## Formal Logic

Background

This topic deals with Logic (rational inquiry). Two main aims are to:

* Learn a new language, the language of first-order logic (FOL).
* Learn about the notion of logical consequence, and about how one goes about establishing whether some claim is or is not a logical consequence of other accepted claims

Language, Proof and Logic is an educational software package used to teach formal logic using a textbook and four software programs.

Software for this course:

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Named after | Use | Description |
| Tarski’s World | Alfred Tarski | Atomic languages | Teaches the basic first-order language and its semantics using a model theoretic-like approach, where the "world" consists of a little grid and some simple objects |
| Fitch | Frederic Brenton Fitch | Proofs | Fitch-style calculus for checking first-order proofs |
| Boole | George Boole | Boolean connectives | Construction of truth tables and related notions (tautology, tautological consequence, etc.) |

**Lesson 0**

First-Order Logic

This is a symbolic, artificial language of science used in rational inquiry. It incorporates elements of human languages to precisely explain, without ambiguity, informal notions like grammaticality, meaning, truth and proof.

Example

Atomic Languages

Before we can use FOL, we need to understand how to construct atomic sentences, the simplest sentences in FOL.

Atomic Language

Object

**name**

e.g. *a*

**Predicate symbol**

e.g. *Cube()*

**name**

e.g. Cl*aire*

**Function symbol**

e.g. *father\_of()*

*Makes a claim about the world Refers to object in the world*

*i.e. “Claire” denotes a person in the world*

In FOL, every symbol comes with a fixed “arity”, a number that tells you how many names it needs to form an atomic sentence.

In Tarski’s world (a tool for interpreting logic), the following atomic sentences are used:

|  |  |  |
| --- | --- | --- |
| Atomic Sentence | Arity | Interpretation |
| Tet(a) | 1 | *a* is a tetrahedron |
| Cube(a) | 1 | *a* is a cube |
| Dodec(a) | 1 | *a* is a dodecahedron |
| Small(a) | 1 | *a* is small |
| Medium(a) | 1 | *a* is medium |
| Large(a) | 1 | *a* is large |
| SameSize(a,b) | 2 | *a* is the same size as *b* |
| SameShape(a,b) | 2 | *a* is the same shape as *b* |
| Larger(a,b) | 2 | *a* is larger than *b* |
| Smaller(a,b) | 2 | *a* is smaller than *b* |
| SameRow(a,b) | 2 | *a* is in the same row as *b* |
| Adjoins(a,b) | 2 | *a* and *b* are located on adjacent (but not diagonally) squares |
| LeftOf(a,b) | 2 | *a* is located nearer to the left edge of the grid than *b* |
| RightOf(a,b) | 2 | *a* is located nearer to the right edge of the grid than *b* |
| FrontOf(a,b) | 2 | *a* is located nearer to the front edge of the grid than *b* |
| BackOf(a,b) | 2 | *a* is located nearer to the back of the grid than *b* |
| Between(a,b,c) | 3 | *a*, *b* and *c* are in the same row, column,  or diagonal, and *a* is between *b* and *c* |

**Lesson 1**

Formal properties

**Properties of Binary Operations**

Identity:

* Properties of one expression can translate to another if the expressions are equivalent

Inverse:

* The opposite.

Idempotent:

* This means that no matter applied multiple times without changing the result beyond the initial application.

*Pressing the “on” button on a calculator is idempotent,* *it has the same effect whether done once or multiple times.*

**Rules of replacement (**"can be replaced in a logical proof with".)

Commutativity (Symmetrical) Property:

* P Q Q P
* P Q Q P

Associativity:

* P (Q R) (P Q) R
* P (Q R) (P Q) R

Distributivity:

* P (Q R) (P Q) (P R)
* P (Q R) (P Q) (P R)

Double Negation:

* P P
* P P

De Morgan’s Law: Negate all the names AND connectives

* (P Q) P Q
* (P Q) P Q

Transposition:

* (P Q) Q P

Exportation:

* (P Q) R P (Q R)

Material implication:

* (P Q) Q

Tautology:

* P P P (idempotency of conjunction)
* P P P (idempotency of disjunction)

Conjunction is **idempotent**.

Conjunction is **commutative** (symmetrical).

**Lesson 2**

Sound Arguments

Logicians use logical consequence to link together conclusions (statements) to premises (preceding statements).

Example

Cube(a) *a* is a cube

a = b *a* is the same object as *b*

therefore, Cube(b) *b* is a cube

Identity Relations **I–R–S-T**

(Also known as logic axioms for identity)

Four important principles that hold of the identity relation:

**IDENTITY**

=**Elim**: **indiscernibility of identicals.**

If b = c, then whatever holds of b holds of c.

*Eliminate or remove parts*

=**Intro**: **Reflexivity of identity**

b = b, is always true in FOL

*Introduce or add parts*

**Symmetry of Identity**

If b = c, then c = b

**Transitivity of Identity.**

If a = b, and b = c, then a = c

substitution

**REFLEXIVITY**

x = x

**SYMMETRY**

x = y y = x

**TRANSIVITY**

x = y y = z

How to give informal proofs using arguments

Example

RightOf(b,c)

LeftOf(d,e)

b = d

LeftOf(c,e)

We are told that b is to the right of c.

So c must be to the left of b, since right of and left of are inverses of one another. And since b = d, c is left of d, by the indiscernibility of identicals. But we are also told that d is left of e, and consequently c is to the left of e, by the transitivity of left of.

This is our desired conclusion.

