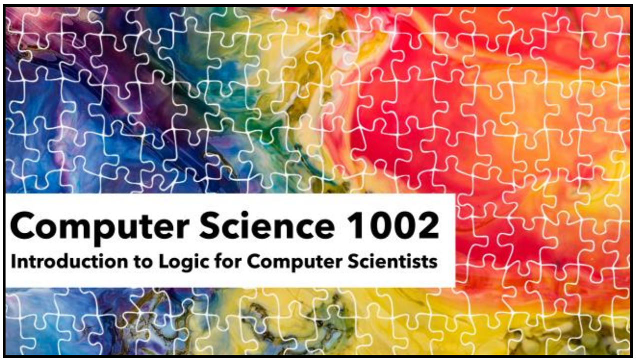


1

Stereotypes puzzle

- Susan is 28 years old, single, outspoken, and very bright. She majored in philosophy. As a student she was deeply concerned with issues of discrimination and social justice and also participated in anti-pipeline demonstrations.

  Please rank the following possibilities by how likely they are. List them from least likely to most likely. Susan is:

1. a kindergarten teacher
2. works in a bookstore and takes yoga classes
3. an active feminist
4. a psychiatric social worker
5. a member of an outdoors club
6. a bank teller
7. an insurance salesperson
8. a bank teller and an active feminist

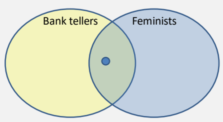
2

Stereotypes puzzle

- Susan is 28 years old, single, outspoken, and very bright. She majored in philosophy. As a student she was deeply concerned with issues of discrimination and social justice and also participated in anti-pipeline demonstrations.

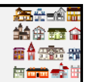
  Please rank the following possibilities by how likely they are. List them from least likely to most likely. Susan is:

1. a kindergarden teacher
2. works in a bookstore and takes yoga classes
3. an active feminist
4. a psychiatric social worker
5. a member of an outdoors club
6. a bank teller
7. an insurance salesperson
8. a bank teller and an active feminist



3

Scenarios and sets



- Want to reason about more general scenarios
- Rather than just true/false, vary over objects:
  - even numbers, integers, primes
  - people that are and are not bank tellers,
  - pairs of animals in the same ecosystem...
- Want multiple properties of these objects:
  - an even number that is divisible by 4 and > 10,
  - a person that is also a bank teller...

4

5

Sets

- A **set** is a collection of objects.
  - The order of elements does not matter.
  - There are no duplicates.

  - S_1={1, 2, 3},
  - S_2 = {Cathy, Alaa, Keiko, Daniela}
- Can define a set of all elements satisfying some condition:
  - {x | such that x ... } is called **set builder notation**

  - $S_3 = \{\, x \mid x$ is an even number }
  - PEOPLE = {x | x is a person living on Earth now}
  - BANKTELLERS = { x | x is a person who is a bank teller}

6

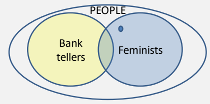
Special sets

- Notation for some special sets:
  - Empty set $\emptyset$ (set with no elements in it)
  - Natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$
    - Sometimes start with 0.
  - Integers $\mathbb{Z} = \{\ldots -2, -1, 0, 1, 2, \ldots\}$
  - Rational numbers $\mathbb{Q} = \left\{\frac{m}{n} \;\middle|\; m, n \text{ integers}, n \neq 0\right\}$
  - Real numbers $\mathbb{R}$
  - Binary strings $\{0,1\}^*$

7

Set elements

- Notation $a \in S$ ("a in S") means that an element a belongs to the set S, and $a \notin S$ states that a is not in S
  - Susan ∈ PEOPLE. Susan ∉ BANKTELLERS
  - $0.5 \in \mathbb{R}$. $0.5 \notin \mathbb{Z}$

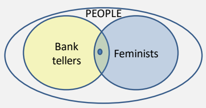


- Also, can write $x \in S$ for a variable x (in particular in set builder notation)
  - BANKTELLERS = { $x \in PEOPLE$ | x is a bank teller}
  - EVEN = { $x \in \mathbb{Z}$| x is divisible by 2 }

8

Subsets and equality

- When all elements of a set S_1 are also elements of a set S_2, we say that S_1 is a **subset** of S_2, written $S_1 \subseteq S_2$
  - BANKTELLERS ⊆ PEOPLE
  - $EVEN \subseteq \mathbb{Z}$
  - $\mathbb{N} \subseteq \{x \in \mathbb{Z} \mid x \geq 1\}$

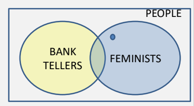


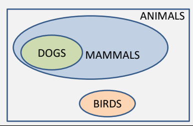
- $S_1 = S_2$ when $S_1 \subseteq S_2$ and $S_2 \subseteq S_1$
  - $\mathbb{N} = \{x \in \mathbb{Z} \mid x \geq 1\}$
- S_1 is a **proper subset** of a S_2, written $S_1 \subset S_2$ when $S_1 \subseteq S_2$, but $S_1 \neq S_2$
  - BANKTELLERS ⊂ PEOPLE, $EVEN \subset \mathbb{Z}$.

9

Venn diagrams

- Circles are sets
  - BANKTELLERS, FEMINISTS, DOGS, MAMMALS, BIRDS, EVEN
- Square is the universe
  - PEOPLE, ANIMALS, $\mathbb{R}$
- Dots are elements
  - Susan, 0.5
- A circle for one set inside another means subset
  - DOGS ⊆MAMMALS
- Circles not overlapping indicates there are no common elements (they are **disjoint**)
  - MAMMALS is disjoint from BIRDS





How can we do logic with sets?

10

11

Predicates

- A **predicate** $P(x)$ is a "proposition with a variable". Here, x is a variable that can take values from a set D called the **domain** or **universe** of $P(x)$.
  - Variable names are usually x, y, z, or words like $name$, usually lowercase.
  - Predicate names are usually P(), $Q(\ \ )$, and capitalized words:
    - $P(y)$, $Q(z)$, $P(name)$, Even(x)
- Examples: Feminist(x), Bankteller(person), P(y).
  - In the first two cases, D is PEOPLE.
  - $x \in S$ is also a predicate for a specific S with the variable x
    - Here, D is the universe to which elements of S belong.

