KRITERION, Nr.11 (1997), pp.3–15

Hannes Leitgeb

A CLASS OF n-VALUED STATEMENT CALCULI: MANY UNIVERSES STATEMENT CALCULUS

0. Introduction

Since the arrival of many-valued logic in the 1920's (Lukasiewicz[1], Post[2]) several suggestions for many-valued statement calculi (for an overview see e.g. Rosser[3], Zinovév[4], [5]) have been made; our considerations are based on a many-valued logic that was proposed by Gotthard Günther in the 1950's – 1970's (most of his papers are collected in Günther[6]). Starting from Hegelian philosophy Günther partly worked out his logic in formal terms and discussed his ideas with some of the leading logicians of the time (e.g. Kurt Gödel). Holding an assistant professorship at the Biological Computer Laboratory (BCL) in Urbana, Illinois (later being professor of philosophy in Hamburg, Germany) Günther had the possibility to develop his logic in tight connection with the arising disciplines of cybernetics and artificial intelligence. Although his work was continued by several authors (Kaehr[7], [8], Mitterauer[9], [10], Thomas[11], Ditterich[12], Na[13]) of which Mitterauer's papers have been most important for us, no semantic/axiomatic description of the Günther-logic, in association with examples and an interpretation, has been proposed. The purpose of this paper is to fill this gap (at least partially) by defining what we call "*Many Universes Logic*" (from now on abbreviated: MUL) in form of a many-valued statement calculus. To avoid any confusation (and because we are aware of the fact that our work is already an interpretation of Günther's) we will not use the terminology of Günther and his followers but more usual terms, plus that of a *logical universe*, which will be defined later. From our point of view the attractiveness of Günther's ideas lies in the fact that he does *not* increase the number of *truth values* from 2 – the classical ones – to n (and interpret them afterwards), but increases the number of *domains* (that will be called logical universes), in which classical 2-valued logic holds, leading to n truth values, the interpretation of which may be *raised from the interpretation of the 2 classical truth values*. Zinovév[5] characterized the Günther-logic as follows: "[It is a] description of connections between different semantic levels of knowledge, every level of which is ruled by a two-valued logic."

Before we introduce definition, semantics and axiomatization of MUL together with some hypotheses regarding its application, we will give examples (and thereby a motivation) of how the step from classical to Many Universes Logic can be taken. In these examples we will try to show how simple operations on computer programs that are written in the language of classical 2-valued logic may lead to MUL.

1. Motivation

Have a look at the following simple program routine (in a PASCAL/MODULA like notation):

Example 1

```
1. IF P(a)=TRUE THEN
2.   Q(b):=FALSE;
3. ELSE
4.   Q(b):=TRUE;
5. END;
```

(where P and Q are predicates, a and b variables defined on the domain of P and Q, and TRUE and FALSE constants representing e.g. integer values 1 and 0; we will always say “TRUE” and “FALSE”, if we refer to the so-called constants in computer programs, we will say “True” and “False” or respectively “T” and “F”, if we refer to the usual truth values of classical 2-valued logic, and we will say “true” and “false”, if we refer to their usual meaning, i.e. applied to statements).

IF the truth value of P(a) is set to, say, TRUE at the start of the program, the truth value of Q(b) will be set to FALSE and the program stops. Mark that nothing essential is changed in the program (and of course its output), if we replace line 3 (“ELSE”) by the following:

```
3’. ELSIF P(a)=FALSE THEN
```

(but also mark that lines 3 and 3’ will differ essentially, if the range of P(a) is larger than the set {TRUE, FALSE} !)

Also everything stays the same, if we define the constants TRUE and FALSE not as integers 1 and 0 but, say, 1 and 2 (if all logical connectives that could occur in the program like “NOT”, “AND”, “OR” etc. are redefined in the same manner, if they haven’t been defined using the constants TRUE and FALSE but directly integers 1 and 0). So much for trivialities.

Now consider two computers that communicate with each other (here to communicate means: to share common objects). Call the program of computer 1 program 1 and respectively that of computer 2 program 2. To enable experiments we will simulate (not adequately, but – for our purpose – sufficiently) computer 1-2 interaction by implementing program 1 and 2 as procedures of a simulation program (that could then be run on one computer), which mutually calls program 1 and 2 (first program 1, secondly program 2, thirdly program 1,...) continuing until the user stops the simulation.

Let e.g. program 1 and 2 be the following:

Example 2

```
1. PROCEDURE Program1;
2. IF P(a)=TRUE THEN
3.   Q(b):=FALSE;
4. ELSIF P(a)=FALSE THEN
5.   Q(b):=TRUE;
6. END;

1. PROCEDURE Program2;
2. IF P(a)=TRUE THEN
3.   Q(b):=TRUE;
4. ELSIF P(a)=FALSE THEN
5.   Q(b):=FALSE;
6. END;
```

(with the same definitions as used in example 1).

