Autodescriptivity: beware!

Non-classical logics, and in particular many-valued logics, are increasingly used in the study of formal aspects of computing. For example, a recent paper by P. F. Gibbins in this *Journal* presents a 3-valued propositional logic for VDM. In the use of such logics one naturally relies on earlier work done by logicians, a case in point being Gibbins's use of the concept of autodescriptivity, introduced by N. Rescher. The purpose of this Note is to sound a warning that Rescher's exposition of autodescriptivity is seriously flawed, and to clarify the autodescriptivity of the logic of VDM.

1. Introduction

The propositional logic of VDM is the 3-valued Kleene *logic of partial functions* (LPF), which the truth tables are:

			&			∨			⇒		
p	$\sim p$		T	⊥	F	T	⊥	F	T	⊥	F
T	F	T	T	⊥	F	T	T	T	T	⊥	F
⊥	⊥	⊥	⊥	⊥	F	T	⊥	⊥	T	⊥	⊥
F	T	F	F	F	F	T	⊥	F	T	T	T

Since this logic has no tautologies the notion of valid inference cannot as in classical propositional logic be captured by truth tables. For this reason Gibbins[1] presented in this *Journal* a sequent calculus formalisation of logical consequence in LPF. In passing, Gibbins remarks that LPF is *autodescriptive*, and cites as reference Rescher's book[2] on many-valued logics (still the *de facto* standard despite being out of print). Rescher's notion of autodescriptivity is meant to capture in terms of conditions on truth tables the idea that a logic can serve as its own metalogic. It is unclear whether or not Gibbins cites Rescher by way of justification, but either way it is appropriate to sound a warning note: *Rescher's exposition of autodescriptivity is seriously flawed.*

2. Autodescriptivity

To define autodescriptivity, notice first that the truth table of a (say) binary infix connective $*$ can be described by sentences of the form

2.1. If the truth value of p is i and the truth value of q is j, then the truth value of

$$p * q \text{ is } k$$

(where p and q are propositional variables, and i, j and k are truth values). The user now defines a new binary prefix operator **V**, with its own truth table, thinking of '**V**ip' as meaning 'The truth value of p is i'. Taking the 'and' and 'if...then...' of 2.1 to be the connectives '&' and '⇒' of the logic under consideration, we get an object-language formula

2.2.
$$\mathbf{V}ip \;\&\; \mathbf{V}jq \Rightarrow \mathbf{V}k(p * q)$$

for each entry in the truth table of each binary connective $*$ of the logic. The value k is constrained to be the actual entry in the (i,j)th position of the truth table for $*$. (Unary and even n-ary connectives can evidently be treated likewise.) Each such formula 2.2 has a truth table, and may or may not be a tautology.

Rescher defines a many-valued logic to be *autodescriptive with respect to* **V** iff 2.2 is a tautology for each and every entry of the truth table of each and every connective of that logic.

Two questions arise. First: does a sentence of the form 'the truth value of p is i' only take on the classical value T (*true*) and F (*false*), or can it also take on non-classical values? If the latter, the truth table of **V** can potentially contain all truth values of the logic, and in this case Rescher says the assignment operator **V** is *diversified*. Second: in calling a logic autodescriptive when the truth tables of its connectives yield only tautologies (via 2.2), do we or do we not count the assignment operator **V** among the connectives of the logic? If we do, and like the others it yields only tautologies, Rescher calls the logic *strongly* autodescriptive.

3. A logical error

It is a common logical error to confuse '*A* if *B*' with '*A* only if *B*'. Not even logicians, it seems, are immune to this: Rescher makes exactly this mistake in his exposition of autodescriptivity. On p. 85 of his book there is a paragraph intended to establish that a logic must be autodescriptive when three given conditions are met. Namely (using '/·/' to indicate truth values of formulae): (1) /**V**ip/ is designated if and only if $i = /p/$, (2) implication has the *modus ponens* feature, and (3) conjunction has the feature that if p & q assumes a designated value then both p and q must do so. In this paragraph Rescher says:

> For then (a) above could fail to assume a designated truth value only when its antecedent is designated and its consequent undesignated. But the former can now happen only when both **V**ip and **V**jq take designated values, i.e. iff both $i = /p/$ and $j = /q/$.

The first point to note about this is that the '(a)' in this quotation is a misprint. The context indicates that it is not the sentence (a) at the top of p. 85 that is meant, but the formula (1) in the middle of p. 84, which appears here (slightly adapted) as the formula 2.2. This being so, the important point to note next is that Rescher's second condition is not sufficient to substantiate the claim that formula 2.2 'could fail to assume a designated value only when its antecedent is designated and its consequent undesignated'. For the second condition simply says that implication must have the *modus ponens* feature, and on p. 84 this is defined as follows: '...if p and $p \Rightarrow q$ take designated values then so does q'. This means that if p is designated and q is undesignated then $p \Rightarrow q$ is undesignated. But it does not mean that $p \Rightarrow q$ is undesignated *only* when p is designated and q is undesignated. Rescher has simply blundered.