**Lesson 3**

Logical Possibility (propositional Equivalence)

**A close up of a sign

Description automatically generatedTautology.**

The saying of the same thing twice over in different words

A tautology is a formula which is "**always true"**

The negation of a tautology is a TT-Contradiction

P P

A close up of a sign

Description automatically generated

**Contradiction. TT-Contradiction**

A combination of statements, ideas, or features which are opposed to one another.

A tautology is a formula which is "**always false"**

The negation of a TT-Contradiction is a Tautology

P P



**Contingency. TT-Contingency**

Neither a tautology nor contradiction (at least one F and one T in its truth table)

P Q R

In Tarski’s world, we use the truth table to show that certain sentences cannot possibly be false. In this world, the truth table method works only in one direction: when it says that a sentence is logically necessary, then it is.

A screenshot of a cell phone

Description automatically generated

Types of possibilities

iff at least one row on its truth table reads T

<https://www.ocf.berkeley.edu/~brianwc/courses/logic/notes04.html>

|  |  |  |
| --- | --- | --- |
| **1** | **TT-Possible** | iff at least one row on its truth table reads T under it’s main connective |
|  | Tautology | iff every row on its truth table reads T under it’s main connective |
| **2** | **TW-Possible** | iff it is true in at least one world in Tarski's World (can be built in Tarski’s World) |
| **3** | **TW-necessary** | iff it is true in every world in Tarski's World |
|  | tautologically equivalent | iff every row of their joint truth table assigns the same values to each |

**Lesson 4**

Boolean Connectives

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Meaning | Alternatives | Description |
|  | Not |  | Negation. |
|  | And |  | Conjunction. |
|  | Or |  | Disjunction. |

Below are the game rule’s in Tarski’s world.

Game rule for Negation () (01)

|  |  |
| --- | --- |
| P | P |
| T | F |
| F | T |

–

We avoid using double negatives, it usually does not make a difference.

If you commit to the truth of P, you commit to the falsity of P.

Game rule for Conjunction () (1000)

False unless both conjuncts are true

|  |  |  |
| --- | --- | --- |
| P | Q | P Q |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

If you commit to the truth of P ^ Q then you have implicitly committed yourself to the truth of each of P and Q

Game rule for Disjunction () (1110)

True unless both disjuncts are false

|  |  |  |
| --- | --- | --- |
| P | Q | P Q |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

If you commit yourself to the truth of P Q, then Tarski's World will make you live up to this by committing yourself to the truth of one or the other

When two sentences are logically equivalent, each is a logical consequence of

the other.

**Lesson 5**

Conditionals (Logical Consequence)

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Meaning | Alternatives | Description |
|  | If… then |  | Material Conditional (Implication) |
|  | Iff  OR  “Just in case” |  | Material biconditional (Equivalence). "can be replaced in a logical proof with". |

Logical truth - A sentence is a logical consequence of a set of sentences if it is impossible for that sentence to be false when all the sentences in the set are true.

Game rule for Implication () (1011)

True, unless antecedent is true when consequent is false

|  |  |  |
| --- | --- | --- |
| P | Q | P Q |
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

In Tarski’s World, P Q is another way of saying ¬P Q.

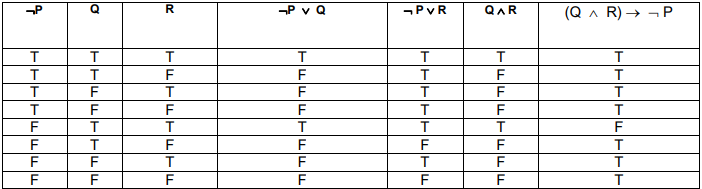
example

If Max is home then Claire is at the library

Home(max) Library (claire)

Example: May 2015 Q1.9

What interpretation can be made about the third sentences as a consequence of 1 and 2?

1. ¬P ∨ Q
2. ¬P ∨ R
3. (Q ∧ R) → ¬ P

The third sentence is neither a

tautological consequence *apply tautology game rule: always T*

nor a logical consequence *apply implication game rule: always T unless* *predicate is T F*

of the first and second sentences.

Game rule for Biconditional () (1001)

True if and only if P and Q have the same truth value

|  |  |  |
| --- | --- | --- |
| P | Q | P Q |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

example

Max is home if and only if Claire is at the library

Home(max) Library (claire)

**Lesson 6**

<https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

Truth Tables

1. Calculate number of rows we need

*For n distinct statements, we need rows*

1. Fill in the columns and their headings for each distinct statement
2. Apply game rule to the remaining columns

Example:

Construct a truth table for the following FOL sentence

¬(p ∨ q) → r

Step 1: Calculate number of rows

*three distinct statements p, q and r. we need or 8 rows*

|  |  |  |
| --- | --- | --- |
| p | q | r |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Step 2: Fill in the columns for the 3 distinct statements

*Column 1: Alternate with half of 8 (4 T’s, 4 F’s)*

*Column 2: Alternate with half of 4 (2 T’s, 2 F’s, 2 T’s, 2 F’s)*

*Column 3: Alternate with half of 2 (T F T F T F T F)*

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| p | q | r |
| T | T | T |
| T | T | F |
| T | F | T |
| T | F | F |
| F | T | T |
| F | T | F |
| F | F | T |
| F | F | F |

