A Note on (In)Compatibility Relations

SEBASTIAN SEQUOIAH-GRAYSON*

Formal Epistemology Project
University of Groningen

Abstract

Non–symmetric Incompatibility relations, or non–symmetric compatibility relations, are a standard method for introducing a split negation pair on a frame. Another standard method is to reject commutation for the frame. The first task is to examine the relationship between non–symmetric incompatibility relations and non–symmetric compatibility relations and commutation failure on frames. The second task is to look at the sort of points that may constitute such frames. Taking the points as information states, what type of information states may operate in non–symmetric incompatibility/compatibility environments? The proposal made here is that databases consisting of sub–propositional information states, or data–points, are a respectable place to start.

keywords: compatibility frames, perp frames, noncommutation, split negation, van Benthem, Dunn, Restall, Wansing.

1 Introduction

Non–symmetry on an incompatibility relation $\perp$, or on a compatibility relation C, is a standard method for introducing a split negation pair onto a frame. Another standard method is to reject commutation for the frame. One task here is to examine the relationship between non–symmetric incompatibility relations and non–symmetric compatibility relation and commutation failure on frames. The other task is to look at the sort of points that may constitute such frames.

In section 2 we introduce compatibility frames and a split negation pair defined in terms of them. In section 3 we introduce perp frames and the resulting defined split negation pair. In section 4 we introduce noncommuting frames and the correspondingly defined split negation pair. In section 5 we translate the ternary relation R of substructural frame semantics into its operational

*Postdoctoral Research Fellow, Department of Theoretical Philosophy, *University of Groningen* - The Netherlands. Senior Research Associate, IEG - Computing Laboratory, *University of Oxford*.

semantics counterpart. In section 6 we look at the connections between a split negations pair defined in terms of a noncommuting frame, and in terms of C and $\perp$. In section 7 we lay out the relationships between (non)commutation on R and (a)symmetry on C and $\perp$. In section 8 we examine cases of non–symmetric (in)compatibility. In section 9 we precisify the conceptual framework underpinning much of what has transpired in sections 2–8, and consider its limits with respect to various database types.

2 Compatibility Frames

A compatibility frame (Restall (2000)) $\mathbf{F_C}$ is a triple $\langle S, \sqsubseteq, C \rangle$. S is a set of information states $x, y, z \ldots$. $\sqsubseteq$ is a partial order of informational development/inclusion such that $x \sqsubseteq y$ is taken to mean that the information carried by y is a development of the information carried by x. C is a binary compatibility relation where xCy means that the information carried by x is compatible with the information carried by y. If xCy then there is no information not carried by x that is carried by y. C interacts with $\sqsubseteq$ in the following manner:

$$\text{If } x \sqsubseteq y \text{ and } yCz, \text{ then } xCz. \tag{1}$$

$$\text{If } x \sqsubseteq z \text{ and } yCz, \text{ then } yCx. \tag{2}$$

In what follows, we will often write $x, y, \ldots \in \mathbf{F_X}$ as shorthand for $x, y, \ldots \in S$ where $S \in \mathbf{F_X}$. Reading $x \Vdash A$ as "the information state x carries information of type A", in this case we can give the following evaluation conditions for our a split negation pair in terms of C as follows:

$$x \Vdash {\sim}A \textit{ iff } \text{for each } y \in \mathbf{F_C} \text{ s.t. } xCy,\ y \nVdash A. \tag{3}$$

$$x \Vdash \neg A \textit{ iff } \text{for each } y \in \mathbf{F_C} \text{ s.t. } yCx,\ y \nVdash A. \tag{4}$$

Importantly, our split negation pair is preserved only under the assumption that C is non–symmetric. That is, that:

$$xCy \not\Rightarrow yCx \tag{5}$$

Question 1: How plausible an assumption is (5)?

