CSE 311 Lecture 09: Proof Strategies

Emina Torlak and Kevin Zatloukal

Topics

Predicate logic proofs

A review and continuation of Lecture 08.

Natural language proofs

From formal proofs to natural language proofs.

Proof strategies

Proof by contrapositive, counterexamples, and proof by contradiction.

Predicate logic proofs

A review and continuation of Lecture 08.

Inference rules for quantifiers

Elim ∀ $$\frac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$$

Intro ∃ $$\frac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$$

Intro ∀ $$\frac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

Elim ∃ $$\frac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$$

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Predicate logic proofs can use …

Predicate logic inference rules

Applied to whole formulas only.

Predicate logic equivalences

Even on subformulas.

Propositional logic inference rules

Applied to whole formulas only.

Propositional logic equivalences

Even on subformulas.

Key takeaways from last lecture: domain properties

Proofs for a specific domain can use the properties of the domain.

- If a predicate is defined with a logical formula, use that formula in your proof.
- Otherwise, use domain properties to establish that a predicate is true.

Key takeaways from last lecture: domain properties

Proofs for a specific domain can use the properties of the domain.

- If a predicate is defined with a logical formula, use that formula in your proof.
- Otherwise, use domain properties to establish that a predicate is true.

1.	$2 = 2 \cdot 1$	**Arithmetic**
2.	$\exists y.\ 2 = 2 \cdot y$	**Intro ∃: 1**
3.	$\text{Even}(2)$	**Definition of Even: 2**
4.	$\text{Prime}(2)$	**Property of integer 2**
5.	$\text{Even}(2) \land \text{Prime}(2)$	**Intro ∧: 3, 4**
6.	$\exists x.\ \text{Even}(x) \land \text{Prime}(x)$	**Intro ∃: 5**

Domain of discourse

Integers

Predicate definitions

$\text{Even}(x) ::= \exists y.\ x = 2 \cdot y$

$\text{Prime}(x) ::=$ "x is prime"

Key takeaways from last lecture: domain properties

Proofs for a specific domain can use the properties of the domain.

- If a predicate is defined with a logical formula, use that formula in your proof.
- Otherwise, use domain properties to establish that a predicate is true.

1.	$2 = 2 \cdot 1$	**Arithmetic**
2.	$\exists y.\, 2 = 2 \cdot y$	**Intro ∃: 1**
3.	$\mathrm{Even}(2)$	**Definition of Even: 2**
4.	$\mathrm{Prime}(2)$	**Property of integer 2**
5.	$\mathrm{Even}(2) \wedge \mathrm{Prime}(2)$	**Intro ∧: 3, 4**
6.	$\exists x.\, \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$	**Intro ∃: 5**

We are using the logic definition of Even to establish that 2 is Even, and we are using domain property to establish that 2 is Prime.

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

$\mathrm{Prime}(x) ::=$ "x is prime"

Key takeaways from last lecture: Elim ∀ & Intro ∃

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore\ P(a) \text{ for any } a}$ **Intro ∃** $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\, P(x)}$

When applying Elim ∀ to $\forall x.\, P(x)$, you *have to replace all occurrences* of the universal variable x in $P(x)$ with the arbitrary name a.

But when applying Intro ∃ to $P(c)$, you *don't have to replace all occurrences* of c in $P(c)$ with the existential variable x.

Key takeaways from last lecture: Elim ∀ & Intro ∃

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore\ P(a) \text{ for any } a}$ **Intro ∃** $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\, P(x)}$

When applying Elim ∀ to $\forall x.\, P(x)$, you *have to replace all occurrences* of the universal variable x in $P(x)$ with the arbitrary name a.

But when applying Intro ∃ to $P(c)$, you *don't have to replace all occurrences* of c in $P(c)$ with the existential variable x.

1.	$\forall x.\, x = x$	**Given**
2.	$a = a$	**Elim ∀: 1, $\mathbf{a}$ is arbitrary**
3.	$\exists z.\, a = z$	**Intro ∃: 2**
4.	$\forall y.\, \exists z.\, y = z$	**Intro ∀: 3**

Domain of discourse
Integers

Key takeaways from last lecture: Elim ∀ & Intro ∃

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore\ P(a) \text{ for any } a}$ **Intro ∃** $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\, P(x)}$

When applying Elim ∀ to $\forall x.\, P(x)$, you *have to replace all occurrences* of the universal variable x in $P(x)$ with the arbitrary name a.

But when applying Intro ∃ to $P(c)$, you *don't have to replace all occurrences* of c in $P(c)$ with the existential variable x.

