Non-Classical Knowledge Representation and Reasoning

Italian National PhD Course on AI, 2024

Umberto Straccia & Giovanni Casini

CNR - ISTI, Pisa, Italy
http://www.straccia.info
{umberto.straccia, giovanni.casini}@isti.cnr.it

Outline

Classical Logics and Knowledge Representation and Reasoning (KRR)

Propositional Logic

Propositional Logic: Basic Ideas

The elementary building blocks of propositional logic are

- atomic propositions (or simply atoms) that cannot be decomposed any further: E.g.,
  - "The block is red"
  - "It is raining"
- logical connectives "and", "or", "not", by which we can build propositional formulas

Propositional Logic: syntax

Atomic Propositions

- $\bot$ (denoting false)
- $\top$ (denoting true)
- Any letter of the alphabet, e.g.: p
- Any letter of the alphabet with a numeric subscript and/or superscript, e.g.: q_4, p^7, r_2'
- Any alphanumeric string, e.g.: “Tom is the driver”

is an atomic proposition (or simply and atom)

Well-Formed Propositions (WFPs)

1. Every atomic proposition is a wfp
2. If α is a wfp, then so is $(\neg\alpha)$
3. If α and β are wfps, then so are

(conjunction)	$(\alpha \wedge \beta)$	(disjunction)	$(\alpha \vee \beta)$
(implication)	$(\alpha \rightarrow \beta)$	(equivalence)	$(\alpha \leftrightarrow \beta)$

4. Nothing else is a wfp

- Parentheses may be omitted
  - we allow $(p_1 \wedge \cdots \wedge p_n)$ and $(p_1 \vee \cdots \vee p_n)$
- Square brackets may be used instead of parentheses
- The symbols $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ are called logical connectives

Examples of (WFPs)

$$((p \wedge (q \vee c)) \rightarrow d)$$

("Betty drives Tom" $\rightarrow$ ($\neg$ "Tom is the driver"))

Summary: Syntax of Propositional Logic

Countable alphabet Σ of atomic propositions: $a, b, c, \ldots$

$$
\begin{array}{lcll}
\alpha, \beta & \longrightarrow & a & \text{(atom)} \\
 & \mid & \bot & \text{(false)} \\
 & \mid & \top & \text{(true)} \\
 & \mid & (\neg \alpha) & \text{(negation)} \\
 & \mid & (\alpha \wedge \beta) & \text{(conjunction)} \\
 & \mid & (\alpha \vee \beta) & \text{(disjunction)} \\
 & \mid & (\alpha \rightarrow \beta) & \text{(implication)} \\
 & \mid & (\alpha \leftrightarrow \beta) & \text{(equivalence)}
\end{array}
$$

Atom : atomic proposition

Literal : atomic proposition or negated atomic proposition (e.g., a, $\neg b$)

Semantics: Intuition

- Atomic statements can be true (T) or false (F)
- The truth value of formulas is determined by the truth values of the atoms

Example: $(a \vee b) \wedge c$

- If a and b are false and c is true, then the formula is false
- If a and c are true, then the formula is true

Semantics: formally

- A truth value assignment (or interpretation) of the atoms in Σ is a function $\mathcal{I}$:

$$\mathcal{I} : \Sigma \rightarrow \{\mathsf{T}, \mathsf{F}\}$$

- Instead of $\mathcal{I}(a)$ we also write $a^{\mathcal{I}}$
- A formula α is satisfied by an interpretation $\mathcal{I}$, denoted $\mathcal{I} \models \alpha$ iff

$$\begin{array}{lll}
\mathcal{I} \models \top & & \\
\mathcal{I} \not\models \bot & & \\
\mathcal{I} \models a & \text{iff} & a^{\mathcal{I}} = \mathsf{T} \\
\mathcal{I} \models \neg\alpha & \text{iff} & \mathcal{I} \not\models \alpha \\
\mathcal{I} \models \alpha \wedge \beta & \text{iff} & \mathcal{I} \models \alpha \text{ and } \mathcal{I} \models \beta \\
\mathcal{I} \models \alpha \vee \beta & \text{iff} & \mathcal{I} \models \alpha \text{ or } \mathcal{I} \models \beta \\
\mathcal{I} \models \alpha \rightarrow \beta & \text{iff} & \text{if } \mathcal{I} \models \alpha \text{ then } \mathcal{I} \models \beta \\
\mathcal{I} \models \alpha \leftrightarrow \beta & \text{iff} & \mathcal{I} \models \alpha \text{ if and only if } \mathcal{I} \models \beta
\end{array}$$

Example

- Consider the formula α

$$(a \lor b) \land c$$

- Let $\mathcal{I}_1$ be the interpretation

$$\begin{array}{lcl} a^{\mathcal{I}_1} & = & \mathrm{T} \\ b^{\mathcal{I}_1} & = & \mathrm{F} \\ c^{\mathcal{I}_1} & = & \mathrm{T} \end{array}$$

then $\mathcal{I}_1 \models \alpha$

$$\begin{array}{lll} \mathcal{I}_1 \models (a \lor b) \land c & \text{iff} & \mathcal{I}_1 \models (a \lor b) \text{ and } \mathcal{I}_1 \models c \\ & \text{iff} & \mathcal{I}_1 \models (a \lor b) \text{ and } c^{\mathcal{I}_1} = \mathrm{T} \\ & \text{iff} & (\mathcal{I}_1 \models a \text{ or } \mathcal{I}_1 \models b) \text{ and } c^{\mathcal{I}_1} = \mathrm{T} \\ & \text{iff} & (a^{\mathcal{I}_1} = \mathrm{T} \text{ or } b^{\mathcal{I}_1} = \mathrm{T}) \text{ and } c^{\mathcal{I}_1} = \mathrm{T} \end{array}$$

Example

- Consider the formula α

$$(a \vee b) \wedge c$$

- Let $\mathcal{I}_2$ be the interpretation

$$\begin{aligned} a^{\mathcal{I}_2} &= \mathsf{F} \\ b^{\mathcal{I}_2} &= \mathsf{F} \\ c^{\mathcal{I}_2} &= \mathsf{T} \end{aligned}$$

then $\mathcal{I}_2 \not\models \alpha$

Truth Tables

The truth of a formula γ in an interpretation $\mathcal{I}$ (denoted $\gamma^{\mathcal{I}}$) can also be determined using truth tables

α	$\neg\alpha$
T	F
F	T

α	β	$\alpha \wedge \beta$
F	F	F
F	T	F
T	F	F
T	T	T

α	β	$\alpha \vee \beta$
F	F	F
F	T	T
T	F	T
T	T	T

α	β	$\alpha \rightarrow \beta$
F	F	T
F	T	T
T	F	F
T	T	T

α	β	$\alpha \leftrightarrow \beta$
F	F	T
F	T	F
T	F	F
T	T	T

Example

- Consider the formula α

$$(a \vee b) \wedge c$$

- Let $\mathcal{I}_1$ be the interpretation

$$\begin{array}{rcl} a^{\mathcal{I}_1} & = & \mathsf{T} \\ b^{\mathcal{I}_1} & = & \mathsf{F} \\ c^{\mathcal{I}_1} & = & \mathsf{T} \end{array}$$

then $\mathcal{I}_1 \models \alpha$

- In fact, $\alpha^{\mathcal{I}_1} = \mathsf{T}$

$$\begin{array}{rcl} \alpha^{\mathcal{I}_1} & = & (a^{\mathcal{I}_1} \vee b^{\mathcal{I}_1}) \wedge c^{\mathcal{I}_1} \\ & = & (\mathsf{T} \vee \mathsf{F}) \wedge \mathsf{T} \\ & = & \mathsf{T} \wedge \mathsf{T} \\ & = & \mathsf{T} \end{array}$$

Example

- Consider the formula α

$$(a \vee b) \wedge c$$

- Let $\mathcal{I}_2$ be the interpretation

$$\begin{array}{lcl} a^{\mathcal{I}_2} & = & \mathsf{F} \\ b^{\mathcal{I}_2} & = & \mathsf{F} \\ c^{\mathcal{I}_2} & = & \mathsf{T} \end{array}$$

  then $\mathcal{I}_2 \not\models \alpha$
- In fact, $\alpha^{\mathcal{I}_1} = \mathsf{F}$

$$\begin{array}{lcl} \alpha^{\mathcal{I}_2} & = & (a^{\mathcal{I}_2} \vee b^{\mathcal{I}_2}) \wedge c^{\mathcal{I}_2} \\ & = & (\mathsf{F} \vee \mathsf{F}) \wedge \mathsf{T} \\ & = & \mathsf{F} \wedge \mathsf{T} \\ & = & \mathsf{F} \end{array}$$

Semantics: Interpretations as $\{0, 1\}$-functions

- An interpretation can also be specified as a function $\mathcal{I}\colon \Sigma \to \{0, 1\}$
- The intuition is that $a^{\mathcal{I}} = 1$ means that a is True, while $a^{\mathcal{I}} = 0$ means that a is False:

$$\mathcal{I} \models a \quad \text{iff} \quad a^{\mathcal{I}} = 1$$

- The truth $\alpha^{\mathcal{I}}$ of a formula α in $\mathcal{I}$ can be established using the rules:

$$\begin{array}{lcl}
(\neg\alpha)^{\mathcal{I}} & = & 1 - \alpha^{\mathcal{I}} \\
(\alpha \lor \beta)^{\mathcal{I}} & = & \max(\alpha^{\mathcal{I}}, \beta^{\mathcal{I}}) \\
(\alpha \land \beta)^{\mathcal{I}} & = & \min(\alpha^{\mathcal{I}}, \beta^{\mathcal{I}}) \\
(\alpha \to \beta)^{\mathcal{I}} & = & \max(1 - \alpha^{\mathcal{I}}, \beta^{\mathcal{I}}) \\
(\alpha \leftrightarrow \beta)^{\mathcal{I}} & = & 1 - |\alpha^{\mathcal{I}} - \beta^{\mathcal{I}}|
\end{array}$$

Example

- Consider the formula α

$$(a \vee b) \wedge c$$

- Let $\mathcal{I}_1$ be the interpretation

$$\begin{array}{lcl} a^{\mathcal{I}_1} & = & 1 \\ b^{\mathcal{I}_1} & = & 0 \\ c^{\mathcal{I}_1} & = & 1 \end{array}$$

  then $\mathcal{I}_1 \models \alpha$
- In fact, $\alpha^{\mathcal{I}_1} = 1$

$$\begin{array}{lcl} \alpha^{\mathcal{I}_1} & = & (a^{\mathcal{I}_1} \vee b^{\mathcal{I}_1}) \wedge c^{\mathcal{I}_1} \\ & = & \min(\max(1,0),1) \\ & = & \min(1,1) \\ & = & 1 \end{array}$$

Example

- Consider the formula α

$$(a \vee b) \wedge c$$

- Let $\mathcal{I}_2$ be the interpretation

$$\begin{array}{rcl} a^{\mathcal{I}_2} & = & 0 \\ b^{\mathcal{I}_2} & = & 0 \\ c^{\mathcal{I}_2} & = & 1 \end{array}$$

  then $\mathcal{I}_2 \not\models \alpha$

- In fact, $\alpha^{\mathcal{I}_1} = 0$

$$\begin{array}{rcl} \alpha^{\mathcal{I}_2} & = & (a^{\mathcal{I}_2} \vee b^{\mathcal{I}_2}) \wedge c^{\mathcal{I}_2} \\ & = & \min(\max(0, 0), 1) \\ & = & \min(0, 1) \\ & = & 0 \end{array}$$

Semantics: Interpretations as sets

- An interpretation can also be specified as a subset of Σ, i.e. $\mathcal{I} \subseteq \Sigma$
- The intuition is that the atoms in $\mathcal{I}$ are considered True, while the others are considered False:

$$\mathcal{I} \models a \quad \text{iff} \quad a \in \mathcal{I}$$

- For instance, the interpretation $\mathcal{I}$

$$\begin{array}{rcl} a^{\mathcal{I}} & = & \mathsf{T} \\ b^{\mathcal{I}} & = & \mathsf{F} \\ c^{\mathcal{I}} & = & \mathsf{T} \end{array}$$

can be represented as

$$\mathcal{I} = \{a, b\}$$

How many interpretations do exists?

- Suppose there are has n different atoms
- Each atom is either T or F, $\rightarrow$ there are 2^n interpretations
- Example: given α as the formula $(a \vee b) \wedge c$, there are $2^3 = 8$ different interpretations for α

Interpretation	a	b	c	Binay Representation	Set Representation
$\mathcal{I}_1$	F	F	F	$\langle 0, 0, 0 \rangle$	$\emptyset$
$\mathcal{I}_2$	F	F	T	$\langle 0, 0, 1 \rangle$	$\{c\}$
$\mathcal{I}_3$	F	T	F	$\langle 0, 1, 0 \rangle$	$\{b\}$
$\mathcal{I}_4$	F	T	T	$\langle 0, 1, 1 \rangle$	$\{b, c\}$
$\mathcal{I}_5$	T	F	F	$\langle 1, 0, 0 \rangle$	$\{a\}$
$\mathcal{I}_6$	T	F	T	$\langle 1, 0, 1 \rangle$	$\{a, c\}$
$\mathcal{I}_7$	T	T	F	$\langle 1, 1, 0 \rangle$	$\{a, b\}$
$\mathcal{I}_8$	T	T	T	$\langle 1, 1, 1 \rangle$	$\{a, b, c\}$

- The interpretations correspond to all possible subsets of $\{a, b, c\}$
- Note: $\mathcal{I}_j \models \alpha$ iff $j \in \{4, 6, 8\}$

Satisfiability and Validity

- An interpretation $\mathcal{I}$ is a model of α iff $\mathcal{I} \models \alpha$
- An interpretation $\mathcal{I}$ is a model of set *KB* of formulae *KB* iff $\mathcal{I} \models \alpha$ for all $\alpha \in$ *KB*
- A formula α (a set of formulae *KB*) is
  - satisfiable, if there is some $\mathcal{I}$ that satisfies α (*KB*)
  - unsatisfiable, if α is not satisfiable
  - falsifiable, if there is some $\mathcal{I}$ that does not satisfy α
  - valid (i.e. a tautology), if every $\mathcal{I}$ is a model of α
- Two formluae α, β are logically equivalent (denoted $\alpha \equiv \beta$), if for all $\mathcal{I}$:

$$\mathcal{I} \models \alpha \text{ iff } \mathcal{I} \models \beta$$

Examples

- Satisfiable: $a \vee (b \wedge c)$
- Unsatisfiable: $(a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee \neg c)$
- Falsifiable: $a \vee (b \wedge c)$
- Valid: $(a \wedge (a \rightarrow b)) \rightarrow b$
- Logically equivalent: $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$

Some Consequences

Proposition:
- α is valid iff $\neg\alpha$ is unsatisfiable
- α is unsatisfiable iff $\neg\alpha$ is valid

Proposition: $\alpha \equiv \beta$ iff $\alpha \leftrightarrow \beta$ is valid

Proposition: If $\alpha \equiv \beta$, and δ is the result of replacing α in γ by β, then $\gamma \equiv \delta$.

Equivalences (I)

Commutativity
$$\begin{aligned} \alpha \vee \beta &\equiv \beta \vee \alpha \\ \alpha \wedge \beta &\equiv \beta \wedge \alpha \\ \alpha \leftrightarrow \beta &\equiv \beta \leftrightarrow \alpha \end{aligned}$$

Associativity
$$\begin{aligned} (\alpha \vee \beta) \vee \gamma &\equiv \alpha \vee (\beta \vee \gamma) \\ (\alpha \wedge \beta) \wedge \gamma &\equiv \alpha \wedge (\beta \wedge \gamma) \end{aligned}$$

Idempotence
$$\begin{aligned} \alpha \vee \alpha &\equiv \alpha \\ \alpha \wedge \alpha &\equiv \alpha \end{aligned}$$

Absorption
$$\begin{aligned} \alpha \vee (\alpha \wedge \beta) &\equiv \alpha \\ \alpha \wedge (\alpha \vee \beta) &\equiv \alpha \end{aligned}$$

Distributivity
$$\begin{aligned} \alpha \vee (\beta \wedge \gamma) &\equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \\ \alpha \wedge (\beta \vee \gamma) &\equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \end{aligned}$$

Equivalences (II)

Tautology	$\alpha \vee \mathsf{T}$	$\equiv$	T
	$\alpha \vee \neg\alpha$	$\equiv$	T
Unsatisfiability	$\alpha \wedge \mathsf{F}$	$\equiv$	F
	$\alpha \wedge \neg\alpha$	$\equiv$	F
Neutrality	$\alpha \wedge \mathsf{T}$	$\equiv$	α
	$\alpha \vee \mathsf{F}$	$\equiv$	α
Double Negation	$\neg\neg\alpha$	$\equiv$	α
De Morgan Law	$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha) \wedge (\neg\beta)$
	$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha) \vee (\neg\beta)$
Implication	$\alpha \to \beta$	$\equiv$	$(\neg\alpha) \vee \beta$
	$\neg(\alpha \to \beta)$	$\equiv$	$\alpha \wedge (\neg\beta)$
Equivalence	$\alpha \leftrightarrow \beta$	$\equiv$	$(\alpha \to \beta) \wedge (\beta \to \alpha)$
	$\neg(\alpha \leftrightarrow \beta)$	$\equiv$	$(\neg\alpha \wedge \beta) \vee (\neg\beta \wedge \alpha)$

Normal Forms

There exists some standardized forms of formulae:

- Negation Normal Form (NNF): only atoms can be negated
  Example: $(a \vee (\neg b)) \wedge ((\neg c) \rightarrow ((\neg b) \wedge d))$
- Conjunctive Normal Form (CNF): conjunction of disjunctions of literals (called clauses)

  $$(l_{11} \vee l_{12} \vee \ldots \vee l_{1n_1}) \wedge (l_{21} \vee l_{22} \vee \ldots \vee l_{2n_2}) \wedge \ldots \wedge (l_{m1} \vee l_{m2} \vee \ldots \vee l_{mn_m})$$

  Example: $(a \vee (\neg b)) \wedge ((\neg c) \vee (\neg b) \vee c) \wedge (c \vee a \vee (\neg d))$
- Disjunctive Normal Form (DNF): disjunction of conjunctions of literals

  $$(l_{11} \wedge l_{12} \wedge \ldots \wedge l_{1n_1}) \vee (l_{21} \wedge l_{22} \wedge \ldots \wedge l_{2n_2}) \vee \ldots \vee (l_{m1} \wedge l_{m2} \wedge \ldots \wedge l_{mn_m})$$

  Example: $(a \wedge (\neg b)) \vee ((\neg c) \wedge (\neg b) \wedge c) \vee (c \wedge a \wedge (\neg d))$

Normal Forms, cont.

- k-CNF: A CNF in which every clause has at most 3 literals
- Horn clause: clause with at most 1 atom
  Example:

  $$\neg c \vee \neg b \vee c$$

  - Horn clause may be written as

    $$a_1 \wedge \ldots \wedge a_n \rightarrow b$$

- Krom clause: clause with at most 2 literals
  Example:

  $$(\neg a \vee \neg b)$$

  - Krom clause may be written as

    $$l_1 \rightarrow l_2$$

  - Can be represented as a graph, with nodes l_i and edges $\rightarrow$

    $$l_1 \rightarrow l_2$$

Proposition

For every propositional formula there exists

- *one equivalent formula in NNF*
- *one equivalent formula in DNF*
- *one equivalent formula in CNF*
- *one equivalent formula in 3-CNF.*

Transformation into NNF

Apply recursively the following equivalences

$$
\begin{array}{lcl}
\neg\neg\alpha & \equiv & \alpha \\
\neg(\alpha \vee \beta) & \equiv & \neg\alpha \wedge \neg\beta \\
\neg(\alpha \wedge \beta) & \equiv & \neg\alpha \vee \neg\beta \\
\neg(\alpha \rightarrow \beta) & \equiv & \alpha \wedge \neg\beta \\
\neg(\alpha \leftrightarrow \beta) & \equiv & (\neg\alpha \wedge \beta) \vee (\neg\beta \wedge \alpha)
\end{array}
$$

Transformation into CNF

1. Transform into NNF; then
2. Apply recursively the following equivalences

$$\begin{array}{lcl} \alpha \vee (\beta \wedge \gamma) & \equiv & (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \\ \alpha \to \beta & \equiv & \neg\alpha \vee \beta \\ \alpha \leftrightarrow \beta & \equiv & (\alpha \to \beta) \wedge (\beta \to \alpha) \end{array}$$

Transformation into DNF

1. Transform into NNF
2. Apply recursively the following equivalences

$$
\begin{array}{lcl}
\alpha \land (\beta \lor \gamma) & \equiv & (\alpha \land \beta) \lor (\alpha \land \gamma) \\
\alpha \rightarrow \beta & \equiv & (\neg \alpha) \lor \beta \\
\alpha \leftrightarrow \beta & \equiv & (\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)
\end{array}
$$

Transformation into 3-CNF

1. Transform into CNF; then
2. Apply recursively the following equivalence ($n > 3$)

$$(l_1 \vee \ldots \vee l_n) \quad \equiv \quad (l_1 \vee l_2 \vee y) \wedge (\neg y \vee l_3 \vee \ldots \vee l_n)$$

where y is a new atom

Why Normal Forms?

- We can transform propositional formulas, in particular, we can construct their NNF, CNF, 3-CNF and DNF
- DNF tells us something as to whether a formula is satisable. If all disjuncts contain $\bot$ or complementary literals, then no model exists. Otherwise, the formula is satisfiable
- CNF tells us something as to whether a formula is a tautology. If all clauses (i.e., conjuncts) contain $\top$ or complementary literals, then the formula is a tautology. Otherwise, the formula is falsifiable

But,

- the transformation into DNF or CNF may be expensive (exponential in time/space) Example:

$$(a \land b) \lor (c \land d) \quad \overset{CNF}{\mapsto} \quad (a \lor c) \land (a \lor d) \land (b \lor c) \land (b \lor d)$$

Satisfiability of KBs

- A set *KB* of formulae is satisfied iff $\mathcal{I} \models \alpha$ for all $\alpha \in$ *KB*
- An interpretation $\mathcal{I}$ is a model of set *KB* of formulae (denoted $\mathcal{I} \models$ *KB*) iff $\mathcal{I} \models \alpha$ for all $\alpha \in$ *KB*
- A set *KB* of formulae is
  - satisfiable, if there is some $\mathcal{I}$ that satisfies *KB*
  - unsatisfiable, if *KB* is not satisfiable
- A set *KB* of formulae entails a formula α iff α is true in all models of *KB*, i.e.

$$KB \models \alpha \quad \text{iff} \quad \mathcal{I} \models \alpha \text{ for all models of } KB$$

Some Properties of Entailment

Deduction Theorem: $KB \cup \{\alpha\} \models \beta$ iff $KB \models \alpha \rightarrow \beta$

Contraposition Theorem: $KB \cup \{\alpha\} \models \beta$ iff $KB \cup \{\neg\beta\} \models \neg\alpha$

Contradiction Theorem: $KB \models \alpha$ iff $KB \cup \{\neg\alpha\}$ is unsatisfiable

Checking Entailment by Enumeration

- How can we verify whether *KB* $\models \alpha$?
- We enumerate all interpretations $\mathcal{I}$ and verify that:
  1. if $\mathcal{I}$ is a model of *KB* then $\mathcal{I}$ is also a model of α;
     or equivalently
  2. *KB* $\cup \{\neg\alpha\}$ is not satisfied by $\mathcal{I}$ (contradiction theorem)

Example

Consider $KB = \{a, a \rightarrow b\}$ and $\alpha = b$. Let us show that $KB \models \alpha$

	a	b	$a \rightarrow b$	KB	α	$KB \cup \{\neg\alpha\}$
$\mathcal{I}_1$	F	F	T	F	F	F
$\mathcal{I}_2$	F	T	T	F	T	F
$\mathcal{I}_3$	T	F	F	F	F	F
$\mathcal{I}_4$	T	T	T	T	T	F

Hence

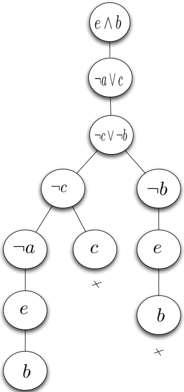
- α is true in all models of KB; or equivalently
- $KB \cup \{\neg\alpha\}$ is unsatisfiable

Therefore, $KB \models \alpha$

Checking KB Satisfiability using analytic tableaux

- A tableaux is tree, where each node is a formula α
- Tableau Inference Rules
  1. If a path contains $\alpha \wedge \beta$ then it contains α and β
  2. If a path contains $\alpha \vee \beta$ then it contains either α or β
- A clash is a path containing α and $\neg\alpha$
- A tableau is clash free if there is path not being a clash
- A tableau is complete if no rule can be applied to a path
- A KB is satisfiable iff there is a clash-free and complete tableau for it

Tableau for $e \wedge b \wedge (\neg a \vee c) \wedge (\neg c \vee \neg b)$



Checking KB Satisfiability using DPLL algorithm

- For $C = l_1 \wedge \ldots \wedge l_n$, C literal consistent iff not both l and $\neg l$ occur in C for some letter l
- literal l occurs pure in C iff $\neg l$ does not occur in C
- $C[l/\top]$ is as C in which any occurrence of l is replaced with $\top$, $C[l/\bot]$ is as C in which any occurrence of l is replaced with $\bot$, and
  - $F \vee \top$ is replaced with $\top$, $F \vee \bot$ is replaced with F

DPLL: Davis-Putnam-Logemann-Loveland

1. Let $C = \bigwedge_{F \in KB} CNF(F)$, where $CNF(F)$ transforms F into CNF
2. Return $DPLL(C)$

Function DPLL(C)

input : A formula C in CNF
output: True if C satisfiable, false otherwise

repeat
  if C *literal consistent* **then**
    return *true*
  end
  if *(C contains a conjunct that is $\bot$)* **OR** *(C not literal consistent)* **then**
    return *false*
  end
  foreach *conjunct in C being a literal l* **do** $C = C[l/\top]$;
  foreach *literal l that occurs pure in C* **do** $C = C[l/\top]$;
until *none of the previous steps is applicable*;
$l := chooseLiteral(C)$;
return $DPLL(C[l/\top])$ **OR** $DPLL(C[l/\bot])$;

Checking KB Satisfiability using Resolution

- A formula $C = C_1 \wedge \ldots \wedge C_i \wedge \ldots \wedge C_n$ in CNF, where $C_i = l_{i_1} \vee \ldots \vee l_{i_{k_i}}$ can be represented as a set of clauses, where a clause is a set of literals
  - $C_i = \{l_{i_1}, \ldots, l_{i_{k_i}}\}$, $C = \{C_1, \ldots, C_n\}$
- Resolution rule:
  from clauses $C = \{\ldots, l, \ldots\}$, $C' = \{\ldots, \neg l, \ldots\}$
  infer $C \cup C' \setminus \{l, \neg l\}$

Proposition

For a KB being a set of clauses, KB unsatisfiable iff the empty clause can be inferred.

Example

Consider $C_1 = \{a, b\}$, $C_2 = \{\neg a, c\}$, $C_3 = \{\neg b\}$, $C_4 = \{\neg c\}$

1. from C_1 and C_2 infer $C_5 = \{b, c\}$
2. from C_4 and C_3 infer $C_6 = \{c\}$
3. from C_6 and C_4 infer $C_7 = \emptyset$

Therefore, $C = \{C_1, C_2, C_3, C_4\}$ is not satisfiable

Checking KB Satisfiability using ILP

- An alternative method for satisfiability checking consists on relying on Integer Linear Programming (ILP)
- Basic idea:
  - For a formula ϕ consider a variable x_ϕ taking values in {0,1}
  - The intuition is that ϕ is true iff $x_\phi = 1$
  - Apply semantic preserving transformations, generating ILP equations
  - Check if the set of inequations has a solution

Consider a knowledge base KB

1. $EQ_{KB} = \emptyset$
2. For all $\phi \in KB$, $EQ_{KB} := EQ_{KB} \cup \{x_\phi = 1, \sigma(\phi)\}$

$$\sigma(\phi) = \begin{cases} x_p \in \{0,1\} & \text{if} \quad \phi = p \\ x_\phi = 1 - x_{\phi'}, \sigma(\phi'), x_\phi \in \{0,1\} & \text{if} \quad \phi = \neg\phi' \\ \begin{array}{l} x_\phi = \min(x_{\phi_1}, x_{\phi_2}) \\ \sigma(\phi_1), \sigma(\phi_2), x_\phi \in \{0,1\} \end{array} & \text{if} \quad \phi = \phi_1 \wedge \phi_2 \\ \begin{array}{l} x_\phi = \max(x_{\phi_1}, x_{\phi_2}) \\ \sigma(\phi_1), \sigma(\phi_2), x_\phi \in \{0,1\} \end{array} & \text{if} \quad \phi = \phi_1 \vee \phi_2 \\ \sigma(\neg\phi_1 \vee \phi_2) & \text{if} \quad \phi = \phi_1 \rightarrow \phi_2 \\ \sigma((\phi_1 \rightarrow \phi_2) \wedge (\phi_2 \rightarrow \phi_1)) & \text{if} \quad \phi = \phi_1 \leftrightarrow \phi_2 \,. \end{cases}$$

Note1: $x = \min(y, z)$ is $x \leq y, x \leq z, x \geq y + z - 1$

Note2: $x = \max(y, z)$ is $x \geq y, x \geq z, x \leq y + z$

Proposition

KB satisfiable iff EQ_{KB} has a solution.

Example

Consider $KB = \{a, a \rightarrow b\}$. Let us show that $KB \models b$

1. Consider $KB' = KB \cup \{\neg b\}$
2. We have to show that KB' not satisfiable
3. Compute $EQ_{KB'}$

$$
\begin{array}{rcl}
EQ_{KB'} & = & \{x_a = 1, x_{\neg a \vee b} = 1, x_{\neg b} = 1\} \\
 & \cup & \{x_{\neg b} = 1 - x_b, x_b \in \{0, 1\}\} \\
 & \cup & \{x_{\neg a} + x_b \geq x_{\neg a \vee b}, x_{\neg a} \leq x_{\neg a \vee b}, x_b \leq x_{\neg a \vee b}\} \\
 & \cup & \{x_{\neg a} = 1 - x_a, x_{\neg a} \in \{0, 1\}\}
\end{array}
$$

4. It can be verified that $EQ_{KB'}$ does not have a solution

Proposition

Checking the satisfiability of a propositional 3-CNF KB is a NP-complete problem.

Exercise: Expert System for Automobile Diagnosis

Knowledge base

$$(\text{GasInTank} \wedge \text{FuelLineOK}) \rightarrow \text{GasInEngine}$$
$$(\text{GasInEngine} \wedge \text{GoodSpark}) \rightarrow \text{EngineRuns}$$
$$(\text{PowerToPlugs} \wedge \text{PlugsClean}) \rightarrow \text{GoodSpark}$$
$$(\text{BatteryCharged} \wedge \text{CablesOK}) \rightarrow \text{PowerToPlugs}$$

Observed

$$\neg\text{EngineRuns}, \text{GasInTank}, \text{PlugsClean}, \text{BatteryCharged}$$

Prove

$$\neg\text{FuelLineOK} \vee \neg\text{CablesOK}$$

First-Order Logic

Pros and Cons of Propositional Logic

- We can already do a lot with propositional logic
  - Propositional logic is declarative
  - Propositional logic allows partial/disjunctive/negated information
  - Propositional logic is compositional
  - Meaning in propositional logic is context-independent
- But it is unpleasant that we cannot access the structure of atomic sentences
  - Atomic formulas of propositional logic are too atomic
  - They are just statements which my be true or false but which have no internal structure
  - Propositional logic assumes the world contains facts

First-Order Logic: Basic Ideas

- In First Order Logic (FOL) the atomic formulas are interpreted as statements about *relationships* between objects
- FOL (like natural language) assumes the world contains:

Objects:	people, houses, numbers, colors, baseball games, wars, ...
Relations:	red, round, prime, brother of, bigger than, part of, comes between, ...
Functions:	father of, best friend, one more than, plus, ...

Predicates and Constants

- Let's consider the statements:

  Mary is female
  John is male
  Mary and John are siblings

- In propositional logic the above statements are atomic propositions:

  MaryIsFemale
  JohnIsMale
  MaryAndJohnAreSiblings

- In FOL atomic statements use predicates, with constants as argument

  $$Female(mary)$$
  $$Male(john)$$
  $$Siblings(mary, john)$$

Variables and Quantifiers

- Let's consider the statements:

  Everybody is male or female
  A male is not a female

- In FOL predicates may have variables as arguments, whose value is bounded by quantifiers

$$\forall x.\text{Male}(x) \lor \text{Female}(x)$$
$$\forall x.\text{Male}(x) \to \neg\text{Female}(x)$$

- Deduction (why?):
  - Mary is not male
  - i.e., $\neg$Male(mary)

Functions

- Let's consider the statement:

  The father of a person is male

- In FOL objects of the domain may be denoted by functions applied to (other) objects:

$$\forall x.\text{Male}(\text{father}(x))$$

Syntax of FOL: atomic sentences

- Countably infinite supply of symbols (signature):
  - variable symbols: $x, y, z, \ldots$
  - n-ary function symbols: $f, g, h, \ldots$
  - individual constants: $a, b, c, \ldots$
  - n-ary predicate symbols: $P, Q, R, \ldots$

Term:

$$
\begin{array}{llll}
t & \longrightarrow & x & \text{(variable)} \\
 & | & a & \text{(constant)} \\
 & | & f(t_1, \ldots, t_n) & \text{(function application)}
\end{array}
$$

Ground Term: terms that do not contain variables

Atomic Formula:

$$
\alpha \longrightarrow P(t_1, \ldots, t_n) \quad \text{(atomic formula)}
$$

Ground Atom: Atom that does not contain variables

Examples

Term: $\text{father}(x), +(x, y)$

Ground Term: $\text{father}(\text{john}), +(2, 3)$

Atom: $\text{Loves}(\text{john}, x)$

Ground Atom: $\text{Loves}(\text{john}, \text{mary})$

Syntax of FOL

Formula:

$$
\begin{array}{llll}
\alpha, \beta & \longrightarrow & P(t_1, ..., t_n) & \text{(atomic formula)} \\
 & \mid & \bot & \text{(false)} \\
 & \mid & \top & \text{(true)} \\
 & \mid & \neg\alpha & \text{(negation)} \\
 & \mid & \alpha \wedge \beta & \text{(conjunction)} \\
 & \mid & \alpha \vee \beta & \text{(disjunction)} \\
 & \mid & \alpha \rightarrow \beta & \text{(implication)} \\
 & \mid & \alpha \leftrightarrow \beta & \text{(equivalence)} \\
 & \mid & \forall x.\alpha & \text{(universal quantification)} \\
 & \mid & \exists x.\alpha & \text{(existential quantification)}
\end{array}
$$

Ground Formula: Formula that does not contain variables

Examples:

- Everyone in Italy is smart:
  $\forall x.\mathrm{In}(x, \mathrm{italy}) \rightarrow \mathrm{Smart}(x)$
- Someone in France is smart:
  $\exists x.\mathrm{In}(x, \mathrm{france}) \wedge \mathrm{Smart}(x)$

Open, Closed and Ground Formula

- A formula with a free variable (not bounded by a quantifier) is called open

$$\forall x.[P(x,y) \leftrightarrow [\exists x.\exists z.[Q(x,y,z) \rightarrow R(x,y)]]]$$

- A formula with no free variables is called closed

$$\forall y.\forall x.[P(x,y) \leftrightarrow [\exists x.\exists z.[Q(x,y,z) \rightarrow R(x,y)]]]$$

- A formula with no variables is called ground

$$[P(a,b) \leftrightarrow [Q(a,b,c) \rightarrow R(a,b)]]$$

Semantics of FOL: intuition

- Just like in propositional logic, a (complex) FOL formula is either true or false with respect to a given interpretation
- An interpretation specifies referents for

constant symbols	$\mapsto$	objects
function symbols	$\mapsto$	functional relations
predicate symbol	$\mapsto$	relations

- An atomic sentence $P(t_1, \ldots, t_n)$ is true in a given interpretation iff the objects referred to by $t_1, \ldots, t_n$ are in the relation referred to by the predicate P
- An interpretation in which a formula is true is called a *model* for the formula

Semantics of FOL: Interpretations

- Interpretation: $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$
  - Δ is an arbitrary non-empty set of objects
  - $\cdot^{\mathcal{I}}$ is a function that maps
    - any constant *a* into an object in Δ:

      $$a^{\mathcal{I}} \in \Delta$$

    - any *n*-ary function symbol *f* to a function:

      $$f^{\mathcal{I}} : \Delta^n \to \Delta$$

    - any *n*-ary predicate symbol *P* to a relation:

      $$P^{\mathcal{I}} \subseteq \Delta^n$$

Interpretation Example

Consider

$$\forall x.\exists y.\mathrm{Loves}(x, \mathrm{friendOf}(y))$$
$$\mathrm{Loves}(a, b)$$

- Interpretation: $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$
  - $\Delta = \{\mathrm{john}, \mathrm{mary}, \mathrm{tim}, \mathrm{claudia}\}$
  - mapping of constants:

$$\begin{array}{lcl} a^{\mathcal{I}} & = & \mathrm{john} \\ b^{\mathcal{I}} & = & \mathrm{mary} \end{array}$$

  - mapping of functions:

$$friendOf^{\mathcal{I}}(d) = \begin{cases} \mathrm{mary} & \text{if} \quad d = \mathrm{john} \\ \mathrm{claudia} & \text{if} \quad d = \mathrm{mary} \\ \mathrm{john} & \text{if} \quad d = \mathrm{tim} \\ \mathrm{tim} & \text{if} \quad d = \mathrm{claudia} \end{cases}$$

  - mapping of predicates:

$$Loves^{\mathcal{I}} = \{\langle \mathrm{john}, \mathrm{mary} \rangle, \langle \mathrm{john}, \mathrm{claudia} \rangle, \langle \mathrm{mary}, \mathrm{tim} \rangle, \langle \mathrm{claudia}, \mathrm{tim} \rangle\}$$

Example (cont.)

The same interpretation can also be represented as:

- Interpretation: $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$
  - $\Delta = \{\text{john}, \text{mary}, \text{tim}, \text{claudia}\}$
  - mapping of constants:

$$\begin{array}{lcl} a^{\mathcal{I}} & = & \text{john} \\ b^{\mathcal{I}} & = & \text{mary} \end{array}$$

  - mapping of functions:

$$\{\text{friendOf}(\text{john}, \text{mary}), \text{friendOf}(\text{mary}, \text{claudia}),\\ \text{friendOf}(\text{tim}, \text{john}), \text{friendOf}(\text{claudia}, \text{tim})\}$$

  - mapping of predicates:

$$\{\text{Loves}(\text{john}, \text{mary}), \text{Loves}(\text{john}, \text{claudia}),\\ \text{Loves}(\text{mary}, \text{tim}), \text{Loves}(\text{claudia}, \text{tim})\}$$

Semantic of FOL: interpretation of ground terms

- Interpretation of ground terms

$$f(t_1, \ldots, t_n)^{\mathcal{I}} = f^{\mathcal{I}}(t_1{}^{\mathcal{I}}, \ldots, t_n{}^{\mathcal{I}})$$

Example:

$$\begin{aligned}(\text{friendOf}(\text{a}))^{\mathcal{I}} &= \text{friendOf}^{\mathcal{I}}(\text{a}^{\mathcal{I}}) \\ &= \text{friendOf}^{\mathcal{I}}(\text{john}) \\ &= \text{mary}\end{aligned}$$

Semantic of FOL: Satisfaction (Model)

- Satisfaction (model of) of ground atoms $P(t_1, \ldots, t_n)$

$$\mathcal{I} \models P(t_1, \ldots, t_n) \quad \text{iff} \quad \langle t_1{}^{\mathcal{I}}, \ldots, t_n{}^{\mathcal{I}} \rangle \in P^{\mathcal{I}}$$

Example:

- $\mathcal{I} \models \text{Loves}(\text{a}, \text{b})$

$$\begin{aligned} \mathcal{I} \models \text{Loves}(\text{a}, \text{b}) \quad & \text{iff} \quad \langle \text{a}^{\mathcal{I}}, \text{b}^{\mathcal{I}} \rangle \in \text{Loves}^{\mathcal{I}} \\ & \text{iff} \quad \langle \text{john}, \text{mary} \rangle \in \text{Loves}^{\mathcal{I}} \end{aligned}$$

- $\mathcal{I} \not\models \text{Loves}(\text{b}, \text{a})$

$$\begin{aligned} \mathcal{I} \not\models \text{Loves}(\text{b}, \text{a}) \quad & \text{iff} \quad \langle \text{b}^{\mathcal{I}}, \text{a}^{\mathcal{I}} \rangle \notin \text{Loves}^{\mathcal{I}} \\ & \text{iff} \quad \langle \text{mary}, \text{john} \rangle \notin \text{Loves}^{\mathcal{I}} \end{aligned}$$

Semantic of FOL: Variable Assignments

- An Interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ maps
  - a variable x into an object in Δ:

$$x^{\mathcal{I}} \in \Delta$$

- Let x be a variable and let $d \in \Delta$ be an object. Then

$$\mathcal{I}_x^d$$

is an interpretation, which is as $\mathcal{I}$, except that x is mapped into d: i.e.

$$z^{\mathcal{I}_x^d} = \begin{cases} z^{\mathcal{I}} & \text{if} \quad z \neq x \\ d & \text{if} \quad z = x \end{cases}$$

For instance,

$$\begin{aligned} a^{\mathcal{I}_x^d} &= a^{\mathcal{I}} \\ f^{\mathcal{I}_x^d} &= f^{\mathcal{I}} \\ P^{\mathcal{I}_x^d} &= P^{\mathcal{I}} \end{aligned}$$

Interpretation Example

Consider: Loves(x, y)

- Interpretation: $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$
  - $\Delta = \{\text{john}, \text{mary}, \text{tim}, \text{claudia}\}$
  - mapping of variables:

$$\begin{aligned} x^{\mathcal{I}} &= \text{john} \\ y^{\mathcal{I}} &= \text{mary} \end{aligned}$$

  - mapping of predicates:

$$\text{Loves}^{\mathcal{I}} = \{\langle \text{john}, \text{mary} \rangle, \langle \text{mary}, \text{tim} \rangle\}$$

- $\mathcal{I} \models \text{Loves}(x, y)$

$$\begin{aligned} \mathcal{I} \models \text{Loves}(x, y) \quad & \text{iff} \quad \langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in \text{Loves}^{\mathcal{I}} \\ & \text{iff} \quad \langle \text{john}, \text{mary} \rangle \in \text{Loves}^{\mathcal{I}} \end{aligned}$$

- $\mathcal{I}_y^{\text{claudia}} \not\models \text{Loves}(x, y)$

$$\begin{aligned} \mathcal{I}_y^{\text{claudia}} \not\models \text{Loves}(x, y) \quad & \text{iff} \quad \langle x^{\mathcal{I}_y^{\text{claudia}}}, y^{\mathcal{I}_y^{\text{claudia}}} \rangle \notin \text{Loves}^{\mathcal{I}_y^{\text{claudia}}} \\ & \text{iff} \quad \langle x^{\mathcal{I}}, \text{claudia} \rangle \notin \text{Loves}^{\mathcal{I}} \\ & \text{iff} \quad \langle \text{john}, \text{claudia} \rangle \notin \text{Loves}^{\mathcal{I}} \end{aligned}$$

Semantics of FOL: Satisfiability of formulae

- An interpretation $\mathcal{I}$ satisfies (is a model of) a formula α (α is true in $\mathcal{I}$), denoted $\mathcal{I} \models \alpha$ iff:

$$
\begin{array}{rcl}
\mathcal{I} \models P(t_1, \ldots, t_n) & \text{iff} & \langle t_1{}^{\mathcal{I}}, \ldots, t_n{}^{\mathcal{I}} \rangle \in P^{\mathcal{I}} \\
\mathcal{I} \models \neg\alpha & \text{iff} & \mathcal{I} \not\models \alpha \\
\mathcal{I} \models \alpha \wedge \beta & \text{iff} & \mathcal{I} \models \alpha \text{ and } \mathcal{I} \models \beta \\
\mathcal{I} \models \alpha \vee \beta & \text{iff} & \mathcal{I} \models \alpha \text{ or } \mathcal{I} \models \beta \\
\mathcal{I} \models \alpha \rightarrow \beta & \text{iff} & \mathcal{I} \models \neg\alpha \vee \beta \\
\mathcal{I} \models \alpha \leftrightarrow \beta & \text{iff} & \mathcal{I} \models (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)
\end{array}
$$

Semantics of FOL: Satisfiability of formulae (cont.)

$$\mathcal{I} \models \forall x.\alpha \quad \text{iff} \quad \text{for all } d \in \Delta, \mathcal{I}_x^d \models \alpha$$

$$\mathcal{I} \models \exists x.\alpha \quad \text{iff} \quad \text{for some } d \in \Delta, \mathcal{I}_x^d \models \alpha$$

- $\mathcal{I}$ satisfies (is a model of) a set of formulae *KB* (denoted $\mathcal{I} \models KB$) iff for each $\alpha \in KB$, $\mathcal{I} \models \alpha$

Example

Interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ with

$$
\begin{array}{rcl}
\Delta & = & \{d_1, \ldots, d_n\} \text{ with } n > 1 \\
\mathrm{x}^{\mathcal{I}} & = & d_1 \quad \mathrm{y}^{\mathcal{I}} = d_2 \\
\mathrm{a}^{\mathcal{I}} & = & d_1 \quad \mathrm{b}^{\mathcal{I}} = d_1 \\
\mathrm{Block}^{\mathcal{I}} & = & \{d_1\} \\
\mathrm{Red}^{\mathcal{I}} & = & \Delta
\end{array}
$$

1. $\mathcal{I} \models \mathrm{Block(a)} \vee \neg\mathrm{Block(a)}$?
2. $\mathcal{I} \models \mathrm{Block(x)} \rightarrow \neg\mathrm{Block(y)}$?
3. $\mathcal{I} \models \forall x.\exists y.[\mathrm{Block(x)} \rightarrow \mathrm{Red(y)}]$?
4. For $KB = \{\mathrm{Block(a)}, \mathrm{Block(b)}, \forall \mathrm{x}.[\mathrm{Block(x)} \rightarrow \mathrm{Red(x)}]\}$

$$\mathcal{I} \models KB \text{ ?}$$

Satisfiability and Validity

Similarly as in propositional logic, a formula α can be satisfiable, unsatisfiable, falsifiable or valid

- α is satisfiable iff there is some model $\mathcal{I}$ of α
- α is unsatisfiable iff there is no model $\mathcal{I}$ of α
- α is falsifiable iff there is some $\mathcal{I}$ not satisfying α
- α is valid (i.e., a tautology) iff every interpretation $\mathcal{I}$ is a model of α

Equivalence

Analogously, two formulas are logically equivalent (denoted $\alpha \equiv \beta$) if for all $\mathcal{I}$ we have

$$\mathcal{I} \models \alpha \quad \text{iff} \quad \mathcal{I} \models \beta$$

Note that P(x) $\not\equiv$ P(y).
Indeed, consider Interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ with

$$\begin{array}{rcl} \Delta & = & \{d_1, d_2\} \\ \mathsf{x}^{\mathcal{I}} & = & d_1 \quad \mathsf{y}^{\mathcal{I}} = d_2 \\ \mathsf{P}^{\mathcal{I}} & = & \{d_1\} \end{array}$$

Entailment

Entailment is defined similarly as in propositional logic.

- A formula α entails a formula β (denoted $\alpha \models \beta$) iff β is true in all models of α
- A set *KB* of formulae entails a formula α (denoted *KB* $\models \alpha$) iff α is true in all models of *KB*

Proposition: *KB* $\models \alpha$ iff *KB* $\cup \{\neg\alpha\}$ is not satisfiable.

Example

$$KB = \{ \text{Human(socrates)}, \; \forall x.[\text{Human(x)} \rightarrow \text{Mortal(x)}] \}$$

$$KB \models \text{Mortal(socrates) ?}$$

Yes.

- Consider a model $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ of *KB*
- Then $\mathcal{I} \models$ Human(socrates), i.e. socrates$^{\mathcal{I}} \in$ Human$^{\mathcal{I}}$
- Then $\mathcal{I} \models \forall x.$[Human(x) $\rightarrow$ Mortal(x)], i.e. Human$^{\mathcal{I}} \subseteq$ Mortal$^{\mathcal{I}}$
- As a consequence, socrates$^{\mathcal{I}} \in$ Mortal$^{\mathcal{I}}$, i.e. $\mathcal{I} \models$ Mortal(socrates)
- Therefore, $\mathcal{I} \models$ Mortal(socrates) in any model $\mathcal{I}$ of *KB*, i.e. *KB* $\models$ Mortal(socrates)

Example

$$
\begin{aligned}
KB &= \{ \quad \text{Block(a)}, \text{Block(b)}, \\
&\qquad \forall x.\exists y.[\text{Block(x)} \rightarrow \text{Red(y)}] \ \}
\end{aligned}
$$

$$
KB \models \text{Red(b)} \ ?
$$

No.
Consider $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ with

$$
\begin{aligned}
\Delta &= \{d_1, d_2\} \\
\mathrm{a}^{\mathcal{I}} &= d_1 \quad \mathrm{b}^{\mathcal{I}} = d_2 \\
\text{Block}^{\mathcal{I}} &= \{d_1, d_2\} \\
\text{Red}^{\mathcal{I}} &= \{d_1\}
\end{aligned}
$$

Then $\mathcal{I} \models KB$, but $\mathcal{I} \not\models$ Red(b).

More Examples

- $\models \forall x.[P(x) \vee \neg P(x)]$
- $P(a) \models \exists x.P(x)$
- $\exists x.[P(x) \wedge [P(x) \rightarrow Q(x)]] \models \exists x.Q(x)$

Equality

- Equality is a special predicate
- Syntax: $t_1 = t_2$, for terms t_1 and t_2
- Semantics: $\mathcal{I} \models t_1 = t_2$ iff ${t_1}^{\mathcal{I}} = {t_2}^{\mathcal{I}}$, i.e., t_1 and t_2 refer to the same object
- Example: two humans are siblings iff they have the same parents

$$\begin{aligned}\forall x.\forall y.[\mathrm{Sibling}(x,y) \quad \leftrightarrow \quad & [\neg(x = y)\wedge \\ & \exists m.\exists f.[\neg(m = f) \wedge \mathrm{Parent}(m,x) \wedge \mathrm{Parent}(f,x)\wedge \\ & \mathrm{Parent}(m,y) \wedge \mathrm{Parent}(f,y)]]]\end{aligned}$$

Notes on Universal Quantification

- "Everyone in Italy is smart":

$$\forall x.[\mathrm{In}(x, \mathrm{italy}) \rightarrow \mathrm{Smart}(x)]$$

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

$$\forall x.[\mathrm{In}(x, \mathrm{italy}) \wedge \mathrm{Smart}(x)]$$

means "Everyone is in Italy and everyone is smart"

Notes on Existential Quantification

- "Someone in France is smart":

$$\exists x.[\mathrm{In}(x, \mathrm{france}) \land \mathrm{Smart}(x)]$$

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

$$\exists x.[\mathrm{In}(x, \mathrm{france}) \rightarrow \mathrm{Smart}(x)]$$

is true if "there is no one in France"

Properties of quantifiers

- $\forall x.\forall y.\alpha$ is the same as $\forall y.\forall x.\alpha$ (why?)
- $\exists x.\exists y.\alpha$ is the same as $\exists y.\exists x.\alpha$ (why?)
- $\exists x.\forall y.\alpha$ is not the same as $\forall y.\exists x.\alpha$ (why?)
  - $\exists \mathrm{x}.\forall \mathrm{y}.\mathrm{Loves}(\mathrm{x}, \mathrm{y})$
    "There is a person who loves everyone in the world"
  - $\forall \mathrm{y}.\exists \mathrm{x}.\mathrm{Loves}(\mathrm{x}, \mathrm{y})$
    "Everyone in the world is loved by at least one person" (not necessarily the same)
- Quantifier duality

$$\begin{aligned} \forall \mathrm{x}.\mathrm{Loves}(\mathrm{x}, \mathrm{beer}) &\equiv \neg\exists \mathrm{x}.\neg\mathrm{Loves}(\mathrm{x}, \mathrm{beer}) \\ \exists \mathrm{x}.\mathrm{Loves}(\mathrm{x}, \mathrm{spinach}) &\equiv \neg\forall \mathrm{x}.\neg\mathrm{Loves}(\mathrm{x}, \mathrm{spinach}) \end{aligned}$$

Equivalences

All propositional equivalences +

$$
\begin{array}{rcl}
(\forall x.\alpha) \wedge \beta & \equiv & \forall x.(\alpha \wedge \beta) \text{ if } x \text{ not free in } \beta \\
(\forall x.\alpha) \vee \beta & \equiv & \forall x.(\alpha \vee \beta) \text{ if } x \text{ not free in } \beta \\
(\exists x.\alpha) \wedge \beta & \equiv & \exists x.(\alpha \wedge \beta) \text{ if } x \text{ not free in } \beta \\
(\exists x.\alpha) \vee \beta & \equiv & \exists x.(\alpha \vee \beta) \text{ if } x \text{ not free in } \beta \\
(\forall x.\alpha) \wedge (\forall x.\beta) & \equiv & \forall x.(\alpha \wedge \beta) \\
(\exists x.\alpha) \vee (\exists x.\beta) & \equiv & \exists x.(\alpha \vee \beta) \\
\neg \forall x.\alpha & \equiv & \exists x.\neg \alpha \\
\neg \exists x.\alpha & \equiv & \forall x.\neg \alpha
\end{array}
$$

Note:

$$
\begin{array}{rcl}
(\forall x.\alpha) \vee (\forall x.\beta) & \not\equiv & \forall x.(\alpha \vee \beta) \\
(\exists x.\alpha) \wedge (\exists x.\beta) & \not\equiv & \exists x.(\alpha \wedge \beta)
\end{array}
$$

Equivalences (cont.)

- Let β^t_x denote the formula obtained from β by replacing all free occurrences of x with the term t
- Let $Q_i \in \{\forall, \exists\}$

$$
\begin{aligned}
(Q_1 x.\alpha) \vee (Q_2 x.\beta) &\equiv Q_1 x.Q_2 y.(\alpha \vee \beta^y_x) \text{ for new variable } y \\
(Q_1 x.\alpha) \wedge (Q_2 x.\beta) &\equiv Q_1 x.Q_2 y.(\alpha \wedge \beta^y_x) \text{ for new variable } y
\end{aligned}
$$

For instance,

$$
\begin{aligned}
(\forall x.p(x)) \vee (\forall x.q(x)) &\equiv \forall x.\forall y.(p(x) \vee q(y)) \\
(\exists x.p(x)) \wedge (\exists x.q(x)) &\equiv \exists x.\exists y.(p(x) \wedge q(y))
\end{aligned}
$$

The Prenex Normal Form

Quantifier prefix + (quantifier free) matrix

$$Q_1 x_1 . Q_2 x_2 . \dots Q_n x_n . \alpha$$

where $Q_i \in \{\forall, \exists\}$ and α does not contain any quantifier

1. Elimination of $\leftrightarrow$ and $\rightarrow$
2. Push $\neg$ inwards
3. Pull quantifiers outwards

For instance

$$\begin{array}{lcl}
\neg\forall x.((\forall y.\exists z.P(x,y,z)) \rightarrow \exists x.Q(x)) & \mapsto & \neg\forall x.(\neg(\forall y.\exists z.P(x,y,z)) \vee \exists x.Q(x)) \\
& \mapsto & \exists x.(\neg(\neg(\forall y.\exists z.P(x,y,z)) \vee \exists x.Q(x))) \\
& \mapsto & \exists x.(\neg\neg(\forall y.\exists z.P(x,y,z)) \wedge \neg\exists x.Q(x)) \\
& \mapsto & \exists x.((\forall y.\exists z.P(x,y,z)) \wedge \forall x.\neg Q(x)) \\
& \mapsto & \exists x.\forall y.\exists z.(P(x,y,z) \wedge \forall x.\neg Q(x)) \\
& \mapsto & \exists x.\forall y.\exists z.\forall u.(P(x,y,z) \wedge \neg Q(u))
\end{array}$$

Skolemization

Elimination of $\exists$ in a prenex normal form

$$\exists x.\alpha \quad \mapsto \quad \alpha_x^c \text{ for new constant } c$$
$$\forall x \exists y.\alpha \quad \mapsto \quad \forall x.\alpha_y^{f(x)} \text{ for new function symbol } f$$

For instance,

$$\exists \mathrm{x}.\forall \mathrm{y}.\exists \mathrm{z}.\forall \mathrm{u}.(\mathrm{P}(\mathrm{x},\mathrm{y},\mathrm{z}) \wedge \neg \mathrm{Q}(\mathrm{u})) \quad \mapsto \quad \forall \mathrm{y}.\exists \mathrm{z}.\forall \mathrm{u}.(\mathrm{P}(\mathrm{c},\mathrm{y},\mathrm{z}) \wedge \neg \mathrm{Q}(\mathrm{u}))$$
$$\forall \mathrm{y}.\exists \mathrm{z}.\forall \mathrm{u}.(\mathrm{P}(\mathrm{c},\mathrm{y},\mathrm{z}) \wedge \neg \mathrm{Q}(\mathrm{u})) \quad \mapsto \quad \forall \mathrm{y}.\forall \mathrm{u}.(\mathrm{P}(\mathrm{c},\mathrm{y},\mathrm{f}(\mathrm{y})) \wedge \neg \mathrm{Q}(\mathrm{u}))$$

Proposition: Let α be a proposition in prenex normal form and let sk(α) its skolemization. Then α is satisfiable iff sk(α) is satisfiable.

Hence any formula can be transformed into a satisfiability preserving form (α quantifier free):

$$\forall x_1.\forall x_2. \ldots .\forall x_n.\alpha$$

Herbrand Interpretation

Consider a formula $\beta := \forall x_1.\forall x_2.\ldots\forall x_n.\alpha$, where α is quantifier free.

Hebrand universe: the smallest set U_β of terms inductively defined as:

- if c is a constant that occurs in α then $c \in U_\beta$. If no constant occurs in α then $c \in U_\beta$ for a new constant c
- if f is an n-ary function symbol occurring in α and $t_1, \ldots, t_n \in U_\beta$, then $f(t_1, \ldots, t_n) \in U_\beta$

Hebrand base: the set B_β of ground atoms such that

- if P is an n-ary predicate symbol occurring in α and $t_1, \ldots, t_n \in U_\beta$, then $P(t_1, \ldots, t_n) \in B_\beta$

Hebrand Interpretation: any subset $\mathcal{I}$ of B_β ($A \in \mathcal{I}$ means that A is true in $\mathcal{I}$)

Herbrand Models

A Herbrand model is a Herbrand interpretion that is a model.
For instance, given

$$\beta := \forall x.\forall y.(P(f(x)) \wedge Q(g(y) \vee P(a))$$

Hebrand universe: $U_\beta = \{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), \ldots\}$

Hebrand base: $B_\beta = \{P(a), Q(a), P(f(a)), P(g(a)), Q(f(a)), Q(g(a)), \ldots\}$

Hebrand Interpretation: Examples,

$$\begin{array}{rcl} \mathcal{I}_1 & = & \{P(a)\} \\ \mathcal{I}_2 & = & \{P(g(a)), Q(f(a))\} \end{array}$$

Hebrand models: Examples,

$$\begin{array}{rcl} \mathcal{I}_1 & \models & \beta \\ \mathcal{I}_2 & \not\models & \beta \end{array}$$

Proposition: $\forall x_1.\forall x_2.\ldots.\forall x_n.\alpha$ (α quantifier free) is satisfiable iff it has a Hebrand model. Hence, any formula is satisfiable iff it has a Herbrand model.

The Conjunctive Normal Form

$\forall$ prefix + (quantifier free) matrix

$$\forall x_1.\forall x_2.\ldots.\forall x_n.(C_1 \wedge C_2 \wedge \ldots \wedge C_k)$$

where each C_j (clause) is a disjunction of literals

Proposition: Any formula can be transformed into a satisfiability preserving Conjunctive Normal Form.