There is no correct answer to this until one has specified the type of information carried by the states $x, y, z \ldots \in S$, which is just to say until one has specified the type of information denoted by A. We have said nothing about this yet. We will have much to say about this later, but before then we need to lay out the details at work in *incompatibility frames.*

3 Perp Frames

A perp frame (or incompatibility frame, Dunn (1994), Dunn (1996)) $\mathbf{F}_\perp$ is a triple $\langle S, \sqsubseteq, \perp \rangle$ where S and $\sqsubseteq$ are as in the compatibility frame above. $\perp$ is a binary incompatibility relation such that $x \perp y$ means that the information carried by x is incompatible with the information carried by y. In this case, we may define a split negation pair as follows:

$$\sim A := \{X : A \perp X\} \tag{6}$$

$$\neg A := \{X : X \perp A\} \tag{7}$$

$\perp$ interacts with $\sqsubseteq$ in the following manner:

$$\text{If } x \sqsubseteq y \text{ and } z \perp x, \text{ then } z \perp y. \tag{8}$$

$$\text{If } x \sqsubseteq y \text{ and } x \perp z, \text{ then } y \perp z. \tag{9}$$

In this case we can give the following evaluation conditions for our a split negation pair in terms of $\perp$ as follows:

$$x \Vdash \sim A \textit{ iff } \text{for each } y \in \mathbf{F}_{\perp} \text{ s.t. } y \Vdash A,\ x \perp y. \tag{10}$$

$$x \Vdash \neg A \textit{ iff } \text{for each } y \in \mathbf{F}_{\perp} \text{ s.t. } y \Vdash A,\ y \perp x. \tag{11}$$

As with C, our split negation is preserved only under the assumption that $\perp$ is non–symmetric. That is, that:

$$x \perp y \not\Rightarrow y \perp x \tag{12}$$

Question 2: How plausible an assumption is (12)?

Question 3: What is the relationship between Question 1 and Question 2?

The answer to Question 2 turns on the very same issues as does the answer to the same question put to C, namely Question 1. This is in itself the answer to Question 3. Questions 1 and 2 turn on the same issues because C and $\perp$ are complements:

$$xCy \textit{ iff } \text{not}\, x \perp y \tag{13}$$

$$x \perp y \textit{ iff } \text{not}\, xCy \tag{14}$$

Their equivalent definitions of negation are most easily appreciated in the context of a split negation pair defined in terms of *noncommuting frames.*

4 Noncommuting Frames

A noncommuting frame $\mathbf{F_{nc}}$ is a triple $\langle S, \sqsubseteq, R\rangle$ where S and $\sqsubseteq$ are as above, and R is a noncomutting ternary relation on the frame such that:

$$Rxyz \not\Rightarrow Ryxz \tag{15}$$

Via the addition of a double implication pair $\langle \rightarrow, \leftarrow \rangle$ and bottom constant $\mathbf{0}$, we can define our split negation pair as follows:

$$\sim A := A \rightarrow \mathbf{0} \tag{16}$$

$$\neg A := \mathbf{0} \leftarrow A \tag{17}$$

The evaluation conditions for our connectives and constant are given by the following:

$$x \Vdash A \rightarrow B \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } Rxyz, \text{ if } y \Vdash A \text{ then } z \Vdash B. \tag{18}$$

$$x \Vdash B \leftarrow A \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } Ryxz, \text{ if } y \Vdash A \text{ then } z \Vdash B. \tag{19}$$

$$x \Vdash \mathbf{0} \text{ for no } x \in \mathbf{F}. \tag{20}$$

In this case, we can give the evaluation conditions for our split negation pair via the following:

$$x \Vdash \sim A[A \rightarrow \mathbf{0}] \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } Rxyz, \text{ if } y \Vdash A \text{ then } z \Vdash \mathbf{0}. \tag{21}$$

$$x \Vdash \neg A[\mathbf{0} \leftarrow A] \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } Ryxz, \text{ if } y \Vdash A \text{ then } z \Vdash \mathbf{0}. \tag{22}$$

In $\mathbf{F_{nc}}$ our split negation pair is only preserved on account of its noncommuting behaviour. Were commutation to be present, then we would loose our split negation.

Question 4: How plausible an assumption is (15)?

Question 5: What is the relationship between Questions 4, 1, and 2?

The answer to this Question 4 turns on the same issues as does the answer to Questions 1 and 2 above. However, the answer to Question 5 is not the same as the answer to Question 4. Although there is a relationship between (in)compatibility non–symmetry and commutation failure, these are not equivalent. Getting clear on all of this is most easily done after interpreting R in robustly informational terms via an operational semantics.

5 Operational Semantics and R

We can make the operation encoded by the ternary relation under an informational interpretation clearer via a rewriting in terms of an operational semantics. The point of an operational semantics is that it makes the semantic operations explicit.

