



Foundations of Knowledge Representation

Lecture 2: Logics for Knowledge Representation

Hannes Straß

based on slides of
Bernardo Cuenca Grau,
Ian Horrocks, and
Przemysław Wałęga



Propositional Logic

We might consider using Propositional Logic

- It is one of the simplest logics
- It can be used to write simple representations of a domain
- There exist reasoning algorithms that exhibit excellent performance in practice
- (Most of) you are already familiar with it!

Syntax: Propositional Alphabet

- 1 Propositional **variables** (**PL**):
basic statements that can be true or false
- 2 The symbols \top (“truth”) and \perp (“falsehood”)
- 3 Propositional **connectives**:
 - \neg : negation (**not**)
 - \wedge : conjunction (**and**)
 - \vee : disjunction (**or**)
 - \rightarrow : implication (**if ... then**)
 - \leftrightarrow : bi-directional implication (**if and only if**)
- 4 Punctuation symbols “(” and “)” can be used to avoid ambiguity

Syntax: Formulas

Atomic formulas (atoms): propositional variables

Formulas: Inductively defined from atoms, \top , and \perp using **connectives**

Examples of formulas:

- If the tumour is benign then it does not have metastasis

$$\textit{Benign} \rightarrow \neg \textit{Metastasis}$$

- A tumour is in Stage 4 if and only if it is not benign

$$\textit{Stage4} \leftrightarrow \neg \textit{Benign}$$

- If a tumour has a treatment, it is surgery, or chemotherapy, or radiotherapy

$$\textit{Treatment} \rightarrow \textit{Surgery} \vee \textit{Chemo} \vee \textit{Radio}$$

Semantics: Interpretations

An **interpretation** \mathcal{I} assigns truth values to propositional variables:

$$\mathcal{I} : \mathbf{PL} \rightarrow \{\text{true}, \text{false}\}$$

An interpretation for a (set of) formulas X interprets the propositional variables occurring in X .

Example: An interpretation \mathcal{I} for the formula $R \rightarrow ((Q \vee R) \rightarrow R)$:

$$\begin{aligned} R^{\mathcal{I}} &= \text{true} \\ Q^{\mathcal{I}} &= \text{false} \end{aligned}$$

A formula with n propositional variables has 2^n interpretations.

Semantics of Formulas

The truth value of the propositional variables in a formula α determines the truth value of α .

$$R \rightarrow ((Q \vee R) \rightarrow R)$$

$$\begin{array}{cc} & \swarrow \quad \searrow \\ R & (Q \vee R) \rightarrow R \end{array}$$

$$\begin{array}{cc} & \swarrow \quad \searrow \\ Q \vee R & R \end{array}$$

$$\begin{array}{cc} & \swarrow \quad \searrow \\ Q & R \end{array}$$

$$R^{\mathcal{I}} = \text{true}$$

$$Q^{\mathcal{I}} = \text{false}$$

$$(Q \vee R)^{\mathcal{I}} = \text{true}$$

$$((Q \vee R) \rightarrow R)^{\mathcal{I}} = \text{true}$$

$$(R \rightarrow ((Q \vee R) \rightarrow R))^{\mathcal{I}} = \text{true}$$

We say that \mathcal{I} is a **model** of α , written $\mathcal{I} \models \alpha$, if \mathcal{I} makes α true.

Given \mathcal{I} and α , checking whether $\mathcal{I} \models \alpha$ can be done effectively, in polynomial time.

Using PL for KR

Propositional Logic provides a simple **KR language**.

To write down a representation of our domain do the following:

- 1 Identify the relevant propositions:

<i>Benign</i>	The tumour is benign
<i>Metastasis</i>	The tumour has metastasis
<i>Stage4</i>	The tumour is in Stage 4
...	

