# AUTOMATED REASONING

## SLIDES 4:

## FORMAL NOTIONS

**Structures and Models**
**Herbrand Interpretations and Models**
**Soundness of Resolution**

**NOTE:**
**Slide 4avii has been updated. Changes are indicated in RED**

**KB - AR - 13**

---

**Properties of Inference Systems:** 4ai

Slides 4 include some material on the properties of inference systems, including material on first order structures. The notion of a Herbrand interpretation, a first order structure with a very particular domain, is introduced and it's explained why Herbrand Interpretations are important for soundness and completeness of resolution. The “Useful Theorem” on Slide 4bii and the Skolemisation property on optional 4di capture this. These properties mean that when proving theorems about resolution it is sufficient to restrict considerations to Herbrand interpretations only, substantially simplifying the proofs. Also, when using refutation as a proof technique to show (un)satisfiability of data, it is sound to consider the clausal form representation of the data.

The proofs of the theorems in Slides 4 (if not given here) can be found either in Appendix1 or in Chapter Notes 1 at www.doc.ic.ac.uk/~kb. ***Note that none of Appendix1 is examinable***.

Note that it is sufficient for this course to understand about Herbrand structures, but if you are familiar with first order structures, you can check your understanding using the following example.

Take Domain = Lists over the (English) alphabet and the following mapping of terms to Domain
a is "a", b is "the", c is "hit"
f(x) is the word formed by appending 's" to x
P(x) is true if x is a correct English word

a) Which of $\forall x\, (P(x) \rightarrow P(f(x)))$, P(a) or P(b) are true in this structure?
b) Choose interpretations for g and Q that make Q(c,g(c)) true in the structure.

The notation $val(S)_I[x/d]$ may be used to denote the valuation of the sentence S in the structure I in which free occurrences of x in S are replaced by Domain element d. For example, $val(P(x))_I[x/'a']$ means " val ((interpretation of P in I)('a'))", which for the structure I given above is " val('a' is a correct English word)". Note that $\forall x.S$ is true in structure I if $val(S)_I[x/d]$ is true in I for every d in the Domain, and $\exists x.S$ is true in structure I if $val(S)_I[x/d]$ is true in I for some d in the Domain.

Note: in $\forall x(P(x) \rightarrow Q(x,y))$ "x" is not free as it is bound by $\forall x$, but "y" is free as it is not bound.

---

## Structures (see ppt) 4aii

- (First order) sentences are written in a language L, which uses formulas and terms constructed from names in the signature Sig(L) = <P, F, C>, where P = predicates, F = function symbols, C = constants.

**Example:** Let Sig(L) = <{P,Q},{f},{a,b}>
S = {$\forall x\, (P(x) \rightarrow P(f(x)))$, P(a), P(b), Q(a, f(a))}
a and b are constants, f has arity 1, P has arity 1 and Q has arity 2

- A structure (or interpretation) I for L consists of
  a non-empty domain D, and
  an interpretation (i.e. a meaning) for each symbol in Sig(L):
- $c \in C$ is interpreted by an element I(c) of D
- f (of arity n) $\in$ F is interpreted by a function I(f) of arity n from $D^n$ to D
- p (of arity m) $\in$ P is interpreted by a relation I(P) of arity m on $D^m$

**Structure 1**: Domain = {integers} = Int
- I(a) = 0, I(b) =2
- I(f) is the function Int -> Int, where x -> x+2 (i.e. the "add 2" function)
- I(P) = even (ie P(x) is true iff x is even)
- I(Q) = less (ie Q(x,y) is true iff x is less than y)

---

## Structures (continued) 4aiii

**Structure 1 again**: Domain = {integers}
I(a) = 0, I(b) =2 I(f) is the function x -> x+2
I(P) = even I(Q) = less

The interpretation I is extended to apply to all ground terms in language L:

I(f(t1, ..., tn)) = I(f)(I(t1), ..., I(tn)) for a functional term of arity n

Also, I(x) =x for a bound variable x

I(P) allows to give a valuation for all ground atoms

val(P(t1, ..., tn)) = val(I(P)(I(t1), ..., I(tm))) for an atom of arity m

val(P(a)) = val(even(0)) = True
val(P(b)) = val(even(2)) = True
val($\forall x\, (P(x) \rightarrow P(f(x)))$) = val($\forall x\, (even(x) \rightarrow even(x+2))$) = True

## Structure 2: a Herbrand Structure

4aiv

Given: Sig(L) = <{P},{f},{a,b}>   S = {∀x (P(x) → P(f(x))),   P(a) ,   P(b)}

**A Herbrand Structure for Sig(L):**

- Domain ={<u>a</u>,<u>b</u>,<u>f(a)</u>,f(b), …, <u>f(f(a))</u>,f(f(b))…,}
  (i.e. the set of *names* of ground terms in L)
- I(a) = <u>a</u>,   I(b) = <u>b</u>,   I(f) = <u>f</u>,   hence I(f(a)) = I(f)(I(a)) = <u>f(a)</u>

**i.e.** terms of the language are, in effect, interpreted as themselves

**NOTE: mapping of constants and functors is fixed in a Herbrand structure**

- I(P) = <u>P</u>

The valuation of atoms can be given explicitly:

**here's one example**: val(P(a)) = val(I(P)(I(a))) = val( <u>P(a)</u>) = True

Value of atoms of the form P($f^n$(a)) = True;  val(P($f^n$(b))) = val(P(b) = False

i.e. <u>P(a)</u> = <u>P(f(a))</u> = <u>P(f(f(a)))</u> =… = True and <u>P(b)</u> = <u>P(f(b))</u> = … = False

A ***Herbrand interpretation*** is represented by a subset of the set of atoms:
for above example we get { P(a), P(f(a)), P(f(f(a))), ..... } (the true atoms)

In this interpretation P(a) and ∀x[P(x) → P(f(x))] are true, but P(b) is false.

## Truth in Structures

4av

• The truth of a sentence S written in L under interpretation I is defined by:

S is a ground atom P(t1,...,tn): S is true iff val(I(P)(I(t1),...,I(tn))) = True

S =¬S1: S is true iff val(S1) = False

S = S1 op S2: S is true iff val(S1) op val(S2) is true (op in {∨, →, ∧, ↔})

S = ∀x(S1): S is true iff val(I(S1)[x/d]) is true for every d in D

S = ∃x(S1): S is true iff val(I(S1)[x/d]) is true for some d in D

I(S1)(x/d) means d replaces occurrences of x in interpreted atoms in S1

A structure I for L is a *model* for a set of sentences S (written in L) if
for every sentence s in S, val(S) is true under I

If S has a model it is *satisfiable*. If S has no models S is *unsatisfiable*.

Exercise: what is the truth of ∃y∀x(P(x,y)) and ∀u∃v(P(u,v)) in the structure with Domain the set of Integers where I(P) = greater-than (ie P(x,y) means x >y)?

## Truth in Structures (Example)

4avi

Given: Sig(L) = <{P},{f},{a,b}>

What is the truth of ∀x (P(x) → P(f(x))) under the Herbrand interpretation "H1" that assigns True to <u>P(a)</u>, <u>P(f(a))</u>, <u>P(f(f(a)))</u>, etc. and false to other atoms?

Answer:

val(∀x (P(x) → P(f(x))) under "H1" = True if
val( H1(∀x (P(x) → P(f(x))) [x/<u>a</u>] ) = val( H1(∀x (P(x) → P(f(x))) [x/<u>f(a)</u>] ) = ....
= val( H1(∀x (P(x) → P(f(x))) [x/<u>b</u>] ) = val( H1(∀x (P(x) → P(f(x))) [x/<u>f(b)</u>] ) = ....
= True

val( H1(∀x (P(x) → P(f(x))) [x/<u>a</u>] ) = val( H1(P)(x) → H1(P)(H1(f(x)) [x/<u>a</u>] )
= val( <u>P(x)</u> → <u>P(f(x))</u> [x/<u>a</u>] )
= val( <u>P(a)</u> → <u>P(f(a))</u> ) = True

Similarly for the other cases.

