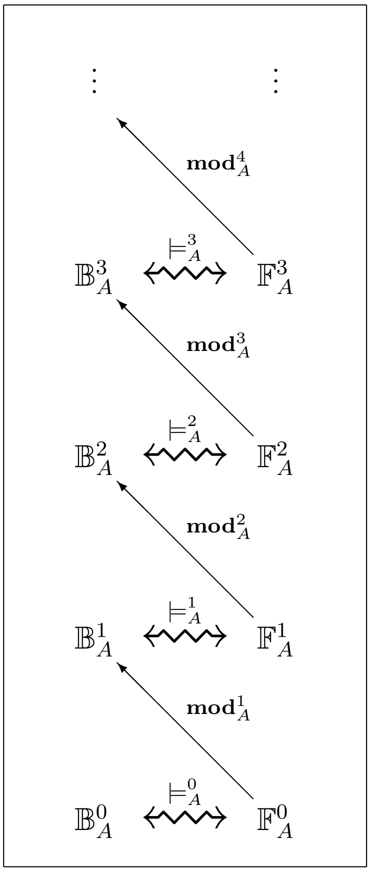
From propositional to hyper–propositional logic

www.bucephalus.org

July 3, 2007



Abstract

"*Hyper–propositional logic*" is our title for a generalization of traditional propositional logic that introduces formulas of arbitrary *degree* k, such that traditional propositional formulas turn into hyper–propositional formulas of first degree.

Primitive formulas of degree $k+1$ are "$\Diamond\sigma$" and "$\Box\sigma$", where σ is a formula of degree k. More complex formulas are constructed by means of conjunctions, disjunctions, and negations, as usual. "$\Diamond\sigma$" and "$\Box\sigma$" may be read "σ is satisfiable" and "σ is valid", respectively, and this is similar to modal logic, however the semantics of hyper–propositional formulas is very different.

Each formula of degree k has possible *interpretations* of degree k, and as usual, such an interpretation is a *model*, if it turns the formula into a true statement. So next and parallel to the hierarchical syntax, the corresponding semantics has arbitrary degrees as well. We call these interpretations *bit tables*, and for each degree we obtain a complete boolean algebra of bit tables. A very strong and elegant property of the whole design is the fact, that the entire model class of a formula actually turns into a single bit table on the next level, i.e. the whole formula algebra of degree k is embedded into the bit table algebra of degree $k+1$.

Contents

1 Bit values, characteristic functions and bit tables **4**

1.1 Bit values . . . 4

1.2 Power sets and characteristic functions . . . 4

1.3 Bit tables and their algebras . . . 5

2 Logical systems and traditional propositional logic in particular **10**

2.1 Optional prelude on logical systems . . . 10

2.2 Traditional propositional logic . . . 11

3 From traditional propositional to hyper–propositional logic **16**

3.1 Applying modal operators to formulas . . . 16

3.2 Higher degree propositional logic . . . 17

3.3 The foundation problem and its solution . . . 19

3.4 The final version of hyper–propositional logic . . . 22

List of Figures

1 Mathematical preliminaries . . . 3

2 Bit values and their algebra . . . 8

3 The isomorphism between $\mathfrak{P}(X)$ and $\mathfrak{Chf}(X)$ . . . 8

4 Bit tables and their algebras . . . 9

5 Traditional propositional logic . . . 15

6 Hyper–propositional logic . . . 23

7 Embedding traditional propositional into hyper–propositional logic . . . 24

Introduction and overview

Main objective of this text is the introduction of the syntax and semantics of *hyper–digital* or *hyper–propositional logic* by showing how it emerges from traditional propositional logic by adding modal operators to the language and finding a consistent and simple foundation for the whole system.

First, we define for each carrier set A and degree k, a set of *bit tables* $\mathbb{B}_A^k$ and a couple of operations on them. The result is the *bit table algebra* $\mathfrak{B}_A^k$. All that is summarized in two figures:

Figure 2 "Bit values and their algebra" and

Figure 4 "Bit tables and their algebra".

Bit tables are the *worlds*, *interpretations* or *semantics* of the systems we are about to develop here. The *syntax* on the other hand are the *(hyper–propositional) formula* sets $\mathbb{F}_A^k$, each one of them also constitutes a *(default) formula algebra* $\mathfrak{F}_A^k$. The important result will be that $\mathfrak{F}_A^k \hookrightarrow \mathfrak{B}_A^{k+1}$, i.e. that each formula algebra has a (very natural) embedding into the bit table algebra of next higher degree. And again, we will summarize all that in

Figure 6 "Hyper–propositional logic"

In fact, these three figures 2, 4 and 6 comprise the whole syntax and semantics of hyper–digital logic. But instead of presenting the system in an axiomatic fashion, we rather take a more narrative approach in this paper and try to explain, how hyper–propositional logic emerges from traditional propositional logic. Accordingly, we don't bother to provide the proofs of the "facts" we are about to state.

Figure 1: Mathematical preliminaries

Sets, functions and relations

- $\emptyset$, $\cap$, $\cup$, $\bigcap$, $\bigcup$, $\setminus$, $\uplus$ are the usual operations on sets, where $X \setminus Y$ is the difference of X and Y (all elements of X, but not in Y), and the disjunct union $X \uplus Y$ is the same as the union $X \cup Y$, but it is only defined for $X \cap Y = \emptyset$.
- $\mathbf{P}(X)$ $:= \{Y \mid Y \subseteq X\}$ denotes the power set of a given set X,
- $\mathbb{N}$ $:= \{0, 1, 2, 3, \ldots\}$ the set of natural numbers, $\mathbb{R}$ is the set of real numbers.
- $\mathbf{card}(X)$ is the cardinality of a given set X.
- $f = \begin{bmatrix} X \longrightarrow Y \\ x \mapsto f(x) \end{bmatrix}$ is our standard notation for a function $f : X \longrightarrow Y$ that maps each $x \in X$ to a unique and well–defined $f(x) \in Y$.
- $R : X \leftrightsquigarrow Y$ is our standard notation for the type expression of a relation R between X and Y. For example, $\leq : \mathbb{R} \leftrightsquigarrow \mathbb{R}$ for the usual linear order $\leq$ on the real numbers.

Quasi–boolean algebras

A quasi–boolean algebra is a structure $\langle B, \sqsubseteq, \equiv, \bot, \top, \sqcap, \sqcup, \neg \rangle$ where

- B is a set
- $\sqsubseteq$ is a *quasi–order* (i.e. a transitive and reflexive) relation on B
- $\equiv$ is the equivalence relation of $\sqsubseteq$, i.e. $x \equiv y$ iff $x \sqsubseteq y$ and $y \sqsubseteq x$
- $\bot$ is a *(quasi–)least element*, i.e. $\bot \sqsubseteq x$ for all $x \in B$
- $\top$ is a *(quasi–)top element*, i.e. $x \sqsubseteq \top$ for all $x \in B$
- $\sqcap$ is a *(quasi–)meet* function, i.e. $x \sqcap y$ is a *greatest lower bound* of $x, y \in B$,
- $\sqcup$ is a *(quasi–)join* function, i.e. $x \sqcup y$ is a *least upper bound* of $x, y \in B$,
- $\neg$ is a *(quasi–)complement* function, i.e. $\neg x \sqcap x \equiv \bot$ and $\neg x \sqcup x \equiv \top$, for all $x \in B$
- $\sqcap$ and $\sqcup$ are mutually *(quasi–)distributive*, i.e. $x \sqcap (y \sqcup z) \equiv (x \sqcap y) \sqcup (x \sqcap z)$ and $x \sqcup (y \sqcap z) \equiv (x \sqcup y) \sqcap (x \sqcup z)$, for all $x, y, z \in B$

Such a quasi–boolean algebra is

- complete, if there are two more functions
  - a supremum function $\coprod$ which returns a least upper bound $\coprod S$ for every $S \subseteq B$,
  - an infimum function $\prod$ which returns a greatest lower bound $\prod S$ for every $S \subseteq B$
- canonic or a boolean algebra, if $\sqsubseteq$ is antisymmetric, i.e. if $\equiv$ is the identity on B.

1 Bit values, characteristic functions and bit tables

1.0.1 Remark

In this section we introduce a whole range of closely related boolean algebras, all of them are complete. We start with the most typical specimens of boolean algebras at all: $\mathfrak{B}$, the one built on just two elements $\mathbf{0}$ and $\mathbf{1}$, and $\mathfrak{P}(X)$ the structure which emerges when the subsets of an arbitrary set X are ordered by the inclusion $\subseteq$. We suppose, we don't need to repeat the concept of a "(complete) boolean algebra" here — it is a common notion for structures that pretty much behave like $\mathfrak{B}$ and $\mathfrak{P}(X)$.
Another common notion is the "characteristic function": it is the same as a "subset", it just has another form. In other words, there is a bijection between subsets of a given set X and all the characteristic functions on X. $\mathfrak{Chf}(X)$ denotes the complete boolean algebra on these charcteristic functions isomorph to $\mathfrak{P}(X)$.
Finally, we built a whole recursive hierarchy on these sets of characteristic functions, similar to an iterated application of the power set operator "$\mathbf{P}(\mathbf{P}(\ldots(\mathbf{P}(X))\ldots))$". That way, we also have characteristic functions of first, second, etc "degree". Our general term for such a construction is "k–degree bit table". Each of those bit table sets constitues a boolean algebra again, we obtain a whole hierarchy of complete boolean algebras $\mathfrak{B}^1_X, \mathfrak{B}^2_X, \mathfrak{B}^3_X, \ldots$, starting with $\mathfrak{B}^1_X = \mathfrak{Chf}(X)$.
The whole matter in this chapter is not very difficult to understand and we state the mentioned theorems without proofs. The main idea is that we built some tools and get used to the notation.

1.1 Bit values

1.1.1 Definition — bit values and their algebra

The bit value set is the two–element set

$$\mathbb{B} := \{ \mathbf{0}, \mathbf{1} \}$$

where $\mathbf{0}$ is the zero bit and $\mathbf{1}$ the unit bit.
The bit value algebra is

$$\mathfrak{B} := \langle \mathbb{B}, \leq, \mathbf{0}, \mathbf{1}, \wedge, \vee, \bigwedge, \bigvee, \text{-} \rangle$$

where the operations are defined as usual, such that $\mathfrak{B}$ is a complete boolean algebra (see figure 2).

1.2 Power sets and characteristic functions

1.2.1 Definition

For every set X we define the power set algebra on X,

$$\mathfrak{P}(X) := \langle \mathbf{P}(X), \subseteq, \emptyset, \mathbb{1}, \cap, \cup, \bigcap, \bigcup, \complement \rangle$$

where $\mathbb{1} := X$ is the full set, $\complement Y := \mathbb{1} \setminus Y$ is the complement of $Y \in \mathbf{P}(X)$ and $\bigcap$ is defined on the whole domain $\mathbf{P}(X)$ by putting $\bigcap \emptyset := \mathbb{1}$.

1.2.2 Fact

For every set X holds:

(1) $\mathbf{card}(\mathbf{P}(X)) = 2^{\mathbf{card}(X)}$

(2) $\mathfrak{P}(X)$ is a complete boolean algebra

1.2.3 Definition — characteristic functions

A characteristic function (on X) is a function χ with codomain $\mathbb{B}$, i.e.

$$\chi : X \longrightarrow \mathbb{B}$$

For such a χ we define

$\mathbf{Unit}(\chi)$	$:= \{x \in X \mid \chi(x) = \mathbf{1}\}$	the unit set of χ
$\mathbf{Zero}(\chi)$	$:= \{x \in X \mid \chi(x) = \mathbf{0}\}$	the zero set of χ

If $X = \{x_1, \ldots, x_n\}$ is finite, we often represent ξ by

$$\chi = \begin{bmatrix} x_1 \mapsto \chi(x_1) \\ x_2 \mapsto \chi(x_2) \\ \vdots \quad \vdots \\ x_n \mapsto \chi(x_n) \end{bmatrix}$$

1.2.4 Example

If $X := \{a, b, c, d, e\}$, a characteristic function on X is given by

$$\chi := \begin{bmatrix} a \mapsto \mathbf{0} \\ b \mapsto \mathbf{1} \\ c \mapsto \mathbf{1} \\ d \mapsto \mathbf{0} \\ e \mapsto \mathbf{1} \end{bmatrix}$$

Then

$$\mathbf{Unit}(\chi) = \{b, c, e\} \qquad \text{and} \qquad \mathbf{Zero}(\chi) = \{a, d\}$$

1.2.5 Definition — characteristic function of a subset

Given a set X. For every $Y \subseteq X$ we define

$$\mathbf{cf}_Y := \mathbf{cf}^X_Y := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \begin{cases} \mathbf{1} & \text{if } x \in Y \\ \mathbf{0} & \text{if } x \notin Y \end{cases} \end{bmatrix}$$

the characteristic function of Y (in X)

1.2.6 Example

As in example 1.2.4, let $X := \{a,b,c,d,e\}$. For $Y := \{b,c,e\}$ we obtain

$$\mathbf{cf}_Y^X = \begin{bmatrix} a \mapsto \mathbf{0} \\ b \mapsto \mathbf{1} \\ c \mapsto \mathbf{1} \\ d \mapsto \mathbf{0} \\ e \mapsto \mathbf{1} \end{bmatrix}$$

Note, that

$$\mathbf{Unit}\left(\mathbf{cf}_Y^X\right) = \{b,c,e\} = Y$$

1.2.7 Fact

For every set X holds:

(1) $\mathbf{cf}_{①} : \mathbf{P}(X) \longrightarrow (X \longrightarrow \mathbb{B})$ is a bijection

(2) $\mathbf{Unit} : (X \longrightarrow \mathbb{B}) \longrightarrow \mathbf{P}(X)$ is the inverse bijection of $\mathbf{cf}_{①}$

(3) $\mathbf{card}(X \longrightarrow \mathbb{B}) = 2^{\mathbf{card}(X)}$

(4) $X = \mathbf{Unit}(\chi) \uplus \mathbf{Zero}(\chi)$, for each $\chi : X \longrightarrow \mathbb{B}$.
In other words, the domain of each characteristic function is a disjunct union of its unit and zero set.

1.2.8 Definition

The characteristic function algebra of a given set X

$$\mathfrak{Chf}(X) := \langle (X \longrightarrow \mathbb{B}), \sqsubseteq, \bot, \top, \sqcap, \sqcup, \prod, \coprod, \neg \rangle$$

is defined by

$$\chi_1 \sqsubseteq \chi_2 \quad \text{iff} \quad \chi_1(x) \leq \chi_2(x) \text{ for all } x \in X$$

$$\bot := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \mathbf{0} \end{bmatrix} \qquad \top := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \mathbf{1} \end{bmatrix}$$

$$\chi_1 \sqcap \chi_2 := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \chi_1(x) \wedge \chi_2(x) \end{bmatrix}$$

$$\chi_1 \sqcup \chi_2 := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \chi_1(x) \vee \chi_2(x) \end{bmatrix}$$

$$\prod \Xi := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \bigwedge\{\chi(x) \mid x \in \xi\} \end{bmatrix}$$

$$\coprod \Xi := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \bigvee\{\chi(x) \mid x \in \xi\} \end{bmatrix}$$

$$\neg \chi := \begin{bmatrix} X \longrightarrow \mathbb{B} \\ x \mapsto \text{-}\chi(x) \end{bmatrix}$$

for all $\chi, \chi_1, \chi_2 : X \longrightarrow \mathbb{B}$ and $\Xi \subseteq (X \longrightarrow \mathbb{B})$.