1. $\forall x.\, x = x$ **Given**
2. $a = a$ **Elim ∀: 1, $\mathbf{a}$ is arbitrary**
3. $\exists z.\, a = z$ **Intro ∃: 2**
4. $\forall y.\, \exists z.\, y = z$ **Intro ∀: 3**

Domain of discourse
Integers

Elim ∀ at 2 replaces all occurrences of x at 1, but Intro ∃ at 3 replaces only one occurrence of a at 2.

A square example

Prove that the square of every even number is even: $\forall x.\, \mathrm{Even}(x) \to \mathrm{Even}(x^2)$.

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\, P(x)}$

Intro ∀ $\dfrac{P(a);\, a \text{ is arbitrary}}{\therefore \forall x.\, P(x)}$

Elim ∃ $\dfrac{\exists x.\, P(x)}{\therefore P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\, \mathrm{Even}(x) \rightarrow \mathrm{Even}(x^2)$.

4. $\forall x.\, \mathrm{Even}(x) \rightarrow \mathrm{Even}(x^2)$

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\, P(x)}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore \forall x.\, P(x)}$

Elim ∃ $\dfrac{\exists x.\, P(x)}{\therefore P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\ \mathrm{Even}(x) \to \mathrm{Even}(x^2)$.

1. Let a be an arbitrary integer.

- Use Intro ∀ on 1 and 2.

3. $\mathrm{Even}(a) \to \mathrm{Even}(a^2)$
4. $\forall x.\ \mathrm{Even}(x) \to \mathrm{Even}(x^2)$ **Intro ∀: 1, 3**

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\ x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

Elim ∃ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\ \mathrm{Even}(x) \to \mathrm{Even}(x^2)$.

1. Let a be an arbitrary integer.
   - 2.1. $\mathrm{Even}(a)$ **Assumption**
   - 2.2.
   - 2.3.
   - 2.4.
   - 2.5.
   - 2.6. $\mathrm{Even}(a^2)$
3. $\mathrm{Even}(a) \to \mathrm{Even}(a^2)$ **Direct Proof Rule**
4. $\forall x.\ \mathrm{Even}(x) \to \mathrm{Even}(x^2)$ **Intro ∀: 1, 3**

- Use Intro ∀ on 1 and 2.
- → so use DRP to get 3.

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\ x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\ P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\ P(x)}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore \forall x.\ P(x)}$

Elim ∃ $\dfrac{\exists x.\ P(x)}{\therefore P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\, \text{Even}(x) \rightarrow \text{Even}(x^2)$.

1. Let a be an arbitrary integer.

   2.1. $\text{Even}(a)$ **Assumption**

   2.2. $\exists y.\, a = 2y$ **Definition of Even: 2.1**

   2.3.

   2.4.

   2.5. $\exists y.\, a^2 = 2y$

   2.6. $\text{Even}(a^2)$ **Definition of Even: 2.5**

3. $\text{Even}(a) \rightarrow \text{Even}(a^2)$ **Direct Proof Rule**
4. $\forall x.\, \text{Even}(x) \rightarrow \text{Even}(x^2)$ **Intro ∀: 1, 3**

- Use Intro ∀ on 1 and 2.
- → so use DRP to get 3.
- Use definition of Even to break down 2.1 and 2.6.

Domain of discourse

Integers

Predicate definitions

$\text{Even}(x) ::= \exists y.\, x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\, P(x)}$

Intro ∀ $\dfrac{P(a);\, a \text{ is arbitrary}}{\therefore \forall x.\, P(x)}$

Elim ∃ $\dfrac{\exists x.\, P(x)}{\therefore P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\, \mathrm{Even}(x) \to \mathrm{Even}(x^2)$.

1. Let a be an arbitrary integer.
   - 2.1. $\mathrm{Even}(a)$ — **Assumption**
   - 2.2. $\exists y.\, a = 2y$ — **Definition of Even: 2.1**
   - 2.3. $a = 2b$ — **Elim ∃: 2.2, b depends on a**
   - 2.4.
   - 2.5. $\exists y.\, a^2 = 2y$
   - 2.6. $\mathrm{Even}(a^2)$ — **Definition of Even: 2.5**
3. $\mathrm{Even}(a) \to \mathrm{Even}(a^2)$ — **Direct Proof Rule**
4. $\forall x.\, \mathrm{Even}(x) \to \mathrm{Even}(x^2)$ — **Intro ∀: 1, 3**

- Use Intro ∀ on 1 and 2.
- → so use DRP to get 3.
- Use definition of Even to break down 2.1 and 2.6.
- Use Elim ∃ on 2.2.

