Non-classical logics

Lecture 5: Many-valued logics

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Until now

Classical logic

- Propositional logic (Syntax, Semantics)
- First-order logic (Syntax, Semantics)

Proof methods (resolution, tableaux)

From now on: Non-Classical logics

- Many-valued logic (finitely-valued; infinitely-valued)

  Syntax, semantics, Automated proof methods (resolution, tableaux)

  Reduction to classical logic

- Modal logics (also description logics, dynamic logic)

  Syntax, semantics, Automated proof methods (resolution, tableaux)

  Reduction to classical logic

- Temporal logic (Linear time; branching time)

  Syntax, semantics, Model checking

From now on: Non-Classical logics

- Many-valued logic (finitely-valued; infinitely-valued)

  Syntax, semantics, Automated proof methods (resolution, tableaux)

  Reduction to classical logic

- Modal logics (also description logics, dynamic logic)

  Syntax, semantics, Automated proof methods (resolution, tableaux)

  Reduction to classical logic

- Temporal logic (Linear time; branching time)

  Syntax, semantics, Model checking

Many-valued logic

- Introduction
- Many-valued logics

  3-valued logic

  finitely-valued logic

  fuzzy logic
- Automated theorem proving (resolution, tableaux)
- Reduction to classical logic

History and Motivation

Many-valued logics were introduced to model undefined or vague information

History and Motivation

Jan Łukasiewicz
Began to create systems of many-valued logic in 1920, using a third value "possible" to deal with Aristotle's paradox of the sea battle.

- Jan Łukasiewicz:
  "On 3-valued logic" (Polish) Ruch Filozoficzny, Vol. 5, 1920.

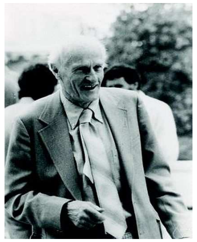
Later, **Jan Łukasiewicz** and **Alfred Tarski** together formulated a logic on n truth values where $n \geq 2$.

- Jan Łukasiewicz:
  Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. Comptes rendus des séance de la Societé des Sciences et des Lettres de Varsovie, Classe III, Vol .23, 1930.
- S. McCall:
  Polish Logic: 1920–1939. Oxford University Press, 1967.

History

Emile L. Post
Introduced (in 1921) the formulation of additional truth degrees with $n \geq 2$ where n is the number of truth values (starting mainly from algebraic considerations).

- Emil Post:
  Introduction to a general theory of elementary propositions. American J. of Math., Vol. 43, 1921.



S. C. Kleene:

Introduced a 3-valued logic in order to express the fact that some recursive functions might be undefined.

Applications of many-valued logic

- independence proofs
- modeling undefined function and predicate values (program verification)
- semantic of natural languages
- theory of logic programming: declarative description of operational semantics of negation
- modeling of electronic circuits
- modeling vagueness and uncertainly
- shape analysis (program verification)

Literature

- J. B. Rosser, A. R. Turquette: Many-valued Logics. North-Holland, 1952.
- N. Rescher: Many-valued Logic. McGraw-Hill, 1989.
- Alasdair Urquhart: Handbook of Philosophical Logic, vol. 3, 1986.
- Bolc und Borowik: Many-Valued Logics. Springer Verlag 1992,

Literature

- Matthias Baaz, Christian G. Fermüller:
  Resolution-Based Theorem Proving for Many valued Logics.
  J. Symb. Comput. 19(4): 353-391 (1995)
- Reiner Hähnle:
  Automated Deduction in Multiple-valued Logics.
  Clarendon Press, Oxford, 1993.
- Grzegorz Malinowski:
  Many-Valued Logics.
  Oxford Logic Guides, Vol. 25, Clarendon Press, Oxford, 1993.
- Siegfried Gottwald
  A Treatise On Many-Valued Logics. Studies in Logic and Computation,
  Vol. 9, Research Studies Press, 2001.

Literature

- Harald Ganzinger and Viorica Sofronie-Stokkermans
  Chaining techniques for automated theorem proving in many-valued logic. ISMVL 2000.

- Viorica Sofronie-Stokkermans and Carsten Ihlemann
  Automated reasoning in some local extensions of ordered structures
  Multiple-Valued Logic and Soft Computing 13(4-6): 397-414, 2007.

