NONDETERMINISTIC AND NONCONCURRENT COMPUTATIONAL SEMANTICS FOR BB$^+$ AND RELATED LOGICS

SHAY ALLEN LOGAN

Abstract. In this paper, we provide a semantics for a range of positive substructural logics, including both logics with and logics without modal connectives. The semantics is novel insofar as it is meant to explicitly capture the computational flavor of these logics, and to do so in a way that builds in both nondeterministic and nonconcurrent computational processes.

1. Introduction

In their paper 'Combinators and Structurally Free Logic' ([6]) Dunn and Meyer give an interpretation of the ternary-relational semantics for relevant logics in computational terms. As they note in a footnote, '[t]here is an alternative "operational semantics" that [they] could be using'. In the same footnote, they briefly gesture in the direction one might go to find such a semantics. They then drop the matter, save for a brief reference in a footnote on the very next page.

As one would expect given the authors of the paper, the claims made are technically unimpeachable. But it turns out that there's some value to be had in actually spelling out the operational semantics in more detail, and the purpose of this paper is to demonstrate that this is so.

We should note that Dunn and Meyer examine a good deal more in their paper, including correspondences between (on one hand) propositional logics, their proof theory, and their their semantics and (on the other hand) combinators, combinatory logics, and typing systems. We will not have space in this paper to effect these extensions; that said, I think they are possible and plan to explore them in future work. Just as for Meyer and Dunn, so also here these extensions will turn the intuitive/philosophical analogy with computational systems into a more robust correspondence mediated by the lambda calculus (in its combinatory guise).

Putting aside future work for now, here's the plan of the current paper. In the following section, we work out the computational metaphor at the heart of this paper and compare it to the one gestured to by Dunn and Meyer in theirs. Intuitive motivation in hand, we turn in §3 to producing the promised semantics. In §4 we prove soundness of **BB**$^+$ for this semantics; §5 is dedicated to completeness of the same. In §6 and §7, we turn to exploring extensions—first to include structural rules then to include modalities. The final section concludes.

2. The Computational Interpretation

Dunn and Meyer's semantics—like the semantics we give in this paper—is a loosely Kripke-style semantics. By this we don't mean anything particularly fancy—just that models consist of a set of points at which formulas are evaluated together with machinery connecting up and operating on the points. In the case of Dunn and Meyer, a crucial piece of the machinery that decorates states is a ternary relation, R, which they frequently interpret as an indexed set of binary relations, R_a in the obvious way; viz. $R_abc =_{df} Rabc$. They then point out that this affords a type of flexibility:

> a set of states can be simultaneously regarded as a set of relations on states, i.e., as a set of possible actions. As we said, a proposition describing a computer can be interpreted as a set of states. But depending on which state the computer is in, it is ready to execute any number of possible actions. A possible action can be viewed as a state transition, and this can be viewed abstractly as just a binary relation between states. [6, p. 507]

Meyer and Dunn's key insight was that this flexibility raises the possibility of using ternary relation semantics to give a semantics for the untyped λ-calculus. The connection between λ-calculi and the above observation about ternary-relational semantics is the following:

> The problem in interpreting the (untyped) λ-calculus of [2], or the combinatorylogic [sic] of [4], has always been how to interpret an expression such as MM, which treats M as simultaneously standing for both a function and an argument. . . .
>
> The discussion above of the ternary relation showed how to achieve the efect of a similar type-defying interpretation: a proposition B can be simultaneously thought of as both a set of states and a set of relations between states . . . Where ρ is a relation and B is a set, the ρ image of $B = \{\chi : \exists\beta \in B\ (\rho\beta\chi)\}$. Going up a type-level, where A is a set of relations it is natural to define a corresponding image $AB = \{\chi : \exists\rho \in A, \exists\beta \in B\ (\rho\beta\chi)\}$, and to think of this as a kind of application. Now regarding ρ as simultaneously a relation and a state, we stick in the ternary relation R . . . rewriting this as
>
> $$AB = \{\chi : \exists\rho \in A, \exists\beta \in B(R_\rho\beta\chi)\} = A \circ B$$
>
> Returning to the computer metaphor, $A \circ B$ is the set of states that can be reached from states in B by applying one of the actions in A. [6, p. 507ff]

The computational interpretation we offer in this paper is, in broad strokes, much the same as the computational interpretation on offer in Dunn and Meyer's paper. There are only a handful of differences worth pointing out.

The first concerns the correction of two type-mismatches that occur in the transition between Dunn and Meyer's semantics and its intended interpretation. The first type mismatch concerns application. Dunn and Meyer interpret $A \circ B$ as "the set of states that can be reached from states in B by applying one of the actions in A". This *sounds* (and the use of '$\circ$' *looks*) like a binary operation mapping pairs of states to a set of states. But they model it not with a binary function from pairs of points to sets of points but with a ternary relation among the points.

Of course, there is at best a hair's breadth of difference between a three-place relation and a two-place function with a set of outputs. That said, there's something wrong-way-rubbing about giving the intuitive explanation of one's semantics in terms of one of these things and using the other in the formal machinery. In the semantics below, I correct this type mismatch by giving a theory in which the semantic correlate of the application operation '$\circ$' is an operation that maps pairs of semantic indices to sets of indices.

The second type-mismatch concerns Dunn and Meyer's choice to use sets of indices to model individual states. Of course, there are well known Stone-ish reasons to talk in terms of sets of points (sometimes called 'UCLA propositions') instead of points.[1] Nonetheless, there's some amount of cognitive dissonance here—one expects the semantic analogue of an individual state to be an individual member of the set of indices, not a subset of the set of indices. This is also corrected in the semantics I present below.

The reader might wonder whether a switch from wrong-way-rubbing to right-way-rubbing in the semantics is enough payoff to justify making these changes. It is. As evidence of this, I'll point out that, once we type-match, we see that there is good reason to explore a broader family of logics than the family Dunn and Meyer examined. In particular, what we will see once we parse the computational metaphor along the lines suggested above is that the bottom-level logic worth looking at isn't the basic relevant logic **B** that Dunn and Meyer end their analysis at, but the yet-weaker relevant logic **BB**.[2]

The basic difference between these logics is that where **B** takes conjunction introduction as an *axiom* in the following form:

$$((A \to B) \wedge (A \to C)) \to (A \to (B \wedge C))$$

the logic **BB** instead takes it to be a rule:

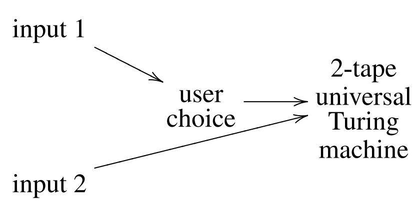
$$\frac{A \to B \qquad A \to C}{A \to (B \wedge C)}$$

The interpretation Dunn and Meyer give of their semantics is, I claim, most naturally implemented in a system that makes exactly this rule-for-axiom swap. I speculatively suggest that the reason they didn't notice this is because the ternary-relational/sets-of-indices-for-states setup they used obscured the semantic interpretation just enough to make it seem plausible that the system to include the axiom rather than the rule.

More to the point, consider the interpretation they offer: $A \to B$ is verified by a state s just if we always *can* reach a state containing B when we apply s to a state containing A. So the logic will verify the conjunction introduction *axiom* just if we always can reach a state containing $A \to (B \wedge C)$ by applying logic to a state containing both $A \to B$ and $A \to C$. But this isn't (as I'll now turn to arguing) the sort of thing that *ought* to follow from logic alone!

If we're given a state s that contains both $A \to B$ and $A \to C$, then what we know about the state is just this: if we apply s to a state containing A, then one of the things we can do is arrive at a state t_1 containing B and another of the things we can do is arrive at a state t_2 containing C. But there's no reason to think that t_1 and t_2 will be the same state and thus no reason to think that by applying s to a state containing A we can arrive at a state containing $B \wedge C$.

In fact, here's a good reason for thinking that *in general*, the computational metaphor doesn't lead us to systems that allow this inference. The system I have in mind is arranged thus:



[1] See [24] and [13], for example.

[2] For more on **BB**, the reader should consult e.g. [14] and, more recently, [25].

We suppose some encoding of Turing machines as strings in a given alphabet. Input 1 and input 2 are both strings in this same alphabet, and the 2-tape universal Turing machine is a universal Turing machine for machines encoded in this way; it thus takes a pair of strings s and t as input and outputs the result of running the Turing machine encoded by s on the string t. We can suppose that strings not encoding Turing machines are treated as, e.g. the identity machine or some such.

The 'user choice' node here represents an intervention: the user selects one machine to run from among the machines encoded on input 1. It's in this sense—and in only this sense—that the picture represents, as suggested in the title of the paper, a nondeterministic semantics. Finally, let's imagine that the inputs are literally written on strips of paper and that the user feeds them into an actual machine that destroys them in the course of doing its computation.

The observation to make, then, is this: a user can be presented with data to use as input 1 that

- supports a machine that turns tapes with data of type A into tapes with data of type B, and
- supports a machine that turns tapes with data of type A into tapes with data of type C, but nonetheless
- doesn't support a machine that turns tapes with data of type A into tapes with data of type $B \wedge C$.

And the reason for this is simple: in order for the user to get from data of type A to data of type $B \wedge C$, she would need to somehow run both the program turning A-data into B-data and the program turning A-data into C-data at the same time. But this isn't possible on the setup as presented. That is, the issue that is preventing our adopting the axiom in question is that, in general, we are allowing room for models of computation that are deeply nonconcurrent: they allow for only one computational process to be implemented at a time. Thus the 'nonconcurrent' in the title of the paper.

