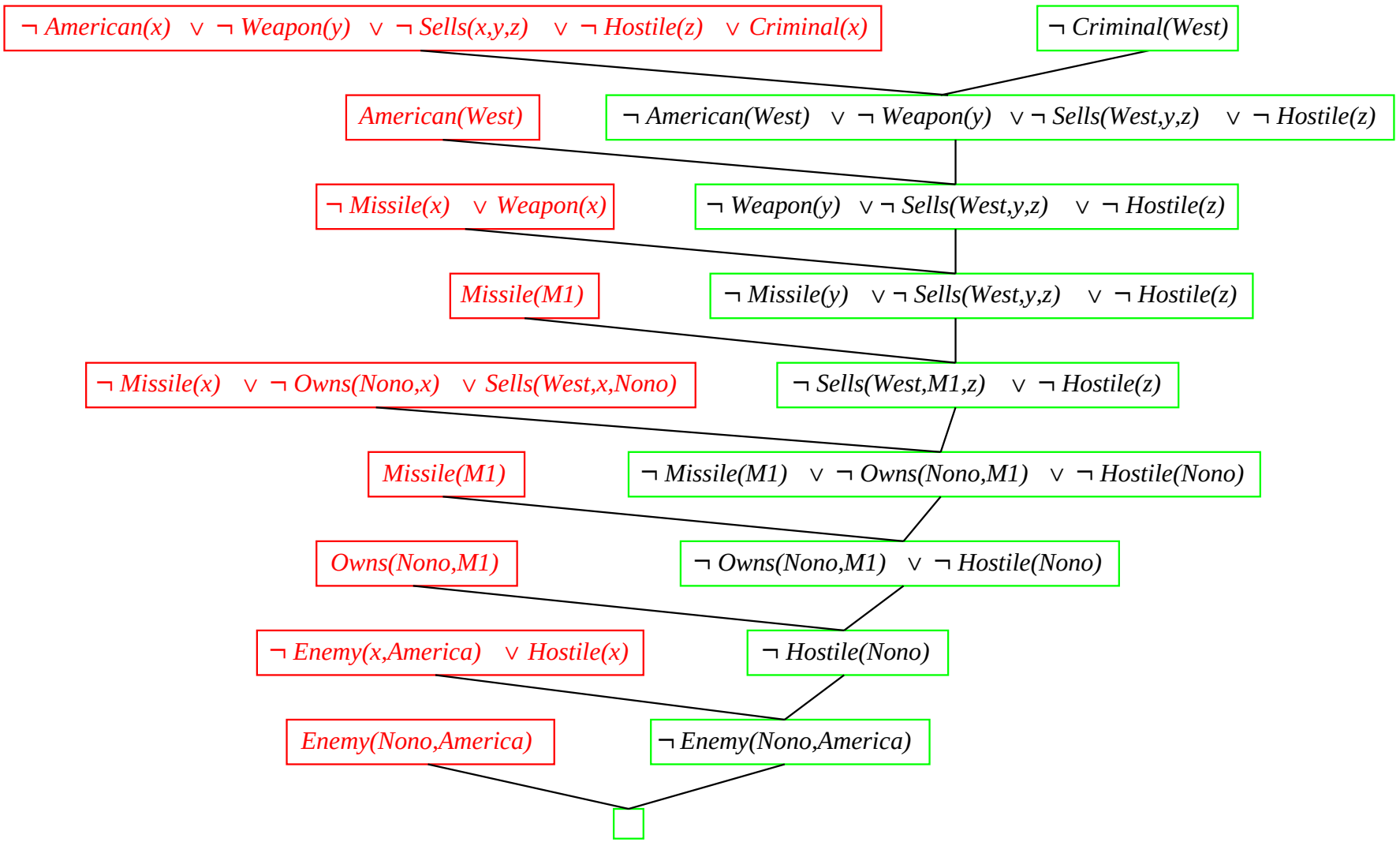
Prova di risoluzione per clausole definite



Un esempio più complesso

Data la seguente conoscenza e query:

Everyone who loves all animals is loved by someone.
Anyone who kills an animal is loved by no one.
Jack loves all animals.
Either Jack or Curiosity killed the cat, who is named Tuna.
Did Curiosity kill the cat?

questa viene rappresentata in FOL come segue (goal negato):

A. $\forall x \ [\forall y \ \ Animal(y) \Rightarrow Loves(x, y)] \Rightarrow [\exists y \ \ Loves(y, x)]$
B. $\forall x \ [\exists y \ \ Animal(y) \wedge Kills(x, y)] \Rightarrow [\forall z \ \ \neg Loves(z, x)]$
C. $\forall x \ \ Animal(x) \Rightarrow Loves(Jack, x)$
D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$
E. $Cat(Tuna)$
F. $\forall x \ \ Cat(x) \Rightarrow Animal(x)$
¬G. $\neg Kills(Curiosity, Tuna)$

Un esempio più complesso

Trasformado tutte le sentenze in CNF otteniamo

A1. $Animal(F(x)) \vee Loves(G(x), x)$

A2. $\neg Loves(x, F(x)) \vee Loves(G(x), x)$

B. $\neg Animal(y) \vee \neg Kills(x, y) \vee \neg Loves(z, x)$

C. $\neg Animal(x) \vee Loves(Jack, x)$

D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$

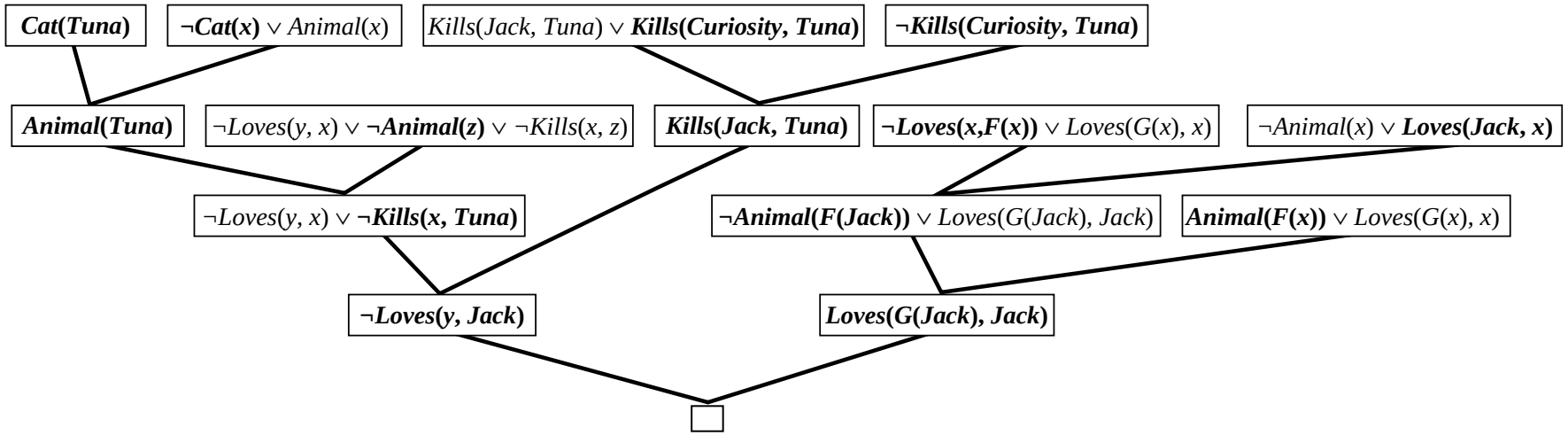
E. $Cat(Tuna)$

F. $\neg Cat(x) \vee Animal(x)$

¬G. $\neg Kills(Curiosity, Tuna)$

Un esempio più complesso

Una prova, usando la Risoluzione, è la seguente:



Goal più generale: Who killed the cat? $\rightarrow$ in FOL: $\exists w \ Kills(w, Tuna)$
Negando otteniamo, in CNF:$\neg Kills(w, Tuna)$

Prova: simile alla precedente dove introduciamo la sostituzione $\{w/Curiosity\}$
Problema: prove non costruttive per goal esistenziali...