A motivating example

B: the sky is blue

R: it rains

U: I take my umbrella

$$(B \longrightarrow \neg R) \wedge (R \longrightarrow U) \wedge (B \longrightarrow \neg U) \wedge R$$

A motivating example

B: the sky is blue

R: it rains

U: I take my umbrella

$$(B \rightarrow \neg R) \wedge (R \rightarrow U) \wedge (B \rightarrow \neg U) \wedge R$$

Description of a situation: (partial) variable assignment $v : \Pi \rightarrow \{0, 1\}$

A	$v(A)$
B	1
R	
U	0

Truth tables in partial logic

v partial valuation.

$v \sqsubseteq v_1$: v_1 is a total variable assignment which extends v.

Example

Description of a situation:

(partial) $v : \Pi \longrightarrow \{0, 1\}$

A	$v(A)$
B	1
R	
U	0

A	$v_1(A)$	$v_2(A)$
B	1	1
R	0	1
U	0	0

$v \sqsubseteq v_1$, $v \sqsubseteq v_2$

$v(F_1 \wedge F_2) = 0$ iff for all v_1 with $v \sqsubseteq v_1$ we have $v_1(F_1 \wedge F_2) = 0$

$v(F_1 \wedge F_2) = 1$ iff for all v_1 with $v \sqsubseteq v_1$ we have $v_1(F_1 \wedge F_2) = 1$

Truth tables for partial logic

$\wedge$	1	undef	0
1	1	undef	0
undef	undef	undef	0
0	0	0	0

$\vee$	1	undef	0
1	1	1	1
undef	1	undef	undef
0	1	undef	0

F	$\neg F$
1	0
undef	undef
0	1

A motivating example

$\wedge$	1	undef	0
1	1	undef	0
undef	undef	undef	0
0	0	0	0

$\vee$	1	undef	0
1	1	1	1
undef	1	undef	undef
0	1	undef	0

F	$\neg F$
1	0
undef	undef
0	1

$$(B \rightarrow \neg R) \wedge (R \rightarrow U) \wedge (B \rightarrow \neg U) \wedge R$$

Description of a situation: (partial) variable assignment $v : \Pi \rightarrow \{0, 1\}$

A	$v(A)$
B	1
R	undef
U	0

F	$v(F)$
$\neg B \vee \neg R$	undef
$\neg R \vee U$	undef
$\neg B \vee \neg U$	1

Another example

Belnap’s 4-valued logic

This particularly interesting system of MVL was the result of research on relevance logic, but it also has significance for computer science applications. Its truth degree set may be taken as

$$M = \{\{\}, \{0\}, \{1\}, \{0, 1\}\},$$

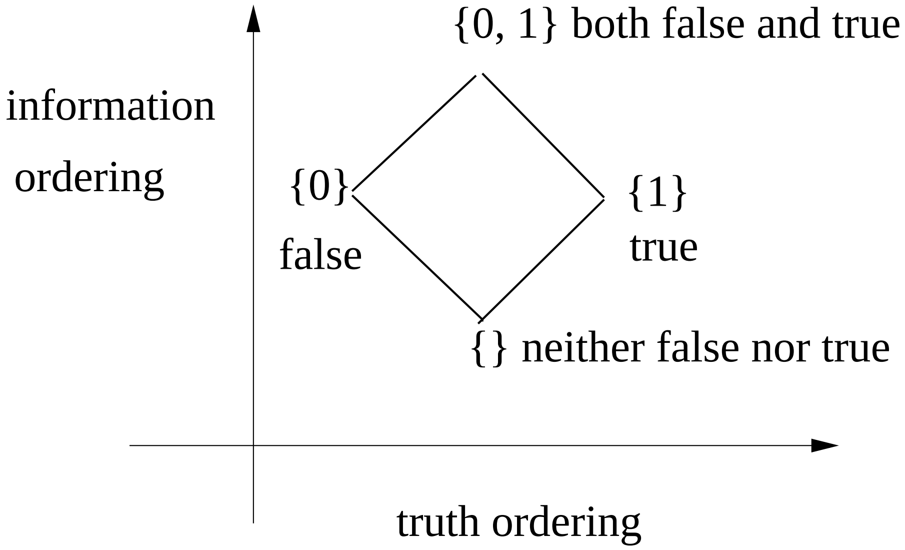
and the truth degrees interpreted as indicating (e.g. with respect to a database query for some particular state of affairs) that there is

- no information concerning this state of affairs,
- information saying that the state of affairs is false,
- information saying that the state of affairs is true,
- conflicting information saying that the state of affairs is true as well as false.

