UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Artificial Intelligence

First-Order Logic: Inference

LESSON 16

prof. Antonino Staiano

M.Sc. In ''Machine Learning e Big Data'' - University Parthenope of Naples

The Resolution Algorithm

- Completeness theorem for predicate logic (Gödel, 1930)
  - *For every first-order sentence α entailed by a given KB* (KB $\models \alpha$) *there exists some inference algorithm that derives α* (KB $\vdash \alpha$) *in a finite number of steps*
  - The opposite does not hold
    - *Predicate logic is semi-decidable*
- A complete inference algorithm for predicate logic: Resolution (1965) based on
  - Converting sentences into Conjunctive Normal Form
  - Using only Resolution inference rule
  - Proof by contradiction
    - to prove $KB \models \alpha$, prove that $KB \land \neg \alpha$ is unsatisfiable (contradictory)
  - Refutation-completeness
    - if $KB \land \neg \alpha$ is unsatisfiable, then resolution derives a contradiction in a finite number of steps

Conjunctive Normal Form for FOL

- First step
  - Convert sentences to conjunctive normal form (CNF)
    - CNF -> conjunction of clauses
      - Each clause a disjunction of literals
        - Literals can contain variables (universally quantified)
- Example
  - $\forall x,y,z$ American(x) ∧ Weapon(y) ∧ Sells(x,y, z) ∧ Hostile(z) ⇒ Criminal(x)
    - In CNF becomes
      - ¬American(x) ∨ ¬Weapon(y) ∨ ¬Sells(x,y, z) ∨ ¬Hostile(z) ∨ Criminal(x)
- Keypoint
  - *Every sentence of FL can be converted into an inferentially equivalent CNF sentence*

Conjunctive Normal Form for FOL

- Same procedure for converting to CNF in propositional logic
  - Main difference -> eliminate existential quantifiers
- Example
  - *Everyone who loves all animals is loved by someone*
  - ∀x [∀y Animal(y) ⇒ Loves(x,y)] ⇒ [∃y *Loves(y, x)*]
    - Steps
      - Eliminate implications (replace P ⇒ Q with ¬P ∨ Q)
        - ∀x ¬[∀y Animal(y) ⇒ Loves(x,y)] ∨ [∃y *Loves(y,x)*]
        - ∀x ¬[∀y ¬Animal(x) ∨ Loves(x,y)] ∨ [∃y *Loves(y,x)*]
      - Move ¬ inwards $\neg\forall x \ p \equiv \exists x \neg p, \quad \neg\exists x \ p \equiv \forall x \neg p$
        - ∀x [∃y ¬(¬Animal(y) ∨ Loves(x,y))] ∨ [∃y *Loves(y,x)*]
        - ∀x [∃y ¬¬Animal(y) ∧ ¬Loves(x,y)] ∨ [∃y *Loves(y,x)*]
        - ∀x [∃y Animal(y) ∧ ¬Loves(x,y)] ∨ [∃y *Loves(y,x)*]
        - *Either there is some animal that x does not love, or (if it is not the case) someone loves x*

Note that ∀y in the premise of ⇒ has become an ∃y

Conjunctive Normal Form for FOL

- Standardize variables
  - Change the name of one of the variables in sentences using the variable name twice $(\exists x\, P(x)) \lor (\exists x\, Q(x))$
  - $\forall x\, [\exists y\, Animal(y) \land \neg Loves(x, y)] \lor [\exists z\, Loves(z, x)]$
- Skolemize
  - a more general form of existential instantiation, where each existential variable is replaced by a Skolem function of the enclosing universally quantified variables
  - $\forall x\, [Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x)]$
    - F and G are Skolem functions
    - The arguments of a Skolem function are all the universally quantified variables in whose scope the existential quantifiers appear
- Drop Universal quantifiers
  - At this point, all the remaining variables are universally quantified, therefore we can drop the quantifier
  - $[Animal(F(x)) \land \neg Loves(x, F(x)) \lor Loves(G(x), x)]$
- Distribute ∨ over ∧
  - $[Animal(F(x)) \lor Loves(G(x), x)] \land [\neg Loves(x, F(x)) \lor Loves(G(x), x)]$
  - CNF consisting of two clauses

