Artificial Intelligence Foundation – JC3001

Lecture 18: Logic Agents III

Prof. Aladdin Ayesh (aladdin.ayesh@abdn.ac.uk)

Dr. Binod Bhattarai (binod.bhattarai@abdn.ac.uk)

Dr. Gideon Ogunniye, (g.ogunniye@abdn.ac.uk)

September 2025

Material adapted from:
Russell and Norvig (AIMA Book): Chapters 7 and 9

UNIVERSITY OF ABERDEEN

Course Progression

- Part 1: Introduction
  1. Introduction to AI ✓
  2. Agents ✓
- Part 2: Problem-solving
  1. Search 1: Uninformed Search ✓
  2. Search 2: Heuristic Search ✓
  3. Search 3: Local Search
  4. Search 4: Adversarial Search ✓
- Part 3: Reasoning and Uncertainty
  1. Reasoning 1: Constraint Satisfaction ✓
  2. **Reasoning 2: Logic and Inference**
  3. Probabilistic Reasoning 1: BNs
  4. Probabilistic Reasoning 2: HMMs
- Part 4: Planning
  1. Planning 1: Intro and Formalism
  2. Planning 2: Algos and Heuristics
  3. Planning 3: Hierarchical Planning
  4. Planning 4: Stochastic Planning
- Part 5: Learning
  1. Learning 1: Intro to ML
  2. Learning 2: Regression
  3. Learning 3: Neural Networks
  4. Learning 4: Reinforcement Learning
- Part 6: Conclusion
  1. Ethical Issues in AI
  2. Conclusions and Discussion

Objectives

- Knowledge-based agents ✓
- Logic - models and entailment ✓
- Propositional and Lifted Inference
  - Resolution
  - Forward and Backward Chaining

Outline

- Propositional logic is **declarative**: pieces of syntax correspond to facts
- Propositional logic allows partial/disjunctive/negated information
  (unlike most data structures and databases)
- Propositional logic is compositional:
  meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- Meaning in propositional logic is context-independent
  (unlike natural language, where meaning depends on context)
- Propositional logic has very limited expressive power
  (unlike natural language)
  E.g., cannot say “pits cause breezes in adjacent squares”
  except by writing one sentence for each square

- Whereas propositional logic assumes world contains **facts**, first-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, theories, Felipe, colours, football games, wars, centuries . . .
  - Relations: red, round, bogus, prime, multistoried . . .,
    brother of, bigger than, inside, part of, has colour, occurred after, owns, comes between, . . .
  - Functions: father of, best friend, second half of, one more than, end of . . .

- Many statements involve **objects** and **variables**:
  - "One plus two equals three"
    Objects: one, two, three, one plus two; **Relation:** equals; **Function:** plus
  - "Squares neighbouring the wumpus are smelly"
    Objects: wumpus, squares; Property: smelly; Relation: neighboring
  - "Evil King John ruled England in 1200"
    Objects: John, England, 1200; **Relation:** ruled; **Properties:** evil, king
  - "x is greater than y"
    Variables: x, y; **Relation:** greater than

- **Constants:** KingJohn, 2, CMU, . . .
- **Predicates:** Brother, >, . . .
- **Functions:** Sqrt, LeftLegOf, . . .
- **Variables:** x, y, a, b, . . .
- **Connectives:** $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
- **Equality:** $=$
- **Quantifiers:** $\forall \exists$

- A.k.a. **FOL**, or **Predicate Logic**
- To understand FOL, we need to understand **predicates** and **quantifiers**

- In predicate logic, we use **predicate symbols** $P, Q, R \ldots$ to represent predicates
- Thus, we can use $P(x)$ to denote the statement that "the x is smelly", and $Q(x, y)$ to denote that "x ruled y"
- In this example, and our understanding of the world, we know that $P(Wumpus)$ is **true** and $Q(Felipe, England)$ is **false**

- When we assign **specific values** to its variables, a predicate has a truth-value, exactly like a proposition in propositional logic
- We say that $P(x)$ is the value of the propositional function P for x
- The number of arguments associated to a predicate symbol is its **arity**
- In formula $R(t_1, t_2, \ldots, t_n)$ we say that R is an **n-arity predicate**

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief
Fuzzy logic	facts + degree of truth	known interval value

- Sentences are true with respect to a model and an interpretation
- Model contains ≥ 1 objects (domain elements) and relations among them
- Interpretation specifies referents for
  constant symbols $\rightarrow$ objects
  predicate symbols $\rightarrow$ relations
  function symbols $\rightarrow$ functional relations
- An atomic sentence $predicate(term_1, \ldots, term_n)$ is true
  iff the objects referred to by $term_1, \ldots, term_n$
  are in the relation referred to by **predicate**

