

17<sup>th</sup> December, 2024. Tuesday Lab no 10 - 80

1) Consider the following English statements

1. If someone suffers from allergies, then he / she sneezes
2. If someone lives with a cat and is allergic to cat, then he / she will suffer from allergies.
3. Tom is a cat
4. Mary is allergic to cats

Represent the above sentences in FOL and prove by FOL resolution  
"Mary sneezes"

Or representation of the above premises in first order logic

1. allergic(x)  $\rightarrow$  sneeze(x)
2. cat(y)  $\wedge$  allergic\_to\_cats(x)  $\rightarrow$  allergic(x)
3. cat(Tom)
4. allergic\_to\_cats(Mary)

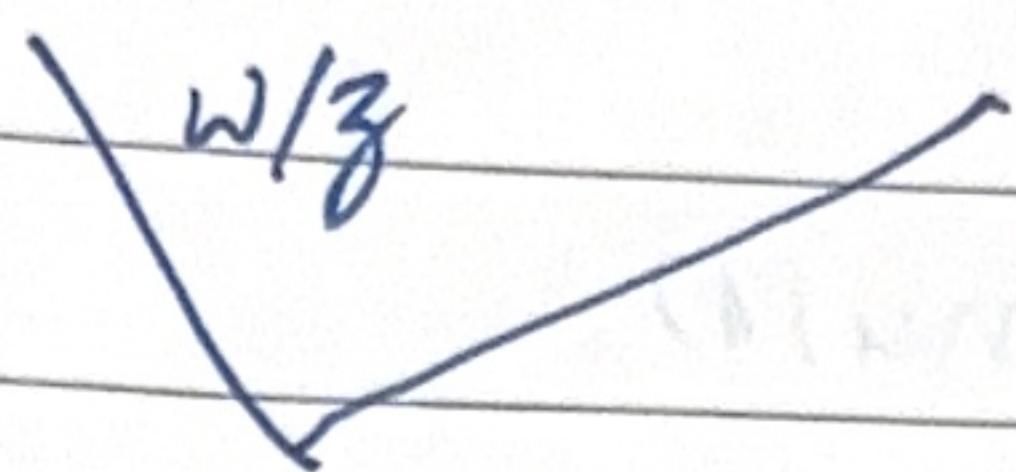
The goal state is sneeze(Mary)

- ~~allergic(x)  $\rightarrow$  sneeze(x)~~
- $\neg$  allergic(x)  $\vee$  sneeze(x)
- $\neg$  cat(y)  $\wedge$  allergic\_to\_cats(x)  $\rightarrow$  allergic(x)
- $\neg$  (cat(y)  $\wedge$  allergic\_to\_cats(x))  $\rightarrow$   $\neg$  allergic(x)
- $\neg$  cat(y)  $\vee$  allergic\_to\_cats(x)  $\rightarrow$   $\neg$  allergic(x)

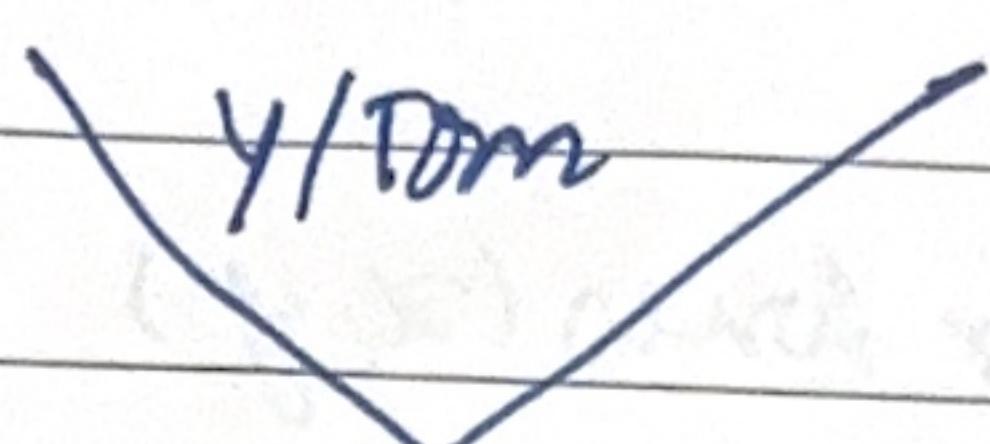
State Program to reach goals:

