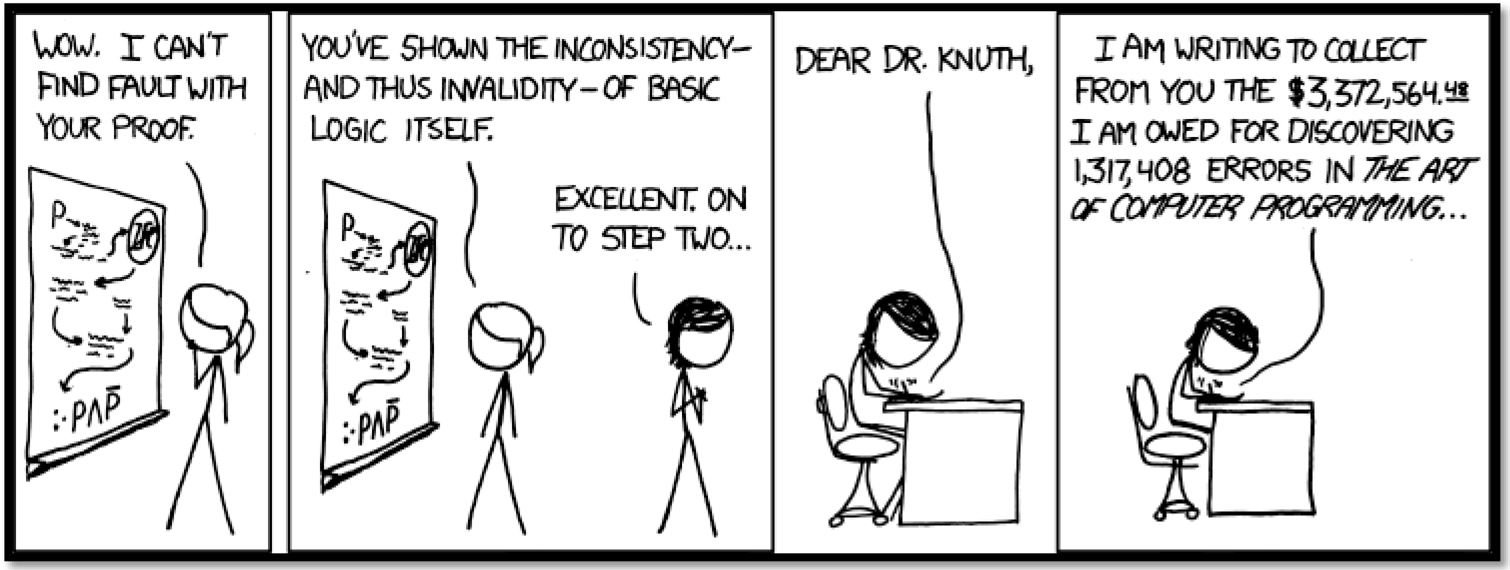
Fall 2015

Lecture 6: Predicate Logic, Logical Inference



$\forall x\ P(x)$

P(x) is true for every x in the domain

read as "for all x, P of x"

$\exists x\ P(x)$

There is an x in the domain for which P(x) is true

read as "there exists x, P of x"

negations of quantifiers

- not every positive integer is prime
- some positive integer is not prime
- prime numbers do not exist
- every positive integer is not prime

$\forall x$ PurpleFruit(x)

Domain:
Fruit

PurpleFruit(x)

Which one is equal to $\neg\forall x$ PurpleFruit(x)?

- $\exists x$ PurpleFruit(x)?

