A Short Introduction to Propositional Logic and First-Order Logic

Xiaojin Zhu

jerryzhu@cs.wisc.edu

Computer Sciences Department

University of Wisconsin, Madison

[Based on slides from Louis Oliphant and Andrew Moore]

Logic

- If the rules of the world are presented formally, then a decision maker can use logical reasoning to make rational decisions.

- Several types of logic:
  - propositional logic (Boolean logic)
  - first order logic (first order predicate calculus)

- A logic includes:
  - syntax: what is a correctly formed sentence
  - semantics: what is the meaning of a sentence
  - Inference procedure (reasoning, entailment): what sentence logically follows given knowledge

Propositional logic syntax

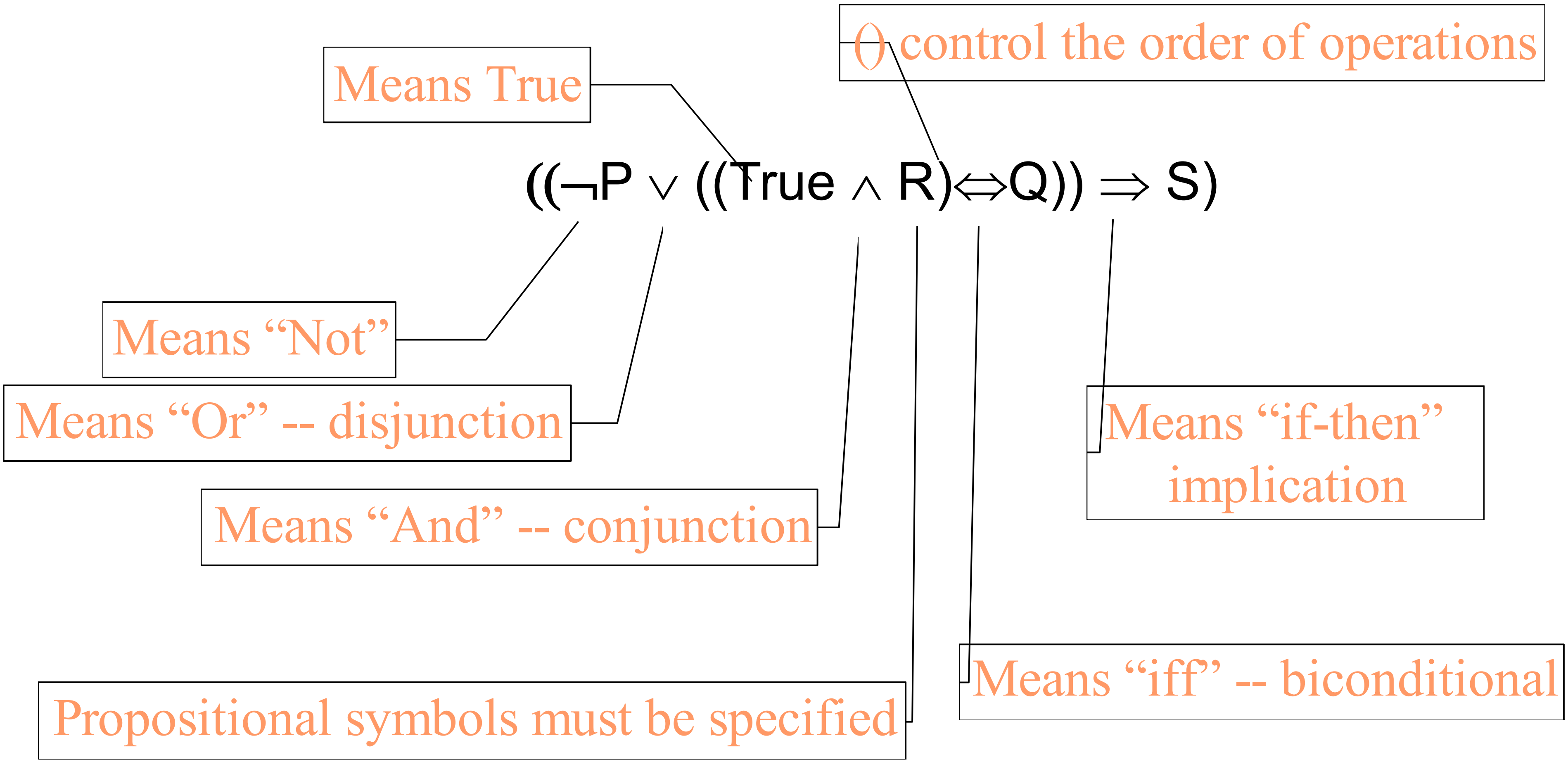
$$
\begin{array}{lll}
\textit{Sentence} & \to & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \to & \text{True} \mid \text{False} \mid \textit{Symbol} \\
\textit{Symbol} & \to & \text{P} \mid \text{Q} \mid \text{R} \mid \ldots \\
\textit{ComplexSentence} & \to & \neg \textit{Sentence} \\
\mid & & (\ \textit{Sentence} \land \textit{Sentence}\ ) \\
\mid & & (\ \textit{Sentence} \lor \textit{Sentence}\ ) \\
\mid & & (\ \textit{Sentence} \Rightarrow \textit{Sentence}\ ) \\
\mid & & (\ \textit{Sentence} \Leftrightarrow \textit{Sentence}\ )
\end{array}
$$