Another example

Belnap's 4-valued logic $\quad M = \{\{\}, \{0\}, \{1\}, \{0,1\}\}$

This set of truth degrees has two natural orderings:



$\wedge, \vee$: sup/inf in the truth ordering

$\sim \{\} = \{\}, \quad \sim \{0,1\} = \{0,1\}, \quad \sim \{0\} = \{1\}, \quad \sim \{1\} = \{0\}$

"Designated" values: (What we can assume to be true)

Computer science: $D = \{\{1\}\}$

Other applications (e.g. information bases): $D = \{\{1\}, \{0,1\}\}$

Many-valued logics

- Syntax
- Semantics
- Applications
- Proof theory / Methods for automated reasoning

1 Syntax

- propositional variables
- logical operations

Propositional Variables

Let Π be a set of propositional variables.

We use letters P, Q, R, S, to denote propositional variables.

Logical operators

Let $\mathcal{F}$ be a set of logical operators.

These logical operators could be the usual ones from classical logic

$$\{\neg/1, \vee/2, \wedge/2, \rightarrow /2, \leftrightarrow /2\}$$

but could also be other operations, with arbitrary arity.

Propositional Formulas

$F_\Pi^\mathcal{F}$ is the set of propositional formulas over Π defined as follows:

$$
\begin{array}{lcll}
F, G, H & ::= & c & \text{(c constant logical operator)} \\
 & \mid & P, \quad P \in \Pi & \text{(atomic formula)} \\
 & \mid & f(F_1, \ldots, F_n) & (f \in \mathcal{F} \text{ with arity } n)
\end{array}
$$

Example: Classical propositional logic

If $\mathcal{F} = \{\top/0, \bot\,/0, \neg/1, \vee/2, \wedge/2, \rightarrow\,/2, \leftrightarrow\,/2\}$ then

F_Π is the set of propositional formulas over Π is defined as follows:

$$
\begin{array}{rcll}
F, G, H & ::= & \bot & \text{(falsum)} \\
& \mid & \top & \text{(verum)} \\
& \mid & P, \quad P \in \Pi & \text{(atomic formula)} \\
& \mid & \neg F & \text{(negation)} \\
& \mid & (F \wedge G) & \text{(conjunction)} \\
& \mid & (F \vee G) & \text{(disjunction)} \\
& \mid & (F \rightarrow G) & \text{(implication)} \\
& \mid & (F \leftrightarrow G) & \text{(equivalence)}
\end{array}
$$

Semantics

We assume that a set $M = \{w_1, w_2, \dots, w_m\}$ of truth values is given.

We assume that a subset $D \subseteq M$ of designated truth values is given.

1. Meaning of the logical operators

$f \in \mathcal{F}$ with arity $n \quad \mapsto \quad f_M : M^n \to M$

(truth tables for the operations in $\mathcal{F}$)

Example 1: If $\mathcal{F}$ consists of the Boolean operations and $M = B_2 = \{0, 1\}$ then specifying the meaning of the logical operations means giving the truth tables for the operations in $\mathcal{F}$

$\neg_b$	
0	1
1	0

$\vee_b$	0	1
0	0	1
1	1	1

$\wedge_b$	0	1
0	0	0
1	0	1

Semantics

We assume that a set $M = \{w_1, \dots, w_m\}$ of truth values is given.

We assume that a subset $D \subseteq M$ of designated truth values is given.

1. Meaning of the logical operators

$f \in \mathcal{F}$ with arity $n \quad \mapsto \quad f_M : M^n \to M$

(truth tables for the operations in $\mathcal{F}$)

Example 2: If $\mathcal{F}$ consists of the operations $\{\vee, \wedge, \neg\}$ and $M = \{0, \text{undef}, 1\}$ then specifying the meaning of the logical operations means giving the truth tables for these operations e.g.

