CS 433 Automated Reasoning 2025

Lecture 2: Substitution and equivalences in propositional logic

Instructor: Ashutosh Gupta

IITB India

Compile date: 2025-01-10

Simplifications for formulas

If we wish to develop algorithms for proof generation, we need more structure in our input.

For example, we simplify equations like $2x + 3 = 1 - x$, before solving them.

We will develop methods for simplifications or turning into normal forms.

Topic 2.1

Substitution theorems

Substitutions

Substitution is an important operation in logic.

Intuitively, we should be able to substitute equivalent subformulas without altering the truth values of formulas.

However, we need a proof to enable us.

In the following, we will prove three theorems.

Substitution theorem

Theorem 2.1

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

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

Proof.

Assume $m \models G$ iff $m \models H$.

We prove the theorem using structural induction over the structure of 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.

...

Substitution theorem (contd.)

Proof(contd.)

Induction step:

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

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

Due to the semantics of the propositional logic, $m \models F_1(G) \circ F_2(G)$ iff $m \models F_1(H) \circ F_2(H)$.

Therefore, $m \models F(G)$ iff $m \models F(H)$.

The negation case is similar. $\square$

Equivalence generalization theorem

Theorem 2.2

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?)

Commentary: If p occurs in H, we split the substitution in two steps. For a fresh q, we first substitute from p to $H[q/p]$ and subsequently q to p. Check if this trick works.

Consider a model m. Let $m' \triangleq \begin{cases} m[p \mapsto 1] & \text{if } m \models H \\ m[p \mapsto 0] & \text{if } m \not\models H. \end{cases}$

Due to the construction of m',

$$m' \models p \text{ iff } m' \models H.\text{(Why?)} \qquad (*)$$

Now we will show that $m \models F(H)$ iff $m \models G(H)$. ...

Commentary: $p \notin \textbf{Vars}(F(H))$ has a subtle assumption. Did $F(H)$ really replace all ps? If we define $F(z) = (z \lor p)$ and use in our theorem for $F(p)$. Then $F(r \lor q)$ has a p. There is a defining step for substitutions like $F(p)$ and it is subsequently used. In our case, the definition is appearing in the theorem.

Equivalence generalization theorem(contd.)

Proof(Contd.)

$$\begin{tikzcd}
m \models F(H) \arrow[d, Leftrightarrow, "p \notin \textbf{Vars}(F(H))"] & m \models G(H) \arrow[d, Leftrightarrow, "p \notin \textbf{Vars}(G(H))"] \\
m' \models F(H) \arrow[d, Leftrightarrow, "\text{Thm 2.1 and *}"] & m' \models G(H) \arrow[d, Leftrightarrow, "\text{Thm 2.1 and *}"] \\
m' \models F(p) \arrow[r, Leftrightarrow, "\text{Due to lhs of this theorem } F(p) \equiv G(p)"'] & m' \models G(p)
\end{tikzcd}$$

Therefore, $m \models F(H)$ iff $m \models G(H)$. Therefore, $F(H) \equiv G(H)$. □

Exercise 2.1

Extend the above argument for simultaneous substitutions

Writing equivalences

The previous theorem allows us to first prove equivalences between formulas over variables then use it for arbitrary formulas.

We will state equivalences using variables instead of generic formulas.

Example 2.1

Since $\neg\neg p \equiv p$*, we can deduce* $\neg\neg(q \oplus r) \equiv (q \oplus r)$

Commentary: Let us prove the previous theorem for simultaneous substitution: If $F(p,q) \equiv G(p,q)$ then for each formulae H_1 and H_2, $F(H_1, H_2) \equiv G(H_1, H_2)$.
Proof. Wlog, we assume p and q do not appear in either H_1 or H_2 Consider model m.

$$\text{Let } m' = \begin{cases} m[p \rightarrow 1, q \rightarrow 1] & \textit{if } m \models H_1 \textit{ and } m \models H_2 \\ m[p \rightarrow 0, q \rightarrow 1] & \textit{if } m \not\models H_1 \textit{ and } m \models H_2 \\ m[p \rightarrow 1, q \rightarrow 0] & \textit{if } m \models H_1 \textit{ and } m \not\models H_2 \\ m[p \rightarrow 0, q \rightarrow 0] & \textit{if } m \not\models H_1 \textit{ and } m \not\models H_2 \end{cases}$$

Due to the construction of m': $m' \models p \textit{ iff } m' \models H_1 \ldots(1)$ and $m' \models q \textit{ iff } m' \models H_2 \ldots(2)$. We the same procedure as given in the previous slide, except for first using Theorem 6.1 and (1) to substitute H_1 by p and then using Theorem 6.1 and (2) to substitute H_2 by q. H_1 before H_2 because even if H_2 is a sub-formula of H_1, substitution will not create problems.