- $\exists x$ $\neg$PurpleFruit(x)?

de Morgan's laws for quantifiers

$$\boxed{\begin{array}{l} \neg\forall x\ P(x) \equiv \exists x\ \neg P(x) \\ \neg\exists x\ P(x) \equiv \forall x\ \neg P(x) \end{array}}$$

de Morgan's laws for quantifiers

$$\boxed{\begin{array}{l} \neg \forall x \; P(x) \equiv \exists x \neg P(x) \\ \neg \exists x \; P(x) \equiv \forall x \neg P(x) \end{array}}$$

"There is no largest integer."

$$\begin{aligned} & \neg \exists x \; \forall y \; (x \geq y) \\ \equiv \; & \forall x \neg \forall y \; (x \geq y) \\ \equiv \; & \forall x \; \exists y \neg (x \geq y) \\ \equiv \; & \forall x \; \exists y \; (y > x) \end{aligned}$$

"For every integer there is a larger integer."

scope of quantifiers

example: Notlargest(x) $\equiv \exists y$ Greater (y, x)

$\equiv \exists z$ Greater (z, x)

truth value:

doesn't depend on y or z "bound variables"

does depend on x "free variable"

quantifiers only act on free variables of the formula they quantify

$\forall x\, (\exists y\, (P(x, y) \rightarrow \forall x\, Q(y, x)))$

example:

Domain = positive integers

$\text{IsMultiple}(x, y)$ = “x is a multiple of y”

$$\forall x \left((x > 1 \wedge \neg(x = y)) \rightarrow \neg \text{IsMultiple}(y, x) \right)$$

$$\equiv \text{Prime}(y)$$

$$\forall x \, \exists y \left((x < y) \wedge \text{Prime}(y) \right)$$

$$\forall x \, \exists y \left((x < y) \wedge \left(\forall x \left((x > 1 \wedge \neg(x = y)) \rightarrow \neg \text{IsMultiple}(y, x) \right) \right) \right)$$

example:

Domain = positive integers

$\text{IsMultiple}(x, y) =$ "x is a multiple of y"

$$\forall x\ \big((x > 1 \wedge \neg(x = y)) \rightarrow \neg\text{IsMultiple}(y, x)\big)$$

$$\equiv \text{Prime}(y)$$

$$\forall x\ \exists y\ \big((x < y) \wedge \text{Prime}(y) \wedge \text{Prime}(y + 2)\big)$$

$$\forall x\ \exists y \begin{pmatrix} (x < y) \wedge \Big(\forall x\ \big((x > 1 \wedge \neg(x = y)) \rightarrow \neg\text{IsMultiple}(y, x)\big)\Big) \\ \wedge \Big(\forall x\ \big((x > 1 \wedge \neg(x = y)) \rightarrow \neg\text{IsMultiple}(y, x)\big)\Big) \end{pmatrix}$$

example:

Domain = positive integers

$\text{IsMultiple}(x, y) =$ "x is a multiple of y"

$$\forall x \left((x > 1 \land \neg(x = y)) \to \neg\text{IsMultiple}(y, x) \right)$$

$$\equiv \text{Prime}(y)$$

$$\forall x\, \exists y \left((x < y) \land \text{Prime}(y) \land \text{Prime}(y+2) \land (x < y^2) \right)$$

$$\forall x\, \exists y \begin{pmatrix} (x < y) \land \left(\forall x \left((x > 1 \land \neg(x = y)) \to \neg\text{IsMultiple}(y, x) \right) \right) \\ \land \left(\forall x \left((x > 1 \land \neg(x = y)) \to \neg\text{IsMultiple}(y, x) \right) \right) \land (x < y^2) \end{pmatrix}$$

