FIRST-ORDER LOGIC

RECAP

Our KB is a conjunction ($\wedge$) of all rules, i.e.,

KB: $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

$R_1 : \neg P_{1,1}$

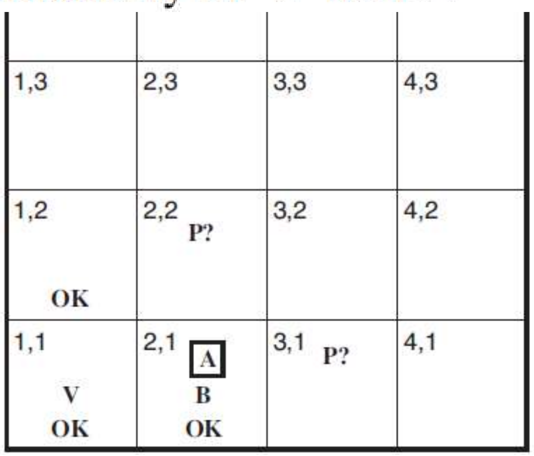
$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

$R_4 : \neg B_{1,1}$

$R_5 : B_{2,1}$

$$
\begin{array}{rcll}
(\alpha \wedge \beta) & \equiv & (\beta \wedge \alpha) & \text{commutativity of } \wedge \\
(\alpha \vee \beta) & \equiv & (\beta \vee \alpha) & \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) & \equiv & (\alpha \wedge (\beta \wedge \gamma)) & \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) & \equiv & (\alpha \vee (\beta \vee \gamma)) & \text{associativity of } \vee \\
\neg(\neg\alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \vee \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) & \equiv & (\neg\alpha \vee \neg\beta) & \text{De Morgan} \\
\neg(\alpha \vee \beta) & \equiv & (\neg\alpha \wedge \neg\beta) & \text{De Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) & \equiv & ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) & \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) & \equiv & ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) & \text{distributivity of } \vee \text{ over } \wedge
\end{array}
$$



FIRST-ORDER LOGIC

- Propositional Logic is **Useful** --- but has **Limited Expressive Power**
  - E.g., cannot say "Pits cause breezes in adjacent squares."
    - except by writing one sentence for each square
  - Needs to refer to objects in the world,
  - Needs to express general rules

- First Order Predicate Calculus (FOPC), or First Order Logic (FOL).
  - FOPC has greatly expanded expressive power, though still limited.
- New Ontology
  - The world consists of OBJECTS (for propositional logic, the world was facts).
  - OBJECTS have PROPERTIES and engage in RELATIONS and FUNCTIONS.
- New Syntax
  - Constants, Predicates, Functions, Properties, Quantifiers.

FIRST ORDER LOGIC

Objects (described by nouns in English, e.g., cat, John, house, wumpus, pit etc)

Relations (described by verbs in English e.g., has color, is bigger than, etc)

Facts describe relationship between objects and evaluate to True or False.

ONTOLOGICAL COMMITMENT

The primary difference between propositional and first-order logic lies in the **ontological commitment** made by each language—that is, what it assumes about the nature of *reality*.

Propositional logic assumes that there are facts that either hold or do not hold in the world.

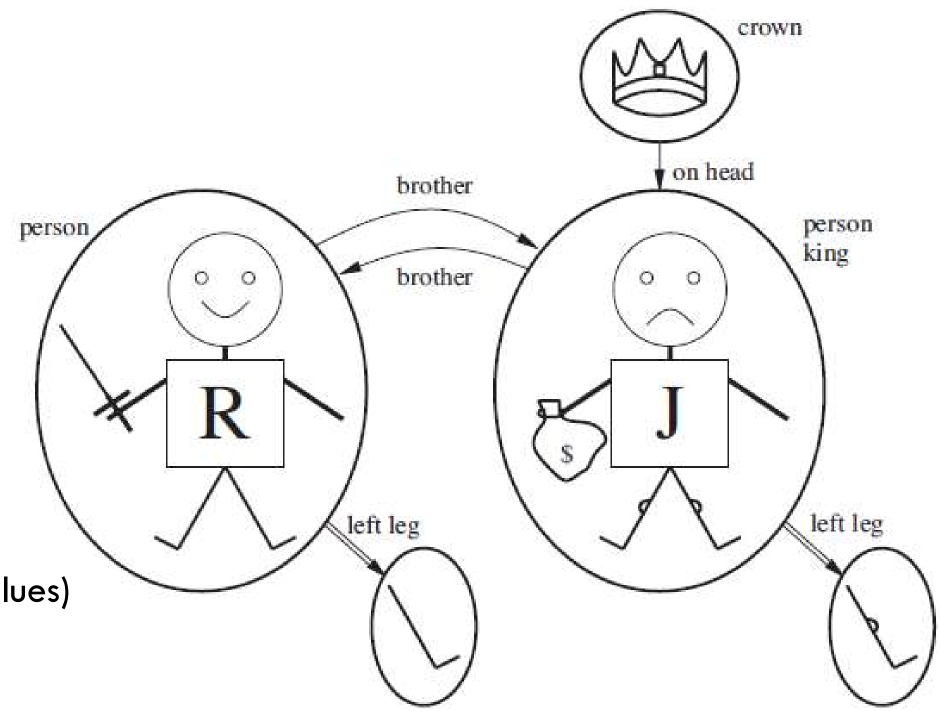
First-order logic assumes more; namely, that the world consists of objects with certain relations among them that do or do not hold

