AI基础

Lecture 8: First-Order Logic

Bin Yang
School of Data Science and Engineering
byang@dase.ecnu.edu.cn
https://binyangdk.github.io/

[Some slides adapted from Philipp Koehn, JHU]

Lecture 7 ILOs

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta} \qquad \frac{\alpha \wedge \beta}{\alpha}.$$

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \quad \text{and} \quad \frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}.$$

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

l_i and m_j are complementary literals

- Knowledge-based agents
  - Wumpus world
- Logic in general
  - Models and entailment
- Propositional logic
  - Symbols, parentheses, and 5 logical connectives: not, and, or, implies, biconditional
  - Inference
    - Model checking: Enumerate the models and check whether α is true in every model in which KB is true
- Equivalence, validity, satisfiability $\quad \alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$.
- Inference rules and theorem proving
  - Search with inference rules
  - Resolution: To prove KB ⊨ α, we show (KB ∧ ¬ α) is unsatisfiable
  - forward chaining, only for Horn KB, data driven
  - backward chaining, only for Horn KB, goal driven
  - (Proofs do not involve model enumerations, while model checking involves.)

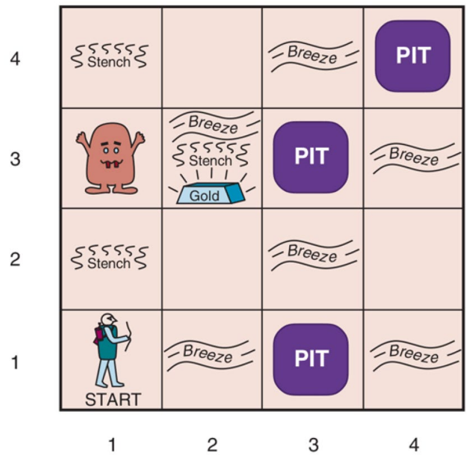
Lecture 8 ILOs

- Limitations of propositional logic
- First-order logic (FOL)
- Syntax and Semantics
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic
  - Unification
  - Forward chaining, backward chaining, resolution

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
    - Models, symbols, interpretations, terms
    - Atomic and complex sentences
    - Quantifiers, universal and existential quantifiers, equality
    - Database semantics
- Inference
    - Reduce FOL to propositional logic
    - Unification
    - Forward chaining, backward chaining, resolution

Pros and cons of propositional logic



- Pros
  - Declarative: knowledge and inference are separate, and inference is entirely domain independent
  - Propositional logic allows partial/disjunctive/negated information
  - Compositionality: meaning of $B_{1,1} \land P_{1,2}$ is derived from the meaning of $B_{1,1}$ and of $P_{1,2}$
- Cons
  - Lacks the expressive power to concisely describe an environment with many objects (unlike natural languages)
  - Cannot say "Pits cause breezes in adjacent squares" except by writing one sentence for each square

$$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})\,.$$

$$B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})\,. \quad \cdots$$

Combining the best of formal and natural languages

- When we look at the syntax of natural language, the most obvious elements include:
- Nouns and noun phrases that refer to **objects** (squares, pits, wumpuses)
  - people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries, ···
- Verbs and verb phrases along with adjectives and adverbs that refer to **relations** among objects (is breezy, is adjacent to, shoots).
  - these can be unary relations or **properties** such as red, round, bogus, prime, multistoried ···,
  - or more general n-ary relations such as brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, ···
- Some of these relations are **functions**—relations in which there is only one "value" for a given "input."
  - father of, best friend, third inning of, one more than, beginning of …

Some examples

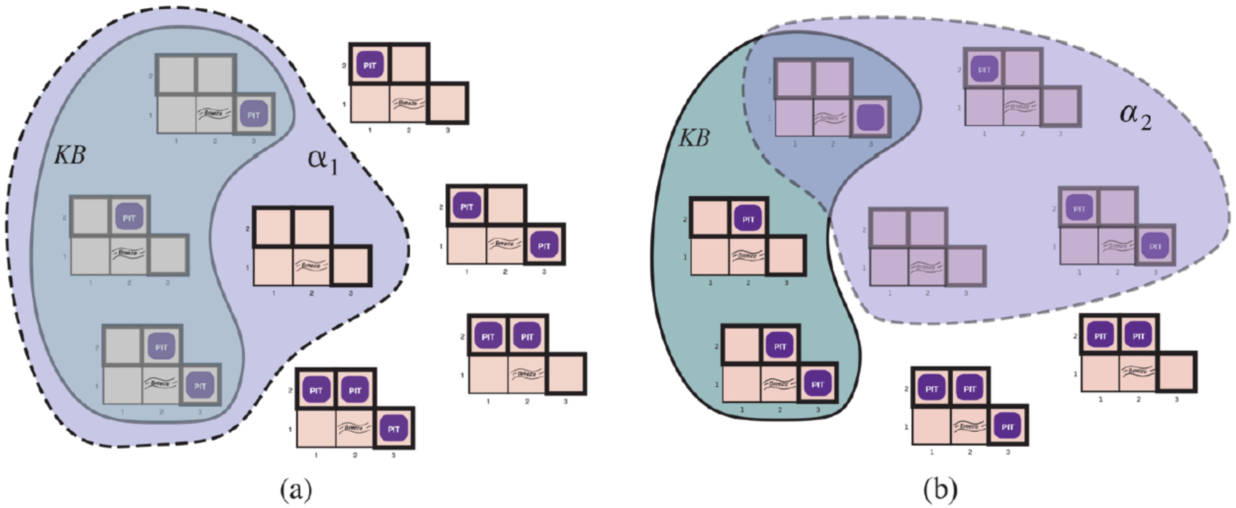
- “One plus two equals three.”
  - Objects: one, two, three, one plus two; Relation: equals; Function: plus.
  - “One plus two” is a name for the object that is obtained by applying the function “plus” to the objects “one” and “two.” “Three” is another name for this object.
- “Squares neighboring the wumpus are smelly.”
  - Objects: wumpus, squares; Property: smelly; Relation: neighboring.
- “Evil King John ruled England in 1200.”
  - Objects: John, England, 1200; Relation: ruled during; Properties: evil, king.
- **First order logic is built around objects and relations.**
  - First-order logic can also express facts about some or all of the objects in the universe. This enables one to represent general laws or rules, such as the statement “Squares neighboring the wumpus are smelly.”

