Knowledge Representation

Introduction to Artificial Intelligence

Chandra Gummaluru
University of Toronto

Version W22.1

- The following is based on material developed by many individuals, including (but not limited to):
  - Sheila McIlraith
  - Bahar Aameri
  - Fahiem Bacchus
  - Sonya Allin

- Part of being an intelligent agent involves being able to infer implicit facts based on known or assumed ones.

  Example: Avatar Aang's Love Life
  - Suppose we knew the following:
    - Anyone who likes someone is sad if they do not like them back.
    - Aang likes Katara.
    - Aang is an fire bender.
    - Aang is sad.
    - Katara does not like all fire benders.
  - Does Katara like Aang?

- Humans develop this ability through experience. Our goal is to instil artificial agents with the same ability.

- Without reasoning, we would have to explicitly remember every fact we've learned.

- To achieve this ability, we do two things:

  1. **Represent** (encode) known statements in our brain.
  2. **Reason** (infer) new statements from the known ones.

- Thus, to achieve this ability artificially, we need to do things:

  1. Develop a **formal languages** to represent statements.
  2. Develop a **reasoning mechanism** for the formal system.

- There are many formal languages and reasoning mechanisms we could use.
- We will consider a representation called **first-order logic** (FOL), and a reasoning mechanism called **resolution**.

Informal Languages versus Formal Languages

- Roughly speaking, the goal of any formal language is to facilitate the expression of knowledge but with strict rules to avoid any ambiguity.
- **Example**: Ambiguity in English
  - What is the appropriate response to the request, "call me an ambulance,"?
    - "okay."
    - "uh...you're an ambulance...".
- The ambiguity arises from the fact that, in English (and other languages), there are multiple interpretations of many words/phrases.

- To avoid such ambiguity, a formal language must define the notion used to build its statements, as well as a system for interpreting those statements.
- The notion is called the **syntax** and the interpretations are the **semantics**.
- So, a formal language needs to provide syntax and a way to introduce semantics. However, it does not provide the semantics themselves.

- FOL syntax consists of the following components:

  1. **variables**, where each variable is, by definition, a **term**.
     - E.g., x.

  2. **functions**, which each map zero or more terms to a single term.
     - E.g., nation(x), which should refer to x's nation.

  3. **predicates**, which each map many terms to true/false.
     - E.g., likes(x, y), which should mean that x likes y.

- An FOL **vocabulary** is a triple, $\mathcal{L} = (\mathcal{V}, \mathcal{F}, \mathcal{P})$, where $\mathcal{V}, \mathcal{F}$, and $\mathcal{P}$ are sets of variables, functions, and predicates, respectively.

- Let $\mathcal{L}$ be a vocabulary.
- For any n-ary $\mathcal{L}$-predicate P, and $\mathcal{L}$-terms, $t_1, \ldots, t_n$, the expression, $P(t_1, \ldots, t_n)$ is called an atomic $\mathcal{L}$-formula.
  - E.g., $\mathrm{likes}(x, \mathrm{nation}(y))$.
- Atomic formula represents the most fundamental statements.

- Non-atomic $\mathcal{L}$-formulae are defined recursively as follows:
  - **negation**: $\neg f$, where f is any $\mathcal{L}$-formula.
  - **disjunction**: $f_1 \vee f_2$, where f_1 and f_2 are $\mathcal{L}$-formulae.
  - **conjunction**: $f_1 \wedge f_2$, where f_1 and f_2 are $\mathcal{L}$-formulae.
  - **implication**: $f_1 \rightarrow f_2$, where f_1 and f_2 are $\mathcal{L}$-formulae.
  - **existential**: $\exists x f$, where x is a variable and f is any $\mathcal{L}$-formula.
  - **universal**: $\forall x f$, where x is a variable and f is any $\mathcal{L}$-formula.

Example: Aang's Love Life

- Suppose we have a vocabulary, $\mathcal{L}$, with:
  - constant symbols: Katara, Aang, Zuko, Azula, fire, water, air, earth
  - variable symbols; x, y
  - function symbols; sibling(x), and nation(x),
  - predicate symbols; person(x), sad(x), likes(x, y), and bender(x, y).
- We can express each fact using our vocabulary:
  1. Anyone who is likes someone is sad if they don't like them back:

     $$\forall x\,(\text{person}(x) \rightarrow (\exists y\,(\text{person}(y) \wedge \text{likes}(x, y) \wedge \neg\text{likes}(y, x)) \rightarrow \text{sad}(x)))$$

  2. Aang likes Katara: $\text{likes}(\text{Aang}, \text{Katara})$.
  3. Aang is a fire bender: $\text{bender}(\text{Aang}, \text{fire})$.
  4. Katara does not like all fire benders: $\exists x \text{bender}(x, \text{fire}) \wedge \neg\text{likes}(\text{Katara}, x)$.
  5. Aang is sad: $\text{sad}(\text{Aang})$.