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\, P(x)}$

Intro ∀ $\dfrac{P(a);\, a \text{ is arbitrary}}{\therefore \forall x.\, P(x)}$

Elim ∃ $\dfrac{\exists x.\, P(x)}{\therefore P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\, \text{Even}(x) \to \text{Even}(x^2)$.

1. Let a be an arbitrary integer.

    2.1. $\text{Even}(a)$ — **Assumption**

    2.2. $\exists y.\, a = 2y$ — **Definition of Even: 2.1**

    2.3. $a = 2b$ — **Elim ∃: 2.2, b depends on a**

    2.4. $a^2 = 4b^2 = 2(2b^2)$ — **Algebra**

    2.5. $\exists y.\, a^2 = 2y$

    2.6. $\text{Even}(a^2)$ — **Definition of Even: 2.5**

3. $\text{Even}(a) \to \text{Even}(a^2)$ — **Direct Proof Rule**
4. $\forall x.\, \text{Even}(x) \to \text{Even}(x^2)$ — **Intro ∀: 1, 3**

- Use Intro ∀ on 1 and 2.
- → so use DRP to get 3.
- Use definition of Even to break down 2.1 and 2.6.
- Use Elim ∃ on 2.2.
- Use algebra on 2.3 to match the body of 2.5.

Domain of discourse

Integers

Predicate definitions

$\text{Even}(x) ::= \exists y.\, x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\, P(x)}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore \forall x.\, P(x)}$

Elim ∃ $\dfrac{\exists x.\, P(x)}{\therefore P(c) \text{ for a specific } c}$

A square example

Prove that the square of every even number is even: $\forall x.\, \mathrm{Even}(x) \rightarrow \mathrm{Even}(x^2)$.

1. Let a be an arbitrary integer.

2.1.	$\mathrm{Even}(a)$	**Assumption**
2.2.	$\exists y.\, a = 2y$	**Definition of Even: 2.1**
2.3.	$a = 2b$	**Elim ∃: 2.2, b depends on a**
2.4.	$a^2 = 4b^2 = 2(2b^2)$	**Algebra**
2.5.	$\exists y.\, a^2 = 2y$	**Intro ∃: 2.4**
2.6.	$\mathrm{Even}(a^2)$	**Definition of Even: 2.5**

3. $\mathrm{Even}(a) \rightarrow \mathrm{Even}(a^2)$ — **Direct Proof Rule**
4. $\forall x.\, \mathrm{Even}(x) \rightarrow \mathrm{Even}(x^2)$ — **Intro ∀: 1, 3**

- Use Intro ∀ on 1 and 2.
- → so use DRP to get 3.
- Use definition of Even to break down 2.1 and 2.6.
- Use Elim ∃ on 2.2.
- Use algebra on 2.3 to match the body of 2.5.
- Use Intro ∃ on 2.4 to get 2.5.

Domain of discourse

Integers

Predicate definitions

$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

Elim ∀ $\dfrac{\forall x.\, P(x)}{\therefore P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore \exists x.\, P(x)}$

Intro ∀ $\dfrac{P(a);\, a \text{ is arbitrary}}{\therefore \forall x.\, P(x)}$

Elim ∃ $\dfrac{\exists x.\, P(x)}{\therefore P(c) \text{ for a specific } c}$

Why list dependencies? To avoid incorrect proofs.

Over the integer domain: $\forall x.\ \exists y.\ y \geq x$ is True but $\exists y.\ \forall x.\ y \geq x$ is False.

Elim ∀ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Elim ∃ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Why list dependencies? To avoid incorrect proofs.

Over the integer domain: $\forall x.\ \exists y.\ y \geq x$ is True but $\exists y.\ \forall x.\ y \geq x$ is False.

1. $\forall x.\ \exists y.\ y \geq x$ Given
2.
3.
4.
5.
6. $\exists y.\ \forall x.\ y \geq x$

Example: an incorrect proof.

Elim ∀ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

Elim ∃ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Why list dependencies? To avoid incorrect proofs.

Over the integer domain: $\forall x.\ \exists y.\ y \geq x$ is True but $\exists y.\ \forall x.\ y \geq x$ is False.

1. $\forall x.\ \exists y.\ y \geq x$ **Given**
2. Let a be an arbitrary integer.
3.
4.
5.
6. $\exists y.\ \forall x.\ y \geq x$ **Intro ∃: 5**

Example: an **incorrect proof**.

