try it with a friend". There is only one such way: $\text{IF}(\text{true}, \text{true})$. "Then" holds "true". And $\text{IF}(\text{true},$

false) = false.

In the next chapter: long sentences — many boxes at once.

Chapter 10. Many boxes

You need the first nine chapters.

Long sentences

In life, sentences can be long: "I have a pen, and a notebook, and a textbook, and my planner". Or: "let's go to the movies, or the park, or the ice rink, or a pool".

OR and AND have two boxes each. So what do we do with a long sentence? Build a chain!

Long OR

"Movies, or park, or ice rink, or pool" is three OR machines, the light of each going into the next:

```
[ OR ] ← ice rink, pool
  ↓ light
[ OR ] ← park
  ↓ light
[ OR ] ← movies
  ↓
light
```

In one line:

```
OR ( movies,  
    OR ( park,  
        OR (ice rink, pool) ) )
```

About the ice rink you know for sure: you saw yourself that it's open. The ice rink is your "true". The rest is up to others — those are tricksters' boxes.

We work from the inside out. Remember the rule from Chapter 7: the light of an inner machine is a fixed note for the next one; the trickster doesn't touch it.

```
OR (ice rink, pool)    = true  
  (ice rink – your "true",  
   pool – Z)  
OR (park, light)      = true  
  (park – Z, light – "true")  
OR (movies, light)    = true
```

Your "true" climbed the chain all the way to the top.

Long OR: one "true" of yours is enough for the whole sentence.

And if there is no "true" of yours anywhere? Then every trickster puts in "false", and the light at the top says **false**.

Where to put the ice rink?

What if the ice rink were not in the innermost machine but outside — "ice rink, or movies, or park, or pool"? Then the inner ORs show "false" — the tricksters made sure of it. But the outer OR gets your "true" — and shows **true**.

It comes out the same. In a long OR it doesn't matter where your "true" stands or how the brackets are placed. In a long AND it doesn't either. Try it yourself: move the ice rink anywhere — you'll get the same result.

Long AND

Now the backpack: "a pen, and a notebook, and a textbook, and my planner".

```
AND ( pen,  
      AND ( notebook,  
            AND (textbook, planner) ) )
```

You packed the pen, the notebook and the textbook yourself. The planner, they say, mom packed — you didn't see. That is the trickster's box.

```
AND (textbook, planner)  = false
  (the trickster puts in "false")
AND (notebook, false)   = false
AND (pen, false)         = false
```

One trickster's box — and the whole sentence is **false**. The innermost AND got Z, the trickster put in "false", and that "false" climbed to the top.

Long AND: one trickster's box ruins the whole sentence.

For a long AND to show "true", **all** the boxes must be your "true"s. Look in the backpack yourself.

Mixed sentences

And if they're mixed? "(Movies or park) and (juice or tea)".

```
AND ( OR (movies, park),
      OR (juice, tea) )
```

The trickster has nowhere to put anything in the outer AND: both its boxes get the lights of the ORs. So everything is decided inside: each OR needs one "true" of yours.

You know for sure about the movies and about the tea:

```
OR (movies, park)      = true
OR (juice, tea)         = true
AND (true, true)       = true
```

But if you only know about the movies, and others decide about the drinks:

```
OR (movies, park)      = true
OR (juice, tea)         = false
AND (true, false)      = false
```

How to find your move quickly

As long as a sentence has only AND and OR, you don't have to work out the whole chain. Go from the top down:

```
OR at the top      one part is enough
AND at the top     all parts are needed
```

(A "part" is the light of an inner machine or a box; you need the part to be "true".)

Keep going down like this until you reach the boxes. There, "true" can only be yours.

(If the sentence has NOT, this shortcut no longer works: NOT flips. Then work it out step by step, as in Chapter 7.)

Try it with a friend

Write a long OR of five boxes — that's four OR machines in a chain. One of you is the trickster and gets every box but one. Choose together which one is yours. What do you put in it?

Then a long AND of five boxes. How many boxes must be yours for you to win?

Puzzles

1. $\text{OR} (Z, \text{OR} (Z, Z))$ — what does the light show?
2. A long AND of ten boxes: nine are your "true"s, and one is Z. What does the light show?
3. $\text{AND} (\text{OR} (Z, Z), \text{OR} (\text{true}, Z))$ — work it out step by step.
4. A long AND of five boxes: four are your "true"s, one is Z. The brackets are placed any way you like. What does the light show?

Think it through first, then check.