Step 3: Apply game rule Apply game rule to the remaining columns

*Column 4: (1,2) Disjunction game rule, true unless both false*

*Column 5: (4) Negation game rule*

*Column 6: (3,5) Implication game rule, True, unless antecedent is true when consequent is false*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| p | q | r | p ∨ q | ¬(p ∨ q) | ¬(p ∨ q) → r |
| T | T | T | T | F | T |
| T | T | F | T | F | T |
| T | F | T | T | F | T |
| T | F | F | T | F | T |
| F | T | T | T | F | T |
| F | T | F | T | F | T |
| F | F | T | F | T | T |
| F | F | F | F | T | F |

Example: May 2015, Q4

Construct a truth table for the following FOL sentence

(A ∨ ¬B) ∧ ¬(A ∨ C)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | B | C | A ∨ C | ¬(A ∨ C) | ¬B | A ∨ ¬B | (A ∨ ¬B) ∧ ¬(A ∨ C) |
| T | T | T | T | F | F | T | F |
| T | T | F | T | F | F | T | F |
| T | F | T | T | F | T | T | F |
| T | F | F | T | F | T | T | F |
| F | T | T | T | F | F | F | F |
| F | T | F | F | T | F | F | F |
| F | F | T | T | F | T | T | F |
| F | F | F | F | T | T | T | T |

*4: (1,3) Disjunction game rule, true unless both false*

*5: (4) Negation game rule*

*6: (2) Negation game rule*

*7: (1,6) Disjunction game rule, true unless both false*

*8: (5,7) Conjunction game rule, False unless both conjuncts are true*

Statement is a TT-Contingency (not all true and not all false)

**Lesson 7**

Arguments

An argument is a series of statements in which one, the conclusion, is meant to follow from or be supported by, the others, the premises.

An argument is sound if it is logically valid and its premises are all true.

Remember:

Valid/Sound Argument

*Logical consequence of its premises*

true premise

true conclusion

true premise

true premise

Valid/Unsound Argument

*Not assured that conclusion is true*

false premise

??? conclusion

true premise

true premise

Invalid Argument

*Not a logical consequence of its premises*

true premise

false conclusion

true premise

true premise

*These are also Counterexamples*

An argument can be valid but unsound if the conclusion is true, but the premises are not all true.

Example: Valid/Sound argument.

All men are mortal. Socrates is a man. So, Socrates is mortal

Example: Invalid argument.

Lucretius is a man. After all, all men are mortal and Lucretius is mortal

*First statement is the conclusion. Lurcetius could be a goldfish*

Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Form | Your commitment | Player to move | Goal |
| T | T | You  Tarski’s World | Choose one of P,Q that is true |
| T | F | Tarski’s World  you | Choose one of P,Q that is false |
| F | either | - | Replace P by P and switch commitment |

**Lesson 8**

Formal Proofs: Methods of Proof

How do you tell if a sentence is a consequence of others?

Counterexample: show that a conclusion is not a logical consequence of it’s premise’s

A Proof: show that a conclusion is a logical consequence of it’s premise’s

Boolean connectives also give rise to two entirely new methods of proof:

* Proof by cases (disjunction elimination)
* Indirect proof: Proof by contradiction (tautologies)

**Proof by cases**

In a proof by cases, we must cover all possible cases that arise in a theorem.

Example: prove if n is an integer

We know that Case 1 Case 2 Case 3

Case 1: , so it follows that

Case 2:

, so it follows that

Case 3:

, so it follows that

Since our inequality is true for all possible cases, we can conclude for all integers

Structure:

1. We know (n cases)
2. Our goal is to prove S
3. If is the case, S follows

OR

1. If is the case, S follows

OR

1. If is the case, S follows

Example

(Home(max) Happy(carl)) (Home(Claire) Happy(scruffy))

We want to prove that either Carl or Scruffy is happy

Happy(carl) Happy(scruffy)

Case 1: Home(max) Happy(carl)

So it follows that Happy(carl) Happy(scruffy)

Case 2: Home(claire) Happy(scruffy)

So it follows that Happy(carl) Happy(scruffy)

**Proof by contradiction** *reductio ad absurdum.*

Remember: Contradiction is a combination of statements, ideas, or features which are opposed to one another.

In proof by contradiction, we assume a proposition is not true. Then through premise and logic, find a contradiction that shows our original premise must have been incorrect

Structure: **proposition**

1. Our goal is to prove S
2. Assume S
3. Prove a contradiction

Structure: **implication (material conditional)**

1. Our goal is to prove P Q
2. Assume P and Q are true
3. Prove a contradiction that shows P Q

OR

1. Prove a contradiction that shows P Q

Example: prove that is irrational

is irrational = S

Therefore, assume that is rational

Therefore, there exist 2 integers where and have no common factors.

Although, there exist no integers where the above is true. Hence, is irrational via contradiction

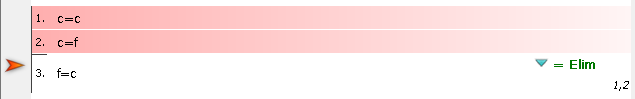
**Lesson 9**

Fitch Proofs: Examples

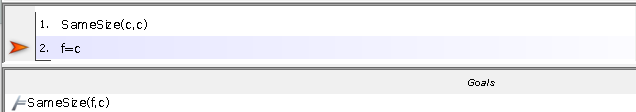
This formal proof method relies on

* Fitch Rules
* Rules of inference

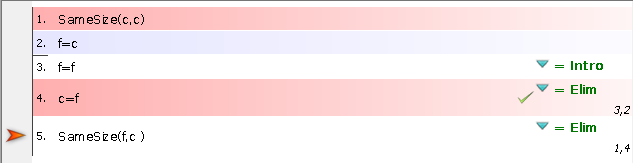
Example: =Elim with names



*Cite: 1. an identity 2. the line you wish to change*

Example: =Elim with predicates

*Cannot use =Eim 1,2 as f=c and SameSize(f,c) are the same.*

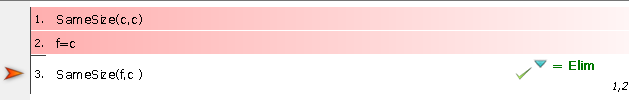


*Add in the line you want to change c=f, Citing 3. an identity 2. the line you wish to change*

