Resolution

Resolution

- Resolution yields a complete inference algorithm
- when coupled with any complete search algorithm.
- Resolution makes use of the inference rules.
- Resolution performs deductive inference.
- Resolution uses proof by contradiction.
- One can perform Resolution from a Knowledge Base.
- A Knowledge Base is a collection of facts or one can even call it a database with all facts.

Algorithm

- Resolution basically works by using the principle of **proof by contradiction.**
- To find the **conclusion we should negate the conclusion.** Then the resolution rule is applied to the resulting clauses.
- Each clause that contains complementary literals is resolved to produce a 2new clause, which can be added to the set of facts (if it is not already present)
- This process continues until one of the two things happen
- • There are no new clauses that can be added
- • An application of the resolution rule derives the empty clause An empty clause shows that the negation of the conclusion is a complete contradiction, hence the negation of the conclusion is invalid or false or the assertion is completely valid or true.

Steps

- Steps for Resolution
- • Convert the given statements in Predicate/Propositional Logic
- • Convert these statements into Conjunctive Normal Form
- • Negate the Conclusion (Proof by Contradiction)
- • Resolve using a Resolution Tree (Unification)

Steps to Convert to CNF (Conjunctive Normal Form)

Every sentence in Propositional Logic is logically equivalent to a conjunction of disjunctions of literals.

A sentence expressed as a conjunction of disjunctions of literals is said to be in Conjunctive normal Form or CNF.

1. Eliminate implication '→'
2. a → b = ~a v b
3. ~ (a ^ b) = ~ a v ~ b .... DeMorgan's Law
4. ~ (a v b) = ~ a ^ ~ b .... DeMorgan's Law
5. ~ (~a) = a

Eliminate Existential Quantifier '∃'

To eliminate an independent Existential Quantifier, replace the variable by a Skolem constant. This process is called as Skolemization.

Example: ∃y: President (y)

Here 'y' is an independent quantifier so we can replace 'y' by any name (say – George Bush).

So, ∃y: President (y) becomes President (George Bush).

To eliminate a dependent Existential Quantifier we replace its variable by Skolem Function that accepts the value of 'x' and returns the corresponding value of 'y.'

Example: ∀x : ∃y : father_of (x, y)

Here 'y' is dependent on 'x', so we replace 'y' by S(x).

So, ∀x : ∃y : father_of (x, y) becomes ∀x : ∃y : father_of (x, S(x)).

Eliminate Universal Quantifier ‘∀’

To eliminate the Universal Quantifier, drop the prefix in PRENEX NORMAL FORM i.e. just drop ∀ and the sentence then becomes in PRENEX NORMAL FORM.

Eliminate AND ‘^’

a ^ b splits the entire clause into two separate clauses i.e. a and b

(a v b) ^ c splits the entire clause into two separate clauses a v b and c

(a ^ b) v c splits the clause into two clauses i.e. a v c and b v c

To eliminate ‘^’ break the clause into two, if you cannot break the clause,

distribute the OR ‘v’ and then break the clause.

Problem Statement:

1. Ravi likes all kind of food.
2. Apples and chicken are food
3. Anything anyone eats and is not killed is food
4. Ajay eats peanuts and is still alive
5. Rita eats everything that Ajay eats

Prove by resolution that Ravi likes peanuts using resolution.

Step 1: Converting the given statements into Predicate/Propositional Logic

i. ∀x : food(x) → likes (Ravi, x)

ii. food (Apple) ^ food (chicken)

iii. ∀a : ∀b: eats (a, b) ^ ~killed (a) → food (b)

iv. eats (Ajay, Peanuts) ^ alive (Ajay)

v. ∀c : eats (Ajay, c) → eats (Rita, c)

vi. ∀d : alive(d) → ~killed (d)

vii. ∀e: ~killed(e) → alive(e)