Answers. 1 — false: there is no "true" of yours anywhere. 2 — false: one trickster's box is enough. 3 — $\text{OR}(Z, Z) = \text{false}$; $\text{OR}(\text{true}, Z) = \text{true}$; $\text{AND}(\text{false}, \text{true}) = \text{false}$. 4 — false, however you place the brackets: the trickster's box will still end up in some AND.

On "Try it with a friend". In a long OR put in "true" — and you've won, whichever box is yours. In a long AND all five must be yours.

In the last chapter: the summary — zero-trust logic.

Chapter 11. Zero-trust logic

You need all the chapters.

What we learned

Let's look back. It all started with an honest machine and a trickster.

Machines. Each one has a table agreed on before the game. The light shows only "true" or "false".

The trickster. He puts his note in after you, and he wants "false". The value of his box is Z. Z is always in the trickster's hands.

Three machines with a secret. On OR put in "true". On NOT-AND — "false". On IF... THEN: if your box is "if", put in "false"; if it is "then", "true". On the others — AND, NOT-OR, EITHER-OR, SAME and the two more from Chapter 5 — the trickster always wins.

NOT. If its only box is the trickster's, the light always says "false".

Chains. The light of one machine is a fixed note for the next. Every machine has its own trickster. That's why it matters how a chain is built: EITHER-OR built

from three machines doesn't behave like the single one.

Laws. Without tricksters every law of ordinary logic holds. With tricksters, some hold and some break — check step by step.

Reasoning. The bridge rule: if $a = \text{true}$ and $\text{IF } (a, b) = \text{true}$, then $b = \text{true}$ as well. It passes "true" along, but doesn't make it out of thin air.

Long sentences. In a long OR one "true" of yours is enough. In a long AND one trickster's box ruins everything.

All of this together is **ZTL** — zero-trust logic.

Where tricksters live today

In Chapter 1 we found tricksters in the class chat, in the weather forecast and in the letter x. There are many more.

Rumors: "somebody told me that...". Who told you? You didn't see.

The label on a package: "this juice has no sugar". You didn't measure it.

The answer of a chatbot or a search engine. You asked a question — and got a note that you didn't put in. Maybe a correct one. Maybe not.

Whoever wrote these notes doesn't necessarily want to fool you. The "trickster" in our game is just a way of saying: **this isn't up to me**. The value of such a box is Z.

What ZTL can do

ZTL checks a sentence step by step, machine by machine. At each step the light shows "true" only if that machine's trickster is powerless. Otherwise — "false".

Remember what that means. "False" is not "wrong" but "can't be guaranteed". And a "true" light at the top of a chain doesn't yet mean that every sentence inside is true. Remember Chapter 7: NOT (NOT (Z)) = true, yet ZTL still doesn't guarantee the box Z itself.

But once you've worked it out step by step, you can see which parts of the sentence rest on your own notes and which on other people's boxes. And where your note is enough to make the trickster powerless.

What ZTL can't do

ZTL **doesn't know what is in the trickster's box**. It doesn't say the rumor is wrong or that the label on the package is untrue.

ZTL **doesn't turn someone else's box into yours.** Only you can do that: go and look for yourself, count for yourself, check for yourself. Then the box will hold your own note — and you can work with that.

Zero trust doesn't mean "everybody lies"

It may seem that ZTL suspects everyone. Let's be honest: by the rules of ZTL even your mom's word is a box Z until you check it yourself.

But whether to trust is up to you, not ZTL. In life we trust people — and rightly so. ZTL doesn't forbid it. It only shows **what your conclusion rests on**: on what you know yourself, or on someone else's word.

And when there are no tricksters and everything is known for sure, ZTL is just ordinary logic.

The last puzzle

An example first. The sentence: "there is a pen **or** a pencil in the backpack". You packed the pen yourself — your "true". You don't know about the pencil: maybe mom packed it, maybe not. That is Z.

$\text{OR}(\text{true}, Z) = \text{true}$

This sentence can be guaranteed, whatever happened to the pencil.

And "there is a pen **and** a pencil in the backpack"?

```
AND (true, Z) = false
```

It can't be guaranteed. That doesn't mean there is no pencil. It means: the pencil isn't up to you.

Now take a simple sentence that you heard or read today — with AND, OR, NOT or IF... THEN. Break it into boxes and work it out step by step. (With IF... THEN, remember which box is "if" and which is "then".)

You already know how to work it out. And which boxes are yours — only you know.

Finally

The machines, the boxes and the tricksters will stay in this book. But the question they taught you to ask will go on with you:

What does my conclusion rest on — on something I know myself, or on someone else's word?