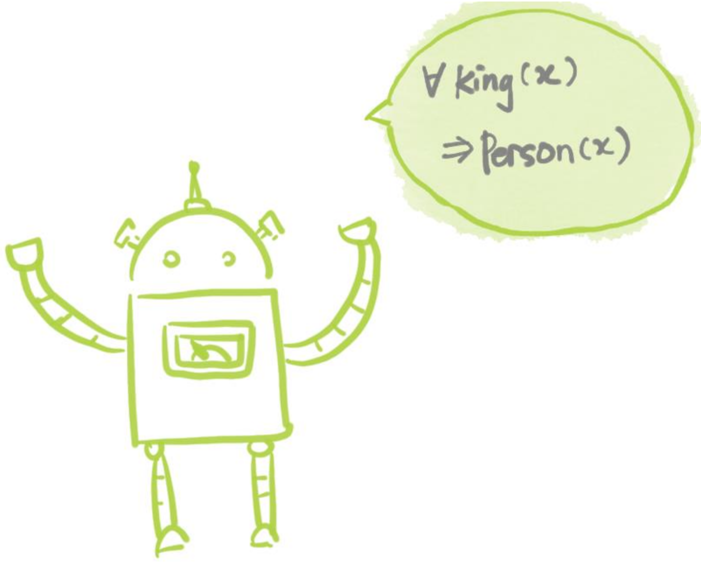
CS5491: Artificial Intelligence

First-order Logic



Instructor: Kai Wang

Recap: Propositional Logic: Syntax

- The proposition symbols P_1, P_2 etc are sentences.
- Negation: If P is a sentence, $\neg P$ is a sentence.
- Conjunction: If P_1 and P_2 are sentences, $P_1 \land P_2$ is a sentence.
- Disjunction: If P_1 and P_2 are sentences, $P_1 \lor P_2$ is a sentence.
- Implication: If P_1 and P_2 are sentences, $P_1 \Rightarrow P_2$ is a sentence.
- Biconditional: If P_1 and P_2 are sentences, $P_1 \Leftrightarrow P_2$ is a sentence.

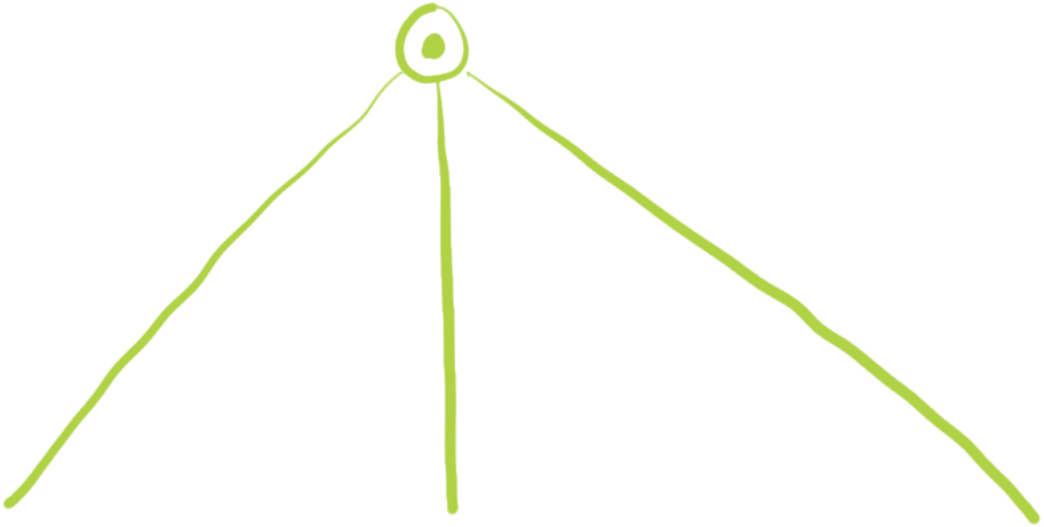
Recap: Inference

The goal of inference is to decide whether $KB \vDash \alpha$. $KB \vDash_i \alpha$ specifically says α can be derived from KB by procedure i.

Soundness: i is sound if whenever $KB \vDash_i \alpha$, it is also true that $KB \vDash \alpha$.