We start by rewriting the ternary relation R in terms of $\sqsubseteq$ and a binary composition operator $\bullet$ that operates on members of S. $\bullet$ is the semantic counterpart to to fusion, or intensional\multiplicative conjunction, $\otimes$. Our rewrite comes out as the following:

$$Rxyz := x \bullet y \sqsubseteq z \tag{23}$$

We read our operationalised ternary relation as something like "the combination of the information carried by x with the information carried by y develops into the information carried by z". In this case, we may rewrite (15) as:

$$x \bullet y \not\Rightarrow y \bullet x \tag{24}$$

In other words, commutation failure corresponds to directional combination. We may also rewrite (18) and (19) as the following:

$$x \Vdash A \rightarrow B \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } x \bullet y \sqsubseteq z, \text{ if } y \Vdash A \text{ then } z \Vdash B. \tag{25}$$

$$x \Vdash B \leftarrow A \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } y \bullet x \sqsubseteq z, \text{ if } y \Vdash A \text{ then } z \Vdash B. \tag{26}$$

Hence (21) and (22) get rewritten as:

$$x \Vdash {\sim}A[A \rightarrow \mathbf{0}] \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } x \bullet y \sqsubseteq z, \text{ if } y \Vdash A \text{ then } z \Vdash \mathbf{0}. \tag{27}$$

$$x \Vdash \neg A[\mathbf{0} \leftarrow A] \textit{ iff } \text{for all } y, z \in \mathbf{F_{nc}} \text{ s.t. } y \bullet x \sqsubseteq z, \text{ if } y \Vdash A \text{ then } z \Vdash \mathbf{0}. \tag{28}$$

We will refer to the ternary relation under its operationalised definition for much of what follows, as it brings to the foreground the very compatibility and incompatibility properties that we are interested in. Now we can clarify the relationship between non–symmetry on C and $\perp$ and noncommutation on R, namely Question 5 from section 4 above.

6 C, $\perp$, and R

Where we have ${\sim}A := A \rightarrow \mathbf{0}$ and $\neg A := \mathbf{0} \leftarrow A$, C and $\perp$ are closely related to R. In this case, we have it that:

$$xCy \textit{ iff } \exists z(x \bullet y \sqsubseteq z) \tag{29}$$

$$yCx \textit{ iff } \exists z(y \bullet x \sqsubseteq z) \tag{30}$$

$$x \perp y \textit{ iff } -\exists z(x \bullet y \sqsubseteq z) \tag{31}$$

$$y \perp x \textit{ iff } -\exists z(y \bullet x \sqsubseteq z) \tag{32}$$

(29)–(32) make sense. (29) and (30) state that two information states are compatible *iff* their is a further information state resulting form their combination, whist (31) and (32) state that two information states are incompatible *iff* there is no information resulting from their combination. Given this, we can rewrite (3) and (4) as the following:

$$x \Vdash {\sim}A \textit{ iff } \text{for each } y \in \mathbf{F_C} \text{ s.t. } x \bullet y \sqsubseteq z,\ y \nVdash A. \tag{33}$$

$$x \Vdash \neg A \textit{ iff } \text{for each } y \in \mathbf{F_C} \text{ s.t. } y \bullet x \sqsubseteq z,\ y \nVdash A. \tag{34}$$

We can also rewrite (10) and (11) as:

$$x \Vdash {\sim}A \textit{ iff } \text{for each } y \in \mathbf{F}_{\perp} \text{ s.t. } y \Vdash A, -\exists z(x \bullet y \sqsubseteq z). \tag{35}$$

$$x \Vdash \neg A \textit{ iff } \text{for each } y \in \mathbf{F}_{\perp} \text{ s.t. } y \Vdash A, -\exists z(y \bullet x \sqsubseteq z). \tag{36}$$

(33) and (35) are equivalent. They tell us the same thing, that information states carrying information of type $\sim A$ can never be applied to an information state carrying information of type A, as such a process is "informationally redundant". Similarly, (34) and (36) are equivalent. They tell us the same thing, that information states carrying information of type $\neg A$ can never have an information state carrying information of type A applied to it, again due to procedural redundancy from the perspective of information generation.

These "procedural redundancy" interpretations are the very meanings of our split negation pair that are directly encoded by our definitions (16) and (17). The evaluation conditions given by (27) and (28) bear this out. Abstracting across directional distinctions, $A \rightarrow \mathbf{0}$ and $\mathbf{0} \leftarrow A$ are simply functions which, by definition, do not return outputs when given information of type A as an input. This is due to null–output $\mathbf{0}$ holding nowhere via (20). Before we do move on to accounting for this direction difference, we need to get a little clearer on the the constraints between commutation on R and symmetry on C and $\perp$.

