

## CS157 Greatest Hits

### The Top 100 Slides of the Course

## Syntax

Propositional Constants: *raining, snowing, cloudy*

Negations:  $\neg$ *raining*

Conjunctions: (*raining*  $\wedge$  *snowing*)

Disjunctions: (*raining*  $\vee$  *snowing*)

Implications: (*raining*  $\Rightarrow$  *cloudy*)

Reductions: (*cloudy*  $\Leftarrow$  *raining*)

Equivalences: (*cloudy*  $\Leftrightarrow$  *raining*)

Nesting:  $((\textit>raining} \vee \textit{snowing}) \Rightarrow \textit{cloudy})$

## Propositional Interpretation

A *propositional interpretation* is an association between the propositional constants in a propositional language and the truth values T or F.

$$p \xrightarrow{i} \text{T}$$

$$q \xrightarrow{i} \text{F}$$

$$r \xrightarrow{i} \text{T}$$

$$p^i = \text{T}$$

$$q^i = \text{F}$$

$$r^i = \text{T}$$

12/6/09

3

## Sentential Interpretation

A *sentential interpretation* is an association between the sentences in a propositional language and the truth values T or F.

$$p^i = \text{T}$$

$$q^i = \text{F}$$

$$r^i = \text{T}$$

$$(p \vee q)^i = \text{T}$$

$$(\neg q \vee r)^i = \text{T}$$

$$((p \vee q) \wedge (\neg q \vee r))^i = \text{T}$$

A propositional interpretation defines a sentential interpretation by application of operator semantics.

12/6/09

4

## Operator Semantics

$\phi$	$\neg\phi$
T	F
F	T

$\phi$	$\psi$	$\phi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

$\phi$	$\psi$	$\phi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

$\phi$	$\psi$	$\phi \Rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

$\phi$	$\psi$	$\phi \Leftarrow \psi$
T	T	T
T	F	T
F	T	F
F	F	T

$\phi$	$\psi$	$\phi \Leftrightarrow \psi$
T	T	T
T	F	F
F	T	F
F	F	T

12/6/09

5

## Evaluation

Interpretation  $i$ :

$$\begin{aligned}
 p^i &= T \\
 q^i &= F \\
 r^i &= T
 \end{aligned}$$

Compound Sentence

$$(p \vee q) \wedge (\neg q \vee r)$$

12/6/09

6

## Truth Tables

A *truth table* is a table of all possible interpretations for the propositional constants in a language.

<i>p</i>	<i>q</i>	<i>r</i>
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

One column per constant.

One row per interpretation.

For a language with  $n$  constants, there are  $2^n$  interpretations.

12/6/09

7

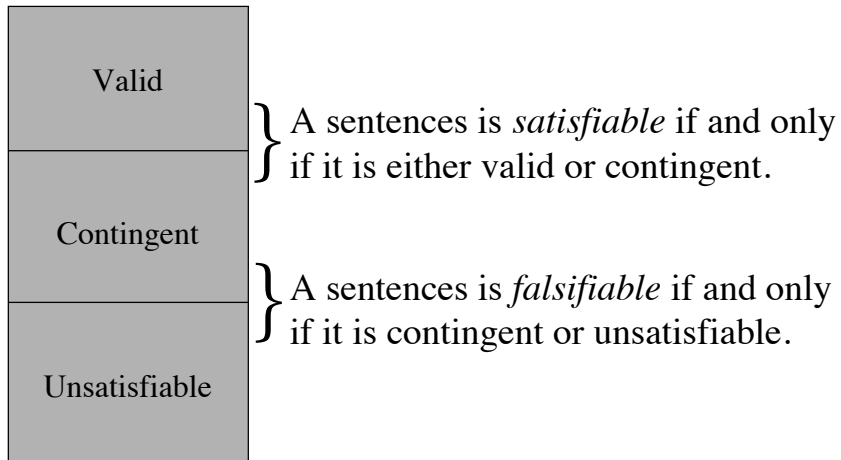
## Properties of Sentences

Valid	A sentence is <i>valid</i> if and only if <i>every</i> interpretation satisfies it.
Contingent	A sentence is <i>contingent</i> if and only if <i>some</i> interpretation satisfies it and <i>some</i> interpretation falsifies it.
Unsatisfiable	A sentence is <i>unsatisfiable</i> if and only if <i>no</i> interpretation satisfies it.

12/6/09

8

## Properties of Sentences



12/6/09

9

## Evaluation Versus Satisfaction

Evaluation:

$$\begin{array}{l} p^i = T \\ q^i = F \end{array} \longrightarrow \begin{array}{l} (p \vee q)^i = T \\ (\neg q)^i = T \end{array}$$

Satisfaction:

$$\begin{array}{l} (p \vee q)^i = T \\ (\neg q)^i = T \end{array} \longrightarrow \begin{array}{l} p^i = T \\ q^i = F \end{array}$$

12/6/09

10

## Logical Entailment

A set of premises  $\Delta$  *logically entails* a conclusion  $\varphi$  (written as  $\Delta \models \varphi$ ) if and only if every interpretation that satisfies the premises also satisfies the conclusion.

$$\{p\} \models (p \vee q)$$

$$\{p\} \not\models (p \wedge q)$$

$$\{p, q\} \models (p \wedge q)$$

12/6/09

11

## Proof (Official Version)

A *proof* of a conclusion from a set of premises is a sequence of sentences terminating in the conclusion in which each item is either:

1. a premise
2. An instance of an axiom schema
3. the result of applying a rule of inference to earlier items in sequence.

12/6/09

12

## Provability

A conclusion is said to be *provable* from a set of premises (written  $\Delta \vdash \varphi$ ) if and only if there is a finite proof of the conclusion from the premises using only Modus Ponens and the Standard Axiom Schemata.

