3- Boolean Semantics (Truth Tables)


York University
Department of Computer Science and Engineering
Overview

• Truth Tables, states
• Tautologies, contradictions, and satisfiable formulae
• Tautological implication
• Substitution
• Schemata
Boolean Semantics

Value of atomic formula

• Truth values: \( t \) (true), \( f \) (false)
  – Note these symbols are not in Boolean Alphabet and they never appear in a wff

• **Definition.** A state \( v \) is a function
  – that assigns the value \( f \) or \( t \) to each Boolean variable, while
  – it assigns necessarily the value \( f \) to the constant \( \bot \) and
  – it assigns necessarily the value \( t \) to the constant \( \top \).

• Notes:
  – This definition gives values to atomic formula only.
  – A state \( v \) is an infinite table.
Boolean Semantics
Value of connectives

• **Definition. (Truth Tables)** Tables describing five functions, called Boolean functions, that take inputs from the set \{f, t\} and return values in the same set.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>$F_\neg(x)$</td>
<td>$F_\lor(x, y)$</td>
<td>$F_\land(x, y)$</td>
<td>$F_\rightarrow(x, y)$</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>
Boolean Semantics

Value of a formula

• Definition. (Value of a formula in a state $v$)

\[
v(p) = \text{whatever we originally assigned to } p; \ t \text{ or } f
\]
\[
v(\top) = t
\]
\[
v(\bot) = f
\]
\[
v\left( (\neg A) \right) = F_{\neg}\left( v(A) \right)
\]
\[
v\left( (A \land B) \right) = F_{\land}\left( v(A), v(B) \right)
\]
\[
v\left( (A \lor B) \right) = F_{\lor}\left( v(A), v(B) \right)
\]
\[
v\left( (A \rightarrow B) \right) = F_{\rightarrow}\left( v(A), v(B) \right)
\]
\[
v\left( (A \equiv B) \right) = F_{\equiv}\left( v(A), v(B) \right)
\]
Boolean meta-variable

• \( p \) (bold \( p \)) is a meta-variable or syntactic variable, i.e. a symbol outside the Boolean alphabet, which we use to refer to any variable.

• So instead of saying
  \[ \nu(p) = \text{whatever we originally assigned to } p \]
  \[ \nu(q) = \text{whatever we originally assigned to } q \]
  \[ \nu(p') = \text{whatever we originally assigned to } p' \]
  Etc......

We say

\[ \nu(p) = \text{whatever we originally assigned to } p \]
Infinite vs. finite tables

• A state \( v \) is by definition an infinite table.

• But intuitively, the value of a formula \( A \) in any state \( v \) should depend only on the values of the variables that occur in \( A \).

• For any formula \( A \), we therefore truncate the state into a finite “appropriate” table.
Occurrence of a variable

• **Definition.** (Occurrence of a variable)
  
  – Atomic case: $p$ occurs in $p$, and $p$ does not occur in $q, \top, \bot$
    (where $q$ is a different variable from $p$)
  
  – $p$ occurs in $(\rightarrow A)$ iff it occurs in $A$

  – $p$ occurs in $(A \circ B)$, $\circ \in \{\land, \lor, \rightarrow, \equiv\}$, iff it occurs in $A$ or $B$ or both.

• **Proposition.** If $v$ and $v'$ are two states that agree on the variables of $A$, then $v(A) = v'(A)$. 
**Tautology/ Satisfiable/ Contradiction**

- **Definition.** A formula A is a **tautology** iff it is true (\(t\)) in all possible states. This is denoted by \(\models_{\text{taut}} A\).

- **Definition.** A formula A is **satisfiable** iff there is at least one state \(v\) where \(v(A) = t\).

- **Definition.** A formula A is **unsatisfiable** or a **contradiction** iff for every state \(v\), we have \(v(A) = f\).
More Definitions!

• We denote sets of formula by capital Greek letters, such as \( \Gamma, \Sigma, \Delta, \Theta \)

• **Definition.** A set of formula \( \Gamma \) is **satisfiable** iff there is at least **one** state \( \nu \) where for **every** formula \( A \) in \( \Gamma \), \( \nu(A) = t \). We say \( \nu \) satisfies \( \Gamma \).

• **Definition.** \( \Gamma \) **tautologically implies** \( A \) iff for **every** state \( \nu \) that satisfies \( \Gamma \), we must have \( \nu(A) = t \).
  
  – This is denoted by \( \Gamma \models_{taut} A \).
  
  – Formulae in set \( \Gamma \) are called the hypotheses or premises
  
  – \( A \) is called the conclusion
Substitution in formulae

\[ A[p := B] = \begin{cases} 
B & \text{if } A = p \\
A & \text{if } A = q \text{ (where } p \neq q), \text{ or} \\
\neg C[p := B] & \text{if } A = \top, \text{ or } A = \bot \\
C[p := B] \circ D[p := B] & \text{if } A = (\neg C) \\
\end{cases} \]

if \( A = (C \circ D) \)
Statements about substitution

• **Proposition.** For any formulae $A$ and $B$ and variable $p$, $A[p:=B]$ is a formula, in other words it is a wff.
  – Provable by induction

• **Proposition.** If $p$ does not occur in $A$, then $A[p:=B]$ is $A$ (unchanged).
  – Provable by induction
Schemata

• A **schema** is a string over the following augmented alphabet:
  
  – The set of Boolean alphabet (\(\lor\)) union
  – \{[, :=, ]\} union
  – \{A, B, C, ....\} union
  – \{p, q, ...\}
  
  \[\text{Syntactic variables}\]

• If we replace all syntactic variables in a schema with any formula or Boolean variable, we will obtain a wff. This formula would be an **instance of the schema**.