*Add in a support line f=f (needs no support lines)*

*Add the goal as a step. Cite 1. An identity, 4. The line you wish to change*

**OR**



*Fitch shortcut: Cite 1. An identity 2. The line you wish to change (can be reversed)*

Example: How to use fitch (ASS 2, Q5.2)

The above video explains how to do step 4 properly

1. Add your goal and add your goal as your final step.

A screenshot of a cell phone

Description automatically generated

1. Separate P (from P V R) into subproof.

We can infer Q from P using → Elim.

To use it, cite a conditional + a line that logically precedes it

A screenshot of a cell phone

Description automatically generated

1. A close up of a logo

   Description automatically generatedJoin P → Q to Q → (R V S) using →Intro

Write the conclusion P → (R V S)

Above this, add the antecedent(P), then the consequent(Q). Cite this for the conclusion

A screenshot of a cell phone

Description automatically generated

1. We are still missing steps inside of our conclusion.

Add the antecedent of P → Q (mainly Q) inside our subproof.

The antecedent can be inferred using → Elim.

A screenshot of a cell phone

Description automatically generated

1. The antecedent of Q → (R V S) (mainly RVS) can now be

inferred using → Elim.

A screenshot of a cell phone

Description automatically generated

1. The conclusion of P → (R V S) can now use the subproof for → Intro.

A screenshot of a cell phone

Description automatically generated

1. Add in ¬Intro (Could be useful later)

A screenshot of a cell phone

Description automatically generated

1. Disjunction intro to get R V S R

TODO: finish this

**Lesson 10**

Methods of Proof: Rules of inference

**Conjunction Rules**

Conjunction Elimination ( Elim)

If you have a conjunction in a proof, you may enter on a new line, any of it’s conjuncts.

Example

Cube(a) Large(a)

Cube(a)

Cannot be used if it is embedded as part of a larger sentence (e.g. where the proof has negation in it)

(Cube(a) Large(a))

(Cube(a)

Conjunction Introduction ( Intro)

If you have several sentences in a proof, you may enter on a new line, their conjunction. May occur in any order

Example

1. P Q
2. Q R

3. P Elim: 1

4. R Elim: 2

5. P R Intro: 3, 4

**Disjunction Rules**

Disjunction Introduction ( Intro)

If you have a sentence on a line in a proof, you may enter on a new line, any disjunction of which it is a disjunct.

Example

1. P
2. Q
3. P Q Intro: 1,2

Disjunction Elimination ( Elim)

Corresponds to the method of proof by cases (cover every single possibility in the theorem). It incorporates the formal device of a ***subproof***.

Example

1. (A B) (C D)

2. A B

3. B Elim: 2

4. B D Intro: 3

5. C D

6. D Elim: 5

7. B D Intro: 6

1. B D Elim: 1,2 -4, 5, -7

To reduce “too many disjuncts” we use Disjunctive normal form (DNF)



<https://www.youtube.com/watch?v=Jg2YCTq_9VM>

DNF: where each disjunct is a conjunction, and

in that conjunction, each conjunct is a literal

*Each conjunct should not be surrounded by brackets*

Steps to get to DNF:  
1. Eliminate arrows

Material implication:

* (P Q) Q

2. Deal with negations

Double Negation:

* P P
* P P

De Morgan’s Law: *Negate all the names AND connectives*

* (P Q) P Q
* (P Q) P Q

3. Deal with conjunction & disjunction

Distributivity:

* P (Q R) (P Q) (P R)
* P (Q R) (P Q) (P R)

Example: May 2015, Q1.7

Convert the following to DNF:

¬[(A ∨ ¬B) ∧ ¬(A ∧ D)] ∨ [C ∧ (A ∨ D)]

*Deal with negations*

≡ ¬[(A ∨ ¬B) ∧ (¬A ∨ ¬D)] ∨ [C ∧ (A ∨ D)]

≡ [(¬A ∧ B)∨ (A ∧D)] ∨ [C ∧ (A ∨ D)]

*Deal with conjunctions/disjunctions*

≡ [(¬A ∧ B)∨ (A ∧D)] ∨ (C ∧ A) ∨ (C ∧ D)

≡ (¬A ∧ B)∨ (A ∧D) ∨ (C ∧ A) ∨ (C ∧ D)

Example: May 2015, Q1.7

Convert the following to DNF:

¬((A ∨ B) ∧ C) ∨ ((A ∨ C) ∧ D)

*Deal with negations*

≡ (¬ (A ∨ B) ∧ ¬C) ∨ ((A ∨ C) ∧ D)

≡ ((¬A ∧ ¬B) ∧ ¬C) ∨ ((A ∨ C) ∧ D)

*Deal with conjunctions/disjunctions*

≡ ((¬A ∧ ¬B) ∧ ¬C) ∨ (A ∧ D) ∨ (C ∧ D)

≡ (¬A ∧ ¬B) ∧ ¬C ∨ (A ∧ D) ∨ (C ∧ D)

**Negation Rules**

Negation Elimination Introduction ( Elim)

This simple rule allows us to eliminate “double negations”

Example

1. A
2. A Elim: 1

To reduce “double negations” we use negation normal form (NNF)

Steps to get to NNF:  
1. Deal with negations

Double Negation:

* P P
* P P

Example

¬¬(A ∨ ¬B) ∧ ¬(¬¬A ∧ ¬B)

= (A ∨ ¬B) ∧ ¬(¬¬A ∧ ¬B)

= (A ∨ ¬B) ∧ ¬(A ∧ ¬B)

= (A ∨ ¬B) ∧ (¬A V B)

A screenshot of a cell phone

Description automatically generatedRemember:

Remember:

The Up tack or falsum () is a logical constant denoting a false proposition in logic, often called "falsum" or "absurdum". It is always assigned the truth value of F.

