Mathematical Logic 2016

Lecture 3: Substitution theorems and Equivalences

Instructor: Ashutosh Gupta

TIFR, India

Compile date: 2016-08-12

Where are we and where are we going?

We have

- defined propositional logic syntax
- assigned meanings to the formulas
- defined the decision problem for the logic
- developed truth tables as a method to decide a formula

We will

- prove substitution theorems
- review some common equivalences

Topic 3.1

Substitution theorems

Substitution theorem

Theorem 3.1

Let $F(p)$, G, and H be formulas. For some model m,

$$\text{if} \quad m \models G \text{ iff } m \models H \quad \text{then} \quad m \models F(G) \text{ iff } m \models F(H)$$

Proof.

Assume $m \models G$ iff $m \models H$. We prove the theorem using structural induction over F.

base case:

$F(p)$ is atomic.

If $F(p) = p$, then $F(G) = G$ and $F(H) = H$. Therefore, hyp holds.

If $F(p) \neq p$, then $F(p) = F(G) = F(H)$. Again, hyp holds.

induction step:

Suppose $F(p) = F_1(p) \circ F_2(q)$ for some binary connective $\circ$.

Due to ind. hyp., $m \models F_1(G)$ iff $m \models F_1(H)$, and $m \models F_2(G)$ iff $m \models F_2(H)$

Due to the semantics of propositional logic,

$m \models F_1(G) \circ F_2(G)$ iff $m \models F_1(H) \circ F_2(H)$. Therefore, hyp holds.

The negation case is symmetric. □

Equivalence generalization theorem

Theorem 3.2
If $F(p) \equiv G(p)$ then for each formula H, $F(H) \equiv G(H)$.

Proof.
Wlog, we assume p does not appear in H.(why?)
Assume $m \models F(H)$. Let b be 1 if $m \models H$ otherwise 0. Let $m' = m[p \mapsto b]$.
Due to Thm 3.1, $m' \models F(p)$. Therefore, $m' \models G(p)$.
Due to Thm 3.1, $m' \models G(H)$. Since $p \not\in \mathbf{Vars}(G(H))$, $m \models G(H)$. □

The above theorem allows us to first prove equivalences between formula over variables then use it for arbitrary formulas.
We will state equivalences using variables instead generic formulas.

Example 3.1
Since $\neg\neg p \equiv p$, we can say $\neg\neg(q \oplus r) \equiv (q \oplus r)$

Exercise 3.1
Extend the argument for simultaneous substitutions.

Subformula Replacement Theorem

Theorem 3.3

Let G,H and $F(p)$ be formulas. If $G \equiv H$ then $F(G) \equiv F(H)$.

Proof.

Due to Thm 3.1, straight forward. □

The above theorem allows us to use known equivalences to modify formulas.

Example 3.2

Since we know $\neg\neg(q \oplus r) \equiv (q \oplus r)$,

$(\neg\neg(q \oplus r) \Rightarrow (r \wedge q)) \equiv ((q \oplus r) \Rightarrow (r \wedge q))$

Exercise 3.2

a. Complete the arguments in the above proof.

b. extend the argument for simultaneous substitutions.

Topic 3.2

Equivalences

Constant connectives

- $\neg\top \equiv \bot$
- $\top \wedge p \equiv p$
- $\top \vee p \equiv \top$
- $\top \oplus p \equiv \neg p$
- $\top \Rightarrow p \equiv p$
- $p \Rightarrow \top \equiv \top$
- $\top \Leftrightarrow p \equiv p$
- $\neg\bot \equiv \top$
- $\bot \wedge p \equiv \bot$
- $\bot \vee p \equiv p$
- $\bot \oplus p \equiv p$
- $\bot \Rightarrow p \equiv \top$
- $p \Rightarrow \bot \equiv \neg p$
- $\bot \Leftrightarrow p \equiv \neg p$

Exercise 3.3

Simplify, the following formulas using the above equivalences

- $\top \Rightarrow \bot$
- $(\top \oplus \top) \oplus \top$
- $p \Rightarrow (\bot \Rightarrow q)$

Negation and other connectives

- $\neg\neg p \equiv p$
- $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- $\neg(p \Rightarrow q) \equiv \neg q \Rightarrow \neg p$
- $\neg(p \oplus q) \equiv \neg p \oplus q \equiv p \Leftrightarrow q$
- $\neg(p \Leftrightarrow q) \equiv p \oplus q$

Associativity

$\wedge$, $\vee$, $\oplus$ are associative

- $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
- $p \vee (q \vee r) \equiv (p \vee q) \vee r$
- $p \oplus (q \oplus r) \equiv (p \oplus q) \oplus r$

Due to associativity, we do not need parenthesis in the following formulas

- $p_1 \wedge \cdots \wedge p_k$
- $p_1 \vee \cdots \vee p_k$
- $p_1 \oplus \ldots \oplus p_k$

The drop of parenthesis is sometimes called flattening.

Exercise 3.4

Prove/Disprove $\Leftrightarrow$ is associative.