`Sidenote:` The 'typing' language used here is meant to remain at an intuitive level. But it seems likely that there's a generalization of the theory of what's known as intersection types—see e.g. [3], [5] and §9.2 of [1]—on offer here. It's incredibly tempting to derail the paper to pursue this—indeed, at several points below the reader is likely to note I have to tie myself in knots to avoid pursuing it—but I think it's best to instead save the matter for another day. The paper contains enough that's new in it already. `End Sidenote`

A natural question now arises: given the nonconcurrency, why should we accept even the *rule* form of conjunction introduction? Shouldn't we be throwing out both?

The answer is no; we shouldn't. The rule form of conjunction introduction is fine. Here's why: theorems represent things that are true of all computational systems meeting the conditions of our models. As we will see below, this can actually be generalized a bit: within a given model, the validities of that model represent the things that are true of all computational systems modeled by that particular model.

So, for example, let's suppose that (perhaps as a result of the particular encodings chosen) in the above setup every time we run a machine on an input tape that has data of type A on it, the result is an output of type B and every time we run a machine on an input tape that has data of type A on it, the result is an output of type C. That is, suppose that both $A \to B$ and $A \to C$ are validities in the system modeling this type of computation. Then if we run our machine on an input tape that has data of type A on it, the result will be an output that is simultaneously of type B and of type C.[3] Thus, it follows from both $A \to B$ and $A \to C$ being validities that $A \to (B \wedge C)$ is as well.

So much for motivation. It's now time to get to the work of presenting our actual semantics.

3. The Semantics

We will be working initially in a language we call $\mathcal{L}$ that is formed from the set $\mathsf{At} = \{a_i\}_{i=1}^{\infty}$ of atomic formulas/propositional variables and the connectives $\wedge$, $\vee$, and $\to$ with the usual formation rules.

We take a frame to be a 5-tuple $F = \langle T_F, \sqcap_F, \sqcup_F, N_F, \otimes_F\rangle$, where

- $\langle T_F, \sqcap_F, \sqcup_F\rangle$ is a lattice. We write $\sqsubseteq_F$ for the ordering induced by this lattice in the usual way.
- N_F is a nonempty subclass of T_F.
- $\otimes_F : T_F \times T_F \longrightarrow 2^{T_F}$.

When it's not needed—and it almost always isn't—we leave off the subscripts.

Each model as a whole represents the space of possible information states reachable using a certain computational architecture. So, for example, there would be a model representing the information states reachable using the '2-tape universal Turing machine' setup outlined above. Here by 'information state' we mean to gesture in the direction of the fact that we take the states to not merely be arrangements of bits and bytes, but to in fact be interpreted. So, e.g., states that contain '$A \wedge B$' will necessarily contain both A and B since, on interpretation, we cannot include the former in the information supported by the state without including the latter. This distinction is necessarily a bit metaphorical and loose—this is yet another place where the right way to manage is to move to a proper typing discipline; we avoid doing so for the same reasons as above.

That said, we think of the parts of a model as follows:

- We will think of T_F as the class of reachable information states. Intuitively, this means that each member of T_F is finitely generated and each will be a *theory* in the sense that it will be closed under the logic of the space F.
- $\sqcap_F$ is intended to be the intersection of theories and $\sqcup_F$ to be the (closed) union of theories.
- N_F we think of as the class of normal theories—that is, those that are contained in the logic.
- $\otimes_F$ is the (in general nondeterministic and nonconcurrent as described above) application operation.

We require that the components of a frame satisfy the following conditions:

F1: If $n \in N$ and $m \sqsubseteq n$, then $m \in N$.
F2: If $n \in N$ and $m \in N$, then $n \sqcup m \in N$.
F3: For all $t \in T$ there is $n \in N$ so that $t \in n \otimes t$.
F4: If $n \in N$ and $u \in n \otimes t$, then $u \sqsubseteq t$.
F5: If $s \sqsubseteq t \in u \otimes v$, then $s \in u \otimes v$ as well.
F6: For all $t \in T$, if $u \sqsubseteq w$, then $t \otimes u \subseteq t \otimes w$ and $u \otimes t \subseteq w \otimes t$.
F7: $(t \otimes u) \cap (s \otimes u) \subseteq (t \sqcap s) \otimes u$
F8: For all s and t, $\{u : s \in u \otimes t\}$ is nonempty and contains a least element.

The philosophical intuitions behind these requirements are as follows:

[3] For those worried about 'inconsistent types' here, we again gesture in the direction of intersection type theory.

For F1: N_F is meant to represent the class of finitely generated subtheories of the logic. F1 then forces us to say something we ought to want to say anyways; viz. that finitely generated subtheories of finitely generated subtheories of the logic are finitely generated subtheories of the logic.

For F2: Since n and m are in N, each represents a finitely generated subtheory of the logic. Their closed union, intuitively, is the theory generated by the union of their generators. So, as required by F2, it too should be a finitely generated subtheory of the logic—which is to say that it should be a member of N.

For F3: A finitely generated theory is generated by the conjunction of its generators. Also, since even the trivial computational process (the one that does nothing at all) supports every instance of $p \to p$, the theory generated by the instance of '$p \to p$' where p is the conjunction of the generators of t ought to be a subtheory of the logic of each frame F (and hence be a member of each N_F). Finally, applying this theory to t should result in a class of theories containing t, since t contains the conjunction of its generators. F3 demands exactly this.

For F4: To be a theory at all is to be closed under the logic. If $n \in N$ and $u \in n \otimes t$, then u is the result of applying some part of the logic to t. So u should, as demanded by F4, already be contained in t.

For F5: If $s \sqsubseteq t$, then intuitively t supports the generators of s. Now suppose that $t \in u \otimes v$. It follows that some program in u, when run on the data provided by v, gives us data that supports t. But as already noted, t supports the generators of s. But since we have a program and data that can get us to something that *contains* support for s, we can interpret this program and data as something provides us with data that supports s. F5 requires that the models obey this sort of reasoning.

For F6: If $s_1 \sqsubseteq s_2$, then intuitively there are no fewer programs in s_2 than there are in s_1. So we ought to get out of $s_2 \otimes t$ at least all the things we get out of $s_1 \otimes t$. A similar intuition motivates the other case.

For F7: Let $v \in (t \otimes u) \sqcap (s \otimes u)$. Then there is a program π_t in t and data $\mu_t \in u$ so that π_t, when applied to μ_t, produces a set of generators for v. Similarly, there is a program π_s in s and data $\mu_s \in u$ so that π_s, when applied to μ_s, produces a set of generators for v. But since each π, when applied to the corresponding μ, produces generators for v, each of them returns v-data. And since each μ is in u, π returns v-data on being supplied u-data. So each π is the type of program that can return generators for v on being supplied generators for u. So there is such a program in the meet of s and t as well, and thus v will also be in $(t \sqcap s) \otimes u$.

For F8: Suppose t is generated by T and s by S. Then any u containing a program that returns S on being supplied T will demonstrate nonemptyness. F8 simply requires that there always be such, which is not unreasonable. The minimality assumption amounts to the claim that we can always produce a program that essentially does this and no more.

Before moving on, let's pause to do a bit of philosophy of logic. A natural question—asked by, among other venerable persons, the referees—is this: how are we to understand the philosophical intuitions above? My preference is that they be read as simultaneously doing all of the following:

- specifying the domain being modeled by my models;
- specifying how the domain in question is being modeled by my models; and
- ensuring that the domain so-specified is in fact modeled in the specified way by my models.

Thus, e.g., in the explanation for F5, I am simultaneously informing the reader that we are only modeling computational systems and situations in which the curtailment in question can be performed and also explaining that the restriction in question ensures that $\otimes$—read as the operation of nonconcurrently, nondeterministically applying a program to some data—in my models in fact models such processes and situations.

Returning to the thread: a *model* M on a frame F is a function $T \longrightarrow 2^{\mathsf{At}}$ satisfying, for all t and u in T, the following two equations:

$$M(t \sqcap u) = M(t) \cap M(u)$$
$$M(t \sqcup u) = M(t) \cup M(u)$$

In addition, we require that a model satisfy the following condition:

M1: For all $a \in \mathsf{At}$, $\{t : a \in v(t)\}$ is nonempty and contains a least element.

M1 captures the intuition that for each atom a, there is a theory generated just by a.

Our semantic clauses are then as follows:

- $t \vDash a$ iff $a \in v(t)$.
- $t \vDash A_1 \wedge A_2$ iff there are u_i with $u_i \vDash A_i$ and $u_1 \sqcup u_2 \sqsubseteq t$.
- $t \vDash A_1 \vee A_2$ iff there are u_i with $u_i \vDash A_i$ and $u_1 \sqcap u_2 \sqsubseteq t$.
- $t \vDash A_1 \to A_2$ iff for all u, if $u_1 \vDash A_1$, then $u_2 \vDash A_2$ for some $u_2 \in t \otimes u_1$.

It's worthwhile to comment on the apparent upside-down-ery of the conjunction and disjunction clauses. On this subject we first note that the inspiration for these clauses is due to the semantics Humberstone presented in [12]. On the other hand, it's not too hard (and is a useful exercise, the solution to which can be found among the lemmas below) to see that the intersection of finitely generated theories t and u is the theory generated by the disjunctions of the generators of t and u. And it's entirely clear that the closed union of t and u will contain each conjunction of generators of t and u. This, together with a small amount of elbow grease, will give what's needed to accept these clauses. The conditional clause, on the other hand, we take to be a transparent semantification of the motivating intuitions raised in the introduction.

To complete our semantic journey, say that A is valid in M iff $n \vDash A$ for some $n \in N$ and that A is valid in a class of frames C iff A is valid in M for all $M \in C$.

Lemma 1 (Heredity)**.** If $s \sqsubseteq t$, and $s \vDash A$, then $t \vDash A$.

Proof. By induction on A. For atoms the result is immediate. For conjunctions and disjunctions, the shape of the corresponding semantic clause makes the result almost immediate. For entailments, it follows from F6. □

Lemma 2. $t \vDash A_1 \wedge A_2$ iff $t \vDash A_1$ and $t \vDash A_2$.