Negation Introduction ( Intro)

A version of the method of indirect proof, or proof by contradiction.

If an assumption made leads to , you may close the subproof and derive as a conclusion the negation of the sentence that was the assumption.

Example

1. P P

2. P Elim: 1

3. P Elim: 2

4. Intro: 2,3

5. (P P) Intro: 4

Negation Introduction ( Intro)

Lines in this proof must be in the form P P (i.e. contradictory).

To use this rule, the two sentences must be identical (symbol for symbol), except for the negation sign at the beginning of one of them.

Example

1. Tet(a) Teb(b)
2. Tet(a) Teb(b)
3. Tet(a) Elim: 2
4. Tet(b) Elim: 2

5. Tet(a)

6. Intro 3,5

7. Tet(b)

8. Intro: 4,7

9. Elim: 1,5-6,7-8

TODO:  
Introduction: Intro

Elimination: Elim

Conditional Intro

Let’s us add the line x=x without citing a support line?

Conditional Elimination

**Lesson 11**

Informal methods of proof: Rules of Inference

**Conditional elimination:** “method of affirming”

P implies Q and P is asserted to be true, therefore Q must be true.

P and P → Q, you may infer Q.

Example: “John will prove a theorem only if he isn't very tired. He slept very well last night, so he'll prove a theorem.”

P: John is not tired

Q: John will prove a theorem

**Biconditional elimination:** “method of affirming for the biconditional”

From P and P ↔ Q , you may infer Q.

From P and Q ↔ P , you may infer Q.

**Contraposition**

P → Q ⇔ ¬Q → ¬P

**“conditional – disjunction” equivalence**

P → Q ⇔ ¬P ∨ Q

**“negated conditional” equivalence**

¬(P → Q) ⇔ P ∧ ¬Q

**“biconditional – conjunction” equivalence**

¬(P → Q) ⇔ P ∧ ¬Q

**“biconditional – disjunction” equivalence**

P ↔ Q ⇔ (P → Q) ∧ (Q → P)

**Lesson 12**

Well-Formed Formulae

We have at our disposal at least two languages:

1. A formal language. An abstract, truth-functional language (Propositional Logic: PL)
2. A natural language (English)

**Translations: Atomic Well-Formed-Formulae (WFF’s)**

A WFF is a string of symbols that is part of the formal language.

Natural language prioritises communication and flexibility

Formal languages prioritise precision and rigidity.

To create a WFF, we need:

The correct components

The correct order

Example: Doctor(x) ∧ Smart(x)

**Lesson 13**

Quantification

**Universal quantifier** ()

Express universal claims, those we express in English using quantified phrases like *everything, each thing, all things,* and *anything*.

Example:

Every doctor is smart.

Negating a universal:

*∀x is now ∃x. Remember to also negate your predicate*

x(Doctor(x) → smart(x))

**Existential Quantifier** ()

Express existential claims, those we express in English

thing, a, an using such phrases as *something, at least one thing, a,* and *an*.

Example:

Negation of an existential:

*∃x is now ∀x. Remember to also negate your predicate*

Some doctor is smart.

x(Doctor(x) → smart(x))

*x is smart*

*every x is smart*

*every doctor is smart*

**Prenex form** (PNF)

Quantifiers are at the start of the formula, with no quantifiers in the body

Example:

xy(Small(x) ∧ Large(y) ∧ FrontOf(y,x))

*Some large object in front of some small object*

xy(Small(x) ∧ Large(y) → FrontOf(y,x))

*Every large object in front of every small object*

**Aristotelian form**

Quantifiers areat the start of the formula and in the body

x(Small(x) ∧ y(Large(y) ∧ FrontOf(y,x))

*Some large object in front of some small object*

*This is a lot easier to read.*

*“Some object is small and has some other property”*

x(Small(x) → y (Large(y) → FrontOf(y,x))

*Every large object in front of every small object*

*We need to re-arrange the formula using:*

Exportation:

* (P Q) R P (Q R)

Four Aristotelian forms:

All P’s are Q’s. ∀x (P(x) → Q(x))

Some P’s are Q’s. ∃x (P(x) ∧ Q(x))

No P’s are Q’s. ∀x (P(x) → ¬Q(x))

Some P’s are not Q’s. ∃x (P(x) ∧ ¬Q(x))

**Distinct Variables**

Remember to explicitly state if you want two distinct objects using ¬ (x = y)

Example: Bill has at least two sisters.

∃x, y ( SisterOf(x, Bill) ∧ SisterOf(y, Bill) ∧ ¬ (x = y))

*There exists someone who is Billy’s sister (x)*

*There exists someone who is Billy’s sister (y)*

*x is not y*

*Billy has at least two sisters*

Example: Every student loves some student.

