CS228 : Logic for Computer Science

Module 1: Propositional Logic

Lecture 2: Parse Trees, Semantics and Problems of Interest

Instructor: S. Akshay

IIT Bombay, India

Logistics

Updates

- Safe should be up and running

Logistics

Updates

- Safe should be up and running
- Piazza is up and running - many interesting questions already!

Logistics

Updates

- Safe should be up and running
- Piazza is up and running - many interesting questions already!
- Problemsheets: This Thursday ...

Logistics

Updates

- Safe should be up and running
- Piazza is up and running - many interesting questions already!
- Problemsheets: This Thursday ...
- Optional Help Session: For a few students, already contacted!

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
- Temporal logics
- Modal logics

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
- Temporal logics
- Modal logics

Recap: Propositional Logic

1. What are propositions?
2. Atomic propositions and how to combine them
3. Syntax of propositional logic: Propositional formulae

Recall: Propositional formulas

Is this a proposition?

It is raining.

Recall: Propositional formulas

Is this a proposition?

It is raining.

Exercise 1.6: Convert these to propositional logic

1. If it rains, you will get wet.

Recall: Propositional formulas

Is this a proposition?

It is raining.

Exercise 1.6: Convert these to propositional logic

1. If it rains, you will get wet.
2. If it does not rain, you will not get wet.

Recall: Propositional formulas

Is this a proposition?

It is raining.

Exercise 1.6: Convert these to propositional logic

1. If it rains, you will get wet.
2. If it does not rain, you will not get wet.
3. If it rains, you will get wet unless you have an umbrella.

Recall: Propositional formulas

Is this a proposition?

It is raining.

Exercise 1.6: Convert these to propositional logic

1. If it rains, you will get wet.
2. If it does not rain, you will not get wet.
3. If it rains, you will get wet unless you have an umbrella.
4. If it rains, you will get wet or you won't get wet.

Recall: Propositional formulas

Is this a proposition?

It is raining.

Exercise 1.6: Convert these to propositional logic

1. If it rains, you will get wet.
2. If it does not rain, you will not get wet.
3. If it rains, you will get wet unless you have an umbrella.
4. If it rains, you will get wet or you won't get wet.

Topic 2.1

Parse Trees

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

$F \in$ **Prop** if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in$ **AP**.

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

$F \in \mathbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \mathbf{AP}$.

Representation of Propositional formulas as strings

- Any $F \in \mathbf{Prop}$ is a string over **AP** and logical connectives

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

$F \in \mathbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \mathbf{AP}$.

Representation of Propositional formulas as strings

- Any $F \in \mathbf{Prop}$ is a string over **AP** and logical connectives
- Two issues:
  1. Not all strings are formulas. e.g., $(\neg)()pq \rightarrow$

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

$F \in \textbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \textbf{AP}$.

Representation of Propositional formulas as strings

- Any $F \in \textbf{Prop}$ is a string over **AP** and logical connectives
- Two issues:
  1. Not all strings are formulas. e.g., $(\neg)()pq \rightarrow$
  2. If we don't bracket properly, we cannot parse it uniquely!

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

$F \in$ **Prop** if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in$ **AP**.

Representation of Propositional formulas as strings

- Any $F \in$ **Prop** is a string over **AP** and logical connectives
- Two issues:
  1. Not all strings are formulas. e.g., $(\neg)()pq \rightarrow$
  2. If we don't bracket properly, we cannot parse it uniquely!
     - A partial solution : Preference order $\neg$ then $\vee, \wedge$ then $\rightarrow, \leftrightarrow$. So $\neg p \vee q \rightarrow r$ is to be read as $((\neg p \vee q) \rightarrow r)$.

Representing Propositional Formulae

A propositional formula is a finite string containing symbols in **AP** and logical connectives.

$F \in$ **Prop** if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in$ **AP**.