Subformula replacement theorem

Theorem 2.3

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

Proof.

Due to Thm 2.1, trivial. $\square$

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

Example 2.2

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

Exercise 2.2

a. Complete the arguments in the above proof.

b. extend the argument for simultaneous substitutions.

Commentary: We had proven theorem 2.3 in the previous lecture using derivation rules. Now we have proven the theorems 2.1- 2.3 again using semantics instead of the derivation rules. There is nothing wrong in doing this. Can we prove theorem 2.2 using derivation rules?

Topic 2.2

Equivalences

Equivalences

- Let us go over a list of useful and easy equivalences for simplification of formulas
- We need to prove their correctness using truth tables. However, we will not present the truth tables in the slides in this lecture.

Commentary: We have seen a few truth tables in the earlier lectures illustrating equivalences. In the exams, you will be expected to illustrate the equivalences using truth tables.

Constant connectives

- $\neg\top \equiv \bot$
- $\top \land p \equiv p$
- $\top \lor p \equiv \top$
- $\top \oplus p \equiv \neg p$
- $\top \Rightarrow p \equiv p$
- $p \Rightarrow \top \equiv \top$
- $\top \Leftrightarrow p \equiv p$
- $\neg\bot \equiv \top$
- $\bot \land p \equiv \bot$
- $\bot \lor p \equiv p$
- $\bot \oplus p \equiv p$
- $\bot \Rightarrow p \equiv \top$
- $p \Rightarrow \bot \equiv \neg p$
- $\bot \Leftrightarrow p \equiv \neg p$

Exercise 2.3

Simplify, the following formulas using the above equivalences

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

Exercise 2.4

Prove $\neg\top \equiv \bot$. *Hint: use semantics.*

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 parentheses in the following formulas

- $p_1 \wedge \dots \wedge p_k = \bigwedge_{i=1}^{k} p_i$
- $p_1 \vee \dots \vee p_k = \bigvee_{i=1}^{k} p_i$
- $p_1 \oplus \ldots \oplus p_k = \bigoplus_{i=1}^{k} p_i$

The drop of parentheses is called flattening.

Exercise 2.5

a. Prove/Disprove $\Leftrightarrow$ is associative. b. Write an inductive proof of flattening.

Commutativity and Absorption law

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

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

$\wedge$ and $\vee$ absorb multiple occurrences

- $p \wedge p \equiv p$
- $p \vee p \equiv p$

Due to associativity, commutativity and absorption law, we define the following notation

- $\bigwedge\{p_1, \dots, p_k\} \triangleq p_1 \wedge \dots \wedge p_k$
- $\bigvee\{p_1, \dots, p_k\} \triangleq p_1 \vee \dots \vee p_k$

Distributivity

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

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

Properties of $\oplus$

- $\top \oplus p \equiv \neg p$
- $\bot \oplus p \equiv p$
- $p \oplus p \equiv \bot$
- $p \oplus \neg p \equiv \top$
- $(p \oplus q) \equiv (p \lor q) \land (\neg p \lor \neg q)$
- $(p \Leftrightarrow q) \equiv (p \lor \neg q) \land (q \lor \neg p)$

Topic 2.3

Negation normal form

Negation and the other connectives

- $\neg\neg p \equiv p$
- $\neg(p \lor q) \equiv \neg p \land \neg q$ (DeMorgan's Law)
- $\neg(p \land q) \equiv \neg p \lor \neg q$ (DeMorgan's Law)
- $\neg(p \Rightarrow q) \equiv p \land \neg q$
- $\neg(p \oplus q) \equiv \neg p \oplus q \equiv p \Leftrightarrow q$
- $\neg(p \Leftrightarrow q) \equiv p \oplus q$

Exercise 2.6

Show that the above equivalences are derivable. For example, $\emptyset \vdash \neg(p \land q) \Leftrightarrow \neg p \lor \neg q$

Expanded DeMorgan

Theorem 2.4

$$\neg(\bigvee_{i=0}^{m} p_i) \equiv \bigwedge_{i=0}^{m} \neg p_i$$

Proof.

