Propositional Logic for KRR

We are going to look at propositional and first order logic, and consider their suitability for use in knowledge representation.

For a comprehensive review of these logics, I recommend:

Logic for Computer Scientists by Ewe Shöning.

Propositional Logic

We might consider using Propositional Logic

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

Syntax: Propositional Alphabet

1. Propositional variables (**PL**):

   basic statements that can be true or false, e.g., *Sunny*
2. The symbols $\top$ ("truth") and $\bot$ ("falsehood")
3. Propositional connectives:
   - $\neg$: negation (not)
   - $\wedge$: conjunction (and)
   - $\vee$: disjunction (or)
   - $\rightarrow$: implication (if … then)
   - $\leftrightarrow$: bi-directional implication (if and only if)
4. Punctuation symbols "(" and ")" can be used to avoid ambiguity

Syntax: Formulas

Atomic formulas (atoms): propositional variables, e.g., *Sunny*

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

- Each atom is a formula.
- $\top$ and $\bot$ are formulas.
- If α is a formula, then $\neg\alpha$ is a formula.
- If α, β are formulas, then so is $(\alpha \circ \beta)$, for $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Examples of formulas:

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

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

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

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

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

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

Semantics: Interpretations

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

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

Then, we extend $\mathcal{I}$ by induction to assign a truth value to complex formulas as follows:

- $\bot^{\mathcal{I}} =$ false.
- $\top^{\mathcal{I}} =$ true.
- $(\neg\alpha)^{\mathcal{I}} =$ true if and only if $\alpha^{\mathcal{I}} =$ false.
- $(\alpha_1 \wedge \alpha_2)^{\mathcal{I}} =$ true if and only if $\alpha_1^{\mathcal{I}} =$ true and $\alpha_2^{\mathcal{I}} =$ true.
- $(\alpha \vee \beta)^{\mathcal{I}} =$ true if and only if $\alpha_1^{\mathcal{I}} =$ true or $\alpha_2^{\mathcal{I}} =$ true (or both).
- $(\alpha \rightarrow \beta)^{\mathcal{I}} =$ false if and only if $\alpha^{\mathcal{I}} =$ true and $\beta^{\mathcal{I}} =$ false.
- $(\alpha \leftrightarrow \beta)^{\mathcal{I}} =$ true if and only if $\alpha^{\mathcal{I}} = \beta^{\mathcal{I}}$.

Semantics

In other words, truth values for complex formulas are given by the table below, where 1 stands for true and 0 for *false*:

α	β	$\neg\alpha$	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$	$\alpha \leftrightarrow \beta$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

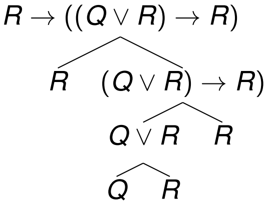
Example: An interpretation $\mathcal{I}$ for the formula $R \rightarrow ((Q \vee R) \rightarrow R)$:

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

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

Semantics of Formulas

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



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

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

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

Using PL for KR

Propositional Logic provides a simple KR language.

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

1. Identify the relevant propositions:

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

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

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

Reasoning with a Knowledge Base

Knowledge Base $\mathcal{K}_1$:

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

Knowledge Base $\mathcal{K}_2$:

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

We would like to answer the following questions:

1. Do our KBs make sense?

   $\mathcal{K}_1$ seems contradictory

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

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

Satisfiability Problem

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

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

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

Satisfiability defined for sets of formulas in the obvious way.

The following knowledge base is unsatisfiable:

$$\begin{aligned}\mathcal{K}_1 \quad = \quad \{ & \mathit{Benign} \wedge \mathit{Stage4} \\ & \mathit{Benign} \leftrightarrow \neg \mathit{Metastasis} \\ & \mathit{Stage4} \rightarrow \mathit{Metastasis} \\ & \ldots \}\end{aligned}$$

Other Reasoning Problems

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

Entailment: An instance is a pair of formulas α, β.
The answer is true if every model of α is also a model of β and false otherwise.

Equivalence: An instance is a pair of formulas α, β.
The answer is true if the set of all models of α and β coincide and false otherwise.

Reductions Between Problems

Intuitively, these problems are strongly related:

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

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

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

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

Expressivity -v- Complexity

Propositional satisfiability is (famously) NP-complete:

Theorem (Cook-Levin)

Propositional satisfiability is an NP-complete problem:

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

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

NO!

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