Completeness: i is complete if whenever $KB \vDash \alpha$, it is also true that $KB \vDash_i \alpha$

Recap: Inference Methods



Resolution

- To show $KB \vDash \alpha$, we show that $KB \wedge \neg\alpha$ is not satisfiable.

- Conjunctive Normal Form: conjunction of disjunctions of literals
  - E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

- Apply resolution to $KB \wedge \neg\alpha$ in CNF

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

  where l_i and m_j are complementary literals.

  until
  - there are no new clauses to be added.
  - two clauses resolve to the empty class, which means $KB \vDash \alpha$.

Resolution

function PL-RESOLUTION(*KB*, α) **returns** *true* or *false*
  inputs: *KB*, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

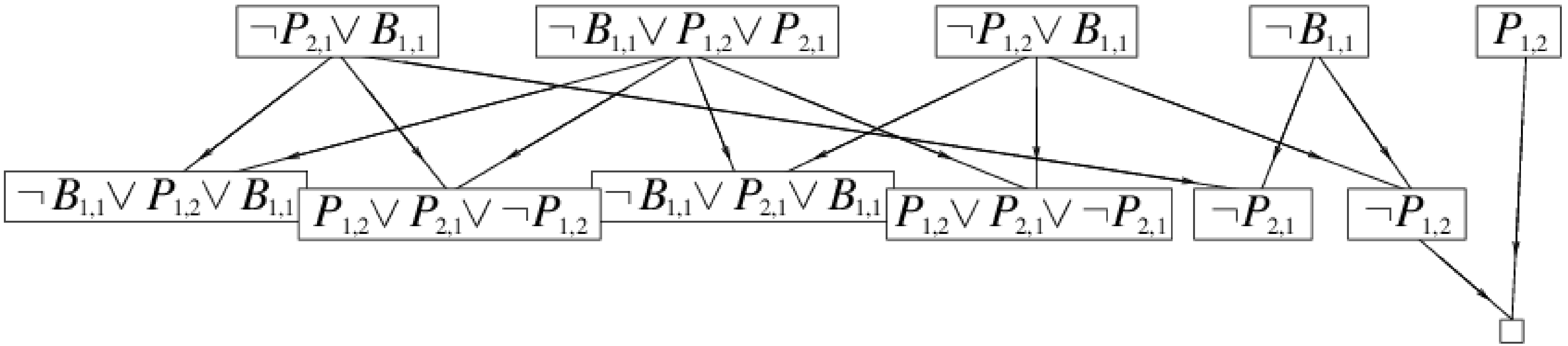
  clauses ← the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  new ← { }
  loop do
    for each C_i, C_j **in** *clauses* **do**
      resolvents ← PL-RESOLVE(C_i, C_j)
      if *resolvents* contains the empty clause **then return** *true*
      new ← *new* ∪ *resolvents*
    if *new* ⊆ *clauses* **then return** *false*
    clauses ← *clauses* ∪ *new*

Resolution in Wumpus

- We take a subset of the knowledge base, say

$$KB = R_2 \wedge R_4 = \left(B_{1,1} \Leftrightarrow \left(P_{1,2} \vee P_{2,1}\right)\right) \wedge \neg B_{1,1}$$

$$KB \wedge \neg\alpha = \left(\neg P_{2,1} \vee B_{1,1}\right) \wedge \left(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}\right) \wedge \left(\neg P_{1,2} \vee B_{1,1}\right) \wedge \left(\neg B_{1,1}\right) \wedge \left(P_{1,2}\right)$$

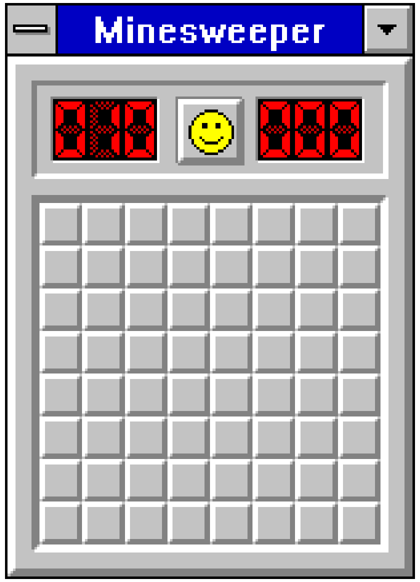


Problems with Propositional Logic

- Consider the game "minesweeper" on a 10x10 field with only one landmine.
- How do you express the knowledge, with propositional logic, that the squares adjacent to the landmine will display the number 1?
- Intuitively with a rule like