$\neg \text{Allergies}(w) \vee \text{Sneeze}(w)$

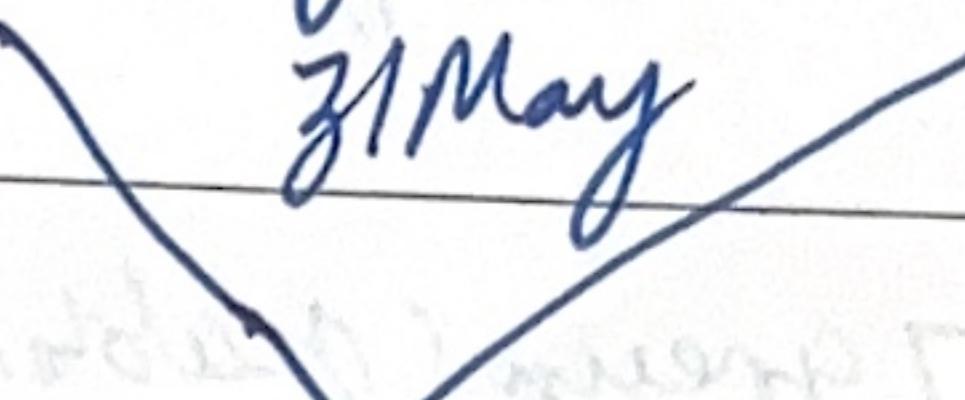
$\neg \text{Cat}(y) \vee \neg \text{AllergicToCat}(y)$   
 $\vee \text{Allergies}(y)$



$\neg \text{Cat}(y) \vee \text{Sneeze}(y) \quad \text{①} \quad \neg \text{AllergicToCat}(y) \quad \text{cat (Tom)}$



$\text{allergicToCat}(May) \quad \text{Sneeze}(y) \vee \neg \text{AllergicToCat}(y)$



Sneeze(May)

$\neg \text{Sneeze}(May)$

{ }

$$\begin{aligned} \therefore x + \bar{x} &= 1 \\ x \cdot \bar{x} &= 0 \end{aligned}$$

- Q) Consider the first order logic (FOL) representations for the following sentences
- Every real number has its corresponding negative
  - Everybody loves somebody
  - There is somebody whom no one loves
  - See an brought everything that Ronaldo brought
  - Pavot is green while rabbit is not.

Q) Find a most general unifier for the set

$$W = \{ p(a, \alpha, f, (g(y))), p(z, f(y), f(u)) \} -$$

$$W = \{ g(f(a), g(x)), g(y, y) \}$$

(i)

 $\forall x \text{ real}(x) \rightarrow \text{negative}(x)$ 

(a)

 ~~$\forall x \text{ real}(x) \exists (\text{negative}(x))$~~ 

(ii)

 $\forall x \exists y \text{ loves}(x, y)$ 

(iii)

 $\forall x \exists y \forall z \text{ loves}(x, y)$ 

(iv)

 $\forall x \text{ brown}(\text{Susan}, x) \rightarrow \text{light}(\text{Ronald}, x)$ 

(v)

 $\text{green}(\text{parrot}) \wedge \neg \text{green}(\text{rabbit})$ 

(vi)

 $\forall x \forall y (\text{Parrot}(x) \rightarrow \text{green}(x) \wedge \text{Rabbit}(y) \rightarrow \text{Green}(y))$ 

P

(i)