- Entailment in propositional logic can be computed by enumerating models
- We **can** enumerate the FOL models for a given KB vocabulary:
  1. For each number of domain elements **n** from **1** to ∞
     2. For each **k-ary** predicate P_k in the vocabulary
        3. For each possible **k-ary** relation on **n** objects
           4. For each constant symbol **C** in the vocabulary
              5. For each choice of referent for **C** from **n** objects . . .
- Computing entailment by enumerating FOL models is not easy!

- Process by which we can substitute variables in two FOL sentences to make them **syntactically identical**
- Examples:

$$UNIFY(Knows(John, x), Knows(John, Jane)) = \{x/Jane\}$$
$$UNIFY(Knows(John, x), Knows(y, Bill)) = \{x/Bill, y/John\}$$
$$UNIFY(Knows(John, x), Knows(y, Mother(y))) = \{y/John, x/Mother(John)\}$$
$$UNIFY(Knows(John, x), Knows(x, Elizabeth)) = failure.$$

function UNIFY(x, y, θ=*empty*) **returns** a substitution to make x and y identical, or *failure*
  if θ = *failure* **then return** *failure*
  else if $x = y$ **then return** θ
  else if VARIABLE?(x) **then return** UNIFY-VAR(x, y, θ)
  else if VARIABLE?(y) **then return** UNIFY-VAR(y, x, θ)
  else if COMPOUND?(x) **and** COMPOUND?(y) **then**
    return UNIFY(ARGS(x), ARGS(y), UNIFY(OP(x), OP(y), θ))
  else if LIST?(x) **and** LIST?(y) **then**
    return UNIFY(REST(x), REST(y), UNIFY(FIRST(x), FIRST(y), θ))
  else return *failure*

function UNIFY-VAR(var, x, θ) **returns** a substitution
  if $\{var/val\} \in \theta$ for some val **then return** UNIFY(val, x, θ)
  else if $\{x/val\} \in \theta$ for some val **then return** UNIFY(var, val, θ)
  else if OCCUR-CHECK?(var, x) **then return** *failure*
  else return add $\{var/x\}$ to θ

Generalized Modus Ponens (GMP)

$$\frac{p'_1, p'_2, \ldots, p'_n, (p_1 \wedge p_2 \wedge \cdots \wedge p_n) \Rightarrow q}{q\theta}$$

where $p'_i\theta = p_i\theta$ for all i

E.g., $\forall x(King(x) \wedge Greedy(x) \Rightarrow Evil(x))$
$King(John)$
$\forall y Greedy(y)$

p'_1 is $King(John)$ $\quad$ p_1 is $King(x)$
p'_2 is $Greedy(y)$ $\quad$ p_2 is $Greedy(x)$
θ is $\{x/John, y/John\}$ $\quad$ q is $Evil(x)$
$q\theta$ is $Evil(John)$

- GMP used with KB of definite clauses (exactly one positive literal)
- All variables assumed universally quantified

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

Can we prove that Col. West is a criminal?

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \Rightarrow Criminal(x)$

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
- Nono . . . has some missiles, i.e., $\exists x Owns(Nono, x) \wedge Missile(x)$:
  $Owns(Nono, M1)$ and $Missile(M1)$

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
- Nono . . . has some missiles, i.e., $\exists x Owns(Nono, x) \wedge Missile(x)$:
  $Owns(Nono, M1)$ and $Missile(M1)$
- . . . all of its missiles were sold to it by Colonel West
  $\forall x(Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono))$

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
- Nono . . . has some missiles, i.e., $\exists x Owns(Nono, x) \wedge Missile(x)$:
  $Owns(Nono, M1)$ and $Missile(M1)$
- . . . all of its missiles were sold to it by Colonel West
  $\forall x(Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono))$
- Missiles are weapons:
  $Missile(x) \Rightarrow Weapon(x)$

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \Rightarrow Criminal(x)$
- Nono . . . has some missiles, i.e., $\exists x Owns(Nono, x) \land Missile(x)$:
  $Owns(Nono, M1)$ and $Missile(M1)$
- . . . all of its missiles were sold to it by Colonel West
  $\forall x(Missile(x) \land Owns(Nono, x) \Rightarrow Sells(West, x, Nono))$
- Missiles are weapons:
  $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as “hostile”:
  $Enemy(x, America) \Rightarrow Hostile(x)$

