Handout 1

Logic is the study of reasoning. The British mathematician and philosopher George Boole (1815–1864) is the man who made logic mathematical. His book *The Mathematical Analysis of Logic* was published in 1847.

Logic can be used in programming, and it can be applied to the analysis and automation of reasoning about software and hardware. This is why it is sometimes considered a part of theoretical computer science. Since reasoning plays an important role in intelligent behavior, logic is closely related to artificial intelligence.

The short book by the German philosopher Gottlob Frege (1848–1925) with the long title *Ideography, a Formula Language, Modeled upon that of Arithmetic, for Pure Thought* (1879), introduced notation that is somewhat similar to what is now called first-order formulas. Frege wrote:

> I believe that I can best make the relation of my ideography to ordinary language clear if I compare it to that which the microscope has to the eye. Because of the range of its possible uses and the versatility with which it can adapt to the most diverse circumstances, the eye is far superior to the microscope... But, as soon as scientific goals demand great sharpness of resolution, the eye proves to be insufficient.
>
> ... I am confident that my ideography can be successfully used wherever special value must be placed on the validity of proofs, as for example when the foundations of the differential and integral calculus are established.

In logic, we distinguish between two languages: the one that is the object of our study and the one that we use to talk about that object. The former is called the *object language*; the latter is the *metalanguage*. In the first part of this course, the object language is the formal language of propositional formulas defined below. The metalanguage is the usual informal language of mathematics and theoretical computer science, which is a mixture of the English language and mathematical notation. The importance of distinguishing between the object language and the metalanguage was emphasized by the mathematician and logician Alfred Tarski (1902–1983), who taught logic at Berkeley since 1942.

Propositional Formulas: Syntax

A *propositional signature* is a set of symbols called *atoms*. (In examples, we will assume that p, q, r are atoms.) The symbols

$$\wedge \quad \vee \quad \rightarrow \quad \leftrightarrow \quad \neg \quad \bot \quad \top$$

are called *propositional connectives*. Among them, the symbols $\wedge$ *(conjunction)*, $\vee$ *(disjunction)*, $\rightarrow$ *(implication)* and $\leftrightarrow$ *(equivalence)* are called *2-place*, or *binary* connectives; $\neg$ *(negation)* is a *1-place*, or *unary* connective; $\bot$ *(false)* and $\top$ *(true)* are *0-place*.

Take a propositional signature σ which contains neither the propositional connectives nor the parentheses (,). The alphabet of propositional logic consists of the atoms from σ, the propositional connectives, and the parentheses. By a *string* we understand a finite string of symbols in this alphabet. We define when a string is a *(propositional) formula* recursively, as follows:

- every atom is a formula,
- both 0-place connectives are formulas,
- if F is a formula then $\neg F$ is a formula,
- for any binary connective $\odot$, if F and G are formulas then $(F \odot G)$ is a formula.

For instance,

$$\neg(p \rightarrow q)$$

and

$$(\neg p \rightarrow q) \tag{1}$$

are formulas; the string

$$\neg p \rightarrow q \tag{2}$$

is not a formula. But very soon (see next page) we are going to introduce a convention according to which (2) can be used as an abbreviation for (1).

Properties of formulas can be often proved by induction. One useful method is strong induction on length (number of symbols). In such a proof, the induction hypothesis is that every formula which is shorter than F has the property P that we want to prove. From this assumption we need to derive that F has property P also. Then it follows that all formulas have property P.

In another useful form of induction, we check that all atoms and 0-place connectives have property P, and that the property is preserved when a new formula is formed using a unary or binary connective. More precisely, we show that

- every atom has property P,
- both 0-place connectives have property P,
- if a formula F has property P then so does $\neg F$,
- for any binary connective $\odot$, if formulas F and G have property P then so does $(F \odot G)$.

Then we can conclude that property P holds for all formulas. This is called "structural induction."

Prove the following assertions.

1.1^c In any prefix of a formula, the number of left parentheses is greater than or equal to the number of right parentheses.

(A *prefix* of a string $a_1 \cdots a_n$ is any string of the form $a_1 \cdots a_m$ where $0 \leq m \leq n$.)

1.2^c Every prefix of a formula F

- is a string of negations (possibly empty), or
- has more left than right parentheses, or
- equals F.

1.3^c No formula can be represented in the form $(F \odot G)$, where F and G are formulas and $\odot$ is a binary connective, in more than one way.

By representing a formula in the form $\neg F$ or $(F \odot G)$ we start "parsing" it. The assertion of the previous problem shows that a formula can be parsed in only one way.