1. Transform the formula into a prenex normal form
2. Apply skolemization
3. Transform the quantifier free matrix into conjunctive normal form in a similar way as for propositional logic

Excercise

$$
\begin{aligned}
KB \quad = \quad & \{Person(john), Person(andrea), Female(susan), Male(bill)\} \\
\cup \quad & \{Loves(andrea, bill), Loves(susan, andrea), HasFriend(john, susan), HasFriend(john, andrea)\} \\
\cup \quad & \{\forall x. Person(x) \leftrightarrow (Male(x) \vee Female(x), \neg\exists x. Male(x) \wedge Female(x)\}
\end{aligned}
$$

$$
KB \models \exists y \exists z. HasFriend(john, y) \wedge Female(y) \wedge Loves(y, z) \wedge Male(z) \ \ ?
$$

Introduction to Semantic Web Languages (SWLs)

The Semantic Web Family of Languages

- Semantic Web family of languages widely used to specify ontologies
- Wide variety of languages
  - RDFS: *Triple language*, -*Resource Description Framework*
    - The logical counterpart is ρdf
  - RIF: *Rule language*, -*Rule Interchange Format*,
    - Relate to the Logic Programming (LP) paradigm
  - OWL 2: *Conceptual language*, -*Ontology Web Language*
    - Relate to Description Logics (DLs)

The cases of RDF and RDFS

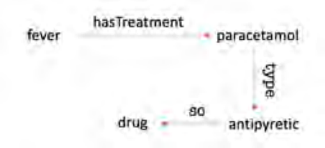
Resource Description Framework Schema (RDFS)

- RDFS: W3C standard and popular logic for KR
- Statements
  - Triples of the form (s, p, o)
  - Informally, binary predicate $p(s, o)$

$$(\texttt{fever}, \texttt{hasTreatment}, \texttt{paracetamol})$$

  - Special predicates: typing and specialisations, etc.

$$(\texttt{paracetamol}, \text{type}, \texttt{antipyretic})$$
$$(\texttt{antipyretic}, \text{sc}, \texttt{drug})$$



- *Knowledge Graphs* may be seen as a special case

The logic of RDF & RDFS: ρdf

Syntax:

- Alphabets:
  - **U** (*RDF URI references*)
  - **B** (*Blank nodes*)
  - **L** (*Literals*)
- For simplicity we will denote unions of these sets simply concatenating their names
- Terms: **UBL** ($a, b, \ldots, w$)
- Variables: **B** (x, y, z)
- Triple:

$$(s, p, o) \in \mathbf{UBL} \times \mathbf{U} \times \mathbf{UBL}$$

  - $s, o \notin \rho$df
  - s subject, p predicate, o object
- Note: e.g. $(\mathrm{type}, \mathrm{sp}, p)$ not allowed

- Graph/Knowledge Base G: set of triples τ
- Ground graph: no blank nodes, i.e. variables
- Map (or *variable assignment*):
  - μ : **UBL** $\rightarrow$ **UBL**, $\mu(t) = t$, for all $t \in$ **UL**

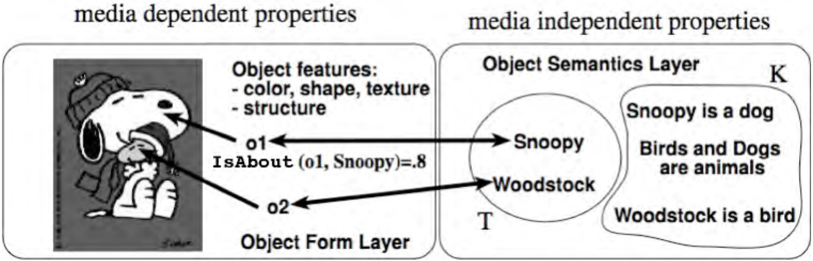
$$\mu(G) = \{(\mu(s), \mu(p), \mu(o)) \mid (s, p, o) \in G\}$$

  - Map μ from G_1 to G_2, and write $\mu : G_1 \rightarrow G_2$
    - if μ is such that $\mu(G_1) \subseteq G_2$

Example

$$
\begin{aligned}
G = \{ & (\texttt{paracetamol}, \textsf{type}, \texttt{antipyretic}), \\
& (\texttt{antipyretic}, \textsf{sc}, \texttt{drugTreatment}), \\
& (\texttt{morphine}, \textsf{type}, \texttt{opioid}), (\texttt{opioid}, \textsf{sc}, \texttt{drugTreatment}), \\
& (\texttt{drugTreatment}, \textsf{sc}, \texttt{treatment}), \\
& (\texttt{brainTumour}, \textsf{type}, \texttt{tumour}), \\
& (\texttt{hasDrugTreatment}, \textsf{sp}, \texttt{hasTreatment}), \\
& (\texttt{hasTreatment}, \textsf{dom}, \texttt{illness}), \\
& (\texttt{hasTreatment}, \textsf{range}, \texttt{treatment}), \\
& (\texttt{hasDrugTreatment}, \textsf{range}, \texttt{drugTreatment}), \\
& (\texttt{fever}, \texttt{hasDrugTreatment}, \texttt{paracetamol}) \\
& (\texttt{brainTumour}, \texttt{hasDrugTreatment}, \texttt{morphine}) \}
\end{aligned}
$$

Example (Ontology-based Multimedia Information Retrieval)



$$G = \left\{ \begin{array}{ll} (\texttt{o1}, \texttt{IsAbout}, \texttt{snoopy}) & (\texttt{o2}, \texttt{IsAbout}, \texttt{woodstock}) \\ (\texttt{snoopy}, \textsf{type}, \texttt{dog}) & (\texttt{woodstock}, \textsf{type}, \texttt{bird}) \\ (\texttt{dog}, \textsf{sc}, \texttt{animal}) & (\texttt{bird}, \textsf{sc}, \texttt{animal}) \end{array} \right\}$$

ρdf (Intentional) Semantics

ρdf interpretation:

$$\mathcal{I} = \langle \Delta_{\mathsf{R}}, \Delta_{\mathrm{DP}}, \Delta_{\mathsf{C}}, \Delta_{\mathsf{L}}, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle \ ,$$

1. Δ_{R} are the resources
2. Δ_{DP} are property names
3. $\Delta_{\mathsf{C}} \subseteq \Delta_{\mathsf{R}}$ are the classes
4. $\Delta_{\mathsf{L}} \subseteq \Delta_{\mathsf{R}}$ are the literal values and contains all the literals in $\mathbf{L} \cap V$
5. $P[\![\cdot]\!]$ is a function $P[\![\cdot]\!] \colon \Delta_{\mathrm{DP}} \to 2^{\Delta_{\mathsf{R}} \times \Delta_{\mathsf{R}}}$
6. $C[\![\cdot]\!]$ is a function $C[\![\cdot]\!] \colon \Delta_{\mathsf{C}} \to 2^{\Delta_{\mathsf{R}}}$
7. $\cdot^{\mathcal{I}}$ maps each $t \in \mathbf{UL} \cap V$ into a value $t^{\mathcal{I}} \in \Delta_{\mathsf{R}} \cup \Delta_{\mathrm{DP}}$, where $\cdot^{\mathcal{I}}$ is the identity for literals; and
8. $\cdot^{\mathcal{I}}$ maps each variable $x \in \mathbf{B}$ into a value $x^{\mathcal{I}} \in \Delta_{\mathsf{R}}$

Models

Intuitively,

- A ground triple (s, p, o) in an RDF graph G will be true under the interpretation $\mathcal{I}$ if
  - p is interpreted as a property name
  - s and o are interpreted as resources
  - the interpretation of the pair (s, o) belongs to the extension of the property assigned to p
- Blank nodes, i.e. variables, work as existential variables: a triple $((x, p, o)$ with $x \in \mathbf{B}$ would be true under $\mathcal{I}$ if
  - there exists a resource s such that (s, p, o) is true under $\mathcal{I}$

ρdf model/entailment

$\mathcal{I} \models G$ if and only if $\mathcal{I}$ satisfies conditions

Simple:

1. for each $(s, p, o) \in G$, $p^{\mathcal{I}} \in \Delta_{\mathrm{DP}}$ and $(s^{\mathcal{I}}, o^{\mathcal{I}}) \in P[\![p^{\mathcal{I}}]\!]$

Subproperty:

1. $P[\![\mathrm{sp}^{\mathcal{I}}]\!]$ is transitive over Δ_{DP}
2. if $(p, q) \in P[\![\mathrm{sp}^{\mathcal{I}}]\!]$ then $p, q \in \Delta_{\mathrm{DP}}$ and $P[\![p]\!] \subseteq P[\![q]\!]$

Subclass:

1. $P[\![\mathrm{sc}^{\mathcal{I}}]\!]$ is transitive over Δ_{C}
2. if $(c, d) \in P[\![\mathrm{sc}^{\mathcal{I}}]\!]$ then $c, d \in \Delta_{\mathrm{C}}$ and $C[\![c]\!] \subseteq C[\![d]\!]$

Typing I:

1. $x \in C[\![c]\!]$ if and only if $(x, c) \in P[\![\mathrm{type}^{\mathcal{I}}]\!]$;
2. if $(p, c) \in P[\![\mathrm{dom}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $x \in C[\![c]\!]$
3. if $(p, c) \in P[\![\mathrm{range}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $y \in C[\![c]\!]$

Typing II:

1. for each $\mathrm{e} \in \rho\mathrm{df}$, $\mathrm{e}^{\mathcal{I}} \in \Delta_{\mathrm{DP}}$;
2. if $(p, c) \in P[\![\mathrm{dom}^{\mathcal{I}}]\!]$ then $p \in \Delta_{\mathrm{DP}}$ and $c \in \Delta_{\mathrm{C}}$
3. if $(p, c) \in P[\![\mathrm{range}^{\mathcal{I}}]\!]$ then $p \in \Delta_{\mathrm{DP}}$ and $c \in \Delta_{\mathrm{C}}$
4. if $(x, c) \in P[\![\mathrm{type}^{\mathcal{I}}]\!]$ then $c \in \Delta_{\mathrm{C}}$.

$G \models H$ if and only if every model of G is also a model of H

Note

- Often $P[\![\mathrm{sp}^{\mathcal{I}}]\!]$ (resp. $C[\![\mathrm{sc}^{\mathcal{I}}]\!]$) is also *reflexive* over Δ_P (resp. Δ_C)
  - We omit this requirement and, thus, do NOT support inferences such as

$$G \models (a, \mathrm{sp}, a)$$
$$G \models (a, \mathrm{sc}, a)$$

    which anyway are of marginal interest

Example ((Model/Entailment))

$$G = \left\{ \begin{array}{ll} (\texttt{o1}, \texttt{IsAbout}, \texttt{snoopy}) & (\texttt{o2}, \texttt{IsAbout}, \texttt{woodstock}) \\ (\texttt{snoopy}, \textsf{type}, \texttt{dog}) & (\texttt{woodstock}, \textsf{type}, \texttt{bird}) \\ (\texttt{dog}, \textsf{sc}, \texttt{animal}) & (\texttt{bird}, \textsf{sc}, \texttt{animal}) \end{array} \right\}$$

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

$$\begin{array}{rcl} \Delta_R & = & \{o1, o2, snoopy, woodstock, dog, bird, animal\} \\ \Delta_P & = & \{IsAbout, \text{type}, \text{sc}\} \\ \Delta_C & = & \{dog, bird, animal\} \\ P[\![IsAbout]\!] & = & \{\langle o1, snoopy\rangle, \langle o2, woodstock\rangle\} \\ P[\![\text{type}]\!] & = & \{\langle snoopy, dog\rangle, \langle woodstock, bird\rangle, \langle snoopy, animal\rangle, \langle woodstock, animal\rangle\} \\ P[\![\text{sc}]\!] & = & \{\langle dog, animal\rangle, \langle bird, animal\rangle\} \\ C[\![dog]\!] & = & \{snoopy\} \\ C[\![bird]\!] & = & \{woodstock\} \\ C[\![animal]\!] & = & \{snoopy, woodstock\} \\ t^{\mathcal{I}} & = & t \text{ for all } t \in \mathbf{UL} \end{array}$$

$\mathcal{I} \models G$ — $\mathcal{I}$ is a model of G

$G \models (snoopy, \text{type}, animal)$ — In all models $\mathcal{I}$ of G, $\langle snoopy, animal\rangle \in P[\![\text{type}]\!]$

Deduction System for RDF & RDFS

- The system is arranged in groups of rules that captures the semantic conditions of models
- In every rule, A, B, C, X, and Y are meta-variables representing elements in **UBL**
- An instantiation of a rule is a uniform replacement of the metavariables occurring in the triples of the rule by elements of **UBL**, such that all the triples obtained after the replacement are well formed RDF triples

Deductive System for ρdf

$G \vdash H$

1. Simple:

   (a) $\frac{G}{G'}$ for a map $\mu : G' \to G$ (b) $\frac{G}{G'}$ for $G' \subseteq G$

2. Subproperty:

   (a) $\frac{(A,\text{sp},B),(B,\text{sp},C)}{(A,\text{sp},C)}$ (b) $\frac{(D,\text{sp},E),(X,D,Y)}{(X,E,Y)}$

3. Subclass:

   (a) $\frac{(A,\text{sc},B),(B,\text{sc},C)}{(A,\text{sc},C)}$ (b) $\frac{(A,\text{sc},B),(X,\text{type},A)}{(X,\text{type},B)}$

4. Typing:

   (a) $\frac{(D,\text{dom},B),(X,D,Y)}{(X,\text{type},B)}$ (b) $\frac{(D,\text{range},B),(X,D,Y)}{(Y,\text{type},B)}$

5. Implicit Typing:

   (a) $\frac{(A,\text{dom},B),(D,\text{sp},A),(X,D,Y)}{(X,\text{type},B)}$

   (b) $\frac{(A,\text{range},B),(D,\text{sp},A),(X,D,Y)}{(Y,\text{type},B)}$

Closure of G:

$$\text{Cl}(G) = \{\tau \mid G \vdash^* \tau\}$$

where $\vdash^*$ is as $\vdash$ except rule $(1a)$ is excluded

- Notion of proof:
  - Let G and H be graphs
  - Then $G \vdash H$ iff there is a sequence of graphs $P_1, \ldots, P_k$ with $P_1 = G$ and $P_k = H$, and for each j ($2 \leq j \leq k$) one of the following holds:
    1. there exists a map $\mu : P_j \to P_{j-1}$ (rule (1a));
    2. $P_j \subseteq P_{j-1}$ (rule (1b));
    3. there is an instantiation $\frac{R}{R'}$ of one of the rules (2)–(5), such that $R \subseteq P_{j-1}$ and $P_j = P_{j-1} \cup R'$.
- The sequence of rules used at each step (plus its instantiation or map), is called a proof of H from G.

Proposition (Soundness and completeness)

The RDFS proof system $\vdash$ is sound and complete for $\models$, that is, $G \vdash H$ iff $G \models H$.

Example (Proof)

$$G = \left\{ \begin{array}{ll} (o1, \mathit{IsAbout}, \mathit{snoopy}) & (o2, \mathit{IsAbout}, \mathit{woodstock}) \\ (\mathit{snoopy}, \text{type}, \mathit{dog}) & (\mathit{woodstock}, \text{type}, \mathit{bird}) \\ (\mathit{dog}, \text{sc}, \mathit{animal}) & (\mathit{bird}, \text{sc}, \mathit{animal}) \end{array} \right\}$$

Let us proof that

$$G \models (\mathit{snoopy}, \text{type}, \mathit{animal})$$

$$\begin{array}{lllll} G & \vdash & (\mathit{snoopy}, \text{type}, \mathit{dog}) & (1) & \text{Rule Simple (b)} \\ G & \vdash & (\mathit{dog}, \text{sc}, \mathit{animal}) & (2) & \text{Rule Simple (b)} \\ G & \vdash & (\mathit{snoopy}, \text{type}, \mathit{animal}) & (3) & \text{Rule SubClass (b) applied to (1) + (2)} \end{array}$$

Some ρdf Properties

1. Every ρdf-graph is satisfiable (i.e. has canonical model)
   - RDFS is paraconsistent
2. $G \vdash H$ if and only if $G \models H$
3. The closure of G is unique and $|\mathrm{Cl}(G)| \in \Theta(|G|^2)$
4. Deciding $G \models H$ is an NP-complete problem
5. If H is ground, then $G \models H$ if and only if $H \subseteq \mathrm{Cl}(G)$
6. There is no triple τ such that $\emptyset \models \tau$
7. RDFS can represent only positive statements,
   e.g. “Paracetamol is a treatment for fever”
   - RDFS with negative statements,
     see [Straccia and Casini, 2022]

     “Opioids and antipyretics are disjoint classes"
     “Radio therapies are non drug treatments"
     “Ebola has no treatment"

   - Note: “Paracetamol is not a treatment for Ebola"
     - Can not be inferred (under OWA)
     - Can be under CWA, but CWA is not acceptable for RDFS

RDFS CQ Answering

- Conjunctive query: is a Datalog-like rule of the form

$$q(\mathbf{x}) \leftarrow \exists \mathbf{y}.\tau_1, \ldots, \tau_n$$

where $\tau_1, \ldots, \tau_n$ are triples in which variables in **x** and **y** may occur (we may omit $\exists \mathbf{y}$)

- The answer set of CQ q is

$$ans(q, G) = \{\mathbf{t} \mid G \cup \{q\} \models q(\mathbf{t})\}$$

- Example:

$$q(x, y) \leftarrow (x, \texttt{creates}, y), (x, \text{type}, \texttt{Flemish}), (x, \texttt{paints}, y), (y, \texttt{exhibited}, \texttt{Uffizi})$$

"Retrieve all the artifacts x created by Flemish artists y, being exhibited at Uffizi Gallery"

- We will also write a query as

$$q(\mathbf{x}) \leftarrow \exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$$

  where $\varphi(\mathbf{x}, \mathbf{y})$ is $\tau_1, \ldots, \tau_n$
- Furthermore, $q(\mathbf{x})$ is called the head of the query, while $\exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$ is is called the body of the query
- Disjunctive query (or, *union of conjunctive queries*) **q**: is, as usual, a finite set of conjunctive queries in which all the rules have the same head
- Example

$$\begin{array}{lcl} q(x,y) & \leftarrow & (x, \mathit{creates}, y), (x, \mathrm{type}, \mathit{Flemish}), (x, \mathit{paints}, y), (y, \mathit{exhibited}, \mathit{Uffizi}) \\ q(x,y) & \leftarrow & (x, \mathit{creates}, y), (x, \mathrm{type}, \mathit{Flemish}), (x, \mathit{paints}, y), (y, \mathit{exhibited}, \mathit{Louvre}) \end{array}$$

  “Retrieve all the artifacts x created by Flemish artists y, being exhibited either at Uffizi Gallery or at the Louvre Museum"

RDFS Query Answering in practice

- A simple query answering procedure for RDFS graphs is the following:
  - Compute the closure of a graph off-line
  - Store the RDFs triples into a Relational database
  - Translate the query into a SQL statement
  - Execute the SQL statement over the relational database
- In practice, some care should be in place due to the large size of data: $\geq 10^9$ triples
- To date, several implmented systems exists

The case of OWL 2

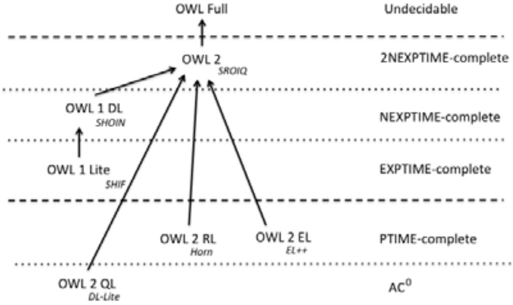
The Web Ontology Language OWL 2

- OWL 2 is a family of the object oriented languages

`class`	Person `partial` Human
	`restriction` (hasName `someValuesFrom` `String`)
	`restriction` (hasBirthPlace `someValuesFrom` Geoplace)

"The class Person *is a subclass of class* Human *and has two attributes:* hasName *having a string as value, and* hasBirthPlace *whose value is an instance of the class* Geoplace*".*

- Description Logics are the logics that stand behind OWL 2
- OWL languages differentiate in syntax and computational complexity of reasoning problems



OWL 2 Profiles

OWL 2 EL

- Useful for large size of properties and/or classes
- The EL acronym refers to the $\mathcal{EL}$ family of DLs
- Basic reasoning problems solved in Poly-time

OWL 2 QL

- Useful for very large volumes of instance data
- Conjunctive query answering via query rewriting and SQL
- OWL 2 QL relates to the DL family *DL-Lite*
- Query answering in LOGSPACE w.r.t. data complexity (size of facts)

OWL 2 RL

- Useful for scalable reasoning without sacrificing too much expressive power
- OWL 2 RL maps to Datalog (an LP language)
- Computational complexity: same as for Datalog, Poly-time w.r.t. data complexity (size of facts), EXPTIME w.r.t. combined complexity (size of knowledge base)

Description Logics (DLs)

The logics behind OWL 2 and its profiles, `http://dl.kr.org/`

- Concept/Class: are unary predicates
- Role or attribute: binary predicates
- Taxonomy: Concept and role hierarchies can be expressed
- Individual: constants
- Operators: to build complex classes out from class names

- Basic ingredients:
  - a:C, called concept assertion, meaning that individual a is an instance of concept/class C

$$\text{a:Person} \sqcap \exists\text{hasChild.Femal}$$

  - (a, b):R, called role assertion, meaning that the pair of individuals $\langle a, b\rangle$ is an instance of the property/role R

$$(\text{tom}, \text{mary})\text{:hasChild}$$

  - $C \sqsubseteq D$, called General Concept Inclusion (GCI), meaning that the class C is a subclass of class D

$$\text{Father} \sqsubseteq \mathit{Male} \sqcap \exists\text{hasChild.Person}$$

Example (Toy Example)

$$
\begin{aligned}
Sex &= Male \sqcup Female \\
Male \sqcap Female &\sqsubseteq \bot \\
Person &\sqsubseteq Human \sqcap \exists hasSex.Sex \\
MalePerson &= Person \sqcap \exists hasSex.Male \\
& functional(hasSex)
\end{aligned}
$$

$$umberto{:}Person \sqcap \exists hasSex.\neg Female$$

$$KB \models umberto{:}MalePerson$$

The DL Family

- A given DL is defined by set of concept and role forming operators
- Basic language: $\mathcal{ALC}$ ($\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplement)

Syntax				Semantics	Example
C, D	$\rightarrow$	$\top$	|	$\top(x)$	
		$\bot$	|	$\bot(x)$	
		A	|	$A(x)$	*Human*
		$C \sqcap D$	|	$C(x) \wedge D(x)$	$Human \sqcap Male$
		$C \sqcup D$	|	$C(x) \vee D(x)$	$Nice \sqcup Rich$
		$\neg C$	|	$\neg C(x)$	$\neg Meat$
		$\exists R.C$	|	$\exists y.R(x,y) \wedge C(y)$	$\exists has\_child.Blond$
		$\forall R.C$		$\forall y.R(x,y) \Rightarrow C(y)$	$\forall has\_child.Human$
$C \sqsubseteq D$				$\forall x.C(x) \Rightarrow D(x)$	$Happy\_Father \sqsubseteq Man \sqcap \exists has\_child.Female$
$a{:}C$				$C(a)$	$John{:}Happy\_Father$

DL Semantics

- Semantics is given in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where
  - $\Delta^{\mathcal{I}}$ is the domain (a non-empty set)
  - $\cdot^{\mathcal{I}}$ is an interpretation function that maps:
    - Concept (class) name A into a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$
    - Role (property) name R into a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
    - Individual name a into an element of $\Delta^{\mathcal{I}}$
  - Interpretation function $\cdot^{\mathcal{I}}$ is extended to concept expressions:

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1{}^{\mathcal{I}} \cap C_2{}^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C_1{}^{\mathcal{I}} \cup C_2{}^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}
\end{aligned}
$$

- Example: assume $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that

$$\begin{array}{rcl} \Delta^{\mathcal{I}} & = & \{a, b, c, d, e, f, 1, 2, 4, 5, 8\} \\ Person^{\mathcal{I}} & = & \{a, b, c, d\} \\ Blonde^{\mathcal{I}} & = & \{b, d\} \\ hasChild^{\mathcal{I}} & = & \{\langle a, b\rangle, \langle a, c\rangle, \langle c, d\rangle, \langle b, c\rangle\} \end{array}$$

  - What is the interpretation of $Person \sqcap \exists hasChild.Blond$?

$$\begin{array}{l} (Person \sqcap \exists hasChild.Blond)^{\mathcal{I}} \\ \quad = Person^{\mathcal{I}} \cap \{x \mid \exists y.\langle x, y\rangle \in hasChild^{\mathcal{I}} \wedge y \in Blond^{\mathcal{I}}\} \\ \quad = \{a, b, c, d\} \cap \{x \mid \exists y.\langle x, y\rangle \in \{\langle a, b\rangle, \langle a, c\rangle, \langle c, d\rangle, \langle b, c\rangle\} \wedge y \in \{b, d\}\} \\ \quad = \{a, b, c, d\} \cap \{a, c\} \\ \quad = \{a, c\} \end{array}$$

- Finally, we say that
  - $\mathcal{I}$ is a model of $C \sqsubseteq D$, written $\mathcal{I} \models C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
  - $\mathcal{I}$ is a model of $a{:}C$, written $\mathcal{I} \models a{:}C$, iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$
  - $\mathcal{I}$ is a model of $(a, b){:}R$, written $\mathcal{I} \models (a, b){:}R$, iff $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$
  - $\mathcal{I}$ is a model of $a = b$, written $\mathcal{I} \models a = b$, iff $a^{\mathcal{I}} = b^{\mathcal{I}}$
  - $\mathcal{I}$ is a model of $a \neq b$, written $\mathcal{I} \models a \neq b$, iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$

DLs and First-Order-Logic

- Satisfiability preserving $\mathcal{ALC}$ mapping to FOL: introduce
  - a unary predicate A for an atomic concept A
  - a binary predicate R for a role R
- Translate as follows

$$\begin{array}{rcl} t(\top, x) & \mapsto & \texttt{true} \\ t(\bot, x) & \mapsto & \texttt{false} \\ t(A, x) & \mapsto & A(x) \\ t(C_1 \sqcap C_2, x) & \mapsto & t(C_1, x) \wedge t(C_2, x) \\ t(C_1 \sqcup C_2, x) & \mapsto & t(C_1, x) \vee t(C_2, x) \end{array}$$

$$\begin{array}{rcl} t(\neg C, x) & \mapsto & \neg t(C, x) \\ t(\exists R.C, x) & \mapsto & \exists y.R(x, y) \wedge t(C, y) \\ t(\forall R.C, x) & \mapsto & \forall y.R(x, y) \Rightarrow t(C, y) \\ t(C \sqsubseteq D) & \mapsto & \forall x.t(C, x) \Rightarrow t(D, x) \\ t(a{:}C) & \mapsto & t(C, a) \\ t((a, b){:}R) & \mapsto & R(a, b) \end{array}$$

- Example:

$$t(HappyFather \sqsubseteq Man \sqcap \exists hasChild.Female) = \\ \forall x.HappyFather(x) \Rightarrow (Man(x) \wedge (\exists y.hasChild(x, y) \wedge Female(y)))$$

$$t(a{:}Man \sqcap \exists hasChild.Female) = \\ Man(a) \wedge (\exists y.hasChild(a, y) \wedge Female(y))$$

Note on DL Naming

$\mathcal{AL}$: $C, D \longrightarrow \top \mid \bot \mid A \mid C \sqcap D \mid \neg A \mid \exists R.\top \mid \forall R.C$

$\mathcal{C}$: Concept negation, $\neg C$. Thus, $\mathcal{ALC} = \mathcal{AL} + \mathcal{C}$

$\mathcal{S}$: Used for $\mathcal{ALC}$ with transitive roles $\mathcal{R}_+$

$\mathcal{U}$: Concept disjunction, $C_1 \sqcup C_2$

$\mathcal{E}$: Existential quantification, $\exists R.C$

$\mathcal{H}$: Role inclusion axioms, $R_1 \sqsubseteq R_2$, e.g. $is\_component\_of \sqsubseteq is\_part\_of$

$\mathcal{N}$: Number restrictions, $(\geq n\ R)$ and $(\leq n\ R)$, e.g. $(\geq 3\ has\_Child)$ (has at least 3 children)

$\mathcal{Q}$: Qualified number restrictions, $(\geq n\ R.C)$ and $(\leq n\ R.C)$, e.g. $(\leq 2\ has\_Child.Adult)$ (has at most 2 adult children)

$\mathcal{O}$: Nominals (singleton class), $\{a\}$, e.g. $\exists has\_child.\{mary\}$.
Note: $a{:}C$ equiv to $\{a\} \sqsubseteq C$ and $(a, b){:}R$ equiv to $\{a\} \sqsubseteq \exists R.\{b\}$

$\mathcal{I}$: Inverse role, R^-, e.g. $isPartOf = hasPart^-$

$\mathcal{F}$: Functional role, f, e.g. $functional(hasAge)$

$\mathcal{R}_+$: transitive role, e.g. $transitive(isPartOf)$

For instance,

$\mathcal{SHIF} = \mathcal{S} + \mathcal{H} + \mathcal{I} + \mathcal{F} = \mathcal{ALCR}_+\mathcal{HIF}$	OWL-Lite
$\mathcal{SHOIN} = \mathcal{S} + \mathcal{H} + \mathcal{O} + \mathcal{I} + \mathcal{N} = \mathcal{ALCR}_+\mathcal{HOIN}$	OWL-DL
$\mathcal{SROIQ} = \mathcal{S} + \mathcal{R} + \mathcal{O} + \mathcal{I} + \mathcal{Q} = \mathcal{ALCR}_+\mathcal{ROIN}$	OWL 2

Semantics of Additional Constructs

- $\mathcal{H}$: Role inclusion axioms, $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff ${R_1}^{\mathcal{I}} \subseteq {R_1}^{\mathcal{I}}$
- $\mathcal{N}$: Number restrictions,
  $(\geq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}}\}| \geq n\}$,
  $(\leq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}}\}| \leq n\}$
- $\mathcal{Q}$: Qualified number restrictions,
  $(\geq n\ R.C)^{\mathcal{I}} = \{x \in |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}| \geq n\}$,
  $(\leq n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}| \leq n\}$
- $\mathcal{O}$: Nominals (singleton class), $\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$
- $\mathcal{I}$: Inverse role, $(R^{-})^{\mathcal{I}} = \{\langle x, y\rangle \mid \langle y, x\rangle \in R^{\mathcal{I}}\}$
- $\mathcal{F}$: Functional role, $I \models fun(f)$ iff $\forall z \forall y \forall z$ if $\langle x, y\rangle \in f^{\mathcal{I}}$ and $\langle x, z\rangle \in f^{\mathcal{I}}$ the $y = z$
- $\mathcal{R}_{+}$: transitive role,
  $(R_{+})^{\mathcal{I}} = \{\langle x, y\rangle \mid \exists z$ such that $\langle x, z\rangle \in R^{\mathcal{I}} \wedge \langle z, y\rangle \in R^{\mathcal{I}}\}$

Basics on Concrete Domains

- Concrete domains: reals, integers, strings, ...
  $(tim, 14)$:*hasAge*
  $(sf, \text{"}SoftComputing\text{"})$:*hasAcronym*
  $(source1, \text{"}ComputerScience\text{"})$:*isAbout*
  $(service2, \text{"}InformationRetrievalTool''\text{)}$:*Matches*
  $Minor = Person \sqcap \exists hasAge. \leq_{18}$
- Semantics: a clean separation between "object" classes and concrete domains
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - Δ_D is an interpretation domain
  - Φ_D is the set of concrete domain predicates d with a predefined arity n and fixed interpretation $d^D \subseteq \Delta_D^n$
  - Concrete properties: $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_D$
- Notation: (D). E.g., $\mathcal{ALC}(D)$ is $\mathcal{ALC}$ + concrete domains

- Example: assume $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that

$$\begin{aligned} \Delta^{\mathcal{I}} &= \{a, b, c, d, e, f, 1, 2, 4, 5, 8\} \\ Person^{\mathcal{I}} &= \{a, b, c, d\} \end{aligned}$$

- Consider the following concrete domain with of some unary predicates ($n = 1$) over reals
  - $\Delta_D = \mathbb{R}$,
  - $\Phi_D = \{=_m, \geq_m, \leq_m, >_m, <_m \mid m \in \mathbb{R}\}$
  - the fixed interpretation of the predicates is

$$\begin{aligned} (=_m)^D &= \{m\} \\ (\geq_m)^D &= \{k \mid k \geq m\} \\ (\leq_m)^D &= \{k \mid k \leq m\} \end{aligned} \qquad \begin{aligned} (>_m)^D &= \{k \mid k > m\} \\ (<_m)^D &= \{k \mid k < m\} \end{aligned}$$

  - Concrete properties: $hasAge^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \mathbb{R}$

$$hasAge^{\mathcal{I}} = \{\langle a, 9\rangle, \langle c, 20\rangle, \langle b, 12\rangle\}$$

  - What is the interpretation of $Person \sqcap \exists hasAge. \leq_{18}$?

$$\begin{aligned} &(Person \sqcap \exists hasAge. \leq_{18})^{\mathcal{I}} \\ &\quad = Person^{\mathcal{I}} \cap \{x \mid \exists y \in \mathbb{R} \text{ such that } \langle x, y\rangle \in hasAge^{\mathcal{I}} \wedge y \in (\leq_{18})^{\mathcal{I}} \\ &\quad = \{a, b, c, d\} \cap \{x \mid \exists y. \langle x, y\rangle \in \{\langle a, 9\rangle, \langle c, 20\rangle, \langle b, 12\rangle\} \wedge y \leq 18\} \\ &\quad = \{a, b, c, d\} \cap \{a, b\} \\ &\quad = \{a, b\} \end{aligned}$$

DL Knowledge Base

- A DL Knowledge Base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where
  - $\mathcal{T}$ is a TBox
    - containing general inclusion axioms of the form $C \sqsubseteq D$,
    - concept definitions of the form $A = C$
    - primitive concept definitions of the form $A \sqsubseteq C$
    - role inclusions of the form $R \sqsubseteq P$
    - role equivalence of the form $R = P$
  - $\mathcal{A}$ is a ABox
    - containing assertions of the form $a{:}C$
    - containing assertions of the form $(a, b){:}R$
- An interpretation $\mathcal{I}$ is a model of $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$ iff $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$, where
  - $\mathcal{I} \models \mathcal{T}$ ($\mathcal{I}$ is a model of $\mathcal{T}$) iff $\mathcal{I}$ is a model of each element in $\mathcal{T}$
  - $\mathcal{I} \models \mathcal{A}$ ($\mathcal{I}$ is a model of $\mathcal{A}$) iff $\mathcal{I}$ is a model of each element in $\mathcal{A}$

Syntax and semantics of the DL $\mathcal{SROIQ}(\mathbf{D})$ (OWL 2)

Concepts	Syntax (C)	FOL Reading of $C(x)$
(C1)	A	$A(x)$
(C2)	$\top$	1
(C3)	$\bot$	0
(C4)	$C \sqcap D$	$C(x) \wedge D(x)$
(C5)	$C \sqcup D$	$C(x) \vee D(x)$
(C6)	$\neg C$	$\neg C(x)$
(C7)	$\forall R.C$	$\forall y.R(x,y) \rightarrow C(y)$
(C8)	$\exists R.C$	$\exists y.R(x,y) \wedge C(y)$
(C9)	$\forall T.\mathbf{d}$	$\forall v.T(x,v) \rightarrow \mathbf{d}(v)$
(C10)	$\exists T.\mathbf{d}$	$\exists v.T(x,v) \wedge \mathbf{d}(v)$
(C11)	$\{a\}$	$x = a$
(C12)	$(\geq m\ S.C)$	$\exists y_1 \ldots \exists y_m. \bigwedge_{i=1}^{m}(S(x,y_i) \wedge C(y_i)) \wedge \bigwedge_{1 \leq j < k \leq m} y_j \neq y_k$
(C13)	$(\leq m\ S.C)$	$\forall y_1 \ldots \forall y_{m+1}. \bigwedge_{i=1}^{m}(S(x,y_i) \wedge C(y_i)) \rightarrow \bigvee_{1 \leq j < k \leq m} y_j = y_k$
(C14)	$(\geq m\ T.\mathbf{d})$	$\exists v_1 \ldots \exists v_m. \bigwedge_{i=1}^{m}(T(x,v_i) \wedge \mathbf{d}(v_i)) \wedge \bigwedge_{1 \leq j < k \leq m} v_j \neq v_k$
(C15)	$(\leq m\ T.\mathbf{d})$	$\forall v_1 \ldots \forall v_{m+1}. \bigwedge_{i=1}^{m}(T(x,v_i) \wedge \mathbf{d}(v_i)) \rightarrow \bigvee_{1 \leq j < k \leq m} v_j = v_k$
(C16)	$\exists S.\mathrm{Self}$	$S(x,x)$

Roles	Syntax (R)	Semantics of $R(x,y)$
(R1)	R	$R(x,y)$
(R2)	R^{-}	$R(y,x)$
(R3)	U	1

Axiom	Syntax (E)	Semantics ($\mathcal{I}$ satisfies E if ...)
(A1)	$a{:}C$	$C(a)$
(A2)	$(a, b){:}R$	$R(a, b)$
(A3)	$(a, b){:}\neg R$	$\neg R(a, b)$
(A4)	$(a, v){:}T$	$T(a, v)$
(A5)	$(a, v){:}\neg T$	$\neg T(a, v)$
(A6)	$C \sqsubseteq D$	$\forall x.C(x) \rightarrow D(x)$
(A7)	$R_1 \ldots R_n \sqsubseteq R$	$\forall x_1 \forall x_{n+1} \exists x_2 \ldots$ $\exists x_n.(R_1(x_1, x_2) \wedge \ldots \wedge R_n(x_n, x_{n+1})) \rightarrow R(x_1, x_{n+1})$
(A8)	$T_1 \sqsubseteq T_2$	$\forall x \forall v.T_1(x, v) \rightarrow T_2(x, v)$
(A9)	trans(R)	$\forall x \forall y \forall z.R(x, z) \wedge R(z, y) \rightarrow R(x, y)$
(A10)	disj(S_1, S_2)	$\forall x \forall y.S_1(x, y) \wedge S_2(x, y) = 0$
(A11)	disj(T_1, T_2)	$\forall x \forall v.T_1(x, v) \wedge T_2(x, v) = 0$
(A12)	ref(R)	$\forall x.R(x, x)$
(A13)	irr(S)	$\forall x.\neg S(x, x)$
(A14)	sym(R)	$\forall x \forall y.R(x, y) = R(y, x)$
(A15)	asy(S)	$\forall x \forall y, S(x, y) \rightarrow \neg S(y, x)$

OWL 2 as Description Logic (excerpt)

Concept/Class constructors:

Abstract Syntax	DL Syntax	Example
Descriptions (C)		
A (URI reference)	A	Conference
owl:Thing	$\top$	
owl:Nothing	$\bot$	
intersectionOf(C_1 C_2 ...)	$C_1 \sqcap C_2$	Reference ⊓ Journal
unionOf(C_1 C_2 ...)	$C_1 \sqcup C_2$	Organization ⊔ Institution
complementOf(C)	$\neg C$	¬ MasterThesis
oneOf(o_1 ...)	$\{o_1, \ldots\}$	{"WISE","ISWC",...}
restriction(R someValuesFrom(C))	$\exists R.C$	∃parts.InCollection
restriction(R allValuesFrom(C))	$\forall R.C$	∀date.Date
restriction(R hasValue(o))	$\exists R.\{o\}$	∃date.{2005}
restriction(R minCardinality(n))	$(\geq n\ R)$	(⩾ 1 location)
restriction(R maxCardinality(n))	$(\leq n\ R)$	(⩽ 1 publisher)
restriction(U someValuesFrom(D))	$\exists U.D$	∃issue.integer
restriction(U allValuesFrom(D))	$\forall U.D$	∀name.string
restriction(U hasValue(v))	$\exists U.=_v\}$	∃series.$=_{\text{"LNCS"}}$
restriction(U minCardinality(n))	$(\geq n\ U)$	(⩾ 1 title)
restriction(U maxCardinality(n))	$(\leq n\ U)$	(⩽ 1 author)

Note: R is an abstract role, while U is a concrete property of arity two.

Axioms:

Abstract Syntax	DL Syntax	Example
Axioms		
Class(A partial $C_1 \ldots C_n$)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$Human \sqsubseteq Animal \sqcap Biped$
Class(A complete $C_1 \ldots C_n$)	$A = C_1 \sqcap \ldots \sqcap C_n$	$Man = Human \sqcap Male$
EnumeratedClass($A\ o_1 \ldots o_n$)	$A = \{o_1\} \sqcup \ldots \sqcup \{o_n\}$	$RGB = \{r\} \sqcup \{g\} \sqcup \{b\}$
SubClassOf($C_1\ C_2$)	$C_1 \sqsubseteq C_2$	
EquivalentClasses($C_1 \ldots C_n$)	$C_1 = \ldots = C_n$	
DisjointClasses($C_1 \ldots C_n$)	$C_i \sqcap C_j = \bot, i \neq j$	$Male \sqcap Female \sqsubseteq \bot$
ObjectProperty(R super (R_1)... super (R_n)	$R \sqsubseteq R_i$	$HasDaughter \sqsubseteq hasChild$
domain(C_1)...domain(C_n)	$(\geq 1\ R) \sqsubseteq C_i$	$(\geq 1\ hasChild) \sqsubseteq Human$
range(C_1)...range(C_n)	$\top \sqsubseteq \forall R.C_i$	$\top \sqsubseteq \forall hasChild.Human$
[inverseof(P)]	$R = P^-$	$hasChild = hasParent^-$
[symmetric]	$R \sqsubseteq R^-$	$similar = similar^-$
[functional]	$\top \sqsubseteq (\leq 1\ R)$	$\top \sqsubseteq (\leq 1\ hasMother)$
[Inversefunctional]	$\top \sqsubseteq (\leq 1\ R^-)$	
[Transitive])	$Tr(R)$	$Tr(ancestor)$
SubPropertyOf($R_1\ R_2$)	$R_1 \sqsubseteq R_2$	
EquivalentProperties($R_1 \ldots R_n$)	$R_1 = \ldots = R_n$	$cost = price$
AnnotationProperty(S)		

Abstract Syntax	DL Syntax	Example
DatatypeProperty(U super (U_1)... super (U_n)	$U \sqsubseteq U_i$	
domain(C_1) ... domain(C_n)	$(\geq 1\ U) \sqsubseteq C_i$	$(\geq 1\ hasAge) \sqsubseteq Human$
range(D_1) ... range(D_n)	$\top \sqsubseteq \forall U.D_i$	$\top \sqsubseteq \forall hasAge.posInteger$
[functional])	$\top \sqsubseteq (\leq 1\ U)$	$\top \sqsubseteq (\leq 1\ hasAge)$
SubPropertyOf($U_1\, U_2$)	$U_1 \sqsubseteq U_2$	$hasName \sqsubseteq hasFirstName$
EquivalentProperties($U_1 \ldots U_n$)	$U_1 = \ldots = U_n$	
Individuals		
Individual(o type (C_1)... type (C_n))	$o{:}C_i$	$tim{:}Human$
value($R_1\, o_1$) ... value($R_n o_n$)	$(o, o_i){:}R_i$	$(tim, mary){:}hasChild$
value($U_1\, v_1$) ... value($U_n v_n$)	$(o, v_1){:}U_i$	$(tim, 14){:}hasAge$
SameIndividual($o_1 \ldots o_n$)	$o_1 = \ldots = o_n$	$president\_Bush = G.W.Bush$
DifferentIndividuals($o_1 \ldots o_n$)	$o_i \neq o_j, i \neq j$	$john \neq peter$
Symbols		
Object Property R (URI reference)	R	hasChild
Datatype Property U (URI reference)	U	hasAge
Individual o (URI reference)	U	tim
Data Value v (RDF literal)	U	"International Conference on Semantic W

Basic Inference Problems (Formally)

Consistency: Check if knowledge is meaningful

- Is $\mathcal{K}$ satisfiability? $\mapsto$ Is there some model $\mathcal{I}$ of $\mathcal{K}$?
- Is C satisfiability? $\mapsto$ $C^{\mathcal{I}} \neq \emptyset$ for some some model $\mathcal{I}$ of $\mathcal{K}$?

Subsumption: structure knowledge, compute taxonomy

- $\mathcal{K} \models C \sqsubseteq D$? $\mapsto$ Is it true that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$?

Equivalence: check if two classes denote same set of instances

- $\mathcal{K} \models C = D$? $\mapsto$ Is it true that $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$?

Instantiation: check if individual a instance of class C

- $\mathcal{K} \models a{:}C$? $\mapsto$ Is it true that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$?

Retrieval: retrieve set of individuals that are instances of calss C

- Compute the set $\{a \mid \mathcal{K} \models a{:}C\}$

Reduction to Satisfiability

Problems are all reducible to KB satisfiability

Subsumption: $\mathcal{K} \models C \sqsubseteq D$ iff $\langle \mathcal{T}, \mathcal{A} \cup \{a{:}C \sqcap \neg D\}\rangle$ not satisfiable, where a is a new individual

Equivalence: $\mathcal{K} \models C = D$ iff $\mathcal{K} \models C \sqsubseteq D$ and $\mathcal{K} \models D \sqsubseteq C$

Instantiation: $\mathcal{K} \models a{:}C$ iff $\langle \mathcal{T}, \mathcal{A} \cup \{a{:}\neg C\}\rangle$ not satisfiable

Retrieval: The computation of the set $\{a \mid \mathcal{K} \models a{:}C\}$ is reducible to the instance checking problem

Non-standard Inferences

There are also some non-standard inferences.

Most Specific Concept: Given $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ and individuals $a_1, \ldots, a_n$, create a most specific concept $C = msc(KB, a_1, \ldots, a_n)$ such that $KB \models a_i{:}C$

Least Common Subsumer: Given $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ and concepts $C_1, \ldots, C_n$, create a most specific concept $C = lcs(KB, C_1, \ldots, C_n)$ such that $KB \models C_i \sqsubseteq C$

Note:

$$
\begin{aligned}
msc(KB, a_1, \ldots, a_n) &= lcs(KB, msc(KB, a_1), \ldots, msc(KB, a_n)) \\
lcs(KB, C_1, \ldots, C_n) &= lcs(KB, lcs(KB, lcs(KB, C_1, C_2), C_3), \ldots,) \ldots
\end{aligned}
$$

Reasoning in DLs

- OWL 2: tableaux based algorithms
- OWL 2 EL: structural based algorithm
- OWL 2 QL: query rewriting based algorithm
- OWL 2 RL: mapping to Datalog

OWL QL

- OWL 2 QL is related to the $DL-Lite$ DL family [Artale et al., 2009]
- $DL-Lite_{core}$, the core language for the whole family (A atomic concept, P atomic role, and P^{-} is its inverse):

$$
\begin{array}{lcl}
B & \longrightarrow & A \mid \exists R \\
C & \longrightarrow & B \mid \neg B \\
 & & \\
R & \longrightarrow & P \mid P^{-} \\
E & \longrightarrow & R \mid \neg R \,.
\end{array}
$$

- Inclusion axioms that are of the form $B \sqsubseteq C$
- $DL-Lite_{\mathcal{R}}$ from $DL-Lite_{core}$ allowing $R \sqsubseteq E$
- $DL-Lite_{\sqcap}$ is obtained from $DL-Lite_{core}$ allowing $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq C$
- $DL-Lite_{\mathcal{F}}$ is obtained by extending $DL-Lite_{core}$ with global functional roles

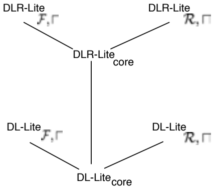


Figure: Excerpt of the DL-Lite family.

- OWL 2 RL is related to the Horn-DL family [Grosof et al., 2003, ter Horst, 2005] (A atomic concept, $m \in \{0, 1\}$, l is a value of the concrete domain, R is an object property, a individual, T is a datatype property):

$$
\begin{array}{lcl}
B & \longrightarrow & A \mid \{a\} \mid B_1 \sqcap B_2 \mid B_1 \sqcup B_2 \mid \exists R.B \mid \exists T.\mathbf{d} \\
C & \longrightarrow & A \mid C_1 \sqcap C_2 \mid \neg B \mid \forall R.C \mid \exists R.\{a\} \mid \forall T.\mathbf{d} \mid \\
 & & (\leq m\ R.B) \mid (\leq m\ R) \mid (\leq m\ T.\mathbf{d}) \\
D & \longrightarrow & \exists R.\{a\} \mid \exists T.=_l \mid D_1 \sqcap D_2 \\
R & \longrightarrow & P \mid P^-
\end{array}
$$

- Inclusion axioms have the form

$$
\begin{array}{lcl}
B & \sqsubseteq & C \\
A & = & D \\
\\
R_1 & \sqsubseteq & R_2 \\
R_1 & = & R_2
\end{array}
$$

- There are others, such as disj(B_1, B_2), dom(R, C), ran(R, C), dom(T, C), fun(R), irr(R), sym(R), asy(R), *trans*$(()R)$, disj(R_1, R_2)

OWL QL and OWL RL can be mapped into Datalog

Excerpt:

$$
\begin{array}{rcl}
\sigma(a{:}A) & \mapsto & A(a). \\
\sigma((a,b){:}R) & \mapsto & R(a,b). \\
\\
\sigma(R_1 \sqsubseteq R_2) & \mapsto & \sigma_{role}(R_2,x,y) \leftarrow \sigma_{role}(R_1,x,y) \\
\sigma(B \sqsubseteq C) & \mapsto & \sigma_h(C,x) \leftarrow \sigma_b(B,x) \\
\\
\sigma_h(\forall R.C,x) & \mapsto & \sigma_h(C,x) \leftarrow \sigma_{role}(R,x,y) \\
\\
\sigma_b(B_1 \sqcap B_2,x) & \mapsto & \sigma_b(B_1,x), \sigma_b(B_2,x) \\
\sigma_b(\exists R.B,x) & \mapsto & \sigma_{role}(R,x,y), \sigma_b(B,y) \\
\\
\sigma_h(A,x) & \mapsto & A(x) \\
\sigma_b(A,x) & \mapsto & A(x) \\
\\
\sigma_{role}(R,x,y) & \mapsto & R(x,y) \\
\sigma_{role}(R^-,x,y) & \mapsto & R(y,x)
\end{array}
$$

where x, y new variables

The case of tableau algorithms

- Tableaux algorithm deciding satisfiability
- Try to build a tree-like model $\mathcal{I}$ of the KB
- Decompose concepts C syntactically
  - Apply tableau expansion rules
  - Infer constraints on elements of model
- Tableau rules correspond to constructors in logic ($\sqcap, \sqcup, \ldots$)
  - Some rules are nondeterministic (e.g., $\sqcup, \leq$)
  - In practice, this means search
- Stop when no more rules applicable or clash occurs
  - Clash is an obvious contradiction, e.g., $A(x), \neg A(x)$
- Cycle check (blocking) may be needed for termination

Negation Normal Form (NNF)

- We have to transform concepts into Negation Normal Form: push negation inside using de Morgan' laws

$$
\begin{array}{rcl}
\neg\top & \mapsto & \bot \\
\neg\bot & \mapsto & \top \\
\neg\neg C & \mapsto & C \\
\neg(C_1 \sqcap C_2) & \mapsto & \neg C_1 \sqcup \neg C_2 \\
\neg(C_1 \sqcup C_2) & \mapsto & \neg C_1 \sqcap \neg C_2
\end{array}
$$

and

$$
\begin{array}{rcl}
\neg(\exists R.C) & \mapsto & \forall R.\neg C \\
\neg(\forall R.C) & \mapsto & \exists R.\neg C
\end{array}
$$

Completion-Forest

- This is a forest of trees, where
  - each node x is labelled with a set $\mathcal{L}(x)$ of concepts
  - each edge $\langle x, y \rangle$ is labelled with $\mathcal{L}(\langle x, y \rangle) = \{R\}$ for some role R (edges correspond to relationships between pairs of individuals)
- The forest is initialised with
  - a root node a, labelled $\mathcal{L}(x) = \emptyset$ for each individual a occurring in the KB
  - an edge $\langle a, b \rangle$ labelled $\mathcal{L}(\langle a, b \rangle) = \{R\}$ for each (a, b):R occurring in the KB
- Then, for each a:C occurring in the KB, set $\mathcal{L}(a) \rightarrow \mathcal{L}(a) \cup \{C\}$
- The algorithm expands the tree either by extending $\mathcal{L}(x)$ for some node x or by adding new leaf nodes.
- Edges are added when expanding $\exists R.C$
- A completion-forest contains a clash if, for a node x, $\{C, \neg C\} \subseteq \mathcal{L}(x)$
- If nodes x and y are connected by an edge$\langle x, y \rangle$, then y is called a successor of x and x is called a predecessor of y. Ancestor is the transitive closure of predecessor.
- A node y is called an R-successor of a node x if y is a successor of x and $\mathcal{L}(\langle x, y \rangle) = \{R\}$.
- The algorithm returns “satisfiable" if rules can be applied s.t. they yield a clash-free, complete (no more rules can be applied) completion forest

$\mathcal{ALC}$ Tableau rules without GCI's

Rule		Description
$(\sqcap)$	if 1.	$C_1 \sqcap C_2 \in \mathcal{L}(x)$ and
	2.	$\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
	then	$\mathcal{L}(x) \to \mathcal{L}(x) \cup \{C_1, C_2\}$
$(\sqcup)$	if 1.	$C_1 \sqcup C_2 \in \mathcal{L}(x)$ and
	2.	$\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$
	then	$\mathcal{L}(x) \to \mathcal{L}(x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$
$(\exists)$	if 1.	$\exists R.C \in \mathcal{L}(x)$ and
	2.	x has no R-successor y with $C \in \mathcal{L}(y)$
	then	create a new node y with $\mathcal{L}(\langle x, y\rangle) = \{R\}$ and $\mathcal{L}(y) = \{C\}$
$(\forall)$	if 1.	$\forall R.C \in \mathcal{L}(x)$ and
	2.	x has an R-successor y with $C \notin \mathcal{L}(y)$
	then	$\mathcal{L}(y) \to \mathcal{L}(y) \cup \{C\}$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D\}$$

x

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D\}$$

x

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D\}$$

$$x$$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$x$$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

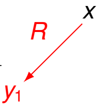
x

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$\mathcal{L}(y_1) = \{C\}$

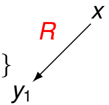


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C\}$$

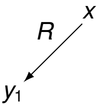


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D\}$$

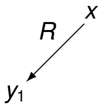


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D\}$$

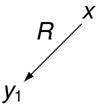


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

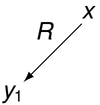
$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg C\}$$



Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$



$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg C\}$$

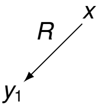
Clash

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D\}$$

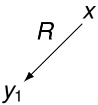


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$$

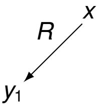


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$$

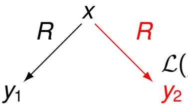


Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$

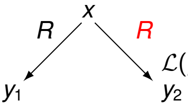


$\mathcal{L}(y_2) = \{D\}$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$



$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$$

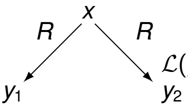
$$\mathcal{L}(y_2) = \{D\}$$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$$



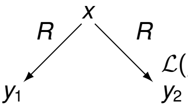
$$\mathcal{L}(y_2) = \{D, \neg C \sqcup \neg D\}$$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$$

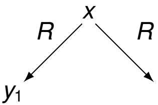


$$\mathcal{L}(y_2) = \{D, \neg C \sqcup \neg D\}$$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$



$$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$$

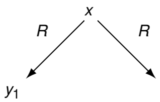
$$\mathcal{L}(y_2) = \{D, \neg C \sqcup \neg D, \neg C\}$$

Example

Is $\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D$ satisfiable? Yes.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.(\neg C \sqcup \neg D), \exists R.D\}$$

$\mathcal{L}(y_1) = \{C, \neg C \sqcup \neg D, \neg D\}$



$\mathcal{L}(y_2) = \{D, \neg C \sqcup \neg D, \neg C\}$

- Finished. No more rules applicable and the tableau is complete and clash-free
- Hence, the concept is satisfiable
- The tree corresponds to a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
  - The nodes are the elements of the domain: $\Delta^{\mathcal{I}} = \{x, y_1, y_2\}$
  - For each atomic concept A, set $A^{\mathcal{I}} = \{z \mid A \in \mathcal{L}(z)\}$
    - $C^{\mathcal{I}} = \{y_1\}$, $D^{\mathcal{I}} = \{y_2\}$
  - For each role R, set $R^{\mathcal{I}} = \{\langle x, y \rangle \mid$ there is an edge labeled R from x to $y\}$
    - $R^{\mathcal{I}} = \{\langle x, y_1 \rangle, \langle x, y_2 \rangle\}$
  - It can be shown that $x \in (\exists R.C \sqcap \forall R.(\neg C \sqcup \neg D) \sqcap \exists R.D)^{\mathcal{I}} \neq \emptyset$

Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$\mathcal{L}(x) = \{\exists R.C \sqcap \forall R.\neg C\}$

x

Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C \sqcap \forall R.\neg C\}$$

x

Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C \sqcap \forall R.\neg C\}$$

x

Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.\neg C\}$$

x

Example

Is $csomeR.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.\neg C\}$$

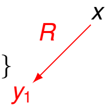
x

Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.\neg C\}$$

$$\mathcal{L}(y_1) = \{C\}$$

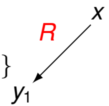


Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.\neg C\}$$

$$\mathcal{L}(y_1) = \{C\}$$

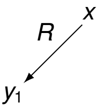


Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.\neg C\}$$

$$\mathcal{L}(y_1) = \{C, \neg C\}$$

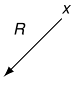


Example

Is $\exists R.C \sqcap \forall R.\neg C$ satisfiable? No.

$$\mathcal{L}(x) = \{\exists R.C, \forall R.\neg C\}$$

$\mathcal{L}(y_1) = \{C, \neg C\}$



Clash

- Finished. No more rules applicable and the tableau is complete, but not clash-free
- Hence, the concept is not satisfiable
- I.e. no model can be built, e.g.
  - $\Delta^{\mathcal{I}} = \{x, y_1\}$
  - $C^{\mathcal{I}} = \{y_1\}$
  - $R^{\mathcal{I}} = \{\langle x, y_1 \rangle\}$
  - is not a model because

$$(\exists R.C \sqcap \forall R.\neg C)^{\mathcal{I}} = (\exists R.C)^{\mathcal{I}} \cap (\forall R.\neg C)^{\mathcal{I}} = \{x\} \cap \emptyset = \emptyset$$

Soundness and Completeness

Theorem

Let $\mathcal{A}$ be an $\mathcal{ALC}$ ABox and F a completion-forest obtained by applying the tableau rules to $\mathcal{A}$. Then

1. *The rule application terminates;*
2. *If F is clash-free and complete, then F defines a (canonical) (tree) model for $\mathcal{A}$; and*
3. *If $\mathcal{A}$ has a model $\mathcal{I}$, then the rules can be applied such that they yield a clash-free and complete completion-forest.*

KBs with GCIs

- We have seen how to test the satisfiability of an ABox $\mathcal{A}$
- But, how can we check if a KB $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable with $\mathcal{T} \neq \emptyset$?
- Basic idea: since $C \sqsubseteq D$ is equivalent to $\top \sqsubseteq nnf(\neg C \sqcup D)$),
  - replace each $C \sqsubseteq D$ with its equivalent form $\top \sqsubseteq nnf(\neg C \sqcup D)$
  - use the rule: for each $\top \sqsubseteq E \in \mathcal{T}$, add E to every node
- But, termination is not guaranteed
  - E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} & = & \{umberto{:}Human\} \end{array}$$

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{\mathit{Human} \sqsubseteq \exists \mathit{hasMother}.\mathit{Human}\} \\ \mathcal{A} & = & \{\mathit{umberto}{:}\mathit{Human}\} \end{array}$$

$\mathcal{L}(\mathit{umberto}) = \{\mathit{Human}, \neg \mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}\}$ $\mathit{umberto}$

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} & = & \{umberto{:}Human\} \end{array}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \neg Human\}$ *umberto*

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{lcl} \mathcal{T} & = & \{\mathit{Human} \sqsubseteq \exists \mathit{hasMother}.\mathit{Human}\} \\ \mathcal{A} & = & \{\mathit{umberto}{:}\mathit{Human}\} \end{array}$$

$\mathcal{L}(\mathit{umberto}) = \{\mathit{Human}, \neg\mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}, \neg\mathit{Human}\}$ *umberto*

Clash

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} & = & \{umberto{:}Human\} \end{array}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human\}$ *umberto*

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{aligned} \mathcal{T} &= \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} &= \{umberto{:}Human\} \end{aligned}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$ $\quad$ *umberto*

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

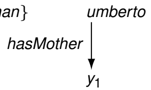
$$\begin{aligned} \mathcal{T} &= \{\mathit{Human} \sqsubseteq \exists \mathit{hasMother}.\mathit{Human}\} \\ \mathcal{A} &= \{\mathit{umberto}{:}\mathit{Human}\} \end{aligned}$$

$\mathcal{L}(\mathit{umberto}) = \{\mathit{Human}, \neg\mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}, \exists \mathit{hasMother}.\mathit{Human}\}$ $\mathit{umberto}$

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$
\begin{aligned}
\mathcal{T} &= \{\mathit{Human} \sqsubseteq \exists \mathit{hasMother}.\mathit{Human}\} \\
\mathcal{A} &= \{\mathit{umberto}{:}\mathit{Human}\}
\end{aligned}
$$

$\mathcal{L}(\mathit{umberto}) = \{\mathit{Human}, \neg\mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}, \exists \mathit{hasMother}.\mathit{Human}\}$



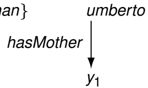
$\mathcal{L}(y_1) = \{\mathit{Human}, \neg\mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}\}$

- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\mathcal{T} = \{Human \sqsubseteq \exists hasMother.Human\}$$
$$\mathcal{A} = \{umberto{:}Human\}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human\}$

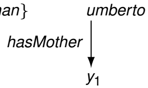


- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$
\begin{aligned}
\mathcal{T} &= \{Human \sqsubseteq \exists hasMother.Human\} \\
\mathcal{A} &= \{umberto{:}Human\}
\end{aligned}
$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$



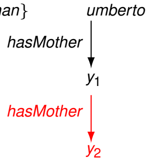
- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} & = & \{umberto{:}Human\} \end{array}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_2) = \{Human, \neg Human \sqcup \exists hasMother.Human\}$



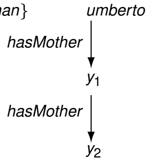
- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{aligned} \mathcal{T} &= \{\mathit{Human} \sqsubseteq \exists \mathit{hasMother}.\mathit{Human}\} \\ \mathcal{A} &= \{\mathit{umberto}{:}\mathit{Human}\} \end{aligned}$$

$\mathcal{L}(\mathit{umberto}) = \{\mathit{Human}, \neg \mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}, \exists \mathit{hasMother}.\mathit{Human}\}$

$\mathcal{L}(y_1) = \{\mathit{Human}, \neg \mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}, \exists \mathit{hasMother}.\mathit{Human}\}$

$\mathcal{L}(y_2) = \{\mathit{Human}, \neg \mathit{Human} \sqcup \exists \mathit{hasMother}.\mathit{Human}\}$



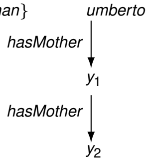
- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} & = & \{umberto{:}Human\} \end{array}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_2) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$



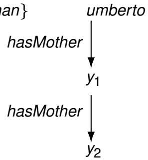
- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{array}{rcl} \mathcal{T} & = & \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} & = & \{umberto{:}Human\} \end{array}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_2) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$



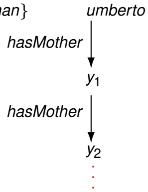
- E.g., consider $KB = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{aligned} \mathcal{T} &= \{Human \sqsubseteq \exists hasMother.Human\} \\ \mathcal{A} &= \{umberto{:}Human\} \end{aligned}$$

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_2) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$



Node Blocking in $\mathcal{ALC}$

- When creating new node, check ancestors for equal label set
- If such a node is found, new node is blocked
- No rule is applied to blocked nodes

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$ — *umberto*

↓ *hasMother*

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$ — y_1

↓ *hasMother*

$\mathcal{L}(y_2) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$ — y_2 blocked: $\mathcal{L}(y_1) = \mathcal{L}(y_2)$

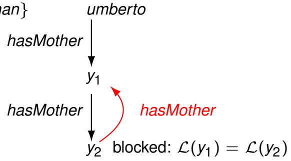
Node Blocking in $\mathcal{ALC}$

- When creating new node, check ancestors for equal label set
- If such a node is found, new node is blocked
- No rule is applied to blocked nodes

$\mathcal{L}(umberto) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_1) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$

$\mathcal{L}(y_2) = \{Human, \neg Human \sqcup \exists hasMother.Human, \exists hasMother.Human\}$



- Block represents cyclical model
  - $\Delta^{\mathcal{I}} = \{umberto, y_1, y_2\}$
  - $Human^{\mathcal{I}} = \{umberto, y_1, y_2\}$
  - $hasMother^{\mathcal{I}} = \{\langle umberto, y_1\rangle, \langle y_1, y_2\rangle, \langle y_2, y_1\rangle\}$

Blocking in $\mathcal{ALC}$

- A non-root node x is blocked if for some ancestor y, y is blocked or $\mathcal{L}(x) = \mathcal{L}(y)$, where y is not a root node.
- A blocked node x is indirectly blocked if its predecessor is blocked, otherwise it is directly blocked.
- If x is directly blocked, it has a unique ancestor y such that $\mathcal{L}(x) = \mathcal{L}(y)$
- if there existed another ancestor z such that $\mathcal{L}(x) = \mathcal{L}(z)$ then either y or z must be blocked.
- If x is directly blocked and y is the unique ancestor such that $\mathcal{L}(x) = \mathcal{L}(y)$, we will say that y blocks x

$\mathcal{ALC}$ Tableau rules with GCI's

Rule		Description
$(\sqcap)$	if 1.	$C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked and
	2.	$\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
	then	$\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$(\sqcup)$	if 1.	$C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked and
	2.	$\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$
	then	$\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$
$(\exists)$	if 1.	$\exists R.C \in \mathcal{L}(x)$, x is not blocked and
	2.	x has no R-successor y with $C \in \mathcal{L}(y)$
	then	create a new node y with $\mathcal{L}(\langle x, y \rangle) = \{R\}$ and $\mathcal{L}(y) = \{C\}$
$(\forall)$	if 1.	$\forall R.C \in \mathcal{L}(x)$, x is not indirectly blocked and
	2.	x has an R-successor y with $C \notin \mathcal{L}(y)$
	then	$\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{C\}$
$(\sqsubseteq)$	if 1.	$\top \sqsubseteq E \in \mathcal{T}$, x is not indirectly blocked and
	2.	$E \notin \mathcal{L}(x) = \emptyset$
	then	$\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{E\}$

Soundness and Completeness

Theorem

Let KB be an $\mathcal{ALC}$ KB and F a completion-forest obtained by applying the tableau rules to KB. Then

1. *The rule application terminates;*
2. *If F is clash-free and complete, then F defines a (canonical) (tree) model for KB; and*
3. *If KB has a model $\mathcal{I}$, then the rules can be applied such that they yield a clash-free and complete completion-forest.*

The case of Logic Programs (LPs)

LPs Basics, for ease, Datalog

- Predicates are *n*-ary
- Terms are variables or constants
- Facts ground atoms
  For instance,
  $$has\_parent(mary, jo)$$
- Rules are of the form
  $$P(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$$
  where
  - $\varphi(\mathbf{x}, \mathbf{y})$ is a formula built from atoms of the form $B(\mathbf{z})$ and connectors $\wedge, \vee, 0, 1$
  - $\mathbf{z}_i$ is a tuple of literals, or variables in $\mathbf{x}, \mathbf{y}$
- For instance,
  $$has\_father(x, y) \quad \leftarrow \quad has\_parent(x, y) \wedge Male(y)$$

Remark

Note that

$$has\_father(x, y) \quad \leftarrow \quad has\_parent(x, y), Male(y)$$

is the same as repplacing "," with $\wedge$

- Extensional database (EDB): set of facts
- Intentional database (IDB): set of rules
- Logic Program $\mathcal{P}$:
  - $\mathcal{P} = EDB \cup IDB$
  - No predicate symbol in *EDB* occurs in the head of a rule in *IDB*
    - The principle is that we do not allow that *IDB* may redefine the extension of predicates in *EDB*
- *EDB* is usually, stored into a relational database

LPs Semantics: FOL semantics

- $\mathcal{P}^*$ is constructed as follows:
  1. set $\mathcal{P}^*$ to the set of all ground instantiations of rules in $\mathcal{P}$
  2. replace a fact $p(\mathbf{c})$ in $\mathcal{P}^*$ with the rule $p(\mathbf{c}) \leftarrow 1$
  3. if atom A is not head of any rule in $\mathcal{P}^*$, then add $A \leftarrow 0$ to $\mathcal{P}^*$
  4. replace several rules in $\mathcal{P}^*$ having same head

$$\left.\begin{array}{c} A \leftarrow \varphi_1 \\ A \leftarrow \varphi_2 \\ \vdots \\ A \leftarrow \varphi_n \end{array}\right\} \text{with } A \leftarrow \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n$$

- Note: in $\mathcal{P}^*$ each atom $A \in B_{\mathcal{P}}$ is head of exactly one rule
- Herbrand Base of $\mathcal{P}$ is the set $B_{\mathcal{P}}$ of ground atoms
- Interpretation is a function $I : B_{\mathcal{P}} \to \{0, 1\}$
- Model $I \models \mathcal{P}$ iff for all $r \in \mathcal{P}^*$ $I \models r$, where $I \models A \leftarrow \varphi$ iff $I(\varphi) \leq I(A)$

- Entailment: for a ground atom $p(\mathbf{c})$

$$\mathcal{P} \models p(\mathbf{c}) \text{ iff all models of } \mathcal{P} \text{ satisfy } p(\mathbf{c})$$

- Least model $M_{\mathcal{P}}$ of $\mathcal{P}$ exists and is least fixed-point of

$$T_{\mathcal{P}}(I)(A) = I(\varphi), \text{ for all } A \leftarrow \varphi \in \mathcal{P}^*$$

- M can be computed as the limit of

$$\begin{array}{lcl} \mathbf{I}_0 & = & \mathbf{0} \\ \mathbf{I}_{i+1} & = & T_{\mathcal{P}}(\mathbf{I}_i) \ . \end{array}$$

Example

$$\mathcal{P} = \begin{cases} Q(x) & \leftarrow & B(x) \\ Q(x) & \leftarrow & C(x) \\ B(a) & & \\ C(b) & & \end{cases} \qquad \mathcal{P}^* = \begin{cases} Q(a) & \leftarrow & B(a) \vee C(a) \\ Q(b) & \leftarrow & B(b) \vee C(b) \\ B(a) & \leftarrow & 1 \\ C(b) & \leftarrow & 1 \end{cases}$$

$\mathbf{I}_i$	$Q(a)$	$Q(b)$	$B(a)$	$B(b)$	$C(a)$	$C(b)$
$\mathbf{I}_0$	0	0	0	0	0	0
$\mathbf{I}_1$	0	0	1	0	0	1
$\mathbf{I}_2$	1	1	1	0	0	1
$\mathbf{I}_3$	1	1	1	0	0	1

- $\mathbf{I}_2 = \mathbf{I}_3$, i.e. $T_{\mathcal{P}}(\mathbf{I}_2) = \mathbf{I}_3 = \mathbf{I}_2$
- $\mathbf{I}_2$ is least fixed-point and, thus, minimal model $M_{\mathcal{P}} = \{Q(a), Q(b), B(a), C(b)\}$

LP Query Answering

Proposition

$\mathcal{P} \models p(t_1, ..., t_n)$ *iff* $M_{\mathcal{P}} \models p(t_1, ..., t_n)$.

