**AUTOMATED REASONING**

**SLIDES 12:**

**PARAMODULATION**
**Using Equality (=) in Data**
**Equality Axioms**
**Equality and Resolution: Paramodulation**
**Controlling use of equality in Resolution:**
  **Hyper-paramodulation**
  **RUE-resolution**
**Equality and Models**

**KB - AR - 13**

## EQUALITY



(1) $T(p,q) \vee T(q,p)$ (p,q are constants)
(2) $\neg T(X,X)$ (3) p=q

**A "Natural" derivation of []**

(1) $T(p,q) \vee T(q,p)$ (2) $\neg T(X,X)$ (3) p=q
(4) (1 + 3) $T(q,q) \vee T(q,p)$ (substitute q for p in T(p,q))
(5) (4 + 2) T(q,p)
(6) (5 + 3) T(q,q) (substitute q for p in T(q,p))
(7) (6 + 2) []

Question:
Given this derivation of [ ] would you expect (1), (2), (3) to be unsatisfiable?
(Hint: replace = by the predicate symbol S.)

- Actually, they do have a model!

So what has gone wrong?



(1) $T(p,q) \vee T(q,p)$ (2) $\neg T(X,X)$ (3) p=q

- They do have a model!

eg Let Domain = {1,2} p -> 1; q -> 2
 Set T(1,1) , T(2,2) both false
 Set T(1,2), T(2,1) both true
 Set =(1,2) is true

- But they do *not* have a H-model in which '=' satisfies the 'equality axioms'.

Informally, the equality axioms state that
 "if t1 = t2 and property P holds for t1, then P holds for t2"

Can you argue that 1, 2 and 3 have no H-model satisfying the above?
Hint: Suppose property P is "T(p,q)"

Paramodulation is a reasoning step that implicitly incorporates the use of such equality properties to generalise the notion of substitution

First let's see how paramodulation operates
Then we'll see why it is correct reasoning

See Optional part for why the simple notion of identity is no use for H-models

DEFN: (**PARAMODULATION**) (generalises simple substitution) (ppt)



if $C1 \equiv L[t] \vee C1'$ (i.e. t occurs in L) , $C2 \equiv r=s \vee C2'$ (or $s=r \vee C2'$) and $r\theta = t\theta$, then the clause $(C1' \vee C2' \vee L[s\theta])\theta$ is called a **paramodulant.**

*Example*

a=b  $L(X) \vee M(X)$
 (P)
 $L(b) \vee M(a)$

Can also obtain: $L(a) \vee M(b)$.

Substitutions occur in 1 argument position at a time.

**Comparison with definition:**

C1 is $L(X) \vee M(X)$. X in L(X) is "t" and C1' is M(X)
C2 is a=b. a is the "r" term and b is the "s" term. C2' is the empty clause

Unify "a" with X. $\theta$ is the substitution X==a, and $a\theta = X\theta$
The term $s\theta$ is "b", since the substitution $\theta$ doesn't affect "b".

The paramodulant clause is therefore $(M(X) \vee L(b))\theta$, which is $M(a) \vee L(b)$

The next slide gives a simple prescription for performing paramodulation

DEFN: (**PARAMODULATION**) (generalises simple substitution) (ppt) 12aiv

if C1≡ L[t]∨ C1' (i.e. t occurs in L) , C2 ≡ r=s ∨ C2' (or s=r ∨ C2') and rθ=tθ, then the clause (C1' ∨ C2' ∨ L[sθ])θ is called a **paramodulant.**

*Example*

a=b     L(X) ∨ M(X)

(P)

L(b) ∨ M(a)

X in L(X) is the "to" term
a in a=b is the "from" term
Unify a with X ( θ is X==a)
C1θ is L(a) ∨ M(a)
Replace a in L(a) by b
Result is L(b) ∨ M(a)

*In general*:
1. Unify the "to" term – the one to be replaced in C1 (t)
and the "from" term – the one in the equality being replaced (r) (mgu is θ)

2. Apply the unifier θ to both clauses C1 and C2 to give C1θ and C2θ

3. Replace the "to" term in C1θ by the term on the other side of the "from" equation – the one in the equality that is the replacement (sθ)

4. The result is the disjunction of C1θ and C2θ after replacement and without the equation.

**SOME MORE EXAMPLES (ppt)** 12av

f(X)=b ∨ C(X)     R(f(a))∨ Q

(P)

C(a) ∨ R(b) ∨ Q

match f(X) with f(a) and replace by b.