12

Instantiation of predicates

- When x in $P(x)$ is replaced with a specific element of D (**instantiated**), the predicate becomes a proposition.
  - Even(3), Feminist(Susan)
- After instantiation, the predicate gets a specific truth value true or false.
  - Even(3) is false. Feminist(Susan) is true.
- $P(x)$ may be true for some values of $x \in D$, and false for other.
  - $Even(x)$ is true for even numbers $x \in \mathbb{Z}$, but false for odd integers.
  - $Feminist(y)$ is true for some $y \in PEOPLE$, and false for others…
  - Here, domain of x is $\mathbb{Z}$, and domain of y is $PEOPLE$
    - $Even(y)$ is not defined for $y \in PEOPLE$, only for elements of $\mathbb{Z}$.

13

Predicates vs. sets

- Predicates and sets are two sides of the same coin
  - For each set S there is a predicate which is true exactly on elements of S
  - For each predicate P there is a set S of values of x on which P is true.

Set S	Predicate P
A collection of elements	Becomes true/false on a given element
$S_P = \{x \mid P(x) \text{ is true}\}$	$P_S(x) \equiv "x \in S"$

- To write formulas, need something that is true/false: predicates!

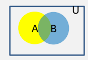
14

Predicate logic

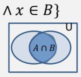
- We can make formulas out of predicates the same way as we did for propositions, but now our formulas have **free variables**:
  - $Even(x) \vee DivisibleByThree(x) \rightarrow \neg Prime(x)$
  - $Feminist(x) \wedge Bankteller(x)$
  - Now scenarios can correspond to values of x.
    - The first formula is false for x=2 since $Even(2) = true$, but $\neg Prime(2) = false$.
- This is called **predicate logic (**or **first-order logic),** as opposed to propositional logic we did so far.

15

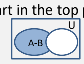
Set operations as formulas



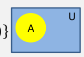
- Let A and B be sets with predicates $x \in A, x \in B$
  - Such as A={1,2,3} and B={ 2,3,4}
- **Intersection** $A \cap B = \{ x \mid x \in A \wedge x \in B\}$
  - The green part of top picture
  - $A \cap B = \{2,3\}$



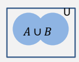
- **Difference** $A - B = \{x \mid x \in A \wedge \neg(x \in B)\}$
  - The yellow part in the top picture.
  - $A - B = \{1\}$



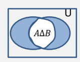
- **Complement** $\overline{A} = \{x \in U \mid \neg(x \in A)\}$
  - The blue part on the Venn diagram
  - If universe U = $\mathbb{N}$, $\overline{A} = \{x \in \mathbb{N} \mid x \notin \{1,2,3\}\}$



- **Union**
  - $A \cup B = \{ x \mid x \in A \vee x \in B\}$
  - The coloured part in the top picture.
  - $A \cup B = \{1,2,3,4\}$



- **Symmetric difference**
  - $A \,\Delta\, B = (A - B) \cup (B - A)$
  - The yellow and blue parts of the top picture.
  - $A\Delta B = \{1,4\}$



16

Predicates with several variables

- Sometimes, want a predicate that depends on more than one variable
  - $LessThan(x, y)$, for $x, y \in \mathbb{R}$, is true if and only if $x < y$
    - Alternatively, can just write $x < y$ to mean $LessThan(x, y)$
  - $Divides(x, y)$, for $x, y \in \mathbb{Z}$, is true if and only if x is a divisor of y
    - $Divides(3,6)$ is true. $Divides(12,4)$ is false.
- A **predicate** $P(x_1, \dots, x_n)$ is a "proposition with variables", where values of the variables $x_1, \dots, x_n$ come from some sets $S_1, \dots, S_n$, called their **domains** or **universes.**
  - Sometimes the domains of its variables are the same, other times different.
  - Order of variables matters.
- For any specific tuple of elements (instantiation of $x_1, \dots, x_n$), a predicate $P(x_1, \dots, x_n)$ is either true or false.
  - For any pair of numbers, x < y can be either true or false:
    - LessThan(1,2), that is $1 < 2$ is true, whereas LessThan(2,1) is false

17

Predicates: arity

- A predicate on one variable is called "unary", on two "binary", on three variables "ternary"
  - in general a predicate on n variables is called n-ary predicate.
  - Number of variables a predicate takes as input is called its $arity$.
- Parent(x,y) is true when the person x is a parent of the person y
  - $Parent$(King George VI, Queen Elizabeth II) is true
  - This is a binary predicate
- Ternary predicates examples:
  - $Sum(x, y, z)$ which is true when $x + y = z$.
  - $Between(x, y, z)$ which is true when $y \leq x$ and $x \leq z$
    - For both of them, can take the domain to be $\mathbb{R}$
  - $Registrations(name, cour, sem)$ which is true when $name$ is the name of a student who takes the course $cour$ in semester sem.

18

19

Quantifiers

- A formula of predicate logic can be evaluated when the values of all variables are known.
  - Alternatively, we might want the formula to hold no matter what the values are.
  - Or wonder if there is any value that makes it true

20

Universal quantifier

- $\forall x \in S\ F(x)$

(pronounced “forall”)

is true when

for all possible values x can take in S

the formula $F(x)$ is true.

- If there is a quantifier over a variable, such variable is not free anymore.
- If there are no free variables, the whole formula evaluates to either true or false.

21

Quantifiers: universal (∀)

- Theorems often look like this: “For all x the following is true”, and then a formula with x as a free variable.
  - For all $x \in \mathbb{Z}$ $DivisibleBySix(x) \rightarrow DivisibleByThree(x)$
  - For all $n \in \mathbb{N}$ if $n > 4$, then $2^n > n^2$
- We write this in predicate logic using the universal quantifier ∀
  - $\forall x \in \mathbb{Z}\ \ DivisibleBySix(x) \rightarrow DivisibleByThree(x)$
  - $\forall n \in \mathbb{N}\ \ n > 4 \rightarrow 2^n > n^2$

Some textbooks put a comma after the quantifier, some period, others nothing.
Some insist that the formula must be in parentheses if it is more than one predicate.

- When in doubt, use parentheses.