If the initial value of P(a) is, say, set to TRUE, the output of the simulation program (which mutually calls program 1 and 2) will be:

Output of Example 2

in integer notation:

	P(a)	Q(b)
Initial value:	1	–
(After calling) PROGRAM 1:	1	0
(After calling) PROGRAM 2:	1	1
(After calling) PROGRAM 1:	1	0
(After calling) PROGRAM 2:	. . .	. . . .

(“–” means that Q(b) has no truth value assigned at the start of the simulation; “. . . .” marks that the output repeats cyclically).

The transition to Many Universes Logic is now accomplished by two steps:

Step 1:
Recall that each program in our examples was formulated in the language of classical 2-valued logic, i.e. there is a set of truth values $\{T_i, F_i\}$ defined for each program i = 1, 2, with $T_i, F_i \in \mathbb{N}_0$ (and of course $T_i \neq F_i$).

Step 2:
To remain within classical 2-valued logic means to choose $T_1 = T_2$ and $F_1 = F_2$; to enter MUL means to generalize by defining a set of (classical) truth values $\{T_1, F_1\}$ for program 1, and a set of (classical) truth values $\{T_2, F_2\}$ for program 2, leaving open the relationship (which is identity or difference between the truth values) between the two sets. This leads to the following definition:

Def. 1: By a *(logical) universe* we mean an ordered pair U = (S, t), where S is a set of two elements, and t is a function, such that t: $S \rightarrow \{T, F\}$ with t bijective.

When we talk about a logical universe $U_i = (S_i, t_i)$ we will often say T_i (the "True" value of universe U_i) instead of $t_i^{-1}(T)$ and F_i (the "False" value of universe U_i) instead of $t_i^{-1}(F)$ (and mostly S_i will be a subset of $\mathbb{N}_0$).

In the case of two universes $U_1 = (S_1, t_1)$, $U_2 = (S_2, t_2)$ there are 7 types (possibilities) of relations between them (up to isomorphy):

(1) $T_1 = 1; F_1 = 0; T_2 = 1; F_2 = 0;$
(2) $T_1 = 1; F_1 = 0; T_2 = 0; F_2 = 1;$
(3) $T_1 = 1; F_1 = 2; T_2 = 2; F_2 = 3;$
(4) $T_1 = 1; F_1 = 2; T_2 = 3; F_2 = 2;$
(5) $T_1 = 2; F_1 = 1; T_2 = 2; F_2 = 3;$
(6) $T_1 = 2; F_1 = 1; T_2 = 3; F_2 = 2;$
(7) $T_1 = 1; F_1 = 2; T_2 = 3; F_2 = 4;$

(up to isomorphy means of course that e.g. T_1=1, F_1=2, T_2=1, F_2=2 would fall under type 1, because $T_1 = T_2$ and $F_1 = F_2$, whereas $T_1 = 3$, $F_1 = 1$, $T_2 = 3$, $F_2 = 2$ would fall under type 5, because $T_1 = T_2$ and $F_1 \neq F_2$, etc.).

Remember that S has to be a set of *two* elements: Therefore always $T_i \neq F_i$ (otherwise $\{T_i, F_i\}$ could not be interpreted as set of classical truth values).

Type 1 represents 2 universes that cannot be differentiated (i.e. the case of classical 2-valued logic is restored). Type 2 consists of two kind of inverse universes – one universe contradicting the other – but is also 2-valued. Types 3 – 6 are 3-valued: S_1 has exactly one element in common with S_2. Type 7 is 4-valued, but S_1 and S_2 have no element in common: In this sense we may view type 7 as the case of 2 "unrelated" classical 2-valued logics.

Now what does all this mean ?

Let us – at first – give the pragmatic answer: We will rewrite example 2, but now we will replace TRUE and FALSE in program 1 by T_1 and F_1, and TRUE and FALSE in program 2 by T_2 and F_2:

Example 2'

```
1. PROCEDURE Program1;
2. IF P(a)=T1 THEN
3.   Q(b):=F1;
4. ELSIF P(a)=F1 THEN
5.   Q(b):=T1;
6. END;

1. PROCEDURE Program2;
2. IF P(a)=T2 THEN
3.   Q(b):=T2;
4. ELSIF P(a)=F2 THEN
5.   Q(b):=F2;
6. END;
```

Now we have to decide of which type (1–7) the 2 universes should be: the selection of type 1 results in the same output as example 2; type 2 corresponds to exchange of line 3 and 5 in program 2; so we select e.g. type 3 and define therefore:

$T_1 := 1; F_1 := 2; T_2 := 2; F_2 := 3;$

The initial value of P(a) in example 2 was set to TRUE: In analogy we may set it now to T_1 (= 1) or T_2 (= 2); let us choose e.g. T_1 :

P(a) := 1;

The output of example 2’ will then be the following:

Output of Example 2’

in integer notation:

	P(a)	Q(b)
Initial value:	1	–
(After calling) PROGRAM 1:	1	2
(After calling) PROGRAM 2:	1	2
(After calling) PROGRAM 1:	1	2
(After calling) PROGRAM 2:	. . . .	. . .

interpretation by program 1:

	P(a)	Q(b)
Initial value:	T_1	–
(After calling) PROGRAM 1:	T_1	F_1
(After calling) PROGRAM 2:	T_1	F_1
(After calling) PROGRAM 1:	T_1	F_1
(After calling) PROGRAM 2:	. . . .	. . .

interpretation by program 2:

	P(a)	Q(b)
Initial value:	#	–
(After calling) PROGRAM 1:	#	T_2
(After calling) PROGRAM 2:	#	T_2
(After calling) PROGRAM 1:	#	T_2
(After calling) PROGRAM 2:	. . . .	. . .