∀x (Student(x) ⇒ ∃y ( Student(y) ∧ Loves(x,y)))

*x and y are not necessarily distinct (every student could love themselves)*

Example: Every student loves some other student.

*x and y are distinct*

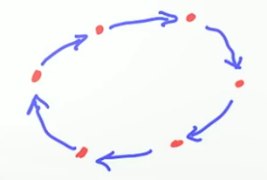
∀x ( Student(x) ⇒ ∃y ( Student(y) ∧ ¬ (x = y) ∧ Loves(x,y) ))

Example: There is a student who is loved by every other student.

∃x ( Student(x) ∧ ∀y ( Student(y) ∧ ¬(x = y) ⇒ Loves(y,x) ))

**Ordering of Mixed Quantifiers**

The translation of sentences may change depending on the order of quantifiers



Example:

∀x∃y Likes(x,y)

*Everybody likes someone*

Example:

∃y∀x Likes(x,y)

*There exists someone who is liked by everybody*

Example: ASS 3 Q7.4

∃x Student(x) ∧ ∀y Student (y) ∧ ¬ (x = y) → Loves(y,x))

*y loves x*

*x is not y*

*There exists some student (x)*

*Every student (y)*

*there exists a student who is loved by every other student*

**Exactly One**

We can also use ordering of mixed quantifiers to indicate that there is exactly one object

Example:

∃x(Tet(x) ∧ ∀y(Tet(y) → x=y)

*There exists a tetrahedron and it has “some other property”*

*There exists a tetrahedron and anything that is a tetrahedron, is identical to that tetrahedron*

*There is exactly one tetrahedron*

**Other Examples**

Example: Billy has at most one sister.

∀x, y ( SisterOf(x, Bill) ∧ SisterOf(y, Bill) ⇒ x = y)

*Everybody could be Billy’s sister (x)*

*y is Billy’s sister*

*x is y*

*Billy has at most one sister*

Example: Billy has exactly one sister.

∃x ( SisterOf(x, Bill) ∧ ∀y ( SisterOf(y, Bill) ⇒ x = y ))

*There exists someone who is Billy’s sister*

*Everyone could be Billy’s sister*

*x is y*

*Billy has exactly one sister*

Example: May2015 Q1.4

∀y∀x((Tet(x) ∧ Dodec(y)) → SameSize(x, y))

Every tetrahedron is the same size of every dodecahedron

*x is the same size as y*

*Every x is the same size as Every y*

*Every tetrahedron is the same size as every dodecahedron*

Example: May2015 Q1.4

Most x [(Cube(x), Most y(Tet(y), FrontOf(x, y)]

*x is in front of y*

*Most x are in front of Most y*

*Most cubes are in front of Most Tetrahedra*

Example: ASS 3 Q7.5

∀xy (SisterOf(x, Billy) ∧ SisterOf(y, Billy) → (x = y))

*???*

Example: ASS 3 Q7.6

¬∃x (Student(x) ∧ ∀y (Student (y) ∧ ¬ (x = y) → Fools(x, y))

*x fools y*

*Some x fools every y*

*Some student fools every other student*

*No student fools every other student*

**TW examples:**

∀x∀y(Cube(x) ∧ Tet(y) → LeftOf(x,y)

Every cube is left of every tetrahedron.

∃x∃y(Cube(x) ∧ Tet(y)∧LeftOf(x,y))

Some cube is left of some tetrahedron.

∀x∀y((Cube(x) ∧ Tet(y)) → ¬LeftOf(x,y))

No cube is left of any tetrahedron

∃x(Cube(x) ∧ ∀y(Tet(y) → ¬LeftOf(x,y)))

Some cube is not left of some tetrahedron

∀x∀y((Tet(x) ∧ Tet(y)) → SameSize(x,y))

Every tetrahedron is the same size

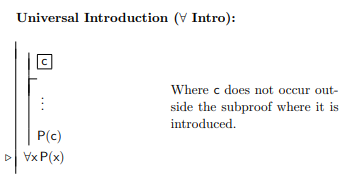
∀x(Dodec(x) → ∃y(Cube(y) ∧ BackOf(x,y)))

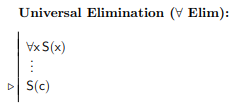
Every dodecahedron has some cube in front of it

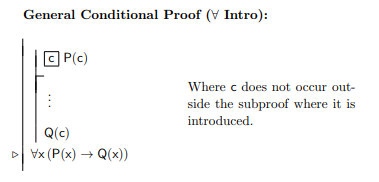
**Lesson 14**

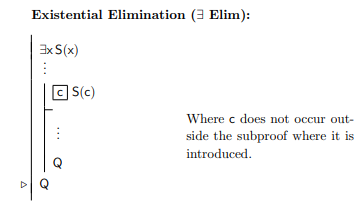
Methods of Proof for Quantifiers

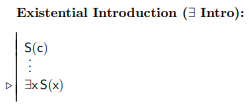
**Universal Quantifier Rules**



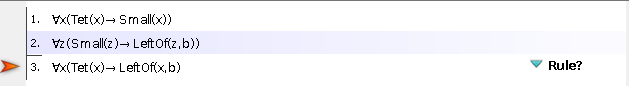




**Existential Quantifier Rules**



Example: <https://www.youtube.com/watch?v=aiE_pzdconI>



Step 1: Add a subproof. Make an assumption about S(c).

