ÉCOLE
POLYTECHNIQUE
DE LOUVAIN

LINFO1361

Intelligence Artificielle

First-Order Logic

Éric Piette

18 April 2024

AIMA Ch. 8
[Artificial Intelligence, Modern Approach.
Russel & Norvig, 4th ed. 2021]

3 weeks ago…. Propositional logic

- Do you remember this?

$$(\neg D \lor \neg B \lor C)$$
$$\land (B \lor \neg A \lor \neg C)$$
$$\land (\neg C \lor \neg B \lor E)$$
$$\land (E \lor \neg D \lor B)$$
$$\land (B \lor E \lor \neg C)$$

- **Propositional logic**! Wumpus, CNF, horn clauses,

Sound and complete, Inference, etc.

Propositional logic

- Declarative
  - Relationships between variables are described through sentences
  - A method for propagating relationships

- Expressive
  - Can represent partial information using disjunction

- Compositional
  - If A means “*It is raining*” and B means “*I like beer*”,
    $A \wedge B$ means “*It is raining and I like beer* ”

- But lacks expressive power to describe the environment concisely
  - E.g. cannot say *"pits cause breezes in adjacent squares “*

Procedural Representation

- Representing knowledge with data structure
    - World[2,2] := Pit

- Limitations
    - How to derive new information from this representation ?
    - How to state that there is a pit in [2,2] OR in [3,1]?
    - How to state that if the Wumpus is in [2,2] then it is not in [3,1]?

- Need a declarative representation

First-Order Logic - basic blocks

- PL assumes the world contains facts

- First-order logic (like natural language) assumes the world contains:
  - **Objects**
    - *people, houses, numbers, colors, …*
  - **Relations**
    - Unary relation : properties of objects
    - N-ary : relations between objects
    - *brother, bigger_than, inside, …*
  - **Functions**
    - *father_of, successor, cons*

Different logics

- **Propositional logic**
  - facts
  - true, false, unknown

- **First-order logic**
  - facts, objects, *relations*
  - true, false, unknown

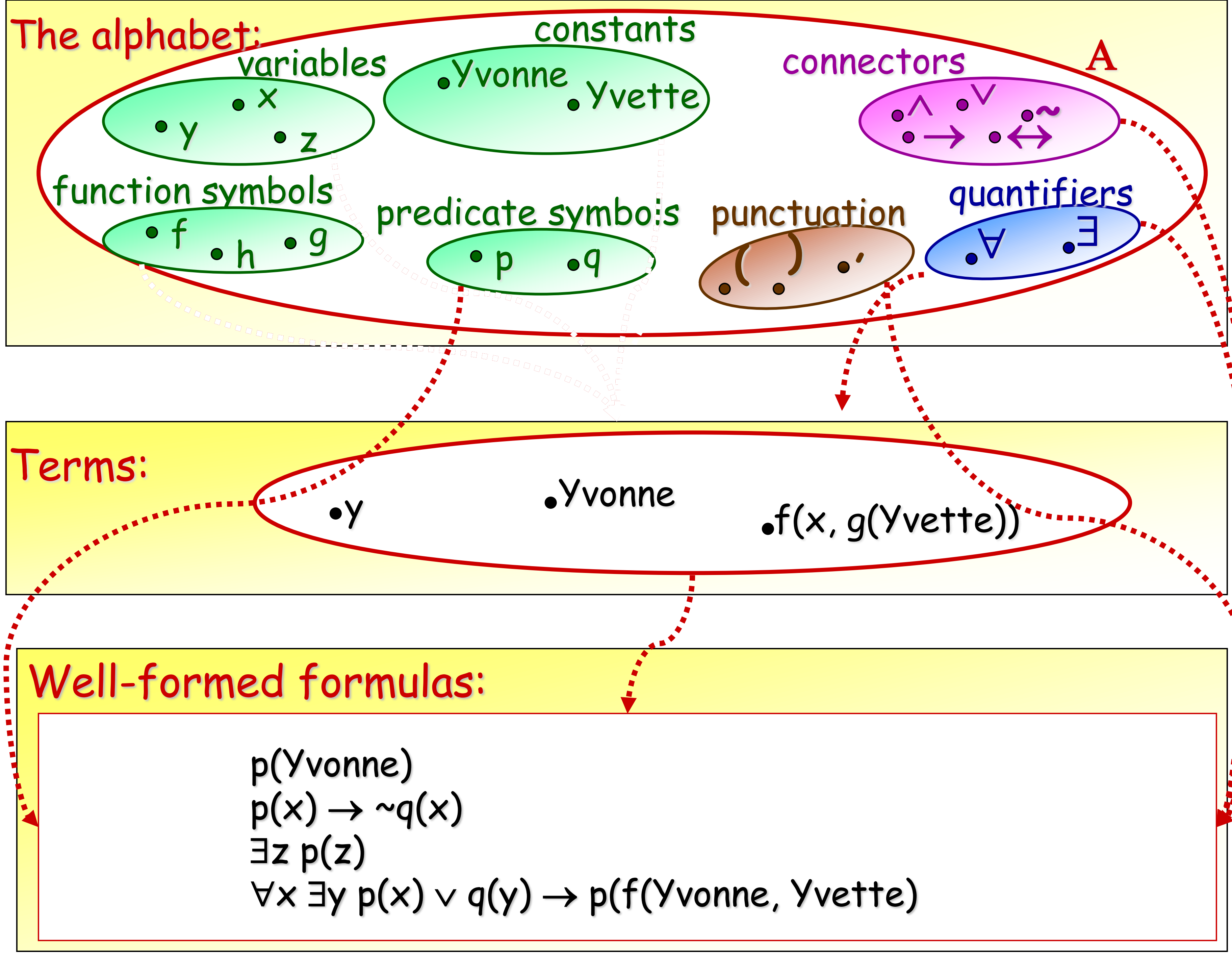
- **Temporal logic**
  - facts, objects, relations, *times*
  - true, false, unknown

- **Probability theory**
  - facts
  - *Degree of belief*

- **Fuzzy logic**
  - facts with degree of truth
  - *Interval value*

- **Higher-order logic**
  - Relations and functions are also objects

- …



The alphabet:

Terms:

Well-formed formulas:

$p(\text{Yvonne})$

$p(x) \rightarrow {\sim}q(x)$

$\exists z\ p(z)$

$\forall x\ \exists y\ p(x) \vee q(y) \rightarrow p(f(\text{Yvonne}, \text{Yvette}))$

Example

- Alphabet:

  { {0}, {x,y}, {s}, {odd,even}, Con, Pun, Quan}

- Terms:

  { 0, s(0), s(s(0)), s(s(s(0))), …
  x, s(x), s(s(x)), s(s(s(x))), …
  y, s(y), s(s(y)), s(s(s(y))), … }

- Well-formed formulas:

  odd(0), even(s(0)), …
  odd(x), odd(s(y)), …
  odd(x) ← even(s(s(x))), …
  ∀x (odd(x) ← even(s(x))), …
  odd(y) ← ∀x (even(s(x))), ...

