## Formal Logic III

Background

This is a continuation of Formal Logic II. The aims are:

propositional logic – declarative sentences, natural deduction, semantics, normal forms

predicate logic – syntax, natural deduction, semantics, undecidability

modal logics – syntax, semantics, logic engineering, natural deduction, multi-agent systems

**Lesson 0**

Propositional logic  
 : Susan saves

: Susan buys a house

: The house has seven bathrooms

: The house is used as a guesthouse

: Peter likes the house

**“only if”**

“The house is used as a guesthouse sonly if it has seven bathrooms”

“Susan buys a house only if the house does not have seven bathrooms”

**“unless”**

“if… then”

“Unless Susan saves, she does not buy a house with seven bathrooms”

“Unless Peter does not like the house, Susan saves and buys a house used as guesthouse”

*Negate one side of the statement*

**“if and only if”**

“logically equivalent to”

“The house is used as a guesthouse if and only if Susan does not buy it, but Peter likes it”

“Peter likes the house if and only it has seven bathrooms or is not used as guesthouse”

*Remember “but” in logic is treated as “AND”*

**Lesson 0**

Symbols

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | phi | Premise/formula |
|  | psi | Conclusion |
|  | Proves | Proves/implies/yields. (Syntactically entails) |
|  | True | Satisfies (models) |
|  | Sequent | Is valid if a proof can be found |
|  |  |  |

We apply proof rules to premises, we get formulas.

We apply more proof rules to formulas, we get a conclusion

Hence:

Example:

Satisfiability:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | Phi | Premise/formula |
|  | Psi | Conclusion |
|  | True | Satisfies (models) |
|  | Model M |  |
|  | Environment |  |
|  | Satisfaction relation |  |
|  | Semantic entailment | Is valid if a proof can be found |

**Lesson 1**

Natural Deduction Rules

Natural deduction is used to examine the validity of arguments. It can sometimes be more efficient than a truth table. This is like Fitch Rules, where the goal is to build a proof, using some type of ruleset.

Recap:

Elim: Eliminate or remove parts

Intro: Introduce or add parts

Example

*Prove that implies*

1 assume a & b.

2 b. &E2 1

3 a. &E1 1

4 b & a. &I 2,3

5 therefore a & b => b & a. =>I 1,4

Conjunction ()

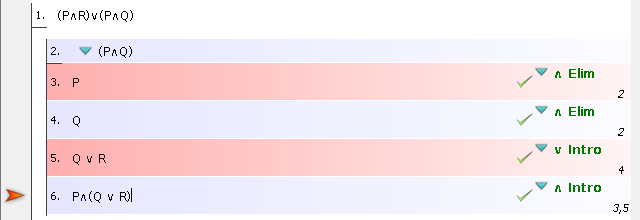
: I am happy

: The sun is shining

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Sentences to cite | Usage |
|  | **And\_Intro.**  Replaces the current goal by the two goals A and B  *I am happy, and the sun is shining*  *is true. is true. Implies is true* | 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. |
|  | **And\_Elim\_1 in H.**  Applies to an assumption of the form H:  And generates a new assumption HO:  *I am happy*  *is true. Implies is true* | (*Elim): for each conjunct, the assumption.*  *Enter every conjunct from the assumption. On a new line* | Remove a conjunct from a previous line containing a conjunction. |
|  | **And\_Elim\_2 in H.**  Applies to an assumption of the form H:  And generates a new assumption HO:  *The sun is shining*  *is true. Implies is true* | (*Elim): for each conjunct, the assumption.*  *Enter every conjunct from the assumption. On a new line* | Remove a conjunct from a previous line containing a conjunction. |
|  | **And\_Elim\_all in H.**  Applies to an assumption of the form H:  And generates two assumptions  H:  HO:  The tactic is then recursively applied to H and HO |  |  |

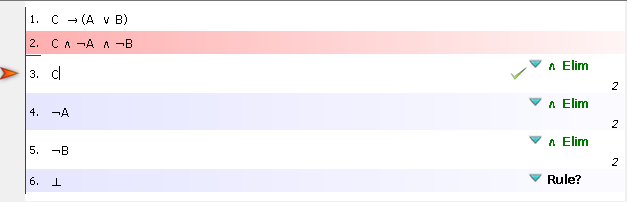
*Intro: Cite as many sentences as you like*