PROPOSITIONAL LOGIC VS FIRST ORDER LOGIC

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

Figure 8.1 Formal languages and their ontological and epistemological commitments.

SYNTAX



Three kinds of symbols,

- Constant: objects
- Predicate: relations
- Functions: functions (can return objects instead of binary values)

Predicate and Functions have arity (number of arguments of a function)

Symbols have an interpretation

Complex Sentences: $\neg$ King(Richard) $\Rightarrow$ King(John) (If Richard isn't the king then john must be.

SYNTAX OF FOL: BASIC SYNTAX ELEMENTS ARE SYMBOLS

- **Constant** Symbols (correspond to English nouns)
  - Stand for objects in the world.
    - E.g., KingJohn, 2, UCI, ...

- **Predicate** Symbols (correspond to English verbs)
  - Stand for relations (maps a tuple of objects to a **truth-value**)
    - E.g., Brother(Richard, John), greater_than(3,2), ...
  - P(x, y) is usually read as "x is P of y."
    - E.g., Mother(Ann, Sue) is usually "Ann is Mother of Sue."

- **Function** Symbols (correspond to English nouns)
  - Stand for functions (maps a tuple of objects to an **object**)
    - E.g., Sqrt(3), LeftLegOf(John), ...

- **Model** (world) = set of domain objects, relations, functions
- **Interpretation** maps symbols onto the model (world)
  - Very many interpretations are possible for each KB and world!
  - Job of the KB is to rule out models inconsistent with our knowledge.

SYNTAX OF FOL: ATOMIC SENTENCES

- **Atomic Sentences** state facts (logical truth values).
  - An **atomic sentence** is a Predicate symbol, optionally followed by a parenthesized list of any argument terms
    - E.g., *Married(Father(Richard), Mother(John))*
  - An **atomic sentence** asserts that some relationship (some predicate) holds among the objects that are its arguments.

- An **Atomic Sentence is true** in a given model if the relation referred to by the predicate symbol holds among the objects (terms) referred to by the arguments.

SYNTAX OF FOL: CONNECTIVES & COMPLEX SENTENCES

- **Complex Sentences** are formed in the same way, and are formed using the same logical connectives, as we already know from propositional logic