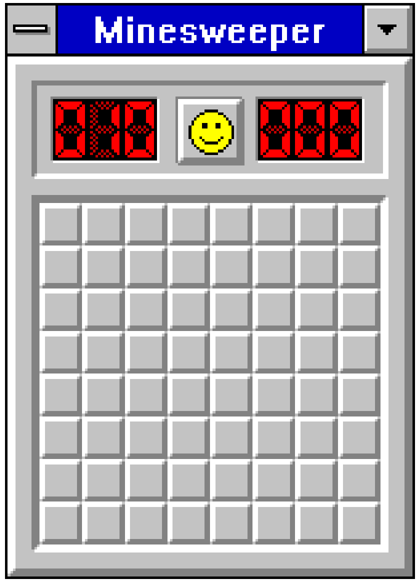
  landmine(x,y)⇒ number 1((neighbors(x,y)))

  but propositional logic cannot do this…



Problems with Propositional Logic

- Propositional logic has to say, e.g., for grid (3,4):
  - Landmine(3,4) ⇒ number 1(2,3)
  - Landmine(3,4) ⇒ number 1(2,4)
  - Landmine(3,4) ⇒ number 1(2,5)
  - Landmine(3,4) ⇒ number 1(3,3)
  - Landmine(3,4) ⇒ number 1(3,5)
  - Landmine(3,4) ⇒ number 1(4,3)
  - Landmine(3,4) ⇒ number 1(4,4)
  - Landmine(3,4) ⇒ number 1(4,5)
- Difficult to express large domains concisely.
- Do not have objects and relations.



More Logics

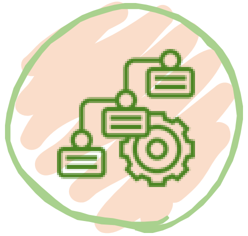
Logic	Primitives	Available knowledge
propositional	facts	true/false/unknown
first-order	facts, objects, relations	true/false/unknown
temporal	facts, objects, relations, times	true/false/unknown
probabilistic theory	facts	degree of belief 0,…,1
fuzzy logic	facts + degree of truth	known internal value

Today

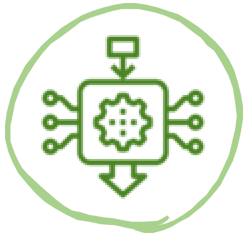
First-order logic syntax

Inference in first-order logic

Today



First-order logic syntax



Inference in first-order logic

First-Order Logic: Syntax

- Constant symbols (i.e., the individuals in the world): Jerry, 2, Green
- Function symbols (mapping individuals to individuals): Sqrt(9), Distance(Madison, Chicago)
- Predicate symbols (mapping from individuals to truth values): Teacher(Jerry, you), Bigger(sqrt(2), x)

- Variable symbols: x, y
- Connectives: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
- Quantifiers: $\forall, \exists$

First-Order Logic: Term

A term is an object in the world.

- Constant: Jerry, 2, Green
- Variables: x, y, a, b, c
- Function($term_1$, …, $term_n$)
  - Sqrt(9), Distance(Madison, Chicago)
  - Maps one or more objects to another object
  - Can refer to an unnamed object: LeftLeg(John)
  - Represents a user defined functional relation

First-Order Logic: Atom

An atom is the smallest true/false expression

- Predicate($term_1$, …, $term_n$)
  - Teacher(Jerry, you), Bigger(sqrt(2), x)
  - Convention: read “Jerry (is) Teacher (of) you”
  - Maps one or more objects to a truth value
  - Represents a user defined relation
- $Term_1$ = $term_n$
  - Radius(Earth)=6400km, 1=2
  - Represents the equality relation when two terms refer to the same

First-Order Logic: Sentence

A sentence is a true/false expression

- Atom
- Complex sentence using connectives: $\land, \lor, \neg, \Rightarrow, \Leftrightarrow$
  - Spouse(Jerry, Jing) $\Rightarrow$ Spouse(Jing, Jerry)
  - Less(11,22) $\land$ Less(22,33)
- Complex sentence using quantifiers $\forall, \exists$

First-Order Logic: Universal Quantifier

- A sentence is true for all values of x in the domain of variable x.

