Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

CSE4108

Artificial Intelligence

Term Assignment 1

Submitted By:

Name: Shimul Paul

ID: 16.02.04.014

Section: A1

Backward Chaining

Backward Chaining is an inference method widely used in artificial intelligence, automated theorem provers & proof assistants. Backward Chaining methodology can be described as working back from a goal. Many Programming languages support backward chaining within their inference engines.

Backward Chaining Properties:

- It is known as a top down approach
- Backward-Chaining is based on modus ponens inference rule.
- In backward chaining, the goal is broken into sub goal or sub-goals to prove the facts true.
- It is called a goal-driven approach, as a list of goals decides which rules are selected & used.
- Backward-Chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, & various AI applications.
- The backward chaining method mostly used a depth-first search strategy for proof.

Steps of working for backward chaining:

Step-1: In the first step, we will take the goal fact & from the goal fact, we'll derive other facts that we'll prove true.

Step-2: We'll derive other facts from goal facts that satisfy the rules.

Step-3: At Step-3, we will extract further fact which infers from facts inferred in Step 2.

Step-4: We'll repeat the same until we get to a certain fact that satisfies the conditions.

So we can say that:

Query → goal → conclusion → premise → subgoal → conclusion → Premise → subgoal → backtracking → -- until Proved or KB exhausted.

Example:

" Hasib is a Parent of Rakib. Sohel is a Parent of Ratan. Manik is a Parent of Hasib and Sohel. Everybody is male. Hasib and Sohel are not same Person. If Hasib and Sohel have the same Parent. Manik and Ratan is male then Hasib is a uncle of Ratan."

Prove that "Hasib is a uncle of Ratan".

For solving the above Problem, first we will convert all the above facts into first order definite clauses, & then we will use a backward chaining algorithm to reach the goal.

Inputs:

```
tupplelist = [('Parent', 'Hasib', 'Rakib'),
              ('Parent', ~~'Hasib'~~ 'Sohel' , 'Ratan'),
              ('Parent', 'Manik', 'Hasib'),
              ('Parent', 'Manik', 'Sohel')]
```

Male = ['Hasib', 'Rakib', 'sohel', 'Ratan', manik']

For this Proof our goal is to prove Hasib is a uncle of Raton."

For this,

Step-1 : Here first we have to declare the tupple list. It contains some facts that was given.

Step-2 : Then we will take input to find uncle.

Step-3 : Then we will run a loop in the tupple list to find the Parent of Rakib and Ratan. we have to make sure that both of them Parents are not the same Person. Then we will execute another one loop for find the Parent both rakib and ratan's Parent. If their Parents are same then Hasib is a uncle of Ratan.

Step-4 : Then we will execute another loop in male list to find that Hasib & sohel is male and their Parent is male.

Another Example:

"As per the law, it is a crime for an American to sell weapons to hostile nations. country A, an enemy of America, has some missiles & all the missiles were sold to it by Robert, who is an American citizen."

Prove that "Robert is a criminal."

From the example we can write these rules:

1. American (p) ∧ weapon(q) ∧ sells (p,q,r) ∧ hostile(r) → Criminal (p)
2. Owns (A, T1)
3. Missile (T1)
4. ? p Missiles (p) ∧ Owns (A, p) → Sells (Robert, p, A)
5. Missile (p) → weapons (p)
6. Enemy (p, America) → Hostile (p)
7. Enemy (A, America).
8. American (Robert)

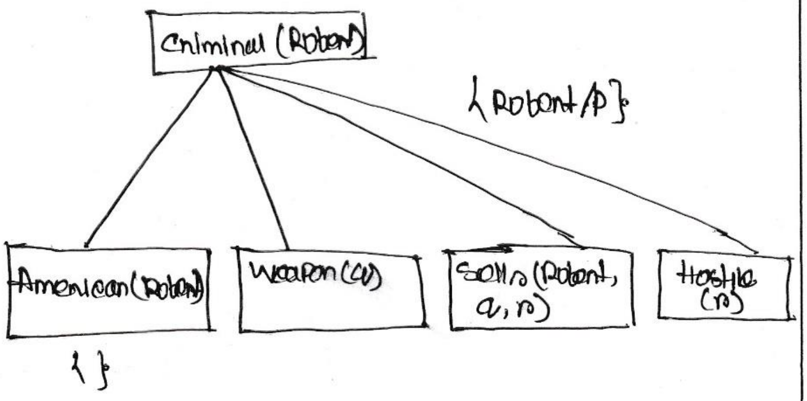
For Backward chaining we will start with our goal predicate which is criminal (Robert) & then infer the other rules.

Step-1: At first, we will take the goal fact. And from the goal fact, we will infer other facts, and at last, we will prove those facts true. So our goal fact is "Robert is a criminal" so following is the predicate of it.