F	$\neg_u F$
1	0
undef	undef
0	1

$\wedge_u$	1	undef	0
1	1	undef	0
undef	undef	undef	0
0	0	0	0

$\vee_u$	1	undef	0
1	1	1	1
undef	1	undef	undef
0	1	undef	0

Semantics

We assume that a set $M = \{w_1, \ldots, w_m\}$ of truth values is given.

We assume that a subset $D \subseteq M$ of designated truth values is given.

1. Meaning of the logical operators

$f \in \mathcal{F}$ with arity $n \quad \mapsto \quad f_M : M^n \rightarrow M$

(truth tables for the operations in $\mathcal{F}$)

Example 2: $\mathcal{F} = \{\vee, \wedge, \neg\}$ and $M = \{\{\}, \{0\}, \{1\}, \{0, 1\}\}$. The truth tables for these operations e.g.

F	F
$\{\ \}$	$\{\ \}$
$\{0\}$	$\{1\}$
$\{1\}$	$\{0\}$
$\{0, 1\}$	$\{0, 1\}$

$\wedge_B$	$\{\ \}$	$\{0\}$	$\{1\}$	$\{0, 1\}$
$\{\ \}$	$\{\ \}$	$\{0\}$	$\{\ \}$	$\{0\}$
$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$
$\{1\}$	$\{\ \}$	$\{0\}$	$\{1\}$	$\{0, 1\}$
$\{0, 1\}$	$\{0\}$	$\{0\}$	$\{0, 1\}$	$\{0, 1\}$

$\vee_B$	$\{\ \}$	$\{0\}$	$\{1\}$	$\{0, 1\}$
$\{\ \}$	$\{\ \}$	$\{\ \}$	$\{1\}$	$\{1\}$
$\{0\}$	$\{\ \}$	$\{0\}$	$\{1\}$	$\{0, 1\}$
$\{1\}$	$\{1\}$	$\{0, 1\}$	$\{1\}$	$\{0, 1\}$
$\{0, 1\}$	$\{1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{1\}$

Semantics

We assume that a set $M = \{w_1, \ldots, w_m\}$ of truth values is given.

We assume that a subset $D \subseteq M$ of designated truth values is given.

2. The meaning of the propositional variables

A Π-valuation is a map

$$\mathcal{A} : \Pi \to M.$$

Semantics

We assume that a set $M = \{w_1, \ldots, w_m\}$ of truth values is given.

We assume that a subset $D \subseteq M$ of designated truth values is given.

3. Truth value of a formula in a valuation

Given an interpretation of the operation symbols $(M, \{f_M\}_{f \in \mathcal{F}})$ and Π-valuation $\mathcal{A} : \Pi \rightarrow M$, the function $\mathcal{A}^* : \Sigma\text{-formulas} \rightarrow M$ is defined inductively over the structure of F as follows:

$$\begin{aligned}\mathcal{A}^*(c) &= c_M (\text{for every constant operator } c \in \mathcal{F})\\ \mathcal{A}^*(P) &= \mathcal{A}(P)\\ \mathcal{A}^*(f(F_1, \ldots, F_n)) &= f_M(\mathcal{A}^*(F_1), \ldots, \mathcal{A}^*(F_n))\end{aligned}$$

For simplicity, we write $\mathcal{A}$ instead of $\mathcal{A}^*$.

Example 1: Classical logic

Given a Π-valuation $\mathcal{A} : \Pi \rightarrow B_2 = \{0,1\}$, the function $\mathcal{A}^*$: Σ-formulas $\rightarrow \{0,1\}$ is defined inductively over the structure of F as follows:

$$
\begin{aligned}
\mathcal{A}^*(\bot) &= 0 \\
\mathcal{A}^*(\top) &= 1 \\
\mathcal{A}^*(P) &= \mathcal{A}(P) \\
\mathcal{A}^*(\neg F) &= \neg_b \mathcal{A}^*(F) \\
\mathcal{A}^*(F \circ G) &= \circ_b(\mathcal{A}^*(F), \mathcal{A}^*(G)) \\
&\quad \text{with } \circ_B \text{ the Boolean function associated with } \circ \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}
\end{aligned}
$$