1.2.9 Example

Three characteristic functions on $X := \{a,b,c,d,e\}$ are given by

$$\chi_1 = \begin{bmatrix} a \mapsto \mathbf{0} \\ b \mapsto \mathbf{0} \\ c \mapsto \mathbf{0} \\ d \mapsto \mathbf{0} \\ e \mapsto \mathbf{0} \end{bmatrix} \qquad \chi_2 = \begin{bmatrix} a \mapsto \mathbf{0} \\ b \mapsto \mathbf{1} \\ c \mapsto \mathbf{1} \\ d \mapsto \mathbf{0} \\ e \mapsto \mathbf{1} \end{bmatrix} \qquad \chi_3 = \begin{bmatrix} a \mapsto \mathbf{0} \\ b \mapsto \mathbf{0} \\ c \mapsto \mathbf{0} \\ d \mapsto \mathbf{1} \\ e \mapsto \mathbf{0} \end{bmatrix}$$

Then

(a) $\chi_1 \sqsubseteq \chi_2$, because $\chi_1(x) \leq \chi_2(x)$ for each $x \in X$

(b) $\chi_3 \not\sqsubseteq \chi_2$, because $\chi_3(d) = \mathbf{1} \not\leq \mathbf{0} = \chi_2(d)$

(c) $\neg\chi_2 = \begin{bmatrix} a \mapsto \text{-}\mathbf{0} \\ b \mapsto \text{-}\mathbf{1} \\ c \mapsto \text{-}\mathbf{1} \\ d \mapsto \text{-}\mathbf{0} \\ e \mapsto \text{-}\mathbf{1} \end{bmatrix} = \begin{bmatrix} a \mapsto \mathbf{1} \\ b \mapsto \mathbf{0} \\ c \mapsto \mathbf{0} \\ d \mapsto \mathbf{1} \\ e \mapsto \mathbf{0} \end{bmatrix}$

(d) $\coprod\{\chi_1,\chi_2,\chi_3\} = \begin{bmatrix} a \mapsto \bigvee\{\mathbf{0},\mathbf{0},\mathbf{0}\} \\ b \mapsto \bigvee\{\mathbf{0},\mathbf{1},\mathbf{0}\} \\ c \mapsto \bigvee\{\mathbf{0},\mathbf{1},\mathbf{0}\} \\ d \mapsto \bigvee\{\mathbf{0},\mathbf{0},\mathbf{1}\} \\ e \mapsto \bigvee\{\mathbf{0},\mathbf{1},\mathbf{0}\} \end{bmatrix} = \begin{bmatrix} a \mapsto \mathbf{0} \\ b \mapsto \mathbf{1} \\ c \mapsto \mathbf{1} \\ d \mapsto \mathbf{1} \\ e \mapsto \mathbf{1} \end{bmatrix}$

1.2.10 Fact

For every set X holds:

(1) $\mathbf{Unit} : \mathfrak{Chf}(X) \cong \mathfrak{P}(X)$
i.e. **Unit** is not only a bijection from $X \longrightarrow \mathbb{B}$ into $\mathbf{P}(X)$, but an isomorphism from $\mathfrak{Chf}(X)$ into $\mathfrak{P}(X)$

(2) $\mathfrak{Chf}(X)$ is a complete boolean algebra.

1.2.11 Remark

Figure 3 shows the typical order diagram (also called "Hasse diagram") of the complete boolean algebra $\mathfrak{P}(X)$, where $X = \{a,b,c\}$ is a simple example of a tree element set. The diagram of the isomorph $\mathfrak{Chf}(X)$ thus looks the same, and the two mutually inverse isomorphisms **Unit** and $\mathbf{cf}_{①}$ point from one diagram to the according places in the other one.

1.3 Bit tables and their algebras

1.3.1 Definition

For every set A and each $k \in \mathbb{N}$ we define

$$\mathbb{B}_A^k := \begin{cases} A & \text{if } k = 0 \\ \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} & \text{if } k > 0 \end{cases}$$

the bit table set on (carrier) A and (degree) k

1.3.2 Remark

For each given A, the (members of the) first bit table sets $\mathbb{B}_A^k$ also have alternative "geometric" names:

$$
\begin{array}{lcll}
\mathbb{B}^0_A & = & A & \text{is also called the bit point set on } A \\
\mathbb{B}^1_A & = & A \longrightarrow \mathbb{B} & \text{are the bit lines on } A \\
\mathbb{B}^2_A & = & (A \longrightarrow \mathbb{B}) \longrightarrow \mathbb{B} & \text{the bit squares on } A \\
\mathbb{B}^3_A & = & ((A \longrightarrow \mathbb{B}) \longrightarrow \mathbb{B}) \longrightarrow \mathbb{B} & \text{the bit cubes on } A \\
\vdots & & \vdots &
\end{array}
$$

1.3.3 Bit table diagrams

In case both A and k are really small, we can note a bit table $\Omega \in \mathbb{B}^k_A$ according to 1.2.3 by

$$
\begin{bmatrix}
\omega_1 & \mapsto & \Omega(\omega_1) \\
\omega_2 & \mapsto & \Omega(\omega_2) \\
\vdots & & \vdots \\
\omega_n & \mapsto & \Omega(\omega_n)
\end{bmatrix}
$$

But instead, we often picture it by its *bit table diagram*, which is a littel more compact, when $k \geq 2$.
For example, let us take $A = \{a, b, c\}$.

(i) For $k = 0$, Ω simply is a member of A and "Ω" very much is its one bit table diagram.

(ii) If $k = 1$ we draw the diagram for $\Omega : A \longrightarrow \mathbb{B}$ in two steps

(1) First we list all the arguments a, b, c of Ω

a	b	c

(2) Then we add for each argument ω its bit value $\Omega(\omega)$

Ω =

a	b	c
$\Omega(a)$	$\Omega(b)$	$\Omega(c)$

(iii) For $k = 2$, $\Omega : \mathbb{B}^1_A \longrightarrow \mathbb{B}$ is again constructed in two steps:

(1) There are $2^3 = 8$ different arguments $\omega_1, \ldots, \omega_n \in \mathbb{B}^1_A$ of Ω. We list all these arguments in a compact list

a	b	c
0	0	0
1	0	0
⋮	⋮	⋮
1	1	1

(2) Then we attach the bit values $\Omega(\omega)$, for each ω:

a	b	c	
0	0	0	$\Omega(\omega_1)$
1	0	0	$\Omega(\omega_2)$
⋮	⋮	⋮	⋮
1	1	1	$\Omega(\omega_8)$

The result is a common diagram from traditional propositional logic, where is it usually called a *truth table*.

(iv) For $k = 3$, we perform the same two steps to produce a bit table diagram for $\Omega \in \mathbb{B}^3_A$:

(1) First list all the $2^8 = 256$ arguments $\omega_1, \ldots, \omega_{256} \in \mathbb{B}^2_A$:

a	b	c				
0	0	0	0	1	...	1
1	0	0	1	1	...	1
⋮	⋮	⋮	⋮	⋮		⋮
1	1	0	0	1	...	1

(2) Then attach the corresponding bit values:

a	b	c				
0	0	0	0	1	...	1
1	0	0	1	1	...	1
⋮	⋮	⋮	⋮	⋮		⋮
1	1	0	0	1	...	1
			$\Omega(\omega_1)$	$\Omega(\omega_2)$	...	$\Omega(\omega_{256})$

This can in principle be done for every $k > 3$ as well. But of course, the size of the diagram grows exponentially. For really small A and k however, the pictures are sometimes useful.

1.3.4

Bit tables with degree higher than 0 are characteristic functions, as defined in 1.2.3. If $\Omega \in \mathbb{B}^k_A$ with $k > 0$, then $\Omega : \mathbb{B}^{k-1}_A \longrightarrow \mathbb{B}$ with

$$\mathbb{B}^{k-1}_A = \textbf{Unit}\,(\Omega) \uplus \textbf{Zero}\,(\Omega)$$

In our standard notation for functions, such a bit table is given by

$$\Omega = \begin{bmatrix} \mathbb{B}^{k-1}_A \longrightarrow \mathbb{B} \\ \omega \mapsto \Omega(\omega) \end{bmatrix}$$

For every $\mathbb{B}^k_A = \mathbb{B}^{k-1}_A \longrightarrow \mathbb{B}$ we can also define the characteristic function algebra of $\mathbb{B}^{k-1}_A$ as introduced in 1.2.8.
Later on it will be useful to clearly distinguish these algebras for different A and k, and therefore, we attach this information to the operation symbols and write "$\sqsubseteq^k_A$", "$\perp^k_A$", "$\sqcap^k_A$" instead of simply "$\sqsubseteq$", "$\perp$", "$\sqcap$" etc.

1.3.5 Definition

For every set A and $k \geq 1$ we define

$$
\begin{array}{lcl}
\mathfrak{B}^k_A & := & \langle \mathbb{B}^k_A, \sqsubseteq^k_A, \perp^k_A, \top^k_A, \sqcap^k_A, \sqcup^k_A, \prod^k_A, \coprod^k_A, \neg^k_A \rangle \\
 & := & \mathfrak{Chf}\left(\mathbb{B}^{k-1}_A\right)
\end{array}
$$

the bit table algebra on (carrier) A and (degree) k.

1.3.6 Operations in general functional representation

So if A and $k \geq 1$ are given, then for all $\Omega, \Omega' \in \mathbb{B}^k_A$ and $\Gamma \subseteq \mathbb{B}^k_A$,

$$\Omega \sqsubseteq^k_A \Omega' \quad \text{iff} \quad \Omega(\omega) \leq \Omega' \text{ for all } \omega \in \mathbb{B}^{k-1}_A$$

$$\bot_A^k = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \mathbf{0} \end{bmatrix} \qquad \top_A^k = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \mathbf{1} \end{bmatrix}$$

$$\neg_A^k \Omega = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \text{-}\Omega(\omega) \end{bmatrix}$$

$$\Omega \sqcap_A^k \Omega' = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \Omega(\omega) \wedge \Omega'(\omega) \end{bmatrix}$$

$$\Omega \sqcup_A^k \Omega' = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \Omega(\omega) \vee \Omega'(\omega) \end{bmatrix}$$

$$\textstyle\prod_A^k \Gamma = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \bigwedge\limits_{\Omega \in \Gamma} \Omega(\omega) \end{bmatrix} \qquad \coprod_A^k \Gamma = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \bigvee\limits_{\Omega \in \Gamma} \Omega(\omega) \end{bmatrix}$$

1.3.7 Operations in diagram representation

If the bit tables are given by their diagrams, the application of the operations becomes very intuitive.
For example, take $A = \{a, b, c\}$, $k = 1$

$$\bot_A^k = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \end{array} \quad \text{and} \quad \top_A^k = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \end{array}$$

If two members of $\mathbb{B}_A^k$, i.e. two bit lines on A are given by

$$\Omega := \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \hline \end{array} \quad \text{and} \quad \Omega' := \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \hline \end{array}$$

then

$$\Omega \sqcap_A^k \Omega' = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{1} \wedge \mathbf{1} & \mathbf{0} \wedge \mathbf{0} & \mathbf{0} \wedge \mathbf{1} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \hline \end{array}$$

$$\neg_A^k \Omega = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \text{-}\mathbf{1} & \text{-}\mathbf{0} & \text{-}\mathbf{0} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \hline \end{array}$$

and

$$\Omega \sqsubseteq_A^k \Omega' \text{ because } \begin{pmatrix} \Omega(a) = \mathbf{1} \leq \mathbf{1} = \Omega'(a) \text{ and} \\ \Omega(b) = \mathbf{0} \leq \mathbf{0} = \Omega'(b) \text{ and} \\ \Omega(c) = \mathbf{0} \leq \mathbf{1} = \Omega'(c) \end{pmatrix}$$

Similar rules hold for other carrier set and degrees. For example, for bit squares (degree=2) on $A = \{a, b\}$ we have

$$\bot_A^k = \begin{array}{|c|c||c|} \hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \end{array} \quad \text{and} \quad \top_A^k = \begin{array}{|c|c||c|} \hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \end{array}$$

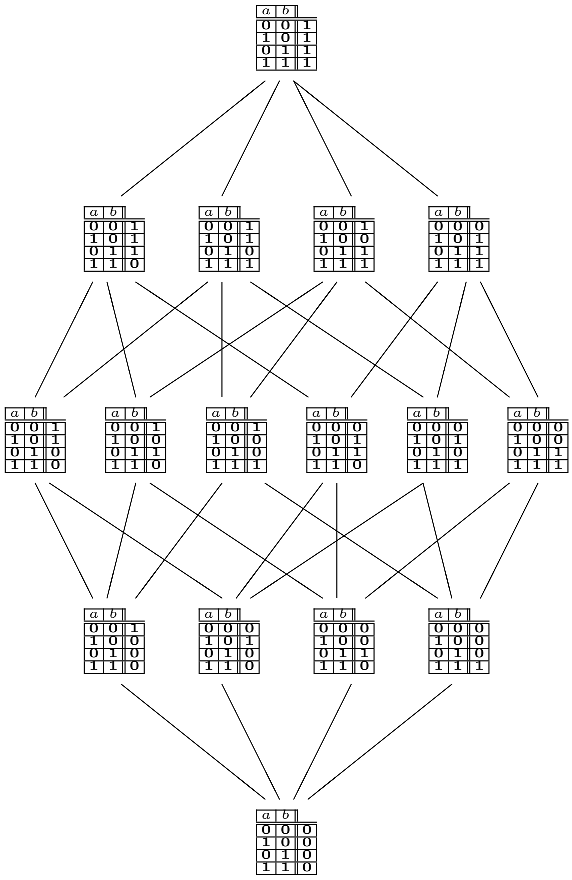
and

$$\begin{array}{|c|c||c|} \hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \end{array} \sqcup_A^k \begin{array}{|c|c||c|} \hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \end{array} = \begin{array}{|c|c||c|} \hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \vee \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \vee \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \vee \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \vee \mathbf{0} \\ \hline \end{array} = \begin{array}{|c|c||c|} \hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \end{array}$$

and so on.

1.3.8 Example

For the same $A = \{a, b\}$ and $k = 2$ the entire boolean algebra $\mathfrak{B}_A^k$ is represented by the following order diagram:



1.3.9 Fact

$\mathfrak{B}_A^k$ is a complete boolean algebra, for every set A and $k \geq 1$.

1.3.10 Remark

We call a boolean algebra degenerated, if it only has a single element. Usually, one expects a boolean algebra to have at least a bottom and a top element, both being different. Otherwise it doesn't really make sense to talk about a "boolean algebra" at all. Some authors therefore exclude degenerated boolean algebras from the definition right away.
The statement of 1.3.9 is true in general, but $\mathfrak{B}_A^k$ is degenerated, if and only if $A = \emptyset$ and $k = 1$. In that case $\mathbb{B}_A^k$ has only one element, the empty function of type $\emptyset \longrightarrow \mathbb{B}$.

Figure 2: Bit values and their algebra

Bit values

$\mathbb{B} := \{\mathbf{0}, \mathbf{1}\}$ is the bit value class, where $\mathbf{0}$ is the zero bit and $\mathbf{1}$ the unit bit.