U=V     T(p,q) ∨ T(q,p)

(P)

T(V,q) ∨ T(q,p)

Identify the "to" and "from" terms

f(X,g(X))=e ∨ T(X)     S(Y,f(g(Y),Z)) ∨ W(Z)

(P)

S(Y,e)∨W( g(g(Y)) )∨T(g(Y) )

match f(X,g(X)) with f(g(Y),Z) (X/g(Y), Z/g(g(Y)) ) and replace f( g(Y),g(g(Y)) ) by e.

**Another Example (from Hodges) for you to check** 12avi

1. T(p,q) ∨T(q,p) (Not everyone is trying equally hard.
¬∀x∀y[¬T(x,y) ∧ ¬T(y,x)] )

2. ¬T(X,X) (No-one tries harder than himself)

3. U=V (There is not more than one person.¬ ∃x∃y¬ [x = y] )

(4) (P. 3+1) T(V1,q) ∨T(q,p) (take instance U1=V1 of (3); match U1 with p and replace by V1)

(5) (4+2) T(q,p)

(6) (P. 5+3) T(V2,p) ( take instance U2=V2 of (3); match U2 with q and replace by V2)

(7) (6+2) []

**Paramodulation** 12bi

*Paramodulation* is the method by which equality is included in resolution refutations. It is a generalisation of equality substitution: if s=t and s occurs in some sentence S, then t can replace s in any of the occurences. Similarly, if t occurs in S, then s can replace t. (See definition on 12aiii.)

The paramodulation rule consists of several steps, given in 12aiv. It is easiest to apply instantiation first, to both the clause containing the equality E as well as to the clause containing the term to which the equality will be applied, so that the term being substituted *from* is the same as the term being substituted *into*. Then apply the equality substitution. The resulting clause, called a *paramodulant*, is the disjunction of the instantiated and substituted clauses (apart from equality E, which is omitted).

We'll see that paramodulation implicitly makes use of the Equality Axiom clausal schema (12bii/12biii) and can be simulated by resolution, in which case there are two distinct phases:
(a) use EQAX2 and equation E to obtain equation E', that can be used to substitute at atom level;
(b) use E' and EQAX3 to make the substitution at atom level.

For (a) there may need to be (none, 1 or more) applications of using the appropriate EQAX2.

For example, suppose the clause a=b ∨ C were to be used (E is a=b). In order to substitute into P(f(a)), an equality of the form f(..)=t is required. From a=b∨ C and the instance (of EQAX2) ¬x=y ∨ f(x)=f(y) we get f(a)=f(b)∨C (E' is f(a)=f(b)); then we can use the instance (of EQAX3) ¬x=y ∨ ¬P(x) ∨ P(y) to obtain P(f(b))∨C. If, instead of P(f(a)), the atom was P(g(f(a))), then an additional instance of EQAX2, ¬x=y ∨ g(x)=g(y), is necessary to obtain g(f(a))=g(f(b))∨C from f(a)=f(b)∨C.
**Exercise:** Show how paramodulation of X=b into P(f(Y),Y) to derive P(f(b),Y) is simulated by resolution and appropriate instances of EQAX2 and EQAX3.