Example 2: Logic of undefinedness

Given a Π-valuation $\mathcal{A} : \Pi \to M = \{0, \text{undef}, 1\}$, the function $\mathcal{A}^*$: Σ-formulas $\to \{0, \text{undef}, 1\}$ is defined inductively over the structure of F as follows:

$$\mathcal{A}^*(\bot) = 0$$

$$\mathcal{A}^*(\top) = 1$$

$$\mathcal{A}^*(P) = \mathcal{A}(P)$$

$$\mathcal{A}^*(\neg F) = \neg_u(\mathcal{A}^*(F))$$

$$\mathcal{A}^*(F \vee G) = \mathcal{A}^*(F) \vee_u \mathcal{A}^*(G)$$

$$\mathcal{A}^*(F \wedge G) = \mathcal{A}^*(F) \wedge_u \mathcal{A}^*(G)$$

Example 3: Belnap's 4-valued logic

Given a Π-valuation $\mathcal{A} : \Pi \to M = \{\{\}, \{0\}, \{1\}, \{0,1\}\}$, the function $\mathcal{A}^*$: Σ-formulas $\to \{\{\}, \{0\}, \{1\}, \{0,1\}\}$ is defined inductively over the structure of F as follows:

$$\begin{aligned}
\mathcal{A}^*(\bot) &= \{0\} \\
\mathcal{A}^*(\top) &= \{1\} \\
\mathcal{A}^*(P) &= \mathcal{A}(P) \\
\mathcal{A}^*(\neg F) &= \ {}_B(\mathcal{A}^*(F)) \\
\mathcal{A}^*(F \vee G) &= \mathcal{A}^*(F) \vee_B \mathcal{A}^*(G) \\
\mathcal{A}^*(F \wedge G) &= \mathcal{A}^*(F) \wedge_B \mathcal{A}^*(G)
\end{aligned}$$

Models, Validity, and Satisfiability

$M = \{w_1, \ldots, w_m\}$ set of truth values
$D \subseteq M$ set of designated truth values
$\mathcal{A} : \Pi \rightarrow M$.

F is valid in $\mathcal{A}$ ($\mathcal{A}$ is a model of F; F holds under $\mathcal{A}$):

$$\mathcal{A} \models F \;:\Leftrightarrow\; \mathcal{A}(F) \in D$$

F is valid (or is a tautology):

$$\models F \;:\Leftrightarrow\; \mathcal{A} \models F \text{ for all } \Pi\text{-valuations } \mathcal{A}$$

F is called satisfiable iff there exists an $\mathcal{A}$ such that $\mathcal{A} \models F$.

Otherwise F is called unsatisfiable (or contradictory).

The logic $\mathcal{L}_3$

Set of truth values: $M = \{1, u, 0\}$.

Designated truth values: $D = \{1\}$.

Logical operators: $\mathcal{F} = \{\vee, \wedge, \neg, \sim\}$.

Truth tables for the operators

$\vee$	0	u	1
0	0	u	1
u	u	u	1
1	1	1	1

$\wedge$	0	u	1
0	0	0	0
u	0	u	u
1	0	u	1

$$v(F \wedge G) = \min(v(F), v(G))$$

$$v(F \vee G) = \max(v(F), v(G))$$

Under the assumption that $0 < u < 1$.

Truth tables for negations

A	$\neg A$	$\sim A$	$\sim \neg A$	$\sim\sim A$	$\neg\neg A$	$\neg \sim A$
1	0	0	1	1	1	1
u	u	1	1	0	u	0
0	1	1	0	0	0	0

Translation in natural language:

$v(A) = 1$ gdw. A is true

$v(\neg A) = 1$ gdw. A is false

$v(\sim A) = 1$ gdw. A is not true

$v(\sim \neg A) = 1$ gdw. A is not false

First-order many-valued logic

$M = \{w_1, \ldots, w_m\}$ set of truth values

$D \subseteq M$ set of designated truth values.

1. Syntax

- non-logical symbols (domain-specific)
  $\Rightarrow$ terms, atomic formulas
- logical symbols $\mathcal{F}$, quantifiers
  $\Rightarrow$ formulae

Signature

A signature

$$\Sigma = (\Omega, \Pi),$$

fixes an alphabet of non-logical symbols, where

- Ω is a set of function symbols f with arity $n \geq 0$, written f/n,
- Π is a set of predicate symbols p with arity $m \geq 0$, written p/m.

If $n = 0$ then f is also called a constant (symbol).
If $m = 0$ then p is also called a propositional variable.
We use letters P, Q, R, S, to denote propositional variables.

