Logic Summary

Ömer Tarik Özyilmaz

January 2020

Contents

0 Disclaimer **4**

1 Atomic Sentences **4**
1.1 Individual Constants 4
1.2 Predicate Symbols 4
1.3 Atomic sentences 4
1.4 General first-order languages 4
1.5 Function symbols 4
1.6 The first-order language of set theory 5
1.7 The first-order language of arithmetic 5

2 The Logic of Atomic Sentences **6**
2.1 Valid and sound arguments 6
2.2 Methods of proof 6
2.3 Formal Proofs 6
2.4 Constructing proofs in Fitch 6
2.5 Demonstrating nonconsequence 6

3 The Boolean Connectives **7**
3.1 Negation symbol: ¬ 7
3.2 Conjunction symbol: ∧ 7
3.3 Disjunction symbol: ∨ 7
3.4 Remarks about the game 8
3.5 Ambiguity and parentheses 8
3.6 Equivalent ways of saying things 8
3.7 Translation 8

4 The Logic of Boolean Connectives **9**
4.1 Tautologies and logical truth . . . 9
4.2 Logical and tautological equivalence . . . 9
4.3 Logical and tautological consequence . . . 9
4.5 Pushing negation around . . . 9
4.6 Conjunctive and disjunctive normal forms . . . 10

5 Methods of Proof for Boolean Logic **10**
5.1 Valid inference steps . . . 10
5.2 Proof by cases . . . 10
5.3 Indirect proof: proof by contradiction . . . 10
5.4 Arguments with inconsistent premises . . . 10

6 Formal Proofs and Boolean Logic **10**

7 Conditionals **11**
7.1 Material conditional symbol $\rightarrow$ . . . 11
7.2 Biconditional symbol $\leftrightarrow$ . . . 11
7.3 Conversational implicature . . . 11
7.4 Truth-functional completeness . . . 11

8 The Logic of Conditionals **12**
8.1 Informal methods of proof . . . 12
8.2 Formal rules of proof for $\rightarrow$ and $\leftrightarrow$ . . . 12

9 Introduction to Quantification **13**
9.1 Variables and atomic wffs . . . 13
9.2 The quantifier symbols $\forall, \exists$ . . . 14
9.3 Wffs and sentences . . . 14
9.4 Semantics for the quantifiers . . . 14
9.5 The four Aristotelian forms . . . 14
9.6 Translating complex noun phrases . . . 14
9.7 Quantifier and function symbols . . . 15

10 The Logic of Quantifiers **16**
10.3 First-order equivalence and DeMorgan's laws . . . 16
10.4 Other quantifier equivalences . . . 16

11 Multiple quantifiers **17**
11.1 Multiple uses of a single quantifier . . . 17
11.2 Mixed quantifiers . . . 17
11.3 The step-by-step method of translation . . . 17
11.4 Paraphrasing English . . . 17
11.5 Ambiguity and context sensitivity . . . 17

11.6 Translations using function symbols 18
11.7 Prenex form 18

12 Methods of Proof for Quantifiers **18**
12.1 Valid quantifier steps 18
12.2 The method of existential instantiation 18
12.3 The method of general conditional proof 18
12.4 Proofs involving mixed quantifiers 18

13 Formal Proofs and Quantifiers **19**
13.1 Universal quantifier rules 19
13.2 Existential quantifier rules 20
13.3 Strategy and tactics 21

17 Advanced Topics in Propositional Logic **21**
17.1 Truth assignments and truth tables 21
17.3 Horn sentences 21

18 Advanced Topics in FOL **22**
18.1 First-order structures 22
18.2 Truth and satisfaction, revisited 22
18.5 Skolemization 22

0 Disclaimer

This summary is NOT enough for the exam, you should also exercise a lot to understand how to answer exam questions. Sometimes I recommend to read the book and that is purely, because some examples are two pages long. These examples can, however, help you understand the terms better. Good luck with your exam!