Representation of Propositional formulas as strings

- Any $F \in$ **Prop** is a string over **AP** and logical connectives
- Two issues:
  1. Not all strings are formulas. e.g., $(\neg)()pq \rightarrow$
  2. If we don't bracket properly, we cannot parse it uniquely!
     - A partial solution : Preference order $\neg$ then $\vee, \wedge$ then $\rightarrow, \leftrightarrow$. So $\neg p \vee q \rightarrow r$ is to be read as $((\neg p \vee q) \rightarrow r)$.
- A better solution: Tree representation!

Representation as Parse trees

Definition: A parse tree of a formula $F \in$ **Prop** is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Representation as Parse trees

Definition: A parse tree of a formula $F \in \mathbf{Prop}$ is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example $$(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$$

Representation as Parse trees

Definition: A parse tree of a formula $F \in \textbf{Prop}$ is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example

$$
\begin{array}{ccc}
 & (p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3))) & \\
\swarrow & & \searrow \\
p_1 & & (\neg p_2 \leftrightarrow (p_1 \wedge p_3))
\end{array}
$$

Representation as Parse trees

Definition: A parse tree of a formula $F \in \textbf{Prop}$ is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example

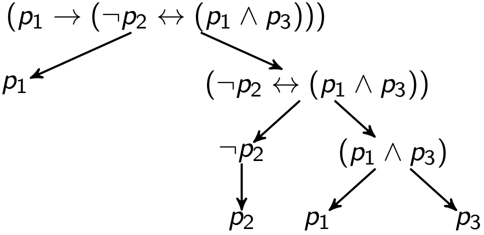
$$\begin{tikzcd}
& (p_1 \to (\neg p_2 \leftrightarrow (p_1 \wedge p_3))) \arrow[dl] \arrow[dr] & & \\
p_1 & & (\neg p_2 \leftrightarrow (p_1 \wedge p_3)) \arrow[dl] \arrow[dr] & \\
& \neg p_2 & & (p_1 \wedge p_3)
\end{tikzcd}$$

Representation as Parse trees

Definition: A parse tree of a formula $F \in$ **Prop** is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example



Theorem

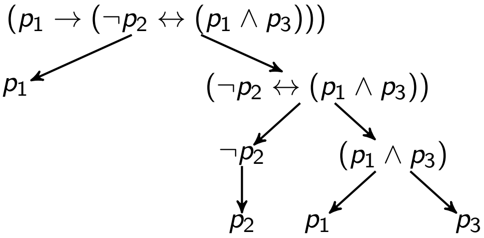
$F \in$ **Prop** iff F has a parse tree

Representation as Parse trees

Definition: A parse tree of a formula $F \in \textbf{Prop}$ is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example



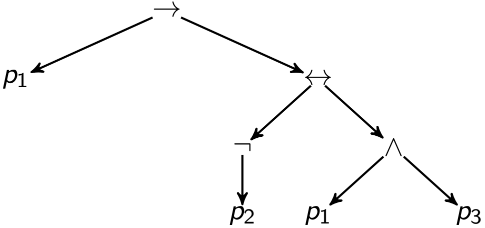
- Therefore, writing the tree allows to check that formula is well-formed.
- Formulas on internal nodes and leaves are called sub-formulas.

Representation as Parse trees

Definition: A parse tree of a formula $F \in \textbf{Prop}$ is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example



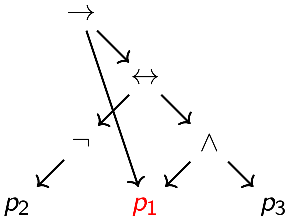
- Therefore, writing the tree allows to check that formula is well-formed.
- Formulas on internal nodes and leaves are called sub-formulas.
- Don't need brackets in the tree!

Representation as Parse trees / DAGs

Definition: A parse tree of a formula $F \in$ **Prop** is a tree s.t.,

- the root is F,
- leaves are atomic propositions, and
- each internal node is formed by applying some logical connective to its children.

Example



- Therefore, writing the tree allows to check that formula is well-formed.
- Formulas on internal nodes and leaves are called sub-formulas.
- Share leaves to get a directed-acyclic graph (DAG) instead of Tree!