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \Rightarrow Criminal(x)$
- Nono . . . has some missiles, i.e., $\exists x Owns(Nono, x) \land Missile(x)$:
  $Owns(Nono, M1)$ and $Missile(M1)$
- . . . all of its missiles were sold to it by Colonel West
  $\forall x(Missile(x) \land Owns(Nono, x) \Rightarrow Sells(West, x, Nono))$
- Missiles are weapons:
  $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as “hostile”:
  $Enemy(x, America) \Rightarrow Hostile(x)$
- West, who is American . . .
  $American(West)$

- . . . it is a crime for an American to sell weapons to hostile nations:
  $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
- Nono . . . has some missiles, i.e., $\exists x Owns(Nono, x) \wedge Missile(x)$:
  $Owns(Nono, M1)$ and $Missile(M1)$
- . . . all of its missiles were sold to it by Colonel West
  $\forall x(Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono))$
- Missiles are weapons:
  $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as “hostile”:
  $Enemy(x, America) \Rightarrow Hostile(x)$
- West, who is American . . .
  $American(West)$
- The country Nono, an enemy of America . . .
  $Enemy(Nono, America)$

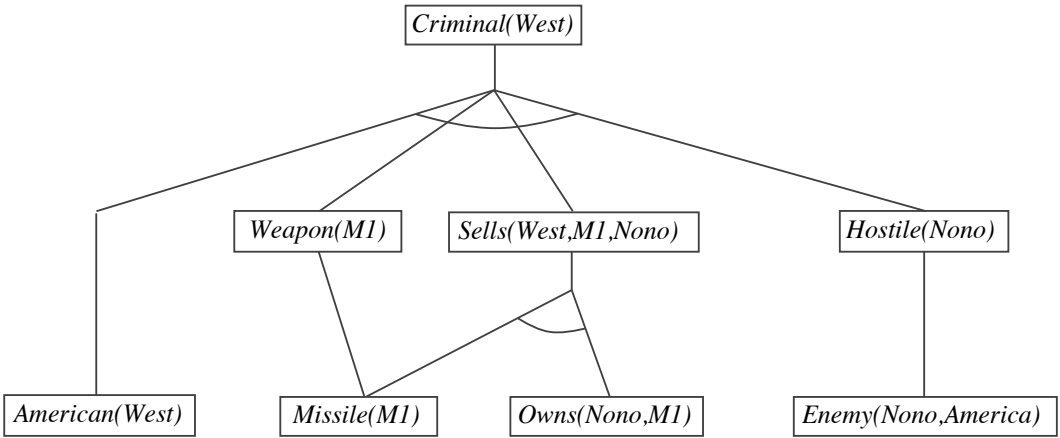
Forward Chaining Algorithm

function FOL-FC-ASK(KB, α) **returns** a substitution or *false*

inputs: KB, the knowledge base, a set of first-order definite clauses
α, the query, an atomic sentence

while *true* **do**
  $new \leftarrow \{\}$ // *The set of new sentences inferred on each iteration*
  for each $rule$ **in** KB **do**
    $(p_1 \wedge \ldots \wedge p_n \Rightarrow q) \leftarrow$ STANDARDIZE-VARIABLES($rule$)
    for each θ such that SUBST($\theta, p_1 \wedge \ldots \wedge p_n$) = SUBST($\theta, p'_1 \wedge \ldots \wedge p'_n$)
        for some $p'_1, \ldots, p'_n$ in KB
      $q' \leftarrow$ SUBST(θ, q)
      if q' does not unify with some sentence already in KB or new **then**
        add q' to new
        $\phi \leftarrow$ UNIFY(q', α)
        if ϕ is not *failure* **then return** ϕ
  if $new = \{\}$ **then return** *false*
  add new to KB

Forward Chaining Proof



- Sound and complete for first-order definite clauses
  (proof similar to propositional proof)
- **Datalog** = first-order definite clauses + **no functions** (e.g., crime KB)
  FC terminates for Datalog in poly iterations: at most $p \cdot n^k$ literals
- May not terminate in general if α is not entailed
  This is unavoidable: entailment with definite clauses is semidecidable

function FOL-BC-ASK(KB, $query$) **returns** a generator of substitutions
  return FOL-BC-OR(KB, $query$, $\{\}$)

