CS4004/CS4504: FORMAL VERIFICATION

Lecture 7: Propositional Logic

Vasileios Koutavas

School of Computer Science and Statistics
Trinity College Dublin

We are working with natural deduction proofs $A_1 \dots A_n \vdash B$ in propositional logic. Deduction rules so far:

- **Conjunction**: $\dfrac{A_1 \quad A_2}{A_1 \wedge A_2} \wedge i \qquad \dfrac{A_1 \wedge A_2}{A_1} \wedge e_1 \qquad \dfrac{A_1 \wedge A_2}{A_2} \wedge e_2$
- **Disjunction**: $\dfrac{A_1}{A_1 \vee A_2} \vee i_1 \qquad \dfrac{A_2}{A_1 \vee A_2} \vee i_2 \qquad \dfrac{A_1 \vee A_2 \quad \boxed{\begin{array}{c} A_1 \\ \dots \\ B \end{array}} \quad \boxed{\begin{array}{c} A_2 \\ \dots \\ B \end{array}}}{B} \vee e$
- **Implication**: $\dfrac{\boxed{\begin{array}{c} A \\ \dots \\ B \end{array}}}{A \to B} \to i \qquad \dfrac{A \quad A \to B}{B} \to e \qquad \dfrac{A_1 \to A_2 \quad \neg A_2}{\neg A_1} \mathrm{MT}$
- **Negation**: $\dfrac{A \quad \neg A}{\perp} \neg e \qquad \dfrac{\boxed{\begin{array}{c} A \\ \dots \\ \perp \end{array}}}{\neg A} \neg i \qquad \dfrac{\perp}{A} \perp e$

Show: $\neg A \vee \neg B \dashv\vdash \neg(A \wedge B)$ (De Morgan)

Show: $\neg A \lor \neg B \dashv\vdash \neg(A \land B)$ (De Morgan)

We can prove the left-to-right direction, but we cannot prove $\neg(A \land B) \vdash \neg A \lor \neg B$

We are missing a last set of rules.

Double negation

We know that $\mathrm{sem}(\neg\neg A) = \mathrm{sem}(A)$, for any A. That is, $\neg\neg A \equiv A$.

Can you derive the following rule?

$$\frac{\neg\neg A}{A}\ \neg\neg e$$

$$\frac{A \qquad \neg A}{\bot}\ \neg e \qquad \frac{\boxed{\begin{array}{c} A \\ \cdots \\ \bot \end{array}}}{\neg A}\ \neg i \qquad \frac{\bot}{A}\ \bot e$$

DOUBLE NEGATION

We know that $\mathrm{sem}(\neg\neg A) = \mathrm{sem}(A)$, for any A. That is, $\neg\neg A \equiv A$.

Can you derive the following rule?

$$\frac{\neg\neg A}{A}\ \neg\neg e$$

It turns out the above rule is not derivable and we need to add it as an axiom to the logic. But we can derive the following with the "standard" rules.

$$\frac{A}{\neg\neg A}\ \neg\neg i$$

If our logic does **not** include $\neg\neg e$ then it is called intuitionistic logic.

If our logic does include $\neg\neg e$ then it is called classical logic.

Show: $(\neg A \to \bot) \vdash A$

$$\frac{A \qquad \neg A}{\bot}\ \neg e \qquad \frac{\boxed{\begin{array}{c} A \\ \cdots \\ \bot \end{array}}}{\neg A}\ \neg i \qquad \frac{\bot}{A}\ \bot e \qquad \frac{A \qquad A \to B}{B} \to e \qquad \frac{\neg\neg A}{A}\ \neg\neg e$$

Show: $(\neg A \to \bot) \vdash A$ Proof by Contradiction (PBC)

$$\frac{A \qquad \neg A}{\bot}\ \neg e \qquad \frac{\boxed{\begin{array}{c} A \\ \cdots \\ \bot \end{array}}}{\neg A}\ \neg i \qquad \frac{\bot}{A}\ \bot e \qquad \frac{A \qquad A \to B}{B} \to e \qquad \frac{\neg\neg A}{A}\ \neg\neg e$$