Syntax

- An ***alphabet*** consists of variables, constants, function symbols, predicate symbols (all user-defined) and of connectors, punctuations and quantifiers

- ***Terms*** are either:
  - variables
  - constants
  - function symbols provided with as many terms as arguments, as the function symbol expects

- ***Well-formed formulas*** are constructed from predicate symbols, provided with terms as arguments, and from connectors, quantifiers and punctuation – according to the rules of the connectors

Syntax

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

Examples

- Brother(Richard, John)
- Married(Father(Richard), Mother(John))
- King(Richard) $\lor$ King(John)
- $\forall$ x King(x) $\Rightarrow$ Person(x)
- $\exists$ x Crown(x) $\Rightarrow$ OnHead(x)
- $\forall$ s Breezy(s) $\Leftrightarrow$ $\exists$ r Adjacent(r, s) $\land$ Pit(r)

Semantics

- Provided by **interpretations** for the basic constructs
  - usually suggested by meaningful names

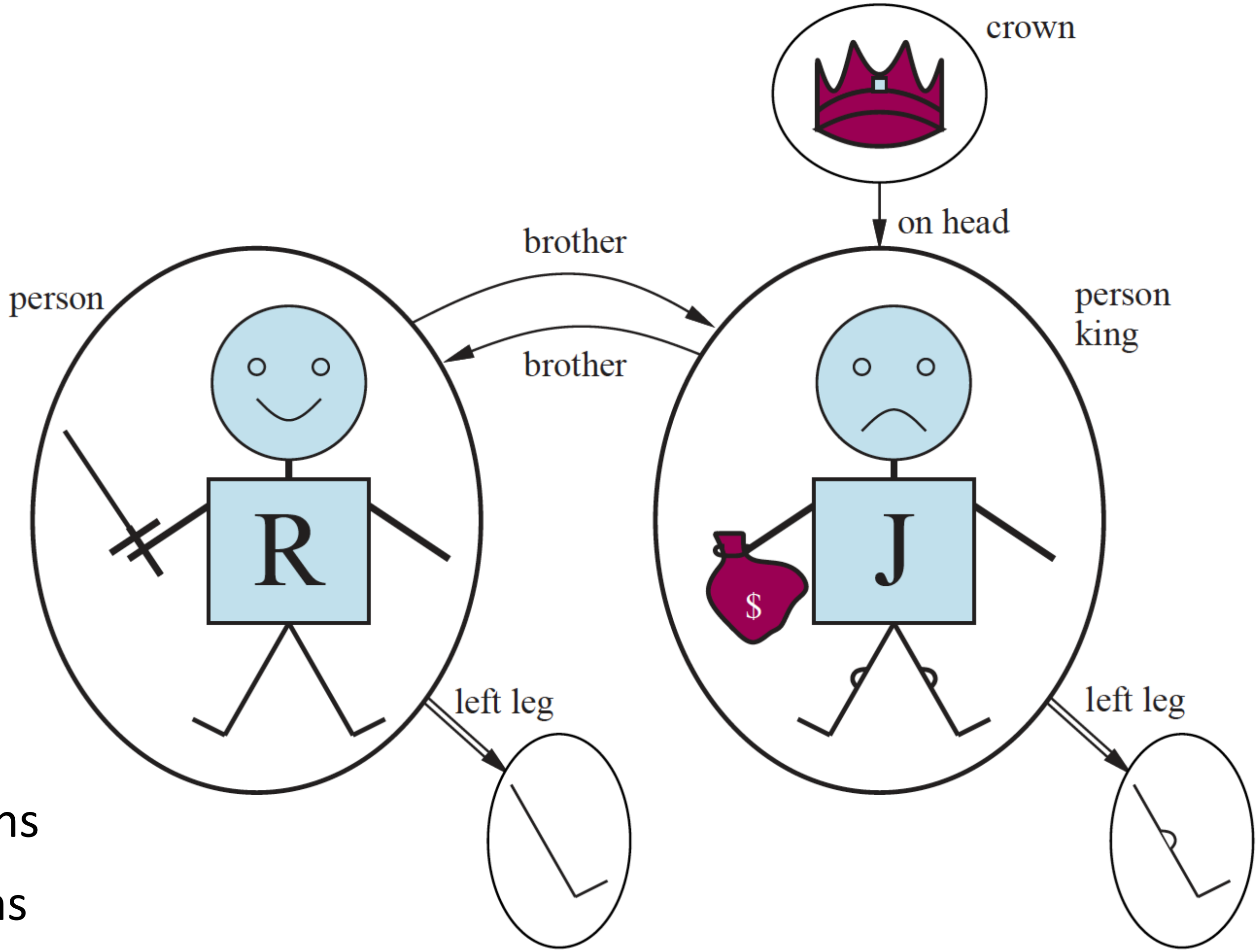
- **Domain** of the interpretation: set of objects
- Constants
  - the interpretation identifies the object in the real world

- Predicate symbols
  - the interpretation specifies the particular relation in the domain
  - Can be defined implicitly, or explicitly through the set of tuples of objects that satisfy the relation

- Function symbols
  - identifies the object referred to by a tuple of objects
  - Can be defined implicitly, or explicitly through tables

Interpretation

- Brother(Richard, John)
- King(Richard) ∨ King(John)
- ∀ x King(x) ⇒ Person(x)
- ∃ x Crown(x) ⇒ OnHead(x)
- ∀ x Person(x) ⇒ leftLeg(x)



