TECHNISCHE UNIVERSITÄT DRESDEN

Computational Logic Group

Sebastian Rudolph (based on slides by Bernardo Cuenca Grau, Ian Horrocks, Przemysław Wałęga)
Faculty of Computer Science, Institute of Artificial Intelligence, Computational Logic Group

Logics for Knowledge Representation

Lecture 2, 20th Oct 2025 // Foundations of Knowledge Representation, WS 2025/26

Course organisation

Registration via Selma

- In case you want to take an exam in this course . . .
- . . . do not forget to register online via the Selma portal!
- (Registration is mandatory for examination.)
- See course web page for links.

Exercise sessions

- Start next week Monday

Propositional Logic

What logic to use for knowledge representation? – It depends.

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 $\bot$ (“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 (in linearised representations of formulas).

Syntax: Formulas

Atomic formulas (atoms): propositional variables

Formulas: Inductively defined from atoms, $\top$, and $\bot$ using connectives

Examples of formulas:

- If the tumour is benign then there is no metastasis

$$\mathit{Benign} \quad \rightarrow \quad \neg \mathit{Metastasis}$$

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

$$\mathit{Stage4} \quad \leftrightarrow \quad \neg \mathit{Benign}$$

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

$$\mathit{Treatment} \quad \rightarrow \quad \mathit{Surgery} \lor \mathit{Chemo} \lor \mathit{Radio}$$

Semantics: Interpretations

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

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

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

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

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

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.

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

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

$Q \vee R \qquad R$

$Q \qquad R$

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

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:

$Benign$	The tumour is benign
$Metastasis$	The tumour has metastasis
$Stage4$	The tumour is in Stage 4
$\ldots$	

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

$$\begin{aligned} Benign \\ Benign \leftrightarrow \neg Metastasis \\ Stage4 \rightarrow Metastasis \\ \ldots \end{aligned}$$

Reasoning with a Knowledge Base

Knowledge Base $\mathcal{K}_1$:

$$\begin{aligned} \mathit{Benign} \land \mathit{Stage4} \\ \mathit{Benign} \leftrightarrow \neg\mathit{Metastasis} \\ \mathit{Stage4} \rightarrow \mathit{Metastasis} \\ \ldots \end{aligned}$$

Knowledge Base $\mathcal{K}_2$:

$$\begin{aligned} \mathit{Benign} \\ \mathit{Benign} \leftrightarrow \neg\mathit{Metastasis} \\ \mathit{Stage4} \rightarrow \mathit{Metastasis} \\ \ldots \end{aligned}$$

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 $\neg\mathit{Stage4}$

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 $R \to ((Q \vee R) \to R)$ the answer is true:
$\quad\mathcal{I}$ 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:
$\quad$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:

$$\mathcal{K}_1 = \{\mathit{Benign} \wedge \mathit{Stage4}, \quad \mathit{Benign} \leftrightarrow \neg\mathit{Metastasis}, \quad \mathit{Stage4} \to \mathit{Metastasis}\}$$

Other Reasoning Problems

Problem

Validity: An instance is a formula α.
The answer is true if every interpretation for α is a model of α and false otherwise.

Problem

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.

Problem

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

Definition

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 aforementioned (and many other) problems can be reduced to (un)satisfiability

Expressivity -v- Complexity

Propositional satisfiability is (famously) NP-complete:

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:

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

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:

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

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.

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

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 . . .

- Light blue represents sets of objects
- Green represents relationships between objects
- Purple 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 $\bot$.
- 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

$\mathit{JohnSmith}$	stands for	the person named John Smith
$\mathit{ssnOf}(\mathit{JohnSmith})$	stands for	the ssn number of John Smith
x	stands for	some object (undetermined)
$\mathit{ssnOf}(x)$	stands for	some ssn number (undetermined)

FOL Syntax: Formulas

An atomic formula (atom) is of the form

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

Examples:

$\mathit{Child}(\mathit{JohnSmith})$	John Smith is a child
$\mathit{JuvenileArthritis}(\mathit{JRA})$	JRA is a juvenile arthritis
$\mathit{Affects}(\mathit{JRA}, \mathit{JohnSmith})$	John Smith is affected by JRA