Variables, Terms

As in classical logic

Atoms

Atoms (also called atomic formulas) over Σ are formed according to this syntax:

$$\begin{array}{rcll} A, B & ::= & p(s_1, \dots, s_m) & ,\ p/m \in \Pi \\ \Big[& \mid & (s \approx t) & \text{(equation)} \quad \Big] \end{array}$$

In what follows we will only consider variants of first-order logic without equality.

Logical Operations

$\mathcal{F}$ set of logical operations

$\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ set of quantifiers

First-Order Formulas

$\mathsf{F}_\Sigma(X)$ is the set of first-order formulas over Σ defined as follows:

$$
\begin{array}{lcll}
F, G, H & ::= & c & (c \in \mathcal{F}, \text{ constant}) \\
 & \mid & A & (\text{atomic formula}) \\
 & \mid & f(F_1, \ldots, F_n) & (f \in \mathcal{F} \text{ with arity } n) \\
 & \mid & QxF & (Q \in \mathcal{Q} \text{ is a quantifier})
\end{array}
$$

Bound and Free Variables

In QxF, $Q \in \mathcal{Q}$, we call F the scope of the quantifier Qx.
An *occurrence* of a variable x is called bound, if it is inside the scope of a quantifier Qx.
Any other occurrence of a variable is called free.

Formulas without free variables are also called closed formulas or sentential forms.

Formulas without variables are called ground.

Semantics

$M = \{1, \ldots, m\}$ set of truth values

$D \subseteq M$ set of designated truth values.

Truth tables for the logical operations:

$$\{f_M : M^n \to M | f/n \in \mathcal{F}\}$$

"Truth tables" for the quantifiers:

$$\{Q_M : \mathcal{P}(M) \to M | Q \in \mathcal{Q}\}$$

Examples: If $M = B_2 = \{0, 1\}$ then

$$\forall_{B_2} : \mathcal{P}(\{0, 1\}) \to \{0, 1\} \qquad \forall_{B_2}(X) = \min(X)$$
$$\exists_{B_2} : \mathcal{P}(\{0, 1\}) \to \{0, 1\} \qquad \exists_{B_2}(X) = \max(X)$$

Structures

An M-valued Σ-algebra (Σ-interpretation or Σ-structure) is a triple

$$\mathcal{A} = (U,\ (f_{\mathcal{A}} : U^n \to U)_{f/n \in \Omega},\ (p_{\mathcal{A}} : U^m \to M)_{p/m \in \Pi})$$

where $U \neq \emptyset$ is a set, called the universe of $\mathcal{A}$.

Normally, by abuse of notation, we will have $\mathcal{A}$ denote both the algebra and its universe.

By $\Sigma\text{-Alg}^M$ we denote the class of all M-valued Σ-algebras.

Assignments

Variable assignments $\beta : X \rightarrow \mathcal{A}$

and extensions to terms $\mathcal{A}(\beta) : T_{\Sigma} \rightarrow \mathcal{A}$ as in classical logic.

Truth Value of a Formula in $\mathcal{A}$ with Respect to β

$\mathcal{A}(\beta) : \mathsf{F}_\Sigma(X) \to M$ is defined inductively as follows:

$$
\begin{aligned}
\mathcal{A}(\beta)(c) &= c_M \\
\mathcal{A}(\beta)(p(s_1, \dots, s_n)) &= p_\mathcal{A}(\mathcal{A}(\beta)(s_1), \dots, \mathcal{A}(\beta)(s_n)) \in M \\
\mathcal{A}(\beta)(f(F_1, \dots, F_n)) &= f_M(\mathcal{A}(\beta)(F_1), \dots, \mathcal{A}(\beta)(F_n)) \\
\mathcal{A}(\beta)(QxF) &= Q_M(\{\mathcal{A}(\beta[x \mapsto a])(F) \mid a \in U\})
\end{aligned}
$$

First-order version of $\mathcal{L}_3$

$M = \{0, u, 1\}$

$D = \{1\}$

$\mathcal{F} = \{\vee, \wedge, \neg, \sim\}$

truth values as the propositional version

$\mathcal{Q} = \{\forall, \exists\}$

$$\forall_M(S) = \begin{cases} 1 & \text{if } S = \{1\} \\ 0 & \text{if } 0 \in S \\ u & \text{otherwise} \end{cases} \qquad \exists_M(S) = \begin{cases} 1 & \text{if } 1 \in S \\ 0 & \text{if } S = \{0\} \\ u & \text{otherwise} \end{cases}$$