Elim ∀ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro ∃ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Intro ∀ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

Elim ∃ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Why list dependencies? To avoid incorrect proofs.

Over the integer domain: $\forall x.\ \exists y.\ y \geq x$ is True but $\exists y.\ \forall x.\ y \geq x$ is False.

1. $\forall x.\ \exists y.\ y \geq x$ — Given
2. Let a be an arbitrary integer.
3. $\exists y.\ y \geq a$ — Elim $\forall$: 1
4.
5.
6. $\exists y.\ \forall x.\ y \geq x$ — Intro $\exists$: 5

Example: an incorrect proof.

Elim $\forall$ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro $\exists$ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Intro $\forall$ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

Elim $\exists$ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Why list dependencies? To avoid incorrect proofs.

Over the integer domain: $\forall x.\ \exists y.\ y \geq x$ is True but $\exists y.\ \forall x.\ y \geq x$ is False.

1. $\forall x.\ \exists y.\ y \geq x$ — **Given**
2. Let a be an arbitrary integer.
3. $\exists y.\ y \geq a$ — **Elim $\forall$: 1**
4. $b \geq a$ — **Elim $\exists$: 3, $\mathbf{b}$ depends on $\mathbf{a}$**
5.
6. $\exists y.\ \forall x.\ y \geq x$ — **Intro $\exists$: 5**

Example: an incorrect proof.

Elim $\forall$ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro $\exists$ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Intro $\forall$ $\dfrac{P(a); a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

Elim $\exists$ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Why list dependencies? To avoid incorrect proofs.

Over the integer domain: $\forall x.\ \exists y.\ y \geq x$ is True but $\exists y.\ \forall x.\ y \geq x$ is False.

1. $\forall x.\ \exists y.\ y \geq x$	**Given**
2. Let a be an arbitrary integer.	
3. $\exists y.\ y \geq a$	**Elim $\forall$: 1**
4. $b \geq a$	**Elim $\exists$: 3, $\mathbf{b}$ depends on $\mathbf{a}$**
5. [redacted]	**Intro $\forall$: 2, 4**
6. $\exists y.\ \forall x.\ y \geq x$	**Intro $\exists$: 5**

Example: an **incorrect proof**.

Can't get rid of a since another name, b, in the same formula depends on it!

Elim $\forall$ $\dfrac{\forall x.\ P(x)}{\therefore\ P(a) \text{ for any } a}$

Intro $\forall$ $\dfrac{P(a);\ a \text{ is arbitrary}}{\therefore\ \forall x.\ P(x)}$

The name a stands for an arbitrary value in the domain. No other name in P depends on a.

Intro $\exists$ $\dfrac{P(c) \text{ for some } c}{\therefore\ \exists x.\ P(x)}$

Elim $\exists$ $\dfrac{\exists x.\ P(x)}{\therefore\ P(c) \text{ for a specific } c}$

The name c is **fresh** and stands for a value in the domain where $P(c)$ is true. List all dependencies for c.

Natural language proofs

From formal proofs to natural language proofs.

Natural language versus (predicate) logic proofs

We often write proofs in English rather than as fully formal proofs.

They are easier for people to read.

(But theorem provers prefer fully formal proofs. :)

English proofs follow the structure of the corresponding formal proofs.

Formal proof methods help to understand how proofs work in English.

And they give clues for how to produce the proofs in English.

A not so odd example in English

Prove that there is an even integer.

1.	$2 = 2 \cdot 1$	**Arithmetic**
2.	$\exists y.\, 2 = 2 \cdot y$	**Intro $\exists$: 1**
3.	$\mathrm{Even}(2)$	**Definition of Even: 2**
4.	$\exists x.\, \mathrm{Even}(x)$	**Intro $\exists$: 3**

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

A not so odd example in English

Prove that there is an even integer.

$2 = 2 \cdot 1$

1.	$2 = 2 \cdot 1$	**Arithmetic**
2.	$\exists y.\ 2 = 2 \cdot y$	**Intro $\exists$: 1**
3.	$\mathrm{Even}(2)$	**Definition of Even: 2**
4.	$\exists x.\ \mathrm{Even}(x)$	**Intro $\exists$: 3**

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Even}(x) ::= \exists y.\ x = 2 \cdot y$

A not so odd example in English

Prove that there is an even integer.

$2 = 2 \cdot 1$

so 2 equals 2 times an integer.

1.	$2 = 2 \cdot 1$	**Arithmetic**
2.	$\exists y.\ 2 = 2 \cdot y$	**Intro $\exists$: 1**
3.	$\mathrm{Even}(2)$	**Definition of Even: 2**
4.	$\exists x.\ \mathrm{Even}(x)$	**Intro $\exists$: 3**

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Even}(x) ::= \exists y.\ x = 2 \cdot y$

A not so odd example in English

Prove that there is an even integer.

