11- Disjunctive algebras

We shall now introduce the notion of disjunctive algebra, which is a structure primarily based on disjunction, negation (for the connectives) and meets (for the universal quantifier). Our main purpose is to draw the comparison with implicative algebras, as an attempt to justify eventually that the latter are at least as general as the former. All along this chapter, we will follow the same rationale which guided the definition of implicative structures, separators, etc... If we will not be able to recover all the disjunctive counterpart of the properties of implicative algebras, we should anyway be convinced in the end that disjunctive algebras do not bring any benefits over the implicative one, in the sense that disjunctive algebras are particular cases of implicative algebras.

The first step in this direction is the definition of disjunctive structures. Our starting point is the fact that in classical logic, the following equivalence holds for all formulas A and B:

$$A \to B \qquad \Leftrightarrow \qquad \neg A \vee B$$

In particular, this equivalence suggests that as long as we are interested in a classical framework, we could as well define the logic with the disjunction and negation as ground connectives. This is for instance the choice of Bourbaki in his *Éléments de mathématique* [21]. The first volume of the famous treatise begins precisely with the introduction of the logical symbols, which are $\neg$, $\vee$ plus two others used to handle substitutions. The first symbolic shorthand which is defined is precisely the implication, and logic is axiomatized by the following schemes:

$$\begin{array}{lcl@{\qquad\qquad}lcl} S1 & : & (A \vee A) \to A & S3 & : & (A \vee B) \to (B \vee A) \\ S2 & : & A \to (A \vee B) & S4 & : & (A \to B) \to ((C \vee A) \to (C \vee B)) \end{array}$$

These logical schemes should give us a guideline in the definition of separators for disjunctive structures.

In the seminal paper introducing linear logic [58], Jean-Yves Girard refines the structure of the sequent calculus LK, introducing in particular two connectives for the disjunctions: $\parr$ and $\oplus$. The first one is said to be multiplicative, while the second one is said to be additive, due to the treatment of contexts in the corresponding rules:

$$\frac{\Gamma \vdash A_1, A_2, \Delta}{\Gamma \vdash A_1 \parr A_2, \Delta}\,(\parr_r) \quad \frac{\Gamma_1, A \vdash \Delta_1 \quad \Gamma_2, B \vdash \Delta_2}{\Gamma_1, \Gamma_2, A_1 \parr A_2 \vdash \Delta_1, \Delta_2}\,(\parr_l) \qquad\qquad \frac{\Gamma \vdash A_i, \Delta}{\Gamma \vdash A_1 \oplus A_2, \Delta}\,(\oplus_r) \quad \frac{\Gamma, A_1 \vdash \Delta \quad \Gamma, A_2 \vdash \Delta}{\Gamma, A_1 \parr A_2 \vdash \Delta}\,(\oplus_l)$$

In the (multiplicative) rules for $\parr$, contexts are indeed juxtaposed, while they are identified in the (additive) rule for $\oplus$. With this finer set of connectives, Girard shows that the usual implication[1] can be retrieved using the multiplicative disjunction:

$$A \to B \quad \triangleq \quad \neg A \parr B$$

[1] To do justice to Girard's approach, the implication which is considered in linear logic, written $\multimap$, is different from the usual one. The difference between both implications is not relevant in our framework.

Dually to these two connectives for the disjunction, two connectives are also introduced for the conjunction! $\otimes$ (multiplicative) and & (additive). Disjunctive and conjunctive connectives are related through some laws of duality which are very similar to De Morgan's laws for classical logic. For instance, the multiplicative connectives verify that $\neg(A \parr B) = \neg A \otimes \neg B$ and $\neg(A \otimes B) = \neg A \parr \neg B$. In particular, this give rises to a second decomposition of the arrow:

$$A \to B \quad \triangleq \quad \neg(A \otimes \neg B)$$

In 2009, Guillaume Munch-Maccagnoni gave a computational account of Girard's presentation for classical logic with a division between multiplicative and additive connectives [126]. In his calculus, named L, each connective corresponds to the type of a particular constructor (or destructor). While L is in essence close to Curien and Herbelin's $\lambda\mu\tilde{\mu}$-calculus (in particular it is presented with the same paradigm of duality between proofs and contexts), the syntax of terms does not include λ-abstraction (and neither does the syntax of formulas includes an implication). The two decompositions of the arrow evoked above are precisely reflected in a decomposition of λ-abstractions (and dually, of stacks) in terms of L constructors.

Notably, the choice of a decomposition corresponds to a particular choice of an evaluation strategy for the encoded λ-calculus. When picking the $\parr$ connective, the corresponding λ-terms are evaluated according to a call-by-name evaluation strategy whose machinery resembles the one of the call-by-name $\lambda\mu\tilde{\mu}$-calculus (see Chapter 4). On the other hand, if the implication is defined through the $\otimes$ connective, the corresponding λ-calculus is reduced in a call-by-value fashion.

We shall begin by considering the call-by-name case, which is closer to the situation of implicative algebras. We start with the presentation of the corresponding fragment of Munch-Maccagnoni's calculus, which we call $L^{\parr}$. In particular, we will see how this calculus induces a realizability model whose structure leads us to the definition of *disjunctive structures*. We will observe that the encoding of λ-terms into $L^{\parr}$ can be directly reflected as an implicative structure induced by each disjunctive structure. Finally, we shall define the notions of (disjunctive) *separator* and *disjunctive algebra*. We will see that, again, any disjunctive algebra can be viewed as an implicative algebra.

11.1 The $L^{\parr}$ calculus

We present here the fragment of L induced by the negative connectives $\parr$, $\neg$ and $\forall$, in order to present afterwards the realizability model it induces. Since this calculus has a lot of similarities with respect to the $\lambda\mu\tilde{\mu}$-calculus, and since the realizability interpretation is akin to the one we gave for the call-by-name $\lambda\mu\tilde{\mu}$-calculus (see Section 4.4.5), we shall try to be concise. In particular, we skip some proofs which can be found either in [126] or in previous chapters.

11.1.1 The $L^{\parr}$ calculus

The $L^{\parr}$-calculus is a subsystem of Munch-Maccagnoni's system L [126], restricted to the negative fragment corresponding to the connectives $\parr$, $\neg^-$ (which we simply write $\neg$ since there is no ambiguity here) and $\forall$. To ease the connection with the syntax of the $\lambda\mu\tilde{\mu}$-calculus, we slightly change the notations of the original paper. The syntax is given by:

Contexts	e^+	::=	$\alpha \mid (e_1^+, e_2^+) \mid [t^-] \mid \mu x.c$
Terms	t^-	::=	$x \mid \mu(\alpha_1, \alpha_2).c \mid \mu[x].c \mid \mu\alpha.c$
Commands	c	::=	$\langle t^- \Vert e^+ \rangle$

Observe in particular that we only have positive contexts and negative terms. We write $\mathcal{E}$ for the set of contexts, $\mathcal{T}$ for the set of terms, C for the set of commands, and $\mathcal{E}_0$, $\mathcal{T}_0$, C_0 for the sets of closed contexts, terms and commands. As for values, they are defined by the following fragment of the syntax[2]:

Values $\qquad V ::= \alpha \mid (V_1, V_2) \mid [t^-]$

We denote by $\mathcal{V}_0$ the corresponding set of closed values.

Since the notations might be a bit confusing regarding the ones we used in previous chapters (especially with respect to the $\lambda\mu\tilde{\mu}$-calculus), we shall say a few words about it:

- (e^+, e^+) are pairs of positive contexts, which we will relate to usual stacks;
- $\mu(\alpha_1, \alpha_2).c$, which binds the co-variables α_1, α_2, is the dual destructor;
- $[t^-]$ is a constructor for the negation, which allows us to embed a negative term into a positive context;
- $\mu[x].c$, which binds the variable x, is the dual destructor;
- $\mu\alpha.c$ and $\mu x.c$ correspond respectively to $\mu\alpha$ and $\tilde{\mu}x$ in the $\lambda\mu\tilde{\mu}$-calculus.

Remark 11.1 (Notations)**.** We shall explain that in (full) L, the same syntax allows us to define terms t and contexts e (thanks to the duality between them). In particular, no distinction is made between t and e, which are both written t, and commands are indifferently of the shape $\langle t^+ \| t^- \rangle$ or $\langle t^- \| t^+ \rangle$. For this reason, in [126] is considered a syntax where a notation $\bar{x}$ is used to distinguish between the positive variable x (that can appear in the left-member $\langle x|$ of a command) and the positive co-variable $\bar{x}$ (resp. in the right member $|x\rangle$ of a command). In particular, the $\mu\alpha$ binder of the $\lambda\mu\tilde{\mu}$-calculus would have been written $\mu\bar{x}$ and the $\tilde{\mu}x$ binder would have been denoted by $\mu\alpha$ (see [126, Appendix A.2]). We thus switched the x and α of L (and removed the bar), in order to stay coherent with the notations in the rest of this manuscript. ⌟

The reduction rules correspond to what could be expected from the syntax of the calculus: destructors reduce in front of the corresponding constructors, both μ binders catch values in front of them and pairs of contexts are expanded if they are not values. As for the η-expansion rules, they are also quite natural:

$$\begin{array}{rcl} \langle \mu[x].c \| [t] \rangle & \to_\beta & c[t/x] \\ \langle t \| \mu x.c \rangle & \to_\beta & c[t/x] \\ \langle \mu\alpha.c \| V \rangle & \to_\beta & c[V/\alpha] \\ \langle \mu(\alpha_1,\alpha_2).c \| (V_1,V_2) \rangle & \to_\beta & c[V_1/\alpha_1, V_2/\alpha_2] \\ \langle t \| (e,e') \rangle & \to_\beta & \langle \mu\alpha.\langle \mu\alpha'.\langle t \| (\alpha,\alpha') \rangle \| e' \rangle \| e \rangle \end{array} \qquad \begin{array}{rcl} c & \to_\eta & \langle \mu\alpha.c \| \alpha \rangle \\ c & \to_\eta & \langle \mu(\alpha_1,\alpha_2).c \| (\alpha_1,\alpha_2) \rangle \\ c & \to_\eta & \langle \mu[x].c \| [x] \rangle \\ c & \to_\eta & \langle x \| \mu x.c \rangle \end{array}$$

where in the last $\to_\beta$ rule, $(e, e') \notin V$.

Finally, we shall present the type system of L$^{⅋}$. In the continuity of the presentation of implicative algebras, we are interested in a second-order settings. Formulas are then defined by the following grammar:

Formulas $\qquad A, B := X \mid A ⅋ B \mid \neg A \mid \forall X.A$

Once again, the type system is similar to the one of the $\lambda\mu\tilde{\mu}$-calculus, in the sense that it is presented in a sequent calculus fashion. We work with two-sided sequents, where typing contexts are defined as usual as finite lists of bindings between variable and formulas:

$$\Gamma ::= \varepsilon \mid \Gamma, x : A \qquad\qquad \Delta ::= \varepsilon \mid \Delta, \alpha : A$$

Sequents are of three kinds, as in the $\lambda\mu\tilde{\mu}$-calculus:

[2]The reader may observe that in this setting, values are defined as contexts, so that we may have called them *covalues* rather than values. We stick to this denomination to stay coherent with the terminology in Munch-Maccagnoni's paper [126].

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \| e \rangle : \Gamma \vdash \Delta}\ (\textsc{Cut})$$

$$\frac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta}\ (ax\vdash) \qquad \frac{(x : A) \in \Gamma}{\Gamma \vdash x : A \mid \Delta}\ (\vdash ax)$$

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \mid \mu x.c : A \vdash \Delta}\ (\mu\vdash) \qquad \frac{c : \Gamma \vdash \Delta, \alpha : A}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}\ (\vdash\mu)$$

$$\frac{\Gamma \mid e_1 : A \vdash \Delta \quad \Gamma \mid e_2 : B \vdash \Delta}{\Gamma \mid (e_1, e_2) : A \parr B \vdash \Delta}\ (\parr\vdash) \qquad \frac{c : \Gamma \vdash \Delta, \alpha_1 : A, \alpha_2 : B}{\Gamma \vdash \mu(\alpha_1, \alpha_2).c : A \parr B \mid \Delta}\ (\vdash\parr)$$

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \mid [t] : \neg A \vdash \Delta}\ (\neg\vdash) \qquad \frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta}\ (\vdash\neg)$$

$$\frac{\Gamma \mid e : A[B/X] \vdash \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta}\ (\forall\vdash) \qquad \frac{\Gamma \vdash t : A \mid \Delta \quad X \notin FV(\Gamma, \Delta)}{\Gamma \vdash t : \forall X.A}\ (\vdash\forall)$$

Figure 11.1: Typing rules for the $L_{\parr,\neg}$-calculus

- $\Gamma \vdash t : A \mid \Delta$ for typing terms,
- $\Gamma \mid e : A \vdash \Delta$ for typing contexts,
- $c : \Gamma \vdash \Delta$ for typing commands.

Just like both connectives $\parr$ and $\neg$ are reflected by a constructor and a destructor in the syntax, in the type system each connective corresponds to a left rule (the introduction rule, for typing the constructor) and to a right rule (the elimination rule, for typing the destructor), in addition to the usual rules for typing variables, μ binders and commands. The type system is given in Figure 11.1.

Remark 11.2 (Universal quantifier)**.** In L, the universal quantification is also reflected by constructors in the syntax. This has the benefits of avoiding the problems of value restriction for the introduction rule. In our particular setting, since all terms are values, the introduction rule does not cause any problem. Beyond that, the realizability model we are going to define is only a pretext to the introduction of disjunctive structures, in which we will interpret the universal quantification by meets. Thus, it would be meaningless for us to introduce a syntactic constructor for the universal quantifier. ⌟

Remark 11.3 (Multiplicativity)**.** We simplified a bit the type system of L to avoid structural rules. Therefore, the rule ($\parr\vdash$) uses the same contexts in both hypotheses and the conclusion, instead of juxtaposing contexts in the conclusion. Both presentations are equivalent since both type systems allow for weakening and contraction. ⌟

11.1.2 Embedding of the λ-calculus

Following Munch-Maccagnoni's paper [126, Appendix E], we can embed the λ-calculus into the $L^{\parr}$-calculus. To this end, we are guided by the expected definition of the arrow:

$$A \to B \ \triangleq\ \neg A \parr B$$

It is easy to see that with this definition, a stack $u \cdot e$ in $A \to B$ (that is with u a term of type A and e a context of type B) is naturally defined as a shorthand for the pair $([u], e)$, which indeed inhabits the type $\neg A \parr B$. Starting from there, the rest of the definitions are straightforward:

$$\begin{array}{rcl}
u \cdot e & \triangleq & ([u], e) \\
\mu([x], \beta).c & \triangleq & \mu(\alpha, \beta).\langle \mu[x].c \| \alpha \rangle \\
\lambda x.t & \triangleq & \tilde{\mu}([x], \beta).\langle t \| \beta \rangle \\
t\,u & \triangleq & \mu\alpha.\langle t \| u \cdot \alpha \rangle
\end{array}$$

These definitions are sound with respect to the typing rules expected from the $\lambda\mu\tilde{\mu}$-calculus:

Proposition 11.4. *The following typing rules are admissible:*

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B} \qquad \frac{\Gamma \vdash u : A \mid \Delta \quad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid u \cdot e : A \to B \vdash \Delta} \qquad \frac{\Gamma \vdash t : A \to B \mid \Delta \quad \Gamma \vdash u : A \mid \Delta}{\Gamma \vdash t\,u : B \mid \Delta}$$

Proof. Each case is directly derivable from L$^{\parr}$ type system. We abuse the notation to denote by (*def*) a rule which simply consists in unfolding the shorthands defining the λ-terms.