5 objects

2 binary relations

3 unary relations

1 unary function

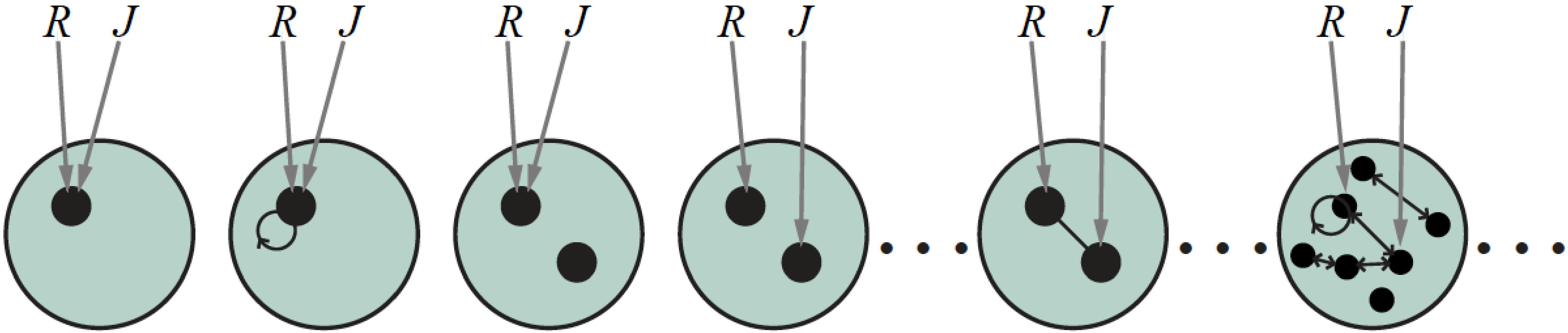
Interpretation

- a set D (the **domain**)
- A (total) function that maps constants to D
- A (total) function that maps function symbols to (total) functions: D → D
- A (total) function that maps predicate symbols to

predicates: D → Booleans

Interpretations

- Examples of interpretations
  - 2 constant symbols (R and J)
  - One binary relation



Intended Interpretation

- Usually, one has a specific interpretation in mind when writing sentences
  - *Richard* refers to Richard the Lionhearty and *John* refers to the evil King John
  - *Brother* refers to the brotherwood relation; *OnHead* is a relation that holds between the crown and King John; *Person*, *King* and *Crown* are unary relations that identify persons, kings, and crown
  - *LeftLeg* refers to the left leg function

- Beware sentences can be interpreted according to any possible interpretation

Terms

- A **term** is a logical expression refering to an object in the (domain of the) interpretation
  - Eg. John, LeftLeg(John)

- Let I be an interpretation
  t_1, ..., t_n be terms and f be a n-ary function symbol
  - f(t_1, ..., t_n) is a term
  - This term refers to the object F(T_1, ..., T_n) in the domain of I
  - where F is the function $\mathbf{D^n \rightarrow D}$ associated to f in I
    and T_1, ..., T_n are the interpretation of the terms t_1, ..., t_n in I

Atomic sentence

- An atomic sentence state a fact (true or false)
- An atomic sentence is composed of a predicate with possible arguments (terms)
    - Brother(Richard, John)
    - Married(Father(Richard, Mother(John))
- An atomic sentence is true or false in an interpretation
- Let I be an interpretation
  t_1, ..., t_n be terms and p be a n-ary predicate symbol
    - $p(t_1, ..., t_n)$ is an atomic formula
    - The formula is true in I if the relation $P(T_1, ...,T_n)$ holds
    - where P is the predicate **$D^n \rightarrow$ Boolean** associated to P in I and T_1, ...,T_n are the interpretation of the terms t_1, ..., t_n in I

Complex sentences

- Built by combining atomic sentences using logical connectives (¬, ∧, ∨, ⇒, ⇔) and parentheses
  - E.g. ¬ Brother(LetfLeg(Richard), John)
  - Brother(Richard, John) ∧ Brother(John, Richard)
  - King(Richard) ∨ King(John)

- A complex sentence is true or false in a given interpretation

- Combinaison of truth values of the atomic sentences (see truth tables of logical connectives in propositional logic)

Quantifiers

- Here comes the power of First-Order Logic

- Can be used to express properties of collections of objects
  - eliminates the need to explicitly enumerate all objects…
  - in a conjunction: universal quantifier $\forall$
  - in a disjunction: existential quantifier $\exists$

Universal Quantifier

- States that a predicate *P* holds for all objects *x* in the domain
  - $\forall x$ King(x) $\Rightarrow$ Person(x)
  - $\forall x$ P(x)

- Given an interpretation I with domain **D**, the sentence is true if and only if all the individual sentences where the variable *x* is replaced by the individual objects in D are true in the given interpretation

Universal Quantifier

- Typically, $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$

- $\forall$ x King(x) $\Rightarrow$ Person(x)
- $\forall$ x King(x) $\wedge$ Person(x)

- What do they mean?

Universal Quantifier

- Typically, $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$

- Correct: $\forall$ x King(x) $\Rightarrow$ Person(x)
- Incorrect:$\forall$ x King(x) $\wedge$ Person(x)
  means “Everyone is a king and is a person”

Existential Quantification

- states that a predicate P holds for some object in the universe
  - $\exists$ x Crown(x) $\wedge$ OnHead(x, John)
  - $\exists$ x P(x)

- Given an interpretation I with domain **D**, the sentence is true if and only if there is at least one of the individual sentences where the variable *x* is replaced by the individual objects in D is true in the given interpretation

Existential Quantifier

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$

- $\exists$ x Crown(x) $\wedge$ OnHead(x, John)
- $\exists$ x Crown(x) $\Rightarrow$ OnHead(x, John)

- What do they mean?

Existential Quantifier

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$

- Correct: $\exists$ x Crown(x) $\wedge$ OnHead(x, John)
- Incorrect: $\exists$ x Crown(x) $\Rightarrow$ OnHead(x, John)
  is true as soon as some object is not a crown

Properties of quantifiers

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$

- $\exists x\ \forall y$ is **not** the same as $\forall y\ \exists x$
  - $\exists x\ \forall y$ Loves(x, y)
  - “???”
  - $\forall y\ \exists x$ Loves(x, y)
  - “???”

Properties of quantifiers

- ∀x ∀y is the same as ∀y ∀x
- ∃x ∃y is the same as ∃y ∃x

- ∃x ∀y is **not** the same as ∀y ∃x
  - ∃x ∀y Loves(x, y)
  - "There is a person who loves everyone in the world"
  - ∀y ∃x Loves(x, y)
  - "Everyone in the world is loved by at least one person"

Connection between quantifiers

- $\forall x\ P$ is equivalent to $\neg \exists x \neg P$
- $\exists x\ P$ is equivalent to $\neg \forall x \neg P$

- This generalises
  - $\neg(P \wedge Q)$ equivalent to $\neg P \vee \neg Q$
  - $\neg(P \vee Q)$ equivalent to $\neg P \wedge \neg Q$

Closed and Ground Formulas

- A ground term (or formula) is a term (or formula) without any variable
  - Brother(LetfLeg(Richard), John) is ground
  - OnHead(**x**, John) is not ground