*All must be conjuncts*



*Elim: for each conjunct, cite 2.*

*Enter every conjunct from 2. On a new line*



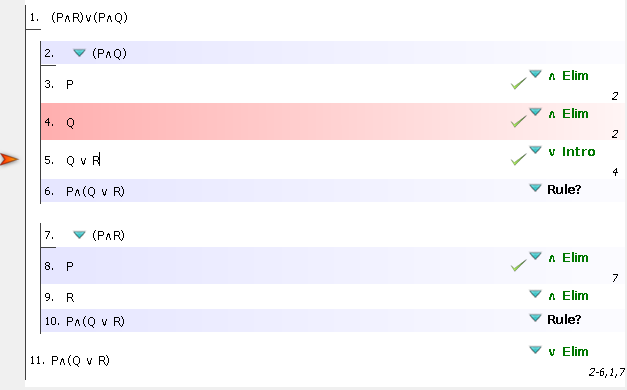
Disjunction ()

: I am happy

: The sun is shining

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Sentences to cite | Usage |
|  | **Or\_Intro\_1**  Replaces the current goal by the goal | 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. |
|  | **Or\_Intro\_2.**  Replaces the current goal by the goal | 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. |
|  | **Or\_Elim in H.**  Applies to an assumption of the form H:  It generates two proof obligations with the assumptions  H: resp.  H: and the current goal | *(∨ Elim) Cite Assumption. And each subproof* | *Each disjunct should be subproof*  *Each subproof should have the same goal as the other*  *The goals in the subproofs should match a new goal outside* |

*(∨ Intro). Cite only one prior line, it will be a disjunct.*

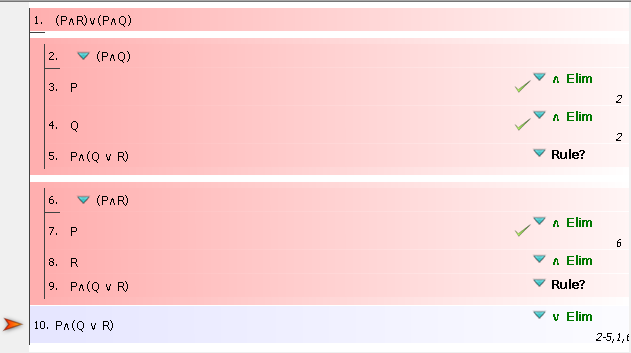


*(∨ Elim) Cite 1. And each subproof*

*Each disjunct should be subproof*

*Each subproof should have the same goal as the other (in this case,)*

*The goals in the subproofs should match a new goal outside (in this case, our intermediate conclusion ,))*

****

Implication ()

: I am happy

: The sun is shining

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Sentences to cite | Usage |
|  | **Impl\_Intro.**  Replaces the current goal by and adds the assumption H: | 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. |
|  | **Impl\_Elim in H and HO.**  Applies to the two assumptions of the form H: and HO: and adds the new assumption H1: | 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. |

*-> Intro:*

*Add a new subproof*

*The assumption of the subproof should be the antecedent ()*

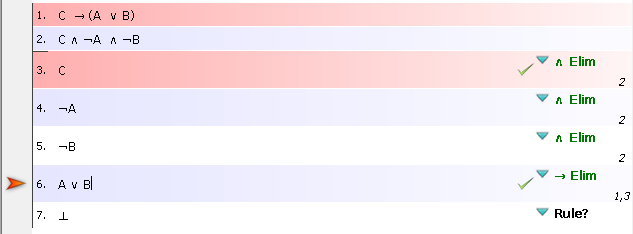
*The conclusion of the subproof should be the consequent ()*

*A picture containing screenshot, drawing

Description automatically generated*

*-> Elim: Cite a conditional (1.)*

*and a sentence that’s an antecedent of the conditional (3.)*

****

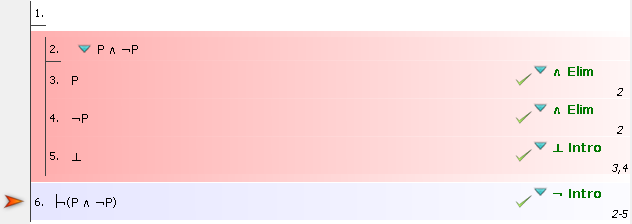
Negation ()