(Note: That $p \Rightarrow q$ is undesignated *only* when p is designated and q is undesignated would be a very strong condition to impose on an implication connective in a many-valued logic, particularly when there is only one designated value. For in this case the truth table for ⇒ has undesignated values only in (part of) one row, namely the row of the designated value. All other entries in the truth table are then the designated value.)

In fact the same logical error occurs a couple of times. The reader diligent enough to check Rescher's book will see, for example, that his third condition on p. 85, which concerns conjunction, does not match the conjunction condition on p. 84, which has the implication reversed. And the claim (a) at the top of p. 85 is false, because it can only make use of the conjunction condition on p. 84, which, for this purpose, would have the implication going the wrong way.

4. Autodescriptivity of LPF

Rescher claims that 'as a consequence of this result' (i.e. the supposed sufficiency of the three given conditions) the 3-valued Kleene logic LPF is autodescriptive with respect to the **V**-operator given by the table

		p		
	V	T	⊥	F
	T	T	⊥	F
i	⊥	⊥	T	⊥
	F	F	⊥	T

This claim is false. As a counterexample, note that if we take in 2.2 $*$ to be &, and i and j to be F and ⊥ respectively, so as to get

4.1.
$$\mathbf{V}Fp \;\&\; \mathbf{V}\perp q \Rightarrow \mathbf{V}F(p \;\&\; q)$$

then this formula has the undesignated value ⊥ when p and q are both assigned value ⊥. (Note: Rescher also claims that the 3-valued Łukasiewicz logic is autodescriptive with respect to this same **V**-operator. This claim, too, is false: take $*$ to be ⇒, i and j to be T and F, respectively, and p and q both to have value ⊥.)

Where does this leave Gibbins's claim that LPF is autodescriptive? To clear the matter up we establish two results.

Result 1. *LPF is strongly autodescriptive with respect to the 2-valued* **V**-*operator defined by the table*:

		p		
	V	T	⊥	F
	T	T	F	F
i	⊥	F	T	F
	F	F	F	T

Proof. Since the **V**-operator is two-valued, it suffices to show that if **V**ip and **V**jq both have value T then:

(1) $\mathbf{V}(\sim i)(\sim p)$ has value T;

(2) for every binary infix connective $*$, $\mathbf{V}(i * j)(p * q)$ has value T, and

(3) $\mathbf{V}(\mathbf{V}ij)(\mathbf{V}pq)$ has value T.

Assume $\mathbf{V}ip$ and $\mathbf{V}jq$ both have value T. The table defining $\mathbf{V}$ shows that $\mathbf{V}ip$ has value T if and only if $/p/ = i$. But then $/\sim p/ = \sim i$; hence $\mathbf{V}(\sim i)(\sim p)$ has value T. For (2), since $/p/ = i$ and $/q/ = j$, it is easy to check that for every $*$ $/p*q/ = i*j$, which implies that $\mathbf{V}(p*q)(i*j)$ has value T. Finally, the truth values of p and q are i and j, respectively where either $i = j$ or $i \neq j$. In either case $\mathbf{V}ij$ and $\mathbf{V}pq$ have the same assignment (either T or F); so $\mathbf{V}(\mathbf{V}ij)(\mathbf{V}pq)$ has value T. ■

Note that the V-operator above satisfies the first of Rescher's constraints: it has $/\mathbf{V}ip/ = \mathrm{T}$ iff $/p/ = i$. It's simplicity derives from the fact that *in addition* it satisfies $/\mathbf{V}ip/ = \mathrm{F}$ iff $/p/ \neq i$. Discarding this latter condition, but still retaining Rescher's first constraint, we can prove:

Result 2. *LPF is not autodescriptive with respect to any diversified* **V***-operator.*

Proof. Let **V** be any diversified **V**-operator (satisfying $/\mathbf{V}ip/ = \mathrm{T}$ iff $/p/ = i$), then $\perp$ appears somewhere in the truth table for **V** – say $/\mathbf{V}ip/ = \perp$ for some value of p and some $i \in \{\mathrm{T}, \perp, \mathrm{F}\}$. For LPF to be autodescriptive $\mathbf{V}ip \Rightarrow \mathbf{V}(\sim i)(\sim p)$ must (for these values) have value T, hence $/\mathbf{V}(\sim i)(\sim p)/ = \mathrm{T}$. Then $/\mathbf{V}(\sim i)(\sim p)$ & $\mathbf{V}ip/ = \perp$, and so, since (taking $*$ to be $\Rightarrow$ in 2.2)

$$\mathbf{V}(\sim i)(\sim p) \ \& \ \mathbf{V}ip \Rightarrow \mathbf{V}(\sim i \Rightarrow i)(\sim p \Rightarrow p)$$

must have value T for LPF to be autodescriptive, we must have $/\mathbf{V}(\sim i \Rightarrow i)(\sim p \Rightarrow p)/ = \mathrm{T}$. Since $/\mathbf{V}ip/ = \perp \neq \mathrm{T}$ we know that $/p/ \neq i$, and simply checking all cases reveals that whenever $/p/ \neq i$ we get $/\sim p \Rightarrow p/ \neq \sim i \Rightarrow i$, hence $/\mathbf{V}(\sim i \Rightarrow i)(\sim p \Rightarrow p)/ \neq \mathrm{T}$. Hence LPF cannot be autodescriptive with respect to **V**. ■

This bears out, though for different reasons, Gibbins's claim that '... though LPF is a non-standard logic in which there are three truth values, its meta-theory is formulable in a standard two-valued meta-language'.