Proof. Suppose $t \vDash A_1 \wedge A_2$. Then there are $u_i \vDash A_i$ with $u_1 \sqcup u_2 \sqsubseteq t$. But then $u_1 \sqsubseteq t$ and $u_2 \sqsubseteq t$. So by Heredity, $t \vDash A_1$ and $t \vDash A_2$.

The converse is immediate from the fact that $t \sqcup t = t$. □

Lemma 3. For all A, $T_A := \{t : t \vDash A\}$ is nonempty and contains a least element i_A.

Proof. For atomic A, this is immediate from M1.

Suppose $A = A_1 \wedge A_2$. By the inductive hypothesis, T_{A_j} contains a least element, i_{A_j}. So $i_{A_j} \vDash A_j$, and thus $i_{A_1} \sqcup i_{A_2} \vDash A_1 \wedge A_2$. So $T_{A_1 \wedge A_2}$ is nonempty. Now let $t \in T_{A_1 \wedge A_2}$. Then there are $u_j \vDash A_j$ with $u_1 \sqcup u_2 \sqsubseteq t$. Thus each $u_j \sqsubseteq t$, so by heredity $t \vDash A_j$. It follows that $i_{A_j} \sqsubseteq t$ and thus that $i_{A_1} \sqcup i_{A_2} \sqsubseteq t$. So $i_{A_1} \sqcup i_{A_2}$ is a lower bound for $T_{A_1 \wedge A_2}$. Since it's also contained in $T_{A_1 \wedge A_2}$, it's immediately a least element as well.

Essentially the same argument, mutatis mutandis, works if $A = A_1 \vee A_2$.

Suppose $A = A_1 \to A_2$. By the inductive hypothesis, T_{A_j} contains a least element, i_{A_j}. By F8, $\{i : i_{A_2} \in u \otimes i_{A_1}\}$ contains a least element, i. We first show that $T_{A_1 \to A_2}$ is nonempty by showing that it contains i as a member. To see this, suppose $w \vDash A_1$. Then $i_{A_1} \sqsubseteq w$. So $i \otimes i_{A_1} \subseteq i \otimes w$. Thus since $i_{A_2} \in i \otimes i_{A_1}$, $i_{A_2} \in i \otimes w$. So since $i_{A_2} \vDash A_2$, $i \vDash A_1 \to A_2$ and thus $i \in T_{A_1 \to A_2}$.

Now let $t \in T_{A_1 \to A_2}$. Then since $i_{A_1} \vDash A_1$, for some $v \in t \otimes i_{A_1}$, $v \vDash A_2$. So $i_{A_2} \sqsubseteq v$. Thus by F5, $i_{A_2} \in t \otimes i_{A_1}$. So $i \sqsubseteq t$ by the definition of i. Thus i is a lower bound for $T_{A_1 \to A_2}$. □

In the remainder, we use the notation given in the statement of this Lemma and let i_A be the least element of $T_A := \{t : t \vDash A\}$.

Lemma 4. If $t_1 \vDash A$ and $t_2 \vDash A$, then $t_1 \sqcap t_2 \vDash A$.

Proof. By induction on A. For atoms, it follows from v being a lattice homomorphism.

Suppose $A = A_1 \wedge A_2$. Then there are $u^i_j \vDash A_i$ so that $u^1_j \sqcup u^2_j \sqsubseteq t_j$. Thus clearly $(u^1_1 \sqcup u^2_1) \sqcap (u^1_2 \sqcup u^2_2) \sqsubseteq t_1 \sqcap t_2$, but also

$$(u^1_1 \sqcup u^2_2) \sqcap (u^1_2 \sqcup u^2_2) = \underbrace{[(u^1_1 \sqcap u^1_2) \sqcup (u^2_1 \sqcap u^1_2)]}_{v_1} \sqcup \underbrace{[(u^1_1 \sqcap u^2_2) \sqcup (u^2_1 \sqcap u^2_2)]}_{v_2}$$

By the inductive hypothesis, $u^i_1 \sqcap u^i_2 \vDash A_i$. So by heredity, $v_i \vDash A_i$. Thus $t_1 \sqcap t_2 \vDash A_1 \wedge A_2$.

Suppose $A = A_1 \vee A_2$. Then there are $u^i_j \vDash A_i$ so that $u^1_j \sqcap u^2_j \sqsubseteq t_j$. Thus clearly $(u^1_1 \sqcap u^2_1) \sqcap (u^1_2 \sqcap u^2_2) \sqsubseteq t_1 \sqcap t_2$, but also $(u^1_1 \sqcap u^2_1) \sqcap (u^1_2 \sqcap u^2_2) = (u^1_1 \sqcap u^1_2) \sqcap (u^2_1 \sqcap u^2_2)$. By the inductive hypothesis, $u^i_1 \sqcap u^i_2 \vDash A_i$. Thus $t_1 \sqcap t_2 \vDash A_1 \vee A_2$.

Finally, suppose $A = A_1 \to A_2$. Then for all u, if $u \vDash A_1$, then there is $v_i \in t_i \otimes u$ so that $v_i \vDash A_2$. It follows by the inductive hypothesis that $v_1 \sqcap v_2 \vDash A_2$. And since $v_1 \sqcap v_2 \sqsubseteq v_i$, F5 gives that $v_1 \sqcap v_2 \in t_i \otimes u$. So by F7, $v_1 \sqcap v_2 \in (t_1 \sqcap t_2) \otimes u$. So $t_1 \sqcap t_2 \vDash A_1 \to A_2$ □

Lemma 5. $A \to B$ is valid in M iff for all $t \in M$, if $t \vDash A$, then $t \vDash B$.

Proof. Suppose $A \to B$ is valid in M. Then for some $n \in N$, $n \vDash A \to B$. Let $t \vDash A$. Then $u \vDash B$ for some $u \in n \otimes t$. But since $u \in n \otimes t$, $u \sqsubseteq t$. Thus by heredity, $t \vDash B$.

Now suppose for all $t \in M$, if $t \vDash A$ then $t \vDash B$. By Lemma 3, T_B has a least element, i_B. By F3, there is $n \in N$ so that $i_B \in n \otimes i_B$. I claim that $n \vDash A \to B$. To see this, let $t \vDash A$. Then $t \vDash B$. So $i_B \sqsubseteq t$. Thus $n \otimes i_B \subseteq n \otimes t$. So since $i_B \in n \otimes i_B$, $i_B \in n \otimes t$ as well. And since $i_B \vDash B$, this finishes the job. □

4. Soundness

BB$^+$ is axiomatized as follows:

Axioms:

(1) $A \to A$

(2) $(A \wedge B) \to A$; $(A \wedge B) \to B$

(3) $A \to (A \vee B)$; $B \to (A \vee B)$

(4) $[A \wedge (B \vee C)] \to [(A \wedge B) \vee (A \wedge C)]$

Rules:

(1) $\dfrac{A \qquad A \to B}{B}$

(2) $\dfrac{A \qquad B}{A \wedge B}$

(3) $\dfrac{A \to B \qquad C \to D}{(B \to C) \to (A \to D)}$

(4) $\dfrac{A \to B_1 \qquad A \to B_2}{A \to (B_1 \wedge B_2)}$

(5) $\dfrac{A_1 \to B \qquad A_2 \to B}{(A_1 \vee A_2) \to B}$

Theorem 6. All theorems of **BB**$^+$ are valid in the class of all frames.

Proof. Fix a model M. We show by induction on the derivation of the theorem in question that it is valid in M. Recall that by F3, for all $t \in T$, there is $n \in N$ so that $t \in n \otimes t$.

A1 Case: Let $i_A \in n \otimes i_A$. Let $t \in T_A$. Then $i_A \sqsubseteq t$. Thus $n \otimes i_A \subseteq n \otimes t$. So $i_A \in n \otimes t$, from which it follows that $n \models A \to A$. The A2 and A3 cases are essentially the same.

A4 Case: Let $i_{(A\wedge B)\vee(A\wedge C)} \in n \otimes i_{(A\wedge B)\vee(A\wedge C)}$. Let $t \models A \wedge (B \vee C)$. Then there are $u_1 \models A$ and $u_2 \models B \vee C$ so that $u_1 \sqcup u_2 \sqsubseteq t$. Since $u_2 \models B \vee C$, there are $u_3 \models B$ and $u_4 \models C$ so that $u_3 \sqcap u_4 \sqsubseteq u_2$. It follows that

$$u_1 \sqcup (u_3 \sqcap u_4) = (u_1 \sqcup u_3) \sqcap (u_1 \sqcup u_4) \sqsubseteq u_1 \sqcup u_2 \sqsubseteq t$$

But also $u_1 \sqcup u_3 \models A \wedge B$ and $u_1 \sqcup u_4 \models A \wedge C$. So $(u_1 \sqcup u_3) \sqcap (u_1 \sqcup u_4) \in T_{(A\wedge B)\vee(A\wedge C)}$. Thus

$$i_{(A\wedge B)\vee(A\wedge C)} \sqsubseteq (u_1 \sqcup u_3) \sqcap (u_1 \sqcup u_4) \sqsubseteq t.$$

It follows that $i_{(A\wedge B)\vee(A\wedge C)} \in n \otimes t$, so that $t \models (A \vee (B \wedge C)) \to ((A \wedge B) \vee (A \wedge C))$.

R1 Case: Let A and $A \to B$ be valid in M and suppose in particular that $n_1 \models A$ and $n_2 \models A \to B$. Then for some $u \in n_2 \otimes n_1$, $u \models B$. But since $n_1 \in N$ and $u \in n_2 \otimes n_1$, F4 gives that $u \sqsubseteq n_1$. Thus by F1, $u \in N$. So B is valid.