: I am happy

: The sun is shining

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Sentences to cite | Usage |
|  | **Not\_Intro.**  Replaces the current goal by by False and adds the assumption H: | Cite only a single sub proof that begins with the opposite of what you hope to prove and ends with Contra. | Begin a sub proof with the opposite of what you want to prove outside of the sub proof. End the sub proof with Contra.  Cite only the sub proof. |
|  | **Impl\_Elim in H and HO.**  Applies to the two assumptions of the form H: and HO: and adds the new assumption H1: False | 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. |

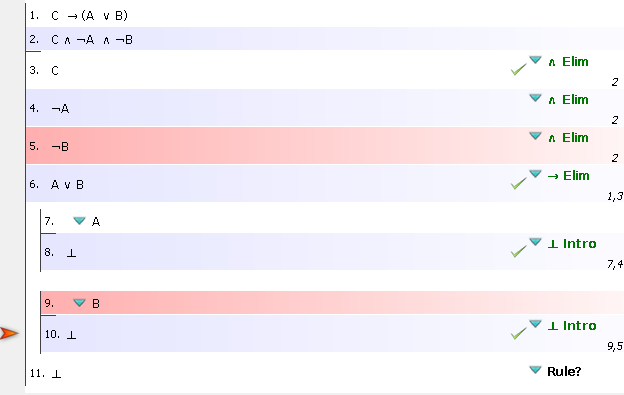
*(¬ Intro) Cite only a single subproof that begins with the opposite of what you hope to prove and ends with Contra*



|  |  |  |  |
| --- | --- | --- | --- |
|  | **PBC**  *Proof by contradiction (Contradiction Introduction)*  Replaces the current goal by False and adds the assumption H: A. | *(⊥ intro)*   * A sentence and, * Exactly that sentence, negated. Cite only two sentences. | Find a sentence and its negation. Cite both and write Contra on a line. |

*(⊥ intro) Cite 4. and 7.*

*(⊥ intro) Cite 5. and 9.*

****

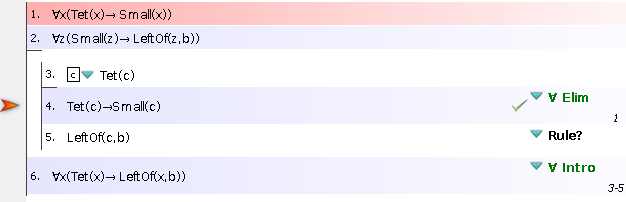
Universal ()

: I am happy

: The sun is shining

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Forall\_Intro.**  Replaces the current goal forall by and adds the variable : to the assumptions. |  |  |
|  | **Forall\_Elim in H with t.**  Applies to an assumption of the form H : forall . It generates a new assumption HO: . |  |  |

*∀ Elim: Cite 1.*



Existential ()

: I am happy

: The sun is shining

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Exists\_Intro with t.**  Replaces the current goal exist by |  |  |
|  | **Exists\_Elim in H.**  Applies to an assumption of the form H : exists . It adss the variable : and the new assumption HO: |  |  |

Proofs practice (ASS1)

**ASS1: Q7**

A picture containing riding

Description automatically generated

*(∨ Elim) Cite 1. And each subproof*

*Each disjunct should be subproof*

*Each subproof should have the same goal as the other (in this case,)*

*The goals in the subproofs should match a new goal outside (in this case, our intermediate conclusion, )*

A screenshot of a cell phone

Description automatically generated

*Elim: for each conjunct, cite the assumption.*

*Enter every conjunct from the assumption. On a new line*

*A screenshot of a cell phone

Description automatically generated*

*Intro: Cite as many sentences as you like*

*All must be conjuncts*

A screenshot of a cell phone

Description automatically generated

*(∨ Intro). Cite only one prior line, it will be a disjunct.*

A screenshot of a cell phone

Description automatically generated

**ASS1: Q8**

A close up of an object

Description automatically generated

*-> Intro:*

*Add a new subproof*

*The assumption of the subproof should be the antecedent ()*

*The conclusion of the subproof should be the consequent ()*

A picture containing table

Description automatically generated

*A picture containing screenshot, beach, table

Description automatically generated*

Use . You already have in subproof. Try get into subproof

*-> Elim: Cite a conditional (1.)*

*and a sentence that’s an antecedent of the conditional (4.)*

*A screenshot of a cell phone

Description automatically generated*

*Intro: Cite as many sentences as you like*

*All must be conjuncts*

*A screenshot of a cell phone

Description automatically generated*

**ASS1: Q9**

A screenshot of a cell phone

Description automatically generated

*-> Intro:*

*Add a new subproof*

*The assumption of the subproof should be the antecedent ()*

*The conclusion of the subproof should be the consequent ()*

A screenshot of a cell phone

Description automatically generated

*(∨ Elim) Cite 1. And each subproof*

*Each disjunct should be subproof*

*Each subproof should have the same goal as the other (in this case)*

*The goals in the subproofs should match a new goal outside (in this case, our intermediate conclusion, )*