Some exercises

Exercise 2.1

- Are these formulas in **Prop**? If so, draw the parse trees and dags for the formulas below:
  1. $(p \wedge (\neg q \rightarrow \neg p))$
  2. $\neg((\neg q \wedge (p \rightarrow r)) \wedge (r \rightarrow q))$
  3. $(s \vee ((\neg p) \rightarrow (\neg p)))$
  4. $(p \rightarrow (\neg q \wedge r \vee \neg p))$

Where are we?

- So far: Syntax — which strings of symbols count as valid formulas.

Where are we?

- So far: Syntax — which strings of symbols count as valid formulas.
- So far: Parse Trees — how to represent a formula's structure (and share sub-formulas via DAGs).

Where are we?

- So far: Syntax — which strings of symbols count as valid formulas.
- So far: Parse Trees — how to represent a formula's structure (and share sub-formulas via DAGs).
- Next: Semantics — given a well-formed formula, what does it actually *mean*? How do we compute its truth value?

Topic 2.2

Semantics

Semantics

What is the meaning of a formula?

- We start with truth values $\mathcal{B} := \{0, 1\}$, where we interpret 0 as False and 1 as True.

Semantics

What is the meaning of a formula?

- We start with truth values $\mathcal{B} := \{0, 1\}$, where we interpret 0 as False and 1 as True.
- Each atomic proposition $p \in$ **AP** is either True or False.

  e.g., "It is raining" is True right now (monsoon season!).

Semantics

What is the meaning of a formula?

- We start with truth values $\mathcal{B} := \{0, 1\}$, where we interpret 0 as False and 1 as True.
- Each atomic proposition $p \in \mathbf{AP}$ is either True or False.

  e.g., "It is raining" is True right now (monsoon season!).
- Each propositional formula $F \in \mathbf{Prop}$ is built from atomic propositions

  e.g., "It is raining $\wedge$ I have an umbrella" is built from two atomic propositions.

Semantics

What is the meaning of a formula?

- We start with truth values $\mathcal{B} := \{0, 1\}$, where we interpret 0 as False and 1 as True.
- Each atomic proposition $p \in$ **AP** is either True or False.

  e.g., "It is raining" is True right now (monsoon season!).
- Each propositional formula $F \in$ **Prop** is built from atomic propositions

  e.g., "It is raining $\wedge$ I have an umbrella" is built from two atomic propositions.
- So "assigning" truth values to **AP** gives "value" for every $F \in$ **Prop**.

  e.g., if both of the above are True, the formula "It is raining $\wedge$ I have an umbrella" is True too.

Semantics

What is the meaning of a formula?

- We start with truth values $\mathcal{B} := \{0, 1\}$, where we interpret 0 as False and 1 as True.
- Each atomic proposition $p \in \mathbf{AP}$ is either True or False.

  e.g., "It is raining" is True right now (monsoon season!).
- Each propositional formula $F \in \mathbf{Prop}$ is built from atomic propositions

  e.g., "It is raining $\wedge$ I have an umbrella" is built from two atomic propositions.
- So "assigning" truth values to **AP** gives "value" for every $F \in \mathbf{Prop}$.

  e.g., if both of the above are True, the formula "It is raining $\wedge$ I have an umbrella" is True too.

Definition

- An *assignment* is a map from $\mathbf{AP} \to \mathcal{B}$.

Semantics

What is the meaning of a formula?

- We start with truth values $\mathcal{B} := \{0, 1\}$, where we interpret 0 as False and 1 as True.
- Each atomic proposition $p \in$ **AP** is either True or False.

  e.g., "It is raining" is True right now (monsoon season!).
- Each propositional formula $F \in$ **Prop** is built from atomic propositions

  e.g., "It is raining $\wedge$ I have an umbrella" is built from two atomic propositions.
- So "assigning" truth values to **AP** gives "value" for every $F \in$ **Prop**.

  e.g., if both of the above are True, the formula "It is raining $\wedge$ I have an umbrella" is True too.