- As a consequence, we may restrict our attention to minimal models only
- Query: is a rule of the form

$$q(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$$

- If $\mathcal{P} \models q(\mathbf{c})$ then $\mathbf{c}$ is called an answer to q
- The answer set of q w.r.t. $\mathcal{P}$ is defined as

$$ans(\mathcal{P}, q) = \{\mathbf{c} \mid \mathcal{P} \models q(\mathbf{c})\}$$

Toy Example

$$\begin{array}{rcl} Q(x) & \leftarrow & B(x) \\ Q(x) & \leftarrow & C(x) \\ B(a) & & \\ C(b) & & \end{array}$$

$$\mathcal{P} \models Q(a) \quad \mathcal{P} \models Q(b) \qquad ans(\mathcal{P}, Q) = \{a, b\}$$

A general top-down query procedure for ground LPs

- Idea: use theory of fixed-point computation of equational systems over $\{0,1\}$
- Assign a variable x_i to an atom $A_i \in B_{\mathcal{P}}$
- Map a rule $A \leftarrow f(A_1,\ldots,A_n) \in \mathcal{P}^*$ into the equation $x_A = f(x_{A_1},\ldots,x_{A_n})$

$$p \leftarrow (q \vee r) \wedge t \text{ is mapped into } x_p = \min(\max(x_q, x_r), x_t)$$

- A LP $\mathcal{P}$ is thus mapped into the equational system, using $\mathcal{P}^*$

$$\left\{ \begin{array}{lcl} x_1 & = & f_1(x_{1_1},\ldots,x_{1_{a_1}}) \\ & \vdots & \\ x_n & = & f_n(x_{n_1},\ldots,x_{n_{a_n}}) \end{array} \right.$$

- f_i is monotone and, thus, the system has least fixed-point, which is the limit of

$$\begin{array}{lcl} \mathbf{y}_0 & = & \mathbf{0} \\ \mathbf{y}_{i+1} & = & \mathbf{f}(\mathbf{y}_i) \,. \end{array}$$

  where $\mathbf{f} = \langle f_1,\ldots,f_n \rangle$ and $\mathbf{f}(\mathbf{x}) = \langle f_1(x_1),\ldots,f_n(x_n) \rangle$
- The least-fixed point is the least model of $\mathcal{P}$

Example

$$\mathcal{P} = \begin{cases} Q(x) & \leftarrow & B(x) \\ Q(x) & \leftarrow & C(x) \\ B(a) & & \\ C(b) & & \end{cases} \qquad \mathcal{P}^* = \begin{cases} Q(a) & \leftarrow & B(a) \vee C(a) \\ Q(b) & \leftarrow & B(b) \vee C(b) \\ B(a) & \leftarrow & 1 \\ C(b) & \leftarrow & 1 \end{cases}$$

$$\begin{cases} x_{Q(a)} & = & \max(x_{B(a)}, x_{C(a)}) \\ x_{Q(b)} & = & \max(x_{B(b)}, x_{C(b)}) \\ x_{B(a)} & = & 1 \\ x_{C(b)} & = & 1 \end{cases}$$

$\mathbf{y}_i$	$x_{Q(a)}$	$x_{Q(b)}$	$x_{B(a)}$	$x_{B(b)}$	$x_{C(a)}$	$x_{C(b)}$
$\mathbf{y}_0$	0	0	0	0	0	0
$\mathbf{y}_1$	0	0	1	0	0	1
$\mathbf{y}_2$	1	1	1	0	0	1
$\mathbf{y}_3$	1	1	1	0	0	1

- $\mathbf{y}_2 = \mathbf{y}_3$, i.e.$\mathbf{f}(\mathbf{y}_2) = \mathbf{y}_3 = \mathbf{y}_2$
- $\mathbf{y}_2$ is least fixed-point and, thus, minimal model

- A simple query answering procedure to determine $ans(\mathcal{P}, q(\mathbf{x}))$:
  1. Convert $\mathcal{P}$ into $\mathcal{P}^*$
  2. Compute the minimal model $M_{\mathcal{P}}$ of $\mathcal{P}^*$, i.e. of $\mathcal{P}$
  3. Store the minimal model $M_{\mathcal{P}}$ of $\mathcal{P}^*$ in a database
  4. Translate $q(\mathbf{x})$ into a SQL statement
  5. Execute the SQL query over the relational database
- Problem: $M_{\mathcal{P}}$ may be huge (exponential in the size of $\mathcal{P}^*$)
- Possible solution: top-down query answering procedure
- First step: a top-down query answering procedure for ground queries
  - Given $q(\mathbf{c})$, check if $\mathcal{P} \models q(\mathbf{c})$ by computing just a fragment of $M_{\mathcal{P}}$ sufficient to answer the question
  - A top-down procedure exists for equational systems
  - Therefore, it works for LPs too

Procedure *Solve*$(\mathcal{S}, Q)$

Input: monotonic system $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, where $Q \subseteq V$ is the set of query variables;

Output: A set $B \subseteq V$, with $Q \subseteq B$ such that the mapping v equals $\mathrm{lfp}(f)$ on B.

$A := Q$, $dg := Q$, $in := \emptyset$, **for all** $x \in V$ **do** $v(x) = 0$, $exp(x) = 0$

while $A \neq \emptyset$ **do**

  select $x_i \in A$, $A := A \setminus \{x_i\}$, $dg := dg \cup s(x_i)$

  $r := f_i(v(x_{i_1}), ..., v(x_{i_{a_i}}))$

  if $r \succ v(x_i)$ **then** $v(x_i) := r$, $A := A \cup (p(x_i) \cap dg)$ **fi**

  if not $exp(x_i)$ **then** $exp(x_i) = 1$, $A := A \cup (s(x_i) \setminus in)$, $in := in \cup s(x_i)$ **fi**

  remove x **from** A **if** $v(x) = \top$

od

$\mathcal{L}$ is complete lattice. For $q(\mathbf{x}) \leftarrow \phi \in \mathcal{P}$, with $\mathrm{s}(q)$ we denote the set of *sons* of q w.r.t. r, i.e. the set of intentional predicate symbols occurring in ϕ. With $\mathrm{p}(q)$ we denote the set of *parents* of q, i.e. the set $\mathrm{p}(q) = \{p_i : q \in \mathrm{s}(p_i, r)\}$ (the set of predicate symbols directly depending on q).

Example

$$\mathcal{P}^* = \left\{ \begin{array}{lcl} a & \leftarrow & b \wedge c \\ c & \leftarrow & a \vee d \\ b & \leftarrow & 1 \\ d & \leftarrow & 1 \end{array} \right. \qquad \left\{ \begin{array}{lcl} x_a & = & \min(x_b, x_c) \\ x_c & = & \max(x_a, x_d) \\ x_b & = & 1 \\ x_d & = & 1 \end{array} \right.$$

$$\mathcal{P}^* \models a \quad ?$$

1. $\mathtt{A} = \{x_a\}, x_i = x_a, \mathtt{A} = \emptyset, \mathtt{dg} = \{x_a, x_b, x_c\}, r = 0, \mathtt{A} = \{x_b, x_c\}, \mathtt{exp}(x_a) = \mathtt{1}, \mathtt{in} = \{x_b, x_c\}$
2. $x_i = x_b, \mathtt{A} = \{x_c\}, r = 1, \mathtt{v}(x_b) = 1, \mathtt{A} = \{x_c, x_a\}, \mathtt{exp}(x_b) = \mathtt{1}$
3. $x_i = x_c, \mathtt{A} = \{x_a\}, \mathtt{dg} = \{x_a, x_b, x_c, x_d\}, r = 0, \mathtt{exp}(x_c) = \mathtt{1}, \mathtt{A} = \{x_a, x_d\}, \mathtt{in} = \{x_a, x_b, x_c, x_d\}$
4. $x_i = x_d, \mathtt{A} = \{x_a\}, r = 1, \mathtt{v}(x_d) = 1, \mathtt{exp}(x_d) = \mathtt{1}, \mathtt{A} = \{x_a, x_c\}$
5. $x_i = x_c, \mathtt{A} = \{x_a\}, r = 1, \mathtt{v}(x_c) = 1$
6. $x_i = x_a, \mathtt{A} = \emptyset, r = 1, \mathtt{v}(x_a) = 1$
7. $\mathtt{stop.\ return\ v}$ (in particular, $\mathtt{v}(x_a) = 1$)

- The fact that only a part of the model is computed becomes evident
  - the computation does not change if we add any program $\mathcal{P}'$ to $\mathcal{P}$ not containing atoms of $\mathcal{P}$
  - a bottom-up computation will consider $\mathcal{P}'$ as well
- Problem: we answer ground queries $q(\mathbf{c})$ only
  - There are too many **c** on which to test $q(\mathbf{c})$
- Solution: generalize *Solve*$(\mathcal{S}, Q)$ to compute ALL answers in one run only
  - Idea: the procedure is as for *Solve*$(\mathcal{S}, Q)$, but we compute answers incrementally

Computing All Answers

- A query is an atom Q (*query atom*) of the form $q(\mathbf{x})$
- For a given n-ary predicate p and a set of answers Δ_p of p, for convenience we represent Δ_p as an n-ary table tab_{Δ_p}, containing the records of the form $\langle c_1, \ldots, c_n \rangle$
- If Δ_p^1 and Δ_p^2 are two sets of answers for p, we write $\Delta_p^1 \preceq \Delta_p^2$ iff $\Delta_p^1 \subset \Delta_p^2$
- Our algorithm is an improved top-down query answering algorithm based on Semi Naive Evaluation for

  Datalog
  1. start by assuming all IDB (Intentional Database) relations empty;
  2. repeatedly evaluate the rules using the EDB (Extensional Database) and the previous IDB, to get a new IDB;
  3. end when no change to IDB.
- Consider a rule $p(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$ with predicates $p_1, \ldots, p_k$ in rule body $\varphi(\mathbf{x}, \mathbf{y})$
- Consider interpretation $\mathcal{I}$

$$\mathcal{I}(p_i(\mathbf{c})) = \begin{cases} 1, & \text{if } \mathbf{c} \in \Delta_{p_i} \\ 0, & \text{otherwise.} \end{cases}$$

- Assume

$$eval(p, \Delta_{p_1}, \ldots, \Delta_{p_k}) = \{\mathbf{c} \mid 1 = \max_{\mathbf{c}'} \mathcal{I}(\varphi(\mathbf{c}, \mathbf{c}'))\} ,$$

  where $\mathbf{c}'$ is a tuple of constants occurring in $\bigcup_i \Delta_{p_i}$
- *eval* can be implemented using SQL queries over relational tables $tab_{\Delta_{p_1}}, \ldots, tab_{\Delta_{p_k}}$
  - E.g.,

$$\mathtt{path}(x, y) \quad \leftarrow \quad \mathtt{edge}(x, y) \vee (\mathtt{path}(x, z) \wedge \mathtt{edge}(z, y))$$

  - $eval(path, \Delta_{edge}, \Delta_{path})$ is

$$\pi_{1,2}(tab_{\Delta_{\mathtt{edge}}}) \cup \pi_{1,4}(tab_{\Delta_{\mathtt{edge}}} \bowtie_{2=3} tab_{\Delta_{\mathtt{path}}}) . \quad (1)$$

Procedure *Answer*$(\mathcal{L}, \mathcal{K}, Q)$

Input: Truth space $\mathcal{L} = \{0, 1\}$, knowledge base $\mathcal{K}$, set Q of query predicate symbols

Output: A mapping $\mathtt{v}$ such that it contains all answers of predicates in Q.

1. $\mathtt{A} := Q$, $\mathtt{dg} := Q$, $\mathtt{in} := \emptyset$, **for all** predicate symbols p in $\mathcal{P}$ **do** $\mathtt{v}(p) = \emptyset$, $\mathtt{exp}(p) = \mathtt{false}$
2. **while** $\mathtt{A} \neq \emptyset$ **do**
3.   **select** $p_i \in \mathtt{A}$, $\mathtt{A} := \mathtt{A} \setminus \{p_i\}$, $\mathtt{dg} := \mathtt{dg} \cup \mathtt{s}(p_i)$
4.   **if** (p_i extensional predicate) $\wedge$ ($\mathtt{v}(p_i) = \emptyset$) **then** $\mathtt{v}(p_i) := tab_{p_i}$
5.   **if** (p_i intentional predicate) **then** $\Delta_{p_i} := eval(p_i, \mathtt{v}(p_{i_1}), \ldots, \mathtt{v}(p_{i_{k_i}}))$
6.   **if** $\Delta_{p_i} \succ \mathtt{v}(p_i)$ **then** $\mathtt{v}(p_i) := \Delta_{p_i}$, $\mathtt{A} := \mathtt{A} \cup (\mathtt{p}(p_i) \cap \mathtt{dg})$ **fi**
7.   **if not** $\mathtt{exp}(p_i)$ **then** $\mathtt{exp}(p_i) = \mathtt{true}$, $\mathtt{A} := \mathtt{A} \cup (\mathtt{s}(p_i) \setminus \mathtt{in})$, $\mathtt{in} := \mathtt{in} \cup \mathtt{s}(p_i)$ **fi**

od

- for predicate symbol p_i, $\mathtt{s}(p_i)$ is the set of predicate symbols occurring in the rule body of p_i, i.e. the *sons* of p_i;
- for predicate symbol p_i, $\mathtt{p}(p_i) = \{p_j : p_i \in \mathtt{s}(p_j)\}$, i.e. the *parents* of p_i;
- in step 5, $p_{i_1}, \ldots, p_{i_{k_i}}$ are all predicate symbols occurring in the rule body of p_i, i.e. the sons $\mathtt{s}(p_i) = \{p_{i_1}, \ldots, p_{i_{k_i}}\}$ of p_i.

Path Example

$$\mathtt{path}(x,y) \leftarrow \mathtt{edge}(x,y) \vee (\mathtt{path}(x,z) \wedge \mathtt{edge}(z,y))$$

$tab_{\mathtt{edge}}$	
c	b
a	c
b	a
a	b

1. $\mathtt{A} := \{\mathtt{path}\}$, $p_i := \mathtt{path}$, $\mathtt{A} := \emptyset$, $\mathtt{dg} := \{\mathtt{path}, \mathtt{edge}\}$, $\Delta_{\mathtt{path}} := \emptyset$
   $\mathtt{exp}(\mathtt{path}) := 1$, $\mathtt{A} := \{\mathtt{path}, \mathtt{edge}\}$, $\mathtt{in} := \{\mathtt{path}, \mathtt{edge}\}$
2. $p_i := \mathtt{path}$, $\mathtt{A} := \{\mathtt{edge}\}$, $\Delta_{\mathtt{path}} := \emptyset$
3. $p_i := \mathtt{edge}$, $\mathtt{A} := \emptyset$, $\Delta_{\mathtt{edge}} \succ \mathtt{v}(\mathtt{edge})$, $\mathtt{v}(\mathtt{edge}) := \Delta_{\mathtt{edge}}$, $\mathtt{A} := \{\mathtt{path}\}$, $\mathtt{exp}(\mathtt{edge}) := 1$
4. $p_i := \mathtt{path}$, $\mathtt{A} := \emptyset$, $\Delta_{\mathtt{path}} \succ \mathtt{v}(\mathtt{path})$, $\mathtt{v}(\mathtt{path}) := \Delta_{\mathtt{path}}$, $\mathtt{A} := \{\mathtt{path}\}$
5. $p_i := \mathtt{path}$, $\mathtt{A} := \emptyset$, $\Delta_{\mathtt{path}} \succ \mathtt{v}(\mathtt{path})$, $\mathtt{v}(\mathtt{path}) := \Delta_{\mathtt{path}}$, $\mathtt{A} := \{\mathtt{path}\}$
6. $p_i := \mathtt{path}$, $\mathtt{A} := \emptyset$, $\Delta_{\mathtt{path}} \succ \mathtt{v}(\mathtt{path})$, $\mathtt{v}(\mathtt{path}) := \Delta_{\mathtt{path}}$, $\mathtt{A} := \{\mathtt{path}\}$
7. $p_i := \mathtt{path}$, $\mathtt{A} := \emptyset$, $\Delta_{\mathtt{path}} = \mathtt{v}(\mathtt{path})$
8. stop. return $\mathtt{v}(\mathtt{path})$

Iteri	Δ_{p_i}	$\mathtt{v}(p_i)$
0.	–	$\mathtt{v}(\mathtt{edge}) = \mathtt{v}(\mathtt{path}) = \emptyset$
1.	$\Delta_{\mathtt{path}} = \emptyset$	–
2.	$\Delta_{\mathtt{path}} = \emptyset$	–
3.	$\Delta_{\mathtt{edge}} = \{\langle a,b\rangle, \langle b,a\rangle, \langle a,c\rangle, \langle c,b\rangle\}$	$\mathtt{v}(\mathtt{edge}) = \Delta_{\mathtt{edge}}$
4.	$\Delta_{\mathtt{path}} = \{\langle a,b\rangle, \langle b,a\rangle, \langle a,c\rangle, \langle c,b\rangle\}$	$\mathtt{v}(\mathtt{path}) = \Delta_{\mathtt{path}}$
5.	$\Delta_{\mathtt{path}} = \{\langle a,a\rangle, \langle a,b\rangle, \langle a,c\rangle, \langle b,a\rangle, \langle b,b\rangle, \langle b,c\rangle, \langle c,a\rangle, \langle c,b\rangle\}$	$\mathtt{v}(\mathtt{path}) = \Delta_{\mathtt{path}}$
6.	$\Delta_{\mathtt{path}} = \{\langle a,a\rangle, \langle a,b\rangle, \langle a,c\rangle, \langle b,a\rangle, \langle b,b\rangle, \langle b,c\rangle, \langle c,a\rangle, \langle c,b\rangle, \langle c,c\rangle\}$	$\mathtt{v}(\mathtt{path}) = \Delta_{\mathtt{path}}$
7.	$\Delta_{\mathtt{path}} = \{\langle a,a\rangle, \langle a,b\rangle, \langle a,c\rangle, \langle b,a\rangle, \langle b,b\rangle, \langle b,c\rangle, \langle c,a\rangle, \langle c,b\rangle, \langle c,c\rangle\}$	–

Uncertainty and Fuzzyness in Logics

Uncertainty vs. Vagueness: a clarification

Uncertainty vs Vagueness: a clarification

- Initial difficulty:
  - Understand the conceptual differences between uncertainty and vagueness
- Main problem:
  - Interpreting a degree as a measure of uncertainty rather than as a measure of vagueness

Uncertain Statements

- A statement is true or false in any world/interpretation
  - We are “uncertain” about which world to consider
  - We may have e.g. a probability or possibility distribution over possible worlds
- E.g., “it will rain tomorrow”
  - We cannot exactly establish whether it will rain tomorrow or not, due to our incomplete knowledge about our world
  - We can estimate to which degree this is probable

- Consider a propositional statement (formula) ϕ
- Interpretation (world) $\mathcal{I} \in \mathcal{W}$,

$$\mathcal{I} : W \to \{0, 1\}$$

- $\mathcal{I}(\phi) = 1$ means ϕ is true in $\mathcal{I}$, denoted $\mathcal{I} \models \phi$
- Each interpretation $\mathcal{I}$ depicts some concrete world
- Given n propositional letters, $|W| = 2^n$
- In uncertainty theory, we do not know which interpretation $\mathcal{I}$ is the actual one

- One may construct a probability distribution over the worlds

$$Pr : W \rightarrow [0, 1]$$
$$\sum_{\mathcal{I}} Pr(\mathcal{I}) = 1$$

- $Pr(\mathcal{I})$ indicates the probability that $\mathcal{I}$ is the actual world
- Probability $Pr(\phi)$ of a statement ϕ in Pr

$$Pr(\phi) = \sum_{\mathcal{I} \models \phi} Pr(\mathcal{I})$$

- $Pr(\phi)$ is the probability of the event: "ϕ is true"

Example (Sport Cars)

- Sport Car:

$$\forall x, hp, sp, ac\ SportCar(x) \quad \leftrightarrow \quad HP(x, hp) \wedge Speed(x, sp) \wedge Acceleration(x, ac)$$
$$\wedge hp \geq 210 \wedge sp \geq 220 \wedge ac \leq 7.0$$

audi_tt

mg

ferrari_enzo

- Ferrari Enzo is a Sport Car: $HP = 651$, $Speed \geq 350$, $Acc. = 3.14$
- MG is not a Sport Car: $HP = 59$, $Speed = 170$, $Acc. = 14.3$
- Is Audi TT 2.0 a Sport Car ? $HP = unknown$, $Speed = 243$, $Acc. = 6.9$
- We can estimate from a training set (Naive Bayes Classification)

$$Pr(SportCar|AudiTT) = Pr(AudiTT|SportCar) \cdot Pr(SportCar) \cdot (1/Pr(AudiTT))$$
$$\approx \frac{Pr(speed \leq 243|SportCar) \cdot Pr(accel \geq 6.9|SportCar) \cdot Pr(SportCar)}{Pr(speed \leq 243) \cdot Pr(accel \geq 6.9)}$$

Vague Statements

- A statement is vague whenever it involves vague concepts or vague objects
  - Heavy rain
  - Tall person
  - Hot temperature
  - The dunes in a desert
- The truth of a vague statement is a matter of degree, as it is intrinsically difficult to establish whether the statement is entirely true or false
  - There are 33 °C. Is it hot?

- A concept is vague whenever its extension is deemed lacking in clarity
  - Aboutness of a picture or piece of text
  - Tall person
  - High temperature
  - Nice weather
  - Adventurous trip
  - Similar proof
- Vague concepts:
  - Are abundant in everyday speech and almost inevitable
  - Their meaning is often subjective and context dependent

- An object is vague whenever its identity is lacking in clarity
  - Dust
  - Cloud
  - Dunes
  - Sun
- Vague objects:
  - Are not identical to anything, except to themselves (reflexivity)
  - Are characterised by a vague identity relation (e.g. a similarity relation)

TripAdvisor: Hotel User Judgments

Vague Statements (cont.)

- A statement is true to some degree, which is taken from a truth space (usually $[0, 1]$)
- The convention prescribing that a proposition is either true or false is changed towards graded propositions
- E.g., “heavy rain”
  - The compatibility of “heavy” in the phrase “heavy rain” is graded and the degree depends on the amount of rain is falling
    - The intensity of precipitation is expressed in terms of a precipitation rate R: volume flux of precipitation through a horizontal surface, i.e. $m^3/m^2s = ms^{-1}$
    - It is usually expressed in mm/h

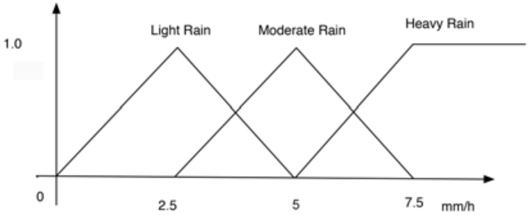
"Heavy rain" continued...E.g., in weather forecasts one may find:

- Rain intensity measured as precipitation rate R: volume flux of precipitation through a horizontal surface, i.e. $m^3/m^2h = mh^{-1}$

  - **Rain.** Falling drops of water larger than 0.5 mm in diameter. "Rain" usually implies that the rain will fall steadily over a period of time;
  - **Light rain.** Rain falls at the rate of 2.6 mm or less an hour;
  - **Moderate rain.** Rain falls at the rate of 2.7 mm to 7.6 mm an hour;
  - **Heavy rain.** Rain falls at the rate of 7.7 mm an hour or more.

- Quite harsh distinction: $$\begin{array}{lcl} R = 7.7mm/h & \rightarrow & \text{heavy rain} \\ R = 7.6mm/h & \rightarrow & \text{moderate rain} \end{array}$$
- This is clearly unsatisfactory, as quite naturally
  - The more rain is falling, the more the sentence "heavy rain" is true
  - Vice-versa, the less rain is falling the less the sentence is true

- In other words, that the sentence "heavy rain" is no longer either true or false, but is intrinsically graded
  - Even if we have complete knowledge about the current world, i.e. exact specification of the precipitation rate
- More fine grained approach:
  - Define the various types of rains as



  - Light rain, moderate rain and heavy rain are vague concepts

- Consider a propositional statement ϕ
- A propositional interpretation $\mathcal{I}$ maps ϕ to a truth degree in $[0, 1]$

$$\mathcal{I}(\phi) \in [0, 1]$$

- I.e., we are unable to establish whether a statement is entirely true or false due the occurrence of vague concept
- Vague statements are truth-functional
    - Degree of truth of a statement can be calculated from the degrees of truth of its constituents
    - Note that this is not possible for uncertain statements
- Example of truth functional interpretation of vague statements:

$$\begin{array}{lcl} \mathcal{I}(\phi \wedge \psi) & = & \min(\mathcal{I}(\phi), \mathcal{I}(\psi)) \\ \mathcal{I}(\phi \vee \psi) & = & \max(\mathcal{I}(\phi), \mathcal{I}(\psi)) \\ \mathcal{I}(\neg\phi) & = & 1 - \mathcal{I}(\phi) \end{array}$$

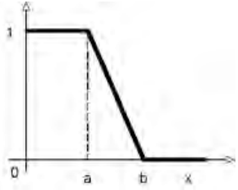
Example

- Sport Car:

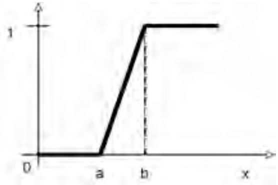
$$\forall x, hp, sp, ac\ SportCar(x) \leftrightarrow 0.3 \cdot HP(x, hp) + 0.2 \cdot Speed(x, sp) + 0.5 \cdot Accel(x, ac)$$

- Each feature, gives a degree of truth depending on the value and the membership function

$$\begin{aligned} HP(x, hp) &= rs(180, 250)(hp) \\ Speed(x, sp) &= rs(180, 240)(sp) \\ Accel(x, ac) &= ls(6.0, 8.0)(ac) \end{aligned}$$



ls(a,b)

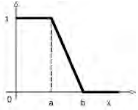


rs(a,b)

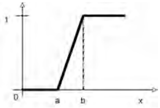
- Degree of truth of $SportCar(AudiTT)$: $0.3 \cdot 0.28 + 0.3 \cdot 1.0 + 0.5 \cdot 0.55 = 0.447$

- The fuzzy membership functions can be learned from a training set (large literature)

$$\begin{aligned} HP(x, hp) &= rs(192, 242)(hp) \\ Speed(x, sp) &= rs(193, 234)(sp) \\ Accel(x, ac) &= ls(6.5, 7.5)(ac) \end{aligned}$$



ls(a,b)



rs(a,b)

- Learned Training Sport Class:

$$\forall x, hp, sp, ac\ TrainingSportCar(x) \leftrightarrow 0.3 \cdot HP(x, hp) + 0.2 \cdot Speed(x, sp) + 0.5 \cdot Accel(x, ac)$$

- Now, a classification method can be applied: e.g. kNN classifier

$$\forall x, hp, sp, ac\ SportCar(x) \leftrightarrow \sum_{y \in Top_k(x)} Similar(x, y) \cdot TrainingSportCar(y)$$

$$\forall x, hp, sp, ac\ Similar(x, y) \leftrightarrow 0.3 \cdot HP(x, hpx) \cdot HP(y, hpy) + 0.2 \cdot Speed(x, spx) \cdot Speed(y, spy) + \\ + 0.5 \cdot Accel(x, acx) \cdot Accel(y, acy)$$

where $Top_k(x)$ is the set of top-k ranked most similar cars to car x

Uncertain & Vague Statements

- Recap:
  - In a probabilistic setting each statement is either true or false, but there is e.g. a probability distribution telling us how probable each interpretation/sentence is

$$\mathcal{I}(\phi) \in \{0,1\}, Pr(\mathcal{I}) \in [0,1] \text{ and } Pr(\phi) = \sum_{\mathcal{I} \models \phi} Pr(\mathcal{I}) \quad \in [0,1]$$

  - In vagueness theory instead, sentences are graded

$$\mathcal{I}(\phi) \in [0,1]$$

Uncertain Vague Statements

- Are there sentences combining the two orthogonal concepts of uncertainty and vagueness?
- Yes, and we use them daily ! E.g.,
  - “Very likely there will be heavy rain tomorrow"
- This type of sentences are called uncertain vague sentences
- Essentially, there is
  - uncertainty about the world we will have tomorrow
  - vagueness about the various types of rain
- Exercise: formalise
  - “Quite unlikely, I will pay to many of you some fair amount of money if the temperature in the following days will be slighlty higher than now”

- Consider a propositional statement ϕ
- A model for uncertain vague sentences:
  - Define probability distribution over worlds $\mathcal{I} \in W$, i.e.

$$Pr(\mathcal{I}) \in [0,1], \sum_{\mathcal{I}} Pr(\mathcal{I}) = 1$$

  - Sentences are graded: each interpretation $\mathcal{I} \in W$ is truth functional and maps sentences into $[0,1]$

$$\mathcal{I}(\phi) \in [0,1]$$

  - For a sentence ϕ, consider the expected truth of ϕ

$$ET(\phi) = \sum_{\mathcal{I}} Pr(\mathcal{I}) \cdot \mathcal{I}(\phi) .$$

- Note: if $\mathcal{I}$ is bivalent (that is, $\mathcal{I}(\phi) \in \{0,1\}$) then $ET(\phi) = Pr(\phi)$

Uncertainty or Vagueness ?

- The distinction between uncertainty and vagueness is not always clear: depends on the assumptions
- (Multimedia) Information Retrieval:

Query: "I'm looking for a house"

System Answer:  score/degree 0.83

- What's behind the computational model?

- Probabilistic model
  - Assumption: a multimedia object is either relevant or not relevant to a query q
  - Score: The probability of being a multimedia object o relevant (Rel) to q

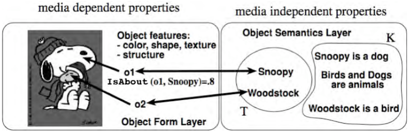
$$score := Pr(Rel \mid q, o)$$

- Vague/Fuzzy model
  - Assumption: a multimedia object o is about a *semantic index term* ($t \in \mathbb{T}$) to some degree in $[0, 1]$
  - The mapping of objects $o \in \mathbb{O}$ to semantic entities $t \in \mathbb{T}$ is called *semantic annotation*

$$F : \mathbb{O} \times \mathbb{T} \to [0, 1]$$

  $F(o, t)$ indicates to which degree the multimedia object o is about the semantic index term t
  - Score: The evaluation of how much the multimedia object o is about the the information need q

$$score := F(o, q)$$



Probability & Propositional Logic

- A statement φ is either true or false
- Due to lack of knowledge we can only estimate to which probability degree they are true or false
- Usually we have a possible world semantics with a distribution over possible worlds
- Possible world: any classical interpretation I, mapping any statement φ into $\{0, 1\}$

$$W = \{I \text{ classical interpretation}\}, \quad I(\varphi) \in \{0, 1\}$$

- Probability distribution: a mapping

$$\mu\colon W \to [0, 1], \quad \mu(I) \in [0, 1]$$

such that

$$\sum_{I \in W} \mu(I) = 1$$

- $\mu(I)$ indicates the probability that the world I is indeed the actual one

- A statement φ corresponds to the event M_ϕ "the set of models of φ", i.e.

$$M_\varphi = \{I \mid I \models \varphi\}$$

- The probability of a statement φ is determined as

$$Pr(\varphi) = Pr(M_\varphi) = \sum_{I \models \varphi} \mu(I)$$

Example

Probabilistic setting:

$$\varphi = \mathit{sprinklerOn} \vee \mathit{wet}$$

W	*sprinklerOn*	*wet*	μ
I_1	0	0	0.1
I_2	0	1	0.2
I_3	1	0	0.4
I_4	1	1	0.3

$$1 = \sum_{I \in W} \mu(I)$$

$$\begin{aligned} Pr(\varphi) &= Pr(\{I_2, I_3, I_4\}) \\ &= 0.2 + 0.4 + 0.3 = 0.9 \end{aligned}$$

Properties of probabilistic formulae

$$
\begin{aligned}
Pr(\varphi \wedge \psi) &= Pr(\varphi) + Pr(\psi) - Pr(\varphi \vee \psi) \\
Pr(\varphi \wedge \psi) &\leq \min(Pr(\varphi), Pr(\psi)) \\
Pr(\varphi \wedge \psi) &\geq \max(0, Pr(\varphi) + Pr(\psi) - 1) \\
Pr(\varphi \vee \psi) &= Pr(\varphi) + Pr(\psi) - Pr(\varphi \wedge \psi) \\
Pr(\varphi \vee \psi) &\leq \min(1, Pr(\varphi) + Pr(\psi)) \\
Pr(\varphi \vee \psi) &\geq \max(Pr(\varphi), Pr(\psi)) \\
Pr(\neg\varphi) &= 1 - Pr(\varphi) \\
Pr(\bot) &= 0 \\
Pr(\top) &= 1
\end{aligned}
$$

Probabilistic Knowledge Bases

- Finite nonempty set of basic events $\Phi = \{p_1, \ldots, p_n\}$
- Event φ: Boolean combination of basic events
- Logical constraint $\psi \Leftarrow \varphi$: events ψ and φ: "φ implies ψ"
- Conditional constraint $(\psi|\varphi)[l, u]$: events ψ and φ, and $l, u \in [0, 1]$: "conditional probability of ψ given φ is in $[l, u]$"
- $\psi \geq l$ is a shortcut for $(\psi|\top)[l, 1]$, $\psi \leq u$ is a shortcut for $(\psi|\top)[0, u]$
- Probabilistic knowledge base $KB = (L, P)$:
  - finite set of logical constraints L
  - finite set of conditional constraints P

Example

Probabilistic knowledge base $KB = (L, P)$:

- $L = \{bird \Leftarrow eagle\}$:

  "Eagles are birds"

- $P = \{(have\_legs \mid bird)[1, 1], (fly \mid bird)[0.95, 1]\}$:

  "Birds have legs"

  "Birds fly with a probability of at least 0.95"

- World I: truth assignment to all basic events in Φ
- $\mathcal{I}_\Phi$: all worlds for Φ
- Probabilistic interpretation Pr: probability distribution on $\mathcal{I}_\Phi$
- $Pr(\varphi)$: sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models \varphi$

$$Pr(\varphi) = \sum_{I \models \varphi} Pr(I)$$

- $Pr(\psi|\varphi)$: if $Pr(\varphi) > 0$, then

$$Pr(\psi|\varphi) = \frac{Pr(\psi \wedge \varphi)}{Pr(\varphi)}$$

- Truth under Pr:
  - $Pr \models \psi \Leftarrow \varphi$ iff $Pr(\psi \wedge \varphi) = Pr(\varphi)$
    (iff $Pr(\psi \Leftarrow \varphi) = 1$)
  - $Pr \models (\psi|\varphi)[l, u]$ iff $Pr(\psi \wedge \varphi) \in [l, u] \cdot Pr(\varphi)$
    (iff either $Pr(\varphi) = 0$ or $Pr(\psi|\varphi) \in [l, u]$)

Example

- Set of basic propositions $\Phi = \{bird, fly\}$.
- $\mathcal{I}_\Phi$ contains exactly the worlds I_1, I_2, I_3, and I_4 over Φ:

	fly	$\neg$*fly*
bird	I_1	I_2
$\neg$*bird*	I_3	I_4

- Some probabilistic interpretations:

Pr_1	*fly*	$\neg$*fly*
bird	19/40	1/40
$\neg$*bird*	10/40	10/40

Pr_2	*fly*	$\neg$*fly*
bird	0	1/3
$\neg$*bird*	1/3	1/3

- $Pr_1(fly \wedge bird) = 19/40$ and $Pr_1(bird) = 20/40$.
- $Pr_2(fly \wedge bird) = 0$ and $Pr_2(bird) = 1/3$.
- $\neg fly \Leftarrow bird$ is false in Pr_1, but true in Pr_2.
- $(fly \mid bird)[.95, 1]$ is true in Pr_1, but false in Pr_2.

Satisfiability and Logical Entailment

- Pr is a model of $KB = (L, P)$ iff $Pr \models F$ for all $F \in L \cup P$
- KB is satisfiable iff a model of KB exists
- $KB \Vdash (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a logical consequence of KB iff every model of KB is also a model of $(\psi|\varphi)[l, u]$
- $KB \Vdash_{tight} (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a tight logical consequence of KB iff l (resp., u) is the infimum (resp., supremum) of $Pr(\psi|\varphi)$ subject to all models Pr of KB with $Pr(\varphi) > 0$

Example

- Probabilistic knowledge base:

  $$KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[0.95,1]\})$$

- KB is satisfiable, since

  Pr with $Pr(bird \wedge eagle \wedge have\_legs \wedge fly) = 1$ is a model

- Some conclusions under logical entailment:

  $KB \Vdash (have\_legs \mid bird)[0.3, 1] KB \Vdash (fly \mid bird)[0.6, 1]$

- Tight conclusions under logical entailment:

$$\begin{aligned} KB &\Vdash_{tight} (have\_legs \mid bird)[1,1] \\ KB &\Vdash_{tight} (fly \mid bird)[0.95,1] \\ KB &\Vdash_{tight} (have\_legs \mid eagle)[1,1] \\ KB &\Vdash_{tight} (fly \mid eagle)[0,1] \end{aligned}$$

Deciding Model Existence / Satisfiability

Theorem: The probabilistic knowledge base $KB = (L, P)$ has a model Pr iff the following system of linear constraints LC over the variables y_r ($r \in R$), where $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$, is solvable:

$$\begin{aligned}
\sum_{r \in R,\, r \models \neg\psi \wedge \varphi} -l\, y_r + \sum_{r \in R,\, r \models \psi \wedge \varphi} (1-l)\, y_r &\geq 0 \quad (\forall (\psi|\varphi)[l,u] \in P), l > 0 \\
\sum_{r \in R,\, r \models \neg\psi \wedge \varphi} u\, y_r + \sum_{r \in R,\, r \models \psi \wedge \varphi} (u-1)\, y_r &\geq 0 \quad (\forall (\psi|\varphi)[l,u] \in P, u < 1) \\
\sum_{r \in R} y_r &= 1 \\
y_r &\geq 0 \quad (\text{for all } r \in R)
\end{aligned}$$

Explanation

- A probability distribution Pr is a model of $(\psi|\varphi)[l, u]$ iff

$$
\begin{array}{rcl}
Pr(\psi \mid \varphi) \in [l, u] & \text{iff} & Pr(\psi \wedge \varphi)/Pr(\varphi) \in [l, u] \\
 & \text{iff} & Pr(\psi \wedge \varphi) \in [l \cdot Pr(\varphi), u \cdot Pr(\varphi)] \\
 & \text{iff} & Pr(\psi \wedge \varphi) \geq l \cdot Pr(\varphi) \text{ and , } Pr(\psi \wedge \varphi) \leq u \cdot Pr(\varphi) \\
Pr(\psi \wedge \varphi) \geq l \cdot Pr(\varphi) & \text{iff} & Pr(\psi \wedge \varphi) - l \cdot Pr(\varphi) \geq 0 \\
 & \text{iff} & Pr(M_{\psi \wedge \varphi}) - l \cdot Pr(M_{\varphi}) \geq 0 \\
 & \text{iff} & Pr(M_{\psi \wedge \varphi}) - l \cdot Pr(M_{\psi \wedge \varphi} \cup M_{\neg \psi \wedge \varphi}) \geq 0 \\
 & \text{iff} & Pr(M_{\psi \wedge \varphi}) - l \cdot Pr(M_{\psi \wedge \varphi}) - l \cdot Pr(M_{\neg \psi \wedge \varphi}) \geq 0 \\
 & \text{iff} & (1 - l) \cdot Pr(M_{\psi \wedge \varphi}) - l \cdot Pr(M_{\neg \psi \wedge \varphi}) \geq 0 \\
 & \text{iff} & (1 - l) \sum_{r \models \psi \wedge \varphi} \mu(r) - l \sum_{lr \models \neg \psi \wedge \varphi} \mu(r) \geq 0 \\
 & \text{iff} & \sum_{r \models \psi \wedge \varphi} (1 - l)\mu(r) + \sum_{l \models \neg \psi \wedge \varphi} (-l)\mu(r) \geq 0
\end{array}
$$

- As we are looking for the values of $\mu(r)$, by setting $y_r = \mu(r)$, any solution to the variables y_r under

$$
\begin{array}{rcl}
\sum_{r \models \psi \wedge \varphi} (1 - l)y_r + \sum_{l \models \neg \psi \wedge \varphi} (-l)y_r & \geq & 0 \\
\sum_{r \in W} y_r & = & 1 \\
y_r & \geq & 0 \text{ for all } r \in W
\end{array}
$$

is a probabilistic model of $(\psi|\varphi)[l, 1]$. The equations for the upper bound are derived similarly.

Computing Tight Logical Consequences

Theorem: Suppose $KB = (L, P)$ has a model Pr such that $Pr(\alpha) > 0$. Then, l (resp., u) such that $KB \,\|\!\!=_{tight} (\beta|\alpha)[l, u]$ is given by the optimal value of the following linear program over the variables y_r ($r \in R$), where $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$:

$$\text{minimize (resp., maximize)} \sum_{r \in R,\, r \models \beta \wedge \alpha} y_r \quad \text{subject to}$$

$$\sum_{r \in R,\, r \models \neg\psi \wedge \varphi} -l\, y_r + \sum_{r \in R,\, r \models \psi \wedge \varphi} (1 - l)\, y_r \;\geq\; 0 \quad (\forall (\psi|\varphi)[l, u] \in P), l > 0$$

$$\sum_{r \in R,\, r \models \neg\psi \wedge \varphi} u\, y_r + \sum_{r \in R,\, r \models \psi \wedge \varphi} (u - 1)\, y_r \;\geq\; 0 \quad (\forall (\psi|\varphi)[l, u] \in P), u < 1$$

$$\sum_{r \in R} y_r \;=\; 1$$

$$y_r \;\geq\; 0 \quad (\text{for all } r \in R)$$

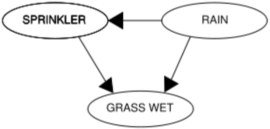
Bayesian Networks

Bayesian network (BN): compact specification of a joint distribution, based on a graphical notation for conditional independencies:

- a set of nodes; each node represents a random variable
- a directed, acyclic graph (link $\approx$ "directly influences")
- a conditional distribution for each node given its parents: $\mathbf{P}(X_i | Parents(X_i))$
- $$Pr(X_1, \ldots, X_n) = \Pi_{i=1}^{n} Pr(X_i \mid parents(X_i)) \ .$$

Any joint distribution can be represented as a BN.

RAIN	SPRINKLER: T	SPRINKLER: F
F	0.4	0.6
T	0.01	0.99



RAIN: T	RAIN: F
0.2	0.8

SPRINKLER	RAIN	GRASS WET: T	GRASS WET: F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Joint probability function is

$$Pr(\texttt{GrassWet}, \texttt{Sprinkler}, \texttt{Rain}) = Pr(\texttt{GrassWet} \mid \texttt{Sprinkler}, \texttt{Rain}) \cdot Pr(\texttt{Sprinkler} \mid \texttt{Rain}) \cdot Pr(\texttt{Rain}) \tag{2}$$

The model can answer questions like "What is the probability that it is raining, given the grass is wet?"

$$\begin{aligned} Pr(\texttt{Rain} = T \mid \texttt{GrassWet} = T) &= \frac{Pr(\texttt{Rain} = T, \texttt{GrassWet} = T)}{Pr(\texttt{GrassWet} = T)} \\ &= \frac{\sum_{Y \in \{T,F\}} Pr(\texttt{Rain} = T, \texttt{GrassWet} = T, \texttt{Sprinkler} = Y)}{\sum_{Y_1, Y_2 \in \{T,F\}} Pr(\texttt{GrassWet} = T, (\texttt{Rain} = Y_1, \texttt{Sprinkler} = Y_2))} \\ &= \frac{0.99 \cdot 0.01 \cdot 0.2 + 0.8 \cdot 0.99 \cdot 0.2}{0.99 \cdot 0.01 \cdot 0.2 + 0.9 \cdot 0.4 \cdot 0.8 + 0.8 \cdot 0.99 \cdot 0.2 + 0 \cdot 0.6 \cdot 0.8} \\ &\approx 0.3577 \end{aligned}$$

Encoding of Bayesian Network in Probabilistic Propositional Logic

- For every node a, we use a propositional letters $a(T)$ (a is true), $a(F)$ (a is false)
- We also need $(a(T) \leftrightarrow \neg a(F)) = 1)$
- If a node a has no parents: $a(T) = p$, where p is its associated probability
- If a node has parents, we encode its associated conditional probability table using conditional probability formulae

$$
\begin{aligned}
(\texttt{Sprinkler}(T) \mid \texttt{Rain}(F)) &= 0.4 \\
(\texttt{Sprinkler}(T) \mid \texttt{Rain}(T)) &= 0.01
\end{aligned}
$$

$$
\begin{aligned}
(\texttt{GrassWet}(T) \mid \texttt{Sprinkler}(F) \wedge \texttt{Rain}(F)) &= 0.0 \\
(\texttt{GrassWet}(T) \mid \texttt{Sprinkler}(F) \wedge \texttt{Rain}(T)) &= 0.8 \\
(\texttt{GrassWet}(T) \mid \texttt{Sprinkler}(T) \wedge \texttt{Rain}(F)) &= 0.9 \\
(\texttt{GrassWet}(T) \mid \texttt{Sprinkler}(T) \wedge \texttt{Rain}(T)) &= 0.99 \,.
\end{aligned}
$$

Independent Choice Logic: Propositional Case

- A knowledge base $KB = \langle P, \mathcal{C} \rangle$ is a set of propositional formulae P together with a choice space $\mathcal{C}$
- A choice space $\mathcal{C}$ is a set $\mathcal{C}$ of choices of the form $\{(A_1 : \alpha_1), \ldots, (A_n : \alpha_n)\}$, where A_i is an atom and the α_i sum-up to 1
- A total choice T is a set of atoms such that from each choice $C_j \in \mathcal{C}$ there is exactly one atom $A_i^j \in C_j$ in T
- The probability of a total choice T is $Pr(T) = Pr(\bigwedge_{A_i^j \in T} A_i^j) = \prod_{A_i^j \in T} \alpha_i^j$
- A query is a propositional formula q. The probability of q w.r.t. KB is

$$Pr(q \mid KB) = \sum_{\{T \mid P \cup T \models q\}} Pr(T)$$

- Example:

$$P = \{a \rightarrow c, b \rightarrow c\}$$
$$\mathcal{C} = \{C_1 = \{a : 0.7, \neg a : 0.3\}, C_2 = \{b : 0.6, \neg b : 0.4\}\}$$

	Total Choice	$Pr(T)$
T_1	$\{a, b\}$	0.42
T_2	$\{a, \neg b\}$	0.28
T_3	$\{\neg a, b\}$	0.18
T_4	$\{\neg a, \neg b\}$	0.12

$$Pr(c \mid KB) = Pr(T_1) + Pr(T_2) + Pr(T_3) = 1 - Pr(T_4) = 0.88$$

Fuzzyness & Logic (Basics)

- Statements involve concepts for which there is no exact definition, such as
  - tall, small, close, far, cheap, expensive, “is about”, “similar to”.
- A statements is true to some degree, which is taken from a truth space
- E.g., “Hotel Verdi is close to the train station to degree 0.83”
- E.g., “The image is about a sun set to degree 0.75”
- Truth space: set of truth values L and an partial order $\leq$
- Many-valued Interpretation: a function I mapping formulae into L, i.e. $I(\varphi) \in L$
- Mathematical Fuzzy Logic: $L = [0, 1]$, but also $\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\}$ for an integer $n \geq 1$

- Problem: what is the interpretation of e.g. $\varphi \wedge \psi$?
  - E.g., if $I(\varphi) = 0.83$ and $I(\psi) = 0.2$, what is the result of $0.83 \wedge 0.2$?
- More generally, what is the result of $n \wedge m$, for $n, m \in [0, 1]$?
- The choice cannot be any arbitrary computable function, but has to reflect some basic properties that one expects to hold for a "conjunction"
- Norms: functions that are used to interpret connectives such as $\wedge, \vee, \neg, \rightarrow$
  - t-norm: interprets conjunction
  - s-norm: interprets disjunction
- Norms are compatible with classical two-valued logic

From Crisp Sets to Fuzzy Sets

- Let X be a universal set of objects
- The power set, denoted 2^A, of a set $A \subset X$, is the set of subsets of A, i.e.,

$$2^A = \{B \mid B \subseteq A\}$$

- Often sets are defined as

$$A = \{x \mid P(x)\}$$

  - $P(x)$ is a statement "x has property P"
  - $P(x)$ is either true or false for any $x \in X$

- Examples of universe X and subsets $A, B \in 2^X$ may be

$$\begin{aligned} X &= \{x \mid x \text{ is a day}\} \\ A &= \{x \mid x \text{ is a rainy day}\} \\ B &= \{x \mid x \text{ is a day with precipitation rate } R \geq 7.5mm/h\} \end{aligned}$$

- In the above case: $B \subseteq A \subseteq X$
- The (crisp) membership function of a set $A \subseteq X$:

$$\chi_A \colon X \to \{0, 1\}$$

where $\chi_A(x) = 1$ iff $x \in A$
- Note that for sets $A, B \in 2^X$

$$A \subseteq B \text{ iff } \forall x \in X.\ \chi_A(x) \leq \chi_B(x)$$

- Fuzzy set A: $\chi_A\colon X \to [0,1]$, or simply

$$A\colon X \to [0,1]$$

- Fuzzy power set over X, is denoted $\tilde{2}^X$, i.e. the set of all fuzzy sets over X
- Example: the fuzzy set

$$C = \{x \mid x \text{ is a day with heavy precipitation rate } R\}$$

is defined via the membership function

$$\chi_C(x) = \begin{cases} 1 & \text{if } R \geq 7.5 \\ (x-5)/2.5 & \text{if } R \in [5, 7.5) \\ 0 & \text{otherwise} \end{cases}$$

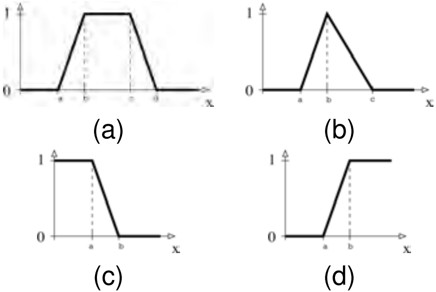
- Cardinality of a fuzzy set A: e.g. using sigma-count

$$|A| = \sum_{x \in X} \chi_A(x)$$

Fuzzy Sets Construction

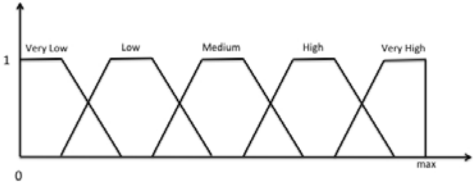
- The usefulness of fuzzy sets depends critically on appropriate membership functions
- Methods for fuzzy membership functions construction is largely addressed in literature

- Fuzzy membership functions may depend on the context and may be subjective
- Shape may be quite different
- Usually, it is sufficient to consider functions

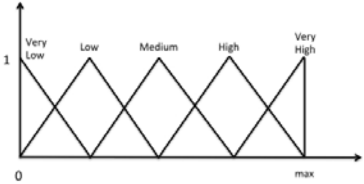


(a) Trapezoidal *trz*(a, b, c, d); (b) Triangular *tri*(a, b, c); (c) left-shoulder *ls*(a, b); (d) right-shoulder *rs*(a, b)

- Simple and typically satisfactory method (numerical domain):
  - uniform partitioning into 5 fuzzy sets

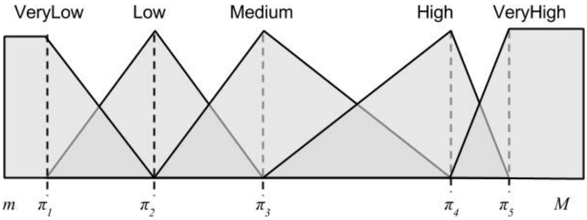


Fuzzy sets construction using trapezoidal functions



Fuzzy sets construction using triangular functions

- Another popular method is based on clustering
- Use Fuzzy C-Means to cluster data into 5 clusters
  - Fuzzy C-Means extends K-Means to accommodates graded membership
- From the clusters $c_1, \ldots, c_5$ take the centroids $\pi_1, \ldots, \pi_5$
- Build the fuzzy sets from the centroids



Fuzzy sets construction using clustering

Norm-Based Fuzzy Set Operations

- Standard fuzzy set operations are not the only ones
- Most notable ones are triangular norms
  - t-norm $\otimes$ for set intersection
  - t-conorm $\oplus$ (also called s-norm) for set union
  - negation $\ominus$ for set complementation
  - implication $\rightarrow$ for set inclusion
- These functions satisfy some properties that one expects to hold

Properties for t-norms and s-norms

Axiom Name	T-norm	S-norm
Taututology/Contradiction	$a \otimes 0 = 0$	$a \oplus 1 = 1$
Identity	$a \otimes 1 = a$	$a \oplus 0 = a$
Commutativity	$a \otimes b = b \otimes a$	$a \oplus b = b \oplus a$
Associativity	$(a \otimes b) \otimes c = a \otimes (b \otimes c)$	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$
Monotonicity	if $b \leq c$, then $a \otimes b \leq a \otimes c$	if $b \leq c$, then $a \oplus b \leq a \oplus c$

Properties for implication and negation functions

Axiom Name	Implication Function	Negation Function
Tautology / Contradiction	$0 \to b = 1,\ a \to 1 = 1,\ 1 \to 0 = 0$	$\ominus 0 = 1,\ \ominus 1 = 0$
Antitonicity	if $a \leq b$, then $a \to c \geq b \to c$	if $a \leq b$, then $\ominus a \geq \ominus b$
Monotonicity	if $b \leq c$, then $a \to b \leq a \to c$	

- By commutativity, $\otimes$ and $\oplus$ are monotone also in the first argument
- $\otimes$ is indempotent if $a \otimes a = a$, for all $a \in [0, 1]$
- Megation function $\ominus$ is involutive iff $\ominus \ominus a = a$, for all $a \in [0, 1]$.
- Salient negation functions are:

  Standard or Łukasiewicz negation: $\ominus_l a = 1 - a$;
  Gödel negation: $\ominus_g a$ is 1 if $a = 0$, else is 0.
- Łukasiewicz negation is involutive, Gödel negation is not.