- 2 Express our knowledge using a set of formulas (**knowledge base**):

$$\begin{aligned} & \textit{Benign} \\ \textit{Benign} & \leftrightarrow \neg \textit{Metastasis} \\ \textit{Stage4} & \rightarrow \textit{Metastasis} \\ & \dots \end{aligned}$$

Reasoning with a Knowledge Base

Knowledge Base \mathcal{K}_1 :

Benign \wedge *Stage4*

Benign $\leftrightarrow \neg$ *Metastasis*

Stage4 \rightarrow *Metastasis*

...

Knowledge Base \mathcal{K}_2 :

Benign

Benign $\leftrightarrow \neg$ *Metastasis*

Stage4 \rightarrow *Metastasis*

...

We would like to answer the following questions:

1 Do our KBs make sense?

\mathcal{K}_1 seems contradictory

2 What is the **implicit knowledge** we can derive from our KBs?

\mathcal{K}_2 seems to imply the formula \neg *Stage4*

Satisfiability Problem

Satisfiability: An instance is a formula α .
The answer is **true** if there exists a model \mathcal{I} of α
and **false** otherwise.

For α the formula $R \rightarrow ((Q \vee R) \rightarrow R)$ the answer is **true**:
 \mathcal{I} assigning R to **true** and Q to **false** is a model of α .

For α the formula $(R \wedge Q) \leftrightarrow (\neg R \vee \neg Q)$ the answer is **false**:
None of the 4 possible interpretations is a model of α .

Satisfiability defined for sets of formulas in the obvious way.

The following knowledge base is **unsatisfiable**:

$$\begin{aligned} \mathcal{K}_1 = \{ & \textit{Benign} \wedge \textit{Stage4} \\ & \textit{Benign} \leftrightarrow \neg \textit{Metastasis} \\ & \textit{Stage4} \rightarrow \textit{Metastasis} \\ & \dots \} \end{aligned}$$

Other Reasoning Problems

Validity: An instance is a formula α .

The answer is **true** if every interpretation for α is a model of α and **false** otherwise.

Entailment: An instance is a pair of formulas α, β .

The answer is **true** if every model of α is also a model of β and **false** otherwise.

Equivalence: An instance is a pair of formulas α, β .

The answer is **true** if the set of all models of α and β coincide and **false** otherwise.

Reductions Between Problems

Intuitively, these problems are strongly related:

- α is valid if and only if $\neg\alpha$ is unsatisfiable
- α and β are equivalent if and only if α entails β and β entails α
- α entails β if and only if $\alpha \wedge \neg\beta$ is unsatisfiable

A **reduction** from problem P_1 to P_2 is a function f such that

- for each input x to P_1 , the answer of P_1 for input x coincides with the answer of P_2 for input $f(x)$,
- given x , the input $f(x)$ can be efficiently computed.

The mentioned before (and many other) problems can be **reduced to (un)satisfiability**

Expressivity -v- Complexity

Propositional satisfiability is (famously) NP-complete:

Theorem (Cook-Levin)

Propositional satisfiability is an NP-complete problem:

- 1 *It is in NP*
- 2 *It is NP-hard: all problems in NP are reducible to it*

So should we just give up (as reasoning is intractable)?

NO!

- Algorithms such as DPLL are effective in practice
- Highly optimised SAT solvers can deal with problems containing millions of propositional variables (www.maxsat.udl.cat)

Limitations of Propositional Logic

Consider the following statements from a medical domain:

- A juvenile disease affects only children or teenagers
- Children and teenagers are not adults
- Juvenile arthritis is a kind of arthritis and a juvenile disease
- Arthritis affects some adults

Let us try to represent these statements in propositional logic:

$$\begin{aligned} JuvDisease &\rightarrow AffectsChild \vee AffectsTeenager \\ Child \vee Teenager &\rightarrow \neg Adult \\ JuvArthritis &\rightarrow JuvDisease \wedge Arthritis \\ Arthritis &\rightarrow AffectsAdult \end{aligned}$$

Limitations of Propositional Logic

Some intuitive consequences of our statements:

- Juvenile arthritis does not affect adults
- Arthritis is not a juvenile disease

We expect the following formulas to follow:

$$\begin{aligned} \text{JuvArthritis} &\rightarrow \neg \text{AffectsAdult} \\ \text{Arthritis} &\rightarrow \neg \text{JuvDisease} \end{aligned}$$

However, neither of them is entailed.