- A closed formula is a formula where each occurrence of a variable x is in the scope of a quantifier ∀x or ∃ x
  - ∃ **x** Crown(**x**) ∧ OnHead(**x**, John) is closed

Logical Reasoning

- An interpretation where α is true is called a **model** of α
- Logical **entailment** between sentences α and β
- $\alpha \vDash \beta$ if and only if, in every interpretation in which α is true, β is also true

Validity and satisfiability

- **Validity**
  - a sentence that is true in ***all*** interpretations
- **Satisfiability**
  - a sentence that is true in ***some*** interpretations (i.e. it has a model)
- **Inconsistency, unsatifiability**
  - a sentence that is **false** in ***all*** interpretation (i.e. it has no model)
- α is valid iff $\neg\alpha$ is unsatisfiable
- α ♦ β iff $(\alpha \wedge \neg \beta)$ is unsatisfiable
  - Proof by refutation (contradiction)

The kinship example

- One’s mother is one’s parent who is a female
  - Give the first-order logic statement with:
    - Two variables m and c, three functions Mother, Female, and Parent

- A grandparent is a parent of one’s parent
  - Give the first-order logic statement with:
    - Two variables g and c, two functions Grandparent, and Parent

- Express the siblinghood is symmetric

The kinship example

- One’s mother is one’s parent who is a female
  - $\forall$c, m Mother(m, c) $\Leftrightarrow$ Parent(m, c) $\wedge$ Female(m)

- A grandparent is a parent of one’s parent
  - Give the first-order logic statement with:
    - Two variables g and c, two functions Grandparent, and Parent

- Express the siblinghood is symmetric

The kinship example

- One’s mother is one’s parent who is a female
  - $\forall c, m$ Mother(m, c) $\Leftrightarrow$ Parent(m, c) $\wedge$ Female(m)

- A grandparent is a parent of one’s parent
  - $\forall c, g$ Grandparent(g, c) $\Leftrightarrow$ $\exists p$ Parent(g, p) $\wedge$ Parent(p, c)

- Express the siblinghood is symmetric

The kinship example

- One’s mother is one’s parent who is a female
  - $\forall$c, m Mother(m, c) $\Leftrightarrow$ Parent(m, c) $\wedge$ Female(m)

- A grandparent is a parent of one’s parent
  - $\forall$c, g Grandparent(g, c) $\Leftrightarrow$ $\exists$p Parent(g, p) $\wedge$ Parent(p, c)

- A theorem: siblinghood is symmetric
  - $\forall$x, y Sibling(x, y) $\Leftrightarrow$ Sibling(y, x)

The number, set and list examples

First-Order Logic can be used to model

- natural numbers
- Sets and subsets
- Lists

Game Description Language (GDL)

- A general game description language in first-order logic

```
(role white) (role black)
(init (cell 1 1 b)) (init (cell 1 2 b)) (init (cell 1 3 b))
(init (cell 2 1 b)) (init (cell 2 2 b)) (init (cell 2 3 b))
(init (cell 3 1 b)) (init (cell 3 2 b)) (init (cell 3 3 b))
(init (control white))
(<= (legal ?w (mark ?x ?y)) (true (cell ?x ?y b))
    (true (control ?w)))
(<= (legal white noop) (true (control black)))
(<= (legal black noop) (true (control white)))
(<= (next (cell ?m ?n x)) (does white (mark ?m ?n))
    (true (cell ?m ?n b)))
(<= (next (cell ?m ?n o)) (does black (mark ?m ?n))
    (true (cell ?m ?n b)))
(<= (next (cell ?m ?n ?w)) (true (cell ?m ?n ?w))
    (distinct ?w b))
(<= (next (cell ?m ?n b)) (does ?w (mark ?j ?k))
    (true (cell ?m ?n b)) (or (distinct ?m ?j)
    (distinct ?n ?k)))
(<= (next (control white)) (true (control black)))
(<= (next (control black)) (true (control white)))
```

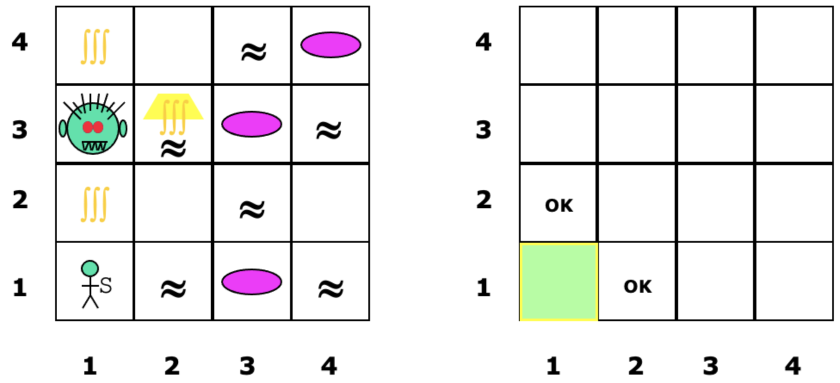
```
(<= (row ?m ?x) (true (cell ?m 1 ?x))
    (true (cell ?m 2 ?x)) (true (cell ?m 3 ?x)))
(<= (column ?n ?x) (true (cell 1 ?n ?x))
    (true (cell 2 ?n ?x)) (true (cell 3 ?n ?x)))
(<= (diagonal ?x) (true (cell 1 1 ?x))
    (true (cell 2 2 ?x)) (true (cell 3 3 ?x)))
(<= (diagonal ?x) (true (cell 1 3 ?x))
    (true (cell 2 2 ?x)) (true (cell 3 1 ?x)))
(<= (line ?x) (row ?m ?x))
(<= (line ?x) (column ?m ?x))
(<= (line ?x) (diagonal ?x))
(<= open (true (cell ?m ?n b))) (<= (goal white 100)
(line x))
(<= (goal white 50) (not open) (not (line x)) (not
(line o)))
(<= (goal white 0) open (not (line x)))
(<= (goal black 100) (line o))
(<= (goal black 50) (not open) (not (line x)) (not
(line o)))
(<= (goal black 0) open (not (line o)))
(<= terminal (line x))
(<= terminal (line o))
(<= terminal (not open))
```

The Wumpus World

More precise axioms than with propositional logic

- Percept has five values
- Time is important
- A typical sentence
  - Percept ([Stench, Breeze, Glitter, None, None], 7)
- The actions are terms
  - Turn(right), Turn(left), Forward, Shoot, Grab, Release
- Computing best action with a query
  - ∃ a: BestAction(a, 7)

Sensors: Stench, Breeze, Glitter, Bump, Scream



The Wumpus World

- Connecting percepts to actions

  $\forall s, b, u, c, t\ \ Percept([s, b, Glitter, u, c], t) \Rightarrow Action(Grab, t)$