Show: $\neg(A \wedge B) \vdash \neg A \vee \neg B$ De Morgan

Basic Propositional Logic Rules:

$$\frac{A_1 \qquad A_2}{A_1 \wedge A_2} \wedge i \qquad\qquad \frac{A_1 \wedge A_2}{A_1} \wedge e_1 \qquad\qquad \frac{A_1 \wedge A_2}{A_2} \wedge e_2$$

$$\frac{A_1}{A_1 \vee A_2} \vee i_1 \qquad\qquad \frac{A_2}{A_1 \vee A_2} \vee i_2 \qquad\qquad \frac{A_1 \vee A_2 \qquad \boxed{\begin{matrix} A_1 \\ \cdots \\ B \end{matrix}} \qquad \boxed{\begin{matrix} A_2 \\ \cdots \\ B \end{matrix}}}{B} \vee e$$

$$\frac{\boxed{\begin{matrix} A \\ \cdots \\ B \end{matrix}}}{A \to B} \to i \qquad\qquad \frac{A \qquad A \to B}{B} \to e$$

$$\frac{A \qquad \neg A}{\bot} \neg e \qquad\qquad \frac{\boxed{\begin{matrix} A \\ \cdots \\ \bot \end{matrix}}}{\neg A} \neg i \qquad\qquad \frac{\bot}{A} \bot e \qquad\qquad \frac{\neg\neg A}{A} \neg\neg e$$

$\vdash A \vee \neg A$ Law of Excluded Middle (LEM)

$(\neg A \to \bot) \vdash A$ PBC

$\neg(A_1 \wedge A_2) \dashv\vdash \neg A_1 \vee \neg A_2$ DeMorgan 1

$A \to (B_1 \vee B_2) \vdash (A \to B_1) \vee B_2$

$A \to B \vdash \neg A \vee B$ Material Implication

Note: $\neg(A_1 \vee A_2) \dashv\vdash \neg A_1 \wedge \neg A_2$ (DeMorgan 2)

does not require a classical proof

All Propositional Logic Rules

Basic Propositional Logic Rules:

$$\frac{A_1 \qquad A_2}{A_1 \wedge A_2}\ \wedge i \qquad \frac{A_1 \wedge A_2}{A_1}\ \wedge e_1 \qquad \frac{A_1 \wedge A_2}{A_2}\ \wedge e_2$$

$$\frac{A_1}{A_1 \vee A_2}\ \vee i_1 \qquad \frac{A_2}{A_1 \vee A_2}\ \vee i_2 \qquad \frac{A_1 \vee A_2 \qquad \boxed{\begin{array}{c} A_1 \\ \cdots \\ B \end{array}} \qquad \boxed{\begin{array}{c} A_2 \\ \cdots \\ B \end{array}}}{B}\ \vee e$$

$$\frac{\boxed{\begin{array}{c} A \\ \cdots \\ B \end{array}}}{A \to B}\ \to i \qquad \frac{A \qquad A \to B}{B}\ \to e$$

$$\frac{A \qquad \neg A}{\bot}\ \neg e \qquad \frac{\boxed{\begin{array}{c} A \\ \cdots \\ \bot \end{array}}}{\neg A}\ \neg i \qquad \frac{\bot}{A}\ \bot e \qquad \frac{\neg\neg A}{A}\ \neg\neg e$$

Derived Propositional Logic Rules:[1]

$$\frac{A}{A}\ \text{COPY} \qquad \frac{A}{\neg\neg A}\ \neg\neg i \qquad \frac{A_1 \rightarrow A_2 \qquad \neg A_2}{\neg A_1}\ \text{MT}$$

$$\frac{\boxed{\begin{array}{c}\neg A\\ \cdots\\ \bot\end{array}}}{A}\ \text{PBC}(\text{proof by contradiction}) \qquad \frac{}{A \vee \neg A}\ \text{LEM}(\text{law of excluded middle})$$

[1] Prove their validity.

Lecture 7, Part 2:
Meta-theory of propositional logic

- gives us a syntax to write logical propositions:

$$A ::= p \mid (\neg A) \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A)$$

- gives us a method for syntactically proving logical entailment
  $A_1, \ldots, A_n \vdash B$
  by applying natural deduction inference rules

  e.g., disjunction:

$$\frac{A_1}{A_1 \vee A_2} \vee i_1 \qquad \frac{A_2}{A_1 \vee A_2} \vee i_2 \qquad \frac{A_1 \vee A_2 \quad \boxed{\begin{array}{c} A_1 \\ \cdots \\ B \end{array}} \quad \boxed{\begin{array}{c} A_2 \\ \cdots \\ B \end{array}}}{B} \vee e$$

- The semantics of the logic interpret formulas as functions (truth tables) and give us a way to find equivalent formulas (even with different truth tables):

$$A_1 \wedge A_2 \rightarrow A_1 \equiv r \vee \neg r \quad \text{means}$$
$$A_1 \wedge A_2 \rightarrow A_1 \models r \vee \neg r \quad \text{and} \quad r \vee \neg r \models A_1 \wedge A_2 \rightarrow A_1$$

Q: Is every provable statement $A_1, \ldots, A_n \vdash B$ valid according to the semantics of the logic?

