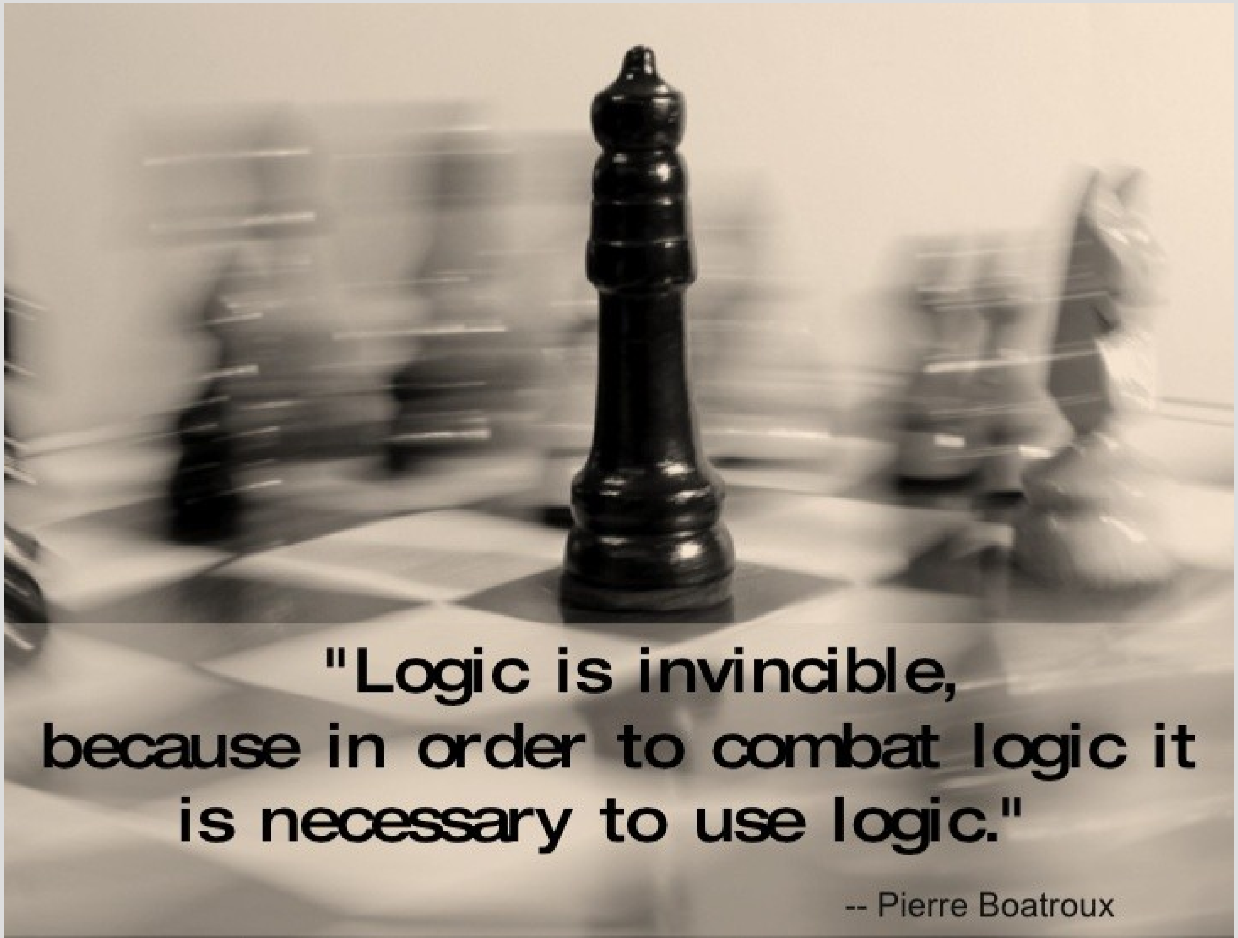
First order logic (Ch. 8)



Review: Propositional logic

Propositional logic builds sentences that relate various symbols with true or false

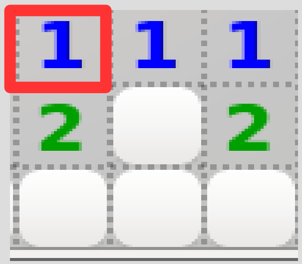
Each symbol is simply a unique identifier, but you cannot "generalize" between them

While this is fairly expressive, it is also quite cumbersome as each part of the environment might need many symbols associated with it

