SC/MATH 1090

3- Boolean Semantics (Truth Tables)

Ref: G. Tourlakis, *Mathematical Logic*, John Wiley & Sons, 2008.

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.

x	y	$F_{\neg}(x)$	$F_{\vee}(x,y)$	$F_{\wedge}(x,y)$	$F_{\rightarrow}(x,y)$	$F_{\equiv}(x,y)$
f	f	t	f	f	t	t
f	t	t	t	f	t	f
t	f	f	t	f	f	f
t	t	f	t	t	t	t

Boolean Semantics
Value of a formula

- **Definition. (Value of a formula in a state *v*)**

$$
\begin{aligned}
v(\mathbf{p}) &= \text{whatever we originally assigned to } \mathbf{p}\text{; } \mathbf{t} \text{ or } \mathbf{f} \\
v(\top) &= \mathbf{t} \\
v(\bot) &= \mathbf{f} \\
v\Big((\neg A)\Big) &= F_{\neg}\Big(v(A)\Big) \\
v\Big((A \wedge B)\Big) &= F_{\wedge}\Big(v(A), v(B)\Big) \\
v\Big((A \vee B)\Big) &= F_{\vee}\Big(v(A), v(B)\Big) \\
v\Big((A \rightarrow B)\Big) &= F_{\rightarrow}\Big(v(A), v(B)\Big) \\
v\Big((A \equiv B)\Big) &= F_{\equiv}\Big(v(A), v(B)\Big)
\end{aligned}
$$

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 <u>any variable</u>.

- So instead of saying

  v(p) = whatever we originally assigned to p

  v(q) = whatever we originally assigned to q

  v(p') = whatever we originally assigned to p'

  Etc……

  We say

  v(**p**) = 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**, $\top$, $\perp$ (where **q** is a different variable from **p**)
  - **p** occurs in ($\neg$ A) iff it occurs in A
  - **p** occurs in (A o B), o$\in$ {$\wedge$, $\vee$, $\rightarrow$, $\equiv$}, 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 Γ is **satisfiable** iff there is at least one state *v* where for every formula A in Γ, *v*(A)=**t**. We say *v* satisfies Γ.
- **Definition**. Γ **tautologically implies** A iff for every state v that satisfies Γ, we must have *v*(A)=**t**.
  - This is denoted by $\Gamma \models_{\text{taut}}$ A.
  - Formulae in set Γ are called the hypotheses or premises
  - A is called the conclusion

Substitution in formulae

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

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 ($\mathcal{V}$) union
  - { [, :=,]} union
  - {A, B, C, ...} union
  - {**p**, **q**, ...}

  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**.