failure.
the results.
...ive step, return

Date 3/12/24
Page 19

LAB - 08

Project title :- Create a knowledge base consisting of FOL logic statement and prove the given query using forward reasoning.

Forward Reasoning Algorithm :-

function FOL-FC-ASK (KB, α) returns a substitution or false
inputs: KB, knowlegde base, a set of first or definite clauses, α, the query, an atomic sentences.
local variables: new, the new sentence infered on each iteration

repeat until new in empty
 new ← { }
 for each rule in KB do
  ($p_1 \wedge \ldots p_n \Rightarrow q$) ← STANDARDIZE - VARIABLES (rules)
  for each θ such that SUBST (θ, $p_1 \wedge \ldots \wedge p_n$)
   = SUBST (θ, $p_1' \wedge \ldots \wedge p_n'$)
 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 fail then return ϕ
 add new to KB
return false.

PTO

Given case study :-

As per the law, it is a crime for an American to sell weapon to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert who is an American citizen. Prove that 'Robert is criminal'.

Representation in FOL

1. It is a crime for an american to sell weapon to hostile nation.
Lets say p, q, and r are variables.
American(p) ∧ Weapon(q) ∧ Sells(p, q, r) ∧ Hostile(r) ⇒ Criminal(p)

2. Country A has some missile.
∃x Owns(A, x) ∧ Missile(x).

Existential instantiation, introducing a new variable T_1
Owns(A, T_1)
Missile(T_1)

3. All of the missiles ~~was~~ were sold to country A by Robert
∀x Missile(x) ∧ Owns(A, x) ⇒ Sells(Robert, x, A)

4. Missiles are weapons
Missile(x) ⇒ Weapon(x)

5. Enemy of America is known as hostile
∀x Enemy(x, America) ⇒ Hostile(x)

6. Robert is an American
American(Robert)

7. The country A, an enemy of America
Enemy(A, America).

output :
gs 'Crimin

forward c

American

Namratha. M.
3/12/2024

r on American to
untry A, an enemy of
all the missiles were
rical citizen.

to sell weapon to

Hostile (r) =)
riminal (p)

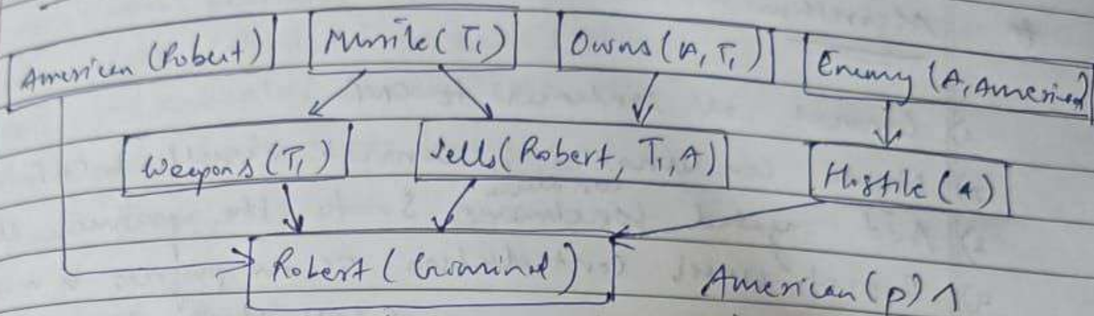
w variable T_1

ntry A by Robert
bert, x, A)

output:

q, 'Criminal (Robert)' provable? True.

forward chaining proof



American(p) $\wedge$
Weapon(q) $\wedge$ sells(p,q,
$\wedge$ Hostiles =) Criminal(p)

Namrata. M.
3/12/2024

LAB-08

Project title :- Create a knowledge base of first order logic and solve query using resolution

Algorithm :-

1) Convert all sentences to CNF
2) Negate conclusion S & convert result to CNF
3) Add negated ~~conclusion~~ condition S to the premise clauses
4) Repeat until contradiction or no progress is made
   a) ~~Resolv~~ Select 2 clauses (call them parent clauses)
   b) Resolve them together, performing all required unification
   c) If resolvent is the empty clause, a contradiction has been found.
   d) If not, add resolvent to the premises. If we succeed in step 4, we have proved the ~~conclusion~~ condition

Proof by Resolution :

Given the KB or Premise

a) John likes all kind of food
b) Apple & vegetables are food
c) Anything anyone eats and not killed is food
d) Anil eats peanuts and still Alive
e) Harry eats anything that anil eats.
f) Anyone who is alive implies not killed
g) Anyone who is alive implies not killed
h) John likes peanuts

=> Representation in FOL

a) $\forall x$: food(x) → likes(John, x)
b) food(Apple) ∧ food(vegetables)
c) $\forall x \forall y$: eats(x, y) ∧ ¬ killed(x) → food(y)
d) eats(Anil, peanuts) ∧ alive(Anil)
e) $\forall x$: eats(Anil, x) → eats(Harry, x)
f) $\forall x$: ¬ killed(x) → alive(x)
g) $\forall x$: alive(x) → ¬ killed(x)
h) likes(John, Peanuts)

Eliminate Implications

a) $\forall x$ ¬ food(x) ∨ likes(John, x)
c) $\forall x \forall y$ ¬ [eats(x, y) ∧ ¬ killed(x)] ∨ food(y)
e) $\forall x$ ¬ eats(Anil, x) ∨ eats(Harry, x)
f) $\forall x$ ¬ [¬ killed(x)] ∨ alive(x)
g) $\forall x$ ¬ alive(x) ∨ ¬ Killed(x)

→ Move negation (¬) inwards

c) $\forall x \forall y$ ¬ eats(x, y) ∨ killed(x) ∨ food(y)
f) $\forall x$ killed(x)] ∨ alive(x)

→ standarize variables

c) $\forall y \forall z$ ¬ eats(y, z) ∨ killed(y) ∨ food(z)
e) $\forall w$ ¬ eats(Anil, w) ∨ eats(Harry, w)
f) $\forall g$ killed(g)] ∨ alive(g)
g) $\forall k$ ¬ alive(k) ∨ ¬ killed(k)

→ Drop universal

a) ¬ food(x) ∨ likes(John, x)
b) food(Apple)
c) food(vegetables)

c) food (vegetables)
d) ¬ eats (Y, Z) ∨ killed (Y) ∨ food (Z)
e) eats (Anil, Peanuts)
f) alive (Anil)
g) ¬ eats (Anil, w) ∨ eats (Harry, w)
h) killed (y) ∨ alive (y)
i) ¬ alive (x) ∨ ¬ killed (K)

Proof :-

¬ likes (John, Peanuts) ¬ food (x) ∨ likes (John, x)
{Peanuts / x}

¬ food (Peanuts) ¬ eats (y, z) ∨ killed (y) ∨ food (z) {Peanuts / z}

¬ eats (y, Peanuts) ∨ killed (y) eats (Anil, Peanuts)
{Anil / y}

killed (Anil) ¬ alive (x) ∨ ¬ killed (x)
{Anil / x}

¬ alive (Anil) alive (Anil)

{ } Hence Proved

Project ...

Algorithm

↳ Alpha (α ...
path with ...
↳ Max wo...
bound de...
↳ In Both ...
where co...
↳ Both ...
same p...
↳ Alpha (α ...
less time ...

Problem.
Apply ...
node (H...