Automated Reasoning 2012 (KB)
PROBLEMS 4: Control, Hyper-resolution and OTTER

QUICKIES:

1. Given: B(a), O(a,b), G(c), O(b,c), B(X) ∨ G(X), ¬B(X) ∨ ¬G(X), ¬O(U,V) ∨ G(U) ∨ ¬G(V)
Find refutations as follows:
(i) using predicate ordering: B<O<G (ii) using locking (you choose the locks)
(iii) using hyper-resolution

LONGER QUESTIONS:

1. **PREDICATE ORDERING**: Rank predicate symbols in some order; within a clause only literals with *lowest ranking* predicate in the order can be resolved upon. e.g. if F < G, and there are F literals in a clause then the F literals *must* be selected before any G literals. Consequently, any G literals in a clause can be resolved upon *only* if there are *no* F literals in the clause. In other words, F literals take priority over G literals. Carry out factoring at any time. (Or, perhaps restrict non-safe factoring to include lowest ranked predicates. Try and analyse the effects of each decision (eg how might it affect subsequent subsumption).

(a) *Try the predicate ordering method on the clauses*:

¬H(a), ¬ F(x) ∨ ¬H(b), Fx ∨ Hx, ¬Gz ∨ ¬Fb, Gu ∨ ¬Fu
Use a saturation search and either the predicate ordering G <F<H or H <F<G.

(b) *Why is the predicate ordering method useless for these clauses?*
P(y,a) ∨ P(f(y),y), P(a,y) ∨ P(y,f(y)), ¬P(x,y) ∨ ¬P(y,x)

2. **LOCKING:** Each literal in a clause is given a (numeric) index (also called a *lock*). A literal can be used in a resolution step only if it is the *lowest indexed*. If several literals have the same lowest lock, then any of them may be selected. Carry out factoring any time. (Or, could restrict non-safe factoring to include at least one lowest locked literal.)

Try the locking strategy on the two sets of clauses:

(i) D(x,a) ∨ D(x,b), ¬D(x,y) ∨ C(x,y), ¬T(x) ∨ ¬C(x,a), T(c), ¬D(c,b)

(ii) P(y,a) ∨ P(f(y),y), P(a,y) ∨ P(y,f(y)), ¬P(x,y) ∨ ¬P(y,x)

Make up indexes and use a different one for each literal. Use a saturation search.

3. Explain how locking could be used to simulate predicate-ordering and (approximately) simulate Hyper-resolution. In the case of Hyper-resolution why is the simulation only approximate?

4. *Use hyper-resolution to derive the empty clause from*:

(a) ¬H(a), ¬F(x) ∨ ¬H(b), Fx ∨ Hx, ¬Gz ∨ ¬Fb, Gu ∨ ¬Fu

(b) D(x,x), ¬D(x,y) ∨ ¬D(y,z) ∨ D(x,z), P(x) ∨ D(g(x),x), P(x) ∨ G(g(x),1),
G(a,1), P(x) ∨ G(x,g(x)), ¬P(y) ∨ ¬D(y,a),
¬G(x,1) ∨ ¬G(a,x) ∨ P(f(x)), ¬G(x,1) ∨ ¬G(a,x) ∨ D(f(x),x)

(c) *Use negative hyper-resolution to derive the empty clause from the clauses of part (a).*

(d) In part (a) suppose each predicate "P" be renamed as a new predicate " ¬P' "; i.e. P' is true exactly when P is false and vice versa. Call the renamed clauses S'.

Check that the negative hyper-resolution derivation from the clauses in part (a) that you found in part (c) is the same as a standard hyper-resolution derivation using the renamed clauses in S'.

5. Show (by hand) how OTTER's SOS and USABLE lists change for the clauses

P(y,a) ∨ P(f(y),y),   P(a,y) ∨ P(y,f(y)),   ¬P(x,y) ∨ ¬P(y,x)

Try for different initial SOS and USABLE lists and for settings of binary resolution and hyper-resolution. Note that in hyper-resolution OTTER defaults to using the following predicate ordering in electrons: predicates are ordered first by arity (ternary before binary before unary, etc.) and then within arity they are ordered alphabetically. Furthermore, electrons are only permitted to resolve on atoms whose predicate is the one coming **latest** in the ordering (so for the clauses of Q2(i) the ranking would be C<D<T, and T atoms are selected from an electron first. (Using the notation in Slides 7, to achieve the same effect the ranking would be T<D<C). To cancel the default order use the command "`clear(order_hyper)`". You can also set your own order using the lex command, for example "`lex([T(_),D(_,_),C(_,_)])`" makes C maximal. The "_" indicates argument positions.

6. Show the completeness of Hyper-resolution (for unsatisfiable ground clauses S) using induction on the number of different atoms occurring in the literals of S. Here are some **Hints**.

Assume S is unsatisfiable, then

i) If exactly one atom occurs in literals of S what form are the clauses? Then show a HR proof exists.

ii) Show there must always be at least one electron in any unsatisfiable clause S.

iii) If an electron is a fact A, form the set S' from S, where S' is the set S without clauses containing A and with all occurrences of ¬A deleted. S' mentions fewer atoms than S and is still unsatisfiable. Assume a HR proof of [] can be found from S' and show how to use this proof to obtain an HR-proof from S.

iv) If there is no fact electron, then choose an atom A in some electron E and form S' as before (ie remove all clauses containing A (including E) and remove ¬A from the rest). Explain why S' is unsatisfiable; next use part (iii) to find a HR derivation of [] from the clauses (S - {E}) + {A}.

Now delete A from E and from any other electrons with A, and remove clauses with ¬A to form S''. Explain why S'' is unsatisfiable. Assume again a HR proof of [] can be found (since the number of different atoms in S'' is smaller than the number in S), and put back A wherever it was deleted. The result will be a derivation of A. Finally, use the refutation from S' to find a HR proof from S.

v) Put all the above together to give a proof by induction.

7. Consider whether hyper-resolution can be combined with predicate ordering, atom ordering or locking. It only makes sense to order electrons - (why?).

Try some examples to see what is possible. As an example of introducing locking on electrons, consider the clauses {PQ, QR, RW, ¬Q¬R, ¬P¬R, ¬Q¬W} with various lockings. You should find that refutations are longer with the extra restrictions.

Challenge: Can you show the combined method is complete (for ground clauses)?

8. (**Optional Case Study**: Atom ordering) (See the notes 7c, as we may not have time to cover it.) Ground atoms are ordered by a total order and non-ground atoms are ordered *only* if all instances will respect the order. Resolve on literals if the atomic part (i.e. ignoring the sign) is minimal in the clause. Note there may be several such literals.

(a) *Try the atom ordering method on the clauses*:

¬H(a),   ¬F(x) ∨ ¬H(b),   Fx ∨ Hx,   ¬Gz ∨ ¬Fb,   Gu ∨ ¬ Fu

Use a saturation search and the atom ordering Fa < Ha < Ga < Fb < Hb < Gb.

(b) *Try the atom ordering method for the clauses:*

P(y,a) ∨ P(f(y),y),   P(a,y) ∨ P(y,f(y)),   ¬P(x,y) ∨ ¬P(y,x)

Order atoms according to the number of occurrences of the symbol *f* in their arguments. If two atoms have an equal number of *f*s, then order by the number in the first argument.