*Universal Elimination:*

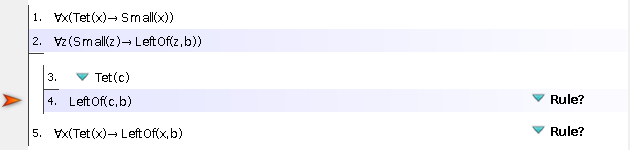
*It’s probably safe to assume the object c is a tetrahedron*



Step 2: Make the goal of the subproof a statement that is in your final goal

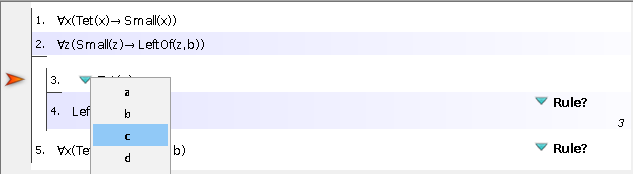
*General Conditional Proof:*

*It’s probably safe to assume that our subproof goal should be LeftOf(c,b)*

**

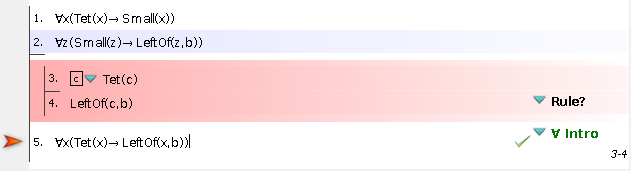
Step 3: Select c as our constant in the submenu

*Introduce the constant c*



Step 4: Cite the entire subproof

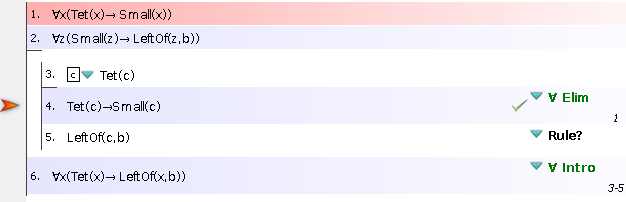
*General Conditional Proof: ∀ Intro*



Step 5: Eliminate the ∀x

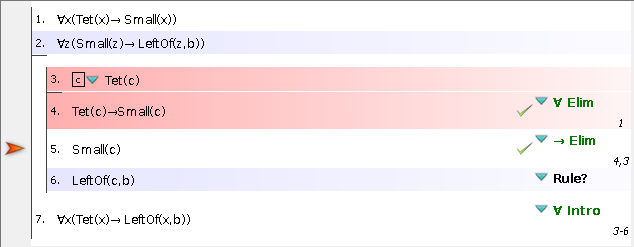
*Universal Elimination: ∀x Elim*

*Cite 1.*



Step 6: Get to our intermediary conclusion, c is small

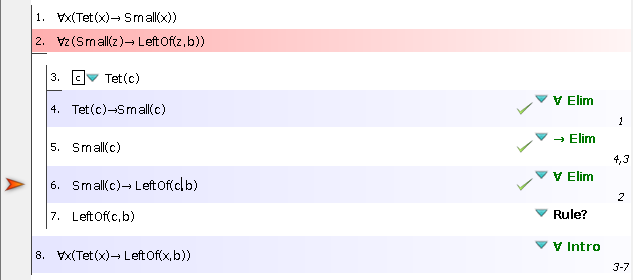
*Cite our object and Elimnate Tet(c)*



Step 7: Eliminate the ∀xon 2.

