Lecture 1: Propositional Logic

- Syntax
- Semantics
- Truth tables
- Implications and Equivalences
- Valid and Invalid arguments
- Normal forms
- Davis-Putnam Algorithm

Atomic propositions and logical connectives

An *atomic proposition* is a statement or assertion that must be true or false.

Examples of atomic propositions are: "5 is a prime" and "program P terminates".

Propositional formulas are constructed from atomic propositions by using *logical connectives*.

Connectives	
0	false
1	true
$\neg$	not
$\wedge$	and
$\vee$	or
$\rightarrow$	conditional (implies)
$\Leftrightarrow$	biconditional (equivalent)

A typical propositional formula is $A \wedge (B \vee C) \rightarrow B$.

The *truth value* of a propositional formula can be calculated from the truth values of the atomic propositions it contains.

Well-formed propositional formulas

The *well-formed formulas* of propositional logic are obtained by using the construction rules below:

- An atomic proposition ϕ is a well-formed formula.
- If ϕ is a well-formed formula, then so is $\neg\phi$.
- If ϕ and ψ are well-formed formulas, then so are $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, and $\phi \Leftrightarrow \psi$.
- If ϕ is a well-formed formula, then so is (ϕ).

Alternatively, can use *Backus-Naur Form (BNF)*:

$$
\begin{array}{lll}
< \text{formula} > & ::= & \text{Atomic Proposition} \\
& | & \neg < \text{formula} > \\
& | & < \text{formula} > \wedge < \text{formula} > \\
& | & < \text{formula} > \vee < \text{formula} > \\
& | & < \text{formula} > \rightarrow < \text{formula} > \\
& | & < \text{formula} > \Leftrightarrow < \text{formula} > \\
& | & (< \text{formula} >)
\end{array}
$$

Truth functions

The truth of a propositional formula $S(x_1, x_2, \ldots, x_n)$ is a function of the truth values of the atomic propositions $x_1, x_2, \ldots, x_n$ it contains.

A *truth assignment* is a mapping that associates a truth value with each of the atomic propositions $x_1, x_2, \ldots, x_n$. Let α be a truth assignment for $x_1, x_2, \ldots, x_n$.

If we identify 0 with *false* and 1 with *true*, we can easily determine the truth value of $S(x_1, x_2, \ldots, x_n)$ under α.

S	$S[\alpha]$
$0/1$	$0/1$
$\neg A$	$1 - A[\alpha]$
$A \wedge B$	$\min(A[\alpha], B[\alpha])$
$A \vee B$	$\max(A[\alpha], B[\alpha])$

The other logical connectives can be handled in a similar manner.

Truth functions are sometimes called *Boolean functions*.

Truth tables for basic logical connectives

A *truth table* shows whether a propositional formula is true or false for each possible truth assignment.

If we know how the five basic logical connectives work, it is easy (in principle) to construct a truth table.

X	$\neg X$
0	1
1	0

X	Y	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \Leftrightarrow Y$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Mistake in table for implication?

Notice that $X \rightarrow Y$ is only $false$ if X is $true$ and Y is $false$.

If X is $false$, then the implication $X \rightarrow Y$ will be $true$. Could this possibly be correct?

Some people feel that it is counterintuitive to say that the implication

"If horses have wings, then elephants can dance"

is true, when we know that horses don't have wings and that elephants can't dance.

There are four possible truth tables for implication $X \rightarrow Y$:

X	Y	T1	T2	T3	T4
0	0	0	1	0	1
0	1	0	0	1	1
1	0	0	0	0	0
1	1	1	1	1	1

Mistake in table for implication?

First Argument:

- If we used T1, then $X \to Y$ would have the same table as $X \wedge Y$.
- If we used T2, then $X \to Y$ would have the same table as $X \Leftrightarrow Y$.
- If we used T3, then $X \to Y$ would have the same table as Y–even worse!

Clearly, each of these three alternatives is unreasonable. Table T4 is the only remaining possibility.

Mistake in table for implication?

Second Argument:

- We would certainly want $X \to (X \vee Y)$ to be a tautology. Let's test each of the four possible choices for $X \to Y$.

		$X \to (X \vee Y)$			
X	Y	T1	T2	T3	T4
0	0	0	1	0	1
0	1	0	0	1	1
1	0	1	1	1	1
1	1	1	1	1	1

Only T4 makes the implication a tautology.