1 Atomic Sentences

1.1 Individual Constants

Individual constants are simply symbols that are used to refer to some fixed individual object, so they work like names in English. Every individual constant must name an existing object. No individual constant can name more than one object. An object can have more names though or even no names at all.

1.2 Predicate Symbols

Predicate symbols are used to express some property of objects or some relation between objects. They are also sometimes called *relation symbols*. Predicates consist of *arguments*, these are the logical subjects of the predicates. How many of these arguments are required to form an atomic sentence is dependent of its *arity* (arity of one means it is unary, two is called binary, three ternary, and so forth). FOL assumes that every predicate is interpreted by a *determinate property* or relation. By a determinate property, we mean a property for which there is a definite fact of the matter whether or not the object has the property. For example, Claire, who is sixteen, is young, but there is no determinate age at which a person stops being young, so this is not a determinate property.

1.3 Atomic sentences

A sentence formed by a predicate followed by the right number of names is called an *atomic sentence*. A *infix notation* is a predicate symbol placed between arguments (i.e. a = b), while a *prefix notation* is a predicate symbol placed before the arguments (i.e. =b).

1.4 General first-order languages

You need to be careful while designing a translation key and need to be as flexible as possible.

1.5 Function symbols

Some first-order languages have other expressions that can appear in atomic sentences, these are *function symbols*. Function symbols allow us to express complex claims that could

not be expressed using names and predicates. Phrases like "Max's father" can be described in the term $father(max)$. It is formed by putting a function symbol $father$ in front of the individual constant max, the result is a *complex term* we use to refer to the father of the person referred to by the name Max. The difference between predicates and function symbols is that combining function symbols with terms does not give you a sentence, but rather something that refers to an object of some sort. The arity of a function symbol is also variable.

1.6 The first-order language of set theory

The language of set theory has the following predicates:

- $=$ (equal to) : A = B iff A en B contain exactly the same elements.
- $\in$ (is a member of) : For example if a = 2 and b is the set $\{2, 4, 6\}$, we can say $a \in b$. This is only true in the case that b is a set and a is a element of that set.
- Sets can be equal if and only if for all x: $x \in A$ iff $x \in B$.
- A is a subset of B, written $A \subseteq B$, iff all elements of A are also elements of B (if $x \in A$ then $x \in B$) {1,2} $\subseteq$ {1,2,3}
- The empty set $\emptyset$ is the set that contains no elements (there is no x such that $x \in \emptyset$)
- Set A is a proper subset, written $A \subsetneq B$, iff A is a subset of B, but B has at least one additional element.
- Intersection: $A \cap B$ is $\{x | x \in A \text{ and } x \in B\}$
- Union: $A \cup B$ is $\{x | x \in A \text{ or } x \in B\}$
- Difference: $A \setminus B$ is $\{x | x \in A \text{ and } x \notin B\}$
- In large set called universe U, we can define complement as:
  $A^c := U \setminus A = \{x | x \in U | x \notin A\}$

1.7 The first-order language of arithmetic

The language of arithmetic contains only the names 1 and 0, the binary relation symbols $=$ and $<$ and two binary function symbols, $+$ and $\times$. Nothing is a term of this language unless it can be obtained by repeated application of 0, 1, $+$ and $\times$.

2 The Logic of Atomic Sentences

2.1 Valid and sound arguments

An *argument* is a series of statements in which one, called the *conclusion*, is meant to be a consequence of the others, called the *premises*. An argument is *valid* if and only if the conclusion must be true in any circumstance in which the premises are true. We say that the conclusion of a logically valid argument is a *logical consequence* of its premises. An argument is *sound* if it is valid and the premises are all true. Logic does not care about soundness, only validity counts, while in science it is the other way around.