  - requires many rules

- Can be simplified by intermediate predicates

  $\forall b, g, u, c, t\ Percept([Stench, b, g, u, c], t) \Rightarrow Stench(t)$

  $\forall b, g, u, c, t\ Percept([None, b, g, u, c], t) \Rightarrow \neg\ Stench(t)$

  $\forall s, g, u, c, t\ Percept([s, Breeze, g, u, c], t) \Rightarrow Breeze(t)$

  $\forall s, g, u, c, t\ Percept([s, None, g, u, c], t) \Rightarrow \neg\ Breeze(t)$

  …

  $\forall t\ Glitter(t) \Rightarrow BestAction(Grab, t)$

The Wumpus World

- Defining squares with [x,y] reference instead of atomic names

- Adjacency between two squares

  $$\forall x,y,a,b\ \ Adjacent([x,y],[a,b]) \Leftrightarrow$$
  $$(x=a \wedge y=b-1) \vee (x=a \wedge y=b+1)$$
  $$\vee (x=a-1 \wedge y=b) \vee (x=a+1 \wedge y=b)$$

- Modeling the breeze of pits

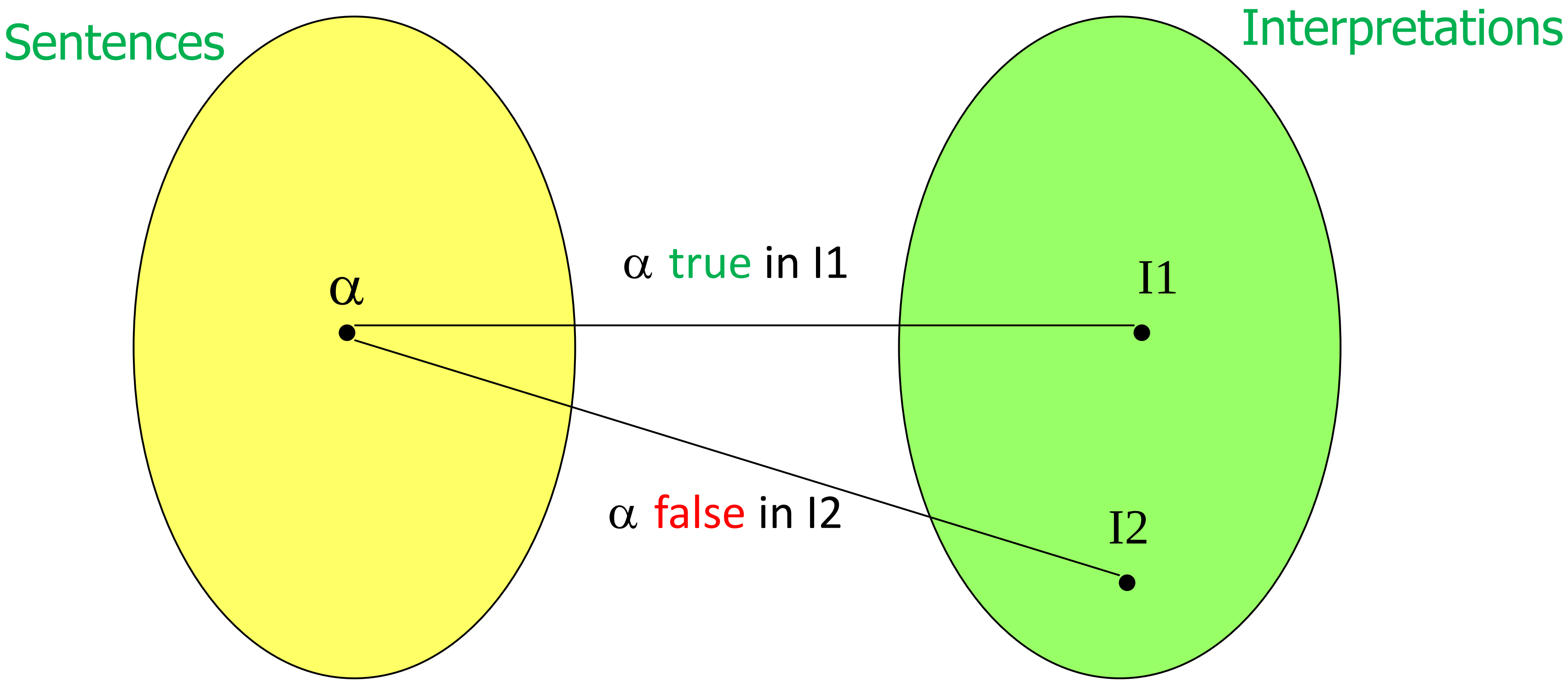
  $$\forall s\ Breezy(s) \Leftrightarrow \exists r\ Adjacent(r, s) \wedge Pit(r)$$

- …

Inference system

$$\begin{tikzcd}
\alpha \models \beta \arrow[d, shift right=2, "\text{Completeness}"'] \\
\alpha \vdash \beta \arrow[u, shift right=2, "\text{Soundness}"']
\end{tikzcd}$$