- Main connective typically is ⇒
  - Forms "if-then" rules
  - "all humans are mammals"

$$\forall x \;\; human(x) \Rightarrow mammal(x)$$

  - Means if x is a human, then x is a mammal.

First-Order Logic: Universal Quantifier

$\forall x \ \ human(x) \Rightarrow mammal(x)$

- It is a big AND: equivalent to the conjunction of all the instantiations of variable x:

  $(human(Jerry) \Rightarrow mammal(Jerry)) \land (human(Jing) \Rightarrow mammal(Jing)) \land \ldots$

- Common mistake is to use $\land$ as the main connective

  $\forall x \ \ human(x) \land mammal(x)$

- This means that everything is human and a mammal!

  $(human(Jerry) \land mammal(Jerry)) \land (human(Jing) \land mammal(Jing)) \land \ldots$

First-Order Logic: Existential Quantifier

- A sentence is true for some value of x in the domain of variable x.

- Main connective typically is $\wedge$
  - "some humans are male"

    $$\exists x \; human(x) \wedge male(x)$$

  - Means there is an x who is a human and is a male

First-Order Logic: Existential Quantifier

$$\exists x \ \ human(x) \land male(x)$$

- It is a big OR: equivalent to the disjunction of all the instantiations of variable x:

  (human(Jerry) ∧ male(Jerry)) ∨ (human(Jing) ∧ male(Jing)) ∨ …

- Common mistake is to use ⇒ as the main connective
  - "some pig can fly"

$$\exists x \ \ pig(x) \Rightarrow fly(x)$$

- This means that there is something not a pig!

  (pig(Jerry) ⇒ fly(Jerry)) ∨ (pig(Jing) ⇒ fly(Jing)) ∨ …

First-Order Logic: Quantifier Properties

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$
- Example:
  - $\forall x\ \forall y$ likes(x,y) meaning that “Everyone likes everyone”.
  - $\forall y\ \forall x$ likes(x,y) meaning that “Everyone is liked by everyone”.

First-Order Logic: Quantifier Properties

- $\forall x\ \exists y$ is the not the same as $\exists y\ \forall x$
- $\exists x\ \forall y$ is the same as $\forall y\ \exists x$
- Example:
  - $\forall x\ \exists y$ likes(x,y) meaning that “Everyone likes someone (can be different)”.
  - $\exists y\ \forall x$ likes(x,y) meaning that “There is someone who is liked by everyone”.

First-Order Logic: Quantifier Properties

- $\forall x\ P(x)$ when negated becomes $\exists x\ \neg P(x)$
- $\exists x\ P(x)$ when negated becomes $\forall x\ \neg P(x)$
- Example:
  - $\forall x$ sleep(x) meaning that "Everyone sleeps".
  - $\exists x\ \neg$sleep(x) meaning that "There is someone who does not sleep".

First-Order Logic: Quantifier Properties

- $\forall x\, P(x)$ is the same as $\neg\exists x\, \neg P(x)$
- $\exists x\, P(x)$ is the same as $\neg\forall x\, \neg P(x)$
- Example:
  - $\forall x$ sleep(x) meaning that “Everyone sleeps”.
  - $\neg\exists x\, \neg$sleep(x) meaning that “There does not exist someone being not asleep”.

Thinking in Logical Sentences

- $King(Richard) \lor King(John)$
- $\neg Brother(LeftLeg(Richard), John)$
- $\forall x\ \forall y\ Brother(x,y) \Rightarrow Sibling(x,y)$
- $In(Paris, France) \land In(Marseilles, France)$
- $\forall c\ Country(c) \land Border(c,Ecuador) \Rightarrow In(c, SouthAmerica)$
- $\forall c\ Country(c) \land Border(c, Spain) \land Border(c, Italy)$

Thinking in Logical Sentences

- Richard has only two brothers, John and Geoffrey.

$Brother(John, Richard) \land Brother(Geoffrey, Richard) \land John \neq Geoffrey \land \forall x\ Brother(x, Richard) \Rightarrow (x{=}John \lor x{=}Geoffrey)$

Thinking in Logical Sentences

- No region in South America borders any region in Europe.