Definition

- An *assignment* is a map from **AP** $\to \mathcal{B}$.
- A *Valuation or a model m* of a propositional formula $F \in$ **Prop** is an assignment of each atomic proposition in F to a truth value.

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in$ **Prop**, denoted $[|F|]$, is a map from $\{0,1\}^{\mathbf{AP}} \to \{0,1\}$.

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in \textbf{Prop}$, denoted $[|F|]$, is a map from $\{0,1\}^{\textbf{AP}} \to \{0,1\}$.

For example, if $F = p \wedge q$ we have:

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in$ **Prop**, denoted $[|F|]$, is a map from $\{0,1\}^{\mathbf{AP}} \rightarrow \{0,1\}$.

For example, for arbitrary formula G and H:

G	H	$G \wedge H$
1	1	1
1	0	0
0	1	0
0	0	0

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in$ **Prop**, denoted $[|F|]$, is a map from $\{0,1\}^{\mathbf{AP}} \to \{0,1\}$.

What about $G \to H$?

G	H	$G \to H$
1	1	1
1	0	0
0	1	1
0	0	1

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in$ **Prop**, denoted $[|F|]$, is a map from $\{0,1\}^{\mathbf{AP}} \rightarrow \{0,1\}$.

What about $G \rightarrow H$?

G	H	$G \rightarrow H$
1	1	1
1	0	0
0	1	1
0	0	1

- What is the truth table for $\neg p \vee q$ and what is its link with $\rightarrow$?

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in$ **Prop**, denoted $[|F|]$, is a map from $\{0,1\}^{\mathbf{AP}} \to \{0,1\}$.

What about $G \to H$?

G	H	$G \to H$
1	1	1
1	0	0
0	1	1
0	0	1

- **Exercise 2.2:** Write down the truth tables for $\vee$, $\neg$, $\leftrightarrow$, $\oplus$.
- Note that Truth table for True is just $\begin{array}{|c|}\hline \top \\ \hline 1 \\ \hline \end{array}$ and False is $\begin{array}{|c|}\hline \bot \\ \hline 0 \\ \hline \end{array}$.

Summarizing the truth values: Truth Tables

- For each assignment of variables, i.e., propositions, we obtain a value of a formula F.
- If we write down the values for ALL possible assignments, we get the "truth table".
- Thus, the semantics of $F \in \mathbf{Prop}$, denoted $[|F|]$, is a map from $\{0,1\}^{\mathbf{AP}} \to \{0,1\}$.

What about $G \to H$?

G	H	$G \to H$
1	1	1
1	0	0
0	1	1
0	0	1

- **Exercise 2.2:** Write down the truth tables for $\vee$, $\neg$, $\leftrightarrow$, $\oplus$.
- Note that Truth table for True is just $\begin{array}{|c|}\hline \top \\ \hline 1 \\ \hline\end{array}$ and False is $\begin{array}{|c|}\hline \bot \\ \hline 0 \\ \hline\end{array}$.

How do we generalize this for an arbitrary formula F?

Evaluating a formula

Question

If $p_1 = 1$, $p_2 = 1$, $p_3 = 0$, what is $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$?

Evaluating a formula using its Parse tree

Question

If $p_1 = 1$, $p_2 = 1$, $p_3 = 0$, what is $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$?

$$(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3))) =?$$

$$p_1 = 1 \qquad (\neg p_2 \leftrightarrow (p_1 \wedge p_3)) =?$$

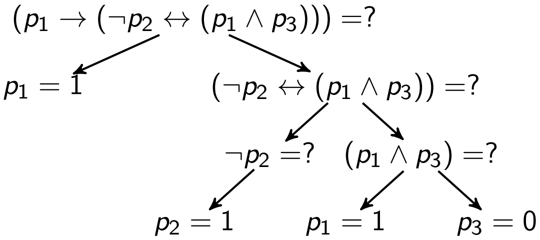
$$\neg p_2 =? \qquad (p_1 \wedge p_3) =?$$

$$p_2 = 1 \qquad p_1 = 1 \qquad p_3 = 0$$

Evaluating a formula using its Parse tree

Question

If $p_1 = 1$, $p_2 = 1$, $p_3 = 0$, what is $(p_1 \to (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$?

