Condition/Decision Duality and the Internal Logic of Extensive Restriction Categories

Robin Kaarsgaard [1,2]

DIKU, Department of Computer Science
University of Copenhagen

Abstract

In flowchart languages, predicates play an interesting double role. In the textual representation, they are often presented as *conditions*, *i.e.*, expressions which are easily combined with other conditions (often via Boolean combinators) to form new conditions, though they only play a supporting role in aiding branching statements choose a branch to follow. On the other hand, in the graphical representation they are typically presented as *decisions*, intrinsically capable of directing control flow yet mostly oblivious to Boolean combination.

While categorical treatments of flowchart languages are abundant, none of them provide a treatment of this dual nature of predicates. In the present paper, we argue that extensive restriction categories are precisely categories that capture such a condition/decision duality, by means of morphisms which, coincidentally, are also called decisions. Further, we show that having these categorical decisions amounts to having an internal logic: Analogous to how subobjects of an object in a topos form a Heyting algebra, we show that decisions on an object in an extensive restriction category form a *De Morgan quasilattice*, the algebraic structure associated with the (three-valued) *weak Kleene logic* $\mathbf{K}_3^{\mathrm{w}}$. Full classical propositional logic can be recovered by restricting to *total* decisions, yielding extensive categories in the usual sense, and confirming (from a different direction) a result from effectus theory that predicates on objects in extensive categories form Boolean algebras.

As an application, since (categorical) decisions are partial isomorphisms, this approach provides naturally reversible models of classical propositional logic and weak Kleene logic.

Keywords: categorical logic, flowchart languages, restriction categories, extensivity, weak Kleene logic

1 Introduction

Flowchart languages are a particular class of imperative programming languages which permit a pleasant and intuitive graphical representation of the control flow of programs. While conceptually very simple, flowchart languages form the foundation for modern imperative programming languages, and have been used for this

---

reason as vehicles for program analysis (*e.g.*, to measure coverage in white-box testing [1]), program transformations (*e.g.*, partial evaluation, see [17]), and to express fundamental properties of imperative programming, such as the equivalence of expressivity in *structured* and *unstructured* programming in the *Böhm-Jacopini theorem* [4] (see also [3, 28]). Figure 1 shows the (textual and graphical) flowchart structures used by structured flowchart languages.

An interesting feature in flowchart languages is the dual presentation of predicates as *conditions* and *decisions*, depending on the context. On the one hand, the textual **if** p **then** c_1 **else** c_2 seems to favor the view of p as a condition, *i.e.*, a predicate which has inherently nothing to do with control flow, but which may easily be combined with other conditions to other conditions to form new ones. In other words, the textual representation considers the branching behaviour to be given by the semantics of **if** ... **then** ... **else** ... rather than by the semantics of p. This view is also emphasized by the usual (big-step) operational semantics of conditionals: Here, predicates are treated as expressions that may be evaluated in a state to yield a Boolean value, which the conditional may then branch on, as in

$$\frac{\langle p, \sigma \rangle \to \textbf{true} \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \textbf{if } p \textbf{ then } c_1 \textbf{ else } c_2, \sigma \rangle \to \sigma'} \qquad \text{and} \qquad \frac{\langle p, \sigma \rangle \to \textbf{false} \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \textbf{if } p \textbf{ then } c_1 \textbf{ else } c_2, \sigma \rangle \to \sigma'} \; .$$

On the other hand, the graphical representation of conditionals in Figure 1(c) seems to rather prefer the view of p as a decision, *i.e.*, a kind of flowchart operation intrinsically capable of directing control flow. That is to say, that this is a structured flowchart (corresponding to a conditional) is purely coincidental; for *unstructured* flowcharts to make sense, p *must* be able to direct control flow on its own. However, where conditions are most naturally composed via the Boolean combinators, the only natural way of composing decisions seems to be in sequence (though this leads to additional output branches).

While categorical models of structured flowchart languages have been widely studied (see, *e.g.*, [2, 10, 11, 24, 26, 27]), none provide a treatment of this dual view of predicates. In this paper, we argue that extensive restriction categories are precisely categories that make clear this dual view on predicates as conditions and decisions, offering both the ease of combination of conditions and the control flow behaviour of decisions. Restriction categories (introduced in [7–9]) are categories of partial maps, in which each morphism is equipped with a *restriction idempotent* that, in a certain sense, gauges how partial that morphism is. Since models of flowchart languages most provide a notion of partiality (due to possible nontermination), restriction categories provide an ideal setting for such models. Coincidentally, the defining feature of extensive restriction categories [3] is the presence of certain morphisms called *decisions*, which play a similar role as the decision view on predicates in flowchart languages.

In this setting, we show that the correspondence between conditions and decisions is exhibited precisely as a natural isomorphism between the *predicate fibration*

---

[3] Note that while extensive restriction categories are strongly connected to extensive categories, they are confusingly *not* extensive in the usual sense of extensive categories [5].

Atomic operations, f.   Sequential composition, $c_1;c_2$.   Conditionals, **if** p **then** c_1 **else** c_2.   While loops, **while** p **do** c.
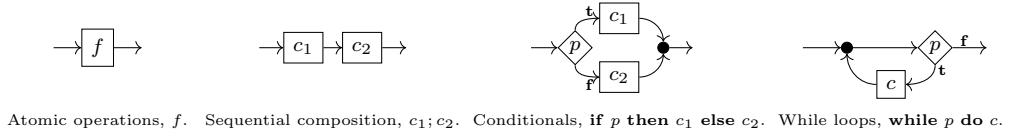
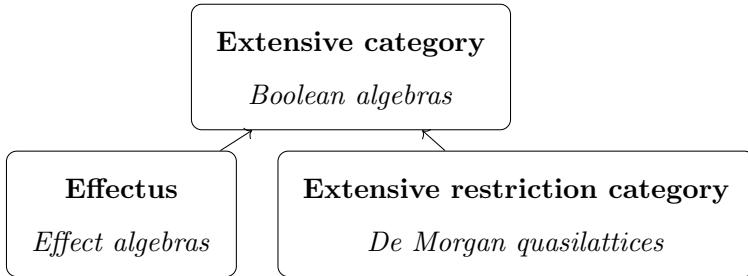Fig. 1. The four flowchart structures.

Fig. 2. Extensive categories, extensive restriction categories, and effecti: their relationships and associated logics.

$\mathrm{Hom}(X, 1+1)$ of predicates and predicate transformers (see also [6, 16]), and the *decision fibration* $\mathrm{Dec}(X)$ of decisions (certain morphisms $X \to X+X$) and decision transformers. We then go on to explore the structure of $\mathrm{Dec}(X)$ (or equivalently, $\mathrm{Hom}(X, 1+1)$), showing that this extends to a fibration over the category of *De Morgan quasilattices* and homomorphisms, which give algebraic semantics [12] to Kleene's *weak logic* $\mathbf{K}_3^{\mathrm{w}}$ [21]. Intuitively, $\mathbf{K}_3^{\mathrm{w}}$ can be seen as a partial version of classical (Boolean) logic. We make this statement precise in this setting by showing that if we restrict ourselves to *total* decisions and decision transformations, classical logic can be recovered. Since the subcategory of objects and total morphisms of a (split) extensive restriction category is an extensive category in the usual sense (see, *e.g.*, [5]), we can use this to provide an alternative proof of a statement from Effectus theory [6, 16] that predicates over each extensive category forms a fibred Boolean algebra via the predicate fibration [6, Prop. 61, Prop. 88]. This yields a relationship diagram of effecti, extensive categories, and extensive restriction categories and their corresponding logics as shown in Figure 2.

This paper is structured as follows: Section 2 gives a brief introduction to extensive restriction categories. Section 3 demonstrates the condition/decision duality of extensive restriction categories by showing that the decision and predicate fibrations are naturally isomorphic; and, as a consequence, that *decisions are a property of the predicates*. Then, in Section 4, we show that the decisions on an object form models of $\mathbf{K}_3^{\mathrm{w}}$, with decision transformers as homomorphisms. By restricting to only *total* decisions, we show that these restrict to models of classical logic. Finally, 5 offers some concluding remarks.

2   Extensive restriction categories

This section gives an introduction to extensive restriction categories as it will be applied in the sections that follow. The experienced reader may safely skip this

section on a first reading, and instead refer back to it as necessary.

Restriction categories are categories equipped with notions of partiality and totality of morphisms. This is done by means of a *restriction combinator*, assigning to each morphism $X \xrightarrow{f} Y$ its *restriction idempotent* $X \xrightarrow{\overline{f}} X$ (subject to certain laws) which may intuitively be thought of as a partial identity defined precisely where f is defined. In this way, restriction categories provide an axiomatic (and relatively light-weight) approach to partiality of morphisms in categories. Formally, restriction categories are defined in the following way:

Definition 2.1 A *restriction structure* on a category consists of a combinator mapping each morphism f to its *restriction idempotent* $\overline{f}$, *i.e.*

$$\frac{X \xrightarrow{f} Y}{X \xrightarrow{\overline{f}} X}$$

subject to the *restriction laws*:

(R1) $f\overline{f} = f$ for all $X \xrightarrow{f} Y$,

(R2) $\overline{f}\,\overline{g} = \overline{g}\,\overline{f}$ for all $X \xrightarrow{f} Y$ and $X \xrightarrow{g} Z$,

(R3) $\overline{f\overline{g}} = \overline{f}\,\overline{g}$ for all $X \xrightarrow{f} Y$ and $X \xrightarrow{g} Z$, and

(R4) $\overline{g}f = f\overline{gf}$ for all $X \xrightarrow{f} Y$ and $Y \xrightarrow{g} Z$.

A category equipped with a restriction structure is called a *restriction category*.

As the name suggests, a restriction structure is a structure on a category rather than a property of it; in particular, a category can be equipped with several different restriction structures. For this reason, we must in principle specify which restriction structure we are using when speaking of a particular category as a restriction category, though this is often omitted when the restriction structure is implicitly given to be a canonical one.

Given that restriction categories are built on a foundation of idempotents, one would expect it to be occasionally useful when all such restriction idempotents split, and indeed this is the case. Say that restriction structure is *split* when all restriction idempotents split, and let $\mathrm{Split}(\mathscr{C})$ denote the category arising from the usual idempotent splitting (*i.e.*, the *Karoubi envelope*) of all restriction idempotents in $\mathscr{C}$. That $\mathrm{Split}(\mathscr{C})$ is a restriction category when $\mathscr{C}$ is follows by [7, Prop. 2.26]

As a canonical example, the category **Pfn** of sets and partial functions is a restriction category, with the restriction idempotent $X \xrightarrow{\overline{f}} X$ for $X \xrightarrow{f} Y$ given by

$$\overline{f}(x) = \begin{cases} x & \text{if } f \text{ is defined at } x \\ \text{undefined} & \text{otherwise} \end{cases}$$

In a restriction category, say that a morphism $X \xrightarrow{f} Y$ is *total* if $\overline{f} = \mathrm{id}_X$, and that

it is a *partial isomorphism* if there exists $Y \xrightarrow{f^\dagger} X$ such that $f^\dagger f = \overline{f}$ and $ff^\dagger = \overline{f^\dagger}$. Partial isomorphisms thus generalize ordinary isomorphisms, as an isomorphism is then a partial isomorphism $X \xrightarrow{f} Y$ such that both f and $f^\dagger$ are total.