 $P(a, x, f(g(y))) \quad P(z, f(z), f(u))$ Substitute  $z$  for  $a$  $P(a, x, f(g(y))) \quad P(a, f(a), f(a))$ On substituting  $a$  by  $g(y)$  ① ~~$P(a, x, f(g(y))) \quad P(a, f(a), f(g(y)))$~~ On substituting  $x$  by  $f(a)$  $P(a, f(a), f(g(y))) \quad P(a, f(a), f(f(y)))$  ~~$x/f(a)$~~ The General Unifier  $g/a, w/g(y)$   
 $x/f(a)$ 

(ii)

 $g(f(a), g(w)) \quad g(g(y))$ 

(vii)

This isn't unifiable as  $y$  cannot accept two values

3)

Convert the following sentences into FOL statements and prove the following statements using both forward & backward chaining.

John likes all kinds of foods

$\forall x \text{ food}(x) \rightarrow \text{like}(\text{John}, x)$

Apples are food  
 $\text{food}(\text{Apples})$

Chicken is food  
 $\text{food}(\text{chicken})$

Anything anyone eats and isn't killed by is food

$\forall x \forall y \text{ eats}(x, y) \wedge \neg \text{killed by}(x) \Rightarrow \text{food}(y)$

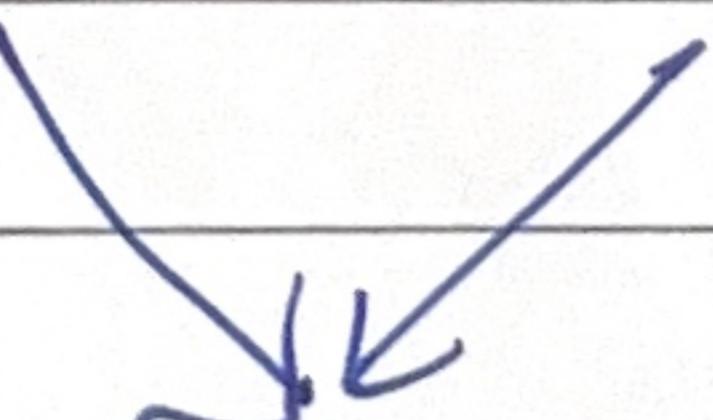
Bill eats peanuts and is still alive  
 $\text{eats}(\text{Bill}, \text{peanuts}) \rightarrow \text{alive}(\text{Bill})$

Sue eats anything Bill eats  
 $\forall x \text{ eats}(\text{Sue}, x) \rightarrow \text{eats}(\text{Sue}, \text{Bill})$

→ An additional required fact:  $\neg \text{killed}(x) \rightarrow \text{alive}(x)$

The forward chaining

$x(\text{Bill}, y) / \text{peanuts}$        $\text{alive}(\text{Bill})$   
 $\text{eats}(\text{Bill}, \text{peanuts})$        $\neg \text{killed}(\text{Bill})$