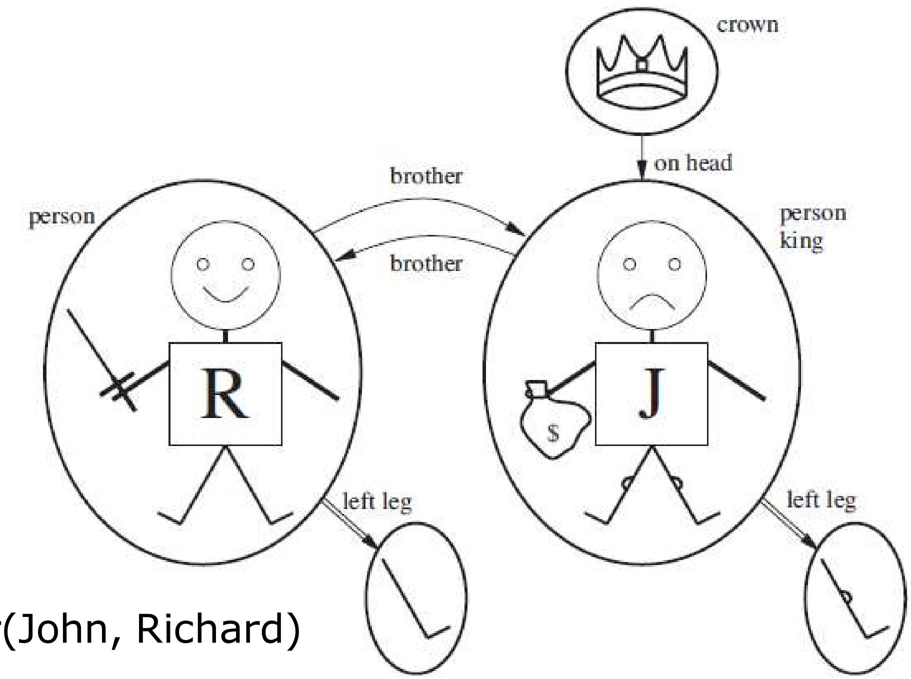
- The **Logical Connectives**:
  - $\Leftrightarrow$ biconditional
  - $\Rightarrow$ implication
  - $\wedge$ and
  - $\vee$ or
  - $\neg$ negation

- **Semantics** for these logical connectives are the same as we already know from propositional logic.

SYNTAX

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} & \rightarrow & (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
& \mid & \neg\, \textit{Sentence} \\
& \mid & \textit{Sentence} \wedge \textit{Sentence} \\
& \mid & \textit{Sentence} \vee \textit{Sentence} \\
& \mid & \textit{Sentence} \;\Rightarrow\; \textit{Sentence} \\
& \mid & \textit{Sentence} \;\Leftrightarrow\; \textit{Sentence} \\
& \mid & \textit{Quantifier}\ \textit{Variable}, \ldots\ \textit{Sentence} \\
\\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term}, \ldots) \\
& \mid & \textit{Constant} \\
& \mid & \textit{Variable} \\
\\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{array}
$$

MODELS IN FOL

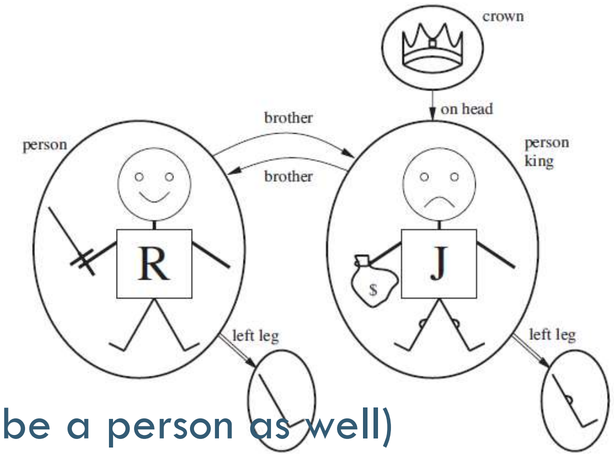


One possible model

Examples

- Brother(Richard, John) $\wedge$ Brother(John, Richard)
- King(Richard) $\vee$ King(John)
- King(John) => $\neg$ King(Richard)

UNIVERSAL QUANTIFICATION ∀



∀ x King(x) ⇒ Person(x) (For every x such that x is a king, x will be a person as well)

⇒ says every king is a person, but every person is not a king.

M={x → Richard, x → John, x → Richard's left leg, x → John's left leg, x → the crown}

Out of these possible five models, there's only one for which (King(x) is True which is x → John and for this value of x, Person(x) will also be True)

We are not claiming for other 4 values that each one of those 4 cannot be a person!

UNIVERSAL QUANTIFICATION

- Universal quantification is equivalent to:
  - Conjunction of all sentences obtained by substitution of an object for the quantified variable.