If you look at the simulation results (in integer notation), you will see that program 1 is started and sets the truth value of Q(b) to F_1 = 2, because the truth value of P(a) is T_1 = 1. Afterwards program 2 is called and – *doesn’t do anything*!! As you can see, the truth value of P(a) is 1, which means *beyond* the alternative of T_2 = 2 and F_2 = 3, and this is the reason why *neither* the case P(a) = T_2 *nor* the case P(a) = F_2 is considered in program 2. In classical 2-valued logic this is of course impossible because of the proposition of the excluded middle. In MUL the situation changes in a subtle way: If you stay *within* a logical universe, the excluded middle still holds rigorously (this means practically that when you use program 1 and 2 separately – without “communication” – they still behave as classical 2-valued programs including the validity of the excluded middle, no matter *which* logical universe they have been assigned to), but the excluded middle of one universe may fail in respect to *another* (different) universe !

That’s why we say that in such a case (like example 2’) the statement P(a) is *irrelevant in respect to* universe U_2, though it is *relevant – namely true – in respect to* universe U_1; or generally:

Def. 2: A statement p is called *irrelevant in respect to* a universe U = (S, t), if the truth value of p is not element of S.
Otherwise p is called *relevant in respect to* U, which means p is either *true in respect to* U or *false in respect to* U.

(The first, who stated that the concept of “relevance” could have something to do with the Günther logic was John W. Campbell, but Campbell’s remark was not taken up by Günther; Campbell[14].)

In this sense truth is always *relative* in MUL – depending on the universe you consider (and a statement may even be *neither true nor false*, i.e. *irrelevant*, in respect to a universe).

Regarding the phenomenon of irrelevance the reader will certainly be reminded of some typical 3-valued logics, wherein two values are interpreted as True and False, and the third value may be interpreted as "*undefined*" (Kleene[15]), "*indeterminate*" (Lukasiewicz [16]) or "*paradox*" (Bocvar[17]). All these concepts differ from MUL by the fact that a statement in MUL is *always true or false in respect to (at least) one universe*, but *may be irrelevant in respect to another universe*, i.e. irrelevance is no (absolute) truth value but a (relative) interpretation of a truth value that is true or false in respect to some (at least one) universe. With so-called "Relevance Logic" the irrelevance concept of MUL only shares part of the name: the former proposing a new theory of implication, namely entailment.

Besides the integer notation of the output of example 2' we have also written down the output values as they are interpreted from the "point of view" of program 1 and 2, i.e. in respect to universes U_1 and U_2: While P(a) is true (T_1) in respect to U_1, it is irrelevant ("#" shall stand for irrelevant) in respect to U_2. Q(b) is false (F_1) in respect to U_1 and true (T_2) in respect to U_2. As you can see, in MUL statements may be *true and false at the same time*, but just *in respect to different universes*. In this sense the proposition of the forbidden contradiction still holds in respect to *one* universe but may fail to hold in respect to *different* universes (Gupta[18] gives the semantics of a 4-valued statement calculus with one truth value interpreted as "both true and false": the difference is again that there is no truth value for "both true and false" in MUL, but a truth value may be true in respect to one universe and false in respect to another).

Mark that the output of our example program is crucially dependent on the type of relation between universes 1 and 2.

Now we will extend our "motivational" view of MUL by the introduction of logical connectives.

2. Logical Connectives

By a *logical connective* we mean of course an element of the alphabet (we will define an appropriate alphabet in the next section) that has associated a function

$$f: M^k \rightarrow M,$$

where k is the number of places of the connective (M is the set of truth values). We will restrict our study to 1- and 2-place logical connectives (as usual in many-valued logic). The number of 1-place connectives is of course n^n ($n = card(M)$), the number of 2-place connectives is $n^{(n \cdot n)}$.

Because of the vast number of connectives for $n \geq 3$ (e.g. the number of 2-place connectives in a 3-valued logic is $3^{(3 \cdot 3)} = 19683$) their interpretation and classification is of special significance. One way to classify them is to apply usual mathematical concepts (e.g.: which connectives are associative? which connectives are T-norms? etc.) Even more important is the *semantic* interpretation of connectives. This is the reason why, from the beginning, various authors in many-valued logic tried to extend the interpretation of classical 2-valued connectives to the many-valued case.

