Propositional and First-Order Logic

1 Knowledge-Based Agents

- Knowledge representation (KR) language
- Knowledge base (KB) – a set of sentences (written in the KR language) describing the agent’s world.
- An inference mechanism
  - What follows (can be deduced) from the KB and the current agent perception.
  - What actions the agent should take.
- Background knowledge – initial knowledge about the world (agent + environment).
- Levels of agent description
  - Knowledge level
  - Logical level – knowledge written in the KR language
  - Implementation level – the implementation of the KR language (compiling/running KR programs)
- Building a knowledge-based agent
  - Declarative approach
  - Learning approach
2 The Wumpus World – a simulation of a KB agent

• Environment
  – 4X4 grid of rooms (cave). Doors between the adjacent (not diagonally) rooms.
  – The agent starts in room (1,1) heading right.
  – Gold in one of the rooms (chosen randomly). If the agent finds the gold it gets 1000 points.
  – A beast (wumpus) in one of the rooms (chosen randomly). If the agent enters this room the beast eats the agent (-1000 points).
  – Pits in some rooms that trap the agent.

• Actuators (agent actions)
  – Move forward, turn 90% left and 90% right
  – Grab the gold (only if in the room with gold)
  – Exit the cave (only if in room (1,1)).

• Sensors (agent perception)
  – In a square adjacent to the wumpus the agent perceives a stench.
  – In a square adjacent to a pit the agent perceives a breeze.
  – In the square where the gold is the agent perceives a glitter.
  – If the agent walks into a wall it perceives a bump.
  – In the initial room (1,1) the agent perceives light.
  – The agent also perceives the coordinates of the room in which it is currently located.

• Goal: grab the gold and exit the cave (the exit door is in room (1,1)).
Figure 1: Wumpus World
3 First-Order Logic – alphabet

- Constants: alphanumerical strings beginning with a lower case letter (or just numbers) – $a$, $b$, $c$, $const1$, 125.
- Functions: $f$, $g$, $h$, or other constants (not numbers).
- Predicates: $p$, $q$, $r$, $father$, $mother$, $likes$, or other constants.
- Logical connectives: $\land$ (conjunction), $\lor$ (disjunction), $\neg$ (negation), $\leftrightarrow$ or $\rightarrow$ (implication) and $\leftrightarrow$ (equivalence).
- Quantifiers: $\forall$ (universal) and $\exists$ (existential).
- Punctuation symbols: (, ) and ,

4 First-Order Logic – terms

- a variable is a term;
- a constant is a term;
- if $f$ is a $n$-argument function ($n \geq 0$) and $t_1, t_2, ..., t_n$ are terms, then $f(t_1, t_2, ..., t_n)$ is a term.
5 First-Order Logic – sentences (formulas)

- if \( p \) is an \( n \)-argument predicate \( (n \geq 0) \) and \( t_1, t_2, ..., t_n \) are terms, then \( p(t_1, t_2, ..., t_n) \) is a formula (called atomic formula or atom);
- if \( F \) and \( G \) are formulas, then \( \neg F, F \land G, F \lor G, F \iff G \) are formulas too;
- if \( F \) is a formula and \( X \) – a variable, then \( \forall X F \) and \( \exists X F \) are also formulas.
- A term/formula without variables is called ground term/formula.

6 Propositional Logic - a subset of FOL

- No variables
- No functions
- Predicates are constants (0-argument predicates)

7 Propositional Logic - examples

Represent assertions that may be true or false.

- \((B \lor A) \land (\neg B \lor \neg C)\) (”Bob is a truth-teller or Amy is a truth-teller. Bob is not a truth-teller or Cal is not a truth-teller.”)
- \(\text{beast}_{31} \rightarrow \text{stench}_{32}\)
- \(\text{beast}_{31}\)
- \(at_{22} \land \neg \text{stench}_{22} \land \neg \text{breeze}_{22} \rightarrow \text{go}_-\text{forward}\)
- \(at_{22} \land \text{heading}_-\text{east} \land \text{go}_-\text{forward} \rightarrow at_{23}\)
- \(x \land \neg y \lor \neg x \land y\) (XOR)
8 First-Order Logic – examples

Represent relations between individual objects or classes of objects.

"For every man there exists a woman that he loves."
(classes of objects ⇒ variables):

∀X∃(Y \text{man}(X) \rightarrow \text{woman}(Y) \land \text{loves}(X,Y))

"John loves Mary." (concrete objects ⇒ constants):

\text{loves}(\text{john}, \text{mary})

"Every student likes every professor."

∀X∀Y(\text{is}(X, \text{student}) \land \text{is}(Y, \text{professor}) \rightarrow \text{likes}(X,Y))

Or (universal quantifiers may be skipped):

\text{is}(X, \text{student}) \land \text{is}(Y, \text{professor}) \rightarrow \text{likes}(X,Y)
9  Clausal form (CNF), Horn clauses

- Literal: an atom or its negation.
- Complementary literals: $A$ and $\neg A$.
- Clause: a disjunction of literals.
- Conjunctive Normal Form (CNF): conjunction of clauses
- Horn clause: a clause with no more than one positive literal.
- Empty clause ($\square$): a clause with no literals (logical constant "false").