- Salient t-norm functions are:

  Gödel t-norm: $a \otimes_g b = \min(a, b)$;

  Bounded difference or Łukasiewicz t-norm:

  $$a \otimes_l b = \max(0, a + b - 1);$$

  Algebraic product or product t-norm: $a \otimes_p b = a \cdot b$;

  Drastic product: $a \otimes_d b =$

  $$\begin{cases} 0 & \text{when } (a, b) \in [0, 1[\times[0, 1[\\ \min(a, b) & \text{otherwise} \end{cases}$$

- Salient s-norm functions are:

  Gödel s-norm: $a \oplus_g b = \max(a, b)$;

  Bounded sum or Łukasiewicz s-norm:

  $$a \oplus_l b = \min(1, a + b);$$

  Algebraic sum or product s-norm: $a \oplus_p b = a + b - ab$;

  Drastic sum: $a \oplus_d b =$

  $$\begin{cases} 1 & \text{when } (a, b) \in]0, 1] \times]0, 1] \\ \max(a, b) & \text{otherwise} \end{cases}$$

Salient properties of norms:

- Ordering among t-norms ($\otimes$ is any t-norm):

$$\otimes_d \leq \otimes \leq \otimes_g$$
$$\otimes_d \leq \otimes_l \leq \otimes_p \leq \otimes_g \ .$$

- The only idempotent t-norm is $\otimes_g$.
- The only t-norm satisfying $a \otimes a = 0$ for all $a \in [0, 1[$ is $\otimes_d$.
- Ordering among s-norms ($\oplus$ is any s-norm):

$$\oplus_g \leq \oplus \leq \oplus_d$$
$$\oplus_g \leq \oplus_p \leq \oplus_l \leq \oplus_d \ .$$

- The only idempotent s-norm is $\oplus_g$.
- The only s-norm satisfying $a \oplus a = 1$ for all $a \in]0, 1]$ is $\oplus_d$.
- The dual s-norm of $\otimes$ is defined as

$$a \oplus b = 1 - (1 - a) \otimes (1 - b) .$$

- Kleene-Dienes implication: $x \to y = \max(1 - x, y)$ is called
- Fuzzy modus ponens: let $a \geq n$ and $a \to b \geq m$
  - Under Kleene-Dienes implication, we infer that if $n > 1 - m$ then $b \geq m$
  - Under r-implication relative to a t-norm $\otimes$, we infer that $b \geq n \otimes m$
- Composition of two fuzzy relations $R_1 : X \times X \to [0, 1]$ and $R_2 : X \times X \to [0, 1]$: for all $x, z \in X$
  - $(R_1 \circ R_2)(x, z) = \sup_{y \in X} R_1(x, y) \otimes R_2(y, z)$
- A fuzzy relation R is transitive iff for all $x, z \in X$ $R(x, z) \geq (R \circ R)(x, z)$

Łukasiewicz, Gödel, Product logic and Standard Fuzzy logic

- One distinguishes three different sets of fuzzy set operations (called fuzzy logics)
  - Łukasiewicz, Gödel, and Product logic
  - Standard Fuzzy Logic (SFL) is a sublogic of Łukasiewicz
    - $\min(a, b) = a \otimes_l (a \rightarrow_l b)$, $\max(a, b) = 1 - \min(1 - a, 1 - b)$

	Łukasiewicz Logic	Gödel Logic	Product Logic	SFL
$a \otimes b$	$\max(a + b - 1, 0)$	$\min(a, b)$	$a \cdot b$	$\min(a, b)$
$a \oplus b$	$\min(a + b, 1)$	$\max(a, b)$	$a + b - a \cdot b$	$\max(a, b)$
$a \rightarrow b$	$\min(1 - a + b, 1)$	$\begin{cases} 1 & \text{if } a \leq b \\ b & \text{otherwise} \end{cases}$	$\min(1, b/a)$	$\max(1 - a, b)$
$\ominus a$	$1 - a$	$\begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}$	$1 - a$

- Mostert–Shields theorem: any continuous t-norm can be obtained as an ordinal sum of these three

Some additional properties

Property	Łukasiewicz Logic	Gödel Logic	Product Logic	SFL
$x \otimes \ominus x = 0$	•			
$x \oplus \ominus x = 1$	•			
$x \otimes x = x$		•		•
$x \oplus x = x$		•		•
$\ominus \ominus x = x$	•			•
$x \to y = \ominus x \oplus y$	•			•
$\ominus (x \to y) = x \otimes \ominus y$	•			•
$\ominus (x \otimes y) = \ominus x \oplus \ominus y$	•	•	•	•
$\ominus (x \oplus y) = \ominus x \otimes \ominus y$	•	•	•	•

- Note: If all conditions in the upper part of a column have to be satisfied then we collapse to classical two-valued logic

Fuzzy Modifiers

- Fuzzy modifiers: interesting feature of fuzzy set theory
- A fuzzy modifier apply to fuzzy sets to change their membership function
  - Examples: very, more_or_less, and slightly
- A fuzzy modifier *m* represents a function

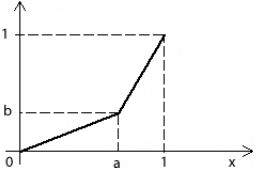
$$f_m \colon [0,1] \to [0,1]$$

Example: $f_{\texttt{very}}(x) = x^2$, $f_{\texttt{more\_or\_less}}(x) = tri(0, x, 1)$, $f_{\texttt{slightly}}(x) = \sqrt{x}$

- Modelling the fuzzy set of very heavy rain:

$$\begin{aligned}\chi_{\texttt{very heavy rain}}(x) &= f_{\texttt{very}}(\chi_{\texttt{heavyrain}}(x)) \\ &= (\chi_{\texttt{heavyrain}}(x))^2 \\ &= (rs(5, 7.5)(x))^2\end{aligned}$$

- A typical shape of modifiers: linear modifiers $lm(a, b)$



- Note: linear modifiers require one parameter c only

$$lm(a, b) = lm(c)$$

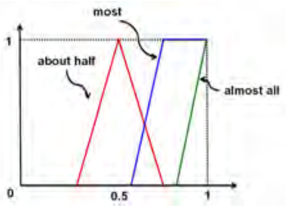
where $a = c/(c + 1)$, $b = 1/(c + 1)$

Fuzzy Quantifiers

- Classical logic has two quantifiers:
  - the universal $\forall$
  - the existential $\exists$
- These are extremal ones among several other linguistic quantifiers, such as
  - all, most, many, about half, few, some
- A quantifier, such as *most*, can be represented as a fuzzy subset ($r \in [0, 1]$)

$$Q : [0, 1] \rightarrow [0, 1]$$

with $Q(0) = 0$, $Q(1) = 1$
  - the membership grade $Q(r)$ indicates the degree to which the proportion r satisfies the linguistic quantifier that Q represents



- Degree of truth of "Most birds fly" is

$$most(\frac{|Bird \wedge Fly|}{|Fly|})$$

Mathematical Fuzzy Logics Basics

- Classical Logics for KR are grounded on Mathematical Logic
- Fuzzy Logics for KR are grounded on Mathematical Fuzzy Logic
- A statement has a degree of truth
- Truth space: set of truth values L
- Given a statement ϕ
  - Fuzzy Interpretation: a function $\mathcal{I}$ mapping ϕ into L, i.e.

$$\mathcal{I}(\varphi) \in L$$

  - Usually

$$\begin{aligned} L &= [0,1] \\ L_n &= \{0, \frac{1}{n}, \ldots, \frac{n-2}{n-1}, \ldots, 1\} \quad (n \geq 1) \end{aligned}$$

- Fuzzy statement: for $r \in [0, 1]$

$$\langle \phi, r \rangle$$

The degree of truth of ϕ is equal or greater than r

- Examples:
  - Fuzzy FOL: $\langle \mathit{RainyDay}(d), 0.75 \rangle$
  - Fuzzy LPs: $\langle \mathit{RainyDay}(d) \leftarrow, 0.75 \rangle$
  - Fuzzy RDFS: $\langle \langle d, \mathit{type}, \mathit{RainyDay} \rangle, 0.75 \rangle$
  - Fuzzy DLs: $\langle d{:}\mathit{RainyDay}, 0.75 \rangle$

- Fuzzy interpretation $\mathcal{I}$:
  - Maps each basic statement p_i into $[0, 1]$
  - Extended inductively to all statements

$$
\begin{array}{lcl}
\mathcal{I}(\phi \land \psi) & = & \mathcal{I}(\phi) \otimes \mathcal{I}(\psi) \\
\mathcal{I}(\phi \lor \psi) & = & \mathcal{I}(\phi) \oplus \mathcal{I}(\psi) \\
\mathcal{I}(\phi \rightarrow \psi) & = & \mathcal{I}(\phi) \rightarrow \mathcal{I}(\psi) \\
\mathcal{I}(\phi \leftrightarrow \psi) & = & \mathcal{I}(\phi \rightarrow \psi) \otimes \mathcal{I}(\psi \rightarrow \phi) \\
\mathcal{I}(\neg \phi) & = & \ominus \mathcal{I}(\phi) \\
\mathcal{I}(\exists x.\phi) & = & \sup_{a \in \Delta^{\mathcal{I}}} \mathcal{I}^a_x(\phi) \\
\mathcal{I}(\forall x.\phi) & = & \inf_{a \in \Delta^{\mathcal{I}}} \mathcal{I}^a_x(\phi) \ ,
\end{array}
$$

where

- $\Delta^{\mathcal{I}}$ is the domain of $\mathcal{I}$
- $\otimes$, $\oplus$, $\rightarrow$, and $\ominus$ are the t-norms, t-conorms, implication functions, a negation functions
- The function $\mathcal{I}^a_x$ is as $\mathcal{I}$ except that x is interpreted as a

Example

In Propositional Lukasiewicz logic:

$$\varphi = \mathit{Cold} \wedge \mathit{Cloudy}$$

$\mathcal{I}$	*Cold*	*Cloudy*	$\mathcal{I}(\varphi)$
$\mathcal{I}_1$	0	0.1	$\max(0, 0 + 0.1 - 1) = 0.0$
$\mathcal{I}_2$	0.3	0.4	$\max(0, 0.3 + 0.4 - 1) = 0.0$
$\mathcal{I}_3$	0.7	0.8	$\max(0, 0.7 + 0.8 - 1) = 0.5$
$\mathcal{I}_4$	1	1	$\max(0, 1 + 1 - 1) = 1.0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- One may also consider the following abbreviations:

$$\begin{aligned}
\phi \wedge_g \psi &\stackrel{\text{def}}{=} \phi \wedge (\phi \to \psi) \\
\phi \vee_g \psi &\stackrel{\text{def}}{=} (\phi \to \psi) \to \phi) \wedge_g (\psi \to \phi) \to \psi) \\
\neg_{\otimes}\phi &\stackrel{\text{def}}{=} \phi \to 0 \\
\langle \phi \leq r \rangle &\stackrel{\text{def}}{=} \langle \neg_l \phi, 1 - r \rangle
\end{aligned}$$

- In case $\to$ is the r-implication based on $\otimes$, then
  - $\wedge_g$ is Gödel t-norm
  - $\vee_g$ is Gödel s-norm
  - $\neg_{\otimes}$ is interpreted as the negation function related to $\otimes$

- $\mathcal{I}$ satisfies $\langle \phi, r \rangle$, or $\mathcal{I}$ is a model of $\langle \phi, r \rangle$

$$\mathcal{I} \models \langle \phi, r \rangle \text{ iff } \mathcal{I}(\phi) \geq r$$

- $\mathcal{I}$ is a model of ϕ if $\mathcal{I}(\phi) = 1$
- Fuzzy knowledge base $\mathcal{K}$: finite set of fuzzy statements
- $\mathcal{I}$ satisfies (is a model of) $\mathcal{K}$: $\mathcal{I} \models \mathcal{K}$ iff it satisfies each element in it
- Best entailment degree of ϕ w.r.t. $\mathcal{K}$:

$$bed(\mathcal{K}, \phi) = \sup \{ r \mid \mathcal{K} \models \langle \phi, r \rangle \}$$

- Best satisfiability degree of ϕ w.r.t. $\mathcal{K}$:

$$bsd(\mathcal{K}, \phi) = \sup_{\mathcal{I}} \{ \mathcal{I}(\phi) \mid \mathcal{I} \models \mathcal{K} \}$$

- Fuzzy Modus Ponens: for r-implication $\to$, for $r, s \in [0, 1]$:

$$\langle \phi, r \rangle, \langle \phi \to \psi, s \rangle \models \langle \psi, r \otimes s \rangle$$

  Informally,

$$\text{from } \varphi \geq r \text{ and } (\varphi \to \psi) \geq s \text{ infer } \psi \geq r \wedge s$$

- Salient equivalences:

$$\begin{aligned} \neg\neg\phi &\equiv \phi \ (\text{Ł}, SFL) \\ \phi \wedge \phi &\equiv \phi \ (G, SFL) \\ \neg(\phi \wedge \neg\phi) &\equiv 1 \ (\text{Ł}, G, \Pi) \\ \phi \vee \neg\phi &\equiv 1 \ (\text{Ł}) \end{aligned}$$

- Salient equivalences:

$$
\begin{aligned}
Ł + G &\equiv \text{Boolean Logic} \\
Ł + \Pi &\equiv \text{Boolean Logic} \\
\mathrm{G} + \Pi &\equiv \text{Boolean Logic}
\end{aligned}
$$

Example

In Lukasiewicz logic:

$$\varphi = \langle \mathit{Cold} \wedge \mathit{Cloudy}, 0.4 \rangle$$

Read: $\mathit{Cold} \wedge \mathit{Cloudy} \geq 0.4$

$\mathcal{I}$	*Cold*	*Cloudy*	$\mathcal{I}(\varphi)$
$\mathcal{I}_1$	0	0.1	$0.4 \rightarrow 0.0 = \min(1, 1 - 0.4 + 0.0) = 0.6$
$\mathcal{I}_2$	0.3	0.4	$0.4 \rightarrow 0.0 = \min(1, 1 - 0.4 + 0.0) = 0.6$
$\mathcal{I}_3$	0.7	0.8	$0.4 \rightarrow 0.5 = \min(1, 1 - 0.4 + 0.5) = 1.0$
$\mathcal{I}_4$	1	1	$0.4 \rightarrow 1.0 = \min(1, 1 - 0.4 + 1.0) = 1.0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\begin{array}{ccc} \mathcal{I}_1 & \not\models & \varphi \\ \mathcal{I}_2 & \not\models & \varphi \\ \mathcal{I}_3 & \models & \varphi \\ \mathcal{I}_4 & \models & \varphi \\ \vdots & \vdots & \vdots \end{array}$$

On Witnessed Models

- Witnessed interpretation $\mathcal{I}$:

$$\begin{aligned} \mathcal{I}(\exists x.\phi) &= \mathcal{I}_x^a(\phi), \text{ for some } a \in \Delta^{\mathcal{I}} \quad (3) \\ \mathcal{I}(\forall x.\phi) &= \mathcal{I}_x^a(\phi), \text{ for some } a \in \Delta^{\mathcal{I}} \quad (4) \end{aligned}$$

- The supremum (resp. infimum) are attained at some point
- Classical interpretations are witnessed
- Fuzzy interpretations may not be witnessed
- E.g., $\mathcal{I}$ is not witnessed as Eq. (3) not satisfied:

$$\begin{aligned} \Delta^{\mathcal{I}} &= \mathbb{N} \\ \mathcal{I}_x^n(A(x)) &= 1 - 1/n < 1, \text{ for all } n \\ \mathcal{I}(\exists x.A(x)) &= \sup_n \mathcal{I}_x^n(A(x)) \\ &= \sup_n 1 - 1/n = 1 \end{aligned}$$

Proposition (Witnessed model property)

In Łukasiewicz logic and SFL over $L = [0, 1]$, or for all cases in which the truth space L is finite, a fuzzy KB has a witnessed fuzzy model iff it has a fuzzy model.

- Not true for Gödel and product logic over $L = [0, 1]$
  - $\neg\forall x\, p(x) \wedge \neg\exists x\, \neg p(x)$ has no classical model
  - In Gödel logic it has no finite model, but has an infinite model: for integer $n \geq 1$, let $\mathcal{I}$ such that $\mathcal{I}(p(n)) = 1/n$

$$\begin{aligned}\mathcal{I}(\forall x\, p(x)) &= \inf_n 1/n = 0 \\ \mathcal{I}(\exists x\, \neg p(x)) &= \sup_n \neg 1/n = \sup 0 = 0\end{aligned}$$

- IMHO: non-witnessed models make little sense in KR
- We will always assume that interpretations are witnessed

Fuzzy Propositional Logic: Reasoning

- We need to distinguish if truth space is $L = [0, 1]$ or $L_n = \{0, \frac{1}{n}, \ldots, \frac{n-2}{n-1}, \ldots, 1\}$
- Case L_n easier: given m propositional letters, there are m^n possible interpretations
- We may use
  - Operational Research
  - Analytic Tableaux, Non-Deterministic Analytic Tableaux
  - Reduction into Classical Propositional Logic

Operational Research: Case Łukasiewicz Logic & SFL

- Basic idea: translate formulae into equational constraints about truth degrees
- For a formula ϕ consider a variable x_ϕ
  - Intuition: x_ϕ will hold the degree of truth of statement ϕ
  - Example: constraints under Łukasiewicz for $\langle \neg\phi, 0.6 \rangle$

$$
\begin{aligned}
x_{\neg\phi} &\in [0,1] \\
x_{\phi} &\in [0,1] \\
x_{\neg\phi} &= 1 - x_{\phi}
\end{aligned}
$$

- We may use Mixed Integer Linear Programming for the encodings of constraints
  For Łukasiewicz:
  - $x_1 \otimes_l x_2 = z$
    $\mapsto \{x_1 + x_2 - 1 \leq z, x_1 + x_2 - 1 \geq z - y, z \leq 1 - y, y \in \{0, 1\}\}$,
    where y is a new variable.
  - $x_1 \oplus_l x_2 = z \mapsto \{x_1 + x_2 \leq z + y, y \leq z, x_1 + x_2 \geq z, y \in \{0, 1\}\}$,
    where y is a new variable.
  - $x_1 \rightarrow_l x_2 = z \mapsto \{(1 - x_1) \oplus_l x_2 = z\}$.

  For SFL:
  - $x_1 \otimes_g x_2 = z$
    $\mapsto \{z \leq x_1, z \leq x_2, x_1 \leq z + y, x_2 \leq z + (1 - y), y \in \{0, 1\}\}$,
    where y is a new variable.
  - $x_1 \oplus_g x_2 = z$
    $\mapsto \{z \geq x_1, z \geq x_2, x_1 + y \geq z, x_2 + (1 - y) \geq z, y \in \{0, 1\}\}$,
    where y is a new variable.
  - $x_1 \rightarrow_{kd} x_2 = z \mapsto (1 - x_1) \oplus_g x_2 = z$.

- Negation Normal Form, $nnf(\phi)$

$$\begin{array}{rcl}
\neg\bot & = & \top \\
\neg\top & = & \bot \\
\neg\neg\phi & \mapsto & \phi \\
\neg(\phi \wedge \psi) & \mapsto & \neg\phi \vee \neg\psi \\
\neg(\phi \vee \psi) & \mapsto & \neg\phi \wedge \neg\psi \\
\neg(\phi \rightarrow \psi) & \mapsto & \phi \wedge \neg\psi \ .
\end{array}$$

1. Transform $\mathcal{K}$ into NNF
2. Initialize the fuzzy theory $\mathcal{T}_{\mathcal{K}}$ and the initial set of constraints $\mathcal{C}_{\mathcal{K}}$ by

$$\begin{aligned} \mathcal{T}_{\mathcal{K}} &= \{\phi \mid \langle \phi, n \rangle \in \mathcal{K}\} \\ \mathcal{C}_{\mathcal{K}} &= \{x_{\psi} \geq n \mid \langle \phi, n \rangle \in \mathcal{K}\} \end{aligned}$$

3. Apply the following inference rules until no more rules can be applied

(var). For variable x_{ϕ} occurring in $\mathcal{C}_{\mathcal{K}}$ add $x_{\phi} \in [0, 1]$ to $\mathcal{C}_{\mathcal{K}}$
($\bar{var}$). For variable $x_{\neg\phi}$ occurring in $\mathcal{C}_{\mathcal{K}}$ add $x_{\phi} = 1 - x_{\neg\phi}$ to $\mathcal{C}_{\mathcal{K}}$
($\bot$). If $\bot \in \mathcal{T}_{\mathcal{K}}$ then $\mathcal{C}_{\mathcal{K}} := \mathcal{C}_{\mathcal{K}} \cup \{x_{\bot} = 0\}$
($\top$). If $\top \in \mathcal{T}_{\mathcal{K}}$ then $\mathcal{C}_{\mathcal{K}} := \mathcal{C}_{\mathcal{K}} \cup \{x_{\top} = 1\}$

($\wedge$). If $\phi \wedge \psi \in \mathcal{T}_{\mathcal{K}}$, then

3.1 add ϕ and ψ to $\mathcal{T}_{\mathcal{K}}$

3.2 $\mathcal{C}_{\mathcal{K}} := \mathcal{C}_{\mathcal{K}} \cup \{x_{\phi} \otimes x_{\psi} = x_{\phi \wedge \psi}\}$

($\vee$). If $\phi \vee \psi \in \mathcal{T}_{\mathcal{K}}$, then

3.1 add ϕ and ψ to $\mathcal{T}_{\mathcal{K}}$

3.2 $\mathcal{C}_{\mathcal{K}} := \mathcal{C}_{\mathcal{K}} \cup \{x_{\phi} \oplus x_{\psi} = x_{\phi \wedge \psi}\}$

($\rightarrow$). If $\phi \rightarrow \psi \in \mathcal{T}_{\mathcal{K}}$, then

3.1 add $nnf(\neg\phi)$ and ψ to $\mathcal{T}_{\mathcal{K}}$

3.2 $\mathcal{C}_{\mathcal{K}} := \mathcal{C}_{\mathcal{K}} \cup \{(1 - x_{nnf(\neg\phi)}) \rightarrow x_{\psi} = x_{\phi \rightarrow \psi}\}$

sat($\mathcal{K}$): $\mathcal{K}$ is satisfiable iff the final set of constraints $\mathcal{C}_{\mathcal{K}}$ has a solution

bed($\mathcal{K}, \phi$):

- Add $\neg\phi$ to $\mathcal{T}_{\mathcal{K}}$
- Add $x_{\neg\phi} \geq 1 - x, x \in [0, 1]$ to $\mathcal{C}_{\mathcal{K}}$, x new
- Compute final set of constraints $\mathcal{C}_{\mathcal{K}}$
- Then, solve the optimisation problem

$$bed(\mathcal{K}, \phi) = \min x. \text{ such that } \mathcal{C}_{\mathcal{K}} \text{ has a solution}$$

bsd($\mathcal{K}, \phi$):

- Add ϕ to $\mathcal{T}_{\mathcal{K}}$
- Add $x_{\phi} \geq x, x \in [0, 1]$ to $\mathcal{C}_{\mathcal{K}}$, x new
- Compute final set of constraints $\mathcal{C}_{\mathcal{K}}$
- Then, solve the optimisation problem

$$bsd(\mathcal{K}, \phi) = \max x. \text{ such that } \mathcal{C}_{\mathcal{K}} \text{ has a solution}$$

Analytical Fuzzy Tableau: Case SFL

- Main property the method is based on:
  - if $\mathcal{I}$ is model of $\langle \phi \wedge \psi, n \rangle$ then $\mathcal{I}$ is a model of both $\langle \phi, n \rangle$ and $\langle \psi, n \rangle$;
  - if $\mathcal{I}$ is model of $\langle \phi \vee \psi, n \rangle$ then $\mathcal{I}$ is a model of either $\langle \phi, n \rangle$ or $\langle \psi, n \rangle$.
  - $\mathcal{I}$ cannot be a model of both $\langle p, n \rangle$ and $\langle \neg p, m \rangle$ if $n > 1 - m$.
- A clash is either
  - a fuzzy statement $\langle \bot, n \rangle$ with $n > 0$; or
  - a pair of fuzzy statements $\langle p, n \rangle$ and $\langle \neg p, m \rangle$ with $n > 1 - m$
- Clash-free: does not contain a clash

1. Transform $\mathcal{K}$ into NNF
2. Initialize the completion $S_{\mathcal{K}} = \mathcal{K}$
3. Apply the following inference rules to $S_{\mathcal{K}}$ until no more rules can be applied
4. We call a set of fuzzy statements $S_{\mathcal{K}}$ complete iff none of the rules below can be applied to $S_{\mathcal{K}}$
5. Note that rule $(\vee)$ is non-deterministic

$(\wedge)$. If $\langle \phi \wedge \psi, n \rangle \in S_{\mathcal{K}}$ and $\{\langle \phi, n \rangle, \langle \psi, n \rangle\} \not\subseteq S_{\mathcal{K}}$, then add both $\langle \phi, n \rangle$ and $\langle \psi, n \rangle$ to $S_{\mathcal{K}}$

$(\vee)$. If $\langle \phi \vee \psi, n \rangle \in S_{\mathcal{K}}$ and $\{\langle \phi, n \rangle, \langle \psi, n \rangle\} \cap S_{\mathcal{K}} = \emptyset$, then add either $\langle \phi, n \rangle$ *or* $\langle \psi, n \rangle$ to $S_{\mathcal{K}}$

$(\rightarrow)$. If $\langle \phi \rightarrow \psi, n \rangle \in S_{\mathcal{K}}$ and $\langle nnf(\neg \phi) \vee \psi, n \rangle \notin S_{\mathcal{K}}$, then add $\langle nnf(\neg \phi) \vee \psi, n \rangle$ to $S_{\mathcal{K}}$

sat$(\mathcal{K})$: $\mathcal{K}$ is satisfiable iff we find a complete and clash-free completion $S_{\mathcal{K}}$ of $\mathcal{K}$

- For BED and BSD we need some more work
- Given $\mathcal{K}$, define

$$\begin{array}{lcl} N^{\mathcal{K}} & = & \{0, 0.5, 1\} \ \cup \ \{n \mid \langle \phi, n \rangle \in \mathcal{K}\} \\ \bar{N}^{\mathcal{K}} & = & N^{\mathcal{K}} \cup \{1 - n \mid n \in N^{\mathcal{K}}\} \\ \epsilon & = & \min\{d/2 \mid n, m \in \bar{N}^{\mathcal{K}}, n \neq m, d = |n - m|\} \end{array}$$

Proposition

Under SFL, given $\mathcal{K}$, then for $n > 0$

$$\mathcal{K} \models \langle \phi, n \rangle \textit{ iff } \mathcal{K} \cup \{\langle \neg\phi, 1 - n + \epsilon \rangle\} \textit{ is not satisfiable .}$$

Moreover, $\mathcal{K}$ is satisfiable iff it has a model over $\bar{N}^{\mathcal{K}}$.

$bed(\mathcal{K}, \phi)$: Find greatest $n \in \bar{N}^{\mathcal{K}}$ such that $\mathcal{K} \models \langle \phi, n \rangle$

$bsd(\mathcal{K}, \phi)$: Find greatest $n \in \bar{N}^{\mathcal{K}}$ such that $\mathcal{K} \cup \{\langle \phi, n \rangle\}$ satisfiable

Non Deterministic Analytic Fuzzy Tableau

- Works for finitely-valued fuzzy propositional logic over L_n
- Works also for SFL (as in place of $[0,1]$, we may use $\bar{N}^{\mathcal{K}}$)
- Basic idea is as for fuzzy tableau, but now we guess the truth degrees
  - ($\wedge$). If $\langle \phi \wedge \psi, n \rangle \in \mathrm{S}_{\mathcal{K}}$, $n_1, n_2 \in L_n$ such that $n_1 \otimes n_2 = n$ and $\{\langle \phi, n_1 \rangle, \langle \psi, n_2 \rangle\} \not\subseteq \mathrm{S}_{\mathcal{K}}$, then add both $\langle \phi, n_1 \rangle$ and $\langle \psi, n_2 \rangle$ to $\mathrm{S}_{\mathcal{K}}$
  - ($\vee$). If $\langle \phi \vee \psi, n \rangle \in \mathrm{S}_{\mathcal{K}}$, $n_1, n_2 \in L_n$ such that $n_1 \oplus n_2 = n$ and $\{\langle \phi, n_1 \rangle, \langle \psi, n_2 \rangle\} \not\subseteq \mathrm{S}_{\mathcal{K}}$, then add both $\langle \phi, n_1 \rangle$ and $\langle \psi, n_2 \rangle$ to $\mathrm{S}_{\mathcal{K}}$
  - ($\rightarrow$). If $\langle \phi \rightarrow \psi, n \rangle \in \mathrm{S}_{\mathcal{K}}$, $n_1, n_2 \in L_n$ such that $n_1 \rightarrow n_2 = n$ and $\{\langle \phi, n_1 \rangle, \langle \psi, n_2 \rangle\} \not\subseteq \mathrm{S}_{\mathcal{K}}$, then add both $\langle \phi, n_1 \rangle$ and $\langle \psi, n_2 \rangle$ to $\mathrm{S}_{\mathcal{K}}$
- A clash is either
  - a fuzzy statement $\langle \bot, n \rangle$ with $n > 0$; or
  - a pair of fuzzy statements $\langle p, n \rangle$ and $\langle \neg p, m \rangle$ such that

$$x_p \geq n, \quad \ominus x_p \geq m, x_p \in L_n$$

  has no solution

Reduction to Classical Propositional Logic: Case SFL over $[0, 1]$

- Given $\mathcal{K}$, we know that we can use

$$L_n = \bar{N}^{\mathcal{K}} = \{\gamma_1, \ldots, \gamma_n\}$$

  with $\gamma_i < \gamma_{i+1}, 1 \leq i \leq n - 1$
- Basic idea: use atom $A_{\geq r}$ to represent

  The truth degree of A has to be equal or greater than r

- Similarly for $A_{>r}$, $A_{\leq r}$ and $A_{<r}$

- To start with, build $Crisp_{L_n}$
  - For all atoms A, for all $1 \leq i \leq n-1, 2 \leq j \leq n-1$

$$A_{\geq \gamma_{i+1}} \rightarrow A_{> \gamma_i}$$
$$A_{> \gamma_j} \rightarrow A_{\geq \gamma_j}$$

- Build $Crisp_{\mathcal{K}}$:

$$Crisp_{\mathcal{K}} = \{\rho(\phi, n) \mid \langle \phi, n \rangle \in \mathcal{K}\} \cup Crisp_{L_n} ,$$

x	y	$\rho(x, y)$
$\top$	c	$\top$
$\bot$	0	$\top$
$\bot$	c	$\bot$ if $c > 0$
A	c	$A_{\geq c}$
$\neg A$	c	$\neg A_{>1-c}$
$\phi \wedge \psi$	c	$\rho(\phi, c) \wedge \rho(\psi, c)$
$\phi \vee \psi$	c	$\rho(\phi, c) \vee \rho(\psi, c)$

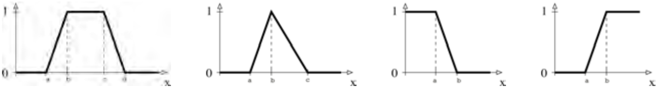
Proposition

Given $\mathcal{K}$ under SFL over L_n, then $\mathcal{K} \models \langle \phi, c \rangle$ iff $\mathcal{K} \cup \{\langle \neg\phi, 1 - c^- \rangle\}$ is not satisfiable, where c^- is the next smaller value than c in L_n

- *sat*($\mathcal{K}$): $\mathcal{K}$ is satisfiable iff $Crisp_{\mathcal{K}}$ satisfiable
- *bed*($\mathcal{K}, \phi$): Find greatest $c \in L_n$ such that $\mathcal{K} \models \langle \phi, c \rangle$
- *bsd*($\mathcal{K}, \phi$): Find greatest $c \in L_n$ such that $\mathcal{K} \cup \{\langle \phi, c \rangle\}$ satisfiable

Fuzzy Concrete Domains

- Allows us to deal with concepts such as young, cheap, cold, etc.
- We allow also crisp constraints such as AlarmSystem $\wedge$ (price $>$ 26,000), AlarmSystem $\rightarrow$ (deliverytime $\geq$ 30)
- Fuzzy membership functions: usually of the form



(a) (b) (c) (d)

Figure: (a) Trapezoidal function $trz(a, b, c, d)$, (b) triangular function $tri(a, b, c)$, (c) left shoulder function $ls(a, b)$, and (d) right shoulder function $rs(a, b)$.

- For instance, AlarmSystem $\wedge$ (price $ls(18000, 22000)$)

Definition (The language $\mathcal{P}(\mathcal{N})$)

Let $\mathcal{A}$ be a set of propositional atoms, and $\mathcal{F}$ a set of pairs $\langle f, D_f \rangle$ each made of a feature name and an associated concrete domain D_f, and let k be a value in D_f. Then the following formulae are in $\mathcal{P}(\mathcal{N})$:

1. every atom $A \in \mathcal{A}$ is a formula
2. if $\langle f, D_f \rangle \in \mathcal{F}$, $k \in D_f$, and $c \in \{\geq, \leq, =\}$ then $(f\ c\ k)$ is a formula
3. if $\langle f, D_f \rangle \in \mathcal{F}$ and c is of the form $ls(a, b)$, $rs(a, b)$, $tri(a, b, c)$, $trz(a, b, c, d)$ then $(f\ c)$ is a formula
4. if ψ and φ are formulae and $n \in [0, 1]$ then so are $\neg\psi$, $\psi \wedge \varphi$, $\psi \vee \varphi$, $\psi \rightarrow \varphi$. We use $\psi \leftrightarrow \varphi$ in place of $(\psi \rightarrow \varphi) \wedge (\varphi \rightarrow \psi)$,
5. if $\psi_1, \ldots, \psi_n$ are formulae, then $w_1 \cdot \psi_1 + \ldots + w_n \cdot \psi_n$ is a formula, where $w_i \in [0, 1]$ and $\sum_i w_i \leq 1$
6. if ψ is a formula and $n \in [0, 1]$ then $\langle \psi, n \rangle$ is a formula in $\mathcal{P}(\mathcal{N})$. If n is omitted, then $\langle \psi, 1 \rangle$ is assumed

Definition (Interpretation and models)

An interpretation $\mathcal{I}$ for $\mathcal{P}(\mathcal{N})$ is a function (denoted as a superscript $\cdot^{\mathcal{I}}$ on its argument) that maps each atom in $\mathcal{A}$ into a truth value $A^{\mathcal{I}} \in [0, 1]$, each feature name f into a value $f^{\mathcal{I}} \in D_f$, and assigns truth values in $[0, 1]$ to formulas as follows:

- for hard constraints, $(f\ c\ k)^{\mathcal{I}} = 1$ iff the relation $f^{\mathcal{I}}\ c\ k$ is true in D_f, $(f\ c\ k)^{\mathcal{I}} = 0$ otherwise
- for soft constraints, $(f\ c)^{\mathcal{I}} = c(f^{\mathcal{I}})$, i.e., the result of evaluating the fuzzy membership function c on the value $f^{\mathcal{I}}$
- $(\neg\psi)^{\mathcal{I}} = \neg\psi^{\mathcal{I}}$, $(\psi \wedge \varphi)^{\mathcal{I}} = \psi^{\mathcal{I}} \wedge \varphi^{\mathcal{I}}$, $(\psi \vee \varphi)^{\mathcal{I}} = \psi^{\mathcal{I}} \vee \varphi^{\mathcal{I}}$, $(\psi \rightarrow \varphi)^{\mathcal{I}} = \psi^{\mathcal{I}} \Rightarrow \varphi^{\mathcal{I}}$ and $(w_1 \cdot \psi_1 + \ldots + w_n \cdot \psi_n)^{\mathcal{I}} = \sum_i w_i \cdot \psi_i^{\mathcal{I}}$
- $\mathcal{I} \models \langle \psi, n \rangle$ iff $\psi^{\mathcal{I}} \geq n$.

Proposition (Reasoning)

Reasoning problems in $\mathcal{P}(\mathcal{N})$ can be solved via MILP, as rs, ls, tri are MILP representable.

Example: Matchmaking

- Suppose we have a buyer and a seller (agents)
  - A car seller sells a sedan car
  - A buyer is looking for a second hand passenger car
  - Both the buyer as well as the seller have preferences (restrictions)
  - There is some background knowledge
- The objective is determine “an optimal” (Pareto optimal) agreement among the two

Matchmaking Example: the Background Knowledge

1. A sedan is a passenger car
2. A satellite alarm system is an alarm system
3. The navigator pack is a satellite alarm system with a GPS system
4. The Insurance Plus package is a driver insurance together with a theft insurance
5. The car colours are black or grey

Matchmaking Example: Buyer's preferences

1. He does not want to pay more than 26000 euro (buyer reservation value)
2. He wants an alarm system in the car and he is completely satisfied with paying no more than 23000 euro, but he can go up to 26000 euro to a lesser degree of satisfaction
3. He wants a driver insurance and either a theft insurance or a fire insurance
4. He wants air conditioning and the external colour should be either black or grey
5. Preferably the price is no more than 22000 euro, but he can go up to 24000 euro to a lesser degree of satisfaction
6. The kilometer warranty is preferrably at least 140000, but he may go down to 160000 to a lesser degree of satisfaction
7. The weights of the preferences 2-6 are, (0.1, 0.2, 0.1, 0.2, 0.4). The higher the value the more important is the preference

Matchmaking Example: Seller's preferences

1. He wants to sell no less than 24000 euro (seller reservation value)
2. If there is an navigator pack system in the car then he is completely satisfied with selling no less than 26000 euro, but he can go down to 24000 euro to a lesser degree of satisfaction
3. Preferably the seller sells the Insurance Plus package
4. The kilometer warranty is preferrably at most 150000, but he may go up to 170000 to a lesser degree of satisfaction
5. If the color is black then the car has air conditioning
6. The weights of the preferences 2-5 are, (0.3, 0.1, 0.4, 0.2). The higher the value the more important is the preference

Matchmaking Example: Encoding

$$\mathcal{T} = \begin{cases} \text{Sedan} \rightarrow \text{PassengerCar} \\ \text{ExternalColorBlack} \rightarrow \neg\text{ExternalColorGray} \\ \text{SatelliteAlarm} \rightarrow \text{AlarmSystem} \\ \text{InsurancePlus} \leftrightarrow \text{DriverInsurance} \wedge \text{TheftInsurance} \\ \text{NavigatorPack} \leftrightarrow \text{SatelliteAlarm} \wedge \text{GPS\_system} \end{cases}$$

Buyer's request:

β = PassengerCar $\wedge$ (price $\leq$ 26000)
β_1 = AlarmSystem $\Rightarrow$ (price , ls(23000, 26000))
β_2 = DriverInsurance $\wedge$ (TheftInsurance $\vee$ FireInsurance)
β_3 = AirConditioning $\wedge$ (ExternalColorBlack $\vee$ ExternalColorGray)
β_4 = (price , ls(22000, 24000))
β_5 = (km_warranty , rs(140000, 160000))
$\mathcal{B} = 0.1 \cdot \beta_1 + 0.2 \cdot \beta_2 + 0.1 \cdot \beta_3 + 0.2 \cdot \beta_4 + 0.2 \cdot \beta_5$

Seller's request:

σ = Sedan $\wedge$ (price $\geq$ 24000)
σ_1 = NavigatorPack $\wedge$ (price , rs(24000, 26000))
σ_2 = InsurancePlus
σ_3 = (km_warranty , ls(150000, 170000))
σ_4 = ExternalColorBlack $\wedge$ AirConditioning
$\mathcal{S} = 0.3 \cdot \sigma_1 + 0.1 \cdot \sigma_2 + 0.4 \cdot \sigma_3 + 0.2 \cdot \sigma_4$

Let

$$KB = \mathcal{T} \cup \{\beta, \sigma\} \cup \{\text{buy} \leftrightarrow \mathcal{B}, \text{sell} \leftrightarrow \mathcal{S}\}$$

Pareto optimal solution:

$$bsd(KB, \text{buy} \wedge_{\Pi} \text{sell}) = 0.651$$

In particular, the final agreement is:

$\text{Sedan}^{\bar{\mathcal{I}}} = 1.0$, $\text{PassengerCar}^{\bar{\mathcal{I}}} = 1.0$, $\text{InsurancePlus}^{\bar{\mathcal{I}}} = 1.0$, $\text{AlarmSystem}^{\bar{\mathcal{I}}} = 1.0$,
$\text{DriverInsurance}^{\bar{\mathcal{I}}} = 1.0$, $\text{AirConditioning}^{\bar{\mathcal{I}}} = 1.0$, $\text{NavigatorPack}^{\bar{\mathcal{I}}} = 1.0$,
$(\text{km\_warranty ls}(150000, 170000))^{\bar{\mathcal{I}}} = 0.5$, i.e. $\text{km\_warranty}^{\bar{\mathcal{I}}} = 160000$,
$(\text{price}, \text{ls}(23000, 26000))^{\bar{\mathcal{I}}} = 0.33$, i.e. $\text{price}^{\bar{\mathcal{I}}} = 24000$,
$\text{TheftInsurance}^{\bar{\mathcal{I}}} = 1.0$, $\text{FireInsurance}^{\bar{\mathcal{I}}} = 1.0$, $\text{ExternalColorBlack}^{\bar{\mathcal{I}}} = 1.0$, $\text{ExternalColorGray}^{\bar{\mathcal{I}}} = 0.0$.

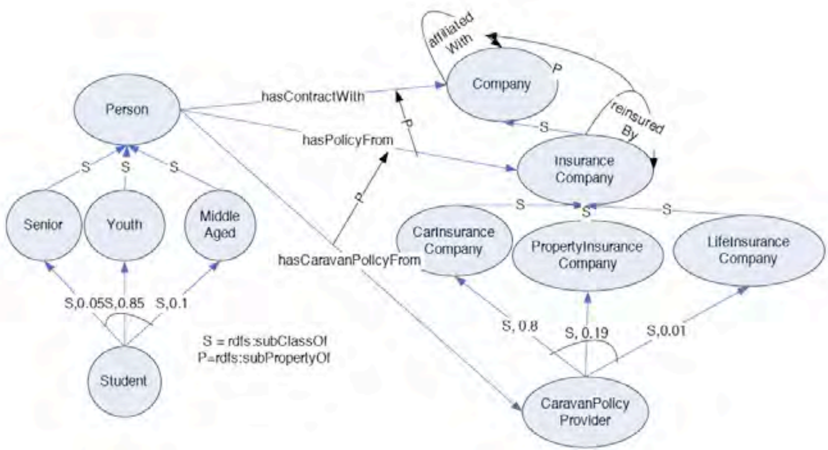
Uncertainity & Fuzzyness in Semantic Web Languages

RDFS

A Probabilistic RDF

- Probabilistic generalization of RDF
- Terminological probabilistic knowledge about classes
- Assertional probabilistic knowledge about properties of individuals
- Assertional probabilistic inference for acyclic probabilistic RDF theories, which is based on logical entailment in probabilistic logic, coupled with a local probabilistic semantics

Example of probabilistic RDF schema tuples



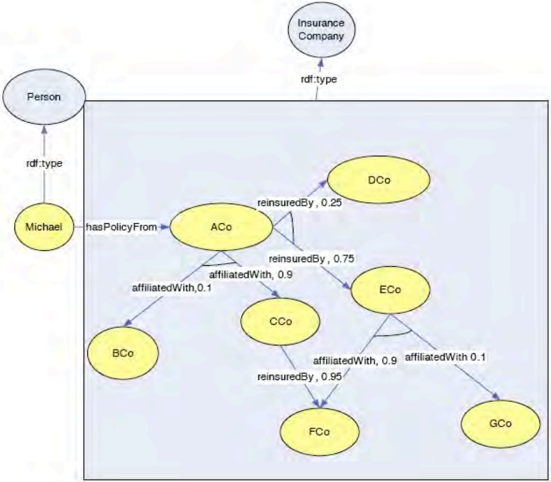
Probabilistic RDF schema tuples

- Non-probabilistic triples:

$$\begin{array}{l}(i, \text{type}, c)\\(p_1, \text{sp}, p_2)\\(p, \text{range}, c)\\(p, \text{dom}, c)\end{array}$$

  - $i \in \mathbf{UB}$ individual (URI reference or blank node)
  - p, p_i properties
  - c class
- Probabilistic schema quadruples: (c, sc, C, μ)
  - c class
  - C set of classes
  - $\mu : C \rightarrow [0, 1]$ with
    - $\sum_{c \in C} \mu(c) = 1$
    - If $(c, \text{sc}, C_1, \mu_1)$ and $(c, \text{sc}, C_2, \mu_2)$ with $C_1 \neq C_2$ then $C_1 \cap C_2 = \emptyset$

Example of probabilistic RDF instance tuples



Probabilistic RDF instance tuples

- Probabilistic instance quadruples:

$$(i, p, V, \mu)$$
$$(i, \text{type}, C, \delta)$$

  - i individual, p property
  - $V \subseteq$ **UBL**, set of individuals or literals
  - μ distribution over V, $\mu : V \to [0, 1]$ with
    - $\sum_{v \in V} \mu(v) \leq 1$
    - If (i, p, V_1, μ_1), (i, p, V_2, μ_2), with $V_1 \neq V_2$ then $V_1 \cap V_2 = \emptyset$
  - C set of classes
  - $\delta : C \to [0, 1]$ with
    - $\sum_{c \in C} \delta(c) \leq 1$
    - If $(i, \text{type}, C_1, \delta_1)$, $(i, \text{type}, C_2, \delta_2)$, then $V_1 = V_2$ and $\delta_1 = \delta_2$
- pRDF theory: a pair (S, R), where S is a set of pRDF schema tuples and R is a set of pRDF instance tuples

Semantics (excerpt)

- *p*-path *P*: for property *p*, *P* is a sequence of *n* tuples $(s_i, p_i, v_i, \gamma_i)$, where
  - for all *i*, $\exists$ (s_i, p_i, V, μ) s.t. $v_i \in V$, $\mu(v_i) = \gamma_i$
  - for all *i*, (p_i, sp^*, p) (sp^* is transitive closure of sp)
  - for all $i \leq n - 1$, $v_i = s_{i+1}$
- A pRDF instance is acyclic if for all properties *p*, there are no cyclic *p*-paths in it
- World: A world *w* is a set of triples (s, p, v) such that either
  - *s* is an individual, *p* is a property and *v* is an individual or literal, or
  - *s* is an individual, *p* is type and *v* is a class
- pRDF interpretation: $\mathcal{I} : W \rightarrow [0, 1]$ with $\sum_{w \in W} \mathcal{I}(w) = 1$

- Satisfaction:
  - $\mathcal{I} \models (s, p, V, \mu)$ iff $\forall v \in V, \mu(v) \leq \sum_{(s,p,v) \in W} \mathcal{I}((s, p, v))$
  - $\mathcal{I} \models (S, R)$ iff
    - $\mathcal{I}$ satisfies all tuples in R
    - for all p-paths $(s_i, p_i, v_i, \gamma_i)_{i \in [1 \ldots n]}$ in (S, R), $\otimes_i \gamma_i \leq \sum_{(s_i, p_i, v_i) \in W} \mathcal{I}((s_i, p_i, v_i))$
    - $\otimes$ is a t-norm
- Entailment: $(S, R) \models (s, p, V, \mu)$ iff any model of (S, R) is a model of (s, p, V, μ)
- Atomic queries: $(?s, p, v, \gamma)$, $(s, ?p, v, \gamma)$, $(s, p, v, ?\gamma)$
- Conjunctive queries: $q_1 \wedge q_2 \wedge \ldots \wedge q_n$, q_i atomic queries

Fuzzy RDF

- Statement (triples) may have attached a degree in $[0, 1]$: for $n \in [0, 1]$

$$\langle (subject, predicate, object), n \rangle$$

- Meaning: the degree of truth of the statement is at least n
- For instance,

$$\langle (o1, IsAbout, snoopy), 0.8 \rangle$$

Fuzzy RDF Syntax

- Fuzzy RDF triple (or Fuzzy RDF atom):

$$\langle \tau, n \rangle \in (\mathbf{UBL} \times \mathbf{U} \times \mathbf{UBL}) \times [0, 1]$$

  - $s \in \mathbf{UBL}$ is the subject
  - $p \in \mathbf{U}$ is the predicate
  - $o \in \mathbf{UBL}$ is the object
  - $n \in (0, 1]$ is the degree of truth
- Example:

$$\langle (\text{audiTT}, \text{type}, \text{SportCar}), 0.8 \rangle$$

Fuzzy RDF Semantics

- Fuzzy RDF interpretation $\mathcal{I}$ over a vocabulary V is a tuple

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle \ ,$$

  where
  - $\Delta_R, \Delta_P, \Delta_C, \Delta_L$ are the interpretations domains of $\mathcal{I}$
  - $P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}}$ are the interpretation functions of $\mathcal{I}$

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

1. Δ_R is a nonempty set of resources, called the domain or universe of $\mathcal{I}$;
2. Δ_P is a set of property names (not necessarily disjoint from Δ_R);
3. $\Delta_C \subseteq \Delta_R$ is a distinguished subset of Δ_R identifying if a resource denotes a class of resources;
4. $\Delta_L \subseteq \Delta_R$, the set of literal values, Δ_L contains all plain literals in $\mathbf{L} \cap V$;
5. $P[\![\cdot]\!]$ maps each property name $p \in \Delta_P$ into a partial function $P[\![p]\!] : \Delta_R \times \Delta_R \to [0, 1]$, i.e. assigns a degree to each pair of resources, denoting the degree of being the pair an instance of the property p;
6. $C[\![\cdot]\!]$ maps each class $c \in \Delta_C$ into a partial function $C[\![c]\!] : \Delta_R \to [0, 1]$, i.e. assigns a degree to every resource, denoting the degree of being the resource an instance of the class c;
7. $\cdot^{\mathcal{I}}$ maps each $t \in \mathbf{UL} \cap V$ into a value $t^{\mathcal{I}} \in \Delta_R \cup \Delta_P$, i.e. assigns a resource or a property name to each element of **UL** in V, and such that $\cdot^{\mathcal{I}}$ is the identity for plain literals and assigns an element in Δ_R to elements in **L**;
8. $\cdot^{\mathcal{I}}$ maps each variable $x \in \mathbf{B}$ into a value $x^{\mathcal{I}} \in \Delta_R$, i.e. assigns a resource to each variable in **B**.

Models

Let G be a graph over ρdf.

- An interpretation $\mathcal{I}$ is a model of G under ρdf, denoted $\mathcal{I} \models G$, iff
  - $\mathcal{I}$ is an interpretation over the vocabulary $\rho\text{df} \cup \mathit{universe}(G)$
  - $\mathcal{I}$ satisfies the following conditions:

Simple:

1. for each $\langle (s, p, o), n \rangle \in G$, $p^{\mathcal{I}} \in \Delta_P$ and $P[\![p^{\mathcal{I}}]\!](s^{\mathcal{I}}, o^{\mathcal{I}}) \geq n$;

Subproperty:

1. $P[\![\text{sp}^{\mathcal{I}}]\!]$ is transitive over Δ_P;
2. if $P[\![\text{sp}^{\mathcal{I}}]\!](p, q)$ is defined then $p, q \in \Delta_P$ and

$$P[\![\text{sp}^{\mathcal{I}}]\!](p, q) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x, y) \implies P[\![q]\!](x, y) \; ;$$

Models (cont.)

Subclass:

1. $P[\![\mathrm{sc}^{\mathcal{I}}]\!]$ is transitive over Δ_C;
2. if $P[\![\mathrm{sc}^{\mathcal{I}}]\!](c,d)$ is defined then $c, d \in \Delta_C$ and

$$P[\![\mathrm{sc}^{\mathcal{I}}]\!](c,d) = \inf_{x \in \Delta_R} C[\![c]\!](x) \implies C[\![d]\!](x) \, ;$$

Typing I:

1. $C[\![c]\!](x) = P[\![\mathrm{type}^{\mathcal{I}}]\!](x,c)$;
2. if $P[\![\mathrm{dom}^{\mathcal{I}}]\!](p,c)$ is defined then

$$P[\![\mathrm{dom}^{\mathcal{I}}]\!](p,c) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x,y) \implies C[\![c]\!](x) \, ;$$

3. if $P[\![\mathrm{range}^{\mathcal{I}}]\!](p,c)$ is defined then

$$P[\![\mathrm{range}^{\mathcal{I}}]\!](p,c) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x,y) \implies C[\![c]\!](y) \, ;$$

Typing II:

1. For each $\mathrm{e} \in \rho\mathrm{df}$, $\mathrm{e}^{\mathcal{I}} \in \Delta_P$
2. if $P[\![\mathrm{dom}^{\mathcal{I}}]\!](p,c)$ is defined then $p \in \Delta_P$ and $c \in \Delta_C$
3. if $P[\![\mathrm{range}^{\mathcal{I}}]\!](p,c)$ is defined then $p \in \Delta_P$ and $c \in \Delta_C$
4. if $P[\![\mathrm{type}^{\mathcal{I}}]\!](x,c)$ is defined then $c \in \Delta_C$

Models (cont.)

Note:

- In the crisp case, if c is a sub-class of d then we impose that $C[\![c]\!] \subseteq C[\![d]\!]$
- This may be seen as the formula

$$\forall x.c(x) \implies d(x) \ ,$$

- The fuzzyfication is

$$P[\![\mathrm{sc}^{\mathcal{I}}]\!](c,d) = \inf_{x \in \Delta_R} C[\![c]\!](x) \implies C[\![d]\!](x) \ ;$$

- Similarly, e.g., “property p has domain c” may be seen as the formula

$$\forall x \forall y.p(x,y) \implies c(x) \ ,$$

- The fuzzyfication is

$$P[\![\mathrm{dom}^{\mathcal{I}}]\!](p,c) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x,y) \implies C[\![c]\!](x) \ .$$

- G entails H under ρdf, denoted $G \models H$, iff
  - every model under ρdf of G is also a model under ρdf of H

Example & Model

$G = \{\langle(\mathit{audiTT}, \mathrm{type}, \mathit{SportsCar}), 0.8\rangle, \langle(\mathit{SportsCar}, \mathrm{sc}, \mathit{PassengerCar}), 0.9\rangle\}$ t-norm: Product

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

$$\begin{array}{rcl}
\Delta_R & = & \{\mathit{audiTT}, \mathit{SportsCar}, \mathit{PassengerCar}\} \\
\Delta_P & = & \{\mathrm{type}, \mathrm{sc}\} \\
\Delta_C & = & \{\mathit{SportsCar}, \mathit{PassengerCar}\} \\
P[\![\mathrm{type}]\!] & = & \{\langle\langle \mathit{audiTT}, \mathit{SportsCar}\rangle, 0.8\rangle, \langle\langle \mathit{audiTT}, \mathit{PassengerCar}\rangle, 0.72\rangle\} \\
P[\![\mathrm{sc}]\!] & = & \{\langle\langle \mathit{SportsCar}, \mathit{PassengerCar}\rangle, 0.9\rangle\} \\
C[\![\mathit{SportsCar}]\!] & = & \{\langle \mathit{audiTT}, 0.8\rangle\} \\
C[\![\mathit{PassengerCar}]\!] & = & \{\langle \mathit{audiTT}, 0.72\rangle\} \\
t^{\mathcal{I}} & = & t \text{ for all } t \in \mathbf{UL}
\end{array}$$

$$\mathcal{I} \models G$$

$\mathcal{I}$ is a model of G

Example (Entailment)

$G = \{\langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle, \langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle\}$ t-norm: Product

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

$$\begin{array}{rcl}
\Delta_R & = & \{\mathit{audiTT}, \mathit{SportsCar}, \mathit{PassengerCar}\} \\
\Delta_P & = & \{\text{type}, \text{sc}\} \\
\Delta_C & = & \{\mathit{SportsCar}, \mathit{PassengerCar}\} \\
P[\![\text{type}]\!] & = & \{\langle\langle \mathit{audiTT}, \mathit{SportsCar}\rangle, 0.8\rangle, \langle\langle \mathit{audiTT}, \mathit{PassengerCar}\rangle, 0.72\rangle\} \\
P[\![\text{sc}]\!] & = & \{\langle\langle \mathit{SportsCar}, \mathit{PassengerCar}\rangle, 0.9\rangle\} \\
C[\![\mathit{SportsCar}]\!] & = & \{\langle \mathit{audiTT}, 0.8\rangle\} \\
C[\![\mathit{PassengerCar}]\!] & = & \{\langle \mathit{audiTT}, 0.72\rangle\} \\
t^{\mathcal{I}} & = & t \text{ for all } t \in \mathbf{UL}
\end{array}$$

$G \models \langle(\mathit{audiTT}, \text{type}, \mathit{PassengerCar}), 0.72\rangle$ In all models $\mathcal{I}$ of G, $P[\![\text{type}]\!](\mathit{audiTT}, \mathit{PassengerCar}) = 0.72$

Deduction System for fuzzy RDF

1. Simple:

   $(a) \quad \frac{G}{G'}$ for a map $\mu : G' \to G \qquad (b) \quad \frac{G}{G'}$ for $G' \subseteq G$

2. Subproperty:

   $$(a) \quad \frac{\langle (A, \mathrm{sp}, B), n\rangle, \langle (B, \mathrm{sp}, C), m\rangle}{\langle (A, \mathrm{sp}, C), n \otimes m\rangle} \qquad (b) \quad \frac{\langle (A, \mathrm{sp}, B), n\rangle, \langle (X, A, Y), m\rangle}{\langle (X, B, Y), n \otimes m\rangle}$$

3. Subclass:

   $$(a) \quad \frac{\langle (A, \mathrm{sc}, B), n\rangle, \langle (B, \mathrm{sc}, C), m\rangle}{\langle (A, \mathrm{sc}, C), n \otimes m\rangle} \qquad (b) \quad \frac{\langle (A, \mathrm{sc}, B), n\rangle, \langle (X, \mathrm{type}, A), m\rangle}{\langle (X, \mathrm{type}, B), n \otimes m\rangle}$$

4. Typing:

   $$(a) \quad \frac{\langle (A, \mathrm{dom}, B), n\rangle, \langle (X, A, Y), m\rangle}{\langle (X, \mathrm{type}, B), n \otimes m\rangle} \qquad (b) \quad \frac{\langle (A, \mathrm{range}, B), n\rangle, \langle (X, A, Y), m\rangle}{\langle (Y, \mathrm{type}, B), n \otimes m\rangle}$$

5. Implicit Typing:

   $$(a) \quad \frac{\langle (A, \mathrm{dom}, B), n\rangle, \langle (C, \mathrm{sp}, A), m\rangle, \langle (X, C, Y), r\rangle}{\langle (X, \mathrm{type}, B), n \otimes m \otimes r\rangle}$$

   $$(b) \quad \frac{\langle (A, \mathrm{range}, B), n\rangle, \langle (C, \mathrm{sp}, A), m\rangle, \langle (X, C, Y), r\rangle}{\langle (Y, \mathrm{type}, B), n \otimes m \otimes r\rangle}$$

Deduction System for Fuzzy RDF (cont.)

- Notion of proof (as for crisp RDF)):
  - Let G and H be graphs
  - Then $G \vdash H$ iff there is a sequence of graphs $P_1, \ldots, P_k$ with $P_1 = G$ and $P_k = H$, and for each j ($2 \leq j \leq k$) one of the following holds:
    1. there exists a map $\mu : P_j \to P_{j-1}$ (rule (1a));
    2. $P_j \subseteq P_{j-1}$ (rule (1b));
    3. there is an instantiation $\frac{R}{R'}$ of one of the rules (2)–(5), such that $R \subseteq P_{j-1}$ and $P_j = P_{j-1} \cup R'$.
- The sequence of rules used at each step (plus its instantiation or map), is called a proof of H from G.

Proposition (Soundness and completeness)

The fuzzy RDF proof system $\vdash$ is sound and complete for $\models$, that is, $G \vdash H$ iff $G \models H$.

Example (Proof)

$$G = \{\langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle, \langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle\}$$

t-norm: Product

Let us proof that

$$G \models \langle(\mathit{audiTT}, \text{type}, \mathit{PassengerCar}), 0.72\rangle$$

$$\begin{array}{lll} G & \vdash & \langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle, \\ G & \vdash & \langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle \\ G & \vdash & \langle(\mathit{audiTT}, \text{type}, \mathit{PassengerCar}), 0.72\rangle \end{array}$$

(1) Rule Simple (b)
(2) Rule Simple (b)
(3) Rule SubClass (b) applied to (1) + (2) using product t-norm

Fuzzy RDFS Query Answering

- Conjunctive query: extends a crisp RDF query and is of the form

$$\begin{array}{rcl}\langle q(\mathbf{x}), s\rangle & \leftarrow & \exists \mathbf{y}.\langle \tau_1, s_1\rangle, \ldots, \langle \tau_n, s_n\rangle, \\ & & s = f(s_1, \ldots, s_n, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h))\end{array}$$

  where
  - τ_i triples involving literals and variables in $\mathbf{x}, \mathbf{y}$
  - $\mathbf{z_i}$ are tuples of literals or variables in $\mathbf{x}$ or $\mathbf{y}$
  - $p_j(\mathbf{t}) \in [0, 1]$
  - f is a *scoring* function $f\colon ([0,1])^{n+h} \to [0, 1]$
- Example:

  $\langle q(x), s\rangle \leftarrow \langle (x, \mathrm{type}, \mathrm{SportCar}), s_1\rangle, (x, \mathrm{hasPrice}, y), s = s_1 \cdot \mathrm{cheap}(y)$

  where e.g. $\mathrm{cheap}(y) = \mathrm{ls}(0, 10000, 12000)(y)$, has intended meaning to “retrieve all cheap sports car"

Fuzzy RDF Query Answering (cont.)

- We will also write a query as

$$\langle q(\mathbf{x}), s\rangle \leftarrow \exists \mathbf{y}.\langle \varphi(\mathbf{x}, \mathbf{y}), \mathbf{s}\rangle \ ,$$

  where
  - $\varphi(\mathbf{x}, \mathbf{y})$ is $\langle \tau_1, s_1\rangle, \ldots, \langle \tau_n, s_n\rangle, s = f(\mathbf{s}, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h))$
  - $\mathbf{s} = \langle s_1, \ldots, s_n\rangle$
- Furthermore, $q(\mathbf{x})$ is called the head of the query, while $\exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$ is is called the body of the query
- Finally, a disjunctive query (or, *union of conjunctive queries*) **q** is, as usual, a finite set of conjunctive queries in which all the rules have the same head
- For instance, the disjunctive query

$$\begin{array}{lll} \langle q(x), s\rangle & \leftarrow & \langle (x, \text{type}, \text{SportCar}), s_1\rangle, (x, \text{hasPrice}, y), s = s_1 \cdot \text{cheap}(y) \\ \langle q(x), s\rangle & \leftarrow & \langle (x, \text{type}, \text{PassengerCar}), s_1\rangle, s = s_1 \end{array}$$

  has intended meaning to retrieve all sports cars or passenger cars

Fuzzy RDF Query Answering (cont.)

- Consider a fuzzy graph G, a query $\langle q(\mathbf{x}), s\rangle \leftarrow \exists \mathbf{y}.\langle \varphi(\mathbf{x}, \mathbf{y}), \mathbf{s}\rangle$, and a vector $\mathbf{t}$ of terms in **UL** and $s \in [0, 1]$
- We say that $\langle q(\mathbf{t}), s\rangle$ is entailed by G, denoted $G \models \langle q(\mathbf{t}), s\rangle$, iff
  - in any model $\mathcal{I}$ of G, there is a vector $\mathbf{t}'$ of terms in **UL**, a vector $\mathbf{s}$ of scores in $[0, 1]$ such that $\mathcal{I}$ is a model of $\langle \varphi(\mathbf{t}, \mathbf{t}'), \mathbf{s}\rangle$ (the scoring atom is satisfied iff s is the value of the evaluation of the score combination function)
- For a disjunctive query $\mathbf{q} = \{q_1, \ldots, q_m\}$, we say that $\langle \mathbf{q}(\mathbf{t}), s\rangle$ is entailed by G, denoted $G \models \langle \mathbf{q}(\mathbf{t}), s\rangle$, iff $G \models \langle q_i(\mathbf{t}), s\rangle$ for some $q_i \in \mathbf{q}$
- We say that s is *tight* iff $s = \sup\{s' \mid G \models \langle \mathbf{q}(\mathbf{t}), s'\rangle\}$
- If $G \models \langle \mathbf{q}(\mathbf{t}), s\rangle$ and s is tight then $\langle \mathbf{t}, s\rangle$ is called an *answer* to $\mathbf{q}$
- The answer set of $\mathbf{q}$ w.r.t. G is defined as

$$ans(G, \mathbf{q}) = \{\langle \mathbf{t}, s\rangle \mid G \models \langle \mathbf{q}(\mathbf{t}), s\rangle,\ s \text{ is tight}\}$$

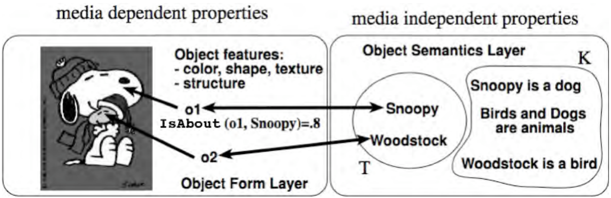
Top-k Retrieval: Given a fuzzy graph G, and a disjunctive query $\mathbf{q}$, retrieve k answers $\langle \mathbf{t}, s\rangle$ with maximal scores and rank them in decreasing order relative to the score s, denoted

$$ans_k(G, \mathbf{q}) = \mathrm{Top}_k\ ans(G, \mathbf{q})\ .$$

Fuzzy RDF Query Answering (cont.)

- A simple query answering procedure is the following:
  - Compute the closure of a graph off-line
  - Store the fuzzy RDF triples into a relational database supporting Top-k retrieval (e.g., RankSQL, Postgres)
  - Translate the fuzzy query into a top-k SQL statement
  - Execute the SQL statement over the relational database
- In practice, some care should be in place due to the large size of data: $\geq 10^9$ triples
- To date, no systems exists

Example



$$G = \left\{ \begin{array}{ll} \langle (o1, IsAbout, snoopy), 0.8 \rangle & \langle (o2, IsAbout, woodstock), 0.9 \rangle \\ (snoopy, \text{type}, dog) & (woodstock, \text{type}, bird) \\ \langle (Bird, \text{sc}, SmallAnimal), 0.7 \rangle & \langle (Dog, \text{sc}, SmallAnimal), 0.4 \rangle \\ (dog, \text{sc}, Animal) & (bird, \text{sc}, Animal) \\ (SmallAnimal, \text{sc}, Animal) & \end{array} \right\}$$

Consider the query

$$\langle q(x), s \rangle \quad \leftarrow \quad \langle (x, IsAbout, y), s_1 \rangle, \langle (y, \text{type}, SmallAnimal), s_2 \rangle, s = s_1 \cdot s_2$$

Then (under any t-norm)

$$ans(G, q) \quad = \quad \{\langle o1, 0.32 \rangle, \langle o2, 0.63 \rangle\}, \quad ans_1(G, q) = \{\langle o2, 0.63 \rangle\}$$

Description Logics

Probabilistic DLs

- Terminological probabilistic knowledge about concepts and roles
- Assertional probabilistic knowledge about instances of concepts and roles (for combining assertional and terminological probabilistic knowledge)
- Terminological and assertional probabilistic inference problems reduced to sequences of linear optimization problems

- Directly extends probabilistic propositional logic
  - In place of atoms we have now concepts as basic events
  - Finite nonempty set of basic events $\Phi = \{C_1, \ldots, C_n\}$, where C_i concept
  - Event φ: Boolean combination of basic events
- Logical constraint $\varphi \sqsubseteq \psi$: "φ is subsumed by ψ"
- Conditional constraints:
  - $(\psi|\varphi)[l, u]$: informally encodes that
    "generally, if an individual is an instance of φ, then it is an instance of ψ with a probability in $[l, u]$"
  - $a : (\psi|\varphi)[l, u]$: informally encodes that
    "if individual a is an instance of φ, then a is an instance of ψ with a probability in $[l, u]$"

Example

$$Eagle \sqsubseteq Bird$$

$$(Fly \mid Bird)[0.95, 1]$$

$$KB \| \!\models_{tight} (Fly \mid Bird)[0.95, 1]$$

$$KB \| \!\models_{tight} (Fly \mid Eagle)[0, 1.0]$$

Reasoning in Probabilistic DLs

- Similar to probabilistic propositional logic via MILP
- A world I is a finite set of basic events $C \in \Phi$ such that $\{C(a) \mid C \in I\} \cup \{\neg C(a) \mid C \in \Phi \setminus I\}$ is satisfiable, where a is a new individual
- Informally, every world I represents an individual a that is fully specified on a in the sense that I belongs (resp., does not belong) to every basic event $C \in I$ (resp., $C \in \Phi \setminus I$)
- We denote by $\mathcal{I}_\Phi$ the set of all worlds relative to Φ
  - Notice that $\mathcal{I}_\Phi$ is finite, since Φ is finite
- A world I satisfies a classical knowledge base $\mathcal{K}$, or I is a model of $\mathcal{K}$, denoted $I \models \mathcal{K}$, iff $\mathcal{K} \cup \{C(a) \mid C \in I\} \cup \{\neg C(a) \mid C \in \Phi \setminus I\}$ is satisfiable, where a is a new individual
- A world I satisfies a basic event $C \in \Phi$, or I is a model of C, denoted $\mathcal{I} \models C$ iff $C \in I$
- The notion of a world I satisfies an event C, or I is a model of C, denoted $\mathcal{I} \models C$, is defined as follows:
  - if $C \in \Phi$ is a basic event then $\mathcal{I} \models C$ iff $C \in I$
  - $I \models \neg C$ iff $I \not\models C$
  - $I \models C \sqcap D$ iff $I \models C$ and $I \models D$

Proposition

Let $\mathcal{K}$ be a classical knowledge base, and let P be a finite set of conditional constraints. Let $R = \{I \in \mathcal{I}_\Phi \mid I \models \mathcal{K}\}$. Then, $\mathcal{K} \cup P$ is satisfiable iff the system of linear constraints LC below over the variables y_r, $r \in R$ is solvable:

$$
\begin{aligned}
\sum_{r\in R,\, r\models\neg\psi\wedge\varphi} -l\, y_r + \sum_{r\in R,\, r\models\psi\wedge\varphi} (1-l)\, y_r &\geq 0 \quad (\forall(\psi|\varphi)[l,u] \in P), l > 0\\
\sum_{r\in R,\, r\models\neg\psi\wedge\varphi} u\, y_r + \sum_{r\in R,\, r\models\psi\wedge\varphi} (u-1)\, y_r &\geq 0 \quad (\forall(\psi|\varphi)[l,u] \in P), u < 1\\
\sum_{r\in R} y_r &= 1\\
y_r &\geq 0 \quad (\textit{for all } r \in R)
\end{aligned}
$$

- In order to compute the tight bounds, just

$$\text{minimize (resp., maximize)} \sum_{r\in R,\, r\models\beta\wedge\alpha} y_r \quad \text{subject to } LC$$

Fuzzy Description Logics

- In classical DLs, a concept C is interpreted by an interpretation $\mathcal{I}$ as a set of individuals
- In fuzzy DLs, a concept C is interpreted by $\mathcal{I}$ as a fuzzy set of individuals
- Each individual is instance of a concept to a degree in $[0, 1]$
- Each pair of individuals is instance of a role to a degree in $[0, 1]$

For a degree n in L or L_n

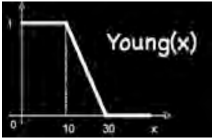
- $\langle a{:}C, n\rangle$ states that a is an instance of concept/class C with degree at least n
- $\langle C_1 \sqsubseteq C_2, n\rangle$ states that class C_1 is ausbclass of C_2 to degree n

Fuzzy OWL 2

- Fuzzy OWL 2 added value:
  - fuzzy concrete domains (e.g., *young*)
  - modifiers (e.g., *very* young)
  - other extensions:
    - aggregation functions: weighted sum, OWA, fuzzy integrals
    - fuzzy rough sets
    - fuzzy spatial relations
    - fuzzy numbers, ...

