Propositional Logic

Question: How do we formalize the definitions and reasoning we use in our proofs?

Where We're Going

- ***Propositional Logic*** (Today)
  - Reasoning about Boolean values.
- ***First-Order Logic*** (Wednesday/Friday)
  - Reasoning about properties of multiple objects.

Propositional Logic

A ***proposition*** is a statement that is, by itself, either true or false.

Some Sample Propositions

- I am not throwing away my shot.
- I’m just like my country.
- I’m young, scrappy, and hungry.
- I’m not throwing away my shot.
- I’m ‘a get a scholarship to King’s College.
- I prob’ly shouldn’t brag, but dag, I amaze and astonish.
- The problem is I got a lot of brains but no polish.

Things That Aren't Propositions

Commands cannot be true or false.

Things That Aren't Propositions

Questions cannot be true or false.

Propositional Logic

- ***Propositional logic*** is a mathematical system for reasoning about propositions and how they relate to one another.
- Every statement in propositional logic consists of ***propositional variables*** combined via ***propositional connectives***.
  - Each variable represents some proposition, such as "You liked it" or "You should have put a ring on it."
  - Connectives encode how propositions are related, such as "If you liked it, then you should have put a ring on it."

Propositional Variables

- Each proposition will be represented by a ***propositional variable***.
- Propositional variables are usually represented as lower-case letters, such as *p, q, r, s,* etc.
- Each variable can take on one of two values: true or false.

Propositional Connectives

- There are seven propositional connectives, many of which will be familiar from programming.
- First, there's the logical "NOT" operation:

$$\neg \boldsymbol{p}$$

- You'd read this out loud as "not *p*."
- The fancy name for this operation is ***logical negation***.

Propositional Connectives

- There are seven propositional connectives, many of which will be familiar from programming.
- Next, there's the logical "AND" operation:

$$\boldsymbol{p \land q}$$

- You'd read this out loud as "*p* and *q*."
- The fancy name for this operation is ***logical conjunction***.

Propositional Connectives

- There are seven propositional connectives, many of which will be familiar from programming.
- Then, there's the logical "OR" operation:

$$\boldsymbol{p \lor q}$$

- You'd read this out loud as "*p* or *q*."
- The fancy name for this operation is ***logical disjunction***. This is an *inclusive* or.

Truth Tables

- A ***truth table*** is a table showing the truth value of a propositional logic formula as a function of its inputs.
- Let's go look at the truth tables for the three connectives we've seen so far:

$$\neg \qquad \wedge \qquad \vee$$

Summary of Important Points

- The ∨ connective is an *inclusive* "or." It's true if at least one of the operands is true.
  - Similar to the `||` operator in C, C++, Java, etc. and the `or` operator in Python.
- If we need an exclusive "or" operator, we can build it out of what we already have.

Try it yourself: Combine the ¬, ∧, and ∨ operators together to form an expression that represents the exclusive or of p and q (something that's true if and only if exactly one of p and q are true).