- **Case** $\mu([x], \alpha).c$:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\Gamma \mid \alpha : \neg A \vdash \Delta, \alpha : \neg A, \beta : B}\,(ax\vdash) \quad \cfrac{c : (\Gamma, x : A \vdash \Delta, \beta : B)}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta, \beta : B}\,(\vdash\mu)}{\langle \mu[x].c \| \alpha \rangle : (\Gamma \vdash \Delta, \alpha : \neg A, \beta : B)}\,(\textsc{Cut})}{\Gamma \vdash \mu(\alpha, \beta).\langle \mu[x].c \| \alpha \rangle : \neg A \parr B \mid \Delta}\,(\vdash\parr)}{\Gamma \vdash \mu([x], \beta).c : \neg A \parr B \mid \Delta}\,(def)$$

- **Case** $\lambda x.t$:

$$\cfrac{\cfrac{\cfrac{\Gamma, x : A \vdash t : B \mid \Delta \quad \cfrac{}{\Gamma \mid \beta : B \vdash \Delta, \beta : B}\,(ax\vdash)}{\langle t \| \beta \rangle : (\Gamma, x : A \vdash \beta : B, \Delta}\,(\textsc{Cut})}{\Gamma \vdash \mu([x], \beta).\langle t \| \beta \rangle : \neg A \parr B \mid \Delta}\,(\vdash\parr)}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta}\,(def)$$

- **Case** $u \cdot e$:

$$\cfrac{\cfrac{\cfrac{\Gamma \vdash u : A \mid \Delta}{\Gamma \mid [u] : A \vdash \Delta}\,(\neg\vdash) \quad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid ([u], e) : \neg A \parr B \vdash \Delta}\,(\parr\vdash)}{\Gamma \mid u \cdot e : A \to B \vdash \Delta}\,(def)$$

- **Case** $t\,u$:

$$\cfrac{\cfrac{\cfrac{\Gamma \vdash t : A \to B \mid \Delta \quad \cfrac{\Gamma \vdash u : A \mid \Delta \quad \cfrac{}{\Gamma \mid \alpha : B \vdash \Delta, \alpha : B}}{\Gamma \mid u \cdot \alpha : A \to B \vdash \Delta, \alpha : B)}}{\langle t \| u \cdot \alpha \rangle : (\Gamma \vdash \Delta, \alpha : B)}\,(\textsc{Cut})}{\Gamma \vdash \mu\alpha.\langle t \| u \cdot \alpha \rangle : B \mid \Delta}\,(\vdash\mu)}{\Gamma \vdash t\,u : B \mid \Delta}\,(def)$$

□

In addition, the above definitions of λ-terms induce the usual rules of β-reduction for the call-by-name evaluation strategy in the Krivine abstract machine (notice that in the KAM, all stacks are values):

Proposition 11.5 (β-reduction)**.** *We have the following reduction rules:*

$$\begin{array}{rcll}
\langle t\,u \| \pi \rangle & \to_\beta & \langle t \| u \cdot \pi \rangle & (\pi \in V^+) \\
\langle \lambda x.t \| u \cdot \pi \rangle & \to_\beta & \langle t[u/x] \| \pi \rangle & (\pi \in V^+)
\end{array}$$

Proof. If $\pi \in V^+$, we have indeed:

$$\langle t\,u \| \pi \rangle = \langle \mu\alpha.\langle t \| u \cdot \alpha \rangle \| \pi \rangle \rightarrow_\beta \langle t \| u \cdot \pi \rangle$$

and:

$$\begin{aligned}\langle \lambda x.t \| u \cdot \pi \rangle &= \langle \tilde{\mu}([x],\beta).\langle t \| \beta \rangle \| ([u],\pi)\rangle \\ &= \langle \mu(\alpha,\beta).\langle \alpha \| \mu[x].\langle t \| \beta \rangle\rangle \| ([u],\pi)\rangle \\ &\rightarrow_\beta \langle [u] \| \mu[x].\langle t \| \pi \rangle\rangle \\ &\rightarrow_\beta \langle t[u/x] \| \pi \rangle\end{aligned}$$

□

At this stage, it is clear that the structure of $L^{⅋}$ allows us to recover all the computational strength of the call-by-name $\lambda\mu\tilde{\mu}$-calculus. As we explained in Section 4.2.4, this also means that we can encode the term $\boldsymbol{cc}$ of the λ_c-calculus, and simulate the Krivine abstract machine. Therefore, $L^{⅋}$ is suitable for the definition of a realizability interpretation through these encodings, but as for the full system L, we can also directly define a realizability model for $L^{⅋}$.

11.1.3 A realizability model based on the $L^{⅋}$-calculus

We briefly go through the definition of the realizability interpretation *à la* Krivine for $L^{⅋}$. The reader should observe that in the end, this interpretation is very similar to the one of the call-by-name $\lambda\mu\tilde{\mu}$-calculus (see Section 4.4.5). As usual, we begin with the definition of a pole:

Definition 11.6 (Pole)**.** A subset $\perp\!\!\!\perp \in C$ is said to be saturated whenever for all $c, c' \in C$, if $c \rightarrow_\beta c'$ then $c \in \perp\!\!\!\perp$. A *pole* is defined as any saturated subset of C_0. ⌟

As it is common in Krivine's call-by-name realizability, falsity values are defined primitively as sets of contexts. Truth values are then defined by orthogonality to the corresponding falsity values. We say that a term t is *orthogonal* (with respect to the pole $\perp\!\!\!\perp$) to a context e we denote by $t \perp\!\!\!\perp e$ when $\langle t \| e \rangle \in \perp\!\!\!\perp$. A term t (resp. a context e) is said to be orthogonal to a set $S \subseteq \mathcal{E}_0$ (resp. $S \subseteq \mathcal{T}_0$), which we write $t \perp\!\!\!\perp S$, when for all $e \in S$, t is orthogonal to e.

Orthogonality satisfies the expected properties of monotonicity:

Proposition 11.7 (Monotonicity)**.** *For any subset S of $\mathcal{T}_0$ (resp. $\mathcal{E}_0$) and any subset $\mathcal{U} \in \mathcal{P}(\mathcal{T}_0)$ (resp. any subset of $\mathcal{P}(\mathcal{E}_0)$), the following holds:*

1. $S \subseteq S^{\perp\!\!\!\perp\perp\!\!\!\perp}$
2. $S^{\perp\!\!\!\perp} = S^{\perp\!\!\!\perp\perp\!\!\!\perp\perp\!\!\!\perp}$
3. $(\bigcap_{S \in \mathcal{U}} S)^{\perp\!\!\!\perp} = \bigcup_{S \in \mathcal{U}} (S^{\perp\!\!\!\perp})$
4. $(\bigcup_{S \in \mathcal{U}} S)^{\perp\!\!\!\perp} \supseteq \bigcap_{S \in \mathcal{U}} (S^{\perp\!\!\!\perp})$

As we explained in more details in chapter 4, the realizability interpretation *à la* Krivine of a calculus given in a sequent calculus presentation (that is whose reduction rules are presented in the shape of an abstract machine) can be derived mechanically from a small-step reduction system. We will not do it in the present case, but it amounts to the case of the call-by-name $\lambda\mu\tilde{\mu}$-calculus. Because of this evaluation strategy (which is induced here by the choice of connectives), a formula A is primitively interpreted by its "falsity value of values", which we write $\|A\|_V$ and call *primitive falsity value*, which is a set in $\mathcal{P}(\mathcal{V}_0)$ (and thus in $\mathcal{P}(\mathcal{E}_0)$). Its *truth value* $|A|$ is then defined by orthogonality to $\|A\|_V$ (and is a set in $\mathcal{P}(\mathcal{T}_0)$), while its *falsity value* $\|A\| \in \mathcal{P}(\mathcal{E}_0)$ is again obtained by orthogonality to $|A|$. Therefore, a universal formula $\forall X.A$ is interpreted by the union over all the possible instantiations for the primitive falsity value of the variable X by a set $S \in \mathcal{P}(\mathcal{V}_0)$. As it is usual in Krivine realizability, to ease the definitions we assume that for each subset S of $\mathcal{P}(\mathcal{V}_0)$, there is a constant symbol $\dot{S}$ in the syntax. The interpretation is given by:

$$\begin{array}{rcl}
\|\dot{S}\|_V & \triangleq & S \\
\|\forall X.A\|_V & \triangleq & \bigcup_{S\in\mathcal{P}(\mathcal{V}_0)} \|A\{X := \dot{S}\}\|_V \\
\|A \parr B\|_V & \triangleq & \{(V_1,V_2) : V_1 \in \|A\|_V \wedge V_2 \in \|B\|_V\} \\
\|\neg A\|_V & \triangleq & \{[t] : t \in |A|\} \\
|A| & \triangleq & \{t : \forall V \in \|A\|_V, t \perp\!\!\!\perp V\} \\
\|A\| & \triangleq & \{e : \forall t \in |A|, t \perp\!\!\!\perp e\}
\end{array}$$

Remark 11.8. One could alternatively prefer to consider the following primitive falsity value:

$$\|A \parr B\|_V \triangleq \{(e_1,e_2) : e_1 \in \|A\| \wedge e_2 \in \|B\|\}$$

As highlighted by Dagand and Scherer [35], the design choice for primitive falsity value results in constraints on the definition of the reduction rules to make them adequate with the definitions. A short Coq development on the proof of adequacy of L$^{\parr}$ typing rules (for the propositional fragment) viewed as an evaluating machine is given to support this claim[3]. In particular, it makes very clear the impact that the choice of definition for $\|A \parr B\|_V$ has on the reduction system. ⌟

We shall now verify that the type system of L$^{\parr}$ is indeed adequate with this interpretation. We first prove the following simple lemma:

Lemma 11.9 (Substitution)**.** *Let A be a formula whose only free variable is X. For any closed formula B, if $S = \|B\|_V$, then $\|A[B/X]\|_V = \|A[\dot{S}/X]\|_V$.*

Proof. Easy induction on the structure of formulas, with the observation that the statement for primitive falsity values implies the same statement for truth values ($|A[B/X]| = |A[\dot{S}/X]|$) and falsity values ($\|A[B/X]\| = \|A[\dot{S}/X]\|$). The key case is for the atomic formula $A \equiv X$, where we easily check that:

$$\|X[B/X]\|_V = \|B\|_V = S = \|\dot{S}\|_V = \|X[\dot{S}/X]\|_V$$

□

The last step before proving adequacy consists in defining substitutions and valuations. We say that a *valuation*, which we write ρ, is a function mapping each second-order variable to a primitive falsity value $\rho(X) \in \mathcal{P}(\mathcal{V}_0)$. A *substitution*, which we write σ, is a function mapping each variable x to a closed term c and each variable α to a closed value $V \in \mathcal{V}_0$:

$$\sigma ::= \varepsilon \mid \sigma, x \mapsto t \mid \sigma, \alpha \mapsto V^+$$

We say that a substitution σ realizes a context Γ and note $\sigma \Vdash \Gamma$ when for each binding $(x : A) \in \Gamma$, $\sigma(x) \in |A|$. Similarly, we say that σ realizes a context Δ if for each binding $(\alpha : A) \in \Delta$, $\sigma(\alpha) \in \|A\|_V$.

We can now state the property of adequacy of the realizability interpretation:

Proposition 11.10 (Adequacy)**.** *Let Γ, Δ be typing contexts, ρ be a valuation and σ be a substitution such that $\sigma \Vdash \Gamma[\rho]$ and $\sigma \Vdash \Delta[\rho]$. We have:*

1. *If V^+ is a positive value such that $\Gamma \mid V^+ : A \vdash \Delta$, then $V^+[\sigma] \in \|A[\rho]\|_V$.*
2. *If t is a term such that $\Gamma \vdash t : A \mid \Delta$, then $t[\sigma] \in |A[\rho]|$.*
3. *If e is a context such that $\Gamma \mid e : A \vdash \Delta$, then $e[\sigma] \in \|A[\rho]\|$.*
4. *If c is a command such that $c : (\Gamma \vdash \Delta)$, then $c[\sigma] \in \perp\!\!\!\perp$.*

Proof. The proof is almost the same as for the proof of adequacy for the call-by-name $\lambda\mu\tilde{\mu}$-calculus. We only give some key cases which are peculiar to this setting. We proceed by induction over the typing derivations. Let σ be a substitution realizing $\Gamma[\rho]$ and $\Delta[\rho]$.

[3] See `https://www.irif.fr/~emiquey/these/coq/Real.RealLPar.html`.

- **Case** ($\vdash \neg$). Assume that we have:

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \tilde{\mu}[x].c : \neg A}\ (\vdash\neg)$$

and let $[t]$ be a term in $\|A[\rho]\|_V$, that is to say that $t \in |A[\rho]|$. We know by induction hypothesis that for any valuation $\sigma' \Vdash (\Gamma, x : A)[\rho]$, $c[\sigma'] \in \perp\!\!\!\perp$ and we want to show that $\mu[x].c[\sigma]\perp\!\!\!\perp[t]$. We have that:

$$\mu[x].c\perp\!\!\!\perp[t] \quad \to_\beta \quad c[\sigma][t/x] = c[\sigma, x \mapsto t]$$

hence it is enough by saturation to show that $c[\sigma][u/x] \in \perp\!\!\!\perp$. Since $t \in |A[\rho]|$, $\sigma[x \mapsto t] \Vdash (\Gamma, x : A)[\rho]$ and we can conclude by induction hypothesis. The cases for ($\mu\ \vdash$), ($\vdash\ \mu$) and ($\vdash\ ⅋$) proceed similarly.

