LAB	08:-
L	nification:
-	->Statements:-
0	Doctors treat patients who are sick.
2	John is adoctor
3	Many is sick
4	Doctors work in hospitals
3	General Hospital is a Hospital
6	John works at General Hospital.
	One application
=>	Quantification Statements:
	COLD COLD COLD COLD COLD COLD COLD COLD
	1) Y2 Yy (Doctor(2) A sicky (Y) -> Treats(2, Y)
	@ Dodox (John)
	3 sick (Mary)
	(4) V2 (Doctor(2) -> 3h(Hospital (W) Works
direct.	A+ (x, h)))
	3 Hospital (General Hospital)
	6 Works At (John, General Mospital)
-	Del-31. State on toin
1365	Unity Statements:
	Fx (Treats (x, Mary))
	* (1) From Statement (1) unity Treats
	To From Statement (1), unify Treats (2, y) with Treats (2, May), binding y= many.
	y=mary
	② Statements (3), (on firm lick (17ay) is true activating statement (i).
0.1	is true activating statement(1).
	3) Use Statement (2) to deduce that

	Page
	Doctors (John), holds so
	A Accordance to the second sec
	x-> John satisfise the query
	2-7 John Satisfise the quary 2 2 (Trusts (2, many)
	the state of the s
	: 2 -> John
	Y-> Mary
	Little of the Control of the Levis of the
	I have the transfer of the second of the sec
	Knowledge base =[
	l" type": "rule" "rule": " Ax My
	Knowledge base = [{"Type": "rule" "rule": "Ax Ay (Doctors(a) A Sicky (y) -> Treats (a,y)),
(X. wie	
	l'type ": " jad" : Doctor (John)")
	I'type": joct , " foct : Sick (mary)"),
- Arth	type " foct fact : (Dodors)
	["type": "fact", "fact": Sick (many)", s' type": "fact", "fact": fact (Dodon's) -> Sh (hospital(h) At A trans
	A) (x, 1)))
	f"type": "lad" "fact"! Hospital (General Hospital)
	5" to 20" (21" 1" ")
	S' type: "fact" "fact": "Works At (John, General Hospital)"?
	7
	anony = { "predid": "Treats " "anguard":
101	quory = { "predid": "Ireats " "orgunaro":
12	
	de unity Clab quory):
Con	des renity (Leb. query): predicate: query ("predicate") torget arg = guery ("arguments") [1) refult = None
1	target ang = glury ("arguments") [1)
	refult = None
	are much of Coltandate at the
100	

Date

for item ["type"] = "nule "and predicate in Homo ['rule']: doctor = None and "sickly" in rule: sick puso o = nlone for fact in Kb:

if fact [" type"]:= "fad" and

Octor of in facts, jad".

olodor - jacts fad". if robult?

noturn (" The query " (query (" predicate))

[nobult] [target - ag 9" is

ensfied: [robult] treats (lagot - ag)]." clse:

noturn of "The quiring " Equivage quiring

['prediction] Result: Unify (Knowledge base, gwy) print (result). Output: John Treats Mary