C. BRINK, I. M. REWITZKY and R. A. SCHMIDT

Department of Mathematics, University of Cape Town, Rondebosch 7700, South Africa.

References

1. P. F. Gibbins, VDM: Axiomatising its Propositional Logic, *The Computer Journal* **31** (6), 510–516 (1988).
2. N. Rescher, *Many-valued Logic*, McGraw-Hill (1969).

Announcement

29–31 OCTOBER 1991

AI✱IA Second Scientific Congress and Industrial Exhibition, Palermo, Italy, organised by **CRES** (Centro per la Ricerca Elettronica in Sicilia), **DIE** (Diapertimento de Ingegneria Elettrica dell'Universitá di Palermo) and **CITC** (Centro Interdipartimentale di Tecnologie della Conoscenza dell'Universitá di Palermo). AI✱IA (Italian Association for Artificial Intelligence) founded in 1988 with the intention of promoting the development of study and research in artificial intelligence and its applications. To this end, among a variety of activities, AI✱IA organises a national congress every other year. The second congress, open to international participation, will be held in Palermo and will focus on high-quality scientific and technical results as well as on innovative industrial applications. Special sessions on industrial experience are envisaged.

Scientific subjects

- Architectures, languages and environments
- Knowledge representation and automated reasoning
- Problem solving and planning
- Knowledge acquisition and automatic learning
- Cognitive models
- Natural language
- Perception and robotics
- Industrial applications of artificial intelligence

Local organisation

Lia Giangreco, Ina Paladino and Giusi Romano, CRES, Viale Regione Siciliana 49, 90046 Monreale, Palermo. Tel. +39-91-6406192/6406197/6404501. Fax +39-91-6406200.

Logistic arrangements

GEA Congressi S.r.l. Via C. Colombo 24, 90142 Palermo. Tel. +39-91-6373418. Fax +39-91-6371625. Telex 910070 ADELFI.

1–2 SEPTEMBER 1991

Sixth Eurographics Workshop on Graphics Hardware, Vienna, Austria

Aims/Scope

Graphics hardware designers today have lots of different options for system design as offered by ASIC design tools and powerful general-purpose modules. On the one hand, silicon compilers offer the short turn-around time of ASIC design and allow for an increased circuit complexity. On the other hand, general-purpose hardware, as available today, offers economically state-of-the-art technologies and can be used to obtain 'cheap' parallel systems.

The aim of the workshop is to bring together leading workers in the field of graphics hardware design, to present their work and to discuss alternative design topics.

The following topics will be included.

- Architectures and interaction
- Architectures and algorithmic aspects
- VLSI design strategy for graphics
- Innovative custom hardware
- Innovative use of general-purpose hardware
- Graphics for VLSI design

Papers and discussions will be published by Springer Verlag.

For further information please contact:
Professor A. Kaufman, Department of Computer Science, State University of New York, Stony Brook, NY 11794–4400, USA. Tel: +1 516 632 8441 or 8471. Fax: +1 516 632 8334. E-mail: ari@sbcs.sunysb.edu.
or
Dr M. Mehl, FhG-AGD, Wilhelminenstr. 7, D-6100 Darmstadt, Germany, Tel.: +49 6151 155 153. Fax: +49 6151 155 199. E-mail: mehl@agd.fhg.de.

30 SEPTEMBER TO 2 OCTOBER 1991

First International Conference of the Austrian Centre for Parallel Computation, Salzburg, Austria

The Austrian Centre for Parallel Computation (ACPC) is a cooperative research organisation founded in 1989 to promote research and education in the field of software for parallel computer systems.

The first international conference of the ACPC is dedicated to a broad variety of subject areas within parallel computation, including high-performance computing. It will take place at the University of Salzburg and is intended as a forum for both researchers and practitioners in the field.

Currently, sessions are planned on the following topics.

- Languages, tools and environments
- Numerical algorithms
- Symbolic computation
- Performance evaluation
- Applications
- Architectures.

Invited speakers include Ken Kennedy (Rice University), Ken Neves (Boeing Computing Services), Klaus Schulten (University of Illinois), Satish Tripathi (University of Maryland), Ulrich Trottenberg (University of Cologne), Steve Wallach (Convex, Dallas), and Paul Wang (Kent State University).

For further information please contact:
Professor P. Zinterhof, Research Institute for Software Technology, University of Salzburg, Hellbrunner Strasse 34, A-5020 Salzburg, Austria. Tel.: +43-(0)662 8044 6701. Fax: +43-(0)662 8044 5010.