A more complex truth table

Let S be the formula

$$(((A \wedge B) \Rightarrow C) \wedge (A \Rightarrow B)) \Rightarrow (A \Rightarrow C).$$

To construct the truth table for S we must consider all possible truth assignments for A, B, and C. In this case there are $2^3 = 8$ such truth assignments. Hence, the table for S will have 8 rows.

A	B	C	$(A \wedge B) \rightarrow C$	$A \rightarrow B$	$A \rightarrow C$	S
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	0	1
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	1	1	1	1	1

In general, if the truth of a formula depends on n propositions, its truth table will have 2^n rows.

Special formulas

A propositional formula S is

- a *tautology* if $S[\alpha] = 1$ for all α.
- a *contradiction* if $S[\alpha] = 0$ for all α.
- *satisfiable* if $S[\alpha] = 1$ for some α.

It is easy to see that $A \vee \neg A$ is a tautology and that $A \wedge \neg A$ is a contradiction,

The truth table on the previous page shows that the formula S is a tautology.

Note that

- A is a contradiction iff $\neg A$ is a tautology.
- A is satisfiable iff $\neg A$ is not a tautology.

Major open problem: Is there a more efficient way to determine if a formula is a tautology (is satisfiable) than by constructing its truth table?

Implications

In the formula $A \Rightarrow B$

- A is the antecedent, hypothesis or premise
- B is the consequent or conclusion

Can be associated with 3 variants:

- Converse: $B \Rightarrow A$
- Inverse: $\neg A \Rightarrow \neg B$
- Contrapositive: $\neg B \Rightarrow \neg A$

$\star$ An implication and its contrapositive are equivalent.

$\star$ **Modus Ponens:** Given A and $A \Rightarrow B$, conclude B.

$\star$ **Modus Tollens:** Given $A \Rightarrow B$ and $\neg B$, conclude $\neg A$.

Equivalences

Two formulae S and T are *equivalent* iff for any truth assignment α we have $S[\alpha] = T[\alpha]$.

Claim: S and T are equivalent iff $S \Leftrightarrow T$ is a tautology.

Some Useful Equivalences that can be used to simplify complex formulas:

$$
\begin{aligned}
A \wedge 0 &\Leftrightarrow 0 \\
A \wedge 1 &\Leftrightarrow A \\
A \wedge A &\Leftrightarrow A \\
A \wedge B &\Leftrightarrow B \wedge A \\
A \wedge (B \wedge C) &\Leftrightarrow (A \wedge B) \wedge C \\
A \wedge (B \vee C) &\Leftrightarrow (A \wedge B) \vee (A \wedge C) \\
A \wedge B &\Leftrightarrow \neg(\neg A \vee \neg B) \\
A \Rightarrow B &\Leftrightarrow \neg A \vee B
\end{aligned}
$$

When is an argument valid?

An *argument* is an assertion that a set of statements, called the *premises*, yields another statement, called the *conclusion*.

An argument is *valid* if and only if the conjunction of the premises implies the conclusion.

In other words, if we grant that the premises are all true, then the conclusion must be true also.

An invalid argument is called a *fallacy*. Unfortunately, fallacies are probably more common than valid arguments.

In many cases, the validity of an argument can be checked by constructing a truth table.

All we have to do is show that the conjunction of the premises implies the conclusion.

Valid and Invalid Arguments

Which of the following arguments are valid?

1. If I am wealthy, then I am happy. I am happy. Therefore, I am wealthy.
2. If John drinks beer, he is at least 18 years old. John does not drink beer. Therefore, John is not yet 18 years old.
3. If girls are blonde, they are popular with boys. Ugly girls are unpopular with boys. Intellectual girls are ugly. Therefore, blonde girls are not intellectual.
4. If I study, then I will not fail basket weaving 101. If I do not play cards to often, then I will study. I failed basket weaving 101. Therefore, I played cards too often.

A More Complicated Example!

The following example is due to Lewis Carroll. Prove that it is a valid argument.

1. All the dated letters in this room are written on blue paper.
2. None of them are in black ink, except those that are written in the third person.
3. I have not filed any of those that I can read.
4. None of those that are written on one sheet are undated.
5. All of those that are not crossed out are in black ink.
6. All of those that are written by Brown begin with “Dear Sir.”
7. All of those that are written on blue paper are filed.
8. None of those that are written on more than one sheet are crossed out.
9. None of those that begin with “Dear sir” are written in the third person.

Therefore, I cannot read any of Brown’s letters.

Lewis Carrol example (cont.)