Bit value algebra

$\mathfrak{B} := \langle \mathbb{B}, \leq, \mathbf{0}, \mathbf{1}, \wedge, \vee, \bigwedge, \bigvee, \text{-} \rangle$ is the bit value algebra, where

$\beta_1 \leq \beta_2$ iff $\beta_1 = \mathbf{0}$ or $\beta_2 = \mathbf{1}$

$$\beta_1 \wedge \beta_2 := \bigwedge \{\beta_1, \beta_2\} \qquad \bigwedge \mathcal{B} := \begin{cases} \mathbf{0} & \text{if } \mathbf{0} \in \mathcal{B} \\ \mathbf{1} & \text{else} \end{cases}$$

$$\beta_1 \vee \beta_2 := \bigvee \{\beta_1, \beta_2\} \qquad \bigvee \mathcal{B} := \begin{cases} \mathbf{1} & \text{if } \mathbf{1} \in \mathcal{B} \\ \mathbf{0} & \text{else} \end{cases}$$

$$\text{-}\beta := \begin{cases} \mathbf{0} & \text{if } \beta = \mathbf{1} \\ \mathbf{1} & \text{else} \end{cases}$$

for all $\beta, \beta_1, \beta_2 \in \mathbb{B}$ and $\mathcal{B} \subseteq \mathbb{B}$.

We also write, for all $\beta_1, \ldots, \beta_n \in \mathbb{B}$ with $n \geq 0$,

$$\bigwedge_{i=1}^{n} \beta_i \text{ for } \bigwedge \{\beta_1, \ldots, \beta_n\} \qquad \text{and} \qquad \bigvee_{i=1}^{n} \beta_i \text{ for } \bigvee \{\beta_1, \ldots, \beta_n\}$$

Theorem

$\mathfrak{B}$ is a complete boolean algebra.

Figure 3: The isomorphism between $\mathfrak{P}(X)$ and $\mathfrak{Chf}(X)$

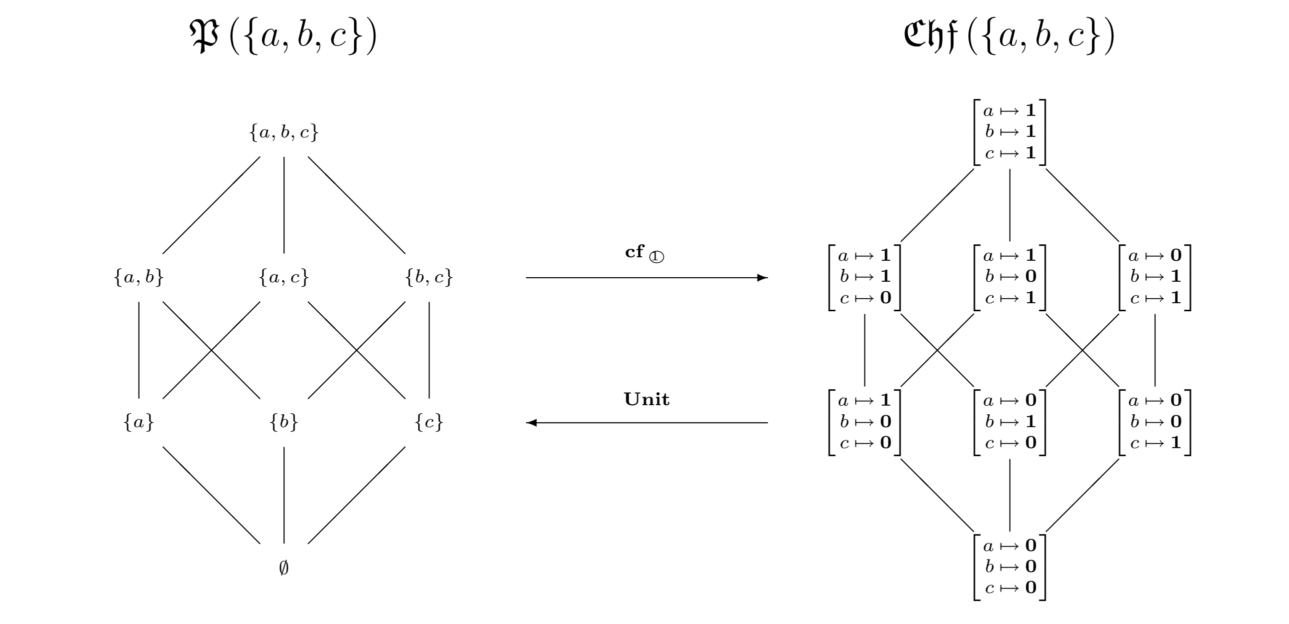


Figure 4: Bit tables and their algebras

Bit tables

For every set A and each natural number k we define

$$\mathbb{B}_A^k \quad := \quad \begin{cases} A & \text{if } k=0 \\ \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} & \text{if } k>0 \end{cases}$$

the bit table set of carrier A and degree k.

In our default notation for functions[a], each bit table $\Omega \in \mathbb{B}_A^k$ with $k \geq 1$ is then given by

$$\Omega = \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \Omega(\omega) \end{bmatrix}$$

Similar to geometry, bit tables of small degree $k = 0,1,2,3$ are also called *bit points*, *bit lines*, *bit squares* and *bit cubes*, respectively. In traditional propositional logic, bit squares are also known as *truth tables*.

Bit table diagrams

If both A and k are finite, we can represent each $\Omega \in \mathbb{B}_A^k$ by its bit table diagram. For example

(1) If $A = \{a,b\}$ and $k=1$ then

$$\begin{array}{|c|c|c|} \hline a & b & c \\ \hline \beta_1 & \beta_2 & \beta_3 \\ \hline \end{array} := \begin{bmatrix} A \longrightarrow \mathbb{B} \\ \begin{bmatrix} a \mapsto \beta_1 \\ b \mapsto \beta_2 \\ c \mapsto \beta_3 \end{bmatrix} \end{bmatrix}$$

(2) If $A = \{a,b\}$ and $k=2$ then

$$\begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \\ 1 & 0 & \beta_2 \\ 0 & 1 & \beta_3 \\ 1 & 1 & \beta_4 \\ \hline \end{array} := \begin{bmatrix} \mathbb{B}_A^1 \longrightarrow \mathbb{B} \\ \begin{bmatrix} \begin{array}{|c|c|} \hline a & b \\ \hline 0 & 0 \\ \hline \end{array} \mapsto \beta_1 \\ \begin{array}{|c|c|} \hline a & b \\ \hline 1 & 0 \\ \hline \end{array} \mapsto \beta_2 \\ \begin{array}{|c|c|} \hline a & b \\ \hline 0 & 1 \\ \hline \end{array} \mapsto \beta_3 \\ \begin{array}{|c|c|} \hline a & b \\ \hline 1 & 1 \\ \hline \end{array} \mapsto \beta_4 \end{bmatrix} \end{bmatrix}$$

(3) If $A = \{a\}$ and $k=3$ then

$$\begin{array}{|c||c|c|c|c|} \hline a & & & & \\ \hline 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ \hline & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \hline \end{array} := \begin{bmatrix} \mathbb{B}_A^2 \longrightarrow \mathbb{B} \\ \begin{bmatrix} \begin{array}{|c||} \hline a \\ \hline \end{array} \\ \begin{array}{|c|c|} \hline 0 & 0 \\ 1 & 0 \\ \hline \end{array} \mapsto \beta_1 \\ \begin{array}{|c||} \hline a \\ \hline \end{array} \\ \begin{array}{|c|c|} \hline 0 & 1 \\ 1 & 0 \\ \hline \end{array} \mapsto \beta_2 \\ \begin{array}{|c||} \hline a \\ \hline \end{array} \\ \begin{array}{|c|c|} \hline 0 & 0 \\ 1 & 1 \\ \hline \end{array} \mapsto \beta_3 \\ \begin{array}{|c||} \hline a \\ \hline \end{array} \\ \begin{array}{|c|c|} \hline 0 & 1 \\ 1 & 1 \\ \hline \end{array} \mapsto \beta_4 \end{bmatrix} \end{bmatrix}$$

Bit table algebras

$\mathfrak{B}_A^k := \langle \mathbb{B}_A^k, \sqsubseteq_A^k, \bot_A^k, \top_A^k, \sqcap_A^k, \sqcup_A^k, \prod_A^k, \coprod_A^k, \neg_A^k \rangle$ is the bit table algebra, for each set A and $k \geq 1$, where

$$\Omega \sqsubseteq_A^k \Omega' \quad \text{iff } \Omega(\omega) \leq \Omega'(\omega) \text{ for all } \omega \in \mathbb{B}_A^{k-1}$$

$$\Omega \sqcap_A^k \Omega' := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \Omega(\omega) \wedge \Omega'(\omega) \end{bmatrix} \qquad \Omega \sqcup_A^k \Omega' := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \Omega(\omega) \vee \Omega'(\omega) \end{bmatrix} \qquad \neg_A^k \Omega := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \text{-}\Omega(\omega) \end{bmatrix}$$

$$\bot_A^k := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \mathbf{0} \end{bmatrix} \qquad \top_A^k := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \mathbf{1} \end{bmatrix} \qquad \textstyle\prod_A^k \Gamma := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \bigwedge\{\Omega(\omega) \mid \Omega \in \Gamma\} \end{bmatrix} \qquad \textstyle\coprod_A^k \Gamma := \begin{bmatrix} \mathbb{B}_A^{k-1} \longrightarrow \mathbb{B} \\ \omega \mapsto \bigvee\{\Omega(\omega) \mid \Omega \in \Gamma\} \end{bmatrix}$$

for all $\Omega, \Omega' \in \mathbb{B}_A^k$ and $\Gamma \subseteq \mathbb{B}_A^k$.

Using bit table diagrams and taking $A = \{a,b\}$ and $k = 2$ for example, the operations are

$$\begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \\ 1 & 0 & \beta_2 \\ 0 & 1 & \beta_3 \\ 1 & 1 & \beta_4 \\ \hline \end{array} \sqsubseteq_A^2 \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \gamma_1 \\ 1 & 0 & \gamma_2 \\ 0 & 1 & \gamma_3 \\ 1 & 1 & \gamma_4 \\ \hline \end{array} \text{ iff } \begin{pmatrix} \beta_1 \leq \delta_1 \text{ and} \\ \beta_2 \leq \delta_2 \text{ and} \\ \beta_3 \leq \delta_3 \text{ and} \\ \beta_4 \leq \delta_4 \end{pmatrix} \qquad \bot_A^2 = \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ \hline \end{array} \qquad \top_A^2 = \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ \hline \end{array}$$

$$\neg_A^2 \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \\ 1 & 0 & \beta_2 \\ 0 & 1 & \beta_3 \\ 1 & 1 & \beta_4 \\ \hline \end{array} = \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \text{-}\beta_1 \\ 1 & 0 & \text{-}\beta_2 \\ 0 & 1 & \text{-}\beta_3 \\ 1 & 1 & \text{-}\beta_4 \\ \hline \end{array} \qquad \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \\ 1 & 0 & \beta_2 \\ 0 & 1 & \beta_3 \\ 1 & 1 & \beta_4 \\ \hline \end{array} \sqcap_A^2 \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \gamma_1 \\ 1 & 0 & \gamma_2 \\ 0 & 1 & \gamma_3 \\ 1 & 1 & \gamma_4 \\ \hline \end{array} = \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \wedge \gamma_1 \\ 1 & 0 & \beta_2 \wedge \gamma_2 \\ 0 & 1 & \beta_3 \wedge \gamma_3 \\ 1 & 1 & \beta_4 \wedge \gamma_4 \\ \hline \end{array} \qquad \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \\ 1 & 0 & \beta_2 \\ 0 & 1 & \beta_3 \\ 1 & 1 & \beta_4 \\ \hline \end{array} \sqcup_A^2 \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \gamma_1 \\ 1 & 0 & \gamma_2 \\ 0 & 1 & \gamma_3 \\ 1 & 1 & \gamma_4 \\ \hline \end{array} = \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & \beta_1 \vee \gamma_1 \\ 1 & 0 & \beta_2 \vee \gamma_2 \\ 0 & 1 & \beta_3 \vee \gamma_3 \\ 1 & 1 & \beta_4 \vee \gamma_4 \\ \hline \end{array}$$

These methods hold similarly for other A and k.

Theorem

$\mathfrak{B}_A^k$ is a complete boolean algebra, for every set A and $k \geq 1$.

---

[a] In our notation we write $f = \begin{bmatrix} X \longrightarrow Y \\ x \mapsto f(x) \end{bmatrix}$ for a function $f : X \longrightarrow Y$ that maps each $x \in X$ to a well–defined $f(x) \in Y$.

2 Logical systems and traditional propositional logic in particular

2.0.11 Introduction

We are about show how hyper–propositional logic can be developed as a generalization of traditional propositional logic. We first start with a general description of a "logical system" and then introduce "(traditional) propositional logic" as a special example.
In itself, our review is incomplete (e.g. we entirely neglect the proof system and the derivation concept of logical systems), the emphasis on certain aspects (like the different representations of an interpretation structure in 2.1.2) might seem awkward, and the whole trip (from "interpretation structures" to "formula algebras") is somewhat counter–intuitive. But it points out the properties that motivate some of the ideas later on.

2.1 Optional prelude on logical systems

2.1.1 Interpretation structures

An interpretation structure is made of three ingredients:[1]

(i) A formal language, usually represented as a set $FORM$ of formulas.
(ii) A potential reality, usually represented as a class INT of possible interpretations.
(iii) A logical semantics, which defines a correspondence between formulas φ and interpretations Ω: φ can be a true formula for Ω. In that case, Ω is also said to satisfy φ or be a model for φ. Otherwise, φ is false for Ω.

2.1.2 Three equivalent representations

In precise mathematical terms, such an interpretation structure can be given in different but equivalent representations:

$\models : INT \leftrightsquigarrow FORM$	the model relation
$\mathbf{Mod} : FORM \longrightarrow \mathbf{P}(INT)$	the model class function
$\mathbf{mod} : FORM \longrightarrow INT \longrightarrow \mathbb{B}$	the model function

"$\Omega \models \varphi$" is saying that Ω is a model for φ. Otherwise, we write "$\Omega \not\models \varphi$", as usual. "$\mathbf{Mod}(\varphi) \subseteq INT$" denotes the class of all the models of φ. And $\mathbf{mod}(\varphi) : INT \longrightarrow \mathbb{B}$, the model function[2] of φ, is the characteristic function of this model class. In other words, $\mathbf{mod}(\varphi)(\Omega)$ is $\mathbf{1}$, if Ω is a model for φ, and $\mathbf{0}$, otherwise.

Most common is the formal definition of a semantics by means of "$\models$". But note, that each of the three representations is equivalent and each one implies the other two.

2.1.3 Transformation rules

Let $FORM$ and INT be two classes.

(1) Given a model relation $\rho : INT \leftrightsquigarrow FORM$
  (a) The model class function of ρ
  $$\mathbf{Mod}_\rho : FORM \longrightarrow \mathbf{P}(INT)$$
  is defined by
  $$\mathbf{Mod}_\rho(\varphi) \; := \; \{\Omega \in INT \mid \Omega\,\rho\,\varphi\}$$
  for all $\varphi \in FORM$.
  (b) The model function of ρ
  $$\mathbf{mod}_\rho : FORM \longrightarrow INT \longrightarrow \mathbb{B}$$
  is defined by
  $$\mathbf{mod}_\rho(\varphi)(\Omega) \; := \; \begin{cases} \mathbf{1} & \text{if } \Omega\,\rho\,\varphi \\ \mathbf{0} & \text{else} \end{cases}$$
  for all $\varphi \in FORM$ and $\Omega \in INT$

(2) Given a model class function $\mathcal{M} : FORM \longrightarrow \mathbf{P}(INT)$
  (a) The model relation of $\mathcal{M}$
  $$\models_{\mathcal{M}} : INT \leftrightsquigarrow FORM$$
  is defined by
  $$\Omega \models_{\mathcal{M}} \varphi \quad \text{iff} \quad \Omega \in \mathcal{M}$$
  for all $\Omega \in INT$ and $\varphi \in FORM$
  (b) The model function of μ
  $$\mathbf{mod}_{\mathcal{M}} : FORM \longrightarrow INT \longrightarrow \mathbb{B}$$
  is defined by
  $$\mathbf{mod}_{\mathcal{M}}(\varphi)(\Omega) \; := \; \begin{cases} \mathbf{1} & \text{if } \Omega \in \mathcal{M} \\ \mathbf{0} & \text{else} \end{cases}$$
  for all $\varphi \in FORM$ and $\Omega \in INT$.

