7. Predicate Calculus

Language & Logic

Dave Parker

University of Birmingham

2017/18

Announcements

- Exercise sheet 4
  - see Canvas for model solutions and feedback (announcement)
  - ask in office hours or class (we'll return to proofs next week)

- Assignment 2
  - on natural deduction for propositional logic; due 5pm Friday

- Module feedback (mid-term questionnaires)
  - lectures & exercise classes: working well
  - content & pace: most people happy
- Suggestions
  - pointers to more examples (i.e. unassessed exercises)

Module syllabus

- Syntax of formal & natural languages
  - grammars, parsing
- Propositional logic
  - truth tables, semantics, proofs via natural deduction
- Predicate calculus
  - proofs via natural deduction
- Program correctness
  - structural induction

Limits of propositional logic

- Here's an example argument
  - how do we check its validity in propositional logic?

> All computer scientists are logical.
>
> Alan is a computer scientist.
>
> Therefore, Alan is logical.

- This lecture: predicate calculus
  - extension of propositional logic
  - also known as first-order logic
  - and sometimes as predicate logic
  - allows us to use variables, quantifiers, predicates, ...

Predicate calculus: Key ingredients

- Variables, e.g., x, y, z,… (to reason about objects/entities)
- Quantifiers
  - $\forall$x […] = "for all x …" (universal)
  - $\exists$x […] = "there exists x such that ..." (existential)
  - [] brackets are used to indicate scope of quantifier
- Constants (specific instances of objects/entities)
  - e.g. a, b, c,…
- Predicates
  - recall: a proposition is a statement that may be true or false
  - a predicate is a statement that may be true or false depending on the values of its arguments
  - P(x) (or P(a)) = "predicate P is true for variable x (or constant a)"
  - alternative notation: Px or Pa

Example

All computer scientists are logical.

Alan is a computer scientist.

Therefore, Alan is logical.

- Predicate symbols (properties of objects)
  - $C(x) = x$ is a computer scientist
  - $L(x) = x$ is logical
- Constant symbols (specific instances of objects)
  - $a =$ Alan
- Argument
  - premise 1: $\forall x\, [\, C(x) \rightarrow L(x)\,]$
  - premise 2: $C(a)$
  - conclusion: $L(a)$
- In sequent form:

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

Predicate calculus – syntax

- Recall the syntax of propositional logic:
  - F → Ap | [¬] F | F [∧] F | F [∨] F | F [→] F
  - Ap → [P] | [Q] | [R] | …

Predicate calculus – syntax

- Extend to the syntax for predicate calculus
  - F → Pr [(] T [)] | [¬] F | F [∧] F | F [∨] F | F [→] F | Q V [[] F []]
  - Pr → [P] | [Q] | [R] | ... (predicate symbol)
  - T → V | C (terms)
  - V → [x] | [y] | [z] | ... (variables)
  - C → [a] | [b] | [c] | ... (constant symbols)
  - Q → [∀] | [∃] (quantifiers)
- (actually a simplified version of predicate calculus)

Predicate calculus – syntax

- Extend to the syntax for predicate calculus
  - F → $\boxed{\text{Pr [(] T [)]}}$ | [¬] F | F [∧] F | F [∨] F | F [→] F | Q V [[] F []]
  - $\boxed{\text{Pr → [P] | [Q] | [R] | ...}}$ (predicate symbol)
  - $\boxed{\text{T → V | C}}$ (terms)
  - $\boxed{\text{V → [x] | [y] | [z] | ...}}$ (variables)
  - $\boxed{\text{C → [a] | [b] | [c] | ...}}$ (constant symbols)
  - Q → [∀] | [∃] (quantifiers)

- (actually a simplified version of predicate calculus)

Predicate calculus – syntax

- Extend to the syntax for predicate calculus
  - F → Pr [(] T [)] | [¬] F | F [∧] F | F [∨] F | F [→] F | Q V [(] F [)]
  - Pr → [P] | [Q] | [R] | ... (predicate symbol)
  - T → V | C (terms)
  - V → [x] | [y] | [z] | ... (variables)
  - C → [a] | [b] | [c] | ... (constant symbols)
  - Q → [∀] | [∃] (quantifiers)