$2 = 2 \cdot 1$
so 2 equals 2 times an integer.
Therefore 2 is even.

1. $2 = 2 \cdot 1$ **Arithmetic**
2. $\exists y.\ 2 = 2 \cdot y$ **Intro ∃: 1**
3. $\text{Even}(2)$ **Definition of Even: 2**
4. $\exists x.\ \text{Even}(x)$ **Intro ∃: 3**

Domain of discourse	**Predicate definitions**
Integers	$\text{Even}(x) ::= \exists y.\ x = 2 \cdot y$

A not so odd example in English

Prove that there is an even integer.

$2 = 2 \cdot 1$
so 2 equals 2 times an integer.
Therefore 2 is even.
Therefore there is an even integer. $\square$

1.	$2 = 2 \cdot 1$	**Arithmetic**
2.	$\exists y.\, 2 = 2 \cdot y$	**Intro ∃: 1**
3.	$\mathrm{Even}(2)$	**Definition of Even: 2**
4.	$\exists x.\, \mathrm{Even}(x)$	**Intro ∃: 3**

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Even}(x) ::= \exists y.\, x = 2 \cdot y$

A square example in English

Prove that the square of every even number is even.

Let a be an arbitrary even integer.

Then, by definition, $a = 2b$
for some integer b, depending on a.
Squaring both sides, we get $a^2 = 4b^2 = 2(2b^2)$.
Since $2b^2$ is an integer, by definition,
a^2 is even.

Since a was arbitrary, it follows that
the square of every even number is even. □

1.	Let a be an arbitrary integer.	
2.1.	$\mathrm{Even}(a)$	Assumption
2.2.	$\exists y.\ a = 2y$	Definition of Even: 2.1
2.3.	$a = 2b$	Elim ∃: 2.2, **b** depends on **a**
2.4.	$a^2 = 4b^2 = 2(2b^2)$	Algebra
2.5.	$\exists y.\ a^2 = 2y$	Intro ∃: 2.4
2.6.	$\mathrm{Even}(a^2)$	Definition of Even: 2.5
3.	$\mathrm{Even}(a) \rightarrow \mathrm{Even}(a^2)$	Direct Proof Rule
4.	$\forall x.\ \mathrm{Even}(x) \rightarrow \mathrm{Even}(x^2)$	Intro ∀: 1, 3

Domain of discourse	Predicate definitions
Integers	$\mathrm{Even}(x) ::= \exists y.\ x = 2 \cdot y$

An odd square example in English

Prove that the square of every odd number is odd.

Domain of discourse	Predicate definitions
Integers	$\mathrm{Odd}(x) ::= \exists y.\, x = 2 \cdot y + 1$

An odd square example in English

Prove that the square of every odd number is odd.

Proof

Let b be an arbitrary odd number.

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Odd}(x) ::= \exists y.\, x = 2 \cdot y + 1$

An odd square example in English

Prove that the square of every odd number is odd.

Proof

Let b be an arbitrary odd number.
Then, $b = 2c + 1$ for some integer c (depending on b).

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Odd}(x) ::= \exists y.\, x = 2 \cdot y + 1$

An odd square example in English

Prove that the square of every odd number is odd.

Proof

Let b be an arbitrary odd number.
Then, $b = 2c + 1$ for some integer c (depending on b).
Therefore, $b^2 = (2c+1)^2 = 4c^2 + 4c + 1 = 2(2c^2 + 2c) + 1$.

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Odd}(x) ::= \exists y.\, x = 2 \cdot y + 1$

An odd square example in English

Prove that the square of every odd number is odd.

Proof

Let b be an arbitrary odd number.
Then, $b = 2c + 1$ for some integer c (depending on b).
Therefore, $b^2 = (2c+1)^2 = 4c^2 + 4c + 1 = 2(2c^2 + 2c) + 1$.
Since $2c^2 + 2c$ is an integer, b^2 is odd.

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Odd}(x) ::= \exists y.\, x = 2 \cdot y + 1$

An odd square example in English

Prove that the square of every odd number is odd.