## Equality Axioms

12bii

Reasoning with equality "naturally" uses the *equality axioms* implicitly

*EQAX1* $\forall x[x=x]$

*EQAX2* $\forall[xi=yi \to f(x1,\ldots,xi,\ldots,xn)=f(x1,\ldots,yi,\ldots,xn)]$

*EQAX3* $\forall[xi=yi \wedge P(x1,\ldots,xi,\ldots,xn) \to P(x1,\ldots,yi,\ldots,xn)]$

*EQAX2 and EQAX3 as clauses:*

*EQAX2* $\forall[\neg xi=yi \vee f(x1,\ldots,xi,\ldots,xn)=f(x1,\ldots,yi,\ldots,xn)]$

*EQAX3* $\forall[\neg xi=yi \vee \neg P(x1,\ldots,xi,\ldots,xn) \vee P(x1,\ldots,yi,\ldots,xn)]$

EQAX2 and EQAX3 are *substitutivity* schema.
There is one axiom for each argument position for each function/predicate.

There is an equivalent form of the Equality Axioms, which are also useful.

*Alternative form for EQAX2 and EQAX3:*

*EQAX2 (Alternative)*

$\forall[\neg x1=y1 \vee \ldots \vee \neg xn=yn \vee f(x1,\ldots,xn) = f(y1,\ldots,yn)]$

*EQAX3 (Alternative)*

$\forall[\neg x1=y1 \vee \ldots \vee \neg xn=yn \vee \neg P(x1,\ldots,xn) \vee P(y1,\ldots,yn)]$

**The Equality Axioms**

12biii

Reasoning with equality in resolution and in tableau implicitly makes use of a set of clausal axiom schema and the reflexivity of equality (EQAX1). There are 2 basic substitutivity schema:

(i) those that deal with substitution at the argument level of atoms (EQAX3), and
(ii) those that deal with substitution at the argument level of terms (EQAX2).
They are given on Slide 12bii.

An alternative form of EQAX combines the schema for each argument place into a single schema that will deal with one or more arguments at the same time. They are:

EQAX2 (Alternative) $\forall[x1=y1 \wedge \ldots \wedge xn=yn \to f(x1,\ldots,xn)=f(y1,\ldots,yn)]$
EQAX3 (Alternative) $\forall[x1=y1 \wedge \ldots \wedge xn=yn \wedge P(x1,\ldots,xn) \to P(y1,\ldots,yn)]$

or as clauses:
EQAX2 (Alternative) $\forall[\neg x1=y1 \vee \ldots \vee \neg xn=yn \vee f(x1,\ldots,xn)=f(y1,\ldots,yn)]$
EQAX3 (Alternative) $\forall[\neg x1=y1 \vee \ldots \vee \neg xn=yn \vee \neg P(x1,\ldots,xn) \vee P(y1,\ldots,yn)]$

**Exercise (a jolly good one!):** Show that the two forms of EQAX are equivalent.
Hint: To show EQAX2(Alternative) implies EQAX2 (and similarly for EQAX3) is easy. You need to use Reflexivity. The other direction is a bit harder.

A discussion of models and interpretations of Equality is given later.

## Using the Equality Axioms in Resolution

12biv

Where, in the "natural" derivation on 12ai, are EQAX used?

To derive line 4, which was $T(q,q) \vee T(q,p)$:
Use EQAX3: $\forall x,y,z[\neg x=y \vee \neg T(x,z) \vee T(y,z)]$ + $(p=q)$ + $T(p,q) \vee T(q,p)$

$p=q$ + EQAX3 ==> $\forall z[\neg T(p,z) \vee T(q,z)]$
$\forall z[\neg T(p,z) \vee T(q,z)]$ + $T(p,q) \vee T(q,p)$ ==> $T(q,q) \vee T(q,p)$

**The refutation in full from (1)-(3) on slide 12ai using EQAX:**
Given (1) $T(p,q) \vee T(q,p)$ (2) $\neg T(X,X)$ (3) $p=q$

(4). $\neg S=Z \vee \neg T(S,W) \vee T(Z,W)$ (EQAX3)
(5) (3+4) $\neg T(p,W) \vee T(q,W)$
+ (1) => $T(q,q) \vee T(q,p)$
(6) (5+2) $T(q,p)$
(7) $\neg S=Z \vee \neg T(W,S) \vee T(W,Z)$ (EQAX3)
(8) (3+7) $\neg T(W,p) \vee T(W,q)$
+ (1) => $T(q,q)$
(9) (8+2) []