Review: Propositional logic

For example: to express just the top left cell of this minesweep, we would need to have:

$$P1,1,1 \land \neg P1,1,2 \land \neg P1,1,3$$
$$\land \neg P1,1,4 \land \neg P1,1,5 \land \neg P1,1,6$$
$$\land \neg P1,1,7 \land \neg P1,1,8 \land \neg P1,1,B$$



Sadly in propositional logic we cannot relate these 9 symbols/literals together as "value of cell [1,1]" (and cannot specify this relationship in general for all cells)

FO logic: definitions

Propositional logic has “propositions” that are either true or false

First order logic (also called “predicate calculus”) has objects and the relation between them is what is important

This can provide a more compact way of expressing the environment (also more complicated since we cannot build truth tables)

FO logic: definitions

There are two basic things in first order logic:

Also called constant symbols

1. Objects which are some sort of noun or "thing" in the environment (e.g. teacher, bat)

2. Relations among objects, which can be:
   2.1. Unary (or properties) which relate to a single object (e.g. red, healthy, boring)
   2.2. n-ary which involve more than one
   2.3. Functions, one "value" for each input

FO logic: definitions

Both unary and n-ary relations are similar, just how many variables are involved

Unary and n-ary relations are predicates

Unary and n-ary relations are true/false values (similar to propositional logic)

$$President(America, Biden) \wedge President(Mexico, AMLO)$$

Functions converts the inputted objects into a single output object (i.e. like coding functions that "return" a ***single*** object)

$$PresidentOf(America) = Biden, PresidentOf(Mexico) = AMLO$$

FO logic: definitions

We can represent any sentence with objects and relations, for example:

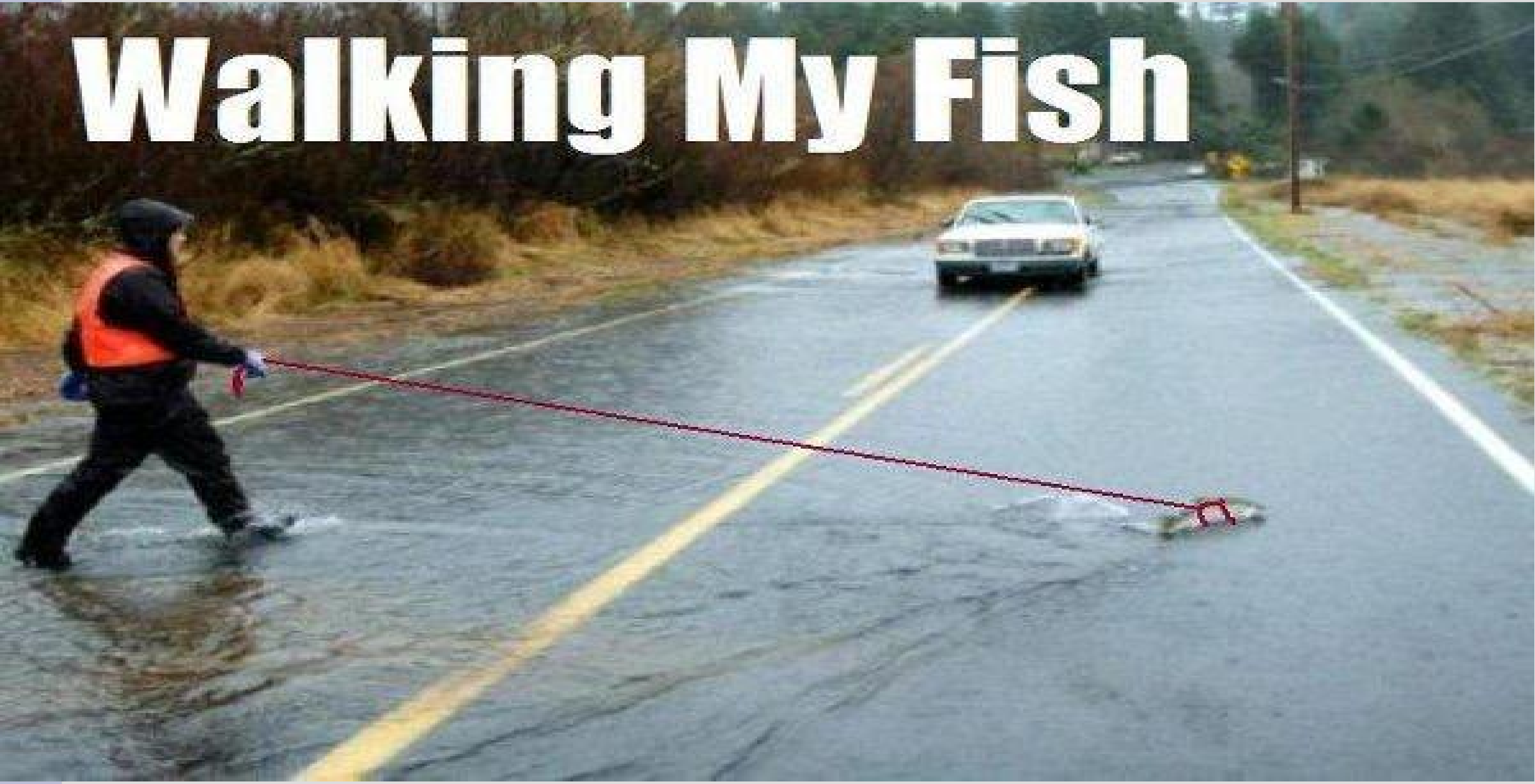
"I am sleepy today"
Object: I, (the "me" of today)
Relations: Sleepy, Today
Logic: Sleepy(Today(I))