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic
  - Unification
  - Forward chaining, backward chaining, resolution

Models of First-Order Logic (FOL)

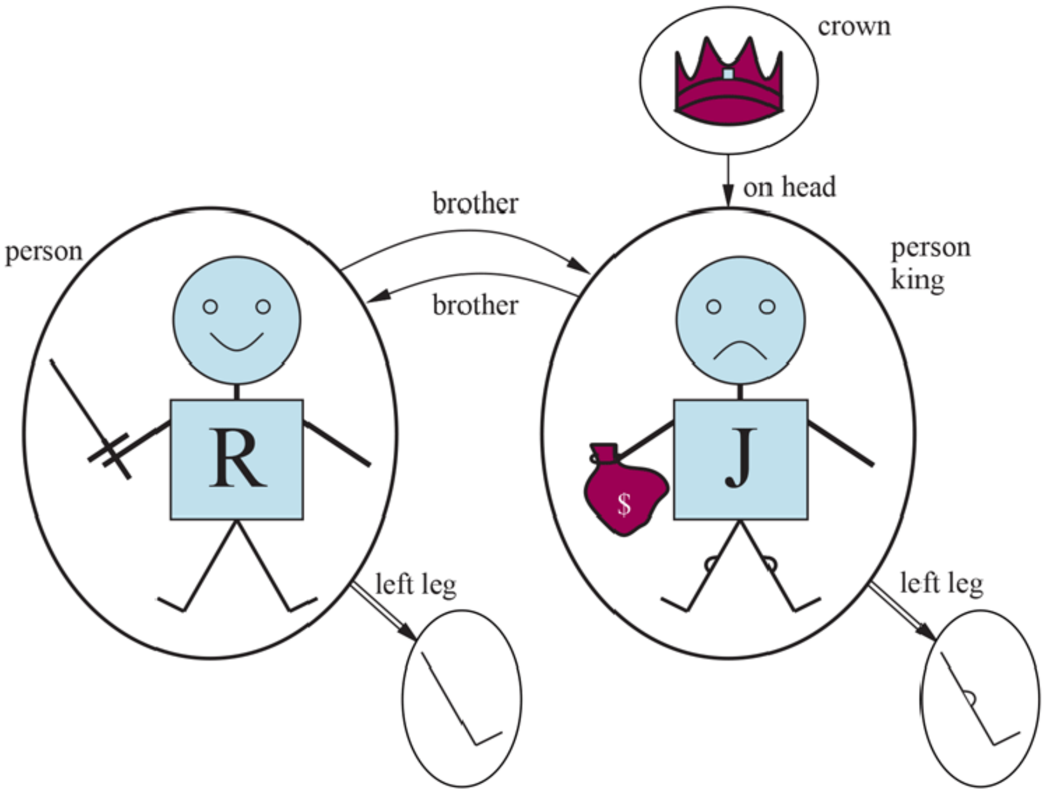
- Models in propositional logic
  - Possible worlds
  - Whether (1, 2), (2, 2), and (3, 1) contain pits.
  - 2^3 possible models.
- Models in FOL
  - FOL has objects
  - The domain of a model is the set of objects.



Possible models for the presence of pits in squares [1,2], [2,2], and [3,1]. The KB corresponding to the observations of nothing in [1,1] and a breeze in [2,1] is shown by the solid line. (a) Dotted line shows models of α_1 (no pit in [1,2]). (b) Dotted line shows models of α_2 (no pit in [2,2]).

Examples of FOL Models

- Five objects
  - **Richard** the Lionheart, King of England from 1189 to 1199;
  - his younger brother, the evil King **John**, who ruled from 1199 to 1215;
  - the **left legs** of Richard and John;
  - a **crown**.



A model containing five objects, two binary relations (brother and on-head), three unary relations (person, king, and crown), and one unary function (left-leg).

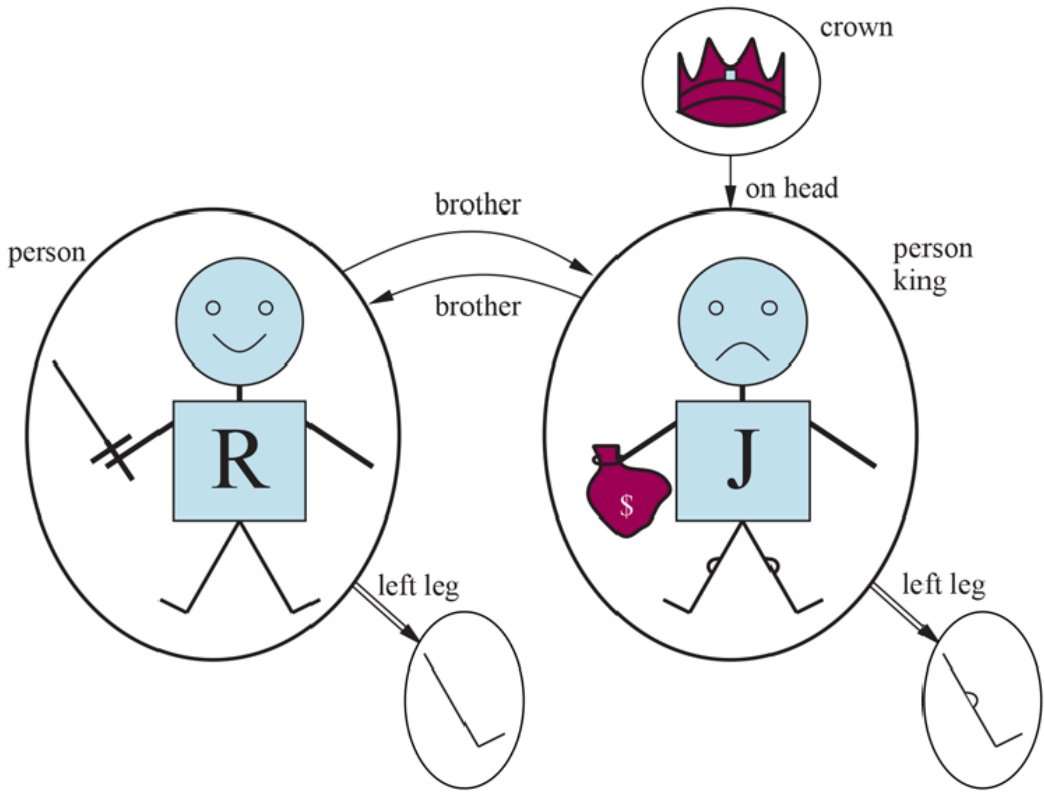
Examples of FOL Models

- Relations
  - A relation is just the set of tuples of objects that are related
  - Brotherhood

$$\{\langle \text{Richard the Lionheart, King John} \rangle, \langle \text{King John, Richard the Lionheart} \rangle\}\ .$$

  - On head
    - ⟨the crown, King John⟩
- Property, unary relationship
  - Person
    - Richard, John
  - King
    - John
- Function
  - Left leg function

$$\langle \text{Richard the Lionheart} \rangle \to \text{Richard's left leg}$$
$$\langle \text{King John} \rangle \to \text{John's left leg}.$$



A model containing five objects, two binary relations (brother and on-head), three unary relations (person, king, and crown), and one unary function (left-leg).

More logics

- Ontological commitment

Figure 8.1

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

Formal languages and their ontological and epistemological commitments.