10  Translating FOL into clausal form (CNF)

1. $\forall X man(X) \rightarrow \exists Y woman(Y) \land loves(X, Y)$
2. $\forall X \neg man(X) \lor \exists Y woman(Y) \land loves(X, Y)$ (removing implications)
3. $\forall X \neg man(X) \lor (woman(s(X) \land loves(X, s(X))))$ (removing the existential quantifiers – skolemization)
4. $\neg man(X) \lor (woman(s(X)) \land loves(X, s(X)))$ (removing the universal quantifiers)
5. $(\neg man(X) \lor woman(s(X)) \land (\neg man(X) \lor loves(X, s(X)))$ (conjunctive normal form)
11 Prolog notation for Horn clauses

\[ A \lor \neg B_1 \lor \neg B_2 \lor \ldots \lor \neg B_m \]
\[(p \leftarrow q = p \lor \neg q)\]

\[ A \leftarrow B_1, B_2, \ldots, B_m \]

Program clause (rule):

\[ A : -B_1, B_2, \ldots, B_m \]

Goal:

\[ : -B_1, B_2, \ldots, B_m \]

or

\[ ? - B_1, B_2, \ldots, B_m \]

Fact:

\[ A \ (\text{single atom}) \]
12 Semantics of Propositional Logic

- Logic constants: true, false ($\Box$)
- Model: truth value assignments to the predicates that make the sentence (formula) true
- Logical consequence (entailment): $P \models Q$, if every model of $P$ is also a model of $Q$.
- Valid formula (tautology): true for any truth value assignment (in every model). Example: $P \lor \neg P$
- A formula is satisfiable (consistent) if it has a model
- A formula is unsatisfiable (inconsistent) if it has no model. Example: $P \land \neg P$
- If $P \models \Box$, then $P$ is unsatisfiable.
- Deduction theorem: $P \models Q \iff P \land \neg Q \models \Box$.
- Determining satisfiability is NP-complete
13 Substitutions

\[ \theta = \{ V_1/t_1, V_2/t_2, \ldots, V_n/t_n \} \]

\[ V_i \neq V_j \; \forall i \neq j, \; t_i \neq V_i, \; i = 1, \ldots, n \]

Example:

\[ t_1 = f(a, b, g(a, b)), \; t_2 = f(A, B, g(C, D)) \]

\[ \theta = \{ A/a, B/b, C/a, D/b \} \]

\[ t_1\theta = t_2 \]

14 Term unification

\[ t_1 = f(X, b, U), \; t_2 = f(a, Y, Z) \]

Unifiers of \( t_1 \) and \( t_2 \): \( \theta_1 = \{ X/a, Y/b, Z/c, U/c \} \), \( \theta_2 = \{ X/a, Y/b, Z/U \} \)

\[ t_1\theta_1 = t_2\theta_1 = f(a, b, c) \]

\[ t_1\theta_2 = t_2\theta_2 = f(a, b, U) \]

\( \theta_2 \) is most general unifier - mgu: \( \exists \theta, \; (t_1\theta_2)\theta = t_1\theta_1 \)

\( \theta = ? \)
15 Semantics of FOL (Logic Programs)

Logic Program (LP): A set of Horn clauses.

Prolog Program: A logic program that also includes control and extra logical components: execution order and cut (!).

Herbrand base ($B_S$): Let $S$ be a set of clauses. $B_S$ is the set of all ground atoms that can be built by using predicate symbols from $S$ and arguments built by combinations of constants and functions from $S$.

Model of clause ($M_C$): $M_C$ is a model of clause $C$, if for all ground instances $C\theta$, there exists either a positive literal $P \in C$, such that $P\theta \in M_C$ or a negative literal $N \in C$, such that $N\theta \notin M_C$.

Empty clause □ has no model.

Least Herbrand model of a set of clauses $S$ ($M_S$): the intersection of all models of $S$.

Intuition:

- Express when a clause or a Logic (Prolog) program is true.
- Depends on the model (the context where the clause appears).
- This model is represented by a set of facts.
Logical consequence (entailment)

$P_1, P_2$ – logic programs.

$P_1 \models P_2$, if every model of $P_1$ is also a model of $P_2$.

$P$ is satisfiable (consistent, true), if $P$ has a model. Otherwise $P$ is unsatisfiable (inconsistent, false).

If $P \models \Box$, then $P$ is unsatisfiable.

**Deduction theorem:** $P_1 \models P_2 \iff P_1 \land \neg P_2 \models \Box$.

**Majot result in LP:** $M_P = \{ A | A$ is a ground atom, $P \models A \}$

**Undecidability of FOL:** The check for $P_1 \models P_2$ is an undesirable problem (semidecidable, i.e. not decidable only if $P_1 \not\models P_2$).

**Decidability of Datalog:** Logic programs without functions (datalog) are decidable.

**Finite/initial models of PL/FOL/Datalog** ($\{p(a), p(f(X)) \leftarrow p(X)\}$)

**How to find $M_P$ (Least Herbrand Model of $P$)?**

- Find all models of $P$ (intractable).
- Use inference rules: procedures $I$ for transforming one formula (program, clause) $P$ into another one $Q$, denoted $P \vdash_I Q$.
- $I$ is correct and complete, if $P \vdash_I P \iff P_1 \models P_2$. 

12