- In FOL, the semantics are provided by what we refer to as a **structure**. The model, $\mathcal{S}$, consists of the following components:

  1. a **domain of discourse**, D, which is a set of relevant elementary objects.
  2. **specializations of functions**, $f^{\mathcal{S}} : D^n \to D$, for each n-ary function, f, so that $f^{\mathcal{S}}$ assigns f within D.
  3. **specializations of predicates**, $p^{\mathcal{S}} \subseteq D^n$, for each n-ary predicate, p, so that $p(t_1, \ldots, t_n)$ is true if and only if $(t_1, \ldots, t_n) \in p^{\mathcal{S}}$.

Example: Avatar Aang's Love Life

- A structure, $\mathcal{S}$ of $\mathcal{L}$ is shown below:
  - $D = \{\textbf{Aang}, \textbf{Katara}, \textbf{Zuko}, \textbf{Azula}, \textbf{water}, \textbf{fire}, \textbf{air}, \textbf{earth}\}$
  - $\text{Katara}^{\mathcal{S}} = \textbf{Katara}$, $\text{Aang}^{\mathcal{S}} = \textbf{Aang}$, $\text{Zuko}^{\mathcal{S}} = \textbf{Zuko}$, $\text{Azula}^{\mathcal{S}} = \textbf{Azula}$
  - $\text{sibling}^{\mathcal{S}}(\textbf{Zuko}) = \textbf{Azula}$, $\text{sibling}^{\mathcal{S}}(\textbf{Azula}) = \textbf{Zuko}$
  - $\text{nation}^{\mathcal{S}}(\textbf{Aang}) = \textbf{air}$, $\text{nation}^{\mathcal{S}}(\textbf{Katara}) = \textbf{water}$, $\text{nation}^{\mathcal{S}}(\textbf{Zuko}) = \textbf{fire}$, $\text{nation}^{\mathcal{S}}(\textbf{Azula}) = \textbf{fire}$
  - $\text{bender}^{\mathcal{S}} = \{\langle \textbf{Aang}, \textbf{water}\rangle, \langle \textbf{Aang}, \textbf{fire}\rangle, \langle \textbf{Aang}, \textbf{air}\rangle, \langle \textbf{Aang}, \textbf{earth}\rangle, \langle \textbf{Katara}, \textbf{water}\rangle, \langle \textbf{Zuko}, \textbf{fire}\rangle, \langle \textbf{Azula}, \textbf{fire}\rangle\}$
  - $\text{likes}^{\mathcal{S}} = \{\langle \textbf{Katara}, \textbf{water}\rangle, \langle \textbf{Aang}, \textbf{Katara}\rangle, \langle \textbf{Katara}, \textbf{Aang}\rangle\}$
  - $\text{person}^{\mathcal{S}} = \{\langle \textbf{Aang}\rangle, \langle \textbf{Katara}\rangle, \langle \textbf{Zuko}\rangle, \langle \textbf{Azula}\rangle\}$
  - $\text{sad}^{\mathcal{S}} = \{\langle \textbf{Aang}\rangle, \langle \textbf{Zuko}\rangle\}$

- An occurrence of a variable, x, in an FOL formula, f, is **bound** if and only if it is in a sub-formula of the form $\forall x f'$ or $\exists x f'$. Otherwise, x is **free**.
- If a formula f contains free variables, then its value depends on what values we assign to those free variables.
- We define an **assignment function**, $\sigma : \mathcal{V} \to D$, so that $\sigma(x)$ is the element in the universe represented by the variable x.
- The value of any $\mathcal{L}$-term is defined recursively with an **extended assignment function**, $\bar{\sigma}$ so that $\bar{\sigma}(x) = \sigma(x)$ and $\bar{\sigma}(f(t_1, \ldots, t_n)) = f^{\mathcal{S}}(\bar{\sigma}(t_1), \ldots, \bar{\sigma}(t_n))$.

Example: Avatar Aang's Love Life

- Suppose we wanted to compute the value of nation(sibling(x)), under $\sigma(x) = \textbf{Azula}$.
- We proceed as follows:

$$\begin{aligned}
\bar{\sigma}\,(\text{nation}(\text{sibling}(x))) &= \text{nation}^{\mathcal{S}}(\bar{\sigma}(\text{sibling}(x))) \\
&= \text{nation}^{\mathcal{S}}\left(\text{sibling}^{\mathcal{S}}(\bar{\sigma}(x))\right) \\
&= \text{nation}^{\mathcal{S}}\left(\text{sibling}^{\mathcal{S}}(\sigma(x))\right) \\
&= \text{nation}^{\mathcal{S}}\left(\text{sibling}^{\mathcal{S}}(\textbf{Azula})\right) \\
&= \text{nation}^{\mathcal{S}}\,(\textbf{Zuko}) \\
&= \textbf{fire}
\end{aligned}$$

- Notice that the value depends on both $\mathcal{S}$ and σ.