- All Cats are Mammals.
  - $\forall x\ Cat(x) \Rightarrow Mammal(x)$

- Conjunction of all sentences obtained by substitution of an object for the quantified variable:

  $Cat(Spot) \Rightarrow Mammal(Spot) \wedge Cat(Rebecca) \Rightarrow Mammal(Rebecca) \wedge$

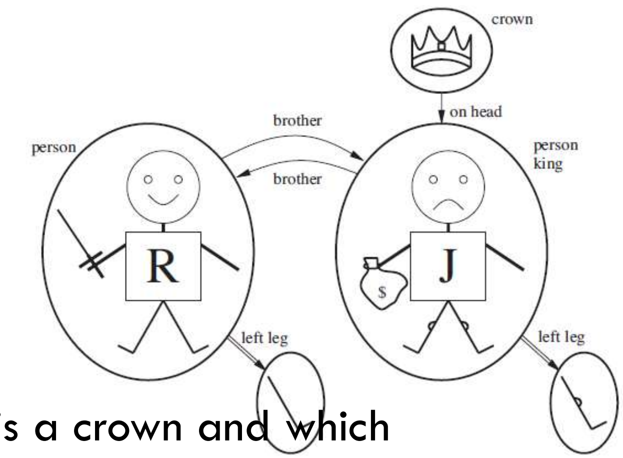
  $Cat(LAX) \Rightarrow Mammal(LAX) \wedge Cat(Shayama) \Rightarrow Mammal(Shayama) \wedge$

  $Cat(France) \Rightarrow Mammal(France) \wedge$

  $Cat(Felix) \Rightarrow Mammal(Felix) \wedge$

  ...

EXISTENTIAL QUANTIFICATION ∃



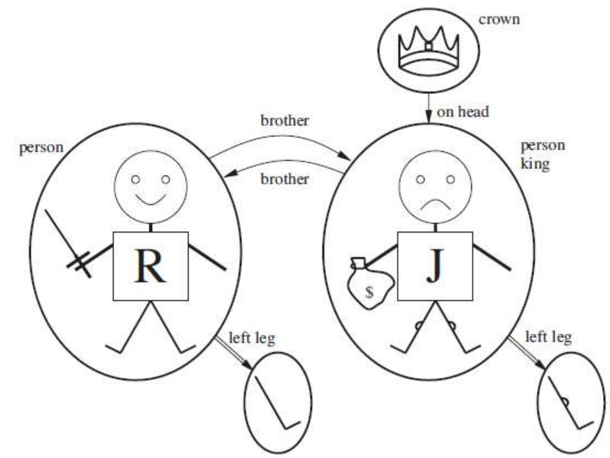
∃ x Crown(x) ∧ onHead(x,John) (There exists atleast one x that is a crown and which is placed on John's head.

Intuitively, the sentence ∃x P says that P is true for at least one object x.

M={x → Richard, x → John, x → Richard's left leg, x → John's left leg, x → the crown}. Therefore one of the following statements MUST BE correct.

- Richard the Lionheart is a crown ∧ Richard the Lionheart is on John's head;
- King John is a crown ∧ King John is on John's head;
- Richard's left leg is a crown ∧ Richard's left leg is on John's head;
- John's left leg is a crown ∧ John's left leg is on John's head;
- The crown is a crown ∧ the crown is on John's head

∃, COMPARING ⇒ WITH ∧



$\exists$ x Crown(x) $\land$ onHead(x,John)

- There exists atleast one x that is a crown and which is placed on John's head.

$\exists$ x Crown(x) $\Rightarrow$ onHead(x,John)

- If x exists such that it's a crown, then its placed on John's head.
- But what for that x, which isn't a crown? i.e., for which Crown(x) is False, it can still imply onHead(x,John) is True.

P	Q	$P \Rightarrow Q$
false	*false*	*true*
false	*true*	*true*
true	*false*	*false*
true	*true*	*true*

EXISTENTIAL QUANTIFICATION

- Existential quantification is equivalent to:
  - Disjunction of all sentences obtained by substitution of an object for the quantified variable.

- Spot has a sister who is a cat.
  - $\exists x$ Sister(x, Spot) $\wedge$ Cat(x)

- Disjunction of all sentences obtained by substitution of an object for the quantified variable:

Sister(Spot, Spot) $\wedge$ *Cat(Spot)* $\vee$ *Sister(Rebecca, Spot)* $\wedge$ *Cat(Rebecca)* $\vee$

Sister(LAX, Spot) $\wedge$ *Cat(LAX)* $\vee$ *Sister(Shayama, Spot)* $\wedge$ *Cat(Shayama)* $\vee$

Sister(France, Spot) $\wedge$ *Cat(France)* $\vee$

Sister(Felix, Spot) $\wedge$ *Cat(Felix)* $\vee$

...

