Resolution Example.

1. Marcus was a man. 2. Marcus was a Pompeian. 3. Marcus was born in 40 A.D.
4. All men are mortal 5. All Pompeians died when the volcano erupted in 79 A.D.
6. No mortal lives longer than 150 years. 7. It is now 1991
8. Alive means not dead. 9. If someone dies, then he is dead at all later times.

Answer the question : Is Marcus dead ?

1. man (Marcus)
2. Pompeian (Marcus)
3. born (Marcus, 40)
4. $\forall x : man(x) \rightarrow mortal(x)$
5. $\forall x : Pompeian(x) \rightarrow died(x, 79)$
6. erupted (volcano, 79)
7. $\forall x: \forall t1: \forall t2: mortal(x) \wedge born(x, t1) \wedge gt(t2-t1, 150) \rightarrow dead(x, t2)$
8. now = 1991
9. $\forall x: \forall t: (alive(x,t) \rightarrow \neg dead(x, t)) \wedge (\neg dead(x, t) \rightarrow alive(x,t))$
10. $\forall x: \forall t1: \forall t2: died(x, t1) \wedge gt(t2, t1) \rightarrow dead(x, t2)$

Axioms in Clause form:

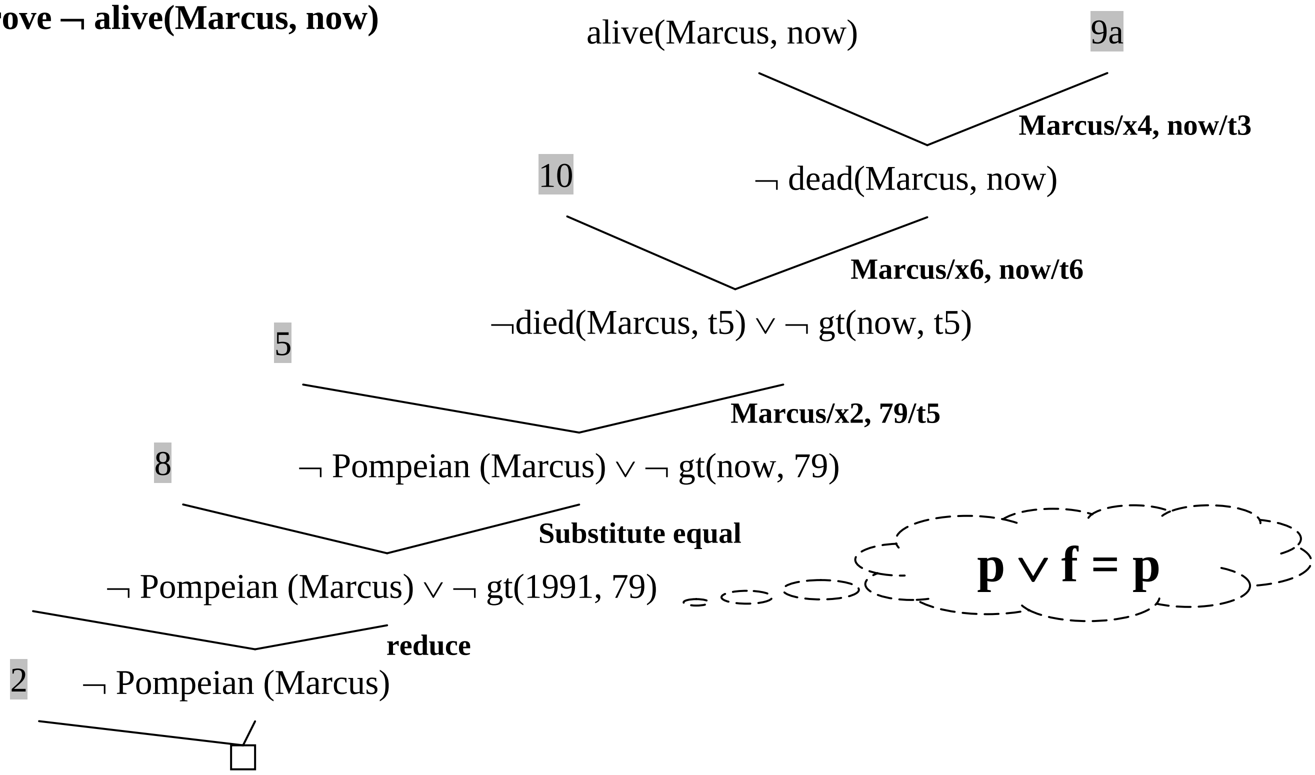
1. man (Marcus)
2. Pompeian (Marcus)
3. born (Marcus, 40)
4. $\neg man(x1) \vee mortal(x1)$
5. $\neg Pompeian(x2) \vee died(x2, 79)$
6. erupted (volcano, 79)
7. $\neg mortal(x3) \vee \neg born(x3, t1) \vee \neg gt(t2-t1, 150) \vee dead(x3, t2)$
8. now = 1991

