Propositional Logic: Semantics (1/3)

CS402, Spring 2016

Shin Yoo

Overview

- Boolean Operators
- Propositional Formulas
- Interpretations

Propositions: a proposition is a declarative sentence. That is, it *can* be declared to be true or false. Examples:

- The sum of the numbers 3 and 5 is equal to 8.
- Jane reacted violently to Jack's accusations.
- Every even natural number greater than 2 is the sum of two prime numbers.
- All Martians like pepperoni on their pizza.

Propositionals are *atomic* and *indecomposable*. We use distinct symbols, $p.q.r. \ldots$, to represent propositions.

Boolean Operators: since propositions are of Boolean type, there are 2^{2^n} n-ary Boolean operators. Each of the n operands can be either true or false, resulting in 2^n Boolean tuples of operands. For each of 2^n tuples, the result of the operation can again be true or false. Hence 2^{2^n}.

For example, the following is the all possible unary Boolean opeartors, $o_1, \ldots, o_4$.

x	o_1	o_2	o_3	o_4
T	T	T	F	F
F	T	F	T	F

Operators o_1 and o_4 are constant, and do not operate on the operand; o_2 is the identity operator. Only o_3 is nontrivially interesting, and is called *negation*.

Binary Boolean Operators: there are 16 binary Boolean operators.

x_1	x_2	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8
T	T	T	T	T	T	T	T	T	T
T	F	T	T	T	T	F	F	F	F
F	T	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	F	T	F

x_1	x_2	o_9	o_{10}	o_{11}	o_{12}	o_{13}	o_{14}	o_{15}	o_{16}
T	T	F	F	F	F	F	F	F	F
T	F	T	T	T	T	F	F	F	F
F	T	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	F	T	F

Trivial operators: o_1 and o_{16} (constant), o_4 and o_6 (projection), o_{11} and o_{13} (negated projection).

Interesting Operators

op	name	symbol	op	name	symbol
o_2	disjunction	$\vee$	o_{15}	nor	$\downarrow$
o_8	conjunction	$\wedge$	o_9	nand	$\uparrow$
o_5	implication	$\rightarrow$	o_{12}		
o_3	reverse implication	$\leftarrow$	o_{14}		
o_7	equivalence	$\leftrightarrow$	o_{10}	exclusive or	$\oplus$

x	y	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\oplus$	$\uparrow$	$\downarrow$
T	T	T	T	T	T	F	F	F
T	F	F	T	F	F	T	T	F
F	T	F	T	T	F	T	T	F
F	F	F	F	T	T	F	T	T

Materialistic Implication: while $p \to q$ is often read "if p then q", it does not mean *causation*, i.e. it does not mean that p caused q. It only means "if p then q" such that $p \to q$ is false only when p is true but q is false (recall the truth table).

Materialistic Implication: while $p \rightarrow q$ is often read "if p then q", it does not mean *causation*, i.e. it does not mean that p caused q. It only means "if p then q" such that $p \rightarrow q$ is false only when p is true but q is false (recall the truth table).

But this also means that $p \rightarrow q$ is equivalent to $\neg p \vee q$.

Materialistic Implication: while $p \to q$ is often read "if p then q", it does not mean *causation*, i.e. it does not mean that p caused q. It only means "if p then q" such that $p \to q$ is false only when p is true but q is false (recall the truth table).

But this also means that $p \to q$ is equivalent to $\neg p \vee q$.
"2 is an odd number $\to$ 2 is an even number" is true.

Materialistic Implication: while $p \to q$ is often read "if p then q", it does not mean *causation*, i.e. it does not mean that p caused q. It only means "if p then q" such that $p \to q$ is false only when p is true but q is false (recall the truth table).

But this also means that $p \to q$ is equivalent to $\neg p \vee q$.
"2 is an odd number $\to$ 2 is an even number" is true.

The more philosophical branch of logic still has a problem with this. Outside mathematics, it is still easy to accept that when (p, q) is (T, F), $p \to q$ is also false. For cases $(T, T), (F, T)$ and (F, F), different accounts of the relationship accept that $p \to q$ is *sometimes* true, but they deny that the conditional is always true in each of these cases.