Fuzzy Concrete Domains

- E.g., *Small*, *Young*, *High*, *etc*. with explicit membership function
- Representation of Young Person:



$$\begin{aligned} \mathit{Minor} &= \mathit{Person} \sqcap \exists \mathit{hasAge}. \leq_{18} \\ \mathit{YoungPerson} &= \mathit{Person} \sqcap \exists \mathit{hasAge}.\mathit{ls}(10, 30) \end{aligned}$$

- Representation of Heavy Rain:

$$\mathit{HeavyRain} = \mathit{Rain} \sqcap \exists \mathit{hasPrecipitationRate}.\mathit{rs}(5, 7.5)$$

Fuzzy Modifiers

- *Very*, *moreOrLess*, *slightly*, etc.
- Representation of Sport Car

$$SportsCar \quad = \quad Car \sqcap \exists speed.very(rs(80, 250))$$

- Representation of Very Heavy Rain

$$VeryHeavyRain = Rain \sqcap \exists hasPrecipitationRate.very(rs(5, 7.5)) \ .$$

Aggregation Operators

- Aggregation operators: aggregate concepts, using functions such as the mean, median, weighted sum operators, etc.
- Allows to express the concept

$$0.3 \cdot \mathit{ExpensiveHotel} + 0.7 \cdot \mathit{LuxuriousHotel} \sqsubseteq \mathit{GoodHotel}$$

  - a good hotel is the weighted sum of being an expensive and luxurious hotel
- Aggregated concepts are popular in robotics:
  - to recognise complex objects from atomic ones

Semantics

The semantics is an immediate consequence of the First-Order-Logic translation of DLs expressions

Interpretation:

$$\mathcal{I} = \Delta^{\mathcal{I}}$$
$$C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$$
$$R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$$

$\otimes$ = t-norm
$\oplus$ = s-norm
$\ominus$ = negation
$\rightarrow$ = implication

Concepts:

Syntax		Semantics		
$C, D \longrightarrow$	$\top \mid$	$\top^{\mathcal{I}}(x)$	$=$	1
	$\bot \mid$	$\bot^{\mathcal{I}}(x)$	$=$	0
	$A \mid$	$A^{\mathcal{I}}(x)$	$\in$	$[0,1]$
	$C \sqcap D \mid$	$(C_1 \sqcap C_2)^{\mathcal{I}}(x)$	$=$	$C_1{}^{\mathcal{I}}(x) \otimes C_2{}^{\mathcal{I}}(x)$
	$C \sqcup D \mid$	$(C_1 \sqcup C_2)^{\mathcal{I}}(x)$	$=$	$C_1{}^{\mathcal{I}}(x) \oplus C_2{}^{\mathcal{I}}(x)$
	$\neg C \mid$	$(\neg C)^{\mathcal{I}}(x)$	$=$	$\ominus C^{\mathcal{I}}(x)$
	$\exists R.C \mid$	$(\exists R.C)^{\mathcal{I}}(x)$	$=$	$\sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)$
	$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(x)$	$=$	$\inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y)\}$

Assertions: $\langle a{:}C, r\rangle$, $\mathcal{I} \models \langle a{:}C, r\rangle$ iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq r$ (similarly for roles)

- individual a is instance of concept C at least to degree r, $r \in [0,1] \cap \mathbb{Q}$

Inclusion axioms: $\langle C \sqsubseteq D, r\rangle$,

- $\mathcal{I} \models \langle C \sqsubseteq D, r\rangle$ iff $\inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \rightarrow D^{\mathcal{I}}(x) \geq r$

Main Inference Problems

Graded entailment: Check if DL axiom α is entailed to degree at least r

- $KB \models \langle \alpha, r \rangle$?

BED: Best Entailment Degree problem

- $bed(KB, \alpha) = \sup\{r \mid KB \models \langle \alpha, r \rangle\}$

BSD: Best Satisfiability Degree problem

- $bsd(KB, C) = \sup_{\mathcal{I} \models KB}\{C^{\mathcal{I}}(a^{\mathcal{I}})\}$, for new individual a

Top-k retrieval: Retrieve the top-k individuals that instantiate C w.r.t. best truth value bound

- $ans_k(KB, C) = Top_k\{\langle a, r \rangle \mid r = bed(KB, a{:}C)\}$

Number Restrictions, Inverse and Transitive roles

- The semantics of the concept $(\geq n\ R)$ is:

$$\exists y_1, \ldots, y_n. \bigwedge_{i=1}^{n} R(x, y_i) \wedge \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j\ .$$

- The semantics of the concept $(\leq n\ R)$ is:

$$(\leq n\ R)^{\mathcal{I}}(x) = \forall y_1, \ldots, y_{n+1}. \bigwedge_{i=1}^{n+1} R(x, y_i) \to \bigvee_{1 \leq i < j \leq n+1} y_i = y_j\ .$$

- Note: $(\geq 1\ R) \equiv \exists R.\top$
- For inverse roles we have for all $x, y \in \Delta^{\mathcal{I}}$

$$R^{\mathcal{I}}(x, y) = R^{\mathcal{I}}(y, x)$$

- For transitive roles R we impose: for all $x, y \in \Delta^{\mathcal{I}}$

$$R^{\mathcal{I}}(x, y) \geq \sup_{z \in \Delta^{\mathcal{I}}} \min(R^{\mathcal{I}}(x, z), R^{\mathcal{I}}(z, y))$$

Fuzzy $\mathcal{SHOIN}(D)$

Concepts:

	Syntax		Semantics
C, D	$\longrightarrow$	$\top$ |	$\top(x)$
		$\bot$ |	$\bot(x)$
		A |	$A(x)$
		$(C \sqcap D)$ |	$C_1(x) \wedge C_2(x)$
		$(C \sqcup D)$ |	$C_1(x) \vee C_2(x)$
		$(\neg C)$ |	$\neg C(x)$
		$(\exists R.C)$ |	$\exists x\ R(x, y) \wedge C(y)$
		$(\forall R.C)$ |	$\forall x\ R(x, y) \rightarrow C(y)$
		$\{a\}$ |	$x = a$
		$(\geq n\ R)$ |	$\exists y_1, \ldots, y_n. \bigwedge_{i=1}^{n} R(x, y_i) \wedge \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j$
		$(\leq n\ R)$ |	$\forall y_1, \ldots, y_{n+1}. \bigwedge_{i=1}^{n+1} R(x, y_i) \rightarrow \bigvee_{1 \leq i < j \leq n+1} y_i = y_j$
		FCC |	$\mu_{FCC}(x)$
		$M(C)$ |	$\mu_M(C(x))$
		$\sum_i w_i \cdot C_i$	$w_1 \cdot C_1(x) + \cdots w_n \cdot C_n(x) \quad (\sum_i w_i = 1)$
R	$\longrightarrow$	P |	$P(x, y)$
		P^- |	$P(y, x)$

Assertions:

	Syntax		Semantics
α	$\longrightarrow$	$\langle a{:}C, r\rangle$ |	$r \rightarrow C(a)$
		$\langle (a, b){:}R, r\rangle$	$r \rightarrow R(a, b)$

Axioms:

	Syntax		Semantics
τ	$\longrightarrow$	$\langle C \sqsubseteq D, r\rangle$ |	$\forall x\ r \rightarrow (C(x) \rightarrow D(x))$, where $\rightarrow$ is r-implication
		$fun(R)$ |	$\forall x \forall y \forall z\ R(x, y) \wedge R(x, z) \rightarrow y = z$
		$trans(R)$	$(\exists z\ R(x, z) \wedge R(z, y)) \rightarrow R(x, y)$

Example (Graded Entailment)

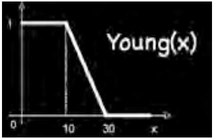
		
audi_tt	*mg*	*ferrari_enzo*

Car	*speed*
audi_tt	243
mg	≤ 170
ferrari_enzo	≥ 350

$$SportsCar = Car \sqcap \exists hasSpeed.very(High)$$

$$\begin{array}{lcl} KB & \models & \langle ferrari\_enzo{:}SportsCar, 1\rangle \\ KB & \models & \langle audi\_tt{:}SportsCar, 0.92\rangle \\ KB & \models & \langle mg{:}\neg SportsCar, 0.72\rangle \end{array}$$

Example (Graded Subsumption)

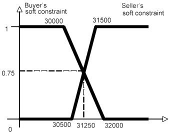


$$\begin{array}{rcl} \mathit{Minor} & = & \mathit{Person} \sqcap \exists \mathit{hasAge}. \leq_{18} \\ \mathit{YoungPerson} & = & \mathit{Person} \sqcap \exists \mathit{hasAge}.\mathit{Young} \end{array}$$

$$\mathit{KB} \models \langle \mathit{Minor} \sqsubseteq \mathit{YoungPerson}, 0.6 \rangle$$

Note: without an explicit membership function of *Young*, this inference cannot be drawn

Example (Simplified Negotiation)



- a car seller sells an Audi TT for 31500 €, as from the catalog price.
- a buyer is looking for a sports-car, but wants to to pay not more than around 30000 €
- classical DLs: the problem relies on the crisp conditions on price

- more fine grained approach: to consider prices as fuzzy sets (as usual in negotiation)
  - seller may consider optimal to sell above 31500 €, but can go down to 30500 €
  - the buyer prefers to spend less than 30000 €, but can go up to 32000 €

$$\begin{array}{lcl} \mathit{AudiTT} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.R(x; 30500, 31500) \\ \mathit{Query} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.L(x; 30000, 32000) \end{array}$$

  - highest degree to which the concept
    $C = \mathit{AudiTT} \sqcap \mathit{Query}$
    is satisfiable is 0.75 (the possibility that the Audi TT and the query matches is 0.75)

  - the car may be sold at 31250 €

Reasoning in Fuzzy $\mathcal{ALC}$, under Zadeh Semantics

- Applies technique based on Mixed Integer Programming (MILP) for fuzzy propositional logic to $\mathcal{ALC}$ calculus
- For each concept assertion α of the form $a{:}C$, we use variable x_α, which holds the degree of truth of α
- It can be shown that

$$
\begin{array}{lcl}
bed(KB, (a,b){:}R) & = & bed(KB \cup \{\langle b{:}B, 1\rangle\}, a{:}\exists R.B) \\
bed(KB, C \sqsubseteq D) & = & \min x \text{ such that } KB \cup \{\langle b{:}C \sqcap \neg D, 1-x\rangle\} \text{ satisfiable} \\
bed(KB, a{:}C) & = & \min x \text{ such that } KB \cup \{\langle a{:}\neg C, 1-x\rangle\} \text{ satisfiable} \\
bsd(KB, C) & = & \min -x \text{ such that } KB \cup \{\langle b{:}C, x\rangle\} \text{ satisfiable}
\end{array}
$$

where b is a new individual and B is a new concept

Satisfiability Testing

- The notion of completion forest $\mathcal{F}$ is similar to the case of $\mathcal{ALC}$
  - $\mathcal{F}$ contains a root node a_i for each individual a_i occurring in $\mathcal{A}$
  - $\mathcal{F}$ contains an edge $\langle a, b\rangle$ for each $\langle (a, b){:}R, n\rangle \in \mathcal{A}$
  - for each $\langle a{:}C, n\rangle \in \mathcal{A}$, we add both C to $\mathcal{L}(a)$ and $x_{a:C} \geq n$ to $\mathcal{C}_\mathcal{F}$
  - for each $\langle (a, b){:}R, n\rangle \in \mathcal{A}$, we add both R to $\mathcal{L}(\langle a, b\rangle)$ and $x_{(a,b):R} \geq n$ to $\mathcal{C}_\mathcal{F}$
- The notion of blocking is as for crisp $\mathcal{ALC}$
- $\mathcal{F}$ is then expanded by repeatedly applying the rules described below
- The completion-forest is complete when none of the rules are applicable
- Then, the bMILP problem on $\mathcal{C}_\mathcal{F}$ is solved

OR-based Fuzzy $\mathcal{ALC}$ Tableau rules with GCI's (Zadeh semantics)

Rule		Description
(var)		For variable $x_{v:C}$ add $x_{v:C} \in [0,1]$ to $\mathcal{C}_{\mathcal{F}}$. For variable $x_{(v,w):R}$, add $x_{(v,w):R} \in [0,1]$ to $\mathcal{C}_{\mathcal{F}}$
$(\bar{A})$	if	$\neg A \in \mathcal{L}(v)$ then add $x_{v:A} = 1 - x_{v:\neg A}$ to $\mathcal{C}_{\mathcal{F}}$
$(\bot)$	If	$\bot \in \mathcal{L}(v)$ then add $x_{v:\bot} = 0$ to $\mathcal{C}_{\mathcal{F}}$
$(\top)$	If	$\top \in \mathcal{L}(v)$ then add $x_{v:\top} = 1$ to $\mathcal{C}_{\mathcal{F}}$
$(\sqcap)$	if	$C_1 \sqcap C_2 \in \mathcal{L}(v)$, v is not indirectly blocked
	then	$\mathcal{L}(v) \to \mathcal{L}(v) \cup \{C_1, C_2\}$, and add $x_{v:C_1} \otimes x_{v:C_2} \geq x_{v:C_1 \sqcap C_2}$ to $\mathcal{C}_{\mathcal{F}}$
$(\sqcup)$	if	$C_1 \sqcup C_2 \in \mathcal{L}(v)$, v is not indirectly blocked
	then	$\mathcal{L}(v) \to \mathcal{L}(v) \cup \{C_1, C_2\}$, and add $x_{v:C_1} \oplus x_{v:C_2} \geq x_{v:C_1 \sqcup C_2}$ to $\mathcal{C}_{\mathcal{F}}$
$(\forall)$	if	$\forall R.C \in \mathcal{L}(v)$, v is not indirectly blocked
	then	$\mathcal{L}(w) \to \mathcal{L}(w) \cup \{C\}$, and add $x_{w:C} \geq x_{v:\forall R.C} \otimes x_{(v,w):R}$ to $\mathcal{C}_{\mathcal{F}}$
$(\exists)$	if	$\exists R.C \in \mathcal{L}(v)$, v is not blocked
	then	create new node w with $\mathcal{L}(\langle v, w\rangle) = \{R\}$ and $\mathcal{L}(w) = \{C\}$, and add $x_{w:C} \otimes x_{(v,w):R} \geq x_{v:\exists R.C}$ to $\mathcal{C}_{\mathcal{F}}$
$(\sqsubseteq)$	if	$\langle C \sqsubseteq D, n\rangle \in \mathcal{T}$, v is not indirectly blocked
	then	$\mathcal{L}(v) \to \mathcal{L}(v) \cup \{C, D\}$, and add $x_{v:D} \geq x_{v:C} \otimes n$ to $\mathcal{C}_{\mathcal{F}}$

Analytical Fuzzy Tableaux: $\mathcal{ALC}$ under SFL over $[0, 1]$

- Works as for classical $\mathcal{ALC}$ on completion forests
  - Node labels $\mathcal{L}(v)$ contain, rather than DL concept expressions, expressions of the form $\langle C, n \rangle$

    "The truth degree of being v instance of C is $\geq n$"
  - Blocking is as for classical $\mathcal{ALC}$
  - The completion forest is expanded by repeatedly applying inference rules
  - The completion-forest is complete when none of the rules are applicable
- Additionally, we adapt the notion of clash: a clash is either
  - $\langle \bot, n \rangle$ with $n > 0$; or
  - a pair $\langle C, n \rangle$ and $\langle \neg C, m \rangle$ with $n > 1 - m$
- Eventually, the initial KB is satisfiable if there is a clash-free complete completion forest

($\sqcap$). If *(i)* $\langle C_1 \sqcap C_2, n\rangle \in \mathcal{L}(v)$, *(ii)* $\{\langle C_1, n\rangle, \langle C_2, n\rangle\} \not\subseteq \mathcal{L}(v)$, and *(iii)* node v is not indirectly blocked, then add $\langle C_1, n\rangle$ and $\langle C_2, n\rangle$ to $\mathcal{L}(v)$.

($\sqcup$). If *(i)* $\langle C_1 \sqcup C_2, n\rangle \in \mathcal{L}(v)$, *(ii)* $\{\langle C_1, n\rangle, \langle C_2, n\rangle\} \cap \mathcal{L}(v) = \emptyset$, and *(iii)* node v is not indirectly blocked, then add some $\langle C, n\rangle \in \{\langle C_1, n\rangle, \langle C_2, n\rangle\}$ to $\mathcal{L}(v)$.

($\forall$). If *(i)* $\langle \forall R.C, n\rangle \in \mathcal{L}(v)$, *(ii)* $\langle R, m\rangle \in \mathcal{L}(\langle v, w\rangle)$ with $m > 1 - n$, *(iii)* $\langle C, n\rangle \notin \mathcal{L}(w)$, and *(iv)* node v is not indirectly blocked, then add $\langle C, n\rangle$ to $\mathcal{L}(w)$.

($\exists$). If *(i)* $\langle \exists R.C, n\rangle \in \mathcal{L}(v)$, *(ii)* there is no $\langle R, n_1\rangle \in \mathcal{L}(\langle v, w\rangle)$ with $\langle C, n_2\rangle \in \mathcal{L}(w)$ such that $\min(n_1, n_2) \geq n$, and *(iii)* node v is not blocked, then create a new node w, add $\langle R, n\rangle$ to $\mathcal{L}(\langle v, w\rangle)$ and add $\langle C, n\rangle$ to $\mathcal{L}(w)$.

($\sqsubseteq$). If *(i)* $\langle \top \sqsubseteq D, n\rangle \in \mathcal{T}$, *(ii)* $\langle D, n\rangle \notin \mathcal{L}(v)$, and *(iii)* node v is not indirectly blocked, then add $\langle D, n\rangle$ to $\mathcal{L}(v)$.

Non-Deterministic Analytic Fuzzy Tableaux

- It's a combination of the analogous method for fuzzy propositional logic and analytical fuzzy tableau
- Works for finitely-valued fuzzy propositional logic over L_n
- Works also for SFL (as in place of $[0, 1]$, we may use $\bar{N}^{\mathcal{K}}$)
- Rule examples:

($\sqcap$). If *(i)* $\langle C_1 \sqcap C_2, m\rangle \in \mathcal{L}(v)$, *(ii)* there are $m_1, m_2 \in L_n$ such that $m_1 \otimes m_2 = m$ with $\{\langle C_1, m_1\rangle, \langle C_2, m_2\rangle\} \not\subseteq \mathcal{L}(v)$, and *(iii)* node v is not indirectly blocked, then add $\langle C_1, m_1\rangle$ and $\langle C_2, m_2\rangle$ to $\mathcal{L}(v)$

($\sqcup$). If *(i)* $\langle C_1 \sqcup C_2, m\rangle \in \mathcal{L}(v)$, *(ii)* there are $m_1, m_2 \in L_n$ such that $m_1 \oplus m_2 = m$ with $\{\langle C_1, m_1\rangle, \langle C_2, m_2\rangle\} \cap \mathcal{L}(v) = \emptyset$, and *(iii)* node v is not indirectly blocked, then add some $\langle C, k\rangle \in \{\langle C_1, m_1\rangle, \langle C_2, m_2\rangle\}$ to $\mathcal{L}(v)$.

($\neg$). If *(i)* $\langle \neg C, m\rangle \in \mathcal{L}(v)$ with $\langle C, \ominus m\rangle \notin \mathcal{L}(v)$ and *(ii)* node v is not indirectly blocked, then add $\langle C, \ominus m\rangle$ to $\mathcal{L}(v)$.

($\forall$). If *(i)* $\langle \forall R.C, m\rangle \in \mathcal{L}(v)$, *(ii)* $\langle R, m_1\rangle \in \mathcal{L}(\langle v, w\rangle)$, *(iii)* there is $m_2 \in L_n$ such that $m_1 \to m_2 \geq m$ with $\langle C, m_2\rangle \notin \mathcal{L}(w)$, and *(iv)* node v is not indirectly blocked, then add $\langle C, m_2\rangle$ to $\mathcal{L}(w)$.

($\exists$). If *(i)* $\langle \exists R.C, m\rangle \in \mathcal{L}(v)$, *(ii)* there are $m_1, m_2 \in L_n$ such that $m_1 \otimes m_2 = m$, *(iii)* there is no $\langle R, m_1\rangle \in \mathcal{L}(\langle v, w\rangle)$ with $\langle C, m_2\rangle \in \mathcal{L}(w)$, and *(iv)* node v is not blocked, then create a new node w, add $\langle R, m_1\rangle$ to $\mathcal{L}(\langle v, w\rangle)$ and add $\langle C, m_2\rangle$ to $\mathcal{L}(w)$.

($\sqsubseteq$). If *(i)* $\langle C \sqsubseteq D, m\rangle \in \mathcal{T}$, *(ii)* there are $m_1, m_2 \in L_n$ such that $m_1 \to m_2 \geq m$, *(iii)* $\{\langle C, m_1\rangle, \langle D, m_2\rangle\} \not\subseteq \mathcal{L}(v)$, and *(iv)* node v is not indirectly blocked, then add $\langle C, m_1\rangle$ and $\langle D, m_2\rangle$ to $\mathcal{L}(v)$.

Reduction to Classical DLs

- Same principle as for the reduction for propositional fuzzy logic
- Needs adaption to the DL constructs: e.g. $\exists, \forall$ and $\sqsubseteq$
- Examples of reduction rules for SFL:

$$
\begin{array}{rcl}
\rho(A, \geq \gamma) & = & A_{\geq \gamma} \\
\rho(C \sqcap D, \geq \gamma) & = & \rho(C, \geq \gamma) \sqcap \rho(D, \geq \gamma) \\
\rho(C \sqcap D, \leq \gamma) & = & \rho(C, \leq \gamma) \sqcup \rho(D, \leq \gamma) \\
\rho(\forall R.C, \geq \gamma) & = & \forall \rho(R, > 1 - \gamma).\rho(C, \geq \gamma) \\
\rho(\forall R.C, \leq \gamma) & = & \exists \rho(R, \geq 1 - \gamma).\rho(C, \leq \gamma) \\
\rho(\exists R.C, \geq \gamma) & = & \exists \rho(R, \geq \gamma).\rho(C, \geq \gamma) \\
\rho(\exists R.C, \leq \gamma) & = & \forall \rho(R, > \gamma).\rho(C, \leq \gamma) \\
\rho(R, \geq \gamma) & = & R_{\geq \gamma} \\
\rho(\langle a{:}C, \gamma \rangle) & = & \{a{:}\rho(C, \geq \gamma)\} \\
\rho(\langle C \sqsubseteq D, n \rangle) & = & \bigcup_{\alpha \in \bar{N}_{+}^{\mathcal{K}}, \alpha \leq n} \{\rho(C, \geq \alpha) \sqsubseteq \rho(D, \geq \alpha)\}
\end{array}
$$

Computational Complexity

The bad news...undecidability!

Proposition

Assume that fuzzy GCIs are restricted to be classical, i.e. of the form $\langle \alpha, 1 \rangle$ only. Then for the following fuzzy DLs, the KB satisfiability problem is undecidable over $[0, 1]$:

1. *$\mathcal{ELC}$ with classical axioms only under Łukasiewicz logic and product logic;*
2. *$\mathcal{ELC}$ under any non Gödelt-norm $\otimes$;*
3. *$\mathcal{ELC}$ with concept assertions of the form $\langle \alpha = n \rangle$ only under any non Gödelt-norm $\otimes$;*
4. *$\mathcal{AL}$ with concept implication operator $\rightarrow$ and concept assertions of the form $\langle \alpha = n \rangle$ only under any non Gödelt-norm $\otimes$.*
5. *$\mathcal{ELC}$ under SFL with weighted sum constructor.*

Some decidability results..

Proposition

The KB satisfiability problem is decidable for

- $\mathcal{SROIQ}$ *under SFL over* $[0, 1]$ *and Gödel logic over* L_n
- $\mathcal{SROIN}$ *under Łukasiewicz logic over* L_n
- $\mathcal{SHI}$ *under any continuous t-norm over* L_n *without TBox*
- $\mathcal{ALC}$ *with concept implication operator* $\rightarrow$*, for any continuous t-norm over* $[0, 1]$ *with acyclicTBox*
- $\mathcal{SHIF}$ *with concept implication operator* $\rightarrow$*, for Łukasiewicz logic over* $[0, 1]$ *with acyclicTBox*
- $\mathcal{SI}$ *under any continuous t-norm over* $[0, 1]$ *without TBox*

Fuzzy DLs Query Answering

- Conjunctive query: similar to fuzzy RDFS CQs:

$$\begin{array}{rcl} \langle q(\mathbf{x}), s\rangle & \leftarrow & \exists \mathbf{y}.\langle \tau_1, s_1\rangle, \ldots, \langle \tau_n, s_n\rangle, \\ & & s = f(s_1, \ldots, s_n, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h)) \end{array}$$

  where

  - $\tau_1, \ldots, \tau_n$ are expressions $A(z)$ or $R(z, z')$, where A is a concept name, R is a role name, z, z' are individuals or variables in $\mathbf{x}$ or $\mathbf{y}$

- Example:

  $\langle q(x), s\rangle \leftarrow \langle \text{SportCar}(x), s_1\rangle, \text{hasPrice}(x, y), s = s_1 \cdot \text{cheap}(y)$

  where e.g. $\text{cheap}(y) = \text{ls}(10000, 12000)(y)$, has intended meaning to retrieve all cheap sports car.

Top-*k* retrieval in tractable DLs: the case of DL-Lite/DLR-Lite

- DL-Lite/DLR-Lite: a simple, but interesting DLs
- Captures important subset of UML/ER diagrams
- Computationally tractable DL to query large databases
- Sub-linear, i.e. LOGSpace in data complexity
  - (same cost as for SQL)
- Good for very large database tables, with limited declarative schema design
- For a CQ query answering procedure, see [Straccia, 2013, Straccia, 2012]
- Can be obtained also by a reduction to fuzzy Datalog

Logic Programs

Probabilistic Logic Programs

- There exists quite many different probabilistic LPs
- We illustrate Probabilistic Datalog under ICL by example
- Logic programs P under different "choices" (Independent Choice Logic)
- Each choice along with P produces a first-order model.
- By placing a probability distribution over the different choices, one then obtains a distribution over the set of first-order models.
- ICL also generalizes Bayesian networks, influence diagrams, Markov decision processes, and normal form games.

Example

- The probability of rain is 0.2

$$\begin{aligned}\texttt{Rain}(x) &\leftarrow \texttt{h}_{\texttt{Rain}}(x)\\ C_{\texttt{Rain}} &= \{h_{\texttt{Rain}}(T), h_{\texttt{Rain}}(F)\}\\ pr(h_{\texttt{Rain}}(T)) &= 0.2\\ pr(h_{\texttt{Rain}}(F)) &= 0.8\end{aligned}$$

- The probability of sprinkler on is 0.4

$$\begin{aligned}\texttt{SprinklerOn}(x) &\leftarrow \texttt{h}_{\texttt{SprinklerOn}}(x)\\ C_{\texttt{SprinklerOn}} &= \{h_{\texttt{SprinklerOn}}(T), h_{\texttt{SprinklerOn}}(F)\}\\ pr(h_{\texttt{SprinklerOn}}(T)) &= 0.4\\ pr(h_{\texttt{SprinklerOn}}(F)) &= 0.6\end{aligned}$$

- If it is raining or the sprinkler is on then the grass is wet

$$\begin{aligned}\texttt{GrassWet}(x) &\leftarrow \texttt{Rain}(x)\\ \texttt{GrassWet}(x) &\leftarrow \texttt{SprinklerOn}(x)\end{aligned}$$

- What is the probability that the grass is wet?

Example (cont.)

- We have to sum up the probabilities of each total choice that added to the program make the query true

$$
\begin{aligned}
\texttt{Rain}(x) &\leftarrow \texttt{h}_{\texttt{Rain}}(x) \\
C_{\texttt{Rain}} &= \{h_{\texttt{Rain}}(T), h_{\texttt{Rain}}(F)\} \\
pr(h_{\texttt{Rain}}(T)) &= 0.2 \\
pr(h_{\texttt{Rain}}(F)) &= 0.8 \\
\\
\texttt{SprinklerOn}(x) &\leftarrow \texttt{h}_{\texttt{SprinklerOn}}(x) \\
C_{\texttt{SprinklerOn}} &= \{h_{\texttt{SprinklerOn}}(T), h_{\texttt{SprinklerOn}}(F)\} \\
pr(h_{\texttt{SprinklerOn}}(T)) &= 0.4 \\
pr(h_{\texttt{SprinklerOn}}(F)) &= 0.6 \\
\\
\texttt{GrassWet}(x) &\leftarrow \texttt{Rain}(x) \\
\texttt{GrassWet}(x) &\leftarrow \texttt{SprinklerOn}(x)
\end{aligned}
$$

Example (cont.)

- Total choice: select a ground atom from each choice

$$
\begin{array}{rcl}
\texttt{Rain}(x) & \leftarrow & \texttt{h}_{\texttt{Rain}}(x) \\
C_{\texttt{Rain}} & = & \{h_{\texttt{Rain}}(T), h_{\texttt{Rain}}(F)\} \\
\texttt{SprinklerOn}(x) & \leftarrow & \texttt{h}_{\texttt{SprinklerOn}}(x) \\
C_{\texttt{SprinklerOn}} & = & \{h_{\texttt{SprinklerOn}}(T), h_{\texttt{SprinklerOn}}(F)\} \\
\texttt{GrassWet}(x) & \leftarrow & \texttt{Rain}(x) \\
\texttt{GrassWet}(x) & \leftarrow & \texttt{SprinklerOn}(x)
\end{array}
$$

B	Total choice
B_1	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(T)$
B_2	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(F)$
B_3	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(T)$
B_4	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(F)$

Example (cont.)

- Total choice B making query true: $P \cup B \models \texttt{GrassWet}(T)$

$$
\begin{aligned}
\texttt{Rain}(x) &\leftarrow \texttt{h}_{\texttt{Rain}}(x) \\
C_{\texttt{Rain}} &= \{h_{\texttt{Rain}}(T), h_{\texttt{Rain}}(F)\} \\
\texttt{SprinklerOn}(x) &\leftarrow \texttt{h}_{\texttt{SprinklerOn}}(x) \\
C_{\texttt{SprinklerOn}} &= \{h_{\texttt{SprinklerOn}}(T), h_{\texttt{SprinklerOn}}(F)\} \\
\texttt{GrassWet}(x) &\leftarrow \texttt{Rain}(x) \\
\texttt{GrassWet}(x) &\leftarrow \texttt{SprinklerOn}(x)
\end{aligned}
$$

B	Total choice	$P \cup B \models \texttt{GrassWet}(T)$
B_1	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(T)$	•
B_2	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(F)$	•
B_3	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(T)$	•
B_4	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(F)$	

Example (cont.)

- Probability of total choice B: $pr(B) = \Pi_{a \in B} pr(a)$
- Condition on pr: $\sum_{a \in C} pr(a) = 1$

$$
\begin{array}{rcl}
\texttt{Rain}(x) & \leftarrow & \texttt{h}_{\texttt{Rain}}(x) \\
pr(h_{\texttt{Rain}}(T)) & = & 0.2 \\
pr(h_{\texttt{Rain}}(F)) & = & 0.8 \\
\\
\texttt{SprinklerOn}(x) & \leftarrow & \texttt{h}_{\texttt{SprinklerOn}}(x) \\
pr(h_{\texttt{SprinklerOn}}(T)) & = & 0.4 \\
pr(h_{\texttt{SprinklerOn}}(F)) & = & 0.6 \\
\\
\texttt{GrassWet}(x) & \leftarrow & \texttt{Rain}(x) \\
\texttt{GrassWet}(x) & \leftarrow & \texttt{SprinklerOn}(x)
\end{array}
$$

B	Total choice	$P \cup B \models \texttt{GrassWet}(T)$	$pr(B)$
B_1	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(T)$	•	0.08
B_2	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(F)$	•	0.12
B_3	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(T)$	•	0.32
B_4	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(F)$		0.48
			1.0

Example (cont.)

- Probability of q: $Pr(q) = \sum_{B, P \cup B \models q} pr(B)$

$$
\begin{aligned}
\texttt{Rain}(x) &\leftarrow \texttt{h}_{\texttt{Rain}}(x) \\
pr(h_{\texttt{Rain}}(T)) &= 0.2 \\
pr(h_{\texttt{Rain}}(F)) &= 0.8 \\
\\
\texttt{SprinklerOn}(x) &\leftarrow \texttt{h}_{\texttt{SprinklerOn}}(x) \\
pr(h_{\texttt{SprinklerOn}}(T)) &= 0.4 \\
pr(h_{\texttt{SprinklerOn}}(F)) &= 0.6 \\
\\
\texttt{GrassWet}(x) &\leftarrow \texttt{Rain}(x) \\
\texttt{GrassWet}(x) &\leftarrow \texttt{SprinklerOn}(x)
\end{aligned}
$$

B	Total choice	$P \cup B \models \texttt{GrassWet}(T)$	$pr(B)$	$Pr(\texttt{GrassWet}(T))$
B_1	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(T)$	•	0.08	+
B_2	$h_{\texttt{Rain}}(T), h_{\texttt{SprinklerOn}}(F)$	•	0.12	+
B_3	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(T)$	•	0.32	+
B_4	$h_{\texttt{Rain}}(F), h_{\texttt{SprinklerOn}}(F)$		0.48	
			1.0	0.52

Fuzzy Logic Programs

- We consider fuzzy Datalog, which extends classical Datalog, where
  - Truth space is $[0, 1]$ or $L_n = \{0, \frac{1}{n}, \ldots, \frac{n-2}{n-1}, \ldots, 1\}$ $(n > 2)$
  - Interpretation is a mapping $I : B_{\mathcal{P}} \to [0, 1]$
  - Generalized LP rules are of the form

$$R(\mathbf{x}) \leftarrow \exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_k(\mathbf{z}_k), p_1(\mathbf{z}'_1), \ldots, p_h(\mathbf{z}'_h))$$

    - Meaning of rules: "take the truth-values of all $R_i(\mathbf{z}_i), p_j(\mathbf{z}'_j)$, combine them using the truth combination function f, and assign the result to $R(\mathbf{x})$"
  - Facts: ground expressions of the form $\langle R(\mathbf{c}), n \rangle$
    - Meaning of facts: "the degree of truth of $R(\mathbf{c})$ is at least n"
  - Fuzzy LP: a set of facts (extensional database) and a set of rules (intentional database). No extensional relation may occur in the head of a rule

- Rules:

$$R(\mathbf{x}) \leftarrow \exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$$

1. $\mathbf{x}$ are the *distinguished variables*;
2. s is the *score variable*, taking values in $[0, 1]$;
3. $\mathbf{y}$ are existentially quantified variables, called *non-distinguished variables*;
4. $\varphi(\mathbf{x}, \mathbf{y})$ is $f(\mathbf{R}(\mathbf{z}), \mathbf{p}(\mathbf{z}'))$, where $\mathbf{R}$ is a vector of predicates R_i and $\mathbf{p}$ is a vector of fuzzy predicates p_j;
5. $\mathbf{z}, \mathbf{z}'$ are tuples of constants in *KB* or variables in $\mathbf{x}$ or $\mathbf{y}$;
6. p_j is an n_j-ary *fuzzy predicate* assigning to each n_j-ary tuple $\mathbf{c}_j$ the *score* $p_j(\mathbf{c}_j) \in [0, 1]$;
7. f is a monotone *scoring* function $f\colon [0, 1]^{k+h} \to [0, 1]$, which combines the scores of the h fuzzy predicates $p_j(\mathbf{c}_j)$ with the k scores $R_i(\mathbf{c}_i)$

Semantics of fuzzy Datalog

- Like for classical Datalog
- $\mathcal{P}^*$ is constructed as follows (as for the classical case):
  1. set $\mathcal{P}^*$ to the set of all ground instantiations of rules in $\mathcal{P}$;
  2. replace a fact $p(\mathbf{c})$ in $\mathcal{P}^*$ with the rule $p(\mathbf{c}) \leftarrow 1$
  3. if atom A is not head of any rule in $\mathcal{P}^*$, then add $A \leftarrow 0$ to $\mathcal{P}^*$;
  4. replace several rules in $\mathcal{P}^*$ having same head

$$\left.\begin{array}{c} A \leftarrow \varphi_1 \\ A \leftarrow \varphi_2 \\ \vdots \\ A \leftarrow \varphi_n \end{array}\right\} \text{with } A \leftarrow \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n \,.$$

- Note: in $\mathcal{P}^*$ each atom $A \in B_{\mathcal{P}}$ is head of exactly one rule
- Herbrand Base of $\mathcal{P}$ is the set $B_{\mathcal{P}}$ of ground atoms
- Interpretation is a function $I : B_{\mathcal{P}} \to [0, 1]$.
- Model $I \models \mathcal{P}$ iff for all $r \in \mathcal{P}^*$ $I \models r$, where $I \models A \leftarrow \varphi$ iff $I(\varphi) \leq I(A)$
- Note:

$$I(f(R_1(\mathbf{c}_1), \ldots, R_k(\mathbf{c}_k), p_1(\mathbf{c}'_1), \ldots, p_h(\mathbf{c}'_h))) = f(I(R_1(\mathbf{c}_1)), \ldots, I(R_k(\mathbf{c}_k)), p_1(\mathbf{c}'_1), \ldots, p_h(\mathbf{c}'_h)))$$

Fuzzy LP Query Answering

- Least model $M_{\mathcal{P}}$ of $\mathcal{P}$ exists and is least fixed-point of

$$T_{\mathcal{P}}(I)(A) = I(\varphi), \text{ for all } A \leftarrow \varphi \in \mathcal{P}^*$$

- M can be computed as the limit of

$$\begin{array}{lcl} \mathbf{I}_0 & = & \mathbf{0} \\ \mathbf{I}_{i+1} & = & T_{\mathcal{P}}(\mathbf{I}_i) \ . \end{array}$$

- Entailment: for a ground expression $\langle q(\mathbf{c}), s\rangle$, $s \in [0, 1]$

$$\mathcal{P} \models \langle q(\mathbf{c}), s\rangle \text{ iff least model of } \mathcal{P} \text{ satisfies } I(q(\mathbf{c})) \geq s$$

- We say that s is *tight* iff $s = \sup\{s' \mid \mathcal{P} \models \langle q(\mathbf{c}), s'\rangle\}$
- If $\mathcal{P} \models \langle q(\mathbf{c}), s\rangle$ and s is tight then $\langle \mathbf{c}, s\rangle$ is called an *answer* to q
- The answer set of q w.r.t. $\mathcal{P}$ is defined as

$$ans(\mathcal{P}, q) = \{\langle \mathbf{c}, s\rangle \mid \mathcal{P} \models \langle q(\mathbf{c}), s\rangle, \ s \text{ is tight}\}$$

Top-k Retrieval: Given a fuzzy LP $\mathcal{P}$, and a query q, retrieve k answers $\langle \mathbf{c}, s\rangle$ with maximal scores and rank them in decreasing order relative to the score s, denoted

$$ans_k(\mathcal{P}, q) = \text{Top}_k \ ans(\mathcal{P}, q) \ .$$

- Fuzzy LPs may be tricky:

$$\begin{array}{lcl} \langle p, 0.1\rangle & & \\ p & \leftarrow & (p+1)/2 \end{array}$$

- In the minimal model the truth of A is 1 (requires ω $T_{\mathcal{P}}$ iterations)!
- There are several ways to avoid this pathological behavior:
  - We may consider $L = \{0, \frac{1}{n}, \frac{2}{n} \ldots, \frac{n-1}{n}, 1\}$, n natural number, e.g. $n = 100$
  - In $A \leftarrow f(B_1, \ldots, B_n)$, f is bounded, i.e. $f(x_1, \ldots, x_n) \leq x_i$

Example: Soft shopping agent

- I may represent my preferences in Logic Programming with the rules

$$
\begin{aligned}
Pref_1(x,p) &\leftarrow HasPrice(x,p) \wedge LS(10000, 14000)(p) \\
Pref_2(x) &\leftarrow HasKM(x,k) \wedge LS(13000, 17000)(k) \\
Buy(x,p) &\leftarrow 0.7 \cdot Pref_1(x,p) + 0.3 \cdot Pref_2(x)
\end{aligned}
$$

ID	MODEL	PRICE	KM
455	MAZDA 3	12500	10000
34	ALFA 156	12000	15000
1812	FORD FOCUS	11000	16000
⋮	⋮	⋮	⋮

- Problem: All tuples of the database have a score:
  - We cannot compute the score of all tuples, then rank them. Brute force approach not feasible.
- Top-k problem: Determine efficiently just the top-k ranked tuples, without evaluating the score of all tuples. E.g. top-3 tuples

ID	PRICE	SCORE
1812	11000	0.6
455	12500	0.56
34	12000	0.50

General top-down query procedure for Many-valued LPs

- Idea: use theory of fixed-point computation of equational systems over truth space (complete lattice or complete partial order)
- Assign a variable x_i to an atom $A_i \in B_{\mathcal{P}}$
- Map a rule $A \leftarrow f(A_1, \ldots, A_n) \in \mathcal{P}^*$ into the equation $x_A = f(x_{A_1}, \ldots, x_{A_n})$
- A LP $\mathcal{P}$ is thus mapped into the equational system

$$\left\{ \begin{array}{lcl} x_1 & = & f_1(x_{1_1}, \ldots, x_{1_{a_1}}) \\ & \vdots & \\ x_n & = & f_n(x_{n_1}, \ldots, x_{n_{a_n}}) \end{array} \right.$$

- f_i is monotone and, thus, the system has least fixed-point, which is the limit of

$$\begin{array}{lcl} \mathbf{y}_0 & = & \mathbf{0} \\ \mathbf{y}_{i+1} & = & \mathbf{f}(\mathbf{y}_i) \ . \end{array}$$

  where $\mathbf{f} = \langle f_1, \ldots, f_n \rangle$ and $\mathbf{f}(\mathbf{x}) = \langle f_1(x_1), \ldots, f_n(x_n) \rangle$
- The least-fixed point is the least model of $\mathcal{P}$
- Consequence: If top-down procedure exists for equational systems then it works for fuzzy LPs too!

Procedure *Solve*$(\mathcal{S}, Q)$
Input: monotonic system $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, where $Q \subseteq V$ is the set of query variables;
Output: A set $B \subseteq V$, with $Q \subseteq B$ such that the mapping v equals $\mathrm{lfp}(f)$ on B.

$A := Q$, $dg := Q$, $in := \emptyset$, **for all** $x \in V$ **do** $v(x) = 0$, $exp(x) = 0$
while $A \neq \emptyset$ **do**
  select $x_i \in A$, $A := A \setminus \{x_i\}$, $dg := dg \cup s(x_i)$
  $r := f_i(v(x_{i_1}), ..., v(x_{i_{a_i}}))$
  if $r \succ v(x_i)$ **then** $v(x_i) := r$, $A := A \cup (p(x_i) \cap dg)$ **fi**
  if not $exp(x_i)$ **then** $exp(x_i) = 1$, $A := A \cup (s(x_i) \setminus in)$, $in := in \cup s(x_i)$ **fi**
od

For $q(\mathbf{x}) \leftarrow \phi \in \mathcal{P}$, with $\mathrm{s}(q)$ we denote the set of *sons* of q w.r.t. r, i.e. the set of intentional predicate symbols occurring in ϕ. With $\mathrm{p}(q)$ we denote the set of *parents* of q, i.e. the set $\mathrm{p}(q) = \{p_i : q \in \mathrm{s}(p_i, r)\}$ (the set of predicate symbols directly depending on q).

Top-k retrieval in LPs

- If the database contains a huge amount of facts, a brute force approach fails:
  - one cannot anymore compute the score of all tuples, rank all of them and only then return the top-k
- Better solutions exists for restricted fuzzy LP languages: Datalog + restriction on the score combination functions appearing in the body

Basic Idea

- We do not compute all answers, but determine answers incrementally
- At each step *i*, from the tuples seen so far in the database, we compute a threshold δ
- The threshold δ has the property that any successively retrieved answer will have a score $s \leq \delta$
- Therefore, we can stop as soon as we have gathered *k* answers above δ, because any successively computed answer will have a score below δ

Procedure *TopAnswers*($\mathcal{K}$, Q, k)
Input: KB $\mathcal{K}$, intensional query relation symbol Q, $k \geq 1$;
Output: Mapping `rankedList` such that `rankedList`(Q) contains top-k answers of Q
Init: $\delta = 1$, **for all** rules $r : P(\mathbf{x}) \leftarrow \phi$ in P **do**
    if P intensional **then** `rankedList`$(P) = \emptyset$;
    if P extensional **then** `rankedList`$(P) = T_P$ **endfor**

1. **loop**
2. `Active` $:= \{Q\}$, `dg` $:= \{Q\}$, `in` $:= \emptyset$,
    for all rules $r : P(\mathbf{x}) \leftarrow \phi$ **do** `exp`$(P, r) =$ `false`;
3. **while** (`Active` $\neq \emptyset$) **do**
4. **select** $P \in$ `A` where $r : P(\mathbf{x}) \leftarrow \phi$, `Active` := `Active` $\setminus \{P\}$, `dg` := `dg` $\cup$ `s`(P, r);
5. $\langle \mathbf{t}, s \rangle :=$ *getNextTuple*(P, r)
6. **if** $\langle \mathbf{t}, s \rangle \neq$ `NULL` **then** insert $\langle \mathbf{t}, s \rangle$ into `rankedList`(P),
    `Active` := `Active` $\cup$ (`p`$(P) \cap$ `dg`);
7. **if not** `exp`(P, r) **then** `exp`$(P, r) =$ `true`,
    `Active` := `Active` $\cup$ (`s`$(P, r) \setminus$ `in`), `in` := `in` $\cup$ `s`(p, r);

    endwhile

8. Update threshold δ;
9. **until** (`rankedList`(Q) does contain k top-ranked tuples with score above δ)
    or (`rL`$'$ = `rankedList`);
10. **return** top-k ranked tuples in `rankedList`(Q);

Procedure *getNextTuple*(P, r)
Input: intensional relation symbol P and rule $r : P(\mathbf{x}) \leftarrow \exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_n(\mathbf{z}_l)) \in P$;
Output: Next tuple satisfying the body of the r together with the score
Init:

loop

1. Generate next new instance tuple $\langle \mathbf{t}, s \rangle$ of P, using tuples in $\texttt{rankedList}(R_i)$ and (RankSQL or Postgres)
2. **if** there is no $\langle \mathbf{t}, s' \rangle \in \texttt{rankedList}(P, r)$ with $s \leq s'$ **then** exit loop

until no new valid join tuple can be generated

3. **return** $\langle \mathbf{t}, s \rangle$ if it exists **else return** `NULL`

Example

Logic Program $\mathcal{P}$ is

$$q(x) \leftarrow p(x)$$
$$p(x) \leftarrow \min(r_1(x, y), r_2(y, z))$$

RecordID	r_1			r_2		
1	a	b	1.0	m	h	0.95
2	c	d	0.9	m	j	0.85
3	e	f	0.8	f	k	0.75
4	l	m	0.7	m	n	0.65
5	o	p	0.6	p	q	0.55
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

What is

$$Top_1(\mathcal{P}, q) = Top_1\{\langle c, s\rangle \mid \mathcal{P} \models q(c, s)\} \ ?$$

$$q(x) \leftarrow p(x)$$
$$p(x) \leftarrow \min(r_1(x, y), r_2(y, z))$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: STOP, top-1 tuple score is equal or above threshold $0.75 = \max(\min(1.0, 0.75), \min(0.7, 0.95))$

Queue	δ
—	0.75

Predicate	*Answers*
q	$\langle e, 0.75\rangle \langle l, 0.7\rangle$
p	$\langle e, 0.75\rangle, \langle l, 0.7\rangle$

$$Top_1(\mathcal{P}, q) = \{\langle e, 0.75\rangle\}$$

Note: no further answer will have score above threshold δ

Threshold computation

For an intentional predicate p, head of a rule $r : p(\mathbf{x}) \leftarrow f(p_1, p_2, \ldots, p_n)$.

- consider a threshold variable δ^p
- with $r.\mathbf{t}_{p_i}^{\perp}$ ($r.\mathbf{t}_{p_i}^{\top}$) we denote the last tuple seen (the top ranked one) in $\texttt{rankedList}(p, r)$
- we define

$$
\begin{aligned}
p_i^{\top} &= \max(\delta^{p_i}, r.\mathbf{t}_{p_i}^{\top}.score) \\
p_i^{\perp} &= \delta^{p_i}
\end{aligned}
$$

- if p_i is an extensional predicate, we define

$$
\begin{aligned}
p_i^{\top} &= r.\mathbf{t}_{p_i}^{\top}.score \\
p_i^{\perp} &= r.\mathbf{t}_{p_i}^{\perp}.score
\end{aligned}
$$

- for rule r we consider the equation $\delta(r)$

$$
\delta^p = \max(f(p_1^{\perp}, p_2^{\top}, \ldots, p_n^{\top}), f(p_1^{\top}, p_2^{\perp}, \ldots, p_n^{\top}), \ldots, f(p^{\top}, p^{\top}, \ldots, p_n^{\perp}))
$$

- consider the set of equations of all equations involving intentional predicates, i.e.

$$
\Delta = \bigcup_{r \in P} \{\delta(r)\} .
$$

- for a query $q(\mathbf{x})$, the threshold δ of the *TopAnswers* algorithm is defined as to be

$$
\delta = \bar{\delta}^q ,
$$

  where $\bar{\delta}^q$ is the solution to δ^q in the minimal solution $\bar{\Delta}$ of the set of equations Δ.
- note that $\bar{\delta}^q$, can be computed iteratively as least fixed-point

Bibliography

Artale, A., Calvanese, D., Kontchakov, R., and Zakharyaschev, M. (2009).
The DL-Lite family and relations.
Journal of Artificial Intelligence Research, 36:1–69.

Baader, F., Peñaloza, R., and Suntisrivaraporn, B. (2007).
Pinpointing in the description logic $\mathcal{EL}^+$.
In *Proceedings of the 30th Annual German Conference on Advances in Artificial Intelligence (KI-07)*, number 4667 in Lecture Notes in Computer Science, pages 52–67, Berlin, Heidelberg. Springer-Verlag.

Grosof, B. N., Horrocks, I., Volz, R., and Decker, S. (2003).
Description logic programs: combining logic programs with description logic.
In *Proceedings of the 12th International Conference on World Wide Web*, pages 48–57. ACM Press.

Klir, G. J. and Yuan, B. (1995).
Fuzzy sets and fuzzy logic: theory and applications.
Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Lloyd, J. W. (1987).
Foundations of Logic Programming.
Springer, Heidelberg, RG.

Lukasiewicz, T. and Straccia, U. (2008).
Managing uncertainty and vagueness in description logics for the semantic web.
Journal of Web Semantics, 6:291–308.

Straccia, U. (2012).
Top-k retrieval for ontology mediated access to relational databases.
Information Sciences, 198:1–23.

Straccia, U. (2013).
Foundations of Fuzzy Logic and Semantic Web Languages.
CRC Studies in Informatics Series. Chapman & Hall.

Straccia, U. and Casini, G. (2022).

A Minimal Deductive System for RDFS with Negative Statements.
In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, pages 351–361.

ter Horst, H. J. (2005).
Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary.
Journal of Web Semantics, 3(2-3):79–115.

Non-Classical Knowledge Representation and Reasoning

Italian National PhD Course on AI, 2024

Umberto Straccia & Giovanni Casini

CNR - ISTI, Pisa, Italy
http://www.straccia.info
{umberto.straccia, giovanni.casini}@isti.cnr.it

Outline

- **Lecture 4**:
  - Nonmonotonic reasoning
  - Conditional reasoning - KLM framework (propositional logic)
- **Lecture 5**:
  - Conditional reasoning - KLM framework (Description logics e RDFS)
  - Belief Change - AGM framework (propositional logic)
- **Lecture 6**:
  - Belief Change - AGM framework (other languages)
  - Paraconsistent logics (brief introduction)

Nonmonotonic and Conditional Reasoning

Monotonicity

- A **logic** is primarily defined by
  - a **language**; and
  - a **consequence relation** (o **entailment relation**).
- **Language**: propositional, first order, modal...
- **Consequence Relation**: A relation that determines what **follows** from any set of premises. Generally defined rigorously on some formal structures (**semantics**).

Monotonicity

Given a language $\mathcal{L}$:

- A **Consequence Relation** $\models : \wp(\mathcal{L}) \times \mathcal{L}$ is a relation between (finite) sets of formulas and a formulas (e.g. $\{a, a \rightarrow b \models b\}$).
- A **Consequence Operation** $\mathcal{C} : \wp(\mathcal{L}) \times \wp(\mathcal{L})$ is a function that associate to any set of formulas KB another set of formulas $\mathcal{C}(KB)$ s.t.:

$$\mathcal{C}(KB) = \{\alpha \mid KB \models \alpha\}$$

Monotonicity

From Lecture 1:

Satisfiability of KBs

- A set *KB* of formulae is satisfied iff $\mathcal{I} \models \alpha$ for all $\alpha \in KB$
- An interpretation $\mathcal{I}$ is a model of set *KB* of formulae (denoted $\mathcal{I} \models KB$) iff $\mathcal{I} \models \alpha$ for all $\alpha \in KB$
- A set *KB* of formulae is
  - satisfiable, if there is some $\mathcal{I}$ that satisfies *KB*
  - unsatisfiable, if *KB* is not satisfiable
- A set *KB* of formulae entails a formula α iff α is true in all models of *KB*, i.e.

$$KB \models \alpha \quad \text{iff} \quad \mathcal{I} \models \alpha \text{ for all models of } KB$$

Monotonicity

Classical logics are characterised by consequence relations that are **Tarskian**.

Definition (Tarskian Consequence Relation)

A consequence relation $\models : \wp(\mathcal{L}) \times \mathcal{L}$ is *tarskian* if it satisfies the following properties:

- **Reflexivity**: $A \models \alpha$ for every $\alpha \in A$.
- **Cut**: $\dfrac{A \cup \{\alpha\} \models \beta \qquad A \models \alpha}{A \models \beta}$
- **Monotonicity**: $\dfrac{A \models \beta}{A \cup \{\alpha\} \models \beta}$

for any set of formulas A and any formulas α, β.

Such properties are mirrored in the classical material implication '$\rightarrow$', due to the deduction theorem (see lecture 1):

$$KB \cup \{\alpha\} \models \beta \text{ iff } KB \models \alpha \rightarrow \beta$$

Monotonicity

The same properties can be formulated for consequence operations

Definition (Tarskian Consequence Operation)

A consequence operation $\mathcal{C} : \wp(\mathcal{L}) \times \wp(\mathcal{L})$ is *tarskian* if it satisfies the following properties:

- **Reflexivity**: $A \subseteq \mathcal{C}(A)$.
- **Cut**: If $A \subseteq B \subseteq \mathcal{C}(A)$, then $\mathcal{C}(B) \subseteq \mathcal{C}(A)$
- **Monotonicity**: $\mathcal{C}(B) \subseteq \mathcal{C}(A)$ whenever $B \subseteq A$

- Monotonicity tells us that augmenting the information in the premises, whatever we had concluded before remains true.
- It represents the necessity of the truth consequence given the truth of the premises.
- It is appropriate for modelling mathematical reasoning, bot not necessarily for other domains.

Monotonicity

"The concept of following logically belongs to the category of those concepts whose introduction into the domain of exact formal investigations was not only an act of arbitrary decision on the side of this or that researcher: in making precise the content of this concept, efforts were made to conform to the everyday 'pre-existing' way it is used. [. . .] the way it is used is unstable, the task of capturing and reconciling all the murky, sometimes contradictory intuitions connected with that concept has to be acknowledged a priori as unrealizable, and one has to reconcile oneself in advance to the fact that every precise definition of the concept under consideration will to a greater or lesser degree bear the mark of arbitrariness." [Tarski, 2002, p.176]

Nonmonotonicity

While monotonic consequence relations are appropriate for reasoning with certain and complete information, there are domains in which we need to draw conclusions while facing incomplete information.

The need to model logical systems appropriate for such domains has become apparent quite early in the program of Artificial Intelligence.

Nonmonotonicity - Frame Problem

- **Frame Problem [McCarthy and Hayes, 1969]**
  It is a main problem in modelling actions. It deals with modelling what remains unchanged after an event in a dynamic world:

  - *Inertia Assumption*: by default, everything is presumed to remain in the state in which it is.

  - α holds. An event *e* happens. If it is not contradictory to assume that *e* does not affect α, we assume that α still holds.

Nonmonotonicity - Frame Problem

- **Frame Problem - Nonmonotonicity**

  Scenario: a robot moves with its arm a small sphere. Our rules tell us that such an action will change the sphere's position. We assume that it will not affect other properties, like the sphere's colour and shape.

  However, if the sphere is made of soft material, we could later discover that the shape of the sphere is changed.

Nonmonotonicity - Closed-World Assumption

- **Closed-World Assumption** (**CWA**) [Reiter, 1978]
  In some contexts we assume that the information we have is complete: if we cannot conclude that α holds, then we assume that α does not hold.

  - Example: Train timetable. We assume that all the trains departing from or arriving at a certain station are only the trains listed in the station's timetable.

Nonmonotonicity - Interested domains

- Some reasoning domains need nonmonotonicity:
  - **Presumptive reasoning**
    - You know that Tweety is a bird, and you conclude that presumably Tweety flies. later you discover that Tweety is a penguin, and consequently does not fly.
  - **Counterfactual reasoning**
    - If Nazis had won WW-II, we would all be under a Nazi regime. But if Nazis had won WW-II and in the 70's there would have been a WW-III won by San Marino, we would not all be under a Nazi regime.
  - **Causal Reasoning**
    - A big hearthquake would cause the collapse of this building. But if we renovate this building, a big hearthquake would not cause its collapse.
  - **Normative Reasoning**
    - You should not kill. But if someone threatens your life, you are allowed to kill.

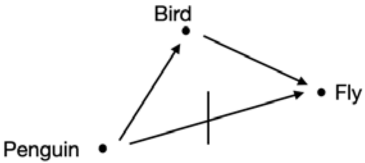
Nonmonotonicity - Default Reasoning

- **Defaults.**

  In general, we refer to the notion of a **default**: a piece of information that formalises some implicit background information that we assume to hold, until we are forced to conclude that that is not the case.

- Many formalisms. Some examples:
  - Inheritance nets
  - Reiter's Defaults
  - McCarthy's Circumscription
  - Negation as Failure
  - Defeasible Conditionals

Inheritance Nets



We have

- nodes (individuals or classes);
- positive links (*defeasible* subclass relations);
- negative links (*defeasible* disjointness relations).

Positive links can be treated as transitive, if no conflict with negative links arises. In case, different decision strategies can be applied to solve such conflicts.

Reiter's Default Rules

Rules of the form

$$\frac{\alpha \ : \ \beta_1, \ldots, \beta_n}{\gamma}.$$

If α (the *prerequisite*) is satisfied, and $\beta_1, \ldots, \beta_n$ (the *justifications*) are all consistent with our KB, then we can conclude γ (the *consequent*).
For example:

$$\frac{\mathit{Bird} \ : \ \mathit{Fly}}{\mathit{Fly}} \qquad \frac{\mathit{Bird} \ : \ \neg \mathit{Penguin}, \neg \mathit{Ostrich}, \neg \mathit{BrokenWing},}{\mathit{Fly}}$$

Given a system of default rules, there can be conflicts among different rules. Different ways or resolving such conflicts defines different ways of reasoning.

McCarthy's Circumscription

Given a KB, we consider only the models that minimise the extension of some propositions (or some predicate). In particular, the extension of the predicate *being abnormal* is minimised.

For example, consider a KB in which we have

- $Bird(x) \wedge \neg Abnorm(x) \rightarrow Fly(x)$;
- $Penguin(x) \rightarrow Bird(x) \wedge Abnorm(x)$;
- $Ostrich(x) \rightarrow Bird(x) \wedge Abnorm(x)$;
- $Eagle(x) \rightarrow Bird(x)$.

We consider only the models of the KB in which the extension of the predicate $Abnorm(x)$ is minimal. That is, it is applied only to the individuals for which it is neccessary (here, penguins and ostriches), while we considers the others as typical subcalsses (here, eagles are treated as typical birds).

Negation as Failure (NAF)

This approach is the most popular for implementing the CWA.

One of the most popular frameworks is **Logic Programming** (see lecture 2), where we reason by using rules like:

$$Fly \leftarrow Bird \wedge \sim BrokenWing$$

where '$\sim$ *BrokenWing*' must be interpreted as 'it cannot be proved *BrokenWing*'.
In this way we can implement CWA.

It is nonmonotonic. For example, from the fact

$$Bird \leftarrow$$

we can conclude *Fly*, but adding also the fact

$$BrokenWing \leftarrow$$

we are not able to activate the first rule anymore.

Negation as Failure (NAF)

Also, there may be conflicting rules. For example:

- $p \leftarrow \sim q$;
- $q \leftarrow \sim p$.

Different formal ways of managing these kind of conflicts defines different kind of nonmonotonic consequence relations.

Deafeasible Conditionals

Here the information is modelled using monotonic conditionals

$$\alpha \rightarrow \beta$$

and defeasible conditionals

$$\alpha \mid\!\sim \beta$$

It is a popular approach to model *if-then* reasoning in particular domains (e.g., presumptive, counterfactuals, causal, and normative reasoning).

Depending on the domain, $\alpha \mid\!\sim \beta$ could be read as 'If α, then presumably β', 'If α were the case, then it would have been β', 'α causes β', or 'If α, then β is mandatory'...

There are strong connections among the different formalisms for nonmonotonic reasoning, that in part have already been investigated.

The basic idea is kind of always the same: we conclude something relying on what we consider the standard situation, if we are not forced to conclude that we are in an exceptional one.

Makinson's book [Makinson, 2005] is a good starting point for gaining a general view and an idea of the basic connections between the formalisms in this area.

Deafeasible Conditionals

Today we focus on the conditional approach, in particular on the framework by Kraus, Lehmann and Magidor (**KLM**) [Kraus et al., 1990].

- Pros:
  - A stronger formal analysis of the kind of reasoning it models
  - A certain resemblance to the way we think
  - It is often possible to implement it on top of classical reasoners, sometimes with computational costs in the same category as the correspondent classical reasoning
- Cons:
  - It may be hard to apply it to logics that are more expressive than PL
  - It may be hard to apply it to logics that are computationally light, without sensibly augmenting the computational costs

KLM framework

KLM Framework

- With **KLM approach** we refer to the semantic approach to conditional reasoning formalised by Kraus, Lehmann and Magidor [Kraus et al., 1990].
- It is a step-stone for conditional reasoning, since they give a complete formal characterisation of an ample class of conditionals.

KLM Framework

- It has been developed for modelling **presumptive reasoning**:

  If it is a bird, then presumably it should be able to fly.

  But the same formal framework is appropriate for modelling also other kinds of reasoning. E.g., normative or counterfactual reasoning.

- The consequence does not follow necessarily from the premises, but only with plausibility.

KLM Framework

Given a propositional language, with formulas $\alpha, \beta, \gamma, \ldots$,

- The conditional $\alpha \mid\sim \beta$ is read "If α holds, then **typically** β holds".
- A knowledge base (KB) consists of a (finite) sets of conditionals

$$KB = \{\alpha_1 \mid\sim \beta_1, \ldots, \alpha_n \mid\sim \beta_n\}$$

- Reasoning with a conditional base: we define an entailment relation $\mid\approx$ that allows to derive new conditionals from a KB. E.g.,

$$\{\alpha_1 \mid\sim \beta_1, \alpha_1 \mid\sim \beta_2\} \mid\approx \alpha_1 \mid\sim (\beta_1 \land \beta_2)$$

KLM Framework

Before considering reasoning with conditionals, let's characterise some *reasoning patterns*, or *closure properties*.

A is a **preferential** set of conditionals if it closed under the following properties:

- Reflexivity (Ref): $\alpha \mid\!\sim \alpha$
- Right Weakening (RW): $\dfrac{\alpha \mid\!\sim \beta,\ \models \beta \rightarrow \gamma}{\alpha \mid\!\sim \gamma}$
- Left Logical equivalence (LLE): $\dfrac{\models \alpha \leftrightarrow \beta,\ \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$
- Right Conjunction (And): $\dfrac{\alpha \mid\!\sim \beta, \alpha \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \wedge \gamma}$
- Disjunction in the Premises (Or): $\dfrac{\alpha \mid\!\sim \gamma, \beta \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma}$
- Cautious Monotonicity (CM): $\dfrac{\alpha \mid\!\sim \beta,\ \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$

KLM Framework

The most interesting property is **Cautious Monotonicity**

- Cautious Monotonicity (CM): $\frac{\alpha \mid\sim \beta, \ \alpha \mid\sim \gamma}{\alpha \wedge \beta \mid\sim \gamma}$

If birds typically fly (*bird* $\mid\sim$ *fly*) and birds typically have feathers (*bird* $\mid\sim$ *feather*), we can conclude that birds with feathers typically fly (*bird* $\wedge$ *feather* $\mid\sim$ *fly*).