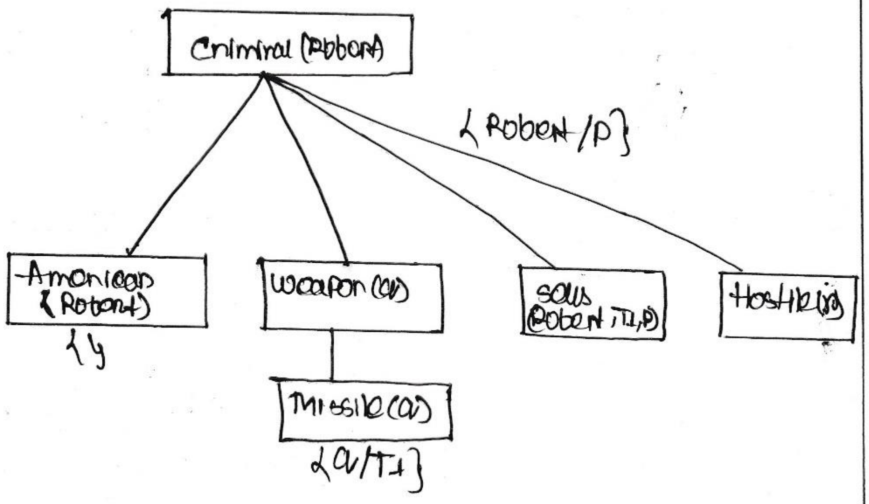
Criminal (Robert)

Step-2: At the second step, we will infer other facts from goal facts which satisfies the rules. So as we can see in Rule-1, the goal predicate criminal (Robert) is present with substitution {Robert/P}. So we will add the facts below the first level and will replace P with Robert.

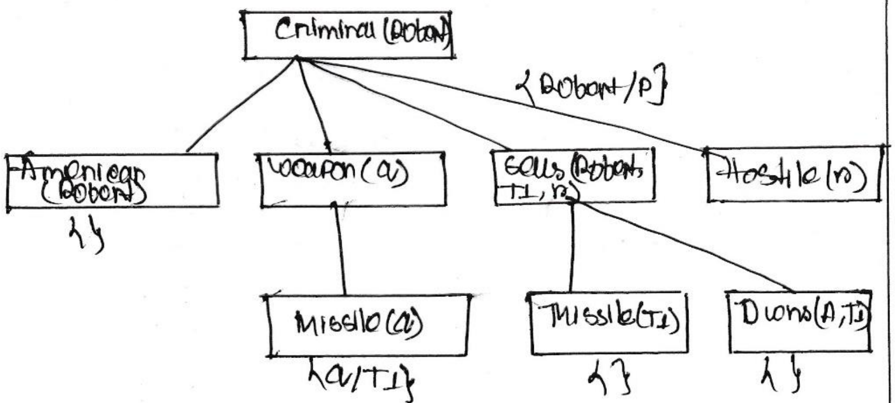
Here we can see American (Robert) is a fact so it is proved here.



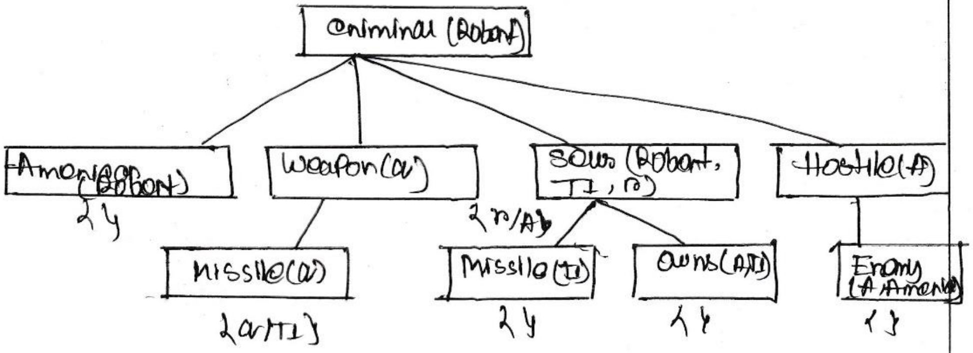
Step-3: At step 3 - we will extract further fact Missile(a) which infer from weapon (a) as it satisfies Rule-(5). Weapon (a) is also true with the substitution of a constant T1 at a.



Step-4! At step 4, we can infer facts Missile(T1) and owns (A,T1) from sells (Robert, T1, r) which satisfies the Rule-4, with the substitution of A in Place r. So these two statements are Proved here.

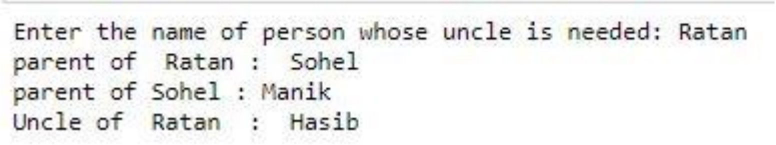


Step 5 At step 5 we can infer the fact enemy (A, America) from Hostile (A) which satisfies Rule-6. And hence all the statements are Proved true using backward ~~tracking~~ chaining.



Python Code:

Input & Output:



```
Enter the name of person whose uncle is needed: Ratan
parent of  Ratan :  Sohel
parent of Sohel : Manik
Uncle of  Ratan  :  Hasib
```

```
tp = [('parent', 'Hasib', 'Rakib'),
  ('parent', 'Sohel', 'Ratan'),
  ('parent', 'Manik', 'Hasib'),
  ('parent', 'Manik','Sohel')]
male = ['Hasib', 'Rakib', 'Sohel', 'Ratan', 'Manik']

name = str(input('Enter the name of person whose uncle is needed: '))
#uncle = 'Hasib'
p = ''
for i in range (len(tp)):
  if(tp[i][0] == 'parent' and tp[i][2]==name):
    p = tp[i][1]
gf = ''
for i in range (len(tp)):
  if(tp[i][0] == 'parent' and tp[i][2]==p):
    gf = tp[i][1]
print('parent of ',name,': ',p,'\nparent of',p,':', gf)
uncle = ''

flag = 0
for i in range (len(tp)):
  if(tp[i][0] == 'parent' and tp[i][1]==gf):
    if(tp[i][2] != p):
      uncle = tp[i][2]
      flag = 1
```

```
            break
if flag == 1:
    print('Uncle of ',name,' : ',uncle)
else:
    print('Not Found')
```