- We write $\mathcal{S} \models f[\sigma]$ to denote that $\mathcal{S}$ **satisfies** the formula, f, under σ.

- By definition, $\mathcal{S} \models P(t_1, \ldots, t_n)[\sigma]$ if and only if $\langle \bar{\sigma}(t_1), \ldots, \bar{\sigma}(t_n) \rangle \in P^{\mathcal{S}}$.

- For other $\mathcal{L}$-formulae, $\models$ is defined recursively as follows:
  - $\mathcal{S} \models (t_1 = t_2)[\sigma]$ if and only if $\bar{\sigma}(t_1) = \bar{\sigma}(t_2)$
  - $\mathcal{S} \models \neg f[\sigma]$ if and only if $\mathcal{S} \not\models f[\sigma]$
  - $\mathcal{S} \models (f_1 \vee f_2)[\sigma]$ if and only if $\mathcal{S} \models f_1[\sigma]$ or $\mathcal{S} \models f_2[\sigma]$
  - $\mathcal{S} \models (f_1 \wedge f_2)[\sigma]$ if and only if $\mathcal{S} \models f_1[\sigma]$ and $\mathcal{S} \models f_2[\sigma]$
  - $\mathcal{S} \models (f_1 \rightarrow f_2)[\sigma]$ if and only if $\mathcal{S} \models (f_1 \vee \neg f_2)[\sigma]$
  - $\mathcal{S} \models (\forall x f)[\sigma]$ if and only if $\mathcal{S} \models f[\sigma[x/m]]$ for all $m \in D$
  - $\mathcal{S} \models (\exists x f)[\sigma]$ if and only if $\mathcal{S} \models f[\sigma[x/m]]$ for some $m \in D$

- Here $\sigma[x/m]$ is defined assuming x is fixed and so that

$$\sigma[x/m](y) = \begin{cases} \sigma(y), y \neq x \\ m, y = x \end{cases}$$

Example: Avatar Aang's Love Life

- Suppose we wanted to compute the value of $\exists x \text{bender}(x, \text{fire}) \land \neg \text{likes}(\text{Katara}, x)$.
- We proceed as follows:

$$\mathcal{S} \models (\exists x\, (\text{bender}(x, \text{fire}) \land \neg \text{likes}(\text{Katara}, x)))\, [\sigma]$$

$\Leftrightarrow \mathcal{S} \models (\text{bender}(x, \text{fire}) \land \neg \text{likes}(\text{Katara}, x))\, [\sigma[x/m]]$ for some $m \in D$
$\Leftrightarrow \mathcal{S} \models \text{bender}(x, \text{fire})[\sigma[x/m]]$ and $\mathcal{S} \not\models \text{likes}(\text{Katara}, x)[\sigma[x/m]]$ for some $m \in D$
$\Leftrightarrow \langle m, \textbf{fire} \rangle \in \text{bender}^{\mathcal{S}}$ and $\langle \textbf{Katara}, m \rangle \notin \text{likes}^{\mathcal{S}}$ for some $m \in D$
The above holds for $m = \textbf{Azula}$.

- A **sentence** is any formula consisting of only bound variables.
- The value of a sentence, s, is independent of the variable assignments, i.e., for any σ and σ',
$$\mathcal{S} \vDash s[\sigma] \text{ if and only if } \mathcal{S} \vDash S[\sigma'].$$
- Thus, for any sentence, s, we simply write $\mathcal{S} \vDash s$ to denote that $\mathcal{S}$ satisfies s.

- Let Φ be a set of sentences.
- We say that Φ is **satisfiable** if there exists some structure $\mathcal{S}$ such that $\mathcal{S} \models s$ for every $s \in \Phi$; such a structure is called a **model** of Φ.
- Say we are given another set of sentences Φ':
  1. If every model of Φ is a model of Φ', we say that Φ' is a consequence of Φ.
  2. If no model of Φ is a model of Φ', we say that Φ' is a contradiction of Φ.
  3. If some models of Φ are models of Φ', but other models of Φ are not models of Φ', then Φ' is neither a consequence of contradiction of Φ.
- Proving logical consequences / contradictions can only be done through a syntactic reasoning mechanism.