12/6/09

13

## Soundness and Completeness

Soundness: Our proof system is *sound*, i.e. if the conclusion is provable from the premises, then the premises propositionally entail the conclusion.

$$(\Delta \vdash \varphi) \Rightarrow (\Delta \models \varphi)$$

Completeness: Our proof system is *complete*, i.e. if the premises propositionally entail the conclusion, then the conclusion is provable from the premises.

$$(\Delta \models \varphi) \Rightarrow (\Delta \vdash \varphi)$$

12/6/09

14

## Metatheorems

Deduction Theorem:  $\Delta \vdash (\varphi \Rightarrow \psi)$  if and only if  $\Delta \cup \{\varphi\} \vdash \psi$ .

Equivalence Theorem:  $\Delta \vdash (\varphi \Leftrightarrow \psi)$  and  $\Delta \vdash \chi$ , then it is the case that  $\Delta \vdash \chi_{\varphi \leftarrow \psi}$ .

12/6/09

15

## Clausal Form

A *literal* is either an atomic sentence or a negation of an atomic sentence.

$$p, \neg p$$

A *clausal sentence* is either a literal or a disjunction of literals.

$$p, \neg p, p \vee q$$

A *clause* is a set of literals.

$$\{p\}, \{\neg p\}, \{p, q\}$$

12/6/09

16

## Conversion to Clausal Form

Implications Out:

$$\varphi_1 \Rightarrow \varphi_2 \rightarrow \neg \varphi_1 \vee \varphi_2$$

$$\varphi_1 \Leftarrow \varphi_2 \rightarrow \varphi_1 \vee \neg \varphi_2$$

$$\varphi_1 \Leftrightarrow \varphi_2 \rightarrow (\neg \varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg \varphi_2)$$

Negations In:

$$\neg \neg \varphi \rightarrow \varphi$$

$$\neg(\varphi_1 \wedge \varphi_2) \rightarrow \neg \varphi_1 \vee \neg \varphi_2$$

$$\neg(\varphi_1 \vee \varphi_2) \rightarrow \neg \varphi_1 \wedge \neg \varphi_2$$

12/6/09

17

## Conversion to Clausal Form

Distribution

$$\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \rightarrow (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$$

$$(\varphi_1 \wedge \varphi_2) \vee \varphi_3 \rightarrow (\varphi_1 \vee \varphi_3) \wedge (\varphi_2 \vee \varphi_3)$$

$$\varphi_1 \vee (\varphi_2 \vee \varphi_3) \rightarrow (\varphi_1 \vee \varphi_2 \vee \varphi_3)$$

$$(\varphi_1 \wedge \varphi_2) \vee \varphi_3 \rightarrow (\varphi_1 \vee \varphi_2 \vee \varphi_3)$$

$$\varphi_1 \wedge (\varphi_2 \wedge \varphi_3) \rightarrow (\varphi_1 \wedge \varphi_2 \wedge \varphi_3)$$

$$(\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \rightarrow (\varphi_1 \wedge \varphi_2 \wedge \varphi_3)$$

Operators Out

$$\varphi_1 \vee \dots \vee \varphi_n \rightarrow \{\varphi_1, \dots, \varphi_n\}$$

$$\varphi_1 \wedge \dots \wedge \varphi_n \rightarrow \varphi_1, \dots, \varphi_n$$

12/6/09

18

## Resolution Principle

General:

$$\frac{\begin{array}{l} \{\varphi_1, \dots, \chi, \dots, \varphi_m\} \\ \{\psi_1, \dots, \neg\chi, \dots, \psi_n\} \end{array}}{\{\varphi_1, \dots, \varphi_m, \psi_1, \dots, \psi_n\}}$$

Example:

$$\frac{\begin{array}{l} \{p, q\} \\ \{\neg p, r\} \end{array}}{\{q, r\}}$$

12/6/09

19

## Soundness and Completeness

A sentence is *provable* from a set of sentences by propositional resolution if and only if there is a derivation of the empty clause from the clausal form of  $\Delta \cup \{\neg\varphi\}$ .

Theorem: Propositional Resolution is sound and complete, i.e.  $\Delta \models \varphi$  if and only if  $\Delta \vdash \varphi$ .

12/6/09

20

## Davis Putnam Procedure

```

function dp ( $\Delta$ )
  {for  $\varphi$  in vocabulary( $\Delta$ )
    do {var  $\Delta' \leftarrow \{\}$ ;
      for  $\Phi_1$  in  $\Delta$  for  $\Phi_2$  in  $\Delta$ 
        such that  $\varphi \in \Phi_1 \ \neg\varphi \in \Phi_2$ 
          do {var  $\Phi' \leftarrow \Phi_1 - \{\varphi\} \cup \Phi_2 - \{\neg\varphi\}$ ;
            if not tautology( $\Phi'$ ) then  $\Delta' \leftarrow \Delta' \cup \{\Phi'\}$ };
           $\Delta \leftarrow \Delta - \{\Phi \in \Delta \mid \varphi \in \Phi \text{ or } \neg\varphi \in \Phi\} \cup \Delta'$ ;
        return {if  $\{\} \in \Delta$  then unsatisfiable else satisfiable}}
  
```

```

function tautology( $\Phi$ )
  { $\varphi \in \Phi$  and  $\neg\varphi \in \Phi$ }
  
```

12/6/09

21

## Davis-Putnam Example

$\{p, q, r\}$	$\{q, r\}$
$\{p, q, \neg r\}$	$\{q, \neg r\}$
$\{p, \neg q, r\}$	$\{\neg q, r\}$
$\{p, \neg q, \neg r\}$	$\{\neg q, \neg r\}$
$\{\neg p, q, r\}$	
$\{\neg p, q, \neg r\}$	$\{r\}$
$\{\neg p, \neg q, r\}$	$\{\neg r\}$
$\{\neg p, \neg q, \neg r\}$	
	$\{\}$

Cost = 16 + 4 + 1 = 21 resolutions

12/6/09

22

## Davis Putnam Logemann Loveland

```
function dpll ( $\Delta$ )
  {var  $\varphi$ ;
   if  $\Delta = \{\}$  then return yes;
   if  $\{\} \in \Delta$  then return no;
    $\varphi \leftarrow$  choose vocabulary( $\Delta$ );
   if dpll(simplify( $\Delta$ ,  $\varphi$ )) return yes
   else return dpll(simplify( $\Delta$ ,  $\neg\varphi$ ))}
```