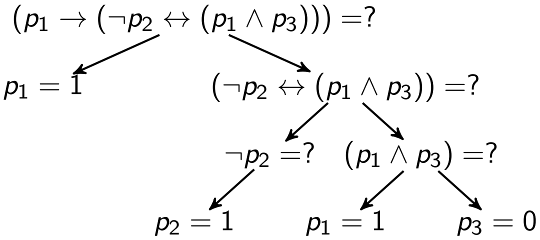


- For a given formula F and a given assignment, how long does it take to evaluate it?

Evaluating a formula using its Parse tree

Question

If $p_1 = 1$, $p_2 = 1$, $p_3 = 0$, what is $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$?

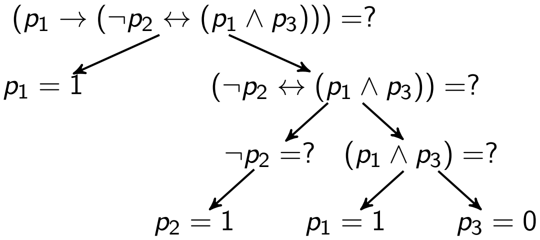


- For a given formula F and a given assignment, how long does it take to evaluate it? A. Linear time in size of the tree!

Evaluating a formula using its Parse tree

Question

If $p_1 = 1$, $p_2 = 1$, $p_3 = 0$, what is $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$?



- For a given formula F and a given assignment, how long does it take to evaluate it? A. Linear time in size of the tree!
- But what is the size of the tree in worst case?

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

p_1	p_2	p_3	$(p_1$	$\rightarrow$	$(\neg$	p_2	$\leftrightarrow ($	p_1	$\wedge$	p_3	$)))$
0	0	0	0			0		0		0	
0	0	1	0			0		0		1	
0	1	0	0			1		0		0	
0	1	1	0			1		0		1	
1	0	0	1			0		1		0	
1	0	1	1			0		1		1	
1	1	0	1			1		1		0	
1	1	1	1			1		1		1	

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

p_1	p_2	p_3	$(p_1$	$\rightarrow$	$(\ \neg$	p_2	$\leftrightarrow ($	p_1	$\wedge$	$p_3\ )))$
0	0	0	0			0		0	0	0
0	0	1	0			0		0	0	1
0	1	0	0			1		0	0	0
0	1	1	0			1		0	0	1
1	0	0	1			0		1	0	0
1	0	1	1			0		1	1	1
1	1	0	1			1		1	0	0
1	1	1	1			1		1	1	1

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

p_1	p_2	p_3	$(p_1$	$\rightarrow$	$(\neg$	p_2	$\leftrightarrow ($	p_1	$\wedge$	$p_3)))$
0	0	0	0		1	0		0	0	0
0	0	1	0		1	0		0	0	1
0	1	0	0		0	1		0	0	0
0	1	1	0		0	1		0	0	1
1	0	0	1		1	0		1	0	0
1	0	1	1		1	0		1	1	1
1	1	0	1		0	1		1	0	0
1	1	1	1		0	1		1	1	1

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

p_1	p_2	p_3	$(p_1$	$\rightarrow$	$(\neg$	p_2	$\leftrightarrow ($	p_1	$\wedge$	$p_3))) $
0	0	0	0		1	0	0	0	0	0
0	0	1	0		1	0	0	0	0	1
0	1	0	0		0	1	1	0	0	0
0	1	1	0		0	1	1	0	0	1
1	0	0	1		1	0	0	1	0	0
1	0	1	1		1	0	1	1	1	1
1	1	0	1		0	1	1	1	0	0
1	1	1	1		0	1	0	1	1	1

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

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

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \land p_3)))$

p_1	p_2	p_3	$(p_1$	$\rightarrow$	$(\neg$	p_2	$\leftrightarrow ($	p_1	$\land$	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

- Can think of it as going up the Parse tree.
- So we are applying an inductive evaluation!