9a. $\neg$ **alive(x4,t3)** $\vee \neg$ **dead(x4, t3)**

9b. **dead(x5, t4)** $\vee$ **alive(x5,t4)**

10. $\neg died(x6, t5) \vee \neg gt(t6, t5) \vee dead(x6, t6)$

prove ¬ alive(Marcus, now)



Example proof

We will now show how to apply the conversion procedure and the resolution refutation procedure on a more complicated example, which is stated in English as:

Jack owns a dog.
Every dog owner is an animal lover.
No animal lover kills an animal.
Either Jack or Curiosity killed the cat, who is named Tuna.
Did Curiosity kill the cat?

First, we express the original sentences (and some background knowledge) in FOL:

A. $\exists x\ Dog(x) \wedge Owns(Jack, x)$
B. $\forall x\ (\exists y\ Dog(y) \wedge Owns(x, y)) \Rightarrow AnimalLover(x)$
C. $\forall x\ AnimalLover(x) \Rightarrow (\forall y\ Animal(y) \Rightarrow \neg Kills(x, y))$
D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$
E. $Cat(Tuna)$
F. $\forall x\ Cat(x) \Rightarrow Animal(x)$

Now we have to apply the conversion procedure to convert each sentence to implicative normal form. We will use the shortcut of writing P instead of $True \Rightarrow P$:

Al. $Dog(D)$
A2. $Owns(Jack, D)$
B. $Dog(y) \wedge Owns(x, y) \Rightarrow AnimalLover(x)$
C. $AnimalLover(x) \wedge Animal(y) \wedge Kills(x, y) \Rightarrow False$
D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$
E. $Cat(Tuna)$
F. $Cat(x) \Rightarrow Animal(x)$

The problem is now to show that *Kills(Curiosity, Tuna)* is true. We do that by assuming the negation, $Kills(Curiosity, Tuna) \Rightarrow False$, and applying the resolution inference rule seven times, as shown in the figure in the next page . We eventually derive a contradiction, *False*, which means that the assumption must be false, and *Kills(Curiosity, Tuna)* is true after all.

In English, the proof could be paraphrased as follows:

> Suppose Curiosity did *not* kill Tuna. We know that either Jack or Curiosity did, thus Jack must have. But Jack owns D, and D is a dog, so Jack is an animal lover. Furthermore, Tuna is a cat, and cats are animals, so Tuna is an animal. Animal lovers don't kill animals, so Jack couldn't have killed Tuna. But this is a contradiction, because we already concluded that Jack must have killed Tuna. Hence, the original supposition (that Curiosity did not kill Tuna) must be wrong, and we have proved that Curiosity *did* kill Tuna.

The proof answers the question "Did Curiosity kill the cat?" but often we want to pose more general questions, like "Who killed the cat?" Resolution can do this, but it takes a little more work to obtain the answer. The query can be expressed as ∃*w Kills(w, Tuna).* If you repeat the proof tree in the following figure, substituting the negation of this query, *Kills(w, Tuna)* ⇒ *False* for the old query, you end up with a similar proof tree, but with the substitution *{w/Curiosity}* in one of the steps. So finding an answer to "Who killed the cat" is just a matter of looking in the proof tree to find the binding of *w*. It is straightforward to maintain a composed unifier so that a solution is available as soon as a contradiction is found.