12/6/09

23

## Simplification

```
function simplify ( $\Delta$ ,  $\varphi$ )
  {var  $\Delta'$ ;
   for  $\Phi \in \Delta$ 
     do {if  $\varphi \in \Phi$  then skip
        else if negation( $\varphi$ )  $\in \Phi$ 
          then  $\Delta' \leftarrow \Delta' \cup \{\Phi - \{\textit{negation}(\varphi)\}\}$ 
          else  $\Delta' \leftarrow \Delta' \cup \{\Phi\}}$ 
```

Example:

$$\textit{simplify}(\{\{p,q\},\{\neg p,r\},\{\neg r,s\}\}, p) = \{\{r\},\{\neg r,s\}\}$$

12/6/09

24

## Words

*Variables* begin with characters from the end of the alphabet (from  $u$  through  $z$ ).

*Constants* begin with digits or letters from the beginning of the alphabet (from  $a$  through  $t$ ).

*Object constants* refer to objects.

*Function constants* denote functions.

*Relation constants* refer to relations.

There is no syntactic distinction between object, function, and relation constants. The type of each such word is determined from context.

## Functional Terms

A *functional term* is an expression formed from an  $n$ -ary function constant and  $n$  terms enclosed in parentheses and separated by commas.

$$\begin{aligned} &father_1(joe) \\ &age_1(joe) \\ &plus_2(x,2) \end{aligned}$$

Functional terms are terms and so can be nested.

$$plus_2(age_1(father_1(joe)),age_1(mother_1(joe)))$$

12/6/09

27

## Relational Sentences

A *relational sentence* is an expression formed from an  $n$ -ary relation constant and  $n$  terms enclosed in parentheses and separated by commas.

$$\begin{aligned} &happy_1(art) \\ &loves_2(art,cathy) \end{aligned}$$

Relational sentences are *not* terms and *cannot* be nested in terms or relational sentences.

No!  $happy_1(person_1(joe))$  No!

$$\begin{aligned} &happy_1(joe) \\ &person_1(joe) \end{aligned}$$

12/6/09

28

## Logical Sentences

Logical sentences in Relational Logic are analogous to those in Propositional Logic.

$\neg \text{loves}(\text{art}, \text{cathy})$   
 $(\text{loves}(\text{art}, \text{betty}) \wedge \text{loves}(\text{betty}, \text{art}))$   
 $(\text{loves}(\text{art}, \text{betty}) \vee \text{loves}(\text{art}, \text{cathy}))$   
 $(\text{loves}(x, y) \Rightarrow \text{loves}(y, x))$   
 $(\text{loves}(x, y) \Leftarrow \text{loves}(y, x))$   
 $(\text{loves}(x, y) \Leftrightarrow \text{loves}(y, x))$

Parenthesization rules are the same as for Propositional Logic.

12/6/09

29

## Quantified Sentences

Universal sentences assert facts about all objects.

$\forall x.(\text{person}(x) \Rightarrow \text{mammal}(x))$

Existential sentence assert the existence of an object with given properties.

$\exists x.(\text{person}(x) \wedge \text{happy}(x))$

Quantified sentences can be nested within other sentences.

$\forall x.\text{apple}(x) \vee \exists x.\text{pear}(x)$   
 $\forall x.\exists y.\text{loves}(x, y)$

12/6/09

30

## Conceptualization

*Universe of Discourse* - a set  $U$  of objects.

$$\{\circ, \bullet\}$$

*Functional Basis Set* - set  $\{f_1, \dots, f_m\}$  of functions on  $U$ .

$$f_i: U^k \rightarrow U$$

*Relational Basis Set* - set  $\{r_1, \dots, r_n\}$  of relations on  $U$ .

$$r_i \subseteq U^k$$

12/6/09

31

## Interpretations

An *interpretation* is a mapping from the constants of a language into elements of a conceptualization  $\langle U, F, R \rangle$ .

$$i: \text{objconst} \rightarrow U$$

$$i: \text{funconst} \rightarrow F$$

$$i: \text{relconst} \rightarrow R$$

The arity of the function and relation constants must match the arity of their interpretations.

12/6/09

32

## Variable Assignments

An *variable assignment* for a conceptualization  $\langle U, F, R \rangle$  is a mapping of variables into  $U$ .

$$v: \text{variable} \rightarrow U$$

Universe of Discourse:

$$U = \{\circ, \bullet\}$$

Example:

$$v(x) = \circ$$

$$v(y) = \bullet$$

$$v(z) = \bullet$$

Example:

$$v(x) = \bullet$$

$$v(y) = \bullet$$

$$v(z) = \bullet$$

12/6/09

33

## Value Assignments

A *value assignment*  $s_{iv}$  based on interpretation  $i$  and variable assignment  $v$  is a mapping from the terms of the language into the universe of discourse that agrees with  $i$  on constants, that agrees with  $v$  on variables, and that, for functional terms, yields the result of applying the interpretation of the functional constant to the values assigned to the argument terms.

$$s_{iv}(\sigma) = i(\sigma)$$

$$s_{iv}(v) = v(v)$$

$$s_{iv}(\pi(\tau_1, \dots, \tau_n)) = i(\pi)(s_{iv}(\tau_1), \dots, s_{iv}(\tau_n))$$

12/6/09

34

## Truth Assignments

A *truth assignment*  $t_{iv}$  based on interpretation  $i$  and variable assignment  $v$  is a mapping from the sentences of the language into  $\{true, false\}$ .

$$t_{iv}: \text{sentence} \rightarrow \{true, false\}$$

The details of the definition are given on the following slides.