- **Cases** ($\neg \vdash$). Trivial by induction hypotheses.
- **Case** ($⅋\ \vdash$). Assume that we have:

$$\frac{\Gamma \mid e_1 : A \vdash \Delta \quad \Gamma \mid u : B \vdash \Delta}{\Gamma \mid (e_1, e_2) : A ⅋ B \vdash \Delta}\ (⅋\vdash)$$

Let then t be a term in $|(A ⅋ B)[\rho]|$, to show that $\langle t \| (e_1, e_2)\rangle \in \perp\!\!\!\perp$, we proceed by anti-reduction:

$$\langle t \| (e, e')\rangle \to_\beta \langle \mu\alpha.\langle \mu\alpha'.\langle t \| (\alpha, \alpha')\rangle \| e'\rangle \| e\rangle$$

It now easy to show, using the induction hypotheses for e and e' that this command is in the pole: it suffices to show that the term $\mu\alpha.\langle \mu\alpha'.\langle t \| (\alpha, \alpha')\rangle \| e'\rangle \in |A|$, which amounts to showing that for any value $V_1 \in \|A\|_V$:

$$\langle \mu\alpha.\langle \mu\alpha'.\langle t \| (\alpha, \alpha')\rangle \| V\rangle \| {\to_\beta} \rangle \langle \mu\alpha'.\langle t \| (V, \alpha')\rangle \| e'\rangle \in \perp\!\!\!\perp$$

Again this holds by showing that for any $V' \in |B|$,

$$\langle \mu\alpha'.\langle t \| (V, \alpha')\rangle \| V'\rangle \to_\beta \langle t \| (V, V')\rangle \in \perp\!\!\!\perp$$

- **Case** ($\vdash \forall$). Trivial.
- **Case** ($\forall\ \vdash$). Assume that we have:

$$\frac{\Gamma \mid e : A[B/X] \vdash \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta}\ (\forall\vdash)$$

By induction hypothesis, we obtain that $e[\sigma] \in \|A[B/X][\rho]\|$; so that if we denote $\|B[\rho]\|_V \in \mathcal{P}(\mathcal{V}_0)$ by S, we have:

$$e[\sigma] \in \|A[\dot{S}/X]\| \subseteq \bigcup_{S \in \mathcal{P}(\mathcal{V}_0)} \|A[\dot{S}/X][\rho]\|_V^{\perp\!\!\!\perp\perp\!\!\!\perp} \subseteq \Big(\bigcup_{S \in \mathcal{P}(\mathcal{V}_0)} \|A[\dot{S}/X][\rho]\|_V\Big)^{\perp\!\!\!\perp\perp\!\!\!\perp} = \|\forall X.A[\rho]\|$$

where we make implicit use of Lemma 11.9. □

As a consequence of the former result and Proposition 11.4, we deduce that the typing rules for the encoded $\lambda\mu\tilde{\mu}$-rules also are adequate with the realizability interpretation.

Corollary 11.11. *The typing rules for $\lambda\mu\tilde{\mu}$-terms are adequate.*

In particular, this means that the realizability interpretation for $L^{⅋}$ is a particular case of the one we define for the call-by-name $\lambda\mu\tilde{\mu}$-calculus in Section 11.1.3.

11.2 Disjunctive structures

Let us examine for a minute the situation to which we arrived. First, insofar as the call-by-name machinery of the $\lambda\mu\tilde{\mu}$-calculus was embeddable into $L^{⅋}$, in particular the Krivine abstract machine for the λ_c-calculus can be recovered in this setting. Therefore, we could have used these embedding to make use of the realizability interpretation for the λ_c-calculus. Schematically, this would have corresponded to the following path:

$L^{⅋}$ - - -> (Call-by-name) $\lambda\mu\tilde{\mu}$-calculus - - -> λ_c-calculus KAM - - -> Realizability model

In particular, thinking of this construction from the point of view of implicative structures, this implies that we could have defined an implicative algebra by proceeding as follows:

$L^{⅋}$ - - -> λ_c-calculus KAM - - -> Implicative structure - - -> Implicative algebra

On the other hand, we saw in the previous section that the $L^{⅋}$ calculus was suitable for the direct definition of a realizability model. The interpretation is induced by the reduction system of $L^{⅋}$, which directly reflects the choice of connectives. Instead of embedding an arrow to obtain in the end an implicative structure, we should expect a direct algebraic counterpart for the structure of the calculus, and obtain a direct algebraic interpretation looking like:

$L^{⅋}$ - - -> Disjunctive structure - - -> Disjunctive algebra

Finally, we know that the realizability model obtained directly from the $L^{⅋}$ calculus somehow contains the realizability model that would have been constructed with the arrow. In other words, the interpretation of $L^{⅋}$ is a particular case of interpretation for a λ_c-calculus enriched with some additional structure. Therefore, we expect that, at the level of algebraic structures, any disjunctive algebra should induce an implicative algebra:

Disjunctive algebra - - -> Implicative algebra

11.2.1 Disjunctive structures

Following the rationale guiding the definition of implicative structure and algebras, we should now define the notion of *disjunctive structure.* Such a structure will then contain two internal laws to reflect the negation and the disjunction from the language of formulas. Regarding the expected commutations, as we choose negative connectives and in particular a universal quantifier, we should define commutations with respect to arbitrary meets. The following properties of the realizability interpretation for $L^{⅋}$ provides us with a safeguard for the definition to come:

Proposition 11.12 (Commutations). *In any $L^{⅋}$ realizability model (that is to say for any pole $\perp\!\!\!\perp$), the following equalities hold:*

1. *If $X \notin FV(B)$, then $\|\forall X.(A ⅋ B)\|_V = \|(\forall X.A) ⅋ B\|_V$.*
2. *If $X \notin FV(A)$, then $\|\forall X.(A ⅋ B)\|_V = \|A ⅋ (\forall X.B)\|_V$.*
3. *$\|\neg(\forall X.A)\|_V = \bigcap_{S\in\mathcal{P}(\mathcal{V}_0)} \|\neg A\{X := \dot{S}\}\|_V$*

Proof. 1. Assume the $X \notin FV(B)$, then we have:

$$\begin{aligned}
\|\forall X.(A ⅋ B)\|_V &= \bigcup_{S\in\mathcal{P}(\mathcal{V}_0)} \|A\{X := \dot{S}\} ⅋ B\|_V \\
&= \bigcup_{S\in\mathcal{P}(\mathcal{V}_0)} \{(V_1, V_2) : V_1 \in \|A\{X := \dot{S}\}\|_V \wedge V_2 \in \|B\|_V\} \\
&= \{(V_1, V_2) : V_1 \in \bigcup_{S\in\mathcal{P}(\mathcal{V}_0)} \|A\{X := \dot{S}\}\|_V \wedge V_2 \in \|B\|_V\} \\
&= \{(V_1, V_2) : V_1 \in \|\forall X.A\|_V \wedge V_2 \in \|B\|\} \quad = \quad \|(\forall X.A) ⅋ B\|_V
\end{aligned}$$

2. Identical.
3. The proof is again a simple unfolding of the definitions:

$$\begin{aligned}\|\neg(\forall X.A)\|_V = \{[t] : t \in |\forall X.A|\} &= \{[t] : t \in \bigcap_{S\in\mathcal{P}(\mathcal{V}_0)} |A\{X := \dot{S}\}|\} \\ &= \bigcap_{S\in\mathcal{P}(\mathcal{V}_0)} \{[t] : t \in |A\{X := \dot{S}\}|]\} = \bigcap_{S\in\mathcal{P}(\mathcal{V}_0)} \|\neg A\{X := \dot{S}\}\|_V\end{aligned}$$

□

In terms of algebraic structure, the previous proposition advocates for the following equalities:

1. $\curlywedge_{b\in B}(a \parr b) = a \parr (\curlywedge_{b\in B} b)$ 2. $\curlywedge_{b\in B}(b \parr a) = (\curlywedge_{b\in B} b) \parr a$ 3. $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

(recall that the order is defined as the reversed inclusion of primitive falsity values (whence $\cap$ is $\curlyvee$) and that the $\forall$ quantifier is interpreted by $\curlywedge$.)

Definition• 11.13 (Disjunctive structure). A *disjunctive structure* is a complete meet-semilattice $(\mathcal{A}, \preccurlyeq)$ equipped with a binary operation $(a, b) \mapsto a \parr b$, called the *disjunction of* $\mathcal{A}$ together with a unary operation $a \mapsto \neg a$ called the *negation of* $\mathcal{A}$, which fulfill the following axioms:

1. Negation is anti-monotonic in the sense that for all $a, a' \in \mathcal{A}$:

   (Contravariance) if $a \preccurlyeq a'$ then $\neg a' \preccurlyeq \neg a$

2. Disjunction is monotonic in the sense that for all $a, a', b, b' \in \mathcal{A}$:

   (Variance) if $a \preccurlyeq a'$ and $b \preccurlyeq b'$ then $a \parr b \preccurlyeq a' \parr b'$

3. Arbitrary meets distributes over both operands of disjunction, in the sense that for all $a \in \mathcal{A}$ and for all subsets $B \subseteq \mathcal{A}$:

   (Distributivity) $\curlywedge_{b\in B}(a \parr b) = a \parr (\curlywedge_{b\in B} b)$ $\curlywedge_{b\in B}(b \parr a) = (\curlywedge_{b\in B} b) \parr a$

4. Negation of the meet of set is equal to the join of the set of negated elements, in the sense that for all subsets $A \subseteq \mathcal{A}$:

   (Commutation) $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

┘

As in the case of implicative structures, the commutation laws imply the value of the internal laws when applied to the maximal element $\top$:

Proposition 11.14. *If $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure, then the following hold for all $a \in \mathcal{A}$:*

1.• $\top \parr a = \top$ *2.•* $a \parr \top = \top$ *3.•* $\neg\top = \bot$

Proof. Using Proposition 9.4 and the axioms of disjunctive structures, we prove:

1. for all $a \in \mathcal{A}$, $\top \parr a = (\curlywedge \emptyset) \parr a = \curlywedge_{x,a\in\mathcal{A}}\{x \parr a : x \in \emptyset\} = \curlywedge \emptyset = \top$
2. for all $a \in \mathcal{A}$, $a \parr \top = a \parr (\curlywedge \emptyset) = \curlywedge_{x,a\in\mathcal{A}}\{a \parr x : x \in \emptyset\} = \curlywedge \emptyset = \top$
3. $\neg\top = \neg(\curlywedge \emptyset) = \curlyvee_{x\in\mathcal{A}}\{\neg x : x \in \emptyset\} = \curlyvee \emptyset = \bot$

□

11.2.2 Examples of disjunctive structures

11.2.2.1 Dummy structure

Example• 11.15 (Dummy disjunctive structure)**.** Given a complete lattice $(\mathcal{L}, \preccurlyeq)$, the following definitions give rise to a dummy structure that fulfills the axioms of Definition 11.13:

$$a \parr b \triangleq \top \qquad \neg a \triangleq \bot \qquad (\forall a, b \in \mathcal{A})$$

The verification of the different axioms is straightforward. ⌟

11.2.2.2 Complete Boolean algebra

Example• 11.16 (Complete Boolean algebras)**.** Let $\mathcal{B}$ be a complete Boolean algebra. It encompasses a disjunctive structure, that is defined by:

- $\mathcal{A} \triangleq \mathcal{B}$
- $a \preccurlyeq b \triangleq a \preccurlyeq b$
- $a \parr b \triangleq a \vee b$
- $\neg a \triangleq \neg a$

$(\forall a, b \in \mathcal{A})$

The different axioms are direct consequences of Proposition 9.7. ⌟

11.2.3 Disjunctive structure of classical realizability

If we abstract the structure of the realizability interpretation of $\mathrm{L}^{\parr}$ (see Section 11.1.3), it is a structure of the form $(\mathcal{T}_0, \mathcal{E}_0, \mathcal{V}_0, (\cdot,\cdot), [\cdot], \perp\!\!\!\perp)$ where $\mathcal{V}_0 \subseteq \mathcal{E}_0$ is the distinguished subset of (positive) values, $(\cdot,\cdot)$ is a binary map from $\mathcal{E}_0^2$ to $\mathcal{E}_0$ (whose restriction to $\mathcal{V}_0$ has values in $\mathcal{V}_0$), $[\cdot]$ is an operation from $\mathcal{T}_0$ to $\mathcal{V}_0$, and $\perp\!\!\!\perp \subseteq \mathcal{T}_0 \times \mathcal{E}_0$ is a relation[4]. From this sextuple, we can define:

- $\mathcal{A} \triangleq \mathcal{P}(\mathcal{V}_0)$
- $a \preccurlyeq b \triangleq a \supseteq b$
- $a \parr b \triangleq (a, b) = \{(V_1, V_2) : V_1 \in a \wedge V_2 \in b\}$
- $\neg a \triangleq [a^{\perp\!\!\!\perp}] = \{[t] : t \in a^{\perp\!\!\!\perp}\}$

$(\forall a, b \in \mathcal{A})$

Proposition 11.17. *The quadruple $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure.*

Proof. We show that the axioms of Definition 11.13 are satisfied.

1. (Contravariance) Let $a, a' \in \mathcal{A}$, such that $a \preccurlyeq a'$ ie $a' \subseteq a$. Then $a^{\perp\!\!\!\perp} \subseteq a'^{\perp\!\!\!\perp}$ and thus
$$\neg a = \{[t] : t \in a^{\perp\!\!\!\perp}\} \subseteq \{[t] : t \in a'^{\perp\!\!\!\perp}\} = \neg a'$$
i.e. $\neg a' \preccurlyeq \neg a$.
2. (Covariance) Let $a, a', b, b' \in \mathcal{A}$ such that $a' \subseteq a$ and $b' \subseteq b$. Then we have
$$a \parr b = \{(V_1, V_2) : V_1 \in a \wedge V_2 \in b\} \subseteq \{(V_1, V_2) : V_1 \in a' \wedge V_2 \in b'\} = a' \parr b'$$
i.e. $a \parr b \preccurlyeq a' \parr b'$.
3. (Distributivity) Let $a \in \mathcal{A}$ and $B \subseteq \mathcal{A}$, we have:
$$\bigcurlywedge_{b \in B} (a \parr b) = \bigcurlywedge_{b \in B} \{(V_1, V_2) : V_1 \in a \wedge e_2 \in b\} = \{(V_1, V_2) : V_1 \in a \wedge V_2 \in \bigcurlywedge_{b \in B} b\} = a \parr (\bigcurlywedge_{b \in B} b)$$
4. (Commutation) Let $B \subseteq \mathcal{A}$, we have (recall that $\bigcurlyvee_{b \in B} b = \bigcap_{b \in B} b$):
$$\bigcurlyvee_{b \in B} (\neg b) = \bigcurlyvee_{b \in B} \{[t] : t \in b^{\perp\!\!\!\perp}\} = \{[t] : t \in \bigcurlyvee_{b \in B} b^{\perp\!\!\!\perp}\} = \{[t] : t \in (\bigcurlywedge_{b \in B} b)^{\perp\!\!\!\perp}\} = \neg(\bigcurlywedge_{b \in B} b)$$

□

Remark 11.18. The same definitions taking $\mathcal{A} \triangleq \mathcal{P}(\mathcal{E}_0)$ instead of $\mathcal{P}(\mathcal{V}_0)$ also satisfy the same properties. ⌟

[4] We could also abstract the different properties axiomatizing the pole and the different sets to obtain some kind of "abstract $\mathrm{L}^{\parr}$ structure", but there is no point in doing this, since it would be less general than the notion of disjunctive structure anyway.