**Note**: intermediate clauses like (5) formed from (4) + (3), or (8) formed from (7) + (3), need not be retained.

## Simulating Paramodulation by Resolution

12bv

*EQAX1* $\forall[x=x]$
*EQAX2* $\forall[\neg xi=yi \vee f(x1,\ldots,xi,\ldots,xn)=f(x1,\ldots,yi,\ldots,xn)]$
*EQAX3* $\forall[\neg xi=yi \vee \neg P(x1,\ldots,xi,\ldots,xn) \vee P(x1,\ldots,yi,\ldots,xn)]$

A resolution simulation of paramodulating $f(x)=x \vee R(x)$ into $P(g(y),y) \vee Q(y))$
to give $P(g(x),f(x)) \vee Q(f(x)) \vee R(x)$

$f(x)=x \vee R(x)$

EQAX2
$\neg u=v \vee g(u)=g(v)$

$R(x) \vee g(f(x))=g(x)$

EQAX3
$\neg u=v \vee \neg P(u,z) \vee P(v,z)$

$R(x) \vee \neg P(g(f(x),z) \vee P(g(x),z)$

$P(g(y),y) \vee Q(y)$

$P(g(x),f(x)) \vee Q(f(x)) \vee R(x)$

Notice that EQAX2 is used first to obtain an equation with a term matching the "to" term (it's g(y) here).

Then EQAX3 is used to make the replacement

Simulation of this kind shows soundness of paramodulation. WHY?

## Equality Axioms also hold for the "=" predicate

12bvi

Can show that EQAX1 and EQAX3 ⇒ symmetry of '='.

1. X=X (EQAX1)
2. ¬U=V ∨ ¬U=Z ∨ V=Z (EQAX3)
   (¬U=V ∨ ¬P(U,Z) ∨ P(V,Z) put = for 'P' )
3. a=b
4. ¬(b=a) (3 and 4 from ¬∀x∀y [x=y → y=x]
5. (2+4) ¬U=b ∨ ¬U=a
6. (5+3) ¬a=a
7. (6+1) []

---

Transitivity can be shown similarly.

EQAX3 ⇒ transitivity of '='.

1. ¬U=V ∨ ¬Z=U ∨ Z=V (EQAX3)
   (¬U=V ∨ ¬P(Z,U) ∨ P(Z,V) put = for 'P' )
2. a=b
3. b=c
4. ¬(a=c) (2, 3 and 4 from ¬∀x∀y [x=y ∧ y=z → y=x]
5. (1+3) ¬Z=b ∨ Z=c
6. (5+2) a=c
7. (6+4) []

## Hyper-paramodulation - A Paramodulation Strategy

12ci

Can combine resolution refinements with the use of equality axioms
Enables to control the use of equality axioms

eg Paramodulation can be combined with hyper-resolution:

In *Hyper-paramodulation*, Hyper-resolution is used for the resolution steps and is forced on the use of EQAX. This leads to some restrictions:

(a) Can only use X=Y if it is an atom in an *electron*.

(b) Can only paramodulate <u>into</u> an *electron*.

- May need specific *instances* of EQAX1 - e.g. f(x) = f(x), g(x,y) = g(x,y),
  or must allow explicit use of EQAX2.

**Example1**: (1) a<b ∨ a=b (2) ¬ a<c (3) b<c (4) ¬x<y ∨ ¬y<z ∨ x<z
(5) 1+3+4: a=b ∨ a<c}
(6) P: 5+3: a<c ∨ a<c ==> a<c (factor) (replace b in b<c by a i.e. use a=b as b=a)
(7) 6+2: []

## Hyper-paramodulation (contd) (ppt)

12cii

In *Hyper-paramodulation*, Hyper-resolution is used for the resolution steps and is forced on the use of EQAX. There are some restrictions:

- Can only use X=Y if it is an atom in an *electron*.
- Can only paramodulate <u>into</u> an *electron*.
- May need specific *instances* of EQAX1 - e.g. f(x) = f(x), g(x,y) = g(x,y),
  or must allow explicit use of EQAX2.

