Ex:

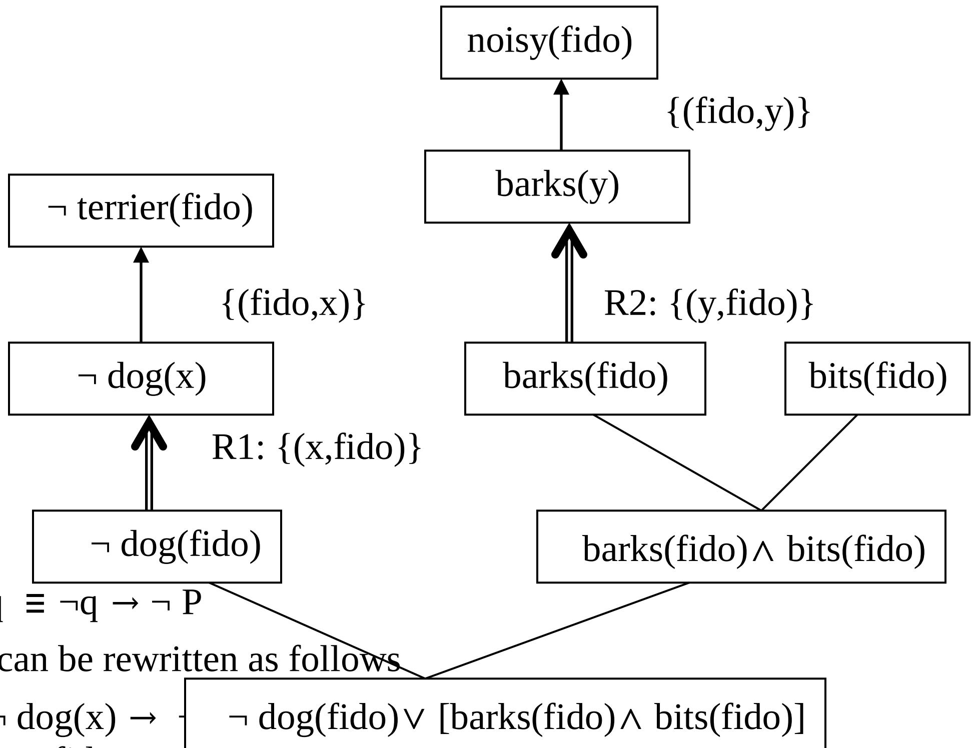
Facts: $\neg$ dog(fido)$\vee$ [barks(fido)$\wedge$ bits(fido)]

Rules: R1: terrier(x) $\rightarrow$ dog(x)

R2: barks(y) $\rightarrow$ noisy(y)

Goal: $\neg$ terrier(z) $\vee$ noisy(z)

P $\rightarrow$ q $\equiv$ $\neg$ P $\rightarrow$ $\neg$q



P $\rightarrow$ q $\equiv$ $\neg$q $\rightarrow$ $\neg$ P

$\therefore$ R1 can be rewritten as follows

R1: $\neg$ dog(x) $\rightarrow$ $\neg$

Goal: z= fido

Goal have been proved

$\neg$ terrier(z) $\vee$ noisy(z)

For z=fido

Composition unifier = { (fido,x),(fido,y)}

$\therefore$ Solution consistent

- Backward AND/OR deduction systems:

Goal: the goal must be in an AND/ OR form. The goal must consist of one or more literals connected by the operators $\vee$ and /or $\wedge$.

Variables in the goal must be existentially quantified. You have to skolomize universally quantified variables replace $\forall$ variables by a function of a $\exists$ variable.

Ex:

$(\exists y)\ (\forall x)\ \{p(x) \rightarrow [q(x,y) \wedge \neg\ [r(x) \vee s(y)]]\}$

$(\exists y)\ (\forall x)\ \{\neg\ p(x) \vee [q(x,y) \wedge \neg\ [r(x) \vee s(y)]]\}$