$\exists x\ (P(x) \land Q(x))$ vs. $\exists x\, P(x) \land \exists x\, Q(x)$

- Bound variable names don't matter

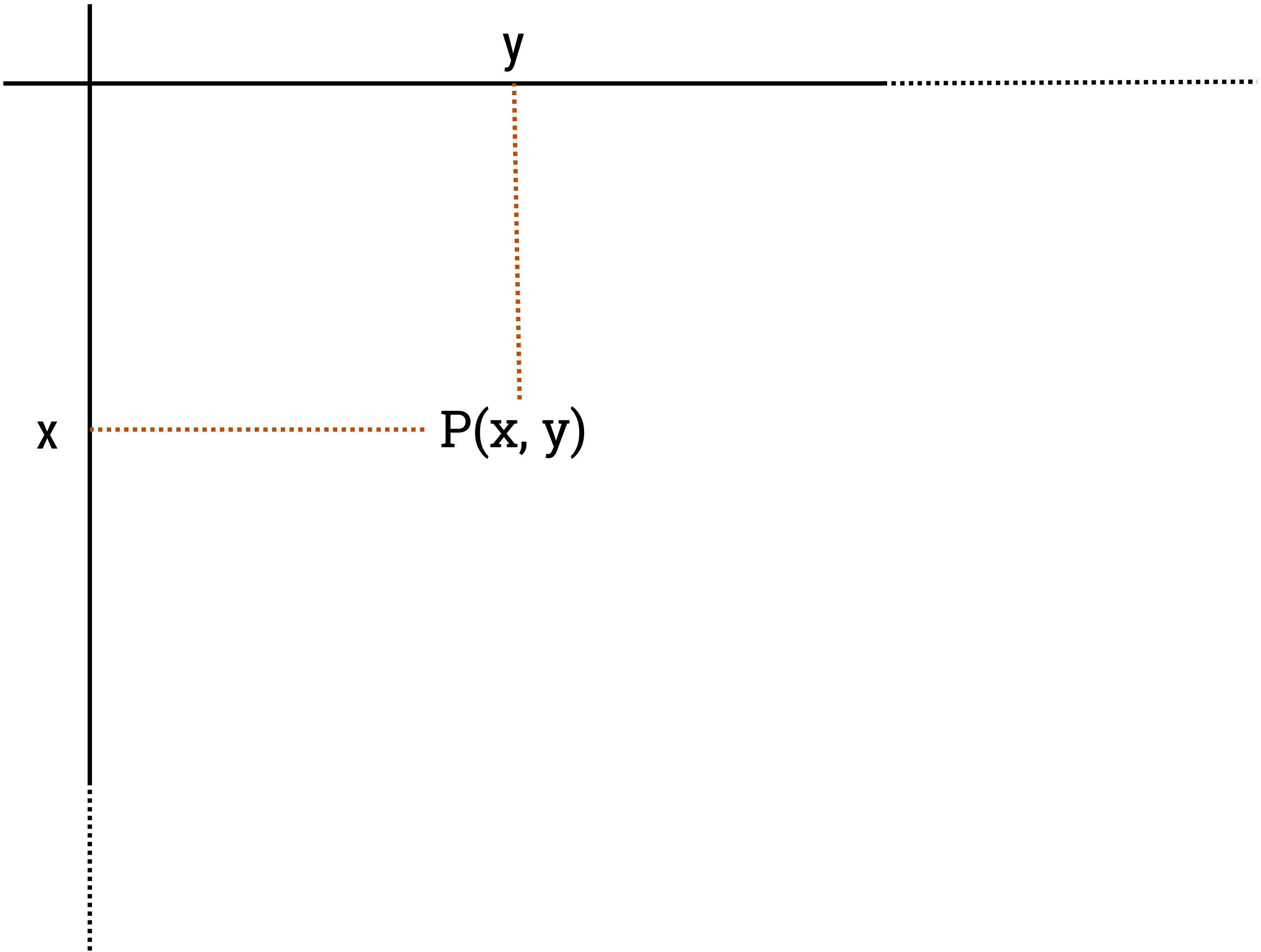
$$\forall x \exists y\, P(x, y) \equiv \forall a \exists b\, P(a, b)$$

- Positions of quantifiers can sometimes change

$$\forall x\, (Q(x) \wedge \exists y\, P(x, y)) \equiv \forall x \exists y\, (Q(x) \wedge P(x, y))$$

- But: order is important...