R2 Case: Let A and B be valid in M and suppose in particular that $n_1 \models A$ and $n_2 \models B$. Then clearly $n_1 \sqcup n_2 \models A \wedge B$. And by F2, $n_1 \sqcup n_2 \in N$. Thus $A \wedge B$ is valid.

R3 Case: Suppose $A \to B$ and $C \to D$ are valid in M and suppose in particular that $n_1 \models A \to B$ and $n_2 \models C \to D$. By Lemma 5, it suffices to show that if $t \models B \to C$, then $t \models A \to D$. So let $t \models B \to C$ and $u \models A$. Then $v \models B$ for some $v \in n_1 \otimes u$. Since $n_1 \in N$, F4 then gives that $v \sqsubseteq u$. Thus $t \otimes v \subseteq t \otimes u$.

Also, since $v \models B$, there is $w \in t \otimes v \subseteq t \otimes u$ so that $w \models C$. And since $w \models C$, there is $x \in n_2 \otimes w$ so that $x \models D$. But since $n_2 \in N$, F4 again gives that $x \sqsubseteq w$. So since $w \in t \otimes u$, $x \in t \otimes u$ as well. Thus $t \models A \to D$. The R4 case is essentially the same.

R5 Case: Suppose $A_i \to B$ are valid, and in particular that $n_i \models A_i \to B$. Let $t \models A_1 \vee A_2$. Then there are $u_i \models A_i$ so that $u_1 \sqcap u_2 \sqsubseteq t$. Since $u_i \models A_i$, there are $v_i \in n_i \otimes u_i$ with $v_i \models B$. Since $n_i \in n_i \otimes u_i$, $v_i \sqsubseteq u_i$. So $v_i \sqcap v_2 \sqsubseteq u_1 \sqcap u_2$. And since $v_i \models B$, $v_1 \sqcap v_2 \models B$. Thus $t \models B$ as well, by heredity. By Lemma 5, this finishes the job.

□

5. Completeness

We prove completeness by constructing a canonical model. Some definitions are needed:

- Given a set of formulas X, and a logic L we say that X is a formal L-theory when the following two conditions are met:
  - If $A \in X$ and $B \in X$, then $A \wedge B \in X$.
  - If $A \in X$ and $A \to B \in L$, then $B \in X$.
- For X a set of formulas, we write $[X]_L$ for the set of formulas L-generated by X, which we define to be $\{B :$ there are $A_i \in X$ with $(A_1 \wedge \cdots \wedge A_n) \to B \in L\}$.

Lemma 7. $[X]_{\mathbf{BB}^+}$ is a formal $\mathbf{BB}^+$-theory for any set X.

Proof. Let $B_1 \in [X]$ and $B_2 \in [X]$. Then there are $A_i^j \in X$ for which $(A_i^1 \wedge \cdots \wedge A_i^{n_i}) \to B_i \in$ $\mathbf{BB}^+$. But then $((A_1^1 \wedge \cdots \wedge A_1^{n_1}) \wedge (A_2^1 \wedge \cdots \wedge A_2^{n_2})) \to (B_1 \wedge B_2) \in \mathbf{BB}^+$. So $B_1 \wedge B_2 \in [X]$.

Now let $B \in [X]$ and $B \to C \in \mathbf{BB}^+$. Since $B \in [X]$, there are $A_i \in X$ so that $(A_1 \wedge \cdots \wedge A_n) \to B \in \mathbf{BB}^+$. But then since $B \to C \in \mathbf{BB}^+$ as well, so also is $(A_1 \wedge \cdots \wedge A_n) \to C$. Thus $C \in [X]$. □

If $X = \{C\}$ is a singleton, we write $[C]_L$ rather than $[\{C\}]_L$. Formal theories of the form $[C]_L$ are called *principal* formal theories. From here on, when it can be determined by context we will drop the subscripted 'L's.

We define the canonical frame F_C to be the tuple $\langle Th^{\mathbf{BB}^+}_{prin}, \cap, \overline{\cup}, \mathbf{BB}^+_{prin}, \otimes_C \rangle$ where

- $Th^{\mathbf{BB}^+}_{prin}$ is the set of principal formal $\mathbf{BB}^+$-theories
- $\cap$ is ordinary intersection,
- $t \,\overline{\cup}\, s = [t \cup s]$,
- $\mathbf{BB}^+_{prin}$ is the set of principal formal subtheories of $\mathbf{BB}^+$.
- $s \otimes_C t = \{[B] : \text{for some } A \in t, A \to B \in s\}$

In general we will omit the subscripted 'C' in $t \otimes_C u$, where t and u are principal formal theories, taking it to be understood from context.

Lemma 8. $[A] \cap [B] = [A \vee B]$.

Proof. Containment right-to-left being clear, it suffices to demonstrate containment left-to-right. For this, suppose $C \in [A]$ and $C \in [B]$. Then $A \to C \in \mathbf{BB}^+$ and $B \to C \in \mathbf{BB}^+$. So by R5, $(A \vee B) \to C \in \mathbf{BB}^+$, and thus $C \in [A \vee B]$. □

Lemma 9. $[A] \,\overline{\cup}\, [B] \subseteq [A \wedge B]$

Proof. Immediate from the definitions. □

Lemma 10. The induced order in the canonical frame is containment.

Proof. Recalling that the induced order on a lattice $\langle L, \sqcap, \sqcup \rangle$ is defined by $s \sqsubseteq t$ iff $s = s \sqcap t$, this is essentially immediate from the definitions involved. □

Lemma 11. $[S] \subseteq [T]$ iff $T \to S \in \mathbf{BB}^+$

Proof. Let $[S] \subseteq [T]$. Then $S \in [T]$. So $T \to S \in \mathbf{BB}^+$.

Let $T \to S \in \mathbf{BB}^+$ and $S' \in [S]$. Then $S \to S' \in \mathbf{BB}^+$. So $T \to S' \in \mathbf{BB}^+$. Thus $S' \in [T]$, so $[S] \subseteq [T]$. □

Lemma 12. The canonical frame is a frame.

Proof. We must verify first that $\langle Th^{\mathbf{BB}^+}_{prin}, \cap, \overline{\cup} \rangle$ is a lattice, then that F_C satisfies F1-F8.

Given the prior lemmas, to show that we've got a lattice on our hands it suffices to observe that each of the following is a theorem:

- $A \to (A \vee (A \wedge B))$ (by A3)
- $(A \vee (A \wedge B)) \to A$ (by A1, A2, and R5)
- $(A \wedge (A \vee B)) \to A$ (by A2)
- $A \to (A \wedge (A \vee B))$ (by A1, A2, and R4)

We deal with the remaining conditions individually:

F1: Immediate.
F2: Immediate from the lemmas.
F3: Let $t = [T]$. Then $n = [T \to T]$ does the job.
F4: Let $n \in \mathbf{BB}^+_{prin}$. If $u = [U] \in n \otimes t$, then for some $A \in t$, $A \to U \in n \subseteq \mathbf{BB}^+$. But then since $A \to U \in \mathbf{BB}^+$, $U \in t$, so $[U] \subseteq t$, as required.

F5: Let $[S] \subseteq [T] \in [U] \otimes [V]$. Then for some $V' \in [V]$, $V' \to T \in [U]$. Since $[S] \subseteq [T]$, $T \to S \in \mathbf{BB}^+$. Thus (by A1 and R3) $(V' \to T) \to (V' \to S) \in \mathbf{BB}^+$. So $V' \to S \in [U]$. So $[S] \in [U] \otimes [V]$.

F6: Immediate.

F7: Let $[S] \in (t_1 \otimes [U]) \cap (t_2 \otimes [U])$. Then there are $U_i \in [U]$ with $U_i \to S \in t_i$. Since $U_i \in [U]$, $U \to U_i \in \mathbf{BB}^+$. Thus $(U_i \to S) \to (U \to S) \in \mathbf{BB}^+$. So $U \to S \in t_i$, and thus $U \to S \in t_1 \cap t_2$. So $[S] \in (t_1 \cap t_2) \otimes [U]$.

F8: Let $s = [S]$ and $t = [T]$. Then clearly $s \in [T \to S] \otimes [T]$. So $\{u : s \in u \otimes t\}$ is nonempty. Also, if $[S] \in u \otimes [T]$, then $T \to S \in u$, and thus $[T \to S] \subseteq u$. So $[T \to S]$ is minimal.

□

The canonical model augments the canonical frame with the function $M_C : t \mapsto t \cap \mathsf{At}$. To see that this satisfies M1, it suffices to observe that for all $a \in \mathsf{At}$, $[a]$ clearly contains a and is minimal among theories that contain a. Thus the canonical model is a model on the canonical frame.

Lemma 13. $M_C, t \vDash A$ iff $A \in t$.

Proof. By induction on A. For atoms it is immediate from the definition of M_C.

Suppose $A = A_1 \wedge A_2$ and $M_C, t \vDash A$. Then there are u_i so that $u_i \vDash A_i$ and $u_1 \overline{\cup} u_2 \subseteq t$. By the inductive hypothesis, since $u_i \vDash A_i$, $A_i \in u_i$. Thus $A_1 \wedge A_2 \in u_1 \overline{\cup} u_2$ and thus $A_1 \wedge A_2 \in t$. On the other hand, if $A \in t$, then clearly $[A_1] \overline{\cup} [A_2] = [A] \subseteq t$. And, by the inductive hypothesis, $[A_i] \vDash A_i$, finishing the job. Mutatis mutandis, the same argument works if $A = A_1 \vee A_2$ as well.

Finally, let $A = A_1 \to A_2$ and suppose $A \in t$ and $u \vDash A_1$. Then by the inductive hypothesis, $A_1 \in u$. Thus $[A_2] \in t \otimes u$. And, again by the inductive hypothesis, $[A_2] \vDash A_2$. So $t \vDash A_1 \to A_2$.