7 Commutation on R and Symmetry on $C(\perp)$

If R is commutative then $C(\perp)$ is symmetric:

$$(x \bullet y \sqsubseteq x \rightarrow y \bullet x \sqsubseteq z) \Rightarrow (xCy \rightarrow yCx) \tag{37}$$

(37) hold because of the following (via (29) and (30)):

$$(x \bullet y \sqsubseteq x \rightarrow y \bullet x \sqsubseteq z) \Rightarrow \exists z(x \bullet y \sqsubseteq x) \rightarrow \exists z(y \bullet x \sqsubseteq z) \tag{38}$$

The constraint in (38) holds only in the left–to–right direction, as there is no constraint from symmetry on $C(\perp)$ to commutation on R. We can bring this out by revealing the suppressed universal quantifiers and scope markers in (38) in order to get the following:

$$\forall x \forall y(\forall z(x \bullet y \sqsubseteq x \rightarrow y \bullet x \sqsubseteq z)) \Rightarrow (\exists z(x \bullet y \sqsubseteq x) \rightarrow \exists z(y \bullet x \sqsubseteq z)) \tag{39}$$

This is just to say that we can have symmetry on $C(\perp)$ without full commutation on R, which is as we would expect. A neat example form Greg Restall (in conversation) bears this out: Define R on the integers by setting $Rxyz$ *iff* $x - y = z$. In this case, it will always be the case that $\exists zRxyz$, hence $\exists zRxyz \rightarrow \exists zRyxz$. However, we have $R101$ (since $1 - 1 = 0$), but not $R011$ (since $0 - 1 \neq 1$), hence we do not have it that $Rxyz \rightarrow Ryzx$ for all x. There will always be such an x, hence we have symmetry, although the relation will not always be true of it, hence we do not have full commutation.

Since there is no in general constraint from symmetry on $C(\perp)$ to full commutation on R, we know that there is no in general constraint from commutation failure on R to symmetry failure on $C(\perp)$. However, we do know that there is an in general constraint from symmetry failure on $C(\perp)$ to commutation failure on R (simply from (37) via logic).

8 Examples of non–symmetry on $C(\perp)$

Now we are in a healthy position to examine the questions put at the end of the first three sections. Q1 and Q2 are equivalent on account of compatibility and incompatibility being equivalent. Due to (37), we know that if we can give examples of symmetry failure on $C(\perp)$, we will have examples of commutation failure for free.

The thing is this: how to make sense of the non–symmetry of xCy (and hence of $x \perp y$). In Wansing (2001), Wansing states that "Obviously, [a split negation pair] coincide[s] if the not implausible assumption is made that incompatibility is a symmetric relation." With a very small amount of license, we will take this comment of Wansing's to be equivalent to the claim that non–symmetry on (in)compatibility *is* an implausible assumption.

The non–symmetry of the relation appears implausible because of the assumption that the information–types carried by the information states is tenseless, propositional information. Although this is standard procedure for elementary extensional logic, it is obviously not a constraint insofar as logical modeling is concerned.

The information carried by the information states may by of any type we stipulate, and some of these types will not support compatibility\incompatibility–symmetry. Still at the propositional level, consider tense–specific action–based propositions such as *Sebastian opens the fridge door* and *Sebastian retrieves a beer from the Fridge*. The compatibility of these informational tokens, and hence of the information states that carry them, is order–sensitive – I do not get my beer without first opening the fridge door.

However, in 2 we stated that $x \Vdash A$ should be read as "the information state x carries information of type A". Note we did not state "the information *that* A. That is, we have left the way open for processing on sub–propositional information types. The $C(\perp)$ asymmetries of sub–propositional information types from Categorical Grammar are suitable robust.

Directional information compatibility (or lack of it) is the mark of much of the informational behaviour of natural languages. Take an intransitive verb such as 'skips', and a noun such as 'Friederike'. 'Freiderike skips' is well–formed, whilst 'skips Friederike' is not. In this case, 'Friederike' is compatible with 'skips', however 'skips' is not compatible with 'Friederike'. The compatibility here is directional, insofar is information generation is concerned. Trivially, it is also the case that we get an non–symmetry of incompatibility here, as 'skips' is incompatible with 'Friederike, but 'Friederike' is not incompatible with 'skips'.