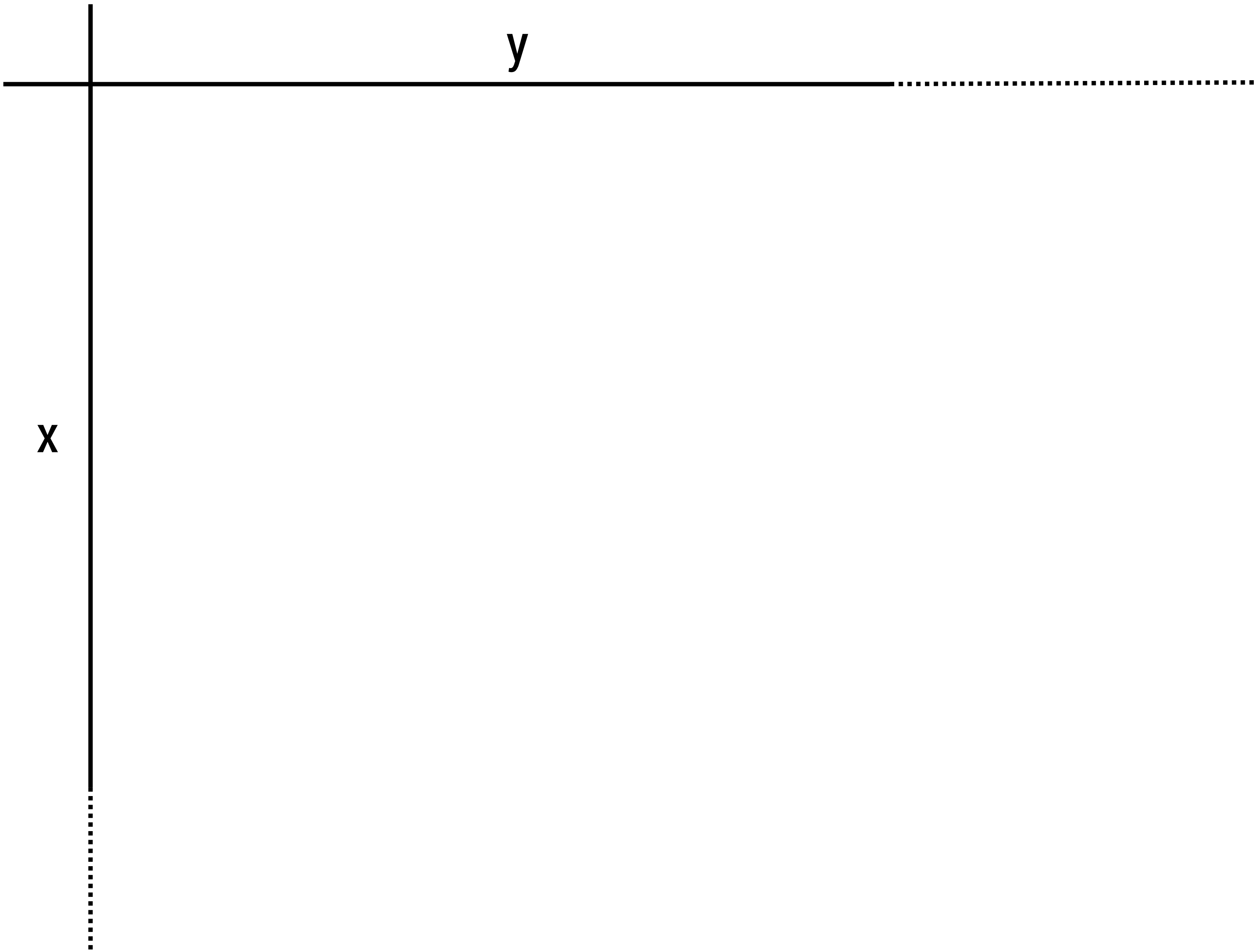
predicate with two variables



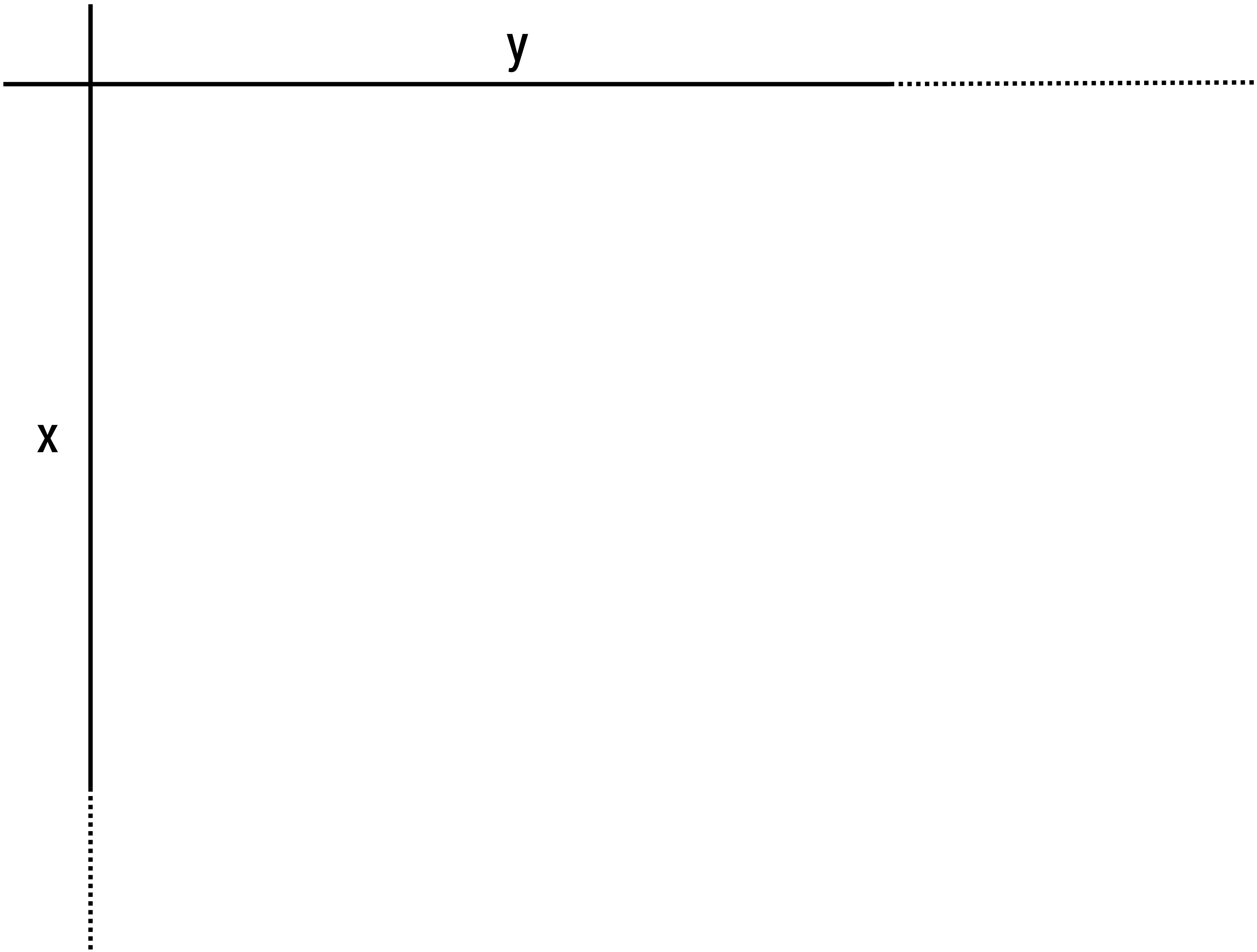
quantification with two variables

expression	**when true**	**when false**
$\forall x \forall y\, P(x, y)$		
$\exists x \exists y\, P(x, y)$		
$\forall x \exists y\, P(x, y)$		
$\exists x \forall y\, P(x, y)$		

