11. Proof by Induction

Language & Logic

Dave Parker

University of Birmingham

2017/18

Last two weeks

- Final topic: proofs of program correctness
  - (see also connections to Elements of Functional Computing)
- This week (week 10)
  - Mon 4pm: lecture
  - Tue 11am: lecture
  - nothing on Thursday
- Next week (week 11)
  - Mon: no lecture
  - Tue 11am and Thu 10am: exercise classes
- Continuous assessment
  - assignment 3 (due 5pm this Friday 1 Dec)

Assignment 3 questions

- Q1b: “them”; “either”

(b) Everyone who loves Bella also loves either Claire or Daisy, and none of them speak French, but at least one person who loves Daisy speaks German.

- Q1/2: sentences vs. arguments vs. theorems

- Prove or disprove equivalences in <u>both</u> directions

Today & tomorrow

- Predicate calculus proofs
  - common questions/errors
- Propositional logic, predicate calculus, proof
  - applications elsewhere in maths & computer science
- Proof by induction
  - mathematical induction
- Structural induction
  - correctness of programs on recursive data structures

Predicate calculus proofs

- Disproof via counterexample
  - how to find? how to present?
  - see e.g. Exercise 8 Q4
  - $\exists x[P(x)] \wedge \forall x[\forall y[(P(x) \wedge P(y)) \to (x = y)]] \ : \ \neg\forall x[P(x)]$
- $\forall$-introduction and $\exists$-elimination
  - take care with the usage restrictions
  - learn/remember the intuition behind them
  - think about the roles of the constants
  - sometimes need several, e.g. Exercise 8 Q5

$$\exists x[\forall y[P(y) \to (x = y)]] \ : \ \forall x[\forall y[(P(x) \wedge P(y)) \to (x = y)]]$$

Logic & proof (so far)

- Propositional logic & predicate calculus
  - Boolean connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$)
  - quantifiers ($\forall$, $\exists$), variables, predicates, identity
  - formulas, arguments, theorems, equivalences
  - translation from natural language
  - formal proof using natural deduction

- Applications of logic & proof
  - program correctness
    - verification, model checking, SAT solvers, theorem provers
  - circuit design, artificial intelligence, knowledge representation
  - basis for formal proofs in mathematics & computer science

Mathematical proofs

- Ingredients of predicate logic are essential for representing mathematical facts, theorems, proofs, e.g.:

- Predicates
  - even(x) = x is even; mult(a,b,x) = x is equal to a times b
- Definitions
  - even integers are multiples of 2,
    i.e. $even(x) \equiv \exists y\ [\ mult(2,y,x)\ ]$
- Theorems
  - if n^2 is even then n is even too,
    i.e. $\forall n\ [\ even(n^2) \rightarrow even(n)\ ]$
- Proof techniques
  - proof by contradiction (¬-introduction)
  - proof by cases (∨-elimination)
  - …

Proof by contradiction

- Theorem:
  - if n^2 is even then n is even too, i.e. $\forall n\ [\ \text{even}(n^2) \rightarrow \text{even}(n)\ ]$
- Proof:

1.	$\text{even}(n^2)$	Hypothesis
2.		
3.		
4.		
5.		
6.		
7.		
8.	$\text{even}(n)$	
9.	$\text{even}(n^2) \rightarrow \text{even}(n)$	$\rightarrow$-introduction$_{1,8}$
10.	$\forall n\ [\ \text{even}(n^2) \rightarrow \text{even}(n)\ ]$	$\forall$-introduction$_{9}$

Proof by contradiction

- Theorem:
  - if n^2 is even then n is even too, i.e. $\forall n\,[\,\mathrm{even}(n^2) \rightarrow \mathrm{even}(n)\,]$
- Proof:

1.	$\mathrm{even}(n^2)$	Hypothesis
2.	$\neg\mathrm{even}(n)$	Hypothesis
3.		
4.		
5.		
6.		
7.	$\neg\neg\mathrm{even}(n)$	$\neg$-introduction$_{2,6}$
8.	$\mathrm{even}(n)$	$\neg\neg$-elimination$_{7}$
9.	$\mathrm{even}(n^2) \rightarrow \mathrm{even}(n)$	$\rightarrow$-introduction$_{1,8}$
10.	$\forall n\,[\,\mathrm{even}(n^2) \rightarrow \mathrm{even}(n)\,]$	$\forall$-introduction$_{9}$

