# 02—Propositional Logic II

CS 5209: Foundation in Logic and AI

Martin Henz

January 21, 2010

Generated on Monday 18th January, 2010, 18:17

# Propositions

**Propositions are Declarative Sentences**

Sentences which one can—in principle—argue as being true or false.

**Examples**

1. The sum of the numbers 3 and 5 equals 8.
2. Jane reacted violently to Jack's accusations.
3. Every natural number $> 2$ is the sum of two prime numbers.

# Sequents as Logical Arguments

A Sequent in English

If *the train arrives late* and
*there are no taxis at the station* then
*John is late for his meeting.*

*John is not late for his meeting.*

*The train did arrive late.*

Therefore, *there were taxis at the station.*

# Focus on Structure, not Content

We are primarily concerned about the structure of arguments in this class, not the validity of statements in a particular domain.

We therefore simply abbreviate sentences by letters such as $p$, $q$, $r$, $p_1$, $p_2$ etc.

Instead of English words such as “if...then”, “and”, “not”, it is more convenient to use symbols such as $\rightarrow$, $\wedge$, $\neg$.

# Sequents in Symbolic Notation

**A Sequent in English**

If *the train arrives late* and *there are no taxis at the station* then *John is late for his meeting.*
*John is not late for his meeting.*
*The train did arrive late.*
Therefore, *there were taxis at the station.*

**The same sequent using letters and symbols**

$p \land \neg q \to r, \neg r, p \vdash q$

**Remaining task**

Develop proof rules that allows us to derive such sequents

# Rules for Conjunction

## Introduction of Conjunction

$$\frac{\phi \qquad \psi}{\phi \land \psi}\,[\land i]$$

## Elimination of Conjunction

$$\frac{\phi \land \psi}{\phi}\,[\land e_1] \qquad \frac{\phi \land \psi}{\psi}\,[\land e_2]$$

# Rules of Double Negation

## Introduction of Double Negation

$$\frac{\phi}{\neg\neg\phi}\,[\neg\neg i]$$

## Elimination of Double Negation

$$\frac{\neg\neg\phi}{\phi}\,[\neg\neg e]$$

# Rules for Eliminating Implication

$$\frac{\phi \qquad \phi \rightarrow \psi}{\psi}[\rightarrow e]$$

$$\frac{\phi \rightarrow \psi \qquad \neg\psi}{\neg\phi}[MT]$$

# Rule for Introduction of Implication

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \rightarrow \psi}\;[\rightarrow i]$$

# Rules for Introduction of Disjunction

$$\frac{\phi}{\phi \vee \psi}[\vee i_i] \qquad \frac{\psi}{\phi \vee \psi}[\vee i_2]$$

# Rule for Elimination of Disjunction

$$\frac{\phi \lor \psi \qquad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \qquad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\;[\lor e]$$

# Proof of $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$

| | | |
|---|---|---|
| 1 | $p \wedge (q \vee r)$ | premise |
| 2 | $p$ | $\wedge e_1$ 1 |
| 3 | $q \vee r$ | $\wedge e_2$ 1 |
| 4 | $q$ | assumption |
| 5 | $p \wedge q$ | $\wedge i$ 2,4 |
| 6 | $(p \wedge q) \vee (p \wedge r)$ | $\vee i_1$ 5 |
| 7 | $r$ | assumption |
| 8 | $p \wedge r$ | $\wedge i$ 2,7 |
| 9 | $(p \wedge q) \vee (p \wedge r)$ | $\vee i_2$ 8 |
| 10 | $(p \wedge q) \vee (p \wedge r)$ | $\vee e$ 3, 4–6, 7–9 |

# A Special Proposition

- Recall: We are only interested in the truth value of propositions, not the subject matter that they refer to (Martian pizzas or whatever).
- Therefore, all propositions that we all agree must be true are the same!
- Example: $p \rightarrow p$
- We denote the proposition that is always true using the symbol $\top$.