Semantics of Propositional formula as Truth Tables

Consider: $(p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$

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

- Can think of it as going up the Parse tree.
- So we are applying an inductive evaluation!
  - $[|F \wedge G|] = 1$ iff $[|F|] = 1$ and $[|G|] = 1$
  - $[|\neg G|] = 1$ iff $[|G|] = 0$
  - Complete the rest!

Back to the example beyond good and evil

- In a country, every person is either good or evil (i.e., NOT good).
- Good people always tell the truth and evil people always lie.
- You meet two citizens A and B. A says, "I am evil or B is good". What are A and B?

Back to the example beyond good and evil

- In a country, every person is either good or evil (i.e., NOT good).
- Good people always tell the truth and evil people always lie.
- You meet two citizens A and B. A says, "I am evil or B is good". What are A and B?

Using propositional logic

- Step 1: Introduce atomic propositions: p_A= A is good, p_B= B is good.
- Step 2: Encode puzzle using the propositional formula: $F = \underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$

Back to the example beyond good and evil

- In a country, every person is either good or evil (i.e., NOT good).
- Good people always tell the truth and evil people always lie.
- You meet two citizens A and B. A says, "I am evil or B is good". What are A and B?

Using propositional logic

- Step 1: Introduce atomic propositions: p_A= A is good, p_B= B is good.
- Step 2: Encode puzzle using the propositional formula: $F = \underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$

Exercise 2.3

Write out the Truth Table for F

Back to the example beyond good and evil

- In a country, every person is either good or evil (i.e., NOT good).
- Good people always tell the truth and evil people always lie.
- You meet two citizens A and B. A says, "I am evil or B is good". What are A and B?

Using propositional logic

- Step 1: Introduce atomic propositions: p_A= A is good, p_B= B is good.
- Step 2: Encode puzzle using the propositional formula: $F = \underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$
- Step 3: Solve the puzzle: Is there a unique assignment of truth values to p_A and p_B under which $[|F|] = 1$? How do we do this?

Back to the example beyond good and evil

- In a country, every person is either good or evil (i.e., NOT good).
- Good people always tell the truth and evil people always lie.
- You meet two citizens A and B. A says, "I am evil or B is good". What are A and B?

Using propositional logic

- Step 1: Introduce atomic propositions: $p_A = A$ is good, $p_B = B$ is good.
- Step 2: Encode puzzle using the propositional formula: $F = \underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$
- Step 3: Solve the puzzle: Is there a unique assignment of truth values to p_A and p_B under which $[|F|] = 1$? How do we do this? Check for 1 in Truth Table!

Back to the example beyond good and evil

- In a country, every person is either good or evil (i.e., NOT good).
- Good people always tell the truth and evil people always lie.
- You meet two citizens A and B. A says, "I am evil or B is good". What are A and B?

Using propositional logic

- Step 1: Introduce atomic propositions: $p_A =$ A is good, $p_B =$ B is good.
- Step 2: Encode puzzle using the propositional formula: $F = \underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$
- Step 3: Solve the puzzle: Is there a unique assignment of truth values to p_A and p_B under which $[|F|] = 1$? How do we do this? Check for 1 in Truth Table!

p_A	p_B	$\neg p_A$	$\neg p_A \vee p_B$	F
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0
1	1	0	1	1

More examples

A prisoner must choose between three rooms. One room has a Lady and other two have tigers. If the prisoner chooses the room with the Lady, he wins! If he chooses a wrong room, he (probably) gets eaten by the tiger!
Each room has a sign in front of it. BUT Only one of the signs is true; the other two are false.

(**Room1**)	*The Lady is not in this room*
(**Room2**)	*The Lady is not in this room*
(**Room3**)	*The Lady is in Room 2*

Which room has the Lady?!

Exercise 2.4

1. Take the encoding of this problem you had written last lecture. Write the Truth Table for that formula.
2. Use the Truth Table to solve the problem.