**Example 2**: (1) a=b (2) ¬P(f(a),f(b) ) (3) P(x,x) (4) x=x

**Note**: (5) P: 1+2: ¬P(f(b),f(b)) % would violate restriction (b)

(6) P: 1+3: P(a,b) % unify 2nd x in (3) with a and then replace by b

Then STUCK! % Either need (4a) f(x)=f(x) or use of EQAX2 +(1)

(7) P:1+4a: f(a)=f(b) % either unify second x in (4a) with a and then replace by b
% or apply EQAX2 using (1)

(8) P: 7+3: P(f(a),f(b)) % unify 2nd x in (3) with f(a) and then replace by f(b)

(9) 8+2: []

## How do the restrictions for Hyper-paramodulation arise?

12ciii

a) Can only use X=Y if it is an atom in an electron.
b) Can only paramodulate into an <u>*electron*</u>.
c) May need specific *instances* of EQAX1 - e.g. f(x) = f(x), g(x,y) = g(x,y),
   or must allow explicit use of EQAX2.

• EQAX3 (eg ¬x=y ∨ ¬ P(…,x,…) ∨ P(…,y…)) is a nucleus – needs 2 electrons; one electron must be the one in which a=b occurs and the other must be the one in which P(…,a,…) occurs. This enforces the two restrictions (a) and (b)

• EQAX2 (eg ¬x=y ∨ f(x) =f(y)) is also a nucleus and needs 1 electron; that must be the one in which a=b occurs; helps enforce (a)
(Remember:
EQAX2 enables terms to be built up for substitution at argument level)

• (c) is caused by (b);
eg cannot make ¬P(f(a),f(b)) into ¬P(f(b),f(b)) using a=b
(say in order to match P(x,x)),
so must derive P(f(a),f(b)) instead from P(x,x); (so can match with ¬P(f(a),f(b)))
and this requires to derive f(a)=f(b) from a=b
either by EQAX2, or from f(x)=f(x) and paramodulation

Can then use paramodulation (match f(a) from f(a)=f(b) with 2nd x in P(x,x))

## RUE-Resolution (Digricoli,Raptis) (Uses the alternative form of EQAX)

*EQAX2 (Alt)* ∀[¬x1=y1 ∨...∨ ¬xn=yn ∨ f(x1,…,xn) = f(y1,…,yn)]
*EQAX3 (Alt)* ∀[¬x1=y1 ∨...∨ ¬xn=yn ∨ ¬L(x1,…,xn) ∨ L(y1,…,yn)]

Given C1≡L(t1,...,tn) ∨ D and C2≡¬L'(t1',...,tn') ∨ E
the RUE-resolvent is D ∨ E ∨ ¬t1=t1' ∨ .... ∨ ¬tn=tn'
where, in EQAX3,
L(t1,...,tn) unifies with L(x1,...,xn) and L'(t1',...,tn') unifies with L(y1,...,yn)

RUE forces a kind of *locking* on use of alternative EQAX
The locking gives ¬x1=y1, ..., ¬xn=yn higher indices than other literals

**Informal example**:
P(a)∨D, ¬P(b) and ¬x=y ∨ ¬P(x) ∨ P(y) (ie C1, C2 and EQAX3) ==> D ∨ ¬a=b

To match P(a) and ¬ P(b) (to resolve C1 and C2) must show a = b.
The goal "show a=b" is represented by ¬a=b
and it is refuted **after** matching P(a), P(b)

12civ

## RUE-Resolution (Contd.) (ppt)

12cv

Given C1≡L(t1,...,tn) ∨ D and C2≡¬L'(t1',...,tn') ∨ E
the RUE-resolvent is D ∨ E ∨ ¬t1=t1' ∨ .... ∨ ¬tn=tn'