NESTED QUANTIFIERS

$\forall$ x $\forall$ y Brother (x, y) $\Rightarrow$ Sibling(x, y)

- All brothers are siblings

$\forall$ x,y Sibling(x, y) $\Leftrightarrow$ Sibling(y, x)   (siblinghood is a symmetric relationship)

- Consecutive quantifiers of the same type can be written as one quantifier with several variables.

ORDER OF QUANTIFICATION

Loves(x, y) means x loves y, (not necessarily the other way around), then,

- ∀ x ∃ y Loves(x, y) ….. Everybody loves somebody / For every x, there exists a y ∀ x (∃ y Loves(x, y)) ….. that x loves
- ∃ y ∀ x Loves(x, y) …… There is someone who is loved by everyone / There exists a y that is loved by every x ∃ y (∀ x Loves(x, y)) ……

X	Y
John	Sara
Rob	Elizabeth
Richard	Meryl
Ned	Cristina

∀ y (∃ x Loves(x, y)) ….. (meaning ?)

∃ x (∀ y Loves(x, y)) ….. (meaning ?)

CONNECTIONS BETWEEN ∀ AND ∃

The two quantifiers are actually intimately connected with each other, through negation.

$\forall x \neg Likes(x, apples)$ is equivalent to $\neg \exists x \, Likes(x, apples)$

- everyone dislikes apples is the same as asserting there does not exist someone who likes them

$\forall x \, Likes(x, IceCream)$ is equivalent to $\neg \exists x \neg Likes(x, IceCream)$

- Everyone likes ice cream means that there is no one who does not like ice cream

DEMORGAN'S LAWS

$\forall x \neg P \equiv \neg \exists x P$

- For every x, P is false $\equiv$ There doesn't exists an x for which P is True

$\neg \forall x P \equiv \exists x \neg P$

- Not for every x is P true $\equiv$ There exists an x for which P is False

$\forall x P \equiv \neg \exists x \neg P$

- For every x, P is True $\equiv$ There does not exists an x for which P is False

$\exists x P \equiv \neg \forall x \neg P$

- There is an x for which P is True $\equiv$ Not for every x is P False

EQUALITY

We can use the **equality symbol** to signify that two terms refer to the same object.

- Father (John)=Henry

To say that Richard has at least two brothers, we would write

- ∃ x, y Brother (x,Richard) ∧ Brother (y,Richard) ∧ ¬(x=y)

The following sentence is true in the model where Richard has only one brother, i.e., where x and y refer to the same object

- ∃ x, y Brother (x,Richard) ∧ Brother (y,Richard)

In both sentences, Richard can have more than 2 brothers

The notation x ≠ y is sometimes used as an abbreviation for ¬(x=y)

Fun with sentences

Brothers are siblings

Fun with sentences

Brothers are siblings

$$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$$

"Sibling" is symmetric

Fun with sentences

Brothers are siblings

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$

"Sibling" is symmetric

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$

One's mother is one's female parent

Fun with sentences

Brothers are siblings

$$\forall x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$$

"Sibling" is symmetric

$$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$$

One's mother is one's female parent

$$\forall x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \land Parent(x, y)).$$

A first cousin is a child of a parent's sibling

Fun with sentences

Brothers are siblings

$$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$$

"Sibling" is symmetric

$$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$$

One's mother is one's female parent

$$\forall\, x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \wedge Parent(x, y)).$$

A first cousin is a child of a parent's sibling

$$\forall\, x, y \;\; FirstCousin(x, y) \;\Leftrightarrow\; \exists\, p, ps \;\; Parent(p, x) \wedge Sibling(ps, p) \wedge Parent(ps, y)$$

MORE FUN WITH SENTENCES

- **“All persons are mortal.”**
- [Use: Person(x), Mortal (x)]

MORE FUN WITH SENTENCES

- **“All persons are mortal.”**
  [Use: Person(x), Mortal (x)]

- $\forall x$ Person(x) $\Rightarrow$ Mortal(x)

- **Equivalent Forms:**
- $\forall x$ $\neg$Person(x) $\vee$Mortal(x)

- **Common Mistakes:**
- $\forall x$ Person(x) $\wedge$ Mortal(x)

MORE FUN WITH SENTENCES

- **“Fifi has a sister who is a cat.”**
- [Use: Sister(Fifi, x), Cat(x)]
-

MORE FUN WITH SENTENCES

- **"Fifi has a sister who is a cat."**
- [Use: Sister(Fifi, x), Cat(x)]