An atom represents a 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

$$\mathit{Child}(\mathit{JohnSmith}), \mathit{Affects}(x, \mathit{JohnSmith})$$

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

$$\neg\mathit{Affects}(\mathit{JRA}, \mathit{JohnSmith}), \neg\mathit{Child}(y)$$

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

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

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

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

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 corresponding quantifier:

$$\mathit{Affects}(\mathit{JRA}, \underline{y}) \to \mathit{Child}(\underline{y}) \lor \mathit{Teenager}(\underline{y})$$

$$\exists x.(\mathit{JuvArthritis}(x) \land \mathit{Affects}(x, \underline{y}) \land \mathit{Adult}(\underline{y}))$$

$$\exists x.(\mathit{JuvArthritis}(x)) \land \mathit{Affects}(\underline{x}, \underline{y}) \land \mathit{Adult}(\underline{y})$$

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.

$$\mathit{Affects}(\mathit{JRA}, \underline{y}) \to \exists x.(\mathit{JuvArthritis}(x)) \land \mathit{Affects}(\underline{x}, \underline{y}) \land \mathit{Adult}(\underline{y}) \qquad \times$$

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

$$\forall x.(\forall y.(\mathit{JuvDisease}(x) \land \mathit{Affects}(x,y) \rightarrow \mathit{Child}(y) \lor \mathit{Teenager}(y)))$$

$$\forall x.(\mathit{Child}(x) \lor \mathit{Teenager}(x) \rightarrow \neg \mathit{Adult}(x))$$

$$\forall x.(\mathit{JuvArthritis}(x) \rightarrow \mathit{Arthritis}(x) \land \mathit{JuvDisease}(x))$$

$$\exists x.(\exists y.(\mathit{Arthritis}(x) \land \mathit{Affects}(x,y) \land \mathit{Adult}(y)))$$

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 $c^{\mathcal{I}} \in \mathbf{D}$.

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

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

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

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

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

Evaluation of Terms

Terms are interpreted as elements of the interpretation domain.

We have already seen how to interpret constants

$$\mathit{JohnSmith}^{\mathcal{I}} = u \quad \mathit{MaryWilliams}^{\mathcal{I}} = v \quad \mathit{JRA}^{\mathcal{I}} = w \quad \ldots$$

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

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

Formula Evaluation

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

Atomic formulas:

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

$$\begin{aligned} \mathit{Child}(\mathit{JohnSmith})^{\mathcal{I},\mathbf{a}} &= \text{true} \quad \text{since} \quad \mathit{JohnSmith}^{\mathcal{I},\mathbf{a}} = u \\ &\qquad\qquad\quad \text{and } \mathit{Child}^{\mathcal{I}} = \{u, v\} \\ \mathit{Affects}(\mathit{JRA}, x)^{\mathcal{I},\mathbf{a}} &= \text{true} \quad \text{since} \quad \mathit{JRA}^{\mathcal{I},\mathbf{a}} = w, \quad x^{\mathcal{I},\mathbf{a}} = u \\ &\qquad\qquad\quad \text{and } \mathit{Affects}^{\mathcal{I}} = \{\langle w, u \rangle\} \end{aligned}$$

Propositional connectives are interpreted as usual:

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

Formula Evaluation

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

Existential quantifiers:

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

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

Universal quantifiers:

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

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

Evaluation of Sentences

For interpreting sentences, assignments are irrelevant.

Consider the sentence

$$\forall x.\forall y.((\mathit{JuvDisease}(x) \land \mathit{Affects}(x,y)) \rightarrow (\mathit{Child}(y) \lor \mathit{Teenager}(y)))$$

and the interpretation $\mathcal{I}$ given as follows:

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

$$\mathit{JuvDisease}^{\mathcal{I}} = \{u\} \qquad \mathit{Child}^{\mathcal{I}} = \{w\}$$

$$\mathit{Teenager}^{\mathcal{I}} = \emptyset \qquad \mathit{Affects}^{\mathcal{I}} = \{\langle u, w\rangle\}$$