(3) Given a model function $\mu : FORM \longrightarrow INT \longrightarrow \mathbb{B}$
  (a) The model relation of μ
  $$\models_\mu : INT \leftrightsquigarrow FORM$$
  is defined by
  $$\Omega \models_\mu \varphi \quad \text{iff} \quad \mu(\varphi)(\Omega) = \mathbf{1}$$
  (b) The model class function of μ
  $$\mathbf{Mod}_\mu : FORM \longrightarrow \mathbf{P}(INT)$$
  is defined by
  $$\mathbf{Mod}_\mu(\varphi) \; := \; \{\Omega \in INT \mid \mu(\varphi)(\Omega) = \mathbf{1}\}$$
  for all $\varphi \in FORM$.

[1] Actually, this definition of an "interpretation structure" meets every binary relation and is thus not suitable for a description of something like a logical system. Later on in ??lindenbaum structure must be a boolean lattic?? , we put more constraints on interpretation systems to be useful.

[2] So we call "**mod**" itself the "model function", but also its application "$\mathbf{mod}(\varphi)$". Later on in the context of hyper-propositional logic, it makes sense to call "$\mathbf{mod}(\varphi)$" the "super–model of φ", because a model of $\varphi \in \mathbb{F}_A^k$ is some $\omega \in \mathbb{B}_A^k$, but $\mathbf{mod}(\varphi) \in \mathbb{B}_A^{k+1}$, i.e. a model has the same degree k as the formula, but the model function has degree $k+1$, hence the "super–model".

2.1.4 Sub– and equivalence

Each interpretation structure with formulas $FORM$, interpretations INT and a semantics, represented by $\models$, **Mod**, **mod**, induces an order on its formulas, usually called "entailment", "consequence" or "implication". We prefer the title "subvalence"[3]) and define two relations

$\Rightarrow : FORM \rightsquigarrow FORM$ the subvalence relation

$\Leftrightarrow : FORM \rightsquigarrow FORM$ the equivalence relation

by putting

$$\begin{aligned}
\varphi \Rightarrow \psi \quad & \text{iff} \quad \forall \Omega \in INT \,.\, (\Omega \models \varphi \text{ implies } \Omega \models \psi) \\
& \text{iff} \quad \mathbf{Mod}\,(\varphi) \subseteq \mathbf{Mod}\,(\psi) \\
& \text{iff} \quad \mathbf{mod}\,(\varphi) \sqsubseteq \mathbf{mod}\,(\psi) \\
\varphi \Leftrightarrow \psi \quad & \text{iff} \quad \forall \Omega \in INT \,.\, (\Omega \models \varphi \text{ iff } \Omega \models \psi) \\
& \text{iff} \quad \mathbf{Mod}\,(\varphi) = \mathbf{Mod}\,(\psi) \\
& \text{iff} \quad \mathbf{mod}\,(\varphi) = \mathbf{mod}\,(\psi)
\end{aligned}$$

for all $\varphi, \psi \in FORM$.

Note, that all this is well–defined. Here, "$\subseteq$" is the order on $\mathbf{P}\,(INT)$ and "$\sqsubseteq$" is the order on the characteristic functions $INT \longrightarrow \mathbb{B}$, introduced in 1.2.8. Both are isomorph structures.

2.1.5 A quasi–boolean algebra of formulas

Obviously, $\langle FORM, \Rightarrow, \Leftrightarrow \rangle$ is a quasi–ordered set. [4]

But for a proper logical system we want this quasi–ordered set to be a quasi–boolean algebra of formulas, i.e. equipped with the following operations:

- a zero or false formula constant $\mathbf{f}$, which is a least element in $\langle FORM, \Rightarrow \rangle$
- a unit or true element $\mathbf{t}$, which is a greatest element
- a (quasi–)meet function $\curlywedge : FORM \times FORM \longrightarrow FORM$, which returns a greatest lower bound $\varphi \curlywedge \psi$ for all $\varphi, \psi \in FORM$.
- a join function $\curlyvee$, which returns a least upper bound $\varphi \curlyvee \psi$, and
- a complement function $\text{-} : FORM \longrightarrow FORM$, such that $\varphi \vee \text{-}\varphi \Leftrightarrow t$ and $\varphi \wedge \text{-}\varphi \Leftrightarrow f$, for all $\varphi \in FORM$.

Note, that all these ingredients are usually not unique, but only unique up to equivalence. This is the difference between a quasi–order and a (partial) ordered structure.

2.1.6 Example

Suppose we have already given the propositional formula set, together with the usual sub– and equivalence, we have different ways to define a meet function $\curlywedge$ for arbitrary arguments φ and ψ. For example,

$$\varphi \curlywedge \varphi \;:=\; [\,\varphi \wedge \psi\,]$$

would be the most obvious choice (the default version). But in fact, we have infinitely many options for the right side of this definition:

$$[\,\psi \wedge \varphi\,] \qquad [\,\varphi \wedge \psi \wedge \varphi\,] \qquad [\,\psi \leftrightarrow \varphi \leftrightarrow \mathbf{t}\,] \qquad \ldots$$

All these approaches produce equivalent results, each one *a* (not *the*) greatest lower bound of φ and ψ. However, these alternatives all produce formulas that increase in size. Alternatively, there are also more sophisticated versions where the result of all operations like $\varphi \curlywedge \psi$ always is in a certain *normal* or even *canonical* form.[5]

2.1.7 Logical system and Lindenbaum algebras

We do not really care here, but we could say that a given interpretation structure with its quasi–ordered set $\langle FORM, \Rightarrow, \Leftrightarrow \rangle$ is (or makes) a logical system, if it is possible to define such a quasi–boolean algebra at all.

There is an alternative criterion for that: we construct the *quotient–structure*[6] of the given quasi–ordered set. If and only if this poset has all the properties of a *boolean lattice*, then the original structure is a logical system in our sense. This boolean quotient structure of a given quasi–ordered set of formulas is usually called the Lindenbaum or Lindenbaum–Tarski algebra.

2.2 Traditional propositional logic

2.2.1 Definition

Pfm (A) the propositional formula set on a given A is recursively defined to comprise the following expressions:

$([a])$	(atomic formula)
$\neg\varphi$	(negation)
$[\,\varphi_1 \wedge \ldots \wedge \varphi_n\,]$	(conjunction)
$[\,\varphi_1 \vee \ldots \vee \varphi_n\,]$	(disjunction)

for all $a \in A$ and $\varphi, \varphi_1, \ldots, \varphi_n \in \mathbf{Pfm}(A)$ with $n \in \mathbb{N}$.

2.2.2 Remark

[3] I like the systematics in the following terminology and notation: *subvalence* "$\Rightarrow$", *equivalence* "$\Leftrightarrow$", *subjunction* "$\rightarrow$", and *equijunction* "$\leftrightarrow$".

[4] $\langle FORM, \Rightarrow, \Leftrightarrow \rangle$ is a quasi–ordered set iff $\Rightarrow$ is a *quasi–order* (i.e. transitive and reflexive) relation on $FORM$ and $\Leftrightarrow$ is its equivalence relation (i.e. $\varphi \Leftrightarrow \psi$ iff $\varphi \Rightarrow \psi$ and $\psi \Rightarrow \varphi$).

[5] See e.g. *Theory and implementation of efficient canonical systems for sentential calculus, based on Prime Normal Forms* on *www.bucephalus.org*.

[6] $\Leftrightarrow$ is an equivalence relation on $FORM$. Its equivalence classes are the $\tilde{\varphi} := \{\psi \in FORM \mid \psi \Leftrightarrow \varphi\}$, for $\varphi \in FORM$. The overall quotient set $FORM/\Leftrightarrow$ is the collection of all these equivalent classes, and the subvalence relation is redefined on it by putting $\tilde{\varphi} \Rightarrow \tilde{\psi}$ iff $\varphi \Rightarrow \psi$. Finally, $\langle FORM/\Leftrightarrow, \Rightarrow \rangle$ is the wanted quotient structure. A quotient structure of a quasi–ordered set is always a poset or (partially) ordered set.

(1) Most authors don't distinguish between atoms and atom formulas, i.e. they write "a" instead of our "$\llparenthesis a \rrparenthesis$". For example, they rather write

"$[\, a \vee \neg b \,]$" instead of our "$[\, \llparenthesis a \rrparenthesis \vee \neg \llparenthesis b \rrparenthesis \,]$"

We call that the convenient form, and in most examples, we also use that version to make things more readable. However and strictly speaking, we insist on the proper "$\llparenthesis a \rrparenthesis$" version for atom formulas. This keeps things clear, especially when it comes to "higher order" issues, where the atom "a" itself can be a more complex form and things might become ambiguous.

(2) The square brackets "$[\ldots]$" are also used as part of the syntax and extra distinction of formulas from terms like "$\mathbf{0} \wedge (\mathbf{1} \vee \mathbf{0})$", which are not formulas, but applications. We don't apply the usual preference rules to eliminate brackets, either.

(4) We defined the conjunction and disjunction for any finite number $n \in \mathbb{N}$ of arguments and we write

- **(a)** $[\wedge]$ and $[\vee]$ for nullary (i.e. $n = 0$), and
- **(b)** $[\wedge \varphi_1]$ and $[\vee \varphi_1]$ for unary conjunctions and disjunctions, respectively.

(5) We extend the stock of expressions by introducing new junctions as abbreviations for more complex formulas:

- **(a)** $\mathbf{t} := [\wedge]$ the true symbol
- **(b)** $\mathbf{f} := [\vee]$ the false symbol
- **(c)** $[\varphi_1 \rightarrow \ldots \rightarrow \varphi_n] := [[\neg\varphi_1 \vee \varphi_2] \wedge \ldots \wedge [\neg\varphi_{n-1} \vee \varphi_n]]$ the subjunction
- **(d)** $[\varphi_1 \leftrightarrow \ldots \leftrightarrow \varphi_n] := [[\neg\varphi_1 \wedge \ldots \wedge \neg\varphi_n] \vee [\varphi_1 \wedge \ldots \wedge \varphi_n]]$ the equijunction

2.2.3 Example

Example formulas of $\mathbf{Pfm}(A)$ for $A = \{a, b, c, \ldots\}$ are the following

- **(i)** $[\, a \wedge \neg b \,]$, which is the convenient form for $[\, \llparenthesis a \rrparenthesis \wedge \neg \llparenthesis b \rrparenthesis \,]$
- **(ii)** $[\, \mathbf{t} \wedge \neg [\neg b \vee a] \wedge b \,]$ wich is the convenient form for $[\, \mathbf{t} \wedge \neg [\neg \llparenthesis b \rrparenthesis \vee \llparenthesis a \rrparenthesis] \wedge \llparenthesis b \rrparenthesis \,]$
- **(iii)** $[\, a \leftrightarrow \mathbf{f} \leftrightarrow \neg\neg a \,]$ which is an abbreviation for the convenient form $[[\neg a \wedge \neg \mathbf{f} \wedge \neg\neg\neg a] \vee [a \wedge \mathbf{f} \wedge \neg\neg a]]$

2.2.4 Remark

A *propositional formula* turn into a *propositions*, i.e. an either false or true statement, by assigning *bit values* to the given *bit variables* or *atoms*. In other words, an *interpretation* for propositional formulas is a characteristic function $\omega : A \longrightarrow \mathbb{B}$ on the given atom class A.
Recall 1.3.2, that $(A \longrightarrow \mathbb{B}) = \mathbb{B}^1_A$. In terms of the bit table terminology and notation, the *bit line set* $\mathbb{B}^1_A$ is the interpretation class for $\mathbf{Pfm}(A)$. We prefer to write $\mathbb{B}^1_A$ instead of $A \longrightarrow \mathbb{B}$ from now on, because this prepares the generalization to hyperpropositional logic later on.
Also recall 2.1.2, that we have three equivalent versions to actually define the interpretation structure for propositional logic. We define them alltogether next in 2.2.5 and just repeat in 2.2.8, that all three are equivalent.

2.2.5 Definition

Given an atom set A.

The propositional model relation on A

$$\models_A : \mathbb{B}^1_A \rightsquigarrow \mathbf{Pfm}(A)$$

is defined as follows:

$$\begin{array}{rcl} \omega \models_A \llparenthesis a \rrparenthesis & \text{iff} & \omega(a) = \mathbf{1} \\ \omega \models_A \neg\varphi & \text{iff} & \omega \not\models_A \varphi \\ \omega \models_A [\varphi_1 \wedge \ldots \wedge \varphi_n] & \text{iff} & \omega \models_A \varphi_i \text{ for all } i \\ \omega \models_A [\varphi_1 \vee \ldots \vee \varphi_n] & \text{iff} & \omega \models_A \varphi_i \text{ for some } i \end{array}$$

for all $\omega \in \mathbb{B}^1_A$, $a \in A$ and $\varphi, \varphi_1, \ldots, \varphi_n \in \mathbf{Pfm}(A)$.

The propositional model class function on A,

$$\mathbf{Mod}_A : \mathbf{Pfm}(A) \longrightarrow \mathbf{P}(\mathbb{B}^1_A)$$

is defined as follows:

$$\begin{array}{rcl} \mathbf{Mod}_A(\llparenthesis a \rrparenthesis) & := & \{\omega : A \longrightarrow \mathbb{B} \mid \omega(a) = \mathbf{1}\} \\ \mathbf{Mod}_A(\neg\varphi) & := & \mathbb{B}^1_A \setminus \mathbf{Mod}_A(\varphi) \\ \mathbf{Mod}_A([\varphi_1 \wedge \ldots \wedge \varphi_n]) & := & \begin{cases} \mathbb{B}^1_A & \text{if } n = 0 \\ \bigcap_{i=1}^{n} \mathbf{Mod}_A(\varphi_i) & \text{else} \end{cases} \\ \mathbf{Mod}_A([\varphi_1 \vee \ldots \vee \varphi_n]) & := & \bigcup_{i=1}^{n} \mathbf{Mod}_A(\varphi_i) \end{array}$$

The propositional model function on A

$$\mathbf{mod}_A : \mathbf{Pfm}(A) \longrightarrow \mathbb{B}^1_A \longrightarrow \mathbb{B}$$

in other words

$$\mathbf{mod}_A : \mathbf{Pfm}(A) \longrightarrow \mathbb{B}^2_A$$

is defined by:

$$\begin{array}{rcl} \mathbf{mod}_A(\llparenthesis a \rrparenthesis)(\omega) & := & \omega(a) \\ \mathbf{mod}_A(\neg\varphi)(\omega) & := & \mathbf{-mod}_A(\varphi)(\omega) \\ \mathbf{mod}_A([\varphi_1 \wedge \ldots \wedge \varphi_n])(\omega) & := & \bigwedge_{i=1}^{n} \mathbf{mod}_A(\varphi_i)(\omega) \\ \mathbf{mod}_A([\varphi_1 \vee \ldots \vee \varphi_n])(\omega) & := & \bigvee_{i=1}^{n} \mathbf{mod}_A(\varphi_i)(\omega) \end{array}$$

for all $\omega \in \mathbb{B}^1_A$, $a \in A$ and $\varphi, \varphi_1, \ldots, \varphi_n \in \mathbf{Pfm}(A)$.