- $\exists x$ Sister(Fifi, x) $\wedge$ Cat(x)

- **Common Mistakes:**
- $\exists x$ Sister(Fifi, x) $\Rightarrow$ Cat(x)

MORE FUN WITH SENTENCES

- **“For every food, there is a person who eats that food.”**
  [Use: Food(x), Person(y), Eats(y, x)]

MORE FUN WITH SENTENCES

- **"For every food, there is a person who eats that food."**
  [Use: Food(x), Person(y), Eats(y, x)]

- $\forall x\ \exists y\ \text{Food}(x) \Rightarrow [\ \text{Person}(y) \wedge \text{Eats}(y, x)\ ]$

- **Equivalent Forms:**
- $\forall x\ \text{Food}(x) \Rightarrow \exists y\ [\ \text{Person}(y) \wedge \text{Eats}(y, x)\ ]$
- $\forall x\ \exists y\ \neg\text{Food}(x) \vee [\ \text{Person}(y) \wedge \text{Eats}(y, x)\ ]$
- $\forall x\ \exists y\ [\ \neg\text{Food}(x) \vee \text{Person}(y)\ ] \wedge [\neg\ \text{Food}(x) \vee \text{Eats}(y, x)\ ]$
- $\forall x\ \exists y\ [\ \text{Food}(x) \Rightarrow \text{Person}(y)\ ] \wedge [\ \text{Food}(x) \Rightarrow \text{Eats}(y, x)\ ]$

- **Common Mistakes:**
- $\forall x\ \exists y\ [\ \text{Food}(x) \wedge \text{Person}(y)\ ] \Rightarrow \text{Eats}(y, x)$
- $\forall x\ \exists y\ \text{Food}(x) \wedge \text{Person}(y) \wedge \text{Eats}(y, x)$

MORE FUN WITH SENTENCES

- **“Every person eats every food.”**
  [Use: Person (x), Food (y), Eats(x, y)]

MORE FUN WITH SENTENCES

- **"Every person eats every food."**
  [Use: Person (x), Food (y), Eats(x, y)]
- $\forall x\ \forall y\ [\ Person(x) \wedge Food(y)\ ] \Rightarrow Eats(x, y)$
- **Equivalent Forms:**
- $\forall x\ \forall y\ \neg Person(x) \vee \neg Food(y) \vee Eats(x, y)$
- $\forall x\ \forall y\ Person(x) \Rightarrow [\ Food(y) \Rightarrow Eats(x, y)\ ]$
- $\forall x\ \forall y\ Person(x) \Rightarrow [\ \neg Food(y) \vee Eats(x, y)\ ]$
- $\forall x\ \forall y\ \neg Person(x) \vee [\ Food(y) \Rightarrow Eats(x, y)\ ]$
- **Common Mistakes:**
- $\forall x\ \forall y\ Person(x) \Rightarrow [Food(y) \wedge Eats(x, y)\ ]$
- $\forall x\ \forall y\ Person(x) \wedge Food(y) \wedge Eats(x, y)$

MORE FUN WITH SENTENCES

- **“All greedy kings are evil.”**

  [Use: King(x), Greedy(x), Evil(x)]

MORE FUN WITH SENTENCES

- **"All greedy kings are evil."**
  [Use: King(x), Greedy(x), Evil(x)]

- $\forall x$ [Greedy(x) $\wedge$ King(x)] $\Rightarrow$ Evil(x)

- **Equivalent Forms:**
- $\forall x$ $\neg$Greedy(x) $\vee$ $\neg$King(x) $\vee$ Evil(x)
- $\forall x$ Greedy(x) $\Rightarrow$ [King(x) $\Rightarrow$ Evil(x)]

- **Common Mistakes:**
- $\forall x$ Greedy(x) $\wedge$ King(x) $\wedge$ Evil(x)

MORE FUN WITH SENTENCES

- **"Everyone has a favorite food."**

  [Use: Person(x), Food(y), Favorite(y, x)]

MORE FUN WITH SENTENCES

- **"Everyone has a favorite food."**
  [Use: Person(x), Food(y), Favorite(y, x)]