2.2 Methods of proof

A *proof* is a demonstration of intermediate conclusions that follows from some premises. *Informal proofs* are proofs written in plain English, while *formal proofs* are proofs that are written in a formalised system (like Fitch or Tarski's world). There are four important principles that hold of the identity relation:

- = Elim: If b = c, then whatever holds of b holds of c. This is also known as the *indiscernibility of identicals*.
- = Intro: Sentences of the form b = b are always true (in fol). This is also known as the *reflexivity of identity*.
- *Symmetry of Identity*: If b = c, then c = b.
- *Transitivity of Identity*: If a = b and b = c, then a = c.

2.3 Formal Proofs

Our system for presenting formal proofs, also known as deductive system, is $\mathcal{F}$ (Fitch). The basic rules are in the textbook.

2.4 Constructing proofs in Fitch

In this section you could do some exercises and test your knowledge in Fitch. Note: the program is called Fitch, but when you write it on paper, the language is called $\mathcal{F}$.

2.5 Demonstrating nonconsequence

Proofs of consequence are proofs that a particular piece of information must be true if the given information, the premises of the argument, are correct. *Proofs of nonconsequence* are proofs that do not follow from the given information. Therefore the conclusion must be false and that can be achieved by introducing a *counterexample*. An example of this is a lawyer that is trying to save his client by proving that his opponent's statement is invalid.

3 The Boolean Connectives

3.1 Negation symbol: ¬

The symbol ¬ is used to express negation in our language, the notion we commonly express in English using terms like 'not' and 'it is not the case that'. *Literal* sentences are atomic sentences or negations of atomic sentences. We abbreviate claims as $\neg(b = c)$ into $b \neq c$. If the sentence $\neg P$ is true the sentence P is not true.

3.2 Conjunction symbol: ∧

The symbol ∧ is used to express conjunction, in English reflected in terms as 'and', 'moreover', and 'but'.

P	Q	$P \wedge Q$
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

Table 1: Truth table for ∧

3.3 Disjunction symbol: ∨

The symbol ∨ is used to express disjunction, where we use 'or' in English. It is used as an inclusive disjunction. This means that the statement is true if *at least* one of the claims is true. If we want to express a *exclusive* disjunction, we can do that like this: $[Home(john) \vee Home(mary)] \wedge \neg[Home(john) \wedge Home(mary)]]$.

P	Q	$P \vee Q$
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Table 2: Truth table for ∨

3.4 Remarks about the game

The Hintikka game is used to find out if a complex sentence is true or false, these are the rules:

- Two players
- Two roles (Abelard (F) and Eloise (T))
- Abelard wins if the sentence is false (F)
- Eloise wins if the sentence is true (T)

There are a couple of scenarios in the game:

1. $A \vee B$: Eloise chooses A or B and the game continues with that choice
2. $A \wedge B$: Abelard chooses A or B and the game continues with that choice
3. $\neg A$: The players swap roles and the game continues with A

3.5 Ambiguity and parentheses

Parentheses must be used whenever ambiguity would result from their omission. In practice, this means that conjunctions and disjunctions must be "wrapped" in parentheses whenever combined by means of some other connective. These sentences are very different for example:

$Home(max) \vee (Home(claire) \wedge Happy(carl))$

$(Home(max) \vee Home(claire)) \wedge Happy(carl)$

3.6 Equivalent ways of saying things

There are some equivalent ways of making claims. The main ones are

- *double negation*: $\neg\neg P \iff P$
- *DeMorgan*: $\neg(P \wedge Q) \iff (\neg P \vee \neg Q)$
- *DeMorgan*: $\neg(P \vee Q) \iff (\neg P \wedge \neg Q)$

3.7 Translation

In order for an FOL sentence to be a good translation of an English sentence, it is sufficient that the two sentences have the same truth values in all possible circumstances, that is, that they have the same truth conditions.

4 The Logic of Boolean Connectives

4.1 Tautologies and logical truth

Logical truths are sentences whose truth is itself a logical necessity (i.e. a = a). A subgroup of these are *tautologies* (when a sentence has only T's in the column under its main connective in a truth table). A sentence is called *TT-possible* if and only if at least one row of the truth table assigns TRUE to the sentence. Sentences that are possible in Tarksi's World are called *TW-possible*. For more detailed information about truth tables:
LPL p. 96-100.