22

Examples with universal quantifier

- “Every bird can fly”
  - Domain $BIRD$
  - Predicate $CanFly(x)$
  - $\forall x \in BIRD\quad CanFly(x)$
    - Not true: penguins

“Any two people are related”
Domain $PEOPLE$
Predicate Related(x, y)
$\forall x \in PEOPLE\ \forall y \in PEOPLE\quad Related\ (x, y)$
Can shorten this as
$\forall x, y\ \in PEOPLE\quad Related(x, y)$

- “I work each day of the week! ☹ “
  - Domain DAYS = {Mon, Tue, Wed, Thu, Fri, Sat, Sun}
  - Predicate Work(x), where $x \in DAYS$
  - $\forall x \in DAYS\quad Work(x)$

23

Examples with universal quantifier

- “Every integer is either even or odd.”
  - Domain $\mathbb{Z}$
  - Predicates $Even(x)$ and $Odd(x)$
  - $\forall x \in \mathbb{Z}\ \ Even(x) \vee Odd(x)$
    - This is true: every integer is even or odd.
- “Adding 1 to any odd integer results in an even number”
  - $\forall x \in \mathbb{Z}\quad Odd(x) \rightarrow Even(x+1)$
    - This is also true.
- “Every odd integer is prime”
  - Domain $\mathbb{Z}$
  - Predicates Prime(x) and $Odd(x)$
  - $\forall x \in \mathbb{Z}\ \ Odd(x) \rightarrow Prime(x)$
    - False: for example 9 is odd but not prime.
- “For every two numbers one is less than the other”
  - Domain $\mathbb{R}$
  - Predicate LessThan(x, y)
    - Can just write it as $x < y$
  - $\forall x \in \mathbb{R}\ \forall y \in \mathbb{R}\ \ x < y \vee y < x$
    - False: x and y may take the same value

24

25

Universal quantifier and truth

- For every formula F of predicate logic with a free variable x, we can write $\forall x \in S\ F(x)$
  - Where $F(a)$ can have quantifiers, too.
  - We call a sentence of the form $\forall x \in S\ \ F(x)$ a **universal statement**
- The formula "$\forall x \in S\ \ F(x)$" is true iff $F(a)$ is true for every $a \in S$.
  - That is, if $a_1, a_2, \dots, a_n, \dots$ is a list of all elements of S, then "$\forall x \in S\ F(x)$" is true iff "$F(a_1) \wedge F(a_2) \wedge \cdots \wedge F(a_n) \wedge \cdots$" is true.
- To evaluate a universally quantified formula $\forall x \in S\ \ F(x)$, check that for each $a \in S$, $F(a)$ is true.

26

Evaluating universally quantified formulas

- Take the domain $D = \{0,1,2\}$, predicates $x \leq y$ and $x \geq y$
  - "$\forall x \in \{0,1,2\}\ \ x \geq 0$" is true for all instantiations of $x \geq 0$:
    $x \geq 0$ true for 0, and 1, and 2 substituted for x in the formula.
  - So $0 \geq 0$ is true, and $1 \geq 0$ is true, and $2 \geq 0$ is also true.
  - That is, when $0 \geq 0\ \wedge 1 \geq 0 \wedge 2 \geq 0$ is true.
    - Which happens to be the case.
  - "$\forall x \in \{0,1,2\}\ \ x \geq 0\ \ \wedge x \leq 2$" is true when
    - $(0 \geq 0 \wedge 0 \leq 2) \wedge (1 \geq 0 \wedge 1 \leq 2) \wedge (2 \geq 0 \wedge 2 \leq 2)$ is true.

27

Evaluating universally quantified formulas

- Take the domain $D = \{0,1,2\}$, predicates $x \leq y$ and $x \geq y$
  - Let's evaluate $\forall x \in D\ \forall y \in D\ \ x \leq y$
  - Now we have two variables, so let's do it one quantifier at a time.
  - First, let's try all values for x then AND the resulting formulas together:
    $\forall x \in D\ \forall y \in D\ \ x \leq y \equiv (\forall y\ \in D\ 0 \leq y) \wedge (\forall y\ \in D\ 1 \leq y) \wedge (\forall y\ \in D\ 2 \leq y)$
  - Now, for each formula in parentheses, try all values of y:
    $\forall y\ \in D\ 0 \leq y \equiv (0 \leq 0) \wedge (0 \leq 1) \wedge (0 \leq 2)$ true
    $\forall y\ \in D\ 1 \leq y \equiv (1 \leq 0) \wedge (1 \leq 1) \wedge (1 \leq 2)$ false
    $\forall y\ \in D\ 2 \leq y \equiv (2 \leq 0) \wedge (2 \leq 1) \wedge (2 \leq 2)$ false
  - Finally, putting it all together, $\forall x \in D\ \forall y \in D\ \ x \leq y$ becomes
    $\big((0 \leq 0) \wedge (0 \leq 1) \wedge (0 \leq 2)\big) \wedge \big((1 \leq 0) \wedge (1 \leq 1) \wedge (1 \leq 2)\big) \wedge ((2 \leq 0) \wedge (2 \leq 1) \wedge (2 \leq 2))$
  - Now we see that $\forall x \in D\ \forall y \in D\ \ x \leq y$ is false
- Evaluating predicates with infinite domains is harder, need proofs.

28

29

Quantifiers and conditionals

- Which of these are true? How can we write them as formulas?
  - All squares are white. All white shapes are squares
  - All circles are blue. All blue shapes are circles.

30

Quantifiers and conditionals

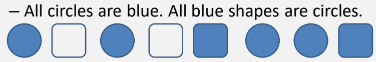
- Which of these are true? How can we write them as formulas?
  - All squares are white. All white shapes are squares
  - All circles are blue. All blue shapes are circles.

  - They all have the structure $\forall x \in S,\ P(x) \rightarrow Q(x)$

31

Quantifiers and conditionals

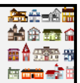
- Which of these are true? How can we write them as formulas?
  - All squares are white. All white shapes are squares
  - All circles are blue. All blue shapes are circles.