Even worse, if we add to our initial formulas the following ones, we obtain an **unsatisfiable** set of formulas.

$$\begin{aligned} \text{JuvArthritis} &\rightarrow \neg \text{AffectsAdult} \\ \text{JuvArthritis} & \end{aligned}$$

Limitations of Propositional Logic

What is going wrong?

- A juvenile disease affects only children or teenagers
- Children and teenagers are not adults
- Juvenile arthritis is a kind of arthritis and a juvenile disease
- Arthritis affects some adults

Intuitively ...

- Green color represents sets of objects
- Blue color represents relationships between objects
- Red color indicates whether a statement holds for “all” or for “some” objects.

We cannot make such distinctions in propositional logic!!!

Limitations of Propositional Logic

We need a language that allows us to

- 1 Represent **sets of objects**
- 2 Represent **relationships between objects**
- 3 Write statements that are true for **some** or **all** objects satisfying certain conditions
- 4 Express everything we can express in propositional logic (**and**, **or**, **implies**, **not**, ...)

Examples of conditions we want to express:

- **For all objects** c ,
if c belongs to **the set of juvenile diseases**
and it **affects** an object d ,
then d belongs to **the set of children**
or to the **set of teenagers**.
- **There exist objects** c, d such that c belongs to the **set of arthritis**
and d belongs to **the set of adults** and c **affects** d .

FOL Syntax: Symbols

A first-order alphabet consists of

- Predicate Symbols, each with a fixed arity

Arthritis Unary Predicate

Affects Binary Predicate

- Function symbols, each with a fixed arity

ssnOf Unary Function Symbol

- Constants: *JohnSmith*, *MaryJones*, *JRA*

- Variables: x , y , z

- Propositional connectives $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$

- Symbols \top and \perp .

- The universal and existential quantifiers: \forall, \exists

FOL Syntax: Terms

Terms stand for specific objects:

- Variables are terms
- Constants are terms
- The application of a function symbol to terms leads to a term

<i>JohnSmith</i>	stands for	the person named John Smith
<i>ssnOf(JohnSmith)</i>	stands for	the ssn number of John Smith
x	stands for	some object (undetermined)
<i>ssnOf(x)</i>	stands for	some ssn number (undetermined)

FOL Syntax: formulas

An atomic formula (atom) is of the form

$P(t_1, \dots, t_n)$ P is an n -ary predicate, t_i are terms

Examples:

Child(*JohnSmith*) John Smith is a child

JuvenileArthritis(*JRA*) JRA is a juvenile arthritis

Affects(*JRA*, *JohnSmith*) John Smith is affected by JRA

An atom represents simple statement:

- similar to atoms in propositional logic,
- but first-order atoms have finer-grained structure.

FOL Syntax: Formulas

Complex formulas:

- Every atom is a formula

$Child(JohnSmith), Affects(x, JohnSmith)$

- \top and \perp are formulas
- If α is a formula, then $\neg\alpha$ is a formula

$\neg Affects(JRA, JohnSmith), \neg Child(y)$

- If α, β are formulas, $(\alpha \circ \beta)$ is a formula for $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$

$Affects(JRA, y) \rightarrow Child(y) \vee Teenager(y)$

- If α a formula and x a variable, $(\forall x.\alpha), (\exists x.\alpha)$ are formulas

$\forall y.(Affects(JRA, y) \rightarrow Child(y) \vee Teenager(y))$
 $\neg(\exists x.\exists y(JuvArthritis(x) \wedge Affects(x, y) \wedge Adult(y)))$