BNF (Backus-Naur Form) grammar in propositional logic

$((\neg P \lor ((\text{True} \land R) \Leftrightarrow Q)) \Rightarrow S$ well formed

$(\neg(P \lor Q) \land \Rightarrow S)$ not well formed

Propositional logic syntax



Propositional logic syntax

- Precedence (from highest to lowest):

$$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$$

- If the order is clear, you can leave off parenthesis.

$\neg P \vee True \wedge R \Leftrightarrow Q \Rightarrow S$ ok

$P \Rightarrow Q \Rightarrow S$ not ok

Semantics

- An interpretation is a complete True / False assignment to propositional symbols
  - Example symbols: P means “It is hot”, Q means “It is humid”, R means “It is raining”
  - There are 8 interpretations (TTT, ..., FFF)
- The semantics (meaning) of a sentence is the set of interpretations in which the sentence evaluates to True.
- Example: the semantics of the sentence $P \vee Q$ is the set of 6 interpretations
  - P=True, Q=True, R=True or False
  - P=True, Q=False, R=True or False
  - P=False, Q=True, R=True or False
- A model of a set of sentences is an interpretation in which all the sentences are true.

Evaluating a sentence under an interpretation

- Calculated using the meaning of connectives, recursively.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

- Pay attention to ⇒
  - "5 is even implies 6 is odd" is True!
  - If P is False, regardless of Q, P⇒Q is True
  - No causality needed: "5 is odd implies the Sun is a star" is True.

Semantics example

$$\neg P \lor Q \land R \Rightarrow Q$$

Semantics example

$$\neg P \vee Q \wedge R \Rightarrow Q$$

P	Q	R	~P	Q^R	~PvQ^R	~PvQ^R->Q
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	1	0	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	0	1
1	1	0	0	0	0	1
1	1	1	0	1	1	1

Satisfiable: the sentence is true under some interpretations

Deciding satisfiability of a sentence is NP-complete

Semantics example

$$(P \land R \Rightarrow Q) \land P \land R \land \neg Q$$

Semantics example

$$(P \wedge R \Rightarrow Q) \wedge P \wedge R \wedge \neg Q$$

P	Q	R	~Q	R^~Q	P^R^~Q	P^R	P^R->Q	final
0	0	0	1	0	0	0	1	0
0	0	1	1	1	0	0	1	0
0	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0
1	0	0	1	0	0	0	1	0
1	0	1	1	1	1	1	0	0
1	1	0	0	0	0	0	1	0
1	1	1	0	0	0	1	1	0

Unsatisfiable: the sentence is false under all interpretations.