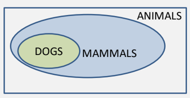
  - They all have the structure $\forall x \in S,\ P(x) \rightarrow Q(x)$
    - All squares are white: for all shapes, if it is a square, then it is white
      - $\forall x \in SHAPES,\ Square(x) \rightarrow White(x)$
    - Different from "All white objects are squares":
      - $\forall x \in SHAPES,\ White(x) \rightarrow Square(x)$

32

Subsets and implication



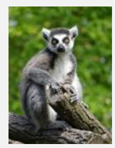
- When all elements of a set S_1 are also elements of a set S_2, we say that S_1 is a **subset** of S_2, written $S_1 \subseteq S_2$
  - Treat $x \in S$ as a predicate. Let U be the universe.
  - $S_1 \subseteq S_2$ iff $\forall x \in U \quad x \in S_1 \rightarrow x \in S_2$
  - $EVEN \subseteq \mathbb{Z}, \mathbb{N} \subseteq \{x \in \mathbb{Z} \mid x \geq 1\}$, DOGS $\subseteq$ MAMMALS
- $S_1 = S_2$ when $S_1 \subseteq S_2$ and $S_2 \subseteq S_1$
  - $\forall x \in U \quad x \in S_1 \leftrightarrow x \in S_2$
  - $\mathbb{N} = \{x \in \mathbb{Z} \mid x \geq 1\}$



33

Quantifiers and conditionals

- All squares are white.
  - $\forall x \in SHAPES\ \ Square(x) \rightarrow White(x)$. False!
- All white shapes are squares
  - $\forall x \in SHAPES\ \ White(x) \rightarrow Square(x)$ True!
- All circles are blue.
  - $\forall x \in SHAPES\ \ Circle(x) \rightarrow Blue(x)$.
- All blue shapes are circles.
  - $\forall x \in SHAPES\ \ Blue(x) \rightarrow Circle(x)$.
- All lemurs live in the trees.
  - $\forall x \in ANIMALS\ \ Lemur(x) \rightarrow LivesIn\ Trees(x)$
- All animals living in the trees are lemurs.
  - $\forall x \in ANIMALS\ \ LivesIn\ Trees(x) \rightarrow Lemur(x)$



34

Quantifiers and conditionals

- Why don't we write them as $\forall x \in SQUARES\ \ White(x)$?
  - Sometimes we can, but the first form is easier to reason about.
  - Besides, the predicate $White(x)$ is more useful when defined for any shapes, rather than only squares!
- More commonly, we use a shorthand called restricted quantifiers (or quantifiers with restricted domain):
  - $\forall x \geq 0\ \ Even(x)$ is a shorthand for $\forall x\ \ (x \geq 0 \rightarrow Even(x))$
    - Why do we write $\rightarrow$ rather than $\wedge$?
    - $\forall x\ \ x \geq 0 \wedge Even(x)$ is immediately false in $\mathbb{Z}$ since some integers are < 0.

35

- Then you should say what you mean,' the March Hare went on.
- `I do,' Alice hastily replied; `at least–at least I mean what I say–that's the same thing, you know.'
- `Not the same thing a bit!' said the Hatter. `You might just as well say that "I see what I eat" is the same thing as "I eat what I see"!'
- `You might just as well say,' added the March Hare, `that "I like what I get" is the same thing as "I get what I like"!'
- `You might just as well say,' added the Dormouse, who seemed to be talking in his sleep, `that "I breathe when I sleep" is the same thing as "I sleep when I breathe"!'

"Alice's Adventures in Wonderland"

by Lewis Carroll

36

37

Formulas: computational view

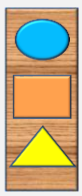
- As you are building a formula, think like a computer scientist:
  - What would a program (machine) that computes its value look like?
  - What are that program's inputs and outputs
  - and what are their types?
  - When would this program output an error?

- Think of operations in the formula computationally.

38

Type checking

- A formula consists of pieces of different types
  - Boolean: taking values true/false
    - Predicates, formulas
    - Can use operations $\neg, \vee, \wedge, \rightarrow$
  - Elements
    - Variables and fixed elements (constants) from the domain.
    - Occur as inputs to predicates and in quantifiers such as $\forall x \in S$
    - Can use operations (functions) from the domain:
      - If elements are numbers, can use $x + y, x \cdot y$, etc
  - Sets of elements
    - Used for domains: only occurs in quantifiers such as $\forall x \in S$



39

Type checking

- A formula consists of pieces of different types
  - Boolean: taking values true/false
  - Elements: e.g. numbers
  - Sets of elements

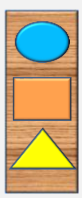
- $\forall x \in \mathbb{Z},\ \ Even(x) \vee Odd(x)$

40

Type checking

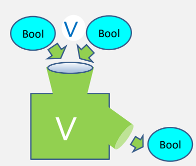
- A formula consists of pieces of different types
  - Boolean: taking values true/false
  - Elements: e.g. numbers
  - Sets of elements

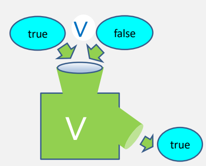
- $\forall x \in \mathbb{Z},\ \ Even(x) \vee Odd(x)$



41

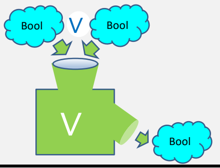
- Propositional variables and formulas are of type Boolean
  - that is, they take values in the set $\{true, false\}$
  - To evaluate a propositional variable, return its truth value.
- Now, treat connectives as machines with inputs and outputs.
  - They take Boolean inputs, and return Boolean outputs.





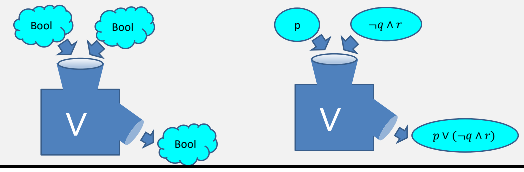
42

- To evaluate a propositional formula, go up its syntax tree:
  - If the formula (node in the tree) is just one variable, return its value
  - If a formula (node) is an operation, apply the respective "evaluation machine":
    - Machines for $\vee, \wedge, \rightarrow, \leftrightarrow$ want two Boolean inputs
    - Machine for $\neg$ wants one Boolean input
    - Each machine outputs a single Boolean value.



43

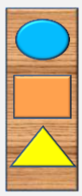
- Propositional variables and formulas are of type Boolean
  - that is, they take values in the set $\{true, false\}$
- If F, G are propositional formulas, then so are $\neg F, F \wedge G, F \vee G, F \rightarrow G$
  - So, operations $\vee, \wedge, \rightarrow, \leftrightarrow$ need to have Boolean type on both sides
    - And $\neg$ has to have a Boolean type following it.
  - The result of all of these operations is also of Boolean type.