From now on, we will abbreviate formulas of the form $(F \odot G)$ by dropping the outermost parentheses in them. We will also agree that $\leftrightarrow$ has a lower binding power than the other binary connectives. For instance,

$$p \vee q \leftrightarrow p \rightarrow r$$

will be viewed as shorthand for

$$((p \vee q) \leftrightarrow (p \rightarrow r)).$$

Finally, for any formulas $F_1, F_2, \ldots, F_n$ $(n > 2)$,

$$F_1 \wedge F_2 \wedge \cdots \wedge F_n$$

will stand for

$$(\cdots(F_1 \wedge F_2) \wedge \cdots \wedge F_n).$$

The abbreviation $F_1 \vee F_2 \vee \cdots \vee F_n$ will be understood in a similar way.

Propositional Formulas: Semantics

The symbols f and t are called *truth values.* An *interpretation* of a propositional signature σ is a function from σ into $\{\mathsf{f}, \mathsf{t}\}$. If σ is finite then an interpretation can be defined by the table of its values, for instance:

p	q	r
f	f	t

(3)

The semantics of propositional formulas that we are going to introduce defines which truth value is assigned to a formula F by an interpretation I.

As a preliminary step, we need to associate functions with all unary and binary connectives: a function from $\{\mathsf{f}, \mathsf{t}\}$ into $\{\mathsf{f}, \mathsf{t}\}$ with the unary connective $\neg$, and a function from $\{\mathsf{f}, \mathsf{t}\} \times \{\mathsf{f}, \mathsf{t}\}$ into $\{\mathsf{f}, \mathsf{t}\}$ with each of the binary connectives. These functions are denoted by the same symbols as the corresponding connectives, and defined by the following tables:

x	$\neg(x)$
f	t
t	f

x	y	$\wedge(x, y)$	$\vee(x, y)$	$\rightarrow(x, y)$	$\leftrightarrow(x, y)$
f	f	f	f	t	t
f	t	f	t	t	f
t	f	f	t	f	f
t	t	t	t	t	t

For any formula F and any interpretation I, the truth value F^I that is *assigned* to F by I is defined recursively, as follows:

- for any atom F, $F^I = I(F)$,
- $\bot^I = \mathsf{f}$, $\top^I = \mathsf{t}$,
- $(\neg F)^I = \neg(F^I)$,
- $(F \odot G)^I = \odot(F^I, G^I)$ for every binary connective $\odot$.

If $F^I = \mathsf{t}$ then we say that the interpretation I *satisfies* F (symbolically, $I \models F$).

1.4 Find a formula F of the signature $\{p, q, r\}$ such that (3) is the only interpretation satisfying F.

1.5 For any formulas $F_1, \ldots, F_n$ $(n \geq 1)$ and any interpretation I,

$$(F_1 \wedge \cdots \wedge F_n)^I = \mathsf{t} \text{ iff } F_1^I = \cdots = F_n^I = \mathsf{t},$$
$$(F_1 \vee \cdots \vee F_n)^I = \mathsf{f} \text{ iff } F_1^I = \cdots = F_n^I = \mathsf{f}.$$

If the underlying signature is finite then the set of interpretations is finite also, and the values of F^I for all interpretations I can be represented by a finite table. This table is called the *truth table* of F. For instance, Problem 1.4 could be stated as follows: Find a formula F whose truth table is

p	q	r	F
f	f	f	f
f	f	t	t
f	t	f	f
f	t	t	f
t	f	f	f
t	f	t	f
t	t	f	f
t	t	t	f

In the following two problems, we assume that the underlying signature is finite: $\sigma = \{p_1, \ldots, p_n\}$.

1.6^c For any interpretation I, there exists a formula F such that I is the only interpretation satisfying F.

1.7^c For any function α from interpretations to truth values, there exists a formula F such that, for all interpretations I, $F^I = \alpha(I)$.

Tautologies

A propositional formula F is a *tautology* if every interpretation satisfies F.

1.8 Determine which of the following formulas are tautologies:

$$(p \to q) \vee (q \to p),$$
$$((p \to q) \to p) \to p,$$
$$(p \to (q \to r)) \to ((p \to q) \to (p \to r)).$$

Equivalent Formulas

A formula F is *equivalent* to a formula G (symbolically, $F \Leftrightarrow G$) if, for every interpretation I, $F^I = G^I$. In other words, $F \Leftrightarrow G$ means that formula $F \leftrightarrow G$ is a tautology.

1.9 (a) Conjunction and disjunction are associative:

$$(F \wedge G) \wedge H \Leftrightarrow F \wedge (G \wedge H),$$
$$(F \vee G) \vee H \Leftrightarrow F \vee (G \vee H).$$

Does equivalence have a similar property?

(b) Conjunction distributes over disjunction:

$$F \wedge (G \vee H) \Leftrightarrow (F \wedge G) \vee (F \wedge H);$$

disjunction distributes over conjunction:

$$F \vee (G \wedge H) \Leftrightarrow (F \vee G) \wedge (F \vee H).$$

Do these connectives distribute over equivalence?