Higher-order logic:
relations and functions operate not only on objects, but also on relations and functions.

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic
  - Unification
  - Forward chaining, backward chaining, resolution

Syntax of FOL

Syntax of Propositional Logic

Figure 7.7

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} &\rightarrow (\, \mathit{Sentence} \,) \\
&\mid\ \neg\, \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \land \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \lor \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \Leftrightarrow \mathit{Sentence}
\end{aligned}
$$

OPERATOR PRECEDENCE : $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$

A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

Relations

Objects

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \boxed{\mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term}, \ldots)} \mid \boxed{\mathit{Term} = \mathit{Term}} \\
\mathit{ComplexSentence} &\rightarrow (\, \mathit{Sentence} \,) \\
&\mid\ \neg\, \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \land \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \lor \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
&\mid\ \mathit{Sentence} \Leftrightarrow \mathit{Sentence} \\
&\mid\ \boxed{\mathit{Quantifier}\ \mathit{Variable}, \ldots\ \mathit{Sentence}} \\
\\
\mathit{Term} &\rightarrow \mathit{Function}(\mathit{Term}, \ldots) \\
&\mid\ \mathit{Constant} \\
&\mid\ \mathit{Variable} \\
\\
\mathit{Quantifier} &\rightarrow \forall \mid \exists \\
\mathit{Constant} &\rightarrow A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} &\rightarrow a \mid x \mid s \mid \cdots \\
\mathit{Predicate} &\rightarrow \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} &\rightarrow \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots
\end{aligned}
$$

OPERATOR PRECEDENCE : $\neg, =, \land, \lor, \Rightarrow, \Leftrightarrow$

The syntax of first-order logic with equality, specified in Backus–Naur form (see page 1030 if you are not familiar with this notation). Operator precedences are specified, from highest to lowest. The precedence of quantifiers is such that a quantifier holds over everything to the right of it.

Symbols and Interpretation

Language	Ontological Commitment (What exists in the world)
Propositional logic	facts
First-order logic	facts, objects, relations

- Symbols
  - **Constant** symbols, which stand for objects
    - Richard, John
  - **Predicate** symbols, which stand for relations
    - Brother, OnHead, Person, King, Crown
  - **Function** symbols, which stand for functions
    - LeftLeg
  - Each predicate and function symbol comes with an **arity** that fixes the number of arguments
- Proposition symbols in propositional logic
  - Facts, true or false, $B_{1,1}$

Symbols and Interpretation

(8.1)

$$\{\langle \text{Richard the Lionheart, King John} \rangle, \langle \text{King John, Richard the Lionheart} \rangle\}\ .$$

- Interpretation
  - specifies exactly which objects, relations and functions are referred to by the constant, predicate, and function symbols
  - *Richard* refers to Richard the Lionheart and *John* refers to the evil King John.
    - There are 25 possible interpretations for constant symbols as there are 5 objects
  - *Brother* refers to the brotherhood relation—that is, the set of tuples of objects given in Equation (8.1) ; *OnHead* is a relation that holds between the crown and King John;
  - *Person*, *King*, and *Crown* are unary relations that identify persons, kings, and crowns.
  - *LeftLeg* refers to the “left leg” function as defined in Equation (8.2)

(8.2)

$$\langle \text{Richard the Lionheart} \rangle \rightarrow \text{Richard's left leg}$$
$$\langle \text{King John} \rangle \rightarrow \text{John's left leg.}$$

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic
  - Unification
  - Forward chaining, backward chaining, resolution

Atomic Sentences

Constant symbols
Function symbols
Predicate symbols → Predicate

Terms → Objects

Predicate → Relations

Atomic sentences
predicate($term_1$, ⋯, $term_n$)

- Terms
  - Logical expression that refers to an **object**.
  - Constant symbols are terms.
  - Function symbols, LeftLeg(John)
- We have
  - Terms referring to objects
  - Predicate symbols referring to relations
- We make atomic sentences
  - predicate($term_1$, ⋯, $term_n$)
  - P(x, y): x is a P of y.
- Examples
  - Brother(Richard, John): Richard is a brother of John.
  - Married(Father(Richard),Mother(John)): Richard the Lionheart’s father is married to King John’s mother

$$\begin{array}{lcl} AtomicSentence & \rightarrow & Predicate \mid Predicate(Term, \ldots) \mid Term = Term \end{array}$$

$$\begin{array}{lcl} Term & \rightarrow & Function(Term, \ldots) \\ & \mid & Constant \\ & \mid & Variable \end{array}$$

Complex sentences

- Complex sentences are made from atomic sentences using logical connectives

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \implies S_2, \quad S_1 \Leftrightarrow S_2$$

$$\neg Brother(LeftLeg(Richard), John)$$
$$Brother(Richard, John) \wedge Brother(John, Richard)$$
$$King(Richard) \vee King(Richard)$$
$$\neg King(Richard) \Rightarrow King(John).$$

$$
\begin{array}{lcl}
ComplexSentence & \rightarrow & (\ Sentence\ ) \\
 & | & \neg\ Sentence \\
 & | & Sentence \wedge Sentence \\
 & | & Sentence \vee Sentence \\
 & | & Sentence \ \Rightarrow\ Sentence \\
 & | & Sentence \ \Leftrightarrow\ Sentence \\
 & | & Quantifier\ Variable, \ldots\ Sentence
\end{array}
$$

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
    - Models, symbols, interpretations, terms
    - Atomic and complex sentences
    - Quantifiers, universal and existential quantifiers, equality
    - Database semantics
- Inference
    - Reduce FOL to propositional logic
    - Unification
    - Forward chaining, backward chaining, resolution

Quantifiers

- Once we have a logic that allows objects, it is only natural to want to express properties of entire collections of objects, instead of enumerating the objects by name.
  - All students in our BaJian class are smart
- Quantifiers let us do this.
- First-order logic contains two standard quantifiers
  - universal and existential.

Universal quantification ∀

- All kings are persons

$$\forall x \;\; King(x) \Rightarrow Person(x).$$

- ∀x P: for all x, P is true, where P is a logical sentence
  - For all x, if x is a king, x is a person.
- Variable x
  - A variable is a term by itself

Model in FOL: 5 objects

$x \rightarrow$ Richard the Lionheart,
$x \rightarrow$ King John,
$x \rightarrow$ Richard's left leg,
$x \rightarrow$ John's left leg,
$x \rightarrow$ the crown.

Richard the Lionheart is a king $\Rightarrow$ Richard the Lionheart is a person.
King John is a king $\Rightarrow$ King John is a person.
Richard's left leg is a king $\Rightarrow$ Richard's left leg is a person.
John's left leg is a king $\Rightarrow$ John's left leg is a person.
The crown is a king $\Rightarrow$ the crown is a person.