In MUL a general semantic interpretation of an arbitrary logical connective C with an associated function f on a set $\{a\} \square M$ of an arbitrary truth value a, if C is 1-place, or on a set $\{(a,b)\} \subseteq M \times M$ of a pair of arbitrary truth values a,b, if C is 2-place, can be given. As the interpretation of a truth value always is *in respect to a universe*, the interpretation of a connective also is always in respect to a universe $U_i = (S_i, t_i)$.

We distinguish between the following cases:

1.) C is 1-place (therefore $f: M \to M$):
$a \in S_i$ or $a \notin S_i$
$f(a) \in S_i$ or $f(a) \notin S_i$
(which leads to 4 possible cases).

2.) C is 2-place (therefore $f: M \times M \to M$):
$a \in S_i$ or $a \notin S_i$
$b \in S_i$ or $b \notin S_i$
$f(a, b) \in S_i$ or $f(a,b) \notin S_i$
(which leads to 8 possible cases).

We said that we will interpret C with its associated function f on a domain consisting of one truth value (if C is 1-place) or of one pair of truth values (if C is 2-place): But because the truth value or the pair of truth values is arbitrary, we will get an interpretation of C on the total domain M (if C is 1-place) or M x M (if C is 2-place).

We have not the place here to interpret each case, but to give an impression we will give two short examples:

Example of 1-place connectives:
case $a \in S_i$, $f(a) \in S_i$

Because of $a \in S_i$ and $f(a) \in S_i$, C may be interpreted as *classical 2-valued connective* on {a} in respect to U_i.

An example is $T_1 := 1$; $F_1 := 2$; $T_2 := 2$; $F_2 := 3$; and $f(T_1) = F_1$;

in this case C may be interpreted as classical 2-valued negation (associated function: f(T) = F, f(F) = T) or contradiction (associated function: f(T) = F, f(F) = F) on $\{1\} = \{T_1\}$ in respect to U_1. If additionally

$$f(F_1) = T_1,$$

C may (only) be interpreted as classical 2-valued negation on {1,2} in respect to U_1 (which shall say that C *may* be interpreted as negation or contradiction on {1} and as negation or tautology – associated function : f(T) = T, f(F) = T – on {2}, but if you take both truth values together, C may *only* be interpreted as classical 2-valued negation on {1,2} in respect to U_1).

Another example: $T_1 := 1$; $F_1 := 0$; $T_2 := 0$; $F_2 := 1$; and $f(1) = 0$; $f(0) = 0$;

Here C may (only) be interpreted as classical 2-valued contradiction on {0, 1} in respect to U_1 and at the same time as classical 2-valued tautology on {0, 1} in respect to U_2.

This means: In MUL one and the same connective may have different interpretations in respect to different universes and the interpretation of connectives does not only depend on their associated value functions but also on the chosen set L of logical universes (the interpretation of the other three cases of 1-place connectives on a set $\{a\} \subseteq M$ is in terms of relevance: In these cases relevance is – so to say – "shifted" from one universe to another). Mark that in general a 1-place connective will be interpreted according to all four cases at the same time (but in respect to different universes).

Remark: From the foregoing explanations it follows that MUL is a *logic of many nega-*

tions (if card(M) ≥ 3) in the following two senses:

(i) Different connectives may be both interpreted as classical 2-valued negation in respect to the same universe $U_i = (S_i, t_i)$ on S_i:

E.g.: $T_1 := 1$; $F_1 := 2$; $T_2 := 2$; $F_2 := 3$; and

$$f_C(1) = 2;\ f_C(2) = 1;\ f_C(3) = 3;$$
$$f_D(1) = 2;\ f_D(2) = 1;\ f_D(3) = 1;$$

(f_C and f_D being the associated functions of connectives C and D).

In this example C and D may both be interpreted as classical 2-valued negation on {1, 2} in respect to U_1 though they are different ($f_C(3) \neq f_D(3)$).

(ii) For every universe $U_i = (S_i, t_i)$ there are connectives that may be interpreted as classical 2-valued negation on S_i in respect to U_i, but must not be interpreted as classical 2-valued negation on an arbitrary subset of M with one element in respect to any universe $U_k = (S_k, t_k)$ with $S_i \neq S_k$:

E.g.: Let $L = \{U_1 = (\{1, 2\}, t_1), ...\}$ be a set of logical universes and

$$f(1) = 2;\ f(2) = 1;\ f(m) = m$$
$$(\text{for all } m \in M\backslash\{1, 2\}).$$

The property we claimed holds because this connective may be interpreted as classical 2-valued negation on {1, 2} in respect to U_1, but must not be interpreted in that way on an arbitrary subset of M with one element in respect to all universes $U_k = (S_k, t_k)$ with $S_i \neq S_k$.