$$\forall x \, \forall y \, P(x, y)$$



$\exists x\, \exists y\, P(x, y)$

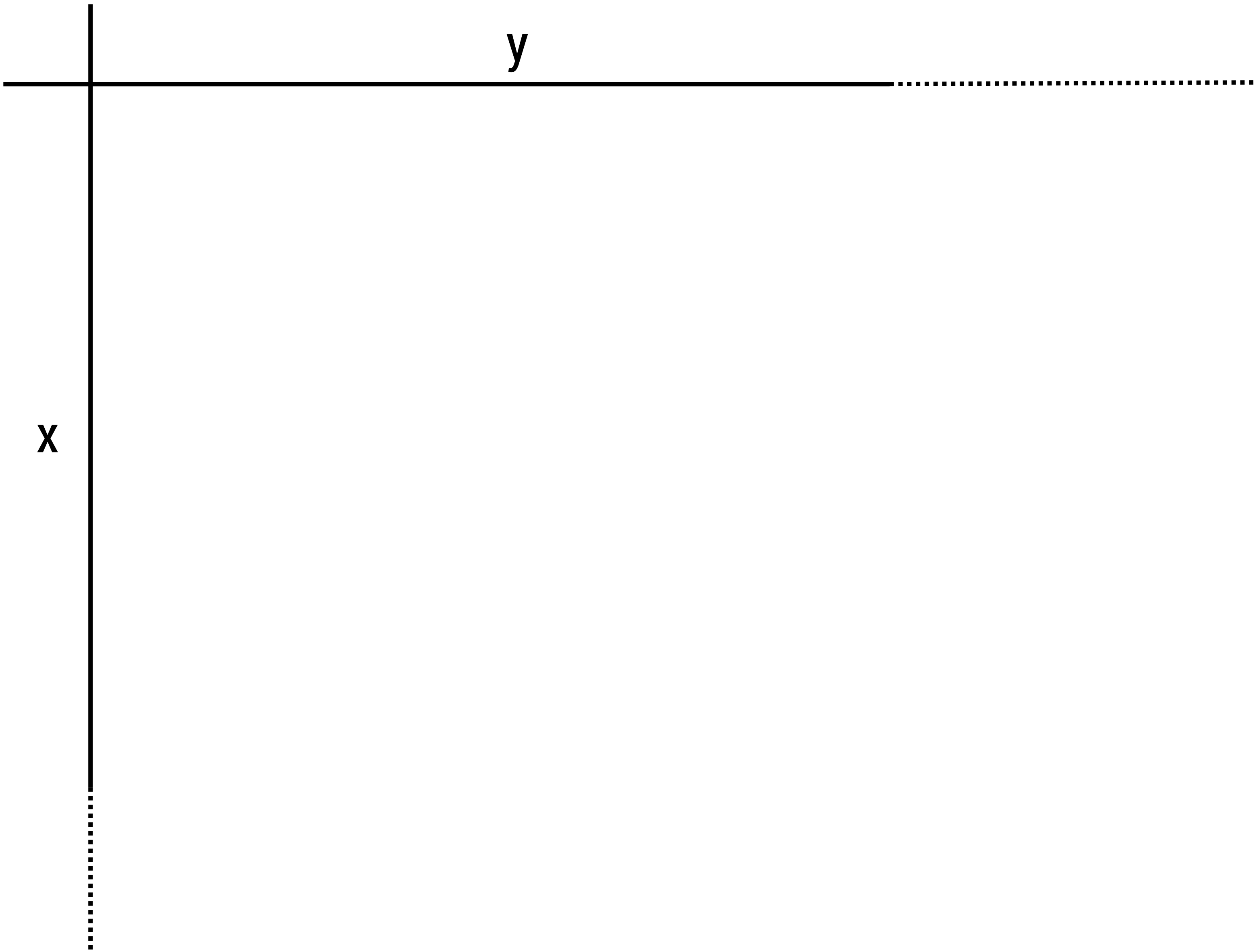


$\forall x \, \exists y \, P(x, y)$

y

x

$$\exists x \, \forall y \, P(x, y)$$



quantification with two variables

expression	**when true**	**when false**
$\forall x \forall y P(x, y)$		
$\exists x \exists y P(x, y)$		
$\forall x \exists y P(x, y)$		
$\exists x \forall y P(x, y)$		