If instead $A_1 \to A_2 \notin t$, then I claim that $[A_2] \notin t \otimes [A_1]$. Notice that by the inductive hypothesis, it follows from this that $t \nvDash A_1 \to A_2$, so this will suffice.

To prove the claim, suppose to the contrary that $[A_2] \in t \otimes [A_1]$. Then for some $A_1' \in [A_1]$, $A_1' \to A_2 \in t$. But since $A_1' \in [A_1]$, $A_1 \to A_1' \in \mathbf{BB}^+$. It follows that $(A_1' \to A_2) \to (A_1 \to A_2) \in \mathbf{BB}^+$. Thus $A_1 \to A_2 \in t$, which is a contradiction. □

Theorem 14. If A is valid in the class of all $\mathbf{BB}^+$ frames, then A is provable in $\mathbf{BB}^+$.

Proof. Suppose A isn't provable in $\mathbf{BB}^+$. Then A isn't in $\mathbf{BB}^+$. So A is in no principle formal subtheory of $\mathbf{BB}^+$. Thus A isn't valid in the canonical model. □

6. Extensions 1: Structural Rules

There are a variety of structural rules we might add to the system. Each of these corresponds to more-or-less plausible conditions on the computation system in question. The extensions we'll examine (together with the corresponding frame conditions) are the 'positive' members of the ones Fine examined in [7] together with clauses that get us up from $\mathbf{BB}^+$ to $\mathbf{B}^+$. To state the frame conditions concisely, it helps to extend $\otimes$ in two ways: to a function $T \times 2^T \longrightarrow 2^T$ and to a function $2^T \times T \longrightarrow 2^T$. We do so by stipulating that $t \otimes S = \bigcup_{s \in S} t \otimes s$ and similarly that $S \otimes t = \bigcup_{s \in S} s \otimes t$.

Lemma 15. Let t, u, and v be formal L-theories for some L extending $\mathbf{BB}^+$. Then if $[B]_L \in t \otimes (u \otimes v)$, then there is $A \to B \in t$ with $[A]_L \in u \otimes v$.

Proof. If $[B]_L \in t\otimes(u\otimes v)$, then for some $A' \to B \in t$, there is $[A]_L \in u\otimes v$ so that $A' \in [A]_L$. But then $A \to A' \in L$. So $(A' \to B) \to (A \to B) \in L$. Thus $A \to B \in t$ □

Lemma 16. Let t, u, and v be formal L-theories for some L extending $\mathbf{BB}^+$. If $[B]_L \in (t \otimes u) \otimes v$, then there is $A \in v$ with $[A \to B]_L \in t \otimes u$.

Proof. Let $[B]_L \in (t \otimes u) \otimes v$. Then for some $[C]_L \in t \otimes u$, there is $A \to B \in [C]_L$ with $A \in u$. Since $[C]_L \in t \otimes u$, there is $D \to C \in t$ with $D \in u$. And since $A \to B \in [C]_L$, $C \to (A \to B) \in L$. So $(D \to C) \to (D \to (A \to B)) \in L$ as well. So $D \to (A \to B) \in t$, and thus $[A \to B] \in t \otimes u$. □

Theorem 17. In the following chart, the logic that extends $\mathbf{BB}^+$ with one of axioms (1)-(7) is sound and complete for the class of frames satisfying the (universal closure of the) constraint listed on the right.

(1)	$((A \to B) \wedge (B \to C)) \to (A \to C)$	$t \otimes (t \otimes u) \subseteq t \otimes u$
(2)	$(A \to B) \to ((B \to C) \to (A \to C))$	$t \otimes (u \otimes v) \subseteq (u \otimes t) \otimes v$
(3)	$(B \to C) \to ((A \to B) \to (A \to C))$	$t \otimes (u \otimes v) \subseteq (t \otimes u) \otimes v$
(4)	$(A \wedge (A \to B)) \to B$	If $u \in t \otimes t$, then $u \sqsubseteq t$
(5)	$(A \to (B \to C)) \to ((A \wedge B) \to C)$	$(t \otimes u) \otimes u \subseteq t \otimes u$
(6)	$A \Rightarrow (A \to B) \to B$	If $n \in N$ and $u \in t \otimes n$, then $u \sqsubseteq t$
(7)	$A \to ((A \to B) \to B)$	$t \otimes u = u \otimes t$
(8)	$((A \to B) \wedge (A \to C)) \to (A \to (B \wedge C))$	If $v_i \in t \otimes u$ then $v_1 \sqcup v_2 \in t \otimes u$
(9)	$((A \to C) \wedge (B \to C)) \to ((A \vee B) \to C)$	$(t \otimes v_1) \cap (t \otimes v_2) \subseteq t \otimes (v_1 \sqcap v_2)$

Proof. We consider each axiom individually. Since in each case we are considering a subclass of the class of all frames, each of Lemmas 1-5 still holds. We write F_n for the class of frames satisfying condition n, and $\mathbf{BB}^+_n$ for the logic axiomatized by $\mathbf{BB}^+$ together with the nth axiom. We leave it to the reader to verify that Lemmas 7-11 hold when replacing $\mathbf{BB}^+$ with $\mathbf{BB}^+_n$.

(1): Soundness part: By Lemma 5, it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash (A \to B) \wedge (B \to C)$, then $t \vDash A \to C$. So suppose $t \vDash (A \to B) \wedge (B \to C)$. Then by Lemma 2, $t \vDash A \to B$ and $t \vDash B \to C$. Now let $u \vDash A$. Then $v \vDash B$ for some $v \in t \otimes u$. Thus $w \vDash C$ for some $w \in t \otimes v$. But then $w \in t \otimes (t \otimes u)$. So $w \in t \otimes u$. So $t \vDash A \to C$.

Completeness part: suppose $[C]_{\mathbf{BB}^+_1} \in t \otimes (t \otimes u)$. By Lemma 15, there is $B \to C \in t$ with $[B]_{\mathbf{BB}^+_1} \in t \otimes u$. But since $[B]_{\mathbf{BB}^+_1} \in t \otimes u$, there is $A \to B \in t$ with $A \in u$. So then $A \to B$ and $B \to C$ are in t and thus since t is a $\mathbf{BB}^+_1$-theory, $A \to C \in t$. So $[C]_{\mathbf{BB}^+_1} \in t \otimes u$.

The arguments for (2) and (3) are similar.

(4): Soundness part: By Lemma 5, it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash A \wedge (A \to B)$, then $t \vDash B$. So suppose $t \vDash A \wedge (A \to B)$. Then by Lemma 2, $t \vDash A$ and $t \vDash A \to B$. So there is $u \in t \otimes t$ with $u \vDash B$. But since $u \in t \otimes t$, $u \sqsubseteq t$. So by heredity, $t \vDash B$ as well.

Completeness part: suppose $[B]_{\mathbf{BB}^+_4} \in t \otimes t$ and $B' \in [B]_{\mathbf{BB}^+_4}$. Then since $[B] \in t \otimes t$, there is $A \to B \in t$ with $A \in t$. So $A \wedge (A \to B) \in t$. Thus $B \in t$. And since $B' \in [B]$, $B \to B' \in \mathbf{BB}^+_4$. So $B' \in t$, and thus $[B] \subseteq t$.

(5): Soundness part: By Lemma 5, it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash A \to (B \to C)$, then $t \vDash (A \wedge B) \to C$. So suppose $t \vDash A \to (B \to C)$ and let $u \vDash A \wedge B$. Then by Lemma 2, $u \vDash A$ and $u \vDash B$.

Since $u \vDash A$, for some $v \in t \otimes u$, $v \vDash B \to C$. Thus for some $w \in v \otimes u$, $w \vDash C$. But then $w \in (t \otimes u) \otimes u \subseteq t \otimes u$. So $t \vDash (A \wedge B) \to C$.

Completeness part: suppose $[C]_{\mathbf{BB}_5^+} \in (t \otimes u) \otimes u$. Then by Lemma 16, there is $B \in u$ with $[B \to C]_{\mathbf{BB}_5^+} \in t \otimes u$. So there is $A \in u$ with $A \to (B \to C) \in t$. But then $(A \wedge B) \to C \in t$. And since $A \in u$ and $B \in u$, Lemma 2 gives that $A \wedge B \in u$. Thus $[C]_{\mathbf{BB}_5^+}$, and thus $(t \otimes u) \otimes u \subseteq t \otimes u$.

(6): Soundness part: consider a model based on a frame satisfying the condition. Suppose $N \ni n \vDash A$. By Lemma 5, to show that $(A \to B) \to B$ is also valid in the model, it suffices to show that if $t \vDash A \to B$, then $t \vDash B$. So, let $t \vDash A \to B$. Then $u \vDash B$ for some $u \in t \otimes n$. But then $u \sqsubseteq t$. So $t \vDash B$ as well.

Completeness part: let $n \in N$ and $[B]_{\mathbf{BB}_6^+} \in t \otimes n$. Then there is $A \in n$ with $A \to B \in t$. But since $A \in n$, $A \in \mathbf{BB}_6^+$. Thus $(A \to B) \to B \in \mathbf{BB}_6^+$. So since $A \to B \in t$, $B \in t$. So $[B]_{\mathbf{BB}_6^+} \subseteq t$.

(7): Soundness part: By Lemma 5, it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash A$, then $t \vDash (A \to B) \to B$. So let $t \vDash A$ and let $u \vDash A \to B$. Then $v \vDash B$ for some $v \in u \otimes t = t \otimes u$. So $t \vDash (A \to B) \to B$.

Completeness part: Let $[B]_{\mathbf{BB}_7^+} \in t \otimes u$. Then there is $A \in u$ with $A \to B \in t$. But since $A \in u$, $(A \to B) \to B \in u$ as well. So $[B]_{\mathbf{BB}_7^+} \in u \otimes t$. The converse containment is established with essentially the same argument.