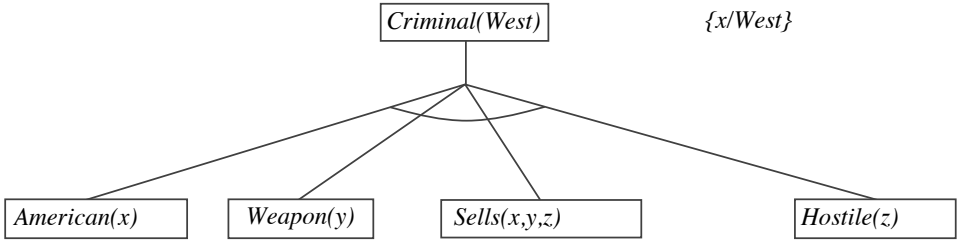
function FOL-BC-OR(KB, $goal$, θ) **returns** a substitution
  for each $rule$ in FETCH-RULES-FOR-GOAL(KB, $goal$) **do**
    $(lhs \Rightarrow rhs) \leftarrow$ STANDARDIZE-VARIABLES($rule$)
    for each θ' **in** FOL-BC-AND(KB, lhs, UNIFY(rhs, $goal$, θ)) **do**
      yield θ'

function FOL-BC-AND(KB, $goals$, θ) **returns** a substitution
  if $\theta = failure$ **then return**
  else if LENGTH($goals$) = 0 **then yield** θ
  else
    $first$,$rest \leftarrow$ FIRST($goals$), REST($goals$)
    for each θ' **in** FOL-BC-OR(KB, SUBST(θ, $first$), θ) **do**
      for each θ'' **in** FOL-BC-AND(KB, $rest$, θ') **do**
        yield θ''