Proof

Let b be an arbitrary odd number.
Then, $b = 2c + 1$ for some integer c (depending on b).
Therefore, $b^2 = (2c+1)^2 = 4c^2 + 4c + 1 = 2(2c^2 + 2c) + 1$.
Since $2c^2 + 2c$ is an integer, b^2 is odd.
The statement follows since b was arbitrary. $\square$

Domain of discourse	**Predicate definitions**
Integers	$\mathrm{Odd}(x) ::= \exists y.\, x = 2 \cdot y + 1$

A rational example in English

A real number x is rational iff there exist integers p and q with $q \neq 0$ such that $x = p/q$.

Prove: "If x and y are arbitrary rational numbers then xy is rational."

Domain of discourse	**Predicate definitions**
Reals	$\text{Rational}(x) \equiv \exists p.\, \exists q.\, x = \frac{p}{q} \wedge \text{Integer}(p) \wedge \text{Integer}(q) \wedge q \neq 0$

A rational example in English

A real number x is rational iff there exist integers p and q with $q \neq 0$ such that $x = p/q$.

Prove: "If x and y are arbitrary rational numbers then xy is rational."

Proof

By the definition of rational, $x = a/b$ for some integers a, b, where $b \neq 0$, and $y = c/d$ for some integers c, d, where $d \neq 0$.

Domain of discourse	**Predicate definitions**
Reals	$\text{Rational}(x) \equiv \exists p.\, \exists q.\, x = \frac{p}{q} \wedge \text{Integer}(p) \wedge \text{Integer}(q) \wedge q \neq 0$

A rational example in English

A real number x is rational iff there exist integers p and q with $q \neq 0$ such that $x = p/q$.

Prove: "If x and y are arbitrary rational numbers then xy is rational."

Proof

By the definition of rational, $x = a/b$ for some integers a, b, where $b \neq 0$, and $y = c/d$ for some integers c, d, where $d \neq 0$.
Multiplying, we get that $xy = (ac)/(bd)$.

Domain of discourse	**Predicate definitions**
Reals	$\text{Rational}(x) \equiv \exists p.\, \exists q.\, x = \frac{p}{q} \wedge \text{Integer}(p) \wedge \text{Integer}(q) \wedge q \neq 0$

A rational example in English

A real number x is rational iff there exist integers p and q with $q \neq 0$ such that $x = p/q$.

Prove: "If x and y are arbitrary rational numbers then xy is rational."

Proof

By the definition of rational, $x = a/b$ for some integers a, b, where $b \neq 0$, and $y = c/d$ for some integers c, d, where $d \neq 0$.
Multiplying, we get that $xy = (ac)/(bd)$.
Since b and d are both non-zero, so is bd; furthermore, ac and bd are integers.

Domain of discourse	**Predicate definitions**
Reals	$\text{Rational}(x) \equiv \exists p.\, \exists q.\, x = \frac{p}{q} \wedge \text{Integer}(p) \wedge \text{Integer}(q) \wedge q \neq 0$

A rational example in English

A real number x is rational iff there exist integers p and q with $q \neq 0$ such that $x = p/q$.

Prove: “If x and y are arbitrary rational numbers then xy is rational.”

Proof

By the definition of rational, $x = a/b$ for some integers a, b, where $b \neq 0$, and $y = c/d$ for some integers c, d, where $d \neq 0$.
Multiplying, we get that $xy = (ac)/(bd)$.
Since b and d are both non-zero, so is bd; furthermore, ac and bd are integers.
It follows that xy is rational, by definition of rational.

Domain of discourse	**Predicate definitions**
Reals	$\text{Rational}(x) \equiv \exists p.\, \exists q.\, x = \frac{p}{q} \land \text{Integer}(p) \land \text{Integer}(q) \land q \neq 0$

Benefits of English proofs

This is more work to write

```
%a = add %i, 1
%b = mod %a, %n
%c = add %arr, %b
%d = load %c
%e = add %arr, %i
store %e, %d
```

than this

```
arr[i] = arr[(i+1) % n];
```

Higher level language is easier because it skips details.

Benefits of English proofs

This is more work to write

```
%a = add %i, 1
%b = mod %a, %n
%c = add %arr, %b
%d = load %c
%e = add %arr, %i
store %e, %d
```

than this

```
arr[i] = arr[(i+1) % n];
```

Higher level language is easier because it skips details.

Formal proofs are the low level language: each part must be spelled out in precise detail.

Benefits of English proofs

This is more work to write

```
%a = add %i, 1
%b = mod %a, %n
%c = add %arr, %b
%d = load %c
%e = add %arr, %i
store %e, %d
```

than this

```
arr[i] = arr[(i+1) % n];
```

Higher level language is easier because it skips details.

Formal proofs are the low level language: each part must be spelled out in precise detail.

English proofs are the high level language.