Universal quantification ∀

$$\forall x\ King(x) \Rightarrow Person(x).$$

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*

Truth tables for the five logical connectives. To use the table to compute, for example, the value of $P \vee Q$ when P is true and Q is false, first look on the left for the row where P is *true* and Q is *false* (the third row). Then look in that row under the $P \vee Q$ column to see the result: *true*.

Model in FOL: 5 objects

$x \rightarrow$ Richard the Lionheart,

$x \rightarrow$ King John,

$x \rightarrow$ Richard's left leg,

$x \rightarrow$ John's left leg,

$x \rightarrow$ the crown.

P	Q	P=>Q
F	T	**T**
T	T	**T**
F	F	**T**
F	F	**T**
F	F	**T**

Richard the Lionheart is a king $\Rightarrow$ Richard the Lionheart is a person.

King John is a king $\Rightarrow$ King John is a person.

Richard's left leg is a king $\Rightarrow$ Richard's left leg is a person.

John's left leg is a king $\Rightarrow$ John's left leg is a person.

The crown is a king $\Rightarrow$ the crown is a person.

We end up asserting the conclusion of the rule just for those objects for which the premise is true and saying nothing at all about those objects for which the premise is false.

A common mistake:
Use conjunction instead of implication

$\forall x \;\; King(x) \wedge Person(x)$

Model in FOL: 5 objects

$x \rightarrow$ Richard the Lionheart,

$x \rightarrow$ King John,

$x \rightarrow$ Richard's left leg,

$x \rightarrow$ John's left leg,

$x \rightarrow$ the crown.

Richard the Lionheart is a king$\wedge$ Richard the Lionheart is a person,

King John is a king$\wedge$ King John is a person,

Richard's left leg is a king$\wedge$ Richard's left leg is a person,

P	Q	P ∧ Q	P=>Q
F	T	F	**T**
T	T	T	**T**
F	F	F	**T**

Mini quiz: All students in our BaJian class of ECNU are smart.
Write in FOL.

Existential Quantification ∃

- ∃x P
    - P is true for at least one object x
    - There exists an x such that, ⋯
    - For some x, ⋯
- King john has a crown on his head

$$\exists x \;\; Crown(x) \wedge OnHead(x, John) \, .$$

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.

Model in FOL: 5 objects

$x \rightarrow$ Richard the Lionheart,
$x \rightarrow$ King John,
$x \rightarrow$ Richard's left leg,
$x \rightarrow$ John's left leg,
$x \rightarrow$ the crown.

P	Q	P∧Q
F	F	F
F	F	F
F	F	F
F	F	F
T	T	T

$\forall$ is often used with =>
$\exists$ is often used with $\wedge$

Some notes

Just as $\Rightarrow$ appears to be the natural connective to use with $\forall$, $\wedge$ is the natural connective to use with $\exists$. Using $\wedge$ as the main connective with $\forall$ led to an overly strong statement in the example in the previous section; using $\Rightarrow$ with $\exists$ usually leads to a very weak statement, indeed. Consider the following sentence:

$$\exists x \;\; Crown(x) \Rightarrow OnHead(x, John).$$

Richard the Lionheart is a crown $\Rightarrow$ Richard the Lionheart is on John's head;
King John is a crown $\Rightarrow$ King John is on John's head;
Richard's left leg is a crown $\Rightarrow$ Richard's left leg is on John's head;

P	Q	P=>Q
F	F	T
F	F	T
F	F	T

An existentially quantified implication sentence is true whenever any object fails to satisfy the premise; hence such sentences really do not say much at all.

Nested Quantifiers: multiple quantifiers

- Brothers are siblings $\forall x\ \forall y\ Brother(x,y) \Rightarrow Sibling(x,y)$.

- Siblinghood is a symmetric relationship $\forall x,y\ Sibling(x,y) \Leftrightarrow Sibling(y,x)$.

- Some confusions when two quantifiers are used with the same variable name:

$$\forall x\ (Crown(x) \lor (\exists x\ Brother(Richard,x))) .$$

The rule is that the variable belongs to the innermost quantifier that mentions it; then it will not be subject to any other quantification.

$$\forall x\ (Crown(x) \lor (\exists z\ Brother(Richard,z))) .$$

∃ x can be replaced by ∃ z

Nested Quantifiers: multiple quantifiers

- Everybody loves somebody $\forall x\ \exists y\ Loves(x,y)$.

- There is someone who is loved by everyone $\exists y\ \forall x\ Loves(x,y)$.

The order of quantification is therefore very important. It becomes clearer if we insert parentheses. $\forall x\ (\exists y\ Loves(x,y))$ says that *everyone* has a particular property, namely, the property that they love someone. On the other hand, $\exists y\ (\forall x\ Loves(x,y))$ says that *someone* in the world has a particular property, namely the property of being loved by everybody.

Mini quiz

Lincoln Quote

You can fool all the people some of the time,
and some of the people all the time,
but you cannot fool all the people all the time.

$$\forall p \; \exists t \;\; Fool(p,t)$$
$$\wedge$$
$$\exists p \; \forall t \;\; Fool(p,t)$$
$$\wedge$$
$$\neg \; \forall p \; \forall t \;\; Fool(p,t)$$

Connections between ∀ and ∃

- Two quantifiers are connected with each other, through negation.

$$\forall x \;\; Likes(x, IceCream) \qquad \neg\exists x \;\; \neg Likes(x, IceCream)$$

$$\exists x \;\; Likes(x, Broccoli) \qquad \neg\forall x \;\; \neg Likes(x, Broccoli)$$

- Because ∀ is really a conjunction over the universe of objects and ∃ is a disjunction, it should not be surprising that they obey De Morgan's rules.

$$\begin{array}{llllll}
\neg\exists x & P \equiv \forall x & \neg P & \neg(P \vee Q) & \equiv & \neg P \wedge \neg Q \\
\neg\forall x & P \equiv \exists x & \neg P & \neg(P \wedge Q) & \equiv & \neg P \vee \neg Q \\
\forall x & P \equiv \neg\exists x & \neg P & P \wedge Q & \equiv & \neg(\neg P \vee \neg Q) \\
\exists x & P \equiv \neg\forall x & \neg P & P \vee Q & \equiv & \neg(\neg P \wedge \neg Q)\,.
\end{array}$$

Equality

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

$$Father(John) = Henry$$

- Different from logical equivalence

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

Negation

- Equality can also be used with **negation** to insist that two terms are not the same object.
  - E.g., to say that Richard has at least two brothers

$$\exists\, x,y \;\; Brother(x, Richard) \wedge Brother(y, Richard) \wedge \neg(x = y)\,.$$

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic, instantiation and propositionalization
  - Unification
  - Forward chaining, backward chaining, resolution

Database semantics

$$Brother(John, Richard) \land Brother(Geoffrey, Richard)\ ,$$

Standard FOL semantics

$$Brother(John, Richard) \land Brother(Geoffrey, Richard) \land John \neq Geoffrey$$
$$\land \forall x\ Brother(x, Richard) \Rightarrow (x = John \lor x = Geoffrey)$$