We prove it by induction over m.

Base case:

If $m = 0$, there is nothing to prove because both sides are same.

Induction step:

Let us assume $\neg(\bigvee_{i=0}^{m} p_i) \equiv \bigwedge_{i=0}^{m} \neg p_i$

Now consider

$$\neg(\bigvee_{i=0}^{m+1} p_i) \equiv \neg(\bigvee_{i=0}^{m} p_i \vee p_{m+1}) \underbrace{\equiv \neg \bigvee_{i=0}^{m} p_i \wedge \neg p_{m+1}}_{\text{Binary DeMorgan Rule}} \underbrace{\equiv \bigwedge_{i=0}^{m} \neg p_i \wedge \neg p_{m+1}}_{\text{Subsitution theorem}} \equiv \bigwedge_{i=0}^{m+1} \neg p_i \qquad \square$$

Negation normal form(NNF)

Definition 2.1

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

Theorem 2.5

For every formula F, there is a formula F' in NNF such that $F \equiv F'$.

Proof.

Due to the standard equivalences, we can always push $\neg$ under the logical connectives. $\square$

Example 2.3

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

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

- There are negations hidden inside $\oplus$, $\Rightarrow$, and $\Leftrightarrow$.
  Sometimes users want to also remove the symbols, while producing NNF.

Often we assume that the formulas are in NNF.

Exercise : NNF

Exercise 2.7

Convert the following formulas into NNF

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

Exercise 2.8

In the above exercises, remove $\Rightarrow$, $\Leftrightarrow$, and $\oplus$ before turning the above into NNF.

Exercise 2.9

*Let us suppose we have access to the parse tree of a formula, which is represented as a directed acyclic graph (DAG) (*not as a tree*). Write an algorithm that produces negation normal form (NNF) of the formula in linear time in the size of the DAG. You may assume the costs of reading from and writing to a map data structure are constant time.*

Formal derivation for NNF

Theorem 2.6

Let F' be the NNF of F. If we have $\Sigma \vdash F$, then we can derive $\Sigma \vdash F'$.

Proof.

We combine the following pieces of proofs for each step of the transformation.

- Derivations for substitutions.
- Derivations for pushing negations inside connectives.

Therefore, we have the derivations. $\square$

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 2.4

Problems

Extended distributivity

Exercise 2.10

Using induction and the distributivity property, show the following

$$\bigvee_{i=0}^{m} \bigwedge_{j=0}^{n_i} p_{ij} \equiv \bigwedge_{j_0=0}^{n_0} \dots \bigwedge_{j_m=0}^{n_m} \bigvee_{i=0}^{m} p_{ij_i}$$

Simplifications

Exercise 2.11

Show $p_1 \oplus \ldots \oplus p_n$ count odd number of one's in $p_1, .., p_n$.

Exercise 2.12

Similar to the above problem characterize the following.

$$\underbrace{p_1 \Leftrightarrow \ldots \Leftrightarrow p_n}_{n}$$

Exercise 2.13

Simplify

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

Exercise 2.14

Simplify

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

Encoding if-then-else

Some propositional logic may also include a ternary operator $ite(p, q, r)$, which encodes that if p is true then q is true, otherwise r is true.

Exercise 2.15

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

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

Simplify

Exercise 2.16

Let $G(x)$ be a formula. Show that the following equivalences hold.

- $F \lor G(F) \equiv F \lor G(\bot)$
- $F \land G(F) \equiv F \land G(\top)$
- $F \Rightarrow G(F) \equiv F \Rightarrow G(\top)$

Monotonic NNF

Definition 2.2

Let $pos(m, F)$ be the set of literals that are true in m and appear in F.

Example 2.4

$pos(\neg p_2 \land (p_1 \lor p_2), \{p_1 \mapsto 1, p_2 \mapsto 0\}) = \{p_1, \neg p_2\}$

Exercise 2.17

Let F be in NNF and does not contain $\oplus$, $\Rightarrow$, and $\Leftrightarrow$. Show that if $m \models F$ and $pos(m, F) \subseteq pos(m', F)$ then $m' \models F$.

Exercise: prove distributivity

Exercise 2.18

Prove/Disprove the following equivalences

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

End of Lecture 2