44

Predicate logic formulas

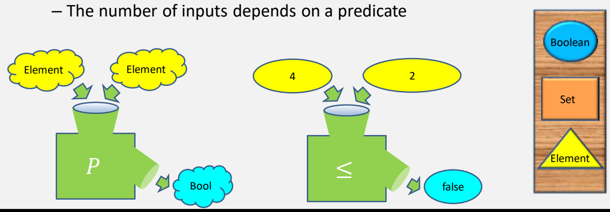
- A predicate logic formula consists of different types
  - Boolean: taking values true/false
  - Elements: e.g. numbers
  - Sets of elements
- We still can use $\vee, \wedge, \rightarrow, \leftrightarrow, \neg$ machines, both for constructing and for evaluating formulas
  - As formulas are of type Boolean
- Also need machines for predicates and quantifiers.



45

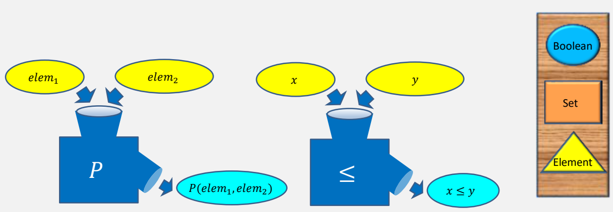
Computational view of predicates

- Predicate machines take elements as inputs and return Boolean
  - Different inputs may be elements from different sets
  - The number of inputs depends on a predicate



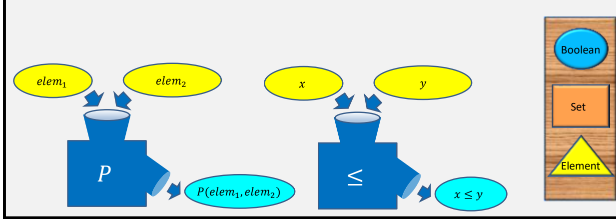
46

- Predicate machines take a list of variables or actual elements of their domain of type element as inputs and return a formula of the form
  - $Predicate(elem_1, elem_2, \ldots)$
    - If there are two inputs, the form such as $elem_1 \leq elem_2$ is also OK



47

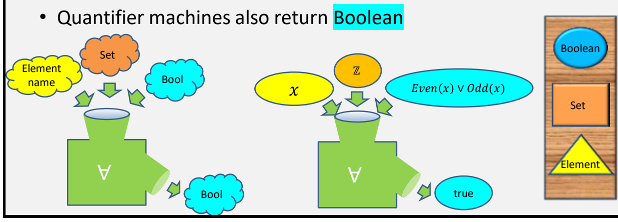
- To construct a predicate correctly, its inputs must take values from corresponding domains
  - Inputs to predicates can be functions, as long as output of the function is from the correct domain: so if the predicate P takes an integer as its input, and x takes integer values, then $P(x+1)$ is OK.



48

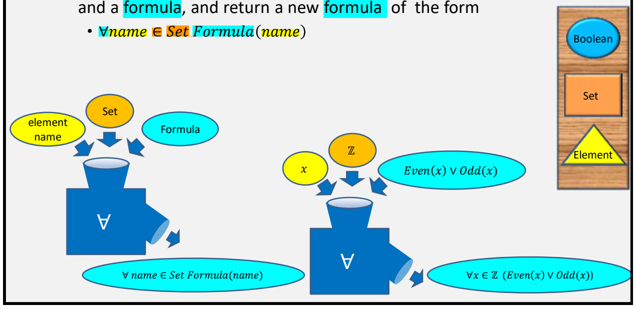
Computational view of quantifiers

- Inputs to quantifier machines are a set, a name of a variable of type element of that set, and a formula with that variable free.
- Quantifier machines also return Boolean



49

– Universal quantifier constructing machine takes a variable name, a set and a formula, and return a new formula of the form
  - $\forall name \in Set\ Formula(name)$



50

Liars paradox puzzle

- The first formulation of the famous liar's paradox, attributed to a Cretan philosopher Epimenides, stated

"All Cretans are liars".

Is this really a paradox?

51

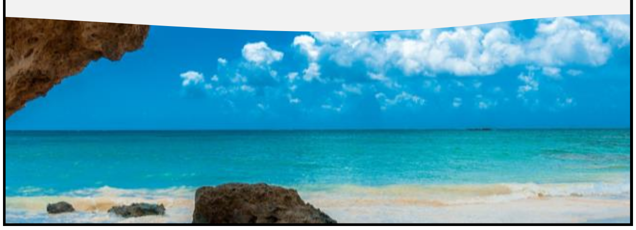
52

Liars paradox puzzle

- The first formulation of the famous liar's paradox, attributed to a Cretan philosopher Epimenides, stated

"All Cretans are liars".

Is this really a paradox?



53

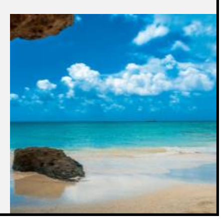
Liars paradox puzzle

- The first formulation of the famous liar's paradox, attributed to a Cretan philosopher Epimenides, stated

"All Cretans are liars".

Is this really a paradox?

– *If "all" is not true, that means that there is a counterexample: a Cretan that does not lie.*
– *So if Epimenides lied, what is true is that there are **some** truth-tellers on Crete (and potentially some liars, too)*
– *And Epimenides is one of the liars.*
– *However, "I am lying" would be a paradox.*



54

Counterexamples

- How to prove that a statement "$\forall x \in S,\ F(x)$" is false?
  - **All** girls hate math.
  - No!
    - **I** don't hate math ☺
  - **Every**body in O'Brian family is tall
    - No, **Jenny** is O'Brian and she is quite short.
  - It is foggy **all** the time, **every** day in St. John's
    - No, **sometimes** it is not foggy (like **today**).

Counterexample:

Element of S for which the formula is false.

One is enough, though more than one is OK.

55

Negation of a universal statement

- What is the negation of "All", that is, what statement is true whenever "$\forall x \in S,\ F(x)$" is false?
  - **All** girls hate math.
  - No!
    - All girls love math?
    - **Some** girls do not hate math! **I** don't.
  - Everybody in O'Brian family is tall
    - No, there is **someone** (**Jenny**) who is O'Brian and short.
  - It is foggy **all** the time, every day in St. John's
    - No, **sometimes** it is not foggy (like **today**).

"Not **All** " ≡ "**Some** not"

As there is a quantifier ∀ for "all", there is also a quantifier for "some": ∃, called **existential quantifier.**

56

Existential quantifier

- $\exists x \in S\ F(x)$

(pronounced "exists")

is true when there is some value a for x in S for which $F(x)$ is true.
  - This $a \in S$ is called a witness.