Summing up: Every universe U_i “induces” connectives that may be interpreted as classical 2-valued negation on S_i in respect to U_i. Maybe the most elementary of those are of the following form: If we say *elementary negation of universe U_i* (let $S_i = \{a, b\}$), we mean a 1-place connective N_i with associated function f: $M \to M$ such that f(a) = b, f(b) = a and f(x) = x for all $x \in M\backslash\{a, b\}$.

Now we will turn to an example of 2-place connectives.

Example of 2-place connectives:
case $a \in S_i$, $b \in S_i$, $f(a, b) \in S_i$

Because of $a, b \in S_i$ and $f(a, b) \in S_i$ the connective C may be interpreted as *classical 2-valued connective* on {(a, b)} in respect to U_i.

Let us examine this case by another program example (at first formulated classically 2-valued):

Example 3

```
1. PROCEDURE Program1;
2. IF (P(a) AND Q(b))=TRUE THEN
3.   Q(b):=FALSE;
4. ELSIF (P(a) AND Q(b))=FALSE
      THEN
5.   Q(b):=TRUE;
6. END;

1. PROCEDURE Program2;
2. IF (P(a) AND Q(b))=TRUE THEN
3.   P(a):=TRUE;
4. ELSIF (P(a) AND Q(b))=FALSE
      THEN
5.   P(a):=FALSE;
6. END;
```

Let e.g. the initial value be *P(a) := TRUE*; *Q(b) := TRUE* (this time a initial value has to be assigned to P(a) *and* Q(b), because the result of “(P(a) AND Q(b))” is of course determined by the truth values of P(a) and Q(b)).

The output of example 3 is this:

Output of Example 3

in integer notion:

	P(a)	Q(b)
Initial value:	1	1
(After calling) PROGRAM 1:	1	0
(After calling) PROGRAM 2:	0	0
(After calling) PROGRAM 1:	0	1
(After calling) PROGRAM 2:	0	1
(After calling) PROGRAM 1:	0	1
(After calling) PROGRAM 2:	. . . .	

If we want to translate example 3 into the language of MUL (as we have done in the previous examples), we have to ask ourselves which function(s) correspond to the function associated with the classical 2-valued "AND" - connective (conjunction): Naturally a connective may be interpreted as "AND" on a set {(a, b)} of a pair of truth values, if its associated function f acts as the function associated with the classical 2-valued "AND", and a connective may be interpreted as "AND" on S_i x S_i (S_i = {T_i, F_i}), if its associated function f acts as the function associated with the classical 2-valued "AND" if restricted to the domain S_i x S_i, i.e.:

$$f(T_i, T_i) = T_i\ ;\ f(T_i, F_i) = F_i\ ;$$
$$f(F_i, T_i) = F_i\ ;\ f(F_i, F_i) = F_i\ ;$$

If e.g. $T_1 := 1$; $F_1 := 2$; $T_2 := 2$; $F_2 := 3$ then e.g. $f_1(x, y) = max(x, y)$ may be interpreted as function associated with "AND" on {1, 2} x {1, 2} in respect to U_1 and on {2, 3} x {2, 3} in respect to U_2. But of course this also holds for other functions, like e.g.

$f_2(x, y)$:	*1*	*2*	*3*
1	1	2	1
2	2	2	3
3	1	3	3

Another example:

$U_1 = (\{1, 2\}, t_1)$,
with $t_1(1) = T$, $t_1(2) = F$ and $f(2,2) = 2$;

Here f may be interpreted as a function associated with classical 2-valued "AND", "OR" (disjunction), etc. (and 6 further connectives) on {(2, 2)} in respect to U_1. If additionally

$$f(1, 1) = 1; f(1, 2) = 1; f(2, 1) = 1;$$

f may only be interpreted as the function associated with classical 2-valued "OR" on {1, 2} x {1, 2} in respect to U_i.

Now let us rewrite example 3 using f(x, y) = max(x, y) as function associated with the connective C by which we will replace the classical 2-valued "AND" in program 1 and 2 of example 3 (with: $T_1 := 1$; $F_1 := 2$; $T_2 := 2$; $F_2 := 3$; and initial values $P(a) := 1$; $Q(b) := 2$):

Example 3'

```
1. PROCEDURE Program1;
2. IF f(P(a), Q(b))= T1 THEN
3.   Q(b):= F1;
4. ELSIF f(P(a), Q(b))= F1 THEN
5.   Q(b):= T1;
6. END;

1. PROCEDURE Program2;
2. IF f(P(a), Q(b))= T2 THEN
3.   P(a):= T2;
4. ELSIF f(P(a), Q(b))= F2 THEN
5.   P(a):= F2;
6. END;
```

This results in:

Output of Example 3'

in integer notation:

	P(a)	Q(b)
Initial value:	1	2
(After calling) PROGRAM 1:	1	1
(After calling) PROGRAM 2:	1	1
(After calling) PROGRAM 1:	1	2
(After calling) PROGRAM 2:	2	2
(After calling) PROGRAM 1:	2	1
(After calling) PROGRAM 2:	2	1
(After calling) PROGRAM 1:	2	1
(After calling) PROGRAM 2:	. . . .	. . .