Respond at pollev.com/zhenglian740

Mathematical Implication

Implication

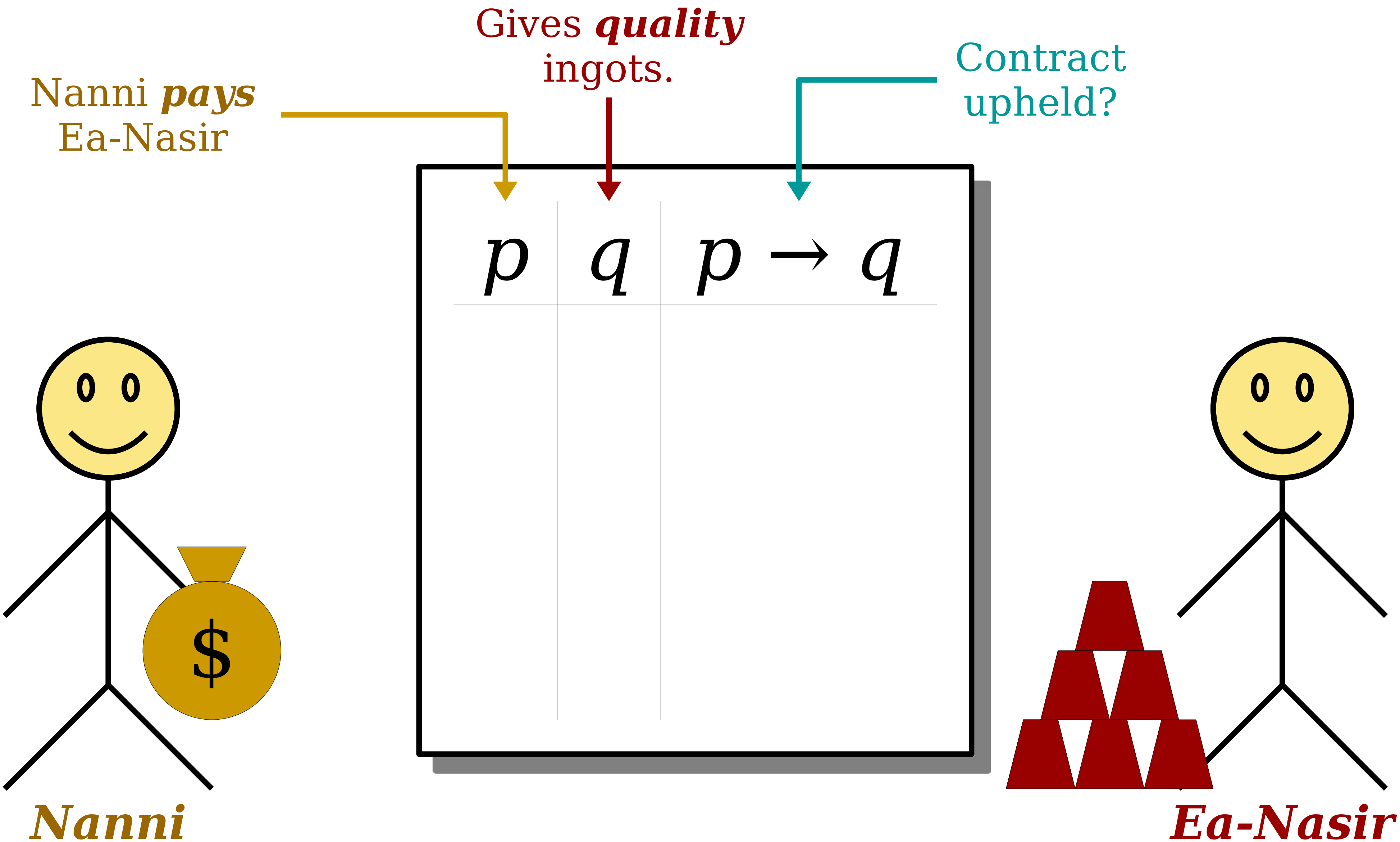
- We can represent implications using this connective:

$$\boldsymbol{p \to q}$$

- You'd read this out loud as "p implies q."
  - The fancy name for this is the ***material conditional***.