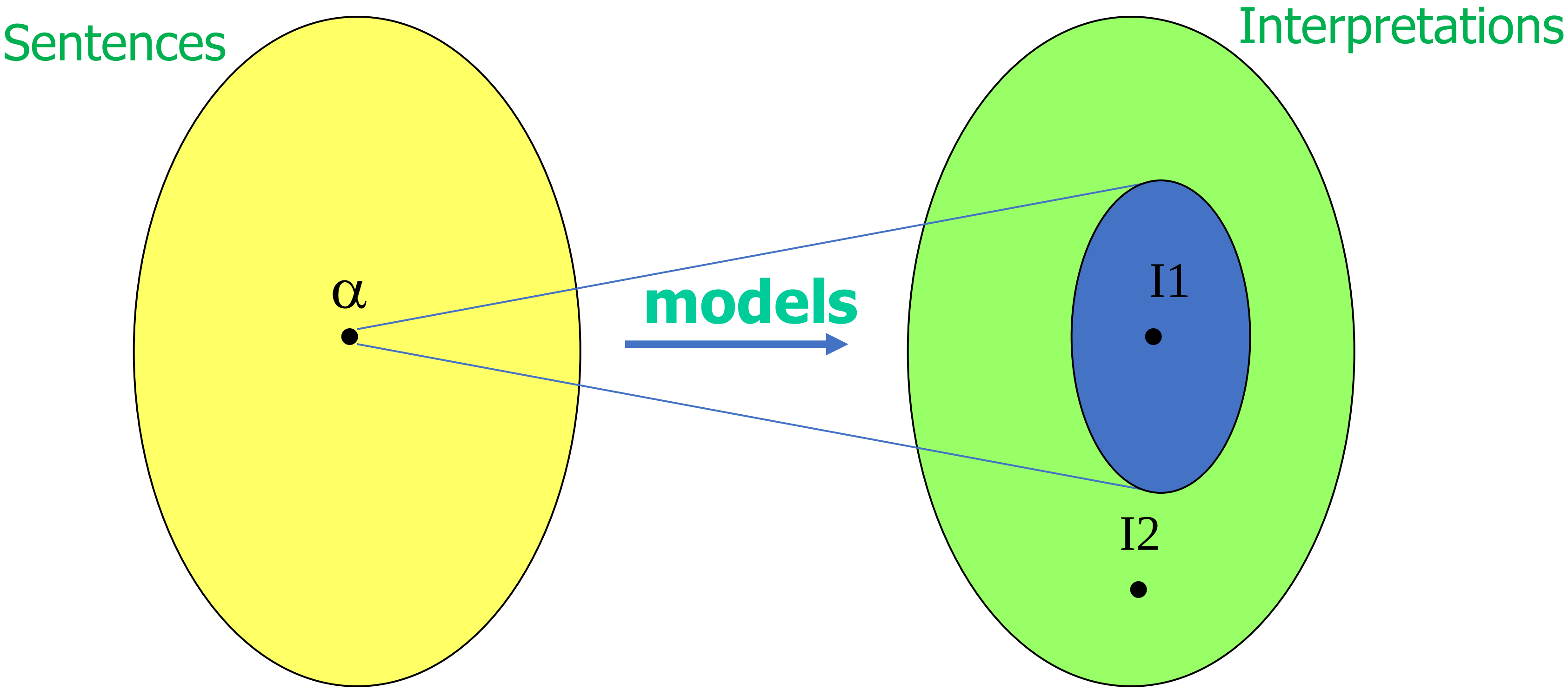
Truth value



How many sentences is there ?

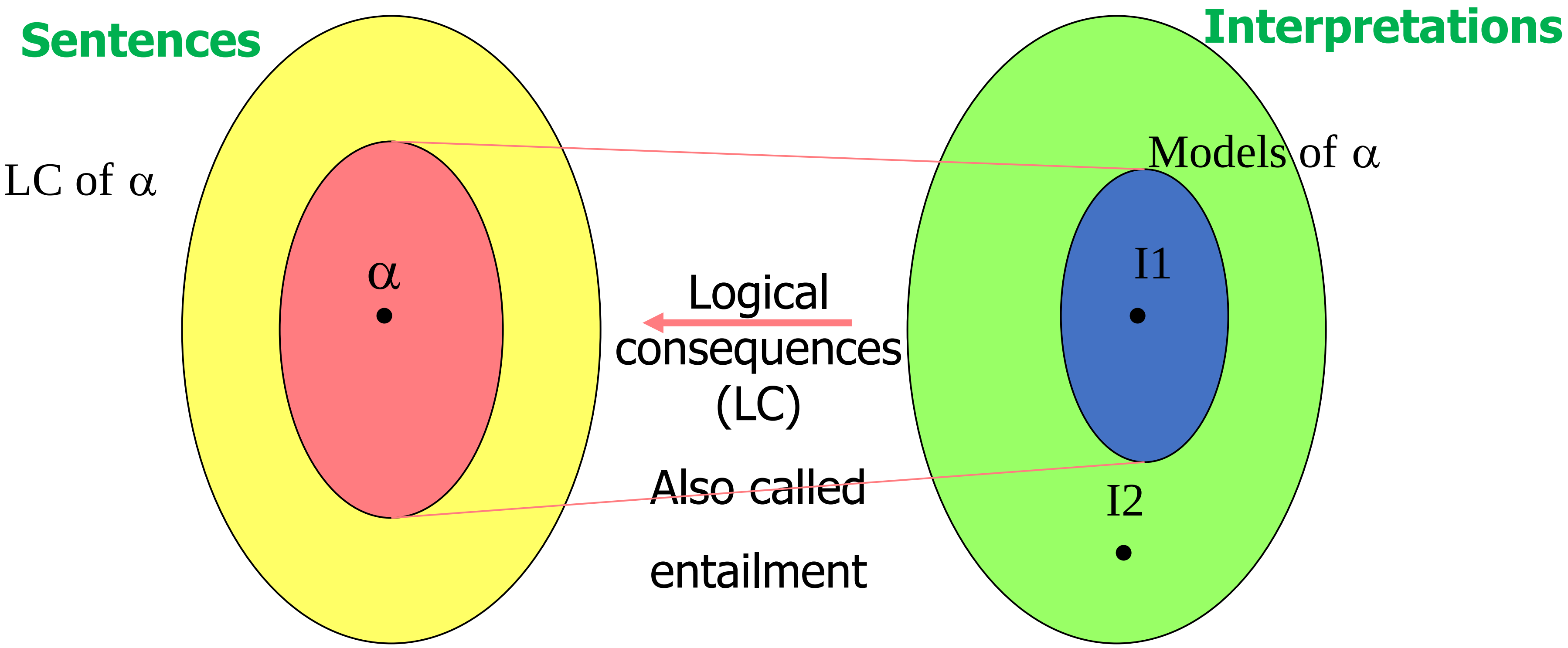
How many interpretations is there ?

Models



How many models is there ?