Database semantics

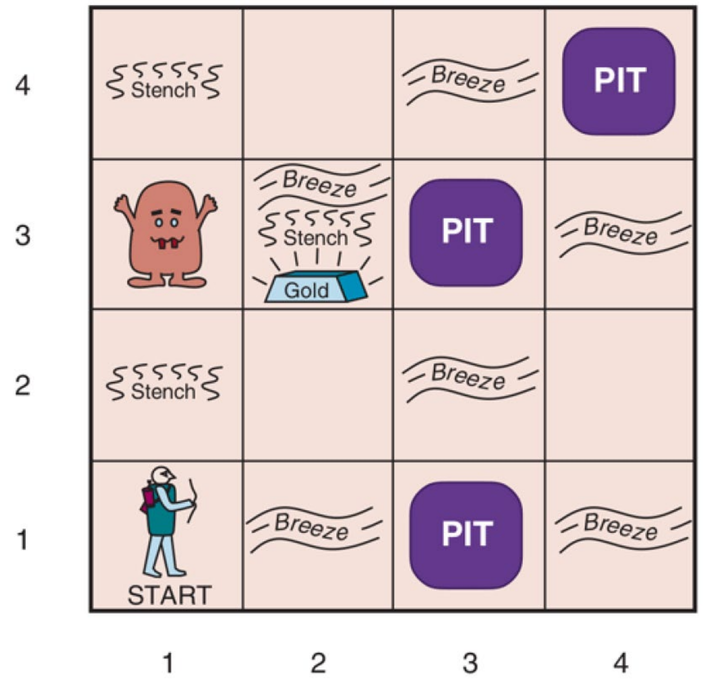
- Suppose that we believe that Richard has two brothers, John and Geoffrey.
  - First, this assertion is true in a model where Richard has only one brother—we need to add John≠Geoffrey.
    - Refer back to “Symbols and Interpretation”
  - Second, the sentence doesn’t rule out models in which Richard has many more brothers besides John and Geoffrey.
- Database semantics vs standard semantics of FOL
  - Unique-names assumption
    - Every constant symbol refer to a distinct object
  - Closed world assumption
    - Atomic sentences not known to be true are in fact false

The Wumpus world

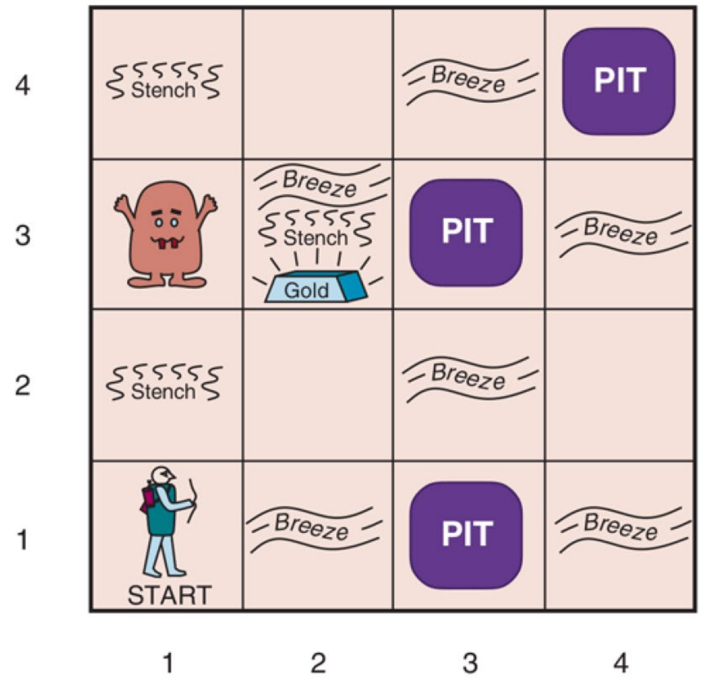
- Perception
  - Percept([Stench, Breeze, Glitter, none, none], 5) .
  - 5 is a timestamp, when the percept occurred.

$$\forall t,s,g,w,c \quad Percept([s,Breeze,g,w,c],t) \Rightarrow Breeze(t)$$
$$\forall t,s,g,w,c \quad Percept([s,None,g,w,c],t) \Rightarrow \neg Breeze(t)$$
$$\forall t,s,b,w,c \quad Percept([s,b,Glitter,w,c],t) \Rightarrow Glitter(t)$$
$$\forall t,s,b,w,c \quad Percept([s,b,None,w,c],t) \Rightarrow \neg Glitter(t)$$

- Simple reflex behavior $\forall t \;\; Glitter(t) \Rightarrow BestAction(Grab,t)$.



The Wumpus world



- The environment
- Objects
  - Squares, pits, the Wumpus
- A square: a list term, e.g., [1, 2]

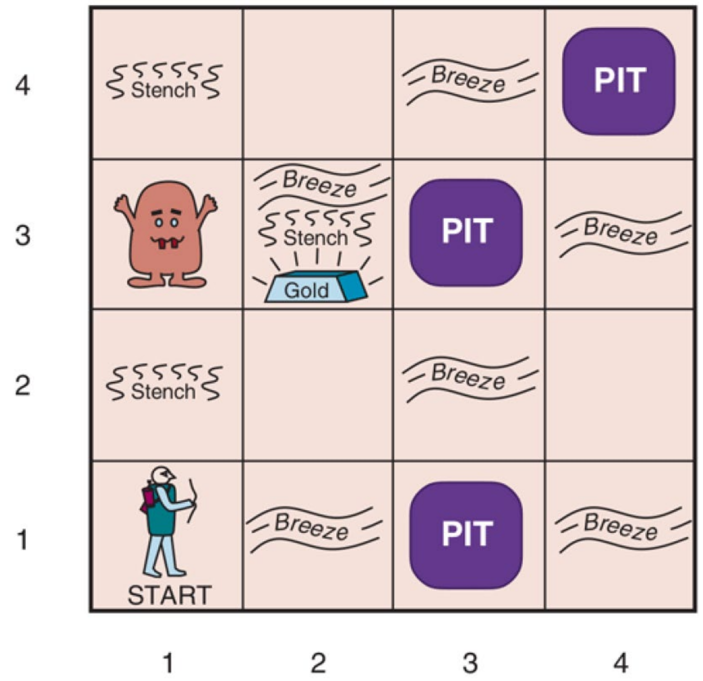
$$\forall x,y,a,b\; Adjacent([x,y],[a,b]) \Leftrightarrow (x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1)).$$

- Use a unary predicate Pit that is true of squares containing pits.
- A constant Wumpus and is fixed to a specific location

$$\forall t\; At(Wumpus,[1,3],t).$$

The Wumpus world

- Perceive Breezy means at least one of the adjacent squares has a pit.