FOL Syntax: Free and Bound Variables

Intuitively, a **free** variable occurrence in a formula is one that does not appear in the scope of a quantifier:

$$\begin{aligned} & \text{Affects}(\text{JRA}, \underline{y}) \rightarrow \text{Child}(\underline{y}) \vee \text{Teenager}(\underline{y}) \\ & \exists x. (\text{JuvArthritis}(x) \wedge \text{Affects}(x, \underline{y}) \wedge \text{Adult}(\underline{y})) \\ & \exists x. (\text{JuvArthritis}(x)) \wedge \text{Affects}(\underline{x}, \underline{y}) \wedge \text{Adult}(\underline{y}) \end{aligned}$$

A variable occurrence is **bound** if it is not free.

A formula is **rectified** if a variable does not appear both free and bound and each quantifier refers to a different variable.

$$\text{Affects}(\text{JRA}, \underline{y}) \rightarrow \exists x. (\text{JuvArthritis}(x)) \wedge \text{Affects}(\underline{x}, \underline{y}) \wedge \text{Adult}(\underline{y}) \quad \times$$

A **sentence** is a formula with no free variable occurrences.

Example FOL Sentences

- A juvenile disease affects only children or teenagers
- Children and teenagers are not adults
- Juvenile arthritis is a kind of arthritis and a juvenile disease
- Arthritis affects some adults

$$\begin{aligned} &\forall x. (\forall y. (JuvDisease(x) \wedge Affects(x, y) \rightarrow Child(y) \vee Teenager(y))) \\ &\quad \forall x. (Child(x) \vee Teenager(x) \rightarrow \neg Adult(x)) \\ &\forall x. (JuvArthritis(x) \rightarrow Arthritis(x) \wedge JuvDisease(x)) \\ &\quad \exists x. (\exists y. (Arthritis(x) \wedge Affects(x, y) \wedge Adult(y))) \end{aligned}$$

FOL Interpretations

As in PL, meaning of sentences given by **interpretations**

An **interpretation** is a pair $\mathcal{I} = \langle \mathbf{D}, \cdot^{\mathcal{I}} \rangle$ where:

- \mathbf{D} is a non-empty set, called the **interpretation domain**.

$$\mathbf{D} = \{u, v, w, s\}$$

- $\cdot^{\mathcal{I}}$ is the **interpretation function** and it associates:

- With each constant c an object $c^{\mathcal{I}} \in \mathbf{D}$.

$$\text{JohnSmith}^{\mathcal{I}} = u \quad \text{MaryWilliams}^{\mathcal{I}} = v \quad \text{JRA}^{\mathcal{I}} = w \quad \dots$$

- With each n -ary function symbol f , a function $f^{\mathcal{I}} : \mathbf{D}^n \rightarrow \mathbf{D}$.

$$\text{ssnOf}^{\mathcal{I}} = \{u \mapsto s, \dots\}$$

- With each n -ary predicate symbol P , a relation $P^{\mathcal{I}} \subseteq \mathbf{D}^n$.

$$\text{Child}^{\mathcal{I}} = \{u, v\} \quad \text{Adult}^{\mathcal{I}} = \emptyset \quad \text{Affects}^{\mathcal{I}} = \{\langle w, u \rangle, \dots\}$$

Evaluation of Terms

Terms are interpreted as elements of the interpretation domain.