We can bring all of this together with operational semantics and multiple–

typing. That is, 'skips' is information of type $\neg\, n \ := \mathbf{0} \leftarrow n$, as well as of type $n \rightarrow s$. That is, if you give 'skips' to the right hand side of a noun, you get a sentence as the informational output, as in (41), but if you give it to the left hand side of a noun, you do not get any informational output, as marked by (40):

$$x \Vdash \neg\, n\, [\mathbf{0} \leftarrow n] \text{ \textit{iff} for all } y, z \text{ s.t. } y \bullet x \sqsubseteq z, \text{ if } y \Vdash n \text{ then } z \Vdash \mathbf{0}. \tag{40}$$

$$x \Vdash n \rightarrow s \text{ \textit{iff} for all } y, z \in \mathbf{F} \text{ s.t. } x \bullet y \sqsubseteq z, \text{if } y \Vdash n \text{ then } z \Vdash s. \tag{41}$$

For information compatibility\incompatibility that runs in the other direction, take as an example the adjective 'happy', and the noun 'Frederike' again. Here, 'happy Friederike' is well–formed, whilst 'Friederike happy' is not. In this case, 'Friederike' is incompatible with 'happy', but 'happy' is not incompatible with 'Friederike'. Equivalently, 'happy' is compatible with 'Friederike', but 'Friederike' is not compatible with 'happy'.

With operational semantics and multiple–typing again, we have 'happy' as type $\sim n \ := n \rightarrow \mathbf{0}$ and type $n \leftarrow n$. That is, if you give 'happy' to the left side of a noun, you get a complex noun phrase as the informational output, as in (43), and if you give 'happy' to the right hand side of a noun, you do not get any informational output, as in (42):

In explicit information processing terms, the types corresponding to 'happy' have the following frame conditions:

$$x \Vdash \sim n\, [n \rightarrow \mathbf{0}] \text{ \textit{iff} for all } y, z \text{ s.t. } x \bullet y \sqsubseteq z, \text{ if } y \Vdash n \text{ then } z \Vdash \mathbf{0}. \tag{42}$$

$$x \Vdash n \leftarrow n \text{ \textit{iff} for all } y, z \in \mathbf{F} \text{ s.t. } y \bullet x \sqsubseteq z, \text{if } y \Vdash n \text{ then } z \Vdash n. \tag{43}$$

Some of the most dramatic examples of $C(\perp)$ non–symmetry then, are the already well–known non–commuting properties of natural language.

9 Conclusion

Concrete examples of $C(\perp)$ non–symmetry are to be found in the dynamic behaviour of the sub–propositional information states underpinning natural language semantics. There is a philosophical aside that some might think threatens at this point, a purported worry about whether or not sub–propositional information really is information in any substantive sense. The concern here is that a necessary condition on information is that is be propositional. That is, are the information states in our frames above really information states, or merely data points?

There are two points to be made at this stage. The first is that metaphysical concerns such as this are in every way operationally irrelevant to the issue at hand. The second is that such concerns start to look merely terminological under a sufficient amount of light – whether you choose to call the the points data points or information states is largely a matter of taste (see Sequoiah-Grayson (2007)). The substantive issues concern the logical properties and behaviours of the models concerned, and these remain happily independent of baptisms.

Proceeding in terms of information states, the crucial moves for bringing non–symmetry on $C(\bot)$ into line with categorical grammar and natural language semantics have been to operationalise the ternary relation R in our various Kripke frames, parsing them in robust information–processing terms. By laying out the relationship between commutation failure on these frames and $C(\bot)$ non–symmetry, we then have a complete link to an information-processing parsing of (in)compatibility between information states. The example of sub–propositional information–processing can be pushed along a little further.

In our operationalised frames above, commutation–failure between particular information states is guaranteed precisely *because* these states are incompatible. We can think of the information states carrying the conditional information types as carrying functional information types, and of the information states carrying the antecedent information types as carrying input information types. In this case, compatibility between the relevant information states means that processing on these states guarantees output success. By contrast, incompatibility between the relevant information states means that processing on these states guarantees output failure. This is a good conceptual framework to use as a heuristic insofar as appreciating why it is that non–symmetry on C amounts to non–symmetry on $\bot$, and *vice versa*. Analysing logical formulas in terms of output success\failure has its proto–manifestation in Groenendijk and Stokhof (1991), and is further developed for various propositional and sub–propositional database types in Sequoiah-Grayson (2010a), Sequoiah-Grayson (2010b), and Sequoiah-Grayson (2009).