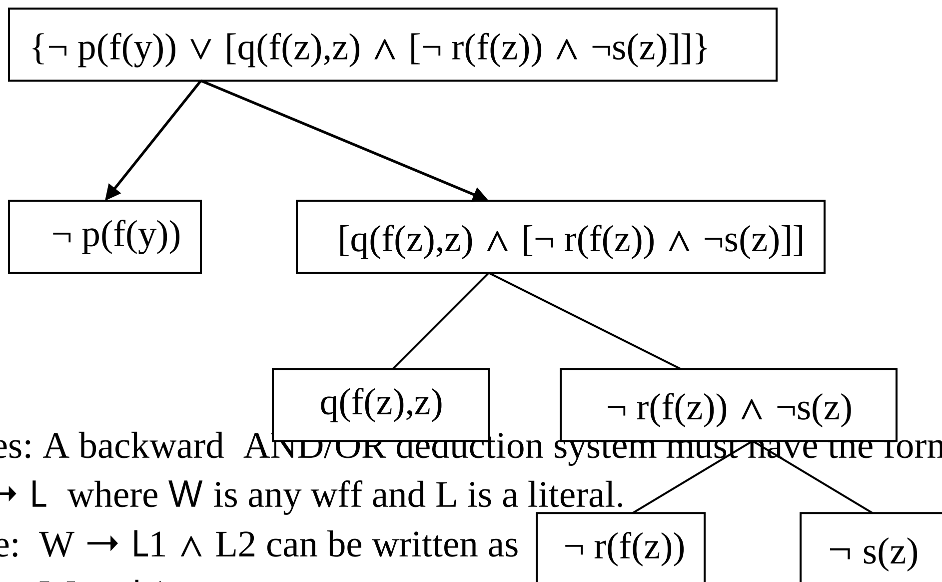
$(\exists y)\ (\forall x)\ \{\neg\ p(x) \vee [q(x,y) \wedge [\neg\ r(x) \wedge \neg s(y)]]\}$

$(\exists y)\ \{\neg\ p(f(y)) \vee [q(f(y),y) \wedge [\neg\ r(f(y)) \wedge \neg s(y)]]\}$

Rename variables over the main disunites in the goal expression

$\{\neg\ p(f(y)) \vee [q(f(z),z) \wedge [\neg\ r(f(z)) \wedge \neg s(z)]]\}$

Represent the goal expression as AND/OR tree such that operator $\vee$ is represented by OR node, and the operator $\wedge$ represented by AND node.



Rules: A backward AND/OR deduction system must have the form

$W \rightarrow L$ where W is any wff and L is a literal.

Note: $W \rightarrow L1 \wedge L2$ can be written as

$W \rightarrow L1$

$W \rightarrow L2$

All conditions that apply to forward rules also apply for backward rules.

Facts: must be conjunction of literals (set of literals).

$W \rightarrow L1 \wedge L2$

$\neg\ W \vee [L1 \wedge L2] \Rightarrow (\neg\ W \vee L1\ )\ (\neg\ W \wedge L2)$

$\neg\ W \vee L1 \equiv W \rightarrow L1$

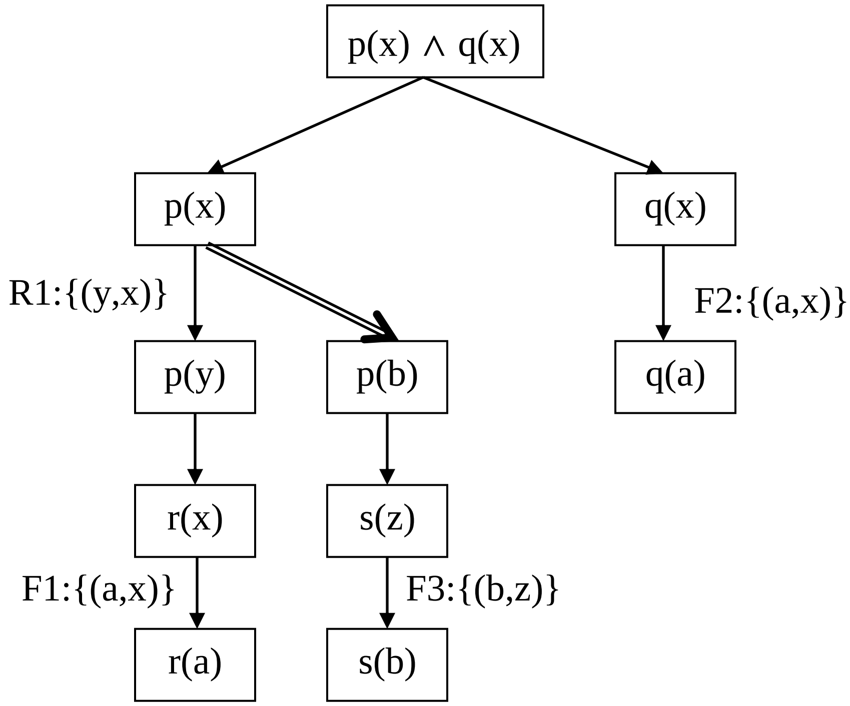
$\neg\ W \wedge L2 \equiv W \rightarrow L2$

Ex: goal: $p(x) \wedge q(x)$

Rules: R1: $r(x) \rightarrow p(y)$

R2: $s(z) \rightarrow p(b)$

Facts: F1: r(a)
F2: q(a)
F3: s(b)



