CS228 Logic for Computer Science 2021

Lecture 3: Semantics and truth tables

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2021-01-08

Topic 3.1

Semantics - meaning of the formulas

Truth values

We denote the set of truth values as $\mathcal{B} \triangleq \{0, 1\}$.

0 and 1 are only distinct objects without any intuitive meaning.

We may view 0 as false and 1 as true, but it is only our emotional response to the symbols.

Model

Definition 3.1

A model is an element of $\text{Vars} \rightarrow \mathcal{B}$.

Example 3.1

$\{p_1 \mapsto 1, p_2 \mapsto 0, p_3 \mapsto 0, \dots\}$ *is a model*

Since Vars is countable, the set of models is non-empty and infinite.

A model m may or may not satisfy a formula F.
The satisfaction relation is usually denoted by $m \models F$ in infix notation.

Propositional Logic Semantics

Definition 3.2

The satisfaction relation $\models$ between models and formulas is the smallest relation that satisfies the following conditions.

- $m \models \top$
- $m \models p$ if $m(p) = 1$
- $m \models \neg F$ if $m \not\models F$
- $m \models F_1 \lor F_2$ if $m \models F_1$ or $m \models F_2$
- $m \models F_1 \land F_2$ if $m \models F_1$ and $m \models F_2$
- $m \models F_1 \oplus F_2$ if $m \models F_1$ or $m \models F_2$, but not both
- $m \models F_1 \Rightarrow F_2$ if if $m \models F_1$ then $m \models F_2$
- $m \models F_1 \Leftrightarrow F_2$ if $m \models F_1$ iff $m \models F_2$

Exercise 3.1

Why $\bot$ is not explicitly mentioned in the above definition?

Example: satisfaction relation

Example 3.2

Consider model $m = \{p_1 \mapsto 1, p_2 \mapsto 0, p_3 \mapsto 0, \dots\}$ *and formula* $(p_1 \Rightarrow (\neg p_2 \Leftrightarrow (p_1 \wedge p_3)))$

$$
\begin{tikzcd}
 & m \not\models (p_1 \Rightarrow (\neg p_2 \Leftrightarrow (p_1 \wedge p_3))) \arrow[dl] \arrow[dr] & & & \\
m \models p_1 & & m \not\models (\neg p_2 \Leftrightarrow (p_1 \wedge p_3)) \arrow[dl] \arrow[dr] & & \\
 & m \models \neg p_2 \arrow[d] & & m \not\models (p_1 \wedge p_3) \arrow[dl] \arrow[dr] & \\
 & m \not\models p_2 & m \models p_1 & & m \not\models p_3
\end{tikzcd}
$$

Exercise 3.2

Formally, write the satisfiability checking procedure .

Satisfiable, valid, unsatisfiable

We say

- *m satisfies F* if $m \models F$,
- F is *satisfiable* if there is a model m such that $m \models F$,
- F is *valid* (written $\models F$) if for each model m $m \models F$, and
- F is *unsatisfiable* (written $\not\models F$) if there is no model m such that $m \models F$.

Exercise 3.3

If F is sat then $\neg F$ is ____. sat or unsat

If F is valid then $\neg F$ is ____. unsat

If F is unsat then $\neg F$ is ____. valid

A valid formula is also called a tautology.

Overloading $\models$: set of models

We extend the usage of $\models$ in the following natural ways.

Definition 3.3

Let M be a (possibly infinite) set of models. $M \models F$ if for each $m \in M$, $m \models F$.

Example 3.3

$\{\{p \to 1, q \to 1\}, \{p \to 1, q \to 0\}\} \models p \vee q$

Exercise 3.4

Which of the following hold?

- $\{\{p \to 1, q \to 1\}, \{p \to 0, q \to 0\}\} \models p$
- $\{\{p \to 1, q \to 1\}\} \models p \wedge q$
- $\{\{p_i \to (k = i) | i \in \mathbb{N}\} | k \in \mathbb{N}\} \models p_1$

Overloading $\models$: set of formulas

Definition 3.4

Let Σ be a (possibly infinite) set of formulas.
$\Sigma \models F$ *if for each model m that satisfies each formula in Σ, $m \models F$.*

- $\Sigma \models F$ is read Σ implies F.
- If $\{G\} \models F$ then we may write $G \models F$.

Example 3.4

$\{p, q\} \models p \lor q$

Exercise 3.5

Which of the following hold?