2.2.6 Remark

We use the conventions:

- **(1)** $\omega \models_A \varphi$ reads "ω satisfies φ" or "φ holds for ω" or "φ is true for ω"
- **(2)** $\omega \not\models_A \varphi$ means that ω does not satisfy φ
- **(3)** $\mathbf{Mod}_A(\varphi)$ is called the model class (on A) of φ.
- **(4)** $\mathbf{mod}_A(\varphi)(\omega)$ is "the bit value of φ at ω",
- **(5)** $\mathbf{mod}_A(\varphi)$ is the model function of φ on A.

2.2.7 Remark

Note, that

$$\omega \models_A [\wedge] \qquad \text{and} \qquad \omega \not\models_A [\vee]$$

in other words

$$\omega \models_A \mathbf{t} \qquad \text{and} \qquad \omega \not\models_A \mathbf{f}$$

which is what one would expect: "true" is true for all interpretations.

2.2.8 Fact

For every set A, each $\varphi \in \mathbf{Pfm}(A)$ and $\omega \in \mathbb{B}^1_A$ holds

$$\omega \models_A \varphi \quad \text{iff} \quad \omega \in \mathbf{Mod}_A(\varphi) \quad \text{iff} \quad \mathbf{mod}_A(\varphi)(\omega) = \mathbf{1}$$

2.2.9 Example

(i) Let $A = \{a, b\}$ and $\varphi \in \mathbf{Pfm}(A)$ be conviently given by $[a \wedge \neg b]$. $\mathbb{B}^1_A$ has four members, namely

$$\omega_1 = \begin{array}{|c|c|}\hline a & b \\ \hline \mathbf{0} & \mathbf{0} \\ \hline\end{array} \quad \omega_2 = \begin{array}{|c|c|}\hline a & b \\ \hline \mathbf{1} & \mathbf{0} \\ \hline\end{array} \quad \omega_3 = \begin{array}{|c|c|}\hline a & b \\ \hline \mathbf{0} & \mathbf{1} \\ \hline\end{array} \quad \omega_4 = \begin{array}{|c|c|}\hline a & b \\ \hline \mathbf{1} & \mathbf{1} \\ \hline\end{array}$$

Accordingly, we have

$$\begin{aligned}
\mathbf{mod}_A(\varphi)(\omega_1) &= \mathbf{mod}_A([a])(\omega_1) \wedge \mathbf{mod}_A(\neg[b])(\omega_1) \\
&= \omega_1(a) \wedge \text{-}\omega_1(b) \\
&= \mathbf{0} \wedge \text{-}\mathbf{0} \\
&= \mathbf{0} \\
\mathbf{mod}_A(\varphi)(\omega_2) &= \omega_2(a) \wedge \text{-}\omega_2(b) = \mathbf{1} \wedge \text{-}\mathbf{0} = \mathbf{1} \\
\mathbf{mod}_A(\varphi)(\omega_3) &= \mathbf{0} \\
\mathbf{mod}_A(\varphi)(\omega_4) &= \mathbf{0}
\end{aligned}$$

so that alltogether

$$\mathbf{Mod}_A(\varphi) = \left\{ \begin{array}{|c|c|}\hline a & b \\ \hline \mathbf{1} & \mathbf{0} \\ \hline\end{array} \right\} \in \mathbf{P}\left(\mathbb{B}^1_A\right)$$

and

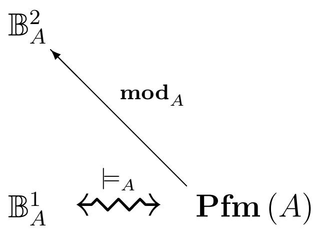
$$\mathbf{mod}_A(\varphi) = \left[\begin{array}{c} \mathbb{B}^1_A \longrightarrow \mathbb{B} \\ \omega \mapsto \begin{cases} \mathbf{0} & \text{if } \omega = \omega_1 \\ \mathbf{1} & \text{if } \omega = \omega_1 \\ \mathbf{0} & \text{if } \omega = \omega_3 \\ \mathbf{0} & \text{if } \omega = \omega_4 \end{cases} \end{array} \right] = \begin{array}{|c|c||c|}\hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \hline\end{array} \in \mathbb{B}^2_A$$

(ii) For the same $A = \{a, b\}$ and the example formulas $\mathbf{f}$ and $\mathbf{t}$ we obtain

$$\mathbf{mod}_A(\mathbf{f}) = \begin{array}{|c|c||c|}\hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \hline\end{array} \qquad \text{and} \qquad \mathbf{mod}_A(\mathbf{t}) = \begin{array}{|c|c||c|}\hline a & b & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline\end{array}$$

2.2.10

In the sequel, we often picture the actual logical system by displaying the correspondence between the syntax on the right and the semantics on the left side. The semantic hierarchy is given by the hierarchy of the bit table classes $\mathbb{B}^k_A$.
So far we only have two bit table classes in this hierarchy on the left and one formula set on the right:



2.2.11 Fact

For each set A and all $\varphi, \varphi_1, \ldots, \varphi_n \in \mathbf{Pfm}(A)$

$$\begin{aligned}
\mathbf{mod}_A(\mathbf{f}) &= \bot^2_A \\
\mathbf{mod}_A(\mathbf{t}) &= \top^2_A \\
\mathbf{mod}_A(\varphi) &= \neg^2_A \mathbf{mod}_A(\varphi) \\
\mathbf{mod}_A([\varphi_1 \wedge \ldots \wedge \varphi_n]) &= \textstyle\prod^2_A \{\mathbf{mod}_A(\varphi_1), \ldots, \mathbf{mod}_A(\varphi_n)\} \\
\mathbf{mod}_A([\varphi_1 \vee \ldots \vee \varphi_n]) &= \textstyle\coprod^2_A \{\mathbf{mod}_A(\varphi_1), \ldots, \mathbf{mod}_A(\varphi_n)\}
\end{aligned}$$

2.2.12 Definition

Let A be a set. We define two relations

$\Rightarrow : \mathbf{Pfm}(A) \rightsquigarrow \mathbf{Pfm}(A)$ — the subvalence relation

$\Leftrightarrow : \mathbf{Pfm}(A) \rightsquigarrow \mathbf{Pfm}(A)$ — the equivalence relation

by putting

$$\begin{aligned}
\varphi \Rightarrow \psi \quad &\text{iff} \quad \forall \omega \in \mathbb{B}^1_A \,.\, \mathbf{mod}_A(\varphi)(\omega) \leq \mathbf{mod}_A(\psi) \\
\varphi \Leftrightarrow \psi \quad &\text{iff} \quad \forall \omega \in \mathbb{B}^1_A \,.\, \mathbf{mod}_A(\varphi)(\omega) = \mathbf{mod}_A(\psi)
\end{aligned}$$

for all $\varphi, \psi \in \mathbf{Pfm}(A)$.

2.2.13 Fact

For every set A and all $\varphi_1, \varphi_2 \in \mathbf{Pfm}(A)$ holds:

$$\begin{aligned}
\varphi_1 \Rightarrow \varphi_2 \quad &\text{iff} \quad \mathbf{Mod}_A(\varphi) \subseteq \mathbf{Mod}_A(\varphi_2) \\
&\text{iff} \quad \mathbf{mod}_A(\varphi_1) \sqsubseteq^2_A \mathbf{mod}_A(\varphi_2) \\
\varphi_1 \Leftrightarrow \varphi_2 \quad &\text{iff} \quad \mathbf{Mod}_A(\varphi) = \mathbf{Mod}_A(\varphi_2) \\
&\text{iff} \quad \mathbf{mod}_A(\varphi_1) = \mathbf{mod}_A(\varphi_2)
\end{aligned}$$

2.2.14 Definition

For every set A, the default propositional formula algebra on A is

$$\mathfrak{Pfm}(A) := \langle \mathbf{Pfm}(A), \Rightarrow, \Leftrightarrow, \mathbf{f}, \mathbf{t}, \wedge, \vee, \text{-} \rangle$$

where

$$\boxed{\varphi \wedge \psi} \;:=\; [\,\varphi \wedge \psi\,] \qquad \text{(conjunction)}$$

$$\boxed{\varphi \vee \psi} \;:=\; [\,\varphi \vee \psi\,] \qquad \text{(disjunction)}$$

$$\boxed{\text{-}\varphi} \;:=\; \neg\varphi \qquad \text{(negation)}$$

2.2.15 Fact

For every set A holds:

$$\mathbf{mod}_A : \mathfrak{Pfm}(A) \hookrightarrow \mathfrak{B}^2_A$$

i.e. $\mathbf{mod}_A$ is an embedding from $\mathfrak{Pfm}(A)$ into $\mathfrak{B}^2_A$.

2.2.16 Remark

The "embedding" in 2.2.15 means as usual, that for all $\varphi, \psi \in$ **Pfm** (A),

$$\varphi \Rightarrow \psi \quad \text{implies} \quad \mathbf{mod}^2_A(\varphi) \sqsubseteq^2_A \mathbf{mod}^2_A(\psi)$$

and

$$\mathbf{mod}^2_A(\varphi \wedge \psi) \;=\; \mathbf{mod}^2_A(\varphi) \sqcap^2_A \mathbf{mod}^2_A(\psi)$$

etc. For example, if $A = \{a, b, c\}$ and

$$\varphi := [\,\neg([b]) \vee ([a])\,] \quad \text{and} \quad \psi := \neg([c])$$

then

$$\mathbf{mod}^2_A(\varphi \wedge \psi) \;=\; \mathbf{mod}^2_A\,([\,[\,\neg([b]) \vee ([a])\,] \wedge \neg([c])\,])$$

=

a	b	c	
0	0	0	1
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0

=

a	b	c	
0	0	0	1
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	1

$\sqcap^2_A$

a	b	c	
0	0	0	1
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0

$$=\; \mathbf{mod}^2_A(\varphi) \sqcap^2_A \mathbf{mod}^2_A(\psi)$$

Figure 5: Traditional propositional logic

Formulas

For every given class A, we define $\mathbf{Pfm}(A)$ the propositional formula class of A as the class comprising the following expressions:

$([a])$	for each $a \in A$	(atomic formula)
$\neg\sigma$	for each $\sigma \in \mathbf{Pfm}(A)$	(negation)
$[\sigma_1 \wedge \ldots \wedge \sigma_n]$	for all $\varphi_1, \ldots \varphi_n \in \mathbf{Pfm}(A)$ with $n \in \mathbb{N}$	(conjunction)
$[\sigma_1 \vee \ldots \vee \sigma_n]$	for all $\varphi_1, \ldots \varphi_n \in \mathbf{Pfm}(A)$ with $n \in \mathbb{N}$	(disjunction)

We write $[\wedge]$ and $[\vee]$ for nullary ($n = 0$) and $[\wedge \varphi_1]$ and $[\vee \varphi_1]$ for unary conjunctions and disjunctions, respectively.

Model function and model class

For every A we define the model function

$$\mathbf{mod}_A : \mathbf{Pfm}(A) \longrightarrow (A \longrightarrow \mathbb{B}) \longrightarrow \mathbb{B}$$

For every $\varphi \in \mathbf{Pfm}(A)$ and $\omega : A \longrightarrow \mathbb{B}$ we give a definition of $\mathbf{mod}_A(\varphi)(\omega)$ by structural induction on the form of φ as follows:

$$\begin{aligned}
\mathbf{mod}_A(([a]))(\omega) &:= \omega(a) \\
\mathbf{mod}_A(\neg\varphi)(\omega) &:= \text{-}\mathbf{mod}_A(\varphi)(\omega) \\
\mathbf{mod}_A([\varphi_1 \wedge \ldots \wedge \varphi_n])(\omega) &:= \bigwedge\{\mathbf{mod}_A(\varphi_1)(\omega), \ldots, \mathbf{mod}_A(\varphi_n)(\omega)\} \\
\mathbf{mod}_A([\varphi_1 \vee \ldots \vee \varphi_n])(\omega) &:= \bigvee\{\mathbf{mod}_A(\varphi_1)(\omega), \ldots, \mathbf{mod}_A(\varphi_n)(\omega)\}
\end{aligned}$$

Furthermore

(α) $\mathbf{mod}_A(\varphi)(\omega)$ is the so–called truth value of φ and (the interpretation) ω

(β) If $\mathbf{mod}_A(\varphi)(\omega) = \mathbf{1}$ we say that "ω is a model for φ" or "ω satisfies φ", and this is expressed by writing $\omega \models \varphi$

(γ) $\mathbf{Mod}_A(\varphi) := \{\omega : A \longrightarrow \mathbb{B} \mid \omega \models \varphi\}$ is the model class of $\varphi \in \mathbf{Pfm}(A)$

(δ) The function $\mathbf{mod}_A(\varphi) : (A \longrightarrow \mathbb{B}) \longrightarrow \mathbb{B}$ is the truth table of φ, and in case of a finite A, this is usually displayed by the typical truth table diagram.

Subvalence and equivalence

For all $\varphi, \psi \in \mathbf{Pfm}(A)$ we define

$\varphi \Rightarrow \psi$ iff $\forall\omega : A \longrightarrow \mathbb{B} \,.\, \mathbf{mod}_A(\varphi)(\omega) \leq \mathbf{mod}_A(\varphi)(\psi)$
iff $\mathbf{Mod}_A(\varphi) \subseteq \mathbf{Mod}_A(\psi)$
iff $\mathbf{mod}_A(\varphi) \sqsubseteq^k_A \mathbf{mod}_A(\psi)$

$\varphi \Leftrightarrow \psi$ iff $\forall\omega : A \longrightarrow \mathbb{B} \,.\, \mathbf{mod}_A(\varphi)(\omega) = \mathbf{mod}_A(\varphi)(\psi)$
iff $\mathbf{Mod}_A(\varphi) = \mathbf{Mod}_A(\psi)$
iff $\mathbf{mod}_A(\varphi) = \mathbf{mod}_A(\psi)$

If $\varphi \Rightarrow \psi$ then we say that "φ is subvalent to ψ" or "φ implies ψ" or "φ entails ψ" or "ψ is a consequence of φ". And $\varphi \Leftrightarrow \psi$ is read as "φ and ψ are equivalent".

The quasi–boolean algebra of propositional formulas

$\mathfrak{Pfm}(A) := \langle \mathbf{Pfm}(A), \Rightarrow, \Leftrightarrow, \mathbf{f}, \mathbf{t}, \wedge, \vee, \text{-} \rangle$ is the default propositional formula algebra, where for all $\varphi, \psi \in \mathbf{Pfm}(A)$

$\mathbf{f} := [\vee]$ $\quad$ $\mathbf{t} := [\wedge]$ $\quad$ $\varphi \wedge \psi := [\varphi \wedge \psi]$ $\quad$ $\varphi \vee \psi := [\varphi \vee \psi]$ $\quad$ $\text{-}\varphi := \neg\varphi$

Theorem

$\mathfrak{Pfm}(A)$ is a quasi–boolean algebra, for every class A.