Since total morphisms are closed under composition and include all identities, they form an important subcategory $\mathrm{Total}(\mathscr{C})$ of any restriction category $\mathscr{C}$. Likewise, partial isomorphisms are closed under composition and include all identities, so all objects and partial isomorphisms of $\mathscr{C}$ form the subcategory $\mathrm{Inv}(\mathscr{C})$. As the notation suggests, this category $\mathrm{Inv}(\mathscr{C})$ is not just a restriction category but an *inverse category* (indeed, it is the *cofree* such [18]) in the usual sense (see [7, 20]).

A useful property of restriction categories is that they come with a natural partial order on homsets (which extends to enrichment in **Poset**) given by $f \leq g$ iff $g\overline{f} = f$. Intuitively, this can be thought of as an *information order*; $f \leq g$ if g can do everything f can do, and possibly more.

Like any other categorical structure, when working in restriction categories we require everything in sight to cooperate with the restriction structure. One of the simplest examples of cooperation with restriction structure is given in the definition of a restriction terminal object: This is simply a terminal object 1 in the usual sense, which further satisfies that the unique map $X \to 1$ is *total* for all objects X. For coproducts, this means that we not only require the restriction category to have all finite coproducts in the usual sense, but also that the coproduct injections $X \xrightarrow{\kappa_1} X + Y$ and $Y \xrightarrow{\kappa_2} X + Y$ are *total*. In this case, we say that the restriction category has *restriction coproducts*. There is also a similar notion of a *restriction zero* object 0: a zero object in the usual sense which additionally satisfies that each zero endomorphism $X \xrightarrow{0_{X,X}} X$ is its own restriction idempotent, *i.e.*, that $\overline{0_{X,X}} = 0_{X,X}$ (or equivalently, that $\overline{0_{X,Y}} = 0_{X,X}$ for all zero morphisms $0_{X,Y}$). When zero morphisms exist, they serve as least element in their homset with respect to the natural ordering, and when a category has restriction coproducts and a restriction zero object, the restriction zero object serves as unit for the restriction coproduct. When this is the case, restriction coproduct injections are further partial isomorphisms (*e.g.*, the partial inverse to $X \xrightarrow{\kappa_1} X + Y$ is $X + Y \xrightarrow{[\mathrm{id},0]} X$).

Extensivity for restriction categories means that the restriction coproducts are particularly well-behaved, in the sense that they admit a *calculus of matrices* [9]. Concretely, this means that each morphism $X \xrightarrow{f} Y + Z$ is associated with a unique morphism $X \xrightarrow{\langle f \rangle} X + X$, its *decision*, which, intuitively, makes the same branching choices as f does, but doesn't do any actual work. Extensive restriction categories are defined as follows.

Definition 2.2 A restriction category with restriction coproducts and a restriction zero is said to be an *extensive restriction category* if each morphism f has a unique *decision* $\langle f \rangle$, *i.e.*

$$\frac{X \xrightarrow{f} Y + Z}{X \xrightarrow{\langle f \rangle} X + X}$$

satisfying the *decision laws*

(D1) $\nabla\langle f\rangle = \overline{f}$ $\hspace{4cm}$ (D2) $(f+f)\langle f\rangle = (\kappa_1 + \kappa_2)f$

where $X + X \xrightarrow{\nabla} X$ is the natural codiagonal $[\mathrm{id}, \mathrm{id}]$.

Note that extensive restriction categories are not extensive in the usual sense – rather, extensive restriction categories are the "partial" version of extensive categories. This connection is made precise by the following proposition due to [9].

Proposition 2.3 *Whenever $\mathscr{C}$ is an extensive restriction category, $\mathrm{Total}(\mathrm{Split}(\mathscr{C}))$ is an extensive category.*

A straightforward example of an extensive restriction category is **Pfn**. Here, the decision $X \xrightarrow{\langle f\rangle} X + X$ of a partial function $X \xrightarrow{f} Y + Z$ is given by

$$
\langle f\rangle(x) = \begin{cases} \kappa_1(x) & \text{if } f(x) = \kappa_1(y) \text{ for some } y \in Y \\ \kappa_2(x) & \text{if } f(x) = \kappa_2(z) \text{ for some } z \in Z \\ \text{undefined if } f \text{ undefined at } x \end{cases}
$$

For further examples and details on extensive restriction categories, see [9].

3 Condition/decision duality

Categorical models of flowcharts are categories with a notion of partiality (due to possible nontermination) and coproducts (corresponding to the control flows of the flowchart). As such, restriction categories with restriction coproducts serve as a good starting point for these. We show in this section that the additional requirement of *extensivity* of the restriction coproduct allows the category to exhibit a condition/decision duality, analogous to the flowchart languages. This manifests in the category as a natural isomorphism between the *decisions* and *predicates* over an object (with their corresponding transformations).

We start with a few technical lemmas regarding the partial order on morphisms in a restriction category as well as properties of decisions in extensive restriction categories.

Lemma 3.1 *It is the case that*

(i) $g \le g'$ *implies* $\overline{gf} \le \overline{g'f}$,

(ii) $\overline{gf} \le \overline{f}$,

(iii) $f \le g$ *implies* $hf \le hg$ *and* $fh' \le gh'$,

(iv) $f \le f'$ *and* $g \le g'$ *iff* $f + g \le f' + g'$.

(v) $f \le \overline{g}$ *implies* $f = \overline{f}$.

Lemma 3.2 *Let $X \xrightarrow{f} Y + Z$ and $X' \xrightarrow{g} X$ be arbitrary morphisms of an extensive restriction category, and $X \xrightarrow{\overline{e}} X$ any restriction idempotent. It is the case that*

(i) $\langle\!\langle f\rangle\!\rangle = \langle f\rangle$

(ii) $\langle f\rangle$ *is a partial isomorphism and*
$$\langle f\rangle^\dagger = \left[\overline{\kappa_1^\dagger f}, \overline{\kappa_2^\dagger f}\right]$$

(iii) $\overline{\langle f\rangle^\dagger} = \overline{\kappa_1^\dagger f} + \overline{\kappa_2^\dagger f}$

(iv) $\overline{\langle f\rangle} = \overline{f}$

(v) $\gamma\langle f\rangle = \langle\gamma f\rangle$

(vi) $\langle\!\langle f\rangle g\rangle = \langle fg\rangle$

(vii) $(\overline{e}+\overline{e})\langle f\rangle = (\overline{e}+\overline{e})\langle f\rangle\overline{e}$

(viii) $\langle f\rangle\overline{e}$ *is a decision and* $\langle f\rangle\overline{e} = (\overline{e}+\overline{e})\langle f\rangle$

(ix) $\langle f\rangle\overline{e} = \langle f\overline{e}\rangle$

(x) $\kappa_i^\dagger\langle f\rangle = \overline{\kappa_i^\dagger f}$

(xi) $\langle g\rangle f = (f+f)\langle gf\rangle$

A few of these identities were shown already in [9]; the rest are mostly straight-forward to derive. Note that a direct consequence of (i) is that $(\langle f\rangle + \langle f\rangle)\langle f\rangle = (\kappa_1 + \kappa_2)\langle f\rangle$; we will make heavy use of this fact in Section 4. Another particularly useful identity is the following, stating intuitively that anything that behaves as a decision in each component is, in fact, a decision.

Lemma 3.3 *If* $\kappa_1^\dagger p = \overline{\kappa_1^\dagger f}$ *and* $\kappa_2^\dagger p = \overline{\kappa_2^\dagger f}$ *then* $p = \langle f\rangle$.

Proof. By Lemma 3.2(ii) $\langle f\rangle^\dagger = \left[\overline{\kappa_1^\dagger f}, \overline{\kappa_2^\dagger f}\right]$. Since $\kappa_1^\dagger p = \overline{\kappa_1^\dagger f}$ and $\kappa_2^\dagger p = \overline{\kappa_2^\dagger f}$ it follows that $\overline{\kappa_1^\dagger f} = (\overline{\kappa_1^\dagger f})^\dagger = (\kappa_1^\dagger p)^\dagger = p^\dagger\kappa_1$ and $\overline{\kappa_2^\dagger f} = (\overline{\kappa_2^\dagger f})^\dagger = (\kappa_2^\dagger p)^\dagger = p^\dagger\kappa_2$ so it follows by the universal mapping property of the coproduct that $p^\dagger = \left[\overline{\kappa_1^\dagger f}, \overline{\kappa_2^\dagger f}\right] = \langle f\rangle^\dagger$, and finally $p = \langle f\rangle$ by unicity of partial inverses. $\square$

As a corollary, p is a decision if $\kappa_1^\dagger p$ and $\kappa_2^\dagger p$ are both restriction idempotents (*i.e.*, if $\kappa_1^\dagger p = \overline{\kappa_1^\dagger p}$ and $\kappa_2^\dagger p = \overline{\kappa_2^\dagger p}$) since all decisions decide themselves (*i.e.*, since $\langle\!\langle p\rangle\!\rangle = \langle p\rangle$).

Theorem 3.4 *There is a functor* $\mathscr{C}^{op} \xrightarrow{\text{Dec}} \mathbf{Set}$ *given by mapping objects to their decisions, and morphisms to decision transformers.*

Proof. Define this functor by $\text{Dec}(X) = \{\langle p\rangle \mid p \in \text{Hom}(X, X + X)\}$ on objects, and by $\text{Dec}(f : Y \to X)(\langle p\rangle) = \langle\!\langle p\rangle f\rangle$ on morphisms. This is contravariantly functorial since $\text{Dec}(\text{id}_X)(\langle p\rangle) = \langle\!\langle p\rangle\,\text{id}_X\rangle = \langle\!\langle p\rangle\!\rangle = \langle p\rangle$ by Lemma 3.2(i), and $\text{Dec}(gf)(\langle p\rangle) = \langle\!\langle p\rangle gf\rangle = \langle\!\langle\!\langle p\rangle g\rangle f\rangle = \text{Dec}(f)(\text{Dec}(g)(\langle p\rangle))$ by Lemma 3.2(vi) and definition of $\text{Dec}(f)$, as desired. $\square$

From now on, we will use the notation $\text{Dec}(Y) \xrightarrow{f^\diamond} \text{Dec}(X)$ for the decision transformation $\text{Dec}(f)$.

This is an example of a fibred category, which have historically been important in categorical presentations of logic, *e.g.*, in topoi (see [15] for a thorough treatment of indexed and fibred categories in categorical logic). In Section 4, we will see that this indexed category extends beyond **Set** to a model of $\mathbf{K}_3^{\text{w}}$. For now, it is sufficient to show the equivalence between conditions (morphisms $X \to 1 + 1$) and decisions (morphisms $X \to X + X$ satisfying the *decision laws* of Definition 2.2).

Theorem 3.5 (Condition/decision duality) *Decisions and predicates are naturally isomorphic in any extensive restriction category with a restriction terminal object:* $\mathrm{Dec}(-) \cong \mathrm{Hom}(-, 1+1)$.