$\neg(\forall x \in S\ F(x)) \equiv \exists x \in S\ \neg F(x)$

$\forall x \in S\ F(x)$ is false iff there exists (∃) a counterexample: an element $a \in S$ such that $\neg F(a)$ is true.

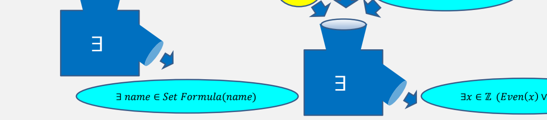
57

Examples with existential quantifier

- "There exists an even prime number."
  - Domain $\mathbb{Z}$
  - Predicates $Even(x)$ and $Prime(x)$
  - $\exists x \in \mathbb{Z},\ \ Even(x) \wedge Prime(x)$
    - This is true: 2 is both even and prime.
- "There is some number between 0 and 1"
  - Domain $\mathbb{R}$, predicate $<$
  - $\exists x \in \mathbb{R},\ \ (0 < x) \wedge (x < 1)$
    - True when the domain is $\mathbb{R}$
      - There is not just one, but lots of real numbers between 0 and 1
    - Would be false if we change the domain to $\mathbb{Z}$
- "n is divisible by m if there is an integer which when multiplied by m gives n"
  - Domain $\mathbb{Z}$, Predicate $=$
  - $\exists z \in \mathbb{Z},\ \ n = mz$
    - True for n= $6, m = 3$; in this case, z = 2
    - False when eg $n = 3,\ m = 2$
  - Here, n and m are still *free* variables, whereas z is *bound* by the quantifier.

58

- Quantifier constructing machines take a variable name, a set and a formula, and return a new formula of the form
  - $\forall name \in Set\ Formula(name)$
  - $\exists name \in Set\ Formula(name)$



59

Truth of existential statements

- $\exists x \in S\ F(x)$ is true when $F(a)$ is true for some $a \in S$
  - "I am teaching some course" is true because I am teaching COMP 1002
  - Here, COMP 1002 is the witness for the truth of "I am teaching some course"
- That is, when $F(a_1) \vee F(a_2) \vee \cdots \vee F(a_n) \vee \cdots$ is true, where $a_1, a_2, \ldots$ is a list of all elements of S
  - Enough to find a single witness $a \in S$ with $F(a)$ true to prove $\exists x \in S\ F(x)$.
  - To show that $\exists x \in S\ F(x)$ is false, show that for all $a \in S$, $F(a)$ is false
    - That is because $\neg(\exists x \in S\ F(x)) \equiv \forall x \in S\ \neg F(x)$

60

Quantifiers in English

- Universal quantifier: usually "every", "all", "each", "any".
  - Every day it is foggy. Each number is divisible by 1.
- "None", "no", "nobody", "nothing" also translate as universal quantifier
  - Nobody works on Sundays ≡ Everybody does not work on Sundays

- Existential quantifier: "some", "a", "exists"
  - Some students got 100% on the first lab.
  - There exists a prime number greater than 100.
- The word "any" can mean either!

61

Quantifiers in English: "any"

- "Any" can have different meanings depending on the context:
- Any = some 
  - Can I have **any** (piece of the) pie?
  - Can I have **some** (piece of the) pie?

- Any = all 
  - **Any** student knows this.
  - **Every** student knows this.

- **Any** student can get an A
- Would **any** student get an A?

62

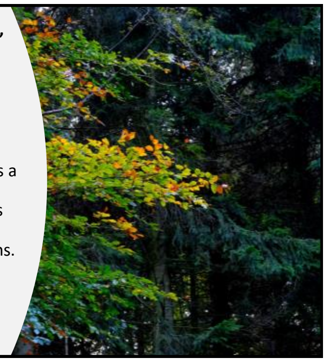
62

63

When negating a quantified formula, the domain always stays the same.

- **Quantifiers flip** ($\forall$ **becomes** $\exists$, $\exists$ **becomes** $\forall$)
- The formula under quantifiers gets a $\neg$ in front, and can be simplified further, treating logical connectives the same way as in propositional logic, and predicates as propositions.

$$\neg(\forall x \in S\ F(x)) \equiv \exists x \in S\ \neg F(x)$$
$$\neg(\exists x \in S\ F(x)) \equiv \forall x \in S\ \neg F(x)$$



64

When negating a quantified formula, the domain stays the same.
Quantifiers flip ($\forall$ becomes $\exists$, $\exists$ becomes $\forall$)
The formula under quantifiers gets a $\neg$ in front, and can be simplified further

Not all **integers** are even ≡ Some **integers** are not even
$\neg(\forall x \in \mathbb{Z}\ Even(x)) \equiv \exists x \in \mathbb{Z}\ \neg Even(x)$

- Some integers are both even and prime – no, none of the integers is both even and prime each integer is not even or not prime
$\neg(\exists x \in \mathbb{Z},\ Even(x) \land Prime\ (x)) \equiv \forall x \in \mathbb{Z}, \neg Even(x) \lor \neg Prime(x)$

- As in propositional case, push negations inside the formula, onto predicates.
  - "NOT (ALL trees have leaves)" is more confusing than "some trees do not have leaves" or "some trees have something other than leaves (e.g., needles).

65

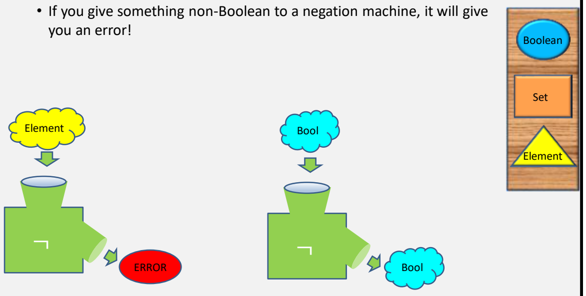
"NOT" is for predicates, not domains!

- For any formula F we can write $\neg F$ to denote a formula which is true if and only if F is false, just like in propositional logic
- F = "All trees have leaves"
  - Is $\neg F$ "non-trees have non-leaves"?...
  - No! What is a non-tree and a non-leaf, anyway?

  $\neg F$ is "Not all trees have leaves"
  - which is the same as "Some trees do not have leaves"
  - we are still talking about trees!
- When simplifying a negated formula, we flip quantifiers and negate subformulas as we would in propositional logic
  - But we never change the domain!