3 From traditional propositional to hyper–propositional logic

3.1 Applying modal operators to formulas

3.1.1 Definition

Mex $(FORM)$ the modalized or modal expression set of a given set $FORM$ is defined to comprise the following expressions:

$\Diamond\varphi$ (diamond) $\Box\varphi$ (box)

for all $\varphi \in FORM$.
$\Diamond\varphi$ also reads "φ is satisfiable" or "sometimes φ"
$\Box\varphi$ also reads "φ is valid" or "always φ"

3.1.2 Example

Let us consider $\mathbf{Mex}(\mathbf{Pfm}(A))$, the modal expressions on propositional formulas with $A = \{a, b, c, \ldots\}$. Examples are

(i) $\Box[\, a \vee \neg a\,]$
(ii) $\Box[\, a \vee \neg b\,]$
(iii) $\Diamond[\, a \vee \neg b\,]$
(iii) $\Diamond[\, a \leftrightarrow \mathbf{f} \leftrightarrow \neg[\, b \wedge a\,]\,]$

3.1.3 Interpretations for modalized expressions

Given an interpretation structure, made of two sets $FORM$ and INT, together with a model relation $\models : INT \rightsquigarrow FORM$.

What kind of interpretations would suit these new modal expressions on $FORM$? More precisely, what class INT' is an appropriate candidate for a model relation

$$\models' : INT' \rightsquigarrow \mathbf{Mex}\,(FORM)$$

Putting $INT' := INT$ doesn't make sense, because if $\omega \in INT$ and $\varphi \in FORM$, then ω makes φ either true (case $\omega \models \varphi$) or false. However, we are looking for interpretations, that make φ not just "true", but "sometimes true" and "always true".

Therefore, we use another approach and put $INT' := \mathbf{P}\,(INT)$. That makes sense now: for each $\mathcal{M} \in \mathbf{P}\,(INT)$ we define

$$\mathcal{M} \models' \Box\varphi \quad \text{iff} \quad \omega \models \varphi \text{ for all } \omega \in \mathcal{M}$$

$$\mathcal{M} \models' \Diamond\varphi \quad \text{iff} \quad \omega \models \varphi \text{ for some } \omega \in \mathcal{M}$$

And since $\mathbf{P}\,(INT)$ is equivalent to $INT \longrightarrow \mathbb{B}$, we have an equivalent model relation defined on characteristic functions: for every $\Omega : INT \longrightarrow \mathbb{B}$ we define

$$\Omega \models' \Box\varphi \quad \text{iff} \quad \omega \models \varphi \text{ for all } \omega \in \mathbf{Unit}\,(\Omega)$$

$$\Omega \models' \Diamond\varphi \quad \text{iff} \quad \omega \models \varphi \text{ for some } \omega \in \mathbf{Unit}\,(\Omega)$$

3.1.4 Interpretations for modalized propositional formulas

We apply the idea of 3.1.3 to our concrete case of propositional formulas and we use the version on characteristic functions by default.

The modal relation for propositional formulas was

$$\models_A : \mathbb{B}^1_A \rightsquigarrow \mathbf{Pfm}\,(A)$$

Accordingly, the modal relation for modalized propositional formulas is of type

$$\models'_A : (\mathbb{B}^1_A \longrightarrow \mathbb{B}) \rightsquigarrow \mathbf{Mex}(\mathbf{Pfm}\,(A))$$

where again $\mathbb{B}^1_A \longrightarrow \mathbb{B}$ is $\mathbb{B}^2_A$.

3.1.5 Definition

The model relation for modalized propositional formulas on a given set A, is defined by

$$\models'_A : \mathbb{B}^2_A \rightsquigarrow \mathbf{Mex}(\mathbf{Pfm}(A))$$

with

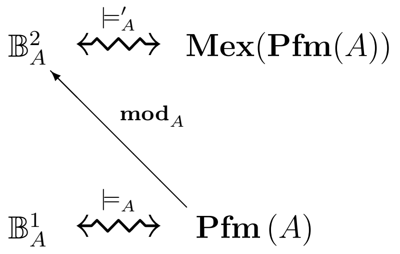
$$\Omega \models'_A \Diamond\varphi \quad \text{iff} \quad \omega \models \varphi \text{ for some } \omega \in \mathbf{Unit}\,(\Omega)$$

$$\Omega \models'_A \Box\varphi \quad \text{iff} \quad \omega \models \varphi \text{ for all } \omega \in \mathbf{Unit}\,(\Omega)$$

for every $\Omega \in \mathbb{B}^2_A$ and each $\sigma \in \mathbf{Pfm}\,(A)$.

3.1.6

When we add these new notions to the diagram of 2.2.10, we obtain the following picture



The new formula class $\mathbf{Mex}(\mathbf{Pfm}\,(A))$ is distinct to the original set $\mathbf{Pfm}\,(A)$ an its semantics is "one degree higher".

3.1.7 Example

Let $A := \{a, b\}$. Let $\Omega \in \mathbb{B}^2_A$ and $\varphi \in \mathbf{Pfm}\,(A)$ be given by

$$\Omega := \begin{array}{|c|c||c|} \hline a & b & \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ \hline \end{array} \quad \text{and} \quad \varphi := [\,\neg a \vee \neg b\,]$$

Then

$$\mathbf{Unit}\,(\Omega) \;=\; \left\{ \begin{array}{|c|c|} \hline a & b \\ \hline 0 & 0 \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline 1 & 1 \\ \hline \end{array} \right\}$$

and

$$\begin{array}{|c|c|} \hline a & b \\ \hline 0 & 0 \\ \hline \end{array} \models_A \varphi \quad \text{because} \quad \text{-}\mathbf{0} \vee \text{-}\mathbf{0} = \mathbf{1} \vee \mathbf{1} = \mathbf{1}$$

$$\begin{array}{|c|c|} \hline a & b \\ \hline 1 & 1 \\ \hline \end{array} \not\models_A \varphi \quad \text{because} \quad \text{-}\mathbf{1} \vee \text{-}\mathbf{1} = \mathbf{0} \vee \mathbf{0} = \mathbf{0}$$

Therefore

- ♣ $\Omega \models'_A \Diamond\varphi$
  i.e. φ is satisfiable in Ω; "sometimes φ" holds in Ω
- ♣ $\Omega \not\models'_A \Box\varphi$
  i.e. φ is not valid in Ω; "always φ" does not hold in Ω

3.1.8 Fact

Given a set A. For every $\varphi \in \mathbf{Pfm}\,(A)$ and $\Omega \in \mathbb{B}^2_A$ holds:

$$\begin{array}{rll} \Omega \models'_A \Box\varphi & \text{iff} & \forall \omega \in \mathbb{B}^1_A \,.\, \Omega(\omega) \le \mathbf{mod}_A(\varphi)(\omega) \\ & \text{iff} & \Omega \sqsubseteq^2_A \mathbf{mod}_A(\varphi) \\ \Omega \models'_A \Diamond\varphi & \text{iff} & \exists \omega \in \mathbb{B}^1_A \,.\, \Omega(\omega) \wedge \mathbf{mod}_A(\varphi)(\omega) = \mathbf{1} \\ & \text{iff} & \Omega \sqcap^2_A \mathbf{mod}_A(\varphi) \neq \bot^2_A \end{array}$$

3.1.9 Fact

Given a set A. For all $\varphi, \psi \in \mathbf{Pfm}\,(A)$ holds:

$$\begin{array}{rll} \mathbf{mod}_A(\psi) \models'_A \Box\varphi & \text{iff} & \psi \Rightarrow \varphi \\ \mathbf{mod}_A(\psi) \models'_A \Diamond\varphi & \text{iff} & [\,\psi \wedge \varphi\,] \not\Leftrightarrow \mathbf{f} \end{array}$$

3.2 Higher degree propositional logic

3.2.1

Modalized propositional formulas $\Diamond\varphi$ and $\Box\varphi$ are statements and it makes sense to combine them to more complex statements again by means of "and", "or", "not" etc. With definition 2.2.1, we already have the means to do that formally: we generalize $\mathbf{Mex}(\mathbf{Pfm}\,(A))$ to $\mathbf{Pfm}(\mathbf{Mex}(\mathbf{Pfm}\,(A)))$.

This way we obtain new formulas such as

$$[\,[\,\Box a \to \Diamond a\,] \vee \neg\neg\Diamond b\,]$$

which would be the *convenient form* (see 2.2.2(1)) of

$$[\,[\,(\![\Box(\![a]\!)]\!) \to (\![\Diamond(\![a]\!)]\!)\,] \vee \neg\neg\Diamond(\![b]\!)\,]$$

The new semantics is defined according to the traditional recipe as well. We increase

$$\models'_A : \mathbb{B}^2_A \rightsquigarrow \mathbf{Mex}(\mathbf{Pfm}\,(A))$$

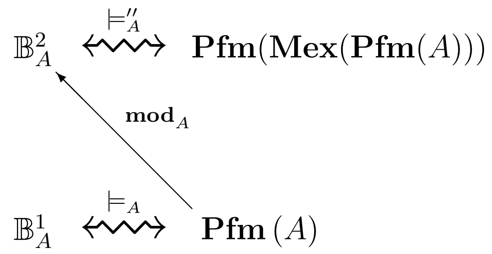
to

$$\models''_A : \mathbb{B}^2_A \rightsquigarrow \mathbf{Pfm}(\mathbf{Mex}(\mathbf{Pfm}\,(A)))$$

by declaring

$$\begin{array}{rll} \Omega \models''_A (\![\sigma]\!) & \text{iff} & \Omega \models'_A \sigma \\ \Omega \models''_A \neg\varphi & \text{iff} & \Omega \not\models''_A \varphi \\ \Omega \models''_A [\,\varphi_1 \wedge \ldots \wedge \varphi_n\,] & \text{iff} & \Omega \models''_A \varphi_1 \text{ and } \ldots \text{ and } \Omega \models''_A \varphi_n \\ \Omega \models''_A [\,\varphi_1 \vee \ldots \vee \varphi_n\,] & \text{iff} & \Omega \models''_A \varphi_1 \text{ or } \ldots \text{ or } \Omega \models''_A \varphi_n \end{array}$$

as usual. That way, our situation from picture 3.1.6 now becomes



3.2.2

The previous picture reveals a pattern that asks for another generalization. Similar to the increase in the semantic hierarchy on the left side from $\mathbb{B}^1_A$ to $\mathbb{B}^2_A$, we have and "upgrade" method for formulas on the right. For every A and $k \ge 1$ we have a "k–degree formula set" F^k_A by putting

$$F^k_A \;:=\; \begin{cases} \mathbf{Pfm}\,(A) & \text{if } k = 1 \\ \mathbf{Pfm}(\mathbf{Mex}(F^{k-1}_A)) & \text{if } k > 1 \end{cases}$$

In analogy to the first steps, we then have a model relation

$$\models^k_A : \mathbb{B}^k_A \rightsquigarrow F^k_A$$

for all $k \ge 2$ as well.
We are going to work out all that in a moment, but we do so with a modified syntax. For degrees $k \ge 2$ we add to each of the operation symbols "$\Diamond$", "$\wedge$", etc. the degree k itself write "$\underset{k}{\Diamond}$", "$\underset{k}{\wedge}$", instead.
The outfit of the resulting formulas is usually redundant and not very appealing. But at this stage, we rather have a transparent than elegant syntax. We want to see for each formula of the new hierarchy, to which level it belongs. Since we allowed nullary junctions like "$[\,\wedge\,]$", we won't be able to see its degree k, unless we write "$[\,\underset{k}{\wedge}\,]$" instead.[7]

[7] Actually, the real motivation for this clear syntax lies beyond this text. It becomes relevant first when we combine the formula sets of all degrees to a single one and compare that with the set of traditional modal formulas (see *www.bucephalus.org* for more information).

3.2.3 Definition

For each set A and $k \in \mathbb{N}$, the k–degree propositional formula set on A,

$$\mathbf{Pfm}^k(A)$$

is recursively defined as follows:

- If $k = 0$ then $\mathbf{Pfm}^0(A) := A$.
- If $k = 1$ then $\mathbf{Pfm}^1(A) := \mathbf{Pfm}\,(A)$.
- If $k \geq 2$ then $\mathbf{Pfm}^k(A)$ is defined to comprise:

$$\left.\begin{array}{r} \underset{k}{\Diamond}\sigma \\ \underset{k}{\Box}\sigma \end{array}\right\} \quad \text{for} \quad \sigma \in \mathbf{Pfm}^{k-1}(A)$$

$$\left.\begin{array}{r} \underset{k}{\neg}\varphi \\ [\,\varphi_1 \underset{k}{\wedge} \dots \underset{k}{\wedge} \varphi_n\,] \\ [\,\varphi_1 \underset{k}{\vee} \dots \underset{k}{\vee} \varphi_n\,] \end{array}\right\} \quad \text{for} \quad \varphi, \varphi_1, \dots, \varphi_n \in \mathbf{Pfm}^k(A)$$

3.2.4 Example

Let $A = \{a, b, c, \dots\}$. A member of $\mathbf{Pfm}^4(A)$ in its convenient form is given by

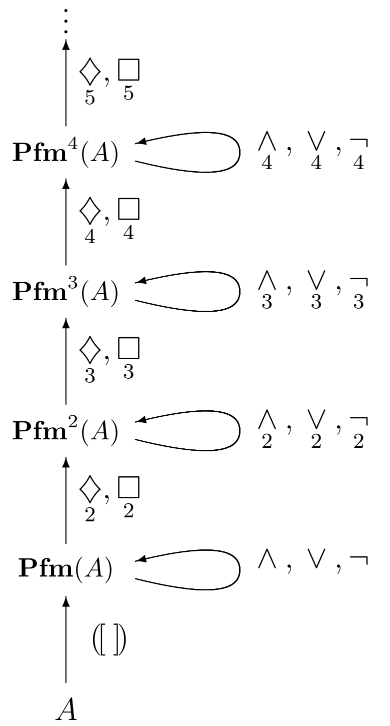
$$\underset{4}{\Diamond}[\underset{3}{\neg}\underset{3}{\Box}[\underset{2}{\Diamond}[a \wedge \neg b] \underset{2}{\vee} \underset{2}{\Box}[c \wedge \neg a]] \underset{3}{\wedge} \underset{3}{\Box}\underset{2}{\Diamond}[a \vee [\wedge]]]$$

We can decompose it to make sure that it is indeed well–formed according to definition 3.2.3:

$$\underset{4}{\Diamond}[\underbrace{\underset{3}{\neg}\underset{3}{\Box}[\underbrace{\underset{2}{\Diamond}\underbrace{[a \wedge \neg b]}_{\in\mathbf{Pfm}(A)} \underset{2}{\vee} \underset{2}{\Box}\underbrace{[c \wedge \neg a]}_{\in\mathbf{Pfm}(A)}}_{\in\mathbf{Pfm}^2(A)}] \underset{3}{\wedge} \underset{3}{\Box}\underbrace{\underset{2}{\Diamond}\underbrace{[a \vee [\wedge]]}_{\in\mathbf{Pfm}(A)}}_{\in\mathbf{Pfm}^2(A)}}_{\in\mathbf{Pfm}^3(A)}]$$

3.2.5 Remark

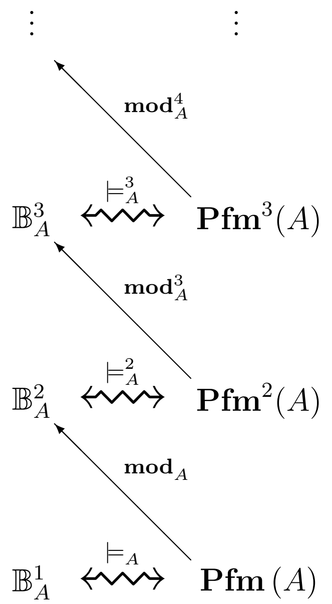
Starting with any given set A we now have an infinite hierarchy of (pairwise disjunct) formula sets: A, $\mathbf{Pfm}\,(A)$, $\mathbf{Pfm}^2(A)$, etc. If we consider the syntactical symbols as functions on these sets, we obtain the following picture:



(Of course, this picture is not entirely correct, because conjunctions and disjunctions don't take formulas, but formula tuples as arguments.)