Semantics example

$$(P \Rightarrow Q) \vee P \wedge \neg Q$$

Semantics example

$$(P \Rightarrow Q) \vee P \wedge \neg Q$$

P	Q	R	~Q	P->Q	P^~Q	(P->Q)vP^~Q
0	0	0	1	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

Tautology: the sentence is true under all interpretations

Knowledge base

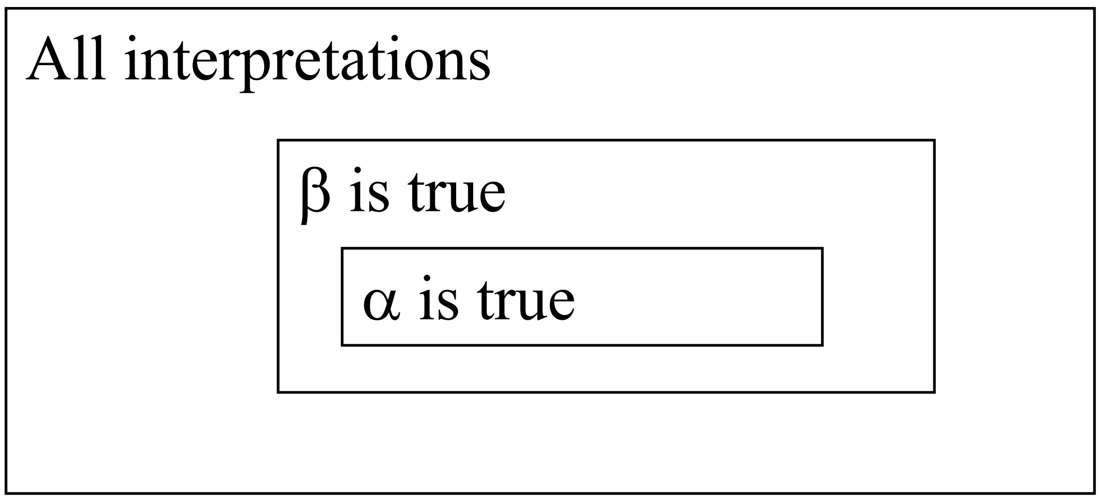
- A knowledge base KB is a set of sentences. Example KB:
  - $TomGivingLecture \Leftrightarrow (TodayIsTuesday \vee TodayIsThursday)$
  - $\neg$ TomGivingLecture
- It is equivalent to a single long sentence: the conjunction of all sentences
  - $(TomGivingLecture \Leftrightarrow (TodayIsTuesday \vee TodayIsThursday)) \wedge \neg TomGivingLecture$
- The model of a KB is the interpretations in which all sentences in the KB are true.

Entailment

- Entailment is the relation of a sentence β logically follows from other sentences α (i.e. the KB).

$$\alpha \models \beta$$

- $\alpha \models \beta$ if and only if, in every interpretation in which α is true, β is also true



Logical equivalences

$$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge$$
$$(\alpha \vee \beta) \equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee$$
$$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge$$
$$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee$$
$$\neg(\neg\alpha) \equiv \alpha \quad \text{double-negation elimination}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{implication elimination}$$
$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination}$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{de Morgan}$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{de Morgan}$$
$$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee$$
$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge$$

You can use these equivalences to modify sentences.

Resolution

- Resolution: a single inference rule
  - Sound: only derives entailed sentences
  - Complete: can derive any entailed sentence
    - Resolution is only refutation complete: if KB $\models \beta$, then KB $\wedge \neg \beta$ |- *empty*. It cannot derive *empty* |- $(P \vee \neg P)$
  - But the sentences need to be preprocessed into a special form
  - But all sentences can be converted into this form

Conjunctive Normal Form (CNF)

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

- Replace all $\Leftrightarrow$ using biconditional elimination
- Replace all $\Rightarrow$ using implication elimination
- Move all negations inward using
  -double-negation elimination
  -de Morgan's rule
- Apply distributivity of $\vee$ over $\wedge$

Convert example sentence into CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ starting sentence

$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

biconditional elimination

$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

implication elimination

$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

move negations inward

$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

distribute $\vee$ over $\wedge$

Resolution steps

- Given KB and β (query)
- Add $\neg \beta$ to KB, show this leads to empty (False. Proof by contradiction)
- Everything needs to be in CNF
- Example KB:
  - $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - $\neg B_{1,1}$
- Example query: $\neg P_{1,2}$