- $\{p, q\} \models p \land q$ ✗
- $\{p \Rightarrow q, q \Rightarrow p\} \models p \Leftrightarrow q$ ✗
- $\{p \Rightarrow q, q\} \models p \oplus q$ ✗
- $\{p \Rightarrow q, \neg q, p\} \models p \oplus q$ ✗

Equivalent

Definition 3.5
Let $F \equiv G$ *if for each model m*

$$m \models F \text{ iff } m \models G.$$

Example 3.5
$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Equisatisfiable and Equivalid

Definition 3.6

Formulas F and G are equisatisfiable *if*

$$F \text{ is sat} \quad \text{iff} \quad G \text{ is sat.}$$

Definition 3.7

Formulas F and G are equivalid *if*

$$\models F \quad \text{iff} \quad \models G.$$

Commentary: The concept of equisatisfiable is used in formula transformations. We often say that after a transformation the formula remained equisatisfiable. Equivalid is the dual concept, rarely used in practice.

Topic 3.2

Decidability of SAT

Notation alert: decidable

A problem is decidable if there is an algorithm to solve the problem.

Propositional satisfiability problem

The following problem is called the satisfiability problem

$$\text{For a given } F \in \mathrm{P}, \text{ is } F \text{ satisfiable?}$$

Theorem 3.1

The propositional satisfiability problem is decidable. → By checking from TT

Proof.

Let $n = |\mathrm{Vars}(F)|$.
We need to enumerate 2^n elements of $\mathrm{Vars}(F) \to \mathcal{B}$.

If any of the models satisfy the formula, then F is sat. Otherwise, F is unsat. □

Exercise 3.6

Give a procedure to decide the validity of a formula. TT

Complexity of the decidability question?

- If we enumerate all models to check satisfiability, the cost is exponential

  → NP hard

- We do not know if we can do better.
- However, there are several tricks that have made satisfiability checking practical for the real-world formulas.

Topic 3.3

Truth tables

Truth tables

Truth tables was the first method to decide propositional logic.

The method is usually presented in slightly different notation. We need to assign a truth value to every formula.

Truth function

A model m is in Vars $\to \mathcal{B}$.

We can extend m to P $\to \mathcal{B}$ in the following way.

$$m(F) = \begin{cases} 1 & m \models F \\ 0 & otherwise. \end{cases}$$

The extended m is called truth function.

Since truth functions are natural extensions of models, we did not introduce new symbols.

Truth functions for logical connectives

Let F and G are logical formulas, and m is a model.
Due to the semantics of the propositional logic, the following holds for the truth functions.

$m(F)$	$m(\neg F)$
0	1
1	0

$m(F)$	$m(G)$	$m(F \land G)$	$m(F \lor G)$	$m(F \oplus G)$	$m(F \Rightarrow G)$	$m(F \Leftrightarrow G)$
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	1	1	0	1	1

Truth table

For a formula F, a truth table consists of $2^{|\mathrm{Vars}(F)|}$ rows. Each row considers one of the models and computes the truth value of F for each of them.

Example 3.6

Consider $(p_1 \Rightarrow (\neg p_2 \Leftrightarrow (p_1 \wedge p_3)))$. *We will not write* $m(.)$ *in the top row for brevity.*

p_1	p_2	p_3	$(p_1$	$\Rightarrow$	$(\neg$	p_2	$\Leftrightarrow ($	p_1	$\wedge$	$p_3\,)))$
0	0	0	0	1	1	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	1
0	1	0	0	1	0	1	1	0	0	0
0	1	1	0	1	0	1	1	0	0	1
1	0	0	1	0	1	0	0	1	0	0
1	0	1	1	1	1	0	1	1	1	1
1	1	0	1	1	0	1	1	1	0	0
1	1	1	1	0	0	1	0	1	1	1

The column under the leading connective has 1s therefore the formula is sat. But, there are some 0s in the column therefore the formula is not valid.

Example : DeMorgan law

Example 3.7

Let us show $p \lor q \equiv \neg(\neg p \land \neg q)$.

p	q	$(p \lor q)$	$\neg$	$(\neg$	p	$\land$	$\neg$	$q)$
0	0	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	1
1	0	1	1	0	1	0	1	0
1	1	1	1	0	1	0	0	1

Since the truth values of both the formulas are same in each row, the formulas are equivalent.

Exercise 3.7

Show $p \land q \equiv \neg(\neg p \lor \neg q)$ *using a truth table*

Commentary: $p \land q \equiv \neg(\neg p \lor \neg q)$ and $p \lor q \equiv \neg(\neg p \land \neg q)$ are called DeMorgan law.

Example : definition of $\Rightarrow$

Example 3.8

Let us show $p \Rightarrow q \equiv (\neg p \lor q)$.

p	q	$(p \Rightarrow q)$	$(\neg$	p	$\lor$	$q)$
0	0	1	1	0	1	0
0	1	1	1	0	1	1
1	0	0	0	1	0	0
1	1	1	0	1	1	1

Since the truth values of both the formulas are same in each row, the formulas are equivalent.

It appears that $\Rightarrow$ is a redundant symbol. We can write it in terms of the other symbols.