Proof. Let $\mathscr{C}$ be an extensive restriction category with a restriction terminal object, and X some object of $\mathscr{C}$; we begin by showing that the mappings

$$X \xrightarrow{\langle f \rangle} X + X \quad \mapsto \quad X \xrightarrow{\langle f \rangle} X + X \xrightarrow{!+!} 1+1 \quad \text{and} \quad X \xrightarrow{p} 1+1 \quad \mapsto \quad X \xrightarrow{\langle p \rangle} X + X$$

between $\mathrm{Dec}(X)$ and $\mathrm{Hom}(X, 1+1)$ yields a bijection. In other words, we must show that $\langle\!\langle (!+!)\langle f \rangle \rangle\!\rangle = \langle f \rangle$ and $p = (!+!)\langle p \rangle$. To show $\langle\!\langle (!+!)\langle f \rangle \rangle\!\rangle = \langle f \rangle$, we show that $\langle f \rangle$ decides $(!+!)\langle f \rangle$ by $\nabla \langle f \rangle = \overline{f} = \overline{\langle f \rangle} = \overline{(!+!)\langle f \rangle}$ using the fact that the unique map $X \xrightarrow{!} 1$ is total by 1 restriction terminal, and by $(((!+!)\langle f \rangle) + ((!+!)\langle f \rangle))\langle f \rangle = ((!+!) + (!+!))(\langle f \rangle + \langle f \rangle)\langle f \rangle = ((!+!) + (!+!))(\langle f \rangle + \langle f \rangle)\langle\!\langle f \rangle\!\rangle = ((!+!) + (!+!))(\kappa_1 + \kappa_2)\langle f \rangle = (\kappa_1 + \kappa_2)(!+!)\langle f \rangle$. Thus $\langle\!\langle (!+!)\langle f \rangle \rangle\!\rangle = \langle f \rangle$, as desired.

To show that $p = (!+!)\langle p \rangle$ for $X \xrightarrow{p} 1 + 1$ we show something slightly more general, namely that $(!+!)f = (!+!)\langle f \rangle$ for any $X \xrightarrow{f} Y + Z$. That $p = (!+!)\langle p \rangle$ then follows as a special case since $\mathrm{id}_{1+1} = (!+!)$ by 1 terminal, so $p = \mathrm{id}_{1+1}\, p = (!+!)p$. This slightly more general statement follows by commutativity of the diagram below.

$$
\begin{array}{ccc}
X & \xrightarrow{\langle f \rangle} & X + X \\
{\scriptstyle f}\downarrow \quad {\scriptstyle (i)} & & \downarrow {\scriptstyle f+f} \quad {\scriptstyle (iii)} \quad \searrow^{!+!} \\
Y + Z & \xrightarrow[\kappa_1 + \kappa_2]{} & (Y+Z) + (Y+Z) \\
& \searrow {\scriptstyle (ii)} \quad \searrow^{!+!} & \\
& & 1+1 \\
& \searrow_{!+!} &
\end{array}
$$

Here, *(i)* commutes by the second axiom of decisions, while *(ii)* and *(iii)* both commute by 1 terminal.

To see that this bijection extends to a natural isomorphism, we must fix some $Y \xrightarrow{f} X$ and chase the diagram

$$
\begin{array}{ccc}
\mathrm{Dec}(X) & \xleftarrow{\;\cong\;} & \mathrm{Hom}(X, 1+1) \\
{\scriptstyle f^{\diamond}}\downarrow & & \downarrow {\scriptstyle f^{*}} \\
\mathrm{Dec}(Y) & \xrightarrow[\cong]{} & \mathrm{Hom}(Y, 1+1)
\end{array}
$$

where we use $f^{\diamond}$ to denote the functorial action $\mathrm{Dec}(f)$, $\mathrm{Dec}(f)(\langle p \rangle) = \langle\!\langle p \rangle f \rangle$. Picking some $\langle g \rangle \in \mathrm{Dec}(X)$ we must have $(!+!)\langle\!\langle g \rangle f \rangle = (!+!)\langle g \rangle f$, which indeed follows by the statement above. On the other hand, picking some $p \in \mathrm{Hom}(X, 1+1)$, chasing yields that we must have $\langle\!\langle p \rangle f \rangle = \langle pf \rangle$, which follows directly by Lemma 3.2 (vi). $\square$

A consequence of this equivalence in extensive restriction categories is that decisions are a property of the *predicates* rather than a property of arbitrary maps, as it is commonly presented. This is shown in the following corollary to Theorem 3.5.

Corollary 3.6 *A restriction category with restriction coproducts, a restriction zero, and a restriction terminal object has all decisions (*i.e.*, is extensive as a restriction category) iff it has all decisions of predicates.*

Proof. It follows directly that having decisions for all morphisms implies having decisions for all predicates. On the other hand, suppose that the category only has decisions for predicates, and let $X \xrightarrow{f} Y + Z$ be an arbitrary morphism. But then, by the proof of Theorem 3.5, the decision for the predicate $X \xrightarrow{f} Y + Z \xrightarrow{!+!} 1 + 1$ decides $X \xrightarrow{f} Y + Z$ (by $\langle (!+!)f \rangle = \langle (!+!) \langle f \rangle \rangle = \langle f \rangle$), and we are done. □

4   The internal logic of extensive restriction categories

Having established the natural isomorphism of decisions and predicates (with their respective transformers) which forms the condition/decision duality at the categorical level, we now turn to their structure. The main result of this section, Theorem 4.7, shows that the decisions $\mathrm{Dec}(X)$ on an object X form a model of $\mathbf{K}_3^{\mathrm{w}}$, and that decision transformers $\mathrm{Dec}(Y) \xrightarrow{f^{\diamond}} \mathrm{Dec}(X)$ are homomorphisms of these models. We first recall Kleene's three valued logics, in particular $\mathbf{K}_3^{\mathrm{w}}$ and its algebraic counterpart of De Morgan quasilattices.

4.1   Kleene's three valued logics and De Morgan quasilattices

Kleene's three valued logics of $\mathbf{K}_3$ (*strong Kleene logic*) and $\mathbf{K}_3^{\mathrm{w}}$ (*weak Kleene logic*), both introduced in [21], are logics based on *partial predicates* with a computational interpretation: Predicates are conceived of as programs which *may not terminate*, but if they do, they terminate with a Boolean truth value as output. In this way, both $\mathbf{K}_3$ and $\mathbf{K}_3^{\mathrm{w}}$ can be thought of as partial versions of classical logic. Here, possible nontermination is handled analogously to how it is handled in domain theory, *i.e.*, by the introduction of a third truth value in addition to truth t and falsehood f, denoted u in Kleene's presentation [21], which should be read as "undefined".

The difference between $\mathbf{K}_3$ and $\mathbf{K}_3^{\mathrm{w}}$ lies in how they cope with undefined truth values. In $\mathbf{K}_3^{\mathrm{w}}$ (see Figure 3), undefinedness is "contagious": if any part of an expression is undefined, the truth value of the entire expression is undefined as well [4]. This fits well into a computation paradigm with possible nontermination and only sequential composition available. In contrast, the semantics of $\mathbf{K}_3$ is to try to recover definite truth values whenever possible, even if part of the computation fails to terminate. For example, in $\mathbf{K}_3$ (and unlike $\mathbf{K}_3^{\mathrm{w}}$), $p \wedge q$ is considered false if

---

[4] This contagious behaviour has also been used to explain other phenomena. In philosophy, $\mathbf{K}_3^{\mathrm{w}}$ is better known as $\mathbf{B}_3$ or *Bochvar's nonsense logic* (see, *e.g.*, [12]), and the third truth value read as "meaningless" or "nonsensical" rather than "undefined". The central idea is that nonsense is contagious: *e.g.*, "2 + 2 = 5 and gobbledygook" is nonsensical even if part of it can be assigned meaning.

Q \ P ∧	t	f	u
t	t	f	u
f	f	f	u
u	u	u	u

(a) Weak conjunction.

Q \ P ∨	t	f	u
t	t	t	u
f	t	f	u
u	u	u	u

(b) Weak disjunction.

P	t	f	u
$\neg P$	f	t	u

(c) Negation.

Fig. 3. The three-valued semantics of $\mathbf{K}_3^{\mathrm{w}}$.

one of p and q is false, even if the other is undefined. While this allows for some recovery in the face of nontermination, computationally it seems to require parallel processing capabilities.

Like classical logic takes its algebraic semantics in Boolean algebras, the corresponding algebraic structure for $\mathbf{K}_3^{\mathrm{w}}$ is that of *De Morgan quasilattices* (see, e.g., [12]). As is sometimes done, we assume these to be distributive; *i.e.*, what we call De Morgan quasilattices are sometimes called *distributive De Morgan quasilattices* or even *(distributive) De Morgan bisemilattices* (see, e.g., [23]). Note that we generally do *not* require these to be bounded, *i.e.*, for top and bottom elements $\top$ and $\bot$ to exist.

Definition 4.1 A *De Morgan quasilattice* (in its algebraic formulation) is a quadruple $\mathfrak{A} = (|\mathfrak{A}|, \neg, \wedge, \vee)$ satisfying the following equations, for all $p, q, r \in |\mathfrak{A}|$:

(i) $p \wedge p = p$,

(ii) $p \vee p = p$,

(iii) $p \wedge q = q \wedge p$,

(iv) $p \vee q = q \vee p$,

(v) $p \wedge (q \wedge r) = (p \wedge q) \wedge r$,

(vi) $p \vee (q \vee r) = (p \vee q) \vee r$,

(vii) $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$,

(viii) $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$,

(ix) $\neg\neg p = p$,

(x) $\neg(p \wedge q) = \neg p \vee \neg q$,

(xi) $\neg(p \vee q) = \neg p \wedge \neg q$,

Further, a De Morgan quasilattice $\mathfrak{A}$ is said to be *bounded* if there exist elements $\bot, \top \in |\mathfrak{A}|$ such that the following are satisfied (for all $p \in |\mathfrak{A}|$):

(xii) $p \wedge \top = p$, and

(xiii) $p \vee \bot = p$.

A homomorphism $\mathfrak{A} \xrightarrow{h} \mathfrak{B}$ of De Morgan quasilattices is a function $|\mathfrak{A}| \to |\mathfrak{B}|$ which preserves $\neg$, $\wedge$, and $\vee$. A homomorphism of bounded De Morgan quasilattices is one which additionally preserves $\top$ and $\bot$.

Being a De Morgan quasilattice is a strictly weaker property than being a Boolean algebra. In particular, a Boolean algebra is a bounded De Morgan quasilattice which further satisfies the *absorption* laws $p = p \wedge (p \vee q)$ and $p = p \vee (p \wedge q)$, and the laws of contradiction and *tertium non datur*, $p \wedge \neg p = \bot$ and $p \vee \neg p = \top$.

De Morgan quasilattices and their homomorphisms form a category which we call **DMQLat**. As for Boolean algebras, one can derive a partial order on De Morgan quasilattices by $p \preccurlyeq q$ iff $p \wedge q = p$, and another one by $p \sqsubseteq q$ iff $p \vee q = q$.

Unlike as for Boolean algebras, however, these do *not* coincide, though they are anti-isomorphic, as it follows from the De Morgan laws that $p \preccurlyeq q$ iff $\neg q \sqsubseteq \neg p$. We will return to these in Section 4 and argue why $\cdot \preccurlyeq \cdot$ is the one more suitable as the entailment relation for $\mathbf{K}_3^w$.

4.2   The internal logic

With $\mathbf{K}_3^w$ and De Morgan quasilattices introduced, we return to the construction of the internal logic. To aid in its presentation (and subsequent proofs), we start by introducing a graphical language of extensive restriction categories, based on the one for cocartesian categories (see, *e.g.*, [25]). Then, we show how the constants and connectives of $\mathbf{K}_3^w$ can be interpreted (Definition 4.2) as decisions on an object (Lemma 4.3). Finally, we show that decisions on an object form a model of $\mathbf{K}_3^w$ (Lemma 4.5), and that decision transformations are homomorphisms of these models (Lemma 4.6), concluding this construction. We go on to explore an important corollary to this construction, namely that if we restrict ourselves from ordinary decisions to total decisions and total decision transformations, we obtain a fibration over Boolean algebras instead (Corollary 4.10 Theorem 4.11). The latter is a well-known property of extensive categories first shown in [6], though this proof uses entirely different machinery.