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 $y = v$:

$$(\mathit{JuvDisease}(u) \land \mathit{Affects}(u,v)) \rightarrow (\mathit{Child}(v) \lor \mathit{Teenager}(v))$$

$$\text{true} \land \text{false} \rightarrow \text{false} \lor \text{false}$$

$$\text{true}$$

Propositional vs FOL Interpretations

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

$$JuvDisease \to AffectsChild \lor AffectsTeenager \qquad (PL)$$

$$\forall x.(\forall y.(JuvDisease(x) \land Affects(x,y) \to Child(y) \lor Teenager(y))) \qquad (FOL)$$

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:

Problem

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

Problem

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

Problem

Equivalence: An instance is a pair of (sets of) sentence(s) X, Y.
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).
Assume 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

$\mathit{Child}(\mathit{JohnSmith})$	John Smith is a child
$\mathit{JuvenileArthritis}(\mathit{JRA})$	JRA is a juvenile arthritis
$\neg\mathit{Affects}(\mathit{JRA}, \mathit{MaryJones})$	Mary Jones not affected by JRA

Usually data consists of (possibly negated) atoms.

But data can also reflect more complex information:

$\mathit{Child}(\mathit{JohnSmith}) \vee \mathit{Child}(\mathit{MaryJones})$ John is a child 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: $\forall x.(\mathit{JuvArthritis}(x) \rightarrow \mathit{Arthritis}(x))$
- Full definitions: $\forall x.(\mathit{JuvArthritis}(x) \leftrightarrow \mathit{Arthritis}(x) \wedge \mathit{JuvDisease}(x))$
- Disjointness statements: $\forall x.(\mathit{Child}(x) \rightarrow \neg \mathit{Adult}(x))$
- Covering statements: $\forall x.(\mathit{Person}(x) \rightarrow \mathit{Adult}(x) \vee \mathit{Child}(x) \vee \mathit{Teenager}(x))$
- Type restrictions: $\forall x.(\forall y.(\mathit{Affects}(x,y) \rightarrow \mathit{Arthritis}(x) \wedge \mathit{Person}(y)))$
- Other general statements:
  $\forall x.(\forall y.(\mathit{JuvDisease}(x) \wedge \mathit{Affects}(x,y) \rightarrow \mathit{Child}(y) \vee \mathit{Teenager}(y)))$

Data vs Terminological Knowledge

- The Data describe 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:

$$\forall y.(\mathit{Affects}(\mathit{JRA}, y) \rightarrow \mathit{Child}(y) \vee \mathit{Teenager}(y))$$

$$\begin{aligned}\forall x.(\mathit{Cont}(x) \rightarrow (x = \mathit{Eur}) \vee (x = \mathit{Asia}) \vee (x = \mathit{Amer})\\ \vee (x = \mathit{Afr}) \vee (x = \mathit{Aus}) \vee (x = \mathit{Antart}))\end{aligned}$$

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

(⇒) We have said nothing about our application

(⇒) Any possible interpretation is a model

We now add to the knowledge base the axiom

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

Any interpretation $\mathcal{I}$ such that

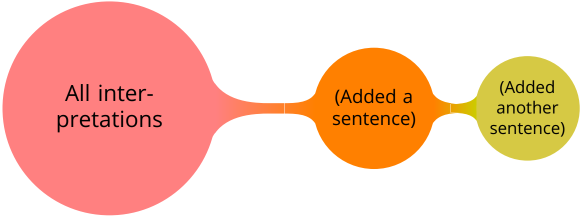
$$\mathit{JuvArthritis}^{\mathcal{I}} \not\subseteq \mathit{Arthritis}^{\mathcal{I}} \cap \mathit{JuvDisease}^{\mathcal{I}}$$