12/6/09

35

## Relational Sentences

A truth assignment satisfies a relational sentence if and only if the tuple of objects denoted by the arguments is a member of the relation denoted by the relation constant.

$$t_{iv}(\rho(\tau_1, \dots, \tau_n)) = \begin{array}{l} true \text{ if } \langle s_{iv}(\tau_1), \dots, s_{iv}(\tau_n) \rangle \in i(\rho) \\ = false \text{ otherwise} \end{array}$$

12/6/09

36

## Logical Sentences

$$t_{iv}(\neg \varphi) = \text{true} \text{ iff } t_{iv}(\varphi) = \text{false}$$

$$t_{iv}(\varphi \wedge \psi) = \text{true} \text{ iff } t_{iv}(\varphi) = \text{true} \text{ and } t_{iv}(\psi) = \text{true}$$

$$t_{iv}(\varphi \vee \psi) = \text{true} \text{ iff } t_{iv}(\varphi) = \text{true} \text{ or } t_{iv}(\psi) = \text{true}$$

$$t_{iv}(\varphi \Rightarrow \psi) = \text{true} \text{ iff } t_{iv}(\varphi) = \text{false} \text{ or } t_{iv}(\psi) = \text{true}$$

$$t_{iv}(\varphi \Leftarrow \psi) = \text{true} \text{ iff } t_{iv}(\varphi) = \text{true} \text{ or } t_{iv}(\psi) = \text{false}$$

$$t_{iv}(\varphi \Leftrightarrow \psi) = \text{true} \text{ iff } t_{iv}(\varphi) = t_{iv}(\psi)$$

12/6/09

37

## Versions

A *version*  $w[v \leftarrow x]$  of a variable assignment  $w$  is the variable assignment that agrees with  $w$  on all variables except  $v$ , which is assigned the value  $x$ .

$$w[v \leftarrow x](\mu) = w(\mu)$$

$$w[v \leftarrow x](v) = x$$

12/6/09

38

## Quantified Sentences

A universally quantified sentence is true in interpretation  $I$  and variable assignment  $\nu$  if and only if the scope is true for  $I$  and every version of  $\nu$ .

$$t_{i\nu}(\forall v.\varphi) = \text{true} \text{ iff } t_{i\nu[v \leftarrow x]}(\varphi) = \text{true} \text{ for all } x \in |I|.$$

An existentially quantified sentence is true in interpretation  $I$  and variable assignment  $\nu$  if and only if the scope is true for  $I$  and some version of  $\nu$ .

$$t_{i\nu}(\exists v.\varphi) = \text{true} \text{ iff } t_{i\nu[v \leftarrow x]}(\varphi) = \text{true} \text{ for some } x \in |I|.$$

12/6/09

39

## Herbrand

The *Herbrand universe* for a set of sentences in Relational Logic (with at least one object constant) is the set of all ground terms that can be formed from just the constants used in those sentences. If there are no object constants, then we add an arbitrary object constant, say  $a$ .

The *Herbrand base* for a set of sentences is the set of all ground atomic sentences that can be formed using just the constants in the Herbrand universe.

12/6/09

40

## Herbrand Theorem

*Herbrand Theorem:* A set of quantifier-free sentences in Relational Logic is satisfiable if and only if it has a Herbrand model.

12/6/09

41

## Modus Ponens

$$\begin{array}{c} \varphi \Rightarrow \psi \\ \varphi \\ \hline \psi \end{array}$$

12/6/09

42

## Universal Generalization

Rule of Inference

$$\frac{\varphi}{\forall v.\varphi}$$

Examples:

$$\frac{p(x)}{\forall x.p(x)}$$

$$\frac{p(x) \Rightarrow q(x)}{\forall x.(p(x) \Rightarrow q(x))}$$

12/6/09

43

## Standard Axiom Schemata

II:  $\varphi \Rightarrow (\psi \Rightarrow \varphi)$

ID:  $(\varphi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \chi))$

CR:  $(\neg\psi \Rightarrow \varphi) \Rightarrow ((\neg\psi \Rightarrow \neg\varphi) \Rightarrow \psi)$   
 $(\psi \Rightarrow \varphi) \Rightarrow ((\psi \Rightarrow \neg\varphi) \Rightarrow \neg\psi)$

EQ:  $(\varphi \Leftrightarrow \psi) \Rightarrow (\varphi \Rightarrow \psi)$   
 $(\varphi \Leftrightarrow \psi) \Rightarrow (\psi \Rightarrow \varphi)$   
 $(\varphi \Rightarrow \psi) \Rightarrow ((\psi \Rightarrow \varphi) \Rightarrow (\varphi \Leftrightarrow \psi))$

OQ:  $(\varphi \Leftarrow \psi) \Leftrightarrow (\psi \Rightarrow \varphi)$   
 $(\varphi \vee \psi) \Leftrightarrow (\neg\varphi \Rightarrow \psi)$   
 $(\varphi \wedge \psi) \Leftrightarrow \neg(\neg\varphi \vee \neg\psi)$

12/6/09

44

## Standard Axiom Schemata (concluded)

UD:  $\forall v.(\varphi \Rightarrow \psi) \Rightarrow (\forall v.\varphi \Rightarrow \forall v.\psi)$

UG:  $\varphi \Rightarrow \forall v.\varphi$   
where  $v$  is not free in  $\varphi$

UI:  $\forall v.\varphi \Rightarrow \varphi[v \leftarrow \tau]$   
where  $\tau$  is free for  $v$  in  $\varphi$

ED:  $\exists v.\varphi \Leftrightarrow \neg \forall v.\neg \varphi$

12/6/09

45

## Freedom

A variable is *free* in a sentence if and only if it occurs outside of the scope any quantifier of that variable.

$x$  is free in  $p(x)$ .

$x$  is free in  $\exists y.p(x,y)$ .

$x$  is free in  $(p(x) \wedge \forall x.q(x))$ .

$x$  is *not* free  $\forall x.p(x)$ .

The statement that a variable is *free* in a sentence is not the same as the statement that a term is *free for* a variable in a sentence!