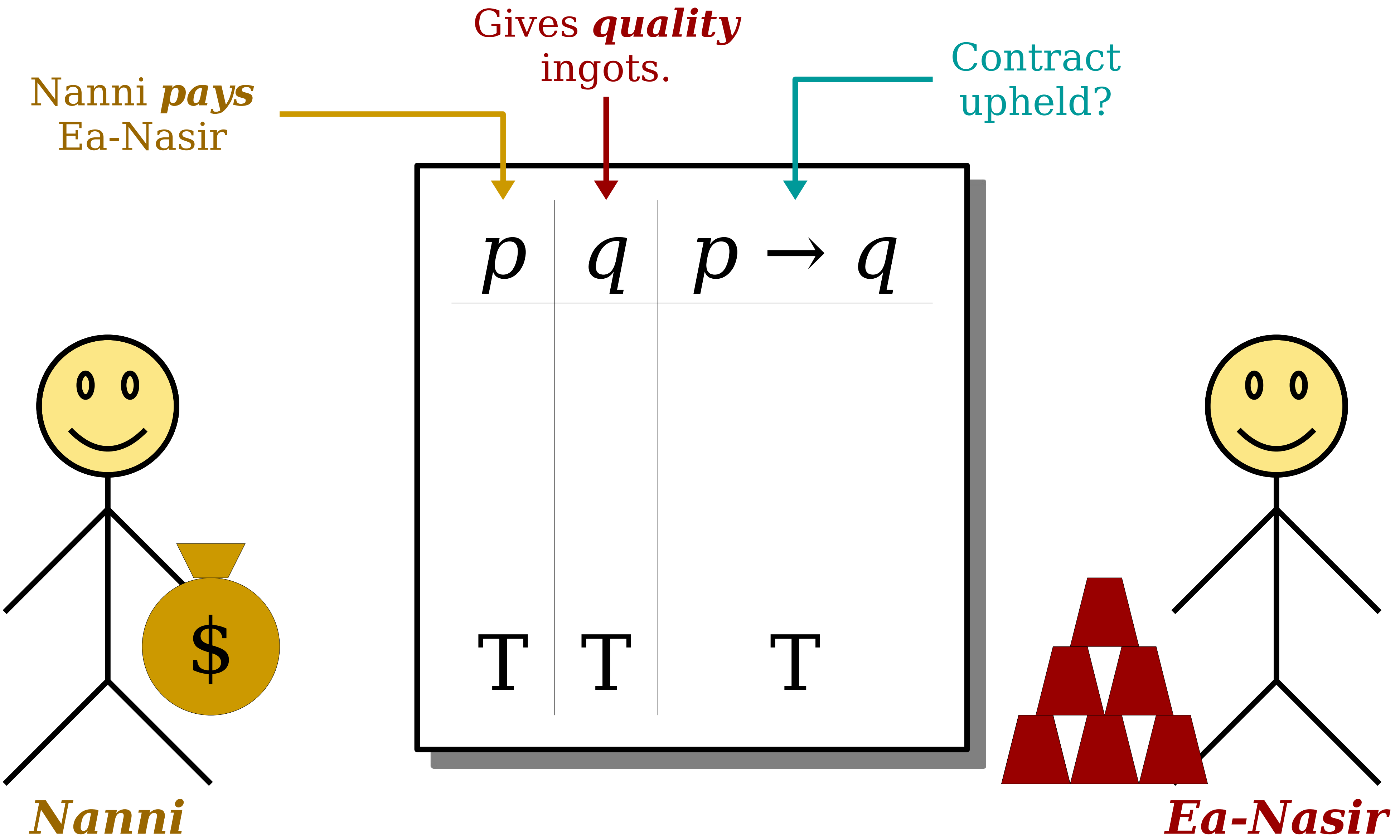
Question: What should the truth table for $p \to q$ look like? Enter your guess as a list of four values to fill in the rightmost column of the table.