$$R_2 : \quad B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}) .$$
$$R_3 : \quad B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1}) .$$
$$\vdots$$

$$\forall s \;\; Breezy(s) \Leftrightarrow \exists r \;\; Adjacent(r,s) \wedge Pit(r) .$$

Propositional Logic

First-order Logic

Concisely describe an environment with many objects

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic, instantiation and propositionalization
  - Unification
  - Forward chaining, backward chaining, resolution

Propositional vs. first-order inference

- Covert the first-order knowledge base to propositional logic and use propositional inference.
- A first step is eliminating universal quantifiers.
  - SUBST({v/g}, α): apply the substitution of v by g
- Universal Instantiation (UI)
  - Replace a universally quantified variable *v* by a ground term *g* (a term without variables)

$$\forall x \;\; King(x) \land Greedy(x) \Rightarrow Evil(x).$$

$\{x/John\}$, $\{x/Richard\}$, and $\{x/Father(John)\}$

$$\frac{\forall v \;\; \alpha}{\textsc{Subst}(\{v/g\},\alpha)}$$

$$\begin{aligned}
&King(John) \land Greedy(John) \Rightarrow Evil(John)\\
&King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard)\\
&King(Father(John)) \land Greedy(Father(John)) \Rightarrow Evil(Father(John)).\\
&\vdots
\end{aligned}$$

Propositional vs. first-order inference

- Existential instantiation
  - Replace an existentially quantified variable with a single **new** constant symbol, called **Skolem constant**
  - C_1 does not appear elsewhere in the knowledge base.

$$\frac{\exists v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}.$$

$$\exists x \;\; Crown(x) \land OnHead(x, John)$$

$$Crown(C_1) \land OnHead(C_1, John)$$

Reduction to propositional inference

Our KB:

$$\forall x \;\; King(x) \land Greedy(x) \Rightarrow Evil(x)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John).$$

and that the only objects are *John* and *Richard*. We apply UI to the first sentence using all possible substitutions, $\{x/John\}$ and $\{x/Richard\}$. We obtain

$$King(John) \land Greedy(John) \Rightarrow Evil(John)$$
$$King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard).$$

Propositionalization

Proposition symbols in propositional logic are facts

- $B_{1,1}$, true or false

Next replace ground atomic sentences, such as $King(John)$, with proposition symbols, such as $JohnIsKing$. Finally, apply any of the complete propositional algorithms in Chapter 7 to obtain conclusions such as $JohnIsEvil$, which is equivalent to $Evil(John)$.

Instantiation

- Universal Instantiation
  - can be applied several times to add new sentences
  - the new KB is logically equivalent to the old KB
- Existential Instantiation
  - can be applied once to replace the existential sentence
  - the new KB is not equivalent to the old
  - but is satisfiable iff the old KB was satisfiable

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic , instantiation and propositionalization
  - Unification
  - Forward chaining, backward chaining, resolution

Unification and First-order Inference

- Generalized Modus Ponens
  - For **atomic** sentences p_i, p_i', and q, where there is a substitution such that

$$\text{SUBST}(\theta, p_i') = \text{SUBST}(\theta, p_i)$$

for all i, then

$$\frac{p_1',\ p_2',\ \ldots,\ p_n',\ (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}.$$

$$\frac{\alpha \Rightarrow \beta,\quad \alpha}{\beta}$$

$$King(x) \wedge Greedy(x) \implies Evil(x)$$

p_1' is $King(John)$ $\quad$ p_1 is $King(x)$

p_2' is $Greedy(y)$ $\quad$ p_2 is $Greedy(x)$

θ is $\{x/John, y/John\}$ $\quad$ q is $Evil(x)$

$\text{SUBST}(\theta, q)$ is $Evil(John)$.

Lifted version

- Generalized Modus Ponens is a lifted version of Modus Ponens—it raises Modus Ponens from ground (variable-free) propositional logic to first-order logic.
- We can develop lifted versions of the forward chaining, backward chaining, and resolution algorithms.
- The key advantage of lifted inference rules over **propositionalization** is that they make only those substitutions that are required to allow particular inferences to proceed.
- Lifted inference rules require finding substitutions that make different logical expressions look identical. This process is called **unification** and is a key component of all first-order inference algorithms.

Unification

- The UNIFY algorithm takes two sentences, p and q, and returns a unifier for them (a substitution) if one exists:

$$\textsc{Unify}(p,q) = \theta \text{ where } \textsc{Subst}(\theta,p) = \textsc{Subst}(\theta,q).$$

- Query: whom does John know?
  - AskVars(Knows(John, x))
  - Answers to this query can be found by finding all sentences in the knowledge base that unify with Knows(John, x)

$$\textsc{Unify}\ (Knows(John,x),\ Knows(John,Jane)) = \{x/Jane\}$$
$$\textsc{Unify}\ (Knows(John,x),\ Knows(y,Bill)) = \{x/Bill, y/John\}$$
$$\textsc{Unify}\ (Knows(John,x),\ Knows(y,Mother(y))) = \{y/John, x/Mother(John)\}$$
$$\textsc{Unify}\ (Knows(John,x),\ Knows(x,Elizabeth)) = \ failure.$$

Outline

- Pros and cons of propositional logic
- First-order logic (FOL) example
- Syntax and semantics of FOL
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic , instantiation and propositionalization
  - Unification
  - Forward chaining, backward chaining, resolution

Lifted Forward Chaining

- Forward Chaining only works for horn clauses
  - A horn clause is a disjunction of literals of which at most one is positive.
  - Or it is in the form of
    - proposition symbol; or
    - (conjunction of symbols) => symbol

$$C, \qquad B \implies A, \qquad C \wedge D \implies B$$

- Lifted forward chaining only works for **first-order definite clauses**
  - A definite clause is either atomic, or is an implication whose antecedent is a conjunction of positive literals and whose consequent is a single positive literal.
  - Existential quantifiers are not allowed
  - Universal quantifiers are always there and are implicit. When you see x, it means ∀x

$$King(x) \wedge Greedy(x) \Rightarrow Evil(x)\ ,$$

Example

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

Prove that West is a criminal

KB:

... it is a crime for an American to sell weapons to hostile nations:
$American(x) \land Weapon(y) \land Sells(x,y,z) \land Hostile(z) \implies Criminal(x)$

Nono ... has some missiles, i.e., $\exists x\ Owns(Nono,x) \land Missile(x)$:
$Owns(Nono, M_1)$ and $Missile(M_1)$

... all of its missiles were sold to it by Colonel West
$Missile(x) \land Owns(Nono,x) \implies Sells(West,x,Nono)$

Missiles are weapons:
$Missile(x) \Rightarrow Weapon(x)$

An enemy of America counts as "hostile":
$Enemy(x, America) \implies Hostile(x)$

West, who is American ...
$American(West)$

The country Nono, an enemy of America ...
$Enemy(Nono, America)$

. . . it is a crime for an American to sell weapons to hostile nations:
$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \implies Criminal(x)$

Nono . . . has some missiles, i.e., $\exists\, x\ Owns(Nono, x) \wedge Missile(x)$:
$Owns(Nono, M_1)$ and $Missile(M_1)$

. . . all of its missiles were sold to it by Colonel West
$Missile(x) \wedge Owns(Nono, x) \implies Sells(West, x, Nono)$

Missiles are weapons:
$Missile(x) \Rightarrow Weapon(x)$

An enemy of America counts as "hostile":
$Enemy(x, America) \implies Hostile(x)$

West, who is American . . .
$American(West)$

The country Nono, an enemy of America . . .
$Enemy(Nono, America)$

Starting from the known facts, it triggers all the rules whose premises are satisfied, adding their conclusions to the known facts.