11.2.4 Interpreting $L^{\parr}$

Following the interpretation of the λ-calculus in implicative structures, we shall now see how $L^{\parr}$ commands can be recovered from disjunctive structures. From now on, we assume given a disjunctive structure $(\mathcal{A}, \preccurlyeq, \parr, \neg)$.

11.2.4.1 Commands

We shall begin with the interpretation of commands. This poses a novel difficulty with respect to the definition of λ-terms in implicative structures. Indeed, we are looking for an interpretation of terms and contexts, that is to say for both the realizers and the opponents (while in implicative structures we only interpreted realizers). Therefore, we first need to understand what it means for a command (in terms of the disjunctive structure) to be well-formed, *i.e.* to be in the pole. For this, we follow the intuition of the passage from a ${}^{\mathcal{K}}$OCA to an AKS (see Proposition 9.34). This translation indeed defines the embedding of a one-sided structure (the ${}^{\mathcal{K}}$OCA, with a set $\mathcal{A}$ of combinators) to a two-sided structure (the AKS, with a set Λ of realizers and a set Π of opponents). The induced AKS is indeed defined with the same domain for terms and stacks $\Lambda = \Pi = \mathcal{A}$. In this setting, the pole $\perp\!\!\!\perp$ is simply defined as the order relation on the ${}^{\mathcal{K}}$OCA: a term $t \in \Lambda$ is orthogonal to a stack $\pi \in \Pi$ if $t \preccurlyeq \pi$. This definition is in accordance with the intuition that the order reflect the quantity of information that a term (resp. stack, formula, etc...) carries: if the term t can defeat its opponent π, *i.e.* if $t \star \pi \in \perp\!\!\!\perp$, it means indeed that t is more defined than π.

We thus define the *commands* of the disjunctive structure $\mathcal{A}$ as the pair (a, b) (which we continue to write $\langle a \| b \rangle$) with $a, b \in \mathcal{A}$, and we define the pole $\perp\!\!\!\perp$ as the ordering relation $\preccurlyeq$. We write $C_{\mathcal{A}} = \mathcal{A} \times \mathcal{A}$ for the set of commands in $\mathcal{A}$ and $(a, b) \in \perp\!\!\!\perp$ for $a \preccurlyeq b$. Besides, we define an ordering on commands which extends the intuition that the order reflect the “definedness” of objects: given two commands c, c' in $C_{\mathcal{A}}$, we say that c is lower than c' and we write $c \trianglelefteq c'$ if $c \in \perp\!\!\!\perp$ implies that $c' \in \perp\!\!\!\perp$. It is straightforward to check that:

Proposition• 11.19. *The relation $\trianglelefteq$ is a preorder.*

Besides, the relation $\trianglelefteq$ verifies the following property of variance with respect to the order $\preccurlyeq$:

Proposition• 11.20 (Commands ordering)**.** *For all $t, t', \pi, \pi' \in \mathcal{A}$, if $t \preccurlyeq t'$ and $\pi' \preccurlyeq \pi$, then $\langle t \| \pi \rangle \trianglelefteq \langle t' \| \pi' \rangle$.*

Proof. Trivial by transitivity of $\preccurlyeq$. □

Finally, it is worth noting that meets are covariant with respect to $\trianglelefteq$ and $\preccurlyeq$, while joins are contravariant:

Lemma• 11.21. *If c and c' are two functions associating to each $a \in \mathcal{A}$ the commands $c(a)$ and $c'(a)$ such that $c(a) \trianglelefteq c'(a)$, then we have:*

$$\bigcurlywedge_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\} \preccurlyeq \bigcurlywedge_{a \in \mathcal{A}} \{a : c'(a) \in \perp\!\!\!\perp\} \qquad\qquad \bigcurlyvee_{a \in \mathcal{A}} \{a : c'(a) \in \perp\!\!\!\perp\} \preccurlyeq \bigcurlyvee_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

Proof. Assume c, c' are such that for all $a \in \mathcal{A}$, $ca \trianglelefteq c'a$. Then it is clear that by definition we have the inclusion $\{a \in \mathcal{A} : c(a) \in \perp\!\!\!\perp\} \subseteq \{a \in \mathcal{A} : c'(a) \in \perp\!\!\!\perp\}$, whence the expected results. □

11.2.4.2 Contexts

We are now ready to define the interpretation of $L^{\parr}$ contexts in the disjunctive structure $\mathcal{A}$. The interpretation for the contexts corresponding to the connectives is very natural:

Definition• 11.22 (Pairing)**.** For all $a, b \in \mathcal{A}$, we let $(a,b) \triangleq a \parr b$. ⌟

Definition• 11.23 (Boxing)**.** For all $a \in \mathcal{A}$, we let $[a] \triangleq \neg a$. ⌟

Note that with these definitions, the encodings of pairs and boxes directly inherit of the properties of the internal law $\parr$ and $\neg$ in disjunctive structures. As for the binder $\mu x.c$, which we write $\tilde{\mu}^+ c$, it should be defined in such a way that if c is a function mapping each $a \in \mathcal{A}$ to a command $c(a) \in C_{\mathcal{A}}$, then $\mu^+.c$ should be "compatible" with any a such that $c(a)$ is well-formed (*i.e.* $c(a) \in \perp\!\!\!\perp$). As it belongs to the side of opponents, the "compatibility" means that it should be greater than any such a, and we thus define it as a join.

Definition• 11.24 (μ^+)**.** For all $c : \mathcal{A} \to C_{\mathcal{A}}$, we define:

$$\mu^+.c := \bigvee_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

⌟

These definitions enjoy the following properties with respect to the β-reduction and the η-expansion (compare with Proposition 10.17):

Proposition 11.25 (Properties of μ^+)**.** *For all functions $c, c' : \mathcal{A} \to C_{\mathcal{A}}$, the following hold:*

1.• *If for all $a \in \mathcal{A}$, $c(a) \trianglelefteq c'(a)$, then $\mu^+.c' \preccurlyeq \mu^+.c$* *(Variance)*
2.• *For all $t \in \mathcal{A}$, then $\langle t \| \mu^+.c\rangle \trianglelefteq c(t)$* *($\beta$-reduction)*
3.• *For all $e \in \mathcal{A}$, then $t = \mu^+.(a \mapsto \langle a \| e\rangle)$* *($\eta$-expansion)*

Proof. 1. Direct consequence of Proposition 11.21.

2,3. Trivial by definition of μ^+.

□

Remark 11.26 (Subject reduction)**.** The β-reduction $c \to_\beta c'$ is reflected by the ordering relation $c \trianglelefteq c'$, which reads *"if c is well-formed, then so is c′"*. In other words, this corresponds to the usual property of subject reduction. In the sequel, we will see that β-reduction rules of $\mathrm{L}^{\parr}$ will always been reflected in this way through the embedding in disjunctive structures. ⌟

11.2.4.3 Terms

Dually to the definitions of (positive) contexts μ^+ as a join, we define the embedding of (negative) terms, which are all binders, by arbitrary meets:

Definition• 11.27 (μ^-)**.** For all $c : \mathcal{A} \to C_{\mathcal{A}}$, we define:

$$\mu^-.c := \bigwedge_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

⌟

Definition• 11.28 ($\mu^{()}c$)**.** For all $c : \mathcal{A}^2 \to C_{\mathcal{A}}$, we define:

$$\mu^{()}.c := \bigwedge_{a,b \in \mathcal{A}} \{a \parr b : c(a,b) \in \perp\!\!\!\perp\}$$

⌟

Definition• 11.29 ($\mu^{[]}$)**.** For all $c : \mathcal{A} \to C_{\mathcal{A}}$, we define:

$$\mu^{[]}.c := \bigwedge_{a \in \mathcal{A}} \{\neg a : c(a) \in \perp\!\!\!\perp\}$$

⌟

These definitions also satisfy some variance properties with respect to the preorder $\trianglelefteq$ and the order relation $\preccurlyeq$, namely, negative binders for variable ranging over positive contexts are covariant, while negative binders intended to catch negative terms are contravariant.

Proposition 11.30 (Variance). *For any functions c, c' with the corresponding arities, the following hold:*

1. *If $c(a) \trianglelefteq c'(a)$ for all $a \in \mathcal{A}$, then $\mu^{-}.c \preccurlyeq \mu^{-}.c'$*
2. *If $c(a,b) \trianglelefteq c'(a,b)$ for all $a,b \in \mathcal{A}$, then $\mu^{()}.c \preccurlyeq \mu^{()}.c'$*
3. *If $c(a) \trianglelefteq c'(a)$ for all $a \in \mathcal{A}$, then $\mu^{[]}.c' \preccurlyeq \mu^{[]}.c$*

Proof. Direct consequences of Proposition 11.21. □

The η-expansion is also reflected as usual by the ordering relation $\preccurlyeq$:

Proposition 11.31 (η-expansion). *For all $t \in \mathcal{A}$, the following holds:*

1. $t = \mu^{-}.(a \mapsto \langle t \| a \rangle)$
2. $t \preccurlyeq \mu^{()}.(a,b \mapsto \langle t \| (a,b) \rangle)$
3. $t \preccurlyeq \mu^{[]}.(a \mapsto \langle t \| [a] \rangle)$

Proof. Trivial from the definitions. □

The β-reduction is reflected by the preorder $\trianglelefteq$:

Proposition 11.32 (β-reduction). *For all $e, e_1, e_2, t \in \mathcal{A}$, the following holds:*

1. $\langle \mu^{-}.c \| e \rangle \trianglelefteq c(e)$
2. $\langle \mu^{()}.c \| (e_1, e_2) \rangle \trianglelefteq c(e_1, e_2)$
3. $\langle \mu^{[]}.c \| [t] \rangle \trianglelefteq c(t)$

Proof. Trivial from the definitions. □

Finally, we call a *$L^{\invamp}$ term with parameters in $\mathcal{A}$* (resp. context, command) any $\mathrm{L}^{\invamp}$ term (possibly) enriched with constants taken in the set $\mathcal{A}$. Commands with parameters are equipped with the same rules of reduction as in $\mathrm{L}^{\invamp}$, considering parameters as inert constants. To every closed $\mathrm{L}^{\invamp}$ term t (resp. context e,command c) we associate an element $t^{\mathcal{A}}$ (resp. $e^{\mathcal{A}}$, $c^{\mathcal{A}}$) of $\mathcal{A}$, defined by induction on the structure of t as follows:

Contexts :

$$\begin{array}{rcl} a^{\mathcal{A}} & \triangleq & a \\ (e_1,e_2)^{\mathcal{A}} & \triangleq & (e_1^{\mathcal{A}}, e_2^{\mathcal{A}}) \\ [t]^{\mathcal{A}} & \triangleq & [t^{\mathcal{A}}] \\ (\mu x.c)^{\mathcal{A}} & \triangleq & \mu^{-}(a \mapsto (c[x:=a])^{\mathcal{A}}) \end{array}$$

Terms :

$$\begin{array}{rcl} a^{\mathcal{A}} & \triangleq & a \\ (\mu\alpha.c)^{\mathcal{A}} & \triangleq & \mu^{-}(a \mapsto (c[\alpha:=a])^{\mathcal{A}}) \\ (\mu(\alpha_1,\alpha_2).c)^{\mathcal{A}} & \triangleq & \mu^{()}(a,b \mapsto (c[\alpha_1:=a,\alpha_2:=b])^{\mathcal{A}}) \\ (\mu[x].c)^{\mathcal{A}} & \triangleq & \mu^{[]}(a \mapsto (c[x:=a])^{\mathcal{A}}) \end{array}$$

Commands: $$\langle t \| e \rangle^{\mathcal{A}} \triangleq \langle t^{\mathcal{A}} \| e^{\mathcal{A}} \rangle)$$

In particular, this definition has the nice property of making the pole $\perp\!\!\!\perp$ (*i.e.* the order relation $\preccurlyeq$) closed under anti-reduction, as reflected by the following property of $\trianglelefteq$:

Proposition 11.33 (Subject reduction). *For any closed commands c_1, c_2 of $L^{\invamp}$, if $c_1 \to_\beta c_2$ then $c_1^{\mathcal{A}} \trianglelefteq c_2^{\mathcal{A}}$,* i.e. *if $c_1^{\mathcal{A}}$ belongs to $\perp\!\!\!\perp$ then so does $c_2^{\mathcal{A}}$.*

Proof. Direct consequence of Propositions 11.25 and 12.22. □

11.2.5 Adequacy

We shall now prove that the interpretation of $\mathrm{L}^{\parr}$ is adequate with respect to its type system. Again, we extend the syntax of formulas to define second-order formulas with parameters by:

$$A, B ::= a \mid X \mid \neg A \mid A \parr B \mid \forall X.A \qquad (a \in \mathcal{A})$$

This allows us to embed closed formulas with parameters into the disjunctive structure $\mathcal{A}$. The embedding is trivially defined by:

$$\begin{array}{rcll} a^{\mathcal{A}} & \triangleq & a & \text{(if } a \in \mathcal{A}) \\ (\neg A)^{\mathcal{A}} & \triangleq & \neg A^{\mathcal{A}} & \\ (A \parr B)^{\mathcal{A}} & \triangleq & A^{\mathcal{A}} \parr B^{\mathcal{A}} & \\ (\forall X.A)^{\mathcal{A}} & \triangleq & \curlywedge_{a\in\mathcal{A}} (A\{X := a\})^{\mathcal{A}} & \end{array}$$

As for the adequacy of the interpretation for the second-order λ_c-calculus, we define substitutions, which we write σ, as functions mapping variables (of terms, contexts and types) to element of $\mathcal{A}$:

$$\sigma ::= \varepsilon \mid \sigma[x \mapsto a] \mid \sigma[\alpha \mapsto a] \mid \sigma[X \mapsto a] \qquad (a \in \mathcal{A},\ x, X \text{ variables})$$

In the spirit of the proof of adequacy in classical realizability, we say that a substitution σ realizes a typing context Γ, which write $\sigma \Vdash \Gamma$, if for all bindings $(x : A) \in \Gamma$ we have $\sigma(x) \preccurlyeq (A[\sigma])^{\mathcal{A}}$. Dually, we say that σ realizes Δ if for all bindings $(\alpha : A) \in \Delta$, we have $\sigma(\alpha) \succcurlyeq (A[\sigma])^{\mathcal{A}}$. We can now prove

Theorem 11.34 (Adequacy)**.** *The typing rules of $\mathrm{L}^{\parr}$ (Figure 11.1) are adequate with respect to the interpretation of terms (contexts,commands) and formulas. Indeed, for all contexts Γ, Δ, for all formulas with parameters A then for all substitutions σ such that $\sigma \Vdash \Gamma$ and $\sigma \Vdash \Delta$, we have:*

1. *for any term t, if $\Gamma \vdash t : A \mid \Delta$, then $(t[\sigma])^{\mathcal{A}} \preccurlyeq A[\sigma]^{\mathcal{A}}$;*
2. *for any context e, if $\Gamma \mid e : A \vdash \Delta$, then $(e[\sigma])^{\mathcal{A}} \succcurlyeq A[\sigma]^{\mathcal{A}}$;*
3. *for any command c, if $c : (\Gamma \vdash \Delta)$, then $(c[\sigma])^{\mathcal{A}} \in \perp\!\!\!\perp$.*

Proof. By induction over the typing derivations.

