& and

V or

~ not

->implies

(Ax) foral

(Ex) exis

Check un satisfiability of

1. ~ p V ~ q
2. P V q
3. ~p V q
4. p V~q

**CDCL**

Decide

< ~p[d] | 1-4 | \* >

Propagate

< ~p[d], ~ q[4] | 1-4 | \* >

Conflict

< ~p[d], ~ q[4] | 1-4 | p Vq>

Explain

*Since the last assign literal il propagated we apply explain, resolution between 4 and 2*

< ~p[d], ~ q[4] | 1-4 | p>

Backjump, th*e last assign literal is ~p that is decided*

*Learn clause 5 p*

< p[5] | 1-5 | \* > *back to search state*

Propagate

< p[5], q[3] | 1-5 | \* >

Conflict

< p[5], q[3] | 1-5 | ~ p V ~ q >

Since I am in conflict without decided literal by Explain I can get the empty clause

Final result UNSAT

**RESOLUTION**

Check validity of the following inference

All cats are felon

No felon is vegetarian

———————

No cat is vegetarian

*Formalisation*

(Ax) (C(x) →F(x))

~ (Ex) ((F(x) & V(x))

————————

~ (Ex) (C(x) & V(x))

After negation the thesis

(Ax) (C(x) →F(x))

~ (Ex) ((F(x) & V(x))

(Ex) (C(x) & V(x))

After skolemization

(Ax) (C(x) →F(x))

(Ax)~((F(x) & V(x))

C(a) & V(a)

Clausification

(Ax) (~ C(x) v F(x))

(Ax) (~F(x) v ~ V(x))

C(a)

V(a)

The herbrand universe only contains a

Instantiation

1. ~ C(a) v F(a)
2. ~F(a) v ~ V(a)
3. C(a)
4. V(a)

Apply resolution:

Resolution between 4+2 gives

5) ~F(a)

Subsumption removes clause 2

Resolution between 5+1 gives

6) ~C(a)

Resolution between 6+3 gives empty clause

Final result UNSAT hence the inference is correct

Apply Congruence closure to

f(g(b))=a

b=c

g(c)=a

~ f(a) =a

Flattering: use new named d for g(b) and e for f(a)

d=g(b)

e=f(a)

f(d) = a

b = c

g(c)=a

~ e = a

**USARE PARENTESI GRAFFE NON QUADRE**

Set of block

[d, g(b)]

[f(d), a, g(c)]

[b,c]

[e, f(a)]

We need to merge first two blocks, because b=c → g(d) = g(c)

[d, g(b), f(d), a, g(c)]

[b,c]

[e, f(a)]

We need to merge first and third blocks, because a = d → f(a) =f(d)

[ d, g(b), f(d), a, g(c), e, f(a) ]

[ b,c ]

Since e and a are in the same block, UNSAT

CAPIRE

Check unsatisfiability of

y-x ≤ 1 x→y weight 1

z-y ≤ -2 y→z weight -2

x ≤ z

In IDL

Rewrite x ≤ z as

x - z ≤ 0 z → x weight 0

Consider the cycle

x —→ y ——> z —→ x

It has has weight 1, + (-2) + 0

So it is negative UNSAT