# 8. Proofs in Predicate Calculus



Language & Logic

**Dave Parker** 

University of Birmingham 2017/18

# **Today**

- Recap: predicate calculus
  - syntax, translation, examples
- Next main topic: Proofs in predicate calculus
  - syntax vs. semantics
  - validity for arguments in predicate calculus
  - syntactic vs. semantic validity
  - soundness & completeness
  - natural deduction for predicate calculus: first steps
- See also Chap 6 (and some of Chap 5) of Tomassi book

# Recap: Predicate calculus

#### Key ingredients

- variables x,y,z
- constant symbols a,b,c
- quantifiers: universal: ∀ ("for all"); existential: ∃ ("there exists")
- predicates (unary, binary, n-ary, nullary)
- e.g. P(x), P(a), L(m,x), ...

#### Extension of propositional logic

- retains all usual connectives:  $\neg$ ,  $\lor$ ,  $\land$ ,  $\rightarrow$ 

### Translation from natural language

- first establish what the domain is
- then identify constants and predicates
- templates for translating several common patterns

# Ex 5 examples

- Examples Ex 5 Q1
- (a) All white animals are mice  $\forall x[W(x) \to M(x)]$
- Common "template"

(b) Basil is a white mouse  $M(b) \wedge W(b)$ 

- No need for any quantifiers
- Re-use the same predicates,
  M(x) and W(x), here

Examples Ex 5 Q2

- Can't write (M(W(b))!
- (k) Everyone who loves Chris loves someone who loves John  $\forall x[L(x,c) \to \exists y[(L(x,y) \land L(y,j))]]$



- Same template as Q1(a), but more complex inner formula
- x and y must be distinct variables<sub>4</sub>

## Semantics: Interpretations

- Syntax (allowable sentences) vs. semantics (meaning)
- An interpretation is an assignment of meaning to the symbols of a formal language
  - usually provides a way to determine the truth value of a sentence
- Recall... In propositional logic:
  - sentence = proposition, e.g. P ∨ Q
  - interpretation = truth assignment (i.e., an assignment of true/false to each atomic proposition)

Р	Q	P V Q
Т	Т	Т
Т	F	Т
F	Т	Т
F	F	F

## Predicate calculus semantics

- How do we define the semantics of the predicate calculus?
  - i.e, what is the truth value of a sentence (formula)?
  - and what is the validity of an argument?
- We need 2 things:
- 1. a domain (also called "domain of discourse" or "universe")
  - a <u>non-empty</u> set of objects/entities
  - e.g. people, cars, program executions, the natural numbers, ...
  - consider e.g. "all computer scientists are logical"
    - translated as ∀x [ L(x) ] or ∀x [ C(x) → L(x) ] ?
- 2. an interpretation
  - defines the meaning of predicates in terms of the domain
  - is Alan logical? is he a computer scientist? etc.

## Semantics of a formula

- Given a domain and an interpretation:
  - we can assign a truth value to each sentence/formula
- Example sentence (assuming the domain = people)
  - $\forall x [C(x) \rightarrow L(x)]$
- An example interpretation:
  - constants:
    - a = Alan; b = Bella
  - predicates:
    - C(a) and C(b) are true
    - L(a) is false but L(b) is true
  - for this interpretation:
    - is ∀x [ C(x) → L(x) ] true?
    - is  $\exists x [C(x) \land L(x)]$  true?

# Validity of an argument

- As for propositional logic, we mostly care about arguments and their validity, rather than particular interpretations
  - e.g.  $\forall x [C(x) \rightarrow L(x)], C(a) : L(a)$  is this valid?
- An argument in predicate logic is valid if and only if
  - for every possible domain and every possible interpretation,
    whenever the premises are all true, the conclusion is true
- An argument in predicate logic is invalid if and only if
  - for some domain, there is a possible interpretation under which all the premises are true and the conclusion is false

# Example arguments

(Taken from the quiz in week 1)

All men are mortal.

Some men are brave.

Therefore, some men are mortal and brave.

All men are brave.

No man is a philosopher.

Therefore, no philosopher is brave.