12/6/09

46

## Substitutability

A term  $\tau$  is *substitutable* for  $v$  in  $\varphi$  if and only if there are no occurrences of  $v$  in the scope of a quantifier of a variable in  $\tau$ .

Some texts say “ $x$  is *free* for  $y$  in  $\varphi$ ” instead of “ $x$  is substitutable for  $y$  in  $\varphi$ ”.

$mother(jane)$  is free for  $y$  in  $hates(jane, y)$ .  
 $mother(x)$  is free for  $y$  in  $hates(jane, y)$ .  
 $mother(x)$  is free for  $y$  in  $\exists z.hates(z, y)$ .  
 $mother(x)$  is *not* free for  $y$  in  $\exists x.hates(x, y)$ .  
 $mother(x)$  is free for  $y$  in  $(\forall x.\forall y.l(x, y) \wedge \exists z.h(z, y))$ .

12/6/09

47

## Formal Proofs

A *formal proof* of  $\varphi$  from  $\Delta$  is a sequence of sentences terminating in  $\varphi$  in which each item is either:

1. a premise (a member of  $\Delta$ )
2. an instance of an axiom schema
3. the result of applying a rule of inference to earlier items in the sequence.

12/6/09

48

## Provability

A sentence  $\varphi$  is *provable* from a set of sentences  $\Delta$  if and only if there is a finite formal proof of  $\varphi$  from  $\Delta$  using only Modus Ponens, Universal Generalization, and the standard axiom schemata.

Soundness Theorem: If  $\varphi$  is provable from  $\Delta$ , then  $\Delta$  logically entails  $\varphi$ .

Completeness Theorem (Godel): If  $\Delta$  logically entails  $\varphi$ , then  $\varphi$  is provable from  $\Delta$ .

12/6/09

49

## Decidability Results

Logical Entailment for Relational Logic is semidecidable.

Logical Entailment for Relational Logic is *not* decidable.

Arithmetic is not finitely axiomatizable in Relational Logic.

12/6/09

50

## Inseado

Implications Out:

$$\varphi_1 \Rightarrow \varphi_2 \rightarrow \neg \varphi_1 \vee \varphi_2$$

$$\varphi_1 \Leftarrow \varphi_2 \rightarrow \varphi_1 \vee \neg \varphi_2$$

$$\varphi_1 \Leftrightarrow \varphi_2 \rightarrow (\neg \varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg \varphi_2)$$

Negations In:

$$\neg \neg \varphi \rightarrow \varphi$$

$$\neg(\varphi_1 \wedge \varphi_2) \rightarrow \neg \varphi_1 \vee \neg \varphi_2$$

$$\neg(\varphi_1 \vee \varphi_2) \rightarrow \neg \varphi_1 \wedge \neg \varphi_2$$

$$\neg \forall v. \varphi \rightarrow \exists v. \neg \varphi$$

$$\neg \exists v. \varphi \rightarrow \forall v. \neg \varphi$$

## Inseado (continued)

Standardize variables

$$\forall x.p(x) \vee \forall x.q(x) \rightarrow \forall x.p(x) \vee \forall y.q(y)$$

Existentials Out (Outside in)

$$\exists x.p(x) \rightarrow p(a)$$

$$\forall x.(p(x) \wedge \exists z.q(x,y,z)) \rightarrow \forall x.(p(x) \wedge q(x,y,f(x,y)))$$

12/6/09

53

## Inseado (continued)

Alls Out

$$\forall x.(p(x) \wedge q(x,y,f(x,y))) \rightarrow p(x) \wedge q(x,y,f(x,y))$$

Distribution

$$\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \rightarrow (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$$

$$(\varphi_1 \wedge \varphi_2) \vee \varphi_3 \rightarrow (\varphi_1 \vee \varphi_3) \wedge (\varphi_2 \vee \varphi_3)$$

12/6/09

54

## Inseado (concluded)

### Operators Out

$$\begin{array}{l} \varphi_1 \wedge \dots \wedge \varphi_n \rightarrow \varphi_1 \\ \dots \\ \varphi_n \\ \varphi_1 \vee \dots \vee \varphi_n \rightarrow \{\varphi_1, \dots, \varphi_n\} \end{array}$$

12/6/09

55

## Unification

A substitution  $\sigma$  is a *unifier* for an expression  $\varphi$  and an expression  $\psi$  if and only if  $\varphi\sigma = \psi\sigma$ .

$$\begin{array}{l} p(X, Y) \{X \leftarrow a, Y \leftarrow b, V \leftarrow b\} = p(a, b) \\ p(a, V) \{X \leftarrow a, Y \leftarrow b, V \leftarrow b\} = p(a, b) \end{array}$$

If two expressions have a unifier, they are said to be *unifiable*. Otherwise, they are *nonunifiable*.

$$\begin{array}{l} p(X, X) \\ p(a, b) \end{array}$$

12/6/09

56

## Most General Unifier

A substitution  $\sigma$  is a *most general unifier (mgu)* of two expressions if and only if it is as general as or more general than any other unifier.

Theorem: If two expressions are unifiable, then they have an mgu that is unique up to variable permutation.

$$p(X, Y) \{X \leftarrow a, Y \leftarrow V\} = p(a, V)$$

$$p(a, V) \{X \leftarrow a, Y \leftarrow V\} = p(a, V)$$

$$p(X, Y) \{X \leftarrow a, V \leftarrow Y\} = p(a, Y)$$

$$p(a, V) \{X \leftarrow a, V \leftarrow Y\} = p(a, Y)$$

12/6/09

57

## Relational Resolution I

$$\{\varphi_1, \dots, \varphi, \dots, \varphi_m\}$$

$$\{\psi_1, \dots, \neg \psi, \dots, \psi_n\}$$

$$\hline \{\varphi_1, \dots, \varphi_m, \psi_1, \dots, \psi_n\} \sigma$$

$$\text{where } \sigma = mgu(\varphi, \psi)$$

12/6/09

58

## Relational Resolution II

$$\frac{\{\varphi_1, \dots, \varphi_m\} \quad \{\psi_1, \dots, \neg\psi, \dots, \psi_n\}}{\{\varphi_1\tau, \dots, \varphi_m\tau, \psi_1, \dots, \psi_n\}\sigma}$$

where  $\sigma = mgu(\varphi\tau, \psi)$   
where  $\tau$  is a variable renaming on  $\varphi$