. . . it is a crime for an American to sell weapons to hostile nations:

$$American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \implies Criminal(x)$$

Nono . . . has some missiles, i.e., $\exists x\ Owns(Nono,x) \wedge Missile(x)$:

$Owns(Nono, M_1)$ and $Missile(M_1)$

. . . all of its missiles were sold to it by Colonel West

$$\boxed{Missile(x) \wedge Owns(Nono,x) \implies Sells(West,x,Nono)}$$

Missiles are weapons:

$$\boxed{Missile(x) \Rightarrow Weapon(x)}$$

An enemy of America counts as "hostile":

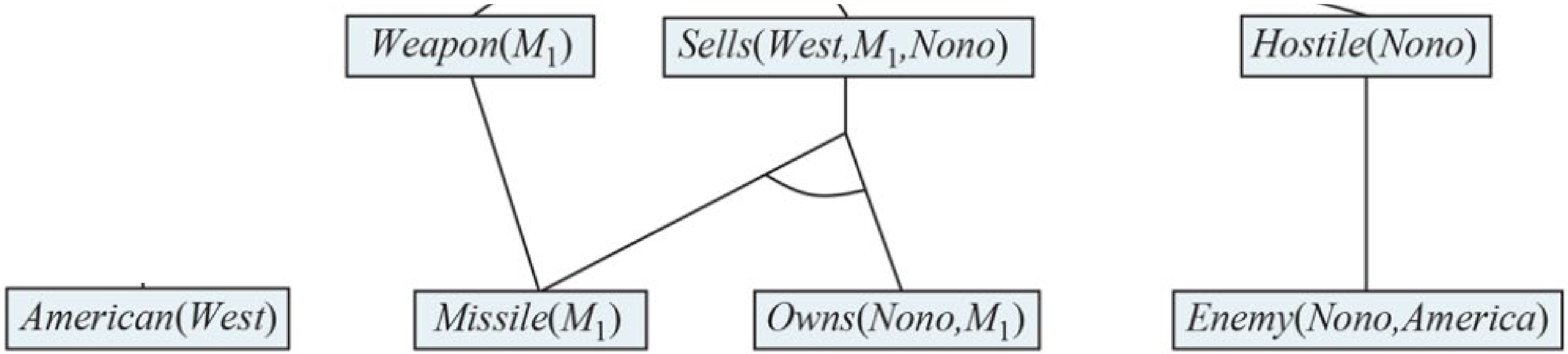
$$\boxed{Enemy(x, America) \implies Hostile(x)}$$

West, who is American . . .

$American(West)$

The country Nono, an enemy of America . . .

$Enemy(Nono, America)$



. . . it is a crime for an American to sell weapons to hostile nations:

$$\boxed{American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \implies Criminal(x)}$$

Nono . . . has some missiles, i.e., $\exists\, x\ Owns(Nono,x) \wedge Missile(x)$:

$Owns(Nono,M_1)$ and $Missile(M_1)$

. . . all of its missiles were sold to it by Colonel West

$Missile(x) \wedge Owns(Nono,x) \implies Sells(West,x,Nono)$

Missiles are weapons:

$Missile(x) \Rightarrow Weapon(x)$

An enemy of America counts as "hostile":

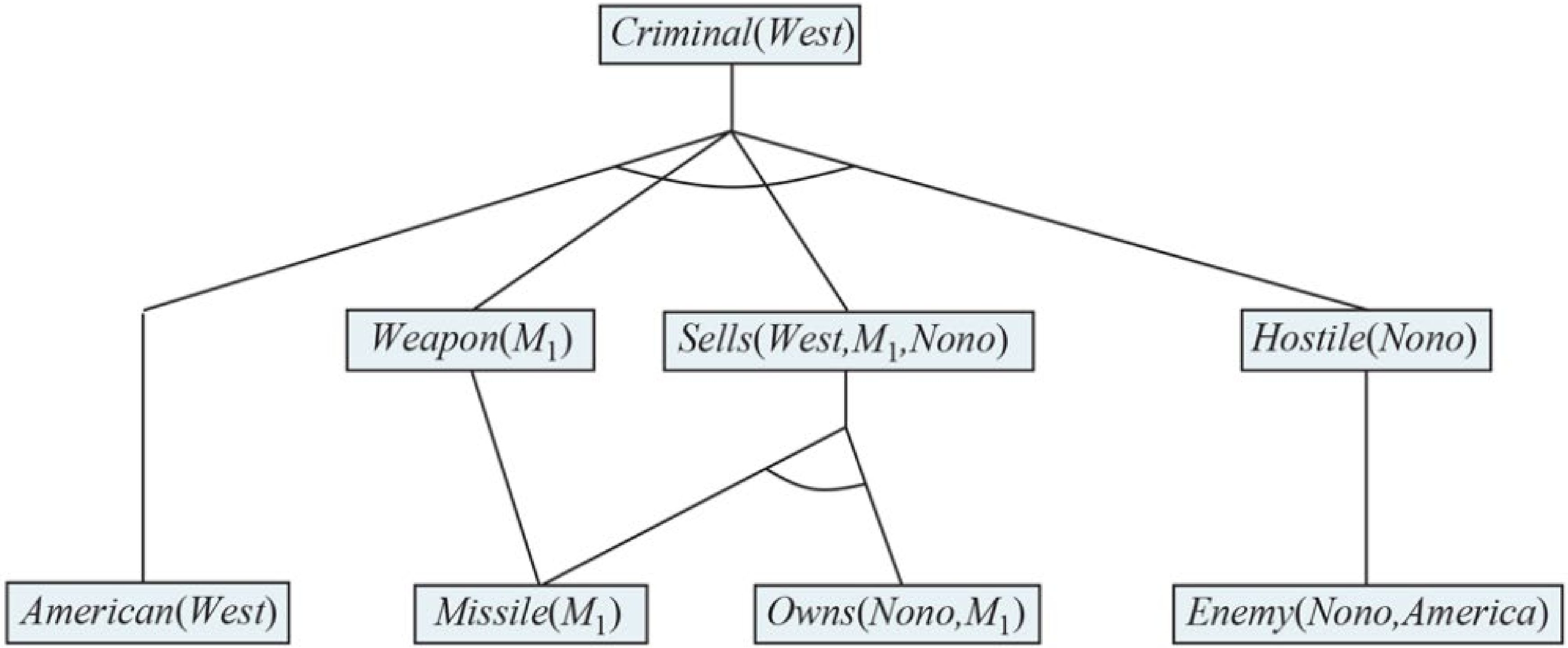
$Enemy(x,America) \implies Hostile(x)$

West, who is American . .