This is clearly very cumbersome!!

We simplify notation when Herbrand structures are used for clausal form, and, with a slight abuse of notation make the following definitions.

## Herbrand Interpretations for clauses (Simplified)

4avii

We identify the terms in the language with their counterparts in the Herbrand structure and write val(S) instead of val(H(S)) for a Herbrand structure H.

<u>**Some Definitions**</u>**:** Let L be a language for a set of **clauses** S

The *Herbrand Universe* HU of L is the set of terms using constants and function symbols in Sig(L). (It is assumed there is always at least one constant.)

The *Herbrand Base* HB of L is the set of ground atoms using terms from HU

An *Herbrand Interpretation* HI of L is a subset of the atoms in HB which are assumed to be true

An *Herbrand model* of S is an Herbrand interpretation M of L that forces val(S) =True for each **clause** in S under M

A ground instance C' of clause C is obtained by applying a ground substitution θ to C. If θ is {$x_1/t_1$,..., $x_n/t_n$}, where $x_i$ are the variables in C and $t_i$ are ground terms, in C' each occurrence of $x_i$ in C has been replaced by $t_i$.

A clause C is true in H if every ground instance of C is true in H.

If S has a Herbrand model we say S is *H-satisfiable*. If not S is *H-unsatisfiable*.

**NOTE**: If Sig(L) includes any function symbols then the Herbrand Universe is infinite.There is assumed always one constant in HU, so HU ≠ ∅.

## Herbrand Interpretations (Example)

4aviii

Sig(L) = < {P,Q,R,S}, {f}, {a,b} >
S=Px ∨ Ry ∨ ¬Qxy, ¬Sz ∨ ¬Rz, Sa, ¬Pf(a) ∨ ¬Pf(b)
(Notation: Px is shorthand for P(x), Qxy is shorthand for Q(x,y), etc.)

- Herbrand Universe = {a,b,f(a),f(b),f(f(a)),f(f(b)), ...}
- Herbrand Base = {Pa,Pb, Pf(a), Pf(b), ... Sa, Sb, Sf(a), Sf(b), ..., Ra, Rb, Rf(a), ..., Qab, Qaa, Qbb,Qba, Qf(a)a, ....}
- One Herbrand interpretation is
  {Sa, all Q atoms except Qaa and Qbb, Pa, Pb, Pf(a), Pf(b), P(f(f(a)),..., }

meaning (informally)
val(Pa) = val(Pb) = ... = True, val(Sa) = True, val(any other S atom)=False,
val(Qaa) = val(Qbb) = False, **val(any other Q atom) = True**
and val(any R atom) = False

- This is **not** a Herbrand *model* of S because val(¬Pf(a) ∨ ¬Pf(b)) = False.
- The HI = { "S" atoms } (i.e. S atoms true, other atoms false) **is** a model of S.

## Soundness and Completeness Properties

4bi

Given a first order language L and a set of clauses C written in L:
C |= ⊥ - (A is inconsistent) means that
there is no structure M (of L) that is a model of C,

There are an infinite number of structures to check using this definition
So how else could we check C |=⊥?

We'll use resolution to derive []
C ⇒* [] - means [] can be deduced from C using resolution

How do we know we get the correct answers?
The two relations |= and ⇒* are equivalent,
as expressed by the *Soundness* and *Completeness* of Resolution:

**Soundness of Resolution** - if C ⇒* ⊥ then C |= ⊥
**Completeness of Resolution** - if C |= ⊥ then C ⇒* ⊥

## Soundness and Completeness of Resolution

4bii

Let C be a set of clauses. Let ⇒* denote "*yields by ≥1 resolution or factor steps*"

**Soundness of Resolution:** if C ⇒* [ ] then C |= ⊥ (or C has no models)

The idea used to show soundness is this:
Show that for **each** resolution step
if the parent clauses are true in a structure then so is the resolvent,
hence, by transitivity, if the initial clauses are true so are all the resolvents
Then, if a resolvent is clearly not true in any structure,
we can conclude the given clauses are not true either.