Figure 4 shows the graphical language of extensive restriction categories, which has the restriction coproduct as its monoidal tensor. The first five gadgets are from cocartesian categories ($\gamma_{X,Y}$ is here the twist map, $[\kappa_2, \kappa_1]$). We add gadgets corresponding to decisions $X \xrightarrow{\langle f \rangle} X + X$, inverses to decisions $X + X \xrightarrow{\langle f \rangle^\dagger} X$ (as all decisions are partial isomorphisms, see Lemma 3.2(ii)), and restriction idempotents $X \xrightarrow{\overline{f}} X$. The gadget for inverses to decisions was inspired by *assertions* in reversible flowcharts (see [29]). Useful derived gadgets include

Just as the graphical language of cocartesian categories, isomorphism or isotopy of diagrams is not enough for coherence – equations only hold in the graphical language up to diagrammatic manipulations corresponding to the decision laws, as well as the diagrammatic manipulations for coproducts (*e.g.*, the commutative monoid axioms and naturality for the codiagonal, the zero morphism laws, etc.). For more on the latter, see [25]. For example, graphically, the decision laws are

As in the example above, when the signature is clear from the context, we omit the object annotations (*e.g.*, X, Y, Z in Figure 4).

With the graphical language in place, we proceed to give the definition of the

$$f + g \qquad [f, g] \qquad \gamma_{X,Y} \qquad 0_{0,X} \quad 0_{X,0} \qquad \langle f \rangle \qquad \langle f \rangle^\dagger \qquad \overline{f}$$
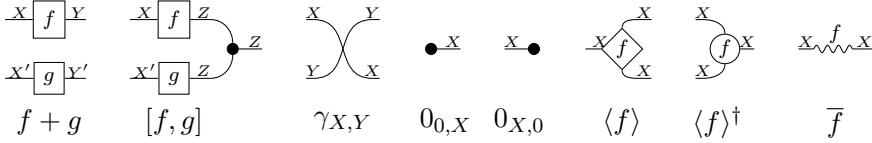
Fig. 4. An overview of the gadgets that make up the graphical language of extensive restriction categories.

internal logic of decisions in an extensive restriction category, *i.e.*, the entailment relation and construction of constants and propositional connectives.

Definition 4.2 In an extensive restriction category, propositional constants and connectives are defined for decisions as follows, using the graphical language:
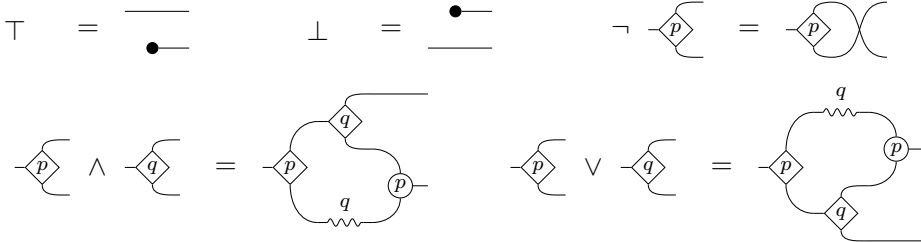
Entailment is defined by $\langle p \rangle \vDash \langle q \rangle$ iff $\langle p \rangle \preccurlyeq \langle q \rangle$ (explicitly, iff $\langle p \rangle \wedge \langle q \rangle = \langle p \rangle$).

For those more textually inclined, this defines $\top = \kappa_1$, $\bot = \kappa_2$, $\neg \langle p \rangle = \gamma \langle p \rangle$, $\langle p \rangle \vee \langle q \rangle = (\langle p \rangle^\dagger + \mathrm{id}) \alpha (\overline{\langle q \rangle} + \langle q \rangle) \langle p \rangle$, and $\langle p \rangle \wedge \langle q \rangle = (\mathrm{id} + \langle p \rangle^\dagger) \alpha (\langle q \rangle + \overline{\langle q \rangle}) \langle p \rangle$.

Intuitively, we think of decisions as representing partial predicates by separating values into *witnesses* and *counterexamples* of that partial predicate (see also [19]). The definitions of $\top$ and $\bot$ express the convention that the first component carries witnesses, while the second component carries counterexamples. Negation of partial predicates then amounts to swapping witnesses for counterexamples and vice versa, *i.e.*, by composing with the symmetry. The intuition behind conjunction (and, dually, disjunction) is less obvious: Using the intuition of decisions as morphisms that tag inputs with a branch but doesn't change it otherwise, we see that a witness of $\langle p \rangle \wedge \langle q \rangle$ has to be a witness of both $\langle p \rangle$ and $\langle q \rangle$, while a counterexample of $\langle p \rangle \wedge \langle q \rangle$ is either a counterexample of $\langle p \rangle$ which is further defined for $\langle q \rangle$ (necessary to ensure commutativity), or a witness of $\langle p \rangle$ which is a counterexample of $\langle q \rangle$. The case for disjunctions is dual.

Before we move on to show that this actually has the logical structure we're after, we first obliged to show that these connectives and constants actually define well-formed decisions. This fact is expressed in the following lemma.

Lemma 4.3 *The constants and connectives of Definition 4.2 are decisions.*

Proof. See appendix.   □

Before we can proceed, we need a small technical lemma.

Lemma 4.4 *Let* *and* *be decisions. It is the case that*

(i) (ii) (iii)
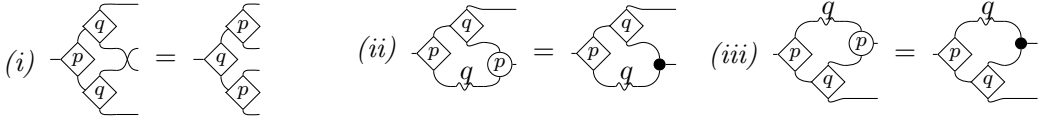
Proof. See appendix. □

The first part of this lemma can be seen as a form of commutativity for decisions – and, indeed, it performs most of the heavy lifting in showing commutativity of conjunction and disjunction. On the other hand, parts (ii) and (iii) shows that we could have defined conjunction and disjunction more simply in Definition 4.2. The reason why we chose the current definition is that it yields entirely *reversible* models (see also [19]), *i.e.*, involving only partial isomorphisms. We will discuss this property further in Section 5. For now, we continue with the internal logic.

Lemma 4.5 $\mathrm{Dec}(X)$ *is a bounded De Morgan quasilattice for any object* X.
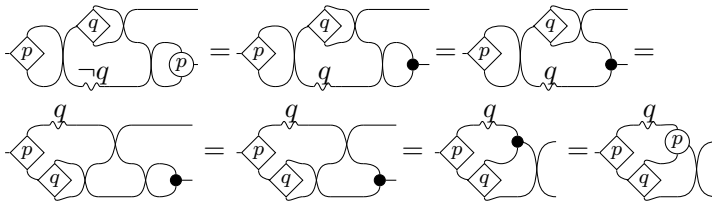
Proof. We show only a few of the cases here using the graphical language. See the appendix for the rest. Idempotence of conjunction, *i.e.*, $\langle p \rangle \wedge \langle p \rangle = \langle p \rangle$, follows by
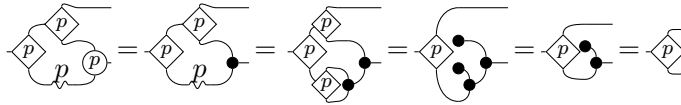
and similarly for disjunction. That $\langle p \rangle \wedge \top = \langle p \rangle$ is shown simply by
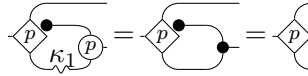
and again, the unit law for disjunction has an analogous proof. The first De Morgan law, that $\neg \langle p \rangle \wedge \neg \langle q \rangle = \neg(\langle p \rangle \vee \langle q \rangle)$

and the proof of the second De Morgan law follows similarly. □

As such, we have that each collection of decisions on an object form a local model of $\mathbf{K}_3^{\mathrm{w}}$, giving us the first part of the fibration. For the second, we need to show that decision transformers preserve entailment and the propositional connectives (though not necessarily the constants). This is shown in the following lemma.

Lemma 4.6 *Let* $X \xrightarrow{f} Y$. *Then* $\mathrm{Dec}(Y) \xrightarrow{f^\diamond} \mathrm{Dec}(X)$ *is a homomorphism of De Morgan quasilattices,* i.e.,

(i) $\langle p \rangle \models \langle q \rangle$ *implies* $f^\diamond(\langle p \rangle) \models f^\diamond(\langle q \rangle)$

(ii) $f^\diamond(\neg \langle p \rangle) = \neg f^\diamond(\langle p \rangle)$

(iii) $f^\diamond(\langle p \rangle \wedge \langle q \rangle) = f^\diamond(\langle p \rangle) \wedge f^\diamond(\langle q \rangle)$

(iv) $f^\diamond(\langle p \rangle \vee \langle q \rangle) = f^\diamond(\langle p \rangle) \vee f^\diamond(\langle q \rangle)$

In addition, if f is total then $f^\diamond$ is a homomorphism of bounded *De Morgan quasilattice; i.e., we also have $f^\diamond(\top) = \top$ and $f^\diamond(\bot) = \bot$.*

Proof. (i) follows by (iii) since $\langle p \rangle \models \langle q \rangle$ iff $\langle p \rangle \preccurlyeq \langle q \rangle$ iff $\langle p \rangle \wedge \langle q \rangle = \langle q \rangle$, which in turn implies that $f^\diamond(\langle p \rangle) \wedge f^\diamond(\langle q \rangle) = f^\diamond(\langle p \rangle \wedge \langle q \rangle) = f^\diamond(\langle p \rangle)$, so $f^\diamond(\langle p \rangle) \preccurlyeq f^\diamond(\langle q \rangle)$ as well, *i.e.*, $f^\diamond(\langle p \rangle) \models f^\diamond(\langle q \rangle)$.

For (ii), we compute $f^\diamond(\neg \langle p \rangle) = f^\diamond(\gamma \langle p \rangle) = f^\diamond(\langle \gamma p \rangle) = \langle\langle \gamma p \rangle f \rangle = \langle \gamma p f \rangle = \gamma \langle p f \rangle = \gamma \langle\langle p \rangle f \rangle = \neg f^\diamond(\langle p \rangle)$ (using Lemma 3.2).

(iii) follows by lengthy but straightforward computation(see appendix).

(iv) is analogous to the previous case. □

Notice the final part regarding preservation of units. Generally, $f^\diamond(\top) = \langle \top f \rangle = \langle \kappa_1 f \rangle$, so $\overline{f^\diamond(\top)} = \overline{\langle \kappa_1 f \rangle} = \overline{\kappa_1 f} = \overline{\overline{\kappa_1} f} = \overline{f}$, so if f is not total, $\langle \kappa_1 f \rangle \neq \kappa_1$ (instead $\langle \kappa_1 f \rangle = \kappa_1 \overline{f}$).