Topic 2.3

Problems of Interest

Satisfiability

Definition (Satisfiability)

$F \in \textbf{Prop}$ is said to be satisfiable if

- there exists a valuation m of F such that $m(F) = 1$,

Satisfiability

Definition (Satisfiability)

$F \in$ **Prop** is said to be satisfiable if

- there exists a valuation m of F such that $m(F) = 1$, i.e., $[|F|] = 1$.

Satisfiability

Definition (Satisfiability)

$F \in$ **Prop** is said to be satisfiable if

- there exists a valuation m of F such that $m(F) = 1$, i.e., $[|F|] = 1$.
- in other words, we say F has a *satisfying* assignment.

Satisfiability

Definition (Satisfiability)

$F \in$ **Prop** is said to be satisfiable if

- there exists a valuation m of F such that $m(F) = 1$, i.e., $[|F|] = 1$.
- in other words, we say F has a *satisfying* assignment.

Examples

- Country with Good and Bad people example
- Lady and the Tiger example
- Many many more!

Satisfiability

Definition (Satisfiability)

$F \in$ **Prop** is said to be satisfiable if

- there exists a valuation m of F such that $m(F) = 1$, i.e., $[|F|] = 1$.
- in other words, we say F has a *satisfying* assignment.

Examples

- Country with Good and Bad people example
- Lady and the Tiger example
- Many many more!

How did we solve it?

Satisfiability

Definition (Satisfiability)

$F \in \textbf{Prop}$ is said to be satisfiable if

- there exists a valuation m of F such that $m(F) = 1$, i.e., $[|F|] = 1$.
- in other words, we say F has a *satisfying* assignment.

Examples

- Country with Good and Bad people example
- Lady and the Tiger example
- Many many more!

How did we solve it?Check for 1 entry.

Validity

Definition: $F \in \mathbf{Prop}$ is said to be valid if

- for all valuations m of F, we have $m(F) = 1$ i.e., $[|F|] = 1$.

Validity

Definition: $F \in$ **Prop** is said to be valid if

- for all valuations m of F, we have $m(F) = 1$ i.e., $[|F|] = 1$.
- Valid formulas are also called tautologies.

Validity

Definition: $F \in \textbf{Prop}$ is said to be valid if

- for all valuations m of F, we have $m(F) = 1$ i.e., $[|F|] = 1$.
- Valid formulas are also called tautologies.

Example: $(p \vee q)$ is not valid but $(p \vee \neg p)$ is valid.

Validity

Definition: $F \in \mathbf{Prop}$ is said to be valid if

- for all valuations m of F, we have $m(F) = 1$ i.e., $[|F|] = 1$.
- Valid formulas are also called tautologies.
- Can we read it from the Truth Table?!

Validity

Definition: $F \in$ **Prop** is said to be valid if

- for all valuations m of F, we have $m(F) = 1$ i.e., $[|F|] = 1$.
- Valid formulas are also called tautologies.
- Can we read it from the Truth Table?! Is $G = (p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$ valid?

Validity

Definition: $F \in \mathbf{Prop}$ is said to be valid if

- for all valuations m of F, we have $m(F) = 1$ i.e., $[|F|] = 1$.
- Valid formulas are also called tautologies.
- Can we read it from the Truth Table?! Is $G = (p_1 \rightarrow (\neg p_2 \leftrightarrow (p_1 \wedge p_3)))$ valid?
- Just check if all entries in col for G are 1.

p_1	p_2	p_3	G
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Theorem

Let $F \in \mathbf{Prop}$. F is satisfiable iff $\neg F$ is not valid.

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Theorem

Let $F \in \mathbf{Prop}$. F is satisfiable iff $\neg F$ is not valid.

Proof:

- ($\implies$) Suppose F is sat. By defn, $\exists$ valuation m of F such that $m(F) = 1$, but then $m(\neg F) = 0$. That is, $\neg F$ is not valid.

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Theorem

Let $F \in \textbf{Prop}$. F is satisfiable iff $\neg F$ is not valid.