Redundancy: the first five binary operators ($\vee, \wedge, \rightarrow, \leftarrow, \leftrightarrow$) can all be defined in terms of any one of them plus negation ($\neg$). For example:

x	y	$x \wedge y$	$\neg y$	$x \rightarrow \neg y$	$\neg(x \rightarrow \neg y)$
T	T	T	F	F	T
T	F	F	T	T	F
F	T	F	F	T	F
F	F	F	T	T	F

x	y	$x \vee y$	$\neg x$	$\neg x \rightarrow y$
T	T	T	F	T
T	F	T	F	T
F	T	T	T	T
F	F	F	T	F

Redundancy: the choice of an interesting set of operators depends on the application.

- In digital circuit design, NAND($\uparrow$), NOR($\downarrow$), and NOT($\neg$) are commonly used to represent all Boolean formulas, mainly because these are more straightforward to implement at the physical, transistor level.
- In mathematics, we are generally interested in one-way logical deductions (from axioms to their implications), so we choose implication and negation.

Definition 1 (2.1)

Propositional Formula: a formula *fml* $\in \mathcal{F}$ is a word that can be derived from the following grammar, starting from the initial non-terminal *fml*:

1. $fml ::= p$ for any $p \in P$
2. $fml ::= \neg fml$
3. $fml ::= fml\ op\ fml$ where $op \in \{\vee, \wedge, \leftarrow, \rightarrow, \leftrightarrow, \downarrow, \uparrow, \oplus\}$

Each derivation of a formula from a grammar can be represented by a derivation tree that displays the application of the grammar rules to the non-terminals.

- Non-terminals: symbols that occur on the left-hand side of a rule
- Terminals: symbols that occur on only the right-hand side of a rule

From the derivation tree we can obtain a formation tree by replacing an fml non-terminal by the child that is an operator or an atom.

Derivation of $p \to q \leftrightarrow \neg p \to \neg q$ using grammar rules.

1. *fml*

Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$

Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$

Derivation of $p \to q \leftrightarrow \neg p \to \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$
4. $p \to fml \leftrightarrow fml$

Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$
4. $p \rightarrow fml \leftrightarrow fml$
5. $p \rightarrow q \leftrightarrow fml$

Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$
4. $p \rightarrow fml \leftrightarrow fml$
5. $p \rightarrow q \leftrightarrow fml$
6. $p \rightarrow q \leftrightarrow fml \rightarrow fml$

Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$
4. $p \rightarrow fml \leftrightarrow fml$
5. $p \rightarrow q \leftrightarrow fml$
6. $p \rightarrow q \leftrightarrow fml \rightarrow fml$
7. $p \rightarrow q \leftrightarrow \neg fml \rightarrow fml$

Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$
4. $p \rightarrow fml \leftrightarrow fml$
5. $p \rightarrow q \leftrightarrow fml$
6. $p \rightarrow q \leftrightarrow fml \rightarrow fml$
7. $p \rightarrow q \leftrightarrow \neg fml \rightarrow fml$
8. $p \rightarrow q \leftrightarrow \neg p \rightarrow fml$

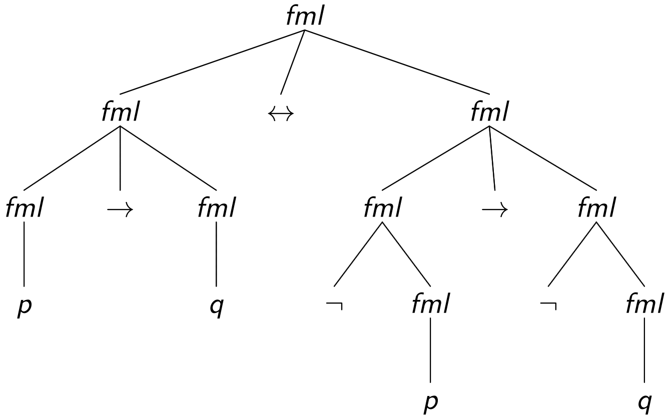
Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $fml \leftrightarrow fml$
3. $fml \leftarrow fml \leftrightarrow fml$
4. $p \rightarrow fml \leftrightarrow fml$
5. $p \rightarrow q \leftrightarrow fml$
6. $p \rightarrow q \leftrightarrow fml \rightarrow fml$
7. $p \rightarrow q \leftrightarrow \neg fml \rightarrow fml$
8. $p \rightarrow q \leftrightarrow \neg p \rightarrow fml$
9. $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg fml$

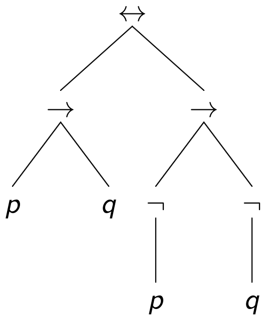
Derivation of $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ using grammar rules.