4.2 Logical and tautological equivalence

If the sentences S and S' are in a *joint truth table* they are *tautologically*, and therefore *logically equivalent*, if and only if every row of the joint truth table assigns the same values to S and S'.

4.3 Logical and tautological consequence

A logical truth is a sentence that is a logical consequence of any set of premises, and logically equivalent sentences are sentences that are logical consequences of one another. *Tautological consequences* are when in all the cases that S is true, S' also is true. Then S' is a tautological consequence of S (so also a logical consequence).

4.5 Pushing negation around

When two sentences are logically equivalent, we can always go from the one to the other. That is why DeMorgan laws can be quite useful in giving proofs. A sentence is in *Negation Normal Form (NNF)* if and only if it is built out of atomic sentences using the three connectives $\wedge, \vee, \neg$, and all negations in it occur directly in front of atomic sentences. A number of further logical equivalences are useful when we wish to simplify complex sentences:

- idempotency of $\wedge$: $P \wedge P \iff P$
- idempotency of $\vee$: $P \vee P \iff P$
- commutativity of $\wedge$: $P \wedge Q \iff Q \wedge P$
- commutativity of $\vee$: $P \vee Q \iff Q \vee P$
- associativity of $\wedge$: $(P \wedge Q) \wedge R \iff P \wedge (Q \wedge R)$
- associativity of $\vee$: $(P \vee Q) \vee R \iff P \vee (Q \vee R)$

4.6 Conjunctive and disjunctive normal forms

A sentence is in *disjunctive normal form (DNF)* if it is a disjunction of one or more conjunctions of one or more literals. A sentence is in *conjunctive normal form (CNF)* if it is a conjunction of one or more disjunctions of one or more literals. Distribution of $\wedge$ over $\vee$ allows you to transform any sentence in negation normal form into disjunctive normal form. Distribution of $\vee$ over $\wedge$ allows you to transform any sentence in negation normal form into conjunctive normal form.

5 Methods of Proof for Boolean Logic

5.1 Valid inference steps

In an informal proof, it is always legitimate to move from a sentence P to another sentence Q if both you and your audience already know that Q is a logical consequence of P. The following patterns are valid:

- From $P \wedge Q$, infer P and infer Q (*conjunction elimination*)
- From P and Q, infer $P \wedge Q$ (*conjunction introduction*)
- From P, infer $P \vee Q (\vee R, etc.)$ (*disjunction introduction*)

5.2 Proof by cases

To prove S from $P_1 \vee ... \vee P_n$ using proof by cases, prove S from each of $P_1, ..., P_n$ (*disjunction elimination*).

5.3 Indirect proof: proof by contradiction

To prove $\neg S$, assume S and prove a contradiction $\perp$.

5.4 Arguments with inconsistent premises

A proof by contradiction shows that the premises are inconsistent. An argument with inconsistent premises is always valid, but more importantly, unsound.

6 Formal Proofs and Boolean Logic

This chapter is mostly just exercising with the rules Chapter 5 introduced. Do the exercises if you are not that good at giving proofs.

7 Conditionals

7.1 Material conditional symbol $\rightarrow$

The symbol $\rightarrow$ is used to combine two sentences P and Q to form a new sentence $P \rightarrow Q$, called a *material conditional*. The sentence P is called the antecedent of the conditional, and Q is called the consequent of the conditional.
The sentence $P \rightarrow Q$ is true if and only if either P is false or Q is true (or both). The

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 3: Truth table for $\rightarrow$

English terms for $\rightarrow$ are, for example: 'if', 'provided that', 'only if', 'implies', 'if then'. 'P if Q' is translated as $Q \rightarrow P$ and 'Unless P, Q' is translated as $\neg P \rightarrow Q$.