A screenshot of a cell phone

Description automatically generated

**Lesson 2**

Parse Trees

consists of three classes of symbols:

Proposition symbols:

Connective symbols:

Quantifiers:

Punctuation symbols:

Example:

*and bind tighter than*

*The main connective is*

*The main connective of the LHS is*

Parse tree to formula:

Example

A picture containing looking, sitting, glasses, person

Description automatically generated

*and bind tighter than*

*Main connectives are and*

**Lesson 3**

Evaluating Sequents

TODO

**Lesson 4**

Algorithms, propositional logic

TODO

**Lesson 5**

Consistency

A set of claims is consistent if it’s logically possible all of them to be true at the same time (not a contradiction)

Let be a theory

is inconsistent if

is consistent if

Example:

All humans are mortal

Some humans are not mortal

*Inconsistent (A)*

All humans are mortal

Simon is immortal

*Consistent, assuming Simon in not human*

All humans are mortal

Simon is immortal

Simon is human

*Inconsistent (A)*

*If Simon is immortal, Simon cannot be human*

**Proof-based:** verify a system using a proof to prove

**Model-based:** computing whether a model satisfies a formula

**Lesson 6**

Satisfaction Relation

<https://en.wikibooks.org/wiki/Formal_Logic/Predicate_Logic/Satisfaction>

We have rules for assigning truth to sentences.

*Every object is true ()*

However, this raises two problems:

[1]

will normally have free variables.

It will normally have free α {\displaystyle \alpha \,\!}

*These are not sentences and do not have a truth value*

[2]

we do not yet have a precise way of saying that φ {\displaystyle \varphi \,\!} is true of every object in the domain

We can solve these problems:

We will need an assignment of objects from the domain to the variables.

We will need to say that a model satisfies (or does not satisfy) a formula with a variable assignment.

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | Phi | Premise/formula |
|  | Psi | Conclusion |
|  | True | Satisfies (models) |
|  | Model M |  |
|  | Environment |  |
|  | Satisfaction relation |  |
|  | Semantic entailment | Is valid if a proof can be found |

The symbol is overloaded

denotes satisfiability

denotes semantic entailment

Whenever ,

Then for all models and look-up tables

Satisfiability:

This is the relation between a wff and a free variable.

We can then define truth in a model in terms of the satisfaction relation:

Text

Description automatically generated

Summary :

<https://math.stackexchange.com/questions/2266491/relation-between-satisfaction-relation-and-truth>

**[1]** We have an alphabet and a language, so there is a universe where all the possible combinations lie, empty word and so on.  
**[2]** We define what a formula is, this makes only some words, let's say blue. Now we have some blue things and some colorless.  
**[3]** We choose some blue's as axioms and this is also suddenly make them green.  
**[4]** Then we define a sequent calculus and this makes everything which are reachable from green words suddenly red. Axioms are also now red.

…

Quantifier sentences and the domain:

<https://www.youtube.com/watch?v=fmf8DjhhLrM>

A sentence of the form is true iff at least one object satisfies

A sentence of the form is true iff every object satisfies

Example:

Only satisfied where is a tetrahedron

Example:

Only satisfied where is a small cube

Example:

*Substitute free occurences of with*

This now is a sentence. If the sentence is true, satisfies

Example: ASS2 Q5

Question 5

*All S’s are Q’s.*

*Some Q’s are not S’s*

The model :

*is the set of integers between 0 and 10*

*is the a subset of*

*is the set of even integers between 0 and 10*

Example: ASS2 Q6

*For this question, make a set large enough that can be true*

The model where the sentence is true:

*is the set of all integers*

: The predicate where is less than

The model where the sentence is false:

*is the set of all integers greater than 0 and less than 10*

: The predicate where is less than

Example: ASS2 Q7

*The model :*

*For the satisfaction relation to be satisfied, we need to test that is true for every object in the model .*

*The ordered pairs and*

*The ordered pairs and*

*The ordered pairs and*

*The ordered pairs and*

*The ordered pairs and*

Therefore, the model where satisfies the sentence