- **Equivalent Forms:**
- $\forall x \; \exists y \; \text{Person}(x) \Rightarrow [\; \text{Food}(y) \wedge \text{Favorite}(y, x) \;]$
- $\forall x \; \text{Person}(x) \Rightarrow \exists y \; [\; \text{Food}(y) \wedge \text{Favorite}(y, x) \;]$
- $\forall x \; \exists y \; \neg\text{Person}(x) \vee [\; \text{Food}(y) \wedge \text{Favorite}(y, x) \;]$
- $\forall x \; \exists y \; [\; \neg\text{Person}(x) \vee \text{Food}(y) \;] \wedge [\; \neg\text{Person}(x) \vee \text{Favorite}(y, x) \;]$
- $\forall x \; \exists y \; [\text{Person}(x) \Rightarrow \text{Food}(y) \;] \wedge [\; \text{Person}(x) \Rightarrow \text{Favorite}(y, x) \;]$

- **Common Mistakes:**
- $\forall x \; \exists y \; [\; \text{Person}(x) \wedge \text{Food}(y) \;] \Rightarrow \text{Favorite}(y, x)$
- $\forall x \; \exists y \; \text{Person}(x) \wedge \text{Food}(y) \wedge \text{Favorite}(y, x)$

MORE FUN WITH SENTENCES

- **“There is someone at UCI who is smart.”**
  [Use: Person(x), At(x, UCI), Smart(x)]

MORE FUN WITH SENTENCES

- **"There is someone at UCI who is smart."**
  [Use: Person(x), At(x, UCI), Smart(x)]
- $\exists x$ Person(x) $\wedge$ At(x, UCI) $\wedge$ Smart(x)
- **Common Mistakes:**
- $\exists x$ [Person(x) $\wedge$ At(x, UCI)] $\Rightarrow$ Smart(x)

MORE FUN WITH SENTENCES

- **"Everyone at UCI is smart."**

  [Use: Person(x), At(x, UCI), Smart(x)]

MORE FUN WITH SENTENCES

- **"Everyone at UCI is smart."**
  [Use: Person(x), At(x, UCI), Smart(x)]

- $\forall x$ [Person(x) $\wedge$ At(x, UCI)] $\Rightarrow$ Smart(x)

- **Equivalent Forms:**
- $\forall x$ $\neg$[Person(x) $\wedge$ At(x, UCI)] $\vee$ Smart(x)
- $\forall x$ $\neg$Person(x) $\vee\neg$At(x, UCI) $\vee$ Smart(x)

- **Common Mistakes:**
- $\forall x$ Person(x) $\wedge$ At(x, UCI) $\wedge$ Smart(x)
- $\forall x$ Person(x) $\Rightarrow$ [At(x, UCI) $\wedge$ Smart(x)]
-

MORE FUN WITH SENTENCES

- **“Every person eats some food.”**

  [Use: Person (x), Food (y), Eats(x, y)]

MORE FUN WITH SENTENCES

- **"Every person eats some food."**
  [Use: Person (x), Food (y), Eats(x, y)]
  -
  - $\forall x\ \exists y\ \text{Person}(x) \Rightarrow [\ \text{Food}(y) \wedge \text{Eats}(x, y)\ ]$
  - **Equivalent Forms:**
  - $\forall x\ \text{Person}(x) \Rightarrow \exists y\ [\ \text{Food}(y) \wedge \text{Eats}(x, y)\ ]$
  - $\forall x\ \exists y\ \neg\text{Person}(x) \vee [\ \text{Food}(y) \wedge \text{Eats}(x, y)]$
  - $\forall x\ \exists y\ [\ \neg\text{Person}(x) \vee \text{Food}(y)\ ] \wedge [\ \neg\text{Person}(x) \vee \text{Eats}(x, y)\ ]$
- **Common Mistakes:**
- $\forall x\ \exists y\ [\ \text{Person}(x) \wedge \text{Food}(y)\ ] \Rightarrow \text{Eats}(x, y)$
- $\forall x\ \exists y\ \text{Person}(x) \wedge \text{Food}(y) \wedge \text{Eats}(x, y)$
-

MORE FUN WITH SENTENCES

- **“Some person eats some food.”**

  [Use: Person (x), Food (y), Eats(x, y)]

MORE FUN WITH SENTENCES

- **“Some person eats some food.”**

  [Use: Person (x), Food (y), Eats(x, y)]

- $\exists x \ \exists y$ Person(x) $\wedge$ Food(y) $\wedge$ Eats(x, y)

- **Common Mistakes:**
- $\exists x \ \exists y$ [Person(x) $\wedge$ Food(y)] $\Rightarrow$ Eats(x, y)