7.2 Biconditional symbol $\leftrightarrow$

The final connective we are going to talk about is called the material biconditional symbol. A sentence of the form $P \leftrightarrow Q$ is true if and only if P and Q have the same truth value.

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

Table 4: Truth table for $\leftrightarrow$

The English terms for $\leftrightarrow$ are, for example: 'if and only if', 'iff', 'exactly if', 'just in case', 'necessary and sufficient'.

7.3 Conversational implicature

If the assertion of a sentence carries with it a suggestion that could be cancelled (without contradiction) by further elaboration by the speaker, then the suggestion is a *conversational implicature*, not part of the content of the original claim.

7.4 Truth-functional completeness

A set of connectives is *truth-functionally complete* if the connectives allow us to express every truth function. An example is the set of Boolean connectives.

8 The Logic of Conditionals

8.1 Informal methods of proof

Three important proof steps:

- Modus ponens: From $P \to Q$, infer Q
- Biconditional elimination: From P and either $P \leftrightarrow Q$ or $Q \leftrightarrow P$, infer Q
- Contraposition: $P \to Q \iff \neg Q \to \neg P$

Some other steps:

- $P \to Q \iff \neg P \vee Q$
- $\neg(P \to Q) \iff P \wedge \neg Q$
- $P \leftrightarrow Q \iff (P \to Q) \wedge (Q \to P)$
- $P \leftrightarrow Q \iff (P \wedge Q) \vee (\neg P \wedge \neg Q)$

The method of condtional proof: To prove $P \to Q$, assume P and prove Q. To prove a number of biconditionals, try to arrange them into a cycle of conditionals.

8.2 Formal rules of proof for $\to$ and $\leftrightarrow$

Conditional Elimination ($\to Elim$):
$P \to Q$
.
.
.
P
.
.
.
Q

Conditional Introduction ($\to Intro$):

| P
|
|-
| .
| .
| .
| Q
|

$P \to Q$

Biconditional Elimination ($\leftrightarrow Elim$):

$P \leftrightarrow Q$
.
.
.
P
.
.
.
Q

Biconditional Introduction ($\leftrightarrow Intro$):

| P
|—
| .
| .
| .
| Q
| Q
|—
| .
| .
| .
| P

$P \leftrightarrow Q$

9 Introduction to Quantification

9.1 Variables and atomic wffs

Variables are kind of placeholders that indicate relationships between quantifiers and the argument positions of various predicates. The language FOL has an infinite number of variables, any of t, u, v, w, x, y and z, with or without numerical subscripts (e.g. $u_9 3 and x$). Variables can occur in *atomic well-formed formulas* in any position normally occupied by a name.

9.2 The quantifier symbols $\forall, \exists$

The symbol $\forall$ is used to express universal claims. In English that would be "for every x" or "for everything". The existential quantifier $\exists$ is used to express existential claims. In English that would be "something" or "for at least one x".

9.3 Wffs and sentences

A well-formed sentence is any n-ary predicate followed by n terms, where terms can contain either variables or constants. Using atomic wffs, we can form more complicated wffs. In a complex wff like: ((Cube(x) $\wedge$ Small(x)) $\rightarrow \exists$y LeftOf(x,y)), the x is a *free or unbound variable* while y is a *bound variable*. Something is called a sentence when there are no free variables in the wff, an example: $\forall$x((Cube(x) $\wedge$ Small(x)) $\rightarrow \exists$y LeftOf(x,y)).

9.4 Semantics for the quantifiers

When checking quantified sentences we need to observe wether they satisfy the atomic wff in a certain environment called *the domain of discourse* (for which x-range we need to check, e.g. Tarski's World). Using this, we can conclude that every sentence of the form $\forall x S(x)$ is true if and only if the wff S(x) is satisfied by every object in the domain of discourse. The same case for every sentence of the form $\exists x S(x)$: this is true true if and only if the wff S(x) is satisfied by some object in the domain of discourse.