(8): Soundness part: By Lemmas 2 and 5, it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash A \to B$ and $t \vDash A \to C$, then $t \vDash A \to (B \wedge C)$. So suppose $t \vDash A \to B$ and $t \vDash A \to C$. Let $u \vDash A$. Then there are $v_i \in t \otimes u$ so that $v_1 \vDash B$ and $v_2 \vDash C$. But then by assumption $v_1 \sqcup v_2 \in t \otimes u$ and by Lemma 2, $v_1 \sqcup v_2 \vDash B \wedge C$. So $t \vDash A \to (B \wedge C)$.

Completeness part: Let $v_i \in t \otimes u$, and suppose $v_i = [V_i]_{\mathbf{BB}_8^+}$. Then there are $A_i \in u$ with $A_i \to V_i \in t$. But since $(A_1 \wedge A_2) \to A_i \in \mathbf{BB}_8^+$ and $V_i \to V_i \in \mathbf{BB}_8^+$, it follows by rule (3) that $(A_i \to V_i) \to ((A_1 \wedge A_2) \to V_i) \in \mathbf{BB}_8^+$. So since $A_i \to V_i \in t$, we also have that $(A_1 \wedge A_2) \to V_i \in \mathbf{BB}_8^+$. So $(A_1 \wedge A_2) \to (V_1 \wedge V_2) \in \mathbf{BB}_8^+$. Finally, since $A_i \in u$, clearly $A_1 \wedge A_2 \in u$. Thus $[V_1 \wedge V_2] = [V_1] \sqcup [V_2] = v_1 \sqcup v_2 \in t \otimes u$.

(9): Soundness part: By Lemma 2 and 5, it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash A \to C$ and $t \vDash B \to C$, then $t \vDash (A \vee B) \to C$. So suppose $t \vDash A \to B$ and $t \vDash A \to C$. Let $u \vDash A \vee B$. Then there are v_i with $v_1 \sqcap v_2 \sqsubseteq u$ and $v_1 \vDash A$ and $v_2 \vDash B$. Then there are $w_i \in t \otimes v_i$ with $w_i \vDash C$. Since $w_1 \sqcap w_2 \sqsubseteq w_i$, it follows by F5 that $w_1 \sqcap w_2 \in t \otimes v_i$. So $w_1 \sqcap w_2 \in (t \otimes v_1) \cap (t \otimes v_2)$. It follows then that $w_1 \sqcap w_2 \in t \otimes (v_1 \sqcap v_2)$. But by Lemma 4, since $w_i \vDash C$, $w_1 \sqcap w_2 \vDash C$. So $t \vDash (A \vee B) \to C$, as required.

Completeness Part: Let $w = [W] \in (t \otimes v_1) \cap (t \otimes v_2)$. Then there are $A_i \in v_i$ with $A_i \to W \in t$. So $(A_1 \to W) \wedge (A_2 \to W) \in t$. Thus $(A_1 \vee A_2) \to W \in t$. And since $A_i \in v_i$, it follows that $A_1 \vee A_2 \in v_1 \cap v_2$. Thus $[W] = w \in t \otimes (v_1 \cap v_2)$ as required.

□

Before turning to the next sort of extension, we'll pause first to think about the computational interpretation of each of the above extensions. We'll work our way from the top to the bottom. It's useful, in doing so, to consider and add notation for a natural operation on programs. In particular, where p_1 and p_2 are programs, we'll write $comp(p_2, p_1)$ for the 'composite' program we get by first running program p_1, then running program p_1. With this in hand, here are the computational interpretation of the semantic restrictions required to effect each of the above extensions of the logic:

For (1): choose $v \in t \otimes (t \otimes u)$. Then there is a program p_1 in t that generates v when applied to data in $t \otimes u$. Thus in fact there are programs p_1 and p_2 in t such that, if p_2 is applied to data in u and p_1 is applied to the data that results, we wind up with data that supports v. If the result is already in $t \otimes u$, then t must support $comp(p_1, p_2)$. Stated more simply, the computational interpretation of the semantic condition is this: data supporting any two programs p_1 and p_2 also supports the program we get by composing these programs. One way to understand this in the case of systems like the one examined in §1 is as an increase in the complexity of the instructions the user can provide to the system: instead of only giving instructions 'run this program on that data', we're now allowing the user to give instructions of the form 'run this program on that data, then run the following program on the resulting data'.

For (2) and (3), the change amounts to a restriction in the behavior of programs when applied to programs. More to the point, what's required for (2) is that each program p be such that, when it is applied to a program q, the result is always a program that executes $comp(q, p)$, while (3) instead requires that the result be $comp(p, q)$. Each of these is a natural-enough sort of restriction one might adopt on the kinds of programs one might consider.

Restrictions (1), (2), and (3) each correspond to modifications of the computational metaphor that keep its rough structure in tact. The remaining restrictions, however, push the bounds of plausibility on the metaphor a bit past the breaking point.

For (4), we are demanding a restriction to something that either corresponds to a type of saturation or a type of omniscience. In either event, what's required is that any data that supports both a program and a piece of information also support the result of executing the program on the information. This can be seen as requiring that we only consider 'saturated' bodies of data—bodies of data that only include programs if they also include all data that result from running that program on the data. Alternatively, it might be taken to require a type of 'computational omniscience'; that is, on assuming that being presented with both a program and data *just is* being presented with the result of running the program on the data.

(5) then takes these notions of saturation/omniscience to the natural next step: given data that support a program and (possibly different) data that support information *just is* being presented with data that support all the programs that result from running the program on the information. Thus, I require not just that data be saturated with respect to itself, but with respect to all the other data around as well. Or, in terms of omniscience, it requires that every time I can generate programs from data, those programs appear to me as already included in the data.

Accommodating (6) and (7) in the computational metaphor is yet more difficult. In some sense, each requires what seem like quite drastic and unnatural restrictions on the data one considers. For (7), for example, what's required is this: a given program p can turn something in u into data d iff u can turn the program p into data d. Thus, u can't contain *anything* that could, by some program p, be manipulated into something that u couldn't produce when applied to p. (6) requires something similar, but only for data that forms a subset of the logic.

(8) and (9) both (as should be expected) explicitly break the nonconcurrency at the heart of the semantics. This is easier to see in (8) than in (9); (8) basically says that if there are programs p_i in t that generate v_i, then there is a program—the program that runs p_1 and p_2 concurrently—that generates $v_1 \sqcup v_2$. (9) on the other hand says that if we are given two different ways of generating data d—say by running p_1 on e_1 or by running p_2 on e_2—then

if we already support both p_1 and p_2, we can generate d by running a program π that we already support on any data that supports the meet of e_1 and e_2. To see how this builds in concurrency, one has to ask what π should be. The answer *isn't* that π should be a program that can take either e_1 or e_2 and return d because such a program can't be guaranteed to generate d whenever it is run on data that support the meet of e_1 and e_2—the meet of e_1 and e_2 might, after all, contain neither e_1 nor e_2. What's needed instead is something that can run on the sort of disjunctive data one expects to find in the meet of e_1 and e_2. Now having a look at the axiom supported by (9), we see what the answer is: what we need is a program that concurrently runs both the program that returns d from e_1 and the program that returns d from e_2, and does this when applied to the 'disjunction' $e_1 \vee e_2$.

6.1. **With Modalities.** A different way to extend $\mathbf{BB}^+$ is by adding modalities to it. We do this by first adding a new unary connective '$\Box$' to the language. From here, the 'obvious' thing to do is to augment our models with a meet-preserving map $' : T \longrightarrow T$. That is, we require that $(t \sqcap s)' = t' \sqcap s'$. It follows that $'$ is also order-preserving, since $t \leq s =_{def} t \sqcap s = t$, and thus if $t \sqsubseteq s$, then $t' \sqcap s' = (t \sqcap s)' = t'$ so that $t' \sqsubseteq s'$. In addition—and this time unlike Fine—we must require that this function satisfy the following constraint:

- For all $t \in T$, $\{s \in T : t \sqsubseteq s'\}$ is nonempty and contains a least element.[4]

The semantic clause for $\Box$ is then defined by $t \vDash \Box A$ iff $t' \vDash A$. Call the set of formulas valid in the resulting semantics $\mathbf{BB}^{\Box+}$.

The problem with this route to providing a semantics for modals is that the completeness proof relies on a conjecture I cannot prove at this time. The conjecture is this:

Conjecture: If t is a principal $\mathbf{BB}^{\Box+}$-theory, then so too is $\{A : \Box A \in t\}$.

Clearly we need to verify this for the expected 'priming' function in the canonical model to actually be well-defined. Showing that the conjecture holds, however, seems to rely on some fairly subtle features of $\mathbf{BB}^{\Box+}$-proofs, so will have to await further work.

What we'll do instead is replace the function $' : T \longrightarrow T$ that we want with a function $'' : T \longrightarrow 2^T$ that does the job. This new function intuitively maps each theory t to the set of finitely generated subtheories of what we wanted to map to. That is $''$ maps to the set of principal subtheories of the 'unboxing', t', of t.

A note: in spite of entirely replacing the single-priming function with the double-priming function, we'll retain the double-priming notation since the single-priming function is, in the completeness parts of the metatheory below, useful to have around.

What we explicitly require is that the function $''$ have all the following features:

- $(t \sqcap s)'' = t'' \cap s''$.
- If $s \in t''$ and $u \sqsubseteq s$, then $u \in t''$.
- If $s_i \in t''$, then $s_1 \sqcup s_2 \in t''$.

From the first point it follows (as we leave to the reader to verify) that if $t \sqsubseteq s$, then $t'' \subseteq s$.

We then adopt the following frame constraint:

F10 For all $t \in T$, $\{s \in T : t \sqsubseteq u$ for some $u \in s''\}$ is nonempty and contains a least element.