Proof:

- ($\Longrightarrow$) Suppose F is sat. By defn, $\exists$ valuation m of F such that $m(F) = 1$, but then $m(\neg F) = 0$. That is, $\neg F$ is not valid.
- ($\Longleftarrow$) Conversely, suppose $\neg F$ is not valid.

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Theorem

Let $F \in \mathbf{Prop}$. F is satisfiable iff $\neg F$ is not valid.

Proof:

- ($\implies$) Suppose F is sat. By defn, $\exists$ valuation m of F such that $m(F) = 1$, but then $m(\neg F) = 0$. That is, $\neg F$ is not valid.
- ($\impliedby$) Conversely, suppose $\neg F$ is not valid. There must exist valuation m' of $\neg F$ such that $m'(\neg F) = 0$.

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Theorem

Let $F \in \textbf{Prop}$. F is satisfiable iff $\neg F$ is not valid.

Proof:

- ($\implies$) Suppose F is sat. By defn, $\exists$ valuation m of F such that $m(F) = 1$, but then $m(\neg F) = 0$. That is, $\neg F$ is not valid.
- ($\impliedby$) Conversely, suppose $\neg F$ is not valid. There must exist valuation m' of $\neg F$ such that $m'(\neg F) = 0$. Then $m'(F) = 1$, which implies that F is satisfiable.

Link between Satisfiability and Validity

What is the link between Satisfiability and Validity?

Theorem

Let $F \in \textbf{Prop}$. F is satisfiable iff $\neg F$ is not valid.

Proof:

- ($\implies$) Suppose F is sat. By defn, $\exists$ valuation m of F such that $m(F) = 1$, but then $m(\neg F) = 0$. That is, $\neg F$ is not valid.
- ($\impliedby$) Conversely, suppose $\neg F$ is not valid. There must exist valuation m' of $\neg F$ such that $m'(\neg F) = 0$. Then $m'(F) = 1$, which implies that F is satisfiable.

In other words, if you have an algorithm to check for Satisfiability of propositional formulas, you can use it to check for validity of formulas and vice versa!

Semantic Entailment

Definition: Consider propositional formulas F, G over $\mathbf{AP}$.

Whenever F evaluates to 1, G also evaluates to 1 — we say, $F \models G$.

Semantic Entailment

Definition: Consider propositional formulas F, G over $\mathbf{AP}$.

Whenever F evaluates to 1, G also evaluates to 1 — we say, $F \models G$.
That is, for all assignments m of $\mathbf{AP}$, if $m(F) = 1$ then $m(G) = 1$.

Semantic Entailment

Definition: Consider propositional formulas F, G over **AP**.

Whenever F evaluates to 1, G also evaluates to 1 — we say, $F \models G$.
That is, for all assignments m of **AP**, if $m(F) = 1$ then $m(G) = 1$.

Exercise 2.5: Do the following hold? Write True/False

- $p \wedge q \models p$
- $p \vee q \models q$

Semantic Entailment

Definition: Consider propositional formulas F, G over **AP**.

Whenever F evaluates to 1, G also evaluates to 1 — we say, $F \models G$.
That is, for all assignments m of **AP**, if $m(F) = 1$ then $m(G) = 1$.

Exercise 2.5: Do the following hold? Write True/False

- $p \wedge q \models p$
- $p \vee q \models q$

Definition (Semantic Equivalence)

We say that F and G are semantically equivalent, denoted $F \equiv G$, if $F \models G$ and $G \models F$.

Semantic Entailment

Definition: Consider propositional formulas F, G over **AP**.

Whenever F evaluates to 1, G also evaluates to 1 — we say, $F \models G$.
That is, for all assignments m of **AP**, if $m(F) = 1$ then $m(G) = 1$.

Exercise 2.5: Do the following hold? Write True/False

- $p \wedge q \models p$
- $p \vee q \models q$

Definition (Semantic Equivalence)

We say that F and G are semantically equivalent, denoted $F \equiv G$, if $F \models G$ and $G \models F$.
Example: $(p \to q) \equiv (\neg p \vee q)$.