- (actually a simplified version of predicate calculus)

Example formulas

- More examples of predicate calculus formulas
  - $\forall x\ [\ P(x) \wedge Q(x)\ ]$ – for all x it is true that P(x) and Q(x)
  - $\forall x\ [\ P(x) \vee \neg Q(x)\ ]$ – for all x it is true that P(x) or not Q(x)
  - $\forall x\ [\ P(x)\ ] \wedge \neg \forall x\ [\ Q(x)\ ]$ – P(x) is true for all x but Q(x) is not

- Existence
  - $\forall x\ [\ G(x) \rightarrow S(x)\ ]$ – all ghosts are scary
  - does this mean there exists at least one ghost?
  - $\exists x\ [\ G(x)\ ]$ – ghosts exist

- Nested quantifiers
  - $\forall x\ [\ P(x) \wedge \exists y\ [\ Q(x) \rightarrow R(y)\ ]\ ]$
  - scope: x and y can only appear inside the corresponding […]

Translating from natural language

- All cars are fast (assume we are talking only about cars)
  - $\forall x\,[\,F(x)\,]$ "everything is A"
- All red cars are fast
  - $\forall x\,[\,R(x) \rightarrow F(x)\,]$ "all As are B"
- Some red cars are fast
  - $\exists x\,[\,R(x) \wedge F(x)\,]$ "some A is B"
  - not: $\exists x\,[\,R(x) \rightarrow F(x)\,]$
- There are no red cars
  - $\neg\exists x\,[\,R(x)\,]$ "there are no As"
  - or: $\forall x\,[\,\neg R(x)\,]$
- No fast cars are purple
  - $\neg\exists x\,[\,F(x) \wedge P(x)\,]$ "no As are B"
  - or: $\forall x\,[\,F(x) \rightarrow \neg P(x)\,]$

Negation

- We have the following equivalences in predicate calculus:
  - $\neg\exists x\ [\ \ldots\ ] \equiv \forall x\ [\ \neg(\ldots)\ ]$
  - $\neg\forall x\ [\ \ldots\ ] \equiv \exists x\ [\ \neg(\ldots)\ ]$
- So existential/universal quantification are dual
- (Think about these as an analog of De Morgan's rules)

- Similarly:
  - $\exists x\ [\ \ldots\ ] \equiv \neg\forall x\ [\ \neg(\ldots)\ ]$
  - $\forall x\ [\ \ldots\ ] \equiv \neg\exists x\ [\ \neg(\ldots)\ ]$
- i.e., we can always rewrite one quantifier in terms of the other using negation
- (Again, the same is true for conjunction/disjunction)

Relationships

- So far, we only considered unary predicates
  - i.e. the arity (number of arguments) is 1
  - e.g. $L(x)$ = x is logical
- We can use binary predicates (arity 2)
  - to represent relationships between objects/entities
  - e.g. $M(a,b)$ = Alice and Bob are married
- More generally: n-ary predicates (arity n)
  - e.g. $P(x_1,x_2,\ldots,x_n)$
- And for completeness: nullary predicates (arity 0)
  - what are these?

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
  - interpretation = assignment of true/false to each atomic prop.

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Interpretations and domains

- How do we define the semantics of the predicate calculus?

- We need:

1. a domain (also called "domain of discourse" or "universe")
   - a non-empty set of objects/entities
   - e.g. people, cars, program executions, the natural numbers, …
   - "all Computer Scientists are logical"
     - translated as $\forall x\ [\ L(x)\ ]$ or $\forall x\ [\ C(x) \rightarrow L(x)\ ]$?
2. an interpretation
   - defines the meaning of predicates in terms of the domain
   - is Alan logical? is he a computer scientist? etc.

- Given a domain and an interpretation:
  - we can assign a truth value to each sentence/formula

Validity

- 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.

Summary

- Predicate calculus
  - key ingredients: variables, quantifiers, predicates
  - syntax: extension of propositional logic
  - translation: quantifiers, negation, relationships
  - semantics: domains, interpretations, validity

- Exercise classes (Tue/Thr)
  - predicate calculus (or questions about natural deduction)

- Next week
  - natural deduction for predicate calculus