It is a very constrained form of the classical Monotonicity:

- Monotonicity (Mon): $\frac{\alpha \mid\sim \gamma}{\alpha \wedge \beta \mid\sim \gamma}$

KLM Framework

If a set of conditionals *A* is closed under all the preferential properties, it is easy to prove that it is also closed under other relevant properties.

For example:

- Cut (CT): $\dfrac{\alpha \mid\sim \beta,\ \alpha \wedge \beta \mid\sim \gamma}{\alpha \mid\sim \gamma}$
- Modus Ponens (MP): $\dfrac{\alpha \mid\sim \beta,\ \alpha \mid\sim \beta \rightarrow \gamma}{\alpha \mid\sim \gamma}$
- Supraclassicality (Sup): $\dfrac{\models \alpha \rightarrow \beta}{\alpha \mid\sim \beta}$

(Sup) tells us that every preferential consequence *extends* classical reasoning.

KLM Framework - Semantics

- Various ways to semantically characterise preferential sets of conditionals.
- **Preferential Interpretations**: most popular semantics. Possible-worlds semantics in the style of modal logics.
- Main idea:

  We interpret "If α, then typically β" as "In all the most typical situations in which α is true, also β is true".

  VS

  Classical case (Tarskian): "If α, then β" holds if "In all the situation in which α is true, also β is true"

KLM Framework - Semantics

Formalisation of the intuition:

we order the classical propositional interpretations (= formalisation of possible situations) according to their relative typicality.

Given two propositional interpretations $\mathcal{I}, \mathcal{J}$,

$$\mathcal{I} \prec \mathcal{J}$$

is read as

$\mathcal{I}$ is more typical than $\mathcal{J}$

KLM Framework - Semantics

Definition (Preferential interpretation - simplified version!)

Given a propositional language $\mathcal{L}$, let $\mathcal{W}$ be the set of all the interpretations defined over $\mathcal{L}$.

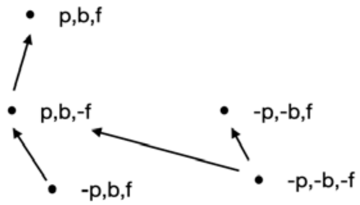
$$\mathcal{P} = \langle \mathcal{M}, \prec_{\mathcal{P}} \rangle$$

- $\mathcal{M} \subseteq \mathcal{W}$ is a set of interpretations;
- $\prec_{\mathcal{P}}$: $\mathcal{M} \times \mathcal{M}$ is a preference partial order over the propositional interpretations.

KLM Framework - Semantics

Example

Preferential model $\mathcal{P}$:

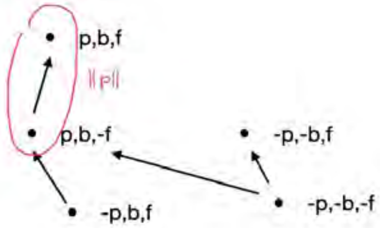


- Each point represents a propositional interpretation.
- *p*, *b*, *f* represent, respectively, 'being a penguin', 'being a bird', and 'being able to fly'.
- $\mathcal{I} \longrightarrow \mathcal{J}$ indicates $\mathcal{I} \prec_{\mathcal{P}} \mathcal{J}$.
- We indicate with $\|\alpha\|_{\mathcal{P}}$ the set of **interpretations satisfying** α in the model $\mathcal{P}$.

KLM Framework - Semantics

Example

Preferential model $\mathcal{P}$:



- Each point represents a propositional interpretation.
- *p*, *b*, *f* represent, respectively, 'being a penguin', 'being a bird', and 'being able to fly'.
- $\mathcal{I} \longrightarrow \mathcal{J}$ indicates $\mathcal{I} \prec_{\mathcal{P}} \mathcal{J}$.
- We indicate with $\|\alpha\|_{\mathcal{P}}$ the set of **interpretations satisfying** α in the model $\mathcal{P}$.

KLM Framework - Semantics

Definition (Preferential interpretation (correct definition!))

Given a propositional language $\mathcal{L}$, let $\mathcal{W}$ be the set of all the interpretations defined over $\mathcal{L}$.

$$\mathcal{P} = \langle \mathcal{S}, I, \prec_{\mathcal{P}} \rangle$$

- $\mathcal{S}$ is a set of objects (states);
- $I : \mathcal{S} \mapsto \mathcal{W}$;
- $\prec_{\mathcal{P}}$: $\mathcal{S} \times \mathcal{S}$ is a preference partial order over the propositional interpretations that satisfies the *smoothness* condition:
  - For every formula α, $\|\alpha\|_{\mathcal{P}} \neq \emptyset$ implies $\min_{\prec_{\mathcal{P}}}(\|\alpha\|_{\mathcal{P}}) \neq \emptyset$, where $\min_{\prec_{\mathcal{P}}}(A) = \{x \in A \mid \nexists y \in A \text{ s.t. } y \prec_{\mathcal{P}} x\}$

KLM Framework - Semantics

Regarding satisfaction, the idea is that a preferential model satisfies the conditional $\alpha \mid\!\sim \beta$ if the *most typical* valuations satisfying α satisfy also β.

Definition (Preferential interpretation - Satisfaction)

Let $\mathcal{P} = \langle \mathcal{S}, l, \prec_{\mathcal{P}} \rangle$ be a preferential interpretation and $\alpha \mid\!\sim \beta$ a conditional.

$$\mathcal{P} \text{ satisfies } \alpha \mid\!\sim \beta \ (\mathcal{P} \Vdash \alpha \mid\!\sim \beta) \text{ iff } \min_{\prec_{\mathcal{P}}}(\|\alpha\|) \subseteq \|\beta\|.$$

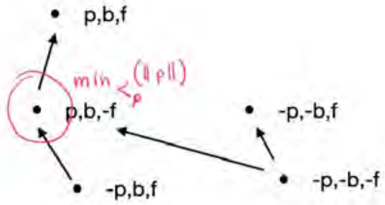
If $\mathcal{P}$ satisfies $\alpha \mid\!\sim \beta$, then $\mathcal{P}$ is a **preferential model** of $\alpha \mid\!\sim \beta$.

Given a set of conditional $KB = \{\alpha_1 \mid\!\sim \beta_1, \alpha_2 \mid\!\sim \beta_2, \ldots\}$, $\mathcal{P}$ is a **preferential model** of KB if $\mathcal{P}$ is a model of every $\alpha_i \mid\!\sim \beta_i \in KB$.

KLM Framework - Semantics

Example

Preferential model $\mathcal{P}$:



- For example, the model $\mathcal{P}$ satisfies $p \mid\sim \neg f$ ($\mathcal{P} \Vdash p \mid\sim \neg f$). That is, typical penguins do not fly.

KLM Framework - Semantics

Kraus, Lehmann and Magidor proved a **representation result**: a **full correspondence** between **preferential sets of conditionals** and **preferential interpretations**.

Theorem ([Kraus et al., 1990])

Let $\mathcal{L} = \{\alpha, \beta, \dots\}$ be a propositional language, $\mathcal{W}$ be the set of propositional interpretations generated by $\mathcal{L}$, and $\mathcal{L}^{\mid\sim} = \{\alpha \mid\sim \alpha, \beta \mid\sim \beta, \alpha \mid\sim \beta, \dots\}$ the conditional language generated from $\mathcal{L}$.

A set of conditionals A ($A \subseteq \mathcal{L}^{\mid\sim}$) is preferential if and only if it corresponds to the set of conditionals satisfied by some preferential interpretation $\mathcal{P} = \langle S, l, \prec_{\mathcal{P}} \rangle$ ($l : S \mapsto \mathcal{W}$).

That is, A is preferential iff there is a $\mathcal{P}$ s.t.

$$A = \{\alpha \mid\sim \beta \mid \mathcal{P} \Vdash \alpha \mid\sim \beta\}.$$

KLM Framework - Preferential Entailment

Now we can model a first form of **reasoning**, that is, an **entailment relation** $\approx_{\mathcal{P}}$ [Lehmann and Magidor, 1992].

Definition (Preferential entailment $\approx_{\mathcal{P}}$)

Let *KB* $= \{\alpha_1 \mid\sim \beta_1, \alpha_2 \mid\sim \beta_2, \ldots\}$ be any set of conditionals and $\gamma \mid\sim \delta$ any conditional.

$$KB \approx_{\mathcal{P}} \gamma \mid\sim \delta$$

if and only if, for every preferential model $\mathcal{P}$ of *KB*,

$$\mathcal{P} \Vdash \gamma \mid\sim \delta.$$

KLM Framework - Preferential Entailment

We have an entailment relation with its semantic characterisation.

Moreover, it is possible to prove that we can reason using the preferential properties as derivation rules.

That is, we have a proof system that uses the preferential properties as derivation rules and is *correct* and *complete* w.r.t. preferential entailment.

KLM Framework - Preferential Entailment

Example

Let

$$KB = \{p \mid\sim \neg f, b \mid\sim f, b \mid\sim ft, p \rightarrow b, r \rightarrow b\},$$

where p, r, b, f, ft represent, respectively, ‘being a penguin’, ‘being a robin’, ‘being a bird’, and ‘being able to fly’, having feathers.

Note: The classical implication $\alpha \rightarrow \beta$ in our KB is an abbreviation for the conditional $\alpha \wedge \neg\beta \mid\sim \bot$, that is satisfied in a preferential model iff no state satisfies $\alpha \wedge \neg\beta$, that is, every state satisfies $\alpha \rightarrow \beta$. No reason to go into further details here.

KLM Framework - Preferential Entailment

Example

From

$$KB = \{p \mid\sim \neg f, b \mid\sim f, b \mid\sim ft, p \to b, r \to b, r \to \neg p\},$$

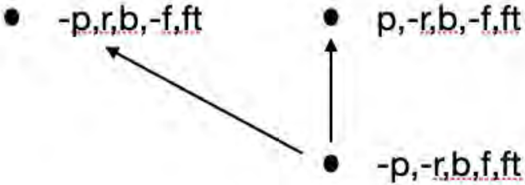
we can conclude, for example, $b \mid\sim f \wedge ft$, since

$$\text{Right Conjunction (And): } \frac{b \mid\sim f, b \mid\sim ft}{b \mid\sim f \wedge ft}$$

KLM Framework - Preferential Entailment

Example

On the other hand, we can prove that $p \mid\!\sim f$ is not derivable from *KB* by creating a counter-model.



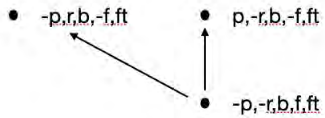
This is a desirable behaviour.

KLM Framework - Preferential Entailment

Example

But we can prove that the preferential entailment is a very weak enatilment relation, since there are a lot of desirable conditionals that we cannot derive.

The model presented before is also a countermodel also for $r \mathrel{|\!\sim} f$.



Since in the KB there is no information saying that robins are atypical birds in some way, we would like to reason about them assuming they are typical birds.

For example, we would like to derive $r \mathrel{|\!\sim} f$.

KLM Framework - Preferential Entailment

Preferential entailment is not able to model one of the main desiderata of presumptive reasoning:

Presumption of Typicality [Lehmann, 1995]:

If we have no reason to conclude that a subclass (e.g. robins) is atypical w.r.t. some super-class (e.g. birds) we have to assume that it inherits all the typical characteristics of the super-class.

KLM Framework - Rational Monotonicity

First, there is a closure property that is interesting from this point of view:

- Rational Monotonicity (RM): $\frac{\alpha \mid\sim \gamma, \alpha \not\mid\sim \neg\beta}{\alpha \wedge \beta \mid\sim \gamma}$

This is a form of constrained monotonicity, stronger than (CM).

$$\frac{b \mid\sim f, b \not\mid\sim \neg r}{r \wedge b \mid\sim f}$$

A preferential set of conditionals that is closed also under (RM) is a **rational** set of conditionals.

Note: since we have $r \rightarrow b$ in the KB, $r \wedge b$ can be substituted simply with r.

KLM Framework - Ranked interpretations

A particular kind of preferential interpretation is introduced.

Definition (Ranked interpretation)

A ranked interpretation $\mathcal{R} = \langle \mathcal{W}, r \rangle$ is s.t. $\mathcal{W}$ is the set of all the propositional interpretations, and the function r is as follows

- $r : \mathcal{W} \mapsto (\mathbb{N} \cup \{\infty\})$ satisfying the following convexity condition: for every $n \in \mathbb{N}$, if $r(\mathcal{I}) = n$ then, for every k s.t. $0 \leq k < n$, there is a $\mathcal{J} \in \mathcal{W}$ for which $r(\mathcal{J}) = k$.

Definition (Ranked interpretation - Satisfaction)

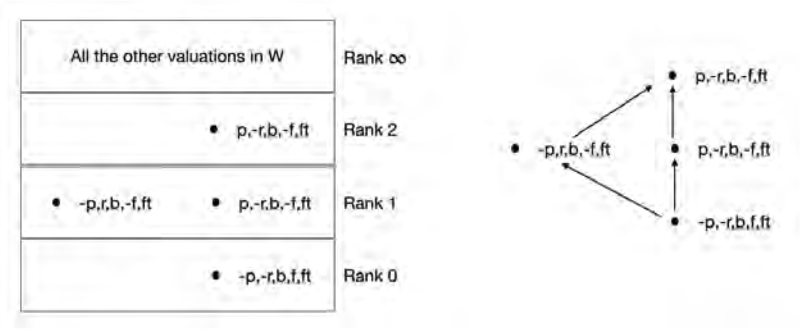
Let $\mathcal{R} = \langle \mathcal{W}, r \rangle$ be a ranked interpretation and $\alpha \mid\!\sim \beta$ a conditional.

$\mathcal{R}$ satisfies $\alpha \mid\!\sim \beta$ ($\mathcal{R} \Vdash \alpha \mid\!\sim \beta$) iff $\min_r(\|\alpha\|) \subseteq \|\beta\|$, where

$$\min_r(\|\alpha\|) = \{\mathcal{I} \in (\|\alpha\|) \mid r(\mathcal{I}) < \infty \text{ and } r(\mathcal{I}) \in \min\{i \mid \mathcal{J} \in (\|\alpha\|) \text{ and } r(\mathcal{J}) = i\}\}$$

KLM Framework - Ranked interpretations

It corresponds exactly to preferential interpretations that are organised in “layers”



A ranked model on the left, and the correspondent preferential model on the right.

KLM Framework - Ranked interpretations

Rational sets of conditionals are characterised by ranked models.

Theorem ([Lehmann and Magidor, 1992])

Let $\mathcal{L} = \{\alpha, \beta, \ldots\}$ be a propositional language, $\mathcal{W}$ be the set of propositional interpretations generated by $\mathcal{L}$, and $\mathcal{L}^{\mid\sim} = \{\alpha \mid\sim \alpha, \beta \mid\sim \beta, \alpha \mid\sim \beta, \ldots\}$ the conditional language generated from $\mathcal{L}$.

A set of conditionals A ($A \subseteq \mathcal{L}^{\mid\sim}$) is rational if and only if it corresponds to the set of conditionals satisfied by some ranked interpretation $\mathcal{R} = \langle W, r \rangle$.

That is, A is rational iff there is a $\mathcal{R}$ s.t.

$$A = \{\alpha \mid\sim \beta \mid \mathcal{R} \Vdash \alpha \mid\sim \beta\}.$$

Rational Closure

Given a set of conditional *KB*, in order to define an entailment relation $\models\!\approx_R$ modelling *Presumption of Typicality* we consider a particular ranked model of *KB*.

1. Given all the ranked models of *KB*, we order them as follows. Let $\mathcal{R} = \langle W, r \rangle, \mathcal{R}' = \langle W, r' \rangle$ be models of *KB*, then

$$\mathcal{R} <_R \mathcal{R}' \text{iff for every } \mathcal{I} \in \mathcal{W},\ r(\mathcal{I}) < r'(\mathcal{I}).$$

2. For every consistent *KB*, we can prove that there is a unique $<_R$-minimum among its models. That is, there is a single model $\mathcal{R}_{KB}$ s.t. $\mathcal{R}_{KB} <_R \mathcal{R}$ for any ranked model $\mathcal{R}$ of *KB* [Giordano et al., 2015].
3. We define the entailment relation $\models\!\approx_R$ as follows:

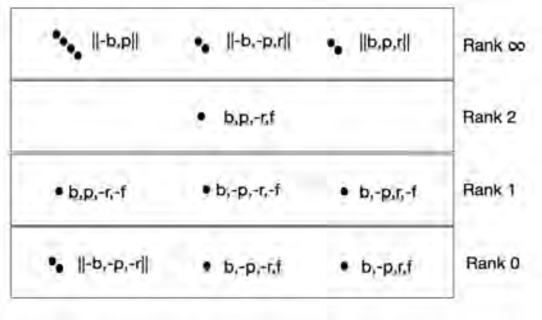
$$KB \models\!\approx_R \alpha \mid\!\sim \beta \text{ iff } \mathcal{R}_{KB} \Vdash \alpha \mid\!\sim \beta$$

It can be proved [Giordano et al., 2015] that this construction corresponds to a well-known consequence relation in non-monotonic logics, that is known as **Rational Closure** [Lehmann and Magidor, 1992] or **System Z** [Pearl, 1990].

KLM Framework - Rational Closure

This is the minimal model of the KB

$$KB = \{p \rightarrow b, r \rightarrow b, p \rightarrow \neg r, b \mathrel{|\!\sim} f, p \mathrel{|\!\sim} \neg f\}.$$



We have $KB \approx_{\mathcal{R}} r \mathrel{|\!\sim} f$, respecting the presumption of typicality.

KLM Framework - Rational Closure

There is another principle that we would like to formalise.

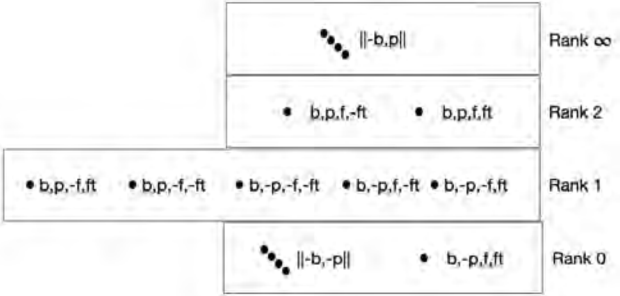
Presumption of Independence [Lehmann, 1995]:

A class a (e.g., birds) typically satisfies the properties b (e.g., flying) and c (e.g., having feathers). A subclass a' (e.g., penguins) of a is atypical, since it does not satisfy b. If we have no reason to conclude that the satisfaction of b has some connection with the satisfaction of c, we should still allow a' to inherit the property c (that is, typical penguins have feathers).
Rational closure does not satisfy the presumption of independence.

KLM Framework - Rational Closure

This is the minimal model of the KB

$$KB = \{p \rightarrow b, b \mid\!\sim f, b \mid\!\sim ft, p \mid\!\sim \neg f\}.$$



The minimal model of *KB* does not satisfy $p \mid\!\sim ft$.

KLM Framework - Rational Closure

- There is the possibility of modelling $\approx_{\mathcal{R}}$ relying only on classical propositional decision procedures.
- There are various proposals built on top of rational closure, extending it, and satisfying the presumption of independence.
- Can all this be adapted to other formalisms, like description logics?

We will consider these issues in the next lecture.

Bibliography

Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2015).
Semantic characterization of rational closure: From propositional logic to description logics.
Artif. Intell., 226:1–33.

Kraus, S., Lehmann, D., and Magidor, M. (1990).
Nonmonotonic reasoning, preferential models and cumulative logics.
Artificial Intelligence, 44(1):167–207.

Lehmann, D. (1995).
Another perspective on default reasoning.
Annals of Mathematics and Artificial Intelligence, 15:61–82.

Lehmann, D. and Magidor, M. (1992).
What does a conditional knowledge base entail?
Artificial Intelligence, 55(1):1–60.

Makinson, D. (2005).
Bridges from classical to nonmonotonic logic, volume 5 of *Texts in computing*.
College Publications.

McCarthy, J. and Hayes, P. (1969).
Some philosophical problems from the standpoint of artificial intelligence.
In Meltzer, B. and Michie, D., editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press.

Pearl, J. (1990).
System z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning.
In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, TARK 90, pages 121–135, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Reiter, R. (1978).
On Closed World Data Bases, pages 55–76.
Springer US, Boston, MA.

Tarski, A. (2002).
On the concept of following logically.
History and Philosophy of Logic, 23(3):155–196.

Non-Classical Knowledge Representation and Reasoning

Italian National PhD Course on AI, 2024

Umberto Straccia & Giovanni Casini

CNR - ISTI, Pisa, Italy
http://www.straccia.info
{umberto.straccia, giovanni.casini}@isti.cnr.it

Recap

In the last lecture:

- The role of nonmonotonicity in Knowledge Representation
- A quick overview of some of the main logical formalisms
- KLM framework:
  - Preferential sets
  - Preferential interpretations and preferential entailment
  - Rational Monotonicity and ranked interpretations
  - Rational Closure

Recap - Preferential sets

A is a **preferential** set of conditionals if it closed under the following properties:

- Reflexivity (Ref): $\alpha \mid\!\sim \alpha$
- Right Weakening (RW): $\dfrac{\alpha \mid\!\sim \beta, \ \models \beta \rightarrow \gamma}{\alpha \mid\!\sim \gamma}$
- Left Logical equivalence (LLE): $\dfrac{\models \alpha \leftrightarrow \beta, \ \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$
- Right Conjunction (And): $\dfrac{\alpha \mid\!\sim \beta, \alpha \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \wedge \gamma}$
- Disjunction in the Premises (Or): $\dfrac{\alpha \mid\!\sim \gamma, \beta \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma}$
- Cautious Monotonicity (CM): $\dfrac{\alpha \mid\!\sim \beta, \ \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$

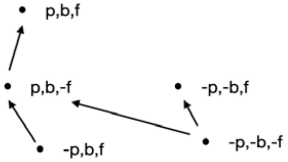
Recap - Preferential intepretations

Definition (Preferential interpretation - simplified version!)

Given a propositional language $\mathcal{L}$, let $\mathcal{W}$ be the set of all the interpretations defined over $\mathcal{L}$.

$$\mathcal{P} = \langle \mathcal{M}, \prec_{\mathcal{P}} \rangle$$

- $\mathcal{M} \subseteq \mathcal{W}$ is a set of interpretations;
- $\prec_{\mathcal{P}}$: $\mathcal{M} \times \mathcal{M}$ is a preference partial order over the propositional interpretations.



Recap - Preferential satisfaction and entailment

Definition (Preferential interpretation - Satisfaction)

Let $\mathcal{P} = \langle \mathcal{S}, I, \prec_{\mathcal{P}} \rangle$ be a preferential interpretation and $\alpha \mid\!\sim \beta$ a conditional.

$$\mathcal{P} \text{ satisfies } \alpha \mid\!\sim \beta \; (\mathcal{P} \Vdash \alpha \mid\!\sim \beta) \text{ iff } \min_{\prec_{\mathcal{P}}}(\|\alpha\|) \subseteq \|\beta\|.$$

Definition (Preferential entailment $\mid\!\approx_{\mathcal{P}}$)

Let *KB* $= \{\alpha_1 \mid\!\sim \beta_1, \alpha_2 \mid\!\sim \beta_2, \ldots\}$ be any set of conditionals and $\gamma \mid\!\sim \delta$ any conditional.

$$KB \mid\!\approx_{\mathcal{P}} \gamma \mid\!\sim \delta$$

if and only if, for every preferential model $\mathcal{P}$ of *KB*,

$$\mathcal{P} \Vdash \gamma \mid\!\sim \delta.$$

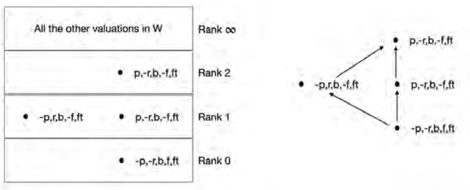
Recap - Rational Monotonicity

Presumption of Typicality [Lehmann, 1995]:

If we have no reason to conclude that a subclass (e.g. robins) is atypical w.r.t. some super-class (e.g. birds) we have to assume that it inherits all the typical characteristics of the super-class.

- Rational Monotonicity (RM): $\dfrac{\alpha \mid\!\sim \gamma, \alpha \not\mid\!\sim \neg\beta}{\alpha \wedge \beta \mid\!\sim \gamma}$

Recap - Rational Closure



A ranked model on the left, and the correspondent preferential model on the right.

- For every consistent *KB*, we can prove that there is a unique $<_R$-minimum among its models. That is, there is a single model $\mathcal{R}_{KB}$ s.t. $\mathcal{R}_{KB} <_R \mathcal{R}$ for any ranked model $\mathcal{R}$ of *KB* [Giordano et al., 2015].
- We define the entailment relation $\approx_R$ as follows:

$$KB \models\!\!\approx_R \alpha \mid\!\sim \beta \text{ iff } \mathcal{R}_{KB} \Vdash \alpha \mid\!\sim \beta$$

Rational Closure and Description Logics

From lecture 2, the language of the Description Logic $\mathcal{ALC}$:

- A given DL is defined by set of concept and role forming operators
- Basic language: $\mathcal{ALC}$ ($\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplement)

Syntax	Semantics	Example
$C, D \rightarrow \top$ |	$\top(x)$	
$\bot$ |	$\bot(x)$	
A |	$A(x)$	$Human$
$C \sqcap D$ |	$C(x) \wedge D(x)$	$Human \sqcap Male$
$C \sqcup D$ |	$C(x) \vee D(x)$	$Nice \sqcup Rich$
$\neg C$ |	$\neg C(x)$	$\neg Meat$
$\exists R.C$ |	$\exists y.R(x,y) \wedge C(y)$	$\exists has\_child.Blond$
$\forall R.C$	$\forall y.R(x,y) \Rightarrow C(y)$	$\forall has\_child.Human$
$C \sqsubseteq D$	$\forall x.C(x) \Rightarrow D(x)$	$Happy\_Father \sqsubseteq Man \sqcap \exists has\_child.Female$
$a{:}C$	$C(a)$	$John{:}Happy\_Father$
$(a, b) : R$	$R(a, b)$	$(John, Mary) : Father\_of$

- TBox ($\mathcal{T}$): a finite set of inclusion axioms ($C \sqsubseteq D$);
- ABox ($\mathcal{A}$): a finite set of assertions about individuals ($a : C \mid (a, b) : R$);
- Knowledge base KB: a pair composed of a Tbox $\mathcal{T}$ and an ABox $\mathcal{A}$ ($KB = \langle \mathcal{T}, \mathcal{A} \rangle$).

Rational Closure and Description Logics

From lecture 2, the semantics of the Description Logic $\mathcal{ALC}$:

- Semantics is given in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where
  - $\Delta^{\mathcal{I}}$ is the domain (a non-empty set)
  - $\cdot^{\mathcal{I}}$ is an interpretation function that maps:
    - Concept (class) name A into a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$
    - Role (property) name R into a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
    - Individual name a into an element of $\Delta^{\mathcal{I}}$
  - Interpretation function $\cdot^{\mathcal{I}}$ is extended to concept expressions:

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(C_1 \sqcap C_2)^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} & = & \{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} & = & \{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}
\end{array}
$$

Defeasible Subsumption

- In the last years there has been a lot of work for introducing defeasible reasoning in formal ontologies.
- Some possible application areas: biomedicine, security, privacy, legal informatics. . .
- Many proposals:
  - Circumscription [Bonatti et al., 2009];
  - Reiter's default [Baader and Hollunder, 1995];
  - Answer Set Programming [Eiter et al., 2008];
  - Novel approaches [Bonatti et al., 2015];
  - Preferential approach [Casini and Straccia, 2010, Giordano et al., 2015, Bonatti, 2019, Britz et al., 2020].

Today we briefly introduce the latter.

Defeasible Subsumption

We can add a new kind of inclusion axioms:

- Defeasible concept subsumption

$$C \sqsubseteq\!\!\!\!\sim D$$

Intuition

- Typical elements of C are in D (exceptional Cs need not)

Example

- $\text{EmpStud} \equiv \text{Student} \sqcap \text{Employee}$
- $\text{Student} \sqsubseteq\!\!\!\!\sim \neg\exists \text{pays.Tax}$
- $\text{EmpStud} \sqsubseteq\!\!\!\!\sim \exists \text{pays.Tax}$
- $\text{EmpStud} \sqcap \text{Parent} \sqsubseteq\!\!\!\!\sim \neg\exists \text{pays.Tax}$

where ‘EmpStud’ represents ‘employed student’.

Defeasible Subsumption

We can formulate the properties for preferential and rational sets of subsumption axioms corresponding to the propositional ones:

$$(\text{Ref})\ C \precsim C \qquad (\text{LLE})\ \frac{C \equiv D, \quad C \precsim E}{D \precsim E} \qquad (\text{And})\ \frac{C \precsim D, \quad C \precsim E}{C \precsim D \sqcap E}$$

$$(\text{Or})\ \frac{C \precsim E, \quad D \precsim E}{C \sqcup D \precsim E} \qquad (\text{RW})\ \frac{C \precsim D, \quad D \sqsubseteq E}{C \precsim E} \qquad (\text{RM})\ \frac{C \precsim D, \quad C \not\precsim \neg E}{C \sqcap E \precsim D}$$

$$(\text{Cons})\ \top \not\precsim \bot$$

Defeasible subsumption - Semantics

Definition (Modular Order)

Given a set X, $\prec \subseteq X \times X$ is modular if there is a ranking function $rk : X \longrightarrow \mathbb{N}$ s.t. for every $x, y \in X$, $x \prec y$ iff $rk(x) < rk(y)$

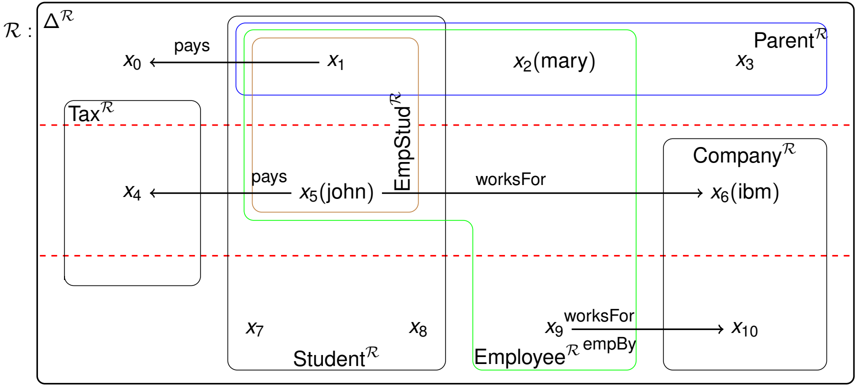
Definition (Modular Interpretation)

A modular interpretation is a ternary tuple $\mathcal{R} = \langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}}, \prec^{\mathcal{R}} \rangle$ where $\langle \Delta^{\mathcal{R}}, \cdot^{\mathcal{R}} \rangle$ is a DL interpretation and $\prec^{\mathcal{R}}$ is a modular order

Intuition

- The domain of interpretation is partitioned into ranks
- All objects are comparable (except if they are in the same rank)

Defeasible subsumption - Semantics



$$\mathcal{R} \Vdash C \mathrel{\sqsubseteq\!\!\!\!\sim} D \text{ iff } \min_{rk}(\|C\|_{\mathcal{R}}) \subseteq \|D\|_{\mathcal{R}}$$

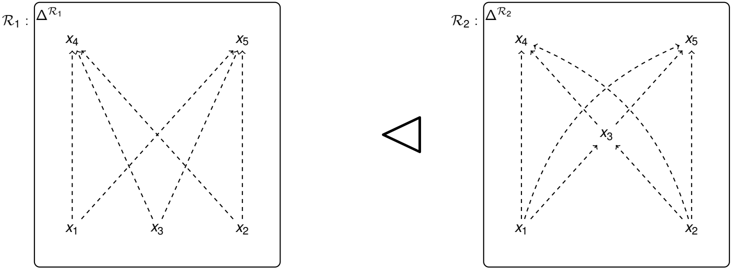
Defeasible subsumption - Semantics

Preferring maximal typicality [Giordano et al., 2015]

Example

Let $\mathcal{R}_1 = \langle \Delta^{\mathcal{R}_1}, \cdot^{\mathcal{R}_1}, \prec^{\mathcal{R}_1} \rangle$ and $\mathcal{R}_2 = \langle \Delta^{\mathcal{R}_2}, \cdot^{\mathcal{R}_2}, \prec^{\mathcal{R}_2} \rangle$ be such that

- $\Delta^{\mathcal{R}_1} = \Delta^{\mathcal{R}_2} = \{x_i \mid 1 \leq i \leq 5\}$ (same domain!), $\cdot^{\mathcal{R}_1} = \cdot^{\mathcal{R}_2}$, $\prec^{\mathcal{R}_1}$ and $\prec^{\mathcal{R}_2}$ as below



Defeasible subsumption - minimal model

- Let us fix an infinite countable domain Δ^R (simplification!) and fix an interpretation function $\cdot^R$.
- Given a TBox $\mathcal{T}$, the $\lhd$-minimal model of $\mathcal{T}$ among those having Δ^R as domain and $\cdot^R$ as interpretation function ($\min_{\lhd}([KB]_{\Delta^R})$) is unique [Giordano et al., 2015].

Definition (Minimal ranked entailment/Rational Closure [Giordano et al., 2015, Britz et al., 2020])

Let $\mathcal{T}$ be a defeasible TBox.

$$KB \approx_R C \sqsubseteq\!\!\!\!\sim D \text{ iff } \min_{\lhd}([\mathcal{T}]_{\Delta^R}) \Vdash C \sqsubseteq\!\!\!\!\sim D$$

- There is an algorithm for computing minimal ranked entailment.
- Input: KB and α; Output: Yes iff $\mathcal{T} \approx_R C \sqsubseteq\!\!\!\!\sim D$
- It can be implemented on top of any classical $\mathcal{ALC}$ reasoner.

Defeasible subsumption - minimal model

If we have also an ABox, then it is possible that the minimal model is not unique.

Example

$$\mathcal{T} = \left\{ \top \sqsubseteq \!\!\!\! \sim A \sqcap \forall r.\neg A, \right\} \qquad \mathcal{A} = \{ \ (a, b) : r \ \}$$

The models for this KB have two minimal configurations:

1. a is at rank 0, and b is exceptional. Hence $a : A$ and $b : \neg A$.
2. b is at rank 0, and a is exceptional. Hence $b : A$.

Different possible approaches:

- **Skeptical**: we take only the conclusions that are common to all the possible options.
- **Credulous**: we take all the conclusions that are satisfied in at least one possible options.
- **Choice**: we choose one specfic option and we take all the conclusions that satisfied in it.

RDFS

- RDFS: W3C standard and popular formalism for KR
- Statements
  - Triples of the form (s, p, o)
  - Informally, binary predicate $p(s, o)$
    - $(\texttt{fever}, \texttt{hasTreatment}, \texttt{paracetamol})$
  - Special predicates: typing and specialisations, etc.
    - $(\texttt{paracetamol}, \text{type}, \texttt{antipyretic})$
    - $(\texttt{antipyretic}, \text{sc}, \texttt{drug})$

Main Steps

- We start from the logic ρdf
  - A minimal, but significant RDFS fragment
  - Covers all essential features of RDFS
- We extend ρdf into $\rho df_\perp = \rho$df + disjointness statements
  - disjointness relationships:

$$(\texttt{opioid}, \perp_\mathsf{c}, \texttt{antipyretic})$$
$$(\texttt{hasDrugAddiction}, \perp_\mathsf{p}, \texttt{usesDrugControlled})$$

- We extend $\rho df_\perp$ into *defeasible* $\rho df_\perp$ adding defeasible information
  - defeasible triples:

$$\langle \texttt{DrugUser}, \mathsf{sc}, \texttt{Young} \rangle$$
$$\langle \texttt{usesDrug}, \mathsf{sp}, \texttt{hasDrugAddiction} \rangle$$

Preliminaries: ρdf

- ρdf: defined on subset of the RDFS vocabulary:

$$\rho\text{df} = \{\text{sp}, \text{sc}, \text{type}, \text{dom}, \text{range}\}$$

Informally,

- (p, sp, q)
  - p is a sub property of property q
- (c, sc, d)
  - c is a sub class of class d
- (a, type, b)
  - a is of type b
- (p, dom, c)
  - domain of property p is c
- $(p, \textit{range}, c)$
  - range of property p is c

Example

$$G = \{(\texttt{yP}, \textsf{sc}, \texttt{hP})$$
$$(\texttt{dU}, \textsf{sc}, \texttt{uhP})$$
$$(\texttt{dU}, \textsf{sc}, \texttt{yP})$$
$$(\texttt{cDU}, \textsf{sc}, \texttt{hP})$$
$$(\texttt{cDU}, \textsf{sc}, \texttt{dU})\}$$

Read:

- yP → 'Young People';
- hP → 'Happy People';
- dU → 'Drug Users';
- uhP → 'Unhappy People';
- cDU → 'Controlled Drug User';

ρdfSemantics

ρdf interpretation:

$$\mathcal{I} = \langle \Delta_{\mathsf{R}}, \Delta_{\mathrm{DP}}, \Delta_{\mathsf{C}}, \Delta_{\mathsf{L}}, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle \ ,$$

1. Δ_{R} are the resources
2. Δ_{DP} are property names
3. $\Delta_{\mathsf{C}} \subseteq \Delta_{\mathsf{R}}$ are the classes
4. $\Delta_{\mathsf{L}} \subseteq \Delta_{\mathsf{R}}$ are the literal values and contains all the literals in $\mathbf{L} \cap V$
5. $P[\![\cdot]\!]$ is a function $P[\![\cdot]\!] \colon \Delta_{\mathrm{DP}} \to 2^{\Delta_{\mathsf{R}} \times \Delta_{\mathsf{R}}}$
6. $C[\![\cdot]\!]$ is a function $C[\![\cdot]\!] \colon \Delta_{\mathsf{C}} \to 2^{\Delta_{\mathsf{R}}}$
7. $\cdot^{\mathcal{I}}$ maps each $t \in \mathbf{UL} \cap V$ into a value $t^{\mathcal{I}} \in \Delta_{\mathsf{R}} \cup \Delta_{\mathrm{DP}}$, where $\cdot^{\mathcal{I}}$ is the identity for literals; and
8. $\cdot^{\mathcal{I}}$ maps each variable $x \in \mathbf{B}$ into a value $x^{\mathcal{I}} \in \Delta_{\mathsf{R}}$

ρdf model/entailment

$\mathcal{I} \models G$ if and only if $\mathcal{I}$ satisfies conditions

Simple:

1. for each $(s, p, o) \in G$, $p^{\mathcal{I}} \in \Delta_{\mathrm{DP}}$ and $(s^{\mathcal{I}}, o^{\mathcal{I}}) \in P[\![p^{\mathcal{I}}]\!]$

Subproperty:

1. $P[\![\mathrm{sp}^{\mathcal{I}}]\!]$ is transitive over Δ_{DP}
2. if $(p, q) \in P[\![\mathrm{sp}^{\mathcal{I}}]\!]$ then $p, q \in \Delta_{\mathrm{DP}}$ and $P[\![p]\!] \subseteq P[\![q]\!]$

Subclass:

1. $P[\![\mathrm{sc}^{\mathcal{I}}]\!]$ is transitive over Δ_{C}
2. if $(c, d) \in P[\![\mathrm{sc}^{\mathcal{I}}]\!]$ then $c, d \in \Delta_{\mathrm{C}}$ and $C[\![c]\!] \subseteq C[\![d]\!]$

Typing I:

1. $x \in C[\![c]\!]$ if and only if $(x, c) \in P[\![\mathrm{type}^{\mathcal{I}}]\!]$;
2. if $(p, c) \in P[\![\mathrm{dom}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $x \in C[\![c]\!]$
3. if $(p, c) \in P[\![\mathrm{range}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $y \in C[\![c]\!]$

Typing II:

1. for each $\mathrm{e} \in \rho\mathrm{df}$, $\mathrm{e}^{\mathcal{I}} \in \Delta_{\mathrm{DP}}$;
2. if $(p, c) \in P[\![\mathrm{dom}^{\mathcal{I}}]\!]$ then $p \in \Delta_{\mathrm{DP}}$ and $c \in \Delta_{\mathrm{C}}$
3. if $(p, c) \in P[\![\mathrm{range}^{\mathcal{I}}]\!]$ then $p \in \Delta_{\mathrm{DP}}$ and $c \in \Delta_{\mathrm{C}}$
4. if $(x, c) \in P[\![\mathrm{type}^{\mathcal{I}}]\!]$ then $c \in \Delta_{\mathrm{C}}$.

$G \models H$ if and only if every model of G is also a model of H

Deductive System for ρdf

$G \vdash H$

1. Simple:

   (a) $\frac{G}{G'}$ for a map $\mu : G' \to G$ (b) $\frac{G}{G'}$ for $G' \subseteq G$

2. Subproperty:

   (a) $\frac{(A,\text{sp},B),(B,\text{sp},C)}{(A,\text{sp},C)}$ (b) $\frac{(D,\text{sp},E),(X,D,Y)}{(X,E,Y)}$

3. Subclass:

   (a) $\frac{(A,\text{sc},B),(B,\text{sc},C)}{(A,\text{sc},C)}$ (b) $\frac{(A,\text{sc},B),(X,\text{type},A)}{(X,\text{type},B)}$

4. Typing:

   (a) $\frac{(D,\text{dom},B),(X,D,Y)}{(X,\text{type},B)}$ (b) $\frac{(D,\text{range},B),(X,D,Y)}{(Y,\text{type},B)}$

5. Implicit Typing:

   (a) $\frac{(A,\text{dom},B),(D,\text{sp},A),(X,D,Y)}{(X,\text{type},B)}$

   (b) $\frac{(A,\text{range},B),(D,\text{sp},A),(X,D,Y)}{(Y,\text{type},B)}$

Extending ρdfinto $\rho df_\perp$

$\rho df_\perp$ Syntax:

- Disjointness predicates: $\perp_c$ and $\perp_p$
  - $(c, \perp_c, d)$: classes c and d are disjoint
  - $(p, \perp_p, q)$: properties p and q are disjoint

Example

$$
\begin{aligned}
G = \{&(\mathtt{yP}, \mathsf{sc}, \mathtt{hP})\\
&(\mathtt{dU}, \mathsf{sc}, \mathtt{uhP})\\
&(\mathtt{dU}, \mathsf{sc}, \mathtt{yP})\\
&(\mathtt{cDU}, \mathsf{sc}, \mathtt{hP})\\
&(\mathtt{cDU}, \mathsf{sc}, \mathtt{dU})\}
\end{aligned}
$$

$$G' = G \cup \{(\mathtt{uhP}, \perp_c, \mathtt{hP})\}$$

$(\mathtt{uhP}, \perp_c, \mathtt{hP})$:
the class *Unhappy People* and the class *Happy People* are disjoint.

$\rho df_\perp$

- Objectives of $\rho df_\perp$ semantics:

  1. Deductive system = ρdf + some additional rules
     - any RDFS reasoner/store may handle the new triples as ordinary triples if it does not want to take account of the extra inference capabilities
  2. Any $\rho df_\perp$ graph is satisfiable.
  3. Computational complexity stays in the same class as ρdf.

For a detailed semantics, see [Casini and Straccia, 2023].

$\rho df_\perp$ Deductive system

From an inference system point of view, new derivation rules are added to the ρdf derivation system. For example:

- Conceptual Disjointness:

  $(a)\quad \frac{(A,\perp_c,B)}{(B,\perp_c,A)} \qquad (b)\quad \frac{(A,\perp_c,B),(C,sc,A)}{(C,\perp_c,B)} \qquad (c)\quad \frac{(A,\perp_c,A)}{(A,\perp_c,B)}$

We define an entailment relation $\models_\perp$ and a derivation relation $\vdash_\perp$ that extend the ρdf ones and $\vdash_\perp$ is correct and complete w.r.t. $\models_\perp$.

Example (Cont.)

From $(uhP, \perp_c, hP)$, (cDU, sc, hP), (cDU, sc, dU) and (dU, sc, uhP) we conclude $(cDU, \perp_c, cDU)$.

$$
\dfrac{\dfrac{\dfrac{(uhP, \perp_c, hP)}{(hP, \perp_c, uhP)} \qquad (cDU, \text{sc}, hP)}{(cDU, \perp_c, uhP)} \qquad \dfrac{(cDU, \text{sc}, dU) \qquad (dU, \text{sc}, uhP)}{(cDU, \text{sc}, uhP)}}{(cDU, \perp_c, cDU)}
$$

Hence, being a controlled drug user is incompatible with being a controlled drug user (that is, *cDU* should be an empty class).

Analogously, from $(uhP, \perp_c, hP)$, (dU, sc, yP), (yP, sc, hP) and (dU, sc, uhP) we conclude $(dU, \perp_c, dU)$.

Defeasible $\rho df_\perp$

Triples $(A, \perp_c, A)$ and $(A, \perp_p, A)$ indicate an *incoherence*, a *conflict* in our graph.

Such conflicts can be solved introducing defeasible reasoning.

We introduce in our language triples:

- $\langle A, sc, B \rangle$: "The instances of the class A are usually also instances of the class B".
- $\langle A, sp, B \rangle$: "The instances of the property A are usually also instances of the property B".

Defeasible $\rho df_\perp$- Minimal Entailment

Ranked $\rho df_\perp$ Interpretations.

A ranked interpretation is a pair $\mathcal{R} = (\mathcal{M}, r)$, where $\mathcal{M}$ is the set of all $\rho df_\perp$ interpretations defined on a fixed set of domains $\Delta_R, \Delta_P, \Delta_C, \Delta_L$, and r is a ranking function over $\mathcal{M}$

$$r : \mathcal{M} \mapsto \mathbb{N} \cup \{\infty\}$$

satisfying a convexity property:

- there is an interpretation $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$;
- for each $i > 0$, if there is an interpretation $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = i$, then there is an interpretation $\mathcal{I}' \in \mathcal{M}$ s.t. $r(\mathcal{I}') = (i - 1)$.

Defeasible $\rho df_\perp$- Minimal Entailment

rank ∞	$\mathcal{M}^F \setminus$ (rank 0 $\cup$ rank 1 $\cup$ rank 2)
rank 2	$\|(s, \mathrm{sc}, b)\| \setminus$ (rank 0 $\cup$ rank 1)
rank 1	$\|(b, \mathrm{sc}, f) \cup (s, \mathrm{sc}, b)\| \setminus$ rank 0
rank 0	$\|(b, \perp_{\mathrm{c}}, b) \cup (s, \mathrm{sc}, b)\|$

$$\begin{aligned}\text{c\_min}(t, \mathcal{R}) \;&=\; \{\mathcal{I} \in \mathcal{M}_{\mathbb{N}} \mid \mathcal{I} \not\models (t, \perp_{\mathrm{c}}, t) \text{ and there is no } \mathcal{I}' \in \mathcal{M}_{\mathbb{N}} \text{ s.t.} \\ &\qquad \mathcal{I}' \not\models (t, \perp_{\mathrm{c}}, t) \text{ and } r(\mathcal{I}') < r(\mathcal{I})\}\ . \\ \text{p\_min}(t, \mathcal{R}) \;&=\; \{\mathcal{I} \in \mathcal{M}_{\mathbb{N}} \mid \mathcal{I} \not\models (t, \perp_{\mathrm{p}}, t) \text{ and there is no } \mathcal{I}' \in \mathcal{M}_{\mathbb{N}} \text{ s.t.} \\ &\qquad \mathcal{I}' \not\models (t, \perp_{\mathrm{p}}, t) \text{ and } r(\mathcal{I}') < r(\mathcal{I})\}\ .\end{aligned}$$

where $\mathcal{M}_{\mathbb{N}} = \{\mathcal{I} \in \mathcal{M} \mid r(\mathcal{I}) \in \mathbb{N}\}$.

E.g., in the above ranked interpretation $\mathcal{R}$, the interpretations in c_min$(b, \mathcal{R})$ will be in rank 1.

Defeasible $\rho df_\perp$- Minimal Entailment

Given a graph G and a fixed set of $\rho df_\perp$-interpretations in our ranked models (see reference for the details), we take under consideration the minimal ranked model for G, that is, the ranked model in which every $\rho df_\perp$-interpretations is ranked as low as possible.

- For every G, the minimal ranked model $\mathcal{R}_{\min G}$ exists and it is unique!

Defeasible $\rho df_\perp$- Minimal Entailment

Minimal Entailment $\models_{\min}$ is defined by the minimal ranked model of a graph G.

$$G \models_{\min} [s, p, o], \text{ iff } \mathcal{R}_{\min G} \models [s, p, o].$$

with $[s, p, o] \in \{(s, p, o), \langle s, p, o \rangle\}$.

We now present (via an example) the decision procedure that is correct and complete w.r.t. $\models_{\min}$.

Defeasible $\rho df_{\perp}$- Example

$$
\begin{aligned}
G' = \{&(\mathtt{yP}, \textbf{sc}, \mathtt{hP})\\
&(\mathtt{dU}, \textbf{sc}, \mathtt{uhP})\\
&(\mathtt{dU}, \textbf{sc}, \mathtt{yP})\\
&(\mathtt{cDU}, \textbf{sc}, \mathtt{hP})\\
&(\mathtt{cDU}, \textbf{sc}, \mathtt{dU})\\
&(\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP})\}
\end{aligned}
$$

Defeasible $\rho df_{\perp}$- Example

$$
\begin{aligned}
G' = \{ & \langle \texttt{yP}, \textbf{sc}, \texttt{hP} \rangle \\
& \langle \texttt{dU}, \textbf{sc}, \texttt{uhP} \rangle \\
& \langle \texttt{dU}, \textbf{sc}, \texttt{yP} \rangle \\
& \langle \texttt{cDU}, \textbf{sc}, \texttt{hP} \rangle \\
& (\texttt{cDU}, \textbf{sc}, \texttt{dU}) \\
& (\texttt{uhP}, \perp_{\textbf{c}}, \texttt{hP}) \}
\end{aligned}
$$

$\langle \texttt{yP}, \textbf{sc}, \texttt{hP} \rangle$:
Young People are usually *Happy People*.

Defeasible $\rho df_\perp$- Ranking

Informally:

1. Create a ranking of the defeasible triples in G.
   - Check the presence of potential conflicts in a graph:
     - translate all the defeasible triples into $\rho df_\perp$ triples.

       $$\langle A, \mathrm{sc}, B \rangle \Rightarrow (A, \mathrm{sc}, B)$$

     - Once a triple $(A, \perp_{\mathrm{c}}, A)$ (resp. $(A, \perp_{\mathrm{p}}, A)$) is derived, all the triples $\langle A, \mathrm{sc}, B \rangle$ (resp. $\langle A, \mathrm{sp}, B \rangle$) are associated to a higher rank.
     - We iterate the procedure.

Defeasible $\rho df_{\bot}$- Example

$$
\begin{aligned}
G' = \{ & \langle \texttt{yP}, \textbf{sc}, \texttt{hP} \rangle \\
& \langle \texttt{dU}, \textbf{sc}, \texttt{uhP} \rangle \\
& \langle \texttt{dU}, \textbf{sc}, \texttt{yP} \rangle \\
& \langle \texttt{cDU}, \textbf{sc}, \texttt{hP} \rangle \\
& (\texttt{cDU}, \textbf{sc}, \texttt{dU}) \\
& (\texttt{uhP}, \bot_{\textsf{c}}, \texttt{hP}) \}
\end{aligned}
$$

Defeasible $\rho df_{\perp}$- Example

$$G' = \{(\texttt{yP}, \textsf{sc}, \texttt{hP})$$
$$(\texttt{dU}, \textsf{sc}, \texttt{uhP})$$
$$(\texttt{dU}, \textsf{sc}, \texttt{yP})$$
$$(\texttt{cDU}, \textsf{sc}, \texttt{hP})$$
$$(\texttt{cDU}, \textsf{sc}, \texttt{dU})$$
$$(\texttt{uhP}, \perp_{\textsf{c}}, \texttt{hP})\}$$

- We derive $(dU, \perp_{\mathsf{c}}, dU)$ and $(cDU, \perp_{\mathsf{c}}, cDU)$.

Defeasible $\rho df_\perp$- Example

$$G' = \{\sout{\langle \mathtt{yP}, \mathbf{sc}, \mathtt{hP} \rangle}$$
$$\langle \mathtt{dU}, \mathbf{sc}, \mathtt{uhP} \rangle$$
$$\langle \mathtt{dU}, \mathbf{sc}, \mathtt{yP} \rangle$$
$$\langle \mathtt{cDU}, \mathbf{sc}, \mathtt{hP} \rangle$$
$$(\mathtt{cDU}, \mathbf{sc}, \mathtt{dU})$$
$$(\mathtt{uhP}, \perp_{\mathbf{c}}, \mathtt{hP})\}$$

- All the defeasible triples with *dU* and *cDU* as first members move to the first rank.

Defeasible $\rho df_{\perp}$- Example

$$
\begin{aligned}
G_1' = \{ & \langle \mathtt{dU}, \mathsf{sc}, \mathtt{uhP} \rangle \\
& \langle \mathtt{dU}, \mathsf{sc}, \mathtt{yP} \rangle \\
& \langle \mathtt{cDU}, \mathsf{sc}, \mathtt{hP} \rangle \\
& (\mathtt{cDU}, \mathsf{sc}, \mathtt{dU}) \\
& (\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP}) \}
\end{aligned}
$$

- Now we consider the information at the first rank.

Defeasible $\rho df_{\perp}$ - Example

$$
\begin{aligned}
G_1' = \{&(\texttt{dU}, \textbf{sc}, \texttt{uhP})\\
&(\texttt{dU}, \textbf{sc}, \texttt{yP})\\
&(\texttt{cDU}, \textbf{sc}, \texttt{hP})\\
&(\texttt{cDU}, \textbf{sc}, \texttt{dU})\\
&(\texttt{uhP}, \perp_{\mathsf{c}}, \texttt{hP})\}
\end{aligned}
$$

- We can still derive $(cDU, \perp_{\mathsf{c}}, cDU)$.

Defeasible $\rho df_\perp$ - Example

$$G'_1 = \{ \sout{\langle \mathtt{dU}, \mathsf{sc}, \mathtt{uhP} \rangle}$$
$$\sout{\langle \mathtt{dU}, \mathsf{sc}, \mathtt{yP} \rangle}$$
$$\langle \mathtt{cDU}, \mathsf{sc}, \mathtt{hP} \rangle$$
$$(\mathtt{cDU}, \mathsf{sc}, \mathtt{dU})$$
$$(\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP}) \}$$

- All the defeasible triples with *cDU* as first member move to the second rank.

Defeasible $\rho df_\perp$- Example

$$G'_2 = \{\langle \texttt{cDU}, \mathsf{sc}, \texttt{hP} \rangle \\ (\texttt{cDU}, \mathsf{sc}, \texttt{dU}) \\ (\texttt{uhP}, \perp_{\mathsf{c}}, \texttt{hP})\}$$

- Now we consider the information at the second rank.

Defeasible $\rho df_{\perp}$- Example

$$
\begin{aligned}
G_2' = \{ & (\mathtt{cDU}, \mathsf{sc}, \mathtt{hP}) \\
& (\mathtt{cDU}, \mathsf{sc}, \mathtt{dU}) \\
& (\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP})\}
\end{aligned}
$$

- We cannot derive any more conflicts. The ranking is done.

Defeasible $\rho df_{\perp}$- Ranking

2 Decision procedure for a query $\langle s, \text{sc}, o\rangle$:

- Given a query $\langle s, \text{sc}, o\rangle$ (resp., $\langle s, \text{sp}, o\rangle$), check the rank of s:
  - check which is the lowest rank in which we do not derive $(s, \perp_{\text{c}}, s)$ (resp., $(s, \perp_{\text{p}}, s)$).
- Given the rank, check whether we can derive (s, sc, o) (resp., (s, sp, o)).

- Deciding whether a graph G defeasibly implies $\langle s, p, o\rangle$ can be done in polynomial time (ground case).

Defeasible $\rho df_{\perp}$- Example

Query: $\langle cDU, \mathsf{sc}, uhP \rangle$.

$$
\begin{aligned}
G' = \{ & \langle \mathtt{yP}, \textbf{sc}, \mathtt{hP} \rangle \\
& \langle \mathtt{dU}, \textbf{sc}, \mathtt{uhP} \rangle \\
& \langle \mathtt{dU}, \textbf{sc}, \mathtt{yP} \rangle \\
& \langle \mathtt{cDU}, \textbf{sc}, \mathtt{hP} \rangle \\
& (\mathtt{cDU}, \textbf{sc}, \mathtt{dU}) \\
& (\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP}) \}
\end{aligned}
$$

- We check at which rank $(cDU, \perp_{\mathsf{c}}, cDU)$ does not hold.

Defeasible $\rho df_\perp$- Example

Query: $\langle cDU, \text{sc}, uhP \rangle$.

$$G' = \{(\texttt{yP}, \textbf{sc}, \texttt{hP})$$
$$(\texttt{dU}, \textbf{sc}, \texttt{uhP})$$
$$(\texttt{dU}, \textbf{sc}, \texttt{yP})$$
$$(\texttt{cDU}, \textbf{sc}, \texttt{hP})$$
$$(\texttt{cDU}, \textbf{sc}, \texttt{dU})$$
$$(\texttt{uhP}, \perp_{\text{c}}, \texttt{hP})\}$$

- Considering the entire graph (rank 0), we derive $(cDU, \perp_{\text{c}}, cDU)$.

Defeasible $\rho df_\perp$- Example

Query: $\langle cDU, \mathsf{sc}, uhP \rangle$.

$$
\begin{aligned}
G'_1 = \{ & \cancel{\langle \mathtt{yP}, \mathbf{sc}, \mathtt{hP} \rangle} \\
& \langle \mathtt{dU}, \mathbf{sc}, \mathtt{uhP} \rangle \\
& \langle \mathtt{dU}, \mathbf{sc}, \mathtt{yP} \rangle \\
& \langle \mathtt{cDU}, \mathbf{sc}, \mathtt{hP} \rangle \\
& (\mathtt{cDU}, \mathbf{sc}, \mathtt{dU}) \\
& (\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP}) \}
\end{aligned}
$$

- Considering the graph at rank 1, we still derive $(cDU, \perp_{\mathsf{c}}, cDU)$.

Defeasible $\rho df_{\bot}$- Example

Query: $\langle cDU, \mathsf{sc}, uhP \rangle$.

$$
\begin{aligned}
G_2' = \{ & \cancel{\langle \mathtt{yP}, \mathsf{sc}, \mathtt{hP} \rangle} \\
& \cancel{\langle \mathtt{dU}, \mathsf{sc}, \mathtt{uhP} \rangle} \\
& \cancel{\langle \mathtt{dU}, \mathsf{sc}, \mathtt{yP} \rangle} \\
& \langle \mathtt{cDU}, \mathsf{sc}, \mathtt{hP} \rangle \\
& (\mathtt{cDU}, \mathsf{sc}, \mathtt{dU}) \\
& (\mathtt{uhP}, \bot_{\mathsf{c}}, \mathtt{hP}) \}
\end{aligned}
$$

- Considering the graph at rank 2, we do not derive $(cDU, \bot_{\mathsf{c}}, cDU)$.

Defeasible $\rho df_{\perp}$- Example

Query: $\langle cDU, \text{sc}, uhP \rangle$.

$$
\begin{aligned}
G'_2 = \{ & \cancel{\langle \texttt{yP}, \textbf{sc}, \texttt{hP} \rangle} \\
& \cancel{\langle \texttt{dU}, \textbf{sc}, \texttt{uhP} \rangle} \\
& \cancel{\langle \texttt{dU}, \textbf{sc}, \texttt{yP} \rangle} \\
& \langle \texttt{cDU}, \textbf{sc}, \texttt{hP} \rangle \\
& (\texttt{cDU}, \textbf{sc}, \texttt{dU}) \\
& (\texttt{uhP}, \perp_{\textbf{c}}, \texttt{hP}) \}
\end{aligned}
$$

- We have to check entailment of $\langle cDU, \text{sc}, uhP \rangle$ w.r.t. this graph.

Defeasible $\rho df_{\perp}$- Example

Query: $\langle cDU, \text{sc}, uhP \rangle$.

$$G_2' = \{\cancel{\langle \mathtt{yP}, \textbf{sc}, \mathtt{hP} \rangle}$$
$$\cancel{\langle \mathtt{dU}, \textbf{sc}, \mathtt{uhP} \rangle}$$
$$\cancel{\langle \mathtt{dU}, \textbf{sc}, \mathtt{yP} \rangle}$$
$$(\mathtt{cDU}, \textbf{sc}, \mathtt{hP})$$
$$(\mathtt{cDU}, \textbf{sc}, \mathtt{dU})$$
$$(\mathtt{uhP}, \perp_{\mathsf{c}}, \mathtt{hP})\}$$

- From this graph we cannot derive (cDU, sc, uhP). Hence $\langle cDU, \text{sc}, uhP \rangle$ is not entailed.

BELIEF CHANGE

The following slides are from a course held at ESSLLI 2018 and prepared with Richard Booth (University of Cardiff)

AGM Theory

Example

Suppose our knowledge base contains the following facts:

A: Sweden is a part of Europe

B: All European swans are white

C: The bird caught in the trap is a swan

D: The bird caught in the trap is from Sweden

From A-D we conclude (using propositional logic)

E: The bird caught in the trap is white

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Example

BUT

Suppose we see that the swan is black. (ie., $\neg E$)

- Want to add $\neg E$ to our database but then database is inconsistent!
- Must change database, but HOW?
- This is the problem of BELIEF REVISION
- Information is valuable. Don't want to throw away unnecessarily.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

2 Main Types of Belief Change

- Revision: $K * \alpha$ is the result of changing K to include α while maintaining consistency.
- Contraction: $K \div \alpha$ is the result of removing α from (the consequences of) K.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Formal Setting

QUESTION: How do we formally represent knowledge bases such as the one in the example?

ANSWER: Use a formal logic $(\mathcal{L}, Cn)$, where:

- $\mathcal{L}$ = set of formulas with which we describe facts
- $Cn : 2^{\mathcal{L}} \to 2^{\mathcal{L}}$ is the consequence relation specifying which formulas follow from any given knowledge base

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Formal Setting

Some assumptions made about (L, Cn) in AGM:

- Language L

  L is closed under propositional connectives

- Consequence operator Cn

  If $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$ (Monotonicity)

  $Cn(A) = Cn(Cn(A))$ (Idempotence)

  $A \subseteq Cn(A)$ (Inclusion)

  } Cn is Tarskian

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Formal Setting

- Consequence operator Cn (continued)

  $\beta \in Cn(A \cup \{\alpha\})$ iff $\alpha \to \beta \in Cn(A)$ (Deduction)

  If $\alpha \in Cl(A)$ then $\alpha \in Cn(A)$ (Supraclassicality)

  (where Cl is classical logical consequence)

  If $\alpha \in Cn(A)$ then $\alpha \in Cn(A')$ for some finite $A' \subseteq A$

  (Compactness)

  $Cn(A \cup \{\alpha\}) \cap Cn(A \cup \{\beta\}) \subseteq Cn(A \cup \{\alpha \vee \beta\})$

  (Disjunction in premisses)

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Formal Setting

LANGUAGE

- L = set of formulas built from some set of propositional atoms $\{p_1, p_2, p_3, \ldots\}$ and connectives $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$.

SEMANTICS

- Set of valuations/interpretations over the set of prop. atoms.
- Each valuation v assigns one of T, F to each p_i.

Notation

- Given $\alpha \in L$, $v \models \alpha \Leftrightarrow v$ evaluates α to T
- $Mod(\alpha) = \{v \mid v \models \alpha\}$
- Given $B \subseteq L$, $Cn(B) = \{\alpha \in L \mid Mod(B) \subseteq Mod(\alpha)\}$

  (the classical logical consequences of B). $\alpha \in Cn(B)$ also written $B \models \alpha$.
- In examples, valuations will often be written as a set of literals, with presence of negation $\neg p$ indicating $v(p) = F$.