9.5 The four Aristotelian forms

The four Aristotelian forms are translated as follows:

1. *All P's are Q's.*: $\forall x(P(x) \rightarrow Q(x))$
2. *Some P's are Q's.*: $\exists x(P(x) \wedge Q(x))$
3. *No P's are Q's.*: $\forall x(P(x) \rightarrow \neg Q(x))$
4. *Some P's are not Q's.*: $\exists x(P(x) \wedge \neg Q(x))$

9.6 Translating complex noun phrases

To translate complex noun phrases, we need to know some ground rules, that will help us translate them easier. An *existential noun phrase* is recognized by the following words: a, an, some, something, at least one, etc. and they are frequently together with $\wedge$ (Aristotelian). An *universal noun phrase* is recognized by the following words: every, each, all, etc. and they are frequently together with $\rightarrow$ (Aristotelian). The order of an English sentence may be different than its FOL version. Next to this, *all P's are Q's* does not imply that there are some P's ("all tetrahedra are cubes." is conversationally false, but in a world without tetrahedra, this sentence is true). Finally, *some P's are Q's* does not imply that not all

P's are Q's ("some freshmen that took that class got an A" does not imply that not all freshman got an A).

9.7 Quantifier and function symbols

Explains how functions and quantifiers work together, pretty straightforward.

10 The Logic of Quantifiers

10.3 First-order equivalence and DeMorgan's laws

If we take the following wffs as an example:

- $P(x) \rightarrow Q(x)$
- $\neg Q(x) \rightarrow \neg P(x)$

We will see that if we replace their free variables with new names, the resulting sentences are logically equivalent so the two wffs are *logically equivalent*.
The DeMorgan laws for quantifiers are, for any wff P(x):

1. $\neg \forall x P(x) \iff \exists x \neg P(x)$
2. $\neg \exists x P(x) \iff \forall x \neg P(x)$

10.4 Other quantifier equivalences

These are other quantifier equivalences:

1. *Pushing quantifiers past* $\wedge$ *and* $\vee$, for any wffs P(x) and Q(x):
   - $\forall x(P(x) \wedge Q(x)) \iff \forall x P(x) \wedge \forall x Q(x)$
   - $\exists x(P(x) \vee Q(x)) \iff \exists x P(x) \vee \exists x Q(x)$
2. *Null quantification*, for any wff P in which x is not free:
   - $\forall x P \iff P$
   - $\exists x P \iff P$
   - $\forall x(P \vee Q(x)) \iff P \vee \forall x Q(x)$
   - $\exists x(P \wedge Q(x)) \iff P \wedge \exists x Q(x)$
3. *Replacing bound variables*, for any wff P(x) and variable y that does not occur in P(x):
   - $\forall x P(x) \iff \forall y P(y)$
   - $\exists x P(x) \iff \exists y P(y)$

11 Multiple quantifiers

11.1 Multiple uses of a single quantifier

A sentence can also contain more than one quantifier and more than one variable (i.e. $\exists x \forall y P(x, y)$). When evaluating a sentence with multiple quantifiers, don't fall into the trap of thinking that distinct variables range over distinct object. In fact, the sentence $\forall x \forall y P(x, y)$ logically implies $\forall x P(x, x)$, and the sentence $\exists x P(x, x)$ logically implies $\exists x \exists y P(x, y)$.

11.2 Mixed quantifiers

With mixed quantifiers, the *order of the quantifiers* and the *order of the variables* are really important. For example the sentence $\exists y \forall x Likes(x, y)$ means that there is someone whom everyone likes, while $\forall x \exists y Likes(x, y)$ means that everyone likes someone. Using this example, we can also see why the order of variables is important. $\exists y \forall x Likes(x, y)$ means that there is someone whom everyone likes, while $\exists y \forall x Likes(y, x)$ means that that there is someone that likes everyone.