Models



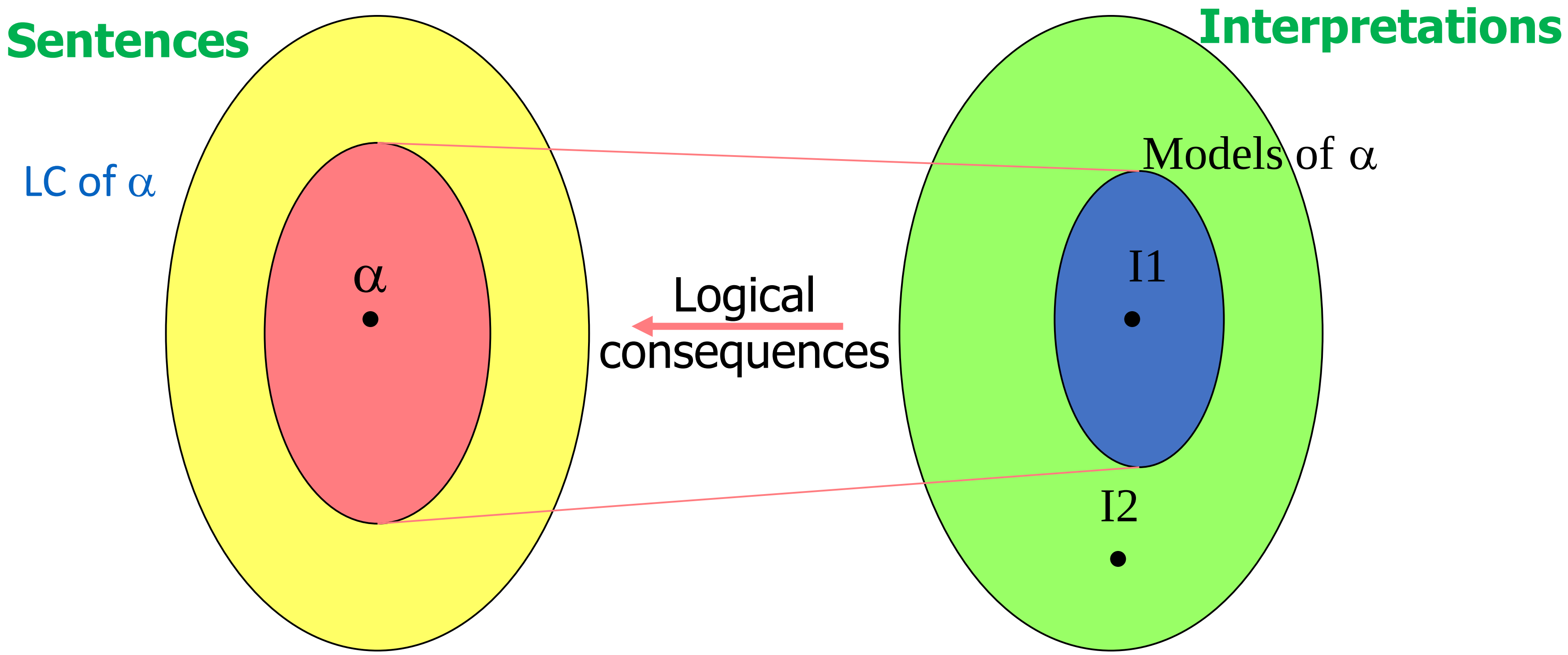
Logical consequences (LC)

Also called entailment

β in LC(α) iff $\alpha \models \beta$

Is it possible to compute this set ?

Completeness

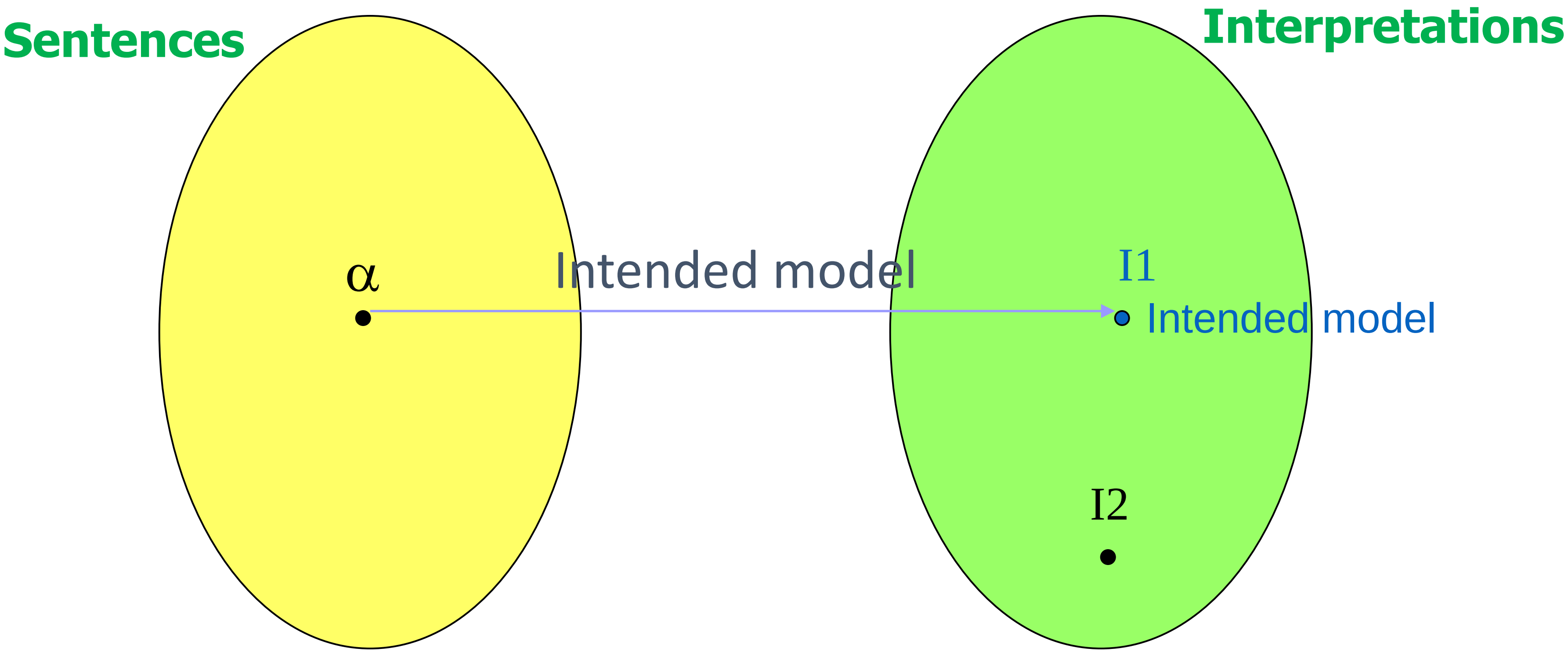


The set LC(α) is computable !