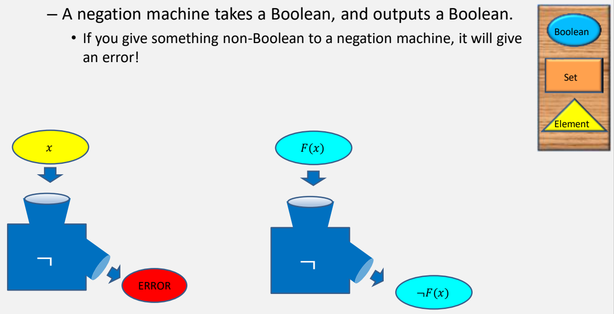
66

– A negation machine takes a Boolean, and outputs a Boolean.
  - If you give something non-Boolean to a negation machine, it will give you an error!



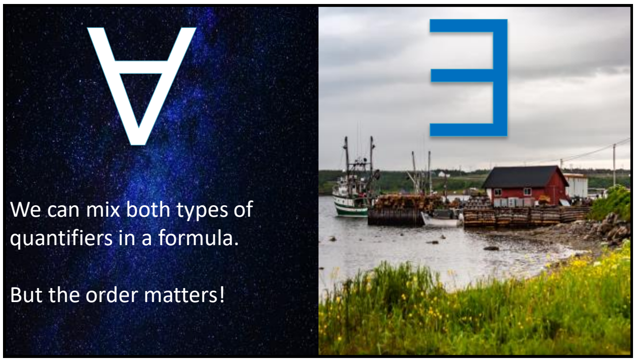
67

– A negation machine takes a Boolean, and outputs a Boolean.
  - If you give something non-Boolean to a negation machine, it will give an error!



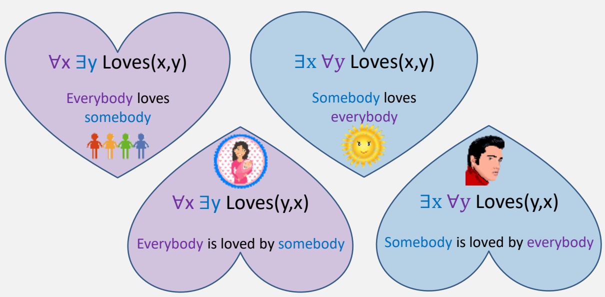
68

69



70

- Predicate: $Loves(x, y)$, true when x loves y. Domain: people.



71

Nested quantifiers

- Order of variables under different quantifiers, the order of quantifiers themselves, matters.
  – Everybody loves somebody is true for people in general
  – Whereas somebody loves everybody is rare (only the sun shines on everyone)
- Order in a predicate matters.
  – To love is is not the same as to be loved.
  – Everyone is loved by their mother
  – yet it is rare to be loved by everyone (except maybe Elvis)

72

Nested quantifiers

- Names of variables don't matter
  - Everybody loves somebody: 

    $\forall x\ \exists y$ Loves(x,y) same as $\forall y\ \exists x$ Loves(y,x) same as $\forall p_1\ \exists p_2\ Loves(p_1, p_2)$
  - Somebody is loved by everybody: 

    $\exists x\ \forall y$ Loves(y,x) same as $\exists y\ \forall x$ Loves(x,y) same as $\exists p_1 \forall p_2 Loves(p_2, p_1)$

73

Nested quantifiers

- When quantifiers of the same time are next to each other, their order does not matter.

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

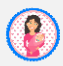
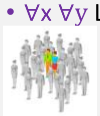
Everybody loves everybody

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

Somebody loves somebody

- First says that every two people love each other: eg when the domain is a happy family
- Second says that somewhere there is a person who feels love (to someone else, or maybe to themselves)
- However, in $\exists x\ \forall y\ \exists z\ F(x, y, z)$ you cannot exchange x with z: 
  - z might be different for each different y, but the same x should work for all of them!
  - Can ignore order only for quantifiers of the same type right next to each other 

74

- $\forall x\ \exists y$ Loves(x,y) Everybody loves somebody 
  -  $\exists x\ \forall y$ Loves(x,y) Somebody loves everybody
- $\forall x\ \exists y$ Loves(y,x) Everybody is loved by somebody 
  -  $\exists x\ \forall y$ Loves(y,x) Somebody is loved by everybody
- $\forall x\ \forall y$ Loves(x,y) Everybody loves everybody 
  -  $\exists x\ \exists y$ Loves(x,y) Somebody loves somebody

75

76

Evaluating sentences with nested quantifiers

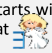
- Take a formula with no free variables: a sentence
  - How do we find out if it is true or false? Play a game!
- Two players, taking turns in order of quantifiers
  - The red player holding $\forall$ suggests counterexamples
  - The green player holding $\exists$ suggests witnesses.
  - If the red player has a way to win no matter what the green player does, then the formula is false
  - If the green player has a way to win no matter what the red player does, then the formula is true.
    - The sentence is always either true or false, so one of them can always win.

I want it false! 

 I want it true!

77

$\forall x\ \in \mathbb{Z}\ \exists y \in \mathbb{Z}\ \forall z \in \mathbb{Z}\ \ x \neq yz$

- Let's first read the formula. It says that no matter what integer x you pick, there is some integer y that does not divide x (does not give x no matter what z you multiply it by).
  - Do you think it is true? Let's play the game!
  - The formula starts with $\forall$, so  goes first. After that  chooses y, then it is  again

78

$\forall x \in \mathbb{Z}\, \exists y \in \mathbb{Z}\, \forall z \in \mathbb{Z}\;\; x \neq yz$

- Let's first read the formula. It says that no matter what integer x you pick, there is some integer y that does not divide x (does not give x no matter what z you multiply it by).
  - Do you think it is true? Let's play the game!
  - The formula starts with ∀, so goes first. After that chooses y, then it is again

$x = 0$ Beat that!

79

$\forall x \in \mathbb{Z}\, \exists y \in \mathbb{Z}\, \forall z \in \mathbb{Z}\;\; x \neq yz$

- Let's first read the formula. It says that no matter what integer x you pick, there is some integer y that does not divide x (does not give x no matter what z you multiply it by).
  - Do you think it is true? Let's play the game!
  - The formula starts with ∀, so goes first. After that chooses y, then it is again

$x = 0$ Beat that!

uhm, $y = 1$?