# Another Special Proposition

- We denote the proposition that is always true using the symbol $\top$.
- Similarly, we denote the proposition that is always false using the symbol $\bot$.
- Example: $p \land \neg p$

# Elimination of Negation

$$\frac{\phi \qquad \neg\phi}{\bot}\,[\neg e]$$

# Introduction of Negation

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\;[\neg i]$$

# Elimination of $\bot$

$$\frac{\bot}{\phi}\,[\bot e]$$

# Basic Rules (conjunction and disjunction)

$$\frac{\phi \qquad \psi}{\phi \wedge \psi}[\wedge i] \qquad \frac{\phi \wedge \psi}{\phi}[\wedge e_1] \qquad \frac{\phi \wedge \psi}{\psi}[\wedge e_2]$$

$$\frac{\phi}{\phi \vee \psi}[\vee i_i] \qquad \frac{\psi}{\phi \vee \psi}[\vee i_2] \qquad \frac{\phi \vee \psi \qquad \begin{array}{|c|}\hline \phi \\ \vdots \\ \chi \\ \hline \end{array} \qquad \begin{array}{|c|}\hline \psi \\ \vdots \\ \chi \\ \hline \end{array}}{\chi}[\vee e]$$

# Basic Rules (implication)

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \rightarrow \psi}\,[\rightarrow i] \qquad \frac{\phi \qquad \phi \rightarrow \psi}{\psi}\,[\rightarrow e]$$

# Basic Rules (negation)

$$\frac{\begin{array}{c}\phi \\ \vdots \\ \bot\end{array}}{\neg\phi}\,[\neg i] \qquad \frac{\phi \qquad \neg\phi}{\bot}\,[\neg e]$$

# Basic Rules (⊥ and double negation)

$$\frac{\bot}{\phi}\ [\bot e]$$

$$\frac{\neg\neg\phi}{\phi}\ [\neg\neg e]$$

# Some Derived Rules: Introduction of Double Negation

$$\frac{\phi}{\neg\neg\phi}\ [\neg\neg i]$$

# Example: Deriving [$\neg\neg$ i] from [$\neg$ i] and [$\neg$ e]

| | | |
|---|---|---|
| 1 | $\phi$ | premise |
| 2 | $\neg\phi$ | assumption |
| 3 | $\bot$ | $\neg$e 1,2 |
| 4 | $\neg\neg\phi$ | $\neg$i 2–3 |

# Some Derived Rules: Modus Tollens

$$\frac{\phi \to \psi \qquad \neg\psi}{\neg\phi}[MT]$$

# Some Derived Rules: Proof By Contradiction

$$\frac{\boxed{\begin{array}{c}\neg\phi \\ \vdots \\ \bot\end{array}}}{\phi}\ [\text{PBC}]$$

# Some Derived Rules: Law of Excluded Middle

$$\frac{}{\phi \vee \neg\phi}\ [\mathrm{LEM}]$$

# Recap: Logical Connectives

- $\neg$: negation of $p$ is denoted by $\neg p$
- $\vee$: disjunction of $p$ and $r$ is denoted by $p \vee r$, meaning at least one of the two statements is true.
- $\wedge$: conjunction of $p$ and $r$ is denoted by $p \wedge r$, meaning both are true.
- $\rightarrow$: implication between $p$ and $r$ is denoted by $p \rightarrow r$, meaning that $r$ is a logical consequence of $p$.

# Formal Description Required

**Use of Meta-Language**

When we describe rules such as $\dfrac{}{\phi \lor \neg\phi}\,[LEM]$ we mean that letters such as $\phi$ can be replaced by *any* formula.

But what exactly is the set of formulas that can be used for $\phi$?

**Allowed**

$(p \land (\neg q))$

**Not allowed**

$) \land p \;\; q \neg ($

# Definition of Well-formed Formulas

Definition

- Every propositional atom $p, q, r, \ldots$ and $p_1, p_2, p_3, \ldots$ is a well-formed formula.
- If $\phi$ is a well-formed formula, then so is $(\neg\phi)$.
- If $\phi$ and $\psi$ are well-formed formulas, then so is $(\phi \land \psi)$.
- If $\phi$ and $\psi$ are well-formed formulas, then so is $(\phi \lor \psi)$.
- If $\phi$ and $\psi$ are well-formed formulas, then so is $(\phi \rightarrow \psi)$.

# Definition very restrictive

How about this formula?

$$p \land \neg q \lor r$$

Usually, this is understood to mean

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

...but for the formal treatment of this section and the first homework, we insist on the strict definition, and exclude such formulas.