An English proof is correct if the *reader* is convinced they can “compile” it to a formal proof if necessary.

Proof strategies

Proof by contrapositive, counterexamples, and proof by contradiction.

Proof by contrapositive

If we assume $\neg q$ and derive $\neg p$, then we have proven that $\neg q \to \neg p$, which is equivalent to proving $p \to q$.

1.1. $\neg q$ **Assumption**

$\dots$

1.3. $\neg p$

2. $\neg q \to \neg p$ **Direct Proof Rule**
3. $p \to q$ **Contrapositive: 2**

Counterexamples

To *disprove* $\forall \mathbf{x}.\ \mathbf{P}(\mathbf{x})$, prove $\exists \mathbf{x}.\ \neg \mathbf{P}(\mathbf{x})$.

Works by DeMorgan's Law: $\neg \forall x.\ P(x) \equiv \exists x.\ \neg P(x)$.

All we need to do is find an x for which $P(x)$ is false.

This x is called a *counterexample*.

Example: disprove that "Every prime number is odd".

Counterexamples

To *disprove* $\forall \mathbf{x}.\, \mathbf{P}(\mathbf{x})$, prove $\exists \mathbf{x}.\, \neg \mathbf{P}(\mathbf{x})$.

Works by DeMorgan's Law: $\neg \forall x.\, P(x) \equiv \exists x.\, \neg P(x)$.

All we need to do is find an x for which $P(x)$ is false.

This x is called a *counterexample*.

Example: disprove that "Every prime number is odd".

2 is a prime number that is not odd.

Proof by contradiction

If we assume p and derive F (a contradiction), then we have proven $\neg p$.

1.1. p **Assumption**

$\ldots$

1.3. F

2. $p \rightarrow \mathsf{F}$ **Direct Proof Rule**
3. $\neg p \vee \mathsf{F}$ **Law of Implication: 2**
4. $\neg p$ **Identity: 3**

An example proof by contradiction

Prove that "No integer is both even and odd."

English proof: $\neg\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x) \equiv \forall x.\ \neg(\mathrm{Even}(x) \wedge \mathrm{Odd}(x))$.

Proof by contradiction

An example proof by contradiction

Prove that "No integer is both even and odd."

English proof: $\neg\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x) \equiv \forall x.\ \neg(\mathrm{Even}(x) \wedge \mathrm{Odd}(x))$.

Proof by contradiction

Let x be an arbitrary integer and suppose that it is both even and odd.

An example proof by contradiction

Prove that "No integer is both even and odd."

English proof: $\neg\exists x.\ \mathrm{Even}(x) \land \mathrm{Odd}(x) \equiv \forall x.\ \neg(\mathrm{Even}(x) \land \mathrm{Odd}(x))$.

Proof by contradiction

Let x be an arbitrary integer and suppose that it is both even and odd.
Then $x = 2a$ for some integer a and $x = 2b + 1$ for some integer b.

An example proof by contradiction

Prove that "No integer is both even and odd."

English proof: $\neg\exists x.\ \mathrm{Even}(x) \wedge \mathrm{Odd}(x) \equiv \forall x.\ \neg(\mathrm{Even}(x) \wedge \mathrm{Odd}(x))$.

Proof by contradiction

Let x be an arbitrary integer and suppose that it is both even and odd.
Then $x = 2a$ for some integer a and $x = 2b + 1$ for some integer b.
Therefore $2a = 2b + 1$ and hence $a = b + \frac{1}{2}$.

An example proof by contradiction

Prove that "No integer is both even and odd."

English proof: $\neg\exists x.\, \mathrm{Even}(x) \wedge \mathrm{Odd}(x) \equiv \forall x.\, \neg(\mathrm{Even}(x) \wedge \mathrm{Odd}(x))$.

Proof by contradiction

Let x be an arbitrary integer and suppose that it is both even and odd.
Then $x = 2a$ for some integer a and $x = 2b + 1$ for some integer b.
Therefore $2a = 2b + 1$ and hence $a = b + \frac{1}{2}$.
But two integers cannot differ by $\frac{1}{2}$ so this is a contradiction.

An example proof by contradiction

Prove that "No integer is both even and odd."

English proof: $\neg\exists x.\, \mathrm{Even}(x) \wedge \mathrm{Odd}(x) \equiv \forall x.\, \neg(\mathrm{Even}(x) \wedge \mathrm{Odd}(x))$.