$\forall c,d\ In(c, SouthAmerica) \land In(d, Europe) \Rightarrow \neg Border(c,d)$

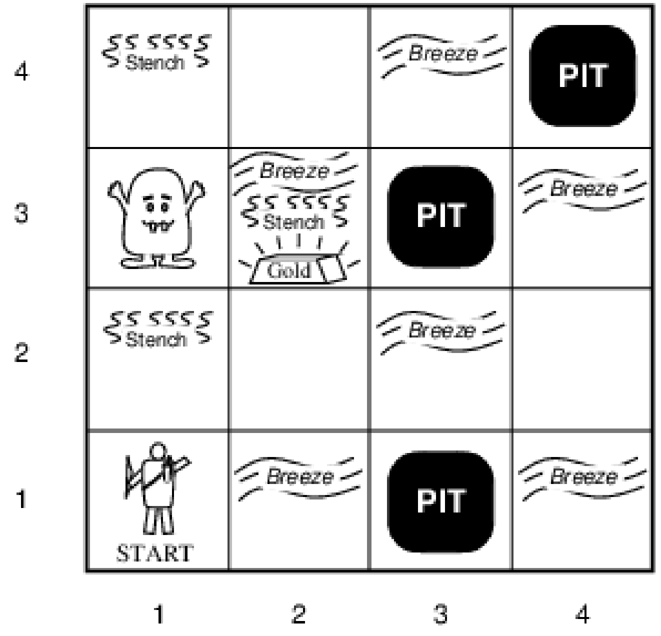
Thinking in Logical Sentences

Clicker question: No two adjacent countries have the same map color.

A. $\exists x\ \exists y\ \text{Country}(x) \wedge \text{Country}(y) \wedge \text{Border}(x,y) \wedge \neg(\text{Color}(x)=\text{Color}(y)) \wedge \neg(x=y)$
B. $\forall x\ \forall y\ \text{Country}(x) \wedge \text{Country}(y) \wedge \text{Border}(x,y) \Rightarrow \neg(\text{Color}(x)=\text{Color}(y))$
C. $\forall x\ \forall y\ \text{Country}(x) \wedge \text{Country}(y) \wedge \text{Border}(x,y) \Rightarrow \neg(\text{Color}(x)=\text{Color}(y)) \wedge \neg(x=y)$
D. $\forall x\ \forall y\ \text{Country}(x) \wedge \text{Country}(y) \wedge (x \neq y) \wedge \text{Border}(x,y) \Rightarrow \neg(\text{Color}(x)=\text{Color}(y))$

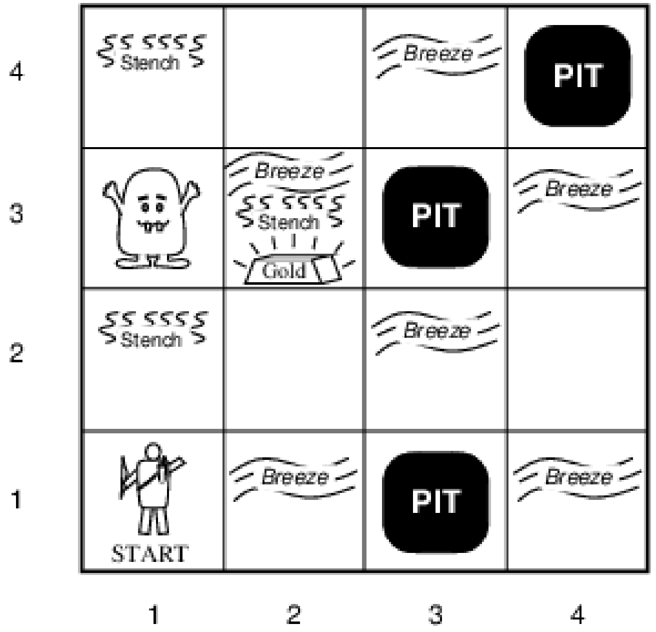
Wumpus in First-Order Logic

- Can include the time domain
  - at time step 4: Percept([Stench, Breeze, Glitter], 4)
  - at time step 6: Percept([None, Breeze, None], 6)
  - Actions can be: Turn(Right), Turn(Left), Forward, Shoot