Notation

BELIEF SETS/THEORIES

- If $B = Cn(B)$ then we call B a belief set, or sometimes theory.
- Belief sets usually denoted by K, K', etc.
- Since we're interested in consequences of knowledge base, easier to assume knowledge bases are belief sets.

Belief revision: The question formalised

Given a belief set $K \subseteq L$, and a formula α

Find $K * \alpha$, the result of revising K to include α

such that the result is consistent.

Revision = contraction + expansion

- Could try $K * \alpha = Cn(K \cup \{\alpha\})$, but this might be inconsistent.
- Strategy: First make changes to K before adding α
  "make some room for α to come in"
- This can be achieved by contracting by $\neg\alpha$ (since $Cn(K \cup \{\alpha\})$ consistent $\iff \neg\alpha \notin K$)
- So we set $K * \alpha = Cn(K \dot{-} \neg\alpha \cup \{\alpha\})$

  (Levi Identity)

Belief contraction

So we attack the contraction problem first:

Given a belief set $K \subseteq L$, and a formula α,
Find $K \div \alpha$, the result of changing K such that
α is no longer a consequence

Partial meet contraction

One of the best-known approaches (Alchourrón, Gärdenfors & Makinson 1985)

3 steps to obtain $K \div \alpha$:

1. Focus on maximal subsets of K that don't entail α.
   Denote by $K \perp \alpha$.
2. Select "best" elements of $K \perp \alpha$ using selection function γ: $\gamma(K \perp \alpha)$
3. Form intersection: $K \div \alpha = \bigcap \gamma(K \perp \alpha)$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Partial meet contraction

- Write $K \div_\gamma \alpha$ to signify dependence on γ.

FORMAL DEFINITION OF $K \perp \alpha$:

$X \in K \perp \alpha$ iff (i) $X \subseteq K$, (ii) $\alpha \notin Cn(X)$, (iii) For any $Y \subseteq K$, if $X \subset Y$ then $\alpha \in Cn(Y)$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Partial meet contraction

FORMAL DEFINITION OF γ:

γ is a selection function for K iff, for all $\alpha \in L$,

(i) if $K \perp \alpha \neq \emptyset$ then $\emptyset \neq \gamma(K \perp \alpha) \subseteq K \perp \alpha$,

(ii) if $K \perp \alpha = \emptyset$ then $\gamma(K \perp \alpha) = \{K\}$.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Characterisation theorem for partial meet contraction

THEOREM (Alchourrón, Gärdenfors, Makinson 1985) $\div = \div_\gamma$ for some selection function γ iff $\div$ satisfies the following properties (known as the basic postulates for contraction):

- $K \div \alpha = Cn(K \div \alpha)$ (Closure)
- If $\alpha \notin Cn(\emptyset)$ then $\alpha \notin K \div \alpha$ (Success)
- $K \div \alpha \subseteq K$ (Inclusion)
- If $\alpha \notin K$ then $K \div \alpha = K$ (Vacuity)
- If $\alpha_1 \equiv \alpha_2$ then $K \div \alpha_1 = K \div \alpha_2$ (Extensionality)
- $K \subseteq Cn(K \div \alpha \cup \{\alpha\})$ (Recovery)

Partial meet revision

- We define p.m. revision via a p.m. contraction operator and the Levi Identity:

$$K *_\gamma \alpha = Cn\left(K \div_\gamma \neg\alpha \cup \{\alpha\}\right)$$

Characterisation theorem for partial meet revision

THEOREM (Alchourrón, Gärdenfors, Makinson 1985) $* = *_\gamma$ for some selection function γ iff $*$ satisfies the following properties (known as the basic postulates for revision):

- $K * \alpha = Cn(K * \alpha)$ (Closure)
- $\alpha \in K * \alpha$ (Success)
- $K * \alpha \subseteq Cn(K \cup \{\alpha\})$ (Inclusion)
- If $\neg\alpha \notin K$ then $Cn(K \cup \{\alpha\}) \subseteq K * \alpha$ (Vacuity)
- If $\alpha_1 \equiv \alpha_2$ then $K * \alpha_1 = K * \alpha_2$ (Extensionality)
- If α is consistent then so is $K * \alpha$ (Consistency)

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

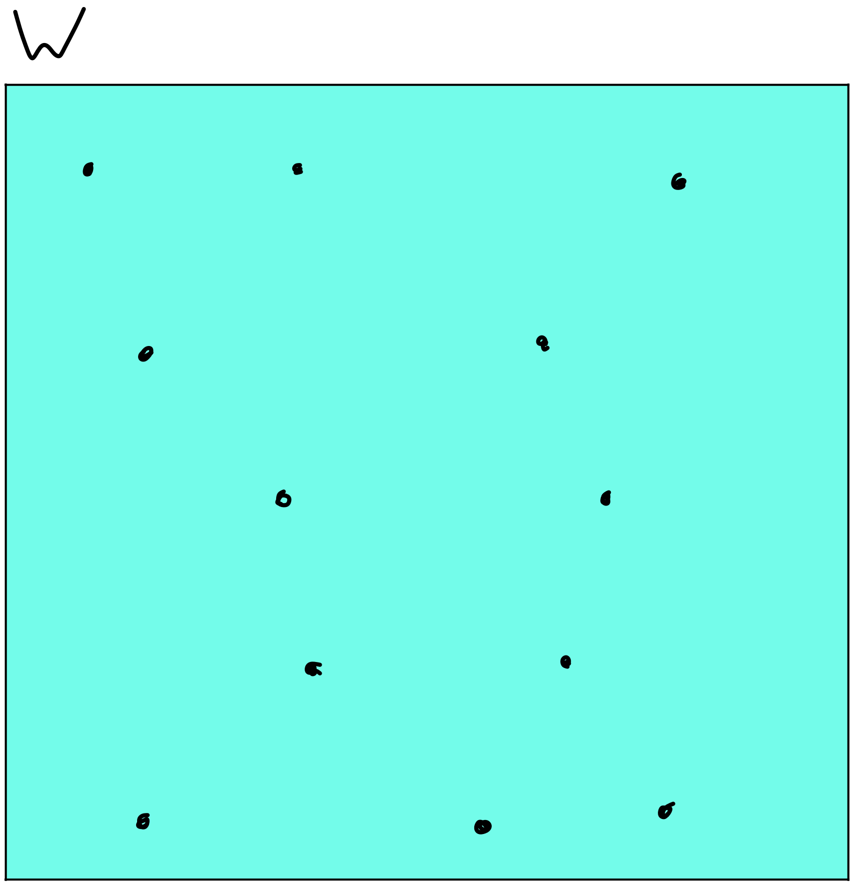
MIREL

Plausibility orderings

- W = set of all interpretations over set of prop. atoms
- Each interpretation assigns one of T, F to each p_i
- Think of them as possible worlds

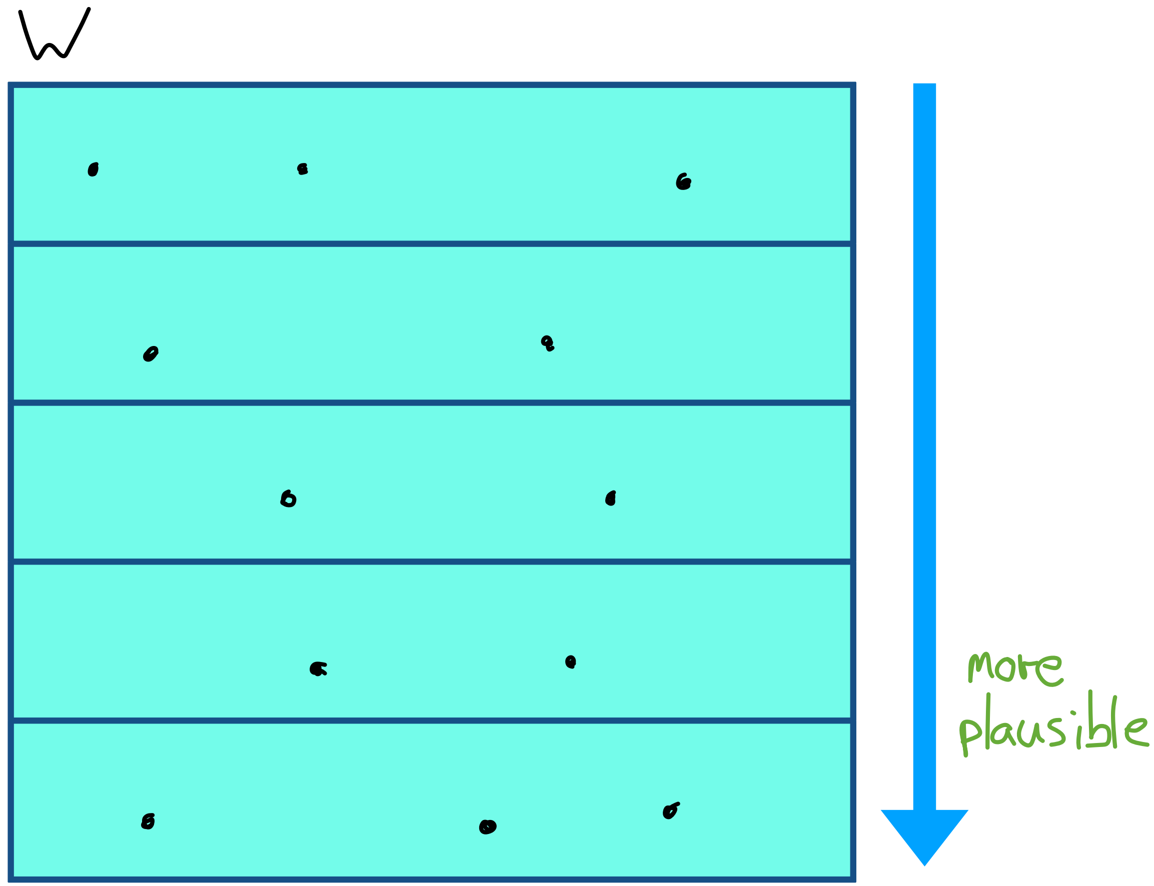
Plausibility orderings

One of these is the true state, but agent doesn't know which one.



Plausibility orderings

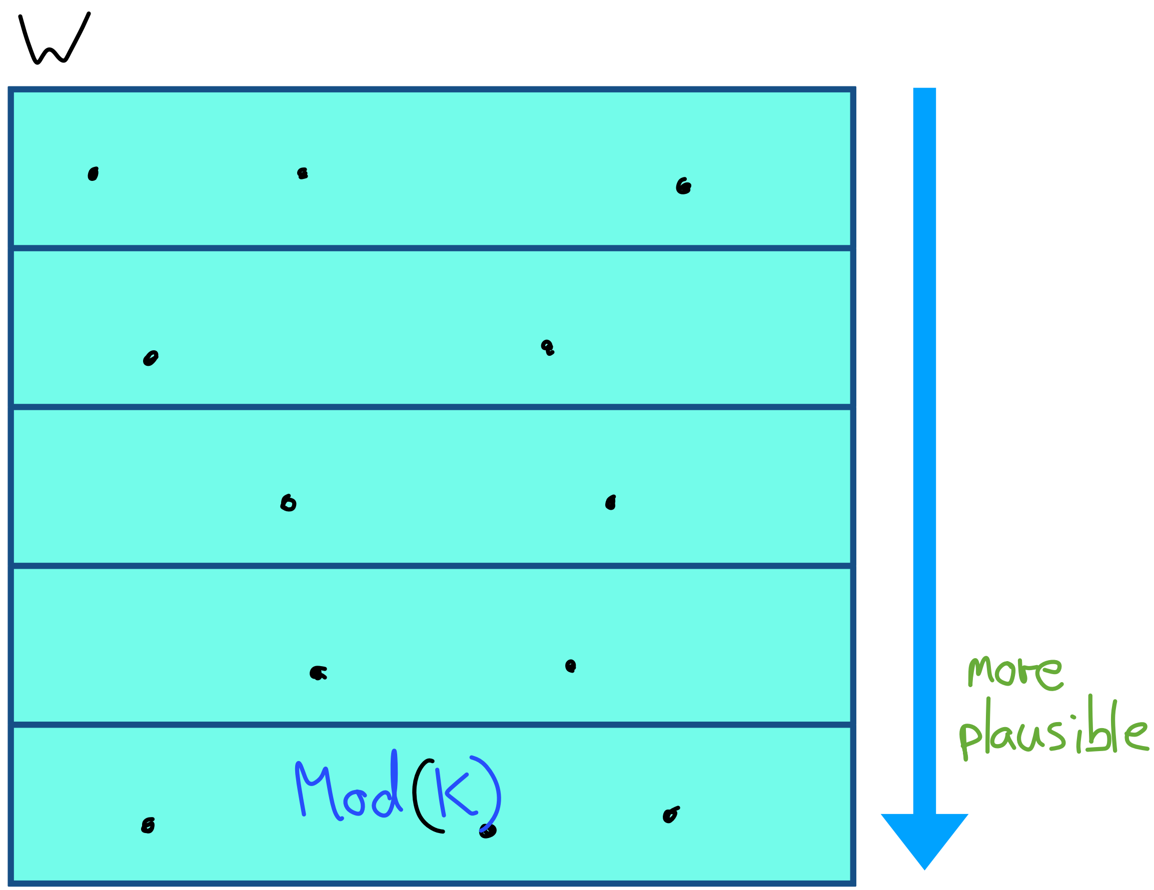
But agent thinks some worlds more plausible than others



Plausibility orderings

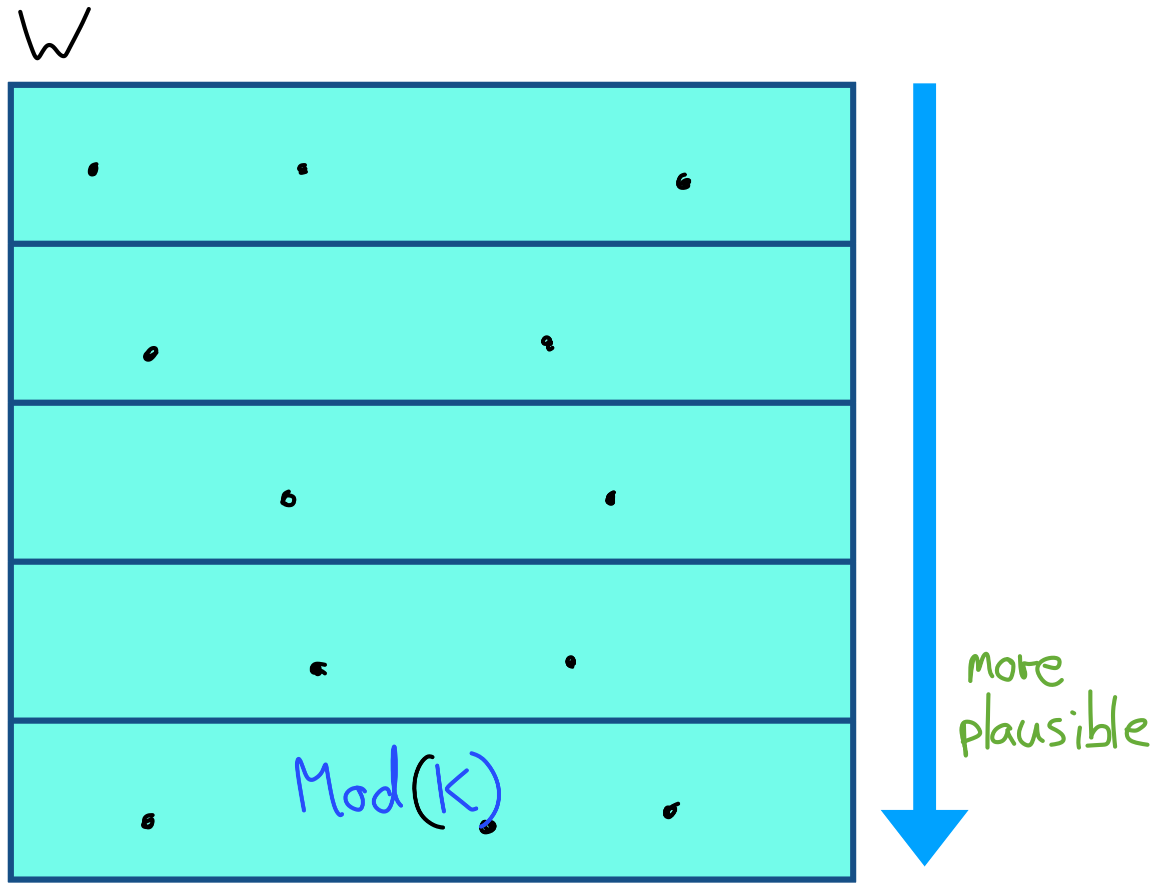
Most plausible are those satisfying K.

(We assume K consistent ($Mod(K) \neq \emptyset$) for simplicity)

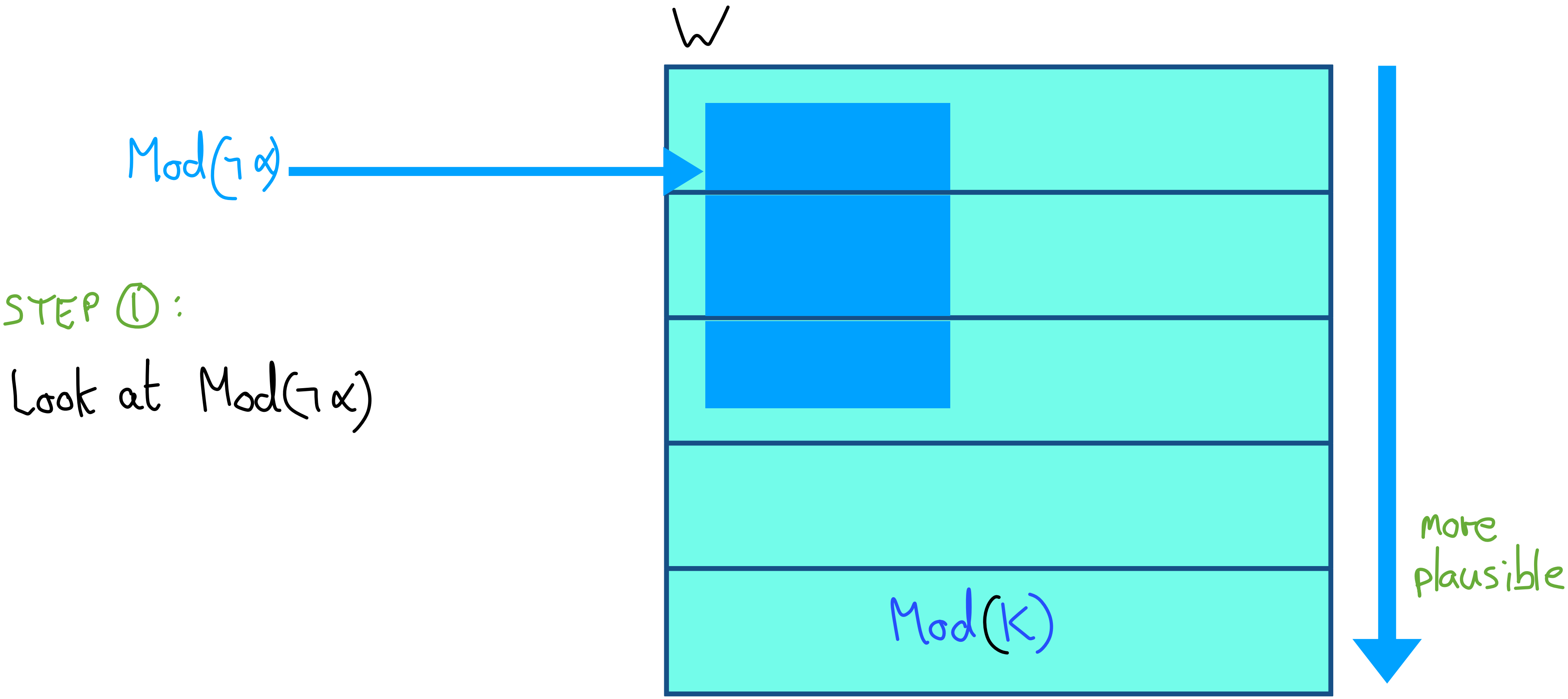


Plausibility orderings

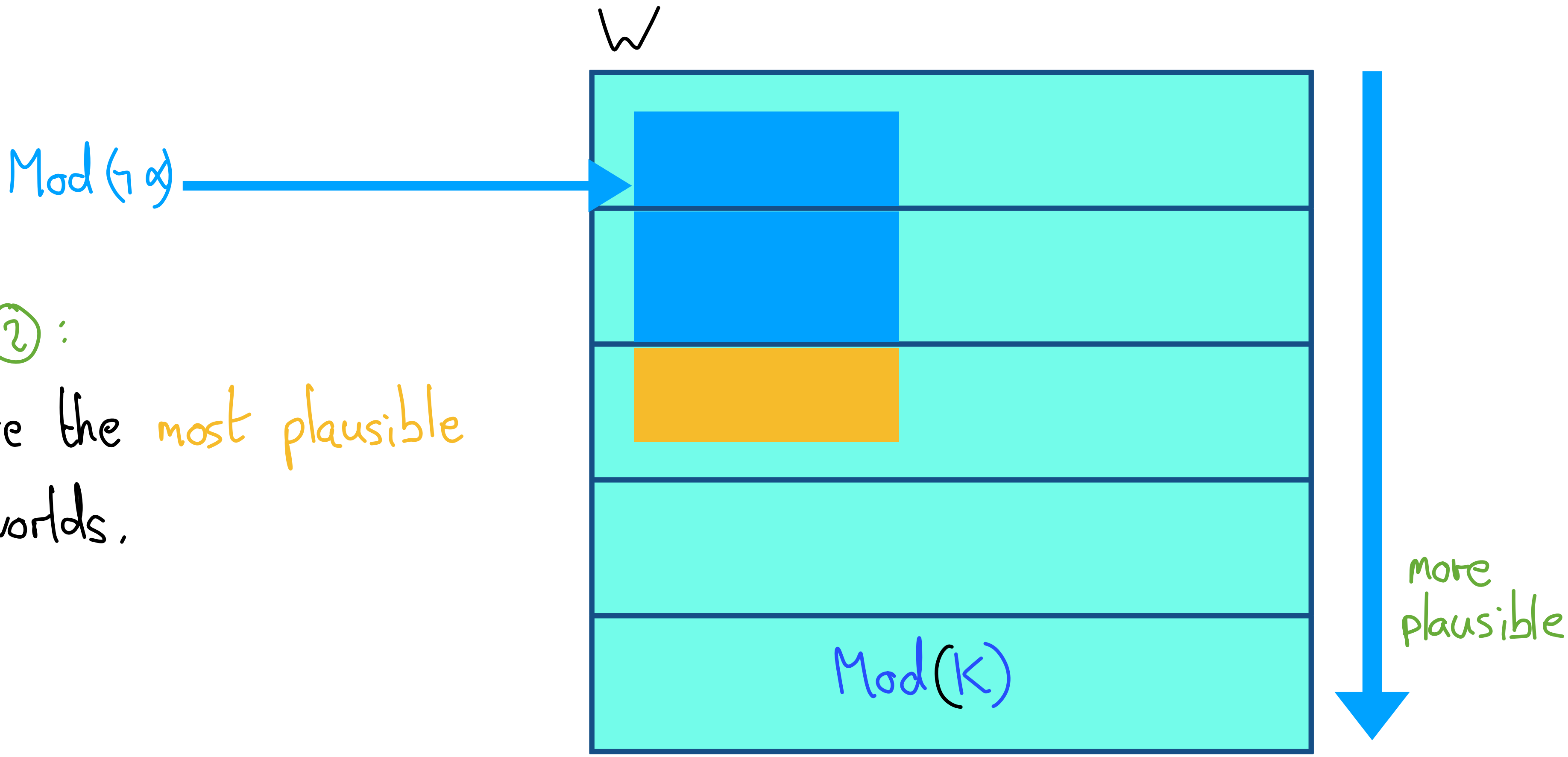
Agent uses this ordering as guide to perform belief change.



Contracting α



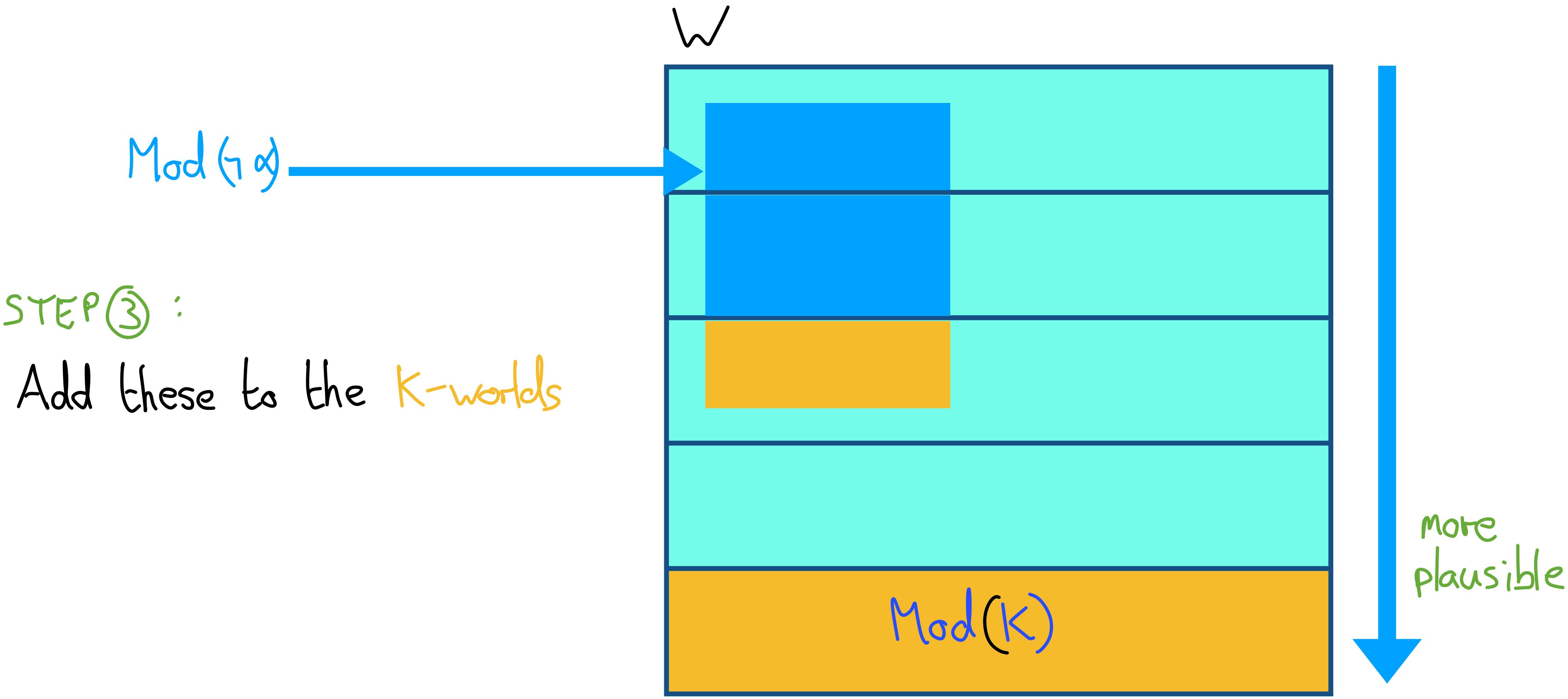
Contracting α



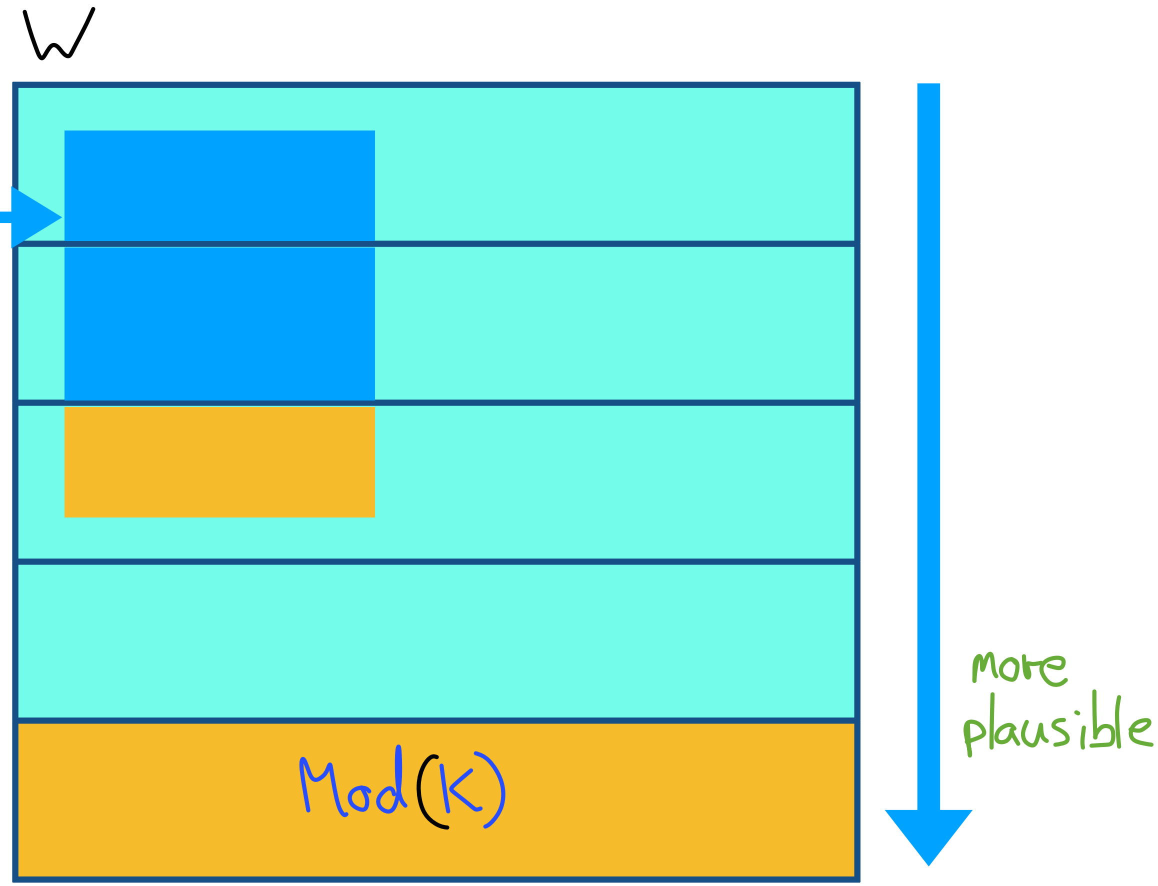
STEP ②:

Locate the most plausible ¬α-worlds.

Contracting α



Contracting α



FINAL STEP:

Take

$K \div \alpha = \{\beta \in L \mid \beta$ is true in all these worlds$\}$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Formal details

Assume a plausibility ordering $\leq$ over W, satisfying the following properties:

- $\leq$ is transitive $(v_1 \leq v_2,\ v_2 \leq v_3 \Rightarrow v_1 \leq v_3)$
- $\leq$ is complete (either $v_1 \leq v_2$ or $v_2 \leq v_1$, for all v_1, v_2)

  (any $\leq$ satisfying these 2 conditions is a total preorder)
- $Mod(K) = min(\leq, W)$

  (for any set X, $min(\leq, X) = \{v \in X \mid v \leq v' \ \forall v' \in X\}$)

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Formal details

From $\leq$, define contraction operator $\dot{-}_{\leq}$ for K by setting, for all $\alpha \in L$:

$$K \dot{-}_{\leq} \alpha = \begin{cases} \{\beta \in L \mid Mod(K) \cup \min(\leq, Mod(\neg\alpha)) \subseteq Mod(\beta)\} & \text{if } \alpha \notin Cn(\emptyset) \\ K & \text{if } \alpha \in Cn(\emptyset) \end{cases}$$

NOTE: $Mod(K \dot{-}_{\leq} \alpha) = Mod(K) \cup \min(\leq, Mod(\neg\alpha))$

(in case $\alpha \notin Cn(\emptyset)$)

Characterisation result

THEOREM (Grove, Katsuno & Mendelzon)

$\div = \div_{\leq}$ for some plausibility order $\leq$ iff $\div$ satisfies the basic postulates for contraction, PLUS:

- $(K \div \alpha) \cap (K \div \beta) \subseteq K \div (\alpha \wedge \beta)$ (Conjunctive Overlap)
- If $\alpha \notin K \div (\alpha \wedge \beta)$ then $K \div (\alpha \wedge \beta) \subseteq K \div \alpha$

(Conjunctive Inclusion)

Revising by α

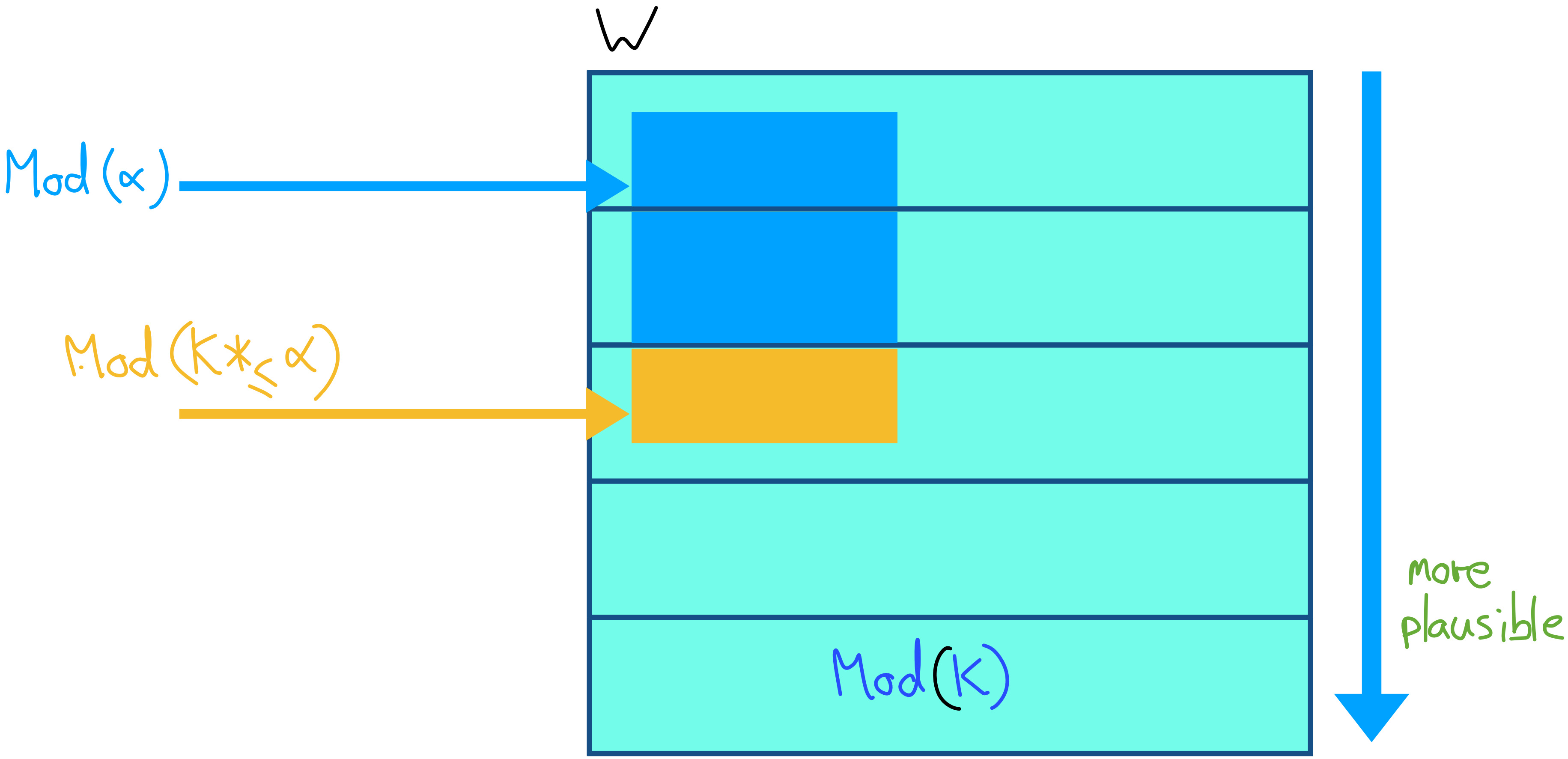
We can use Levi Identity to define $*_{\leq}$ from $\dot{-}_{\leq}$.

$$K *_{\leq} \alpha = Cn\left((K \dot{-}_{\leq} \neg\alpha) \cup \{\alpha\}\right)$$

Then:

$$K *_{\leq} \alpha = \begin{cases} \{\beta \in L \mid \min(\leq, Mod(\alpha)) \subseteq Mod(\beta)\} & \text{if } \neg\alpha \notin Cn(\emptyset) \\ L & \text{if } \neg\alpha \in Cn(\emptyset) \end{cases}$$

The picture



Characterisation result

THEOREM (Grove, Katsuno & Mendelzon)

$* = *_{\leq}$ for some plausibility order $\leq$ iff $*$ satisfies the basic postulates for revision, PLUS:

- $(K * \alpha) \cap (K * \beta) \subseteq K * (\alpha \vee \beta)$ (Disjunctive Overlap)
- If $\neg\alpha \notin K * \beta$ then $Cn(K * \beta \cup \{\alpha\}) \subseteq K * (\alpha \wedge \beta)$ (Subexpansion)

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

The Harper Identity

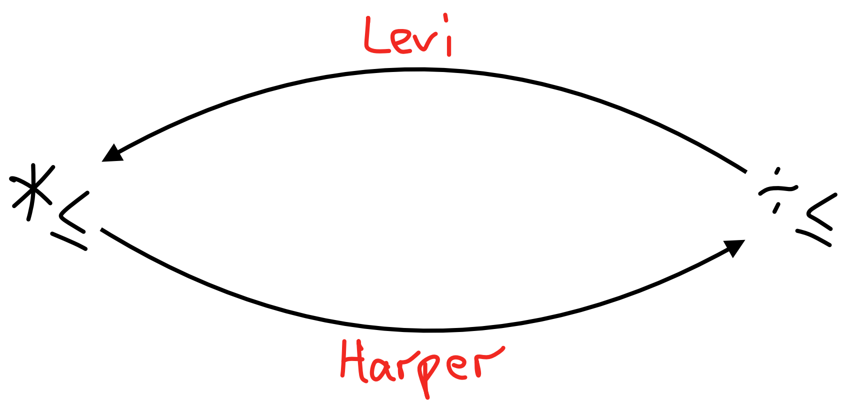
- The Levi Identity tells us how to define revision from contraction.
- What about going to contraction from revision?
- The Harper Identity:

$$K \dot{-} \alpha = (K * \neg\alpha) \cap K$$

The Harper Identity

- The Harper Identity does indeed hold for $*_{\leq}$, $\dot{-}_{\leq}$ defined from the same $\leq$.

- Harper and Levi are inverses of each other



Other approaches

- Belief revision in epistemic logic (Amsterdam)
  - Have explicit Belief modality $B\alpha$
  - Strange things happen to AGM "Success" postulate, e.g. revise by $p \wedge \neg B p$ (Moore sentence). Then $B(p \wedge \neg B p)$ cannot hold.
- Belief base revision (K need not be deductively closed)

Bibliography

- Richard Booth & Thomas Meyer, Belief change, in 'Logic and Philosophy Today, Vol. 1', College Publications, 2011.

Bibliography

Baader, F. and Hollunder, B. (1995).
Embedding defaults into terminological knowledge representation formalisms.
J. Autom. Reason., 14(1):149–180.

Bonatti, P. A. (2019).
Rational closure for all description logics.
Artif. Intell., 274:197–223.

Bonatti, P. A., Faella, M., Petrova, I. M., and Sauro, L. (2015).
A new semantics for overriding in description logics.
Artif. Intell., 222:1–48.

Bonatti, P. A., Lutz, C., and Wolter, F. (2009).
The complexity of circumscription in dls.
J. Artif. Intell. Res., 35:717–773.

Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., and Varzinczak, I. (2020).
Principles of klm-style defeasible description logics.
ACM Trans. Comput. Logic, 22(1).

Casini, G. and Straccia, U. (2010).
Rational closure for defeasible description logics.
In Janhunen, T. and Niemelä, I., editors, *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings*, volume 6341 of *Lecture Notes in Computer Science*, pages 77–90. Springer.

Casini, G. and Straccia, U. (2023).
Defeasible RDFS via rational closure.
Inf. Sci., 643:118409.

Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., and Tompits, H. (2008).
Combining answer set programming with description logics for the semantic web.
Artif. Intell., 172(12-13):1495–1539.

Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2015).
Semantic characterization of rational closure: From propositional logic to description logics.
Artif. Intell., 226:1–33.

Lehmann, D. (1995).
Another perspective on default reasoning.
Annals of Mathematics and Artificial Intelligence, 15:61–82.

Non-Classical Knowledge Representation and Reasoning

Italian National PhD Course on AI, 2024

Umberto Straccia & Giovanni Casini

CNR - ISTI, Pisa, Italy

`http://www.straccia.info`

{umberto.straccia, giovanni.casini}@isti.cnr.it

Recap

In the last lecture:

- Rational closure for Description Logic $\mathcal{ALC}$
- Rational closure for ρdf
- Belief Change - the AGM aopporoach:
  - Contraction
  - Revision

Partial meet contraction

One of the best-known approaches (Alchourrón, Gärdenfors & Makinson 1985)

3 steps to obtain $K \div \alpha$:

1. Focus on maximal subsets of K that don't entail α.
   Denote by $K \perp \alpha$.
2. Select "best" elements of $K \perp \alpha$ using selection function γ: $\gamma(K \perp \alpha)$
3. Form intersection: $K \div \alpha = \bigcap \gamma(K \perp \alpha)$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Characterisation theorem for partial meet contraction

THEOREM (Alchourrón, Gärdenfors, Makinson 1985) $\div = \div_\gamma$ for some selection function γ iff $\div$ satisfies the following properties (known as the basic postulates for contraction):

- $K \div \alpha = Cn(K \div \alpha)$ (Closure)
- If $\alpha \notin Cn(\emptyset)$ then $\alpha \notin K \div \alpha$ (Success)
- $K \div \alpha \subseteq K$ (Inclusion)
- If $\alpha \notin K$ then $K \div \alpha = K$ (Vacuity)
- If $\alpha_1 \equiv \alpha_2$ then $K \div \alpha_1 = K \div \alpha_2$ (Extensionality)
- $K \subseteq Cn(K \div \alpha \cup \{\alpha\})$ (Recovery)

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Partial meet revision

- We define p.m. revision via a p.m. contraction operator and the Levi Identity:

$$K *_\gamma \alpha = Cn(K \div_\gamma \neg\alpha \cup \{\alpha\})$$

Characterisation theorem for partial meet revision

THEOREM (Alchourrón, Gärdenfors, Makinson 1985) $* = *_\gamma$ for some selection function γ iff $*$ satisfies the following properties (known as the basic postulates for revision):

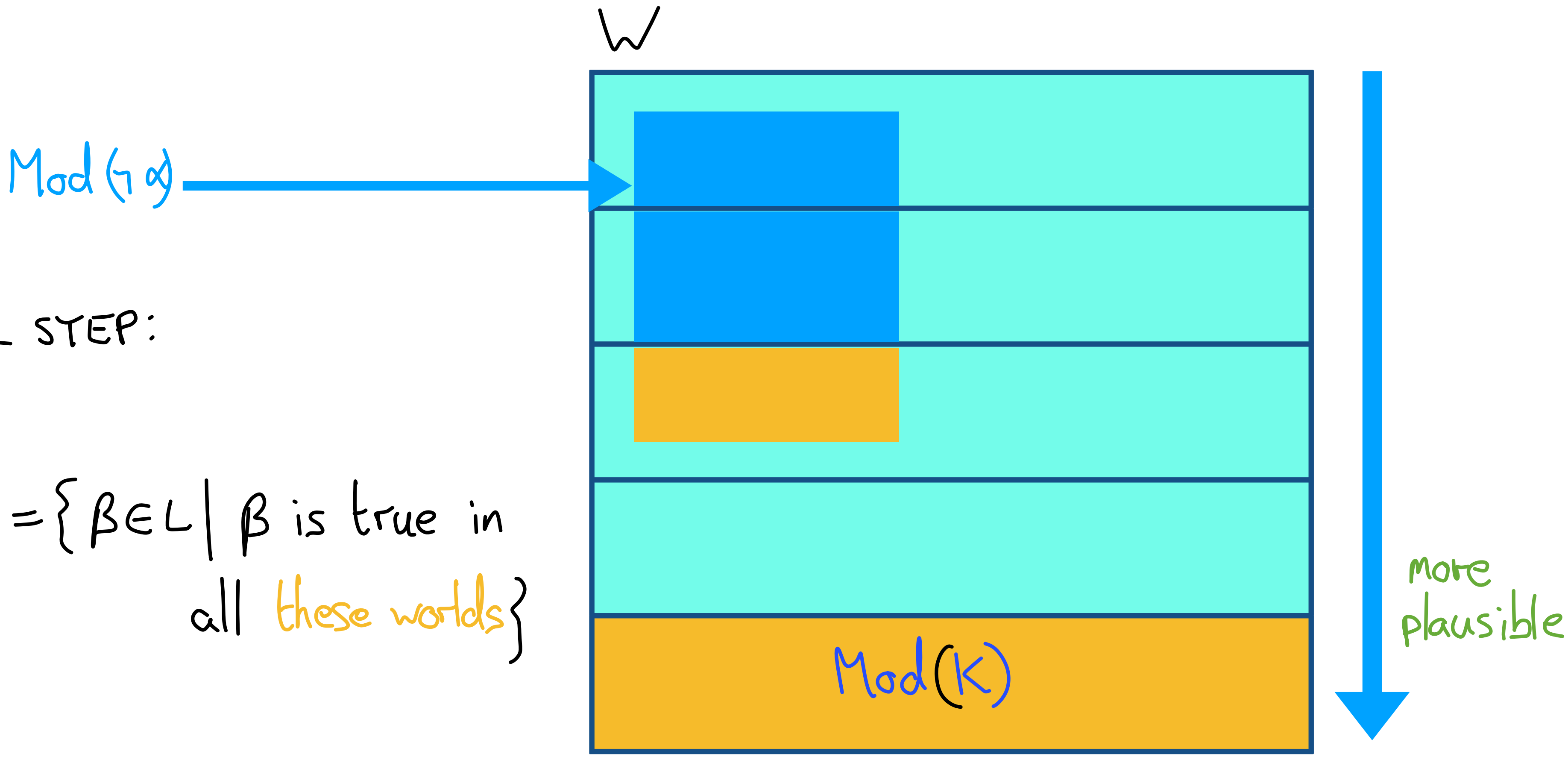
- $K * \alpha = Cn(K * \alpha)$ (Closure)
- $\alpha \in K * \alpha$ (Success)
- $K * \alpha \subseteq Cn(K \cup \{\alpha\})$ (Inclusion)
- If $\neg\alpha \notin K$ then $Cn(K \cup \{\alpha\}) \subseteq K * \alpha$ (Vacuity)
- If $\alpha_1 \equiv \alpha_2$ then $K * \alpha_1 = K * \alpha_2$ (Extensionality)
- If α is consistent then so is $K * \alpha$ (Consistency)

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Contracting α

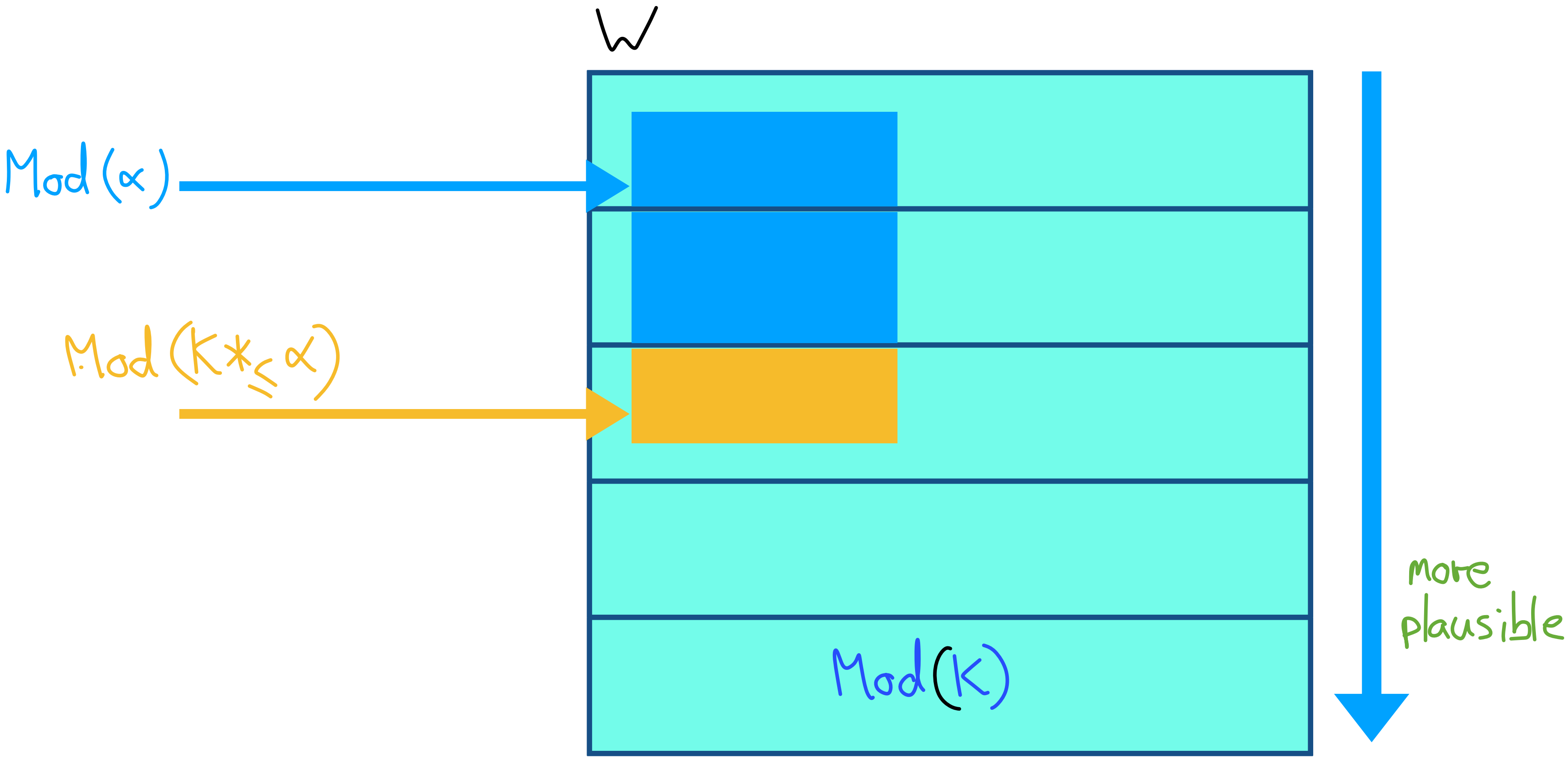


FINAL STEP:

Take

$K \dot{-} \alpha = \{ \beta \in L \mid \beta$ is true in all these worlds $\}$

The picture



BR and Non-monotonicity

Today's topic

In the AGM approach we assume that the underlying consequence operator satisfies some properties. In particular, it is assumed that it is Tarskian and Compact.

- Are these constraints essential for developing and AGM-style analysis?

- Today we consider dropping one of the property that a consequence operator needs to satisfy to be Tarskian: **Monotonicity**.

AGM Assumptions

AGM made some assumptions about the underlying logic $\langle L, Cn \rangle$:

- Language: closed under propositional operators.
- Consequence operator:

  1. Tarskian
     - Monotonicity: if $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$
     - Idempotence: $Cn(A) = Cn(Cn(A))$
     - Inclusion: $A \subseteq Cn(A)$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

AGM Assumptions

2. AGM Assumptions:

- Deduction: $\beta \in Cn(A \cup \{\alpha\})$ iff $(\alpha \to \beta) \in Cn(A)$
- Supraclassicality: if $\alpha \in Cl(A)$ then $\alpha \in Cn(A)$
- Compactness: if $\alpha \in Cn(A)$, then $\alpha \in Cn(A')$ for some finite $A' \subseteq A$
- Disjunction in the premises: $\dfrac{\gamma \in Cn(A \cup \{\alpha\}) \qquad \gamma \in Cn(A \cup \{\beta\})}{\gamma \in Cn(A \cup \{\alpha \vee \beta\})}$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD
uni.lu UNIVERSITÉ DU LUXEMBOURG
MIREL

AGM Assumptions

We may need an analysis of belief change for logics that do not satisfy the above prerequisites.

➡ What happens if we drop some of them?
Can we still develop an AGM-style analysis of belief change?

Today we consider dropping one of the Tarskian properties:

Monotonicity

Non-monotonicity

Recent work in revision of non-monotonic theories:

➡ Answer Set Programming

➡ Conditional Reasoning

Today we will focus on conditional reasoning:

➡ Its relation with belief revision

➡ Known issues

➡ Recent characterisations of BR for conditional reasoning

And at the end also have a look at ASP.

Non-monotonicity

Example

Remember the KB we saw in the first lecture?

- *Sweden is a part of Europe*
- *All European swans are white*
- *The bird caught in the trap is a swan*
- *The bird caught in the trap is from Sweden*

From this we can derive

- *The bird caught in the trap is white*

If we are informed that *the caught bird is a black swan*, we need to revise our KB in order to preserve consistency.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Non-monotonicity

Example

Consider the following slightly modified KB:

- *Sweden is a part of Europe*
- *Typically, European swans are white*
- *The bird caught in the trap is a swan*
- *The bird caught in the trap is from Sweden*

From such a KB we can conclude that

- *Presumably, the bird caught in the trap is white*

Such a conclusion is just tentative.

We are informed that the swan is black

↓

We drop the presumptive conclusion, we do not need to make changes to the KB, since it admits exceptions to the second statement.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Non-monotonicity

There is a connection between Belief Revision and Non-monotonic Reasoning

- Both are aimed at managing potential conflicts among pieces of information
  - Non-monotonic reasoning can manage conflicting information
  - Still, it is possible to have inconsistencies also in non-monotonic KBs

Example

Consider again the KB:

- *Sweden is a part of Europe*
- *Typically, European swans are white*
- *The bird caught in the trap is a swan*
- *The bird caught in the trap is from Sweden*

We are informed that

- *Typically, European swans are blue*

This is a conflict that is problematic also for non-monotonic systems.

Non-monotonicity

➡ Assume we are facing conflicting pieces of information: should such a conflict be managed by the non-monotonic machinery or by some belief change operator?

➡How to characterise such belief change operators for non-monotonic reasoning?

Conditional Reasoning

For non-monotonic conditionals, the following does not hold:

Monotonicity

$$\frac{\alpha \Rightarrow \beta}{\alpha \wedge \gamma \Rightarrow \beta}$$

Conditionals like $bird \Rightarrow fly$ and $penguin \wedge bird \Rightarrow \neg fly$ can coexist consistently.

Note:
non-monotonicity conditional $\neq$ non-monotonicity entailment operator

Monotonic entailment operator Cn:

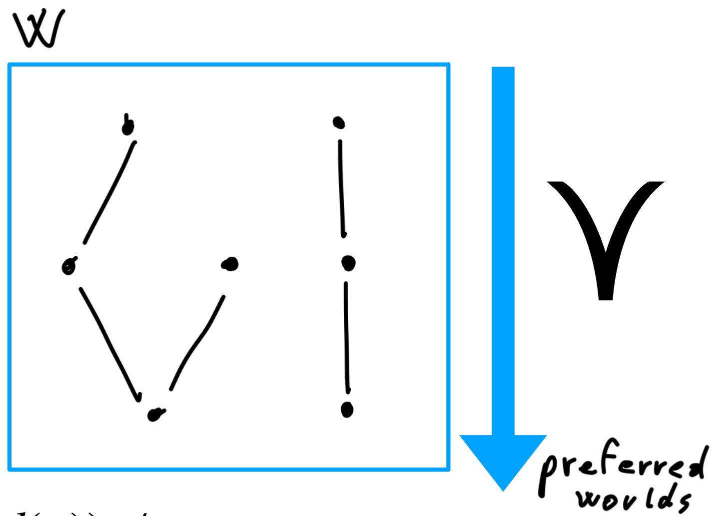
➡ If $\alpha \Rightarrow \beta \in Cn(K)$ then $\alpha \Rightarrow \beta \in Cn(K \cup \{\gamma \Rightarrow \delta\})$

It is compatible with non-monotonic conditionals.

Conditional Reasoning

A popular semantics for non-monotonic conditionals $\alpha \Rightarrow \beta$:preferential semantics.

Interpretations $M = (W, \prec)$:



- W is a (multi)set of possible worlds (propositional valuations)
- $\prec$ is a preference relation defined over W :
    - *transitive*, *asymmetric*, and *smooth*

Smoothness: for every α, if $Mod(\alpha) \neq \emptyset$ then $\min_{\prec}(Mod(\alpha)) \neq \emptyset$, where

$$\min_{\prec}(Mod(\alpha)) = \{w \in Mod(\alpha) \mid \nexists v \in w \text{ s.t. } v \in Mod(\alpha) \text{ and } v \prec w\}$$

$w \prec v$ is read as 'the situation described by w is preferred to the situation described by v'

Conditional Reasoning

A conditional $\alpha \Rightarrow \beta$ is satisfied by an interpretation $M = (W, \prec)$ ($M \Vdash \alpha \Rightarrow \beta$) if the preferred worlds satisfying α satisfy also β. That is,

$$\min_{\prec}(Mod(\alpha)) \subseteq Mod(\beta)$$

Depending on the interpretation we give to the relation $\prec$, the conditionals $\alpha \Rightarrow \beta$ have been interpreted in various ways. For example:

- Expectations: “Typically, if α then β“.
- Obligations: “If α , then it ought to be β”.
- Counterfactual/subjunctive conditionals: “If α were the case, then β would have been the case too”.

Conditional Reasoning

Theorem [Kraus et Al. (1990)]
A conditional entailment operator $Cn(.)$ is closed under the preferential properties iff, for every conditional KB K, $Cn(K)$ can be defined using a preferential model. That is,

$$Cn(K) = \{\alpha \Rightarrow \beta \mid M \Vdash \alpha \Rightarrow \beta\}$$

for some preferential model M.

Special case - **Preferential Closure $Pr(.)$**:

$$Pr(K) = \{\alpha \Rightarrow \beta \mid M \Vdash \alpha \Rightarrow \beta \text{ for all } M \text{ s.t. } M \Vdash K\}$$

$Pr(K)$ is the smallest preferential closure containing K.

$\alpha \Rightarrow \beta \in Pr(K)$ iff $\alpha \Rightarrow \beta$ is derivable from K using the preferential properties

$Pr(.)$ is **Tarskian**!
(and hence **monotonic**)

Conditional Reasoning

Let's consider another property:

$$\text{(RM)} \qquad \frac{\alpha \Rightarrow \beta \qquad \alpha \not\Rightarrow \neg\gamma}{\alpha \wedge \gamma \Rightarrow \beta} \qquad \text{Rational Monotony}$$

necessary for the satisfaction of important reasoning patterns, as

Presumption of typicality:
Given the information at our disposal, we assume we are in the most expected situation.

$$\frac{bird \Rightarrow fly \qquad bird \not\Rightarrow \neg sparrow}{bird \wedge sparrow \Rightarrow fly}$$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Conditional Reasoning

The entailment operators aimed at modelling some kind of presumptive reasoning are usually non-monotonic (and satisfy (RM)).

- I know that typically birds fly ($bird \Rightarrow fly$)
- I hear about some 'Dodo' bird, but I know nothing about it. So, I am not aware whether it is an atypical bird ($bird \not\Rightarrow \neg dodo$)
- With this information, I presume that dodos behave like normal birds ($bird \wedge dodo \Rightarrow fly$)
- Later I am informed that dodos are extinct, and that actually they were very strange birds. Not really a typical bird ($bird \Rightarrow \neg dodo$)
- With this new piece of information, I can to drop the previous conclusion, still satisfying (RM) ($bird \wedge dodo \not\Rightarrow fly$)

$$dodo \wedge bird \Rightarrow fly \in Cn(\{bird \Rightarrow fly\})$$

$$dodo \wedge bird \Rightarrow fly \notin Cn(\{bird \Rightarrow fly, bird \Rightarrow \neg dodo\})$$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

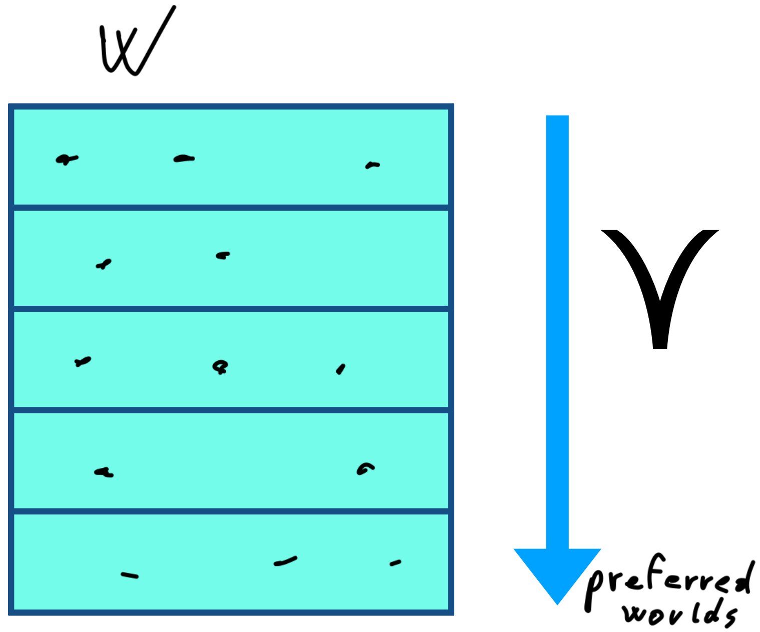
MIREL

Conditional Reasoning

Ranked interpretation $R = (W, \prec)$:

R is a preferential interpretation and $\prec$ satisfies modularity:

If $x \prec y$ then either $z \prec y$ or $x \prec z$



Theorem [Lehmann and Magidor (1992)]
A conditional entailment operator $Cn(.)$ is closed under the preferential properties + (RM) iff, for every conditional KB K, $Cn(K)$ can be defined using a ranked model. That is,

$$Cn(K) = \{\alpha \Rightarrow \beta \mid M \Vdash \alpha \Rightarrow \beta\}$$

for some ranked model R.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

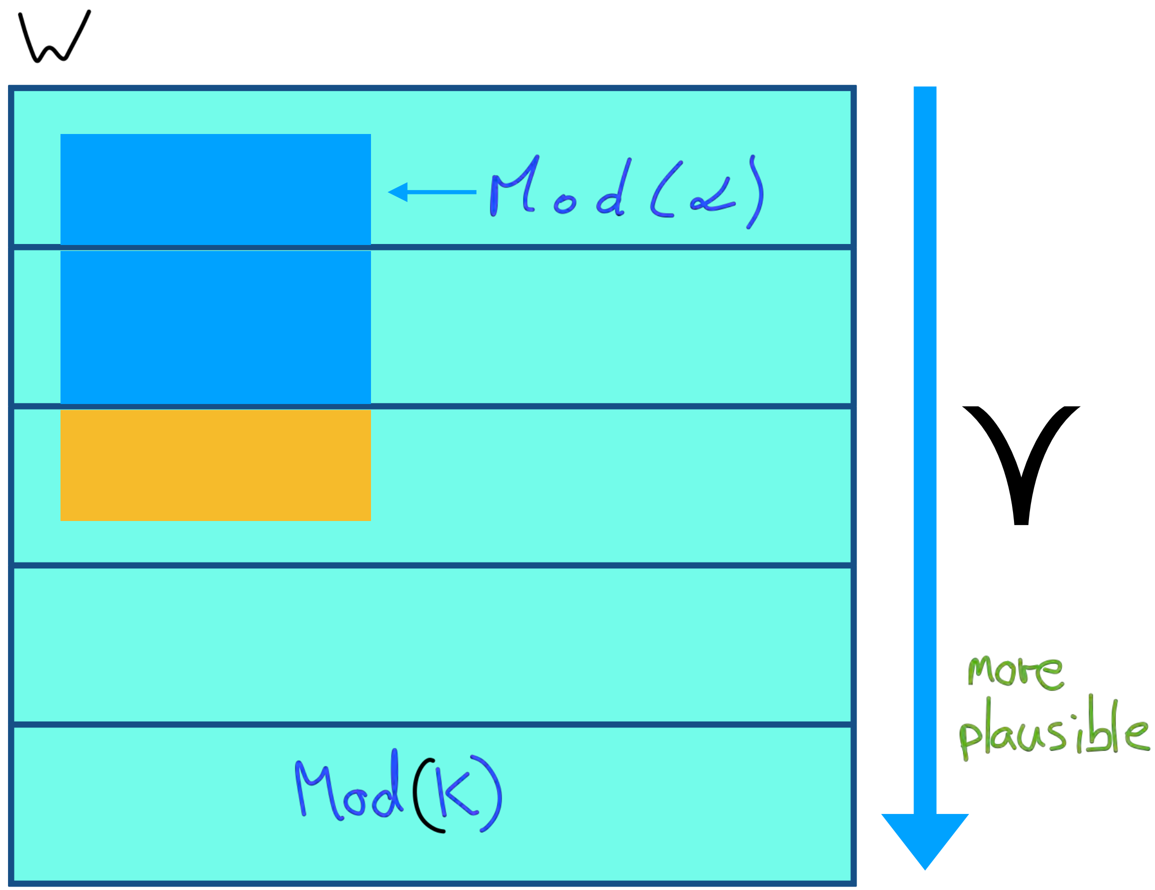
MIREL

Two sides of the same coin

Remember the semantic characterisation of AGM revision?