- So far we've considered:
  - How to understand and *express* things using propositional and predicate logic
  - How to *compute* using Boolean (propositional) logic
  - How to show that different ways of expressing or computing them are *equivalent* to each other
- Logic also has methods that let us *infer* implied properties from ones that we know
  - Equivalence is only a small part of this

applications of logical inference

- Software Engineering
  - Express desired properties of program as set of logical constraints
  - Use inference rules to show that program implies that those constraints are satisfied
- Artificial Intelligence
  - Automated reasoning
- Algorithm design and analysis
  - e.g., Correctness, Loop invariants.
- Logic Programming, e.g. Prolog
  - Express desired outcome as set of constraints
  - Automatically apply logic inference to derive solution

foundations of rational thought…

- Start with hypotheses and facts
- Use rules of inference to extend set of facts
- Result is proved when it is included in the set

an inference rule: *Modus Ponens*

- If p and p $\rightarrow$ q are both true then q must be true

- Write this rule as

$$\frac{p,\ p \rightarrow q}{\therefore\ q}$$

- Given:
  - If it is Monday then you have a 311 class today.
  - It is Monday.

- Therefore, by modus ponens:
  - You have a 311 class today.

Show that r follows from p, $p \rightarrow q$, and $q \rightarrow r$

1.	p	given
2.	$p \rightarrow q$	given
3.	$q \rightarrow r$	given
4.	q	modus ponens from 1 and 2
5.	r	modus ponens from 3 and 4

proofs can use equivalences too

Show that $\neg p$ follows from $p \rightarrow q$ and $\neg q$

1.	$p \rightarrow q$	given
2.	$\neg q$	given
3.	$\neg q \rightarrow \neg p$	contrapositive of 1
4.	$\neg p$	modus ponens from 2 and 3

- Each inference rule is written as:

$$\frac{A,\ B}{\therefore\ C, D}$$

...which means that if both A and B are true then you can infer C and you can infer D.

  – For rule to be correct $(A \wedge B) \rightarrow C$ and $(A \wedge B) \rightarrow D$ must be a tautologies

- Sometimes rules don't need anything to start with. These rules are called axioms:
  – e.g. *Excluded Middle Axiom*

$$\frac{\quad}{\therefore\ p \vee \neg p}$$

simple propositional inference rules

Excluded middle plus two inference rules per binary connective, one to eliminate it and one to introduce it:

$$\frac{p \wedge q}{\therefore\ p,\ q} \qquad \frac{p,\ q}{\therefore\ p \wedge q}$$

$$\frac{p \vee q\ ,\ \neg p}{\therefore\ q} \qquad \frac{p}{\therefore\ p \vee q,\ q \vee p}$$

$$\frac{p,\ p \rightarrow q}{\therefore\ q} \qquad \frac{p \Rightarrow q}{\therefore\ p \rightarrow q}$$

Direct Proof Rule

Not like other rules

important: applications of inference rules

- You can use equivalences to make substitutions of any sub-formula.
- Inference rules only can be applied to whole formulas (not correct otherwise)

e.g.	1.	$p \rightarrow q$	given
	2.	$(p \vee r) \rightarrow q$	intro $\vee$ from 1.

Does not follow! e.g . p=**F**, q=**F**, r=**T**

direct proof of an implication

- $p \Rightarrow q$ denotes a proof of q given p as an assumption

- The direct proof rule:

  If you have such a proof then you can conclude that $p \rightarrow q$ is true

Example:

proof subroutine

1.	p	**assumption**
2.	$p \vee q$	intro for $\vee$ from 1
3.	$p \rightarrow (p \vee q)$	direct proof rule