Resolution preprocessing

- Add $\neg\beta$ to KB, convert to CNF:

  a1: $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1})$

  a2: $(\neg P_{1,2} \vee B_{1,1})$

  a3: $(\neg P_{2,1} \vee B_{1,1})$

  b: $\neg B_{1,1}$

  c: $P_{1,2}$
- Want to reach goal: *empty*

Resolution

- Take any two clauses where one contains some symbol, and the other contains its complement (negative)

$$P \vee Q \vee R \qquad \neg Q \vee S \vee T$$

- Merge (resolve) them, throw away the symbol and its complement

$$P \vee R \vee S \vee T$$

- If two clauses resolve and there's no symbol left, you have reached *empty* (False). KB |= β
- If no new clauses can be added, KB does not entail β

Resolution example

a1: $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1})$

a2: $(\neg P_{1,2} \lor B_{1,1})$

a3: $(\neg P_{2,1} \lor B_{1,1})$

b: $\neg B_{1,1}$

c: $P_{1,2}$

Resolution example

a1: $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1})$

a2: $(\neg P_{1,2} \vee B_{1,1})$

a3: $(\neg P_{2,1} \vee B_{1,1})$

b: $\neg B_{1,1}$

c: $P_{1,2}$

Step 1: resolve a2, c: $B_{1,1}$

Step 2: resolve above and b: *empty*

Efficiency of the resolution algorithm

- Run time can be exponential in the worst case
  - Often much faster
- Factoring: if a new clause contains duplicates of the same symbol, delete the duplicates

$$P \lor R \lor P \lor T \rightarrow P \lor R \lor T$$

- If a clause contains a symbol and its complement, the clause is a tautology and useless, it can be thrown away

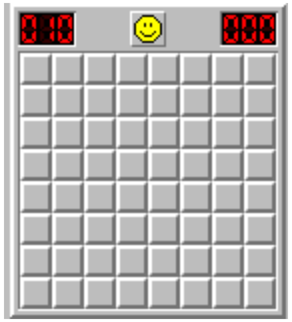
a1: $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1})$

a2: $(\neg P_{1,2} \lor B_{1,1})$

$\rightarrow P_{1,2} \lor P_{2,1} \lor \neg P_{1,2}$ (tautology, throw away)

Problems with propositional logic

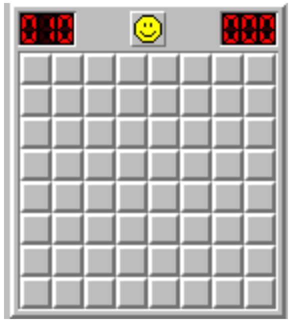
- Consider the game "minesweeper" on a 10x10 field with only one landmine.



- How do you express the knowledge, with propositional logic, that the squares adjacent to the landmine will display the number 1?

Problems with propositional logic

- Consider the game "minesweeper" on a 10x10 field with only one landmine.



- How do you express the knowledge, with propositional logic, that the squares adjacent to the landmine will display the number 1?
- Intuitively with a rule like

$$\text{landmine}(x,y) \Rightarrow \text{number1}(\text{neighbors}(x,y))$$

but propositional logic cannot do this…

Problems with propositional logic

- Propositional logic has to say, e.g. for cell (3,4):
  - Landmine_3_4 $\Rightarrow$ number1_2_3
  - Landmine_3_4 $\Rightarrow$ number1_2_4
  - Landmine_3_4 $\Rightarrow$ number1_2_5
  - Landmine_3_4 $\Rightarrow$ number1_3_3
  - Landmine_3_4 $\Rightarrow$ number1_3_5
  - Landmine_3_4 $\Rightarrow$ number1_4_3
  - Landmine_3_4 $\Rightarrow$ number1_4_4
  - Landmine_3_4 $\Rightarrow$ number1_4_5
  - And similarly for each of Landmine_1_1, Landmine_1_2, Landmine_1_3, …, Landmine_10_10!
- Difficult to express large domains concisely
- Don't have objects and relations
- First Order Logic is a powerful upgrade

Ontological commitment

- Logics are characterized by what they consider to be 'primitives'

Logic	Primitives	Available Knowledge
Propositional	facts	true/false/unknown
First-Order	facts, objects, relations	true/false/unknown
Temporal	facts, objects, relations, times	true/false/unknown
Probability Theory	facts	degree of belief 0…1
Fuzzy	degree of truth	degree of belief 0…1

First Order Logic syntax

- **Term**: an object in the world
  - **Constant**: Jerry, 2, Madison, Green, …
  - **Variables**: x, y, a, b, c, …
  - **Function**($term_1$, …, $term_n$)
    - Sqrt(9), Distance(Madison, Chicago)
    - Maps one or more objects to another object
    - Can refer to an unnamed object: LeftLeg(John)
    - Represents a user defined functional relation

- A **ground term** is a term without variables.