Putting the two lemmas together gives us the main result:

Theorem 4.7 *In every extensive restriction category $\mathscr{C}$, decisions over $\mathscr{C}$ form a fibred De Morgan quasilattice via the decision fibration.*

Proof. By Lemmas 4.5 and 4.6. □

We previously claimed that the conjunction order was the more suitable one for entailment in extensive restriction categories. We are finally ready to state why:

Lemma 4.8 *Entailment is upwards directed in truth and definedness: $\langle p \rangle \models \langle q \rangle$ iff $\kappa_1^\dagger \langle p \rangle \leq \kappa_1^\dagger \langle q \rangle$ and $\overline{\langle p \rangle} \leq \overline{\langle q \rangle}$.*

Proof. See appendix. □

In other words, $\langle p \rangle$ entails $\langle q \rangle$ iff $\langle q \rangle$ is both *at least as true* and *at least as defined* as $\langle p \rangle$ is. That is, entailment preserves not only *truth* (as we expect all entailments to) but also *information* (as we expect of orders on *partial* maps). Compare this to the disjunction partial order for which $\langle p \rangle \sqsubseteq \langle q \rangle$ instead states that $\langle q \rangle$ is less *false* and less *defined* than $\langle p \rangle$: In other words, it prefers for information to be *forgotten* rather than preserved.

We move on now to an important special case of the situation above, which is when only *total* decisions are considered rather than arbitrary ones. For this, we need a small lemma regarding the restriction idempotents of decisions when composed using the propositional connectives.

Lemma 4.9 *We state some facts about restriction idempotents of decisions:*

(i) $\overline{\neg \langle p \rangle} = \overline{\langle p \rangle}$,

(ii) $\overline{\langle p \rangle \wedge \langle q \rangle} = \overline{\langle p \rangle} \, \overline{\langle q \rangle}$,

(iii) $\overline{\langle p \rangle \vee \langle q \rangle} = \overline{\langle p \rangle} \, \overline{\langle q \rangle}$,

(iv) $\overline{\langle p \rangle \wedge \langle q \rangle} \leq \overline{\langle p \rangle}$ and $\overline{\langle p \rangle \wedge \langle q \rangle} \leq \overline{\langle q \rangle}$,

(v) $\overline{\langle p \rangle \vee \langle q \rangle} \leq \overline{\langle p \rangle}$ and $\overline{\langle p \rangle \vee \langle q \rangle} \leq \overline{\langle q \rangle}$.

Proof. See appendix. □

We can now show that total decisions form a fibred Boolean algebra.

Corollary 4.10 $\mathrm{TDec}(X)$ *is a Boolean algebra for any object X, and $f^{\triangle} : \mathrm{TDec}(Y) \to \mathrm{TDec}(X)$ is homomorphism of Boolean algebras for any total $X \xrightarrow{f} Y$.*

Proof. Since $\mathrm{Dec}(X)$ is a De Morgan quasilattice (Lemma 4.5), since total decisions are specifically decisions, and since the constants are total and the connectives preserve totality (Lemma 4.9), it suffices to show that when $\langle p \rangle$ and $\langle q \rangle$ are total they satisfy the *absorption* laws $\langle p \rangle = \langle p \rangle \wedge (\langle p \rangle \vee \langle q \rangle)$ and $\langle p \rangle = \langle p \rangle \vee (\langle p \rangle \wedge \langle q \rangle)$, and the laws of contradiction and *tertium non datur*, $\langle p \rangle \wedge \neg \langle p \rangle = \bot$ and $\langle p \rangle \vee \neg \langle p \rangle = \top$. The first absorption law follows by
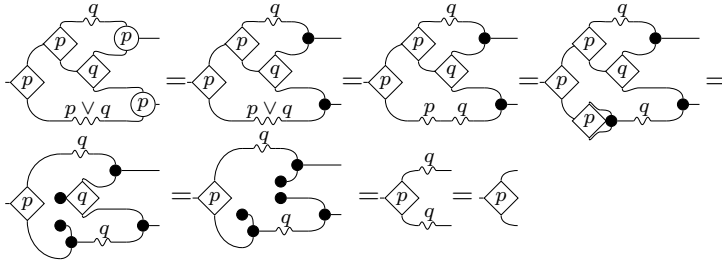
and the other follows analogously. Likewise, the law of contradiction can be shown as
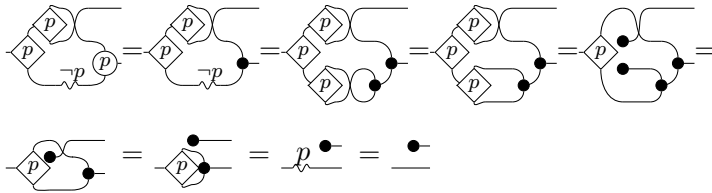
and similarly for *tertium non datur*. □

Using the previous corollary, it follows (see [6] for the original proofs from effectus theory) that predicates over an extensive category form a fibred Boolean algebra.

Theorem 4.11 *Predicates (or, equivalently, decisions) over an extensive category is a fibred Boolean algebra via the predicate fibration (or, equivalently, the decision fibration).*

Proof. Since total decisions on objects form Boolean algebras by Corollary 4.10, it suffices to show that every extensive category arises as the subcategory of total morphisms of an extensive restriction category.

Let $\mathscr{C}$ be an extensive category, and $\mathcal{M}$ denote the collection of all coproduct injections of $\mathscr{C}$. As remarked in [7], this is a stable system of monics, and by Example 4.17 of [8], $\mathrm{Par}(\mathscr{C}, \mathcal{M})$ is a classified restriction category under the $+1$ monad. Since $\mathscr{C}$ has coproducts and $\mathrm{Par}(\mathscr{C}, \mathcal{M})$ is classified, it follows by Proposition 2.3 of [9] that $\mathrm{Par}(\mathscr{C}, \mathcal{M})$ has restriction coproducts. That 0 is a restriction zero in $\mathrm{Par}(\mathscr{C}, \mathcal{M})$ follows straightforwardly, with the span $X \xleftarrow{!_X} 0 \xrightarrow{!_Y} Y$ as the unique zero morphism $X \xrightarrow{0_{X,Y}} Y$. As such, it suffices to show that decisions can be constructed in $\mathrm{Par}(\mathscr{C}, \mathcal{M})$. Let $X \xleftarrow{m} X' \xrightarrow{f} Y + Z$ be an arbitrary morphism of $\mathrm{Par}(\mathscr{C}, \mathcal{M})$. Since $\mathscr{C}$ is extensive it has pullbacks of coproduct injections along arbitrary morphisms, so the two squares

$$
\begin{array}{ccccc}
X_1 & \xrightarrow{m_1} & X' & \xleftarrow{m_2} & X_2 \\
{\scriptstyle f_1}\downarrow & & \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f_2} \\
Y & \xrightarrow{\kappa_1} & Y + Z & \xleftarrow{\kappa_2} & Z
\end{array}
$$

are pullbacks, and so the top row is a coproduct diagram (*i.e.*, $X_1 + X_2 \cong X'$). But then it readily follows that

$$
X \xleftarrow{m} X' \xleftarrow{\cong} X_1 + X_2 \xrightarrow{m_1 + m_2} X' + X' \xrightarrow{m+m} X + X
$$

is a decision for $X \xleftarrow{m} X' \xrightarrow{f} Y + Z$ in $\mathrm{Par}(\mathscr{C}, \mathcal{M})$, and we are done. $\square$

5   Conclusion and future work

Motivated by an observation from flowchart languages that predicates serve a dual role as both condition and decision, we have given an account of extensive restriction categories (due to [7–9]) as categories with an internal logic (namely $\mathbf{K}_3^{\mathrm{w}}$) that internalize this duality, in the form of a natural isomorphism between the predicate fibration and the decision fibration.

We have also extended the graphical language of cocartesian categories to one for extensive restriction categories, and used our results to give an alternative proof of the fact that extensive categories, too, are categories with an internal logic – classical logic. While the graphical language has proven itself useful in proving theorems, it does have its shortcomings. For example, the only way to express restriction idempotents of compositions, such as $\overline{gf}$, is, awkwardly, as $\overset{gf}{\sim\!\!\sim\!\!\sim}$. That is, we would want only one representation of composition as placing gadgets in sequence, but since $\overline{gf}$ cannot generally be expressed as a composite involving only smaller things (*e.g.*, $\overline{f}$ and $\overline{g}$), we are forced in this case to let the textual representation (*i.e.*, juxtaposition) bleed into the graphical language. The graphical notation for decisions has similar issues.

An application of the developed theory is in reversible models of logics, which was also the motivation for defining the connectives in slightly more involved fash-

ion, using partial inverses to decisions rather than the codiagonal. Indeed, the inspiration for using decisions as predicates came from the study of the categorical semantics of reversible flowchart languages (see [13,19]). Since a decision in $\mathscr{C}$ is still a decision in $\text{Inv}(\mathscr{C})$ (see [19]), $\text{Dec}(X)$ is still a De Morgan quasilattice in $\text{Inv}(\mathscr{C})$, though the homomorphisms between fibres differ (*i.e.*, only decision transformers that are partial isomorphisms occur in the decision fibration on $\text{Inv}(\mathscr{C})$).

We have only considered the weak Kleene logic $\mathbf{K}_3^{\mathrm{w}}$ here, as it can be constructed by purely sequential means. However, we conjecture that the strong Kleene logic $\mathbf{K}_3$ can be modelled as well in extensive restriction categories if additionally a *parallel* composition operator such as finite joins (see [14]) is available. Finally, just the propositional fragment of $\mathbf{K}_3^{\mathrm{w}}$ and classical logic has been considered in this paper. Though decisions on an object yields a fibred category with a logical structure, we have not explored extensions to models of first-order logics, *e.g.*, by investigating the feasibility of adjoints to substitution, as in the standard trick due to Lawvere [22] (see also [15]).

References

[1] P. Ammann and J. Offutt. *Introduction to Software Testing*. Cambridge University Press, 1st edition, 2008.

[2] M. A. Arbib and E. G. Manes. Partially additive categories and flow-diagram semantics. *Journal of Algebra*, 62(1):203 – 227, 1980.

[3] E. Ashcroft and Z. Manna. The translation of 'go to' programs into 'while' programs. In C. V. Freiman, J. E. Griffith, and J. L. Rosenfeld, editors, *Proceedings of IFIP Congress 71*, volume 1, pages 250–255. North-Holland, 1972.

[4] C. Böhm and G. Jacopini. Flow diagrams, Turing machines and languages with only two formation rules. *Communications of the ACM*, 9(5):366–371, 1966.

[5] A. Carboni, S. Lack, and R. Walters. Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra*, 84(2):145–158, 1993.

[6] K. Cho, B. Jacobs, B. Westerbaan, and A. Westerbaan. An introduction to effectus theory. See http://arxiv.org/abs/1512.05813, 2015.

[7] J. R. B. Cockett and S. Lack. Restriction categories I: Categories of partial maps. *Theoretical Computer Science*, 270(1–2):223–259, 2002.

[8] J. R. B. Cockett and S. Lack. Restriction categories II: Partial map classification. *Theoretical Computer Science*, 294(1–2):61–102, 2003.

[9] R. Cockett and S. Lack. Restriction categories III: Colimits, partial limits and extensivity. *Mathematical Structures in Computer Science*, 17(4):775–817, 2007.

[10] C. C. Elgot. Monadic computation and iterative algebraic theories. In H. E. Rose and J. C. Shepherdson, editors, *Logic Colloquium*, pages 175–230. North Holland, 1975.

[11] C. C. Elgot. Structured programming with and without GO TO statements. *IEEE Transactions on Software Engineering*, SE-2:41–53, 1976.