interpretation by program 1:

	P(a)	Q(b)
Initial value:	T_1	F_1
(After calling) PROGRAM 1:	T_1	T_1
(After calling) PROGRAM 2:	T_1	T_1
(After calling) PROGRAM 1:	T_1	F_1
(After calling) PROGRAM 2:	F_1	F_1
(After calling) PROGRAM 1:	F_1	T_1
(After calling) PROGRAM 2:	F_1	T_1
(After calling) PROGRAM 1:	F_1	T_1
(After calling) PROGRAM 2:	. . . .	. . .

interpretation by program 2:

	P(a)	Q(b)
Initial value:	#	T_2
(After calling) PROGRAM 1:	#	#
(After calling) PROGRAM 2:	#	#
(After calling) PROGRAM 1:	#	T_2
(After calling) PROGRAM 2:	T_2	T_2
(After calling) PROGRAM 1:	T_2	#
(After calling) PROGRAM 2:	T_2	#
(After calling) PROGRAM 1:	T_2	#
(After calling) PROGRAM 2:	. . . .	. . .

A fixpoint is reached with P(a) = 2 and Q(b) = 1.

Note that similar properties hold as in the case of 1-place connectives regarding different interpretation of one and the same connective in respect to different universes (e.g. $T_1 := 1$; $F_1 := 0$; $T_2 := 0$; $F_2 := 1$; and $f(x, y) = min(x, y)$: in this example C may be interpreted as classical 2-valued "AND" on {0, 1} x {0, 1} in respect to U_1, but at the same time as classical 2-valued "OR" on {0, 1} x {0, 1} in respect to U_2; this underlines why we said that universes of the relation type 2 are kind of "inverse" !) and also regarding the interpretation of connectives that depend on their associated functions *and* on the chosen set L of logical universes. E.g. if $T_1 := 1$; $F_1 := 2$; $T_2 := 3$; $F_2 := 2$; and $f(x, y) = max(x, y)$ would accomplish no adequate translation of example 3 into MUL, because C may only be interpreted as classical 2-valued "OR" on {2, 3} x {2, 3} in respect to U_2, but not as "AND"; one adequate possibility for f would be:

f(x, y):	*1*	*2*	*3*
1	1	2	1
2	2	2	2
3	1	2	3

Also note that each program P_i will still behave classically 2-valued if restricted to $M = S_i$ (or if you choose universes of relation type 1) and communication being cancelled. The other 7 cases of the interpretation of a 2-place connective on a set {(a, b)} have again no classical analogue (but new interesting properties regarding relevance that we have not the place to elaborate more intensively).

For the semantics that we will give afterwards the following classic result is of importance (this result you can e.g. find in Zinovév[5]; the proposition itself goes back to Webb):

Proposition 1 (Webb):
Let M = {1, 2, ..., n}, F = {f, p} with $f: M \times M \rightarrow M$ and $p: M \rightarrow M$, such that f(x, y) = max(x, y) and

p(x):	1	2	...	n-1	n .
	2	3	...	n	1

Then F is functionally complete (i.e. every function associated with an arbitrary connective can be composed of f and p).

Now, this f has the nice property that for every universe $U_i \in L$ (= set of logical universes; $i \in I$ with I a set of indices) with $U_i = (S_i, t_i)$ and $S_i \subseteq \mathrm{I\!N}_0$ f is the associated function of classical 2-valued "OR" (if $t_i(\max(S_i)) = T$) or "AND" (if $t_i(\max(S_i)) = F$) on S_i in respect to U_i. If additionally the graph G = (V, E) such that V = M and $E = \{\{a, b\} \mid \exists\, i \in I: S_i = \{a, b\}\}$, is connected, it is easy to show that p can be composed of the set $Tr = \{\tau_i \mid \tau_i = (T_i\ F_i)$ and $i \in I\}$ (I is again the set of indices: see the definition of L; by $(T_i\ F_i)$ we mean the transposition $\tau_i(T_i) = F_i$, $\tau_i(F_i) = T_i$) and $\tau_i(x) = x$ for all $x \in M \backslash S_i$), since: if G is connected, it contains a spanning tree; see e.g. Berge[19], p. 141 for the proposition that a set Tr of n-1 transpositions generates the symmetric group S_n, if and only if (M, Tr) is a tree with Tr viewed as set of edges; and since $p \in S_n$, we are done. The τ_i have the nice property that they are exactly the elementary negations of universes U_i, which we have defined previously: In this sense the set $F = \{f = \max(x, y)\} \cup Tr$ is not only functionally complete, if G is connected, and consists of functions with a nice interpretation, but can also be understood as kind of an extension of the classical 2-valued case, where for M = {0, 1} and F = {associated functions of $\neg, \wedge, \vee$} also functional completeness can be proved.