**Example1**: (1) a<b ∨ a=b (2) ¬a<c (3) b<c (4) x<z ∨ ¬y<z ∨ ¬x<y
(5) RUE (2+3): ¬a=b ∨ ¬c=c (6) (5+1): a<b ∨ ¬c=c
(7) (4+6+3): a<c ∨ ¬c=c (9) (7+2): ¬c=c (10) (9+reflex): [ ]

**Example2**: (1)P(x,x,a) (2) ¬P(b,y,y) (3) a=b
(4) RUE: (1+2): ¬x=b ∨ ¬x=y ∨ ¬a=y (Match arguments)
Now there are several solutions:
Solution 1: x==b, y==b (match with EQAX1on lits 1 and 2 and (3) on lit 3)
Solution 2: x==b, y==a (match with EQAX1 on lits 1 and 3, use (3) as b=a)
Solution 3: x==a, y==b (match with (3) for all literals)
Solution 4: x==a, y==a (match with EQAX1 on last 2 literals)

Can also use some simplification steps to reduce literals of the form ¬t1=t2
eg ¬f(a)=f(b) can reduce to ¬a=b by EQAX2 implicitly
¬x=a can reduce to x==a by EQAX1 implicitly
**These are optional steps.**

**Notes on Hyper-paramodulation** 12cvi

The simulation of Hyper-paramodulation using Hyper-resolution (HR) and equality axioms shows soundness of paramodulation. For completeness, we'd like to show that a hyper-paramodulation refutation can be constructed from a HR refutation using also EQAX.

Suppose there is a HR refutation using EQAX. Then
Use of EQAX3 simulates a hyper-paramodulation step already
Use of EQAX2 can also be turned into a hyper-paramodulation step using reflexive axioms such as f(x)=f(x). **(Details an exercise.)**

**Notes on RUE-resolution**
RUE-resolution is an alternative to paramodulation as a way of including EQAX implicitly into the deduction. It can, informally, be interpreted as trying to impose locking onto the use of equality axioms. It is as though some kind of locking strategy is applied to EQAX3 such that the non-equality literals must be resolved (with other clauses) before any other useful resolvents can be made using these axioms. i.e. the equality literals are locked highest in EQAX3. The alternative form of EQAX3 (and EQAX2) are the most appropriate to use here. That is:

EQAX2 (Alternative) ∀[¬x1=y1 ∨ ... ∨ ¬xn=yn ∨ f(x1,…,xn)=f(y1,…,yn)]
EQAX3 (Alternative) ∀[¬x1=y1 ∨ ... ∨ ¬xn=yn ∨ ¬P(x1,…,xn) ∨ P(y1,…,yn)]

Thus the basic step is to match two potentially complementary literals with the two "P" literals in the appropriate EQAX3 schema. The result is a disjunction of inequalities, which can then be resolved with either EQAX1, EQAX2, or equations in the data.

12cvii

**Notes on RUE-resolution (continued)**

If the RUE-resolvent includes an equality ¬t1=t2 such that t1 and t2 are not different constants, then further simplifications may be applied using either EQAX1 or EQAX2.

For instance:
If t1 and t2 are identical terms, then resolve with EQAX1.

If t1 and t2 are functional terms f(x1,...,xn) and f(y1,...,yn) (and there is no appropriate positive equality matching ¬t1=t2), then resolve with the EQAX2 (for f) to get ¬x1=y1∨... ∨¬xn=yn. Can possibly apply further simplifications to each of the inequalities so introduced.

If t1 or t2 is a variable, then could resolve with EQAX1 to instantiate the variable. But note that ¬t1=t2 might also resolve with some other equality present in the data.

In all 3 cases the original inequality will be eliminated.

**Exercise (good one):**
Compare the use of RUE-resolution and Paramodulation for the 3 clauses
(1) P(x,x,a), (2) ¬P(b,y,y), (3) b=a.