A natural language lexicon is just one database amongst many, both propositional and sub–propositional. Given the simplicity of the operational semantics, we should expect there to be a wide variety of database types to which such an analysis may be put. However, it is not obvious, indeed far from it, that *all* database types will be "cooperative". A group of agents in a communicative setting is a database if anything is. Dynamic Epistemic and Public Announcement logics (see Baltag, Moss, and Solecki (1998), Baltag and Smetts (2008), van Benthem (2009), and van Ditmarsh, van der Hoek, and Kooi (2008)) have been successful in capturing a great deal of the dynamic logical behaviour of such databases. In van Benthem (2010), van Benthem makes a strong case for there being *in principle* reasons to suspect that an operational analysis built on the back of categorical grammars breaks down when confronted with the subtleties at work in multi–agent scenarios such as these. At present this remains an open, and interesting problem.[1]

[1]My sincerest thanks to Greg Restall and Johan van Benthem for their good advice on many points. Many thanks also to the organisers and participants of Logica 2010. In particular, I would like to thank Marie Duzie, and Andreas Pietz, both both their comments and for their encouragement. An earlier version of this talk was presented at the ILLC in Amsterdam, and I would like to thank all of the participants, especially Dora Achourioti, Paul Dekker, and Catarina Dutilh, both for their insights and for their constructive criticism. I would also liek to take this opportunity to thank Michal Pelis and Vita Puncochar for their tireless efforts

References

Baltag, A., Moss, L., and Solecki, S. (1998): The logic of Public Announcements, Common Knowledge and Private Suspicions, *Proceedings of Tark'98*, 43–56, Morgan Kaufmann Publishers.

Baltag, A, and Smetts, S. (2008): The Logic of Conditional Doxastic Actions, forthcoming in R. van Rooij and K. Apt (rds.): *Texts in Logic and Games*, Special issue on New Perspectives on Games and Interaction, Amsterdam University Press.

van Benthem, J. (2009): Logical Dynamics of Information and Interaction, *manuscript*.

van Benthem, J. (2010): Categorical versus Modal Information Theory, *forthcoming* in *Linguistic Analysis*.

van Ditmarsh, H., van der Hoek, W., and Kooi, B, (2008): Dynamic Epistemic Logic, Springer, The Netherlands.

Dunn, J. M. (1994): Start and Perp: Two Treatments of Negation, in J. E. Tomberlin (ed.): *Philosophical Perspectives* vol. 7, pp. 331–357.

Dunn, J. M. (1996): Generalised Ortho Negation, in Heinrich Wansing (ed.): Negation: A Notion in Focus, pp. 3-26, Walter de Gruyter, Berlin.

Groenendijk, J., and Stokof, M. (1991): Dynamic Predicate Logic, *Linguistics and Philosophy*: 14, pp. 33–100.

Restall, G. (2000): Defining Double Negation Elimination, *Logic Journal of the IGPL*: 8.6, pp. 853–860.

Sequoiah-Grayson, S. (2010a): Epistemic Closure and Commuting, Nonassociating Residuated Strucrures, forthcoming in *Synthese*.

Sequoiah-Grayson, S. (2010b): Lambek Calculi with 0 and Test–Failure in DPL, *forthcoming* in *Linguistic Analysis*.

Sequoiah-Grayson, S. (2009): Dynamic Negation and Negative Information, *Review of Symbolic Logic*: 2.1, 233–248.

Sequoiah-Grayson, S. (2007): The Metaphilosophy of Information, *Minds and Machines*: 17.3, pp. 331-344.

Wansing, H. (2001): Negation, in L. Goble (ed.): *The Blackwell Guide to Philosophical Logic*, pp. 415–436.

editing the Yearbook itself.

Sebastian Sequoiah-Grayson
-Postdoctoral Research Fellow, Department of Theoretical Philosophy, *University of Groningen* - The Netherlands.
-Senior Research Associate, IEG - Computing Laboratory, *University of Oxford*.
Email: s.sequoiah-grayson@rug.nl
Projectpage: www.formalphilosophy.org
Homepage: http://logic.tsd.net.au