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



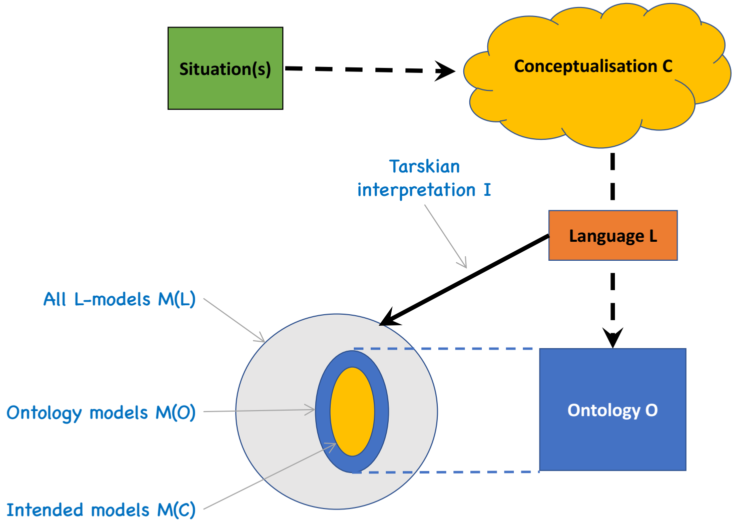
By adding FOL statements to a knowledge base we gain knowledge:

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

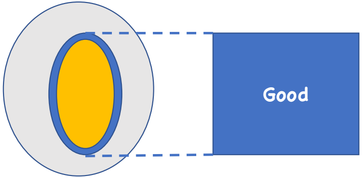
Two special cases:

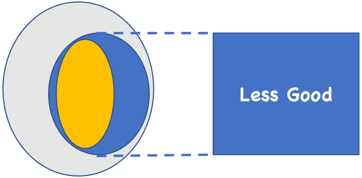
- New sentence entailed by previous ones: models stay the same
  Redundant knowledge
- Knowledge base becomes unsatisfiable: no models, everything follows
  Meaningless knowledge (error in the modeling)

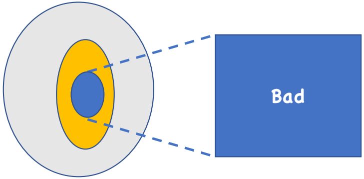
Ontological Modelling

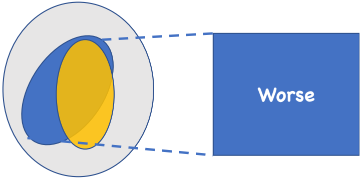


Ontological Modelling

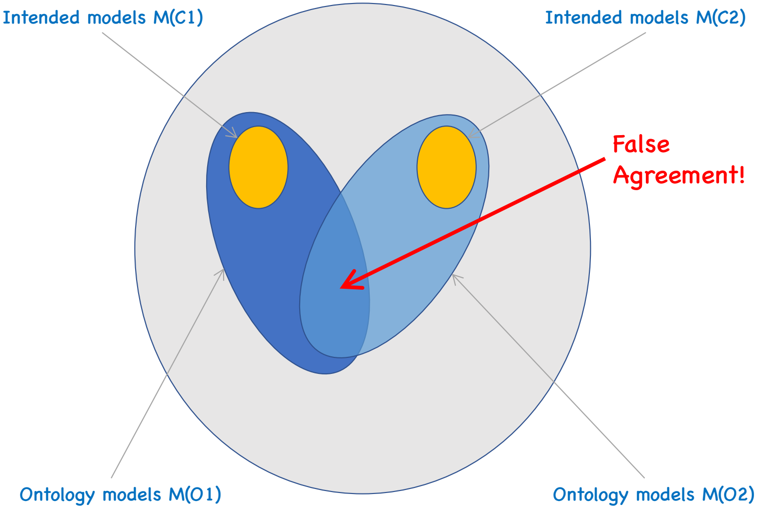








Ontological Modelling



The Role of Reasoning

Why are reasoning problems (satisfiability, entailment) useful?

1. Detect errors
   - $\Rightarrow$ Knowledge base becomes unsatisfiable
   - $\Rightarrow$ We get an unintuitive (and “wrong”) entailment
   - $\Rightarrow$ We don’t get an intuitive (and “right”) entailment
2. Discover new knowledge
   - $\Rightarrow$ Things we weren’t aware we knew
3. Richer query answers
   - $\Rightarrow$ 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 vs Complexity

Theorem

FOL satisfiability is an undecidable problem, i.e. there is no procedure that given any set $\mathcal{S}$ of first order sentences:

1. always terminates,
2. returns true if and only if $\mathcal{S}$ 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 p)
- Vague knowledge (Ian is tall)
- ...

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