Commutativity

$\wedge$, $\vee$, $\oplus$, $\Leftrightarrow$ are commutative

- $(p \wedge q) \equiv (q \wedge p)$
- $(p \vee q) \equiv (q \vee p)$
- $(p \oplus q) \equiv (q \oplus p)$
- $(p \Leftrightarrow q) \equiv (q \Leftrightarrow p)$

Absorption law

- $p \land p \Leftrightarrow p$
- $p \lor p \Leftrightarrow p$

Due to associativity, commutativity and absorption law, we define the following notation with a clear meaning

- $\bigwedge\{p_1, \dots, p_k\} \triangleq p_1 \land \dots \land p_k$
- $\bigvee\{p_1, \dots, p_k\} \triangleq p_1 \lor \dots \lor p_k$

Distributivity

$\wedge$, $\vee$ distribute over each other

- $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Exercise 3.5

Prove/Disprove $p \oplus (q \wedge r) \equiv (p \oplus q) \wedge (p \oplus r)$

Properties of $\oplus$

- $\top \oplus p \equiv \neg p$
- $\bot \oplus p \equiv p$
- $p \oplus p \equiv \bot$
- $p \oplus \neg p \equiv \top$

Simplify

- All tools include a simplify procedure using the presented equivalences
- $\oplus$ and $\Leftrightarrow$ are difficult connectives, because they result in larger formula if one aims to remove them. We will learn soon how to deal with the operators.

Topic 3.3

Expressive power of propositional logic

Boolean functions

A finite boolean function is $\mathcal{B}^n \rightarrow \mathcal{B}$

A formula $F(p_1, \ldots, p_n)$ with $\textbf{Vars}(F) = \{p_1, \ldots, p_n\}$ can be viewed as boolean function f that is defined as follows.

$$\text{for each model } m, f(m(p_1), \ldots, m(p_n)) = m(F)$$

We say F represents f.

Expressive power

Theorem 3.4

For each finite boolean function f, there is a formula F that represents f.

Proof.

Let $f : \mathcal{B}^n \rightarrow \mathcal{B}$. We construct a formula $F(p_1, \dots, p_n)$ to represent f.
Let $p_i^0 \triangleq \neg p_i$ and $p_i^1 \triangleq p_i$.
Let $(b_1, \dots, b_n) \in \mathcal{B}^n$. Let

$$F_{(b_1,\dots,b_n)} = \begin{cases} (p_1^{b_1} \wedge \cdots \wedge p_n^{b_n}) & \text{if } f(b_1, \dots, b_n) = 1 \\ \bot & \text{otherwise} \end{cases}$$

$F \triangleq F_{(0,\dots,0)} \vee \cdots \vee F_{(1,\dots,1)}$ □

Exercise 3.6

a. Show $F(p_1, \dots, p_n)$ represents f.

b. Show $\wedge$ alone can not express all boolean functions

Minimal logical connectives

We used

- 2 0-ary,
- 1 unary, and
- 5 binary

connectives to describe the propositional logic.

However, it is not the minimal set needed for the maximum expressivity.

Example 3.3

$\neg$ and $\vee$ can define the whole propositional logic.

- $\top \equiv p \vee \neg p$ *for some $p \in$* **Vars**
- $\bot \equiv \neg\top$
- $(p \wedge q) \equiv \neg(\neg p \vee \neg q)$
- $(p \oplus q) \equiv (p \wedge \neg q) \vee (\neg p \wedge q)$
- $(p \Rightarrow q) \equiv (\neg p \vee q)$
- $(p \Leftrightarrow q) \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$

Exercise 3.7

a. Show $\neg$ and $\wedge$ can define all other connectives

b. Show $\oplus$ alone can not define $\neg$

Universal connective

Let $\overline{\wedge}$ be a binary connective with the following truth table

$m(F)$	$m(G)$	$m(F \overline{\wedge} G)$
0	0	1
0	1	1
1	0	1
1	1	0

Exercise 3.8

a. Show $\overline{\wedge}$ can define all other connectives

b. Are there other universal connectives?

Topic 3.4

Negation normal form

Negation normal form(NNF)

Definition 3.1
A formula is in NNF if $\neg$ appears only in front of the propositional variables.

Theorem 3.5
For every formula F there is another formula F' in NNF s.t. $F \equiv F'$.

Proof.
Recall from the tautologies that we can always push negation inside the operators. □

- Often we assume that the formulas are in negation normal form.
- However, there are negations hidden inside $\oplus$, $\Rightarrow$, and $\Leftrightarrow$. In practice, these symbols are also expected to be removed while producing NNF

Exercise 3.9
Write an efficient algorithm to covert a propositional formula to NNF?

Example :NNF

Example 3.4

Consider $\neg(((p \vee \neg s) \oplus r) \Rightarrow q)$

$\equiv \neg q \Rightarrow \neg((p \vee \neg s) \oplus r)$

$\equiv \neg q \Rightarrow \neg(p \vee \neg s) \oplus r$

$\equiv \neg q \Rightarrow (\neg p \wedge \neg\neg s) \oplus r$

$\equiv \neg q \Rightarrow (\neg p \wedge s) \oplus r$

Exercise 3.10

Convert the following formulas into NNF

- $\neg(p \Rightarrow q)$
- $\neg(\neg((s \Rightarrow \neg(p \Leftrightarrow q))) \oplus (\neg q \vee r))$

Exercise 3.11

Are there any added difficulties if the formula is given as a DAG not as a tree?

Topic 3.5

Exercises

Simplifications

Exercise 3.12

Simplify

$$\underbrace{p \oplus \ldots \oplus p}_{n} \oplus \underbrace{\neg p \oplus \ldots \oplus \neg p}_{k} \equiv ?$$

Exercise 3.13

Simplify

$$(p \vee (p \oplus y)) \Rightarrow (p \wedge q) \wedge (r \wedge \neg p)$$

Encoding if-then-else

Exercise 3.14

Show the following two encodings of $ite(p, q, r)$ *equivalent.*

1. $(p \wedge q) \vee (\neg p \wedge q)$
2. $(p \Rightarrow q) \wedge (\neg p \Rightarrow q)$

Expressive power

Exercise 3.15
Show $\neg$ and $\oplus$ is not as expressive as propositional logic.

Exercise 3.16
Prove/disprove:
if-then-else is fully expressive

End of Lecture 3