We need (1) **Useful Theorem (*)** which states that
S has a Herbrand model iff S has some model
≡ S has no Herbrand models iff S has no models
Hence to show C |= ⊥ it is sufficient to show C $|=_H$ ⊥

and (2) a single resolution or factoring step is sound with respect to H-models:
if S ⇒ R then S $|=_H$ R (where R is a resolvent or factor from S)

where S $|=_H$ R holds iff
for every H-model M, if M is a model of S then M is a model of R
(Details and proofs of (1) and (2) are in Slides A1b in Appendix 1).

## Proving Soundness of Resolution

4biii

**Soundness of Resolution:** if C ⇒* [ ] then C |= ⊥ (or C has no models)

**We argue as follows:**

for a refutation C ⇒ C+R1 ⇒ C+R1+R2 ⇒ ... ⇒ C+ ... + [ ]

**we have by (2) from Slide 4bii (==> reads as implies)**

C has H-model ==> C+R1 has H-model ==> C+R1+R2 has H-model ==>... ==> C+R1+...+Rn has H- model ≡

C+R1+...+Rn has no H-model ==> C+R1+...+Rn-1 has no H-model ==>... ==> C+R1 has no H-model ==> C has no H-model

Now suppose that resolvent Rn is the empty clause []:
since [] has no models, C+R1+...+Rn has no H-models,
hence .... C has no H-models.

**Hence by (1) from Slide 4bii** C has no models at all

**Completeness of Resolution:** if C |= ⊥ then C ⇒* [ ]

See Slides 5

## Summary of Slides 4

4ci

**1.** Herbrand interpretations are first order structures which use a fixed mapping between terms in the Language and the domain of the structure. In particular, terms (constants or functional terms such as f(a)) map to (the names of) themselves.

**2.** Any set of sentences S has a model iff S has a Herbrand model.

**3.** Resolution is sound and complete: Derivation of [ ] from a set of clauses S by resolution and factoring implies that $S \models \bot$ , and if $S \models \bot$ then there is a resolution (and factoring) derivation of [ ] from S.

**4.** Soundness of resolution depends on the soundness of a single resolution or factoring step: if $S \Rightarrow R$ then $S \models_H R$ and hence $S \models_H S+\{R\}$.

## START of OPTIONAL MATERIAL (SLIDES 4)

Resolution and the General case - given sentences not clauses

## The General Case

4di

In general, we'd like to show that we can use resolution to show unsatisfiability of any set of sentences.

In optional material on Slides 3d, conversion to clauses used Skolemisation (Step 3)

3. *Skolemise* - existential-type quantifiers are removed and bound variable occurrences of x in ∃xS are replaced by *Skolem constants* or *Skolem functions*

Now, all non-*Skolemisation* steps in the conversion to clausal form are equivalences

But, although Skolemised(S) $\models$ S, it is **not** true that S $\models$ Skolemised(S).

eg f(a) |= ∃x.f(x) – if f(a) is true then there is an x (namely a) s.t. f(x) is true.

But ∃x.f(x) does not imply f(a). Whatever the x that makes f(x) true, it need not be a.

However, it **is** true that Skolem(S) is unsatisfiable iff S is unsatisfiable. **(**)**
And this is what we need. (See Slides A1c in Appendix 1 for proof.)

In general:
To show Data |=Conclusion we convert {Data, ¬Conclusion} to clauses C.
Then
Data |=Conclusion iff {Data,¬Conclusion} is unsatisfiable (by definition)
iff C has no models (by (**) above)
iff C has no H-models **(by Useful Theorem (*) on 4bii)**
iff C⇒* [ ] (by Soundenss and Completeness of resolution)