3.2.6

We define the semantics for this hierarchy of formulas:



On the first level, $\models_A$ is traditional model relation and $\mathbf{mod}_A$

the model function from ??. For all $k \geq 2$, the $\models^k_A$ and $\mathbf{mod}^{k+1}_A$ are now defined according to our approach so far.

3.2.7 Definition

For every set A and each $k \geq 2$ we define a model relation

$$\models^k_A : \mathbb{B}^k_A \leftrightsquigarrow \mathbf{Pfm}^k(A)$$

as follows:

$$\Omega \models^k_A \underset{k}{\Diamond}\sigma \quad \text{iff} \quad \begin{cases} \exists \omega \in \mathbf{Unit}(\Omega) \;.\; \omega \models_A \sigma & \text{if } k = 2 \\ \exists \omega \in \mathbf{Unit}(\Omega) \;.\; \omega \models^{k-1}_A \sigma & \text{if } k > 2 \end{cases}$$

$$\Omega \models^k_A \underset{k}{\Box}\sigma \quad \text{iff} \quad \begin{cases} \forall \omega \in \mathbf{Unit}(\Omega) \;.\; \omega \models_A \sigma & \text{if } k = 2 \\ \forall \omega \in \mathbf{Unit}(\Omega) \;.\; \omega \models^{k-1}_A \sigma & \text{if } k > 2 \end{cases}$$

$$\text{for all } \Omega \in \mathbb{B}^k_A \text{ and } \sigma \in \begin{cases} \mathbf{Pfm}(A) & \text{if } k = 2 \\ \mathbf{Pfm}^{k-1}(A) & \text{if } k > 2 \end{cases}$$

$$\Omega \models^k_A \underset{k}{\neg}\varphi \quad \text{iff} \quad \Omega \not\models^k_A \varphi$$

$$\Omega \models^k_A [\varphi_1 \underset{k}{\wedge} \ldots \underset{k}{\wedge} \varphi_n] \quad \text{iff} \quad \Omega \models^k_A \varphi_i \text{ for all } i$$

$$\Omega \models^k_A [\varphi_1 \underset{k}{\vee} \ldots \underset{k}{\vee} \varphi_n] \quad \text{iff} \quad \Omega \models^k_A \varphi_i \text{ for some } i$$

$$\text{for all } \Omega \in \mathbb{B}^k_A \text{ and } \varphi, \varphi_1, \ldots, \varphi_n \in \mathbf{Pfm}^k(A)$$

For every set A and each $k \geq 2$ we define a model function

$$\mathbf{mod}^{k+1}_A : \mathbf{Pfm}^k(A) \longrightarrow \mathbb{B}^k_A \longrightarrow \mathbb{B}$$

by putting

$$\mathbf{mod}^{k+1}_A(\underset{k}{\Diamond}\sigma)(\Omega) := \begin{cases} \bigvee_{\omega \in \mathbf{Unit}(\Omega)} \mathbf{mod}_A(\sigma)(\omega) & \text{if } k = 2 \\ \bigvee_{\omega \in \mathbf{Unit}(\Omega)} \mathbf{mod}^k_A(\sigma)(\omega) & \text{if } k > 2 \end{cases}$$

$$\mathbf{mod}^{k+1}_A(\underset{k}{\Box}\sigma)(\Omega) := \begin{cases} \bigwedge_{\omega \in \mathbf{Unit}(\Omega)} \mathbf{mod}_A(\sigma)(\omega) & \text{if } k = 2 \\ \bigwedge_{\omega \in \mathbf{Unit}(\Omega)} \mathbf{mod}^k_A(\sigma)(\omega) & \text{if } k > 2 \end{cases}$$

$$\text{for all } \Omega \in \mathbb{B}^k_A \text{ and } \sigma \in \begin{cases} \mathbf{Pfm}(A) & \text{if } k = 2 \\ \mathbf{Pfm}^{k-1}(A) & \text{if } k > 2 \end{cases}$$

$$\mathbf{mod}^{k+1}_A(\underset{k}{\neg}\varphi)(\Omega) := \text{-}\mathbf{mod}^{k+1}_A(\varphi)(\Omega)$$

$$\mathbf{mod}^{k+1}_A([\varphi_1 \underset{k}{\wedge} \ldots \underset{k}{\wedge} \varphi_n])(\Omega) := \overset{n}{\underset{i=1}{\wedge}} \mathbf{mod}^{k+1}_A(\varphi_i)(\Omega)$$

$$\mathbf{mod}^{k+1}_A([\varphi_1 \underset{k}{\vee} \ldots \underset{k}{\vee} \varphi_n])(\Omega) := \overset{n}{\underset{i=1}{\vee}} \mathbf{mod}^{k+1}_A(\varphi_i)(\Omega)$$

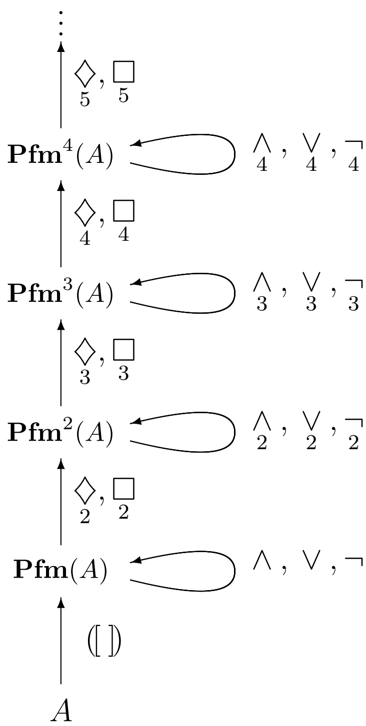
$$\text{for all } \Omega \in \mathbb{B}^k_A \text{ and } \varphi, \varphi_1, \ldots, \varphi_n \in \mathbf{Pfm}^k(A).$$

3.3 The foundation problem and its solution

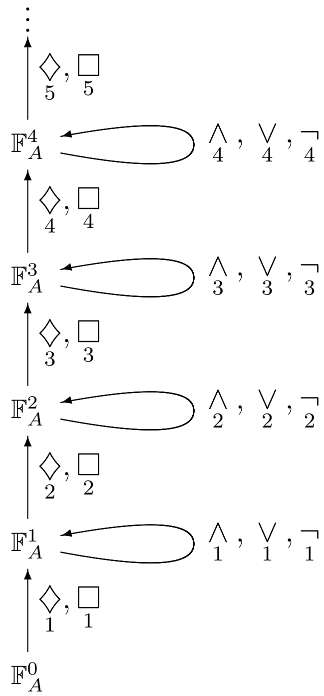
3.3.1

We now have a syntactical hierarchy of *higher–degree propositional formulas* and a parallel semantical hierarchy of *bit tables*, as displayed in the diagramm of 3.2.6. But something is not perfect, yet. Its first level $\mathbf{Pfm}(A)$ itself doesn't fit in entirely and is asking for the possibilities of a more elegant design.

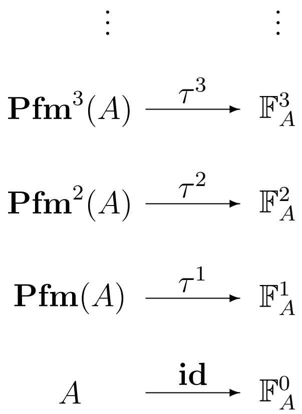
Consider the picture from 3.2.5 again



When we replace the atom formula symbol "([ ])" by "$\underset{1}{\Diamond}$" and "$\underset{1}{\Box}$", we obtain a much more elegant syntax. With new names "$\mathbb{F}^k_A$" for the formula sets this is:

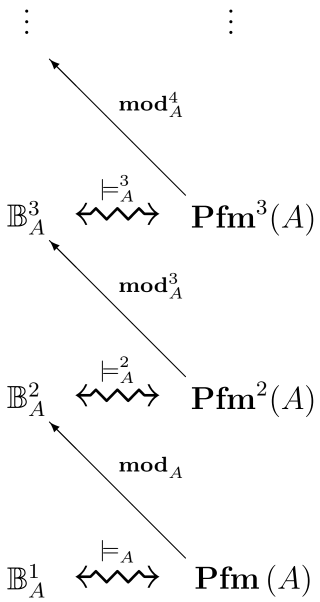


For every degree k we have a translator τ^k,

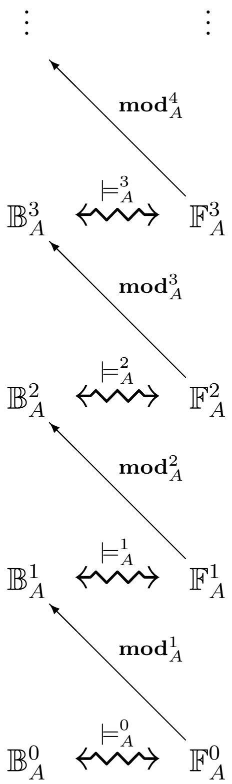


and all these τ^k are "natural" or "conservative" in the sense that they translation conjunctions into conjunctions, negations into negations etc. In other words, all these translations are determined by the *atomic translator*, which tells us how the atomic formulas "$[\![a]\!]$" in $\mathbf{Pfm}\,(A)$ are translated into $\mathbb{F}^1_A$.

Together with this quest for a modification of the syntax goes a change in semantics. Recall the semantical hierarchy so far:



which is defined recursively as well. Does this hierarchy have a nicer foundation, more systematically built on one level lower?



It would be nice to have a system where all this fits together. We call this quest the "foundation problem" and we will see, that it has a very nice solution. But first of all, let us work out the definitions for a proper statement of the problem in 3.3.6.

3.3.2 Definition

Given a set A and $k \in \mathbb{N}$. The hyper–propositional formula set on (carrier) A and (degree) k, written

$$\mathbb{F}^k_A$$

is recursively defined as follows:

- If $k = 0$ then $\mathbb{F}^0_A := A$.
- If $k \geq 1$ we define $\mathbb{F}^k_A$ to comprise the following expressions:

$$\left.\begin{array}{c} \underset{k}{\Diamond}\sigma \\ \underset{k}{\Box}\sigma \end{array}\right\} \quad \text{for} \quad \sigma \in \mathbb{F}^{k-1}_A$$

$$\left.\begin{array}{c} \underset{k}{\neg}\varphi \\ [\,\varphi_1 \underset{k}{\wedge} \ldots \underset{k}{\wedge} \varphi_n\,] \\ [\,\varphi_1 \underset{k}{\vee} \ldots \underset{k}{\vee} \varphi_n\,] \end{array}\right\} \quad \text{for} \quad \varphi, \varphi_1, \ldots, \varphi_n \in \mathbb{F}^k_A$$

3.3.3 Definition

An atomic model relation on a given set A is a relation

$$\rho : A \rightsquigarrow A$$

Each such atomic model relation ρ induces a k–degree model relation of ρ

$$\models^k_\rho : \mathbb{B}^k_A \rightsquigarrow \mathbb{F}^k_A$$

for every $k \in \mathbb{N}$, which is recursively defined as follows:

- If $k = 0$ then

$$\alpha \models^0_\rho \varphi \quad \text{iff} \quad \alpha\, \rho\, \varphi$$

for every $\alpha \in \mathbb{B}^0_A = A$ and $\varphi \in \mathbb{F}^0_A = A$

- If $k > 0$ then

$$\Omega \models^k_\rho \underset{k}{\Diamond}\sigma \quad \text{iff} \quad \omega \models^{k-1}_\rho \sigma \text{ for some } \omega \in \mathbf{Unit}\,(\Omega)$$

$$\Omega \models^k_\rho \underset{k}{\Box}\sigma \quad \text{iff} \quad \omega \models^{k-1}_\rho \sigma \text{ for all } \omega \in \mathbf{Unit}\,(\Omega)$$

$$\Omega \models^k_\rho \underset{k}{\neg}\varphi \quad \text{iff} \quad \Omega \not\models^k_A \varphi$$

$$\Omega \models^k_\rho [\,\varphi_1 \underset{k}{\wedge} \ldots \underset{k}{\wedge} \varphi_n\,] \quad \text{iff} \quad \Omega \models^k_\rho \varphi_i \text{ for all } i$$

$$\Omega \models^k_\rho [\,\varphi_1 \underset{k}{\vee} \ldots \underset{k}{\vee} \varphi_n\,] \quad \text{iff} \quad \Omega \models^k_\rho \varphi_i \text{ for some } i$$

for all $\Omega \in \mathbb{B}^k_A$, all $\sigma \in \mathbb{F}^{k-1}_A$ and $\varphi, \varphi_1, \ldots, \varphi_n \in \mathbb{F}^{k-1}_A$

3.3.4 Definition

An atomic translator of a given set A is a function

$$\tau : A \longrightarrow \mathbb{F}^1_A$$

Each such atomic translator τ induces a k–degree translator of τ

$$\tau^k : \mathbf{Pfm}^k(A) \longrightarrow \mathbb{F}^k_A$$

for every $k \geq 1$, which is recursively defined as follows:

- If $k = 1$ then

$$\begin{aligned}
\tau^1\left(([a])\right) &:= \tau(a) \\
\tau^1\left(\neg\varphi\right) &:= \underset{1}{\neg}\tau^1(\varphi) \\
\tau^1\left([\,\varphi_1 \wedge \ldots \wedge \varphi_n\,]\right) &:= [\,\tau^1(\varphi_1) \underset{1}{\wedge} \ldots \underset{1}{\wedge} \tau^1(\varphi_n)\,] \\
\tau^1\left([\,\varphi_1 \vee \ldots \vee \varphi_n\,]\right) &:= [\,\tau^1(\varphi_1) \underset{1}{\vee} \ldots \underset{1}{\vee} \tau^1(\varphi_n)\,]
\end{aligned}$$

- If $k \geq 2$ then

$$\begin{aligned}
\tau^k\left(\underset{k}{\Diamond}\sigma\right) &:= \underset{k}{\Diamond}\tau^{k-1}(\sigma) \\
\tau^k\left(\underset{k}{\Box}\sigma\right) &:= \underset{k}{\Box}\tau^{k-1}(\sigma) \\
\tau^k\left(\underset{k}{\neg}\varphi\right) &:= \underset{k}{\neg}\tau^k(\varphi) \\
\tau^k\left([\,\varphi_1 \underset{k}{\wedge} \ldots \underset{k}{\wedge} \varphi_n\,]\right) &:= [\,\tau^k(\varphi_1) \underset{k}{\wedge} \ldots \underset{k}{\wedge} \tau^k(\varphi_n)\,] \\
\tau^k\left([\,\varphi_1 \underset{k}{\vee} \ldots \underset{k}{\vee} \varphi_n\,]\right) &:= [\,\tau^k(\varphi_1) \underset{k}{\vee} \ldots \underset{k}{\vee} \tau^k(\varphi_n)\,]
\end{aligned}$$

3.3.5 Example

An example of an atom translation is given by

$$\tau \;:=\; \begin{bmatrix} A \longrightarrow \mathbb{F}^1_A \\ a \mapsto \underset{1}{\Diamond}a \end{bmatrix}$$

If we take example 3.2.4 again, where $\varphi \in \mathbf{Pfm}^4(A)$ was given by

$$\varphi \;=\; \underset{4}{\Diamond}[\,\underset{3}{\neg}\underset{3}{\Box}[\,\underset{2}{\Diamond}[\,a \wedge \neg b\,] \underset{2}{\vee} \underset{2}{\Box}[\,c \wedge \neg a\,]\,] \underset{3}{\wedge} \underset{3}{\Box}\underset{2}{\Diamond}[\,a \vee [\wedge]\,]\,]$$

then the translation $\tau^4(\varphi) \in \mathbb{F}^4_A$ is

$$\underset{4}{\Diamond}[\,\underset{3}{\neg}\underset{3}{\Box}[\,\underset{2}{\Diamond}[\,\underset{1}{\Diamond}a \underset{1}{\wedge} \underset{1}{\neg}\underset{1}{\Diamond}b\,] \underset{2}{\vee} \underset{2}{\Box}[\,\underset{1}{\Diamond}c \underset{1}{\wedge} \underset{1}{\neg}\underset{1}{\Diamond}a\,]\,] \underset{3}{\wedge} \underset{3}{\Box}\underset{2}{\Diamond}[\,\underset{1}{\Diamond}a \underset{1}{\vee} [\underset{1}{\wedge}]\,]\,]$$

3.3.6 Definition foundation problem

Given some set A. The foundation problem is the question, if an atom translator τ and an atomic relation ρ exist, such that

$$\Omega \models^k_\rho \tau^k(\varphi) \quad \text{iff} \quad \Omega \models^k_A \varphi$$

for each $k \geq 1$ and all $\Omega \in \mathbb{B}^k_A$ and $\varphi \in \mathbf{Pfm}^k(A)$.