This is the reason why we will base the semantics of MUL (in the next chapter) on this set $F = \{f = \max(x, y)\} \cup Tr$ of functions (by the way: we have now another reason for the decision to restrict ourselves to 1- and 2-place connectives, because – as we have seen – every connective can be defined by connectives with associated functions in F and the functions in F are just 1- or 2-place !)

We are now in the position to define semantics for MUL.

3. Semantics

(i) Let *L* be the *set of logical universes*, i.e. $L = \{U_i \mid U_i = (S_i, t_i)$ *and* $i \in I\}$ ($L \neq \emptyset$), where I is a set of indices, for all $i \in I$ U_i is a logical universe and for all $i \in I$: $S_i \subseteq \mathrm{I\!N}_0$.

(ii) Let *V* be the *set of statement variables*, i.e. $V = \{p, q, r, p_1, p_2, p_3, \ldots\}$.

(iii) Let *A* be the *alphabet*, i.e. $A = V \cup \{C\} \cup \{N_i \mid i \in I\} \cup \{(,)\}$.

(iv) Let *S* be the *set of statements* (we will not speak of "statement forms"), i.e. S is defined inductively in the following way:

(a) $\forall v \in V: v \in S$
(b) $\forall s \in S: (N_i\ s) \in S$
(c) $\forall s, t \in S: (s\ C\ t) \in S$

(v) Let *M* be the *set of truth values*, i.e. $M = \bigcup S_i$; $i \in I$

(vi) Let *D* be the *set of designated truth values*, i.e. $D = \{a \in M \mid \exists\, i \in I: t_i^{-1}(T) = a\}$ (as you can see the notion of a "designated truth value" in MUL is drawn back to the one designated truth value in classical 2-valued logic, namely True).

(vii) Let *e* be the *evaluation function*, i.e. *e:* $S \rightarrow M$, *with:*
(a) $\forall s \in S: \forall i \in I: e((N_i\ s)) = (a_i\ b_i) \cdot e(s)$, *where* $S_i = \{a_i, b_i\}$; (a_i, b_i) *is the transposition of* a_i *and* b_i; "$\cdot$" *shall mean: to apply the transposition* $(a_i\ b_i)$ *to the value of e(s).*
(b) $\forall s, t \in S: e((s\ C\ t)) = \max(e(s), e(t))$

(viii) Let $s \in S$.

Def. 3: s is called *tautology*, if for all evaluation functions e holds: $e(s) \in D$.

Regarding the axiomatization of MUL the following proposition is decisive.

4. Axiomatization

Proposition 2: For the semantics of MUL ((i) – (viii)) with M = {1,..., n} and D = {1, ..., d} (d < n) there is a sound and complete axiomatization concerning tautologies, if the graph G = (V, E) such that V = M and E = {{a, b} □□ i □ I: S_i = {a, b}} is connected.

The axioms are the following:

(i) Choose some two-place connective and some one-place connectives J_k (1 □ k □ n), which are *plausible* or satisfy *standard conditions* (for the definition of these terms see Rosser[3]).

(ii) A "chain symbol" □ is defined recursively:

(a) If v < u, then $\underset{i=u}{\overset{v}{□}} P_i\ Q := Q$.

(b) If v ≥ u, then $\underset{i=u}{\overset{v}{□}} P_i\ Q := P_v \quad \underset{i=u}{\overset{v-1}{□}} P_i\ Q$.

(iii) Then for all statements P, Q, R, and P_1, P_2, the following statements are theorems:

A 1. Q (P Q).
A 2. (P (Q R)) (Q (P R)).
A 3. (P Q) ((Q R) (P R)).
A 4. $(J_k(P) \quad (J_k(P) \quad Q)) \quad (J_k(P) \quad Q)$ with 1 □ k □ n.
A 5. $\underset{i=1}{\overset{n}{□}} (J_i(P) \quad Q)Q$.
A 6. $J_i(P) \quad P$ with i = 1, ..., d.
A 7. $J_{p2}(P_2) \quad (J_{p1}(P_1) \quad J_{max(p1,\, p2)}((P_1\ C\ P_2)))$ with p1, p2 □ M, and:
$J_{p1}(P_1) \quad J_{(a\ \ b)\ (p1)}((N_i\ P_1))$
with p1 □ M for all U_i □ L: S_i = {a, b}.

R1. If P is a theorem and P Q is a theorem, then Q is a theorem.

(mark that the condition that G is connected is just needed to get functional completeness from the connectives C and N_i with i □ I.)

Proof: For the proof see Rosser[3]. The connectives used by Rosser[3] can be composed by C and N_i (i □ I) because of functional completeness; the definition of tautology (or, in their terminology: acceptable statement) used by Rosser[3] corresponds to our definition of tautology.

5. Discussion

To study a logic like MUL has got different reasons: One is to get new views on classical 2-valued logic. E.g. may some of the most important tautologies of classical 2-valued logic like the law of double negation or the de Morgan rules be shown to have interesting extensions in MUL (which lead to permutation groups in general and automorphism groups in particular).