- **Case** (Cut). Assume that we have:

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \| e \rangle : \Gamma \vdash \Delta}\ (\text{Cut})$$

By induction hypotheses, we have $(t[\sigma])^{\mathcal{A}} \preccurlyeq A[\sigma]^{\mathcal{A}}$ and $(e[\sigma])^{\mathcal{A}} \succcurlyeq A[\sigma]^{\mathcal{A}}$. By transitivity of the relation $\preccurlyeq$, we deduce that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (e[\sigma])^{\mathcal{A}}$, so that $(\langle t \| e \rangle[\sigma])^{\mathcal{A}} \in \perp\!\!\!\perp$.

- **Case** ($\vdash ax$). Straightforward, since if $(x : A) \in \Gamma$, then $(x[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma])^{\mathcal{A}}$. The case $(ax \vdash)$ is identical.

- **Case** ($\vdash \mu$). Assume that we have:

$$\frac{c : \Gamma \vdash \Delta, \alpha : A}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}\ (\vdash\mu)$$

By induction hypothesis, we have that $(c[\sigma, \alpha \mapsto (A[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Then, by definition we have:

$$((\mu\alpha.c)[\sigma])^{\mathcal{A}} = (\mu\alpha.(c[\sigma]))^{\mathcal{A}} = \bigcurlywedge_{b\in\mathcal{A}} \{b : (c[\sigma, \alpha \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \preccurlyeq (A[\sigma])^{\mathcal{A}}$$

- **Case** ($\mu\ \vdash$). Similarly, assume that we have:

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \mid \mu x.c : A \vdash \Delta}\ (\mu \vdash)$$

By induction hypothesis, we have that $(c[\sigma, x \mapsto (A[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Therefore, we have:

$$((\mu x.c)[\sigma])^{\mathcal{A}} = (\mu x.(c[\sigma]))^{\mathcal{A}} = \bigcurlyvee_{b \in \mathcal{A}} \{b : (c[\sigma, x \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \succcurlyeq (A[\sigma])^{\mathcal{A}}\,.$$

- **Case** ($⅋\ \vdash$). Assume that we have:

$$\frac{\Gamma \mid e_1 : A_1 \vdash \Delta \quad \Gamma \mid e_2 : A_2 \vdash \Delta}{\Gamma \mid (e_1, e_2) : A_1 ⅋ A_2 \vdash \Delta}\ (⅋ \vdash)$$

By induction hypotheses, we have that $(e_1[\sigma])^{\mathcal{A}} \succcurlyeq (A_1[\sigma])^{\mathcal{A}}$ and $(e_2[\sigma])^{\mathcal{A}} \succcurlyeq (A_2[\sigma])^{\mathcal{A}}$. Therefore, by monotonicity of the $⅋$ operator, we have:

$$((e_1, e_2)[\sigma])^{\mathcal{A}} = (e_1[\sigma], e_2[\sigma])^{\mathcal{A}} = (e_1[\sigma])^{\mathcal{A}} ⅋ (e_2[\sigma])^{\mathcal{A}} \succcurlyeq (A_1[\sigma])^{\mathcal{A}} ⅋ (A_2[\sigma])^{\mathcal{A}}\,.$$

- **Case** ($\vdash\ ⅋$). Assume that we have:

$$\frac{c : \Gamma \vdash \Delta, \alpha_1 : A_1, \alpha_2 : A_2}{\Gamma \vdash \mu(\alpha_1, \alpha_2).c : A_1 ⅋ A_2 \mid \Delta}\ (\vdash ⅋)$$

By induction hypothesis, we get that $(c[\sigma, \alpha_1 \mapsto (A_1[\sigma])^{\mathcal{A}}, \alpha_2 \mapsto (A_2[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Then by definition we have

$$((\mu(\alpha_1, \alpha_2).c)[\sigma])^{\mathcal{A}} = \bigcurlywedge_{a, b \in \mathcal{A}} \{a ⅋ b : (c[\sigma, \alpha_1 \mapsto a, \alpha_2 \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \preccurlyeq (A_1[\sigma])^{\mathcal{A}} ⅋ (A_2[\sigma])^{\mathcal{A}}\,.$$

- **Case** ($\neg\ \vdash$). Assume that we have:

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \mid [t] : \neg A \vdash \Delta}\ (\neg \vdash)$$

By induction hypothesis, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma])^{\mathcal{A}}$. Then by definition of $[\,]^{\mathcal{A}}$ and covariance of the $\neg$ operator, we have:

$$([t[\sigma]])^{\mathcal{A}} = \neg(t[\sigma])^{\mathcal{A}} \succcurlyeq \neg(A[\sigma])^{\mathcal{A}}.$$

- **Case** ($\vdash\ \neg$). Assume that we have:

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta}\ (\vdash \neg)$$

By induction hypothesis, we have that $(c[\sigma, x \mapsto (A[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Therefore, we have:

$$((\mu[x].c)[\sigma])^{\mathcal{A}} = (\mu[x].(c[\sigma]))^{\mathcal{A}} = \bigcurlywedge_{b \in \mathcal{A}} \{\neg b : (c[\sigma, x \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \preccurlyeq \neg(A[\sigma])^{\mathcal{A}}.$$

- **Case** ($\forall\ \vdash$) . Assume that we have:

$$\frac{\Gamma \vdash e : A\{X := B\} \mid \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta}\ (\forall \vdash)$$

By induction hypothesis, we have that $(e[\sigma])^{\mathcal{A}} \succcurlyeq ((A\{X := B\})[\sigma])^{\mathcal{A}} = (A[\sigma, X \mapsto (B[\sigma])^{\mathcal{A}}])^{\mathcal{A}}$. Therefore, we have that $(e[\sigma])^{\mathcal{A}} \succcurlyeq (A[\sigma, X \mapsto (B[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \succcurlyeq \bigcurlywedge_{b \in \mathcal{A}} \{A\{X := b\}[\sigma]^{\mathcal{A}}\}$.

- **Case** $(\vdash \forall)$. Similarly, assume that we have:

$$\frac{\Gamma \vdash t : A \mid \Delta \quad X \notin FV(\Gamma,\Delta)}{\Gamma \vdash t : \forall X.A}\ (\vdash\forall)$$

By induction hypothesis, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma,X \mapsto b])^{\mathcal{A}}$ for any $b \in A$. Therefore, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq \curlywedge_{b\in\mathcal{A}}(A\{X := b\}[\sigma]^{\mathcal{A}})$. □

11.3 From disjunctive to implicative structures

11.3.1 The induced implicative structure

Recall that the implication is defined in terms of the disjunction and the negation by:

$$a \overset{\parr}{\rightarrow} b \triangleq \neg a \parr b$$

This definition can be reflected at the level of disjunctive structures in the sense that it directly induces an implicative structure:

Proposition• 11.35. *If $(\mathcal{A},\preccurlyeq,\parr,\neg)$ is a disjunctive structure, then $(\mathcal{A},\preccurlyeq,\overset{\parr}{\rightarrow})$ is an implicative structure.*

Proof. We need to show that the definition of the arrow fulfills the expected axioms:

1. (Variance) Let $a,b,a',b' \in \mathcal{A}$ be such that $a' \preccurlyeq a$ and $b \preccurlyeq b'$, then we have:

$$a \overset{\parr}{\rightarrow} b = \neg a \parr b \preccurlyeq \neg a' \parr b' = a' \overset{\parr}{\rightarrow} b'$$

since $\neg a \preccurlyeq \neg a'$ by contra-variance of the negation and $b \preccurlyeq b'$.

2. (Distributivity) Let $a \in \mathcal{A}$ and $B \subseteq \mathcal{A}$, then we have:

$$\curlywedge_{b\in B}(a \overset{\parr}{\rightarrow} b) = \curlywedge_{b\in B}(\neg a \parr b) = \neg a \parr (\curlywedge_{b\in B} b) = a \overset{\parr}{\rightarrow} (\curlywedge_{b\in B} b)$$

by distributivity of the infimum over the disjunction.

□

Therefore, we can again define for all a,b of $\mathcal{A}$ the application ab as well as the abstraction λf for any function f from $\mathcal{A}$ to $\mathcal{A}$;

$$ab \triangleq \curlywedge\{c \in \mathcal{A} : a \preccurlyeq b \overset{\parr}{\rightarrow} c\} \qquad\qquad \lambda f \triangleq \curlywedge_{a\in\mathcal{A}}(a \overset{\parr}{\rightarrow} fa)$$

We get for free the properties of these encodings in implicative structures:

Proposition 10.15 (Properties of abstraction and application)**.** *The following properties hold for all $a,a',b,b',c \in \mathcal{A}$ and for all $f,g : \mathcal{A} \to \mathcal{A}$,*

1.• *If $a \preccurlyeq a'$ and $b \preccurlyeq b'$, then $ab \preccurlyeq a'b'$.* *(Monotonicity of application)*
2.• *If $f(a) \preccurlyeq g(a)$ for all $a \in \mathcal{A}$, then $\lambda f \preccurlyeq \lambda g$.* *(Monotonicity of abstraction)*
3.• *$(\lambda f)a \preccurlyeq fa$.* *($\beta$-reduction)*
4.• *$a \preccurlyeq \lambda(x \mapsto ax)$.* *($\eta$-expansion)*
5.• *If $ab \preccurlyeq c$ then $a \preccurlyeq b \overset{\parr}{\rightarrow} c$.* *(Adjunction)*

11.3.2 Interpretation of the λ-calculus

Up to this point, we defined two ways of interpreting a λ-term into a disjunctive structures, either through the implicative structure which is induced by the disjunctive one, or by embedding into the $L^{\parr}$-calculus which is then interpreted within the disjunctive structure. As a sanity check, we verify that both coincide.

Lemma• 11.36. *The shorthand $\mu([x],\alpha).c$ is interpreted in $\mathcal{A}$ by:*

$$(\mu([x],\alpha).c)^{\mathcal{A}} = \bigcurlywedge_{a,b\in\mathcal{A}} \{(\neg a) \parr b : c[x := a, \alpha := b] \in \preccurlyeq\}$$

Proof.

$$\begin{aligned}
\mu([x],\alpha).c)^{\mathcal{A}} &= (\mu(x_0,\alpha).\langle \mu[x].c \| x_0\rangle)^{\mathcal{A}} \\
&= \bigcurlywedge_{a',b\in\mathcal{A}} \{a' \parr b : (\langle \mu[x].c[\alpha := b] \| a'\rangle)^{\mathcal{A}} \in \preccurlyeq\} \\
&= \bigcurlywedge_{a',b\in\mathcal{A}} \{a' \parr b : (\bigcurlywedge_{a\in\mathcal{A}} \{\neg a : c^{\mathcal{A}}[x := a, \alpha := b] \in \preccurlyeq\} \preccurlyeq a'\} \\
&= \bigcurlywedge_{a,b\in\mathcal{A}} \{(\neg a) \parr b : c^{\mathcal{A}}[x := a, \alpha := b] \in \preccurlyeq\}
\end{aligned}$$

□

Proposition 11.37 (λ-calculus). *Let $\mathcal{A}^{\parr} = (\mathcal{A}, \preccurlyeq, \parr, \neg)$ be a disjunctive structure, and $\mathcal{A}^{\rightarrow} = (\mathcal{A}, \preccurlyeq, \xrightarrow{\parr})$ the implicative structure it canonically defines, we write ι for the corresponding inclusion. Let t be a closed λ-term (with parameter in $\mathcal{A}$), and $[\![t]\!]$ his embedding in $L^{\parr}$. Then we have*

$$\iota(t^{\mathcal{A}^{\rightarrow}}) = [\![t]\!]^{\mathcal{A}^{\parr}}$$

where $t^{\mathcal{A}^{\rightarrow}}$ (resp. $t^{\mathcal{A}^{\parr}}$) is the interpretation of t within $\mathcal{A}^{\rightarrow}$ (resp. $\mathcal{A}^{\parr}$).

In other words, this proposition expresses the fact that the following diagram commutes:

$$\begin{tikzcd}
\lambda\text{-calculus} \arrow[r, "{[\![\ ]\!]}"] \arrow[d, "{[\ ]^{\mathcal{A}^{\rightarrow}}}"'] & L^{\parr} \arrow[d, "{[\ ]^{\mathcal{A}^{\parr}}}"] \\
(\mathcal{A}^{\rightarrow}, \preccurlyeq, \rightarrow) \arrow[r, "\iota"] & (\mathcal{A}^{\parr}, \preccurlyeq, \parr, \neg)
\end{tikzcd}$$

Proof. By induction over the structure of terms.

- **Case** a for some $a \in \mathcal{A}^{\parr}$. This case is trivial as both terms are equal to a.