[12] V. K. Finn and R. Grigolia. Nonsense logics and their algebraic properties. *Theoria*, 59(1–3):207–273, 1993.

[13] R. Glück and R. Kaarsgaard. A categorical foundation for structured reversible flowchart languages. In A. Silva, editor, *The Thirty-third Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIII)*, volume 336 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2017.

[14] X. Guo. *Products, Joins, Meets, and Ranges in Restriction Categories*. PhD thesis, University of Calgary, 2012.

[15] B. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, first edition, 1999.

[16] B. Jacobs. New directions in categorical logic, for classical, probabilistic and quantum logic. *Logical Methods in Computer Science*, 11(3), 2015.

[17] N. D. Jones, C. K. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall International, 1993.

[18] R. Kaarsgaard, H. B. Axelsen, and R. Glück. Join inverse categories and reversible recursion. *Journal of Logical and Algebraic Methods in Programming*, 87:33–50, 2017.

[19] R. Kaarsgaard and R. Glück. A categorical foundation for structured reversible flowchart languages: Soundness and adequacy. *Logical Methods in Computer Science*, 14(3):1–38, 2018.

[20] J. Kastl. Inverse categories. In H.-J. Hoehnke, editor, *Algebraische Modelle, Kategorien und Gruppoide*, volume 7 of *Studien zur Algebra und ihre Anwendungen*, pages 51–60. Akademie-Verlag, 1979.

[21] S. C. Kleene. *Introduction to metamathematics*. North Holland, 1st edition, 1952.

[22] F. W. Lawvere. Adjointness in foundations. *Dialectica*, 23:281–296, 1969.

[23] A. Ledda. Stone-type representations and dualities for varieties of bisemilattices. *Studia Logica*, 106(2):417–448, 2018.

[24] E. G. Manes and M. A. Arbib. *Algebraic approaches to program semantics*. Springer, 1986.

[25] P. Selinger. A survey of graphical languages for monoidal categories. In B. Coecke, editor, *New Structures for Physics*, pages 289–355. Springer, 2011.

[26] G. Ştefănescu. An algebraic theory of flowchart schemes. In P. Franchi-Zannettacci, editor, *CAAP '86*, volume 214 of *Lecture Notes in Computer Science*, pages 60–73. Springer, 1986.

[27] G. Ştefănescu. On flowchart theories part I: The deterministic case. *Journal of Computer and System Sciences*, 35(2):163 – 191, 1987.

[28] M. Williams and H. Ossher. Conversion of unstructured flow diagrams into structured form. *The Computer Journal*, 21(2):161–167, 1978.

[29] T. Yokoyama, H. B. Axelsen, and R. Glück. Fundamentals of reversible flowchart languages. *Theoretical Computer Science*, 611:87–115, 2016.

A   Omitted proofs

Proof. [Proof of Lemma 3.2] For (i) and (ii), see [9]. (iii) follows by (ii) since

$$\overline{\langle f \rangle^{\dagger}} = \overline{[\kappa_1^{\dagger} f, \kappa_2^{\dagger} f]} = \overline{\overline{\kappa_1^{\dagger} f} + \overline{\kappa_2^{\dagger} f}} = \overline{\kappa_1^{\dagger} f} + \overline{\kappa_2^{\dagger} f}.$$

For (iv), $\overline{\langle f \rangle} = \overline{\langle f \rangle} = \overline{\overline{\nabla} \langle f \rangle} = \overline{\nabla \langle f \rangle} = \overline{\overline{f}} = \overline{f}$. To show (v) we show that $\gamma \langle f \rangle$ decides γf, since $\nabla \gamma \langle f \rangle = \nabla \langle f \rangle = \overline{f} = \overline{\overline{f}} = \overline{\overline{\gamma} f} = \overline{\gamma f}$ and

$$((\gamma f) + (\gamma f)) \gamma \langle f \rangle = \gamma ((\gamma f) + (\gamma f)) \langle f \rangle = \gamma (\gamma + \gamma)(f + f) \langle f \rangle = \gamma (\gamma + \gamma)(\kappa_1 + \kappa_2) p$$
$$= \gamma ((\gamma \kappa_1) + (\gamma \kappa_2)) p = \gamma (\kappa_2 + \kappa_1) p = (\kappa_1 + \kappa_2) \gamma p$$

which was what we wanted. We show (vi) analogously by showing that $\langle \langle f \rangle g \rangle$ decides $\langle fg \rangle$ since $\nabla \langle \langle f \rangle g \rangle = \overline{\langle f \rangle g} = \overline{\overline{\nabla} \langle f \rangle g} = \overline{\nabla \langle f \rangle g} = \overline{\overline{f} g} = \overline{fg}$ and

$$((fg) + (fg)) \langle \langle f \rangle g \rangle = ((f \overline{f} g) + (f \overline{f} g)) \langle \langle f \rangle g \rangle = ((f \nabla \langle f \rangle g) + (f \nabla \langle f \rangle g)) \langle \langle f \rangle g \rangle$$
$$= ((f \nabla) + (f \nabla))((\langle f \rangle g) + (\langle f \rangle g)) \langle \langle f \rangle g \rangle$$
$$= ((f \nabla) + (f \nabla))(\kappa_1 + \kappa_2) \langle f \rangle g = ((f \nabla \kappa_1) + (f \nabla \kappa_2)) \langle f \rangle g$$
$$= (f + f) \langle f \rangle g = (\kappa_1 + \kappa_2) fg \ .$$

For (vii), we observe that $\overline{(\overline{e}+\overline{e})\langle f\rangle} = \overline{\nabla(\overline{e}+\overline{e})\langle f\rangle} = \overline{\overline{e}\nabla\langle f\rangle} = \overline{\overline{e}f} = \overline{e}\overline{f} = \overline{f}\overline{e} = \overline{\langle f\rangle\overline{e}}$ so $(\overline{e}+\overline{e})\langle f\rangle = (\overline{e}+\overline{e})\langle f\rangle\overline{(\overline{e}+\overline{e})\langle f\rangle} = (\overline{e}+\overline{e})\langle f\rangle\overline{\langle f\rangle\overline{e}} = (\overline{e}+\overline{e})\langle f\rangle\overline{e}$. To show (viii), we show that the two morphism decide one another. We see that $\langle f\rangle\overline{e}$ decides $(\overline{e}+\overline{e})\langle f\rangle$ since $\nabla\langle f\rangle\overline{e} = \overline{f}\overline{e} = \overline{\langle f\rangle\overline{e}} = (\overline{e}+\overline{e})\langle f\rangle$ (see (vii) above) and

$$(((\overline{e}+\overline{e})\langle f\rangle) + ((\overline{e}+\overline{e})\langle f\rangle))\langle f\rangle\overline{e} = ((\overline{e}+\overline{e}) + (\overline{e}+\overline{e}))(\langle f\rangle + \langle f\rangle)\langle f\rangle\overline{e}$$
$$= ((\overline{e}+\overline{e}) + (\overline{e}+\overline{e}))(\langle f\rangle + \langle f\rangle)\langle\!\langle f\rangle\!\rangle\overline{e} = ((\overline{e}+\overline{e}) + (\overline{e}+\overline{e}))(\kappa_1 + \kappa_2)\langle f\rangle\overline{e}$$
$$= (\kappa_1 + \kappa_2)(\overline{e}+\overline{e})\langle f\rangle\overline{e} = (\kappa_1 + \kappa_2)(\overline{e}+\overline{e})\langle f\rangle$$

where $(\overline{e}+\overline{e})\langle f\rangle = (\overline{e}+\overline{e})\langle f\rangle\overline{e}$ by (vii); thus $\langle f\rangle\overline{e}$ decides $(\overline{e}+\overline{e})\langle f\rangle$, i.e., $\langle\!\langle(\overline{e}+\overline{e})\langle f\rangle\rangle\!\rangle = \langle f\rangle\overline{e} = \langle\!\langle\langle f\rangle\overline{e}\rangle\!\rangle$ (the latter by (i)). In the other direction, $\nabla(\overline{e}+\overline{e})\langle f\rangle = \overline{e}\nabla\langle f\rangle = \overline{e}\overline{f} = \overline{f}\overline{e} = \overline{\langle f\rangle\overline{e}}$ and it is the case that

$$(((\langle f\rangle\overline{e}) + (\langle f\rangle\overline{e}))(\overline{e}+\overline{e})\langle f\rangle = ((\langle f\rangle\overline{e}) + (\langle f\rangle\overline{e}))(\overline{e}+\overline{e})\langle f\rangle\overline{e}$$
$$= (((\langle f\rangle\overline{ee}) + (\langle f\rangle\overline{ee}))\langle f\rangle\overline{e} = (((\langle f\rangle\overline{e}) + (\langle f\rangle\overline{e}))\langle f\rangle\overline{e}$$
$$= ((\langle f\rangle\overline{e}) + (\langle f\rangle\overline{e}))\langle\!\langle f\rangle\overline{e} = (\kappa_1 + \kappa_2)\langle f\rangle\overline{e}$$

For (ix) we have that $\langle f\rangle\overline{e} = \langle\!\langle f\rangle\overline{e}\rangle$ by (viii) and get $\langle f\rangle\overline{e} = \langle\!\langle f\rangle\overline{e}\rangle = \langle f\overline{e}\rangle$ by (vi).

For (x), by (ii) $\langle f\rangle^\dagger = \left[\overline{\kappa_1^\dagger f}, \overline{\kappa_2^\dagger f}\right]$ so $\kappa_i^\dagger\langle f\rangle = \kappa_i^\dagger\left[\overline{\kappa_1^\dagger f}, \overline{\kappa_2^\dagger f}\right]^\dagger = \left(\left[\overline{\kappa_1^\dagger f}, \overline{\kappa_2^\dagger f}\right]\kappa_i\right)^\dagger = \overline{\kappa_i^\dagger f}^\dagger = \overline{\kappa_i^\dagger f}$.