In other words is the proof system sound?

Theorem (Soudness of proof rules)

For any provable statement $A_1, \ldots, A_n \vdash B$ it is valid that $A_1, \ldots, A_n \models B$.

Q: Is every provable statement $A_1, \ldots, A_n \vdash B$ valid according to the semantics of the logic?

In other words is the proof system sound?

Theorem (Soudness of proof rules)

For any provable statement $A_1, \ldots, A_n \vdash B$ it is valid that $A_1, \ldots, A_n \models B$.

Proof by a form of induction. We will learn more about inductive proofs in the following weeks.

*Soundness is very important: we can't derive something false from the proof system.

Q: Do we have enough proof rules so that any valid $A_1, \ldots, A_n \models B$ we can be proved syntactically as $A_1, \ldots, A_n \vdash B$?

In other words is the proof system complete?

Theorem (Completeness of proof rules)

For any valid sequent $A_1, \ldots, A_n \models B$ it is provable that $A_1, \ldots, A_n \vdash B$.

Proof: see book 1.4.4.

*Completeness means we can prove any valid propositional logic theorem, using only the syntactic proof system. This is a very strong statement, not true for many other logics.

Q: Do we have enough proof rules so that any valid $A_1, \ldots, A_n \models B$ we can be proved syntactically as $A_1, \ldots, A_n \vdash B$?

In other words is the proof system complete?

Theorem (Completeness of proof rules)

For any valid sequent $A_1, \ldots, A_n \models B$ it is provable that $A_1, \ldots, A_n \vdash B$.

Proof: see book 1.4.4.

*Completeness means we can prove any valid propositional logic theorem, using only the syntactic proof system. This is a very strong statement, not true for many other logics.

**Natural deduction is not the only sound and complete system for doing propositional proofs. Exercise 1.2.6 in the book shows another: the sequent calculus, a system of rules to transform valid sequents to other valid sequents.

Q: is it possible to write an algorithm that **decides** whether $A_1, \ldots, A_n \vdash B$ is a valid sequent?

Q: is it possible to write an algorithm that **decides** whether $A_1, \ldots, A_n \vdash B$ is a valid sequent?

We only need an algorithm to decide whether $\models A$:

$A_1, \ldots, A_n \vdash B$ by a theorem, is equivalent to
$\vdash A_1 \to \ldots \to A_n \to B$ by soundness and completeness, is equivalent to
$\models A_1 \to \ldots \to A_n \to B$

Q: is it possible to write an algorithm that **decides** whether $A_1, \ldots, A_n \vdash B$ is a valid sequent?

We only need an algorithm to decide whether $\models A$:

$A_1, \ldots, A_n \vdash B$ by a theorem, is equivalent to
$\vdash A_1 \to \ldots \to A_n \to B$ by soundness and completeness, is equivalent to
$\models A_1 \to \ldots \to A_n \to B$

There are many ways to do this. One is to turn formulas into Conjunctive Normal Form (CNF).

CNF is a a formula which has the following structure:

- → It contains **literals** L which are either atoms (e.g., p) or their negation (e.g., $\neg p$)
- → It composes literals into **clauses** using disjunction ($\vee$)
- → It composes clauses into a **formula** using conjunction ($\wedge$)

CNF is a a formula which has the following structure:

- → It contains **literals** L which are either atoms (e.g., p) or their negation (e.g., $\neg p$)
- → It composes literals into **clauses** using disjunction ($\vee$)
- → It composes clauses into a **formula** using conjunction ($\wedge$)

example:

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

CNF is a a formula which has the following structure:

- It contains **literals** L which are either atoms (e.g., p) or their negation (e.g., $\neg p$)
- It composes literals into **clauses** using disjunction ($\vee$)
- It composes clauses into a **formula** using conjunction ($\wedge$)

example:

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

CNF formulas **do not contain**:

- double negation
- implication

A CNF formula is valid iff every clause contains a literal and its negation. (why?)

Valid formulas:

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

Not valid formulas:

$$p$$
$$(P \vee q)$$
$$(q \vee p \vee r \vee \neg q) \wedge (\neg p \vee s \vee p) \wedge (s)$$

A CNF formula is valid iff every clause contains a literal and its negation. (why?)