- **Case** $\lambda x.u$. We have $[\![\lambda x.u]\!] = \mu([x],\alpha).\langle [\![t]\!] \| \alpha\rangle$ and

$$\begin{aligned}
(\mu([x],\alpha).\langle [\![t]\!] \| \alpha\rangle)^{\mathcal{A}^{\parr}} &= \bigcurlywedge_{a,b\in\mathcal{A}} \{\neg a \parr b : ([\![t[x := a]]\!]^{\mathcal{A}^{\parr}}, b) \in \perp\!\!\!\perp\} \\
&= \bigcurlywedge_{a,b\in\mathcal{A}} \{\neg a \parr b : [\![t[x := a]]\!]^{\mathcal{A}^{\parr}} \preccurlyeq b\} \\
&= \bigcurlywedge_{a\in\mathcal{A}} (\neg a \parr [\![t[x := a]]\!]^{\mathcal{A}^{\parr}})
\end{aligned}$$

On the other hand,

$$\iota([\lambda x.t]^{\mathcal{A}^{\to}}) = \iota(\bigcurlywedge_{a\in\mathcal{A}} (a \xrightarrow{\parr} (t[x := a])^{\mathcal{A}^{\to}})) = \bigcurlywedge_{a\in\mathcal{A}} (\neg a \parr \iota(t[x := a]^{\mathcal{A}^{\to}}))$$

Both terms are equal since $[\![t[x := a]]\!]^{\mathcal{A}^{\parr}} = \iota(t[x := a])^{\mathcal{A}^{\to}})$ by induction hypothesis.

- **Case** $u\,v$.

On the one hand, we have $[\![u\,v]\!] = \mu(\alpha).\langle[\![u]\!]\|([[\![v]\!]],\alpha)\rangle$ and

$$\begin{aligned}(\mu(\alpha).\langle[\![u]\!]\|([[\![v]\!]],\alpha)\rangle)^{\mathcal{A}^{\parr}} &= \bigcurlywedge_{a\in\mathcal{A}} \{a : ([\![u]\!]^{\mathcal{A}^{\parr}}, (\neg[\![v]\!]^{\mathcal{A}^{\parr}} \parr a)) \in \perp\!\!\!\perp\} \\ &= \bigcurlywedge_{a\in\mathcal{A}} \{a : [\![u]\!]^{\mathcal{A}^{\parr}} \preccurlyeq (\neg[\![v]\!]^{\mathcal{A}^{\parr}} \parr a)\}\end{aligned}$$

On the other hand,

$$\iota([u\,v]^{\mathcal{A}^{\to}}) = \iota(\bigcurlywedge_{a\in\mathcal{A}} \{a : (u^{\mathcal{A}^{\to}}) \preccurlyeq (v^{\mathcal{A}^{\to}}) \xrightarrow{\parr} a\}) = \bigcurlywedge_{a\in\mathcal{A}} \{a : \iota(u^{\mathcal{A}^{\to}}) \preccurlyeq \neg(\iota(v^{\mathcal{A}^{\to}}) \parr a\}))$$

Both terms are equal since $[\![u]\!]^{\mathcal{A}^{\parr}} = \iota(u^{\mathcal{A}^{\to}})$ and $[\![v]\!]^{\mathcal{A}^{\parr}} = \iota(v^{\mathcal{A}^{\to}})$ by induction hypotheses. □

11.4 Disjunctive algebras

11.4.1 Separation in disjunctive structures

We shall now introduce the notion of disjunctive separator. To this purpose, we adapt the definition of implicative separators, using Bourbaki's axioms for the disjunction and the negation instead of Hilbert's combinators **s** and **k**. We recall these axioms, which are taken from [21, p.25], to which we added the fifth one:

$$\begin{array}{lcl} S1 & : & (A \lor A) \to A \\ S2 & : & A \to (A \lor B) \\ S3 & : & (A \lor B) \to (B \lor A) \\ S4 & : & (A \to B) \to ((C \lor A) \to (C \lor B)) \\ S5 & : & (A \lor (B \lor C)) \to ((A \lor B) \lor C) \end{array}$$

Remark 11.38 (About S5)**.** The last axiom will mostly be used to swap the premises of an arrow from $A \to B \to C$ to $B \to A \to C$. In his book, Bourbaki does not need such an operation since he is interested in the provability of such an arrow, for which he can introduce A and B as hypotheses and try to prove C using these hypotheses in an arbitrary order. Therefore, the order of the premises is somehow irrelevant in his approach. On the opposite, we shall now contemplate the notion of separation (just like in the previous chapter). Typically, we will have to determine whether an element $a \to b$ belongs to a given separator, which is different from determining if b belongs to the separator knowing that a is in it. In this sense, we are facing a situation which is different from Bourbaki's setting.

Besides, viewed as a combinator, the fifth axiom is clearly independent from the others: it is the only one that allows us to decompose the operand of a disjunction as a disjunction itself (S1-S4 only consider premises/conclusions of the form A, $A \lor B$ or $(\neg A) \lor B$). Even though this informal argument could appear as not enough convincing, we believe that the question of knowing whether S5 is an axiom properly speaking is not of big interest here. If it is, then there is no point in considering the stronger notion of (non-associative) disjunctive algebra since all the realizability algebras are associative. If it is not, this simply means that there is a way to compile the corresponding combinator thanks to the first four, just like **i** can be retrieved by **skk** in implicative algebras. ⌟

Let $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ be a fixed disjunctive structure. We thus define the combinators that canonically correspond to the previous axioms:

$$\begin{array}{rcl}
s_1^{\parr} & \triangleq & \curlywedge_{a\in\mathcal{A}} [(a \parr a) \to a] \\
s_2^{\parr} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} [a \to (a \parr b)] \\
s_3^{\parr} & \triangleq & \curlywedge_{a,b\in\mathcal{A}} [(a \parr b) \to b \parr a] \\
s_4^{\parr} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} [(a \to b) \to (c \parr a) \to (c \parr b)] \\
s_5^{\parr} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}} [(a \parr (b \parr c)) \to ((a \parr b) \parr c)]
\end{array}$$

Separators for $\mathcal{A}$ are defined similarly to the separators for implicative structures, replacing the combinators $\mathbf{K}, \mathbf{S}$ and $\mathbf{cc}$ by the previous ones.

Definition• 11.39 (Separator)**.** We call *separator* for the disjunctive structure $\mathcal{A}$ any subset $\mathcal{S} \subseteq \mathcal{A}$ that fulfills the following conditions for all $a, b \in \mathcal{A}$:

(1) If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$ (upward closure)

(2) s_1, s_2, s_3, s_4 and s_5 are in $\mathcal{S}$ (combinators)

(3) If $a \to b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$ (closure under modus ponens)

A separator $\mathcal{S}$ is said to be *consistent* if $\perp \notin \mathcal{S}$. ⌟

Remark• 11.40 (Alternative definition)**.** As for implicative structures (Remark 10.29), in presence of condition (1), condition (3) is equivalent to the following condition:

(3') If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $ab \in \mathcal{S}$ (closure under application)

The proof is exactly the same:

- (3) ⇒(3'): If $a \in \mathcal{S}$ and $b \in \mathcal{S}$, since $a \preccurlyeq b \to ab$ (Section 11.3.1) by upward closure we have $b \to ab \in \mathcal{S}$, and thus $ab \in \mathcal{S}$ by modus ponens .
- (3')⇒(3): If $a \in \mathcal{S}$ and $a \to b \in \mathcal{S}$, then $(a \to b)a \in \mathcal{S}$ by closure under application. Since $(a \to b)a \preccurlyeq b$ (Section 11.3.1) by upward closure we conclude that $b \in \mathcal{S}$.

⌟

Definition• 11.41 (Disjunctive algebra)**.** We call *disjunctive algebra* the given of a disjunctive structure $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ together with a separator $\mathcal{S} \subseteq \mathcal{A}$. A disjunctive algebra is said to be consistent if its separator is. ⌟

Remark 11.42. The reader may notice that in this chapter, we do not distinguish between classical and intuitionistic separators. Indeed, $L^{\parr}$ and the corresponding fragment of the sequent calculus are intrinsically classical. As we shall see thereafter, so are the disjunctive algebras: the negation is always involutive modulo the equivalence $\cong_{\mathcal{S}}$ (Proposition 11.58). ⌟

Example• 11.43 (Complete Boolean algebras)**.** Once again, if $\mathcal{B}$ is a complete Boolean algebra, $\mathcal{B}$ induces a disjunctive structure in which it is easy to verify that the combinators $s_1^{\parr}, s_3^{\parr}, s_3^{\parr}, s_4^{\parr}$ and $s_5^{\parr}$ are equal to the maximal element $\top$. Therefore, the singleton $\{\top\}$ is a valid separator for the induced disjunctive structure and any non-degenerated complete Boolean algebras thus induces a consistent disjunctive algebra. In fact, the filters for $\mathcal{B}$ are exactly its separators. ⌟

11.4.2 Disjunctive algebra from classical realizability

Recall that any model of classical realizability based on the $L^{⅋}$-calculus induces a disjunctive structure, where:

$$\begin{array}{lll} \bullet\ \mathcal{A} \triangleq \mathcal{P}(\mathcal{V}_0) & \bullet\ a ⅋ b \triangleq (a,b) = \{(e_1,e_2) : e_1 \in a \wedge e_2 \in b\} & \\ \bullet\ a \preccurlyeq b \triangleq a \supseteq b & \bullet\ \neg a \triangleq [a^{\perp\!\!\!\perp}] = \{[v] : v \in a^{\perp\!\!\!\perp}\} & (\forall a,b \in \mathcal{A}) \end{array}$$

As in the implicative case, the set of formulas realized by a closed term[5], that is to say:

$$\mathcal{S}_{\perp\!\!\!\perp} \triangleq \{a \in \mathcal{P}(V_0^+) : a^{\perp\!\!\!\perp} \cap \mathcal{T}_0 \neq \emptyset\}$$

defines a valid separator. The conditions (1) and (3) are clearly verified (for the same reasons as in the implicative case), but we should verify that the formulas corresponding to the combinators are indeed realized.

Let us then consider the following closed terms:

$$\begin{array}{lcl} PS_1 & \triangleq & \mu([x],\alpha).\langle x \| (\alpha,\alpha)\rangle \\ PS_2 & \triangleq & \mu([x],\alpha).\langle \mu(\alpha_1,\alpha_2).\langle x\|\alpha_1\rangle \| \alpha\rangle \\ PS_3 & \triangleq & \mu([x],\alpha).\langle \mu(\alpha_1,\alpha_2).\langle x\|(\alpha_2,\alpha_1)\rangle \| \alpha\rangle \\ PS_4 & \triangleq & \mu([x],\alpha).\langle \mu([y],\beta).\langle \mu(\gamma,\delta).\langle y\|(\gamma,\mu z.\langle x\|([z],\delta)\rangle)\rangle \|\beta\rangle\|\alpha\rangle \\ PS_5 & \triangleq & \mu([x],\alpha).\langle \mu(\beta,\alpha_3).\langle \mu(\alpha_1,\alpha_2).\langle x\|(\alpha_1,(\alpha_2,\alpha_3))\rangle \|\beta\rangle\|\alpha\rangle \end{array}$$

Proposition 11.44. *The previous terms have the following types in $L^{⅋}$:*

1. $\vdash PS_1 : \forall A.(A ⅋ A) \to A \mid$
2. $\vdash PS_2 : \forall AB.A \to A ⅋ B \mid$
3. $\vdash PS_3 : \forall AB.A ⅋ B \to B ⅋ A \mid$
4. $\vdash PS_4 : \forall ABC.(A \to B) \to (C ⅋ A \to C ⅋ B) \mid$
5. $\vdash PS_5 : \forall ABC.(A ⅋ (B ⅋ C)) \to ((A ⅋ B) ⅋ C) \mid$

Proof. Straightforward typing derivations in $L^{⅋}$. □

We deduce that $\mathcal{S}_{\perp\!\!\!\perp}$ is a valid separator:

Proposition 11.45. *The quintuple $(\mathcal{P}(\mathcal{V}_0), \preccurlyeq, ⅋, \neg, \mathcal{S}_{\perp\!\!\!\perp})$ as defined above is a disjunctive algebra.*

Proof. Conditions (1) and (3) are trivial. Condition (2) follows from the previous proposition and the adequacy lemma for the realizability interpretation of $L^{⅋}$ (Proposition 11.10). □

11.4.3 About the combinators

The interpretations of the terms PS_1, PS_2, PS_3 and PS_5 are equal to their principal types.

Proposition• 11.46. *We have:*

$$(PS_1)^{\mathcal{A}} = \bigcurlywedge_{a \in \mathcal{A}} ((a ⅋ a) \to a)$$

[5] As in the $\lambda\mu\tilde{\mu}$-calculus (see Section 4.4.5), proof-like terms in $L^{⅋}$ simply correspond to closed terms.