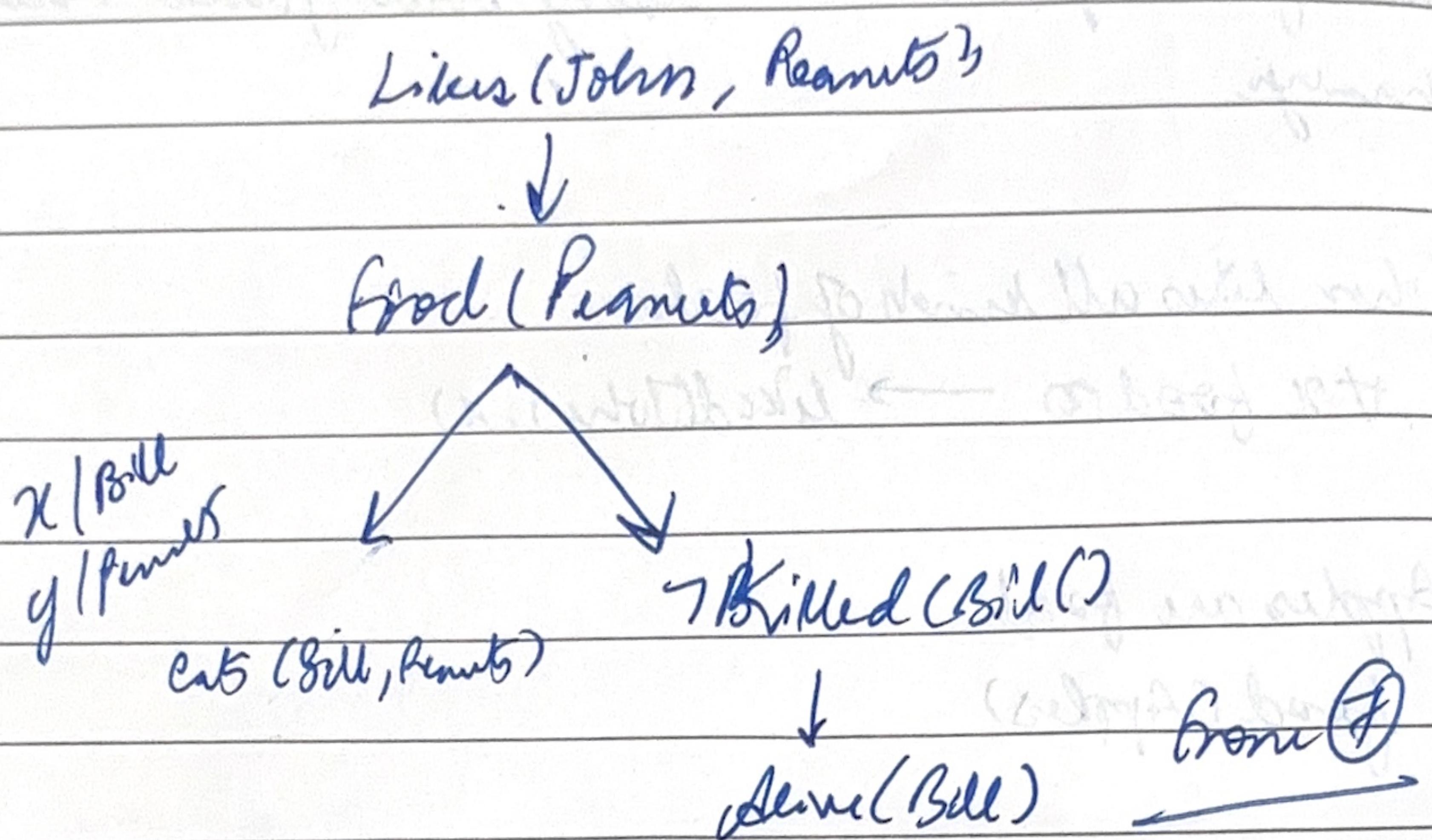
$\text{food}(\text{Peanuts})$

$\downarrow x / \text{peanuts}$

$\text{likes}(\text{John}, \text{peanuts})$

From ①

Backward Drawing:



4) function MINMAX-DECISION (state) returns an action  
return arg max<sub>a</sub> ∈ actions(<sub>s</sub>) MIN-VALUE(RESULT(<sub>s</sub>, a))

function MAX-VALUE (state) returns a utility value  
if TERMINAL TEST (state) then return UTILITY (state)  
 $v \leftarrow -\infty$   
for each a in actions (state) do  
 $v \leftarrow \text{MAX} (v, \text{MIN-VALUE}(\text{RESULT} (s, a)))$   
return v

function MIN-VALUE (state) returns a utility value  
if TERMINAL TEST (state) then return UTILITY (state)  
 $v \leftarrow \infty$   
for each a in ACTIONS (state) do  
 $v \leftarrow \text{MIN} (v, \text{MAX-VALUE} (\text{RESULT} (s, a)))$   
return v