Skolemization

- A more general form of Existential Instantiation must be applied when an existential quantifier appears in the scope of a universal quantifier:

$$\forall x, \ldots \exists y, \ldots \alpha[x, \ldots, y \ldots]$$

- For instance
  - from $\forall x\ \exists y\ Loves(x,y)$ (Everybody loves somebody)
  - it is not correct to derive $\forall x\ Loves(x,A)$ (Everybody loves A)
    - the latter sentence means that everybody loves the same person

Skolemization

- Instead of a constant symbol, a new function symbol, known as the Skolem function, must be introduced with as many arguments as universally quantified variables. Therefore, from:

$$\forall x,\ldots\ \exists y,\ldots\ \alpha[x,\ldots,y\ \ldots]$$

the correct application of EI derives:

$$\forall x,\ldots\ \alpha[x,\ldots,F_1(x),\ldots]$$

- For instance, from

$$\forall x\ \exists y\ \mathit{Loves}(x,y)$$

one can correctly derive

$$\forall x\ \mathit{Loves}(x,F(x))$$

where F maps any individual x to someone loved by x

The Resolution Inference Rule

- Two clauses (standardized apart) can be resolved if they contain complementary literals
  - FOL literals are complementary if one unifies with the negation of the other

$$\frac{l_1 \lor \cdots \lor l_k, \qquad m_1 \lor \cdots \lor m_n}{SUBST(\theta, \quad l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \cdots \lor l_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n)}$$

where UNIFY(l_i, $\neg m_i$) = θ

- Example

  [Animal(F(x)) ∨ Loves(G(x), x)] and [¬Loves(u,v) ∨ ¬ Kills(u, v)] becomes [Animal(F(x)) ∨ ¬ Kills(G(x), x)]
  - By eliminating the complementary literals Loves(G(x), x) and ¬Loves(u,v) with the unifier θ = {u/G(x), v/x}
- Apply resolution steps to CNF (KB ∧ ¬**α**); complete for FOL

Example Proof

- Resolution proves that (KB ∧ ¬**α**) is unsatisfiable
- Crime example
  - The sentences in CNF are
  - ¬American(x) ∨ ¬Weapon(y) ∨ ¬Sells(x,y, z) ∨ ¬Hostile(z) ∨ Criminal(x)
  - ¬Missile(x) ∨ ¬ Owns(Nono, x) ∨ ¬Sells(West, x, Nono)
  - ¬Enemy(x, America) ∨ Hostile(x)
  - ¬Missile(x) ∨ Weapon(x)
  - Owns(Nono, M1)   Missile(M1)
  - American(West)   Enemy(Nono, America)
  - We also include the negated goal ¬ Criminal(x)