Proof. By definition, we have:

$$(PS_1)^{\mathcal{A}} = (\mu([x],\alpha).\langle x\|(\alpha,\alpha)\rangle)^{\mathcal{A}} = \curlywedge_{\alpha,x\in\mathcal{A}} \{x \to \alpha : x \preccurlyeq (\alpha \parr \alpha)\}$$

Let α, x be elements of $\mathcal{A}$ such that $x \preccurlyeq \alpha \parr \alpha$. Then by covariance of the arrow and definition of the meet, we deduce that:

$$\curlywedge_{a\in\mathcal{A}} \{(a \parr a) \to a\} \preccurlyeq (\alpha \parr \alpha) \to \alpha \preccurlyeq x \to \alpha$$

and this being true for any $\alpha, x \in \mathcal{A}$, we obtain:

$$\curlywedge_{a\in\mathcal{A}} \{(a \parr a) \to a\} \preccurlyeq \curlywedge_{\alpha,x\in\mathcal{A}} \{x \to \alpha : x \preccurlyeq (\alpha \parr \alpha)\} = (PS_1)^{\mathcal{A}}$$

The converse inequality can be proved the same way, or can be directly deduced using Proposition 12.29 and the adequacy $\mathrm{L}^{\parr}$ typing rules (Proposition 11.34). □

Proposition• 11.47. *We have:*

$$(PS_2)^{\mathcal{A}} = \curlywedge_{a,b\in\mathcal{A}} (a \to a \parr b)$$

Proof. By definition, we have:

$$(PS_2)^{\mathcal{A}} = (\mu([x],\alpha).\langle \mu(\alpha_1,\alpha_2).\langle x\|\alpha_1\rangle\|\alpha\rangle)^{\mathcal{A}} = \curlywedge_{\alpha,x\in\mathcal{A}} \{x \to \alpha : \curlywedge_{\alpha_1,\alpha_2\in\mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_1\} \preccurlyeq \alpha\}$$

Using the distributivity of meets over the disjunction, one observe that for any fixed a:

$$\curlywedge_{\alpha_1,\alpha_2\in\mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_1\} = \Big(\curlywedge_{\alpha_1\in\mathcal{A}} \{\alpha_1 : x \preccurlyeq \alpha_1\}\Big) \parr \Big(\curlywedge_{\alpha_2\in\mathcal{A}} \{\alpha_2\}\Big) = x \parr \bot$$

Therefore, we can directly prove that:

$$(PS_2)^{\mathcal{A}} = \curlywedge_{\alpha,x\in\mathcal{A}} \{x \to \alpha : x \parr \bot \preccurlyeq \alpha\} = \curlywedge_{\alpha,x\in\mathcal{A}} \{x \to x \parr \bot\} = \curlywedge_{a,b\in\mathcal{A}} \{a \to (a \parr b)\}$$

□

Proposition• 11.48. *We have:*

$$(PS_3)^{\mathcal{A}} = \curlywedge_{a,b\in\mathcal{A}} (a \parr b \to b \parr a)$$

Proof. We want to prove the inequality from right to left, the other one being a consequence of semantic typing. By definition, we have:

$$(PS_3)^{\mathcal{A}} = (\mu([x],\alpha).\langle \mu(\alpha_1,\alpha_2).\langle x\|(\alpha_2,\alpha_1)\rangle\|\alpha\rangle)^{\mathcal{A}} = \curlywedge_{\alpha,x\in\mathcal{A}} \Big\{x \to \alpha : \curlywedge_{\alpha_1,\alpha_2\in\mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\} \preccurlyeq \alpha\Big\}$$

Let α, x be elements of $\mathcal{A}$ such that $\curlywedge_{\alpha_1,\alpha_2\in\mathcal{A}}\{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\} \preccurlyeq \alpha$. Using the variance of the arrow we obtain that:

$$x \to \curlywedge_{\alpha_1,\alpha_2\in\mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\} \preccurlyeq x \to \alpha$$

Using the commutation of meet and par, we have:

$$x \to \bigcurlywedge_{\alpha_1,\alpha_2 \in \mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\} = \bigcurlywedge_{\alpha_1,\alpha_2 \in \mathcal{A}} \{x \to \alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\}$$

Let then α_1, α_2 be elements of $\mathcal{A}$ such that $x \preccurlyeq \alpha_2 \parr \alpha_1$, using the variance of the arrow, we deduce that:

$$\bigcurlywedge_{a,b \in \mathcal{A}} (a \parr b \to b \parr a) \preccurlyeq \alpha_1 \parr \alpha_2 \to \alpha_2 \parr \alpha_1 \preccurlyeq x \to \alpha_2 \parr \alpha_1$$

Recollecting the pieces, we deduce (by introduction of the meet over α_1, α_2) that:

$$\bigcurlywedge_{a,b \in \mathcal{A}} (a \parr b \to b \parr a) \preccurlyeq x \to \bigcurlywedge_{\alpha_1,\alpha_2 \in \mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\} \preccurlyeq x \to \alpha$$

and finally (by introduction of the meet over α, x) that:

$$\bigcurlywedge_{a,b \in \mathcal{A}} (a \parr b \to b \parr a) \preccurlyeq \bigcurlywedge_{\alpha,x \in \mathcal{A}} \left\{x \to \alpha : \bigcurlywedge_{\alpha_1,\alpha_2 \in \mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_2 \parr \alpha_1\} \preccurlyeq \alpha\right\} = (PS_3)^{\mathcal{A}}$$

□

Proposition• 11.49. *We have:*

$$(PS_5)^{\mathcal{A}} = \bigcurlywedge_{a,b,c \in \mathcal{A}} ((a \parr (b \parr c)) \to ((a \parr b) \parr c)$$

Proof. Once more, we only want to prove the inequality from right to left, the other one being a consequence of semantic typing. By definition, we have:

$$\begin{aligned}
(PS_5)^{\mathcal{A}} &= (\mu([x],\alpha).\langle \mu(\beta,\alpha_3).\langle \mu(\alpha_1,\alpha_2).\langle x \| (\alpha_1,(\alpha_2,\alpha_3))\rangle \| \beta\rangle \| \alpha\rangle)^{\mathcal{A}} \\
&= \bigcurlywedge_{\alpha,x \in \mathcal{A}} \left\{x \to \alpha : \bigcurlywedge_{\beta,\alpha_3 \in \mathcal{A}} \left\{\beta \parr \alpha_3 : \bigcurlywedge_{\alpha_1,\alpha_2 \in \mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_1 \parr (\alpha_2 \parr \alpha_3)\} \preccurlyeq \beta\right\} \preccurlyeq \alpha\right\} \\
&= \bigcurlywedge_{x,\beta,\alpha_3 \in \mathcal{A}} \left\{x \to (\beta \parr \alpha_3) : \bigcurlywedge_{\alpha_1,\alpha_2 \in \mathcal{A}} \{\alpha_1 \parr \alpha_2 : x \preccurlyeq \alpha_1 \parr (\alpha_2 \parr \alpha_3)\} \preccurlyeq \beta\right\} \\
&= \bigcurlywedge_{x,\alpha_3,\alpha_1,\alpha_2 \in \mathcal{A}} \{x \to (\alpha_1 \parr \alpha_2) \parr \alpha_3 : x \preccurlyeq \alpha_1 \parr (\alpha_2 \parr \alpha_3)\}
\end{aligned}$$

Let $x, \alpha_3, \alpha_1, \alpha_2$ be elements of $\mathcal{A}$ such that $x \preccurlyeq \alpha_1 \parr (\alpha_2 \parr \alpha_3)$. Using the covariance of the arrow on the left, and by definition of meets, we get that:

$$\bigcurlywedge_{a,b,c \in \mathcal{A}} ((a \parr (b \parr c)) \to ((a \parr b) \parr c) \preccurlyeq \alpha_1 \parr (\alpha_2 \parr \alpha_3) \to (\alpha_1 \parr \alpha_2) \parr \alpha_3 \preccurlyeq x \to (\alpha_1 \parr \alpha_2) \parr \alpha_3$$

Thus, we can conclude (by introduction of the meet over $x, \alpha_3, \alpha_2, \alpha_1$) that:

$$\bigcurlywedge_{a,b,c \in \mathcal{A}} ((a \parr (b \parr c)) \to ((a \parr b) \parr c) \preccurlyeq \bigcurlywedge_{x,\alpha_3,\alpha_1,\alpha_2 \in \mathcal{A}} \{x \to (\alpha_1 \parr \alpha_2) \parr \alpha_3 : x \preccurlyeq \alpha_1 \parr (\alpha_2 \parr \alpha_3)\} = (PS_5)^{\mathcal{A}}$$

□

Remark• 11.50. Before turning to the study of the internal logic of disjunctive algebras, we should say a word on the missing equality for PS_4 and $s_4^{\parr}$. In contrast with the other four $L^{\parr}$ terms, PS_4 makes use of a context $\mu x.c$. Through the embedding, this binder is translated into a join and we get:

$$PS_4^{\mathcal{A}} = \bigcurlywedge_{x,\alpha \in \mathcal{A}} \{x \to \alpha : \bigcurlywedge_{y,\beta \in \mathcal{A}} \{y \to \beta : \bigcurlywedge_{\gamma,\delta \in \mathcal{A}} \{\gamma \parr \delta : y \preccurlyeq \gamma \parr \bigcurlyvee_{z \in \mathcal{A}} \{z : x \preccurlyeq z \to \delta\}\} \preccurlyeq \beta\} \preccurlyeq \alpha\}$$

By manipulation of the meets with their commutation, we can reduce it to:

$$PS_4^{\mathcal{A}} = \bigwedge_{x,b,c\mathcal{A}} \{x \to (c \parr \bigvee_{z\in\mathcal{A}} \{z : x \preccurlyeq z \to b\}\} \to (c \parr b)\}$$

Nonetheless, this is *a priori* the best we can do, in the absence of commutation law for the join. In particular, there is no way to prove that $s_4^{\parr} = \bigwedge_{a,b,c\in\mathcal{A}}((a \to b) \to (c \parr a) \to (c \parr b))$ is lower than this term, given a fixed x, there is no way to find two elements a and b such that $x \preccurlyeq a \to b$. Of course, if the disjunctive algebra has extra commutations (of joins with the negation and the disjunction), the equality holds, but in this case the disjunctive algebra is in fact a Boolean algebra. ⌟

11.4.4 Internal logic

11.4.4.1 Entailment

As in the case of implicative algebras, we define a relation of entailment:

Definition• 11.51 (Entailment). For all $a,b \in \mathcal{A}$, we say that a *entails* b and write $a \vdash_{\mathcal{S}} b$ if $a \to b \in \mathcal{S}$. We say that a and b are equivalent and write $a \cong_{\mathcal{S}} b$ if $a \vdash_{\mathcal{S}} b$ and $b \vdash_{\mathcal{S}} a$. ⌟

From the combinators, we directly get that:

Proposition 11.52 (Combinators). *For all $a,b,c \in \mathcal{A}$, the following holds:*

1.• $(a \parr a) \vdash a$
2.• $a \vdash (a \parr b)$
3.• $(a \parr b) \vdash (b \parr a)$
4.• $(a \to b) \vdash (c \parr a) \to (c \parr b)$
5.• $a \parr (b \parr c) \vdash (a \parr b) \parr c$

Proposition 11.53 (Preorder). *For any $a,b,c \in \mathcal{A}$, the following holds:*

1.• $a \vdash_{\mathcal{S}} a$ *(Reflexivity)*
2.• *if $a \vdash_{\mathcal{S}} b$ and $b \vdash_{\mathcal{S}} c$ then $a \vdash_{\mathcal{S}} c$* *(Transitivity)*

Proof. We first that (2) holds by applying twice the closure by modus ponens, then we use it with the relation $a \vdash_{\mathcal{S}} a \parr a$ and $a \parr a \vdash_{\mathcal{S}}$ proven above to get 1. □

We could pursue our investigation about the properties of the entailment relation as we did in implicative algebras. Unfortunately, in comparison with the implicative setting, we are lacking of a powerful proof tool. Indeed, remember that for implicative algebras, we were able to compute directly with truth values, mainly thanks to the fact that any separator contains all closed λ-terms. This statement was proven using the combinatorial completeness of the separators **K** and **S** with respect to the λ-calculus. Here, we are in a situation drastically different: first of all, we do not have any clue about a potential completeness of $PS_1,\ldots,PS_5$ with respect to $\mathrm{L}^{\parr}$. And even if we were having such a result, since $PS_4^{\mathcal{A}}$ is not equal to $s_4^{\parr}$, we still could not use it to prove that every closed $\mathrm{L}^{\parr}$ term is in the separator.

In a nutshell, we are in a situation where we have to do realizability with only a finite set of realizers, and the possibility of examining the structure of falsity values case by case. In particular, most of the proof we present thereafter rely on technical lemmas requiring tedious and boring proofs. We shall skip some details, taking advantage of our formalization which should help the reader to convince himself that we are not hiding difficulties under the carpet. The key lemma in this situation is the closure of the separator under application (condition (3')). Indeed, it allows us to prove the following technical lemmas, which are generalized forms of modus ponens and transitivity, compatible with meets:

Lemma• 11.54 (Generalized modus ponens)**.** *For all subsets $A, B \subseteq \mathcal{A}$, if $\curlywedge_{a\in A, b\in B}(a \to b) \in \mathcal{S}$ and $(\curlywedge_{a\in A} a) \in \mathcal{S}$, then $(\curlywedge_{b:B} b) \in \mathcal{S}$.*

Proof. Let $A, B \subseteq A$ be two subsets of $\mathcal{A}$ such that $t_{ab} \triangleq \curlywedge_{a\in A, b\in B}(a \to b) \in \mathcal{S}$ and $t_a \triangleq (\curlywedge_{a\in A} a) \in \mathcal{S}$. Then by closure under application, we have $t_{ab} t_a \in \mathcal{S}$. Using the upward closure, it only remains to prove that:

$$t_{ab} t_a \preccurlyeq (\bigcurlywedge_{b:B} b)$$

which is an easy manipulation of meets using the adjunction. □

Lemma• 11.55 (Generalized transitivity)**.** *For any subsets $A, B, C \subseteq \mathcal{A}$, if $\curlywedge_{a\in A, b\in B}(a \to b) \in \mathcal{S}$ and $\curlywedge_{b\in B, c\in C}(b \to c) \in \mathcal{S}$, then $\curlywedge_{a\in A, c\in C}(a \to c) \in \mathcal{S}$.*

Proof. Let $A, B, C \subseteq \mathcal{A}$ be some fixed sets, such that $t_{ab} \triangleq \curlywedge_{a\in A, b\in B}(a \to b) \in \mathcal{S}$ and $t_{bc} = \curlywedge_{b\in B, c\in C}(b \to c) \in \mathcal{S}$. Then we have $s_4^{\parr}\, t_{ab}\, t_{bc} \in \mathcal{S}$, and it suffices to show that

$$s_4^{\parr}\, t_{bc}\, t_{ab} = \left(\bigcurlywedge_{a,b,c\in\mathcal{A}} (a \to b) \to (c \parr a) \to (c \parr b)\right) t_{bc}\, t_{ab} \preccurlyeq \bigcurlywedge_{a\in A, c\in C} (a \to c)$$

This is proved again by a straightforward manipulation of the meets using the adjunction. □

As a corollary, we can for instance use the previous lemma to show that:

Proposition• 11.56 (I)**.** *We have $\boldsymbol{I}^{\mathcal{A}} = \curlywedge_{a\in A}(a \to a) \ \in \mathcal{S}$.*

Proof. Simple application of Lemma 11.55 to compose $s_2^{\parr}$ and $s_1^{\parr}$. □

11.4.4.2 Negation

We can relate the primitive negation to the one induced by the underlying implicative structure:

Proposition 11.57 (Implicative negation)**.** *For all $a \in \mathcal{A}$, the following holds:*

1.• $\neg a \vdash_{\mathcal{S}} a \to \bot$ *2.•* $a \to \bot \vdash_{\mathcal{S}} \neg a$

Proof. We prove in both cases a slightly more general statement, namely that the meet over all a, b or the corresponding implication belongs to the separator. The first item follows directly from the fact that $s_2^{\parr}$ belongs to the separator, since $\curlywedge_{a\in\mathcal{A}}(\neg a) \to (a \to \bot) = \curlywedge_{a\in\mathcal{A}}(\neg a) \to (\neg a \parr \bot)$.