Inference system

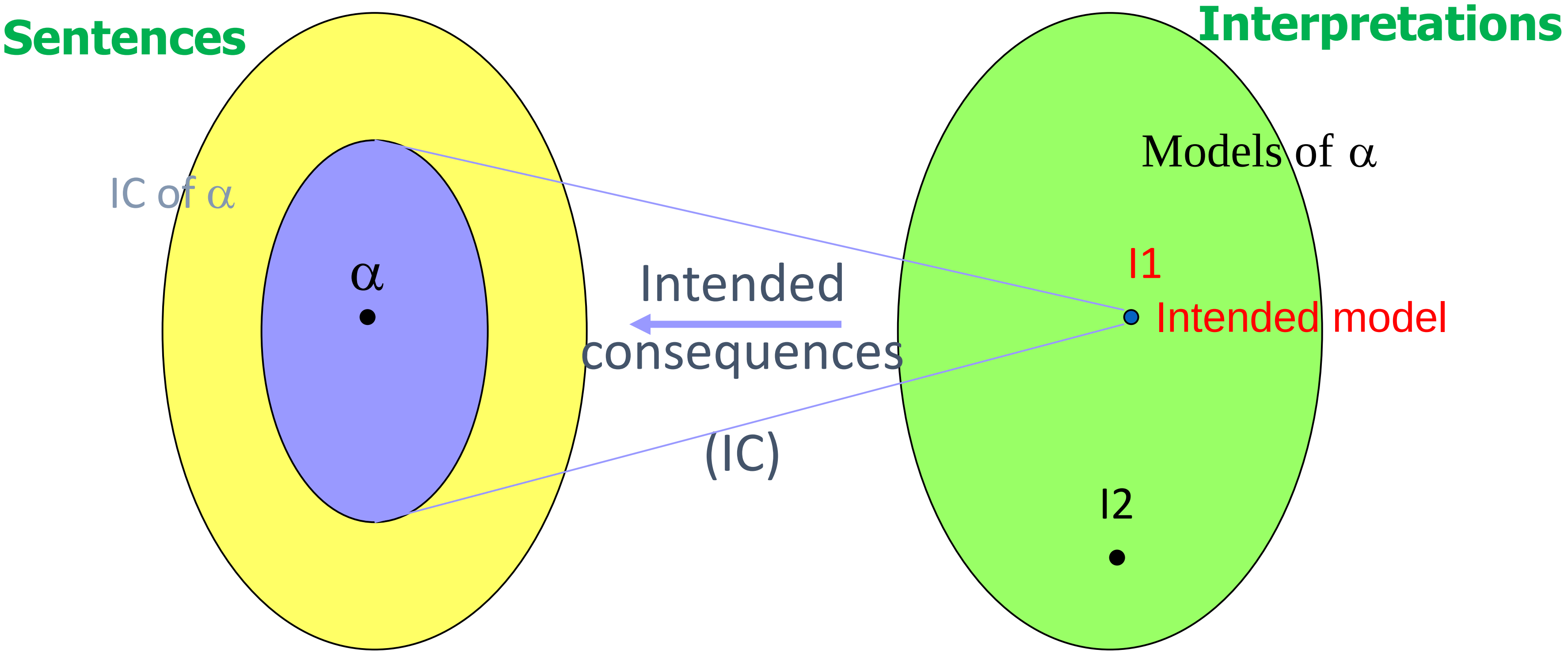
$$\begin{tikzcd} \alpha \models \beta \arrow[d, shift right, "\text{Completeness}"'] \\ \alpha \vdash \beta \arrow[u, shift right, "\text{Soundness}"'] \end{tikzcd}$$

Intended Interpretation



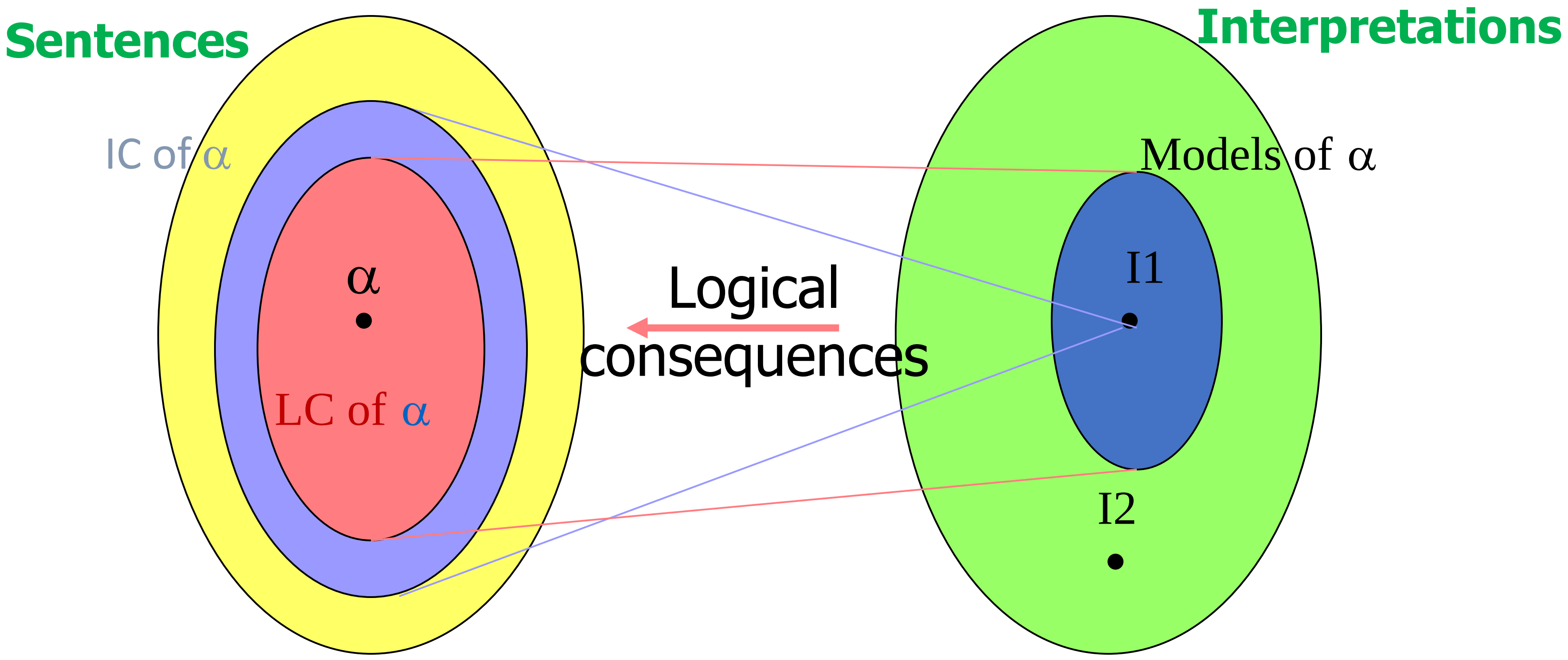
I1 is the intended interpretation (a model)

Intended Consequences



β in IC(α) iff β is true in the intended model I1

IC Versus LC



The set LC(α) is a subset of IC(α)

In the next episode…

Inference in First-Order Logic

AIMA Ch. 9
[Artificial Intelligence, Modern Approach.
Russel & Norvig, 4th ed. 2021]

Next class on 02/05/2024