FOL syntax

- **Atom**: smallest T/F expression
  - **Predicate**(term$_1$, …, term$_n$)
    - Teacher(Jerry, you), Bigger(sqrt(2), x)
    - Convention: read "Jerry (is)Teacher(of) you"
    - Maps one or more objects to a truth value
    - Represents a user defined relation
  - **term$_1$ = term$_2$**
    - Radius(Earth)=6400km, 1=2
    - Represents the equality relation when two terms refer to the same object

FOL syntax

- **Sentence**: T/F expression
  - Atom
  - Complex sentence using connectives: $\wedge \vee \neg \Rightarrow \Leftrightarrow$
    - Spouse(Jerry, Jing) $\Rightarrow$ Spouse(Jing, Jerry)
    - Less(11,22) $\wedge$ Less(22,33)
  - Complex sentence using quantifiers $\forall$, $\exists$
- Sentences are evaluated under an interpretation
  - Which objects are referred to by constant symbols
  - Which objects are referred to by function symbols
  - What subsets defines the predicates

FOL quantifiers

- Universal quantifier: ∀
- Sentence is true **for all** values of x in the domain of variable x.

- Main connective typically is ⇒
  - Forms if-then rules
  - “all humans are mammals”

    $$\forall x\ \mathbf{human(x)} \Rightarrow \mathbf{mammal(x)}$$

  - Means if x is a human, then x is a mammal

FOL quantifiers

$$\forall x \ \texttt{human(x)} \Rightarrow \texttt{mammal(x)}$$

- It's a big AND: Equivalent to the conjunction of all the instantiations of variable x:

$$(\texttt{human(Jerry)} \Rightarrow \texttt{mammal(Jerry)}) \land$$
$$(\texttt{human(Jing)} \Rightarrow \texttt{mammal(Jing)}) \land$$
$$(\texttt{human(laptop)} \Rightarrow \texttt{mammal(laptop)}) \land \ldots$$

- Common mistake is to use $\land$ as main connective

$$\forall x \ \texttt{human(x)} \land \texttt{mammal(x)}$$

- This means everything is human and a mammal!

$$(\texttt{human(Jerry)} \land \texttt{mammal(Jerry)}) \land$$
$$(\texttt{human(Jing)} \land \texttt{mammal(Jing)}) \land$$
$$(\texttt{human(laptop)} \land \texttt{mammal(laptop)}) \land \ldots$$

FOL quantifiers

- Existential quantifier: ∃
- Sentence is true **for some** value of x in the domain of variable x.

- Main connective typically is ∧
  - "some humans are male"

    $\exists x\ \mathbf{human(x)} \wedge \mathbf{male(x)}$

  - Means there is an x who is a human and is a male

FOL quantifiers

$$\exists x\ \mathrm{human}(x) \wedge \mathrm{male}(x)$$

- It's a big OR: Equivalent to the disjunction of all the instantiations of variable x:

$$(\mathrm{human}(\mathrm{Jerry}) \wedge \mathrm{male}(\mathrm{Jerry})) \vee$$
$$(\mathrm{human}(\mathrm{Jing}) \wedge \mathrm{male}(\mathrm{Jing})) \vee$$
$$(\mathrm{human}(\mathrm{laptop}) \wedge \mathrm{male}(\mathrm{laptop})) \vee \ldots$$

- Common mistake is to use $\Rightarrow$ as main connective
  - "Some pig can fly"

$$\exists x\ \mathrm{pig}(x) \Rightarrow \mathrm{fly}(x) \quad \textbf{(wrong)}$$