## SOME PROPERTIES OF EQAX

12di

- Useful models are those in which '=' satisfies EQAX at ground level.
- An **E-interpretation** is an H-interpretation HI , which satisfies:
  t=t is true in HI for all t in the Herbrand Universe
  if s=t is true in HI then t=s is true in HI
  if s=t and t=r are true in HI then s=r is true in HI
  if s=t is true in HI then f(s)=f(t) is true in HI for every functor f
  (and similarly generalised to functors of arity > 1)
  if s=t and L[s] are true in HI then L[t] is true in HI
- S is **E-unsatisfiable** if S has no E-interpretations.
- **(Corollary)** S is E-unsatisfiable iff S+EQAX is unsatisfiable.
- **Completeness Result:** (Peterson 1983) If S is E-unsatisfiable, then [] can be derived from S ∪ {X=X} by paramodulation and resolution.
- Paramodulation allows the properties of '=' to be taken into account implicitly and to avoid using them explicitly.
- **(Theorem)** A set of clauses S is E-unsatisfiable iff S has no models in which '=' is interpreted as the identity relation (called normal models). (See Optional material for these slides)

## Summary of Slides 12

12ei

1. The use of equality is ubiquitous in every day reasoning. It uses the natural rule of substitution. Given an equality atom such as p=q, occurrences of p may be repalced by q (or vice versa) in any context.

2. Equality reasoning implicitly makes use of equality axiom schema. We called these schema EQAX1 (Reflex), EQAX2 (for building up equations between terms) and EQAX3 (for substitution).

3. In resolution theorem provers the natural rule of equality substitution is generalised to paramodulation, in which the equality may be one disjunct of a clause, and may involve variables, both in the equality and/or the context.

4.Paramodulation leads to a large increase in the search space, especially when equalities have variables, since they will match many contexts. e.g. given f(x)=x, even if the equality is restricted so that only occurrences of the RHS may be substituted for occurrences of the LHS, there are four places in which the equality can be used in the context P(f(f(y)),y). (**What are they?**)

5. The completeness of paramodulation and resolution states that E-(un)satisfiability can be checked using paramodulation.

6. In the context of H-models and equality, (un)satisfiability is defined through the notion of E-(un)satisfiability and E-interpretations.

7. Ways to control paramodulation have been investigated. *Hyper-paramodulation* is one way, in which hyper-resolution restrictions are imposed on the use of equality substitution axioms, as well as the ordinary clauses. These restrictions constrain both the equality used to provide the substitution and the literal being substituted into to belong to an electron. For completeness, functional instances of EQAX1 (Reflex) may be needed.

8. A second control method is *RUE resolution*, in which the equality literals in equality axioms (EQAX3) are always the last literals to be resolved upon. This enforces resolution on the two "P" literals in such axioms, which results in ``matching'' the literals and generating negative equality literals that can be interpreted as goals to be derived. e.g. P(f(f(y)),y) can be RUE-resolved with ¬P(f(a),a): first match corresponding terms: f(f(y))=f(a) and y=a and then set them as goals (i.e. negate them) yielding ¬f(f(y))=f(a) ∨ ¬y=a, which gives ¬f(y)=a ∨ ¬y=a. These have to be proved from the given data.

12eii

# START of OPTIONAL MATERIAL (SLIDES 12)

Notes on Normal Models
Equality in Otter

12fi

**Models including the Equality Literal: Notes on Normal Models(1)**

***Herbrand Models and Equality***: You may already have come across interpretations and satisfiability when equality is involved and learned that "=" is normally interpreted as identity. That is, Val("="(p,q)) = Val(I("=")(I(p), I(q))) holds iff I(p) = I(q). That is, iff "p" and "q" are mapped to the same domain element. On slide 12aii we apparently found a model because we did not observe this convention.

Now, you have also learned that by definition a H-interpretation maps each term to its name, and thus "p" and "q" will map to different domain elements and the above convention cannot be incorporated in the context of H-models and H-interpretations. (Recall that I("p") = p and I("q")=q, and p and q are unique domain elements. This is why, in the context of resolution and Herbrand interpretations we use the equality axioms instead. The theorem on slide 12di states that the two approaches are equivalent for refutations.