Resolution Proof: Definite Clauses

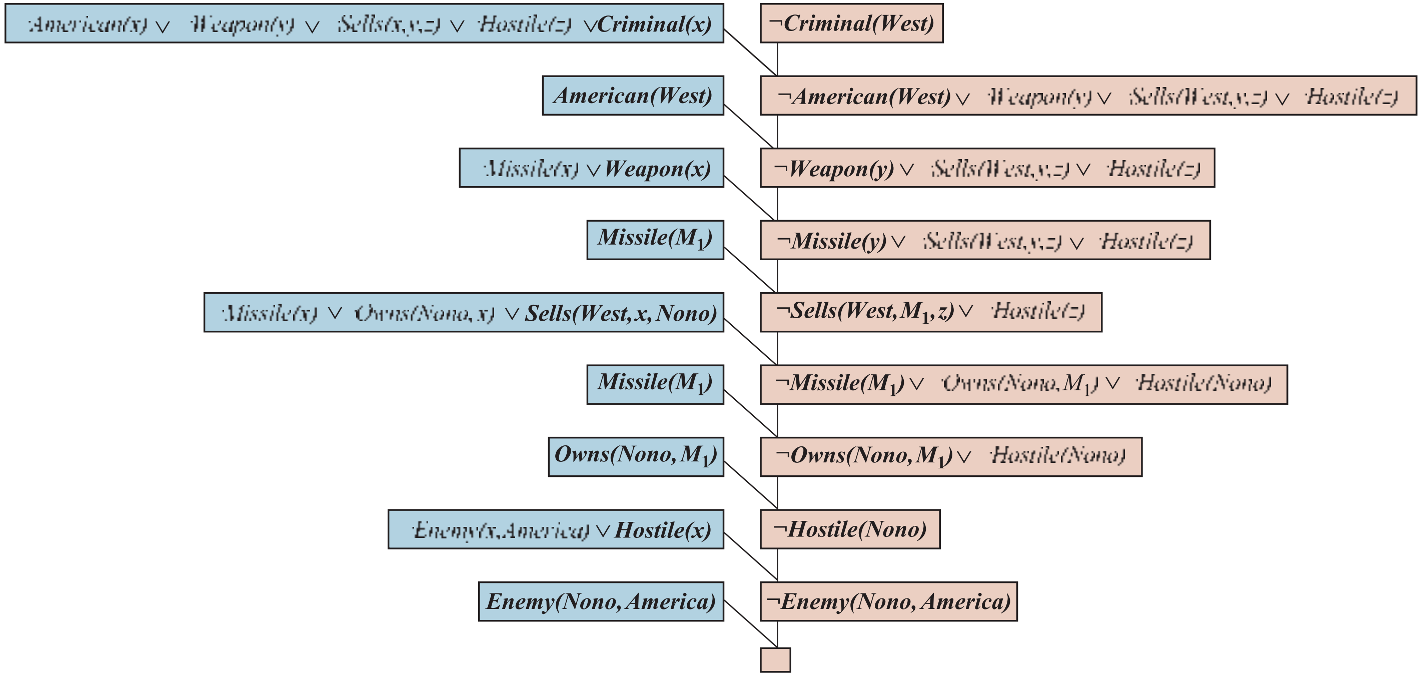

Figure 9.10 A resolution proof that West is a criminal. At each resolution step, the literals that unify are in bold and the clause with the positive literal is shaded blue.

Gödel's Incompleteness Theorem

- There are true arithmetic sentences that cannot be proved
- For any set of true sentences of number theory, and in particular any set of basic axioms, there are true sentences that cannot be proved from those axioms
- We can never prove all the theorems of mathematics within any given system of axioms

Applications of FC, BC and Resolution

- FC
  - Encoding condition-action rules to recommend actions, based on a data-driven approach
    - Production systems (production: condition-action rules)
    - Expert systems
- BC
  - Logic programming languages (e.g. Prolog), used for
    - Rapid prototyping
    - Symbol processing applications (compilers, NL parsers, …)
- Resolution
  - Main application -> theorem provers, used for
    - Assisting mathematicians
    - Proof checking
    - Verification and synthesis of hardware and software

Assignments

- Choose a topic from the list (next slide) and provide
  - Your problem specifications (by the end of the course, early June 2024)
  - The Python implementation and a Jupiter Notebook step-by-step explanation
    - 1 week ahead of the exam
- Possible libraries
  - Pygame
  - Pylogic
  - PyPlan
  - PyCogent
  - PySAT

List of projects

1. *LogicalBattleShip* (at most 10x10 grid)
2. *LogicalTict-Tac-Toe*
3. *PropostionalLogicMinesweeper* (6x6 grid, at most 8x8)
4. *First-Order Logic Wumpus World*
   - Exploring the Wumpus World
5. *FirstOrdeLogic HarryPotter World*
   - Gain knowledge from the Harry Potter Saga
6. *FOL Detective AI*
   - Imagine a spy or crime story and try to find the culprit