FOL quantifiers

$$\exists x\ \text{human}(x) \land \text{male}(x)$$

- It's a big OR: Equivalent to the disjunction of all the instantiations of variable x:

$$(\text{human}(\text{Jerry}) \land \text{male}(\text{Jerry})) \lor$$
$$(\text{human}(\text{Jing}) \land \text{male}(\text{Jing})) \lor$$
$$(\text{human}(\text{laptop}) \land \text{male}(\text{laptop})) \lor \ldots$$

- Common mistake is to use $\Rightarrow$ as main connective
  - "Some pig can fly"

$$\exists x\ \text{pig}(x) \Rightarrow \text{fly}(x)\ \ \textbf{(wrong)}$$

- This is true if there is something not a pig!

$$(\text{pig}(\text{Jerry}) \Rightarrow \text{fly}(\text{Jerry})) \lor$$
$$(\text{pig}(\text{laptop}) \Rightarrow \text{fly}(\text{laptop})) \lor \ldots$$

FOL quantifiers

- Properties of quantifiers:
  - $\forall$**x** $\forall$**y** is the same as $\forall$**y** $\forall$**x**
  - $\exists$**x** $\exists$**y** is the same as $\exists$**y** $\exists$**x**

- Example:
  - $\forall$**x** $\forall$**y** **likes(x,y)**

    Everyone likes everyone.
  - $\forall$**y** $\forall$**x** **likes(x,y)**

    Everyone is liked by everyone.

FOL quantifiers

- Properties of quantifiers:
  - $\forall$**x** $\exists$**y** is not the same as $\exists$**y** $\forall$**x**
  - $\exists$**x** $\forall$**y** is not the same as $\forall$**y** $\exists$**x**

- Example:
  - $\forall$**x** $\exists$**y** **`likes(x,y)`**

    Everyone likes someone (can be different).
  - $\exists$**y** $\forall$**x** **`likes(x,y)`**

    There is someone who is liked by everyone.

FOL quantifiers

- Properties of quantifiers:
  - $\forall$**x P(x)** when negated becomes $\exists$**x** $\neg$**P(x)**
  - $\exists$**x P(x)** when negated becomes $\forall$**x** $\neg$**P(x)**

- Example:
  - $\forall$**x sleep(x)**

    Everybody sleeps.
  - $\exists$**x** $\neg$**sleep(x)**

    Somebody does not sleep.

FOL quantifiers

- Properties of quantifiers:
  - $\forall$**x P(x)** is the same as $\neg\exists$**x** $\neg$**P(x)**
  - $\exists$**x P(x)** is the same as $\neg\forall$**x** $\neg$**P(x)**

- Example:
  - $\forall$**x sleep(x)**

    Everybody sleeps.
  - $\neg\exists$**x** $\neg$**sleep(x)**

  There does not exist someone who does not sleep.

Professor Snape's Puzzle

Danger lies before you, while safety lies behind,
Two of us will help you, whichever you would find,
One among us seven will let you move ahead,
Another will transport the drinker back instead,
Two among our number hold only nettle-wine,
Three of us are killers, waiting hidden in line
Choose, unless you wish to stay here forevermore
To help you in your choice, we give you these clues four:
First, however slyly the poison tries to hide
You will always find some on nettle wine's left side
Second, different are those who stand at either end
But if you would move onward, neither is your friend;
Third as you see clearly, all are different size
Neither dwarf nor giant hold death in their insides;
Fourth, the second left and the second on the right
Are twins once you taste them, though different at first sight.

1. $\exists x\ A(x) \wedge (\forall y\ A(y) \Rightarrow x=y)$
2. $\exists x\ B(x) \wedge (\forall y\ B(y) \Rightarrow x=y)$
3. $\exists x \exists y\ W(x) \wedge W(y) \wedge \neg(x=y) \wedge (\forall z\ W(z) \Rightarrow z=x \vee z=y)$
4. $\forall x\ \neg(A(x) \vee B(x) \vee W(x)) \Rightarrow P(x)$
5. $\forall x \forall y\ W(x) \wedge L(y,x) \Rightarrow P(y)$
6. $\neg(P(b1) \wedge P(b7))$
7. $\neg(W(b1) \wedge W(b7))$
8. $\neg A(b1)$
9. $\neg A(b7)$
10. $\neg P(b3)$
11. $\neg P(b6)$
12. $(P(b2) \wedge P(b6)) \vee (W(b2) \wedge W(b6))$