# Backus Naur Form: A more compact definition

Backus Naur Form for propositional formulas

$$\phi ::= p|(\neg\phi)|(\phi \wedge \phi)|(\phi \vee \phi)|(\phi \rightarrow \phi)$$

where $p$ stands for any atomic proposition.

# Inversion principle

How can we show that a a formula such as

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

is well-formed?

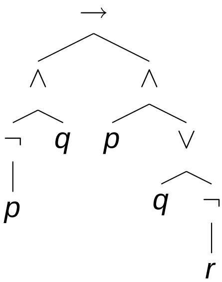
Answer: We look for the only applicable rule in the definition (the last rule in this case), and proceed on the parts.

# Parse trees

A formula

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

...and its parse tree:



# Meaning of propositional formula

**Meaning as mathematical object**

We define the meaning of formulas as a function that maps formulas and valuations to truth values.

**Approach**

We define this mapping based on the structure of the formula, using the meaning of their logical connectives.

# Truth values and valuations

Definition

The set of truth values contains two elements T and F, where T represents “true” and F represents “false”.

Definition

A *valuation* or *model* of a formula $\phi$ is an assignment of each propositional atom in $\phi$ to a truth value.

# Meaning of logical connectives

The meaning of a connective is defined as a truth table that gives the truth value of a formula, whose root symbol is the connective, based on the truth values of its components.

| $\phi$ | $\psi$ | $\phi \wedge \psi$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

# Truth tables of formulas

Truth tables use placeholders of formulas such as $\phi$:

| $\phi$ | $\psi$ | $\phi \land \psi$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

Build the truth table for given formula:

| $p$ | $q$ | $r$ | $(p \land q)$ | $((p \land q) \land r)$ |
|---|---|---|---|---|
| T | T | T | T | T |
| T | T | F | T | F |
| $\vdots$ | | | | |

# Truth tables of other connectives

| $\phi$ | $\psi$ | $\phi \lor \psi$ |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

| $\phi$ | $\psi$ | $\phi \to \psi$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

| $\phi$ | $\neg\phi$ |
|---|---|
| T | F |
| F | T |

| $\top$ |
|---|
| T |

| $\bot$ |
|---|
| F |

# Constructing the truth table of a formula

| $p$ | $q$ | $(\neg p)$ | $\neg q$ | $p \to \neg q$ | $q \lor \neg p$ | $(p \to \neg q) \to (q \lor \neg p)$ |
|---|---|---|---|---|---|---|
| T | T | F | F | F | T | T |
| T | F | F | T | T | F | F |
| F | T | T | F | T | T | T |
| F | F | T | T | T | T | T |

# Validity and Satisfiability

**Validity**

A formula is *valid* if it computes T for all its valuations.

**Satisfiability**

A formula is *satisfiable* if it computes T for at least one of its valuations.

# Semantic Entailment

**Definition**

If, for all valuations in which all $\phi_1, \phi_2, \ldots, \phi_n$ evaluate to T, the formula $\psi$ evaluates to T as well, we say that $\phi_1, \phi_2, \ldots, \phi_n$ semantically entail $\psi$, written:

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi$$

# Soundness of Propositional Logic

**Soundness**

Let $\phi_1, \phi_2, \ldots, \phi_n$ and $\psi$ be propositional formulas. If $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$, then $\phi_1, \phi_2, \ldots, \phi_n \models \psi$.

# Completeness of Propositional Logic

**Completeness**

Let $\phi_1, \phi_2, \ldots, \phi_n$ and $\psi$ be propositional formulas. If $\phi_1, \phi_2, \ldots, \phi_n \models \psi$, then $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.

# Provability

Definition

If there is a natural deduction proof of $\psi$ using the premises $\phi_1, \phi_2, \ldots, \phi_n$, we say that $\psi$ is *provable* from $\phi_1, \phi_2, \ldots, \phi_n$ and write

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$$

# Semantic Entailment

Definition

If, for all valuations in which all $\phi_1, \phi_2, \ldots, \phi_n$ evaluate to T, the formula $\psi$ evaluates to T as well, we say that $\phi_1, \phi_2, \ldots, \phi_n$ semantically entail $\psi$, written:

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi$$

# Some More Definitions

**Semantic equivalence**

Let $\phi$ and $\psi$ be formulas of propositional logic. We say that $\phi$ and $\psi$ are semantically equivalent iff $\phi \models \psi$ and $\psi \models \phi$ hold. We write $\psi \equiv \phi$.