Wumpus in First-Order Logic

- Can encode complex rules
  - $\forall$t, s, g, m, c Percept([s,b,Glitter,m,c],t) $\Rightarrow$ Glitter(t)
  - $\forall$t Glitter(t) $\Rightarrow$ BestAction(Grab, t)
  - $\forall$x, y, a, b Adjacent([x,y], [a,b]) $\Leftrightarrow$ (x=a $\wedge$ (y=b-1 $\vee$ y=b+1)) $\vee$ (y=b $\wedge$ (x=a-1 $\vee$ x=a+1))



Today

First-order logic syntax

Inference in first-order logic

Inference Rules

- Inference rules for the propositional logic:
  - Modus ponens

$$\frac{A \Rightarrow B, A}{B}$$

  - Resolution

$$\frac{A \vee B, \neg B \vee C}{A \vee C}$$

- Additional inference rules are needed for sentences with quantifiers and variables.

Variable Substitutions

- Variable in sentences can be substituted with terms.

- Substitution is a mapping from variables to terms.
  - $\{x_1/t_1, x_2/t_2 \cdots\}$

- Example:
  - SUBST({x / Sam, y / Pam}, likes(x,y)) = like(Sam, Pam)
  - SUBST({x / z, y / fatherof(John)}, likes(x,y)) = like(z, fatherof(John))

Inference Rules for Quantifiers

- Universal elimination: substituting a variable with a constant

$$\frac{\forall x\ \phi(x)}{\phi(a)}$$

- Example:
  - ∀x Likes(x, IceCream)

    Likes(Ben, IceCream)

Inference Rules for Quantifiers

- Existential elimination: substituting a variable with a constant that does not appear elsewhere in the KB, i.e., Skolem constant.

$$\frac{\exists x\ \phi(x)}{\phi(a)}$$

- Example:
  - $\exists x$ Kill(x, Victim)

    Kill(Murderer, Victim)
  - $\exists x$ Crown(x) $\wedge$ OnHead(x, John)

    Crown(C_1) $\wedge$ OnHead(C_1 , John)

Propositionalization

- Suppose the KB contains just the following:
  - $\forall x\ King(x) \land Greedy(x) \Rightarrow Evil(x)$
  - King(John)
  - Greedy(John)
  - Brother(Richard, John)

- Instantiating the universal sentence in all possible ways:
  - $King(John) \land Greedy(John) \Rightarrow Evil(John)$
  - $King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard)$
  - King(John)
  - Greedy(John)
  - Brother(Richard, John)

Problems with Propositionalization

- Propositionalization generates lots of irrelevant sentences.
- With p k-ary predicates and n constants, there are pn^k instantiations.
- Example:
  - ∀x King(x) ∧ Greedy(x) ⇒ Evil(x)
  - King(John)
  - ∀y Greedy(y)
  - Brother(Richard, John)

Unification

Unification takes two similar sentences and computes the substitution that makes them look the same, if it exists

$$UNIFY(p, q) = \sigma \quad s.t.\ SUBST(\sigma, p) = SUBST(\sigma, q)$$

- Example:
  - *UNIFY*(Knows(John, x), Knows(John, Jane)) = {x / Jane}
  - *UNIFY*(Knows(John, x), Knows(y, Ann)) = {x / Ann, y / John}
  - *UNIFY*(Knows(John, x), Knows(y, MotherOf(y))) = {x / MotherOf(John), y / John}
  - *UNIFY*(Knows(John, x), Knows(x, Elizabeth)) = fail

Generalized Modus Ponens

- If there exists a substitution σ such that $SUBST(\sigma, A_i) = SUBST(\sigma, A_i')$ for all i=1,2,…,n, then

$$\frac{A_1 \wedge A_2 \wedge \cdots A_n \Rightarrow B, A_1' \wedge A_2' \wedge \cdots A_n'}{SUBST(\sigma, B)}$$

- Example:
  - ∀x King(x) ∧ Greedy(x) ⇒ Evil(x)
  - King(John)
  - ∀y Greedy(y)
  - Brother(Richard, John)

A_1 = King(x), A_1' = King(John)
A_2 = Greedy(x), A_2' = Greedy(y)
σ={x / John, y / John} , B = Evil(x)
$SUBST(\sigma, B)$= Evil(John)