For the second item, the first step is to apply Lemma 11.55 with the following hypotheses:

$$\bigcurlywedge_{a\in\mathcal{A}} ((a \to \bot) \to a \to \neg a) \in \mathcal{S} \qquad\qquad \bigcurlywedge_{a\in\mathcal{A}} (a \to \neg a) \to \neg a \in \mathcal{S}$$

The statement on the left hand-side is proved by subtyping from the identity. On the right hand-side, we use twice Lemma 11.54 to prove that:

$$\bigcurlywedge_{a\in\mathcal{A}} (a \to a) \to (\neg a \to \neg a) \to (a \to \neg a) \to \neg a \ \in \mathcal{S}$$

The two extra hypotheses are trivially subtypes of the identity again. This statement follows from this more general property (recall that $a \to a = \neg a \parr a$):

$$\bigcurlywedge_{a,b\in\mathcal{A}} (a \parr b) \to a + b$$

that we shall prove thereafter (see Proposition 11.59). □

Additionally, we can show that the principle of double negation elimination is valid with respect to any separator:

Proposition 11.58 (Double negation)**.** *For all $a \in \mathcal{A}$, the following holds:*

1.$^\bullet$ $a \vdash_{\mathcal{S}} \neg\neg a$ *2.*$^\bullet$ $\neg\neg a \vdash_{\mathcal{S}} a$

Proof. The first item is easy since for all $a \in \mathcal{A}$, we have $a \to \neg\neg a = (\neg a) \parr \neg\neg a \cong_{\mathcal{S}} \neg\neg a \parr \neg a = \neg a \to \neg a$. As for the second item, we use Lemma 11.55 and Proposition 11.57 to it reduce to the statement:

$$\bigcurlywedge_{a \in \mathcal{A}} ((\neg a) \to \bot) \to a \quad \in \mathcal{S}$$

We use again Lemma 11.55 to prove it, by showing that:

$$\bigcurlywedge_{a \in \mathcal{A}} ((\neg a) \to \bot) \to (\neg a) \to a \quad \in \mathcal{S} \qquad\qquad \bigcurlywedge_{a \in \mathcal{A}} ((\neg a) \to a) \to (\neg a) \to a \quad \in \mathcal{S}$$

where the statement on the left hand-side from by subtyping from the identity. For the one on the right hand-side, we use the same trick as in the last proof in order to reduce it to:

$$\bigcurlywedge_{a \in \mathcal{A}} (a \to \neg a) \to (a \to a) \to (\neg a \to a) \to a) \ \in \mathcal{S}$$

□

11.4.4.3 Sum type

As in implicative structures, we can define the sum type by:

$$a + b \triangleq \bigcurlywedge_{c \in \mathcal{A}} ((a \to c) \to (b \to c) \to c) \qquad (\forall a, b \in \mathcal{A})$$

We can prove that the disjunction and this sum type are equivalent from the point of view of the separator:

Proposition 11.59 (Implicative sum type)**.** *For all $a, b \in \mathcal{A}$, the following holds:*

1.$^\bullet$ $a \parr b \vdash_{\mathcal{S}} a + b$ *2.*$^\bullet$ $a + b \vdash_{\mathcal{S}} a \parr b$

Proof. We prove in both cases a slightly more general statement, namely that the meet over all a, b or the corresponding implication belongs to the separator. For the first item, we have:

$$\bigcurlywedge_{a,b \in \mathcal{A}} (a \parr b) \to a + b = \bigcurlywedge_{a,b,c \in \mathcal{A}} (a \parr b) \to (a \to c) \to (b \to c) \to c$$

Swapping the order of the arguments, we prove that $\bigcurlywedge_{a,b,c \in \mathcal{A}} (b \to c) \to (a \parr b) \to (a \to c) \to c \in \mathcal{S}$. For this, we use Lemma 11.55 and the fact that:

$$\bigcurlywedge_{a,b,c \in \mathcal{A}} (b \to c) \to (a \parr b) \to (a \parr c) \ \in \mathcal{S} \qquad\qquad \bigcurlywedge_{a,c \in \mathcal{A}} (a \parr c) \to (a \to c) \to c \ \in \mathcal{S}$$

The left hand-side statement is proved using $s_4^{\parr}$, while on the right hand-side we prove it from the fact that:

$$\bigcurlywedge_{a,c \in \mathcal{A}} (a \to c) \to (a \parr c) \to c \parr c \ \in \mathcal{S}$$

which is a subtype of $s_4^{\parr}$, by using Lemma 11.55 again with $s_1^{\parr}$ and by manipulation on the order of the argument.

The second item is easier to prove, using Lemma 11.55 again and the fact that:

$$\bigcurlywedge_{a,b\in\mathcal{A}} a+b \to (a \to (a \parr b)) \to (b \to (a \parr b)) \to (a \parr b) \in \mathcal{S}$$

which is a subtype of $\mathbf{I}^{\mathcal{A}}$ (which belongs to $\mathcal{S}$). The other part, which is to prove that:

$$\bigcurlywedge_{a,b\in\mathcal{A}} ((a \to (a \parr b)) \to (b \to (a \parr b)) \to (a \parr b)) \to (a \parr b) \quad \in \mathcal{S}$$

follows from Lemma 11.54 and the fact that $\bigcurlywedge_{a,b\in\mathcal{A}}(a \to (a \parr b))$ and $\bigcurlywedge_{a,b\in\mathcal{A}}(b \to (a \parr b))$ are both in the separator.

□

11.4.5 Induced implicative algebras

We shall now prove that the combinators defining implicative separators also belong to any disjunctive separator. Since conditions (1) and (3) of disjunctive and implicative separators are equal, this will in particular prove that any disjunctive algebra is a particular case of implicative algebra.

Proposition• 11.60 (Combinator $\mathbf{K}^{\mathcal{A}}$). *For any disjunctive algebra $(\mathcal{A}, \preccurlyeq, \parr, \neg, \mathcal{S})$, we have $\mathbf{K}^{\mathcal{A}} \in \mathcal{S}$.*

Proof. This directly follows by upwards closure from the fact that $\bigcurlywedge_{a,b\in\mathcal{A}} a \to (b \parr a) \in \mathcal{S}$. □

Proposition• 11.61 (Combinator $\mathbf{s}^{\mathcal{A}}$). *For any disjunctive algebra $(\mathcal{A}, \preccurlyeq, \parr, \neg, \mathcal{S})$, we have $\mathbf{s}^{\mathcal{A}} \in \mathcal{S}$.*

Proof. We make several applications of Lemmas 11.55 and 11.54 consecutively. We prove that:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \;\in \mathcal{S}$$

is implied by Lemma 11.55 and:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} ((a \to b \to c) \to (b \to a \to c)) \in \mathcal{S} \qquad \text{and} \qquad \bigcurlywedge_{a,b,c\in\mathcal{A}} ((b \to a \to c) \to (a \to b) \to a \to c) \in \mathcal{S}$$

The statement on the left hand-side is an ad-hoc lemma, while the other is proved by generalized transitivity (Lemma 11.54), using a subtype of $s_4^{\parr}$ as hypothesis, from:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} ((a \to b) \to (a \to a \to c)) \to (a \to b) \to a \to c \in \mathcal{S}$$

The latter is proved, using again generalized transitivity with a subtype of $s_4^{\parr}$ as premise, from:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} (a \to a \to c) \to (a \to c) \in \mathcal{S}$$

This is proved using again Lemmas 11.55 and 11.54 with $s_5^{\parr}$ and a variant of $s_4^{\parr}$. □

Proposition• 11.62 (Combinator $\mathbf{cc}^{\mathcal{A}}$). *For any disjunctive algebra $(\mathcal{A}, \preccurlyeq, \parr, \neg, \mathcal{S})$, we have $\mathbf{cc}^{\mathcal{A}} \in \mathcal{S}$.*

Proof. We make several applications of Lemmas 11.55 and 11.54 consecutively. We prove that:

$$\bigwedge_{a,b\in\mathcal{A}} ((a \to b) \to a) \to a \ \in \mathcal{S}$$

is implied by generalized modus ponens (Lemma 11.55) and:

$$\begin{array}{ll} & \bigwedge_{a,b\in\mathcal{A}}((a \to b) \to a) \to (\neg a \to a \to b) \to \neg a \to a \ \in \mathcal{S} \\ \text{and} & \bigwedge_{a,b\in\mathcal{A}}((\neg a \to a \to b) \to \neg a \to a) \to a \ \in \mathcal{S} \end{array}$$

The statement above is a subtype of $s_4^{⅋}$, while the other is proved, by Lemma 11.55, from:

$$\begin{array}{ll} & \bigwedge_{a,b\in\mathcal{A}}((\neg a \to a \to b) \to \neg a \to a) \to \neg a \to a \ \in \mathcal{S} \\ \text{and} & \bigwedge_{a\in\mathcal{A}}((\neg a) \to a) \to a \ \in \mathcal{S} \end{array}$$

The statement below is proved as in Proposition 11.58, while the statement above is proved by a variant of the modus ponens and:

$$\bigwedge_{a,b\in\mathcal{A}} (\neg a \to a \to b) \ \in \mathcal{S}$$

We conclude by proving this statement using the connections between $\neg a$ and $a \to \bot$, reducing the latter to:

$$\bigwedge_{a,b\in\mathcal{A}} (a \to \bot) \to a \to b \ \in \mathcal{S}$$

which is a subtype of the identity. □

As a consequence, we get the expected theorem:

Theorem• 11.63. *Any disjunctive algebra is a classical implicative algebra.*

Proof. The conditions of upward closure and closure under modus ponens coincide for implicative and disjunctive separators, and the previous propositions show that **K**, **S** and **cc** belong to the separator of any disjunctive algebra. □

Corollary 11.64. *If t is a closed λ-term and $(\mathcal{A}, \preccurlyeq, ⅋, \neg, \mathcal{S})$ a disjunctive algebra, then $t^{\mathcal{A}} \in \mathcal{S}$.*

11.4.6 From implicative to disjunctive algebras

On the converse direction, we could wonder whether it is possible to get a disjunctive algebra from an implicative one. The first step in this direction would be to define a disjunctive structure from an implicative structure, and to this end, the natural candidates for the disjunction and the negation are:

$$a ⅋ b \triangleq a + b \qquad\qquad \neg a \triangleq a \to \bot$$

Indeed, we saw that in the implicative algebra underlying any disjunctive algebra $(\mathcal{A}, \preccurlyeq, ⅋, \neg, \mathcal{S})$, we had the equivalences $a ⅋ b \cong_{\mathcal{S}} a + b$ and $\neg a \cong_{\mathcal{S}} a \to \bot$ (Propositions 11.57 and 11.59).

However, there is no reason for the required laws of commutation:

$$\bigwedge_{b\in B} (a + b) = a + (\bigwedge_{b\in B} b) \qquad \bigwedge_{b\in B} (b + a) = (\bigwedge_{b\in B} b) + a \qquad (\bigwedge_{a\in A} a) \to \bot = \bigvee_{a\in A} (a \to \bot)$$

to hold in an implicative structure. If we focus on the particular case of implicative algebras arising from an abstract Krivine structure (or alternatively in any Krivine realizability model), the equality for the negation holds, but the equalities for the sum type are not true in general. More precisely, they hold in the case were the arrow commutes with the joins, in which case we know that any separator on such a structure will induce a forcing tripos. Nonetheless, in the case where these equalities hold, it is easy to see that any implicative algebra induces a disjunctive algebra since the axioms $s_1^{⅋}, s_2^{⅋}, s_3^{⅋}, s_4^{⅋}, s_5^{⅋}$ are all realized by closed λ-terms. Writing $\neg_\bot a$ for $a \to \bot$, we have:

Proposition 11.65. *If* $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ *is an implicative algebra and* $(\mathcal{A}, \preccurlyeq, +, \neg_\bot)$ *is a disjunctive structure, then* $(\mathcal{A}, \preccurlyeq, +, \neg_\bot, \mathcal{S})$ *is a disjunctive algebra.*

Proof. The conditions of upward closure and closure under modus ponens coincide for implicative and disjunctive separators, and finding realizers for $s_1^{⅋}, s_2^{⅋}, s_3^{⅋}, s_4^{⅋}, s_5^{⅋}$ (with $⅋ = +$) is an easy exercise of λ-calculus. ◻

In other words, implicative algebras which induce disjunctive algebras through[6] $+$ and $\cdot \to \bot$ are particular cases of implicative algebras satisfying extra properties of commutation.

11.5 Conclusion

Since any disjunctive algebra is a particular case of implicative algebra, it is clear that the construction leading to the implicative tripos can be rephrased in this framework. In particular, the same criterion allows us to determine whether the implicative tripos is isomorphic to a forcing tripos. Notably, a disjunctive algebra with extra-commutations for the disjunction $⅋$ and the negation $\neg$ with arbitrary joins will induce an implicative algebra where the arrow commutes with arbitrary joins. Therefore, the induced tripos would collapse to a forcing situation (see Section 10.4.4.2).

Of course, we could reproduce the whole construction (that is studying the product of disjunctive structures, then the quotient by the uniform separator, and verifying the necessary conditions for the functor $\mathcal{T} : I \mapsto \mathcal{A}^I/\mathcal{S}[I]$ to be a tripos) directly in the setting of disjunctive algebras. Nonetheless, insofar as we are interested in the most general framework (and especially in existence of triposes which are not isomorphic to forcing triposes), there is no point in doing this. Indeed, the main conclusion that we draw from this chapter is the following slogan:

Implicative algebras are more general than disjunctive algebras.

In particular, even though we are still missing some properties which would be convenient to be able to use disjunctive algebras in practice, the former slogan dissuades us to pursue in this direction. Nonetheless, we should point out the main feature that is missing in our analysis of disjunctive algebras, namely a computational completeness with respect to $\mathrm{L}^{⅋}$. We obtained in the end that any closed λ-term is in the separator of any disjunctive algebra, which provides us with the possibility of proving that a given element belongs to the separator by finding the adequate realizer. Especially, since we know that the disjunction $a ⅋ b$ is equivalent, with respect to separators, to the sum type $a + b$ (and similarly for the negation $\neg a$ and the implication $a \to \bot$), any formula can be realized by a λ-term for the equivalent formula encoded with $+$ and $\neg_\bot$. However, this is not really convenient in practice and it would be nice to be able to realize formulas directly through $\mathrm{L}^{⅋}$ terms. We do not know if this is possible in the absolute. It would make sense to prove that the combinators $s_1^{⅋}, s_2^{⅋}, s_3^{⅋}, s_4^{⅋}, s_5^{⅋}$ are complete with respect to $\mathrm{L}^{⅋}$ terms, but all our attempts in this direction have shown to be unsuccessful.

[6] Of course, one could still argue that there are maybe better candidates for embedding a negation and a disjunction into implicative structures. Inasmuch as the disjunction and negation that are obtained in the construction of the implicative tripos are $+$ and $\neg_\bot$, we believe this choice to be legitimate.