1. fml
2. $\mathit{fml} \leftrightarrow \mathit{fml}$
3. $\mathit{fml} \leftarrow \mathit{fml} \leftrightarrow \mathit{fml}$
4. $p \rightarrow \mathit{fml} \leftrightarrow \mathit{fml}$
5. $p \rightarrow q \leftrightarrow \mathit{fml}$
6. $p \rightarrow q \leftrightarrow \mathit{fml} \rightarrow \mathit{fml}$
7. $p \rightarrow q \leftrightarrow \neg \mathit{fml} \rightarrow \mathit{fml}$
8. $p \rightarrow q \leftrightarrow \neg p \rightarrow \mathit{fml}$
9. $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg \mathit{fml}$
10. $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$

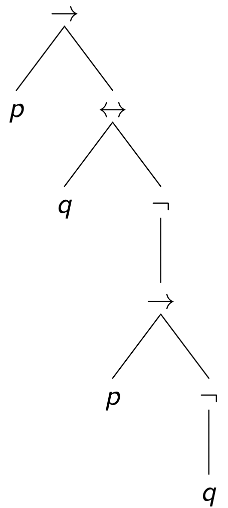
Derivation Tree: represents how non-terminals are expanded using which rules.



Formation Tree: shows the structure of the formula $p \to q \leftrightarrow \neg p \to \neg q$.



OK.



???

Removing Ambiguity: formation trees are unique, linear representation such as $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ are not. There are a few ways to resolve this ambiguity.

- **Polish Notation:** essentially, formulate linear representation by visiting the formation tree depth-first preorder (i.e. starting from the root, visit the current node, visit the left subtree, visit the right subtree, recursively).
  - $\leftrightarrow \rightarrow pq \rightarrow \neg p \neg q$
  - $\rightarrow p \leftrightarrow q \neg \rightarrow \neg p \neg q$
- **Use parentheses:** change the grammar slightly so that $fml ::= p$ for any $p \in P$, $fml ::= (\neg fml)$, and $fml ::= (fml\ op\ fml) \ldots$, etc.
  - $((p \rightarrow q) \leftrightarrow ((\neg p) \rightarrow (\neg q)))$
  - $(p \rightarrow (q \leftrightarrow (\neg(p \rightarrow (\neg q)))))$
- Define precedence and associativity: parentheses are needed only when the formula deviates from the precedence.

Removing Ambiguity: formation trees are unique, linear representation such as $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$ are not. There are a few ways to resolve this ambiguity.

- Define **precedence** and **associativity**: parentheses are needed only when the formula deviates from the precedence. We naturally recognize $a * b * c + d * e$ as $(((a * b) * c) + (d * e))$. Similarly.
  - From high to low precedence: $\neg, \wedge, \uparrow, \vee, \downarrow, \rightarrow, \leftrightarrow$
  - Assume right associativity, i.e. $a \vee b \vee c$ means $(a \vee (b \vee c))$.

  With minimal use of parentheses, the previous two formulation trees can be represented as:
  - $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$
  - $p \rightarrow (q \leftrightarrow \neg(p \rightarrow \neg q))$

Structural induction

Theorem 1 (2.5)

Theorem 2.5. To show property(A) for all formulas $A \in \mathcal{F}$, it suffices to show:

- *Base case: property(p) holds for all atoms $p \in \mathcal{P}$*
- *Induction step:*
  - *Assuming property(A), the property$(\neg A)$ holds.*
  - *Assuming property(A_1) and property(A_2), then property$(A_1$ op $A_2)$ hold, for each of the binary operators.*

Exercise: Prove that every propositional formula can be equivalently expressed using only $\uparrow$.

Interpretations

Definition 2 (2.6)

An assignment ν is a function $\nu : \mathcal{P} \rightarrow \{T, F\}$.