$American(West)$

The country Nono, an ene

$Enemy(Nono, Americ$



Lifted Backward chaining

- Start with query
- Check if it can be derived by given rules and facts
  - apply rules that infer the query
  - recurse over pre-conditions

... it is a crime for an American to sell weapons to hostile nations:

$American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \implies Criminal(x)$

Nono ... has some missiles, i.e., $\exists x\ Owns(Nono, x) \land Missile(x)$:

$Owns(Nono, M_1)$ and $Missile(M_1)$

... all of its missiles were sold to it by Colonel West

$Missile(x) \land Owns(Nono, x) \implies Sells(West, x, Nono)$

Missiles are weapons:

$Missile(x) \Rightarrow Weapon(x)$

An enemy of America counts as "hostile":

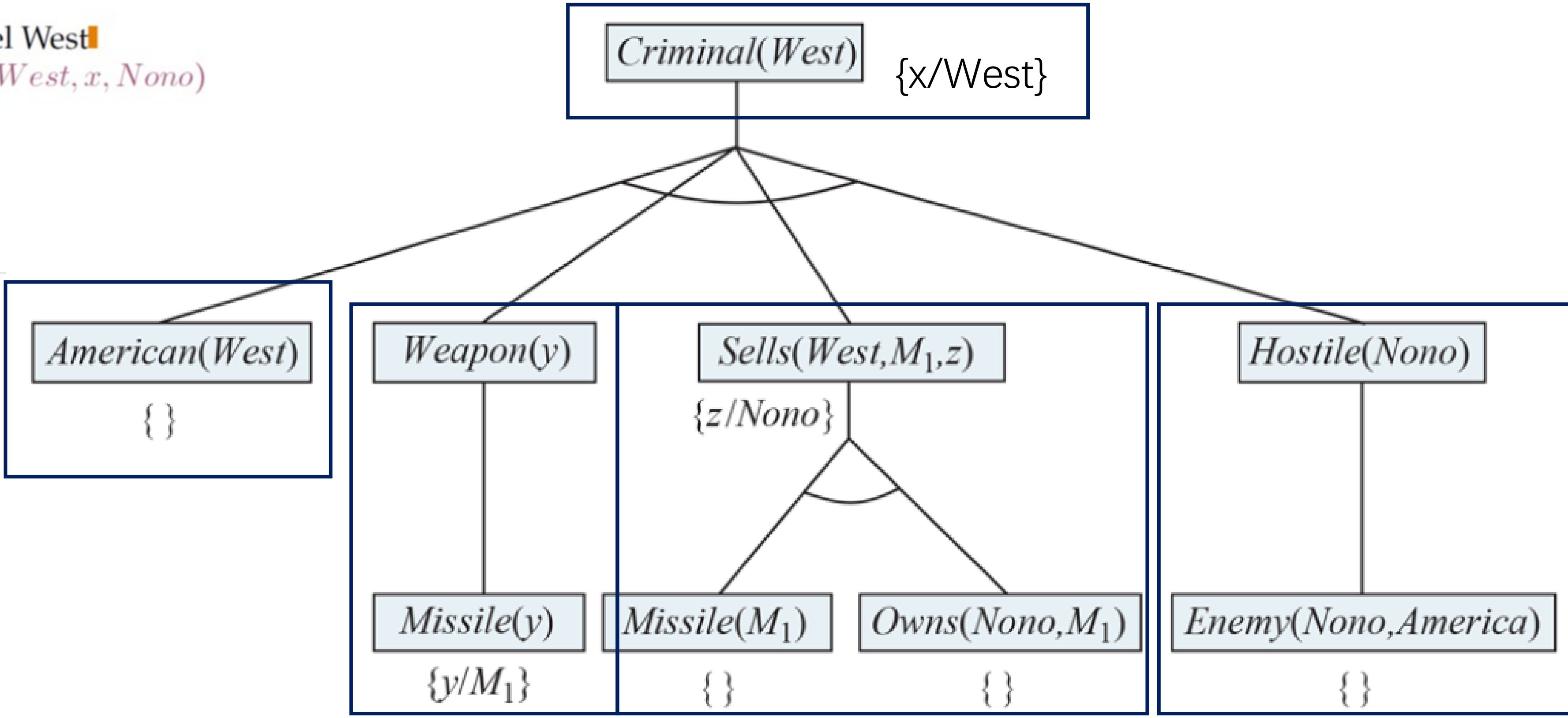
$Enemy(x, America) \implies Hostile(x)$

West, who is American ...

$American(West)$

The country Nono, an enemy of America ...

$Enemy(Nono, America)$



Proof tree constructed by backward chaining to prove that West is a criminal. The tree should be read depth first, left to right. To prove *Criminal(West)*, we have to prove the four conjuncts below it. Some of these are in the knowledge base, and others require further backward chaining. Bindings for each successful unification are shown next to the corresponding subgoal. Note that once one subgoal in a conjunction succeeds, its substitution is applied to subsequent subgoals. Thus, by the time FOL-BC-Ask gets to the last conjunct, originally *Hostile*(z), z is already bound to *Nono*.

Resolution in propositional logic

- Resolution inference

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

  - l_i and m_j are complementary literals
- Example

$$\frac{P_{1,3} \vee P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}} \qquad\qquad \frac{P_{1,1} \vee P_{3,1}, \qquad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}\,.$$

- Resolution is sound and complete for propositional logic

Resolution in first-order logic

- Resolution inference

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\textsc{Subst}(\theta, \ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)}$$

  - Unify(l_i, $\neg m_j$)=Θ

$$\frac{\neg Rich(x) \vee Unhappy(x) \qquad Rich(Ken)}{Unhappy(Ken)}$$

$$\theta = \{x/Ken\}$$

Lecture 8 ILOs

FOL:
- Objects, Terms
- Relations, Predicates
- Concise, Quantifiers

Propositional Logic

Propositionalization

Instantiation
- Universal instantiation
- Existential instantiation

- Limitations of propositional logic
- First-order logic (FOL)
- Syntax and Semantics
  - Models, symbols, interpretations, terms
  - Atomic and complex sentences
  - Quantifiers, universal and existential quantifiers, equality
  - Database semantics
- Inference
  - Reduce FOL to propositional logic, instantiation and propositionalization
  - Unification
  - Forward chaining, backward chaining, resolution