It was built using a specific class of preferential models, the ones in which $\prec$ is modular.

The worlds in the yellow part define $K \star \alpha$



Two sides of the same coin

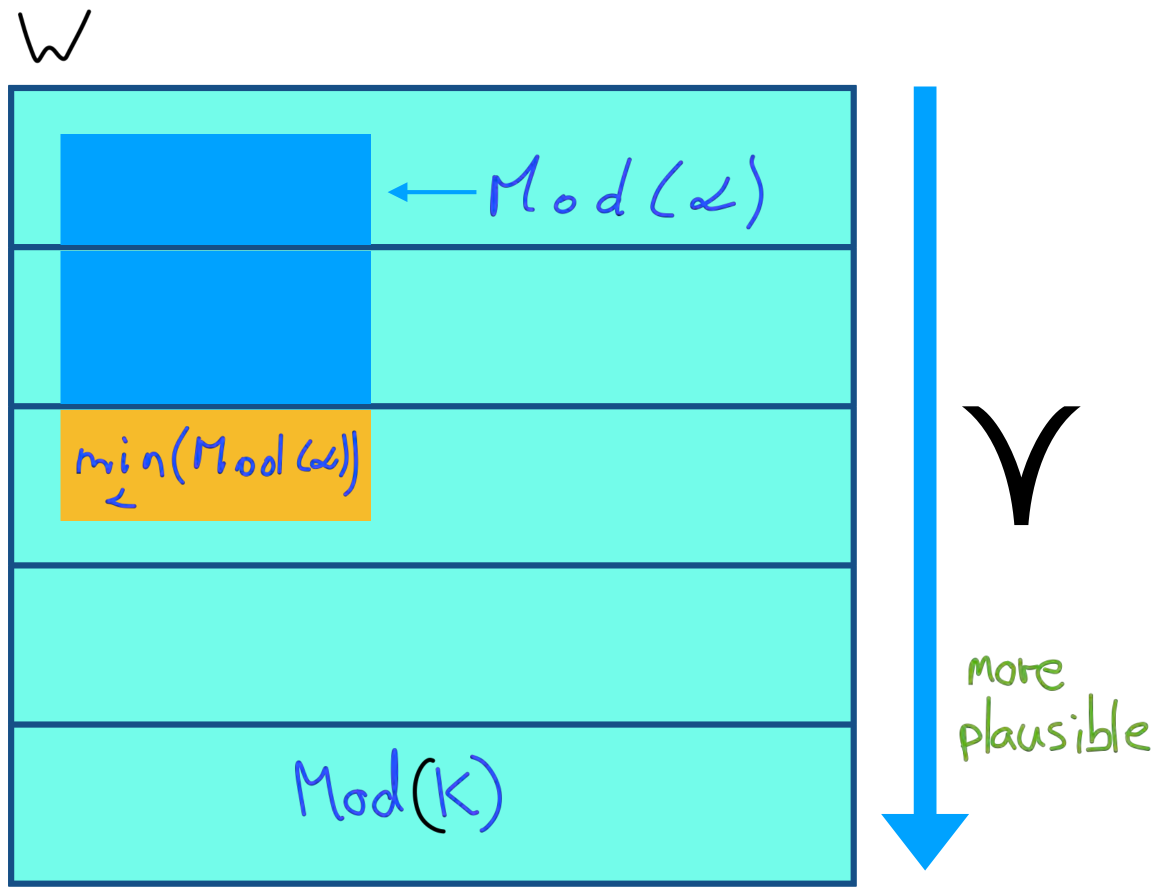
The yellow part corresponds to

$\min_{\prec}(Mod(\alpha))$

So, there is a strong correspondence:

in the same ranked model, $\alpha \Rightarrow \beta$ holds iff

$\beta \in K \star \alpha$



Two sides of the same coin

There is the possibility of representing revision policies via subjunctive conditionals:

$$\beta \in K \star \alpha \textbf{ iff } \alpha \Rightarrow \beta$$

That is, the revision policies can be represented via conditionals interpreted as "if α were the case, then β would hold".

Question: Can we extend the revision operators to a language containing the correspondent conditionals?

Implementing such a step would allow also to revise the revision policies.

Two sides of the same coin

Answer: **No!**

Or at least not easily.

Gärdenfors' Impossibility result [Gärdenfors (1988)],
here in Rott's version [Rott (1989)]:

Assume a logic $\langle L, Cn \rangle$ where:

- Language L: Propositional language + conditionals $\alpha \Rightarrow \beta$
- Operator $Cn(.)$: operator for L s.t. it corresponds to propositional logic for the propositional fragment (and it is monotonic over the entire L)

Belief Revision Model $\langle \mathbf{K}, \star \rangle$: $\mathbf{K}$ is a set of Cn-theories (closed under propositional expansion) and $\star$ is a revision operation on the theories K in $\mathbf{K}$ satisfying :

$(\star\ 2)$ If $\alpha \notin Cn(\varnothing)$, then $\alpha \in K \star \alpha$ *(Success)*

$(\star\ 4)$ If $\neg\alpha \notin K$, then $Cn(K \cup \{\alpha\}) \subseteq K \star \alpha$ *(Vacuity)*

$(\star\ 6)$ If $\neg\alpha \notin Cn(\varnothing)$, then $\bot \notin K \star \alpha$ *(Consistency)*

(GRT) $\beta \in K \star \alpha$ iff $\alpha \Rightarrow \beta \in K$ *(Ramsey Test)*

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Two sides of the same coin

Let B be a theory in $\mathbf{K}$, and let α, β be two *contingent* propositions, that is, s.t.

$$\alpha \notin Cn(\emptyset); \neg\alpha \notin Cn(\emptyset); \beta \notin Cn(\emptyset); \neg\beta \notin Cn(\emptyset)$$

and such that $\{\alpha \vee \beta, \neg\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \vee \neg\beta\} \cap B = \emptyset$

It turns out that $\perp \in Cn(B \cup \{\alpha\}) \star \neg\alpha$

In contradiction with

$(\star\ 6)$ If $\neg\alpha \notin Cn(\emptyset)$, then $\perp \notin K \star \alpha$ $(Consistency)$

A lot has been discussed about the implications of Gardenfors' result, and its hidden assumptions.

There have been some interesting proposals about the revision of conditionals respecting the Ramsey test, but avoiding the impossibility result.

All the discussion was focused on the Ramsey test and the subjunctive interpretation of conditionals

BR of non-monotonic KBs

Does Gardenfors' result prevent the application of an AGM approach to non-monotonic preferential reasoning?

Preferential conditionals have also other interpretations beyond the subjunctive one. Belief change is interesting also in a non-monotonic conditional framework.

Example

We have a knowledge base B containing the following information:

- vertebrate red blood cells have a nucleus ($v \rightarrow n$);
- avian red blood cells are vertebrate red blood cells ($a \rightarrow v$);
- mammalian red blood cells are vertebrate red blood cells ($m \rightarrow v$);
- mammalian red blood cells don't have a nucleus ($m \rightarrow \neg n$).

We must conclude that mammalian red blood cells do no exist ($m \rightarrow \bot$).

BR of non-monotonic KBs

Case 1.

We know that mammalian red blood cells exist, and we want to enforce such information ($m \to \bot$ should be contracted).

- In classical monotonic belief change: the contraction of $m \to \bot$ results into the elimination of some piece of information, for example $v \to n$;
- It could be preferable to weaken $v \to n$ into its defeasible version $v \Rightarrow n$ (vertebrate red blood cells usually have nucleus).
- The non-monotonic inference machinery will take care of treating m as an exceptional subclass of v.

$$B' = \{v \Rightarrow n, a \to v, m \to v, m \to \neg n\}$$

BR of non-monotonic KBs

Case 2.

Assuming our non-monotonic machinery is well-behaved,

- from B' we can conclude that avian red blood cells presumably have a nucleus ($a \Rightarrow n$)

$$B' = \{v \Rightarrow n, a \rightarrow v, m \rightarrow v, m \rightarrow \neg n\}$$

- But we are informed that avian red blood cells usually do not have a nucleus ($a \Rightarrow \neg n$). Since $a \Rightarrow n$ was a presumptive conclusion, the non-monotonic entailment relation should take care of eliminating such a conclusion once faced with conflicting evidence ($a \Rightarrow \neg n$).

- The introduction of $a \Rightarrow \neg n$ should correspond to a simple addition:

$$B'' = \{v \Rightarrow n, a \Rightarrow \neg n, a \rightarrow v, m \rightarrow v, m \rightarrow \neg n\}$$

BR of non-monotonic KBs

Case 3.

$$B'' = \{v \Rightarrow n, a \Rightarrow \neg n, a \rightarrow v, m \rightarrow v, m \rightarrow \neg n\}$$

We are then informed that $a \Rightarrow n$.

But, even in the most trivial non-monotonic reasoning, $B'' \vDash a \Rightarrow \neg n$

We have now a choice:

- If we are interested only in preserving ***logical consistency*** (avoid $\top \Rightarrow \bot$), then we can simply add $a \Rightarrow n$ to B'', and conclude $a \Rightarrow \bot$.
- If we want to preserve ***coherence***, then we have to ``readjust'' the KB to avoid $a \Rightarrow \bot$.

As it is intended in the field of logic-based ontologies, a KB is ***coherent*** if every class (i.e., atomic proposition) a that has been introduced in the language can in principle be populated (we cannot conclude $a \Rightarrow \bot$).

BR of non-monotonic KBs

There are two critical points in modelling belief change for conditional non-monotonic reasoning:

1. Revising, are we interested in preserving consistency or coherence?

- We want to add to our KB a conditional $\alpha \Rightarrow \beta$. Do we consider a potential conflict if the addition of $\alpha \Rightarrow \beta$ enforces the derivation of $\top \Rightarrow \bot$ (logical inconsistency) or it is sufficient the derivation of $\alpha \Rightarrow \bot$ (incoherence)?

This is a contextual issue, associated to the domain we are modelling.

BR of non-monotonic KBs

2. Management of the potential conflicts.

- We have a non-monotonic consequence operator Cn and a conditional base K. Let $\alpha \Rightarrow \neg\beta \in Cn(K)$.
  We receive the information $\alpha \Rightarrow \beta$, that is in conflict with $Cn(K)$.
  We need to know whether $\alpha \Rightarrow \neg\beta$ is a necessary or a defeasible consequence of K.
  In the former case, we have a conflict and we need to revise the base (Case 3 of the example), in the latter there is no need of actual revision, since the non-monotonic machinery will eliminate the conflict (Case 2 of the example).

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

BR for preferential conditionals

This second point is a formal question: given a non-monotonic closure Cn and a conditional base K, which conditionals in $Cn(K)$ are a necessary consequence of Cn. Monotonicity gives us the answer.

A closure operator Cl is called the *monotonic core* of a non-monotonic closure Cn if, for every conditional base B,B',

(i) $B \subseteq B'$ implies $Cl(B) \subseteq Cl(B')$;

(ii) $Cl(B) \subseteq Cn(B)$;

(iii) for every closure operator Cl' satisfying (i) and (ii), $Cl'(B) \subseteq Cl(B)$.

Given a non-monotonic entailment relation, the existence of a monotonic core needs to be proved.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

BR for preferential conditionals

We consider the class of **supra-preferential cumulative operators** Cn:

- Supra-preferential:
  - ➡ Cn is closed under the preferential properties
  - ➡ If a set of conditionals K has a model ($\top \Rightarrow \bot \notin Pr(K)$), then $\top \Rightarrow \bot \notin Cn(K)$ (consistency preservation)
- Cumulative:
  - ➡ **if** $B \subseteq B' \subseteq Cn(B)$**, then** $Cn(B') = Cn(B)$

This covers an ample class of non-monotonic operator Cn that are definable using preferential semantics.

BR for preferential conditionals

Proposition [Casini & Meyer (2017)]

Given a supra-preferential closure operator Cn, its monotonic core is the preferential closure Pr.

Characterising belief revision for supra-preferential operators:

1. Model it for the monotonic core (preferential closure Pr).

2. Then model it for the non-monotonic operator Cn.

Contraction and preferential closure

Translation of the basic AGM contraction postulates in the conditional framework:

$(\div 1)$ $K \div \alpha = Cn(K \div \alpha)$

$(\div 2)$ $K \div \alpha \subseteq K$

$(\div 3)$ If $\alpha \notin Cn(K)$, then $K \div \alpha = K$

$(\div 4)$ If $\alpha \notin Cn(\emptyset)$, then $\alpha \notin K \div \alpha$

$(\div 5)$ If $\alpha \leftrightarrow \beta \in Cn(\emptyset)$, then $K \div \alpha = K \div \beta$

$(\div 6)$ $K \subseteq Cn((K \div \alpha) \cup \{\alpha\})$

(-1) $K^{-}_{\alpha \Rightarrow \beta} = Pr(K^{-}_{\alpha \Rightarrow \beta})$

(-2) $K^{-}_{\alpha \Rightarrow \beta} \subseteq K$

(-3) If $\alpha \Rightarrow \beta \notin Pr(K)$, then $K^{-}_{\alpha \Rightarrow \beta} = K$

(-4) If $\alpha \Rightarrow \beta \notin Pr(\emptyset)$, then $\alpha \Rightarrow \beta \notin K^{-}_{\alpha \Rightarrow \beta}$

(-5) If $\alpha \Rightarrow \beta \equiv_{Pr} \alpha' \Rightarrow \beta'$, then $K^{-}_{\alpha \Rightarrow \beta} = K^{-}_{\alpha' \Rightarrow \beta'}$

(-6) $K \subseteq Pr(K^{-}_{\alpha \Rightarrow \beta} \cup \{\alpha \Rightarrow \beta\})$

Contraction and preferential closure

Remember the partial meet contraction in AGM belief revision?

The remainder set $K \perp \alpha$ contains all the maximal subtheories of K that do not contain α

$$K \div \alpha = \bigcap \gamma(K \perp \alpha)$$

Working with preferential theories K, we can define the equivalent notion in the conditional framework:

- $K \perp (\alpha \Rightarrow \beta)$ is the set of the maximal (preferential) subtheories of K that do not contain $\alpha \Rightarrow \beta$
- - is a partial meet contraction operator if it can be defined as $K^{-}_{\alpha \Rightarrow \beta} = \bigcap \gamma(K \perp (\alpha \Rightarrow \beta))$

  where γ behaves as in the propositional case:

  ✦ If $K \perp (\alpha \Rightarrow \beta) \neq \varnothing$, then $\varnothing \neq \gamma(K \perp (\alpha \Rightarrow \beta)) \subseteq K \perp (\alpha \Rightarrow \beta)$

  ✦ If $K \perp (\alpha \Rightarrow \beta) = \varnothing$, then $\gamma(K \perp (\alpha \Rightarrow \beta)) = \{K\}$

Contraction and preferential closure

Monotonic Core Pr.
Postulates for Contraction $-$

The postulates for contraction are as follows (where $\equiv_{\mathrm{Pr}}$ refers to preferential equivalence):

- (-1) $K^{-}_{\alpha\Rightarrow\beta} = Pr(K^{-}_{\alpha\Rightarrow\beta})$ $(-\ closure)$
- (-2) $K^{-}_{\alpha\Rightarrow\beta} \subseteq K$ $(-\ inclusion)$
- (-3) If $\alpha \Rightarrow \beta \notin Pr(K)$, then $K^{-}_{\alpha\Rightarrow\beta} = K$ $(-\ vacuity)$
- (-4) If $\alpha \Rightarrow \beta \notin Pr(\emptyset)$, then $\alpha \Rightarrow \beta \notin K^{-}_{\alpha\Rightarrow\beta}$ $(-\ success)$
- (-5) If $\alpha \Rightarrow \beta \equiv_{Pr} \alpha' \Rightarrow \beta'$, then $K^{-}_{\alpha\Rightarrow\beta} = K^{-}_{\alpha'\Rightarrow\beta'}$ $(-\ extensionality)$
- (-6) $K \subseteq Pr(K^{-}_{\alpha\Rightarrow\beta} \cup \{\alpha \Rightarrow \beta\})$ $(-\ recovery)$

Theorem [Casini & Meyer (2017)]

A contraction operator $-$ for preferential entailment Pr satisfies (-1) – (-6) iff it is a partial meet contraction operator.

Preferential revision

Monotonic Core *Pr*. Postulates for Revision • (consistency preservation)

The postulates for revision for consistency preservation are as follows:

- $(\bullet\ 1)$ $K^{\bullet}_{\alpha\Rightarrow\beta} = Pr(K^{\bullet}_{\alpha\Rightarrow\beta})$ (• *closure*)
- $(\bullet\ 2)$ $K^{\bullet}_{\alpha\Rightarrow\beta} \subseteq Pr(K \cup \{\alpha \Rightarrow \beta\})$ (• *inclusion*)
- $(\bullet\ 3)$ If $\top \Rightarrow \bot \notin Pr(K \cup \{\alpha \Rightarrow \beta\})$, then $Pr(K \cup \{\alpha \Rightarrow \beta\}) \subseteq K^{\bullet}_{\alpha\Rightarrow\beta}$ (• *vacuity*)
- $(\bullet\ 4)$ $\alpha \Rightarrow \beta \in K^{\bullet}_{\alpha\Rightarrow\beta}$ (• *success*)
- $(\bullet\ 5)$ If $\alpha \Rightarrow \beta \equiv_{Pr} \alpha' \Rightarrow \beta'$, then $K^{\bullet}_{\alpha\Rightarrow\beta} = K^{\bullet}_{\alpha'\Rightarrow\beta'}$ (• *extensionality*)
- $(\bullet\ 6)$ If $\top \Rightarrow \bot \notin Pr(\alpha \Rightarrow \beta)$, then $\top \Rightarrow \bot \notin Pr(K^{\bullet}_{\alpha\Rightarrow\beta})$ (• *consistency*)
- $(\bullet +)$ $K^{\bullet}_{\alpha\Rightarrow\beta} = Pr(K^{\bullet}_{\top\Rightarrow\alpha\rightarrow\beta} \cup \{\alpha \Rightarrow \beta\})$ (• *extra*)

Preferential revision

Levi-style Identity for consistency preservation:

$$K^{\bullet}_{\alpha\Rightarrow\beta} := Pr(K^{-}_{\top\Rightarrow\alpha\wedge\neg\beta} \cup \{\alpha \Rightarrow \beta\}) \qquad (1)$$

Theorem [Casini & Meyer (2017)]

A revision operator $\bullet$ for preferential entailment Pr satisfies $(\bullet\ 1) - (\bullet\ 6)$ and $(\bullet +)$ iff it can be defined, via (1), from a contraction operator satisfying the postulates $(-\ 1) - (-\ 6)$

Preferential revision

Monotonic Core Pr.
Postulates for Revision $\circ$ (coherence preservation)

The postulates for revision for coherence preservation are as follows:

- ($\circ$ 1) $K^{\circ}_{\alpha\Rightarrow\beta} = Pr(K^{\circ}_{\alpha\Rightarrow\beta})$ ($\circ$ *closure*)
- ($\circ$ 2) $K^{\circ}_{\alpha\Rightarrow\beta} \subseteq Pr(K \cup \{\alpha \Rightarrow \beta\})$ ($\circ$ *inclusion*)
- ($\circ$ 3) If $\alpha \Rightarrow \bot \notin Pr(K \cup \{\alpha \Rightarrow \beta\})$, then $Pr(K \cup \{\alpha \Rightarrow \beta\}) \subseteq K^{\circ}_{\alpha\Rightarrow\beta}$ ($\circ$ *vacuity*)
- ($\circ$ 4) $\alpha \Rightarrow \beta \in K^{\circ}_{\alpha\Rightarrow\beta}$ ($\circ$ *success*)
- ($\circ$ 5) If $\alpha \Rightarrow \beta \equiv_{Pr} \alpha' \Rightarrow \beta'$, then $K^{\circ}_{\alpha\Rightarrow\beta} = K^{\circ}_{\alpha'\Rightarrow\beta'}$ ($\circ$ *extensionality*)
- ($\circ$ 6) If $\alpha \Rightarrow \bot \notin Pr(\alpha \Rightarrow \beta)$, then $\alpha \Rightarrow \bot \notin Pr(K^{\circ}_{\alpha\Rightarrow\beta})$ ($\circ$ *coherence*)

Preferential revision

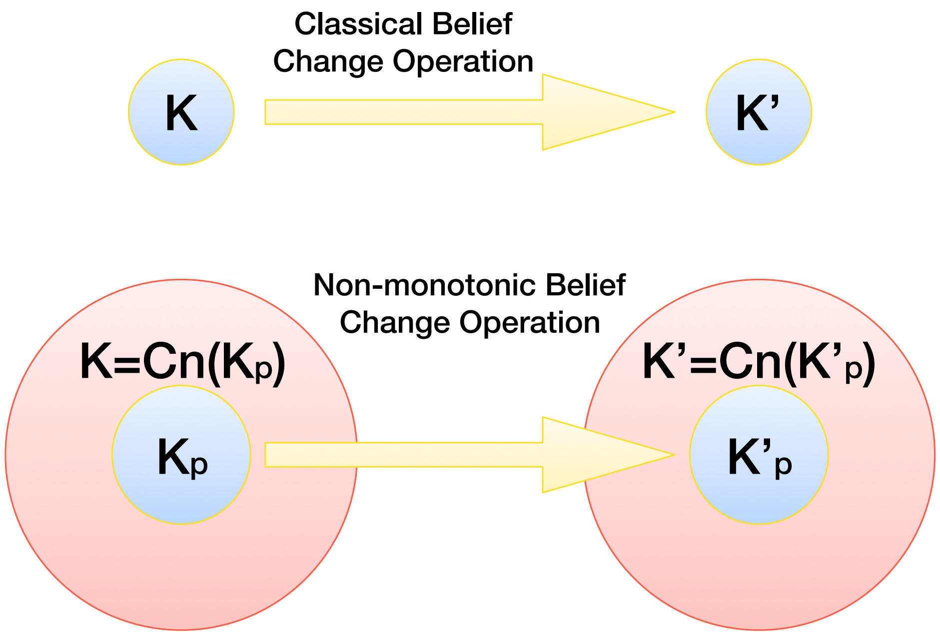
Levi-style Identity for consistency preservation:

$$K^{\circ}_{\alpha \Rightarrow \beta} := Pr(K^{-}_{\alpha \Rightarrow \neg\beta} \cup \{\alpha \Rightarrow \beta\}) \qquad (2)$$

Theorem [Casini et Al. (2018)]

A revision operator **∘** for preferential entailment Pr satisfies $(\circ\ 1)\ -\ (\circ\ 6)$ iff it can be defined, via (2), from a contraction operator satisfying the postulates $(-\ 1)\ -\ (-\ 6)$

BR for preferential conditionals



We have characterised contraction and revision for the monotonic core.

In order to characterise revision w.r.t. a Cn-theory K, we need to keep track and refer to it's monotonic core (K_p):

- We want to add $A \Rightarrow B$ to K. An actual revision needs to be done only if there is a conflict with the monotonic core K_p.

Revision of a non-monotonic theory would always keep track of the theory and its monotonic core.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

BR for preferential conditionals

Given Cn-theory K, we need to keep track and refer to it's monotonic core (K_p)

Non-monotonic Closure Cn. Postulates for Revision $\odot$ (consistency preservation)

The postulates for revision for consistency preservation are as follows:

- ($\odot$ 1) $K^{\odot}_{\alpha\Rightarrow\beta} = Cn(K^{\odot}_{\alpha\Rightarrow\beta})$ ($\odot$ *closure*)
- ($\odot$ 2) $\exists K'$ s.t. $Cn(K') = Cn(K^{\odot}_{\alpha\Rightarrow\beta})$ and $K' \subseteq Pr(K_p \cup \{\alpha \Rightarrow \beta\})$ ($\odot$ *generator inclusion*)
- ($\odot$ 3) If $\top \Rightarrow \bot \notin Pr(K_p \cup \{\alpha \Rightarrow \beta\})$, then $Cn(K_p \cup \{\alpha \Rightarrow \beta\}) \subseteq K^{\odot}_{\alpha\Rightarrow\beta}$ ($\odot$ *vacuity*)
- ($\odot$ 4) $\alpha \Rightarrow \beta \in K^{\odot}_{\alpha\Rightarrow\beta}$ ($\odot$ *success*)
- ($\odot$ 5) If $\alpha \Rightarrow \beta \equiv_{Pr} \alpha' \Rightarrow \beta'$, then $K^{\odot}_{\alpha\Rightarrow\beta} = K^{\odot}_{\alpha'\Rightarrow\beta'}$ ($\odot$ *extensionality*)
- ($\odot$ 6) If $\top \Rightarrow \bot \notin Pr(\alpha \Rightarrow \beta)$, then $\top \Rightarrow \bot \notin Pr(K^{\odot}_{\alpha\Rightarrow\beta})$ ($\odot$ *consistency*)

BR for preferential conditionals

Non-monotonic Closure Cn. Postulates for Revision $\otimes$ (coherence preservation)

The postulates for revision for consistency preservation are as follows:

- ($\otimes$ 1) $K^{\otimes}_{\alpha\Rightarrow\beta} = Cn(K^{\otimes}_{\alpha\Rightarrow\beta})$ ($\otimes$ *closure*)
- ($\otimes$ 2) $\exists K'$ s.t. $Cn(K') = Cn(K^{\otimes}_{\alpha\Rightarrow\beta})$ and $K' \subseteq Pr(K_p \cup \{\alpha \Rightarrow \beta\})$ ($\otimes$ *generator inclusion*)
- ($\otimes$ 3) If $\alpha \Rightarrow \bot \notin Pr(K_p \cup \{\alpha \Rightarrow \beta\})$, then $Cn(K_p \cup \{\alpha \Rightarrow \beta\}) \subseteq K^{\odot}_{\alpha\Rightarrow\beta}$ ($\otimes$ *vacuity*)
- ($\otimes$ 4) $\alpha \Rightarrow \beta \in K^{\otimes}_{\alpha\Rightarrow\beta}$ ($\otimes$ *success*)
- ($\otimes$ 5) If $\alpha \Rightarrow \beta \equiv_{Pr} \alpha' \Rightarrow \beta'$, then $K^{\otimes}_{\alpha\Rightarrow\beta} = K^{\otimes}_{\alpha'\Rightarrow\beta'}$ ($\otimes$ *extensionality*)
- ($\otimes$ 6) If $\alpha \Rightarrow \bot \notin Pr(\alpha \Rightarrow \beta)$, then $\alpha \Rightarrow \bot \notin Pr(K^{\otimes}_{\alpha\Rightarrow\beta})$ ($\otimes$ *coherence*)

BR for preferential conditionals

Theorem [Casini & Meyer (2017)]

A revision operator $\odot$ for suprapreferential entailment Cn satisfies $(\odot\ 1)\ -\ (\odot\ 6)$ iff there is a preferential revision operator $\bullet$ satisfying the postulates $(\bullet\ 1)\ -\ (\bullet\ 6)$ s.t.

$$K^{\odot}_{\alpha\Rightarrow\beta} = Cn(K^{\bullet}_{p_{\alpha\Rightarrow\beta}})$$

Theorem [Casini & Meyer (2017)]

A revision operator $\otimes$ for suprapreferential entailment Cn satisfies $(\otimes\ 1)\ -\ (\otimes\ 6)$ iff there is a preferential revision operator $\circ$ satisfying the postulates $(\circ\ 1)\ -\ (\circ\ 6)$ s.t.

$$K^{\otimes}_{\alpha\Rightarrow\beta} = Cn(K^{\circ}_{p_{\alpha\Rightarrow\beta}})$$

The semantic characterisation of the above operations will be presented in [Casini et Al. (2018)]

Example

We have the following KB B, that is closed by a supra-preferential closure operator Cn:

$$horse \Rightarrow tall; \quad horse \Rightarrow black;$$

$$horse \Rightarrow live\,.\,in\,.\,farm$$

Let's consider some new pieces of information:

- $horse \Rightarrow \neg(tall \wedge live\,.\,in\,.\,farm)$
- $horse \wedge black \Rightarrow \neg tall$
- $horse \wedge brown \Rightarrow \neg tall$

Example

We have the following KB B, that is closed by a supra-preferential closure operator Cn:

$$horse \Rightarrow tall; \quad horse \Rightarrow black;$$

$$horse \Rightarrow live.in.farm$$

Let's consider some new pieces of information:

- $horse \Rightarrow \neg(tall \wedge live.in.farm)$
- $horse \wedge black \Rightarrow \neg tall$
- $horse \wedge brown \Rightarrow \neg tall$

How do we manage the introduction of each of these pieces of information, starting from B?

Example

Knowledge base B: $horse \Rightarrow tall$; $horse \Rightarrow black$; $horse \Rightarrow live.in.farm$

Preferential Properties (defining the monotonic core):

(REF)	$\alpha \Rightarrow \alpha$	Reflexivity
(CT)	$\dfrac{\alpha \Rightarrow \beta \quad \alpha \wedge \beta \Rightarrow \gamma}{\alpha \Rightarrow \gamma}$	Cut (Cumulative Trans.)
(CM)	$\dfrac{\alpha \Rightarrow \beta \quad \alpha \Rightarrow \gamma}{\alpha \wedge \beta \Rightarrow \gamma}$	Cautious Monotony
(LLE)	$\dfrac{\alpha \Rightarrow \gamma \quad \models \alpha \equiv \beta}{\beta \Rightarrow \gamma}$	Left Logical Equival.
(RW)	$\dfrac{\alpha \Rightarrow \beta \quad \beta \models \gamma}{\alpha \Rightarrow \gamma}$	Right Weakening
(OR)	$\dfrac{\alpha \Rightarrow \gamma \quad \beta \Rightarrow \gamma}{\alpha \vee \beta \Rightarrow \gamma}$	Left Disjunction

New conditionals:

- $horse \Rightarrow \neg(tall \wedge live.in.farm)$
- $horse \wedge black \Rightarrow \neg tall$
- $horse \wedge brown \Rightarrow \neg tall$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

ASP

The investigation of non-monotonic contraction in the conditional framework has to be done.

It is instead at the base of the approach to revision for logic programs in [Zhuang et Al. (2016)].

Disjunctive logic programs are based on rules of the form:

$$a_1; \ldots, a_m \leftarrow b_1, \ldots, b_n, \textbf{not } c_1, \ldots, \textbf{not } c_0$$

ASP

Consider the following program:

- $Teach(John) \leftarrow Prof(John),$ **not** $Admin(John)$
- $Prof(John) \leftarrow$

  From this we conclude $\{Prof(John), Teach(John)\}$
  We are informed that

- $\leftarrow Teach(John)$
  that is in conflict with the previous program (no answer set).

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

ASP

Consider the following program:

- $Teach(John) \leftarrow Prof(John),$ **not** $Admin(John)$
- $Prof(John) \leftarrow$

  From this we conclude $\{Prof(John), Teach(John)\}$
  We are informed that

- $\leftarrow Teach(John)$
  that is in conflict with the previous program (no answer set).

We can fix the situation in two ways:

- We eliminate some rule in the program, or
- We add $Admin(John) \leftarrow$ to the program.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

ASP

[Zhuang et Al. (2016)] characterise belief change in the framework of grounded disjunctive logic programs defining an operator $P \star Q$ s.t.:

P and Q are two programs, and $P \star Q$ gives back a consistent program containing Q.

In case of conflict, either P is weakened, or more rules are added.

If the latter solution is impossible, it means that the conflict between P and Q is a ***monotonic inconsistency***.

Essential Bibliography

Preferential conditionals:

- S. Kraus, D. Lehmann, M. Magidor (1990), *Nonmonotonic Reasoning, Preferential Models and Cumulative Logics*. Artificial Intelligence, 44, pp. 167-207.

- D. Lehmann, M. Magidor (1992), *What Does a Conditional Knowledge Base Entail?*. Artificial Intelligence, 55, pp. 1-60

AGM vs. conditionals:

- P. Gärdenfors (1988), *Knowledge in Flux*. MIT Press (reprinted by College Publications, 2008)

- H. Rott (1989), *Conditionals and Theory Change: Revisions, Expansions and Additions*. Synthese, 81, pp. 91–113

Revision of Conditional KBs:

- G. Casini, T. Meyer (2017), *Belief Change in a Preferential Non-monotonic Framework*. Proc. of IJCAI 2017, pp. 929-935

- G. Casini, E. Fermé, T. Meyer, I. Varzinczak (2018), *A Semantic Perspective on Belief Change in a Preferential Non-Monotonic Framework*. Proceedings of KR 2018.

Revision in ASP:

- Z. Zhuang, J. Delgrande, A. Nayak, A. Sattar (2016), Reconsidering AGM-Style Belief Revision in the Context of Logic Programs. Proc. of ECAI 2016, pp. 671-679

BR and Description Logics

Today's topic

Can we use the AGM approach as a basis to model belief change in the area of Formal Ontologies?

- We take under consideration the family of Description Logics, the logical counterpart of the most popular formalism in the area, the OWL family.

- It is an area in which it is important to properly manage the dynamics of information.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Preliminaries - Description Logics

DLs represent the logical foundation for the OWL family of languages, providing them with a formal semantics and allowing the development of reasoners.

DLs allow the definition of two components of a KB, corresponding to two kinds of information.

- The **TBox**, capturing information on a general, conceptual level.
- The **ABox**, capturing information about individuals.

Preliminaries - Description Logics

The statements contained in the TBox are general concept inclusions (GCIs):

$$C \sqsubseteq D$$

Read as "the concept C is subsumed by the concept D" (equivalently, the class C is a subclass of the class D).

C and D are *concepts* (classes, sets of individuals), that are built from two sets:

- Concept Names $\mathsf{N}_{\mathscr{C}} := \{A_1, A_2, \ldots\}$
- Role Names $\mathsf{N}_{\mathscr{R}} := \{r_1, r_2, \ldots\}$

Preliminaries - Description Logics

The concepts can be built from $\mathsf{N}_{\mathscr{C}}$ and $\mathsf{N}_{\mathscr{R}}$ using various operators. For example:

- Propositional Connectives: $\sqcap, \sqcup, \neg$
- Logical Constants: $\top, \bot$
- Quantifiers: $\forall, \exists, \geq_n, \leq_n, \ldots$

In the DL *ALC*, for example, concepts can be constructed in the following way

$$C ::= A \mid (C_1 \sqcap C_2) \mid (C_1 \sqcup C_2) \mid \neg C \mid \exists r . C \mid \forall r . C$$

Preliminaries - Description Logics

The Semantics is given by means of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where:

- $\Delta^{\mathcal{I}}$ is a nonempty set (domain);
- $\cdot^{\mathcal{I}}$ is a mapping (interpretation function) defined as follows (in *ALC*):

NAMES:			
concept	A	$Vehicle$	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role	r	$hasPart$	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
tautology	$\top$		$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
contradiction	$\bot$		$\bot^{\mathcal{I}} = \emptyset$
CONNECTIVES:			
conjunction	$C \sqcap D$	$Vehicle \sqcap Red$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$Vehicle \sqcup Red$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\neg Vehicle$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
RESTRICTIONS:			
existential	$\exists r . C$	$\exists hasPart . Wheel$	$\{x \mid \exists y$ **s.t.** $(x, y) \in r^{\mathcal{I}}$ **and** $y \in C^{\mathcal{I}}\}$
universal	$\forall r . C$	$\forall hasPart . Metal$	$\{x \mid \forall y$, **if** $(x, y) \in r^{\mathcal{I}}$ **then** $y \in C^{\mathcal{I}}\}$

Preliminaries - Description Logics

For example, the expression

$$Vehicle \sqcap \exists\ hasPart\,.\,Wheel$$

indicates the class of the vehicles that have at one wheels. While the concept inclusion

$$Sparrow \sqsubseteq Bird$$

indicates that Sparrows are Birds.

- An interpretation satisfies a GCI ($\mathcal{I} \vDash C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- A TBox is a finite set of GCIs: **T** $= \{C_i \sqsubseteq D_i \mid 1 \leq i \leq n\}$
- $\mathcal{I}$ is a model of a TBox **T** if $\mathcal{I} \vDash C \sqsubseteq D$ for all the $C \sqsubseteq D \in$ **T**

Preliminaries - Description Logics

The ABox captures knowledge on an individual level.
We add to the vocabulary:

- Individual Names $\mathsf{N}_{\mathcal{O}} := \{a, b, c, \ldots\}$

The interpretation function $\cdot^{\mathcal{I}}$ is extended with:

- If $a \in \mathsf{N}_{\mathcal{O}}$, then $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$

The ABox can contain:

- Concept Assertions: $C(a)$, where $\mathcal{I} \models C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- Role Assertions: $r(a, b)$, where $\mathcal{I} \models r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
- An ontology $\mathcal{O} = (\mathbf{T}, \mathbf{A})$ is composed by a TBox **T** and an ABox A

By $\alpha, \beta, \ldots$ we indicate either a GCI or an ABox assertion.

Preliminaries - Description Logics

- $\mathcal{I}$ is a **model** of $\mathcal{O} = (\mathsf{T}, \mathsf{A})$ if $\mathcal{I}$ is a model of both T and A
- $\mathcal{O}$ is **consistent** if it has a model.
- $\mathcal{O}$ is **coherent** if each concept name A in $\mathcal{O}$ is satisfiable w.r.t. $\mathcal{O}$ (there is a model $\mathcal{I}$ of $\mathcal{O}$ s.t. $\mathcal{I} \not\models A \sqsubseteq \bot$
- A statement α is **entailed** by an ontology $\mathcal{O}(\mathcal{O} \models \alpha)$ if every model of $\mathcal{O}$ satisfies α.
  - If $\mathcal{O}$ is **inconsistent**, we will have $\mathcal{O} \models \top \sqsubseteq \bot$
  - If $\mathcal{O}$ is **incoherent**, we will have $\mathcal{O} \models A \sqsubseteq \bot$ for some A

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Preliminaries - Description Logics

Exercise:

Let's try to create a model for the following TBox **T** :

T

$Rat \sqsubseteq Mammal$ $\quad$ $Python \sqsubseteq Reptile$

$Mammal \sqcup Reptile \sqsubseteq Animal$

$Animal \sqsubseteq Mammal \sqcup Reptile$

$\exists hasPet\,.\,Python \sqsubseteq \exists hasPet\,.\,Rat$

$Python \sqsubseteq \exists\ eating\,.\,Rat$

Let's add also an ABox **A** :

A

$Python(Jimmy)$ $\quad$ $Rat(Bob)$ $\quad$ $Rat(Karl)$

$hasPet(Peter, Jimmy)$

Try also to find a counter-model

BR and Description Logics

Semantic Web and Formal Ontologies are areas in which managing the dynamics of information is particularly important.

The possibility of occurrence of conflicts is high, due, for example, to:

- Frequent updating of the information;
- Merging of ontologies;
- Pieces of information from different sources.

BR and Description Logics

Main questions:

- Can we apply the AGM approach to Description Logics (DLs)?
  - DLs have different expressivity w.r.t. Propositional Logic
  - We need to take under consideration not only the preservation of Consistency, but also the preservation of Coherence.

BR and Description Logics

A lot of work has been dedicated to define and implement procedures for debugging ontologies, that is, modifying ontologies that result inconsistent or incoherent.

Most of the work has been dedicated to the definition of specific procedures for debugging.

Some of the proposed procedures are often in line with a Base Revision approach [see, e.g., Horridge et Al. (2009)].

BR and Description Logics

However, the works in ontology debugging usually lack the kind of analysis in line with the Belief Revision approach, defining the desired properties that a change procedure should satisfy and characterising the classes of procedures satisfying them.

First step to develop such an analysis in the DL framework: check whether the AGM operations can be modelled inside the DL framework.

AGM Compliance

As we saw yesterday, AGM made some assumptions about the underlying logic $\langle L, Cn \rangle$:

- Language: closed under propositional operators.
- Consequence operator:

    1. Tarskian

        - Monotonicity: if $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$
        - Idempotence: $Cn(A) = Cn(Cn(A))$
        - Inclusion: $A \subseteq Cn(A)$

AGM Compliance

2. AGM Assumptions:

- Deduction: $\beta \in Cn(A \cup \{\alpha\})$ iff $(\alpha \rightarrow \beta) \in Cn(A)$
- Supraclassicality: if $\alpha \in Cl(A)$ then $\alpha \in Cn(A)$
- Compactness: if $\alpha \in Cn(A)$, then $\alpha \in Cn(A')$ for some finite $A' \subseteq A$
- Disjunction in the premises: $\dfrac{\gamma \in Cn(A \cup \{\alpha\}) \qquad \gamma \in Cn(A \cup \{\beta\})}{\gamma \in Cn(A \cup \{\alpha \vee \beta\})}$

AGM Compliance

Are all the above conditions necessary to define a contraction operator satisfying the six basic AGM postulates, or just sufficient?

Once we assume a Tarskian consequence operator, what are the necessary conditions for AGM contraction?

➡ Notion of AGM Compliance [Flouris et Al. (2005)]

AGM Compliance

Remember the basic AGM postulates?

$(\div 1)$	$K \div \alpha = Cn(K \div \alpha)$	(Closure)
$(\div 2)$	$K \div \alpha \subseteq K$	(Inclusion)
$(\div 3)$	If $\alpha \notin Cn(K)$, then $K \div \alpha = K$	(Vacuity)
$(\div 4)$	If $\alpha \notin Cn(\varnothing)$, then $\alpha \notin K \div \alpha$	(Success)
$(\div 5)$	If $\alpha \leftrightarrow \beta \in Cn(\varnothing)$, then $K \div \alpha = K \div \beta$	(Extensionality)
$(\div 6)$	$K \subseteq Cn((K \div \alpha) \cup \{\alpha\})$	(Recovery)

AGM Compliance

Note:

Once we assume a Tarskian consequence operator Cn, defining a contraction operator that satisfies $(\div 1) - (\div 5)$ is not problematic:

Let K be a Cn-theory. Let $\div$ be a contraction operator s.t., for every non-tautological α , $K_\alpha^\div$ is s.t.:

- If $\alpha \notin K_\alpha^\div$, then $K_\alpha^\div = K$
- Otherwise
  - $K_\alpha^\div \subset K$
  - $K_\alpha^\div = Cn(K_\alpha^\div)$
  - $\alpha \notin K$

It is easy to prove that any contraction satisfying these properties satisfies $(\div 1) - (\div 5)$, and that a contraction operator like that is always definable if Cn is Tarskian.

The potential problems rise if we consider also$(\div 6)$

AGM Compliance

A logic $\langle L, Cn \rangle$ is AGM-compliant if it is possible to define for it a contraction operation satisfying the six AGM postulates.

Let A, K be two sets of formulas in the logic $\langle L, Cn \rangle$ s.t. $K = Cn(K)$ and $Cn(\emptyset) \subset Cn(A) \subset K$

Given a set of formulas A, let $K^{-}(A)$ be defined as

$$K^{-}(A) := \{K' \mid Cn(K') \subset Cn(K) \text{ and } Cn(K' \cup A) = Cn(K)\}$$

$\langle L, Cn \rangle$ is decomposable if, for every A, K , $K^{-}(A) \neq \emptyset$.

Theorem [Flouris et Al. (2006)]

A logic $\langle L, Cn \rangle$ is AGM-compliant iff it is decomposable.

AGM Compliance

A logic $\langle L, Cn \rangle$ is AGM-compliant if it is possible to define for it a contraction operation satisfying the six AGM postulates.

Let A, K be two sets of formulas in the logic $\langle L, Cn \rangle$ s.t. $K = Cn(K)$ and $Cn(\emptyset) \subset Cn(A) \subset K$

Note: These are **proper** subset relations!

Given a set of formulas A, let $K^{-}(A)$ be defined as

$$K^{-}(A) := \{K' \mid Cn(K') \subset Cn(K) \text{ and } Cn(K' \cup A) = Cn(K)\}$$

$\langle L, Cn \rangle$ is decomposable if, for every A, K , $K^{-}(A) \neq \emptyset$.

Theorem [Flouris et Al. (2006)]

A logic $\langle L, Cn \rangle$ is AGM-compliant iff it is decomposable.

AGM Compliance

Most of the DLs are not AGM-compliant!

➡Another problem is the relation between Contraction and Revision: the expressivity of the language not always allows to re-formulate Levi's Identity.
We cannot express the negation of a GCI (we had an analogous problem with the conditionals yesterday).

This limits should not prevent from applying an AGM-like approach in the DL framework.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

BR in Description Logics

Not a lot of work has been done in analysing belief change in DLs from the point of view of the AGM approach.

Two relevant exceptions:

- Ribeiro, Wassermann (2008), *Base Revision for Ontology Debugging*, Journal of Logic and Computation, 19 (5), pp. 721-743.

  This paper analyses belief change for expressive DLs like $\mathcal{SHIF}$ and $\mathcal{SHOIN}$, in the framework of Base Revision.

- Zhuang, Wang, Wang, Qi (2016), *DL-Lite Contraction and Revision*, JAIR, 56, pp. 329-378.

  This paper deals with theory change for the DL-Lite family, a family of low-complexity DLs.
  We will focus on this paper, as a representative example of the problem.

DL-Lite

The DL-Lite family is a family of DLs with constrained expressivity and in which the decision problems are computationally feasible.

We introduce $DL-Lite_{core}$, the simplest logic in the family.

The vocabulary is composed by:

- A finite set of *Atomic Concepts*, that we indicate using $A_1, A_2, \ldots$
- A finite set of *Atomic Roles*, that we indicate using $P_1, P_2, \ldots$
- The *negation operator* $\neg$
- The *role inversion* function $\cdot^{-}$
- The *logical constants* $\top$, $\bot$
- The quantifier $\exists$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

DL-Lite

We can build:

- *Basic Concepts*: $B \to A \mid \exists R$
- *General Concepts*: $C \to B \mid \neg B$
- *Basic Roles*: $R \to P \mid P^{-}$

The TBox can contain the following kinds of Inclusions:

$$B \sqsubseteq C \quad \top \sqsubseteq C \quad B \sqsubseteq \bot$$

The ABox can contain the following kinds of statements:

$$A(a) \qquad P(a, b)$$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Preliminaries - Description Logics

Exercise:

The ontology specified before is a $DL-Lite_{core}$ ontology?

T

$$Rat \sqsubseteq Mammal \qquad Python \sqsubseteq Reptile$$
$$Mammal \sqcup Reptile \sqsubseteq Animal$$
$$Animal \sqsubseteq Mammal \sqcup Reptile$$
$$\exists hasPet.Python \sqsubseteq \exists hasPet.Rat$$
$$Python \sqsubseteq \exists\, eating.Rat$$

A

$$Python(Jimmy) \qquad Rat(Bob) \qquad Rat(Karl)$$
$$hasPet(Peter, Jimmy)$$

AGM Compliance

The DL-Lite family not AGM-compliant!

Example

Consider a DL-Lite TBox T $= \{ \top \sqsubseteq \neg A_1 \}$.
Let T' $= \{A_2 \sqsubseteq \neg A_1\}$. T' is a TBox s.t. $Cn(\mathrm{T'}) \subset Cn(\mathrm{T})$.

Is there a DL-Lite TBox T* s.t. $Cn(\mathrm{T^*}) \subset Cn(\mathrm{T})$ and $Cn(\mathrm{T'} \cup \mathrm{T^*}) = Cn(\mathrm{T})$?

Let's try to find one!

DL-Lite

Issues that need to be taken under consideration:

- Not AGM-compliant: how do we deal with the impossibility of satisfying the six basic AGM postulates?
- No proper negation: we cannot use Levi's Identity
- Consistency vs. Coherence: we need to consider the satisfaction of both these constraints.
- Implementability: Each consistent DL ontology has infinite models; that is a problem in order to have implementable semantic-based procedures.
- Multiple revision tasks: revision of the TBox, of the ABox alone (keeping a background TBox fixed), or of the TBox+ABox?

ct-type semantics

The first issue that Zhuang and others address is the semantics.

We want the semantics to be *succinct*, that is, the models should be finite and avoid the redundancy of information, in order to help w.r.t. *computational efficiency*.

- *ct-type semantics* (Core TBox type semantics).

ct-type semantics

The *ct-types* are a kind of finite, succinct interpretations appropriate for the $DL-Lite_{core}$ TBoxes.

- Let $\mathscr{B}$ be the (finite) set of basic concepts:

$$\mathscr{B} := \{A_1, \ldots, A_n, \exists P_1, \ldots, \exists P_m, \exists P_1^-, \ldots, \exists P_m^-\}$$

- Let Ω_c^t be the power set of $\mathscr{B}$

Ω_c^t can be interpreted like a set of propositional valuations

ct-type semantics

Let **T** be a $DL-Lite_{core}$ TBox.

An element v of Ω^t_c is a *propositional model* of **T** if $v \models \neg C \vee D$ for every $C \sqsubseteq D \in$ **T**

Let $\|$ **T** $\|^t_c$ be the set of propositional models of **T** ($\|$ **T** $\|^t_c \subseteq \Omega^t_c$).
$\|$ **T** $\|^t_c$ accounts for most of the inclusions enforced by **T** in $DL-Lite_{core}$, apart from

$$\frac{\exists R \sqsubseteq \bot}{\exists R^- \sqsubseteq \bot} \qquad \frac{\exists R^- \sqsubseteq \bot}{\exists R \sqsubseteq \bot}$$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

ct-type semantics

A ct-type τ is a *ct-model* of a TBox T if

1. $\tau \in \| \mathrm{T} \|_c^t$
2. If $\mathrm{T} \models \exists r \sqsubseteq \bot$ then $\exists r \notin \tau$

Let $|\mathrm{T}|_c^t$ be set of ct-models of a TBox T

$\mathrm{T} \models_c^t B \sqsubseteq C$ if $\tau \models \neg B \vee C$ for every $\tau \in |\mathrm{T}|_c^t$.

ct-types semantics is succinct, every TBox has a finite number of models, and it gives the correct characterisation of DL-entailment.

Theorem [Zhuang et Al.(2016)]

Let T be a $DL - Lite_{core}$ TBox, and $B \sqsubseteq C$ a $DL - Lite_{core}$ inclusion

$$\mathrm{T} \models_c^t B \sqsubseteq C \text{ iff } \mathrm{T} \models B \sqsubseteq C$$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

ct-type semantics

ct-types give us an alternative semantics for TBox reasoning in the simplest $DL-Lite$: $DL-Lite_{core}$

Slightly more complex semantical structures are defined for a more expressive $DL-Lite$ ($DL-Lite_R$), and to characterise ABox reasoning, but we will not introduce them.

Since we consider only ct-types we will describe only the operators for TBox changes in $DL-Lite_{core}$.

Belief change for the ABox or for the entire ontology can be defined analogously, referring to the dedicated semantic constructions.

ct-type semantics

Problem:
Under ct-type semantics, we lose the bijection between sets of interpretations and TBoxes

Let M be a set of ct-types. $\mathbf{T}$ is a corresponding TBox for M iff

- $M \subseteq |\mathbf{T}|_c^t$
- there is no TBox $\mathbf{T}'$ s.t. $M \subseteq |\mathbf{T}'|_c^t \subseteq |\mathbf{T}|_c^t$

ct-type semantics

If M is coherent (for every atomic $A, v \nvDash A \sqsubseteq \bot$ for some $v \in M$), then the corresponding TBox for M is unique.

An operator $\mathscr{T}$ is introduced, s.t. it takes as input a set of ct-types M

$\mathscr{T}(M) =$

- the closure of the corresponding TBox **T** (Cn(**T**)), if M is coherent;
- **T**$_\bot$ otherwise

Contraction

The authors define a contraction operator $\barwedge$ using ct-interpretations and a choice function.

Let ϕ be a set of inclusion statements $B \sqsubseteq C$.

- $|\phi|^t_c$ indicates the ct-models of A
- $|\neg\phi|^t_c := \Omega^t_c \setminus |\phi|^t_c$

Let γ be a choice function over the ct-interpretations s.t.

- If $M \neq \emptyset$, then $\emptyset \subseteq \gamma(M) \subseteq M$

where $M \subseteq \Omega^t_c$.
γ is *faithful* w.r.t. **T** iff

- If $|\mathbf{T}|^t_c \cap M \neq \emptyset$, then $\gamma(M) = |\mathbf{T}|^t_c \cap M$

Contraction

A contraction operation $\barwedge$ is defined using ct-models. Let **T** be a TBox and ϕ be a set of inclusion statements

$\barwedge$ is *T-contraction operator* if it can be defined as

$$\mathbf{T} \barwedge \phi := \mathscr{T}(|\mathbf{T}|_c^t \cup \gamma(|\neg\phi|_c^t))$$

where γ is faithful w.r.t. **T**

What about the postulates?

$DL-Lite_{core}$ is not AGM compliant, *recovery* cannot be saved

$$(\barwedge\ 6)\quad \mathbf{T} \subseteq Cn((\mathbf{T} \barwedge \phi) \cup \{\phi\})$$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Contraction

$(\barwedge\ 1) - (\barwedge\ 5)$ are just translated in the intuitive way:

$(\barwedge\ 1)$ **T** $\barwedge \phi = Cn((\mathbf{T} \barwedge \phi)$

$(\barwedge\ 2)$ **T** $\barwedge \phi \subseteq$ **T**

$(\barwedge\ 3)$ **If T** $\not\models \phi$**, then T** $\barwedge A =$ **T**

$(\barwedge\ 4)$ **If** $\emptyset \not\models \phi$ **for some** $B \sqsubseteq C \in \phi$**, then T** $\barwedge \phi \not\models \phi$

$(\barwedge\ 5)$ **If** $\phi \equiv \psi$**, then T** $\barwedge \phi =$ **T** $\barwedge \psi$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UNIVERSITÉ DU LUXEMBOURG

MIREL

Contraction

The postulate of ***Disjunctive Elimination*** is added

$(\barwedge - de)$ **If T** $\models \psi$ **and** $|\mathrm{T} \barwedge \phi|_c^t \subseteq |\phi|_c^t \cup |\psi|_c^t$**, then T** $\barwedge \phi \models \psi$

That in the propositional version was

If $\psi \in K$ **and** $\phi \vee \psi \in K \div \phi$**, then** $\psi \in K \div \phi$

Theorem [Zhuang et Al. (2016)]

$\barwedge$ is a T-contraction operator for a TBox **T** iff $\barwedge$ satisfies $(\barwedge\ 1) - (\barwedge\ 5)$ and $(\barwedge - de)$

The authors also specify a **computationally tractable** procedure implementing T-contractions.

Revision

We will skip the characterisation of the **T-revision operators** $\circledast$. We just mention two important points:

- Since there is not the possibility of using Levi's Identity, the revision operators are defined independently, using another kind of semantic choice function.
- The operators are defined with the goal of preserving coherence, instead of consistency.

Also for revision the authors propose a procedure implementing T-revision operators and working in polynomial time.

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

uni.lu UNIVERSITÉ DU LUXEMBOURG

MIREL

Essential Bibliography

- G. Flouris, D. Plexousakis, G. Antoniou (2005), *On Applying the AGM Theory to DLs and OWL*. Proc. Of ISWC 2005, LNCS 3729, pp. 216-231

- G. Qi, F. Yang (2008), *A survey of Revision Approaches in Description Logics*. Proc. of RR 2008, LNCS 5341, pp. 74-88

- M. Horridge, B. Parsia, U. Sattler (2009), The OWL Explanation Workbench: A Toolkit for Working with Justifications for Entailments in OWL Ontologies. Technical Report [url: https://tinyurl.com/yau48d37]

- M. M. Ribeiro, R. Wassermann (2009), *Base Revision for Ontology Debugging*. , Journal of Logic and Computation, 19 (5), pp. 721-743

- M. M. Ribeiro, R. Wassermann, G. Flouris, G. Antoniou (2013), *Minimal Change: Relevance and Recovery Revisited*. Artificial Intelligence, 201, pp. 59-80

- Z. Zhuang, Z. Wang, K. Wang, G. Qi (2016), *DL-Lite Contraction and Revision*. JAIR, 56, pp.329-378

Ex Falso Quodlibet

A principle of classical logic:

Ex Falso Quodlibet (EFQ): $$\frac{\alpha \qquad \neg\alpha}{\beta}$$

From a contradiction we can conclude any formula.

Ex Falso Quodlibet

The classical notion of consequence relation enforces this principles also for modern logic.

From Lecture 1:

Satisfiability of KBs

- A set *KB* of formulae is satisfied iff $\mathcal{I} \models \alpha$ for all $\alpha \in KB$
- An interpretation $\mathcal{I}$ is a model of set *KB* of formulae (denoted $\mathcal{I} \models KB$) iff $\mathcal{I} \models \alpha$ for all $\alpha \in KB$
- A set *KB* of formulae is
  - satisfiable, if there is some $\mathcal{I}$ that satisfies *KB*
  - unsatisfiable, if *KB* is not satisfiable
- A set *KB* of formulae entails a formula α iff α is true in all models of *KB*, i.e.

$$KB \models \alpha \quad \text{iff} \quad \mathcal{I} \models \alpha \text{ for all models of } KB$$

Ex Falso Quodlibet

Such a condition is equivalent to:

'The set of the models of the premises ($\|KB\|$) is a subset of the set of models of the consequence ($\|\alpha\|$)', that is:

$$\|KB\| \subseteq \|\alpha\|$$

Assume there is a contradiction in our premises, and we want to check whether

$$A \cup \{\alpha, \neg\alpha\} \models \beta$$

for some formula β.

Ex Falso Quodlibet

$A \cup \{\alpha, \neg\alpha\} \models \beta$ holds if and only if

$$\|A \cup \{\alpha, \neg\alpha\}\| \subseteq \|\beta\|. \tag{1}$$

The contradiction in the premises implies that $A \cup \{\alpha, \neg\alpha\}$ has no model, that is $\|A \cup \{\alpha, \neg\alpha\}\| = \emptyset$.

Condition (1) is trivially satisfied, since $\emptyset \subseteq \|\beta\|$ for any set $\|\beta\|$.

$$A \cup \{\alpha, \neg\alpha\} \models \beta$$

holds for any β.

Ex Falso Quodlibet

We need to avoid this "explosion".
Two possible strategies:

- Avoiding contradictions $\Rightarrow$ Belief Change.
- Avoiding the **EFQ** rule $\Rightarrow$ Paraconsistent Logics.

When we have to deal with huge knowledge bases, with pieces of information coming from different sources, the presence of contradiction is highly probable, and repairing the KB could be impossible.

Paraconsistent reasoning can then be a possible solution.

Paraconsistent Logics

Many proposals. Some are very close to belief change:

- Systems that allow conjunction only if consistent [Rescher and Manor, 1970].

  Given a knowledge base *KB*, let KB^* be the set of all the maximal consistent subsets of *KB*. That is

  $$KB^* = \{A \subseteq KB \mid \text{for every } B \text{ s.t. } A \subset B \subseteq KB,\ B \models \bot\}$$

  $$KB \models \alpha \text{ if and only if } A \models \alpha, \text{ for every } A \subseteq KB^*.$$

Paraconsistent Logics

Particularly relevant is the many-valued (MV) approach:

- Connection with the MV approach that has developed into the fuzzy logics.
- The most popular in computer science.

Originally proposed by Asenjo [Asenjo, 1966], who introduced a propositional logic with 3 possible truth values: True (*t*), False (*f*), and **Both** (*b*).

$\neg$	
t	f
b	b
f	t

$\wedge$	t	b	f
t	t	b	f
b	b	b	f
f	f	f	f

$\vee$	t	b	f
t	t	t	t
b	t	b	b
f	t	b	f

Very close to the Kleene and Łukasiewicz's original systems with 3 truth values.

Paraconsistent Logics

The notion of consequence relation can be reformulated as *KB* $\models \alpha$ if and only if, for every interpretation $\mathcal{I}$,

> If every formula in *KB* is either *t* or *b*, then α is either *t* or *b*.

We can avoid EXQ. For example, the contradiction is a consequence of the KB:

$$\alpha \wedge \neg\alpha, \beta \models \alpha \wedge \neg\alpha$$

but we do not have explosion, since

$$\alpha \wedge \neg\alpha, \beta \not\models \neg\beta$$

Paraconsistent Logics

- Various proposals in this line, using 3 (true, false, both) or 4 (true, false, both, neither) true value.
- From the semantical side, for example we can use a two interpretation functions, one to determine truth, and one falsehood.
- we have already seen an example last week with $\rho df_{\bot}^{\neg}$ [Straccia and Casini, 2022].

$\rho df_{\bot}^{\neg}$ interpretation:

$$\mathcal{I} = \langle \Delta_{\mathsf{R}}, \Delta_{\mathrm{DP}}, \Delta_{\mathsf{C}}, \Delta_{\mathsf{L}}, \mathsf{P}^{+}[\![\cdot]\!], \mathsf{P}^{-}[\![\cdot]\!], \mathsf{C}^{+}[\![\cdot]\!], \mathsf{C}^{-}[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

Some constraints:

- for domain element t, there is unique complement $\neg t$ ($\neg\neg t$ is t)
- $\mathsf{C}^{+}[\![\cdot]\!]$ and $\mathsf{C}^{-}[\![\cdot]\!]$ are functions $\Delta_{\mathsf{C}} \to 2^{\Delta_{\mathsf{R}}}$ with $\mathsf{C}^{+}[\![\neg c]\!] = \mathsf{C}^{-}[\![c]\!]$
- $\mathsf{P}^{+}[\![\cdot]\!]$ and $\mathsf{P}^{-}[\![\cdot]\!]$ are functions $\Delta_{\mathrm{DP}} \to 2^{\Delta_{\mathsf{R}} \times \Delta_{\mathsf{R}}}$ with $\mathsf{P}^{+}[\![\neg p]\!] = \mathsf{P}^{-}[\![p]\!]$
- $\cdot^{\mathcal{I}}$ maps each $t \in \mathbf{UL} \cap V$, that is not of the form $\star_c$, into a value $t^{\mathcal{I}} \in \Delta_{\mathsf{R}} \cup \Delta_{\mathrm{DP}}$, such that $(\neg t)^{\mathcal{I}} = \neg t^{\mathcal{I}}$

4-Valued Intentional Semantics for $\rho df_{\perp}^{\neg}$

We will have:

- $\mathsf{P}^{+}[\![]\!]$ and $\mathsf{C}^{+}[\![]\!]$
  - Positive extensions of $P[\![]\!]$ and $C[\![]\!]$
- $\mathsf{P}^{-}[\![]\!]$ and $\mathsf{C}^{-}[\![]\!]$
  - Negative extensions of $P[\![]\!]$ and $C[\![]\!]$
- $\mathsf{C}^{+}[\![c]\!]$ denotes the set of resources known to be instances of class c
- $\mathsf{C}^{-}[\![c]\!]$ denotes the set of resources known not to be instances of class c
- Positive and negative extensions need not to be the complement of each other
  - $r \notin \mathsf{C}^{+}[\![c]\!]$ does not imply necessarily that $r \in \mathsf{C}^{-}[\![c]\!]$
  - $\mathsf{C}^{-}[\![c]\!]$ is not enforced to be $\Delta_{\mathsf{R}} \setminus \mathsf{C}^{+}[\![c]\!]$

Bibliography

Asenjo, F. G. (1966).
A calculus of antinomies.
Notre Dame J. Formal Logic, 7(1):103–105.

Rescher, N. and Manor, R. (1970).
On inferences from inconsistent premises.
Theory and Decision, 1(2):179–217.

Straccia, U. and Casini, G. (2022).
A minimal deductive system for RDFS with negative statements.
In Kern-Isberner, G., Lakemeyer, G., and Meyer, T., editors, *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel, July 31 - August 5, 2022*.