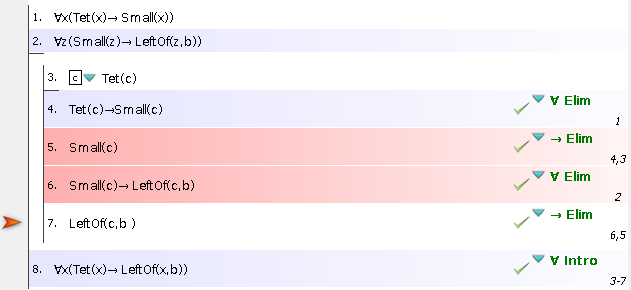
*Universal Elimination: ∀x Elim*

*Cite 2.*



Step 8: Get our conclusion

*Cite our object and Elimnate Small(c)*



**Lesson 15**

Fitch Rule Summary

<https://www.ocf.berkeley.edu/~brianwc/courses/logic/rulesummary.html>

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sentences you can prove** | **Sentences to cite** | **How to use** |
| **Identity Introduction** (= Intro) | Self-Identity (a=a, b=b, c=c …) | None | Introduce a Self-Identity on any line of a proof and cite nothing, using the rule = Intro. |
| **Identity Elimination** (= Elim) | Any sentence using at least one name Large(a), Smaller(b, c), Home(max) … | * Identity * A sentence you are replacing a name in, that uses at least one of the names from the identity sentence cited. Always cite just two prior lines. | Introduce a sentence on any line of a proof that changes one or more occurrences of a name from a previous sentence. Cite that sentence you are changing, and cite the identity sentence that says the change you are making is legitimate. |
| **Conjunction Introduction** (And Intro) | A Conjunction | Any. May cite as many prior lines as you like, and each will be a conjunct. | Introduce a new conjunction on any line of a proof by citing each of the conjuncts from prior lines. These conjuncts must be alone on the line cited. |
| **Conjunction Elimination** (And Elim) | Any | Cite one conjunction only. | Instructions for use: Remove a conjunct from a previous line containing a conjunction. |
| **Disjunction Introduction** (Or Intro) | A Disjunction | Any. Cite only one prior line, it will be a disjunct. | You can cite any prior sentence available and create a disjunction containing as one conjunct the prior line cited and as another disjunct any sentence you like. |
| **Disjunction Elimination** (Or Elim) | Any | Must cite one disjunction, a subproof for each disjunct within that disjunction, and nothing else. | Cite a disjunction, create a subproof for each disjunct that begins with each disjunct in turn. End each subproof with the exact same goal, and then that identical goal sentence is justified outside of the subproofs. |
| **Negation Elimination** (Not Elim) | Any | Cite only a negation of a negation. | If there is a sentence with at least two negations on it, you can take the negations off, two at a time, with this rule. Cite only one sentence. |
| **Negation Introduction** (Not Intro) | Any | Cite only a single subproof that begins with the opposite of what you hope to prove and ends with Contra . | Begin a subproof with the opposite of what you want to prove outside of the subproof. End the subproof with Contra . Cite only the subproof. |
| **Contradiction Introduction** (Contra Intro) | Contra only | * A sentence and, * Exactly that sentence, negated. Cite only two sentences. | Find a sentence and it's negation. Cite both and write Contra on a line. |
| **Contradiction Elimination** (Contra Elim) | Any | You must cite only a single line containing Contra. | If you prove Contra on a line you may cite that line and write any sentence you please on a subsequent line. |
| **Conditional Introduction** (-> Intro) | Only a Conditional | You must cite only a single subproof. | To prove a conditional statement, make a subproof that begins with the antecedent and ends with the consequent. |
| **Conditional Elimination** (-> Elim) | Any | You must cite exactly two sentences:   1. a conditional and, 2. a sentence that is the antecedent of the conditional in 1). | You can only prove the consequent of the conditional cited in 1) above. |
| **Biconditional Introduction** (<-> Intro) | Only a Biconditional | You must cite exactly two subproofs. | To prove a biconditional statement, make a subproof that begins with the left and ends with the right and make another subproof that begins with the right and ends with the left. |
| **Biconditional Elimination** (<-> Elim) | Any | You must cite exactly two sentences   1. a Biconditional and, 2. a sentence that is either the left or right side of the biconditional in 1). | Instructions for use: You prove one side of the biconditional cited in 1) above. |
| Rules/Mechanisms outside of Fitch | | | |
| **Taut Con** | Any | Any/Varies | Only use to prove things based merely on the logic of connectives, things that could be done with our other normal rules (minus the identity rules). |
| **FO Con** | Any | Any/Varies | Only use to prove things based merely on the logic of connectives and identity, things that could be done with our other normal rules including the identity rules. (If the sentence could be proven without identity rules, use Taut Con instead.) |
| **Ana Con (Analytical consequence)** | Any | Any/Varies | Use to prove things that are true based on the meanings of the predicates in Tarski's World, and that cannot be proven with any other normal rule or Con rule. |
| **Reit (Reiteration)** | Any | Any/Varies | Allows you to simply REITerate, or repeat, any prior available line with no changes whatsoever. Remember, lines from previously completed subproofs may not be reiterated. |

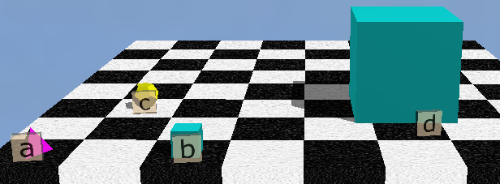
■ Should not be used in a formal proof

**Lesson 16**

Tarski’s World

Example: May 2015, Q1.2

Given the TW, which statements are true?



1. ∀y(Dodec(y) → SameCol(y,d))

*Every dodecahedron is in the same column as shape d*

1. ∀x (Tet(x) → (Small(x) ∧ LeftOf(x, b)))

*Every tetrahedron is small and on the left of shape b*

Only 2 is true (1 should read same “column” not row)

Example: May 2015, Q1.3 Given the TW above, which statements are true?

1. ∀x(SameRow(x,c) → Cube(x))

*Every cube is in the same row as shape c*

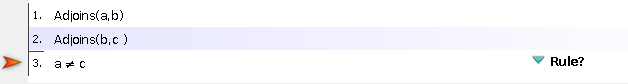
2. BackOf(d,b)

Only 2 is true

**Nonconsequence and counterexamples**

The premise(s) must be true

The conclusion must be false

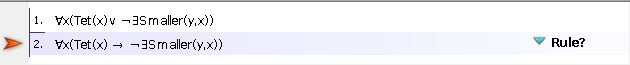
Example:

*If a and c are the same shape*



Example: May 2015 Q7

Provide a counter-example where the premise is true but the conclusion not



∀x(Tet(x) ∨ ¬∃ySmaller(y,x))

*y is not smaller than x*

*some y are smaller than every x*

*some shapes are smaller than every tetrahedron*

*No shapes are smaller than every tetrahedron*

**

∀x(Tet(x) → ∃ySmaller(y,x))

*y is smaller than x*

*some y are smaller than every x*

*some shapes are smaller than every tetrahedron*

Exam Curriculum

Propositional logic:

~~logic of atomic sentences (FOL) L0~~

~~WFF’s L12~~

~~Boolean connectives (TW Game rules) L4~~

~~Logical Consequence (TW Game rules) L5~~

~~Nonconsequence (counterexamples) L7 L8~~

~~Counterexamples in TW L12~~

*Do more examples of this!!*

~~Logical Possibility (propositional Equivalence) L3 L11~~

Formal proofs: Fitch proofs L9

Rule summary L15

Rules of inference (formal proofs) L10

~~NNF~~

~~DNF~~

Informal proofs:

Rules of Inference L11

Quantifiers:

~~logic of quantifiers L13~~

~~Methods of Proof for Quantifiers L14~~