The set of all objects that we are using are called the <u>domain</u>

"I howl at full moons"
Objects: Me, Moon Relations: Full, Howl
Logic: Full(Moon) => Howl(Me)

FO logic: definitions

Let's identify objects and relations in this:



FO logic: definitions

Objects:
Person, Car, Road, Fish, Leash

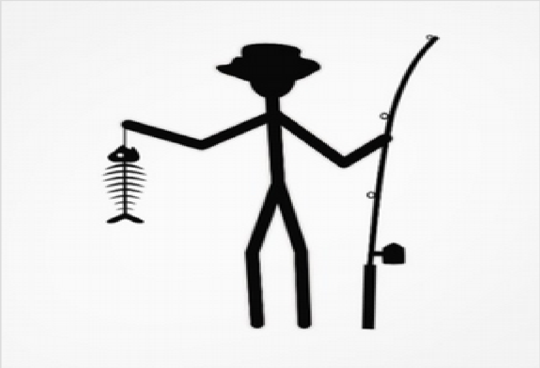
Relations:
Unary: Wet(Fish), Wet(Road), Wet(Car)
n-ary: OnTopOf(Person, Road),
OnTopOf(Car, Road), OnTopOf(Fish, Road)
Functions: attached(Person, Leash) = Fish

FO logic: definitions

You find objects and relations (what type):

FO logic: definitions

Objects:
StickPerson, Fish, Pole, Hat, SP'sLeftLeg



Relations examples....
unary: Black(StickPerson)
n-ary: Hold(StickPerson, Fish),
    Hold(StickPerson, Pole)
functions:
    OnHead(StickPerson), LeftLeg(StickPerson)

FO logic: definitions

The “arguments” to relations are assumed to be order dependent (not symmetrical)

For example: Hold(StickPerson, Fish) might imply “StickPerson holds Fish”

This is not a symmetric relationship, so Hold(Fish, StickPerson) conveys a different meaning

FO logic: definitions

Can represent relations as “tuples”
(generalize “pair” for more than 2 elements)

For example the “Hold” relation might be:
{<StickPerson, Fish>, <StickPerson, Pole>}

For functions, we normally provide the result:
OnHead:
<StickPerson> → Hat
<Fish> → String

FO logic: definitions

Side note:
Functions have to be defined for all possible objects in our use of first-order logic

So with the "OnHead" function in the last example, we would also need to define:

OnHead(Pole) = Pole
OnHead(Hat) = Hat
OnHead(SP'sLeftLeg) = StickPerson (?)

Syntax

Objects and relations form the basis of first order logic, but we also expand our syntax with three things:

1. Quantifiers (existential and universal)
2. Variables (much in the math sense)
3. Equality (as in “=” not “ $\Longleftrightarrow$ ” or “$\equiv$”)

Otherwise we have a similar syntax to propositional logic (implies, AND, OR, etc.)

Existential quantifier

The existential quantifier is $\exists$, which means "there exists ..."

For example, if I had a variable "x", then...

$$\exists x \; Santa(x)$$

... means "Santa exists" or "Someone is Santa"

if quantifier on far left without parenthesis, assume applies to whole sentence

$$\exists x \; Person(x) \wedge InClass(x) \wedge Hungry(x)$$

... means "Someone in class is hungry" or "At least one person in class is hungry"

Existential quantifier

The existential quantifier is $\exists$, which means "there exists ..."