(c) De Morgan's laws

$$\neg(F \wedge G) \Leftrightarrow \neg F \vee \neg G,$$
$$\neg(F \vee G) \Leftrightarrow \neg F \wedge \neg G$$

show how to transform a formula of the form $\neg(F \odot G)$ when $\odot$ is conjunction or disjunction. Find similar transformations for the cases when $\odot$ is implication or equivalence.

(d) Implication distributes over conjunction:

$$F \rightarrow (G \wedge H) \Leftrightarrow (F \rightarrow G) \wedge (F \rightarrow H).$$

Find a similar transformation for $(F \vee G) \rightarrow H$.

(e) To simplify a formula means to find an equivalent formula that is shorter. Simplify the following formulas:

(i) $F \leftrightarrow \neg F$,

(ii) $F \vee (F \wedge G)$,

(iii) $F \wedge (F \vee G)$,

(iv) $F \vee (\neg F \wedge G)$.

(v) $F \wedge (\neg F \vee G)$.

1.10^c (a) For each of the formulas

$$p \wedge q,\ p \vee q,\ p \leftrightarrow q,\ \neg p, \top$$

find an equivalent formula that contains no connectives other than $\rightarrow$ and $\bot$.
(b) For each of the formulas

$$p \rightarrow q,\ p \wedge q$$

find an equivalent formula that contains no connectives other than $\leftrightarrow$ and $\vee$.

Adequate Sets of Connectives

1.11^c For any formula, there exists an equivalent formula that contains no connectives other than $\rightarrow$ and $\bot$.

In this sense, $\{\rightarrow, \bot\}$ is an "adequate" set of connectives.

1.12 If the underlying signature is non-empty then each of the sets

$$\{\wedge, \neg\},\ \{\vee, \neg\},\ \{\rightarrow, \neg\}$$

is adequate.

1.13^c Any propositional formula equivalent to $\bot$ contains at least one of the connectives $\bot$, $\neg$.

This fact shows that the set $\{\wedge, \vee, \rightarrow, \leftrightarrow, \top\}$ is not adequate.

Disjunctive and Conjunctive Normal Forms

A *literal* is an atom or the negation of an atom. A *simple conjunction* is a formula of the form $L_1 \wedge \cdots \wedge L_n$ $(n \geq 1)$, where $L_1, \dots, L_n$ are literals. A formula is in *disjunctive normal form* if it has the form $C_1 \vee \cdots \vee C_m$ $(m \geq 1)$, where $C_1, \dots, C_m$ are simple conjunctions.

1.14^c If the underlying signature is non-empty then any formula is equivalent to a formula in disjunctive normal form.

A *simple disjunction* is a formula of the form $L_1 \vee \cdots \vee L_n$ $(n \geq 1)$, where $L_1, \dots, L_n$ are literals. A formula is in *conjunctive normal form* if it has the form $D_1 \wedge \cdots \wedge D_m$ $(m \geq 1)$, where $D_1, \dots, D_m$ are simple disjunctions.

1.15^c Let F be a formula in disjunctive normal form. Show that $\neg F$ is equivalent to a formula in conjunctive normal form.

1.16^c If the underlying signature is non-empty then any formula is equivalent to a formula in conjunctive normal form.

Satisfiability and Entailment

A set Γ of formulas is *satisfiable* if there exists an interpretation that satisfies all formulas in Γ, and *unsatisfiable* otherwise.

1.17 A set $\{F_1, \dots, F_n\}$ is unsatisfiable iff $\neg(F_1 \wedge \cdots \wedge F_n)$ is a tautology.

1.18^c Let Γ be a set of literals. Show that Γ is satisfiable iff there is no atom A for which both A and $\neg A$ belong to Γ.

For any atom A, the literals A, $\neg A$ are said to be *complementary* to each other. Thus the assertion of the last problem can be expressed as follows: A set of literals is satisfiable iff it does not contain complementary pairs.

A set Γ of formulas *entails* a formula F (symbolically, $\Gamma \models F$), if every interpretation that satisfies all formulas in Γ satisfies F also. Note that the notation for entailment uses the same symbol as the notation for satisfaction introduced earlier, the difference being that the expression on the left is an interpretation (I) in one case and a set of formulas (Γ) in the other. The formulas entailed by Γ are also called the *logical consequences* of Γ.

1.19 $F_1, \dots, F_n \models G$ iff $(F_1 \wedge \cdots \wedge F_n) \rightarrow G$ is a tautology.

(In the first of these expressions, we dropped the braces $\{\,\}$ around $F_1, \dots, F_n$.)

1.20 For any set Γ of formulas and any formula F, $\Gamma \models F$ iff the set $\Gamma \cup \{\neg F\}$ is unsatisfiable.