We have already seen how to interpret constants

$$\text{JohnSmith}^{\mathcal{I}} = u \quad \text{MaryWilliams}^{\mathcal{I}} = v \quad \text{JRA}^{\mathcal{I}} = w \quad \dots$$

To interpret terms, we need to interpret (free) variables by means of a mapping from variables to domain elements (an **assignment**)

Given \mathcal{I} and assignment \mathbf{a} , we can interpret any term. Let \mathcal{I} be as before and \mathbf{a} map x to u :

$$\begin{aligned} \text{JohnSmith}^{\mathcal{I}, \mathbf{a}} &= u \\ x^{\mathcal{I}, \mathbf{a}} &= u \\ (\text{ssnOf}(x))^{\mathcal{I}, \mathbf{a}} &= \text{ssnOf}^{\mathcal{I}}(u) = s \end{aligned}$$

Formula Evaluation

Given \mathcal{I} and \mathbf{a} , a formula is interpreted as either true or false

Atomic formulas:

$$P(t_1, \dots, t_n)^{\mathcal{I}, \mathbf{a}} = \text{true} \quad \text{iff} \quad \langle t_1^{\mathcal{I}, \mathbf{a}}, \dots, t_n^{\mathcal{I}, \mathbf{a}} \rangle \in P^{\mathcal{I}} \quad \text{e.g.:}$$

$$\text{Child}(\text{JohnSmith})^{\mathcal{I}, \mathbf{a}} = \text{true} \quad \text{since} \quad \text{JohnSmith}^{\mathcal{I}, \mathbf{a}} = u \\ \text{and} \quad \text{Child}^{\mathcal{I}} = \{u, v\}$$

$$\text{Affects}(\text{JRA}, x)^{\mathcal{I}, \mathbf{a}} = \text{true} \quad \text{since} \quad \text{JRA}^{\mathcal{I}, \mathbf{a}} = w, \quad x^{\mathcal{I}, \mathbf{a}} = u \\ \text{and} \quad \text{Affects}^{\mathcal{I}} = \{\langle w, u \rangle\}$$

Propositional connectives are interpreted as usual:

$$\begin{aligned} (\neg \text{Child}(\text{JohnSmith}))^{\mathcal{I}, \mathbf{a}} &= \text{false} \\ (\text{Affects}(\text{JRA}, x) \wedge \text{Child}(\text{JohnSmith}))^{\mathcal{I}, \mathbf{a}} &= \text{true} \\ (\text{Child}(\text{JohnSmith}) \rightarrow \neg \text{Child}(\text{JohnSmith}))^{\mathcal{I}, \mathbf{a}} &= \text{false} \end{aligned}$$

Formula Evaluation

Given \mathcal{I} and \mathbf{a} , a formula is interpreted as either true or false

Existential quantifiers:

$$(\exists x. \textit{Affects}(\textit{JRA}, x))^{\mathcal{I}, \mathbf{a}_\emptyset} = \text{true}$$

since there exists an assignment \mathbf{a} extending \mathbf{a}_\emptyset such that $\textit{Affects}(\textit{JRA}, x)^{\mathcal{I}, \mathbf{a}} = \text{true}$

Universal quantifiers:

$$(\forall x. \textit{Affects}(\textit{JRA}, x))^{\mathcal{I}, \mathbf{a}_\emptyset} = \text{false}$$

since it is not true that, for any assignment \mathbf{a} extending \mathbf{a}_\emptyset , $\textit{Affects}(\textit{JRA}, x)^{\mathcal{I}, \mathbf{a}} = \text{true}$.

Evaluation of Sentences

For interpreting sentences, assignments are irrelevant

$$\forall x.(\forall y.(JuvDisease(x) \wedge Affects(x, y) \rightarrow Child(y) \vee Teenager(y)))$$

And the interpretation \mathcal{I} given as follows:

$$\begin{aligned} \mathbf{D} &= \{u, v, w\} \\ JuvDisease^{\mathcal{I}} &= \{u\} \\ Child^{\mathcal{I}} &= \{w\} \\ Teenager^{\mathcal{I}} &= \emptyset \\ Affects^{\mathcal{I}} &= \{\langle u, w \rangle\} \end{aligned}$$