**Validity**

If $\models \phi$ holds, we call $\phi$ *valid*.

# Soundness of Natural Deduction

**Soundness**

Let $\phi_1, \phi_2, \ldots, \phi_n$ and $\psi$ be propositional formulas. If $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$, then $\phi_1, \phi_2, \ldots, \phi_n \models \psi$.

# Proof outline

Proof by structural induction on the proof (as a graph) for $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.

1. **Base case:** Trivial proof. It must be a premise. Thus the sequent is $\phi \vdash \phi$. Whenever $\phi$ evaluates to T, $\phi$ evaluates to T. Thus $\phi \models \phi$.
2. **Inductive step:** Analyze every possible application of a proof rule, and use hypothesis to show $\phi_1, \phi_2, \ldots, \phi_n \models \psi$.

# Completeness of Propositional Logic

**Completeness**

Let $\phi_1, \phi_2, \ldots, \phi_n$ and $\psi$ be propositional formulas. If $\phi_1, \phi_2, \ldots, \phi_n \models \psi$, then $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.

# Proof outline

To show: If $\phi_1, \phi_2, \ldots, \phi_n \models \psi$, then $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.

Step 1: Show $\models \phi_1 \rightarrow (\phi_2 \rightarrow (\phi_3 \rightarrow (\cdots(\phi_n \rightarrow \psi)\cdots)))$.

Step 2: Show $\vdash \phi_1 \rightarrow (\phi_2 \rightarrow (\phi_3 \rightarrow (\cdots(\phi_n \rightarrow \psi)\cdots)))$.

Step 3: Show $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.

In Step 2, construct proof for each line of the truth table, and assemble the proofs into a single proof. This proof uses structural induction on the formula $\psi$.

# Questions about Propositional Formula

- Is a given formula valid?
- Is a given formula satisfiable?
- Is a given formula invalid?
- Is a given formula unsatisfiable?
- Are two formulas equivalent?

# Decision Problems

**Definition**

A *decision problem* is a question in some formal system with a yes-or-no answer.

**Examples**

The question whether a given propositional formula is satisifiable (unsatisfiable, valid, invalid) is a decision problem.

The question whether two given propositional formulas are equivalent is also a decision problem.

# How to Solve the Decision Problem?

**Question**

How do you decide whether a given propositional formula is satisfiable/valid?

**The good news**

We can construct a truth table for the formula and check if some/all rows have T in the last column.

# Satisifiability is Decidable

**An algorithm for satisifiability**

Using a truth table, we can implement an *algorithm* that returns *"yes"* if the formula is satisifiable, and that returns *"no"* if the formula is unsatisfiable.

**Decidability**

Decision problems for which there is an algorithm computing "yes" whenever the answer is "yes", and "no" whenever the answer is "no", are called *decidable*.

**Decidability of satisfiability**

The question, whether a given propositional formula is satisifiable, is decidable.

# The Bad News

**Concern**

In practice, propositional formulas can be large. Example:
`http://www.comp.nus.edu.sg/~cs5209/prop.txt`

**Techniques so far inadequate**

Proving satisfiability/validity using truth tables or natural deduction is impractical for large formulas.

# Is there a *practical* way of deciding satisfiability?

**Question**

Is there an *efficient* algorithm that decides whether a given formula is satisfiable?

**More precisely...**

Is there a *polynomial-time* algorithm that decides whether a given formula is satisfiable?

**Answer**

We do not know!

# What *do* we know about satisfiability?

**Truth assignment as witness**

If the answer is “yes”, then a satisfying truth assignment can serve as a proof that the answer is indeed “yes”.

**Witness for satisfiability**

Such a proof is called a *witness*.

**Checking the witness**

We can quickly check whether indeed the witness assignment makes the formula true. This can be done in time proportional to the size of the formula.

# The Complexity Class NP

**Definition**

Decision problems for which the “yes” answer has a proof that can be checked in polynomial time, are called *NP*.