Respond at pollev.com/zhenglian740



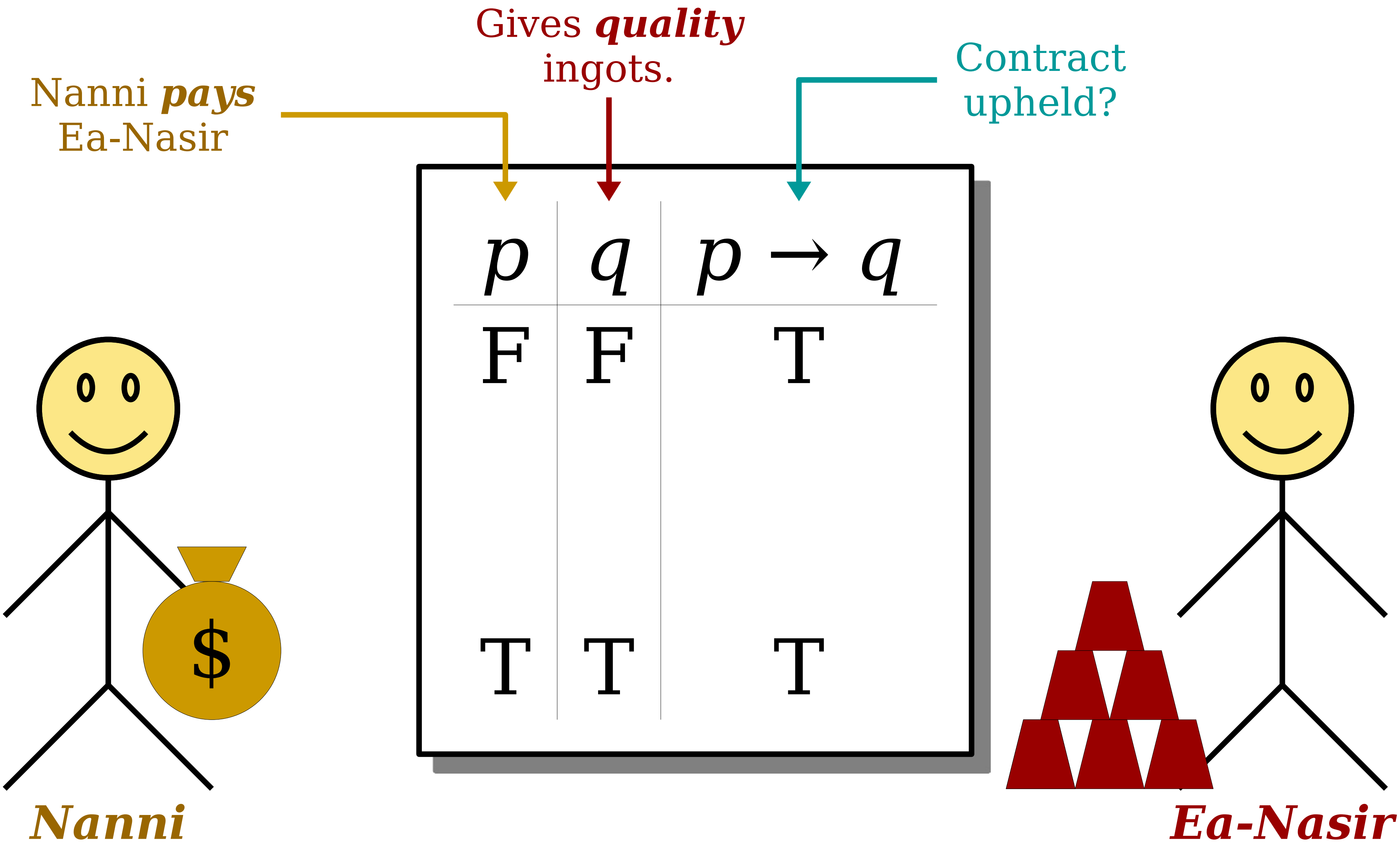
Ancient Contract:

If Nanni pays money to Ea-Nasir, then
Ea-Nasir will give Nanni quality copper ingots.



