AI2 Module 2
Tutorial 1

Bonnie Webber
School of Informatics

1 Most General Unifiers (mgu)

Recall that in unification, the *most general unifier* (mgu) is the least constrained *substitution* that makes two clauses match/unify with each other.

What is the mgu for each pair of clauses below, and what is the result of applying it to the pair? If there is no mgu, explain why. (Remember that the systematic procedure we have for finding mgu's is the Unify algorithm in R&N Section 9.2, p.278.)

1. $p(a,b,b)$
   $p(X,Y,Z)$
2. $q(Y,g(a,b))$
   $q(g(X,X),Y)$
3. $older(father(Y),Y)$
   $older(father(X),john)$
4. $knows(father(Y),Y)$
   $knows(X,X)$

N.B. Prolog's unifier behaves differently on the last example than Russell and Norvig's unification algorithm. When you are next on-line, try it and see what you get. Try to understand the basis for the difference. (Adapted from R&N, 2^{nd} edition, Ex. 9.4)

2 Representation

Represent the following sentences in first-order logic (FOL), choosing whatever **vocabulary** of constants, predicate symbols, and function symbols you need in order to capture what the sentences express.

1. Horses and cows are mammals.
2. An offspring of a horse is a horse.
3. Bluebeard is a horse.
4. Bluebeard is Charlie's parent.
5. Offspring and parent are inverse relations.
6. Every mammal has a parent

Now convert your logical representation of (1)-(6) into *Horn clauses* to use in Generalised Modus Ponens (GMP). Recall that a *Horn clause* is either an atomic clause or an implication whose LHS consists of a conjunction of (positive) atomic clauses and whose RHS consists of a single atomic clause. Horn clauses do not have explicit quantifiers: Existential quantifiers should be eliminated through the primitive inference rule of **Existential Instantiation** and universal quantifiers, by simply treating all variables as being universally quantified. (Adapted from R&N 2^{nd} edition, Ex. 9.9)

3 Generalised Modus Ponens

Here we use your Horn Clause representations for (1)-(6) to reason with GMP using **backward chaining**. (Adapted from R&N 2nd edition, Ex. 9.10)

1. Draw the proof tree generated by **exhaustive** backward chaining for the query $\exists h$. *Horse(h)*, where clauses are matched **in the order given**.
2. How many solutions for *h* actually follow from your sentences?
3. Can you think of a way to find them all?