Example : definition of $\Leftrightarrow$

Example 3.9

Let us show $p \Leftrightarrow q \equiv (p \Rightarrow q) \land (q \Rightarrow p)$.

p	q	$(p \Leftrightarrow q)$	$(p$	$\Rightarrow$	$q)$	$\land$	$(q$	$\Rightarrow$	$p)$
0	0	1	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1	0	0
1	0	0	1	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1

Example: definition $\oplus$

Example 3.10

Let us show $(p \oplus q) \equiv (\neg p \land q) \lor (p \land \neg q)$ *using truth table.*

p	q	$(p \oplus q)$	$(\neg$	p	$\land$	$q)$	$\lor$	$(p$	$\land$	$\neg$	$q)$
0	0	0	1	0	0	0	0	0	0	1	0
0	1	1	1	0	1	1	1	0	0	0	1
1	0	1	0	1	0	0	1	1	1	1	0
1	1	0	0	1	0	1	0	1	0	0	1

Exercise 3.8

Show $(p \oplus q) \equiv (\neg p \lor \neg q) \land (p \lor q)$

Example: associativity

Example 3.11

Let us show $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

p	q	r	$(p$	$\wedge$	$q)$	$\wedge$	r	p	$\wedge$	$(q$	$\wedge$	$r)$
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	1	0	0	1	1	1
1	0	0	1	0	0	0	0	1	0	0	0	0
1	0	1	1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	0	0	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1

Exercise: associativity

Exercise 3.9

Prove/disprove using truth tables

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

Exercise: distributivity

Exercise 3.10

Prove/disprove using truth tables prove that $\wedge$ distributes over $\vee$ and vice-versa.

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

Tedious truth tables

- We need to write 2^n rows even if a simple observation about the formula prove (un)satisfiability.

  For example,

  - $(a \vee (c \wedge a))$ is sat (why? - no negation)
  - $(a \vee (c \wedge a)) \wedge \neg(a \vee (c \wedge a))$ is unsat (why?- contradiction at top level)

- We should be able to take such shortcuts?

We will see methods that will allow us to take such shortcuts. But not now!

Topic 3.4

Expressive power of propositional logic

Boolean functions

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

A formula F with $\mathrm{Vars}(F) = \{p_1, \dots, p_n\}$ can be viewed as a Boolean function f that is defined as follows.

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

We say F represents f.

Example 3.12

Formula $p_1 \vee p_2$ represents the following function

$$f = \{(0,0) \rightarrow 0, (0,1) \rightarrow 1, (1,0) \rightarrow 1, (1,1) \rightarrow 1\}$$

A Boolean function is another way of writing truth table.

Expressive power

Theorem 3.2

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 to represent f.

Let $p_i^0 \triangleq \neg p_i$ and $p_i^1 \triangleq p_i$.

$$\text{For } (b_1, \dots, b_n) \in \mathcal{B}^n, \text{ let } \quad F_{(b_1,\dots,b_n)} \triangleq \begin{cases} (p_1^{b_1} \land \dots \land p_n^{b_n}) & \text{if } f(b_1, \dots, b_n) = 1 \\ \perp & \text{otherwise.} \end{cases}$$

$$F \triangleq \underbrace{F_{(0,\dots,0)} \lor \dots \lor F_{(1,\dots,1)}}_{\text{All Boolean combinations}}$$

We used only three logical connectives to construct F

→ SoP

□

Exercise 3.11

Workout if F really represents f.

Insufficient expressive power

If we do not have sufficiently many logical connectives, we cannot represent all Boolean functions.

Example 3.13

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

To prove this we show that Boolean function $f = \{0 \rightarrow 1, 1 \rightarrow 1\}$ can not be achieved by any combination of $\wedge$s.

We setup induction over the sizes of formulas consisting a variable p and $\wedge$.

Commentary: We are assuming that only one variable occurs in the formula, since there is exactly one input to f. Our definition of “represents” requires the number of variables must match the arity of the function.

Insufficient expressive power II

base case:
Only choice is p.(why?) For $p = 0$, the function does not match.

induction step:
Let us assume that formulas F and G of size less than $n-1$ do not represent f.
We can construct a longer formula in the following way.

$$(F \wedge G)$$

The formula does not represent f, because we can always pick(why?) a model when F or G produces 0.

Therefore $\wedge$ alone is not expressive enough.

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 3.14

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

- $\top \equiv p \lor \neg p$ *for some* $p \in \mathrm{Vars}$
- $\bot \equiv \neg\top$
- $(p \land q) \equiv \neg(\neg p \lor \neg q)$
- $(p \oplus q) \equiv (p \land \neg q) \lor (\neg p \land q)$
- $(p \Rightarrow q) \equiv (\neg p \lor q)$
- $(p \Leftrightarrow q) \equiv (p \Rightarrow q) \land (q \Rightarrow p)$

Exercise 3.12

a. Show $\neg$ and $\land$ can define all the other connectives *b. Show $\oplus$ alone can not define $\neg$*

Topic 3.5

Problems

Semantics

Exercise 3.13

Show $F(\perp/p) \wedge F(\top/p) \models F \models F(\perp/p) \vee F(\top/p)$.