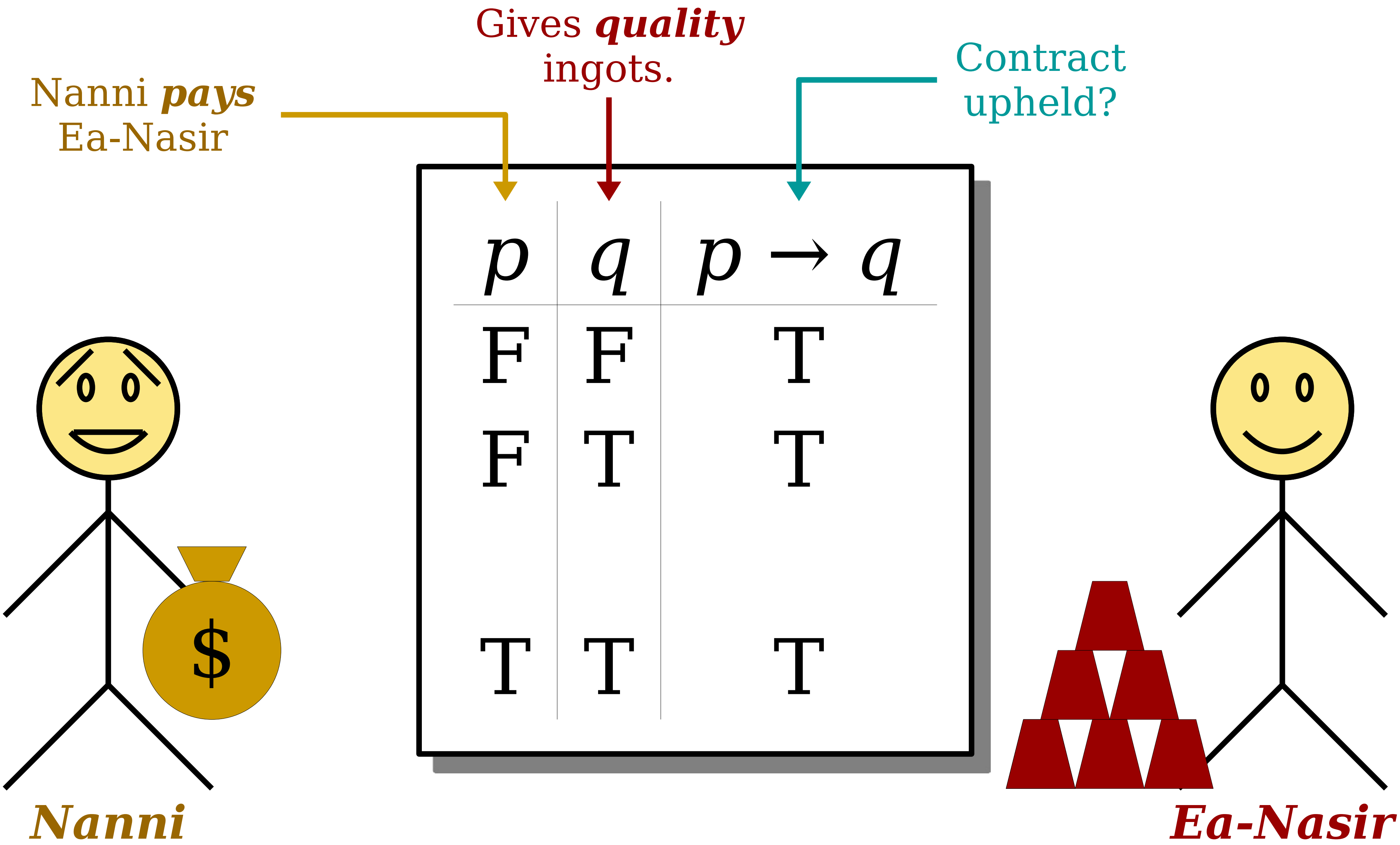
Ancient Contract:

If Nanni pays money to Ea-Nasir, then
Ea-Nasir will give Nanni quality copper ingots.



Ancient Contract:

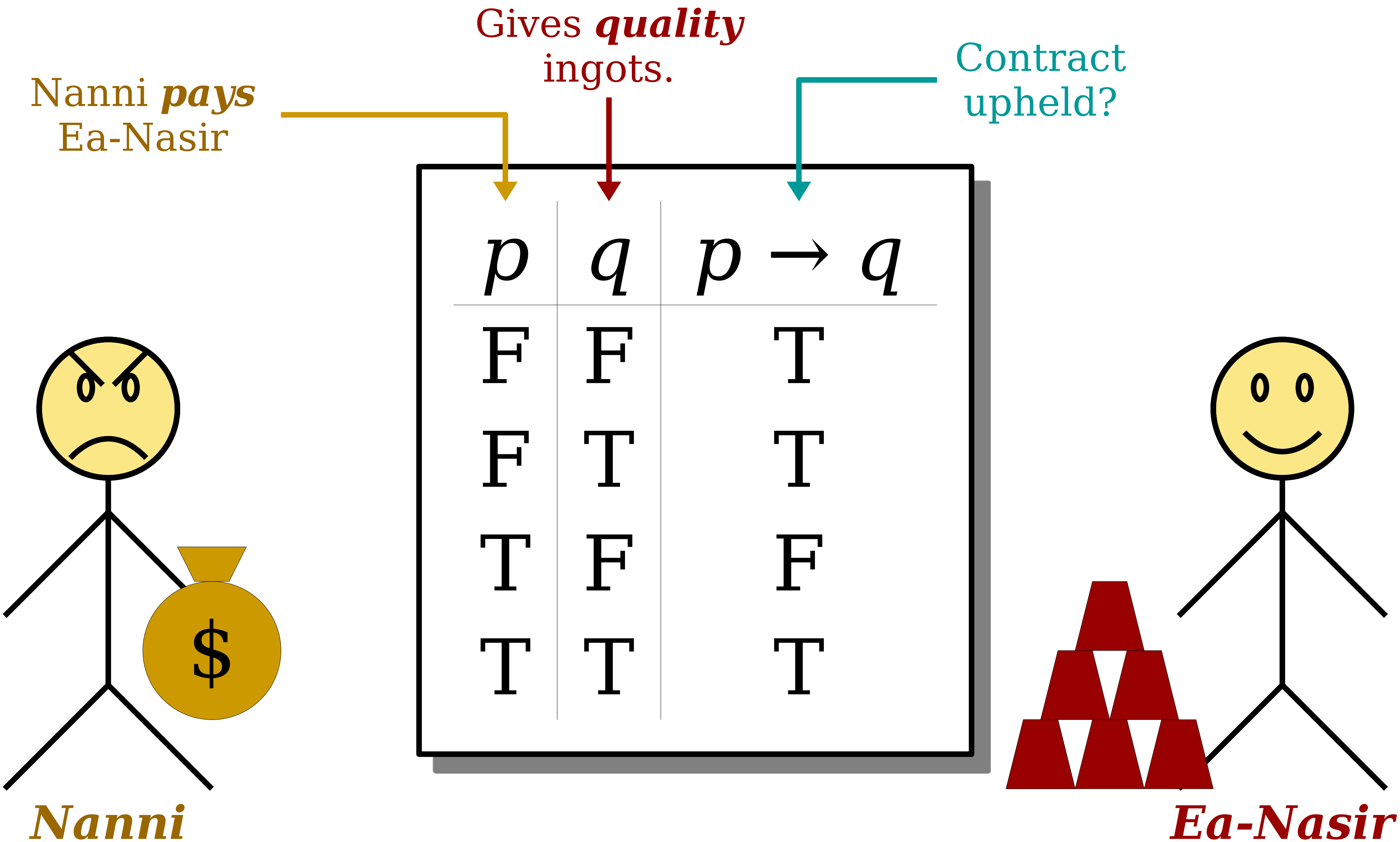
If Nanni pays money to Ea-Nasir, then
Ea-Nasir will give Nanni quality copper ingots.



p	q	$p \to q$
F	F	T
F	T	T
T	T	T

Ancient Contract:

If Nanni pays money to Ea-Nasir, then
Ea-Nasir will give Nanni quality copper ingots.



Ancient Contract:

If Nanni pays money to Ea-Nasir, then
Ea-Nasir will give Nanni quality copper ingots.

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

An implication is false only when the antecedent is true and the consequent is false.

Every formula is either true or false, so these other entries have to be true.

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

Important observation:
The statement $p \to q$ is true whenever $p \land \neg q$ is false.

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

An implication with a false antecedent is called ***vacuously true***.

An implication with a true consequent is called ***trivially true***.

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

Please commit this table to memory. We're going to need it, extensively, over the next couple of weeks.

Fun Fact: The Contrapositive Revisited

The Biconditional Connective

The Biconditional Connective

- On Friday, we saw that "p if and only if q" means both that $p \rightarrow q$ and $q \rightarrow p$.
- We can write this in propositional logic using the ***biconditional*** connective:

$$\boldsymbol{p \leftrightarrow q}$$

- This connective's truth table has the same meaning as "p implies q and q implies p."