The formula with no quantifiers must evaluate to true in \mathcal{I} for all values $x, y \in \mathbf{D}$. Example for $x = u$ and $y = v$:

$$\begin{aligned} JuvDisease(u) \wedge Affects(u, v) \rightarrow Child(v) \vee Teenager(v) \\ \text{true} \wedge \text{false} \rightarrow \text{true} \vee \text{false} \\ \text{true} \end{aligned}$$

Propositional vs FOL Interpretations

More complicated to give meaning to FOL than to PL formulas:

$JuvDisease \rightarrow AffectsChild \vee AffectsTeenager$ (PL)

$\forall x.(\forall y.(JuvDisease(x) \wedge Affects(x, y) \rightarrow Child(y) \vee Teenager(y)))$ (FOL)

PL Interpretations

- Assigns truth values to atoms
- The truth value of complex formulas determined by induction

Example formula has 8 possible interpretations and 7 models

FOL interpretations

- Specify the domain for quantifiers to quantify over
- Interpret constants, predicates, functions
- Assign objects to variables

Example formula has ∞ possible interpretations and ∞ models

Basic Reasoning Problems in FOL

Exactly the same ones as in Propositional Logic

Satisfiability: An instance is a (set of) sentence(s) X .
The answer is **true** if X has a model and **false** otherwise.

Entailment: An instance is a pair of (sets of) sentence(s) X, Y .
The answer is **true** if every model of X is also a model of Y
and **false** otherwise.

Equivalence: An instance is a pair of (sets of) sentence(s) X, Y .
The answer is **true** if the set of all models of X and Y coincide
and **false** otherwise.

Again, these problems are **reducible to satisfiability**

The Process of Knowledge Engineering

Starts with a **problem/application**:

FOL-based KR is being used in several countries to describe electronic patient records (e.g., by specifying knowledge about human anatomy, drugs, surgical procedures, and so on).

We have been hired to write a FOL knowledge base about different types of arthritis (to be used by a medical research company in the annotation of patient data)

Next, we need to **gather requirements**

- Find out what kind of data will be in the application
(\Rightarrow) Usually, no access to the actual data
- Meet (or work closely with) with the company's domain experts
- Gather relevant documentation about the domain

Outcome: diagrams and list of textual descriptions

Establishing the Vocabulary

Start from a textual description or diagram:

- A juvenile disease affects only children or teenagers
- Children and teenagers are not adults
- Juvenile arthritis is a kind of arthritis and a juvenile disease
- Arthritis affects some adults

Identify the important types of objects (unary FOL predicates):

juvenile disease, child, teenager, adult, ...

Identify the important types of relationships (n-ary FOL predicates)

affects, ...

Identify the important functions (none in this particular case)

Basic Facts

Now that we have the basic vocabulary, we can acquire the **data**

Child(*JohnSmith*) John Smith is a child

JuvenileArthritis(*JRA*) JRA is a juvenile arthritis

\neg *Affects*(*JRA*, *MaryJones*) Mary Jones not affected by JRA

Usually data consists of (possibly negated) atoms.

But data can also reflect more complex information:

Child(*JohnSmith*) \vee *Child*(*MaryJones*) Either John or Mary is a child

In our case, the medical company will take care of the data

Terminological Axioms

Sentences describing the general meaning of predicate and function symbols (independently of the concrete data)

- Sub-type statements

$$\forall x. (\text{JuvArthritis}(x) \rightarrow \text{Arthritis}(x))$$

- Full definitions:

$$\forall x. (\text{JuvArthritis}(x) \leftrightarrow \text{Arthritis}(x) \wedge \text{JuvDisease}(x))$$

- Disjointness statements:

$$\forall x. (\text{Child}(x) \rightarrow \neg \text{Adult}(x))$$

- Covering statements:

$$\forall x. (\text{Person}(x) \rightarrow \text{Adult}(x) \vee \text{Child}(x) \vee \text{Teenager}(x))$$

- Type restrictions:

$$\forall x. (\forall y. (\text{Affects}(x, y) \rightarrow \text{Arthritis}(x) \wedge \text{Person}(y)))$$