For (xi), we compute

$$\langle g\rangle f = (\kappa_1^\dagger + \kappa_2^\dagger)(\kappa_1 + \kappa_2)\langle g\rangle f = (\kappa_1^\dagger + \kappa_2^\dagger)((\langle g\rangle f) + (\langle g\rangle f))\langle\!\langle g\rangle f$$
$$= (\kappa_1^\dagger + \kappa_2^\dagger)((\langle g\rangle f) + (\langle g\rangle f))\langle gf\rangle = (\kappa_1^\dagger + \kappa_2^\dagger)(\langle g\rangle + \langle g\rangle)(f + f)\langle gf\rangle$$
$$= ((\kappa_1^\dagger\langle g\rangle) + (\kappa_2^\dagger\langle g\rangle))(f + f)\langle gf\rangle = (\overline{\kappa_1^\dagger g} + \overline{\kappa_2^\dagger g})(f + f)\langle gf\rangle$$
$$= (f + f)(\overline{\kappa_1^\dagger gf} + \overline{\kappa_2^\dagger gf})\langle gf\rangle = (f + f)[\overline{\kappa_1^\dagger gf}, \overline{\kappa_2^\dagger gf}]\langle gf\rangle = (f + f)\overline{\langle gf\rangle^\dagger}\langle gf\rangle$$
$$= (f + f)\langle gf\rangle$$

where we use that $\kappa_i^\dagger\langle g\rangle = \overline{\kappa_i^\dagger g}$ by (xi), and $\langle gf\rangle^\dagger = [\overline{\kappa_1^\dagger gf}, \overline{\kappa_2^\dagger gf}]$ by (ii). $\qquad\square$

Proof. [Proof of Lemma 4.3] That $\top = \kappa_1$ and $\bot = \kappa_2$ are decisions is shown in [9]. That $\neg\langle p\rangle = \gamma\langle p\rangle$ is a decision follows by $\gamma\langle p\rangle = \langle\gamma p\rangle$ by Lemma 3.2(v). To see that $\langle p\rangle \wedge \langle q\rangle$ is a decision, it suffices by Lemma 3.3 to show that $\kappa_1^\dagger(\langle p\rangle \wedge \langle q\rangle) =$

$\overline{\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle)}$ and $\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle) = \overline{\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle)}$. We compute

$$
\begin{aligned}
\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle) &= \kappa_1^\dagger(\mathrm{id} + \langle p \rangle^\dagger)\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \mathrm{id}\,\kappa_1^\dagger\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \kappa_1^\dagger\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \kappa_1^\dagger\kappa_1^\dagger(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \kappa_1^\dagger\langle q \rangle\kappa_1^\dagger\langle p \rangle \\
&= \overline{\kappa_1^\dagger q}\,\overline{\kappa_1^\dagger p}
\end{aligned}
$$

so $\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle) = \overline{\kappa_1^\dagger q}\,\overline{\kappa_1^\dagger p} = \overline{\overline{\kappa_1^\dagger q}\,\overline{\kappa_1^\dagger p}} = \overline{\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle)}$. Further

$$
\begin{aligned}
\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle) &= \kappa_2^\dagger(\mathrm{id} + \langle p \rangle^\dagger)\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \langle p \rangle^\dagger\kappa_2^\dagger\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \langle p \rangle^\dagger(\kappa_2^\dagger + \mathrm{id})(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle \\
&= \langle p \rangle^\dagger((\kappa_2^\dagger\langle q \rangle) + \overline{\langle q \rangle}))\langle p \rangle \\
&= \langle p \rangle^\dagger(\overline{\kappa_2^\dagger q} + \overline{q}))\langle p \rangle \\
&\leq \langle p \rangle^\dagger(\overline{q} + \overline{q}))\langle p \rangle \\
&= \langle p \rangle^\dagger\langle p \rangle\overline{q} \\
&= \overline{\langle p \rangle}\overline{q} \\
&= \overline{p}\,\overline{q}
\end{aligned}
$$

so since $\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle) \leq \overline{p}\,\overline{q}$ it follows that $\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle) = \overline{\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle)}$, and so finally $\langle p \rangle \wedge \langle q \rangle$ is a decision by Lemma 3.3. The case for $\langle p \rangle \vee \langle q \rangle$ is entirely analogous. $\square$

Proof. [Proof of Lemma 4.6(iii)] By Lemma 3.3 it suffices to show that $\kappa_1^\dagger f^\diamond(\langle p \rangle \wedge \langle q \rangle) = \kappa_1^\dagger(f^\diamond(\langle p \rangle) \wedge f^\diamond(\langle q \rangle))$ and $\kappa_2^\dagger f^\diamond(\langle p \rangle \wedge \langle q \rangle) = \kappa_2^\dagger(f^\diamond(\langle p \rangle) \wedge f^\diamond(\langle q \rangle))$, Firstly we expand $f^\diamond(\langle p \rangle \wedge \langle q \rangle) = \langle(\langle p \rangle \wedge \langle q \rangle)f\rangle$ and $f^\diamond(\langle p \rangle) \wedge f^\diamond(\langle q \rangle) = \langle\!\langle p \rangle f\rangle \wedge \langle\!\langle q \rangle f\rangle = \langle pf\rangle \wedge \langle qf\rangle$. Then we compute

$$
\begin{aligned}
\kappa_1^\dagger(f^\diamond(\langle p \rangle) \wedge f^\diamond(\langle q \rangle)) &= \kappa_1^\dagger\langle(\langle p \rangle \wedge \langle q \rangle)f\rangle = \overline{\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle)f} \\
&= \overline{\kappa_1^\dagger(\mathrm{id} + \langle p \rangle^\dagger)\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle f} = \overline{\mathrm{id}\,\kappa_1^\dagger\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle f} \\
&= \overline{\kappa_1^\dagger\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle f} = \overline{\kappa_1^\dagger\langle q \rangle\kappa_1^\dagger\langle p \rangle f} \\
&= \overline{\overline{\kappa_1^\dagger\langle q \rangle}\,\overline{\kappa_1^\dagger\langle p \rangle}f} = \overline{\overline{\kappa_1^\dagger\langle q \rangle}f\,\kappa_1^\dagger\langle p \rangle f} \\
&= \overline{f\kappa_1^\dagger\langle q \rangle f\,\kappa_1^\dagger\langle p \rangle f} = \overline{\overline{f}\,\overline{\kappa_1^\dagger\langle q \rangle f}\,\kappa_1^\dagger\langle p \rangle f} \\
&= \overline{\overline{f}\,\kappa_1^\dagger\langle q \rangle f\,\kappa_1^\dagger\langle p \rangle f} = \overline{\kappa_1^\dagger\langle q \rangle f\overline{f}\,\kappa_1^\dagger\langle p \rangle f} \\
&= \overline{\kappa_1^\dagger\langle q \rangle f\overline{f}\,\kappa_1^\dagger\langle p \rangle f} = \overline{\kappa_1^\dagger\langle q \rangle f\,\kappa_1^\dagger\langle p \rangle f}
\end{aligned}
$$

and

$$
\begin{aligned}
\kappa_1^\dagger(f^\diamond(\langle p\rangle \wedge f^\diamond(\langle q\rangle))) &= \kappa_1^\dagger\langle pf\rangle \wedge \langle qf\rangle = \overline{\kappa_1^\dagger(\overline{\langle pf\rangle} \wedge \langle qf\rangle)}\\
&= \overline{\kappa_1^\dagger(\mathrm{id} + \langle pf\rangle)\alpha(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle} = \overline{\mathrm{id}\,\kappa_1^\dagger\alpha(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle}\\
&= \overline{\kappa_1^\dagger\alpha(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle} = \overline{\kappa_1^\dagger\kappa_1^\dagger(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle}\\
&= \overline{\kappa_1^\dagger\langle qf\rangle\kappa_1^\dagger\langle pf\rangle} = \overline{\kappa_1^\dagger\langle qf\rangle}\,\overline{\kappa_1^\dagger\langle pf\rangle}\\
&= \kappa_1^\dagger\langle qf\rangle\,\kappa_1^\dagger\langle pf\rangle
\end{aligned}
$$

so $\kappa_1^\dagger(f^\diamond(\langle p\rangle) \wedge f^\diamond(\langle q\rangle)) = \kappa_1^\dagger(f^\diamond(\langle p\rangle) \wedge f^\diamond(\langle q\rangle))$. For the second part,

$$
\begin{aligned}
\kappa_2^\dagger f^\diamond(\langle p\rangle \wedge \langle q\rangle) &= \kappa_2^\dagger\langle((\langle p\rangle \wedge \langle q\rangle)f\rangle = \overline{\kappa_2^\dagger(\overline{\langle p\rangle \wedge \langle q\rangle})f}\\
&= \overline{\kappa_2^\dagger(\mathrm{id} + \langle p\rangle^\dagger)\alpha(\langle q\rangle + \overline{\langle q\rangle})\langle p\rangle f} = \overline{\langle p\rangle^\dagger\kappa_2^\dagger\alpha(\langle q\rangle + \overline{\langle q\rangle})\langle p\rangle f}\\
&= \overline{\langle p\rangle^\dagger(\kappa_2^\dagger + \mathrm{id})(\langle q\rangle + \overline{\langle q\rangle})\langle p\rangle f} = \overline{\langle p\rangle^\dagger((\kappa_2^\dagger\langle q\rangle) + \overline{\langle q\rangle})\langle p\rangle f}\\
&= \overline{\langle p\rangle^\dagger(\kappa_2^\dagger\langle q\rangle + \overline{\langle q\rangle})\langle p\rangle f} = \overline{(\kappa_2^\dagger\langle q\rangle + \overline{\langle q\rangle})\langle p\rangle^\dagger\langle p\rangle f}\\
&= \overline{(\overline{\kappa_2^\dagger\langle q\rangle} + \overline{\langle q\rangle})\langle p\rangle f} = \overline{((\kappa_2^\dagger\langle q\rangle) + \langle q\rangle)\langle p\rangle f}\\
&= \overline{((\kappa_2^\dagger\langle q\rangle) + \langle q\rangle)\langle p\rangle\overline{\langle p\rangle}f} = \overline{((\kappa_2^\dagger\langle q\rangle\overline{\langle p\rangle}) + (\langle q\rangle\overline{\langle p\rangle}))\langle p\rangle f}\\
&= ((\kappa_2^\dagger\langle q\rangle\overline{\langle p\rangle}f) + (\langle q\rangle\overline{\langle p\rangle}f))\langle pf\rangle
\end{aligned}
$$

and

$$
\begin{aligned}
\kappa_2^\dagger(f^\diamond(\langle p\rangle) \wedge f^\diamond(\langle q\rangle)) &= \kappa_2^\dagger(\langle pf\rangle \wedge \langle qf\rangle) = \overline{\kappa_2^\dagger(\overline{\langle pf\rangle} \wedge \langle qf\rangle)}\\
&= \overline{\kappa_2^\dagger(\mathrm{id} + \langle pf\rangle^\dagger)\alpha(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle} = \overline{\langle pf\rangle^\dagger\kappa_2^\dagger\alpha(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle}\\
&= \overline{\langle pf\rangle^\dagger(\kappa_2^\dagger + \mathrm{id})(\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle} = \overline{\langle pf\rangle^\dagger((\kappa_2^\dagger\langle qf\rangle) + \overline{\langle qf\rangle})\langle pf\rangle}\\
&= \overline{\langle pf\rangle^\dagger(\kappa_2^\dagger\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle} = \overline{(\kappa_2^\dagger\langle qf\rangle + \overline{\langle qf\rangle})\langle pf\rangle^\dagger\langle pf\rangle}\\
&= \overline{(\overline{\kappa_2^\dagger\langle qf\rangle} + \overline{\langle qf\rangle})\langle pf\rangle} = \overline{(\overline{\kappa_2^\dagger\langle q\rangle f} + \overline{\langle q\rangle f})\langle pf\rangle}\\
&= \overline{((\kappa_2^\dagger\langle q\rangle f) + (\langle q\rangle f))\langle pf\rangle} = \overline{((\kappa_2^\dagger\langle q\rangle f) + (\langle q\rangle f))\langle pf\rangle\overline{\langle pf\rangle}}\\
&= \overline{((\kappa_2^\dagger\langle q\rangle f) + (\langle q\rangle f))\langle pf\rangle\overline{\langle p\rangle}f} = \overline{((\kappa_2^\dagger\langle q\rangle f\overline{\langle p\rangle}f) + (\langle q\rangle f\overline{\langle p\rangle}f))\langle pf\rangle}\\
&= \overline{((\kappa_2^\dagger\langle q\rangle\overline{\langle p\rangle}f) + (\langle q\rangle\overline{\langle p\rangle}f))\langle pf\rangle}
\end{aligned}
$$

so also $\kappa_2^\dagger f^\diamond(\langle p\rangle \wedge \langle q\rangle) = \kappa_2^\dagger(f^\diamond(\langle p\rangle) \wedge f^\diamond(\langle q\rangle))$, which finally gives us $f^\diamond(\langle p\rangle \wedge \langle q\rangle) = f^\diamond(\langle p\rangle) \wedge f^\diamond(\langle q\rangle)$. □

Proof. [Proof of 4.9] (i) $\overline{\neg\langle p\rangle} = \overline{\gamma\langle p\rangle} = \overline{\overline{\gamma}\langle p\rangle} = \overline{\mathrm{id}\langle p\rangle} = \overline{\langle p\rangle}$.