12/6/09

59

## Relational Resolution III (Final Version)

$$\frac{\Phi \quad \Psi}{((\Phi' - \{\phi\})\tau \cup (\Psi' - \{\neg\psi\}))\sigma}$$

where  $\phi \in \Phi'$ , a factor of  $\Phi$   
where  $\neg\psi \in \Psi'$ , a factor of  $\Psi$   
where  $\sigma = mgu(\varphi\tau, \psi)$   
where  $\tau$  is a variable renaming on  $\varphi$

12/6/09

60

## Provability

A *resolution derivation* of a clause  $\varphi$  from a set  $\Delta$  of clauses is a sequence of clauses terminating in  $\varphi$  in which each item is

- (1) a member of  $\Delta$  or
- (2) the result of applying the resolution to earlier items.

A sentence  $\varphi$  is *provable* from a set of sentences  $\Delta$  by resolution if and only if there is a derivation of the empty clause from the clausal form of  $\Delta \cup \{\neg\varphi\}$ .

A resolution *proof* is a derivation of the empty clause from the clausal form of the premises and the negation of the desired conclusion.

12/6/09

61

## Soundness and Completeness

*Metatheorem:* Provability using the Relational Resolution Principle is sound and complete for Relational Logic (without equality).

12/6/09

62

## Answer Extraction Method

Alternate Method for Logical Entailment: To determine whether a set  $\Delta$  of sentences logically entails a closed sentence  $\varphi$ , rewrite  $\Delta \cup \{\varphi \Rightarrow goal\}$  in clausal form and try to derive *goal*.

Method for Answer Extraction: To get values for free variables  $v_1, \dots, v_n$  in  $\varphi$  for which  $\Delta$  logically entails  $\varphi$ , rewrite  $\Delta \cup \{\varphi \Rightarrow goal(v_1, \dots, v_n)\}$  in clausal form and try to derive  $goal(v_1, \dots, v_n)$ .

Intuition: The sentence  $(q(z) \Rightarrow goal(z))$  says that, whenever,  $z$  satisfies  $q$ , it satisfies the “goal”.

12/6/09

63

## Strategies

Elimination Strategies (Constraints on clauses):

- Identical Clause Elimination
- Pure Literal Elimination
- Tautology Elimination
- Subsumption Elimination

Restriction Strategies (Constraints on inferences):

- Unit Restriction
- Input Restriction
- Linear Restriction
- Set of Support Restriction

12/6/09

64

## Clauses and Chains

A *clause* is a set of literals.

$$\{p, \neg q, \neg r\}$$

A *chain* is a sequence of literals

$$\langle p, \neg q, \neg r \rangle$$

12/6/09

65

## Ordered Resolution

$$\frac{\langle \varphi, \varphi_1, \dots, \varphi_m \rangle \quad \langle \neg \psi, \psi_1, \dots, \psi_n \rangle}{\langle \varphi_1, \dots, \varphi_m, \psi_1, \dots, \psi_n \rangle \sigma}$$

where  $\sigma = mgu(\varphi, \psi)$

12/6/09

66

## Semi-Ordered Resolution

$$\frac{\langle \varphi_1, \dots, \varphi_i, \dots, \varphi_m \rangle \quad \langle \psi, \psi_1, \dots, \psi_n \rangle}{\langle \varphi_1, \dots, \varphi_m, \psi_1, \dots, \psi_n \rangle \sigma}$$

where  $\sigma = mgu(\neg\varphi_i, \psi)$

12/6/09

67

## Contrapositives

A *contrapositive* of a chain is a permutation in which a different literal is placed at the front.

Chain:  $\langle p, \neg q, \neg r \rangle$

Contrapositive:  $\langle \neg q, p, \neg r \rangle$

Contrapositive:  $\langle \neg r, p, \neg q \rangle$

The contrapositives of a chain are logically equivalent to the original chain.

12/6/09

68

## Model Elimination

*Model Elimination* is a variant of Ordered Resolution that incorporates the Linearity Restriction in the definition of the rules of inference.

Using Model Elimination alone, it is possible to build a theorem prover that is sound and complete for all of Relational Logic.

Moreover, it works with the Set of Support strategy and the Input Restriction!!!

12/6/09

69

## Normal and Reduced Literals

Normal Literals:

$$\begin{array}{c} p \\ \neg q \end{array}$$

Reduced Literals:

$$\begin{array}{c} [p] \\ [\neg q] \end{array}$$

Chains:

$$\langle p, \neg q, [p], r \rangle$$

12/6/09

70

## Model Elimination Rules

Reduction

$$\frac{\langle \varphi, \varphi_1, \dots, \varphi_m \rangle \quad \langle \psi, \psi_1, \dots, \psi_n \rangle}{\langle \varphi_1, \dots, \varphi_m, [\psi], \psi_1, \dots, \psi_n \rangle \sigma}$$

Cancellation

where  $\sigma = mgu(\neg\varphi, \psi)$

$$\frac{\langle \varphi, \varphi_1, \dots, \varphi_m, [\psi], \psi_1, \dots, \psi_n \rangle}{\langle \varphi_1, \dots, \varphi_m, [\psi], \psi_1, \dots, \psi_n \rangle \sigma}$$

where  $\sigma = mgu(\neg\varphi, \psi)$

Dropping

$$\frac{\langle [\varphi], \varphi_1, \dots, \varphi_m \rangle}{\langle \varphi_1, \dots, \varphi_m \rangle}$$

12/6/09

71

## Epilog

Epilog is a theorem prover for Relational Logic. It is sound and complete. It is at least as efficient as Model Elimination, and it is arguably more efficient. It is somewhat more intuitive than ordinary Resolution.