For example, if I had a variable "x", then...

$$\exists x \; Santa(x)$$

... means "Santa exists" or "Someone is Santa"

$$\exists x \; Person(x) \wedge InClass(x) \wedge Hungry(x)$$

... means "Someone in class is hungry" or "At least one person in class is hungry"

Variables

A variable is a place-holder for any object

So if we had 3 objects, {Sue, Alex, Devin}, we could formally write:

$$\exists x \ Santa(x)$$

As...

$$Santa(Sue) \lor Santa(Alex) \lor Santa(Devin)$$

... or in English: “Someone is Santa”, “Santa is Sue, Alex or Devin”

Universal quantifier

The universal quantifier is denoted by $\forall$ means "for all ..."

Thus, $\forall x\ Santa(x)$ ... means "Everyone is a Santa"

If our objects were again {Sue, Alex, Devin}, then this would mean:

$$Santa(Sue) \wedge Santa(Alex) \wedge Santa(Devin)$$

Quantifier

As $\exists$ is basically ORs and $\forall$ is ANDs, we can apply De Morgan's laws:

$$\neg(\exists x\ Santa(x)) \equiv \forall x\ \neg Santa(x)$$

In words “No Santa exists” is the same as “Everyone is not Santa” (or “No one is Santa”)

You can have multiple quantifiers as well:

$$\exists x \exists y\ SnapChat(x, y) \equiv \exists x, y\ SnapChat(x, y)$$

This means “Two people are snapchatting”
(Note: this could also mean snapchatting self)

Quantifier

The order of quantifiers also matters:

$\forall x \exists y \ Mother(x) = y$ means "For every person x, they have some mother y" or "All people have some mother"

However in the opposite order:

$\exists y \forall x \ Mother(x) = y$ means "There is some person y, who is the mother to everyone" or "Everyone has the same mother"

Quantifier

Write these two sentences in logic:

1. “Someone is happy yet sleepy”

2. “Everyone in class is thinking”

Quantifier

Write these these two sentences in logic:

1. "Someone is happy yet sleepy"

$$\exists x \; Person(x) \wedge Happy(x) \wedge Sleepy(x)$$

2. "Everyone in class is thinking"

$$\forall x \; \Big(Person(x) \wedge InClass(x) \Big) \Rightarrow Thinking(x)$$

Normally this is the case:

For "$\exists$" you use $\wedge$

For "$\forall$" you use $\Rightarrow$

Equality

In logic, equality means two things are the same (much as it does in math)

For example, $Sue = Alex$ would imply Sue and Alex are the same people

This is often useful with variables:

$$\forall x, y \; \neg(x = y) \Rightarrow \neg(Midterm(x) = Midterm(y))$$

... which means "No two (different) people have the same midterm score" (unique scores)

Assumptions

Being completely expressive in first order logic can be difficult at times

In the last statement you need the "$\neg(x = y)$" (which I will abbreviate often as: "$x \neq y$") to ensure that the variable does not reference the same person/object

However, in general two objects could be the same thing...

Assumptions

Try to formally express:

“My only brothers are Bob and Jack”

Assumptions

Try to formally express:
"My only brothers are Bob and Jack"

$$
\begin{aligned}
&Brother(James, Bob)\\
\wedge &Brother(James, Jack)\\
\wedge &Jack \neq Bob\\
\wedge &\forall x\ Brother(James, x) \Rightarrow (x = Bob \vee x = Jack)
\end{aligned}
$$

This is overly complicated as we have to specify that everyone else is not my brother and that Jack and Bob are different people

Assumptions

For this reason, we make 3 assumptions:

1. Objects are unique (i.e. $Bob \neq Jack$ always)

2. Only objects I have specified exist
    (i.e. I assume a person $Davis$ does not exists as I never mentioned them)

3. (sometimes) All un-said sentences are false
    Thus, if I only say: $Brother(James, Bob)$
    then I imply: $\neg Brother(James, Jack)$

Assumptions

These assumptions make it easier to write some sentences more compactly

Under assumptions 1. and 3., “My sisters are Alice and Grace” can be represented as:

$$Sister(James, Alice) \wedge Sister(James, Grace)$$

Assumption 3. does make it harder to say more general sentences, such as: “Two of my sisters are Alice and Grace”