80

$\forall x \in \mathbb{Z}\, \exists y \in \mathbb{Z}\, \forall z \in \mathbb{Z}\;\; x \neq yz$

- Let's first read the formula. It says that no matter what integer x you pick, there is some integer y that does not divide x (does not give x no matter what z you multiply it by).
  - Do you think it is true? Let's play the game!
  - The formula starts with ∀, so goes first. After that chooses y, then it is again

$x = 0$ Beat that!

uhm, $y = 1$?

Nope! Set $z = 0$

and now $0 = 1 \cdot 0$

The formula is false!

Mwahahaha!

81

$\forall x \in \{1,2,3\}\, \exists y \in \mathbb{Z}\, \forall z \in \mathbb{Z}\;\; x \neq yz$

- Now we changed the domain of x, so x can only take values 1, 2 or 3.
  - Do you think it is true now? Let's play the game and find out!
  - So the formula is true.
    - No matter what x among allowed 1,2,3 he takes, she can reply with $y = x + 1$ and win.
    - Note that her choice of y depends on what value of x he picks.
    - But since he has to choose x before she chooses y, it is OK, as she would know his choice for the value of x. The order of quantifiers matters!

Say $x = 2$

Then let $y = 3$?

Say $z = 0?..$

But $2 \neq 0 \cdot 3$

I give up. No matter what z I try, $2 \neq z \cdot 3$

You win ☹ Arrrgh!

82

$\forall x \in \mathbb{Z}\, (x = 0 \vee \exists y \in \mathbb{Z}\, \forall z \in \mathbb{Z}\;\; x \neq yz)$

- Now let's try a more complex formula, which says that for every integer x either $x = 0$ or there is an integer y that does not divide x
  - After the first step where sets x, the formula under $\forall x$ splits into ∨ of two.
  - The first, $x = 0$, can be evaluated right away
    - He'd lose immediately if he sets x to be 0
    - So he'd better choose a non-zero value for x
  - Still, no matter what x he takes, she can always reply with $y = x + 1$ (if $x < 0$, then $y = x - 1$) and win

Say $x = 5$

Then let $y = 6$?

Say $z = 0?..$

But $5 \neq 0 \cdot 6$

I give up. No matter what z I try, $5 \neq z \cdot 6$

You win ☹ Arrrgh!

83

$\forall x \in \mathbb{Z}\, \forall y \in Z\;\; \exists z \in \mathbb{Z}\;\; xz = yz)$

- If there are several quantifiers of the same type, the corresponding player keeps getting a turn
  - Here, the formula starts with two ∀
  - So gets to go twice, setting both x and y.
  - Only then gets her turn to set z.

Say $x = 2$

Say $y = 3$

Take $z = 0$

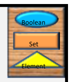
It works ☺

84

85

86

Variables and terms



- Simplest formulas are predicates: $P(x, y),\ x_1 = x_2,\ Even(num)$
  - Here, x, y, x_1, x_2, num are variables, that is, placeholders for elements in the domains of the predicates.
  - Instead of a variable, can have an expression (also called a *function* or a *term*), which takes values of variables and returns a value in the domain:
    - $x + 1,\ x \cdot y$ are terms
    - Can now use them in $Even(x + 1)$, or in $x \cdot y = z$
    - E.g. if the domain of Even( ) and = is $\mathbb{Z}$, then x, y, z take values in $\mathbb{Z}$, so the value of $x + 1 \in \mathbb{Z}$, and the value of $x \cdot y \in \mathbb{Z}$ no matter what values x, y, z take.
  - If the formula is a predicate, then all the variables it takes are free.
    - Including all variables mentioned in the terms, as x, y, z in $x \cdot y = z$

87

Naming conventions

- **Names of variables do not matter**, as long as you use the same name every time you refer to a specific placeholder in a formula.
  $\forall x \in \mathbb{Z}\ (Even(x) \lor Odd(x))$ is the same as $\forall y \in \mathbb{Z}\ (Even(y) \lor Odd(y))$
- The convention is to use either lowercase letters, usually at the end of the alphabet (x, y, z, w, x_1, x_2) or lowercase words (num for number) for variables
  - Just like unknowns in arithmetic equations
  - Letters a, b, c more often used for specific elements.
- Use capitalized words ($Even, LessThan$) or P, Q for predicates.
  - Often use P, Q when the interpretation of the predicate is not specified

88

Prenex normal form

- Better to avoid using same names for different variables – it is confusing.
  - $\forall x\,(\exists y\ P(x, y)) \land (\exists y\, Q(x, y)) \equiv \forall x\,(\exists\, y\, P(x, y)) \land (\exists z\, Q(x, z))$
    $\equiv \forall x\, \exists y\, \exists z\, P(x, y) \land Q(x, z)$
- The final line is an example of a special form of a predicate logic formula with all quantifiers in front followed by a formula without quantifiers, called *prenex normal form.*
  - Here, the names must be different to differentiate variables, as they all have the same scope: the whole formula.

89

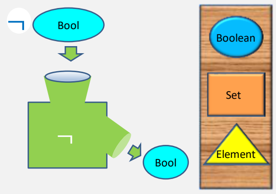
Constructing predicate logic formulas

- A single predicate is a predicate logic formula
  - $P(x, y),\ x_1 = x_2,\ Even(num)$
  - In these formulas, x, y, x_1, x_2, num are free variables
- If F and G are predicate logic formulas, then so are
  $F \land G \qquad F \lor G \qquad F \to G \qquad F \leftrightarrow G \qquad \neg F$
- If $F(x)$ is a predicate logic formula with a free variable x from the domain S, then so are $\forall x \in S\ F(x)$ as well as $\exists x \in S\ F(x)$
  - Here x is no longer a free variable, it gets bound by the quantifier.
  - When the domain is clear, just write $\forall x\ F(x)$ (respectively, $\exists x\ F(x)$)
  - Use parentheses to avoid ambiguity: $\forall x\,(G(x) \to H)$ is not equivalent to $(\forall x\, G(x)) \to H$

90

Make sure to type-check!

- Always think which types can be combined with which operations,
  - and what the result looks like!
- Remember that if you give an evaluation machine incorrectly formatted input, it will give you an error.
  - "$Even(x), Odd(x)$" ERROR!
  - "$\exists$ $Even$ " ERROR!
  - "$Even(x = y)$" ERROR!
  - "$\forall x \in \mathbb{Z}\ \neg x$" $ERROR$!



91