# Semantic and syntactic validity

- For propositional logic, we saw two methods to show validity of an argument
  - truth table construction & analysis (semantic validity)
  - natural deduction proofs (syntactic validity)
- Semantic validity
  - an argument is valid iff, whenever all the premises are true,
    then the conclusion is also true
- Syntactic validity
  - an argument is valid iff the conclusion can be derived from the premises by means of stipulated rules of inference
- Notation
  - semantic validity:  $A \rightarrow B$ ,  $\neg B \models \neg A$
  - syntactic validity:  $A \rightarrow B$ ,  $\neg B \vdash \neg A$

# Soundness & completeness

#### Soundness

- means any syntactically valid argument is semantically valid
- i.e.  $A_1, ..., A_n \vdash B$  implies  $A_1, ..., A_n \models B$
- means that our proofs are correct

#### Completeness

- $-A_1, ..., A_n \models B \text{ implies } A_1, ..., A_n \vdash B$
- means we can construct a proof of any valid argument

#### Natural deduction

- for propositional logic (and predicate calculus)
- is sound and complete
- which means we can prove any valid argument

## Natural deduction

- How do we prove validity in the predicate calculus?
  - no way to enumerate all possible domains/interpretations
  - so use natural deduction (which, again, is sound and complete)
- Natural deduction for predicate calculus
  - extension of the case for propositional logic
  - i.e., we can use all the existing inference rules
  - (and all exist theorems hold)
- We add 4 new inference rules
  - − ∀-elimination (universal-elimination)
  - ∃-introduction (existential-introduction)
  - − ∀-introduction (universal-introduction)
  - ∃-elimination (existential-elimination)

## ∀-elimination

- Basic idea:
  - given a universal quantification, we can infer a particular instance
- Example

All computer scientists are logical.

Alan is a computer scientist.

Therefore, Alan is logical.

• Argument  $\forall x [C(x) \rightarrow L(x)], C(a) : L(a)$ 

• Proof:

- 1.  $\forall x [C(x) \rightarrow L(x)]$  Premise {1}
- 2.  $C(a) \rightarrow L(a)$   $\forall$ -elimination<sub>1</sub> {1}
- 3. C(a) Premise {2}
- 4.  $L(a) \rightarrow -elimination_{2,3} \{1,2\}$

## ∀-elimination

• The inference rule:

$$\frac{\forall x [ \varphi(x) ]}{\varphi(a)} \quad \forall -elimination$$

- Notation (here and in the later rules):
  - $-\phi(x)$  is a formula in which variable x appears
  - (x is called a free variable)
  - $\phi(a)$  is  $\phi(x)$  with each instance of x replaced by constant a
- Intuition for ∀-elimination:
  - universal quantification is like a conjunction
  - $\forall x [P(x)] \equiv P(a) \land P(b) \land P(c) \land ...$
  - so ∀-elimination is like ∧-elimination

## ∃-introduction

#### Basic idea

- from a particular instance, we can infer an existential quantification

#### The inference rule:

$$\frac{\varphi(a)}{\exists x [\varphi(x)]} \exists -introduction$$

- where  $\phi(x)$  and  $\phi(a)$  are as defined for the previous rule

#### Intuition

- existential quantification is like a disjunction
- $-\exists x [P(x)] \equiv P(a) \vee P(b) \vee P(c) \vee ...$
- so ∃-introduction is similar to ∨-introduction

# Example - 3-introduction

• Is this argument valid?

```
- \forall x [P(x)] : \exists x [P(x)]
```

- Yes (remember that domains are always non-empty)
- Here is the proof (using our 2 new inference rules):

```
 ∀x [ P(x) ] Premise {1}
 P(a) ∀-elimination<sub>1</sub> {1}
 ∃x [ P(x) ] ∃-introduction<sub>2</sub> {1}
```

- Question: Is this argument valid?
  - $\forall x [P(x) \rightarrow Q(x)] : \exists x [Q(x)]$

## ∀-introduction

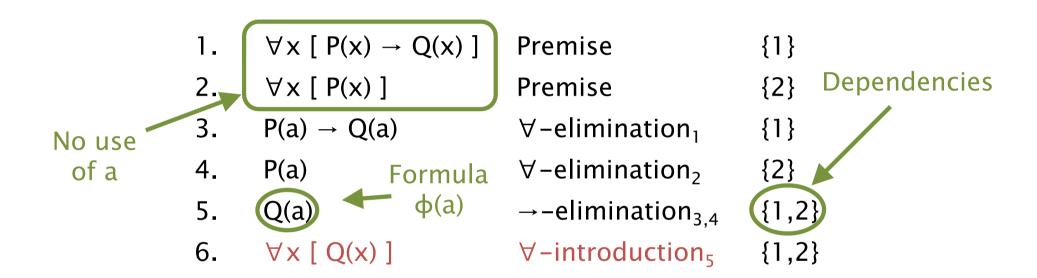
• The inference rule:

$$\frac{\varphi(a)}{\forall x \ [\ \varphi(x)\ ]} \ \forall -introduction$$

- But there are conditions...
- Under what circumstances can we infer this?
  - a needs to be a "typical" example of x
  - i.e., we must know nothing else about a
- To be more precise...
  - $\forall$ -introduction can <u>only</u> be applied to a formula  $\varphi(a)$  if a does <u>not</u> appear in any of the dependencies for  $\varphi(a)$ ,

# Example - ∀-introduction

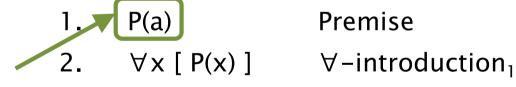
• Prove:  $\forall x [P(x) \rightarrow Q(x)], \forall x [P(x)] : \forall x [Q(x)]$ 

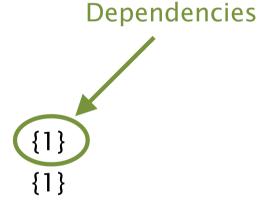


- Note: this is a common pattern in predicate calculus proofs
  - ∀-elimination (for new symbol a) then ∀-introduction

# ∀-introduction: Why the condition?

- Why do we need the condition on the a?
- An example: Is this argument valid?
  - $P(a), \neg P(b) : \forall x [P(x)]$
- No. Here is an (incorrect) proof





Contains

a

- The condition for ∀-introduction is not satisfied
  - so the inference rule cannot be applied here

## ∃-elimination

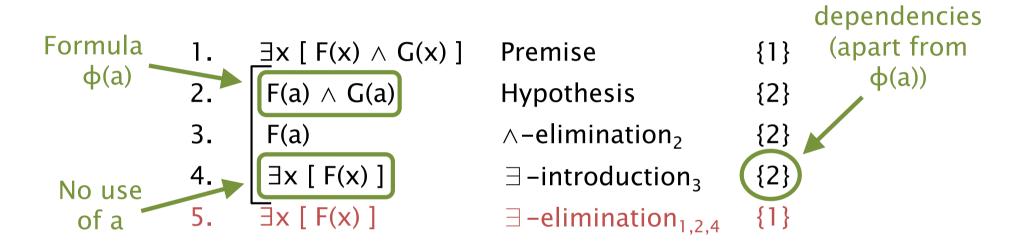
- Intuition for ∃-elimination:
  - recall: existential quantification is like a disjunction
  - $-\exists x [P(x)] \equiv P(a) \vee P(b) \vee P(c) \vee ...$
  - so ∃-elimination is like ∨-elimination

$$\frac{\exists x \ [\ \varphi(x)\ ] \quad \varphi(a) \vdash C}{C} \quad \exists -elimination$$

- Like for ∀-introduction
  - a needs to be a "typical" example of x
- So:
  - the constant a in  $\phi(a)$  must <u>not</u> appear in C, nor in any of the dependencies of C (except the hypothesis  $\phi(a)$  itself)

# Example – 3-elimination

Prove: ∃x [ F(x) ∧ G(x) ] : ∃x [ F(x) ]



- Note: this is also a common pattern in predicate calculus proofs
  - ∃-introduction inside ∃-elimination

No other

# Examples – 3-elimination

• Prove:  $\exists x [F(x) \land G(x)] : \exists x [G(x) \land F(x)]$ 

• Prove:  $\forall x [G(x) \rightarrow G(b)], \exists x [G(x)] : G(b)$ 

# Summary

- Natural deduction for predicate calculus
  - extends natural deduction for propositional logic
- ∀-elimination
  - infer a particular instance  $\phi(a)$  from  $\forall x [\phi(x)]$
- ∃-introduction
  - infer  $\exists x [ \phi(x) ]$  from a particular instance  $\phi(a)$
- ∀-introduction
  - infer  $\forall x [ \phi(x) ]$  from  $\phi(a)$  for a "typical" a
- ∃-elimination
  - infer C from  $\exists x [ \phi(x) ]$  by inferring C for a "typical" a