- In other words, ν assigns one of the truth values, T or F to every atom.
- From now on, we use two new syntax terms, "true" and "false", which are *syntactic tokens*.
- On the other hand, T and F are *truth values*.
- $\mathit{fml} ::= \mathit{true} | \mathit{false}$ where $\nu(\mathit{true}) = T$ and $\nu(\mathit{false}) = F$.

Combining this with inductive definition, we can extend the assignment to functions, i.e. $\nu : \mathcal{F} \rightarrow \{T, F\}$. In other words, we inductively decide whether a propositional formula is true or false. In this case, ν is called an **interpretation**.

Interpretations

Theorem 2 (2.9)

An assignment can be extended to **exactly one** *interpretation.*

Theorem 3 (2.10)

Let $\mathcal{P}' = \{p_1, \ldots, p_n\} \subseteq \mathcal{P}$ be the atoms appearing in $A \in \mathcal{F}$. Let ν_1 and ν_2 be assignments that agrees on $\mathcal{P}'$, that is, $\nu_1(p_i) = \nu_2(p_i)$ for all $p_i in \mathcal{P}'$. Then, the interpretations ν_1 and ν_2 agree on A, i.e. $\nu_1(A) = \nu_2(A)$.

Example 1 (2.7)

Let $A = (p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ and let ν the assignment such that $\nu(p) = F$ and $\nu(q) = T$, and $\nu(p_i) = T$ for all other $p_i \in \mathcal{P}$. Extend ν to an interpretation of A.

1. $\nu(p \rightarrow q) = T$

Example 1 (2.7)

Let $A = (p \to q) \leftrightarrow (\neg q \to \neg p)$ and let ν the assignment such that $\nu(p) = F$ and $\nu(q) = T$, and $\nu(p_i) = T$ for all other $p_i \in \mathcal{P}$. Extend ν to an interpretation of A.

1. $\nu(p \to q) = T$
2. $\nu(\neg q) = F$

Example 1 (2.7)

Let $A = (p \to q) \leftrightarrow (\neg q \to \neg p)$ and let ν the assignment such that $\nu(p) = F$ and $\nu(q) = T$, and $\nu(p_i) = T$ for all other $p_i \in \mathcal{P}$. Extend ν to an interpretation of A.

1. $\nu(p \to q) = T$
2. $\nu(\neg q) = F$
3. $\nu(\neg p) = T$

Example 1 (2.7)

Let $A = (p \to q) \leftrightarrow (\neg q \to \neg p)$ and let ν the assignment such that $\nu(p) = F$ and $\nu(q) = T$, and $\nu(p_i) = T$ for all other $p_i \in \mathcal{P}$. Extend ν to an interpretation of A.

1. $\nu(p \to q) = T$
2. $\nu(\neg q) = F$
3. $\nu(\neg p) = T$
4. $\nu(\neg q \to \neg p) = T$

Example 1 (2.7)

Let $A = (p \to q) \leftrightarrow (\neg q \to \neg p)$ and let ν the assignment such that $\nu(p) = F$ and $\nu(q) = T$, and $\nu(p_i) = T$ for all other $p_i \in \mathcal{P}$. Extend ν to an interpretation of A.

1. $\nu(p \to q) = T$
2. $\nu(\neg q) = F$
3. $\nu(\neg p) = T$
4. $\nu(\neg q \to \neg p) = T$
5. $\nu((p \to q) \leftrightarrow (\neg q \to \neg p)) = T$

Example 1 (2.7)

Let $A = (p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ and let ν the assignment such that $\nu(p) = F$ and $\nu(q) = T$, and $\nu(p_i) = T$ for all other $p_i \in \mathcal{P}$. Extend ν to an interpretation of A.

1. $\nu(p \rightarrow q) = T$
2. $\nu(\neg q) = F$
3. $\nu(\neg p) = T$
4. $\nu(\neg q \rightarrow \neg p) = T$
5. $\nu((p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)) = T$

Example 2 (2.8)

$\nu(p \rightarrow (q \rightarrow p)) = T$, but $\nu((p \rightarrow q) \rightarrow p) = F$. This shows that $p \rightarrow q \rightarrow p$ is ambiguous.