Interpretation of quantifiers

$$\begin{array}{llll}
\mathcal{A}(\beta)(\forall x F(x)) = 1 & \textit{iff} & \text{for all } a \in U_{\mathcal{A}}, & \mathcal{A}(\beta[x \mapsto a])(F(x)) = 1 \\
\mathcal{A}(\beta)(\forall x F(x)) = 0 & \textit{iff} & \text{for some } a \in U_{\mathcal{A}}, & \mathcal{A}(\beta[x \mapsto a])(F(x)) = 0 \\
\mathcal{A}(\beta)(\forall x F(x)) = u & & \text{otherwise} &
\end{array}$$

$$\begin{array}{llll}
\mathcal{A}(\beta)(\exists x F(x)) = 1 & \textit{iff} & \text{for some } a \in U_{\mathcal{A}}, & \mathcal{A}(\beta[x \mapsto a])(F(x)) = 1 \\
\mathcal{A}(\beta)(\exists x F(x)) = 0 & \textit{iff} & \text{for all } a \in U_{\mathcal{A}}, & \mathcal{A}(\beta[x \mapsto a])(F(x)) = 0 \\
\mathcal{A}(\beta)(\forall x F(x)) = u & & \text{otherwise} &
\end{array}$$

Models, Validity, and Satisfiability

F is valid in $\mathcal{A}$ under assignment β:

$$\mathcal{A}, \beta \models F \quad :\Leftrightarrow \quad \mathcal{A}(\beta)(F) \in D$$

F is valid in $\mathcal{A}$ ($\mathcal{A}$ is a model of F):

$$\mathcal{A} \models F \quad :\Leftrightarrow \quad \mathcal{A}, \beta \models F, \text{ for all } \beta \in X \rightarrow U_{\mathcal{A}}$$

F is valid:

$$\models F \quad :\Leftrightarrow \quad \mathcal{A} \models F, \text{ for all } \mathcal{A} \in \Sigma\text{-alg}$$

F is called satisfiable iff there exist $\mathcal{A}$ and β such that $\mathcal{A}, \beta \models F$. Otherwise F is called unsatisfiable.

Entailment

$N \models F$ $:\Leftrightarrow$ for all $\mathcal{A} \in \Sigma$-alg and $\beta \in X \rightarrow U_{\mathcal{A}}$:

if $\mathcal{A}(\beta)(G) \in D$, for all $G \in N$, then $\mathcal{A}(\beta)(F) \in D$.

Models, Validity, and Satisfiability in $\mathcal{L}_3$

F is valid in $\mathcal{A}$ under assignment β:

$$\mathcal{A}, \beta \models_3 F \quad :\Leftrightarrow \quad \mathcal{A}(\beta)(F) = 1$$

F is valid in $\mathcal{A}$ ($\mathcal{A}$ is a model of F):

$$\mathcal{A} \models_3 F \quad :\Leftrightarrow \quad \mathcal{A}, \beta \models_3 F, \text{ for all } \beta \in X \to U_{\mathcal{A}}$$

F is valid (or is a tautology):

$$\models_3 F \quad :\Leftrightarrow \quad \mathcal{A} \models_3 F, \text{ for all } \mathcal{A} \in \Sigma\text{-alg}$$

F is called satisfiable iff there exist $\mathcal{A}$ and β such that $\mathcal{A}, \beta \models_3 F$. Otherwise F is called unsatisfiable.

Entailment in $\mathcal{L}_3$

$N \models_3 F$:$\Leftrightarrow$ for all $\mathcal{A} \in \Sigma$-alg and $\beta \in X \to U_{\mathcal{A}}$:

if $\mathcal{A}(\beta)(G) = 1$, for all $G \in N$, then $\mathcal{A}(\beta)(F) = 1$.

Observations

- Every $\mathcal{L}_3$-tautology is also a two-valued tautology.
- Not every two-valued tautology is an $\mathcal{L}_3$-tautology.
  Example: $F \vee \neg F$.