11.3 The step-by-step method of translation

To translate a multiple quantifiers sentence, we need to treat the consecutive sentences as single quantifiers. For example *Each cube is to the left of a tetrahedron* can first be translated to $\forall x(Cube(x) \rightarrow something)$. Then the sentence *x is to the left of a tetrahedron* can be translated to $\exists y(Tet(y) \wedge LeftOf(x, y))$. So the final form would be $\forall x(Cube(x) \rightarrow \exists y(Tet(y) \wedge LeftOf(x, y))$.

11.4 Paraphrasing English

Sometimes we can't translate a sentence using the step-by-step method, because we first need to paraphrase the English sentence. For example: *If a freshman takes a logic class, then he or she must be smart* is translated by the method as $\exists x(Freshman(x) \wedge \exists y(LogicClass(y) \wedge Takes(x, y))) \rightarrow Smart(x)$, but this does not work, as the last x is free. If we, however paraphrase the English sentence to *Every freshman who takes a logic class must be smart*, we can form the sentence $\forall x[(Freshman(x) \wedge \exists y(LogicClass(y) \wedge Takes(x, y))) \rightarrow Smart(x)]$ and this is correct. These sentences that can't be translated instantly are called *donkey sentences*. For another example see page 310 of the book.

11.5 Ambiguity and context sensitivity

Sometimes a sentence can be ambigious (e.g. *Every minute a man is mugged*). In a case of ambiguity, the translation is *context sensitive*.

11.6 Translations using function symbols

Anything you can express using an n-ary function symbol can also be expressed using an n+1-ary relation symbol, but at a cost in terms of complexity of the sentences used.

11.7 Prenex form

A sentence is in *prenex form* if any quantifiers contained in it are out in front. Any sentence is logically equivalent to one in prenex form. To understand how to put a sentence in prenex form, you would have to read the pages 320-323 of the book.

12 Methods of Proof for Quantifiers

12.1 Valid quantifier steps

Universal elimination/instantiation means that we can infer S(c) from $\forall x S(x)$ so long as c denotes an object in the domain of discourse. From that S(c) we can also infer $\exists x S(x)$ and that is called *existential introduction/generalization.*

12.2 The method of existential instantiation

If $\exists x S(x)$ is the case, we can give a name to one of the objects that satisfies S(x) as long as that name is not in use. Let's call that object c. By doing that we can assume S(c). This is called *existential elimination.*

12.3 The method of general conditional proof

One of the most important methods of proof involves reasoning about an arbitrary object of a particular kind in order to prove a universal claim about all such object. This is called *general conditional proof.* We can achieve that proof by using *universal introduction/generalization.* For example:

$\forall x(Cube(x) \rightarrow Small(x))$
$\forall x Cube(x)$

$\forall x Small(x)$

12.4 Proofs involving mixed quantifiers

By using the methods and proofs of the earlier sections, you can do proofs involving mixed quantifiers. I would recommend reading the pages 338-341 of the book to understand these concepts fully, as the explanation is well done there.

13 Formal Proofs and Quantifiers

13.1 Universal quantifier rules

In the examples you are going to see, there is a *boxed constant* (e.g. $\boxed{c}$). You can think of boxed constants as "Let c denote an arbitrary object satisfying P(c)".

General conditional proof:

$\boxed{a} P(a)$

.
.
.
Q(a)

$\forall x(P(x) \rightarrow Q(x))$

Universal Introduction:

$\boxed{a}$

.
.
.
P(a)

$\forall x P(x)$

An example of these proofs and previous ones:

1.$\forall x(P(x) \rightarrow Q(x))$
2.$\forall z(Q(z) \rightarrow R(z))$

| 3.$\boxed{d}P(d)$
| 4.$P(d) \rightarrow Q(d)$
| 5.$Q(d)$
| 6.$Q(d) \rightarrow R(d)$
| 7.$R(d)$
8.$\forall x(P(x) \rightarrow R(x))$

13.2 Existential quantifier rules

Existential Introduction

$S(c)$
.
.
$\exists x S(x)$

Existential elimination

$\exists x S(x)$
.
.
| $\boxed{c}S(c)$
| .
| .
| .
| Q
Q

For a more complex example: page 358 LPL.