3.3.7 Fact the solution of the foundation problem

The foundation problem has a solution, given by the atomic translator

$$\mathbf{t} : A \longrightarrow \mathbb{F}^1_A \quad \text{with} \quad \mathbf{t}\,(a) := \underset{1}{\Diamond}a \quad \text{for all} \quad a \in A$$

and the identity on A as the atomic model relation, i.e.

$$\mathbf{id} : A \rightsquigarrow A \quad \text{with} \quad \mathbf{id}\,(a) = a \quad \text{for all} \quad a \in A$$

3.4 The final version of hyper–propositional logic

3.4.1 Definition

For every set A and each $k \in \mathbb{N}$ we define a system of hyper–propositional logic of (carrier) A and (degree) k as follows:

(1) Syntax:
The hyper–propositional formula set $\mathbb{F}^k_A$ was defined in 3.3.2.

(2) Interpretation structure:
The model relation (of k and A)

$$\models^k_A : \mathbb{B}^k_A \rightsquigarrow \mathbb{F}^k_A$$

is defined by

$$\Omega \models^k_A \varphi \quad \text{iff} \quad \Omega \models^k_{\mathbf{id}} \varphi \qquad \text{for all } \Omega \in \mathbb{B}^k_A \text{ and } \varphi \in \mathbb{F}^k_A$$

According to 2.1.3, this also provides us, for every $\varphi \in \mathbb{F}^k_A$, with the definition of the model class

$$\mathbf{Mod}^k_A(\varphi) := \{\Omega \in \mathbb{B}^k_A \mid \Omega \models^k_A \varphi\}$$

and a model function or super–model

$$\mathbf{mod}^{k+1}_A(\varphi) := \begin{bmatrix} \mathbb{B}^k_A \longrightarrow \mathbb{B} \\ \Omega \mapsto \begin{cases} \mathbf{1} & \text{if } \Omega \models^k_A \varphi \\ \mathbf{0} & \text{else} \end{cases} \end{bmatrix}$$

(3) Quasi–boolean order:
According to 2.1.4, we also have a subvalence and equivalence relation on $\mathbb{F}^k_A$, defined by

$$\varphi \Rightarrow^k_A \psi \quad \text{iff} \quad \forall \Omega \in \mathbb{B}^k_A \,.\, \left(\Omega \models^k_A \varphi \text{ implies } \Omega \models^k_A \psi\right)$$
$$\varphi \Leftrightarrow^k_A \psi \quad \text{iff} \quad \forall \Omega \in \mathbb{B}^k_A \,.\, \left(\Omega \models^k_A \varphi \text{ iff } \Omega \models^k_A \psi\right)$$

(4) Quasi–boolean lattice of formulas
According to 2.1.5 and similar to the default propositional formula $\mathfrak{Pfm}A$ we define the default hyper–propositional formula algebra of A and k as

$$\mathfrak{F}^k_A := \langle \mathbb{F}^k_A, \Rightarrow^k_A, \Leftrightarrow^k_A, \mathbf{f}^k, \mathbf{t}^k, \wedge^k, \vee^k, \neg^k \rangle$$

where

$$\mathbf{f}^k := [\bigvee_k] \qquad \mathbf{t}^k := [\bigwedge_k]$$
$$\varphi \wedge^k \psi := [\varphi \bigwedge_k \psi] \qquad \varphi \vee^k \psi := [\varphi \bigvee_k \psi]$$
$$\neg^k \varphi := \neg_k \varphi$$

3.4.2 Fact

For every set A and $k \geq 1$,

- $\mathfrak{F}^k_A$ is a quasi–boolean algebra
- $\mathbf{mod}^{k+1}_A$ is an embedding of $\mathfrak{F}^k_A$ into $\mathfrak{B}^{k+1}_A$

3.4.3 Fact

For each set A holds: $\mathbf{t}^1 : \mathbf{Pfm}(A) \hookrightarrow \mathbb{F}^1_A$, i.e. $\mathbf{t}^1$ is an embedding of $\mathfrak{Pfm}A$ into $\mathfrak{F}^1_A$.

3.4.4 Remark

In figure 6 we summarize the syntax and semantics of hyper–propositional logic. Figure 5 was a summarized repetition of the syntax and semantics of traditional logic. Figure 7 repeats 3.4.3.

Figure 6: Hyper–propositional logic

Formulas

For every set A and $k \in \mathbb{N}$ we define $\mathbb{F}^k_A$ the (hyper–propositional) formulas of carrier A and degree k recursively as follows

(i) If $k = 0$ then $\mathbb{F}^0_A := A$.

(ii) If $k > 0$ then $\mathbb{F}^k_A$ comprises the following expressions:

$$\left.\begin{array}{ll} \underset{k}{\Diamond}\sigma & \text{(diamond)} \\ \underset{k}{\Box}\sigma & \text{(box)} \end{array}\right\} \text{for all } \sigma \in \mathbb{F}^{k-1}_A \qquad \left.\begin{array}{ll} \underset{k}{\neg}\varphi & \text{(negation)} \\ [\,\varphi_1 \underset{k}{\wedge} \dots \underset{k}{\wedge} \varphi_n\,] & \text{(conjunction)} \\ [\,\varphi_1 \underset{k}{\vee} \dots \underset{k}{\vee} \varphi_n\,] & \text{(disjunction)} \end{array}\right\} \text{for all } \varphi, \varphi_1, \dots, \varphi_n \in \mathbb{F}^k_A$$

We write $[\underset{k}{\wedge}]$ and $[\underset{k}{\vee}]$ for nullary, and $[\underset{k}{\wedge}\varphi_1]$ and $[\underset{k}{\vee}\varphi_1]$ for unary conjunctions and disjunctions, respectively.

Super–models and model classes

For every class A and every natural number $k \in \mathbb{N}$ we define the (super–) model function

$$\mathbf{mod}^{k+1}_A : \mathbb{F}^k_A \longrightarrow \mathbb{B}^k_A \longrightarrow \mathbb{B}$$

where $\mathbf{mod}^{k+1}_A(\varphi)(\Omega)$ is defined, for each $\varphi \in \mathbb{F}^k_A$ and $\Omega \in \mathbb{B}^k_A$, by induction on k as follows:

(i) If $k = 0$ then $\varphi \in \mathbb{F}^0_A = A$ and $\Omega \in \mathbb{B}^0_A = A$ and

$$\mathbf{mod}^1_A(\varphi)(\Omega) \;:=\; \begin{cases} \mathbf{1} & \text{if } \varphi = \Omega \\ \mathbf{0} & \text{else} \end{cases}$$

(ii) If $k > 0$, we define by structural induction on the form of φ as follows:

$$\begin{array}{rcl} \mathbf{mod}^{k+1}_A\left(\underset{k}{\Diamond}\sigma\right)(\Omega) & := & \bigvee\{\mathbf{mod}^k_A(\sigma)(\omega) \mid \omega \in \mathbb{B}^{k-1}_A, \Omega(\omega) = \mathbf{1}\} \\ \mathbf{mod}^{k+1}_A\left(\underset{k}{\Box}\sigma\right)(\Omega) & := & \bigwedge\{\mathbf{mod}^k_A(\sigma)(\omega) \mid \omega \in \mathbb{B}^{k-1}_A, \Omega(\omega) = \mathbf{1}\} \\ \mathbf{mod}^{k+1}_A\left(\underset{k}{\neg}\varphi\right)(\Omega) & := & \text{-}\mathbf{mod}^{k+1}_A(\varphi)(\Omega) \\ \mathbf{mod}^{k+1}_A\left([\,\varphi_1 \underset{k}{\wedge} \dots \underset{k}{\wedge} \varphi_n\,]\right)(\Omega) & := & \bigwedge\left\{\mathbf{mod}^{k+1}_A(\varphi_1)(\Omega), \dots, \mathbf{mod}^{k+1}_A(\varphi_n)(\Omega)\right\} \\ \mathbf{mod}^{k+1}_A\left([\,\varphi_1 \underset{k}{\vee} \dots \underset{k}{\vee} \varphi_n\,]\right)(\Omega) & := & \bigvee\left\{\mathbf{mod}^{k+1}_A(\varphi_1)(\Omega), \dots, \mathbf{mod}^{k+1}_A(\varphi_n)(\Omega)\right\} \end{array}$$

Furthermore:

(α) $\mathbf{mod}^{k+1}_A(\varphi)(\Omega) \in \mathbb{B}$ is the so–called truth value of φ and (the interpretation) Ω

(β) If $\mathbf{mod}^{k+1}_A(\varphi)(\Omega) = \mathbf{1}$ we say that "Ω is a model for φ" or "Ω satisfies φ", and this is also expressed by writing $\Omega \models \varphi$.

(γ) Accordingly and for each given $\varphi \in \mathbb{F}^k_A$, its model class is a subset of $\mathbb{B}^k_A$, defined by

$$\mathbf{Mod}^k_A(\varphi) \;:=\; \{\Omega \in \mathbb{B}^k_A \mid \mathbf{mod}^{k+1}_A(\varphi)(\Omega) = \mathbf{1}\}$$

(δ) Note, that for each $k \in \mathbb{N}$, $\mathbf{mod}^{k+1}_A : \mathbb{F}^k_A \longrightarrow \mathbb{B}^{k+1}_A$, because $\mathbb{B}^{k+1}_A = (\mathbb{B}^k_A \longrightarrow \mathbb{B})$ (hence the superscript "$k+1$" in "$\mathbf{mod}^{k+1}_A$"). We call $\mathbf{mod}^{k+1}_A(\varphi) \in \mathbb{B}^{k+1}_A$ the super–model or truth table of $\varphi \in \mathbb{F}^k_A$.

Subvalence and equivalence

Given A and k, we define two relations on $\mathbb{F}^k_A$. For all $\varphi, \psi \in \mathbb{F}^k_A$ let

$$\begin{array}{lll} \varphi \Rightarrow^k_A \psi & \text{iff} & \forall\Omega \in \mathbb{B}^k_A \,.\, (\Omega \models \varphi \text{ implies } \Omega \models \psi) \\ & \text{iff} & \mathbf{Mod}^k_A(\varphi) \subseteq \mathbf{Mod}^k_A(\psi) \\ & \text{iff} & \mathbf{mod}^{k+1}_A(\varphi) \sqsubseteq^{k+1}_A \mathbf{mod}^{k+1}_A(\psi) \end{array} \qquad \begin{array}{lll} \varphi \Leftrightarrow^k_A \psi & \text{iff} & \forall\Omega \in \mathbb{B}^k_A \,.\, (\Omega \models \varphi \text{ iff } \Omega \models \psi) \\ & \text{iff} & \mathbf{Mod}^k_A(\varphi) = \mathbf{Mod}^k_A(\psi) \\ & \text{iff} & \mathbf{mod}^{k+1}_A(\varphi) = \mathbf{mod}^{k+1}_A(\psi) \end{array}$$

If $\varphi \Rightarrow^k_A \psi$ then we say that "φ is subvalent to ψ" or "φ implies ψ" or "φ entails ψ" or "ψ is a consequence of φ". And $\varphi \Leftrightarrow^k_A \psi$ is read as "φ and ψ are equivalent".

The quasi–boolean lattice of formulas

$\mathfrak{F}^k_A := \langle \mathbb{F}^k_A, \Rightarrow^k_A, \Leftrightarrow^k_A, \mathbf{f}^k, \mathbf{t}^k, \wedge^k, \vee^k, \neg^k \rangle$ is the default formula algebra of A and k, where for all $\varphi, \psi \in \mathbb{F}^k_A$

$$\mathbf{f}^k := [\underset{k}{\vee}] \qquad \mathbf{t}^k := [\underset{k}{\wedge}] \qquad \varphi \wedge^k \psi := [\varphi \underset{k}{\wedge} \psi] \qquad \varphi \vee^k \psi := [\varphi \underset{k}{\vee} \psi] \qquad \neg^k \varphi := \underset{k}{\neg}\varphi$$

Theorem

$\mathfrak{F}^k_A$ is a quasi–boolean algebra, for every A and $k \geq 1$.

Theorem

$\mathbf{mod}^{k+1}_A : \mathfrak{F}^k_A \hookrightarrow \mathfrak{B}^{k+1}_A$, i.e. $\mathbf{mod}^{k+1}_A : \mathbb{F}^k_A \longrightarrow \mathbb{B}^{k+1}_A$ is an embedding of $\mathfrak{F}^k_A$ into $\mathfrak{B}^{k+1}_A$, for all A and $k \geq 1$.

Figure 7: Embedding traditional propositional into hyper–propositional logic

Definition

For every set A we put

$$\mathbf{t}^1 \;:=\; \left[\begin{array}{c} \mathbf{Pfm}\,(A) \longrightarrow \mathbb{F}^1_A \\ \left[\begin{array}{rcl} ([a]) & \mapsto & \underset{1}{\Diamond} a \\ \neg\sigma & \mapsto & \underset{1}{\neg}\mathbf{t}^1(\sigma) \\ [\,\sigma_1 \wedge \ldots \wedge \sigma_n\,] & \mapsto & [\,\mathbf{t}^1(\sigma_1) \underset{1}{\wedge} \ldots \underset{1}{\wedge} \mathbf{t}^1(\sigma_n)\,] \\ [\,\sigma_1 \vee \ldots \vee \sigma_n\,] & \mapsto & [\,\mathbf{t}^1(\sigma_1) \underset{1}{\vee} \ldots \underset{1}{\vee} \mathbf{t}^1(\sigma_n)\,] \end{array}\right] \end{array}\right]$$

Theorem

$\mathbf{t}^1 : \mathfrak{Pfm}\,(A) \hookrightarrow \mathfrak{F}^1_A$ for every set A, i.e. $\mathbf{t}^1$ is an embedding of $\mathfrak{Pfm}\,(A)$ into $\mathfrak{F}^1_A$.