Valid formulas:

$$(p \lor \neg p)$$
$$(q \lor p \lor r \lor \neg q) \land (\neg p \lor s \lor p) \land (\neg s \lor s)$$

Not valid formulas:

$$p$$
$$(P \lor q)$$
$$(q \lor p \lor r \lor \neg q) \land (\neg p \lor s \lor p) \land (s)$$

The above gives an efficient algorithm to check validity of CNF formulas ($O(n)$ to the size of the formula).

Every fomula can be transformed to an equivalent CNF formula by the following method:

1. replace implication using the theorem: $A \rightarrow B \equiv \neg A \vee B$
2. push all negations inwards using De Morgan laws:

$$\neg(A_1 \wedge A_2) \equiv \neg A_1 \vee \neg A_2 \qquad \neg(A_1 \vee A_2) \equiv \neg A_1 \wedge \neg A_2$$

3. remove double negations: $\neg\neg A \equiv A$
4. distribute `and` over `or`: $(A_1 \wedge A_2) \vee B \equiv (A_1 \vee B) \wedge (A_2 \vee B)$

Every fomula can be transformed to an equivalent CNF formula by the following method:

1. replace implication using the theorem: $A \rightarrow B \equiv \neg A \vee B$
2. push all negations inwards using De Morgan laws:

$$\neg(A_1 \wedge A_2) \equiv \neg A_1 \vee \neg A_2 \qquad \neg(A_1 \vee A_2) \equiv \neg A_1 \wedge \neg A_2$$

3. remove double negations: $\neg\neg A \equiv A$
4. distribute **and** over `or`: $(A_1 \wedge A_2) \vee B \equiv (A_1 \vee B) \wedge (A_2 \vee B)$

The above conversion outputs in the worst case an exponentially large formula ($O(2^n)$ to the size of the input formula).

Convert to CNF and check the validity of the formulas:

- $\neg p \wedge q \rightarrow p \wedge (r \rightarrow q)$
- $p \rightarrow q \rightarrow r$
- $(p \rightarrow q \rightarrow r) \rightarrow (p \wedge q \rightarrow r)$
- $\perp \rightarrow p$
- $p \rightarrow \top$

Satisfiability: Given A, is there a model which makes A true?

Q: Can we decide satisfiability?

Satisfiability: Given A, is there a model which makes A true?

Q: Can we decide satisfiability?

Theorem

The satisfiability problem is decidable, and NP-complete

So there are known algorithms but they are not efficient in the worst case.

Satisfiability: Given A, is there a model which makes A true?

Q: Can we decide satisfiability?

Theorem

The satisfiability problem is decidable, and NP-complete

So there are known algorithms but they are not efficient in the worst case.

But there are efficient algorithms for a **some CNF formulas**: Horn clauses

A CNF formula is a horn formula if all its clauses have at most one positive literal

$$\neg p \vee \neg q \vee r \quad \text{becomes} \quad p \wedge q \rightarrow r$$
$$\neg p \vee \neg q \quad \text{becomes} \quad p \wedge q \rightarrow \bot$$
$$p \quad \text{becomes} \quad \top \rightarrow p$$

Algorithm: Inputs a Horn formula and maintains a list of literals, $\perp$, and $\top$ in the formula.
It **marks** the literals in this list as follows:

- → it marks $\top$ if it exists in the list
- → If there is a conjuct

  $$L_1 \wedge \ldots L_n \to L'$$

  and all $L_1 \wedge \ldots L_n$ are marked then mark L'. Repeat (2) until no more such conjuncts.
- → if $\perp$ marked then output “unsatisfiable” and stop
- → else output “satisfiable” and stop

Algorithm: Inputs a Horn formula and maintains a list of literals, $\perp$, and $\top$ in the formula.
It **marks** the literals in this list as follows:

- it marks $\top$ if it exists in the list
- If there is a conjuct
  $$L_1 \wedge \ldots L_n \to L'$$
  and all $L_1 \wedge \ldots L_n$ are marked then mark L'. Repeat (2) until no more such conjuncts.
- if $\perp$ marked then output "unsatisfiable" and stop
- else output "satisfiable" and stop

This is a $O(n)$ algorithm.

- $(p \wedge q \wedge s \to p) \wedge (q \wedge r \to p) \wedge (p \wedge s \to s)$
- $(p \wedge q \wedge s \to \bot) \wedge (q \wedge r \to p) \wedge (\top \to s)$
- $(p \wedge q \wedge s \to \bot) \wedge (p \wedge s \to q) \wedge (s \to p) \wedge (\top \to s)$
- $(p \wedge q \wedge s \to \bot) \wedge (s \to p) \wedge (\top \to s)$