Another way to study MUL is to give a philosophical interpretation of its formalism: This may lead to an ontology of "many realities" (each human being confronted with her/his reality), each reality inducing a classical 2-valued logic (a logical universe). For more on this see Günther[6] and Mitterauer[10]. In the opinion of Günther everyone that makes statements uses classical 2-valued logic. But: According to Günther it is an *unfounded assumption* that everyone uses the *same* classical 2-valued logic ! It is only known that they use *some* classical 2-valued logic, but the classical 2-valued logics *different* persons (machines) may use, *need not coincide* (in the sense that their truth values may differ) !

If you accept this idea as a working assumption, it leads immediately to a logic, where a parameter has been added to classical 2-valued logic by which the latter is being multiplied. The usage of this parameter should be to *define in a strict and formal way the bearer of logic* "*into*" *logic itself* (a parameter of "point of view", if you want) ! MUL is additionally characterized by the facts that each statement has a unique truth value, which may be interpreted as classical truth value in respect to at least one universe, and each universe is definitely 2-valued.

Another question is: To what area(s) (outside mathematical logic) could MUL be applied ?

In general, the area in which we would expect the greatest chance of useful application of MUL is that of communication (human-to-human, human-to-computer, computer-to-computer): Communication is the area in which statements are made by different humans (or computers) based on different realities with different meanings (and we feel that most communication problems stem from these differences in reality and meaning). If MUL could play a role in communication, the decisive question of communication would be: To which universe (out of a set of logical universes) should *I* assign *your* statement ? By formalizing and answering such questions MUL could be a helpful tool in the area of distributed artificial intelligence (as an example the Müller-Lyer visual illusion can be shown to have a – though simple – model in MUL: two computer agents "discussing" the illusion from different point of views; the programs of the agents written in the language of MUL).

Last but not least we want to thank the following people for their assistance without which this paper would not exist: Prof. Mitterauer; Prof. Czermak; Prof. Werner; Prof. Reitboeck; Dr. Wolf.

This paper was supported by the Medizinische Forschungsgesellschaft Salzburg.

References

[1] Lukasiewicz, J. (1920): "O logice trojwartosciowej". In: *Ruch Filosoficzny* 5, pp. 170–171.

[2] Post, E.L. (1921): "Introduction to a general theory of elementary propositions". In: *American Journal of Mathematics* 43, pp. 163–185.

[3] Rosser, J.B./Turquette, A.R. (1977): *Many-Valued Logics*. Greenwood Press.

[4] Zinovév, A.A. (Sinowjew) (1968): *Über mehrwertige Logik*. Deutscher Verlag der Wissenschaften. Berlin.

[5] Zinovév, A.A. (1963): *Philosophical Problems of Many-Valued Logic*. D. Reidel. Dordrecht-Holland.

[6] Günther, G. (1980): *Beiträge zur Grundlegung einer operationsfähigen Dialektik*. Vol. I–III. Felix Meiner Verlag. Hamburg.

[7] Kaehr, R. (1981): "Materialien zur Formalisierung der dialektischen Logik und der Morphogrammatik". In: Günther, G.: *Idee und Grundriß einer nicht-Aristotelischen Logik*. Felix Meiner Verlag. Hamburg.

[8] Kaehr,R./Goldammer von, E. (1988): "...Again Computers and the Brain". In: *Journ. Molecular Electronics* 4, pp. 31–37.

[9] Mitterauer, B. (1994): "Biokybernetik der Depression". In: *Der informierte Arzt*. Sonderpublikation.

[10] Mitterauer, B. (1980): "Die Logik des Wahns". In: *Confinia Psychiatrica* 23. pp. 173–186.

[11] Thomas, G.G. (1982): "On Permutographs". In: *Supplemento ai Rendiconti*

del Circolo Matematico di Palermo. Serie II/2.

[12] Ditterich, J. (1982): “Logikwechsel und Theorie selbstreferentieller Systeme”. In: D. Hombach (ed.): *Zukunft der Gegenwart*. Zeta 01. Rotation. Berlin, pp. 120–155.

[13] Na, H.S.H./Foerster V.H./Günther, G. (1964): *On Structural Analysis of Many-Valued Logic*. University of Illinois. Illinois.

[14] Campbell, J.W.: A letter to Günther.

[15] Kleene, S.C. (1952): *Introduction to Metamathematics*. Amsterdam, Groningen, New York.

[16] Lukasiewicz, J. (1930): “Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls”. In: *CR Varsovie*, cl. III, 23 (1930), pp. 51–77.

[17] Bocvar, D.A. (1938): “Ob odnom trechznacnom iscislenii i ego primenenii k analizu paradoksov klassiceskogo iscislenija”. In: *Mathematiceskij Sbornik* 4, pp. 287–308.

[18] Gupta, A./Belnap, N. (1993): *The Revision Theory of Truth*. The MIT Press. Cambridge.

[19] Berge, C. (1971): *Principles of Combinatorics*. Academic Press. New York, San Francisco, London.