Truth tables

Exercise 3.14

Prove/disprove validity of the following formulas using truth tables.

1. $(p \Rightarrow (q \Rightarrow r)) \Leftrightarrow ((p \land q) \Rightarrow r))$
2. $p \land (q \oplus r) \Leftrightarrow (p \land q) \oplus (q \land r)$
3. $(p \lor q) \land (\neg q \lor r) \Leftrightarrow (p \lor r)$
4. $\perp \Rightarrow F$ *for any* F

Expressive power

Exercise 3.15

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

Exercise 3.16

Prove/disprove that the following subsets of connectives are fully expressive.

- $\lor, \oplus$ ✗
- $\bot, \oplus$
- $\Rightarrow, \oplus$
- $\lor, \land$ ✗
- $\Rightarrow, \bot$ ✓

Expressive power(2)

Exercise 3.17
Prove/disprove: if-then-else is fully expressive

Exercise 3.18
Show $\Rightarrow$ *alone can not express all the Boolean functions*

All minimal combinations*

Exercise 3.19

List all minimal subsets of the logical connectives that are fully expressive.

Encode boolean functions***

Exercise 3.20

Find smallest formulas that encode the following functions over n inputs

- *Encode parity function*
- *Encode majority function*

$\models$ vs. $\Rightarrow$

Exercise 3.21

Using truth table prove the following

- $F \models G$ *if and only if* $\models (F \Rightarrow G)$.
- $F \equiv G$ *if and only if* $\models (F \Leftrightarrow G)$.

Exercise: downward saturation

Exercise 3.22

Let us suppose we only have connectives $\wedge$, $\vee$, or $\neg$ in our formulas. Consider a set Σ of formulas such that

1. *for each* $p \in \text{Vars}$, $p \notin \Sigma$ *or* $\neg p \notin \Sigma$
2. *if* $\neg\neg F \in \Sigma$ *then* $F \in \Sigma$
3. *if* $(F \wedge G) \in \Sigma$ *then* $F \in \Sigma$ *and* $G \in \Sigma$
4. *if* $\neg(F \vee G) \in \Sigma$ *then* $\neg F \in \Sigma$ *and* $\neg G \in \Sigma$
5. *if* $(F \vee G) \in \Sigma$ *then* $F \in \Sigma$ *or* $G \in \Sigma$
6. *if* $\neg(F \wedge G) \in \Sigma$ *then* $\neg F \in \Sigma$ *or* $\neg G \in \Sigma$

Show that Σ is satisfiable, i.e., there is a model that satisfies every formula in Σ.

Exercise 3.23

Given algorithm that extends a set Σ into a set of the formula that satisfy the above. Can we use the algorithm as a satisfiability checker?

Commentary: Please note that the above does not hold if we drop any of the six conditions. You need to show that all six are needed.

Exercise: counting models

Exercise 3.24

Let propositional variables p, q, are r be relevant to us. There are eight possible models to the variables. Out of the eight, how many satisfy the following formulas?

1. p
2. $p \lor q$
3. $p \lor q \lor r$
4. $p \lor \neg p \lor r$

Exercise: 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 3.25

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

b. Are there other universal connectives?

$\neg = (F \overline{\wedge} F)$

$\vee = (F \overline{\wedge} F) \overline{\wedge} (G \overline{\wedge} G)$

$F \wedge G = \neg(\neg F \vee \neg G)$

Topic 3.6

Extra slides: sizes of models

Size of models

A model must assign value to all the variable, since it is a complete function.

However, we may not want to handle such an object.

In practice, we handle partial models. Often, without explicitly mentioning this.

Partial models

Let $m|_{\mathrm{Vars}(F)} : \mathrm{Vars}(F) \rightarrow \mathcal{B}$ and for each $p \in \mathrm{Vars}(F)$, $m|_{\mathrm{Vars}(F)}(p) = m(p)$

Theorem 3.3

If $m|_{\mathrm{Vars}(F)} = m'|_{\mathrm{Vars}(F)}$ then $m \models F$ iff $m' \models F$

Proof sketch.

The procedure to check $m \models F$ only looks at the $\mathrm{Vars}(F)$ part of m. Therefore, any extension of $m|_{\mathrm{Vars}(F)}$ will have same result either $m \models F$ or $m \not\models F$. $\square$

Definition 3.8

We will call elements of $\mathrm{Vars} \hookrightarrow \mathcal{B}$ as partial models.

Exercise 3.26

Write the above proof formally.

End of Lecture 3