- Other general statements:

$$\forall x. (\forall y. (\text{JuvDisease}(x) \wedge \text{Affects}(x, y) \rightarrow \text{Child}(y) \vee \text{Teenager}(y)))$$

Data vs Terminological Knowledge

- The **Data** describes specific objects
(\Rightarrow) Sentences without variables or quantifiers (usually atoms)
- **Terminological axioms** describe general properties of the application domain, independently of the data.
(\Rightarrow) Universally quantified sentences with no constants

This separation is not theoretically “clean” in FOL:

$$\begin{aligned} & \forall y. (\textit{Affects}(\textit{JRA}, y) \rightarrow \textit{Child}(y) \vee \textit{Teenager}(y)) \\ & \forall x. (\textit{Cont}(x) \rightarrow (x = \textit{Eur}) \vee (x = \textit{Asia}) \vee (x = \textit{Amer}) \\ & \quad \vee (x = \textit{Afr}) \vee (x = \textit{Aus}) \vee (x = \textit{Antart})) \end{aligned}$$

But it is conceptually and practically **very useful**.

Set of Terminological Axioms often called an **Ontology**

Ontology + Data often called a **Knowledge Base**

Model Selection

Initially, we have no data or terminological axioms

(\Rightarrow) We have said nothing about our application

(\Rightarrow) Any possible interpretation is a model

We now add to the knowledge base the axiom

$$\forall x. (JuvArthritis(x) \rightarrow Arthritis(x) \wedge JuvDisease(x))$$

Any interpretation \mathcal{I} such that

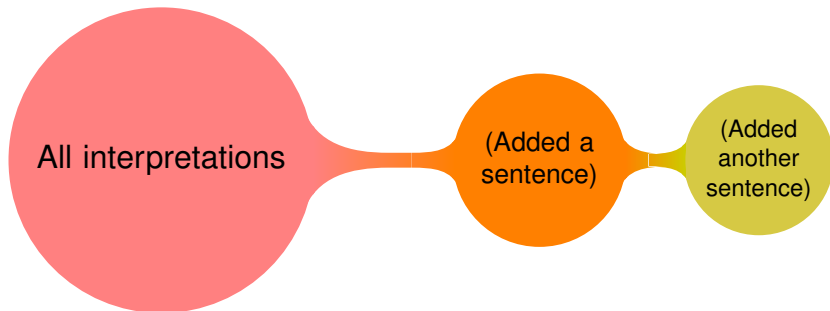
$$JuvArthritis^{\mathcal{I}} \not\subseteq Arthritis^{\mathcal{I}} \cap JuvDisease^{\mathcal{I}}$$

is no longer a model

By writing down a FOL sentence we have:

- Discarded (possibly infinitely many) models
- Selected the models consistent with our statement

Model Selection



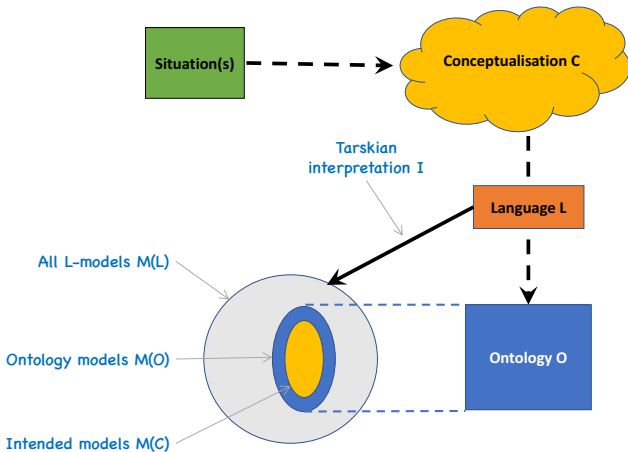
By adding FOL statements to a knowledge base we **gain knowledge**:

- Reduce the number of models
- Obtain new logical consequences (recall entailment definition)

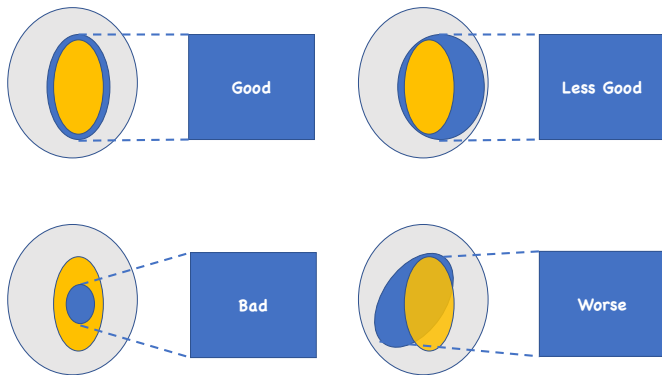
Two special cases:

- New sentence entailed by previous ones: models stay the same
Redundant knowledge
- Knowledge base becomes unsatisfiable: no models, everything follows
Meaningless knowledge (error in the modeling)

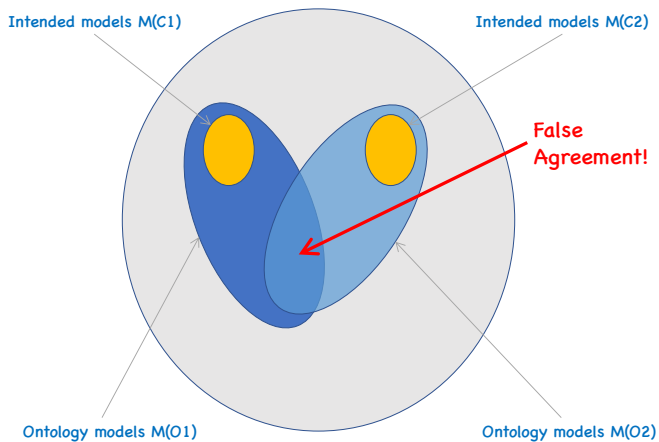
Ontological Modelling



Ontological Modelling



Ontological Modelling



The Role of Reasoning

Why are reasoning problems (satisfiability, entailment) useful?

- 1 Detect errors
 - ⇒ Knowledge base becomes unsatisfiable
 - ⇒ We get an unintuitive (and “wrong”) entailment
 - ⇒ We don’t get an intuitive (and “right”) entailment
- 2 Discover new knowledge
 - ⇒ Things we weren’t aware we knew
- 3 Richer query answers
 - ⇒ Retrieve more (relevant) data

Without reasoning, knowledge engineering becomes unfeasible

- 1 Knowledge bases grow very large (1,000s of sentences)
- 2 Errors are difficult to detect manually
- 3 Query answers do not take knowledge into account

Expressivity -v- Complexity

Theorem

FOL satisfiability is an *undecidable* problem: there is no procedure that given any set of first order sentences S :

- 1 Always terminates
- 2 Returns true if and only if S is satisfiable

Proof idea: [proof beyond the scope of this course]

- 1 Define a computable function f which takes a Turing Machine M to a sentence $f(M)$ in FOL.
- 2 M does not halt on the empty tape if and only if $f(M)$ has a model

(The Halting problem on the empty tape is undecidable)

So should we just give up (reasoning is intractable)?

MAYBE!

- Highly optimised FOL theorem provers are effective in practice
- But still can't cope with realistic KR problems

Limitations of FOL

FOL is powerful, but still can't capture

- Transitive closure (Ancestor is the transitive closure of Parent)
- Defaults and exceptions (Birds fly by default; Penguins are an exception)
- Probabilistic knowledge (Children suffer from JRA with probability x)
- Vague knowledge (Ian is Tall)
- ...

We will return to some of these issues later in the course