Features:

- Rule Form instead of Clausal Form
- Backward Chaining variant of the ME rule
- Iterative Deepening rather than Breadth-First Search

12/6/09

72

## Rule Form

Premises are expressed as *rules*.

$$\begin{array}{ll} \langle p \rangle & p \Leftarrow \\ \langle \neg p \rangle & \neg p \Leftarrow \\ \langle r, \neg p, \neg q \rangle & r \Leftarrow p \wedge q \end{array}$$

Conclusions are expressed as *questions*.

$$\begin{array}{ll} \langle p \rangle & \neg p? \\ \langle \neg p \rangle & p? \\ \langle \neg p, \neg q, r \rangle & p \wedge q \wedge \neg r? \end{array}$$

12/6/09

73

## Backward Chaining

Backward Chaining is the same as reduction except that it works on rule form rather than clausal form.

$$\frac{\varphi \Leftarrow \varphi_1 \wedge \dots \wedge \varphi_m \quad \psi \wedge \psi_1 \wedge \dots \wedge \psi_n?}{\varphi_1 \wedge \dots \wedge \varphi_m \wedge [\psi] \wedge \psi_1 \wedge \dots \wedge \psi_n? \sigma}$$

where  $\sigma = mgu(\varphi, \psi)$

Reduced literals need be retained only for non-Horn premises.

Cancellation and Dropping are analogous.

12/6/09

74

## Equality

An equation  $\sigma = \tau$  is true in an interpretation  $i$  if and only if the terms in the equation refer to the same object in the universe of discourse.

$$\sigma^i = \tau^i$$

## Example

Interpretation:

$$i(a) = \circ$$

$$i(b) = \bullet$$

$$i(c) = \circ$$

$$i(f) = \{\circ \rightarrow \bullet, \bullet \rightarrow \circ\}$$

$$i(r) = \{\langle \circ, \bullet \rangle, \langle \bullet, \bullet \rangle\}$$

Satisfied Sentences

$$a = a$$

$$a \neq b$$

$$a = c$$

$$b = b$$

$$12/6/09 \quad b \neq c$$

$$a = f(b)$$

$$b = f(a)$$

$$b = f(c)$$

$$c = f(b)$$

$$c = f(f(c))$$

$$a = f(f(a))$$

$$a = f(f(c))$$

$$b = f(f(b))$$

$$c = f(f(a))$$

$$c = f(f(c))$$

77

## Unique Names Assumption

In many applications, one makes the assumption that every object has a unique name. This is called the *unique names assumption* (UNA).

The upshot is that a difference in name implies a difference in referent.

$$\sigma = \tau \Leftrightarrow \sigma^i = \tau^i$$

The unique names assumption is *not* true in general!!!

Question: How does one express the unique names assumption in Relational Logic?

12/6/09

78

## Equational Reasoning

Axioms (reflexivity, symmetry, transitivity, subst)  
Paramodulation and Demodulation

- x RUE (Disagreement Sets)
- x AC-unification
- x Rewrite Rules

12/6/09

79

## Equality Axioms

Reflexivity

$$\forall x. x=x$$

Symmetry:

$$\forall x. \forall y. (x=y \Rightarrow y=x)$$

Transitivity:

$$\forall x. \forall y. \forall z. (x=y \wedge y=z \Rightarrow x=z)$$

12/6/09

80

## Equality Axioms in Rule Form

Reflexivity

$$x=x$$

Symmetry:

$$x=y \Leftrightarrow y=x$$

Transitivity:

$$x=z \Leftrightarrow x=y \wedge y=z$$

12/6/09

81

## Equality Proof

- |     |  |          |
|-----|--|----------|
| 1.  | $b = a$                                    | Premise  |
| 2.  | $b = c$                                    | Premise  |
| 3.  | $x = x$                                    | Equality |
| 4.  | $x = y \Leftrightarrow y = x$              | Equality |
| 5.  | $x = z \Leftrightarrow x = y \wedge y = z$ | Equality |
| 6.  | $a = c?$                                   | Goal     |
| 7.  | $a = y \wedge y = c?$                      | 5,6      |
| 8.  | $y = a \wedge y = c?$                      | 4,7      |
| 9.  | $b = c?$                                   | 1,8      |
| 10. | ?  | 2,9      |

12/6/09

82

## Equality Problem

- |     |   |          |
|-----|---|----------|
| 1.  | $f(a) = b$  | Premise  |
| 2.  | $f(b) = a$  | Premise  |
| 3.  | $x = x$   | Equality |
| 4.  | $x = y \Leftrightarrow y = x$                         | Equality |
| 5.  | $x = z \Leftrightarrow x = y \wedge y = z$            | Equality |
| 6.  | $f(f(a)) = a?$  | Goal     |
| 7.  | $a = f(f(a))?$  | 4,6      |
| 8.  | $f(f(a)) = y \wedge y = a?$                           | 5,6      |
| 9.  | $f(f(a)) = w \wedge w = y \wedge y = a?$              | 5,8      |
| 10. | $f(f(a)) = v \wedge v = w \wedge w = y \wedge y = a?$ | 5,9      |

12/6/09

83

## Flattening

Equivalence:

$$f(f(a))=a \Leftrightarrow \exists x.(f(a)=x \wedge f(x)=a)$$

Rewrite:  $f(f(a))=a$

As:  $\exists x.(f(a)=x \wedge f(x)=a)$

As:  $f(a)=c \wedge f(c)=a$

As:  $f(a)=c$

$f(c)=a$

Rewrite:  $f(f(a))=a?$

As:  $\exists x.(f(a)=x \wedge f(x)=a)?$

As:  $f(a)=x \wedge f(x)=a?$

12/6/09

84

## Proof With Flattening

- |    |                                       |                |
|----|---------------------------------------|----------------|
| 1. | $f(a) = b$                            | Premise        |
| 2. | $f(b) = a$                            | Premise        |
| 3. | $x = x$                               | Equality       |
| 4. | $x = y \Leftarrow y = x$              | Equality       |
| 5. | $x = z \Leftarrow x = y \wedge y = z$ | Equality       |
| 6. | $f(a) = x \wedge f(x) = a?$           | $f(f(a)) = a?$ |
| 7. | $f(b) = a?$                           | 1,6            |
| 8. | ?                                     | 2,7            |