Let

- p be “the letter is dated,”
- q be “the letter is written on blue paper,”
- r be “the letter is written in black ink,”
- s be “the letter is written in the third person,”
- t be “the letter is filed,”
- u be “I can read the letter,”
- v be “the letter is written on one sheet,”
- w be “the letter is crossed out,”
- x be “the letter is written by Brown,”
- y be “the letter begins with ‘Dear Sir’ “

Lewis Carrol example (cont.)

Now, we can write the argument in propositional logic.

1. $p \rightarrow q$
2. $\neg s \rightarrow \neg r$
3. $u \rightarrow \neg t$
4. $v \rightarrow p$
5. $\neg w \rightarrow r$
6. $x \rightarrow y$
7. $q \rightarrow t$
8. $\neg v \rightarrow \neg w$
9. $y \rightarrow \neg s$

Therefore $x \rightarrow \neg u$

Negation Normal Form

Some more useful equivalences:

- $\neg\neg P \Leftrightarrow P$
- $\neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$
- $\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$
- $\neg(P \rightarrow Q) \Leftrightarrow (P \wedge \neg Q)$

Negation Normal Form

The *negation* of

$$Q \rightarrow ((P \wedge \neg R) \vee \neg S)$$

is simply

$$\neg[Q \rightarrow ((P \wedge \neg R) \vee \neg S)].$$

This may not be very useful. Often desirable to simplify formula as much as possible using four tautologies above.

$$\begin{aligned}
\neg[Q &\rightarrow ((P \wedge \neg R) \vee \neg S)] \\
Q &\wedge \neg[((P \wedge \neg R) \vee \neg S)] \\
Q &\wedge (\neg[(P \wedge \neg R)] \wedge \neg[\neg S]) \\
Q &\wedge (\neg[(P \wedge \neg R)] \wedge S) \\
Q &\wedge ((\neg P \vee \neg[\neg R]) \wedge S) \\
Q &\wedge ((\neg P \vee R) \wedge S)
\end{aligned}$$

The resulting formula is said to be in *negation normal form*.

Disjunctive Normal Form

- Every propositional formula is equivalent to a formula in *disjunctive normal form (DNF)*:

  $$(A_{11} \wedge A_{12} \wedge \ldots \wedge A_{1n_1}) \vee (A_{21} \wedge \ldots \wedge A_{2n_2}) \vee \ldots (A_{m1} \wedge \ldots \wedge A_{mn_m})$$

  where each A_{ij} is a literal (an atomic proposition or the negation of one).

  In short: $\bigvee_i \bigwedge_j A_{ij}$.

- Every propositional formula is equivalent to a formula in *conjunctive normal form (CNF)*:

  $$(A_{11} \vee A_{12} \vee \ldots \vee A_{1n_1}) \wedge (A_{21} \vee \ldots \vee A_{2n_2}) \wedge \ldots (A_{m1} \vee \ldots \vee A_{mn_m})$$

  where each A_{ij} is a literal.

  In short: $\bigwedge_i \bigvee_j A_j$.

How hard is it to check if CNF formula is a tautology? How about DNF? How about checking for satisfiability instead?

Connectives

From CNF (or DNF) it follows that no connectives other than $\{\vee, \wedge, \neg\}$ are really needed.

Since $A \wedge B$ is equivalent to $\neg(\neg A \vee \neg B)$, we only need $\{\vee, \neg\}$.

Likewise, $\{\wedge, \neg\}$ is sufficient.

Likewise, $\{\Rightarrow, \neg\}$ is sufficient.

But $\{\wedge, \vee\}$ is not sufficient.

NAND

Consider the binary connective $x \otimes y := \neg(x \wedge y)$.

Claim: $\{\otimes\}$ alone is sufficient.

Deciding satisfiability

The fastest known algorithms for deciding propositional satisfiability are based on the Davis-Putnam Algorithm.

A *unit clause* is a clause that consists of a single literal.

function Satisfiable (clause list S) **returns** boolean;
    /* unit propagation */
    repeat
        for each unit clause $L \in S$ **do**
            delete from S every clause containing L
            delete $\neg L$ from every clause of S in which it occurs
        end for
        if S is empty **then return** TRUE
        else if null clause is in S then **return** FALSE **end if**
    until no further changes result **end repeat**
    /* splitting */
    choose a literal L occurring in S
    if Satisfiable ($S \cup \{L\}$) **then return** TRUE
    else if Satisfiable ($S \cup \{\neg L\}$ **then return** TRUE
    else return FALSE **end if**
end function