The semantic clause for $\Box$, on the other hand, becomes the following: $t \vDash \Box A$ iff $u \vDash A$ for some $u \in t''$.

[4]In [7], no mention is actually made of the need for $'$ to be order-preserving, but since Fine relies on heredity just as much as I do, it seems we should charitably take this to be an essentially typographical error.

With regard to the computational metaphor, we can understand the double-prime function (and thus the box) as a call to a family of libraries. The call is, initially, very weak—just a pointer in fact.[5] As we strengthen the semantics, the call will get stronger. For example, the axiom $\Box A \to A$ strengthens the call from a pointer to an import—all the data one calls is loaded as additional data. Thus, in an analogy that is particularly poignant as I struggle to compile this document, when so-augmented $\Box A$ behaves a bit like the LaTeX command `\includepackage{A}`.

Our first task is to verify that all our requisite lemmas still work.

Lemma 18. Suppose M is a modal model. Then all of the following hold:

(a) If $s \sqsubseteq t$, then if $s \vDash A$ then $t \vDash A$.
(b) $t \vDash A_1 \wedge A_2$ iff $t \vDash A_1$ and $t \vDash A_2$.
(c) For all A, T_A is nonempty and contains a least element.
(d) If $t_1 \vDash A$ and $t_2 \vDash A$, then $t_1 \sqcap t_2 \vDash A$.
(e) $A \to B$ is valid in M iff for all $t \in M$, if $t \vDash A$, then $t \vDash B$.

Proof. We need only examine (a), (c), and (d) to get the rest. And in these cases, we need only examine the new clause in the corresponding inductions.

For (a), suppose $s \sqsubseteq t$ and $s \vDash \Box A$. Then $u \vDash A$ for some $u \in s''$. But since $s \sqsubseteq t$, $s'' \subseteq t''$. Thus $t \vDash \Box A$ as well.

For (c), suppose $A = \Box A_1$. By the inductive hypothesis, T_{A_1} has a least element, i_{A_1}. By F10, $\{s \in T : i_{A_1} \sqsubseteq u \text{ for some } u \in s''\}$ is nonempty and has a least element, j. Thus, for some $u \in j''$, $i_{A_1} \sqsubseteq u$. So $u \vDash A_1$, and thus $j \vDash \Box A_1$, so $j \in T_A$. Now suppose $t \in T_A$. Then $t \vDash \Box A_1$, so for some $u \in t''$, $u \vDash A_1$. So $i_{A_1} \sqsubseteq u$. Thus $j \sqsubseteq t$, from which it follows that j is a least member of T_A.

For (d), suppose $t_i \vDash \Box A$. Then there are $u_i \in t_i$ with $u_i \vDash A$. By the inductive hypothesis, $u_1 \sqcap u_2 \vDash A$. But $u_1 \sqcap u_2 \sqsubseteq u_i$, so $u_1 \sqcap u_2 \in t_i''$, and thus $u_1 \sqcap u_2 \in t_1'' \cap t_2'' = (t_1 \sqcap t_2)''$. Thus $t_1 \sqcap t_2 \vDash \Box A$. □

Theorem 19. $\mathbf{BB}^{\Box+}$ is axiomatized by **BB** plus the following axiom and rule:

- $(\Box A \wedge \Box B) \to \Box(A \wedge B)$
- $\dfrac{A \to B}{\Box A \to \Box B}$

Proof. Soundness part: By Lemma 18(e), to show that $(\Box A \wedge \Box B) \to \Box(A \wedge B)$ is valid in the above semantics it suffices to show that if $t \vDash \Box A \wedge \Box B$, then $t \vDash \Box(A \wedge B)$. So suppose $t \vDash \Box A \wedge \Box B$. Then by Lemma 18(b), $t \vDash \Box A$ and $t \vDash \Box B$. So there are $u \in t''$ and $v \in t''$ with $u \vDash A$ and $v \vDash B$. But then $u \sqcup v \in t''$ as well and clearly $u \sqcup v \vDash A \wedge B$. Thus $t \vDash \Box(A \wedge B)$.

Now suppose that $n \vDash A \to B$. By Lemma 18(e), to show that $\Box A \to \Box B$ is valid, it suffices to show that if $t \vDash \Box A$, then $t \vDash \Box B$. So let $t \vDash \Box A$. Then $u \vDash A$ for some $u \in t''$. Thus there is $v \in n \otimes u$ so that $v \vDash B$. But since $v \in n \otimes u$, $v \sqsubseteq u$. So $v \in t''$ as well. Thus $t \vDash \Box B$.

For completeness, we define the canonical model mostly as before, but add that $t''^c = \{[A] : [A] \subseteq t'^c\}$ where $t'^c = \{B : \Box B \in t\}$. It then suffices to establish the following:

- $(t \cap s)''^c = t''^c \cap s''^c$.
- If $s \in t''^c$ and $u \subseteq s$, then $u \in t''^c$.
- If $s_i \in t''^c$, then $s_1 \overline{\cup} s_2 \in t''^c$.

[5]Perhaps in this weakest case, it's best to think of '□' as the 'citation modal'—it points to other work without directly including that work in the work at hand.

- For all $t \in Th^{\mathbf{BB}^{\Box+}}_{prin}$, $\{s \in Th^{\mathbf{BB}^{\Box+}}_{prin} : t \subseteq u$ for some $u \in s''^{C}\}$ is nonempty and contains a least element.
- For all $t \in Th^{\mathbf{BB}^{\Box+}}_{prin}$, $t \vDash \Box A$ iff $\Box A \in t$.

From the first four it will follow that the canonical model is a model, and the last will allow us to augment the proof of Lemma 13 as expected. From there we finish the job in the expected way. Throughout the rest of this proof, we omit for the sake of readability the subscripted 'C's.

For the first point, let $[U] \in ([T] \cap [S])''$. Then $[U] \in [T \vee S]''$, so $[U]$ is a principal subtheory of $[T \vee S]' = \{A : (T \vee S) \to \Box A \in \mathbf{BB}^{\Box+}\}$. Thus, there are $A_i \in [T \vee S]'$ so that $(A_1 \wedge \cdots \wedge A_n) \to U \in \mathbf{BB}^{\Box+}$. But then $(T \vee S) \to \Box A_i \in \mathbf{BB}^{\Box+}$, and thus $T \to \Box A_i \in \mathbf{BB}^{\Box+}$ and $S \to \Box A_i \in \mathbf{BB}^{\Box+}$. So $[U]$ is a principal subtheory of $[T]'$ and of $[S]'$ as well. So $[U] \in [T]'' \cap [S]''$.

For the other direction, suppose $[U] \in [T]'' \cap [S]''$. Then $[U]$ is a principal subtheory of $[T]'$ and a principal subtheory of $[S]'$. So $T \to \Box U \in \mathbf{BB}^{\Box+}$ and $S \to \Box U \in \mathbf{BB}^{\Box+}$. It follows that $(T \vee S) \to \Box U \in \mathbf{BB}^{\Box+}$, so $[U]$ is a principal subtheory of $[T \vee S]' = ([T] \cap [S])'$ as well. So $[U] \in ([T] \cap [S])''$.

The second and third points follow almost immediately from the definitions.

For the fourth point, again let $t = [T]$ and consider the theory $[\Box T]$. Clearly $T \in [\Box T]'$, and thus $[T] \in [\Box T]''$. So $t \subseteq u$ for some $u \in [\Box T]''$. Thus the set is nonempty. Now suppose s is in the set. Then for some $u \in s''$, $t \subseteq u$. So $T \in u$. So $T \in s'$, and thus $\Box T \in s$. So $[\Box T] \subseteq s$, so $[\Box T]$ is minimal.

For the final point, note that $t \vDash \Box A$ iff $u \vDash A$ for some $u \in t''$ iff (by the inductive hypothesis) $A \in u$ iff $\Box A \in t$. $\Box$

We can naturally extend $''$ to a function $2^T \longrightarrow 2^T$ by defining $S'' = \bigcup_{s \in S} s''$. Using this, we can then give semantic characterizations of the same extensions of the basic modal logic $\mathbf{BB}^{\Box+}$ as above. As with Theorem 17, we will pause after proving the theorem to give computational interpretations of each extension.

Theorem 20. In the following chart, the logic that extends $\mathbf{BB}^{\Box+}$ with one of axioms (10)-(13) is sound and complete for the class of frames satisfying the (universal closure of the) constraint listed on the right.

(10)	$\Box A \to A$	if $u \in t''$, then $u \sqsubseteq t$
(11)	$\Box A \to \Box\Box A$	$t'' \subseteq (t'')''$
(12)	$\Box(A \to B) \to (\Box A \to \Box B)$	$\bigcup_{v \in t'', w \in u''} v \otimes w \subseteq (t \otimes u)''$
(13)	$\dfrac{A}{\Box A}$	$N \subseteq N''$.

Proof. We again consider each axiom individually

(10): Soundness part: By Lemma 18(e), it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash \Box A$, then $t \vDash A$. So let $t \vDash \Box A$. Then $u \vDash A$ for some $u \in t''$. But then $u \sqsubseteq t$, so Lemma 18(a) gives that $t \vDash A$.

Completeness part: suppose $A \in u \in [T]''^{C}_{\mathbf{BB}^{\Box+}_8}$. Then $\Box A \in t$. So since $\Box A \to A \in \mathbf{BB}^{\Box+}_8$, $A \in t$. So $u \subseteq t$.

(11): Soundness part: By Lemma 18(e), it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash \Box A$, then $t \vDash \Box\Box A$. So let $t \vDash \Box A$. Then there is $u \in t''$ so that $u \vDash A$. But since $t'' \subseteq (t'')''$, it follows that for some $v \in t''$, $u \in v''$. So since $u \vDash A$, $v \vDash \Box A$ and thus $t \vDash \Box\Box A$.