(ii) We have

$$
\begin{aligned}
\overline{\langle p \rangle \wedge \langle q \rangle} &= \overline{(\mathrm{id} + \langle p \rangle^\dagger)\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{(\mathrm{id} + \nabla)\alpha(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{\overline{(\mathrm{id} + \nabla)\alpha}(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{\mathrm{id}(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{(\langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{(\overline{\langle q \rangle} + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{\langle p \rangle \overline{\langle q \rangle}} \\
&= \overline{\langle p \rangle}\,\overline{\langle q \rangle}
\end{aligned}
$$

(iii) follows by analogous reasoning to (ii). We have (iv) immediately by (ii) since $\overline{\langle p \rangle \wedge \langle q \rangle} = \overline{\langle p \rangle}\,\overline{\langle q \rangle} \le \overline{\langle p \rangle}$ directly, and likewise $\overline{\langle p \rangle \wedge \langle q \rangle} = \overline{\langle p \rangle}\,\overline{\langle q \rangle} \le \overline{\langle q \rangle}$. (v) follows analogously. □

Proof. [Proof of Lemma 4.8] Assume $\kappa_1^\dagger \langle p \rangle \le \kappa_1^\dagger \langle q \rangle$ and $\overline{\langle p \rangle} \le \overline{\langle q \rangle}$. By Lemma 3.3, to show $\langle p \rangle \wedge \langle q \rangle = \langle p \rangle$ (*i.e.*, $\langle p \rangle \vDash \langle q \rangle$) it suffices to show that $\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle) = \kappa_1^\dagger \langle p \rangle$ and $\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle) = \kappa_2^\dagger \langle p \rangle$.

Since
$$
\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle) = \overline{\kappa_1^\dagger p}\,\overline{\kappa_1^\dagger q} = \overline{\kappa_1^\dagger \langle p \rangle}\,\overline{\kappa_1^\dagger \langle q \rangle} = \overline{\overline{\kappa_1^\dagger \langle p \rangle}} = \kappa_1^\dagger \langle p \rangle
$$
by the proof of Lemma 4.3, $\kappa_1^\dagger \langle p \rangle \le \kappa_1^\dagger \langle q \rangle$, and Lemma 3.2, proving the first part. For the second part,

$$
\begin{aligned}
\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle) &= \overline{\overline{\kappa_2^\dagger(\langle p \rangle \wedge \langle q \rangle)}} = \overline{\overline{\langle p \rangle^\dagger(\kappa_2^\dagger \langle q \rangle + \overline{\langle q \rangle})\langle p \rangle}} \\
&= \overline{\overline{\langle p \rangle^\dagger}(\kappa_2^\dagger \langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} = \overline{(\overline{\kappa_1^\dagger \langle p \rangle} + \kappa_2^\dagger \langle p \rangle)(\kappa_2^\dagger \langle q \rangle + \overline{\langle q \rangle})\langle p \rangle} \\
&= \overline{((\overline{\kappa_1^\dagger \langle p \rangle}\,\kappa_2^\dagger \langle q \rangle) + (\kappa_2^\dagger \langle p \rangle\,\overline{\langle q \rangle}))\langle p \rangle} = \overline{((\overline{\kappa_1^\dagger \langle p \rangle}\,\overline{\kappa_1^\dagger \langle q \rangle}\,\kappa_2^\dagger \langle q \rangle) + (\kappa_2^\dagger \langle p \rangle\,\overline{\langle q \rangle}))\langle p \rangle} \\
&= \overline{((\overline{\kappa_1^\dagger \langle p \rangle}\,0) + (\kappa_2^\dagger \langle p \rangle\,\overline{\langle q \rangle}))\langle p \rangle} = \overline{(0 + (\kappa_2^\dagger \langle p \rangle\,\overline{\langle q \rangle}))\langle p \rangle} \\
&= \overline{(0 + (\kappa_2^\dagger \langle p \rangle \overline{\langle q \rangle}))\langle p \rangle} = \overline{(0 + (\kappa_2^\dagger \langle p \rangle \overline{\langle p \rangle}\,\overline{\langle q \rangle}))\langle p \rangle} \\
&= \overline{(0 + (\kappa_2^\dagger \langle p \rangle \overline{\langle p \rangle}))\langle p \rangle} = \overline{(0 + (\kappa_2^\dagger \langle p \rangle))\langle p \rangle} \\
&= \overline{((0\,\overline{\kappa_1^\dagger \langle p \rangle}) + (\kappa_2^\dagger \langle p \rangle))\langle p \rangle} = \overline{(0 + \mathrm{id})(\overline{\kappa_1^\dagger \langle p \rangle} + \kappa_2^\dagger \langle p \rangle)\langle p \rangle} \\
&= \overline{(0 + \mathrm{id})\langle p \rangle^\dagger \langle p \rangle} = \overline{(0 + \mathrm{id})\langle p \rangle} = \overline{\kappa_2^\dagger \langle p \rangle} = \overline{\overline{\kappa_2^\dagger \langle p \rangle}} = \kappa_2^\dagger \langle p \rangle
\end{aligned}
$$

In the other direction, suppose that $\langle p \rangle \vDash \langle q \rangle$, *i.e.*, $\langle p \rangle \wedge \langle q \rangle = \langle p \rangle$. Then $\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle) = \kappa_1^\dagger \langle p \rangle = \kappa_1^\dagger \langle p \rangle$, but since we also know that $\kappa_1^\dagger(\langle p \rangle \wedge \langle q \rangle) = \overline{\kappa_1^\dagger \langle p \rangle}\,\kappa_1^\dagger \langle q \rangle$ (see above), it follows that
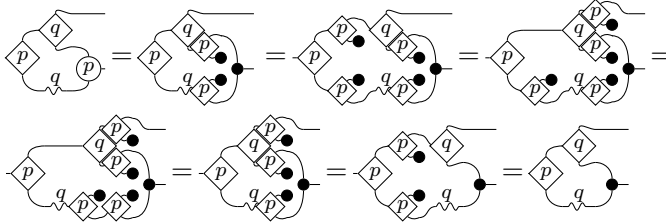
$$
\overline{\kappa_1^\dagger \langle q \rangle \kappa_1^\dagger \langle p \rangle} = \overline{\kappa_1^\dagger \langle q \rangle}\,\overline{\kappa_1^\dagger \langle p \rangle} = \overline{\kappa_1^\dagger \langle p \rangle}\,\overline{\kappa_1^\dagger \langle q \rangle} = \overline{\overline{\kappa_1^\dagger \langle p \rangle}} = \kappa_1^\dagger \langle p \rangle
$$

that is, $\kappa_1^\dagger \langle p \rangle \leq \kappa_1^\dagger \langle q \rangle$. That $\overline{\langle p \rangle} \leq \overline{\langle q \rangle}$ follows by Lemma 4.9, as we thus have $\overline{\langle p \rangle} = \overline{\langle p \rangle \wedge \langle q \rangle} \leq \overline{\langle q \rangle}$. □

Proof. [Proof of Lemma 4.4] To prove (i), it suffices to show that their partial inverses are equal, since partial inverses are unique. We show this as follows:
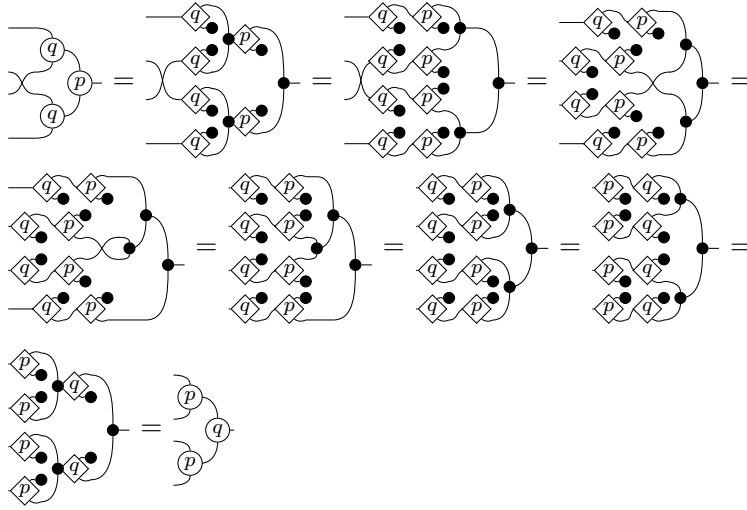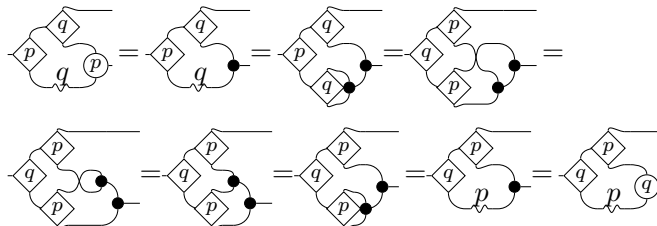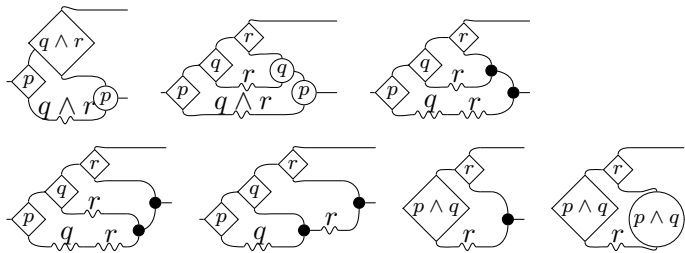

(ii) follows by
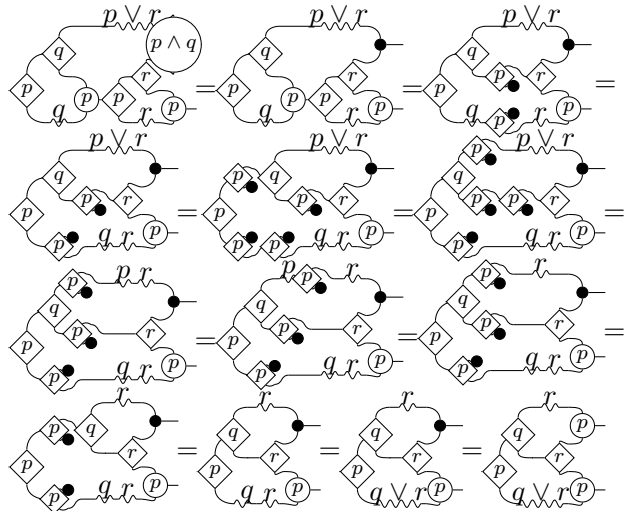

(iii) follows by analogous argument to (ii). □

Proof. [Proof of Lemma 4.5] Commutativity of conjunction follows by

and commutativity of disjunction analogously. Associativity of conjunction is demonstrated by

and associativity of disjunction can be shown similarly. For distributivity of conjunction over disjunction,

and the dual distributive law follows symmetrically. Finally, the double negation law then follows simply by

which concludes the proof. □