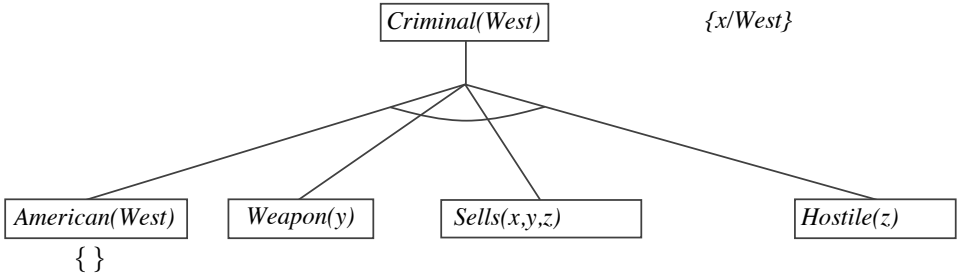
Backward Chaining Example

$$\boxed{Criminal(West)}$$

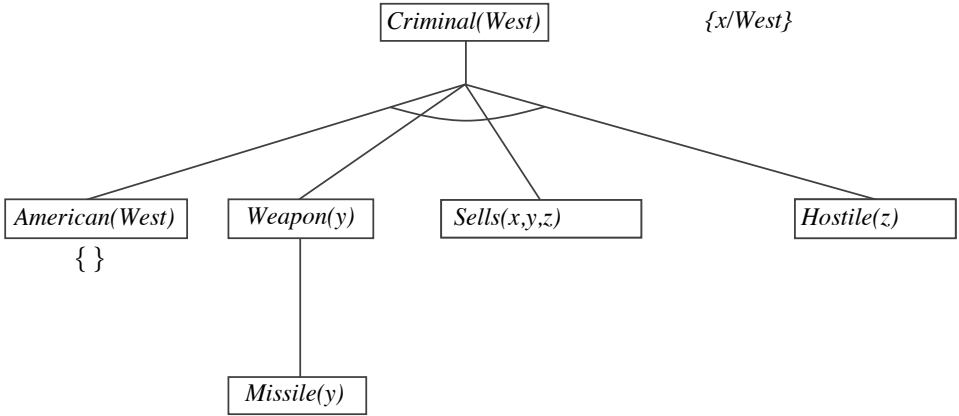
Backward Chaining Example



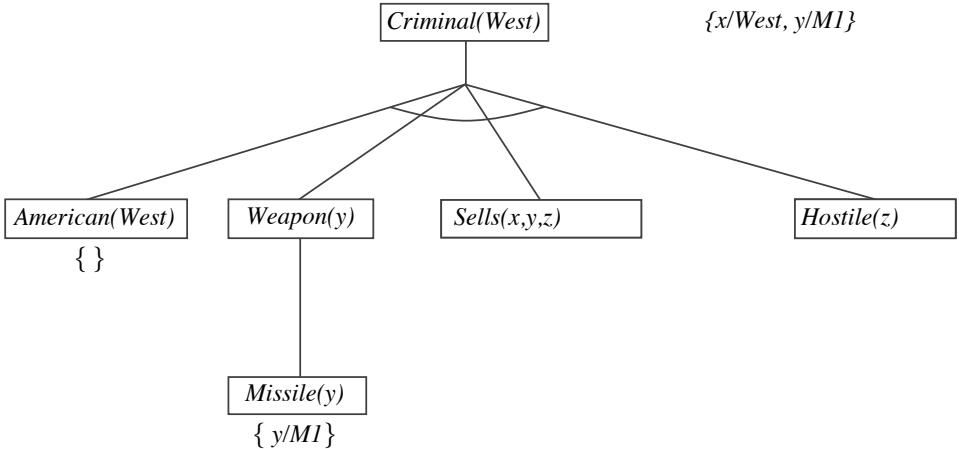
Backward Chaining Example



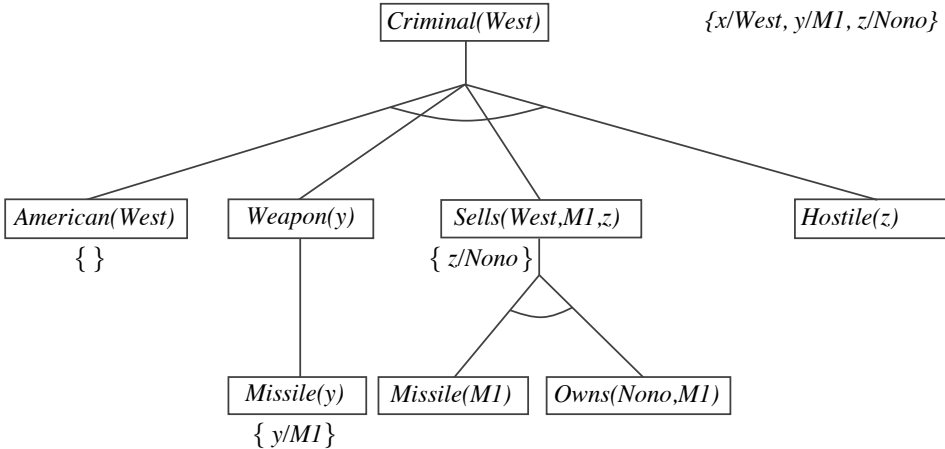
Backward Chaining Example



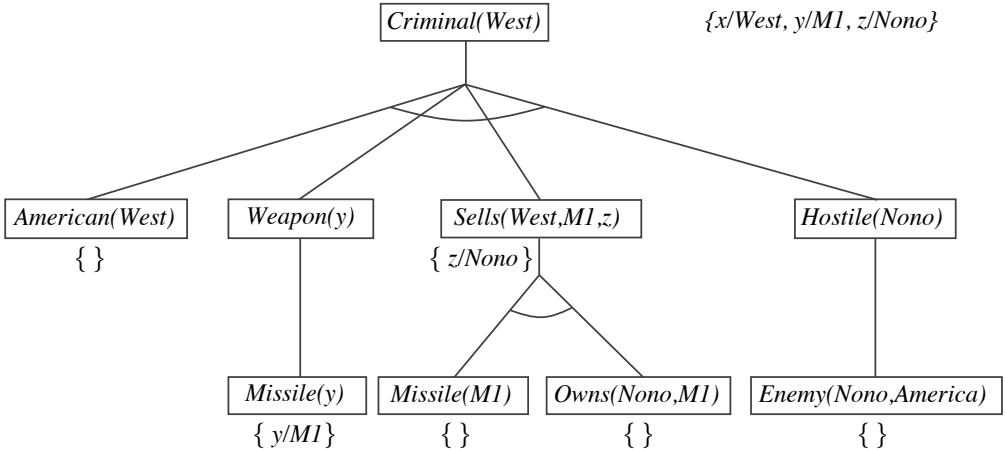
Backward Chaining Example



Backward Chaining Example



Backward Chaining Example



Properties of Backward Chaining

- Depth-first recursive proof search: space is linear in size of proof
  Incomplete due to infinite loops $\Rightarrow$ fix by checking current goal against every goal on stack
- Inefficient due to repeated subgoals (both success and failure)
  $\Rightarrow$ fix using caching of previous results (extra space!)
- Widely used (without improvements!) for
  logic programming

- Logic - models and entailment
- Propositional and Lifted Inference
  - Resolution
  - Forward and Backward Chaining

Any Questions.