13.3 Strategy and tactics

Some tactics you can use while proving with universal and existential quantifiers are:

- Always be clear what the sentence means
- Find an informal proof and try to formalize this
- Working backwards can be very useful in proving universal claims
- Working backwards does not work well with existential claims
- If you get stuck: proof by contradiction

17 Advanced Topics in Propositional Logic

17.1 Truth assignments and truth tables

To model truth tables, we are going to use set theory from now on. To do this we set up a *truth assignment*, a function from the atomic sentences into the set {TRUE, FALSE}. It models a single row of a complete truth table for the language. Using truth assignments, we can define for example tautologies easier. For a sentence S and a truth assignment h, S is a tautology if every h has S coming out true.

17.3 Horn sentences

To understand what Horn sentences are we first need to know what *positive and negative literals* are. Literals are atomic sentences (positive) and their negations (negative). A sentence that is in conjunctive normal form (CNF) and every disjunction in this sentence having at most one positive literal is a *Horn sentence*. An example:
$\neg Home(claire) \wedge (\neg Home(max) \vee Happy(carl))$

Any Horn sentence is logically equivalent to a conjunction of conditional statements of the following three forms, where the A and B stand for ordinary atomic sentences:

1. $(A1 \wedge ... \wedge An) \rightarrow B$
2. $(A1 \wedge ... \wedge An) \rightarrow \bot$
3. $\top \rightarrow B$

Using these rules, we can perform an efficient method on a computer to check tt-satisfiability of Horn sentences, known as the *satisfaction algorithm for Horn sentences*.

18 Advanced Topics in FOL

18.1 First-order structures

The FOL-equivalent of truth assignments are called *first-order structures*. These structures can be used to design a world. When modeling a world, we need to think of a couple of things. The first one is the *domain of discourse* (D), in the case of a world, the objects of a world. Then we need to assign predicates to certain subsets, we call this *extensions of predicates*, like assigning *Cube* to the subset **Cu** or *Larger* to the subset **La**. An example of the last one can be written as:

$La = \{\langle b2, b1\rangle, \langle b3, b1\rangle, \langle b3, b2\rangle\}$

(In a world where b3 is the is largest object, then b2, and lastly b1.) Lastly, to model a world, we sometimes need to name an object. Say, for example, we want to name object b2 'c'. We say that b2 is the *referent* of the name c. To do this we mostly use a naming function, that assigns that object to that certain name. To represent this world with one object, we package all these parts of a world into the mathematical object $\mathfrak{M}$. This function $\mathfrak{M}$ is defined on the predicates of the language, the names of the language, and the quantifier symbol $\forall$. Such a function is called a first-order structure if:

1. $\mathfrak{M}(\forall)$ is a nonempty set D, called the domain of discourse of $\mathfrak{M}$
2. P is an n-ary predicate symbol of the language then $\mathfrak{M}(P)$ is a set of n-tuples x1,..., xn of elements of D. This set is called the extension of P in $\mathfrak{M}$. It is required that the extension of the identity symbol consist of all pairs $\langle x, x\rangle$, for $x \in D$.
3. c is any name of the language, then $\mathfrak{M}(c)$ is an element of D, and is called the referent of c in $\mathfrak{M}$.

18.2 Truth and satisfaction, revisited

This paragraph is about semantics and cannot be summarized here. LPL 516-522 and videos about this subject should help you understand this technique.

18.5 Skolemization

Given a sentence of the form $\forall x \exists y P(x, y)$ in some first-order language, we *Skolemize* it by choosing a function symbol f not in the language and writing $\forall x P(x, f(x))$. Every world that makes the *Skolemization* true also makes the original sentence true. Every world that makes the original sentence true can be turned into one that makes the Skolemization true by interpreting the function symbol f by a function f which picks out, for any object b in the domain, some object c such that they satisfy the wff P(x, y).