12/6/09

85

## Substitution Axiom

Flattening Rule:

$$f(f(a))=a \Leftrightarrow \exists x.(f(a)=x \wedge f(x)=a)$$

Substitution Axiom:

$$f(x)=z \Leftarrow x=y \wedge f(y)=z$$

12/6/09

86

## Proof With Substitution

- |     |   |              |
|-----|---|--------------|
| 1.  | $f(a) = b$  | Premise      |
| 2.  | $f(b) = a$  | Premise      |
| 3.  | $x = x$   | Equality     |
| 4.  | $x = y \Leftrightarrow y = x$                     | Equality     |
| 5.  | $x = z \Leftrightarrow x = y \wedge y = z$        | Equality     |
| 6.  | $f(x) = z \Leftrightarrow x = y \wedge f(y) = z?$ | Substitution |
| 7.  | $f(f(a)) = a?$                                    | Goal         |
| 8.  | $f(a) = y \wedge f(y) = a?$                       | 6,7          |
| 9.  | $f(b) = a?$                                       | 1,8          |
| 10. | ?   | 2,9          |

12/6/09

87

## Notes

Substitution axioms for relation constants too.

$$p(x) \Leftrightarrow x=y \wedge p(y)$$

Substitution axioms for multiple arguments

$$p(x,y)=z \Leftrightarrow x=u \wedge y=v \wedge p(u,v)$$

Need one substitution for each function and relation constant.

12/6/09

88

## Demodulation

$$\frac{\{\varphi_1, \dots, \varphi_n\}}{\{\tau_1 = \tau_2\}} \\ \frac{\{\varphi_1, \dots, \varphi_n\}[\tau_1\sigma \leftarrow \tau_2\sigma]}{\text{where } \tau \text{ occurs in } \varphi_i} \\ \text{where } \tau_1\sigma = \tau$$

12/6/09

89

## Examples

$$\frac{p(a, f(b, g(a, h(b))), c), d)}{b = e} \\ \frac{p(a, f(e, g(a, h(e))), c), d)}{}$$

$$\frac{p(a, f(b, g(a, h(b))), c), d)}{g(x, y) = j(x)} \\ \frac{p(a, f(b, j(a), c), d)}{}$$

12/6/09

90

## Non-Examples

Unit Equations Only

$$\frac{p(a, g(a, b), c) \quad g(a, y) = y \Leftarrow p(y)}{p(a, g(a, b), c) \Leftarrow p(b)}$$

Variables Substituted in Equation Only

$$\frac{p(a, g(x, b), c) \quad g(a, y) = j(y)}{p(a, j(b), c)}$$

12/6/09

91

## Problems With Demodulation

Cannot bind variables in expression:

$$\frac{father(pat) = quincy \quad older(father(x), x)}{older(quincy, pat)}$$

Equation must be a unit clause

$$\frac{father(x) = y \Leftarrow x = pat \wedge y = quincy \quad older(father(x), x)}{older(quincy, pat) \Leftarrow x = pat \wedge y = quincy}$$

12/6/09

92

## Paramodulation

$$\{\varphi_1, \dots, \varphi_n\}$$
$$\{\psi_1, \dots, \tau_1 = \tau_2, \dots, \psi_n\}$$
$$\frac{\{\varphi_1, \dots, \varphi_n, \psi_1, \dots, \psi_n\}}{\{\varphi_1, \dots, \varphi_n, \psi_1, \dots, \psi_n\} \sigma[\tau_1 \sigma \leftarrow \tau_2 \sigma]}$$

where  $\tau$  occurs in  $\varphi_i$

where  $\tau_1 \sigma = \tau_2 \sigma$

12/6/09

93

## Example

$$\{p(f(x,b),x),q(x)\}$$
$$\{f(a,y) = y,r(y)\}$$
$$\frac{\{p(f(x,b),x),q(x)\}}{\{p(b,a),q(a),r(b)\}}$$

12/6/09

94

## Proof With Paramodulation

1.  $\{father(pat) = quincy\}$  Premise
2.  $\{older(father(x),x)\}$  Premise
3.  $\{\neg older(quincy,pat)\}$  Goal
4.  $\{older(quincy,pat)\}$  1,2
5.  $\{\}$  3,4

12/6/09

95

## Proof With Paramodulation

1.  $\{p(a)\}$  Premise
2.  $\{p(b)\}$  Premise
3.  $\{a = c, b = c\}$   $a = c \vee b = c$
4.  $\{\neg p(c)\}$  Goal
5.  $\{\neg p(a), b = c\}$  3,4
6.  $\{\neg p(a), \neg p(b)\}$  4,5
7.  $\{\neg p(b)\}$  1,6
8.  $\{\}$  2,7

12/6/09

96

## Differences

- (1) Demodulation requires a unit equation.
- (2) Demodulation binds variables in equation only.
- (3) Demodulation deletes parent.

12/6/09

97

## Power

Theorem: Resolution and Paramodulation (together with the reflexivity axiom) are refutation complete for all of Relational Logic\*.

\*including equality

12/6/09

98

## Details

Wednesday 7:00pm in Skilling Auditorium.

One hour and 15 minutes of three hours.

Closed book.

Easy.

12/6/09

99

## Hints

1. Study the Problems on the Problem Sets.
2. Study the Problems on the Problem Sets.
3. Study the Problems on the Problem Sets.
4. Study the Problems on the Problem Sets.
5. Study the Problems on the Problem Sets.
6. Study the Problems on the Problem Sets.
7. Study the Problems on the Problem Sets.
8. Study the Problems on the Problem Sets.
9. Study the Problems on the Problem Sets.
10. View and understand the guest lectures.

12/6/09

100