We have to check for consistency for the first solution
{(y,x)}
{(a,x)} :- composition unifier= {(a,x),(y,x)}
{(a,x)}
Consistency checking for the second solution
R2:{(b,x)}
F3:{(b,z)}
F2: {(a,x)}
Composition unifier = {(a,x),(b,x),(b,z)}

$\therefore$ not consistent

Ex: F1: dog(fido)
F2: ¬ barks(fido)
F3: big(fido)
F4: small(myrtile)

R1: big(x1) ∧ dog(x2) → friendly(x1)

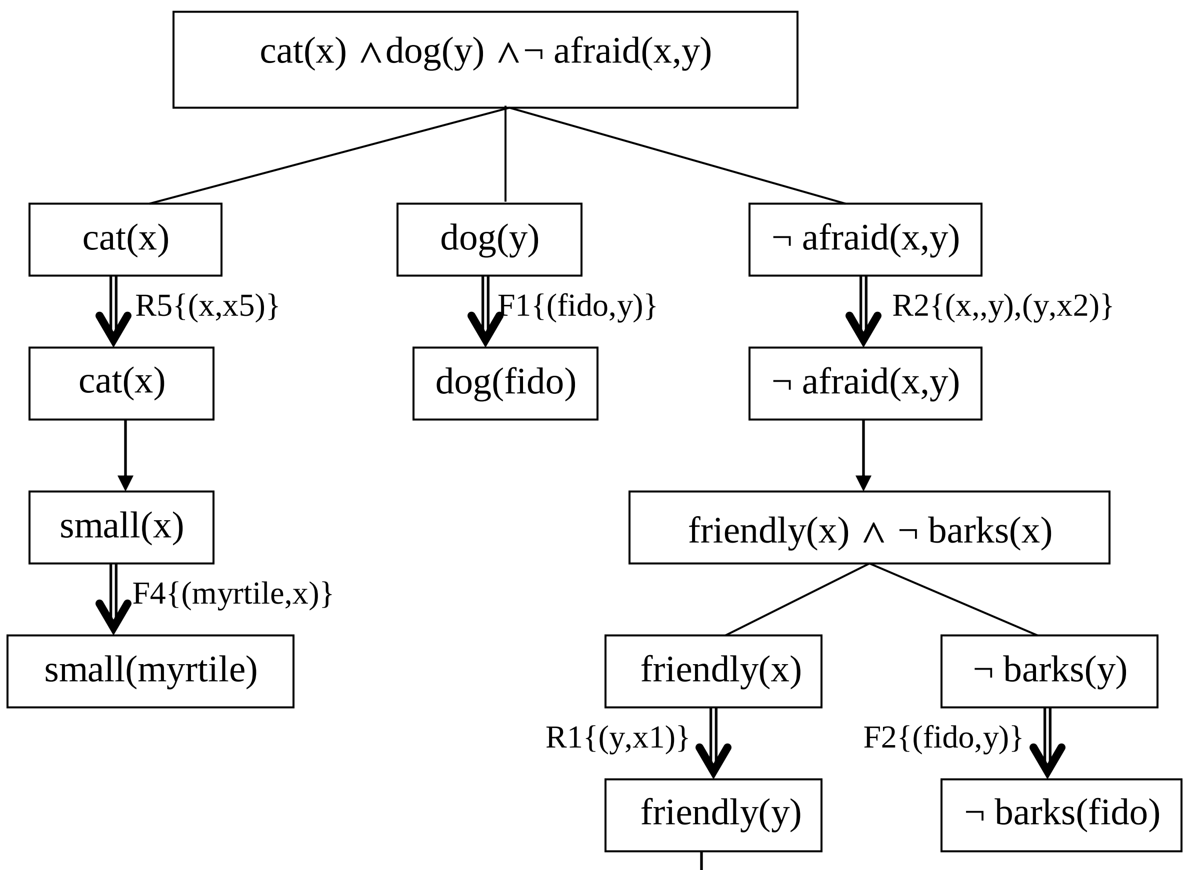
R2: friendly(x2) ∧ ¬ barks(x2) → ¬ afraid(y2,x2)

R3: dog(x3) → animal(x3)

R4: cat?(x4) → animal(x4)

R5: small(x5) → cat(x5)

Goal: ∃x∃y [cat(x) ∧dog(y) ∧¬ afraid(x,y)]



Cosistant unifier ={(myrtile,x),(x,x),(fido,y), (x,y2),(y,x1)}

x5
Myrtile → x
y

x2
fido → y
x1

∴ consistant