Generalized Resolution Rule

- If the substitution is computed without failure, i.e., $\sigma = UNIFY\left(\phi_i, \neg\psi_j\right) \neq fail$,

$$\frac{\phi_1 \vee \phi_2 \vee \cdots \phi_k, \psi_1 \vee \psi_2 \vee \cdots \psi_n}{SUBST(\sigma, \phi_1 \vee \cdots \vee \phi_{i-1} \vee \phi_{i+1} \cdots \vee \phi_k, \vee \psi_1 \vee \cdots \vee \psi_{j-1} \vee \psi_{j+1} \cdots \psi_n)}$$

- Example:

$$\rightarrow \quad \frac{P(x) \vee Q(x), \neg Q(John) \vee S(y)}{P(John) \vee S(y)}$$

Inference with Resolution Rule

- Proof by contration
  - Prove that KB, $\neg\alpha$ is unsatisfiable.
- Main procedures:
  - Convert KB, $\neg\alpha$ to CNF with ground terms and universal variables only.
  - Apply repeatedly the resolution rule while keeping track of the consistency of substitutions.
  - Stop when empty set (contradiction) is derived or no more new resolvents (conclusions) follow.

Conversion to CNF

- Eliminate implications and logical equivalences
  - $(p \Rightarrow q) \rightarrow (\neg p \vee q)$
- Move negations inside
  - $\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$, $\neg(p \vee q) \rightarrow (\neg p \wedge \neg q)$
  - $\neg\forall xp \rightarrow \exists x\neg p, \neg\exists xp \rightarrow \forall x\neg p$
  - $\neg\neg p \rightarrow p$
- Standardize variables
  - $(\forall x\, P(x)) \vee (\exists x\, Q(x)) \rightarrow (\forall x\, P(x)) \vee (\exists y\, Q(y))$

Conversion to CNF

- Move all quantifiers left
  - $\left(\forall x\, P(x)\right) \vee \left(\exists x\, Q(x)\right) \rightarrow \forall x \exists y P(x) \vee Q(y)$
- Skolemization
  - $\exists y\, P(A) \vee Q(y) \rightarrow P(A) \vee Q(B)$
  - $\forall x \exists y\, P(x) \wedge Q(y) \rightarrow \forall x\, P(x) \wedge Q(F(x))$
- Drop universal quantifiers
  - $\forall x\, P(x) \vee Q\left(F(x)\right) \rightarrow P(x) \vee Q(F(x))$
- Convert to CNF using the distribution laws.
  - $p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \wedge r)$

Example

- Suppose the KB contains just the following:

  - John like all kinds of food.
  - Apples are food.
  - Anything anyone eats and isn’t killed by is food.
  - Bill eats peanuts and is still alive.
  - Sue eats everything Bill eats.

Prove that “John likes peanuts”.

Example

- Represent the KB with first-order logic sentences.

- John like all kinds of food.
- Apples are food.
- Anything anyone eats and isn't killed by is food.
- Bill eats peanuts and is still alive.
- Sue eats everything Bill eats.

- $\forall x\ (Food(x) \Rightarrow Like(John, x))$
- $Food(Apple)$
- $\forall x\ \forall y\ (Eat(x,y) \land \neg Killed\ by(x,y)) \Rightarrow Food(y)$
- $Eat(Bill, penut) \land \neg Killed\ by(Bill,peanut)$
- $\forall x\ (Eat(Bill,x) \Rightarrow Eat(Sue,x))$