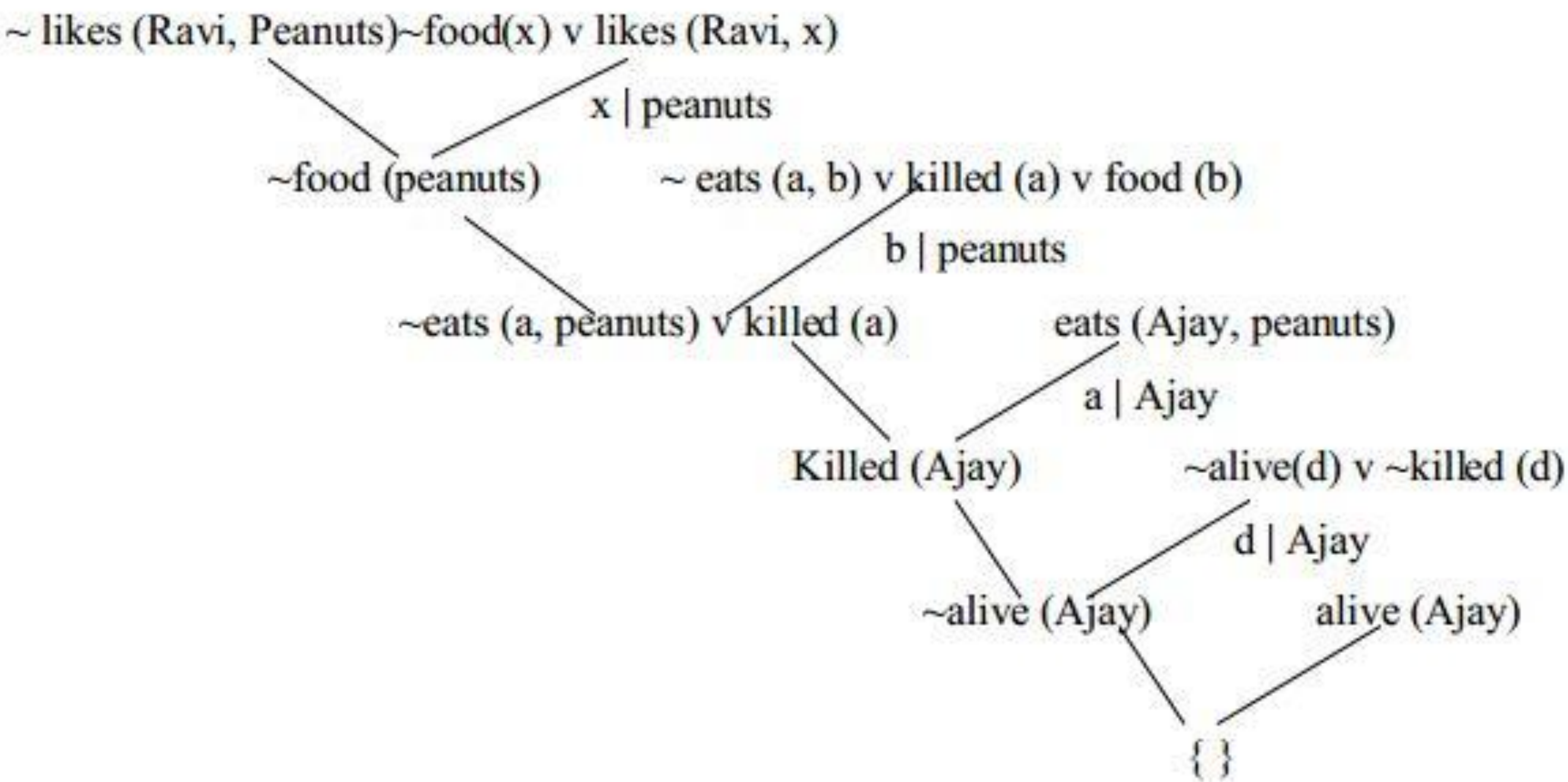
Conclusion: likes (Ravi, Peanuts)

Step 2: Convert into CNF

i. ~food(x) v likes (Ravi, x)
ii. Food (apple)
iii. Food (chicken)
iv. ~ eats (a, b) v killed (a) v food (b)
v. Eats (Ajay, Peanuts)
vi. Alive (Ajay)
vii. ~eats (Ajay, c) V eats (Rita, c)
viii. ~alive (d) v ~ killed (d)
ix. Killed (e) v alive (e)
Conclusion: likes (Ravi, Peanuts)

Negate the conclusion

~ likes (Ravi, Peanuts)


~ likes (Ravi, Peanuts)
~food(x) v likes (Ravi, x)
x | peanuts
~food (peanuts)
~ eats (a, b) v killed (a) v food (b)
b | peanuts
~eats (a, peanuts) v killed (a)
eats (Ajay, peanuts)
a | Ajay
Killed (Ajay)
~alive(d) v ~killed (d)
d | Ajay
~alive (Ajay)
alive (Ajay)
{ }

Uses of Resolution in Today’s World

- Used widely in AI.
- Helps in the development of computer programs to automate reasoning and theorem proving

For work out

All hounds howl at night.

Anyone who has any cats will not have any mice.

Light sleepers do not have anything which howls at night.

John has either a cat or a hound.

(Conclusion) If John is a light sleeper, then John does not have any mice.