Question: What should the truth table for $p \leftrightarrow q$ look like? Enter your guess as a list of four values to fill in the rightmost column of the table.

Respond at pollev.com/zhenglian740

Biconditionals

- The ***biconditional*** connective $p \leftrightarrow q$ is read "p if and only if q."
- Here's its truth table:

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Biconditionals

- The ***biconditional*** connective $p \leftrightarrow q$ is read "p if and only if q."
- Here's its truth table:

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

One interpretation of $\leftrightarrow$ is to think of it as equality: the two propositions must have equal truth values.

True and False

- There are two more “connectives” to speak of: true and false.
  - The symbol ⊤ is a value that is always true.
  - The symbol ⊥ is value that is always false.
- These are often called connectives, though they don't connect anything.
  - (Or rather, they connect zero things.)

Proof by Contradiction

- Suppose you want to prove p is true using a proof by contradiction.
- The setup looks like this:
  - Assume p is false.
  - Derive something that we know is false.
  - Conclude that p is true.
- In propositional logic:

$$(\neg p \to \bot) \to p$$

Operator Precedence

- How do we parse this statement?

$$\neg x \to y \lor z \to x \lor y \land z$$

- Operator precedence for propositional logic:

$$\neg$$

$$\land$$

$$\lor$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$\neg x \to y \vee z \to x \vee y \wedge z$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow y \vee z \rightarrow x \vee y \wedge z$$

- Operator precedence for propositional logic:

$$\neg$$
$$\wedge$$
$$\vee$$
$$\rightarrow$$
$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to y \vee z \to x \vee y \wedge z$$

- Operator precedence for propositional logic:

$$\neg$$
$$\wedge$$
$$\vee$$
$$\to$$
$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to y \lor z \to x \lor (y \land z)$$

- Operator precedence for propositional logic:

$$\neg$$

$$\boldsymbol{\land}$$

$$\lor$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to y \lor z \to x \lor (y \land z)$$

- Operator precedence for propositional logic:

$$\neg$$

$$\land$$

$$\boldsymbol{\lor}$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow (y \vee z) \rightarrow (x \vee (y \wedge z))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to (y \vee z) \to (x \vee (y \wedge z))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to ((y \vee z) \to (x \vee (y \wedge z)))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to ((y \lor z) \to (x \lor (y \land z)))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\land$$

$$\lor$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- The main points to remember:
  - $\neg$ binds to whatever immediately follows it.
  - $\wedge$ and $\vee$ bind more tightly than $\to$.
- We will commonly write expressions like $p \wedge q \to r$ without adding parentheses.
- For more complex expressions, we'll try to add parentheses.
- Confused? ***Please ask!***

The Big Table

Connective	Read Aloud As	C++ Version	Fancy Name
$\neg$	"not"	`!`	Negation
$\wedge$	"and"	`&&`	Conjunction
$\vee$	"or"	`||`	Disjunction
$\to$	"implies"	*see PS2!*	Implication
$\leftrightarrow$	"if and only if"	*see PS2!*	Biconditional
$\top$	"true"	**`true`**	Truth
$\bot$	"false"	**`false`**	Falsity

Time-Out for Announcements!

Office Hours

- Office hours start today. Think of them as “drop-in help hours” where you can ask questions on problem sets, lecture topics, etc.
  - Check the Guide to Office Hours on the course website for the schedule.
- Most office hours are held online. A few are hybrid.
- Once you arrive, sign up on QueueStatus so that we can help people in the order they arrived:

  https://queuestatus.com/queues/2774
- Office hours are much less crowded earlier in the week than later.
- Thursday (July 4th) is a university holiday. We are still planning to host OHs, but watch for any updates.

Finding a Problem Set Partner

Looking for a problem set partner?

- Meet folks in lecture!
- Meet folks in office hours!
- Check out our pinned thread on EdStem!
- Fill out our matchmaking form!
  - First round of matches have been sent out.
  - We will perform a second round of matching this Friday!

Back to CS103!