Proof by contradiction

- Theorem:
  - if n^2 is even then n is even too, i.e. $\forall n\ [\ even(n^2) \rightarrow even(n)\ ]$
- Proof:

1.	$even(n^2)$	Hypothesis
2.	$\neg even(n)$	Hypothesis
3.	$n = 2k+1$	(since n is odd)
4.	$n^2 = (2k+1)^2 = 4k^2+4k+1 = 2(2k^2+2k)+1$	(expansion)
5.	$\neg even(n^2)$	(from above)
6.	$\perp$	$\wedge$-introduction$_{1,5}$
7.	$\neg\neg even(n)$	$\neg$-introduction$_{2,6}$
8.	$even(n)$	$\neg\neg$-elimination$_{7}$
9.	$even(n^2) \rightarrow even(n)$	$\rightarrow$-introduction$_{1,8}$
10.	$\forall n\ [\ even(n^2) \rightarrow even(n)\ ]$	$\forall$-introduction$_{9}$

Proof by induction

- Mathematical induction
  - proof technique for statements of the form $\forall n \, [\, P(n) \,]$
  - where n is a natural number

- Two steps:
  - 1. Base case:
    - e.g. prove that $P(0)$ is true
  - 2. Inductive step:
    - assume that $P(k)$ is true, prove that $P(k+1)$ is true
    - $P(k)$ is called the inductive hypothesis

- Conclude
  - $P(n)$ is true for all n

Proof by induction

- Inference rule:

$$\frac{P(0) \quad \forall k\,[\,P(k) \rightarrow P(k+1)\,]}{\forall n\,[\,P(n)\,]} \quad \text{by induction}$$

- Why it works:

1.	$P(0)$	Premise
2.	$\forall k\,[\,P(k) \rightarrow P(k+1)\,]$	Premise
3.	$P(0) \rightarrow P(1)$	$\forall$–elimination
4.	$P(1)$	$\rightarrow$–elimination
5.	$P(1) \rightarrow P(2)$	$\forall$–elimination
6.	$P(2)$	$\rightarrow$–elimination
7.	…	…

Mathematical induction – Example

- Prove: $\forall n \, [\, \Sigma_{i=0\ldots n}\, i = n(n+1)/2 \,]$
  - i.e. $\forall n \, [\, P(n) \,]$ where $P(n)$: $\Sigma_{i=0\ldots n}\, i = n(n+1)/2$

- Base case $(n=0)$:
  - LHS: $\Sigma_{i=0\ldots n}\, i = 0$
  - RHS: $n(n+1)/2 = (0 \times 1)/2 = 0$
  - so $P(0)$ is true

- Inductive step
  - assume inductive hypothesis $P(k)$: $\Sigma_{i=0\ldots k}\, i = k(k+1)/2$
  - then prove $P(k+1)$, i.e. $\Sigma_{i=0\ldots k+1}\, i = (k+1)(k+2)/2$
  - $\Sigma_{i=0\ldots k+1}\, i = (0+1+\ldots+k+k+1) = (\Sigma_{i=0\ldots k}\, i) + (k+1)$

    $= k(k+1)/2 + (k+1)$ (using inductive hypothesis)

    $= (k(k+1) + 2(k+1))/2 = (k^2+3k+2)/2 = (k+1)(k+2)/2$
  - so $P(k) \rightarrow P(k+1)$ and therefore $\forall n \, [\, P(n) \,]$

Next up: Structural induction

- Inference rule:

$$\frac{P(0) \quad \forall k\ [\ P(k) \rightarrow P(k+1)\ ]}{\forall n\ [\ P(n)\ ]} \text{ by induction}$$

- We could rewrite this as:

$$\frac{P(0) \quad \forall k\ [\ P(k) \rightarrow P(\mathrm{succ}(k))\ ]}{\forall n\ [\ P(n)\ ]}$$

- And in fact "succ" could be any recursive definition…