Example

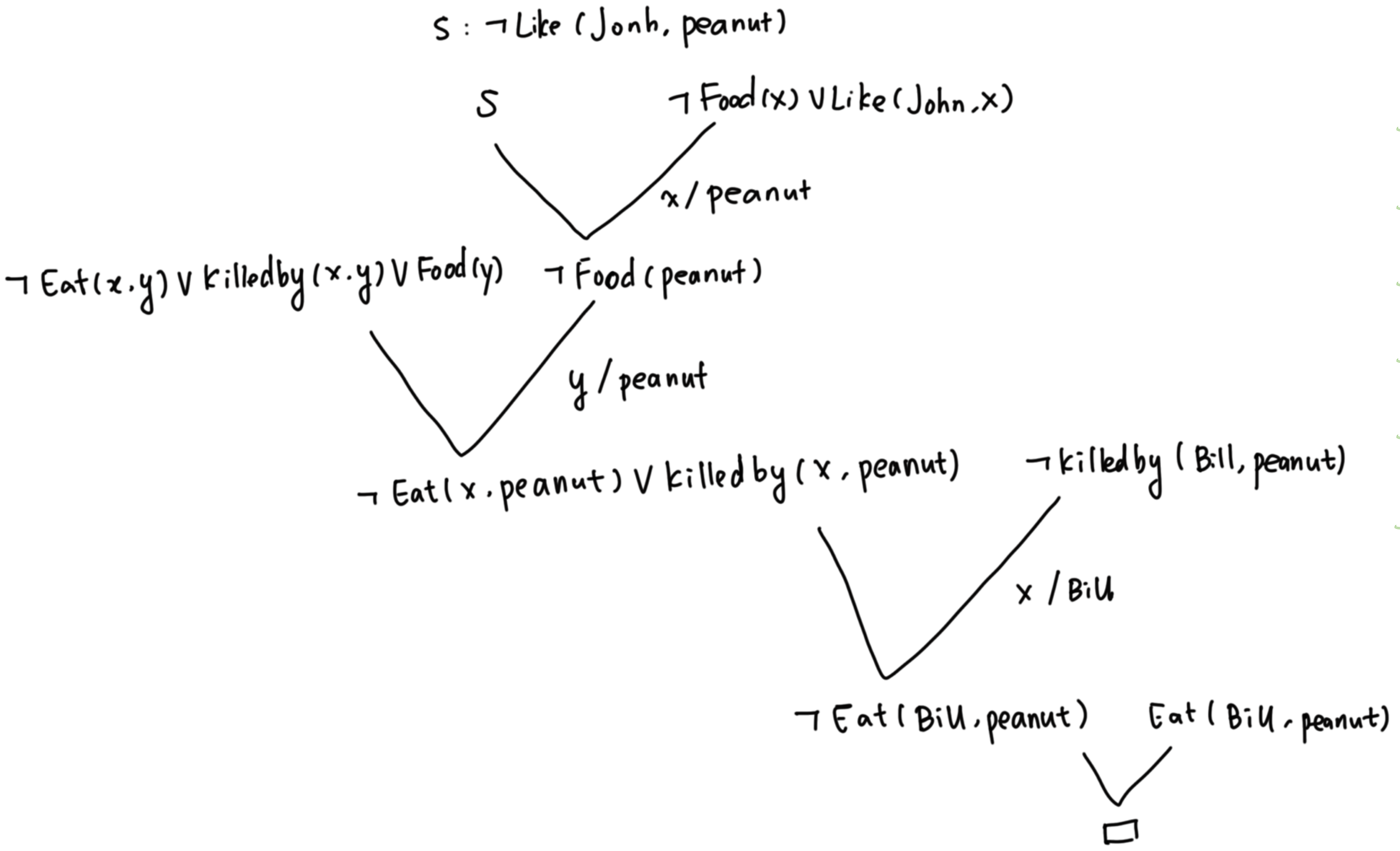
- Convert the FOL sentences into the CNF form.

- $\forall x\ (Food(x) \Rightarrow Like(John, x))$
- $Food(Apple)$
- $\forall x\ \forall y\ (Eat(x,y) \land \neg Killed\ by(x,y)) \Rightarrow Food(y)$
- $Eat(Bill, penut) \land \neg Killed\ by(Bill,peanut)$
- $\forall x\ (Eat(Bill,x) \Rightarrow Eat(Sue,x))$

- $\neg Food(x) \lor Like(John, x)$
- $Food(Apple)$
- $\neg Eat(x,y) \lor Killed\ by(x,y)) \lor Food(y)$
- $Eat(Bill, penut)$
- $\neg Killed\ by(Bill,peanut)$
- $\neg Eat(Bill,x) \lor Eat(Sue,x)$

Example

- Resolution



- ¬Food(x) ∨ Like(John, x)
- Food(Apple)
- ¬Eat(x,y) ∨ Killed by(x,y)) ∨ Food(y)
- Eat(Bill, penut)
- ¬Killed by(Bill,peanut)
- ¬Eat(Bill,x) ∨ Eat(Sue,x)

Goals

- Understand the downsides of propositional logic.
- Understand the syntax and semantic of first-order logic.
- Learn to formulate the first-order logic for real-world problems.
- Understand the generalized inference rules for first-order logic.
- Know how to implement the forward / backward chaining and resolution.

Important This Week

- Do more exercises in Chapter 8 in the textbook.