Recap So Far

- A ***propositional variable*** is a variable that is either true or false.
- The ***propositional connectives*** are
  - Negation: $\neg p$
  - Conjunction: $p \land q$
  - Disjunction: $p \lor q$
  - Implication: $p \to q$
  - Biconditional: $p \leftrightarrow q$
  - True: $\top$
  - False: $\bot$

Translating into Propositional Logic

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

"I won't see a total solar eclipse if I'm not in the path of totality."

Question: How would you express this statement in propositional logic?

Respond at pollev.com/zhenglian740

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

"I won't see a total solar eclipse if I'm not in the path of totality."

$$\neg a \to \neg b$$

“$\boldsymbol{p}$ if $\boldsymbol{q}$”

translates to

$$\boldsymbol{q \rightarrow p}$$

It does *not* translate to

⚠ $\boldsymbol{p \rightarrow q}$ ⚠

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

c: There is a total solar eclipse today.

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

c: There is a total solar eclipse today.

"If I will be in the path of totality, but there's no solar eclipse today, I won't see a total solar eclipse."

Question: How would you express this statement in propositional logic?

Respond at pollev.com/zhenglian740

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

c: There is a total solar eclipse today.

“If I will be in the path of totality, but there's no solar eclipse today, I won't see a total solar eclipse.”

$$a \land \neg c \to \neg b$$

“***p*, but *q***”

translates to

$\boldsymbol{p \wedge q}$

The Takeaway Point

- When translating into or out of propositional logic, be very careful not to get tripped up by nuances of the English language.
    - In fact, this is one of the reasons we have a symbolic notation in the first place!
- Many prepositional phrases lead to counterintuitive translations; make sure to double-check yourself!

Propositional Equivalences

Quick Question:

What would I have to show you to convince you that the statement $\boldsymbol{p \land q}$ is false?

Quick Question:

What would I have to show you to convince you that the statement $p \lor q$ is false?

de Morgan's Laws

- Using truth tables, we concluded that

$$\neg(p \wedge q)$$

  is equivalent to

$$\neg p \vee \neg q$$

- We also saw that

$$\neg(p \vee q)$$

  is equivalent to

$$\neg p \wedge \neg q$$

- These two equivalences are called ***De Morgan's Laws***.

de Morgan's Laws in Code

- ***Pro tip:*** Don't write this:

```
if (!(p() && q())) {
    /* … */
}
```

- Write this instead:

```
if (!p() || !q()) {
    /* … */
}
```

- (This even short-circuits correctly!)

An Important Equivalence

- Earlier, we talked about the truth table for $p \to q$. We chose it so that

$$\boldsymbol{p \to q} \quad \textbf{\textit{is equivalent to}} \quad \boldsymbol{\neg(p \land \neg q)}$$

- Later on, this equivalence will be incredibly useful:

$$\boldsymbol{\neg(p \to q)} \quad \textbf{\textit{is equivalent to}} \quad \boldsymbol{p \land \neg q}$$

Another Important Equivalence

- Here's a useful equivalence. Start with

$$p \to q \quad \textit{is equivalent to} \quad \neg(p \land \neg q)$$

- By de Morgan's laws:

$$\begin{aligned} p \to q \quad & \textit{is equivalent to} \quad \neg(p \land \neg q) \\ & \textit{is equivalent to} \quad \neg p \lor \neg\neg q \\ & \textit{is equivalent to} \quad \neg p \lor q \end{aligned}$$

- Thus $p \to q$ is equivalent to $\neg p \lor q$

Another Important Equivalence

- Here's a useful equivalence. Start with

$$p \to q \quad \textit{is equivalent to} \quad \neg(p \land \neg q)$$

- By de Morgan's laws:

$$p \to q \quad \textit{is equivalent to}$$

$$\textit{is equivalent to}$$

$$\textit{is equivalent to}$$

If p is false, then $\neg p \lor q$ is true.
If p is true, then q has to be true for the whole expression to be true.

- Thus $p \to q$ is equivalent to $\neg p \lor q$

Next Time

- ***First-Order Logic***
  - Reasoning about groups of objects.
- ***First-Order Translations***
  - Expressing yourself in symbolic math!