Proof by contradiction

Let x be an arbitrary integer and suppose that it is both even and odd.
Then $x = 2a$ for some integer a and $x = 2b + 1$ for some integer b.
Therefore $2a = 2b + 1$ and hence $a = b + \frac{1}{2}$.
But two integers cannot differ by $\frac{1}{2}$ so this is a contradiction.
Therefore no integer is both even and odd. $\square$

Fun strategy: proof by computer

Use an automated theorem prover:

```
; No integer is both even and odd.

(define-fun even ((x Int)) Bool
  (exists ((y Int)) (= x (* 2 y))))

(define-fun odd ((x Int)) Bool
  (exists ((y Int)) (= x (+ (* 2 y) 1))))

(define-fun claim () Bool
  (not (exists ((x Int)) (and (even x) (odd x)))))

(assert (not claim)) ; proof by contradiction

(check-sat)
```

Fun strategy: proof by computer

Use an automated theorem prover:

```
; No integer is both even and odd.

(define-fun even ((x Int)) Bool
  (exists ((y Int)) (= x (* 2 y))))

(define-fun odd ((x Int)) Bool
  (exists ((y Int)) (= x (+ (* 2 y) 1))))

(define-fun claim () Bool
  (not (exists ((x Int)) (and (even x) (odd x)))))

(assert (not claim)) ; proof by contradiction

(check-sat)
```

While this example works, proofs of arbitrary formulas in predicate logic *cannot* be automated. But *interactive theorem provers* can still help by checking your formal proof and filling in some low-level details for you.

Fun fact: counterexamples & contradiction in verification

Automated verifiers work by counterexample and contradiction proofs.

Recall that program verification involves proving that a program P satisfies a specification S on all inputs x: $\forall x.\, p(x) \rightarrow s(x)$, where p and s are formulas encoding the semantics of P and S.

Fun fact: counterexamples & contradiction in verification

Automated verifiers work by counterexample and contradiction proofs.

Recall that program verification involves proving that a program P satisfies a specification S on all inputs x: $\forall x.\, p(x) \rightarrow s(x)$, where p and s are formulas encoding the semantics of P and S.

The program verifier sends the formula $\mathbf{\exists x.\, p(x) \land \neg s(x)}$ to the prover.

$\neg\forall x.\, p(x) \rightarrow s(x) \equiv \exists x.\, \neg(p(x) \rightarrow s(x)) \equiv \exists x.\, \neg(\neg p(x) \lor s(x)) \equiv \exists x.\, p(x) \land \neg s(x).$

Fun fact: counterexamples & contradiction in verification

Automated verifiers work by counterexample and contradiction proofs.

Recall that program verification involves proving that a program P satisfies a specification S on all inputs x: $\forall x.\ p(x) \rightarrow s(x)$, where p and s are formulas encoding the semantics of P and S.

The program verifier sends the formula $\mathbf{\exists x.\ p(x) \land \neg s(x)}$ to the prover.

$\neg\forall x.\ p(x) \rightarrow s(x) \equiv \exists x.\ \neg(p(x) \rightarrow s(x)) \equiv \exists x.\ \neg(\neg p(x) \lor s(x)) \equiv \exists x.\ p(x) \land \neg s(x)$.

If the prover finds a counterexample, we know the program is incorrect.

The counterexample is a concrete input (test case) on which the program violates the spec.

Fun fact: counterexamples & contradiction in verification

Automated verifiers work by counterexample and contradiction proofs.

Recall that program verification involves proving that a program P satisfies a specification S on all inputs x: $\forall x.\, p(x) \rightarrow s(x)$, where p and s are formulas encoding the semantics of P and S.

The program verifier sends the formula $\mathbf{\exists x.\, p(x) \wedge \neg s(x)}$ to the prover.

$\neg\forall x.\, p(x) \rightarrow s(x) \equiv \exists x.\, \neg(p(x) \rightarrow s(x)) \equiv \exists x.\, \neg(\neg p(x) \vee s(x)) \equiv \exists x.\, p(x) \wedge \neg s(x).$

If the prover finds a counterexample, we know the program is incorrect.

The counterexample is a concrete input (test case) on which the program violates the spec.

If no counterexample exists, we know the program is correct.

Because this is proof by contradiction! The prover assumed $\exists x.\, p(x) \wedge \neg s(x)$ and arrived at false (“unsat”).

Summary

Formal (logic) proofs follow well-defined rules and are easy to check.

- They can be checked mechanically.
- And are used in the construction of critical software.

English proofs correspond to those rules but are easier for people to read.

- Easily checkable in principle.

Simple proof strategies already do a lot.

- Later we will cover a specific strategy that applies to loops and recursion (mathematical induction).