Standard approaches to incorporating equality in tableau and first order logic introduce the notion of *normal* models, in which the equality predicate ***is*** interpreted as identity. I.e. if p=q is true, then p and q must be interpreted as the same domain element. Consider again the clauses on slide 12ai: (1) T(p,q)∨T(q,p) (2) ¬T(X,X) (3) p=q

Let I(p)=I(q) = s (say). Then Val(p=q) = Val(I(=)(I(p),I(q))) = Val(I(=)(s,s)). If I(=) is identity, then I(=)(s,s) is True. Hence Val(T(p,q)) = Val (I(T)(s,s)) = Val (T(q,p)) = Val(T(p,p)) = Val(T(q,q)). Then (1) and (2) cannot both be true

12fii

**Notes on normal models (2):**

**Justification of Corollary on Slide 12di**:
We show the contrapositive: S is E-satisfiable iff S+EQAX is satisfiable.

(if case:) Let M be (any) model of S+EQAX; then there is also a H-model of S+EQAX (see slides 4). But this is an E-interpretation by definition, so S is E-satisfiable.

(only if case:) On the other hand, suppose S is E-satisfiable and let M be an E-interpretation that satisfies S; then M also satisfies the EQAX by definition.

(**Proof outline of Theorem on Slide 12di**)

**(Theorem)** A set of clauses S is E-unsatisfiable iff S has no models in which '=' is interpreted as the identity relation (called normal models).

(only if case:) Suppose S is E-unsatisfiable - then S+EQAX are unsatisfiable and S has no normal model, for such a model would violate the assumption.

(if case:) On the other hand, if S +EQAX are satisfied by some model M, i.e. they are E-satisfiable, then S+EQAX have a H-model H; this H is therefore an E-interpretation. E-satisfiable clauses S have normal models as well, formed by considering the equivalence classes imposed by the given equalities as domain elements and constructed from H (see Chapter notes on paramodulation on my webpage www.doc.ic.ac/~kb for the construction details). See example on next slide.

12fiii

**Notes on normal models (3):**

**Example**: Given: S is the set of facts p=q, T(p,q), ¬T(X,X).

Suppose T(p,q), p=q, q=p, p=p, q=q are true and T(p,p), T(q,p), T(q,q) are false.

This is ***not*** an E-interpretation as it doesn't satisfy the following instance of EQAX3: ¬p=q ∨ ¬T(p,q) ∨ T(q,q).

Now let S' be S without ¬T(X,X) (i.e. S' is the set of facts p=q and T(p,q)).
Suppose all atoms are true, then both facts in S' are true in this (Herbrand) E-interpretation.
However, it is not a normal model as it satisfies p=q, yet p and q are not mapped to the same domain element.

A normal model M for S' might use the domain {d}, and the mapping p->d, q->d.

Suppose M sets T(d,d) true and interprets "=" as the identity relation (i.e. d=d is true).

M satisfies p=q (which is interpreted as d=d), and clearly satisfies the equality axioms.

In general, to obtain a normal model must ensure that all terms that are equal to one another, i.e. in the same equivalence class, are mapped to the same domain element. The domain of the normal model consists of the names of the equivalence classes (c.f. d in the example.)

12gi

**Using Equality in Otter**

Otter includes many settings for controlling paramodulation. It is possible to restrict whether paramodulation is allowed **from** either the left or the right sides of an equality literal (para_from_left, para_from_right), or **to** either the left or the right sides of an equality literal (para_into_left, para_into_right), (all set by default).

Paramodulation from variables and into variables can be controlled (para_from_vars, para_into_vars), (both clear by default).

Paramodulation from or into units can be enforced (para_into_units_only, para_from_units_only), (both clear by default).

There are some other flags which are useful if orderings have bee imposed, in that equalities are ordered. Used in conjunction with the para_from_left and para_from_right flags enables the search space to be quite reduced.

Of course, completeness is not guaranteed if certain steps are forbidden. But the search space is much reduced, and often a refutation can still be found. The Otter manual gives full details.