Completeness part: $[U]_{\mathbf{BB}_9^{\Box+}} \in t''^c$. Then $\Box U \in t$. So since $\Box U \to \Box\Box U \in \mathbf{BB}_9^{\Box+}$, $\Box\Box U \in t$ as well. Thus, $[\Box U] \in t''^c$. And since $[U] \in [\Box U]''^c$, it follows that $[U] \in (t''^c)''^c$.

(12): Soundness part: By Lemma 18(e), it suffices to show that in any model based on a frame satisfying the condition, if $t \vDash \Box(A \to B)$, then $t \vDash \Box A \to \Box B$. So let $t \vDash \Box(A \to B)$ and $u \vDash \Box A$. Then $v \vDash A \to B$ for some $v \in t''$ and $w \vDash A$ for some $w \in u''$. Thus $x \vDash B$ for some $x \in v \otimes w$. So since $\bigcup_{\substack{v \in t'' \\ w \in u''}} v \otimes w \subseteq (t \otimes u)''$, $x \in (t \otimes u)''$ as well. It follows that for some $y \in t \otimes u$, $x \in y''$. And since $x \vDash B$, $y \vDash \Box B$. So $t \vDash \Box A \to \Box B$.

Completeness part: Let $[X]_{\mathbf{BB}_{10}^{\Box+}} \in v \otimes w$ for some $v \in t''^c$ and $w \in u''^c$. Then there is $Y \in w$ for which $Y \to X \in v$. Since $v \in t''^c$, $v \subseteq t'^c$. And since $w \in u''^c$, $w \subseteq u'^c$. Thus $\Box(Y \to X) \in t$ and $\Box Y \in u$. And since $\Box(Y \to X) \to (\Box Y \to \Box X) \in \mathbf{BB}_{10}^{\Box+}$, we then get that $\Box Y \to \Box X \in t$ and thus that $[\Box X]_{\mathbf{BB}_{10}^{\Box+}} \in t \otimes u$. So $[X]_{\mathbf{BB}_{10}^{\Box+}} \in (t \otimes u)''^c$.

(13): Soundness part: Suppose we have a model satisfying the condition. Let $n \vDash A$ for some $n \in N$. Then since $N \subseteq N''$, $n \in m''$ for some $m \in N$. So $m'' \vDash \Box A$.

Completeness part: suppose $[A]_{\mathbf{BB}_{11}^{\Box+}} \in N$. Then since $\mathbf{BB}_{11}^{\Box+}$ is closed under the rule, $[\Box A]_{\mathbf{BB}_{11}^{\Box+}} \in N$. So $[A]_{\mathbf{BB}_{11}^{\Box+}} \in N''^c$.

□

The first two axioms have very natural computational interpretations: (10) requires that library called by □ be actually loaded—that is, that its contents be made part of the current theory. (11) requires that libraries called by libraries be called. Together (10) and (11) then demand that we load libraries and that we load libraries loaded by libraries.

(12)'s computational interpretation is fairly straightforward as well. It requires (essentially) that application be extended piecewise to libraries. That is, anything I can get by applying some part of the library t calls to some part of the library u calls must be part of the library of something I can get by applying t to u. There is a certain sort of concurrency being baked in here—intuitively, when applying t plus library_t to u plus library_u, what (12) requires is that we apply something in t to u and *simultaneously* apply part of library_t to library_u. This is, of course, unsurprising since (12) is a modal analogue of the concurrency-destroying (8).

Finally, (13) is probably best thought of as a restriction on the notion of *ab*normality. Explicitly, (13) requires that each normal theory must be part of the library called by some (not necessarily distinct) normal theory. Thus, nothing that counts as normal is such that calling it as a library immediately makes a theory abnormal. Computationally, the idea is that a call to something that implements the support relation cannot be the sort of thing that guarantees one is looking at data that is *not* implementing the support relation.

Conclusion and Future Work

The computational metaphor first pointed out by Dunn and Meyer in 1997 is, it seems, still ripe for exploration. We've clearly just scratched the surface here, and we already see that it naturally extends in a way that incorporates nondeterministic and nonconcurrent types of computation. The resulting semantic theory—which seems to more smoothly incorporate Dunn and Meyer's motivating intuitions—naturally captures the very weak relevant logic $\mathbf{BB}^+$. And straightforward restrictions on the class of models then give rise to various well-known strengthenings of this logic. Altogether this suggests a rich and powerful new semantics for an interesting class of logics.

On the modal front, while the semantics is certainly capable of handling everything we've thrown at it, the metaphor seems a bit more fragile. In particular, while it seems that modalities obeying (10), (11), and (13) are all compatible with the computational metaphor we've been examining, modalities obeying (12) are not. We're left then with six of the thirteen axioms we've examined—axioms (1)-(3) and axioms (10), (11), and (13)—that can plausibly be added without doing damage to the computational metaphor in play. Intuitively, adding any collection of these axioms should also be compatible with the computational metaphor. This leaves us with a whopping 36 computationally interesting logics worth exploring already. And of course, we've only looked at the low-hanging fruit here—axioms that have been considered already in other contexts. That said, the fact that the same semantic paradigm works across such a broad range of different extensions suggests that the computational metaphor underlying the semantics is fairly robust. It also suggests novel ways of interpreting several systems that have heretofore seen very little exploration.

Of course, the elephants in the room are (a) negation and (b) the conjecture. On (a), given the computational perspective I've advocated here, intuitionistic-style negations are quite natural, but what exactly it means for a negation to be 'intuitionistic' in the setting at hand is less than clear (though some useful thoughts in this direction can be found in [19]). It would also be interesting to examine a range of other negations, including both relevant negations and the 'boolean' negations that have been explored in the relevant literature in various places (see e.g. [17] and the 'relevant' chapters in [10] to start).

Two other natural extensions worth examining are the extension to quantificational logics and the extension to justification logics, whose study in the relevant realm is in its infancy (see [21] and [23] for the state of the art). In the former case, it will be interesting to see whether, how, and to what extent the well-known difficulties relevant logics face with Tarksian quantification (see [9] for the locus classicus) arise here. On the latter front, there's hope for Curry-Howard-ish things (see e.g. [11]) to push the computational metaphor further. In particular, to the extent that justification terms record proofs and proofs just are programs, one would expect that we might be able to say something more concrete about the computational content of the semantics at hand.

Also worth examining is the extent to which the semantics I've given here is amenable to *non-canonical* models. One has the sense that models of the sort specified will be very close to being canonical. But it isn't clear (a) how to measure 'closeness' to the canonical model nor (b) how to go about even *trying* to construct non-canonical models. I think both of these actually present rather interesting lines to explore in this area, but will have to leave them to later work.

References

[1] Katalin Bimbó. *Combinatory logic: Pure, applied and typed*. CRC Press, 2011.

[2] Alonzo Church. *The Calculi of Lambda-Conversion*. Princeton, NJ, USA: Princeton University Press, 1941.

[3] Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame journal of formal logic*, 21(4):685–693, 1980.

[4] Haskell B. Curry and Robert Feys. *Combinatory Logic, vol. I.* Amsterdam (North Holand), 1972.

[5] Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, and Betti Venneri. The "relevance" of intersection and union types. *Notre Dame Journal of Formal Logic*, 38(2):246–269, 1997.

[6] J Michael Dunn and Robert K Meyer. Combinators and structurally free logic. *Logic Journal of IGPL*, 5(4):505–537, 1997.

[7] Kit Fine. Models for entailment. *Journal of Philosophical Logic*, pages 347–372, 1974.

[8] Kit Fine. Semantics for quantified relevance logic. *Journal of Philosophical Logic*, pages 27–59, 1988.
[9] Kit Fine. Incompleteness for quantified relevance logics. In *Directions in relevant logic*, pages 205–225. Springer, 1989.
[10] Dov M Gabbay and Heinrich Wansing. *What is negation?* Springer, 1999.
[11] William A Howard. The formulae-as-types notion of construction. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, 44:479–490, 1980.
[12] IL Humberstone. Operational semantics for positive R. *Notre Dame J. Formal Log.*, 29(1):61–80, 1988.
[13] Peter T Johnstone. *Stone spaces*, volume 3. Cambridge university press, 1982.
[14] Peter Lavers. Generalising tautological entailment. *Logique et Analyse*, 37(147/148):367–377, 1994.
[15] Shay Allen Logan. Deep fried logic. *Erkenntnis*, 87(1):257—286, 2022.
[16] Shay Allen Logan. The universal theory tool building toolkit is substructural. In *Alasdair Urquhart on Nonclassical and Algebraic Logic and Complexity of Proofs*, pages 261–285. Springer, 2022.
[17] Graham Priest. Boolean negation and all that. *Journal of Philosophical Logic*, 19(2):201–215, 1990.
[18] Stephen Read. *Relevant logic*. Oxford: Blackwell, 1988.
[19] Gemma Robles and José M Méndez. *Routley-Meyer ternary relational semantics for intuitionistic-type negations*. Academic Press, 2018.
[20] Richard Routley, Val Plumwood, Robert K Meyer, and Ross T Brady. *Relevant logics and their rivals*. 1982.
[21] Nenad Savic and Thomas Studer. Relevant justification logic. *FLAP*, 6(2):397–412, 2019.
[22] John Slaney. A general logic. *Australasian Journal of Philosophy*, 68(1):74–88, 1990.
[23] Shawn Standefer. Weak relevant justification logics. *Journal of Logic and Computation*, 2022.
[24] Marshall H Stone. The theory of representation for boolean algebras. *Transactions of the American Mathematical Society*, 40(1):37–111, 1936.
[25] Andrew Tedder. Information flow in logics in the vicinity of BB. *The Australasian Journal of Logic*, 18(1):1–24, 2021.