Un esempio più complesso

Problema: prove non costruttive per goal esistenziali...

$\neg Kills(w, Tuna)$ risolve (tramite $\{w/Curiosity\}$) con
$Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$ generando
$Kills(Jack, Tuna)$, che risolve di nuovo (tramite $\{w/Jack\}$) con
$\neg Kills(w, Tuna)$ portando alla clausola vuota

Ma w in questa prova è legato a due valori *distinti*!

Possibili soluzioni:

- variabili query legate ad un solo valore (implica capacità di effettuare backtracking sui possibili legami)
- aggiunta termine speciale (*answer literal*) al goal negato:
  $\neg Kills(w, Tuna) \vee Answer(w)$
  che per la prova non costruttiva darebbe
  $Answer(Curiosity) \vee Answer(Jack)$ (che non costituisce una risposta)

Strategie di risoluzione

♢ Unit Clause

Si preferisce effettuare la risoluzione con una delle sentenze costituita da un singolo letterale (clausola unitaria)

♢ Unit Resolution

Forma ristretta di risoluzione in cui ogni passo di risoluzione deve coinvolgere una clausola unitaria (incompleta in generale, completa per clausole di Horn)

♢ Set of Support

Basata sull'insieme di supporto da cui si preleva una delle sentenze e dove si pone il risolvente. Esempio di insieme di supporto: il goal (e tutti i risolventi derivati da esso)

♢ Input Resolution

Combina una sentenza in input (KB e Query) con il risolvente corrente. Tipica struttura a spina di pesce.

♢ Linear Resolution

Come Input Resolution, ma ammette anche la combinazione del risolvente corrente con suoi avi.

♢ Subsumption

Elimina tutte le sentenze che sono "subsumed" (più specifiche di altre). Es. $P(A)$ e $P(A) \vee P(B)$ sono più specifiche di $P(x)$.

Uguaglianza

Fino ad ora l'uguaglianza ($=$) non è stata trattata.

Esistono fondamentalmente 3 approacci diversi:

1. Assiomatizzazione
   Idea base: si aggiungono assiomi che descrivono le proprietà dell'uguaglianza ed assiomi opportuni per ogni predicato e funzione.

2. Aggiunta di una nuova regola di inferenza (p.e., Demodulation, Paramodulation)
   Idea base: se $x = y$ e $UNIFY(x, z) = \theta$,
   rimpiazza z con $SUBST(\theta, y)$.

3. Estensione dell'algoritmo di unificazione
   Idea base: unifica termini equivalenti,
   p.e. $1 + 2$ e $2 + 1$.

Completezza Risoluzione: schema prova

Ogni insieme di sentenze S è rappresentabile in forma normale congiuntiva

Si assuma S insoddisfacibile e in forma normale congiuntiva

Teorema di Herbrand

Qualche insieme S' di istanze ground è insoddisfacibile

Teorema di Risoluzione per termini ground

La risoluzione può trovare una contraddizione in S'

Lemma di Lifting

Prova tramite risoluzione dell'esistenza di una contraddizione per S

Riassunto

Inferenza in FOL

♢ Proposizionalizzazione ed uso di unificazione per evitare la istanziazione di variabili coinvolte in una prova.

♢ Modus Ponens Generalizzato (GMP) usa l'unificazione e permette l'applicazione di forward e backward chaining su clausole definite.

♢ GMP è completo per clausole definite, anche se determinare le conseguenze logiche è un problema semidecidibile. Per Datalog (no funzioni e clausole definite) il problema è decidibile.

♢ Forward chaining è completo e polinomiale per Datalog.

♢ Backward chaining è usato in Programmazione Logica (p.e., Prolog) e rinforzato con opportune tecniche di compilazione.

♢ Risoluzione generalizzata è completa per KB in forma normale congiuntiva (CNF). Se però si usa il principio di induzione $\Rightarrow$ non completa.