**Origin of name**

NP stands for “**N**on-deterministic **P**olynomial time”.

**Original definition**

NP is the set of decision problems solvable in polynomial time by a non-deterministic Turing machine.

# Some History

- The concept of NP-completeness was introduced by Stephen Cook in 1971 at the 3rd Annual ACM Symposium on Theory of Computing.
- At the conference, there was a fierce debate whether there could be a polynomial time algorithm to solve such problems.
- John Hopcroft convinced the delegates that the problem should be put off to be solved at some later date.
- In 1972, Richard Karp presented 21 NP-completed problems.
- Cook and Leonid Levin proved independently that propositional satisifiability is in this class.

# Algorithms for Proving Satisfiability of $\psi$

- Transform $\neg\psi$ into Conjunctive Normal Form *ncnf* and prove validity (non-validity) of *ncnf*
- Transform $\psi$ into Conjunctive Normal Form *cnf* and search for a satisfying valuation
  - Complete algorithms: guaranteed to terminate with correct answer
    example: DPLL
  - Incomplete algorithms: Return “yes” for some satisfiable formulas, and run forever for other satisfiable formulas and all unsatisfiable formulas; example: WalkSAT
- Transform $\psi$ into DAG; return “yes” for some satisfiable formulas, return “no” for some unsatisfiable formulas, return “don’t know” otherwise; example: linear solver (1.6.1)

# Conjunctive Normal Form

Definition

A literal $L$ is either an atom $p$ or the negation of an atom $\neg p$. A formula $C$ is in *conjunctive normal form* (CNF) if it is a conjunction of clauses, where each clause is a disjunction of literals:

$$\begin{aligned} L &::= p \mid \neg p \\ D &::= L \mid L \vee D \\ C &::= D \mid D \wedge C \end{aligned}$$

# Examples

$(\neg p \lor q \lor r) \land (\neg q \lor r) \land (\neg r)$ is in CNF.
$(\neg p \lor q \lor r) \land ((p \land \neg q) \lor r) \land (\neg r)$ is not in CNF.
$(\neg p \lor q \lor r) \land \neg(\neg q \lor r) \land (\neg r)$ is not in CNF.

# Usefulness of CNF

**Lemma**

A disjunction of literals $L_1 \vee L_2 \vee \cdots \vee L_m$ is valid iff there are $1 \leq i, j \leq m$ such that $L_i$ is $\neg L_j$.

How to disprove

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

Disprove any of:

$$\models (\neg q \vee p \vee r) \qquad \models (\neg p \vee r) \qquad \models q$$

# Usefulness of CNF

**Lemma**

A disjunction of literals $L_1 \vee L_2 \vee \cdots \vee L_m$ is valid iff there are $1 \leq i, j \leq m$ such that $L_i$ is $\neg L_j$.

How to prove

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

Prove all of:

$$\models (\neg q \vee p \vee q) \qquad \models (p \vee r \neg p) \qquad \models (r \vee \neg r)$$

# Usefulness of CNF

**Proposition**

Let $\phi$ be a formula of propositional logic. Then $\phi$ is satisfiable iff $\neg\phi$ is not valid.

**Satisfiability test**

We can test satisfiability of $\phi$ by transforming $\neg\phi$ into CNF, and show that some clause is not valid.

# Transformation to CNF

**Theorem**

Every formula in the propositional calculus can be transformed into an equivalent formula in CNF.

# Algorithm for CNF Transformation

1. Eliminate implication using:
   $A \rightarrow B \equiv \neg A \vee B$
2. Push all negations inward using De Morgan's laws:
   $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$
   $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$
3. Eliminate double negations using the equivalence $\neg\neg A \equiv A$
4. The formula now consists of disjunctions and conjunctions of literals. Use the distributive laws
   $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
   $(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$
   to eliminate conjunctions within disjunctions.

# Example

$$
\begin{aligned}
(\neg p \rightarrow \neg q) \rightarrow (p \rightarrow q) &\equiv \neg(\neg\neg p \vee \neg q) \vee (\neg p \vee q) \\
&\equiv (\neg\neg\neg p \wedge q) \vee (\neg p \vee q) \\
&\equiv (\neg p \wedge q) \vee (\neg p \vee q) \\
&\equiv (\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q)
\end{aligned}
$$